

# VIDEO SYSTEMS in an IT ENVIRONMENT

The Basics of Networked Media and File-Based Workflows



2<sup>ND</sup> EDITION

Al Kovalick



# **Video Systems in an IT Environment: The Basics of Networked Media and File-Based Workflows**

This page intentionally left blank

# **Video Systems in an IT Environment: The Basics of Networked Media and File-Based Workflows**

**Second Edition**

**Al Kovalick**



AMSTERDAM • BOSTON • HEIDELBERG • LONDON •  
NEW YORK • OXFORD • PARIS • SAN DIEGO •  
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO  
Focal Press is an imprint of Elsevier





Focal Press is an imprint of Elsevier  
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA  
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Copyright © 2009, Elsevier Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: [permissions@elsevier.com](mailto:permissions@elsevier.com). You may also complete your request on-line via the Elsevier homepage (<http://elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

**Library of Congress Cataloging-in-Publication Data**

Application submitted

**British Library Cataloguing-in-Publication Data**

A catalogue record for this book is available from the British Library.

ISBN: 978-0-240-81042-3

For information on all Focal Press publications visit our website at <a href="http://www.elsevierdirect.com">www.elsevierdirect.com</a>
--

09 10 11 12 13 5 4 3 2 1

Printed in the United States of America

Working together to grow libraries in developing countries
---

<a href="http://www.elsevier.com">www.elsevier.com</a>   <a href="http://www.bookaid.org">www.bookaid.org</a>   <a href="http://www.sabre.org">www.sabre.org</a>
--

<b>ELSEVIER</b>	<b>BOOK AID</b> International	<b>Sabre Foundation</b>
-----------------	----------------------------------	-------------------------

# Dedication

This book is dedicated to my parents, Al and Virginia, and to my loving wife, May, who provided constant support and encouragement during the entire project.

# Acknowledgments

This book would never have seen the light of day if it were not for many friends and colleagues who assisted me along the way. Some helped as technical reviewers, some as consultants, some as encouraging voices, and all as solid supporters. I am grateful to all of you. Thanks go to Frans DeJong, Santosh Doss, John Footen, Brad Gilmer, Jacob Gsoedl, Mark Johnston, Greg Lowitz, Bill Moren, Harlan Neugeboren, Charles Poynton, Michel Proulx, John Schmitz, Clyde Smith, and Joanne Tracy.

This page intentionally left blank

# Contents

INTRODUCTION .....	ix
<b>CHAPTER 1</b> Networked Media in an IT Environment.....	1
<b>CHAPTER 2</b> The Basics of Professional Networked Media .....	33
<b>CHAPTER 3A</b> Storage System Basics .....	81
<b>CHAPTER 3B</b> Storage Access Methods .....	121
<b>CHAPTER 4</b> Software Technology for A/V Systems .....	157
<b>CHAPTER 5</b> Reliability and Scalability Methods .....	197
<b>CHAPTER 6</b> Networking Basics for A/V .....	231
<b>CHAPTER 7</b> Media Systems Integration.....	267
<b>CHAPTER 8</b> Security for Networked A/V Systems.....	317
<b>CHAPTER 9</b> Systems Management and Monitoring .....	345
<b>CHAPTER 10</b> The Transition to IT: Issues and Case Studies.....	373
<b>CHAPTER 11</b> A Review of A/V Basics .....	399
<b>Appendix A:</b> Fast Shortcuts for Computing $2^N$ .....	435
<b>Appendix B:</b> Achieving Frame Accuracy in a Non-frame Accurate World .....	437
<b>Appendix C:</b> Grid, Cluster, Utility, and Symmetric Multiprocessing Computing.....	439
<b>Appendix D:</b> The Information Flood—One Zettabyte of Data.....	443
<b>Appendix E:</b> 8B/10B Line Coding.....	445
<b>Appendix F:</b> Digital Hierarchies .....	447
<b>Appendix G:</b> 270 Million—A Magic Number in Digital Video .....	451
<b>Appendix H:</b> A Novel A/V Storage System .....	453
<b>Appendix I:</b> Is It Rabbits Multiplying or Is It Streaming?.....	457
<b>Appendix J:</b> How to Evaluate a Video Server .....	459
<b>Appendix K:</b> Blade Servers .....	463
<b>Appendix L:</b> Solid State Discs Set Off Flash Flood .....	465
<b>Appendix M:</b> Will Ethernet Switches Ever Replace Traditional Video Routers?.....	467
GLOSSARY .....	469
INDEX .....	485

This page intentionally left blank

# Introduction

*There is a tide in the affairs of men, which, taken at the flood, leads on to fortune;  
omitted, all the voyage of their life is bound in shallows and in miseries.*  
—William Shakespeare

Astute sailors know the optimal time to catch the tidal flood toward the harbor. If it is missed, a ship may be caught in a storm or stranded at sea. An able captain and crew never pass up favorable currents. Today there is a different tidal flood that many captains of ship are seeking to ride to safe harbor. What is it? It is the tidal swell of information technology (IT)<sup>1</sup> that is being leveraged to create compelling video systems<sup>2</sup> and file based—"tapeless"—workflows for broadcasters and other professional operations. In the big picture, we are at the emergent stages of video systems designed from hybrid combinations of IT standard platforms (storage, servers, routers, networks, firewalls, middleware, software platforms, Internet, Web services, archives, etc.) and traditional A/V methods and technology.

If you are conversant only in IT methods or comfortable only with traditional video techniques, then the hybrid combination may seem a bit strange and worrying. Will IT methods, systems, and techniques be responsive enough for the demands of real-time video? Can IT meet a 99.9999 percent reliability goal? Can you run video over and through IT-based links and switches? Will network congestion cause dropouts in your video? Will a virus or worm take

---

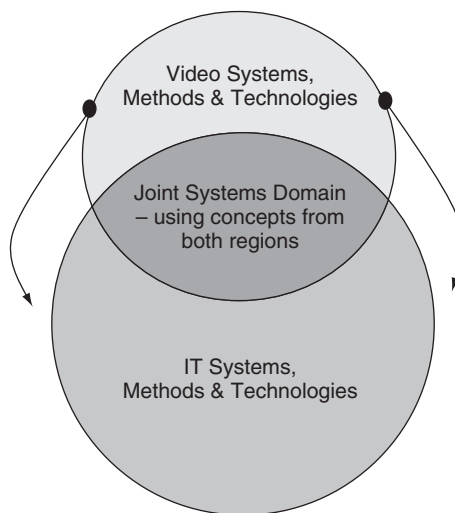
<sup>1</sup>The "IT" term is used throughout this book to refer to the standard platforms, systems, and methods that comprise information technology as used by business process worldwide.

<sup>2</sup>In this book the term "video systems" includes audio systems and still graphics. As a composite, they are denoted by the term "A/V" or "AV" systems. The hybrid acronym AV/IT describes systems that use a combination of IT and traditional A/V technologies.

you off air? Can you upgrade your system while it is in use? Will the short life spans of IT equipment lead to an unprofitable ROI and constant retooling headaches? Is using IT too risky for your demanding operations? Are the software components stable enough for mission-critical applications? Can you use A/V+IT technologies and create a “Broadcast IT” system? These and countless other concerns are discussed and resolved in this book. First, let us look a bit deeper at the interesting cross-section of A/V plus IT.

Figure IN.1 depicts the two domains of interest to us and their all important overlap. As the workflows, methods, and technology of the IT world and those of traditional video mix and combine, compelling new formulations emerge. The IT sphere consists of domain experts plus all the standard infrastructure and systems that make up IT. However, the traditional time-based media sphere consists of domain experts, video-specific links and routers, VTRs, cameras, A/V editors, on-air graphics, effects processors, vision mixers, and much more. The overlap region gathers selected components together from each domain, thereby creating IT-based media workflows.

Which domain has a greater gravitational pull on the other? In 2008, IT equipment alone was estimated to be a \$1.7 trillion worldwide market (source: Forrester Research), while the entire worldwide broadcast equipment market is estimated at \$15 billion (source: DIS Consulting). This is about a 115:1 ratio, and the smaller of the two is drawn to the larger to take advantage of the many levers that IT can provide for video system design. The arrows in the figure imply the gradual consumption of the A/V domain by the IT domain. Almost all new A/V installations, across the board in scale, have large components of IT



**FIGURE IN.1** *The joint systems domain of hybrid A/V+IT systems.*

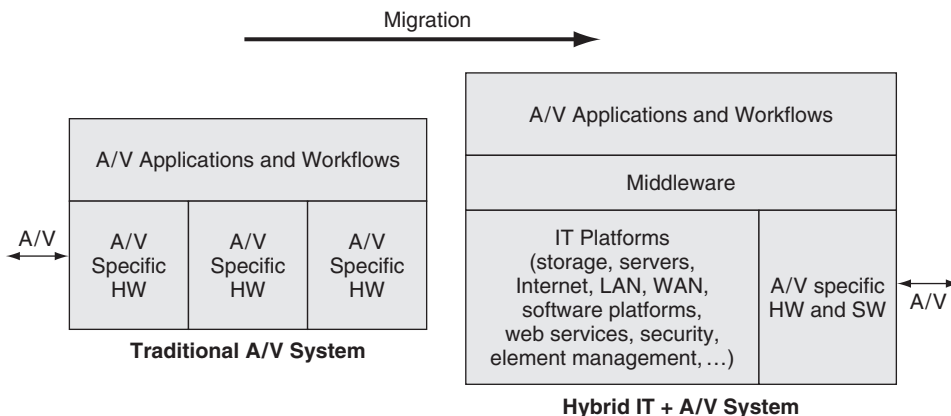
(see Chapter 10) Gone are the days when the exhibitors at NAB or IBC use IT as a differentiator. IT, for the most part, is becoming invisible and woven into the fabric of A/V solutions.

It is the compelling mix of A/V+IT that is our focus. Our approach is judiciously biased toward the understanding of how A/V systems can leverage IT techniques and tools. The chapters that follow cover IT in relation to the workflow needs of video systems. The intention is not to fully describe media technology but rather to explain IT in the light of video systems.

Figure IN.2 illustrates a traditional A/V system on the left and a hybrid A/V+IT system on the right. Traditional is composed of custom A/V components, specialized software, and exotic technologies—usually in small volumes. However, the hybrid mix leverages standard products from information technology and adds A/V-specific elements only as needed by the workflow requirements. Some modern hybrid systems are 90 percent IT and 10 percent A/V in terms of technology. Just a few years ago, the fabric of a typical A/V system was 90 percent A/V specific with just a pinch of IT. Chapter 1 outlines the solid business and technical motivations for the migration to hybrid systems.

## SCOPE OF THE BOOK

Admittedly, the world of video systems spans from the sophisticated workings of a CNN newsroom to a simple home video network. This book's coverage will not boil the ocean. Rather, the concentration is focused on the A/V workflows used by professional broadcast, educational, government, business, and postproduction industries. There are thousands of TV stations and other video facilities worldwide that have not yet made the IT plunge, so this is timely material.



**FIGURE IN.2** The move to IT reduces the amount of custom A/V gear.



Digital A/V finds application in distribution (Web, satellite, digital cable, mobile, digital terrestrial) of content to home, business, and mobile. Home networks are catching fire too. However, our coverage focuses on production processes and not distribution or home networking. Nonetheless, many of the principles covered in these chapters are applicable to any digital A/V network.

The discussions are relevant not only to broadcasters but to media professionals in Fortune 1000 companies, government agencies, small businesses, cable MSOs, production facilities, and movie studios. Event videographers and prosumers are already seeing the gradual invasion of IT into their space.

So who are the target readers for this book?

- IT professionals—Domain experts, system administrators, directors, system engineers, security managers, CIOs, and support staff
- A/V media professionals—Domain experts, chief engineers, VPS of engineering, engineering managers, directors, systems integrators, design engineers, maintenance staff, technicians, facility planners, A/V equipment vendors, A/V sales personnel, and support staff

For media professionals, IT is framed in the context of A/V systems, i.e., in what ways can IT help do your job better. For IT professionals, A/V is framed in the context of IT systems, i.e., how can IT be used to create A/V systems. The level of coverage is moderately technical, providing practical and actionable information for the following purposes:

- Understanding the forces causing the migration toward networked media
- Explaining file-based methods and how “tapeless” concepts improve workflow efficiency contrasted to stream-based methods.
- Appreciating the basics of networked media
- Evaluating a video system’s architectures, reliability, and scalability
- Understanding the fundamentals of networking, data servers, storage systems, data archive, and security as applied to networked media
- Comprehending the fundamental industry standards that apply to IT and A/V infrastructures
- Evaluating the trends for networked media solutions and technology
- Providing insight into software platforms and their trade-offs
- Learning the support and maintenance themes for these hybrid systems
- Knowing what questions to ask of potential equipment suppliers
- Reducing the FUD<sup>3</sup> and social uneasiness that surround /AV/IT systems

---

<sup>3</sup>Acronyms are used throughout the book. Usually, they are explained upon introduction, whereas in other cases, no definition is provided. When in doubt, check the Glossary or Internet sources.

For sure, the information in this book concentrates more on IT in the context of A/V than solely on traditional A/V basics. However, Chapter 11 provides an overview of A/V basics. If you are new to A/V concepts, then it may be wise to review this chapter first.

IT means choice. Universal platforms, standards, and flexibility all embody IT. The focus of the chapters that follow is on the application of A/V plus IT methods to build, operate, and support video systems in an IT-networked environment. If all this is alien to you, do not lose hope. Hang on and this book will turn alien to familiar. Do not become a prisoner of your point of view—widen out and explore the new vistas.

So, are you going with the flow? Are you catching the tidal wave that is changing A/V systems forever? Let us ride this ship into safe harbors and enjoy the benefits of converged AV/IT systems. Yes, let us start on our journey of illuminating A/V and IT systems in the light of each other's context.

This page intentionally left blank

# Networked Media in an IT Environment

## CONTENTS

1.0	Introduction	2
1.1	What is Networked Media?	2
1.2	Motivation Toward Networked Media	5
1.2.1	Force #1: Network Infrastructure and Bandwidth	7
1.2.2	Force #2: CPU Compute Power	9
1.2.3	Force #3: Storage Density, Bandwidth, and Power	12
1.2.4	Force #4: IT Systems Manageability	15
1.2.5	Force #5: Software Architectures	16
1.2.6	Force #6: Interoperability	17
1.2.7	Force #7: User Application Functionality	19
1.2.8	Force #8: Reliability and Scalability	19
1.2.9	The Eight Forces: A Conclusion	20
1.3	Three Fundamental Methods of Moving A/V Data	21
1.4	Systemwide Timing Migration	23
1.5	Can “IT” Meet the Needs of A/V Workflows?	23
1.6	Advantages and Disadvantages of Methods	28
1.6.1	Trade-off Discussion	28
1.7	It’s A Wrap: Some Final Words	31
	References	31

### 1.0 INTRODUCTION

Among his many great accomplishments, Sir Isaac Newton discovered three fundamental laws of physics. Law number one is often called the *law of inertia* and is stated as *Every object in a state of uniform motion remains in that state unless an external force is applied to it.*

By analogy, this law may be applied to the recent state of A/V system technology. The traditional methods (*state of uniform motion*) of moving video [serial digital interface (SDI), composite...] and storing video (tape, VTRs) assets are accepted and comfortable to the engineering and production staff, fit existing workflows, and are proven to work. Some facility managers feel, “If it’s not broken don’t fix it.” Ah, but the second part of the law states “... *unless an external force is applied to it.*” So, what force is moving A/V systems today into a new direction—the direction of networked media? Well, it is the force of information technology (IT)<sup>1</sup> and all that is associated with it. Is this a benign force? Will its muscle be beneficial for the broadcast and professional A/V production businesses? What are the advantages and trade-offs of this new direction? These issues and many more are investigated in the course of this book. First, what is networked media?

### 1.1 WHAT IS NETWORKED MEDIA?

The term *network* in the context of our discussions is limited to a system of digital interconnections that communicate, move, or transfer information. This primarily includes traditional IT-based LAN (Ethernet in all forms), WAN (Telco-provided links), and Fibre Channel network technologies. Some secondary linkages such as IEEE-1394, USB, and SCSI are used for very short haul connectivity. The secondary links have limited geographical reach and are not as fully routable and extensible as the primary links.

In contrast to traditional A/V equipment,<sup>2</sup> networked media relies on technology and components supplied by IT equipment vendors to move, store, and manipulate A/V assets. With all respect to the stalwart SDI router, it is woefully lacking in terms of true networkability. Only by Herculean feats can SDI links be networked in similar ways to what Ethernet and Internet Protocol (IP) routing can offer.

---

<sup>1</sup> IT storage and networking concepts are used universally in business systems worldwide. See the Introduction for background on IT.

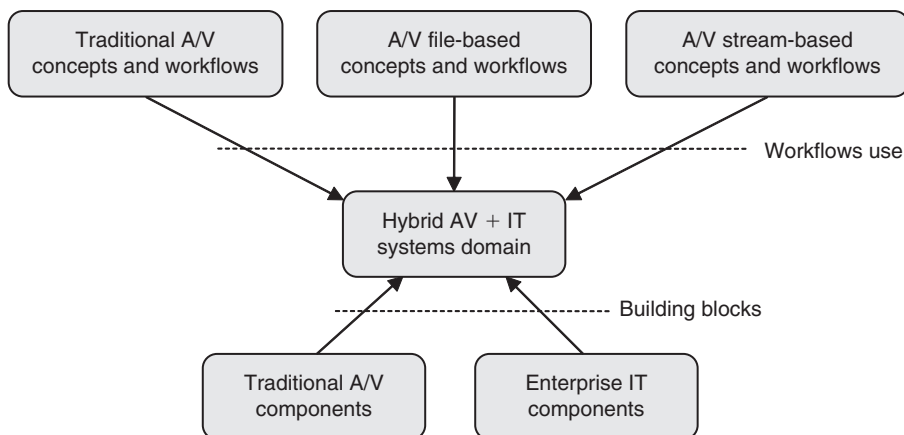
<sup>2</sup> If you are not familiar with traditional A/V techniques, consider reviewing Chapter 11 for a general overview.

The following fundamental methods and concepts are examples of networked media.

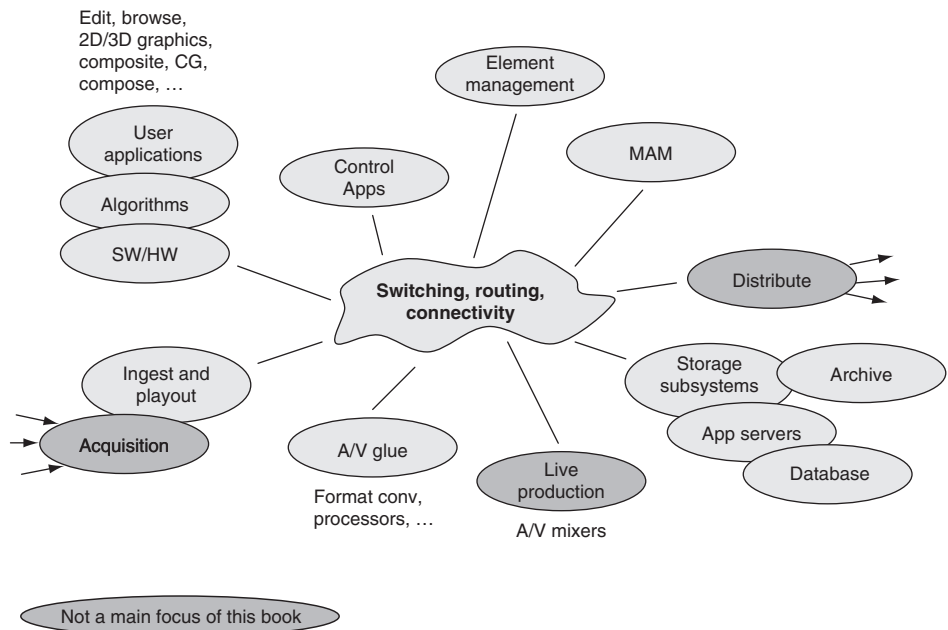
- Direct-to-storage media ingest, edit, playout, process, and so on
- 100 percent reliable file transfer methods
- A/V streaming over IT networks
- Media/data routing and distribution using Ethernet LAN connectivity, Fibre Channel, WAN, and other links with appropriate switching
- Networkable A/V components (media clients): ingest ports, edit stations, data servers, caches, playout ports, proxy stations, controllers, A/V process stations, and so on
- A/V-as-data archive; not traditional videotape archive

For the most part, file-based technology and workflows (so-called tapeless) use networked media techniques. So, file-based technology is implemented using elements of AV + IT systems and is contrasted to stream-based throughout this book. Also, the AV/IT systems domain is a superset of the file-based concepts domain. Figure 1.1 illustrates the relationships between the various actors in the AV/IT systems domain.

The world of networked media spans from a simple home video network to large broadcast and postproduction facilities. There are countless applications of the concepts in the list just given, and many are described in the course of the book. We will concentrate on the subset that is the realm of the professional/enterprise (and prosumer) media producer/creator. Figure 1.2 illustrates the domain of the general professional video system, whether digital or not.



**FIGURE 1.1** Professional video system components.



**FIGURE 1.2** *Switching, routing, connectivity.*

The components are connected via the routing domain to create an unlimited variety of systems to perform almost any desired workflow. Examples of these systems include the following:

1. Analog based (analog tape + A/V processing + analog connectivity)
2. Digitally based (digital tape + A/V processing + digital connectivity)
3. Networked based (data servers + A/V processing + networked connectivity)
4. Hybrid combinations of all the above

The distinction between digitally based and networked based may seem inconsequential, as networks are digital in nature. Think of it this way: all networks are digital, but not all digital interconnectivity is net-workable. The ubiquitous SDI link is certainly digital, but it is not easily networkable. Over the course of discussions, our focus highlights item #3 as primary, with the others taking on supporting roles. Items #1 and #2 are defined for our discussions as “traditional A/V” compared to item #3, which is referred to as “AV/IT or IT-based AV” throughout this book.

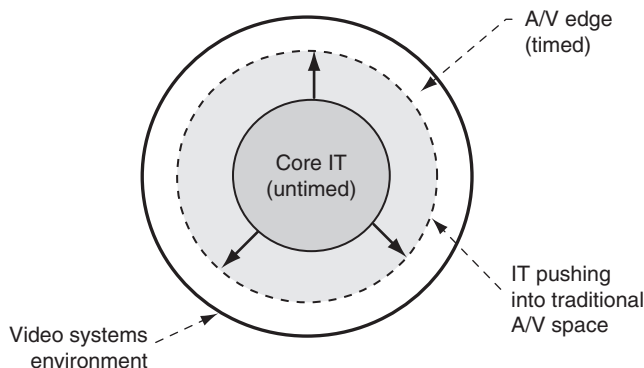
Again, looking at Figure 1.1, most of the components may be combined in various ways to make up an IT-based professional video system. However, three elements have extended applications beyond our consideration. The world of media acquisition and distribution is enormous and will not be considered in all its glory. Also, media distribution methods using terrestrial RF broadcast,

cable TV networks, the Web, and satellite are beyond our scope. Additionally, live (sporting events, news, etc.) production methods (field cameras, vision mixers) fall into a gray area in terms of the application of IT. However, most new field cameras don't use videotape; instead, they use file-based optical disc or flash memory for storage. These offer nonlinear access and network ports.

## 1.2 MOTIVATION TOWARD NETWORKED MEDIA

Over the past few years, there has been a gradual increase in new A/V products that steal pages from the playbook of IT methods. Figure 1.3 shows the changing nature of video systems. At the core are untimed, asynchronous IT networks, data servers, and storage subsystems. At the edges are traditional timed (in the horizontal and vertical raster-scanning sense) A/V circuits and links that interface to the core. The core is expanding rapidly and consuming many of the functionalities that were once performed solely by A/V-specific devices. This picture likely raises many questions in your mind. How can not-designed-for-video equipment replace carefully designed video gear? How far can this trend continue before all notion of timed video has disappeared? What is fueling the expansion? Will the trend reverse itself after poor experiences have accumulated? Our discussions will answer these questions.

There is no single motivational force responsible for the shift to IT media. There are at least two levels of motivational factors: business related and technology related. At the business level there is what may be called the prime directive. Simply put, owners and managers of video and broadcast facilities are demanding, *"I want more and better but with less."* That is a tall order, but this directive is driving many purchasing decisions every day. More what? More compelling content, more distribution channels, more throughput. Better what? Better quality (HD, for example), more compelling imagery, better production



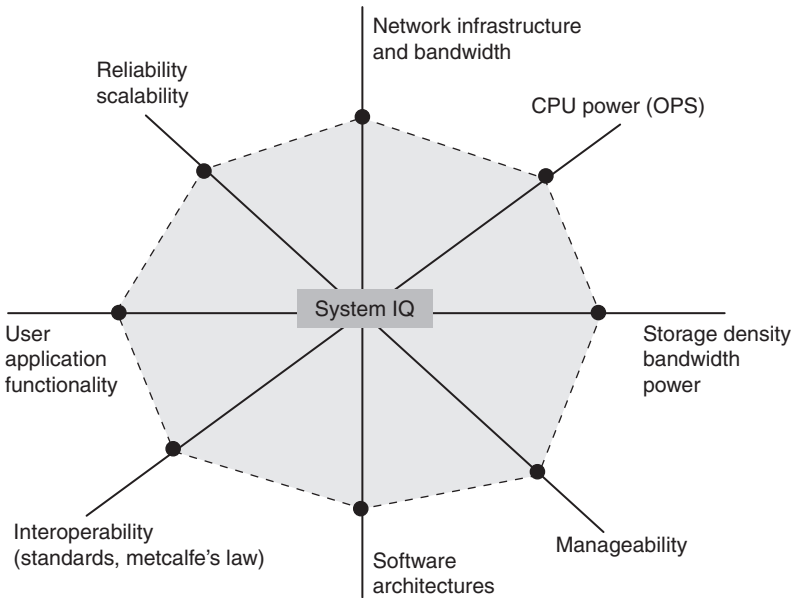
**FIGURE 1.3** The expansion of the IT universe into A/V space.



value, better branding. Less what? Less capital spending, less ongoing operational cost, fewer maintenance headaches. All these combine to create value and the real business driver—more profit. Of course, there are many aspects to more/better/less, but let us focus our attention on the technical side of the operations. If we want to achieve more/better/less, the technology selection is key. The following sections examine this aspect.

Of course, there are issues with the transition to the AV/IT environment from the comfortable world of traditional A/V video. All is not peaches and cream. The so-called move to IT has lots of baggage. The following sections focus on the positive workflow-related benefits of the move to IT. However, in Chapter 10, several case studies examine real-world examples of those who took the bold step to create hybrid IT and A/V environments. In that chapter you will feel the pains and joys of the implementers on the bleeding edge. In that consideration we examine the cultural, organizational, operational, and technical implications of the move to IT.

At least eight *technical* forces are combining to create a resulting vector that is moving media systems in the direction of IT. Let us call the area enclosed by the boundary contour of Figure 1.4 the *system IQ*. This metric is synthetic, but consider the area (bigger is better) as a measure of a system's "goodness" to meet or exceed a user's requirements. Each of the eight axes is labeled with one of the forces. Let us devote some time to each force and add insight into their individual significance. Also, for each force, a measure of workflow improvement



**FIGURE 1.4** Eight forces enabling the new AV/IT infrastructure.

due to the force is described. After all, without an improvement in cost savings, quality, production value, resource utilization, or process delay, a force would be rather feeble. Although the forces are numbered, this is not meant to imply a priority to their importance.

### 1.2.1 Force #1: Network Infrastructure and Bandwidth

The glue of any IT system is its routing and connectivity network. The faster and wider the interconnectivity, the more access any node has to another node. But of what benefit is this to a media producer? What are the workflow improvements? Networks break the barrier of geography and allow for distributed workflows that are impossible using legacy A/V equipment. For example, imagine a joint production project with collaborating editors in Tokyo, New York City, and London (or among different editors in a campus environment). Over a WAN they can share a common pool of A/V content, access the same archive, and creatively develop a project using a coordinated workflow management system. File transfer is also enabled by LANs and WANs. Does file transfer improve workflow efficiency? Consider the following steps for a typical videotape-based copy and transfer cycle:

1. Create a tape dub of material—delay and cost.
  - a. Check quality of dub—delay and cost.
  - b. Separately package any closed caption files, audio descriptive narration files (SAP channel), and ratings information.
2. Deliver to recipient using land-based courier—delay and cost.
3. Receive package, log it, and distribute to end user—delay mainly.
  - a. Integrate the closed caption and descriptive notation ready for playout.

## THE PERFECT VIDEO SYSTEM

The late itinerant Hungarian mathematician Paul Erdos developed the idea of “The Book of Mathematical Proofs” written by God. In his spare time, God filled it with perfect mathematical proofs. For every imaginable mathematical problem or puzzle that one can posit, the book contains a correspondingly elegant and beautifully simple proof that cannot be improved upon. Erdos imagined that all the proofs developed by mere mortal mathematicians could only hope to equal those in the “Book.” We too can imag-

ine a similar book filled with perfectly ideal video systems designed to match all the requirements of their users. Of the many architectural choices, of the many equipment preferences, and of the many design decisions, our book would contain a video system that could not be improved upon for a given set of user workflow requirements. True, such a book is a dream. However, many of the principles discussed in these chapters would make up the fabric and backbone of our book.



4. Ingest into archive or video server system (and enter any metadata)—delay and cost.
  - a. QA ingested material—delay and cost.
5. Archive videotape—cost to manage and store it, format obsolescence worries.

It is obvious that the steps are prone to error, are costly, and add delay. Let us look at the corresponding file transfer workflow:

1. Locate target file(s) to transfer.
2. Initiate and transfer file(s) to end station—minimum delay for transfer (seconds to hours, depending on desired transfer speed).

Additionally, file-associated metadata are included in the transfer, thereby eliminating another cause of error—manual metadata logging. The transferred file integrity is 100 percent guaranteed accurate.

What are the advantages? No QA process steps—or very short ones—delay cut from days to minutes and guaranteed delivery (not lost or stuck in shipment) to the end user. All in all, file transfer improves the workflow of making a copy and distribution of a program in meaningful ways. The walls of the traditional video facility are crumbling, and the new virtual facility is an anywhere-anytime operation. So what are the technology trends for LANs and WANs?

Not all that long ago, Ethernet seemed stuck indefinitely at 100 Mbps. Fortunately, there is a continual press forward to higher bandwidths and reach of networks. Today it is not uncommon to see 10-Gbps Ethernet links and routers in high-end data centers.

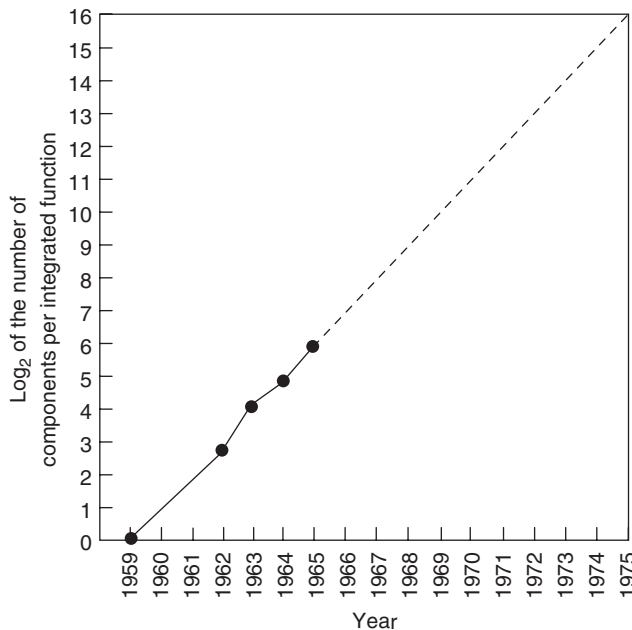
Let us take a tangent for a moment and investigate the very high end of connectivity. Using wavelength division multiplexing on optical fiber, researchers at Alcatel-Lucent Bell Labs (4/07) have proven that a WDM optical system is capable of delivering ~50,000 Gbps of data on one strand of fiber. Using 320 different wavelengths, each carrying a 156.25-Gbps payload, they postulate that the astronomical rate of ~50 Tbps is achievable per strand of fiber (see Appendix F).

Let us assume that we have encoded an immense collection of MPEG movies and programs each at 5 Mbps. At this rate, one could transmit *10 million* different programs simultaneously on one single fiber. Since most fiber cables carry 200+ strands, one properly snaked cable could serve *2 billion* homes, each accessing a unique program. Ah, so many channels, so few people. Amazing? Yes, but tomorrow promises even greater bandwidths. What is the point of this hyperbolic illustration? Video distribution and production workflows will be impacted greatly by these major advances in connectivity. Fasten your seat belt and hold on for a wild ride.

### 1.2.2 Force #2: CPU Compute Power

In a nutshell, it all follows from Moore's law. Simply put, Gordon Moore from Intel stated that integrated circuit density doubles every 18 months. The law had been in effect since 1965 and will likely continue at least until 2015, according to Moore in statements made in 2005. Initially, the doubling occurred every 12 months, but it has slowed to a doubling every ~24 months due to CPU complexity. Figure 1.5 shows the famous diagram redrawn from Moore's original paper (*Cramming more components onto integrated circuits*) (Moore 1965) and shows the doubling trend every 12 months. This diagram is the essence of Moore's law. Early among Intel's CPUs was the 8008 with 2,500 transistors. As a graduate student at UC Berkeley, the author wrote an 8008 program to control elevator operations. In 2008 the Dual-Core Intel Itanium 2 Processor had 1.72 billion transistors per die (2 CPUs). The Nvidia 9800 class, dual GPU, computes 1.1 trillion operations per second, bringing world-class 3D reality to the display. The Sony ZEGO technology platform is based on the Cell Engine chipset and is targeted at real-time HD video transforms and 2D/3D image rendering. Hence, the predictive power of Moore's law.

This law was not the first but the *fifth* paradigm to provide exponential growth of computing. Starting in 1900 with purely electromechanical systems, relays followed in the 1940s, then vacuum tubes, then transistors, and then integrated circuits. Since 1900, "computing power" has increased 10 trillion

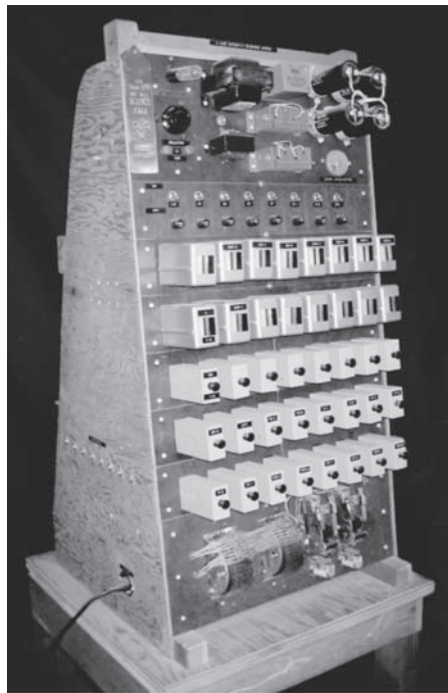


**FIGURE 1.5** Moore's law: Graph from his original paper.  
Source: *Electronics*, Volume 38, Number 8, April 19, 1965.

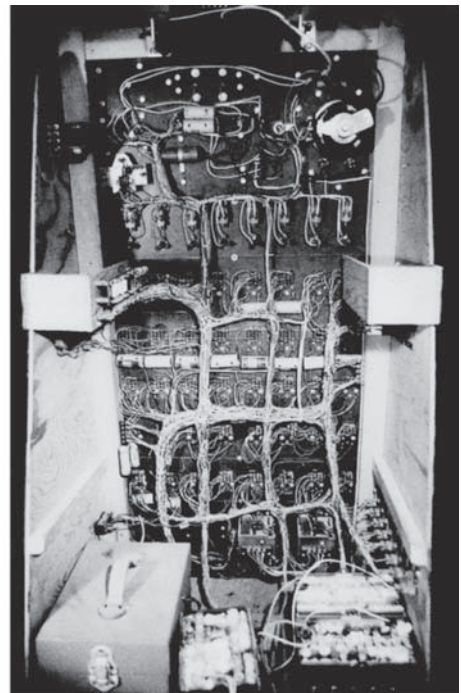
times. Our appetite for computing power is growing to consume all available power.

Demonstrating one of the paradigms of computation, while a Lowell High School student, the author designed and built an eight-line, relay-based, automatic telephone system for a San Francisco Science Fair. Figure 1.6 shows the final 60-relay design. Relay logic was relatively straightforward, and the sound of the relays completing a call was always a kick. For a teenager, transistors were way too quiet. The top of the unit is the power supply, the midsection has 40 of the 60 relays, and the lower section has two dial-activated rotary relays and two line-finder rotary relays. In the rear is the dial tone generator, batteries, and some additional relays. Not shown is a sound-proof box containing relays for generating the 20 Hz ringing voltage and various timing intervals.

Video processing needs a huge amount of computing power to perform real-time or “human fast” operations. Once left to the domain of purpose-built video circuits, CPUs are now performing three-dimensional (3D) effects, noise filtering, compositing, compressing video (a la MPEG), and completing other mathematically intensive operations in real time or faster. It is only getting easier



Front view of phone system



Rear view of phone system

**FIGURE 1.6** *Eight-line, relay-based, automatic telephone system.  
From: May Kovalick.*

to manipulate digital video, which has consigned traditional video circuits to a smaller and smaller part of the overall system.

Running CPUs in parallel, dual, and quad cores, for example, increases the total processing power available to applications. AMD has announced a 12-core processor chip targeted at servers, due out in 2010. The computing power of these systems is enormous, and performance can exceed a trillion operations per second (TOPS). There are more details on this subject in Appendix C.

On the memory front, the cost of one megabyte of DRAM has dropped precipitously from \$5,000 in 1977 (Hennessy et al. 2003) to \$.02 in 2010e (Source: Objective Analysis) in constant dollars. This is a 250,000 factor decrease in only 33 years. At least two video server manufacturers offer a RAM-based server while eschewing the disk drive completely. One worry is processor power consumption. The rule has been a doubling of processor power every 36 months. This is an untenable progression, since the cooling requirements become very difficult to meet. For example, the Intel Pentium Processor XE draws 160 watts. Fortunately, the new multicore versions use less than half this power.

All in all, CPU and memory price/performance are ever decreasing to the benefit of media system designers and their users. Incidentally, CPU clock speed has increased by a factor of 750 $\times$  from the introduction of the 8008 in 1972 until the Pentium4 in 2000.

So what is the workflow improvement? Fewer devices are needed to accomplish a given set of operations. The end-to-end processing chain has fewer links. Many A/V operations can be performed in real time using off-the-shelf commodity CPUs. There are, however, a few specialized processors that are optimized for certain tasks and application spaces.

In the area of specialized processors, the list includes

- Graphics processors (NVIDIA and AMD/ATI Technologies, for example)
- Embedded processors (Intel, Infineon, TI, and Motorola, for example)

## WHAT WILL YOU DO WITH A 2 BILLION TRANSISTOR CPU?



Is such a device overkill? Consider some CPU-based software A/V applications:

- Encode and decode MPEG HD video in real time
- Perform 3D effects in real time much like dedicated graphics processors do today
- Format conversions, transcoding in faster than real time
- Three-dimensional animation rendering

- Real-time image recognition
- Complex video processing
- Compressed domain processing

As processing power increases, there will be less dependence on special-purpose hardware to manipulate video. It is possible that video processing HW will become a relic of the past—time will tell.

- Media processors (TI, Analog Devices, and Philips, for example)
- Network processors (IBM, Intel, Xelerated, and a host of others) that clock in at 91.3

The world's most powerful computing platforms are documented at [www.top500.org](http://www.top500.org). The SX-9, from NEC, is capable of calculating 839 teraflops—or 839 trillion floating-point operations per second—and was considered the world's most powerful computing platform in 2008.

Fast I/O is required to keep up with increasing CPU speeds. One of the leaders in this area is the PCI Express bus (PCIe or PCI-E). Not to be confused with PCI-X, this is an implementation of the PCI bus that uses existing PCI programming concepts and communications standards. However, it is based on serial connectivity, not parallel as with the original PCI bus. The basic “X1” link has a peak data bandwidth of 2 Gbps. The link is bidirectional, so the effective data transfer rate is 4 Gbps. Links may be bundled and are referred to as X1, X4, X8, and X16. An X16 bus structure supports 64 Gbps of throughput. All this is good news for A/V systems. The link uses 8B/10B encoding (see Appendix E).

There is every reason to be optimistic about the future of the “CPU,” especially for A/V computing. But will it become the strong link in the computation chain of otherwise weak elements? Fortunately not, as the other forces grow in strength, too.

### 1.2.3 Force #3: Storage Density, Bandwidth, and Power

At 3 o'clock on Figure 1.4 is the dimension of storage density (cost/GB), storage bandwidth<sup>3</sup> [cost/(Mbps)], and power consumed (W/GB). For all metrics, smaller is better. Unless you have been living in a cave for the past 20 years, it is obvious that disk drive capacity per unit has been climbing at an astronomical rate. Much of the technology that makes up drives and other storage media also follows Moore's law, hence the capacity increase. The dimension of storage is a broad topic. The four main storage means are hard disk drives (HDD), optical disk, tape, and DRAM/Flash. The application spaces for these are as follows:

1. HDD—video servers, file/database servers, Web servers, PCs of all types, personal video recorders, embedded products (portable music players)
2. Optical disk—DVD (4.7 GB single sided), CD (700 MB), Blu-ray (up to 25 GB single sided, 50 GB double sided), and other lesser known devices. Some sample applications are
  - a. Consumer DVD—SD and HD
  - b. Sony XDCAM HD using a variation of the Blu-ray format for field acquisition at 50 GB per disc
  - c. Archive, backup

---

<sup>3</sup> The terms *bandwidth* and *data rate* are equivalent in a colloquial sense.

3. Tape—traditional videotape, archive, backup. Archive tape technology is discussed in Chapter 3A.
4. RAM/Flash—RAM is being used to replace disc and tape for select applications, including deep cache. The Panasonic P2 camera is a good example of a professional camera that has only removable Flash cards as the media store. In 2009, 32 GB modules are common and able to store 160 minutes of DV25 video. P2 cards have writing speeds of 640 Mbps, exceeding conventional Flash memory. Sony also offers a Flash-based camera, the XDCAM EX, using the SxS PRO cards. By 2010, 128 GB Flash cards will be in use.
5. SSD—The Solid State Disc is a device that mimics an HDD using identical I/O ports and form factor but with Flash memory replacing rotating platters for storage. There are numerous reasons to replace HDD with SSD in some applications. See Appendix L for a review of the pros and cons.

The hard disk is having an immense impact on the evolution of IT systems. Consider the implications of Figure 1.7, reprinted from an article by IBM researchers (Morris et al. 2003).

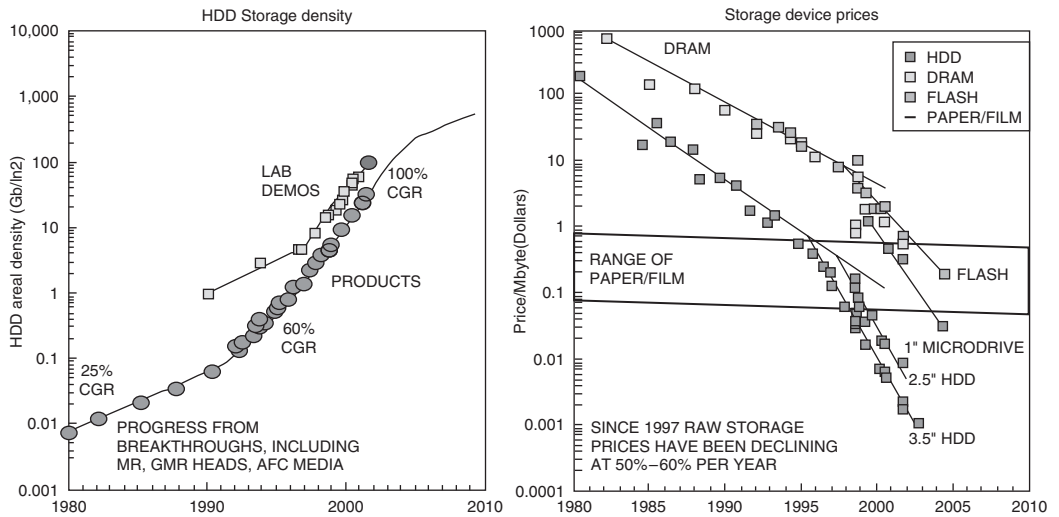
Storage density is currently over 250 Gbits/in.<sup>2</sup> on the surface of the rotating platter. This increased at a compound growth rate (CGR) of 60 percent per year in 2008 and enabled 2.5-inch form factor HD drives with capacities of 1 TB. This rate is expected to slow down modestly to about 40 percent per year in 2010. HDD prices have decreased by about 5 orders of magnitude (100,000:1) since 1980, whereas storage systems' prices have decreased by a factor of 2.5 orders of magnitude. The faster fall in HDD prices compared to system prices implies that HDDs are a smaller overall part of storage systems. Chapter 3A discusses storage systems in detail. Raw HDD prices have been falling 50–60 percent a year since 1997.

It is enlightening to forecast the future of HDD performance, keeping in mind that fortune telling is risky business. So using Figure 1.7 and extrapolating to 2010, we should expect to see HDD capacities of around 1.5TB per 2.5-inch unit at a cost of \$40 in constant dollars. Using the most advanced audio compression (64 Kbps), a single HDD could store 1 million tracks of music (3 min average length). Imagine the world's collection of music on your personal computer or in your pocket. All this bodes well for professional video systems, too. Video/file servers with 10TB of combined storage (91 days' worth of continuously playing content at SD-compressed rates) will be routine. Even at HD compressed production rates of say 150 Mbps, *one* 1.5TB HDD will store 2.2 hr of material and at 19.3 Mbps (ATSC payload rate) will store nearly 17 hr.

More storage for less is the trend, and it will likely continue. When integrated into a full-featured chassis with high-performance I/O, RAID protection and monitoring, the system price per GB will be higher in all cases.



HDD storage density is improving at 100 percent per year (currently over 100 Gbit/in.<sup>2</sup>).  
The price of storage is decreasing rapidly and is now significantly cheaper than paper or film.



**FIGURE 1.7** Storage media performance trends.  
Source: IBM.



### Storage Rule of Thumb

Ten megabits per second compressed video consumes 4.5 GB/hr of storage. Use this convenient data point to scale to other rates.

The development to higher capacities has other side benefits too. Note the following trends:

- Internal HDD R/W sustained rates (best case, max) are currently at 1,000 Mbps (1 TB drive) and are increasing at 40 percent per year for SAS class HDD units. The actual achieved I/O for normal transactional loads will be lower due to random head seek and latency delays.
- Power per GB for HDD units is dropping at 60 percent per year. In 2009 a 1 TB, R/W active drive consumes on the order of .01 W/GB. This is crucial in large data centers that have hundreds of TB of storage. Storage systems consume an order of magnitude or more of power per GB due to the added components overhead.

Are there any workflow improvements? Oh yes, and in spades. This force is single-handedly driving IT into broadcast and other professional A/V applications. Consider the case of the video server. In 1995, HP pioneered and

introduced the world's first mission-critical MPEG-based video server (the MediaStream Server) for the broadcast TV market. Initially, the product used individual 9 GB hard drives in the storage arrays. In 2009, storage arrays support 1TB drives. Now that is progress. Video servers enable hands-free, automated operations for many A/V applications.

### **1.2.3.1 SCSI Versus ATA Drives**

Two different types of HDD have emerged: one is the so-called SCSI HDD and the other is the ATA (IDE) drive. In many ways the drives are similar. The SCSI drive is aimed at enterprise data centers where top-notch performance is required. The ATA drive is aimed at the PC market where less performance is acceptable. Because of the different target markets, the common perception is that SCSI drives are the right choice for high-end applications and ATA drives are for home use and light business. A comparative summary follows:

- ATA drives are about one-third the price of SCSI drives.
- SCSI drives have a top platter spin of 15,000 rpm, whereas ATA tops at 7200.
- ATA drives have a simpler and less flexible I/O interface than SCSI.
- ATA consumes less power.
- ATA drives sport 1TB capacities in 2009, more than SCSI drives.
- The reliability edge is given to SCSI due to more testing during the R&D cycle.

Because of the lower price of the ATA HDD, many video product manufacturers have found ways to use ATA drives in their RAID-based storage systems. The biggest deficit in the ATA drive is the R/W head access time, which is determined by the platter rotational speed. In the world of A/V storage, the faster SCSI platter rotation speed is not necessarily a big advantage. For the enterprise data center, the average HDD R/W transaction block size is 4–8 KB. However, for A/V data transactions, several MB is a normal R/W block size (video files are huge). There is a complete discussion of this topic in Chapter 3A.

The ATA is on the ascension for A/V systems. Working around the less than ideal specs of the ATA drive yields big cost savings. These drives are almost always bundled with RAID, which improves overall reliability. Look for the ATA drive to become the centerpiece for A/V storage systems. In addition, some drive manufacturers specialize in ATA (and SATA) drives and offer specs that compete very favorably with SCSI on most fronts. See Chapter 5 for more details on HDD reliability.

### **1.2.4 Force #4: IT Systems Manageability**

Unmanaged equipment can quickly become the chaotic nightmare of searching for bad components and repairing them while trying to sustain service. Long ago, the IT community realized the necessity to actively manage the routers,

switches, servers, LAN and WAN links, and even software applications that comprise an IT system. However, most legacy A/V-specific equipment has no standard way to report errors, warnings, or status messages. Ad hoc solutions from each vendor have been the norm compared to the standardized methods that the IT industry embraces.

Managed equipment yields savings with less downtime, faster diagnosis, and fewer staff members to manage thousands of system components. Entire industries have risen to provide embedded software for element status and error reporting, management protocol software, and, most importantly, monitoring stations to view and notify of the status of all system components. This includes the configuration and performance of the system under scrutiny. There are sufficient standards to create a vendor plug-and-play environment so users have their choice of products when creating a management environment. However, there will always be vendor-specific aspects of element management for which only they will provide management applications.

No one doubts that the IT management wave will be adopted by many A/V equipment manufacturers over the next few years. The IT momentum, coupled with the advantages of the approach, spells doom for unmanaged equipment. Of course, the A/V industry must standardize the A/V-specific aspects of element management. See Chapter 9 for an extended discussion. Let us leave the topic at this juncture. Has this improved the workflow to produce or generate video programming? Well, only indirectly. With less downtime and more accessible resources, workflows will literally flow better.

### 1.2.5 Force #5: Software Architectures

There are two main forces in software systems today. They both have their adherents and detractors, and siding with one faction or the other can be a religious experience. It is obvious to almost anyone that Microsoft wields a mighty sword and that many professional application developers use their Windows OS and .NET software framework for design. The other camp is the Linux-based Open Source movement with backing by IBM, HP, and countless others who advocate open systems (see, for example, [www.sourceforge.net](http://www.sourceforge.net), [www.openoffice.org](http://www.openoffice.org), [www.linux.org](http://www.linux.org)).

Closely associated with this is the Java EE framework (and several very good development platforms) as an alternative to the Microsoft .NET programming environment. Java is now open sourced. The .NET and Java camps have built up a momentum of very credible solutions. Are there other alternatives? Yes, but they are niche players, and the sum total of their influence will be small. Next, a little background on the status of the OS market.

The lion's share of the OS marketplace comes from two segments, namely enterprise servers (database servers, Web servers, file servers, and so on) and client based (desktop). After these two behemoths, many smaller segments follow, such as the embedded OS, mobile phones, and more. Market Share by Net

Applications analysts estimated that Microsoft Windows controlled 92 percent of the desktop OS real estate in early 2008. Worldwide, Apple MacOS gets 7.6 percent, and the Linux desktop is at 1 percent.

For the worldwide server space in early 2008, Gartner stated that UNIX commands a 7 percent share, Microsoft Windows a 67 percent share, and Linux a 23 percent share. The server OS market is now a two horse race. In 2005 UNIX was in first place. As server OS virtualization gains ground, some expect that Linux will have an edge over Windows. Time will tell.

Does the selection of an OS and programming language development platform bring end user workflow improvements? Admittedly, this is a complex question. The advantages are first felt by the equipment manufacturers. How? Using either Java or .NET programming paradigms produces efficiencies in product development, product enhancements, and change management. If end users are given access to the code base or APIs for systems integration and enhancement purposes, then they will reap the advantages of these programming environments. Also, if the IT staff is trained and comfortable with these environments, then any needed upgrades or patches are more likely to be implemented without issue or anxiety.

Workflow improvements will come from the power of the software applications produced by either of these environments. Also, their flexibility (well-documented, open programming interfaces) will allow for software enhancements to be made to meet changing business needs. Software-based systems allow for great flexibility in creating and changing workflows. Older non-IT-based A/V systems can be rigid in their topology. IT frees us to create almost any A/V and data workflow imaginable. Many video facilities already have one or more programmers on staff to effect software changes when business needs dictate. Look forward to a big leap in customer-developed solutions that work in harmony with vendor-provided equipment. The topic of programming environments is discussed in Chapter 4.

### 1.2.6 Force #6: Interoperability

Writer John Donne once said, "No man is an island; every man is a piece of the continent." Much has been written for and against his proclamation of the dependent need for others. For our discussion, we will side with the affirmative but apply the sentiment to islands of IT. Gone are the days of isolated islands of operations. Gone are the days of proprietary connectivity. Today, end users of video gear expect access to the Internet, email, compatible file formats, easy access to storage, and workflows that meet their needs for flexibility and production value. "Give me the universe of access and only then am I satisfied" is the mantra.

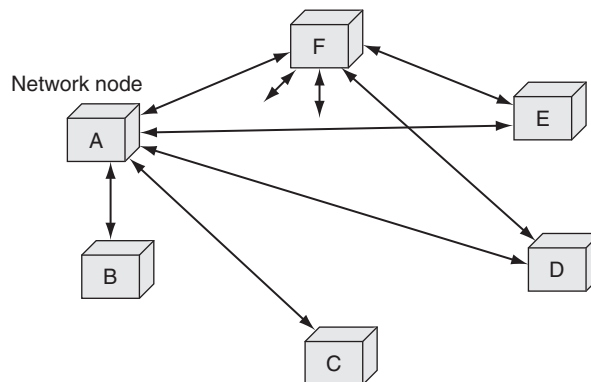
Does this mean that operational "islands" are a bad idea? By no means. Whether for security, reliability, control, application focus, or some other reason, equipment islands defined by their operational characteristics will be a design choice.

Robert Metcalfe, the inventor of Ethernet (Gilder 1993), once declared a decree now known as Metcalfe's law: "The value of a network of interconnected devices grows as the square of connected elements." What did he mean? Well, consider an email system of two members. Likely, boring and limited. But a billion member population is much more interesting and useful. So too with media interconnectivity. As the number of connected elements and users grows, the power of productivity grows as the square of the connected devices. Collaborative works, file sharing, common metadata, and media are all powerfully leveraged when networked. Networking also adds layers of software complexity, which must be managed by the IT staff.

So Metcalfe's law is the response to the plea, "Please, I want more productivity." Standards foster interconnectivity. SMPTE (Society of Motion Picture and Television Engineers), the EBU (European Broadcast Union), ARIB (Japan), the IEEE (Ethernet, for example), the ITU/ISO/IEC (MPEG, for example), and W3C (Web standards HTML and XML, for example) develop the standards that make Metcalfe's law a reality. There is more discussion on standards and user forums such as the Advanced Media Workflow Association (AMWA) in Chapter 2. Is there a demonstrative workflow improvement? Yes, in terms of nearly instant user/device access to A/V content, processors, access to metadata, and user collaboration.

You may wonder why the synergy of a system is a function of the square of the number of attached nodes. Consider that most communication paths are between nodal pairs in a network. For example, node A may request a file from node Z, which is only one possible choice for A. With  $N$  nodes there are roughly  $N^2$  number of combinations for 1:1 bidirectional communication; A can communicate with B or C or D, B can communicate with C or D, and so on until  $N^2$  combinations are accumulated—hence Metcalfe's law.

Figure 1.8 shows some of the pairwise combinations in a population of  $N = 6$ . For this case, there are  $2 * (5 + 4 + 3 + 2 + 1) = 30$  pairwise combinations (each



**FIGURE 1.8** Metcalfe's law: Combinations of unidirectional pairwise communication paths tend toward  $N^2$  as  $N$  (number of nodes) becomes large.

bidirectional path is counted as two unidirectional paths, hence the factor of 2 multiplier). However, 36 would be the value based on  $6^2$ . For  $N = 25$ , Metcalfe's law predicts 625 when there are 600 paths in actuality. The actual number of pairwise communication paths is  $N^2 - N$  so as  $N$  tends to be large the  $-N$  term is a diminishingly small correction, as Metcalfe must have known.

### 1.2.7 Force #7: User Application Functionality

Application functionality is now largely defined and accessed via graphical user interfaces (GUIs) and APIs. Many of the hard surfaces of old have been replaced by more flexible soft interfaces. Oh sure, there is still a need for hard surface interfaces for applications such as live event production with camera switching, audio control, and video server control. Nonetheless, most user interfaces in a media production facility will be soft based, thereby allowing for change with a custom look and feel. A GUI as defined by a manufacturer may also be augmented by end user-chosen "helper" applications such as media management, browsing, and so on. Using drag-and-drop functionality, a helper application can provide data objects to the main user application. In the end, soft interfaces are the ultimate in flexibility and customization.

Another hot area of interest is Web services. In brief, a Web service can be any business or data processing function that is made available over a network. Web services are components that accomplish a well-defined purpose and make their interfaces available via standard protocols and data formats. These services combine the best aspects of component-based development and Web infrastructure. Web services provide an ideal means to expose business (and A/V process) functions as automated and reusable interfaces. Web services offer a uniform mechanism for enterprise resources and applications to interface with one another. The promise of "utility computing" comes alive with these services. Imagine A/V service operators (codecs, converters, renders, effects, compositors, searching engines, etc.) being sold as components that other components or user applications can access at will to do specific operations. Entire workflows may be composed of these services driven by a user application layer that controls the logic of execution. There are already standard methods and data structures to support these concepts. There is a deeper discussion of these ideas in Chapter 4.

### 1.2.8 Force #8: Reliability and Scalability

The world's most mission-critical software systems run in an IT environment. Airline reservation systems, air traffic control, online banking, stock market transaction processing, and more depend on IT systems. There are four basic methods to improve a system's reliability:

- Minimize the risk that a failure will occur.
- Detect malfunctions quickly.
- Keep repair time quick.
- Limit impact of the failure.

In Chapter 5 there is an extensive discussion of reliability, availability, and scalability. Also, enterprise and mission-critical systems often need to scale from small to medium to large during their lifetime. Due to the critical nature of their operations, live upgrading is often needed, so scalability is a crucial aspect of an IT system.

Many video systems (broadcast TV stations, for example) also run mission-critical operations and share the same reliability and scalability requirements as banking and stock market transaction processing but with the added constraint of real-time response. IT-based A/V solutions may have all or some of the following characteristics:

- A/V glitch-free, no single point of failure (NSPOF) fault tolerance
- Real-time A/V access, processing, and distribution
- Off-site mirrors for disaster recovery
- Nearly instantaneous failover under automatic detection of a failed component
- Live upgrades to storage, clients, critical software components, and failed components
- Storage redundancy using RAID and other strategies

These characteristics have a very positive impact on workflow. Keeping systems alive and well keeps users happy. True fault-tolerant operations are practical and in use every day in facilities worldwide. As business and workflow requirements change, IT systems are able to keep pace by enabling changes of all critical components while in operation. All these aspects are addressed in more detail in Chapter 5. The bottom line is this: IT can meet the most critical needs of mission-critical video systems. Many systems offer better performance and reliability than purpose-built video equipment.

### 1.2.9 The Eight Forces: A Conclusion

The eight forces just described do indeed improve video system workflows by being more cost effective, reliable, higher performing, and flexible. The combined vector of all eight forces is moving video system design away from purpose-built, rigid, traditional A/V links toward an IT-based infrastructure. The remainder of this book delves deeper into each force and provides added information and insight. Several years ago, even the thought of building complex A/V systems with IT components seemed a joke. Today, the maturity of IT and its far-reaching capability grants it an honored place in video systems design. During the course of this book, several case studies will show impressive evidence of real-world systems with an IT backbone.

Despite the positive forces described, there exists a lot of fear, uncertainty, and doubt (FUD) surrounding AV/IT systems. Many of those well grounded in traditional A/V methods may find stumbling blocks at every step. The chapters

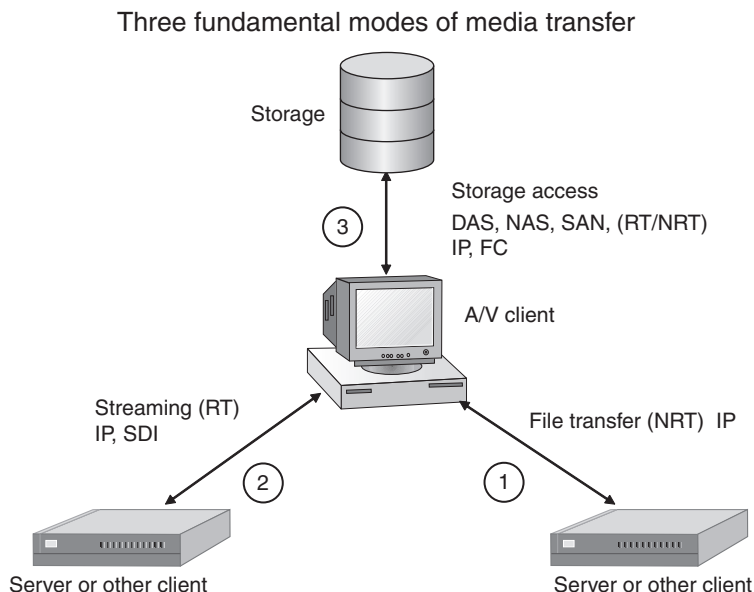
that follow will provide convincing evidence that AV/IT can indeed meet the challenges of mission-critical small, medium, and large video system designs.

### 1.3 THREE FUNDAMENTAL METHODS OF MOVING A/V DATA

There are three chief methods of moving A/V assets between devices/domains using IT. In Figure 1.9 these means are shown connected to the central “A/V client.” This client represents any digital device that exchanges media with another device. These methods form the base concepts for file-based and so-called tapeless workflows. The means are

1. **File transfer** using LAN/WAN in NRT or pseudo RT
2. **Streaming A/V** using LAN/WAN
  - a. Included is A/V streaming using traditional links
3. **Direct-to-Storage** real-time (RT) or non-real-time (NRT) A/V data access
  - a. DAS, SAN, and NAS storage access

Storage access, streaming A/V, and file transfer are all used in different ways to build video systems. The notion of an A/V *stream* is common in the Web delivery of media programming. In practice, any A/V data sent over a network or link in RT is a stream. For Figure 1.9, a client is some device that has a means to input/output A/V information over a link of some sort. Some systems depend exclusively on one method, whereas another may use a hybrid mix of all three. Each method has its strong and weak points, and selecting one over



**FIGURE 1.9** Three fundamental methods used to move A/V data.



## FILE-BASED AND TAPELESS WORKFLOWS: ARE THEY DIFFERENT?



The sense of *tapeless*, in A/V systems, implies video-tapeless. In fact, many A/V systems use tape for archive, so they are not truly tapeless. Archive tape is alive and well, as discussed in Chapter 3A. On the other hand, *file-based* is more encompassing a term. File-based

concepts transcend tape and are used to create networkable, IT-enabled, nonlinear-accessible, and malleable media workflows. The designation *networked media* covers file-based and IP streaming, so it defines a superset of IT-based media flows.

another requires a good knowledge of the workflows that need to be supported. Chapters 3A and 3B review storage access, and Chapter 2 reviews streaming and file transfer.

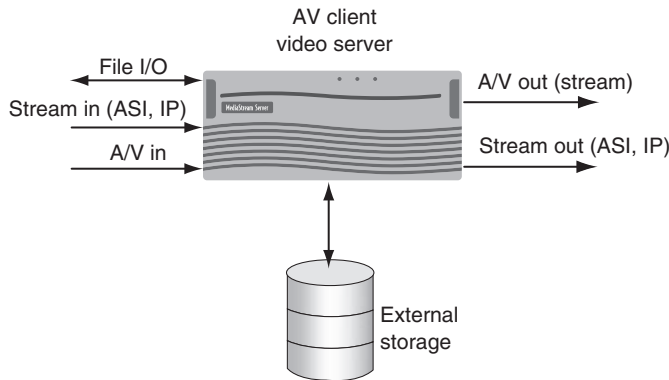
Two acronyms, *RT* and *NRT*, are used repeatedly throughout the book, so they deserve special mention. *RT* is used to represent an activity such as A/V streaming or storage access that occurs in the sense of video or audio real time. *NRT* is, as expected, an activity that is not *RT* but slower (1/10 real time) or faster (5× real time). *NRT*-based systems are less demanding in terms of quality of service (QoS). These two concepts are intrinsic to many of the themes in this book.

In Figure 1.9, each of the three links represents one of the A/V mover techniques. For example, one client may exchange files with another client, or the central client may R/W to storage directly. The client in the center supports all three methods. The diagram represents the logical view of these concepts. The physical view may, in fact, consolidate the “links” into one or more actual links. For example, storage access, IP streaming, and file transfer can use a single LAN link. The flows are separated at higher levels by the application software running on the client.

A practical example of the three-flow model is illustrated with the common video server (the A/V client) in Figure 1.10. The I/O demonstrates streaming using IP LAN, traditional A/V I/O, file transfer I/O, and storage access. The most basic video server supports only A/V I/O with internal storage, whereas a more complete model would support all three modes. All modes may be used simultaneously or separately, depending on the installed configuration. When evaluating a server, ask about all three modes to fully understand its capabilities.

Of course, there are other ways to move A/V assets (tape, optical disk manual transport), but these three are the focus of our discussions. Throughout the book, these means are discussed and dissected to better appreciate the advantages/disadvantages of each method.

One of the characteristics that help define a video system is the notion of A/V timing. Some systems have hundreds (or thousands) of video links that need to be frame accurate, lip synced to audio, and aligned for switching between



**FIGURE 1.10** Video server with support for files, streams, and storage access.

sources. The following section discusses the evolution from traditional A/V timing to that of a hybrid AV/IT system.

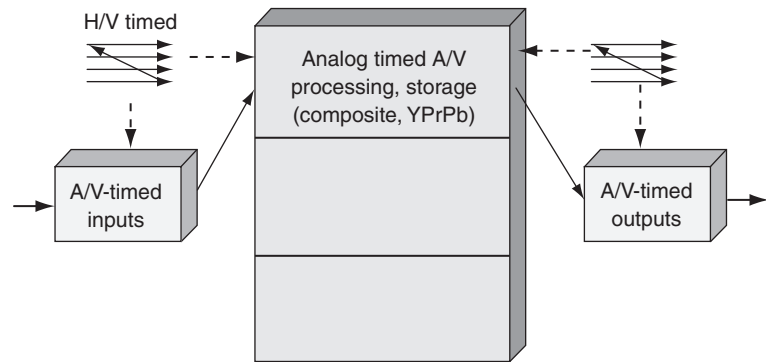
## 1.4 SYSTEMWIDE TIMING MIGRATION

When IT and A/V are mentioned in the same breath, many seasoned technical professionals express signs of worry. After all, is video not a special data type because of the precise horizontal and vertical timing relationships? It turns out that the needed timing can be preserved and still rely on IT at the *core* of the system. Figures 1.11, 1.12, and 1.13 show the migration from an all analog system to an AV/IT one. In Figure 1.11 every step of video processing and I/O needs to preserve the H/V timing. In Figure 1.12 the timing is easier to preserve due to the all-digital nature of the processing. In Figure 1.13 the H/V timing is evident only at the edges for traditional I/O and display purposes. Figure 1.13 is the hybrid AV/IT model for much of the discussion in this book.

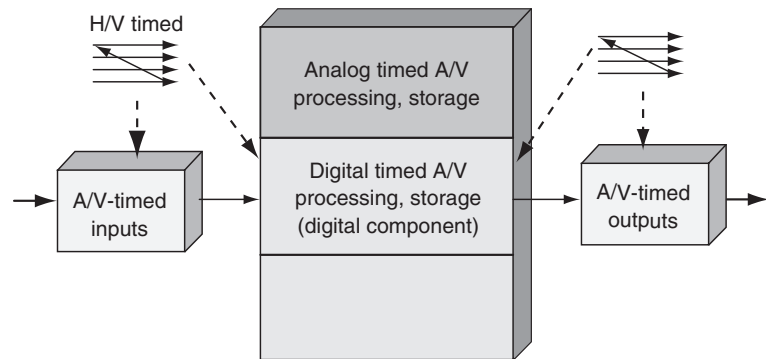
As discussed earlier, streaming using IT links is practical for some applications. If Figure 1.13 has no traditional A/V I/O and only network links used for I/O, then the notion of H/V timing is lost at the edge of the system. In this case, there is a need for new timing methods that are IT based and frame accurate. In 2009, there are very few pure IT-only video systems. In a few years, pure IT-only systems may become popular; until then, let us pass on this particular special case for now, although it is revisited in Chapter 2. Next, let us see how well AV/IT configuration fares compared to its older cousins.

## 1.5 CAN “IT” MEET THE NEEDS OF A/V WORKFLOWS?

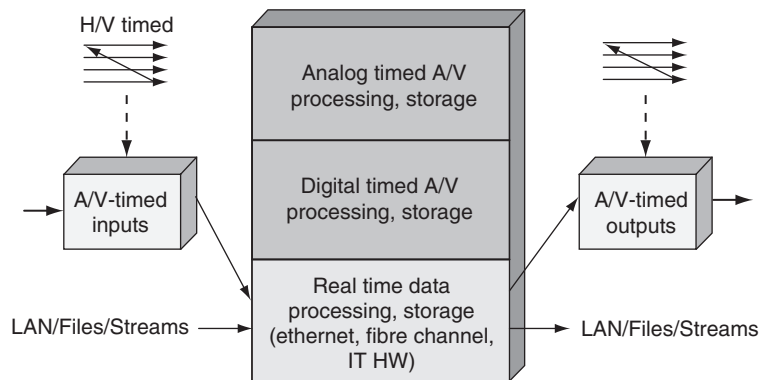
There are several important metrics when comparing traditional video to networked media system performance. These are the measures that the technical staff normally quantifies when calibrating or tuning a system. Figure 1.14 shows



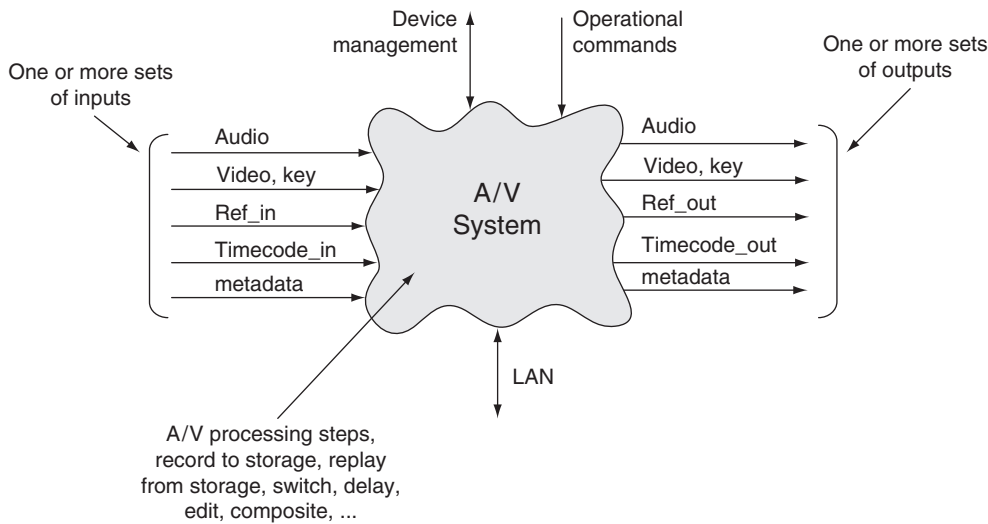
**FIGURE 1.11** *Traditional analog.*



**FIGURE 1.12** *Traditional digital.*



**FIGURE 1.13** *Hybrid AV + IT system.*



**FIGURE 1.14** *Traditional A/V system performance model.*

a simple A/V system where a perfectly aligned A/V signal set serves as an input along with a corresponding video reference signal and a time code (see Glossary) signal. The system may perform any process steps (delay, switch, process A/V, route, store, replay, etc.) to the input signals, including the control signals, and the output signals are always some function of the inputs. As a result, the outputs are referenced to the inputs in well-defined ways.

Under this group of conditions, the output should be completely deterministic and measurable to a set of specifications. The question then becomes whether a system composed of the hybrid mix of AV/IT can work as well as or better than a traditional A/V-only system. Imagine the system in Figure 1.14 composed of a hybrid AV/IT mix and measured to a set of specs. Then convert the system to a traditional A/V-only system and make the same measurements. How much—if any—would the measurements differ? Ideally, the AV/IT system would equal or exceed all measured specs. Is this the case? Are there some “sweet spots” for either system and, if so, what are they? Let us find out. The chief metrics (check Glossary if needed) of interest to us are as follows:

1. **A/V lip-sync.** This measures the amount of time delay between the audio signal and the video signal on the system output. Any deviation from zero shows the classic lip-sync characteristic. The input A/V alignment has exactly zero deviation.
2. **Video keying.** If an input key signal is present, does the keying operate without artifacts? Is it always frame aligned to the video fill signal?
3. **Frame accuracy.** Is the video output frame accurate to the output video reference (or input reference if desired)?

4. **Time code accuracy.** Is the time code output perfectly correlated (if needed) to the time code input value? Is the output time code perfectly correlated to the output video signal?
5. **Video reference accuracy.** If required, is the output video reference perfectly correlated to a video reference?
6. **A/V delay.** Is the A/V delay (latency) through the system acceptable and constant? For some cases the acceptable delay should be less than a line of video ( $<62\text{ |xs}$ ), whereas for others it may need to be several hours or more.
7. **Ancillary data.** If auxiliary information is present in the input signals (VBI, HANC, VANC, AES User Data, etc.), are they preserved as desired?
8. **Quality, noise, and artifacts.** Is there any noise or other undesired A/V properties on the output signals? Are the output A/V quality and timing as desired?
9. **Glitching.** Are there any undesired interruptions (illegal A/V, dropouts, etc.) on the output signals? How well does the system perform when the input signals exhibit glitching or are disconnected?
10. **Deterministic control.** Are all system operations deterministic? Does every operational command (switch, route, play, record, etc.) consistently function as expected?
11. **Wide geographic environments.** Can systems be created cost effectively across large distances?
12. **A/V storage.** Can the input signals be recorded and replayed on command? What are the delays?
13. **A/V processing.** Can the input A/V signals be processed at will for any reasonable operation?
14. **Metric drifting.** Do any of the measured spec values drift over time?
15. **User metadata.** Are user metadata supported?
16. **Systems management.** How are A/V devices managed? What are methods for alarm reporting? How is the configuration determined and maintained?
17. **Live A/V switching.** Traditional A/V switching is very mature. Can an AV/IT system do as well for all functional requirements?

All these metrics are indicative of system performance. If any one is not compliant, the entire system may be unusable. A detailed analysis of each element is beyond our scope, but the summary overview is appropriate. Table 1.1

**Table 1.1** Summary of Traditional A/V Versus AV/IT Metrics

System Metric	Traditional A/V	Hybrid AV/IT
1. A/V lip-sync	Needs careful attention to keep A + V in sync.	Needs careful attention to keep A + V in sync.
2. Video keying	Needs careful attention to keep V + K in sync.	Needs careful attention to keep V + K in sync.
3. Frame accuracy (FA)	Requires careful design.	Requires careful design. FA design requires new techniques when using IT.
4. Time code accuracy	Requires careful design.	Requires careful design.
5. Video ref accuracy	Requires careful design.	Requires careful design.
6. A/V delay	May require occasional calibration. Long delays are difficult to achieve; short ones (one video frame) are easier.	Deterministic delays are straightforward. Delays of <1 s may not be easy to achieve for some applications due to buffering delays.
7. Ancillary data	Straightforward.	Straightforward.
8. Quality, noise, and artifacts	If analog based, may be subject to noise injection and subsequent artifacts.	Digital systems less likely to add or be influenced by noise sources. Perfect, repeatable quality.
9. Glitching	Careful design required to avoid glitching due to various anomalous conditions.	Careful design required to avoid glitching due to various anomalous conditions.
10. Deterministic control	Mature today. Traditional RS422 control link is not networkable.	May be perfectly deterministic. Move toward all LAN-based control under way.
11. Wide geography	Difficult and expensive to preserve all the system specs.	Straightforward and a sweet spot for IT.
12. A/V storage	May be done with VTRs and robotics as needed. Expensive and impractical for some scenarios.	May be done with disc or RAM storage arrays. Large, dropout-free storage. Tape free for most operations.
13. A/V processing	Straightforward if done with digital HW.	Straightforward and networkable.
14. Metric drifting	If any analog components, it is likely; if digital, unlikely.	Unlikely.
15. User metadata	No support for full-featured metadata on VTRs and many other components.	Metadata are of primary importance and support is common.
16. Systems management	Not mature and lacking in standards and functionality.	Very mature for general IT devices.
17. Live A/V switching (e.g., camera selection)	Commonly done to produce live sports, news, and drama events.	Not common in 2009 due to the difficulty in live switching of a LAN signal frame accurately.

provides a high-level overview of what features the different systems can provide and describes their “sweet spots.” Note that traditional A/V systems can and often are composed of a predominance of digital components (or analog + digital mix), but they lack the full-featured networking and infrastructure of IT-based solutions.

## 1.6 ADVANTAGES AND DISADVANTAGES OF METHODS

### 1.6.1 Trade-off Discussion

Table 1.1 lists several “sweet spots” for AV/IT, namely No. 6, 8, 10, 11, 12, 13, 15, and 16.

- **No. 6.** Long delays are ideally implemented with IT storage systems. Long delays may be needed for time delay applications, censorship, or other needs.
- **No. 8.** Digital systems may be designed to be noise and artifact free with the highest level of repeatable quality.
- **No. 10.** See later.
- **No. 11.** Systems can span a campus, city, country, or the world without affecting video quality. There are countless such systems in use today.
- **No. 12.** IT storage methods are accepted for video servers and are in use daily in media production and distribution facilities. For all but deep archives, this means the demise of the VTR (tape free at last) for most record/replay operations. Tape is still in common use for field camera acquisition, but even this is waning.
- **No. 13.** There are no geographic constraints on where the A/V processors may be located. Clusters of render farms, codecs, effects processors, format converters, and so on may be networked to form a virtual utility pool of A/V processing.
- **No. 15.** User-descriptive metadata importing, storage, indexing, querying, and exporting are considered a cornerstone feature of IT systems. Chapter 7 investigates metadata standards and utilization in more detail.
- **No. 16.** IT systems management has a major advantage over the ad hoc methods in current use today (see Chapter 9).

Number 10 has the IT edge too for many applications. Traditional device control has relied on the sturdy RS422 serial link carrying device commands. Sure, it is trustworthy and reliable, but it is not networkable, is yet another link to manage and route, and has limited reach. LAN-based control is the natural replacement. There is one concern with LAN replacing RS422 serial linking: determinism. RS422 routing is normally a direct connection from controller to

device under control, and its QoS is excellent and proven. As has been the case, a video server with 10 I/O ports has 10 RS422 control ports! This is messy and inefficient in terms of control logic, but it is proven. However, a LAN may be routed through a shared network with at least some small delay. A *single* LAN connection to a device may control all aspects of its operation with virtually no geographic constraints. In fact, it is not uncommon to control distributed devices from a central location using LAN and WAN connectivity. This cannot be done using a purely RS422 control and reporting system. There are several ways to deal with the networking delay and jitter, which are covered in detail in Chapter 7.

Number 17 in Table 1.1 is one area where traditional A/V methods are superior. This scenario assumes that event cameras have a LAN-like output that carries the live video signal. The signals are fed into a LAN-based switcher operated by a technical director where camera feeds are selected and processed for output. There are no accepted standards for frame-accurately aligning the isolated camera output signals over a LAN. Furthermore, there are no commercial LAN-based (for I/O) video switchers. Most professional cameras use a coax Triax link (or alternative fiber optic link for HD feeds) from camera to the production control room.

The missing link for live camera production using IT networking is time-synchronous Ethernet. Current Ethernet uses an asynchronous timing model. The IEEE 802.1 Audio/Video Bridging Task Group (LANs/MANs) is developing three specifications that will enable time-synchronized, excellent QoS, low latency A/V streaming services through 802 networks. These are referenced as 802.1AS, 802.1Qat, and 802.1Qav ([www.ieee802.org/1/pages/avbridges.html](http://www.ieee802.org/1/pages/avbridges.html)). Once these standards are mature (sometime in 2009 likely), expect vendors to offer A/V devices with time-synchronous Ethernet ports. When this happens and it is embraced by our industry, traditional video links will start to wane. How long before the stalwart SDI link goes the way of composite video link for professional systems?

Note too that Ethernet switches (MAC layer 2) have less I/O jitter and delay compared to most (IP layer 3) routers. Ethernet frames (1,500B normally) may be switched with less delay than a 64 KB IP packet. So, for live event switching, layer 2 may be applied with better results than layer 3. However, IP routing offers more range and scale, so methods to support both layers 2 and 3 are being developed. See Appendix M for more insights.

Some researchers have built test systems for live event production that are almost entirely LAN/WAN and IT based. The European Nuggets (Devlin et al. 2004) project developed a proof of concept live production IT-based system. In their demo system, camera control, streaming A/V over networks, metadata management, MXF, live proxies, and camera switching are all folded into one comprehensive demo. Their work is on the bleeding edge of using IT techniques for live production. The Nuggets effort is a good indicator of the promise of IT-based live production.

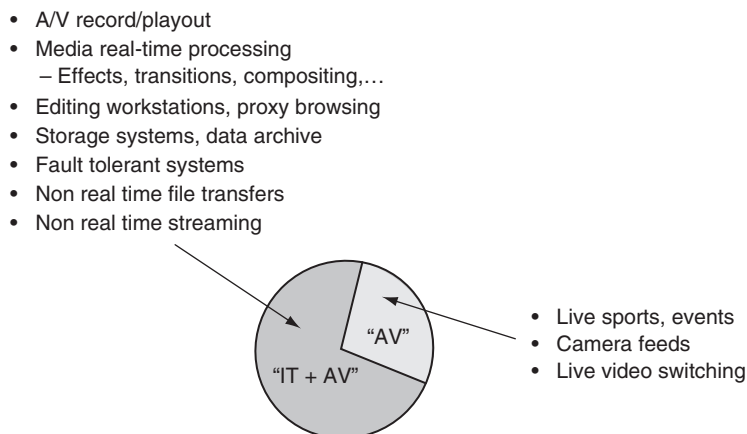


Another effort to leverage IT in postproduction is MUPPITS, or Multiple User Post-Production IT Services. This group brings together key players in the UK postproduction value-chain to investigate, develop, and demonstrate a new service-oriented approach to film and broadcast postproduction. See [www.muppits.org.uk](http://www.muppits.org.uk).

As technology matures, it seems likely that video systems for any set of user requirements can and will be implemented using IT methods. Admittedly, it may take many years for live event HD production to migrate to an all-IT environment. Despite the latency drawback, most other video system application requirements are easily achieved today with AV/IT configurations. See Figure 1.15 for the production sweet spots for traditional A/V versus AV/IT in 2009 and likely sometime beyond. Note too that the AV + IT portion of the pie is file based for virtually all operations. Even non-RT streaming has a file origination and end point in most cases.

For a traditional TV station, 90 percent of daily operations have little need for live stream switching. Using non-real-time file transfers to move A/V files can replace SDI in many cases. Sure, live news with field reporting requires video switching. In practice, most station operations can use a mix of AV + IT gear. However, ESPN's HD Digital Center facility has 10 million A/V cross points in its router infrastructure. Their lifeblood is live events. In this case, because the need to switch streams under human control is great, traditional SDI is required in bulk. Still, the ESPN facility has its share of IT elements (Hobson).

Traditional A/V has one added benefit not cited in Table 1.1: familiarity. The engineers, technicians, and staff responsible for the care and feeding of the media infrastructure may have many years of experience. Moving to an AV/IT infrastructure requires new skills and changes in thinking. Some staff may resist or find the change uncomfortable. Others will welcome the change



**FIGURE 1.15** Sweet spots for traditional and hybrid A/V systems.

and embrace it as progress and improvement. Change management is always a challenge. We now have some track record of facilities that made the switch. Some are broadcasters who made the switch to an IT infrastructure while on air, whereas others had the advantage of building a new “green field” facility where existing operations (if any) were not of concern. The challenges and rewards of building these new systems are reviewed in Chapter 10.

## 1.7 IT'S A WRAP: SOME FINAL WORDS

So can AV/IT meet the challenge of replacing (and improving) the traditional A/V infrastructure? For all but a few areas of operations, the answer is a resounding yes! There is every reason to believe that IT methods will eventually become the bedrock of all media operations. True, there will always be a few cases in which traditional A/V still has an edge, but IT has a momentum that is difficult to derail. Do not let a corner case become the driving decision not to consider IT. The words of the brilliant Charles Kettering (GM research chief) seem truer today than when he spoke them in 1929: “*Advancing waves of other people’s progress sweep over unchanging man and wash him out.*” Do not get washed out but seek to understand the waves of IT that are now crashing on the shores of traditional A/V.

Now that we have established the beneficial aspects behind the move to IT, let us move to expand on the concepts outlined in this chapter. The next chapter reviews the basics of networked media and file-based techniques as related to A/V systems. The chapters that follow it will provide yet more insights and explanations for the major themes in IT as related to A/V systems. As Winston Churchill once said, “Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.”

## REFERENCES

- Chen, X., *Transporting Compressed Digital Video*, Kluwer Academic Publishers, (2002). Chapter 4.
- Gilder, G., *Metcalfe’s Law and Legacy*, Forbes ASAP, (September 13, 1993).
- J. Hennessy, D. Patterson, *Computer Architectures*, 3rd edition, 2003, page 15, Morgan Kaufmann.
- Hobson, E., and Szypulski, T., *The Design and Construction of ESPN’s HD Digital Center in Bristol, Conn.* SMPTE: Technical Conference Pasadena. 2004.
- Moore, G. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 19, 1965.
- Morris, R. J. T., and Truskowski, B. J. The evolution of storage systems. *IBM Systems Journal*, 42(2), July 2003.
- Devlin, B., Heber, H., Lacotte, J. P., Ritter, U., van Rooy, J., Ruppel, W., and Viollet, J. P. Nuggets and MXF: Making the networked studio a reality. *SMPTE Motion Imaging Journal*, July/August 2004.

This page intentionally left blank

# The Basics of Professional Networked Media

## CONTENTS

2.0	Introduction	34
2.1	Core Elements	35
2.1.1	The Application Client	36
2.1.2	The Router	36
2.1.3	The Ethernet Switch	37
2.1.4	The Firewall and Intrusion Prevention System	38
2.1.5	File and Application Servers	38
2.1.6	Storage Subsystems	39
2.1.7	Networking Infrastructure	39
2.1.8	Systems Management and Device Control	40
2.1.9	Software for All Seasons	41
2.2	Standards	41
2.2.1	General Scope of Technology Committees	43
2.2.2	The Eight SMPTE Technology Committees	43
2.3	A/V Media Clients	44
2.3.1	Class 1 Media Client	44
2.3.2	Class 2 Media Client	45
2.3.3	Class 3 Media Client	46
2.3.4	Class 4 Media Client	47
2.3.5	The Classes in Perspective	48
2.4	File Transfer, Streaming, and Direct-to-Storage Concepts	48
2.4.1	File Transfer Concepts	50
2.4.2	Streaming Concepts	62
2.4.3	Direct-to-Storage Concepts	68
2.5	The Three Planes	71

<b>2.6</b>	<b>Interoperability Domains</b>	<b>72</b>
2.6.1	Domains 1 and 2: Streaming and File Transfer Interfaces	73
2.6.2	Domain 3: Control Interface	74
2.6.3	Domain 4: Management Interface	74
2.6.4	Domain 5: Storage Subsystem Interface	74
2.6.5	Domains 6 and 7: Wrappers and Essence Formats	74
2.6.6	Interop Conclusions	75
2.7	Tricks for Making IT Elements Work in Real Time	75
2.8	Using IT Methods to Route Traditional A/V Signals	78
2.9	It's a Wrap: A Few Final Words	80
	References	80

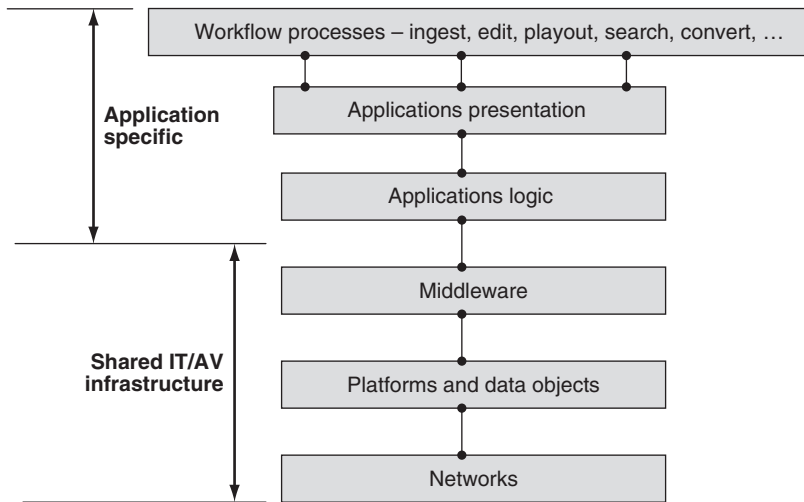
## 2.0 INTRODUCTION

This chapter reviews the essential elements of networked media as they relate to the professional production of A/V-based materials. Of course, it is not possible to present a detailed explanation of each building block, so let us call the coverage an “evaluation strength” treatment. What is this? If you need to evaluate various HW/SW architectures, system components, or systems issues, the coverage in this chapter (and others) provides you with the tutoring to ask the right questions. Rudyard Kipling said (paraphrased), “I keep six honest friends—what, why, when, how, where and who.” In the end, you will be able to ask probing and intelligent questions when evaluating and specifying AV/IT systems.

This section does a broad-brush coverage, while the remainder of the book dissects the same subjects to uncover their subtleties and deeper points. The following chapters probe deeper into select subjects:

- Chapters 3A and 3B—Storage systems
- Chapter 4—Software technology for A/V systems
- Chapter 5—Reliability and scalability methods
- Chapter 6—Networking basics for A/V
- Chapter 7—Media systems integration
- Chapter 8—Security for networked A/V systems
- Chapter 9—Systems management and monitoring
- Chapter 10—The transition to IT: Issues and case studies
- Chapter 11—A review of A/V basics

At the highest level is generic IT architecture. Figure 2.1 shows this six-tier architecture. It is deliberately abstract, as it can represent almost any process-oriented workflow. Networked media, as defined for our purposes, is more than pure networking. It encompasses all the stages in Figure 2.1. The diagram is split into two domains; the application-specific one and the shared IT infrastructure.



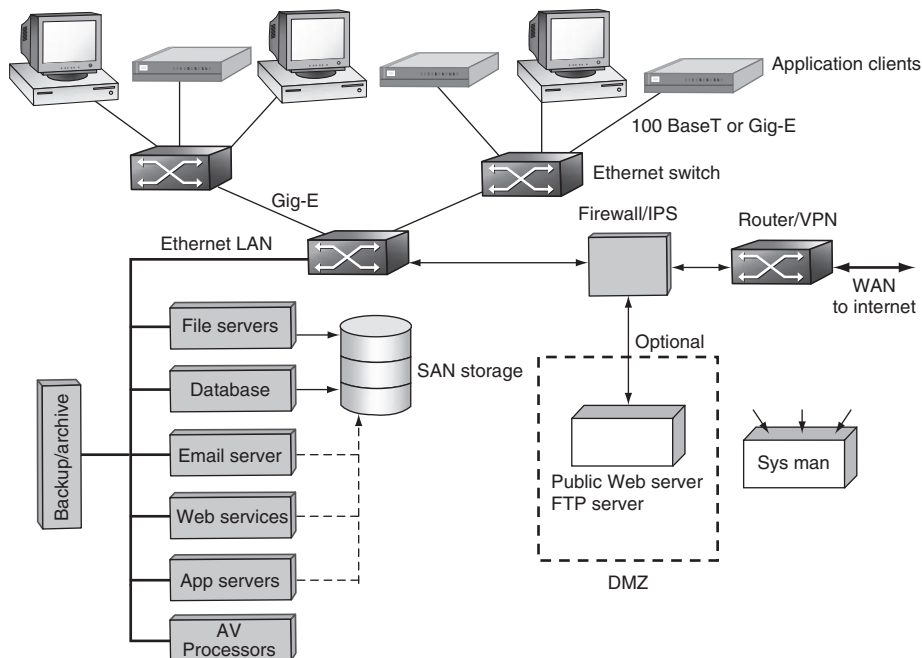
**FIGURE 2.1** *Generic six-layer IT architecture.*

The application layers define the various applications needed to support a workflow. The lower layers provide the services and resources that the application logic calls upon. Next, let us peel back the onion on the bottom three layers in Figure 2.1. The higher layers are considered in other chapters.

## 2.1 CORE ELEMENTS

Figure 2.2 provides a 5-mile high view of a generic IT architecture for the enterprise. This is the physical representation of the logical view of the three lower layers in Figure 2.1, along with the application clients. Also, most of the infrastructure in Figure 2.2 would be found in a typical hybrid AV/IT architecture. Missing from Figure 2.2 are miscellaneous A/V links, video/audio routers, cameras, VTRs, logo inserters, video servers, and so on. IT architectures that are fine-tuned for A/V are discussed in other chapters. Nonetheless, Figure 2.2 forms the foundational elements for our discussions throughout the book. The main elements to be discussed here are

- Application clients
- The router
- Ethernet switching
- The firewall and intrusion prevention system
- File and application servers
- Storage subsystems
- Network infrastructure (LANs, WANs)
- Systems management and device control
- Software for all seasons



**FIGURE 2.2** *Simplified enterprise network architecture.*

### 2.1.1 The Application Client

Clients come in all shapes and flavors. The application clients perform A/V I/O, video editing and compositing, browsing, other media-related functions, or standard enterprise applications. This book uses the term *application client* in a general way, implying some device that accesses storage or servers or other systemwide resources. If the client does A/V processing, then it may or may not have A/V I/O ports. For example, a nonlinear editor (NLE) may have A/V file access from a storage element in the network but it may not have physical A/V I/O ports. Another client may support only A/V I/O and not any human interface. This type of element is common for capturing content from live satellite feeds or as an A/V playout device. The network attach may be an Ethernet port at 100 Mbps, Gigabit Ethernet (Gig-E), or Fibre Channel. Why choose one over the other? These and other client-related aspects are discussed later in this chapter. See, too, Section 2.2, “A/V Media Clients,” later in this chapter.

### 2.1.2 The Router

The router connects the facility to the outside world or to other company sites using IP routing. A router has three fundamental jobs. The first is to compute the best path that a packet should take through the network to its destination. This computation accounts for various policies and network constraints. The

second job of the router is to forward packets received on an input interface to the appropriate output interface for transmission across the network. Routers offer a selection of WAN interfaces<sup>1</sup> (SONET, T1/E1, T3/E3, ATM, Frame Relay, ISDN, and more) and LAN (Ethernet with IP for our discussions) interfaces. The third major router function is to temporarily store packets in large buffer memories to absorb the bursts and temporary congestion that frequently occur and to queue the packets using a priority-weighted scheme for transmission. Some routers also support a virtual private network (VPN) function for the secure tunneling of packets over the public Internet.

### 2.1.3 The Ethernet Switch

Most campus networks are composed of a cascade of Ethernet switches. The switching may occur at the Ethernet level (layer 2) or the IP level (layer 3), although the distinction is not enforced here. Think of the switch as a *subset* of a full-fledged router; it is simpler and less costly than routers, although there are numerous technical differences too. Layer 2 and 3 switching and routing methods are covered in more detail in Chapter 6. Switches are the tissue of an enterprise network. Most switches can support “wire speed” packet forwarding at the Ethernet line rate (100 Mbps or 1 Gbps commonly). Small Ethernet switches (24 ports) are inexpensive with a per port cost of only \$15.

Packet switches (and routers) have throughput latency that is ideally in the 1- to 20- $\mu$ s range but can grow much bigger in the presence of packet congestion. IP switches may be classed as asynchronous devices, whereas SDI A/V routers are isochronous (equal timed) in nature with perfectly timed (fixed latency, very small jitter, and bit accurate timing) I/O ports. So it is apparent that routing A/V packets is not exactly equivalent to SDI routing. It is precisely this issue that convinces some that IP switches cannot be used to route live A/V signals. After all, if live video needs to be perfectly timed to within a nano-second or so, then the IP switch is not fit for the job. This issue is addressed later in the chapter with the conclusion that switches (and routers) can indeed be used to switch live A/V packets.

The switch internal data routing structure is either shared memory or switch fabric and can reach speeds of 400 Gbps non-blocking throughput for a carrier-class large size switch (200 ports). Terabit per second fabrics exist and are used in very high-end switches and routers. See [www.avici.com](http://www.avici.com) and [www.juniper.com](http://www.juniper.com) for added insights. Like routers, they can support small I/O latencies in the 1- to 20- $\mu$ s range, which is ideal for moving live A/V streams.

A switch is deceptively simple from the connectivity point of view; it is just Ethernet I/O. Looking deeper, we see that it must support 20 or more networking standards for efficient packet forwarding, Ethernet frame forwarding,

---

<sup>1</sup> See Appendix F for more information on WANs.



secure device management, flow control, VLAN support, class of service, packet inspection, multicast, and a multitude of other protocols. Routers are more sophisticated and may support 35+ different protocols. There are many vendors to choose from, and the price/performance ratio is improving with each generation thanks again to Moore's law.

#### 2.1.4 The Firewall and Intrusion Prevention System

A computer firewall protects private networks and their internal nodes from malicious external intrusion, resulting in a compromise of system or data integrity or a denial of service. It is usually a secure HW/SW device that acts to filter every packet that transits between a secure and unsecured network. It must have at least two network interfaces: one for the network it is intended to protect and one for the risky network—such as the Internet. The earliest computer firewalls were simple routers. The term *firewall* comes from the fact that when a network is segmented into different parts, damage that could spread from one subnet to another is stopped—just as fire doors or firewalls stop a fire.

An Internet firewall examines all traffic routed between a private network and the Internet. Every packet must meet a preselected set of criteria; otherwise, it is dropped. A network firewall filters both inbound and outbound traffic. Disallowing internal access to select external locations is vital in guaranteeing secure and legitimate business operations. It can log all attempts to enter the private network and trigger alarms when hostile or unauthorized entry is attempted. Firewalls can filter packets based on their source, destination addresses, or port numbers. This is known as address filtering. Firewalls can also filter based on application layer protocols such as HTTP, FTP, or Telnet.

The Intrusion Prevention System (IPS) is a sort of super firewall. It filters at the content layer. For example, it may look for and prevent attacks from incoming worms, ill-formed protocols, and other higher layer tricks. A more detailed coverage is outlined in Chapter 8.

#### 2.1.5 File and Application Servers

Servers come in many forms and are a common element in an IT infrastructure. Their basic function is to host software applications that clients can access over the network. These functions may be done in an A/V real-time sense or in non-real time. The common functions are shown in Figure 2.2. For the purposes of A/V functions, the following are of interest:

- **File server**—This type of server stores/retrieves A/V files for access by clients. Normally, the delivery of files over the network would be in non-real-time. The server usually appears as a networked drive (//K: or //MyServer:) to the client. Files can be moved to and from application clients.
- **A/V processor**—This networked resource is used for processing A/V essence. Typical functions are compressing/decompressing, file format

conversion (DV to MPEG), 3D effects, proxy file video generation, and more. Most often, the processing is not in real time, but live streams may also be supported.

- **File gateway**—This is the boundary point for transferring files to and from external sources. A gateway may do file conversion, proxy video file generation, bandwidth management, and more.
- Control and scheduling, media asset management (MAM), element management, and other services offered on a server platform.

See Chapter 3A for more details on servers and their architecture.

### 2.1.6 Storage Subsystems

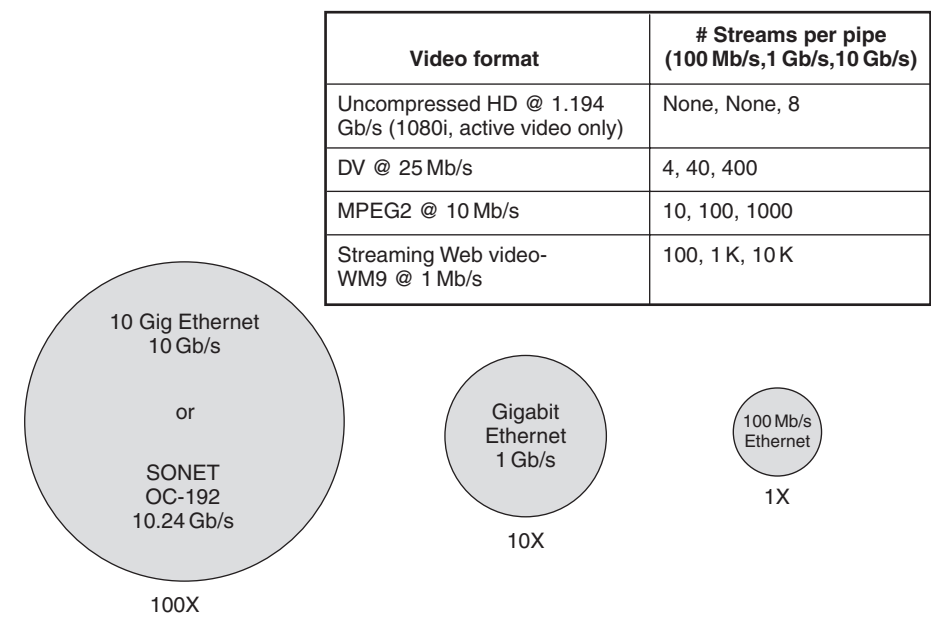
Many enterprises use SAN-based storage to reduce the number of independent storage systems. A SAN provides for a *single storage pool* to be shared by different nodes. For media centric IT systems, there is often the need for huge amounts of high-availability, *real-time* storage. Terabytes of online storage (thousands of hours of standard definition video) with aggregate bandwidths of gigabits per second are not uncommon for large installations. The *real-time* spec is a frequent requirement for systems that read/write live A/V to storage. SAN and NAS storage systems are discussed in Chapter 3B. Additionally, there is discussion about non-real-time and real-time storage QoS and their application spaces.

### 2.1.7 Networking Infrastructure

The Ethernet LANs in Figure 2.2 connect to create networks of virtually any size. These systems are mature with powerful features, such as

- Scalability from a few networked elements to thousands.
- Scalability to any required data throughput.
- Reliability “to the budget”; i.e., the reliability will be only as good as you can afford.
- LANs can be configured with virtually 100 percent uptime for the core routing and switching components. QoS per link can be defined offering an overall media-friendly infrastructure.
- LAN segmentation for building virtual network islands for media production intelligently isolated from the normal IT company operations.
- Network management of all components. Faults, warnings, status, and systems configuration information readily available from intuitive GUIs.
- WAN connectivity for wide area distribution of streams and files.

Of course, a LAN is more than Ethernet connectivity. A smorgasbord of protocols run on the Ethernet links, such as IP, TCP, UDP, HTTP, IPSec, and many



**FIGURE 2.3** *Filling LAN/WAN pipes with video streams.  
Not to scale, ideal, no overhead.*

more. They are mature and used universally in a variety of real-world environments. These protocols are discussed in Chapter 6.

Throughout the book, networks are used as transport for A/V streams and files. Figure 2.3 illustrates how many simultaneous, real-time video streams of different rates can fit into the listed pipes. At the top of Figure 2.3 is uncompressed 1080i HD (see Glossary) at 1.194 Gbps image payload data rate. Standard Gigabit Ethernet cannot carry even one stream of this format. The bottom of Figure 2.3 shows that SONET OC-192 can carry ~10,000 streams of Web video (see Appendix F). The packing densities are ideal, and a realistic packing may be 70 percent of these values due to a variety of link, transport, and format overheads.

Compression increases packing density greatly. Fully uncompressed digital cinema at so-called 4 K × 2 K resolution reaches ~9.6 Gbps, and some vendors use exotic InfiniBand links to move the images. Using MPEG2 compression, the same program can fit nicely into a 19.3-Mbps ATSC pipe. Newer HD compression methods can squeeze even more, reaching ~8 Mbps—a whopping 1,200:1 bit savings, albeit with a loss of the pristine quality.

2.1.8 Systems Management and Device Control

Element management methods are de rigueur; otherwise, chaos would reign in the IT center. Chapter 9 covers systems management but especially in the context of AV/IT systems. A/V device control is not commonly practiced in a

traditional IT center. However, the world of scheduled broadcast TV requires the coordinated recording and playback of many video frame-accurate signals. As such, a dedicated class of application called *automation* is commonly applied to this end. Device automation is considered in Chapter 7.

### 2.1.9 Software for All Seasons

At every layer in Figure 2.2, software plays a major role. The software may be classified into the following domains:

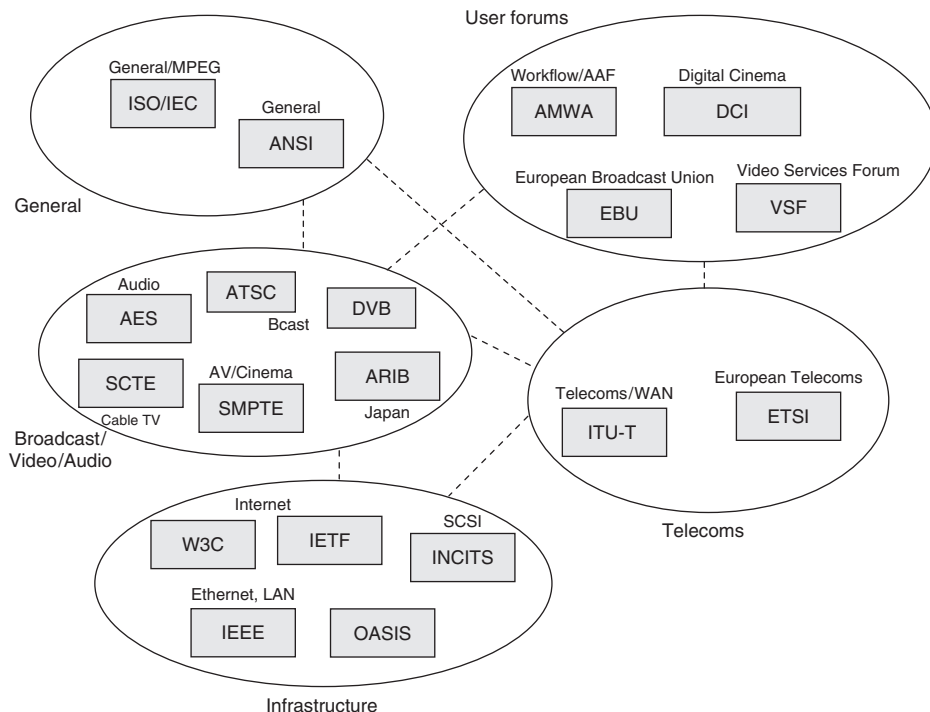
- Application and services related
- Operating systems (OS) for clients, servers, and other devices
- Middleware protocols for client and server communications
- Web services
- Programming frameworks

These technologies work hand in hand to complete a solution. These domains are discussed in Chapter 4. See (Britton) for more information too.

## 2.2 STANDARDS

It has been said that the nice thing about standards is that there are so many to choose from. So true—and for good reason. Without standards—and lots of them—interoperability would be hopeless. Figure 2.2 could never exist (with heterogeneous components) without standards. Based on our personal experience, sitting through hundreds of hours of debates and the due diligence of standards development, we know the pain is worth the gain. Figure 2.4 outlines a short list of standards bodies, user groups, and industry associations that are active in both IT and A/V spaces.

The mother of all standards bodies for A/V (broadcast, post, and cinema) is the Society of Motion Picture and Television Engineers ([www.smpite.org](http://www.smpite.org)). The Audio Engineering Society (AES) also contributes a significant effort. The International Telecommunications Union (ITU) is the world's largest telecom body, and the ISO/IEC (two separate bodies that cooperate) has developed many standards, with MPEG and JPEG being the most significant for our space. The European Telecommunications Standards Institute (ETSI) works in fields that are germane to European interests, such as the DVB broadcast standards in association with the EBU. The IEEE and Internet Engineering Task Force (IETF) have developed thousands of standards (request for comments, RFCs) for networking interoperability. The W3C contributes important recommendations, such as XML, HTML, SVG, MathML, SMIL, Timed Text, and Web services. Among user groups, the Advanced Media Workflow Association (AMWA) is responsible for the Advanced Authoring Format (AAF) and is also creating recommendations for flexible media workflows using AAF, MXF, Web services, and SOA principles.



**FIGURE 2.4** *World of AV/IT standards.*

One new activity, sponsored by the W3C, is “Video in the Web.” The goal of this activity is to make video a “first class citizen” of the Web. The group has goals of building an architectural foundation that enables the creation, navigation, searching, linking, and distribution of video, effectively making video an integral part of the Web. Today, video is an afterthought, a dangling object forced to fit as needed. One goal is to standardize methods to identify spatial and temporal clips. Having global identifiers for clips would allow substantial benefits, including linking, bookmarking, caching, indexing, and displaying associated metadata. Not one of the existing solutions is fully satisfactory.

Many of these bodies have liaison connections to other bodies, and the dashed lines in Figure 2.4 indicate these relationships. For example, every SMPTE standard may have a corresponding ANSI (U.S., master body) standard. Normally, user groups have no power to set standards but they can make recommendations that sometimes become de facto standards, much like what AAF has become for edit decision list exchange. Not all liaison connections are shown in Figure 2.4.

Standards are the fabric that hold modern IT systems together. If you want to know more, most of the standards bodies have Web sites in the form of [www.NAME.org](http://www.NAME.org). Some of the standard documents are free for the asking (such

as the IETF; learn about TCP, for example, by downloading RFC 793), whereas others require payment (such as SMPTE and the ITU) per document. Many professionals in the broadcast and cinema industry subscribe to the SMPTE CD-ROM, a collection of all their standards for easy access.<sup>2</sup>

Of all the standards groups, SMPTE is the most active in video systems and digital cinema standardization. To keep in tune with SMPTE activities and receive its excellent *Imaging Journal*, consider becoming a member ([www.smpte.org](http://www.smpte.org)). The following is a summary of the work efforts (obtained from SMPTE documents) of the technology committees that develop standards.

### 2.2.1 General Scope of Technology Committees

To develop SMPTE engineering documents, review existing documents to ensure that they are current with established engineering practices and are compatible with international engineering documents where possible; recommend and develop test specifications, methods, and materials; and prepare tutorial material on engineering subjects for publication in the SMPTE *Imaging Journal* or for other means of dissemination benefiting the Society and the industry.

### 2.2.2 The Eight SMPTE Technology Committees

**Essence 10E**—The general scope as it applies to electronic capture, generation, editing, mastering, archiving, and reproduction of image, audio, subtitles, captions, and any other master elements required for distribution across multiple applications.

#### Applications Committees:

**Film 20F**—The general scope as it applies to application of mastered essence to theatrical film distribution, including media and component creation, marking, laboratory methods, reproduction, packaging, projection, and related topics; additionally, film capture, editing, and recording.

**D-Cinema 21DC**—The general scope as it applies to application of mastered essence to theatrical digital distribution, including compression, encryption, wrapping, marking, packaging, media, logging, playout, projection, reproduction, and related topics.

**Television 22TV**—The general scope as it applies to application of mastered essence to television distribution, including compression, encryption, wrapping, marking, packaging, media, control, display presentation, reproduction, and related topics.

**Broadband 23B**—Application of mastered essence to electronic broadband distribution including compression, encryption, wrapping, marking, packaging, tracking/control, presentation, reproduction, and related topics.

---

<sup>2</sup> For an excellent table summary of professional digital tape and compression standards, visit [www.smpte.org/smpte\\_store/standards](http://www.smpte.org/smpte_store/standards).

**Infrastructure Committees:**

**Metadata/Registries 30MR**—The general scope as it applies to definition and implementation of the SMPTE Registration Authority, used to identify digital assets and associated metadata. Additionally, the common definition of metadata semantic meaning across multiple committees.

**Files Structures 31FS**—The general scope as it applies to definition of common wrapper structures for storage, transmission, and use in the carriage of all forms of digital content components.

**Network/Facilities Infrastructure 32NF**—The general scope as it applies to definition and control of elements supporting the infrastructures of content production and distribution facilities, including file management, transfer protocols, switching mechanisms, and physical networks that are both internal and external to the facility, excluding unique final distribution methods.

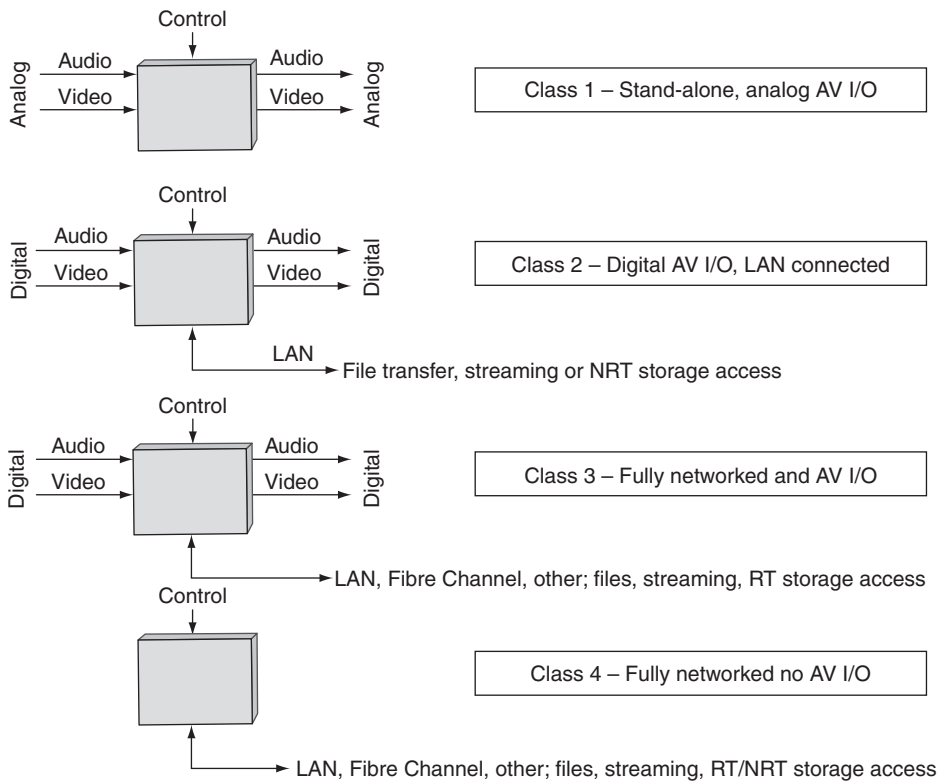
## 2.3 A/V MEDIA CLIENTS

Looking again at Figure 2.2, let us morph it slightly to give it a more media centric personality. Without changing anything in Figure 2.2, we can add the notion of A/V by defining the *A/V application client*. So, consider the top level to be a mix of A/V clients and non-A/V clients. The necessary traditional A/V routing infrastructure is not shown in Figure 2.2 and is left out to simplify the overall diagram. See Chapter 10 for a discussion of a more dedicated, full-featured, AV/IT system based on the principles developed in this book.

Broadly, there are four different classes of A/V client. Figure 2.5 shows their general I/O characteristics. The I/O notation is abbreviated, as time code signals and video reference inputs are omitted because they are not relevant to our immediate discussion. Also, the definitions are not meant to be rigid. Other hybrid combinations will exist. For example, clients with digital A/V inputs but no outputs or vice versa (ingest stations, cameras, etc.) are not specifically shown. The point of the four classifications is to discern the major client types and their characteristics. Let us consider each class and its applications. Also, no one class is superior to another class. Each has its strengths and weaknesses depending on the intended application. Some clients also have management interfaces, but these are not shown to simplify the diagrams.

### 2.3.1 Class 1 Media Client

The simplest class, class 1, is shown for completeness. It is all analog I/O (composite, component video), standalone with no network connection. It is the legacy class of A/V client and includes VTRs, analog amplifiers, linear switchers for edit suites, master control stations, special effects devices, and so on. This class has gone out of favor; it does not take advantage of digital or networked advantages. Still, there are many facilities with rooms filled with these devices in legacy A/V systems.



**FIGURE 2.5** Four classes of media clients.

### 2.3.2 Class 2 Media Client

Class 2 is distinguished with the inclusion of coax-based serial digital interface (SDI) I/O for A/V connectivity and LAN connectivity. The LAN port may be used for file exchange and A/V streaming but *not* access to networkable RT storage by definition. Support for device-internal storage is common. Audio I/O is also included using the AES/EBU interface.

A sample of this class of client is the Sony IMX eVTR, which is an MPEG-based tape deck with SDI I/O and a LAN for MXF file export. Another example is the Panasonic P2 Camera with LAN (and removable Flash memory card) for exporting stored clips. Consider, too, the stalwart “small” (or configured as such) video server. Doremi Lab’s V1 server, Harris’s Nexio AMP, Avid’s Thunder, Omneon’s MediaDeck, and Thomson GVG’s M-Series iVDR are all class 2 devices with internal (or bundled) storage. Graphics devices such as Chyron’s Duet family, Insciber’s Inca family, and Avid’s Deko family are also examples of class 2 devices. True, some of them may also be optionally attached to external real-time storage. In this case, these devices are considered class 3 devices (see Appendix J for more information on the workhorse video server).



Other examples of class 2 devices are the countless models of standalone nonlinear editors (NLE) from many vendors. They normally have digital I/O and LAN to access files over a network. Most A/V test equipment is of class 2 but may not have any A/V output ports. The class 2 device was the first to bridge the traditional A/V and IT worlds back in the mid 1990s. At the time, it was groundbreaking A/V technology.

### 2.3.3 Class 3 Media Client

Class 3 is a fully networked device; it is a turbocharged class 2. It has not only SDI I/O, but also real-time access to external networkable storage. Of course, it also has LAN access for file exchange and streaming. The access to real-time external storage is the major differentiating characteristic of this class compared to class 2. Storage connectivity may use one or more of the following methods with current top data rates:

1. Fibre Channel (up to 2/4 Gbps per link)—SAN related.
2. LAN using Ethernet (up to 10 Gbps)—NAS and SAN related.
3. IEEE 1394 and USB 2.0 for connecting to small, local storage systems.
4. This mode offers limited network access to storage.

These methods are discussed in greater detail in Chapter 3A. Methods 1 and 2 are in heavy use in the IT infrastructure and are crucial to building any AV/IT system. IEEE 1394 and USB 2.0 are limited in their range and networkability, but they find niche applications for low-cost clients.

In theory, networked clients can access storage from any location. This is a powerful feature, and it enables A/V I/O to be placed in the most convenient location independent of storage location. This freeness of location has its trade-offs: an excellent link QoS is required to connect the client with the storage. Nonetheless, some systems will take advantage of this leverage point and more so as technology matures. Some class 3 devices also have internal HDD storage.

A/V data storage systems are core to the new media facility. There are many ways to construct these systems, they come in a variety of forms, and they allow for whole new workflows that were impossible just a few years ago. These aspects are discussed in Chapters 3A, 3B, and 7. Of course, relying on data storage systems to keep all A/V digital assets represents a major shift in operations: no more videotapes to manage and no more lost digital assets (if managed properly). There are several schools of thought regarding storage access by a client or server. Some of the methods are

- A. Networkable RT storage pool available to all authorized attached clients. Classes 3 and 4 use this.
- B. Networkable NRT storage pool available to all authorized attached clients. NRT storage has a relaxed QoS compared to RT storage. Classes 2 and 4 use this.

- C. Islands of NRT storage. This is like B except there is not one consolidated pool of storage.

Method A is used with class 3 and 4 clients. Methods B and C are more appropriate for class 2 and 4 devices but may also apply to class 3. In Method C, files may reside on various file servers or arrays throughout a facility. If this occurs, then the asset management may become a serious nightmare, due to the balkanization of resources. In fact, most modern systems use either A or B for their design methodology.

The distinction between RT and NRT is important in several regards. RT storage is used by clients for streaming A/V processes such as recording and playout. NRT is used by clients for file transfer, offline A/V processing, meta-data access, database access, and so on.

Large multichannel video server I/O devices are good examples of class 3 clients. Instances are Avid's AirSpeed, Harris's Nexio AMP server, Omneon's Spectrum Media Server, the SeaChange Media Client attached to its Broadcast Media Library, and Thomson/GVG's K2 media client/server system.

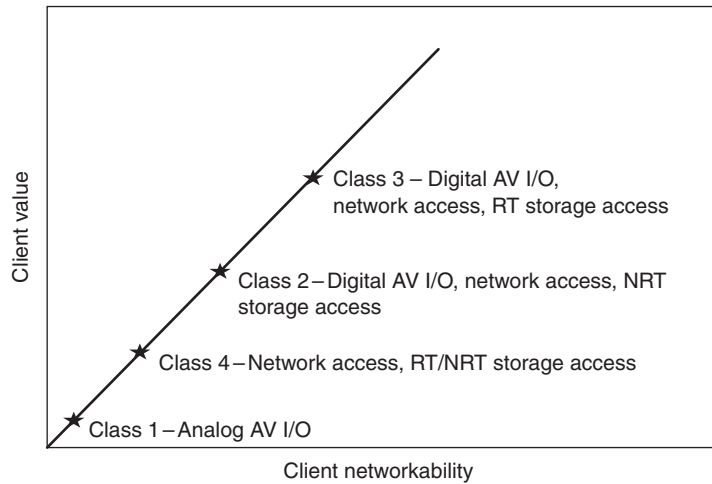
Other examples are networked nonlinear editors: Apple's Final Cut Pro (Final Cut Studio too), Avid's Media Composer family, Quantel's family of sQ editors, and others. This class of media client offers the most flexibility compared to the other classes.

The digital audio workstation (DAWS) has unique requirements as a class 3 device. Many such devices (ProTools from Avid, or Logic from Apple, for example) support 192 simultaneous audio tracks at 48 KHz. This requires 26 MBps raw throughput. For sure, one HDD can support this rate. However, when it is divided among 192 file accesses, this is problematic due to possible small file chunk R/W and possible fragmentation issues. The bottom line is that a DAWS places a demanding I/O workload on a typical storage system.

### 2.3.4 Class 4 Media Client

Class 4 is a class 3 or 2 device without any A/V digital I/O. This client type is a fully networked station that does not have a need to ingest or play out A/V materials. Some class 4 stations may use NRT file transfer for importing/exporting files. Other stations may access a RT storage pool, whereas others may support streaming.

Examples of this client type include visual content browsers (low-rez normally), reduced functionality NLE stations, graphics authoring (Adobe Illustrator and Photoshop, for example) QA stations, asset managers, file gateways, file format converters, some storage, DRM authoring stations, file distribution schedulers, A/V processors, and displays. One special version of this client type is a control device that sends operational commands to other clients to record, play, route, convert, retrieve, and so on. Importantly, no A/V data pass through a control-type media client. Automation vendors offer a variety of media client control devices.



**FIGURE 2.6** *The relative value of media clients.*

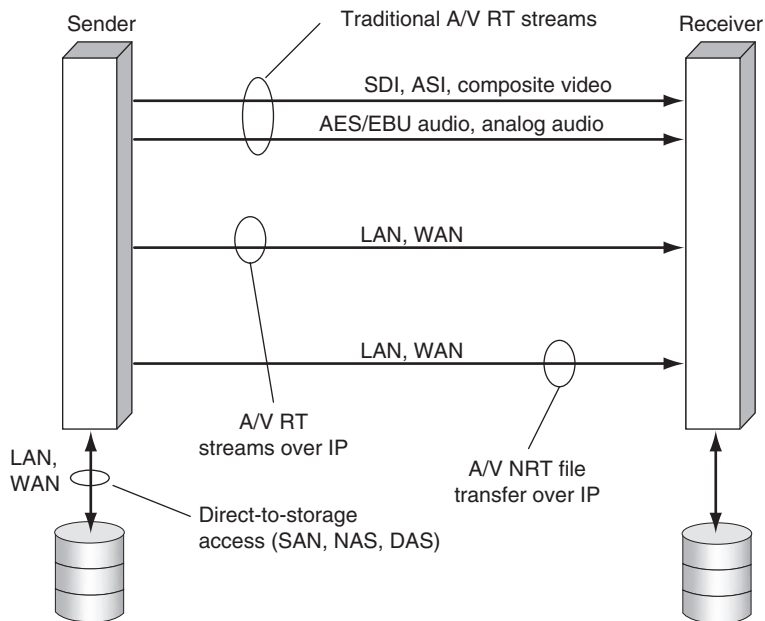
Another special case of this class is a software service such as a Web service. The service (with no UI) may perform a data conversion of some nature. This is covered in more detail in Chapter 4. Viewed in this light, the term *client* is not the ideal moniker but will suffice for our discussions.

### 2.3.5 The Classes in Perspective

Figure 2.6 is a simple chart showing the value of networked clients. The class 3 client is the most useful due to its fully digital nature and networkability. The price/performance ratio, however, will always be a selection criterion, so all client classes will find ample use in video systems as designers seek to drive down the total cost. In general, class 4 is the least expensive from a hardware point of view, and class 3 is the most expensive. Of course, the software costs per device can range from insignificant to very expensive, so there is no simple way to classify the price/performance across all classes of clients. In the final analysis, classes 2–4 will find ample use across a wide range of new video system designs.

## 2.4 FILE TRANSFER, STREAMING, AND DIRECT-TO-STORAGE CONCEPTS

As mentioned in Chapter 1, three methods are key to moving A/V information in a video system. Most client types may use combinations of these methods. Figure 2.7 shows a system that has all three types of A/V data-moving means. When you are considering the three, you may compare them against each other in three pairs or consider them separately by their own standalone merits. Analyzing the methods



**FIGURE 2.7** Streams, file transfer, and storage access.

by direct 1:1 comparisons (files versus streaming, files versus direct to storage, direct to storage versus streaming) is a bit tedious and somewhat obscures the overall trade-offs. To complicate things even more, because direct to storage may be NRT or RT access, our classification will analyze each method separately with some 1:1 for the most important characteristics.

In summary, the high-level characteristics of each method are

- **File transfer**—Move files from one device to another (or point to multipoint). NRT transfers are easy to implement using FTP or HTTP, have no geographic limitations, and have 100 percent guaranteed file transfer.
- **Streaming A/V**—Stream data from one element to another (or point to multipoint), usually in real time. Traditional A/V streaming (SDI, composite) is common in video systems. IT-based streaming is mature as well but is used mainly for content distribution to end users. Voice over IP (VOIP) is a form of audio streaming.
- **Direct to storage**—Random read/write data access by a storage attached client. Attached clients may have access to a clustered file system for a full view of all stored files. Storage access may be NRT or RT. Of course, RT access is more demanding on the connect infrastructure. Storage access may be via DAS, SAN, or NAS methods. See Chapter 3B.

**Table 2.1** A/V Storage and Data Rate Appetites

A/V Format	Byte Rate per Hour	Comments
64 Kbps compressed audio for an iPod	28.8 MB/hr,	1,000 hr is ~29 GB, a small-capacity HDD
2 Mbps MPEG video for a PVR	900 MB/hr,	100 2 hr movies is ~180 GB, one ATA disc
10 Mbps MPEG video	4.5 GB/hr	Typical rate for video servers storing SD commercials for TVOne
25 Mbps DV video or HDV video, or ~28 Mbps with audio and overhead	11.25 GB/hr video only, or 12.65 GB/hr with audio,	180 GB HDD can store 16 hr of DV or HDV video
166 Mbps uncompressed 4:2:2 SD video, 8-bit (720 H × 480 V × 30 FPS)	75.4 GB/hr	Uncompressed A/V has a huge appetite
995.3 Mbps for uncompressed 4:2:2 HD 1,920 × 1,080 × 30 imaging	448 GB/hr	The ATSC/DTV standard compresses video to 19.3 Mbps, shaving off 98 percent
384 Mbps per frame for 4 K H by 2 K V uncompressed (4:4:4, 16 bits per RGB pixel) digital cinema	4.145 TB/hr (24 FPS), 9.216 Gbps	Requires massive storage and I/O bandwidth for pristine quality

When you are transferring, streaming, or storing, it is wise to know the storage and data rate appetite of the A/V data structures. Table 2.1 outlines some popular formats and their vital statistics. See Chapter 11 for a discussion of A/V formats.

It is always good to have a rule of thumb when computing storage requirements. The best metric to memorize is the one in row 3 of Table 2.1. By knowing that 10 Mbps video requires 4.5 GB/hr, you can easily scale for other values because of the convenient factor of 10 scaling.

So 1 hr of DV requires  $28/10 \times 4.5 = 12.65$  GB/hr. Many of the chapters in this book refer to A/V storage and bandwidth requirements, so it is a good idea to become familiar with these stats. Please note that *Mbps* means megabits per second and *MBps* means megabytes per second. A lowercase *b* represents bits and a capital *B* represents bytes—8 bits.

When you are building an AV/IT system, which method is best to achieve the most flexible workflows—file transfer, streaming, direct to storage, or some hybrid combination? What are the trade-offs among the methods? Let us consider these questions.

**2.4.1 File Transfer Concepts**

File transfer should be viewed as a hierarchy of steps or processes. One three-layer segmentation for the layering is as follows:

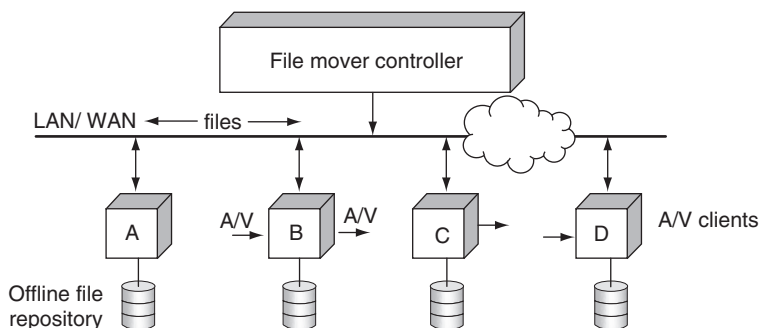
- **Top**—Scheduling, batching, file audits, reports, other metrics. These are useful for the what/who/when/where of transfers. See companies such as Signiant and FileCatalyst for examples.

- **Middle**—This is a transaction step for establishing the file transfer between two entities. Elements such as formal setup, initiation/terminate, security, metadata associated with the file, negotiation of transfer protocol to be used (FTP, HTTPS, etc.), restarts, error recovery, and more. See SMPTE standard 2032 (Media Dispatch Protocol, MDP) for an example of this function.
- **Bottom**—Protocol for actual file transfer. This could be FTP, HTTP, HTTPS, or another choice. Considerations for performance, security, and reliability are made at this layer.

The discussions to follow address the lowest layer for the most part, but other layers are covered in passing.

Figure 2.8 is used as the basis for our discussion. Clients A/B/C/D can all send and receive files. Frankly, each of these devices may also be a server, but let us use the client name for now. The file mover block is some sort of external control logic (automation system) that directs NRT file movement from a sender to a receiver(s). Additionally, individual clients can pull or push files from/to other clients in a peer-to-peer arrangement. Files can be transferred in a point-to-point or point-to-multipoint fashion. Each client has local storage of various capacities. File transfer between a source and a destination is often referred to as “store and forward,” although the moniker is losing favor. The file mover has the master database of the location of all files (or can locate a file using queries as needed). Client A may be the master file repository, but this is not a requirement for a file-based architecture.

Table 2.2 lists the general characteristics of a file transfer. Although not cited, many of the features of file transfer are different from either streaming or direct-to-storage methods. If you had to deliver an elevator pitch summary of file transfer, the top three points would be as follows: 100 percent error-free delivery is possible, NRT delivery normally, point-to-point is the most common mode.



**FIGURE 2.8** NRT file transfer topology example.

Table 2.2	Feature Set for General-Purpose File Transfer
	Move files between sender and receiver. Support for reading files (pull a file from a remote host) and writing files (push a file to a remote host)
	Point-to-point or point-to-multipoint support
	Guaranteed delivery of files or best effort with continuation strategies after a link interruption
	Faster, approximately equal, or slower than real-time transfer speeds
	Files are moved entirely or partial transfers
	Files protected by password, user name, and access groups
	LAN and WAN support
	Large file support (100 + GB class)

Table 2.3 (in reference to Figure 2.8) outlines the foremost advantages and disadvantages of the file transfer method. One acronym needs explaining: Just in Time File Transfer (JITFT). In a file-centric system, a client may need a file for a critical operation at a certain time of day. The file mover, for example, may schedule the file transfer (move a file from client D to client B) such that a needy client receives the target file “just in time” as a worst case. There is a delicate balance between sending files too soon (hogging the client’s local storage) and too late (may miss a deadline due to the NRT nature of file transfer). JITFT is a way to think about these issues.

Table 2.3	Advantages and Disadvantages of NRT Transfers
NRT File Transfer Advantages	NRT File Transfer Disadvantages
100 percent error-free delivery possible or best effort with FEC.	Media management can be complex due to files being scattered among many clients. Who owns the master file copy? Where is the master file? How many copies are floating around? Who has delete privileges?
No need for instant failover in case transfer link or network switch fails. Easy to ensure fault tolerance using alternate path diversity.	Delivery latency can be an enemy in a JITFT system. This can be a major issue for nondeterministic A/V requirements; for example, <i>play file ABC now</i> .
Uses standard IT-based file servers and JITFT logic to deliver files.	An NLE client may need hundreds of files for an edit session. Moving files from external storage to an edit client can be very slow; lost productivity.
Delivery rate may be set from much less than RT to much more than RT.	Cannot implement streaming workflows easily.
No reliance on RT storage QoS, less expensive \$/GB of storage.	Partial file transfers not common.

**MOVING FILES: IT IS TIME VERSUS MONEY**

When you are transferring an HD or SD file, you may choose a low rate link to save money. Slow links result in long delivery times—possibly many hours. However,

an improved QoS (more money) can reduce the delivery time—several seconds is achievable. So, as with many things in life, money saves time.

The NRT moniker is a bit vague in terms of the actual file transfer time. For some systems, a 1/10 file transfer speed (1 min of program material takes 10 min to move) is sufficient, whereas other system designs may require  $\times 10$  file transfer speed. The rated speed of transfer and the infrastructure performance are intimately connected. This topic will not be studied in detail, but it should be obvious that  $\times 10$  file transfer speed requires a storage and link QoS approaching RT, if not much better. Aspects such as component failover during a fast transfer need to be accounted for to guarantee JITFT performance. The analysis in Table 2.3 assumes that  $\text{NRT} < \text{RT}$  in terms of speeds. If NRT is defined as  $\gg \text{RT}$ , then many of the advantages/disadvantages are inaccurate. Of course, a JITFT-based design can also use a wide mix of NRT speeds, making the system design more difficult in terms of QoS and failover requirements.

The advantages/disadvantages comments in Table 2.3 are not meant to be row aligned; i.e., there is not necessarily a correlation between the advantage and the disadvantage on the same row.

#### **2.4.1.1 Reliable File Transfer Techniques**

The bedrock protocols for file transfer over networks use the TCP, UDP, and IP families. TCP is a packet-reliable protocol and is widely used. UDP is a “send-and-hope” methodology and also finds wide use. Table 2.4 outlines the chief methods used to transfer files using both TCP and UDP. Most of the entries support multiparty transfers. Some are best-effort delivery using an FEC, and some offer 100 percent error-free delivery. Refer to Chapter 6 for more information on IP, TCP, and UDP.

Let us consider the first four methods in Table 2.4. FTP (RFC 969) is the granddaddy of all file transfer methods and is used daily to move millions of files. In practice, FTP does not formally support partial file transfer (start-end points within a file) or point-to-multipoint transfers. To reduce the deficiencies of FTP, the IETF is developing FLUTE - RFC 3926. This is a new file transfer protocol for doing point-to-multipoint transfers, and it supports a massive number of simultaneous receivers over unidirectional IP channels. Because there is no back channel, each receiver uses FEC methods to recover modest amounts of data lost (if any) during transmission.



**Table 2.4** File Transfer Method Classifications

Applications	Transport Layers	Significant Aspects
FTP/browser HTTP/browser	Bidirectional TCP/IP. Other TCP-like protocols	Very common TCP acknowledges every packet but provides poor throughput over long-fat-pipes; 100 percent reliable transfer
FLUTE	Unidirectional using IP Multicast	Massive point-to-multipoint file transfer using FEC to correct for errors. New
GridFTP	Bidirectional TCP/IP	Multipoint distribution and partial file transfer. Specialized
Desktop drag-drop file transfer	TCP/IP supporting CIFS, NFS, or AFP	These are general file access protocols that may be used to move entire or partial files. Ubiquitous
ATSC, DVB use for sending program guides and related data	Unidirectional UDP data wheel based on ISO/IEC 13818-6 DSM-CC spec	Data sent as a repeating wheel with a simple FEC. Possibly long delay to get 100 percent reliable file since no back channel. Good for point-to-multipoint
Solutions from Kencast and others. With FTP-like support too	Unidirectional UDP with sophisticated FEC	Data sent once with sophisticated FEC for 5–30 percent (typically) data loss recovery. Good for point-multipoint satellite distribution and lossy networks with 100 percent reliable transfers within FEC limits. Excellent throughput
Solutions from Kencast and others. With FTP-like support too	Unidirectional UDP with sophisticated FEC and back channel using TCP or other protocol	As above but with back channel to request packet resend once FEC correct limit is reached. 100 percent reliable over very lossy channels with excellent throughput
Telestream, Aspera Software, and others	Unidirectional UDP, no FEC, with back channel signaling for error recovery and BW management	Simple, very fast, works well in low loss channels

The third row in Table 2.4 outlines GridFTP ([www.globus.org](http://www.globus.org)). This is a protocol for use in grid computing (Appendix C). It extends the standard FTP protocol with facilities, such as multistreamed transfer and partial file transfer. The effort is currently not under the sponsorship of the IETF. Files may also be transferred using the desktop drag-and-drop method common to many computer applications. In this case, the underlying protocol is not FTP, but usually CIFS, NFS, AFP, or another remote file access protocol (see Chapter 3B). It is apparent that there are various file transfer methods in use, but

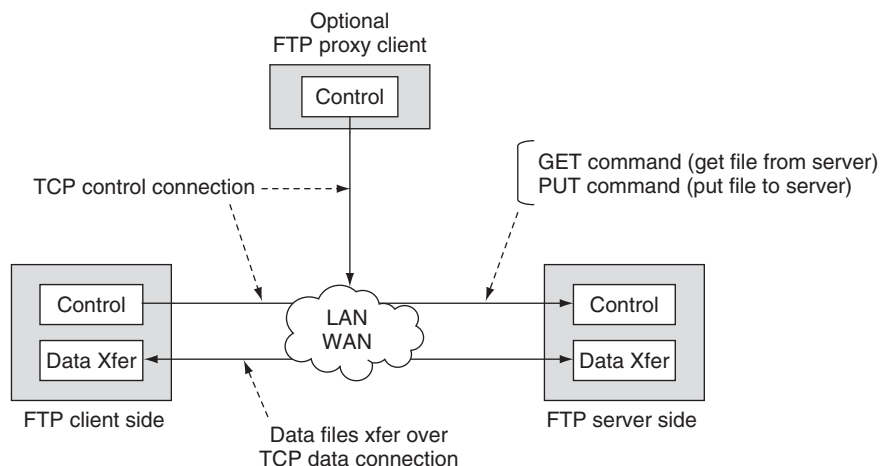
garden-variety FTP does the lion's share of work today for simple point-to-point applications.

### The Ubiquitous FTP/TCP Method

Developed in the 1970s at UC Berkeley, the File Transfer Protocol (FTP) is a wonderful tool for point-to-point file transfers using bidirectional links. Figure 2.9 shows the basic client/server FTP arrangement. Separate logical control and data connections are shown. In general, the client establishes a TCP control connection over a LAN/WAN with the server side. The server may accept more than one client connection (sometimes hundreds). Commands are sent to request the server to transfer files or accept files.

Typical commands are GET\_File to request a file transfer (server to client) and PUT\_File to send a file (client to server). The server responds by pushing or pulling files over the data connection. No file data move over the control connection. Of course, using TCP guarantees a 100 percent correct file transfer even over congested networks. FTP works quite well over a confined LAN, and transfer speeds can be exceptional. Transferring large files (gigabytes) is always a challenge if the link is slow (or long) because the transfer can take many hours.

For long-distance transfers, TCP is not ideal and, in fact, can be as much as *100 times slower* than alternative transport means. Why is this? Simply put, TCP's reliability scheme limits its data throughput inversely with network latency and packet loss. When packet loss is detected, TCP reduces its "congestion window" size exponentially. The window size defines the maximum number of bytes in the transmission pipeline before the receiving end acknowledges receipt to the sender. After recovering from a packet loss, TCP increases the



**FIGURE 2.9** FTP client and server with control and data logical connections.

window linearly and slowly. Hence, aggressive back-off and slow ramp-up limit the TCP throughput considerably. See Chapter 6 for a discussion of how and why this limits throughput.

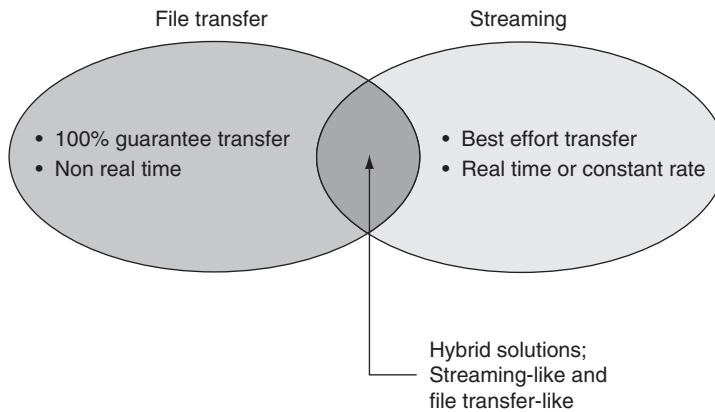
Another important aspect of FTP is a supported mode called “proxy client.” Many everyday FTP transactions involve the client and server as discussed. But what if a third party (the proxy; for example, an automation system) needs to initiate a file transfer between the client and the server? How is this done, and does the proxy receive any file data? In Figure 2.9 the proxy client is shown with only a control connection.

Consider an automation system (the proxy client) requesting an archive system (the FTP server) to send a file to an on-air video server (the FTP client). The proxy client initiates the transaction, and file data move between the two end points. From the standpoint of the FTP server, it is blind to where the request originated. The request could have come from the proxy or the standard FTP client. See RFC 959 (Figure 2.9) for more information on FTP and the proxy client.

Formally, FTP does not support *partial* file transfer, say, from a mark-in point to a mark-out point in the file. This hobbles applications that want to transfer 1 min out of a 60 min program, for example. Some vendors have used the optional FTP SITE command for this purpose. This method is being copied by others, and it may become a de facto standard. One FTP application vendor that supports the proxy client mode and the SITE command is FlashFXP ([www.flashfxp.com](http://www.flashfxp.com)), but there are others. Stepping away from TCP and its cousins, let us consider the fourth row from the bottom in Table 2.4.

### Data Wheel File Transfer Method

Consider the case of transferring a single file to thousands (or millions) of set-top boxes and televisions over a satellite or cable system. This is the case for the distribution of the electronic program guides. It is not practical to use a packet-reliable protocol such as TCP, so the Digital Audio Video Council (DAVIC, active in 1994–1997) invented the data carousel as a repeating data stream. Data are sent in packets (UDP in functionality and concept) to all end points with a modest forward error correction (FEC) to correct some errors at the receiver point. Data repeat as a wheel, so if, say, 5 percent of the data file was not received on the first pass, then the missing packets will likely be recovered on the next or subsequent passes. The data transfer rate may be very slow, but the patient receiver will be rewarded with a 100 percent data file, given sufficient passes of the data wheel. DVB and the ATSC standards specify this method for sending “files” to set-top box receivers. This protocol is an example of a hybrid file transfer and data stream. Figure 2.10 shows the features of a transport means that is part pure file transfer and part streaming. Given the limited use of this protocol, other more efficient means have been invented, as referenced in the last three rows of Table 2.4.



**FIGURE 2.10** File transfer and streaming domains.

## IP MULTICAST

IP Multicast is a bandwidth-efficient method (and set of standards) for moving data across a network to multiple locations simultaneously. Multicast requires that all the IP switches in the network cooperate. See

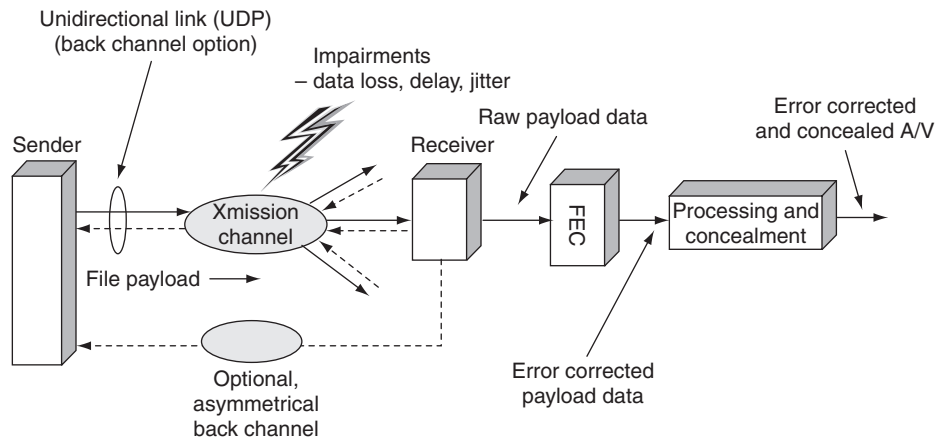
<http://en.wikipedia.org/wiki/Multicast> for a tutorial on the subject. It is not commonly implemented for professional A/V applications and is slowly gaining maturity.



## Sophisticated FEC Methods

Forward error correction is a general means to correct for a fixed percentage (say, 10 percent) of packet loss at the received end of a file transfer without needing to contact the sending site for retransmission. Files may be sent over unidirectional links (UDP in concept) at much greater rates than TCP will allow, but UDP lacks the reliability layer of TCP. Also, UDP naturally supports point-to-multipoint transfers. Figure 2.11 outlines a general system that relies on UDP for file transmission and FEC to correct receive errors. All errors within the correctable range will be repaired with 100 percent reliability. A return channel may be used to augment the FEC for requesting packet resends when the loss exceeds the correctable range (>10 percent, for example).

There is no free lunch, however. To correct for, say, 20 percent data loss, the FEC overhead is at least 20 percent and usually more. The FEC overhead may be located at the end of every packet/data block or distributed throughout the data stream. Algorithm designers choose from a variety of methods to correct for errors. The common audio CD uses a data-recording method called eight to fourteen modulation (EFM). When this is coupled with Reed-Solomon coding (plus data interleaving!), burst errors of up to 3,500 bits (2.4 mm ~ 0.1 inch wide defect on media) can be completely corrected. Even a horribly scratched



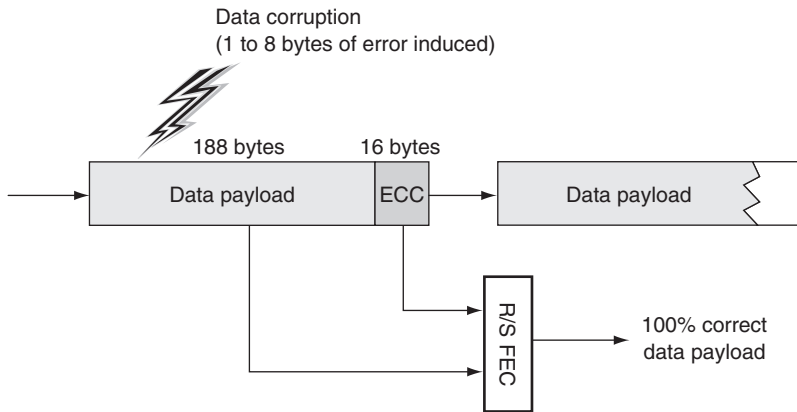
**FIGURE 2.11** File transfer delivery chain over unidirectional links with FEC.

audio CD (or video DVD) will likely play with 100 percent fidelity thanks to Mr. Reed and Mr. Solomon.

There are several methods that provide for error correction: Hamming codes, Fire codes, Reed-Solomon, BCH codes, Tornado codes, and countless proprietary codes like the ones from Kencast ([www.kencast.com](http://www.kencast.com)); for example, see (Morelos 1991). See Chapter 5 for examples of how RAID works to correct for errors from disk drives. The overall winner in terms of efficiency for burst error correction is the Reed-Solomon coding scheme, although other coding schemes may be more efficient under a variety of operating modes. These methods often support the common FTP programming interface semantics (GET, PUT, etc.), although not the FTP protocol layer.

Just how good is the Reed-Solomon FEC? In the R/S world,  $N$  total data symbols with  $K$  user message symbols have an FEC overhead of  $N - K$  data symbols. So if  $N = 204$  and  $K = 188$ , then 16 “overhead” symbols are needed to perform the FEC correction. It turns out that  $(N - K)/2$  symbols may be corrected by the RS algorithm. If each symbol is a byte, then for this case 8 bytes may be corrected with an overhead of 16 bytes. True, this method is not 100 percent efficient (one bit overhead per corrected bit would be ideal), but it is relatively easy to compute, good at large block error recovery, and well understood.

Looking at Figure 2.12, we can see that a 16-byte electronic correction code (ECC) is appended to the end of a 188-byte data payload for this example. This is an example of the ubiquitous DVB-ASI transport stream format. The 16-byte overhead can correct for up to 8 corrupted payload bytes as applied by the FEC algorithm. Using more ECC bits corrects for more corrupt payload bits. But what if not all the errors can be corrected? Of course, if the receiver can ask for a resend, then the bad bits may be recovered. This raises issues such as the latency for the resend, the logic to do this over a field of, say, thousands of



**FIGURE 2.12** Using FEC to correct errors: An example using R/S.

receivers, the requirement for a back channel, and so on. Under some circumstances, *concealing* an error when the FEC has run out of stream may be better than requesting a resend of data.

An example of this is to silence a portion of an audio track during the period of corrupt data samples. Another case is to replicate the previous video frame when the next frame is corrupt. Muting, duplicating information, or estimating has its limits naturally, but it works remarkably well when the method is not overused. The audio CD format uses a combination of FEC and error concealment to reconstruct errors in the audio stream. Of course, it is not feasible to conceal some data errors any more than it is practical to estimate the missing digits in a corrupt phone number (or metadata or compressed coefficients and so on). As with all methods to recover data, the designer needs to use prudence in applying the right mix of FEC, resend, and concealment. Concealment can be applied only after A/V decompression has occurred (if any) because it is nearly impossible to conceal most errors in the compressed domain.

### File Transfers Using UDP Without FEC

The last row in Table 2.4 includes methods using UDP without any FEC for reliable data payload transfers. During the course of the file transfer, the receiver acknowledges only bad or missing packets using a separate TCP-based (or UDP) message sent to the sender. The sender responds by retransmitting the requested packets. The HyperMAP protocol from Telestream ([www.telestream.net](http://www.telestream.net)) is an example of such a protocol. UDP payloads with negative acknowledgments can be very efficient, especially over a channel with low loss or in a point-to-multipoint transfer. A channel with 3 percent packet errors can achieve ~97 percent of the throughput of the raw link. HyperMAP has some added A/V-friendly features too, such as resume interrupted transfers, resume a transfer from a different location, and pause/resume a transfer. It beats FTP/TCP

in terms of performance at the cost of being a proprietary protocol. There are many variations on this theme in use today.

Another way to speed up data transfers is to use a *WAN accelerator*. This small appliance is placed at each end of a WAN link and uses protocol tricks to increase data transfer rates. How is this done? One method is to replace any TCP streams (as used by FTP) with UDP streams and appropriate error recovery methods. The protocol remapping techniques are non-standard, but the appliances hide this fact from all upper-level user applications. It is not unusual to get a 30–100×-improvement in data transfer speeds across a long distance WAN connection.

This class of products falls under the umbrella of wide area file services (WAFS). Besides TCP speedup, WAFS appliances also increase the performance of storage access over a WAN. See Chapter 3B for more insights on WAFS.

## SWARMS OF FILES



You likely have heard of iMesh, Kazaa, Morpheus, or other applications for file transfer across the Web. They are used commonly to transfer A/V files using peer-to-peer transfer—sometimes illegally. The reference to them is provided as an example only and not as an endorsement. Still, these programs are commonly used for legal file exchange as well. There is no file mover equivalent for these programs, and clients initiate and exchange files with other clients. One novel peer-to-peer transfer method

comes from BitTorrent. Using the BitTorrent software, a client downloads pieces of a file from several clients simultaneously until all the pieces are collected.

A “swarm” is a collection of clients all connected to share a given file. By each client sharing only pieces of a file, more clients can share and receive a given file simultaneously. See [www.bittorrent.com](http://www.bittorrent.com) for more information and links on swarming.

### Choosing the Best Method

So which of the methods in Table 2.4 is the best? Well, the answer depends on your needs. FTP/TCP works quite well for short hops (roundtrip delay < than 20 Ms) within a campus environment. Very large FTP/TCP throughputs on the order of 350 Mbps and greater may be achieved over Ethernet-based LANs. FTP is the de facto standard for file transfer, and most equipment vendors support it. For wide area file distribution, FEC methods work very well and have wide industry acceptance. The proven fact that non-TCP transport layers can deliver files up to 100× faster is a huge impetus to use TCP alternatives such as UDP coupled with a powerful FEC (or with negative acknowledgments) or WAN accelerators.

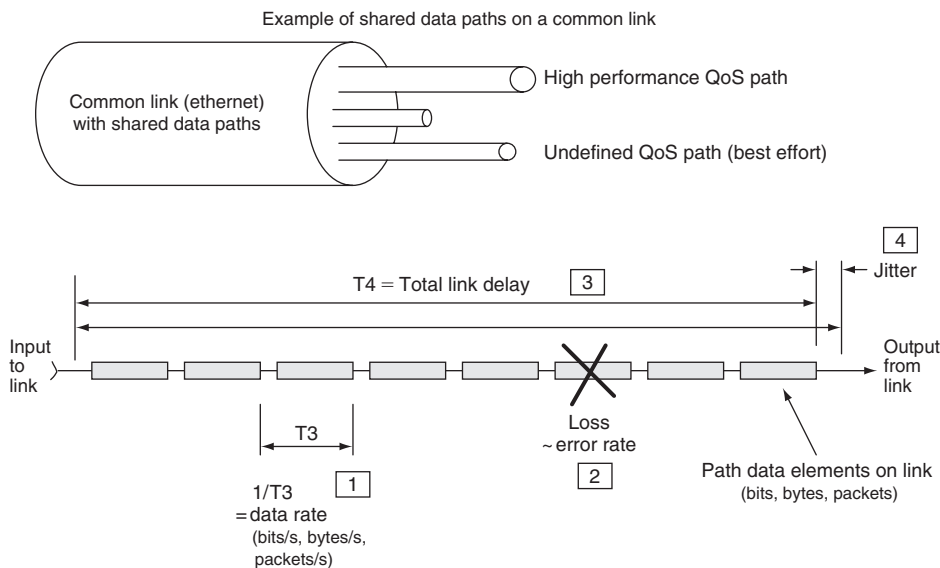
#### 2.4.1.2 QoS Concepts

Quality of service is a concept crucial to many aspects of A/V—both traditional and IT based. Typically, QoS metrics define the quality of transmission on any link that connects two points. Thus, the concepts apply to file transfer,

streaming, and storage access. An Internet connection, a video link from a sports venue, or a satellite stream may all be classified by their QoS metrics. Of course, QoS concepts can be applied to just about any process, such as application serving or storage access. Specific metrics are applied as appropriate.

The *big four* QoS link and network parameters are *data rate*, *loss*, *delay*, and *jitter* (*variation in delay*). These metrics apply to digital paths only. Ideally, a 270-Mbps multihop SDI path QoS is 270 Mbps rate guaranteed, loss is near zero, delay is approximately wire speed, and jitter is near zero. The QoS of a consumer's Internet<sup>3</sup> end-to-end connection is more relaxed with variable data rates (DSL or cable rates), some packet loss (0.2 percent for traffic is typical), modest delay (~5 Ms for intracity end points), and some jitter expected (~20 percent of delay average). Aiming for an ideal QoS may be a waste of money if a relaxed QoS will meet your needs. After all, the tighter the QoS, the more expensive the link. Regardless of the QoS for any practical link, an error can occur in the received video stream. Despite SDI's well-defined QoS, it always offers *best effort* delivery. Receive errors may be concealed in some manner.

Figure 2.13 illustrates the four key QoS metrics for an end-to-end network path. Figure 2.13 shows how each metric is measured in principle. Also, some links (such as Ethernet) may carry many different unrelated connections



**FIGURE 2.13** Network path QoS: The big four defined.

<sup>3</sup> These values are point-in-time averages, so your mileage will vary.



**Table 2.5** Streaming Versus File Transfer

Streaming	File Transfer
Best-effort quality (typically), unidirectional link normally	Guaranteed 100 percent delivery (typically), bidirectional link normally
Synchronous, isochronous, or asynchronous delivery choices	Asynchronous delivery advantage
Delivery time pacing 1×, 4×, etc.	Any desired delivery time
Point to multipoint is natural	Point to multipoint requires complex logic
Tight QoS desired	Relaxed QoS

simultaneously—each with its own QoS requirements. Whenever paths share a common carrier, attention is needed to keep them separate and with no coin-terference. Isolating paths is not easy because one rogue path (data bursting over their assigned limit, for example) can ruin the QoS of the entire link. It is normally left to the end point transmitting gear to meter data and set the QoS rate. Many WAN links meter data rates, usually at the edges, and will throttle data to their assigned range.

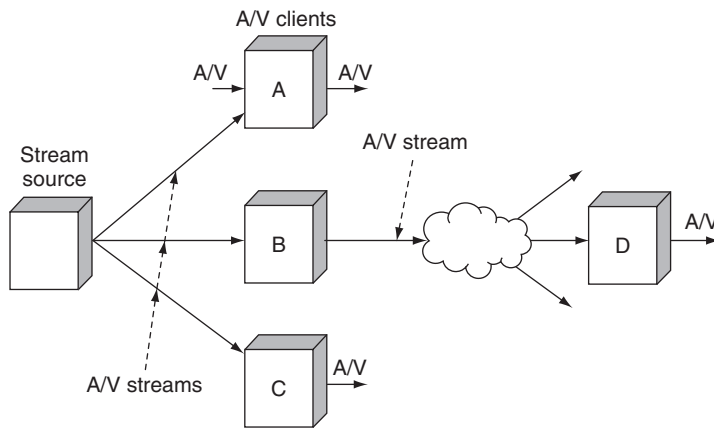
The required QoS of an A/V-related link depends on whether the link is used for file transfer or streaming video. These two methods are compared in the following section.

2.4.2 Streaming Concepts

In addition to file transfer, streaming is a second popular method used to move digital media between two or more devices. Five main characteristics highlight the differences between streaming and file transfer (see Table 2.5). Examples of streaming abound: digital TV broadcast over terrestrial and satellite links; traditional SMPTE 259M SDI and composite links; Intranet (campus) A/V streaming; and popular Web media as distributed by radio stations, news orga-nizations, and other sources of streaming A/V.

Some links are designed to natively carry real-time streaming, such as the ubiquitous SDI, and, in the WAN sense, T1/E1, SONET, and other Telco links. Digital broadcast standards ATSC and DVB support streaming. Still others such as asynchronous Ethernet can be made to carry streams using appropri-ate IP-based protocols. More variations on the streaming theme are discussed later.

Each streaming method uses completely different means but achieves the same end result—namely, the best-effort (usually) delivery of real-time A/V materials to one or more receivers. In some applications, streaming may do the job of a file transfer and vice versa, so it is prudent to compare these two meth-ods. Figure 2.14 shows a simple example of a point-to-multipoint streaming configuration. Let us consider each row of Table 2.5 (Kovalick 1998).



**FIGURE 2.14** Real-time streaming topology example.

Normally, streamed content is delivered in a *best-effort* fashion. If there is no back channel and only a modest FEC is used, then there is no way to guarantee reliability. Of course, with a back channel, the quality can be outstanding, but streaming is implemented most efficiently without a return channel. If the packet loss is overwhelming, then the efficiency of FEC is lost. A streaming application with a heavy FEC or back channel use may indeed be considered a file transfer method, as Figure 2.10 indicates in the overlap area. Most practical streaming applications in use today provide for best-effort delivery.

Row 2 of Table 2.5 outlines the different timing relationships that a delivery link may use. Digital video streams can be isochronous, synchronous, or even asynchronous. These timing relationships are covered later in this section. Isochronous and synchronous connectivity put strict requirements on the QoS for the link.

Another important aspect of comparison is the delivery timing as listed in row 3 of Table 2.5. For live events, a RT-streamed link may be the only practical choice. But for many other applications (copies, distribution, archive, etc.), a NRT file transfer is preferred. In fact, choosing the delivery time adds a degree of flexibility not always available in streamed video. It may be prudent to do a program dub at 100× without loss of quality or, to conserve resources, distribute a program at 1/10 real time. Of course, you can use streaming to mimic a file transfer, but the *best-effort* delivery of most streaming methods results in file transfer having the quality edge.

The point-to-multipoint nature of most simple streaming methods is difficult to match using file transfer (row 4 of Table 2.5). True, as shown in the section on file transfer, there are methods to do multipoint transfers, but they are sophisticated and not as simply implemented as streaming for the most part. Consider the case of the traditional SDI link. To send an SDI-based signal to,

say, 10 receivers is a snap. Merely run the source SDI signal through a 1:10 distribution amplifier or use a SDI router with multiple ganged output ports. See Appendix I for insights into a novel multipoint streaming method.

Finally (row 5 of Table 2.5), the cost of networking infrastructure is dropping much faster than that of video-specific infrastructure. Networking components are following price curves that are tied directly to Internet infrastructure costs, as discussed in Chapter 1. In general, file transfer has the cost edge for NRT applications, as the associated QoS can be very poor and still achieve a 100 percent reliable file reception. Of course, a loose QoS may be used for RT streaming, but the rate, loss, delay, and latency specs must be accounted for. One example of this is Internet telephony or voice over IP (VOIP). VOIP is a RT streaming operation using modest QoS; therefore, some long-distance telephone calls have less than desired quality.

If you have followed the discussion to this point, then it should be apparent that file transfer and streaming both have a place in video systems design. File transfer has an overall edge in terms of guaranteed quality compared to streaming for NRT content distribution, but RT streaming beats file transfer when live programming is sent from a source to one or more simultaneous receivers. LAN-based streaming for professional applications is rare despite its everyday use over the Internet and business intranets. WAN-based streaming (point-to-point trunking) has found some niche applications. MXF has support for streaming and some early tests show promise. See (Devlin 2003) for more information on using MXF for streaming applications.

#### **2.4.2.1 Streaming Delivery and Timing Methods**

A streaming receiver should be able to “view/process” the received stream in real time. The notion of clocking is usually associated with a stream because the receiver normally needs knowledge of the sender’s time references to re-create the target signal. A good example of a timed stream is a typical NTSC/PAL broadcast TV signal. Another example is that of video moving over an *SDI* or composite link. Many Telcos offer A/V streaming wide area connectivity using the analog TV-1 service, a digital 45 Mbps compressed video service, or a 270 Mbps uncompressed service. Accessing Web video on a PC is normally a streaming operation (or progressive download). Streams can be sent over just about any kind of link if the appropriate (rate, loss, delay, and jitter) considerations are met. A stream may be sent over a path using the following forms of connectivity:

- **Isochronous (equally timed bits) links.** In this case the medium (SDI and AES/EBU links) has an inherent bit clock to guarantee precise timing relationships. The clock is embedded into the on-the-wire format. The transmit clock may be recovered directly at the receiver from the link’s data bit stream. In a well-provisioned link, there is zero data loss, very low jitter, and low delay. This link was designed with RT streaming as the goal.

- **Synchronous (with time) links.** With synchronous links, end-to-end timing coordination is accomplished using a systemwide common clock. SONET and SDH are good examples of synchronous links. See Appendix F for more information on telecom links. Streaming A/V over a synchronous link requires extra effort compared to using SDI alone, but low data loss, jitter, and delay can be achieved.
- **Plesiochronous (almost synchronous) links.** In this case the sender clock and receiver clock are not 100 percent locked, but are very closely in sync. Two plesiochronous signals are arbitrarily close in frequency to some defined precision. These signals are not sourced from the same clock and, after a time, they become skewed. Their relative closeness allows a switch to cross-connect, switch, or in some way process the signals.

The inaccuracy of timing may force a receiver to repeat or delete data (video) in order to handle buffer underflow or overflows. Examples of these links are the ubiquitous T1/T3, E1/E3 telecom links. The clock drift specs are a function of the timing hierarchy as defined and used in the telecom world. If time stamp clock recovery methods are used in association with these links, true loss-free A/V streaming is possible.

- **Asynchronous data links.** Ethernet is a typical asynchronous link. True, it has a notion of clocking (100 Base-T has a nominal clock of 100 Mbps), but the timing is loose and there is no concept (normal use) of clocking coherence between Ethernet links. Streamed A/V may be carried over asynchronous links if the time stamp methods described for synchronous communications (see later) are applied. With error correction or using TCP/IP, excellent low loss and jitter characteristics may be obtained. One trade-off is typically long delays compared to the other links discussed.

It is possible to achieve outstanding A/V streaming quality using asynchronous or plesiochronous links if attention is given to coordinating the end point timing relationships. Let us call this technique *synchronous communications*.

## Synchronous Communications

With synchronous communications, the timing coordination is accomplished by synchronizing the transmitting and receiving devices to a common clock signal by some means. The links may be tightly clocked (SONET/ATM), loosely clocked (Ethernet), or not clocked at all (Internet connectivity). Methods for a receiver to recover a sender's clock are as follows:

- **The sender and receiver use the same clock.** They may use a GPS-based global clock, for example, or use independent Cesium clocks such that any frequency drift is inconsequential. Another method is to use the network time protocol (NTP, RFC 1305) at both ends. SONET is an

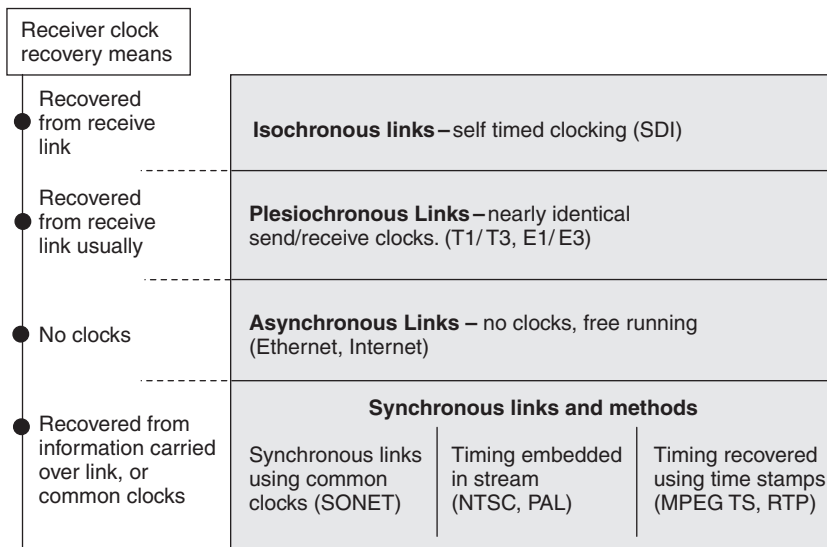
example of a WAN link using common clocks. Alternatively, IEEE-1588 Precision Time Protocol (PTP) provides timing signals over IP/Ethernet and can achieve 10- to 100- $\mu$ s accuracy in campus networks. This accuracy is about one horizontal line of video (see <http://ieee1588.nist.gov>). SMPTE and the EBU are co-sponsoring new standards for precision time distribution in the studio/facility environment. They should be available in late 2009 or soon thereafter. Refer also to the work from the IEEE's 802.1 Audio/Video Bridging Task Group. This group is developing time-synchronized, excellent QoS, low-latency A/V streaming services through 802 networks. The standards are referenced as 802.1AS, 802.1Qat, and 802.1Qav ([www.ieee802.org/1/pages/avbridges.html](http://www.ieee802.org/1/pages/avbridges.html)).

- **Derive a clock from received embedded time stamps.** This is commonly used for Web A/V streaming, MPEG broadcast (DVB, ATSC, and satellite) for home TV, and other applications.
- **Derive signal timing from the received signal stream.** A PAL or NTSC signal is designed so that a receiver may recover the sender's "clock." The common video composite signal also supports receiver timing recovery. A receiver uses the embedded horizontal and vertical timing information to re-create the sender's notion of the same timing. If designed properly, a receiver can stay locked indefinitely to the sender's clock references.

Note that synchronous end-to-end communications are not necessarily dependent on some sort of synchronous link. True, the tighter the link clocking, the easier it may be to stream A/V. But think of *synchronous links* and *synchronous communications* as different concepts. That is why it is possible to achieve respectable A/V live streaming over the Web—one of the most egregious examples of a non-synchronous network.

Figure 2.15 shows a segmentation of methods to achieve end-to-end streaming over links of various sorts. For most real-time streams, the receiver needs to recover the transmitter's original sense of timing before the A/V can be viewed or otherwise processed live.

The SDI link has a special place in the heart of the video systems designer. It is possible to create a complete A/V system with thousands of SDI links all completely lock-stepped in sync (video line and video frame synced). As a result, live frame-accurate switching (camera choice or other source choice) between the various SDI links is a relatively trivial matter. Switching is not so easy when it comes to using the other links in Figure 2.15. See Appendix B for some insight into synchronizing multiple independent A/V signals from non-isochronous sources. Still, IT links can be used for many typical A/V streaming applications with proper care.



**FIGURE 2.15** End-to-end communication methods for A/V streaming.

## Push-and-Pull Streaming

There is one more aspect of streaming to contemplate before we leave the subject: stream flow control. There are two general methods to move a stream from a sender to a receiver: push it out from the sender without regard for the receiver or pull it from the sender under control of the receiver. A broadcast TV station operates using the *push* scenario. No end point receiver can throttle the incoming stream—take it or leave it. The sender must push the output stream with the exact flow rate that the receiver will consume it. In the second method, the receiver *pulls* data from the sender. A viewing PC asks for, say, the next 5 s of video, and the sender complies by sending it. When the receiver has consumed 4 s, it asks the sender for more and so on. This method allows for the receiver to control the stream precisely according to its needs (display video, for example). Frankly, push streaming is used most commonly. In fact, most Web-based consumer A/V streaming is UDP push based. One special use of pull streaming permits the receiver to pause, fast forward, or rewind a stream by commanding the sender to act like a virtual VTR. Of course, physics will not allow a receiver to fast forward a live stream.

## Interactive Streaming

Interactive Web-based streaming (start, stop, pause, fast forward, rewind, etc.) requires a client feedback mechanism to control the source server. In Web applications, the real-time streaming protocol (RTSP, RFC 2326) may be used for this purpose. No A/V data are transported over RTSP; it is a stream control method. Rather, the real-time transport protocol (RTP, RFC 1889, 3550, and

3497) is used for streaming transport. RTP provides end-to-end network transport functions suitable for applications transmitting real-time A/V over multicast or unicast network services. RTP does not address resource reservation and does not guarantee QoS for real-time services. These protocols will find some use in professional, high-quality applications, especially for point-to-point trunking applications.

### High-Quality, High-Rate Streaming

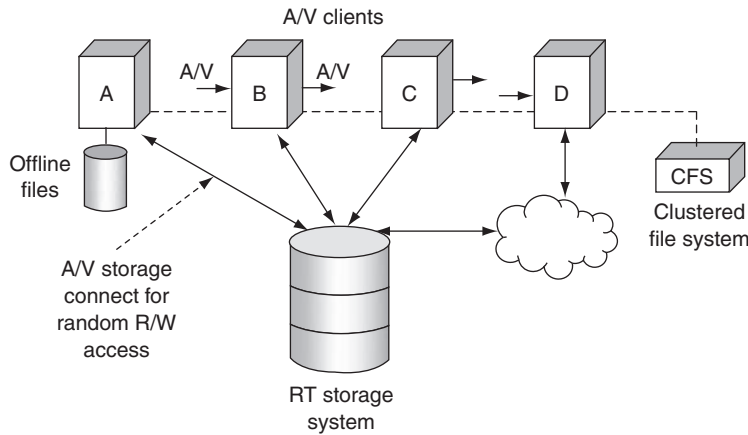
Many streaming applications are for low data rate, end-user needs. However, there is an application class for very high-rate professional “trunking” over IP networks. Point-to-point links between A/V facilities (sports, news, and campus events) have been provided for years using T3/E3, SONET/SDH, and other connectivity. These links can be very expensive and often require compressed A/V to reduce the data rate. With the advent of available IP access, several vendors offer SDI extenders over IP.

One such product is from Path 1 Networks ([www.path1.com](http://www.path1.com)). The Cx1000 IP Video Gateway delivers uncompressed RT SDI (270 Mbps) over IP networks. The unit maps the SDI data input onto a 1 Gb Ethernet link, and a second remote unit unwraps IP data and outputs it on an SDI link, resulting in transparent trunking. The Ethernet link may be WAN connected for even greater reach. Using clock recovery and FEC methods, the gateway provides excellent QoS in the presence of link jitter (250 Ms) and packet loss.

Well, that’s it for the coverage of streaming. These concepts are referenced throughout the book and are fundamental to AV/IT systems. For more information on professional streaming methods and applications, refer to the Video Services Forum. This industry group specializes in streaming A/V transport technology ([www.videoservicesforum.org](http://www.videoservicesforum.org)).

### 2.4.3 Direct-to-Storage Concepts

Figure 2.16 shows a simplified plan of a shared storage model for attached clients. In this case, clients A/B/C/D all have RT read/write access to a single storage system. NRT access will not be analyzed in as much detail because NRT is a relaxed form of the RT case. In the RT case, external storage *appears* as local to each client. All files are stored in the same repository, and there is no requisite for files to be moved (file transfer) between clients or from the storage to a client. Individual clients may have local physical storage, but this is not a requirement. Individual clients may also be restricted from accessing all the available storage and from deleting files. User access rights and permissions may be set for any client to guarantee a well-managed storage system. With direct access to storage, clients have the ability to randomly access read/write data into any stored “file” in real time. For the shared storage model to be most effective, all clients also need to share a common file system. Let us assume this exists for our analysis. Clustered file systems (CFS) are discussed in detail



**FIGURE 2.16** Real-time direct-to-storage topology example.

in Chapter 3A and provide for a true file-sharing environment among attached clients. Without a CFS, only the storage *hardware* may be shared.

Regarding the RT aspect of storage, it can also come in different flavors. By strict definition, RT storage allows for A/V-stored data to be read/write in A/V *real time* without exceeding the loss, delay, and reliability specs for the system. Playing a Super Bowl or World Cup commercial (\$2.2 million for 30 s) to air from RT storage will require a higher level of QoS than, say, supporting the QoS needs of a five-station edit system. As a result, the overall QoS of either an RT or NRT system is a strong factor of system requirements.

Table 2.6 lists the advantages/disadvantages of the shared storage model. An advantage in a row is not necessarily aligned with a corresponding disadvantage in the same row. Treat the columns as independent evaluations. To gain more insight, compare Table 2.6 to Table 2.3 and Table 2.5 on file transfer and streaming.

The biggest plus of the shared storage model is the fact that all clients have immediate RT read/write random access to all storage. This is not a feature of streaming or file transfer. Also, workflows may be designed in almost any form because files do not need to be moved before A/V operations can start. A general conclusion is that shared storage trumps file transfer, and file transfer trumps streaming for the majority of operational needs in an AV/IT facility. This conclusion must be taken in context. Sure, there are sweet spots for each method. If the application is live production, then streaming using SDI is required. In practice, facility SDI links are routed to provide access and reach. Also, file transfer is more appropriate than streaming for delivering files in NRT over unreliable links or long distance. Each method needs to be considered on its own merits and in relation to a particular target workflow. Examples of all methods are to be found in facilities worldwide.



**Table 2.6** Real-Time Shared Storage Access

Advantages	Disadvantages
Storage is immediately available to all clients/users (with permission) all the time for random R/W access.	Demanding QoS to support all client access in RT high-availability.
Media management is simplified by having all content in one repository. No versions of files in unknown places.	RT storage is more difficult to design and test than for NRT. A CFS is needed for true file sharing because only storage hardware is shared.
No JITFT file mover logic or prequeuing is needed. Files need to be transferred only when importing/exporting to external devices.	Single storage system for all A/V data puts content at risk if there is a total failure. Use of mirrored storage or RAID methods mitigates this.
Streaming workflows are a natural fit: <ul style="list-style-type: none"><li>■ Ingest to store.</li><li>■ Edit while ingesting from store.</li><li>■ Playout while editing from store.</li></ul> This is a common workflow for time critical sports highlight packages, for example.	Not trivial to perform hot upgrades; adding more storage, updating storage controller software, and so on. NRT storage may relax this slightly.

**2.4.3.1 Using File Transfer to Create a Virtual Shared Storage Model**

Figure 2.8 shows each client with individual storage. Figure 2.16 shows a similar configuration but in a shared storage model. If the individual storage pools in Figure 2.8 all had identical file content, then each client would see the same files. At times, there is a design advantage for individual storage to act like shared storage. To put it another way, if Figure 2.8 smells and feels like Figure 2.16, then some aspects of data management and control are simplified.

How can the two figures be made to act as equals from the client perspective? One way is to mirror all file content on all storage in Figure 2.8. This way, each client sees the same storage as the others. Of course, there is a waste of  $N - 1$  times the individual storage for  $N$  clients. Also, there is a time delay to copy content from, say, D to A, B, and C. Then, too, there is the extra bandwidth needed to make the  $N - 1$  copies. For example, if a video clip is ingested into client D (Figure 2.8), it needs to be copied (the faster, the better or JITFT) to A, B, and C so they each have access to the file. Ideally, A/B/C/D always have identical files. File copying is one way to make Figure 2.8 look like Figure 2.16 from a data access view. Of course, if client B modifies a file, the others need to get a fresh copy. This is not an issue for playout-centric systems but would be for editing-centric systems, for example.

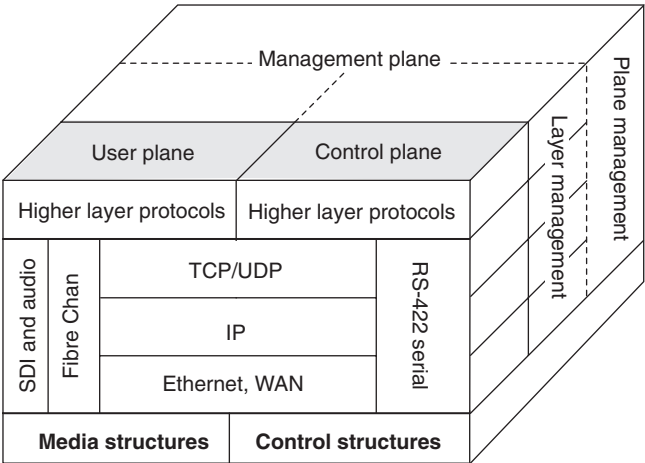
However, is the price of wasted bandwidth and storage worth the cost? With disc drive capacity at 1TB and bandwidth getting less expensive, some designers are working with this model. Its most virtuous aspect is no need for a sophisticated clustered file system. The price to manage file movement, deal with deletes and file changes, support the needed copy bandwidth, and have sufficient mirror storage is worth the effort and cost to eliminate the need for a CFS for some workflows. This reasoning is especially valid when only a few clients are needed but may break down if, say, five or more clients are needed. There are many ways to look at this problem and using JITFT reduces some of the constraints if the workflow allows for it. Also, because many systems do not require a complete file mirror on all clients, this decreases the need for  $N - 1$  times the storage. With proper knowledge of what files are needed for use, copying is reduced greatly at the cost of precise knowledge of how the files will be used. This is not always well known at the time of file creation, so doing a brute-force copy is often simpler.

Overall reliability is a strong suit for this method (Figure 2.8). If client A dies, D can take over. Relying on small “edge servers,” each with its own storage, simplifies the system design, albeit at the cost of file management. As with most engineering choices, the trade-offs are critical and will work for some applications but not for others. A strong case may be made for the clustered file system and eliminating all the file movements. Most large broadcast TV server systems for on-air playout (say, five or more clients supporting 16+ channels of A/V) use the CFS method, whereas some smaller systems (two or three clients) use the file mirror method.

## 2.5 THE THREE PLANES

Is there a unified way to visualize all the disparate elements in Figure 2.2 (with A/V media clients)? The diagram has the inherent traits of *data flow*, *control*, and *systems management* even though the concepts may not be apparent from a high level. Figure 2.17 is a pictorial of these three planes or layers. Each one has an associated protocol stack. Consider the following:

- **Data or user layer**—A/V data moving across links in RT or NRT. The data types may be all manner of audio, video, metadata, and general user data. This plane is alternatively called *data* or *user*. One term describes the *data* aspects of the plane, whereas the *user* handle denotes the applications-related aspects.
- **Control layer**—This is the control aspect of a video system and may include automation protocols for devices (A/V clients, VTRs, data servers, etc.), live status, configuration settings, and other control aspects.
- **Management layer**—Element management for alarms, warnings, status, diagnostics, self-test, parameter measurements, remote access, and other functions.



**FIGURE 2.17** *The three planes: Data/user, control, and management.*

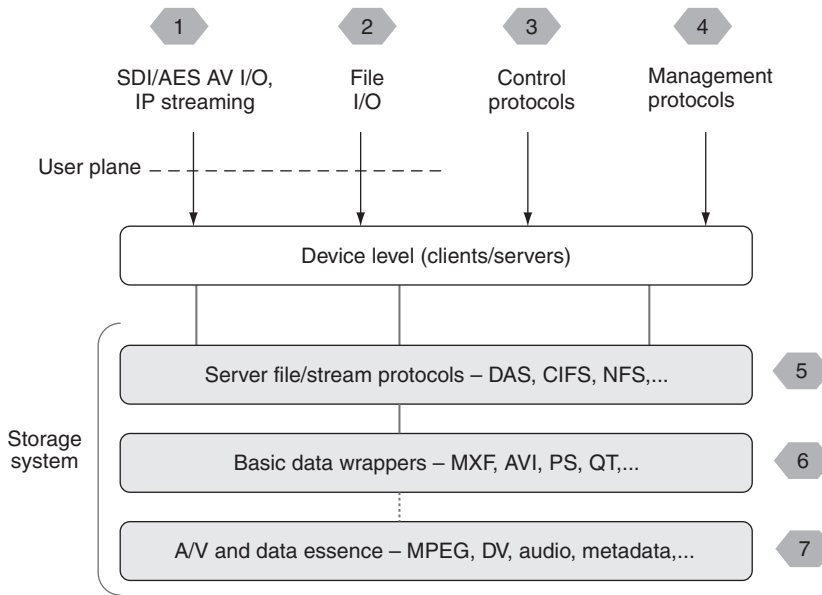
This model has been used for years in the telecom’s world with mature stacks across all three domains. In this case, the data plane is the actual data/protocols related to a telephone conversation, the control plane is the logic/protocols needed to establish and manage a call, and the management plane manages switching systems and configures them for subscriber-calling features. The famous Signaling System (SS7) protocol is used worldwide as the control protocol. SS7 is an architecture for performing out-of-band signaling in support of the call-establishment, billing, routing, and information-exchange functions of the Public Switched Telephone Network (PSTN). The legacy of the telecom’s logical view of the three planes is being applied to IT systems and hence AV/IT systems. Of course, the three stacks are completely different for our needs, but the overall concepts are still valid.

The detailed examination of the three layers and their implications is left to Chapter 7.

## 2.6 INTEROPERABILITY DOMAINS

Interoperability is mandatory for creating practical media workflows. This section outlines seven chief domains that contribute to interoperability between an A/V system and external points. The seven numbered interop domains are shown in Figure 2.18. For sure, there other domains, but these seven are frequently encountered. Domains 1 and 2, 3, and 4 comprise the data/user, control, and management planes, respectively, as outlined in Figure 2.17. Domains 5, 6, and 7 divide up the stack of a storage system, a user/data plane.

Frankly, older non-AV/IT systems had more interoperable functionality because there were fewer data formats, control protocols, and interfaces. There



**FIGURE 2.18** Seven key interoperability domains.

was a time when a video system could be built using one VTR control protocol and links for composite video (SD only) and analog audio. Today there is more of everything—more formats, more interfaces, and more protocols—so the move to AV/IT provides more choice with the associated headaches of more chance of interoperability problems between vendors' gear. To be fair, even without the IT component, advanced A/V interfaces and data structures are more complex than those from just a few years ago.

Think of Figure 2.18 as representing a simple A/V system with combinations of I/O, control ports, management ports, and storage access. Viewing all seven together provides a higher-level view of how to realize cross-product interoperability across a range of system components.

### 2.6.1 Domains 1 and 2: Streaming and File Transfer Interfaces

Domains 1 and 2 have several dimensions, including file and streaming protocols and file formats. FTP has become de facto for file transfer, but there are other choices, as Table 2.4 outlines. However, file incompatibility can be a cause of grief.

Streaming using IT means (LAN/WAN) is not well established for professional A/V systems, despite its common use in the Web space. Oh sure, IP streaming is used for low-bit rate proxy browsing, but high data rate, IP-based streaming is not common except for some A/V trunking applications....

Traditional A/V streaming interfaces such as SDI, composite, ASI, AES/EBU audio, and so on are well documented and supported by most equipment. If a vendor supports one of these interfaces, there is little that can go wrong. This interface point is likely in the first place in terms of interoperability and maturity. In the context of the three planes, this is a data plane element.

### 2.6.2 Domain 3: Control Interface

The control interface is vital for automated operations. Some, but not all, elements have a control interface point. For now, let us postpone a discussion of domain 3 until Chapter 7.

### 2.6.3 Domain 4: Management Interface

Management interfaces are required for monitoring device health, for diagnostics, and configurations. This interface is discussed in detail in Chapter 9.

### 2.6.4 Domain 5: Storage Subsystem Interface

Clients can connect to storage using several mature protocols. There are two classes of storage interfaces:

- DAS (direct attached storage) using SCSI, Fibre Channel, USB2, IEEE 1394, or other interfaces
- Networked-attached clients using SAN and NAS technologies

If you have ever connected a disk array to a PC or server, you have implemented DAS. When a client is outfitted with a Fibre Channel I/O card and connects to a common storage array, a SAN is being implemented. Whenever a client accesses stored files on a networked file server, NAS protocols are being put to work. NAS is a client/server file-level access protocol that provides transparent networked file access. NAS-connected clients are more common than SAN-connected clients due to the popularity of high-bandwidth file servers and Ethernet/IP networking.

Chapter 3B is dedicated to the coverage of DAS, SAN, and NAS. The bottom line is this: for client interoperable connectivity to storage, standards should be used. Always ask your providing vendor what protocols it uses for client-to-storage connectivity.

### 2.6.5 Domains 6 and 7: Wrappers and Essence Formats

The next pieces in the interop puzzle are the file wrapper and essence formats. A wrapper format (interface 6) is not the same as a compression format. A wrapper is often a content-neutral format that carries various lower-level A/V data structures. The simplest wrapper may only provide for multiplexing of A + V. The ubiquitous AVI file type is a wrapper and can carry MPEG, DV, Motion-JPEG, audio, and other formats. These are often called *A/V essence*

formats (interface 7). Another wrapper format is MXF. This is a professional wrapper format standardized by SMPTE (377M and others) and has profiles for carrying many different video, audio, and metadata formats. Interface point 6 exists only if the stored files are wrapped by AVI, MXF, QT, or similar.

Also at layer 7 are metadata. Often, metadata are packaged within XML but not exclusively. There are currently several SMPTE efforts to standardize metadata for a variety of A/V system operational needs. Other general data are stored at this level too.

File format translation between external types and internal formats may be required. For example, an external AVI file may enter (or exit) a system, but the internal format supports only MXF. So the format converter remaps the A/V formats as needed. Several companies offer file conversion gateways, among which are Telestream's FlipFactory, Digital Rapids' Transcode Manager, Rhozet's Carbon Coder, AnyStream's Agility Platform, products from Masstech, and others. Despite having some new format standards for file exchange, such as MXF, there are boatloads of older formats that cannot be ignored. File conversion has some trade-offs, such as speed of conversion, potential loss of A/V quality, testing of format conversions, and proper handling of metadata. Chapter 7 covers format conversion in more detail.

### 2.6.6 Interop Conclusions

The seven interface points are crucial in achieving interoperability with external users and interfaces. Successful interfacing is founded on standards and commonly used protocols and formats. For sure, there are other interface points such as vendor-supplied APIs, but these seven form the core for most A/V systems. In complex systems there will likely be some interface incompatibility, but these issues can be managed. For the most part, SMPTE, the IETF, and ITU/ISO/IEC are the organizations responsible for standardizing data, control, and management planes.

## 2.7 TRICKS FOR MAKING IT ELEMENTS WORK IN REAL TIME

For many years, purpose-built gear was needed to acquire, move, process, and output A/V signals. When some early adopter equipment companies decided to use a mix of A/V and IT elements, there were objections from many quarters. Following are some classic objections to using IT components:

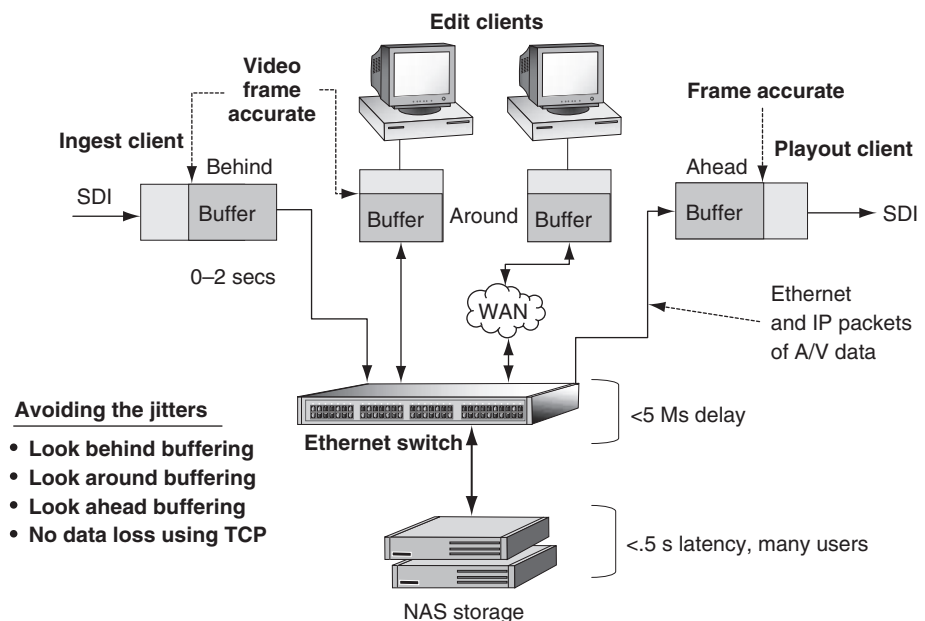
- They are not designed to pass frame-accurate RT video and audio.
- There will be too much packet loss, jitter, and delay.
- Ethernet is asynchronous and SDI is isochronous, so the two cannot interoperate.
- Available data rate is not well defined in an IT network.
- Building frame-accurate systems is impossible.

Then, too, there are objections about security threats such as viruses, worms, spyware, and other network-related pests. Security-related issues are covered in Chapter 8. But what about these other objections? Let us find out.

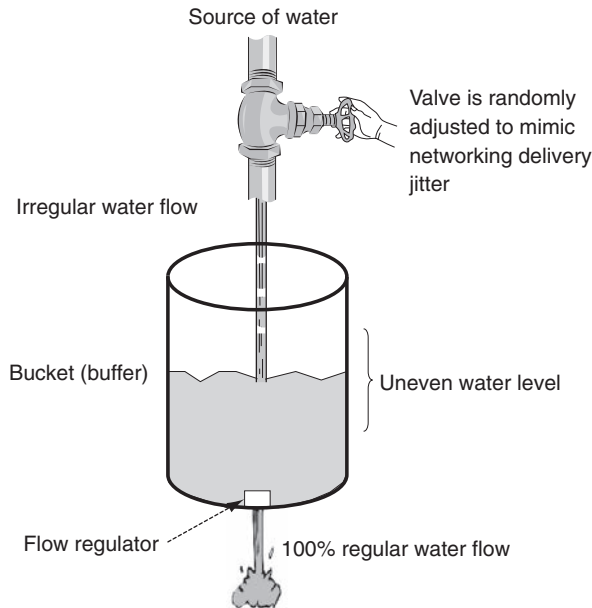
Figure 2.19 serves as the landscape for the basis of our discussion. The configuration shows NAS-attached clients, an Ethernet IP switch, and a file server (NAS storage). The clients access storage using the TCP/IP protocol suite. There are several areas where latency, jitter, and loss can accumulate if not managed properly. The trick to realizing excellent client performance is in the judicious use of the I/O buffer. Each client has a small buffer of a few seconds and they may be described as follows:

- *Look ahead* buffer as used in the ingest client.
- *Look around* buffer as used in the NLE client.
- *Look behind* buffer as used in the playout client.

Buffering is a way to smooth out IT component I/O irregularities. Of course, buffering also adds delay, and at times this may induce workflow problems or cause a client to appear as sluggish in response to a command. Generally, the more buffering that is used, the more irregularities and network problems may be hidden. So, in a way, *more buffering solves everything* (related to data jitter smoothing), yet *time delay is evil* (related to response times). So, indeed, the careful use of buffering can balance the needs of response time and smoothing needs. Hence, the art of buffering.



**FIGURE 2.19** Making IT components work in real time.



**FIGURE 2.20** Using a buffer to smooth out irregularities.

The general idea of buffering to smooth out irregularities is seen in Figure 2.20. A bucket is filled with an irregular flow of water. The input flow has an *average rate* that is constant, but the instantaneous rate will vary. A valve is adjusted randomly to represent irregular filling from uneven data delivery mechanisms inherent in the network switching, routing, lost packet recovery, storage delivery latency, and so forth. At times the bucket is nearly full, at other times nearly empty, but it never overflows or underflows. A regulator at the base of the bucket adjusts the output flow to be exactly even with no variations of flow rate regardless of the level of water in the bucket. As a result, the bucket acts like a buffer to smooth out any input irregularities and allows for a smooth output flow. The average input flow must equal the output flow for the method to work. When the source type of liquid changes, the bucket will need to be purged so as not to mix two dissimilar liquid types.

Using the bucket analogy, let us apply it to Figure 2.19.

The *look behind buffer* in the ingest client receives a regular flow of streaming A/V and sends it to storage over the network. The buffer has video stored from the past—hence, the look behind name. With a second or so of buffering, most IT-caused irregularities can be smoothed out as data are written to the storage. There may be occasions when a much bigger buffer is needed. Consider the case in which the storage array is offline for minutes longer. Under this condition, the ingest client can cache long periods of incoming video and write



it to storage when the storage connectivity resumes. See the section on caching in Chapter 3B.

The *look around buffer* in the NLE client smooths out I/O requests to storage. This buffer lets a user look around (jog, shuttle) a point in the time line with a human-fast response. Again, using bigger buffers (caching) yields even more advantages.

Finally, there is the *look ahead buffer*. This one is located in the playout client and queues A/V data for frame-accurate playout. In many video applications, there is ample time to queue a file into the buffer space before playout starts. However, output buffers may need to be “burst filled or charged” at up to  $5 \times \text{RT}$  rates to accommodate immediate requests for A/V playout. The burst filling burdens the infrastructure with much higher data rates but usually for less than 1 second. The design quandary is to support large surges of buffer charging yet not saddle the system with a commensurate cost increase due to increased peak rates.

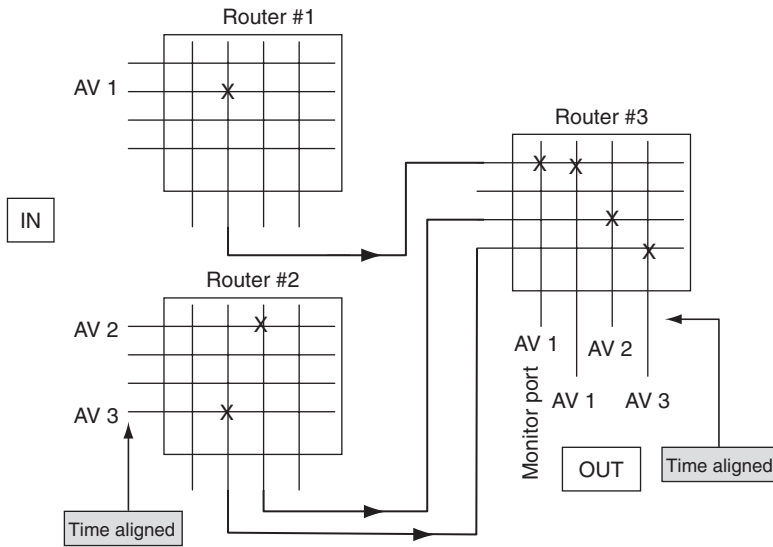
The system in Figure 2.19 has a limitation, however. The minimum “in to out” delay can be large, on the order of several seconds. There is recording latency due to input buffering, playout latency due to the output buffer, and storage array access latency. When the storage is Flash memory based, the in/out minimum delay may be less than a second. In the final analysis, buffering is the magic needed to coerce an IT infrastructure to behave in an A/V-civilized manner.

## 2.8 USING IT METHODS TO ROUTE TRADITIONAL A/V SIGNALS

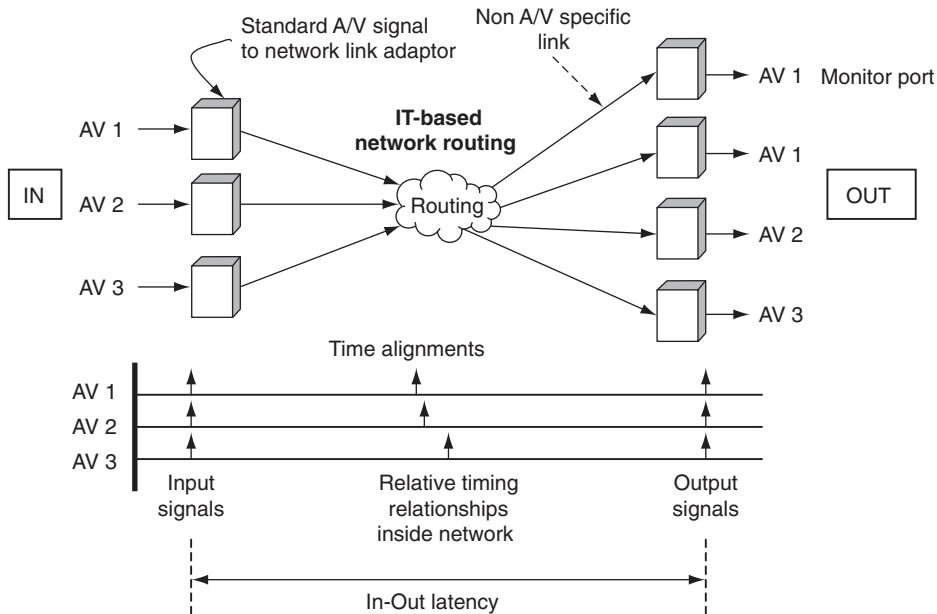
One of most common operations in a facility is to route streamed A/V signals. Figure 2.21 illustrates a traditional A/V routing structure using SDI links. Three SDI routers switch the A/V signals. Two key features of this configuration are very small in/out latency (on the order of tens of nanoseconds), and H/V timing is easy to establish at any point in the chain. Another feature is easy splitting of a signal for more than one output. In this case, AV 1 is fed to two outputs, one for signal monitoring.

What is the IT-based counterpart? Figure 2.22 shows an example. Ideally, the in-to-out relationships should be the same as in Figure 2.21. The router layer may be IP switching or other non-video-specific means. Today there are no IT-based routing means to duplicate the configuration specs found in Figure 2.21. Why not? The following summarizes the issues:

- IP networks have latency on the order of tens of microseconds to many milliseconds for campus size routes.
- H/V timing is lost during network routing. H/V timing must be re-established at the output ports. Protocols such as RTP support



**FIGURE 2.21** Traditional A/V router example.



**FIGURE 2.22** Using IT-based routing to transport A/V signals.

time-stamped streaming over IP networks, but even this does not guarantee recovery of the input H/V timing.

- Point to multipoint (splitting an input signal into one or more outputs) is difficult to achieve. IP multicast supports this but is not commonly used within an A/V facility.

The reality is that traditional A/V routing is a marvel and will not be replaced soon by IT means for those applications that demand it.

Duplicating the features of SDI routing is not easy, but over time methods will be developed as IT pushes deeper into all aspects of A/V. The concept of converting SDI signals to go over a WAN was mentioned earlier in this chapter, but this is normally for point-to-point video trunking and is an *extender* of SDI and not a replacement for it.

Importantly, the culture of using SDI and AES/EBU for all linking misses opportunities to use alternate non-real-time means such as file transfer. As a result, IT-based networking will replace timed A/V networks where it makes economic sense coupled with workflow efficiencies.

## 2.9 IT'S A WRAP: A FEW FINAL WORDS

This chapter outlined the essential elements of an AV/IT system. It is the basis of the remaining chapters in this book. Despite the fast-changing world of IT technology and products, the ideas in this chapter will not soon become stale with age. The comparisons between file transfer, streaming, and direct to storage are time-honored methods that will transcend any particular vendor's products. The interoperability domains will only become more mature as industry experience accumulates. In the end, the information in these sections is a good foundation for understanding the essentials of networked media systems.

## REFERENCES

- Britton, C. (2000). *IT Architectures and Middleware*: Addison-Wesley.
- Devlin, B., et al. (2003). Nuggets and MXF—Making the Networked Studio a Reality. *IBC 2003 Technical Conference Proceedings*, 94.
- Kovalick, A. (August 1998). A Reference Architecture for Digital A/V. *SMPTE Journal*.
- Morelos-Zaragoza, R. H. (1991). *The Art of Error Correcting Coding*. Hoboken, NJ: Wiley & Sons 07030.

# Storage System Basics

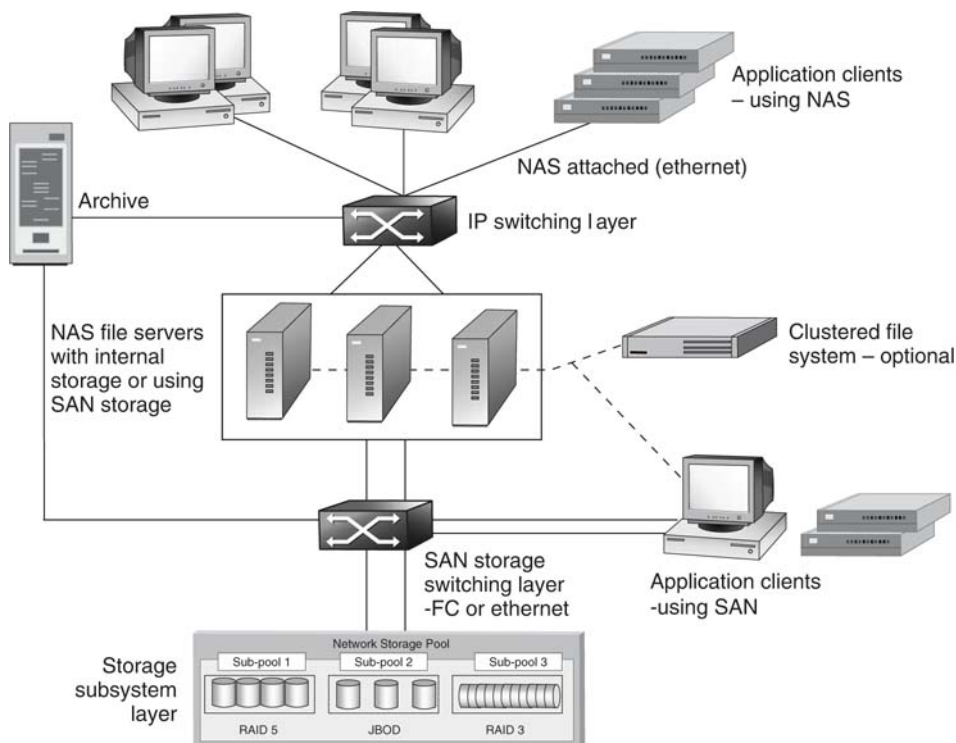
## CONTENTS

3A.0 Introduction to Storage Systems	82
3A.0.1 The Client and IP Switching Layers	83
3A.0.2 The Server Layer	83
3A.0.3 The Storage Switching Layer	84
3A.0.4 The Storage Layer	85
3A.0.5 Long-Term Archive	86
3A.0.6 Storage Software	86
3A.1 Storage Virtualization and File System Methods	87
3A.1.1 Storage Virtualization (SV)	87
3A.1.2 Clustered File System (CFS)	90
3A.1.3 Volume Management	92
3A.1.4 Distributed File System (DFS)	93
3A.1.5 Virtualization or CFS: How to Choose	94
3A.2 Client Transaction Types and Storage Performance	96
3A.2.1 Optimizing Storage Array Data Throughput	97
3A.2.2 Fragmentation, OS Caching, and Command Reordering	100
3A.2.3 Storage System Benchmarks	101
3A.3 Storage Subsystems	103
3A.3.1 HDD Capacity and Access Data Rate	103
3A.3.2 Aggregate Array I/O Rates	105
3A.3.3 General Storage Requirements	108
3A.4 JBOD and RAID Arrays	108
3A.5 NAS and SAN Storage	109
3A.6 Object Storage	109
3A.6.1 Deduplication	110
3A.7 Hierarchical and Archival Storage	111
3A.7.1 Data Flows Across Tiered Storage	112
3A.7.2 Managing Storage	114
3A.7.3 Archive Storage Choices	115
3A.8 It's a Wrap: Some Final Words	119
References	120

### 3A.0 INTRODUCTION TO STORAGE SYSTEMS

The core of any AV/IT system is its *storage and file server* infrastructure. After all, that is the place where the crown jewels are stored: the A/V and metadata content. Storage is a big topic so the treatment is divided into two main parts covering two chapters. This chapter, 3A, discusses the basics of storage systems: networked architecture, virtualization, file systems, transaction types, HDD performance, transaction optimization, RAIDs, clustering, and hierarchical storage, among other topics. Chapter 3B analyzes storage access methods (DAS, SAN, and NAS). Between these two chapters, the essentials of storage, with focused attention on A/V requirements, are covered. You may need to bounce between the two chapters, while reading either one, to get a full appreciation for the concepts and acronyms.

The landscape is first described using Figure 3A.1. This figure gives a high-level view of the domain of focus. This model has five horizontal layers. We study each layer briefly, with more detail added during the course of this chapter and the next. There are many real-world products and systems that have their roots in this configuration.



**FIGURE 3A.1** General view of a storage and file server infrastructure.

Consider an A/V edit cluster of  $N$  clients (craft editors and browsers), NAS or SAN attached. They may connect to one or more servers over a network. The servers in turn have access to storage. Clients may edit directly off the storage (direct to RT storage model) or use file transfer methods to load projects directly to the edit client.

Or consider an ingest and playout system. In this case the clients are ingest and/or playout nodes. Under control of a scheduler/automaton program or manual trigger, each of the clients will perform the record/playout operation on command. Some commercially available distributed video servers use the NAS method for client connectivity, and some use the SAN method. Most large-scale video servers from the major vendors are based on either of these two methods. See, for example, server products and systems from Avid, EVS, GVG/Thomson, Harris, Quantel, Omneon, SeaChange, Sony, and others. Small, standalone servers are often self-contained with no SAN or NAS storage connectivity, although they usually have a LAN port for file transfer support.

Clients may be A/V processors that are programmed to do effects, coding, or conversions of all kinds. One commercial example of this is the FlipFactory file conversion gateway from TeleStream.

Incidentally, Media Asset Management (MAM), automation control, systems management, and A/V proxy servers are not shown in Figure 3A.1. These elements are not relevant to the discussion at hand. Nevertheless, these elements are vital to any real-world A/V system, and their contributions are discussed in other chapters. Also not shown are any traditional (non-IT-based) A/V links. These links may always be added as needed.

### **3A.0.1 The Client and IP Switching Layers**

The first (top) layer is the application client. The client types are discussed in detail in Chapter 2. Each client can access the NAS file server over a network. Technically, a NAS-attached client accesses the storage layer via the file servers. In some cases, however, the server layer will also provide application services, such as file format conversion, encoding/decoding of the stored essence, caching, bandwidth regulation, and more.

The second layer is IP switching. This can be as simple as a \$100 switch or a complex campuswide mesh of switches. The reliability can be minimal or extend all the way to a fault-resilient network with various strategies of failover. Although Ethernet is the most common link, other less common links exist but are not the subject of this discussion. It is possible to design and operate this layer with excellent QoS with support for RT client access to the servers. See Chapter 6 for more information on switching.

### **3A.0.2 The Server Layer**

The third layer is the server subsystem. Servers may be located anywhere across the network. In general, they may be storage servers or application servers that

execute application code. The simplest configuration is a single NAS file server attached to storage. Microsoft offers the Windows Storage Server; many vendors use this as the core file system for their NAS products. At the other end of the spectrum is cluster computing with a mesh of servers working together as one. Cluster computing strategies range from a few independent servers that are load balanced to hundreds that appear as one virtual server. Fault tolerance and scalability are paramount in a cluster.

Grid computing is another technique that is differentiated from cluster computing (see Appendix C). The key distinction between clusters and grids is mainly in the way resources are managed. In the case of clusters, the resource allocation is performed by a centralized resource manager, and all nodes work together cooperatively as a single unified resource. In the case of grids, each node has its own resource manager, and overall there is no single server view as with clustering.

The big five players in the enterprise hardware server market are listed in Table 3A.1. The revenue includes any loaded OS software if present. The worldwide server market was \$54.4 billion in 2007.

One unique incarnation of a server is called a blade. Compared to a stand-alone rack-mounted server, a blade is a server on a card that mounts into a multcard enclosure. The packing density, cost, and efficiency (shares' power supplies, enclosure, and other elements) are outstanding. Blade servers accounted for 8 percent of all servers sold in 2007.

3A.0.3 The Storage Switching Layer

Layer four (Figure 3A.1) is the storage switching layer. In the simplest of cases, a server connects to a single storage array with barely a hint of switching or none at all. At the other end of the scale, the storage switching layer is a complex Fibre<sup>1</sup> Channel switching fabric (SAN) with failover mechanisms built

Table 3A.1 Worldwide Server Market		
Vendor	2007 Revenue <sup>a</sup>	Market Share
IBM	\$17,336	31.9%
Hewlett-Packard	\$15,415	28.3%
Dell	\$6,145	11.3%
Sun Microsystems	\$5,861	10.8%
Fujitsu/Fujitsu Siemens	\$2,676	4.9%
Others	\$6,988	12.8%
All vendors	\$54,421	100.0%

<sup>a</sup>Revenues are in millions. From IDC's Worldwide Quarterly Server Tracker.

<sup>1</sup> In this context, *fibre* has always followed the British spelling rather than the American *fiber*. This is a legacy of the standard body's efforts to differentiate it from older *fiber* optic cabling schemes.

in. More recently, some storage arrays support native Ethernet SAN connectivity based on iSCSI (SCSI protocol over TCP/IP). Fibre Channel has owned this space since 1998 so the migration to new methods will take some time. Note, too, that SAN clients connect directly to storage over Fibre Channel bypassing the server layer. A SAN-connected client (2 Gbps FC link, for example) has access to ~1,600 Mbps of storage bandwidth. Until recently, this type of performance has been available only using Fibre Channel.

Ethernet has won the war of enterprise connectivity and is pushing Fibre Channel lower in the value chain, although the two will coexist for many years to come. The trends to Ethernet/IP are very interesting, but legacy Fibre Channel SANs will not be replaced overnight. It is possible to build all five layers of Figure 3A.1 with only Ethernet/IP connectivity. During this transition, companies such as Brocade, Cisco, and HP provide gateways that link IP and Fibre Channel to create hybrid SANs. As a result, the modern SAN is composed of pure Fibre Channel at one end of the scale, a hybrid of IP and FC in the middle, and iSCSI at the all-Ethernet end of the spectrum. Replacing FC with Ethernet/IP has many implications, which are discussed in this chapter.

### 3A.0.4 The Storage Layer

Layer five, at the bottom, is the user storage layer. Of course, the storage system could be represented as tiered but is not here for simplicity. A tiered taxonomy is covered in Section 3A.7.1, “Data Flows Across Tiered Storage.” Storage systems range from a simple external USB2-connected array up to many Petabytes of Fibre Channel (or Ethernet)-connected arrays. At the high end, the arrays are complex systems of many drives (hundreds) with RAID protection and mirrored components to provide for the ultimate in reliability. For an example, of the ultra high end, see the TagmaStore from Hitachi Data Systems. There are various clever architectures from different vendors, all claiming some unique advantage in performance (access bandwidth + storage capacity + low access latency), reliability or packing density or usability (connectivity + management + backup + support) or price, or some combination of all of these. Storage is big business.

IDC ([www.idc.com](http://www.idc.com)) reports the 2007 worldwide disc storage systems<sup>2</sup> factory revenue was approximately \$26.3 billion (see Table 3A.2).

---

<sup>2</sup> IDC defines a disc storage system as a set of storage elements, including controllers, cables, and host bus adapters, associated with three or more disc drives [direct attach storage device (DASD)/hard disc drive (HDD)]. A system may be located outside or within a server cabinet. The average cost of the disc storage systems does not include infrastructure storage hardware (i.e., switches) and non-bundled storage software.



**Table 3A.2** Worldwide Disc Storage Systems Factory Revenue

Vendor	2007 Revenue <sup>a</sup>	Market Share
1. IBM	\$5,289	20.1%
2. HP	\$5,111	19.4%
3. EMC	\$3,995	15.2%
4. Dell	\$2,471	9.4%
5. Hitachi	\$1,522	5.8%
6. Network Appliance	\$1,482	5.6%
Others	\$6,465	24.6%
All Vendors	\$26,335	100%

<sup>a</sup>From IDC, March 6, 2008. Revenues are in millions.

The storage subsystem shown in Figure 3A.1 is a mix of various types, ranging from RAM-based to disc-based RAID systems of various flavors.

If managed properly, a distributed storage system may appear as one array. When the methods of storage virtualization and/or clustered file systems (discussed next) are used, just about any mix of storage technologies may be combined into a homogeneous whole. While it is true that the storage system may be built from optical, holographic, or tape media, these are considered as archive or backup formats due to their slow access times.

### 3A.0.5 Long-Term Archive

The last piece of the puzzle in Figure 3A.1 is the long-term archive. This component may connect into layer two (NAS attach), layer four (SAN attach), or be accessed via FTP depending on the design of the overall system. Archives are normally based on long-term storage with removable tape or optical media. Most commercial archives have access times considerably slower than HDD arrays but offer much greater storage density. For most A/V applications, it is possible to find a content balance among online storage (HDD-based), near-line, and offline storage. For more on this topic, see Section 3A.7, “Hierarchical and Archival Storage,” later in this chapter.

### 3A.0.6 Storage Software

End users will continue to invest strongly in various forms of storage software. IDC forecasts the worldwide storage software market to pass the \$17 billion mark by 2012, representing a 9.6 percent compound annual growth rate from 2007 through 2012. Storage systems are a strategic purchase for the media enterprise, so buy smart and insist on the performance, scale, and reliability that meets your A/V business needs.

## 3A.1 STORAGE VIRTUALIZATION AND FILE SYSTEM METHODS

With so many different devices connecting to the same storage system in Figure 3A.1, who is the traffic cop that regulates access? How does any one server or SAN client manage its storage pool? Who assigns access rights? What prevents a client from writing over the data space of another client? Who owns the directory tree? Well, there are two general ways to solve these sticky problems: storage virtualization and use of a clustered file system. Storage virtualization is considered first.

### 3A.1.1 Storage Virtualization (SV)

Storage virtualization partitions the entire array or collection of arrays into blocks that are individually assigned to select servers or SAN clients. This method insulates the requesting devices from the physical storage by providing a layer of indirection (filtering, mapping, aliasing) between the request for stored data and the physical address of that data. There are various ways to accomplish storage virtualization. In effect, the available storage is carved up into virtual pools that act as independent storage arrays. There is no common view of *all* storage from the client's perspective but only the portions they are allowed to see.

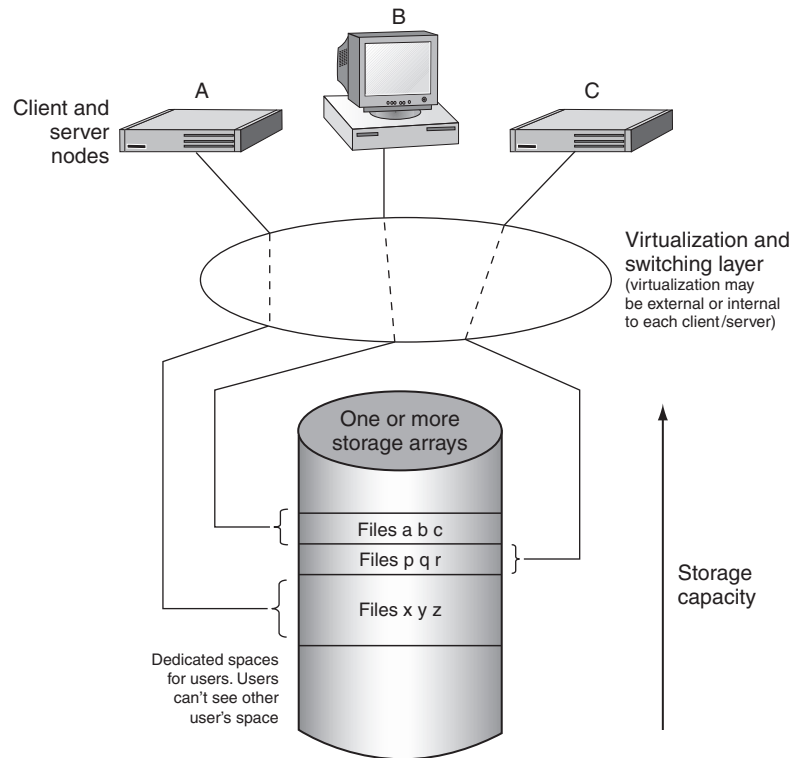
For example, in Figure 3A.2 server A can see only files X, Y, and Z; client B can access only files A, B, and C; and server C has access to files P, Q, and R. The respective files reside in sections of memory that are assigned to the attached clients and servers. There is no shared file system view; rather, memory is carved up and apportioned as needed. Dividing up the memory this way guarantees access rights; server A cannot mess with files that belong to server C and so on. However, attached devices cannot share files easily because they are walled off from each other. Sometimes this is an advantage, and other times it is not.

So why do it? Here are a few of the main reasons to use virtualization:<sup>3</sup>

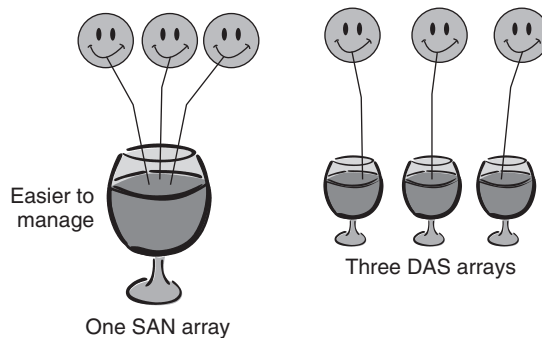
- **You can manage all storage with a centralized application:** This reduces labor costs to manage heterogeneous storage systems.
- **Users and applications have better access to storage:** User access to storage is not limited by geography or the capacity of an isolated storage module.
- **IT administrators can manage more storage:** Gartner Group estimates that managers can increase the amount they can administer by at least a factor of six if storage is consolidated.
- **You can lower physical costs of consolidated storage:** Existing storage is used more efficiently because one pool (a SAN) is apportioned rather than managing DAS islands. In effect, it is more efficient to manage one

---

<sup>3</sup> www.veritas.com has several white papers on virtualization of storage.



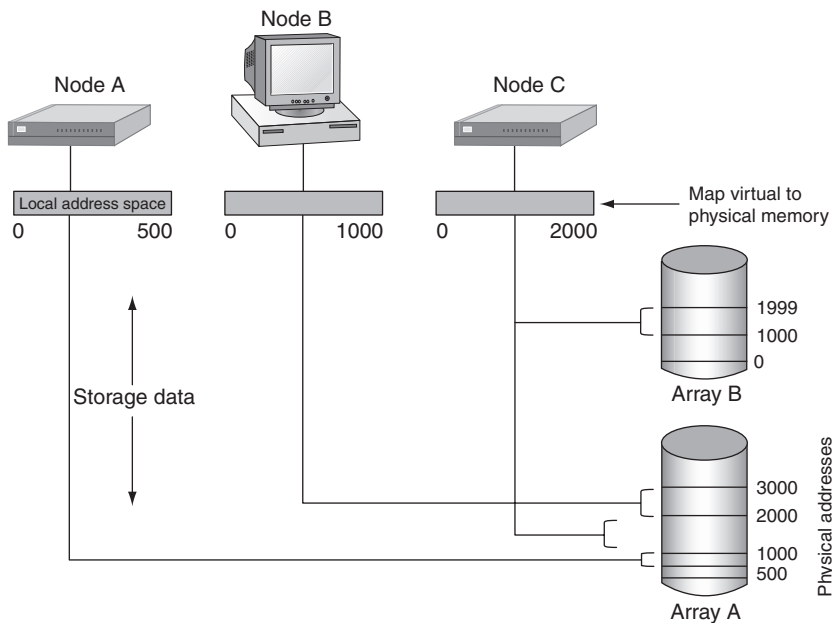
**FIGURE 3A.2** Example of storage virtualization: Sharing storage HW.



**FIGURE 3A.3** A SAN is easier and less costly to manage than islands of DAS.

pool with  $N$  straws drawing from it than  $N$  pools each with one straw (see Figure 3A.3).

- **You can scale with more reliability compared to DAS pools:** Scale by adding arrays and then map their access as needed to multiple requesters.
- **You can allocate capacity on demand.**



**FIGURE 3A.4** Virtualization using address mapping.

Virtualization is accomplished in several ways. The most popular is to map each node's assigned storage space to a physical address. This is done using the mapping, aliasing, and filtering of requested addresses.

Figure 3A.4 shows a diagram of the concept. Node A has a storage address range from 0 to 500, which maps onto array A's physical address 500 to 1,000. Node C has an address range from 0 to 2,000, which maps into two different arrays, each contributing a 1,000 and 1,001 address, respectively. Mapping is implemented in various ways, and logical unit number (LUN) masking is a popular choice. If done in the storage array controller, then it does the mapping based on the identity of the requesting node. Some vendors support the virtualization operation at the node level, switch, or storage array level, so the virtualization layer in Figure 3A.2 is a logical view, not a physical one.

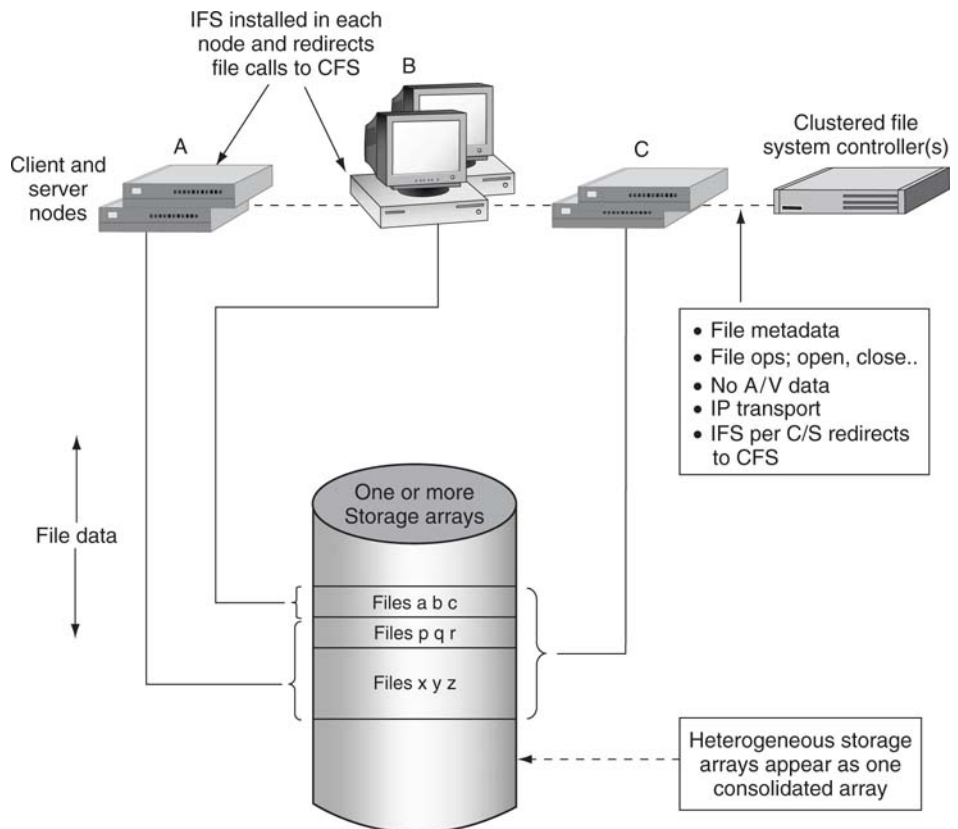
The debate about where to locate the virtualization logic is an interesting one. The ANSI/INCITS group has released the fabric application interface standard (FAIS). This standardized API facilitates LUN masking, storage pooling, mirroring, and other advanced functions at the fabric switch level. As a result, FAIS enables advanced storage services, which adds value to the storage system and reduces the need for proprietary solutions.

The main public companies that support storage virtualization are the names in Table 3A.2, plus Brocade Communications Systems, Cisco Systems, and Symantec. There are many other smaller vendors offering hardware and software virtualization solutions.

### 3A.1.2 Clustered File System (CFS)

A second method for managing a storage pool is to provide every attached node (SAN clients and NAS servers) with a single, universal view of file storage. A clustered file system allows multiple attached devices to have read and write access via a single file system for all available storage. A CFS is also called a *shared disc file system*. However, this name is vague and somewhat misleading. With storage virtualization, only the physical storage is shared. With a CFS, however, storage *and* files are shared. File sharing is complex because clients/servers may R/W to any file (with permissions), thereby creating potential for multiple users to write over the same portion of data. So file-locking mechanisms are needed to ensure that files are reliably opened, modified, and closed. A full-featured, fault-tolerant CFS is a thing of beauty. Incidentally, no actual user data pass through the CFS controller; only file system metadata are managed.

In Figure 3A.5, the CFS is made available to all nodes. In the figure, it is not important how the nodes connect to storage, only that they do and all see a



**FIGURE 3A.5** Example of using a CFS: Sharing storage HW and files.

shared file system. In reality, each node has an installable file system (IFS) software component for redirecting all user application file calls to the remote CFS controller instead of the local file system. In Figure 3A.5, node B may access only files a, b, and c, whereas node C has access to all the files. This is markedly different from virtualization. With a CFS, any node has potential access to any file in storage given the preassigned access rights. There are no inherent walls as with virtualization; all storage is accessible from any node in principle.

There are three main methods for a node to access a file system:

- Node uses only its internal file system. For example, a Windows server or client is based on Microsoft's NTFS file system. This file system manages any direct attached storage (DAS or SAN based).
- Node connects to remote server (NAS attach) and sees that server's FS (as in `\\Company_Server\your_files` format).
- Node has an installable file system software component to access an external CFS view of the storage. Each node can be part of a SAN or NAS. A node (or user of the node) selects the CFS view of the file system as it would select the CD-ROM drive or external file server (as in X: drive). The nodal platforms may be of any type (Linux, Windows, Mac, etc.), and a corresponding IFS is needed for each platform.

There is little magic in implementing the first two choices; they are commonly used. However, implementing a CFS is non-trivial. Why? Some of the common features are

- Negotiation of all R/W access to a heterogeneous pool of storage from a heterogeneous pool (Windows, Linux, Mac) of requesting nodes. Not all CFS implementations support all node types.
- Simultaneous users (hundreds or more) accessing up to millions of files.
- Data striping across storage arrays to increase access bandwidth.
- Data block locking, file locking, and directory locking.
- Bandwidth control to support QoS demands.
- Fault-tolerant operation using dual auto failover CFS metadata controllers. It may also implement a journaled FS (JFS) for supporting CFS failure with graceful recovery by performing a rollback to some stable FS state.
- Access control settings per user/group/campus.

The CFS must be a responsible citizen; it is a single point of file metadata for all member nodes. If it faults in some way, everyone is unhappy.

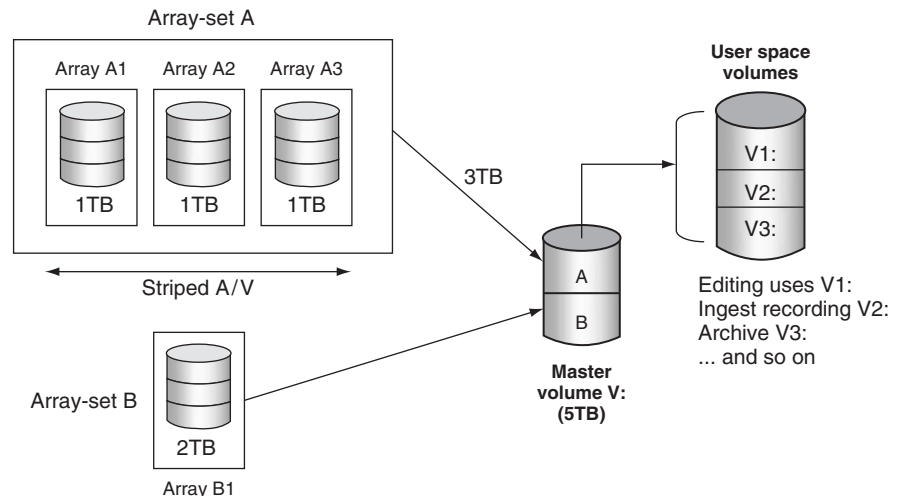
The CFS is a masterful administrator. It manages all the metadata that define a typical hierarchical FS: the storage access rights and permissions per

user/group, address maps that associate physical user memory with directory/file names, and tools to create/read/write/delete files. Each node has a low-bandwidth TCP/IP (or even UDP) connection to the CFS, and over this channel all CFS requests/responses are made. The CFS must potentially support hundreds of simultaneous requests across the entire FS spectrum of operations. Indeed, creating and testing a CFS is a massive undertaking.

One welcome addition to the CFS world is NFS V4.1, the so-called Parallel NFS (pNFS). NFS is discussed in Chapter 3B. Basically, NFS enables a client to access single file servers across a network. pNFS enables a CFS with a global name space for all clustered file servers. Detailed coverage of this is beyond the scope of this book. However, pNFS is the first standardized approach to implementing a CFS and may gain traction as the preferred method over the next few years.

### 3A.1.3 Volume Management

Whether a system is configured for virtualization or a CFS, the concept of volume management is important. Many practical systems combine a pool of disc arrays into one or more volumes for user access. By hiding individual arrays and mapping them into volumes, you can more easily allocate array storage to individual users or groups. Figure 3A.6 illustrates a small system with four arrays. Array set A has three arrays, each with individual drives. The A/V files are striped across all three arrays. Striping provides for improved non-blocking access to files. Striping is discussed later in this chapter. Array set B is only one array. All four are virtually combined to be one 5TB volume V. Users do not know if their files are stored on arrays A or B and should not care in most cases.



**FIGURE 3A.6** Example of volume organization.

In some systems, volume V: can be further divided into user or group space. When a volume manager application is used, virtual volumes V1, V2, and V3 may be created and assigned as needed to the user community. The amount of assigned storage may be changed as needed for business processes. In advanced systems, user utilization and department billing are included. Volume management conceals the details of array configurations and provides simple volumes (V1:, V2:, V3:, etc.) for user access. For a simple example of a volume manager under Windows XP, access Control Panel, Admin Tools, Computer Management, and Storage.

### 3A.1.4 Distributed File System (DFS)

A distributed file system is differentiated from a CFS in how the FS is implemented. Both attempt to create a single file system image for all storage, but they do so in different ways. With a CFS, file metadata reside on a separate and dedicated FS controller (or more than one for fault tolerance). With a DFS, the FS is distributed among the nodes (servers normally) such that the FS function spans servers and is a part of each server node. Think of a DFS as a way to repackage a CFS by folding its functionality into the servers that access storage. Since a DFS spans, say  $N$  servers, they must all cooperate together to create the single FS image. It is easy to imagine that this is complex in terms of guaranteeing reliability, stability, and scalability. DFS success stories are rare, but the Andrew File System (AFS) is probably the most successful. It was developed at Carnegie Mellon University and is now available through open source as OpenAFS ([www.openafs.org](http://www.openafs.org)). AFS is supported by Linux as well.

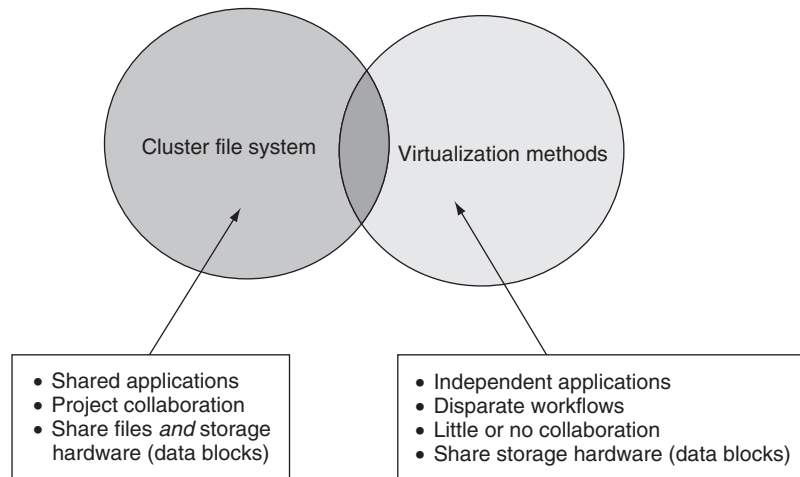
Unfortunately, the term *DFS* is a bit overloaded. Indeed, even Microsoft offers a form of distributed file system, but it is not a DFS in the true sense. To make matters worse, many authors refer to a DFS as a CFS and vice versa because the two are so intimately related. However, Microsoft's use of DFS is limited.

With Microsoft DFS,<sup>4</sup> users no longer need to know the actual physical location of files in order to access them. For example, if marketing places files on multiple servers in a domain, DFS makes it appear as though all the marketing files are on a single server. This ability eliminates the need for users to go to multiple locations on the network to find the information they need. DFS also provides many other benefits, including fault tolerance and load-sharing capabilities, making it ideal for all types of organizations. As a result, DFS locates the actual files across a domain of servers and *consolidates their appearance* for easy browsing. The actual files remain on disparate servers and storage systems, but the view of these appears as consolidated to a user searching for files. Microsoft uses DFS in this limited way. It is not the same as a true CFS or its derivatives.

---

<sup>4</sup> This paragraph is paraphrased from Microsoft's description of its DFS. See [www.microsoft.com/dfs](http://www.microsoft.com/dfs).





**FIGURE 3A.7** Individual benefits for CFS and virtualization.

### 3A.1.5 Virtualization or CFS: How to Choose

So when would a facility use virtualization, and when would a CFS (or true DFS) be more appropriate? Figure 3A.7 shows that the selection of one versus the other is based on how users operate. With A/V applications, because users tend to share files, work in collaboration, and use the same types of tools, a CFS is appropriate. In a business setting (running SAP, ERP, CRM), users rarely collaborate between applications, so virtualization suits the usage patterns. In fact, both are used in organizations worldwide.

In the technical and general business computing arena, several CFS vendors offer general-purpose (COTS) products:

- SGI's CXFS (Clustered eXtended File System) 64-bit FS for Irix, Linux, and Windows access to storage (SAN) or servers (NAS). Supports up to 9 million terabytes of storage (9 Exabytes). This was designed to support A/V applications.
- IBM's GPFS (General Parallel File System) for heterogeneous node access to a massive clustered server system. This CFS (actually a DFS) is not sold separately but only in conjunction with GPFS-clustered servers. IBM also offers the SANergy FS as a standalone product.
- Symantec's (acquired Veritas) Storage Foundation CFS. This is used in general enterprise applications. The Veritas CFS delivers a cohesive solution that provides direct access to shared disks and files from all heterogeneous nodes in the cluster.
- Quantum's StorNext FS for heterogeneous node access to storage.

- Apple's 64-bit Xsan CFS.
- Sun's open source Lustre FS. In testing on production supercomputing clusters with 1,000 clients, Lustre achieves an aggregate parallel I/O throughput of 11.1 GBps, utilizing in excess of 90 percent of the available raw I/O bandwidth. It is Linux based.
- Red Hat's 64-bit Global File System (GFS) for Linux server clusters.
- Sanbolic's Melio FS for heterogeneous node access to storage.

In addition to general-purpose CFS solutions, some captive CFS solutions are bundled with A/V SAN/NAS systems. These CFS products are included as a captive part of a vendor's overall storage solution. In some cases the CFS is buried in the infrastructure and is not called out as a CFS, but it is. Most of the systems support NLE, video server I/O, and other nodal functions for news production, transmission, and general edit applications. Basically, if an A/V workflow requires file sharing among collaborators, then a CFS is an enabler. Following are some of the leading A/V systems that include a bundled CFS:

- Avid's Unity ISIS storage platform for general editing, postproduction, and news production applications.
- Harris's NEXIO advanced media platform (AMP).
- Omneon's MediaGrid System with the EFS (Extended File System, a CFS).
- Quantel uses the Quantum StorNext CFS alongside its sQServer storage system. This is a COTS CFS integrated into an A/V environment.
- SeaChange Broadcast MediaCluster play-to-air video server with a custom-embedded CFS (actually a DFS) configured to work with each of the nodes in the cluster.
- Thomson/GVG K2 client/server with support for Windows-based transmission servers and news production nodes. This system uses the StorNext CFS from Quantum.

For the most part, these A/V vendors chose to build their own CFS and not use a COTS version. Why? There are several reasons. One is for total control in mission-critical environments. Another is based on a make-versus-buy analysis. Still another is performance related to reliability, failover speed (not all COTS versions are optimized for A/V applications), and integration with embedded client operating systems (non-Microsoft). Time will tell if the COTS versions will beat out the custom versions in the market. The COTS versions have massive engineering behind them and are feature rich. The A/V-specialized ones are niche products and may well get sidelined as the industry matures. Some of the COTS versions are A/V friendly, and this trend will most likely continue.

3A.2 CLIENT TRANSACTION TYPES AND STORAGE PERFORMANCE

Client applications conduct exchanges with servers and storage using a wide range of transaction types. Figure 3A.8 shows three of the most typical:

- **Complex transaction.** An example is to obtain the rights to edit a valuable piece of content. This may require exchanging user name, password, condition information, and purpose and also granting a use token. Note that the storage may be accessed several times during the transaction. A database (SQL Server, Oracle, MySQL, etc.) may be involved with this type of transaction.
- **Simple transaction.** An example is a NAS-attached client asking the server layer to read a block of video MPEG data from the storage array.
- **Direct R/W transaction.** An example is a direct R/W to storage using a DAS or SAN connection.

For each of these transactions, the storage system (HDD based for this analysis) is accessed, and its efficiency of performance depends on the access patterns. Four main aspects contribute to the overall performance of a storage system.

1. R/W patterns: Size of R/W block *and* random or sequential access to the HDD surface
2. Mix of reads versus writes from 100 percent read (0 percent write) to 100 percent write (0 percent read)

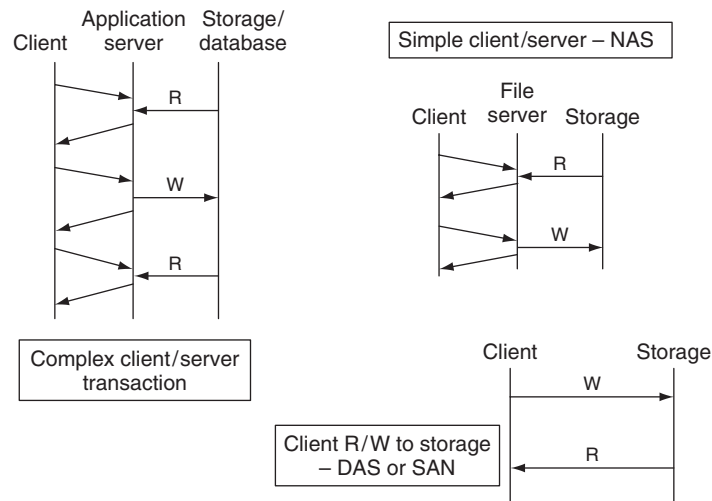


FIGURE 3A.8 Storage transaction types.

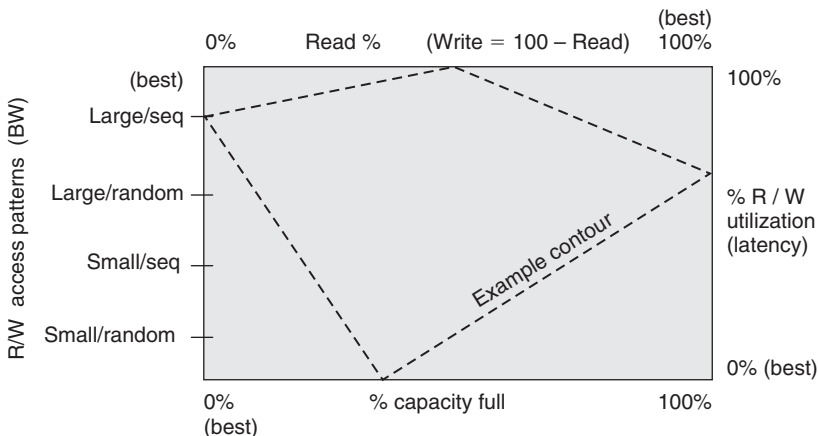
3. Utilization of the array from 0 to 100 percent usage
4. Capacity remaining from 0 to 100 percent

Each of these factors impacts various performance metrics. Figure 3A.9 shows a “billiard table” diagram with each side being one of the four factors just described. It is assumed that the array is not laboring under any failure modes (RAID data recovery in use). The purpose of the diagram is to encapsulate all four of the metrics into one visual imprint. Starting at the bottom left and moving clockwise around the outside, the left axis is a measure of user data access patterns (strong influencer of access bandwidth), the top is a measure of the mix of R/W transactions (influencer of access bandwidth), the right axis is the total throughput utilization (measure of latency), and the bottom axis is a measure of the remaining capacity of the array. For *each* of the four axes, moving clockwise is an improving metric. For example, the top of the right axis indicates 100 percent utilization of the array. For this case, the R/W requests form a deep queue, and therefore transaction latency is inevitable due to the heavy loading. However, the lower portion of the right axis is 0 percent utilization, so the occasional R/W request gets immediate response because there are no other requests to compete for storage resources. These four elements are developed in the following sections.

### 3A.2.1 Optimizing Storage Array Data Throughput

Considering item 1 from the previous list, the two most important metrics that influence individual disc performance are block size and data seek method:

- Large R/W blocks (0.5 to 2 MB block), medium R/W blocks, or small R/W blocks (4 to 64 KB blocks).



**FIGURE 3A.9** Storage system performance billiard table diagram.

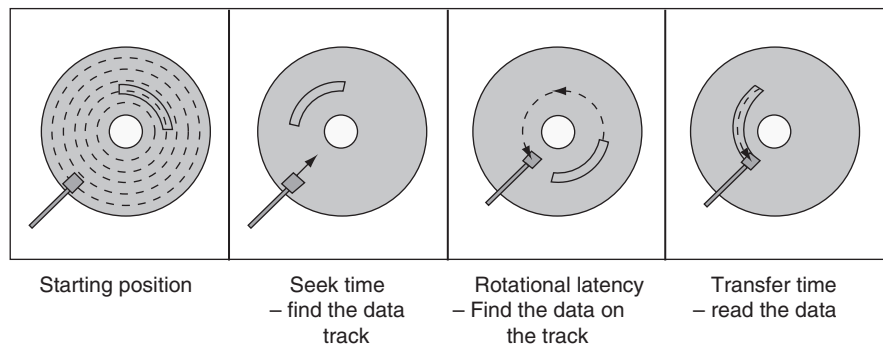
■ Random or sequential R/W data access:

Random: The next R/W may be at any physical address on the disc, causing the HDD head to seek to a new track.

Sequential: The next R/W address follows in order so that the R/W head of the disc does not have to seek a new track.

These factors are not relevant if the storage system is RAM based because RAM has no concept of rotating surfaces or R/W head movement. When you are using a HDD-based array, however, the *seek time and rotational latency* are very important. These are illustrated in Figure 3A.10. *Seek time* is the time required for a disc drive's read/write head to move to a specific track on the disc platter. Seek time does not include latency nor the time it takes for the controller to send signals to the read/write head. *Rotational latency* is the delay between when the controller starts looking for a specific block of data on a track (under the head) and when that block rotates around to where it can be read by the read/write head. On average, it is half the time needed for a full rotation (which depends on the rotational speed, or rpm, of the disc). It is because of the seek time and rotational latency that the block size (large/small) and random/sequential access methods have such a big impact on drive R/W performance. As we shall see, the rpm of the platter (spans 7.5 to 15K usually) is not a key factor in the average data throughput for large A/V data blocks. First, here is an illustration to help you better appreciate these issues.

Let us say that you plan to go to Maui, Hawaii, for vacation. The airplane trip is analogous to the seek and latency times before you arrive. Once you arrive on Maui, you likely want to stay as long as possible. The longer you stay, the less bearing the initial air travel delay has on your overall journey. If it takes 5 hr to reach the island and you stay 2 weeks, then the one-way travel time is only 1.5 percent of your total vacation time. However, if you stay only for 5 hr and then hop on the red eye to return home, the one-way travel time was a huge part of the overall journey (50 percent). What a waste! This is the same as when doing a R/W transaction to a HDD. The bigger the data block, the more time is spent reading/writing compared to the travel time to get there.



**FIGURE 3A.10** Key access metrics for HDD performance.

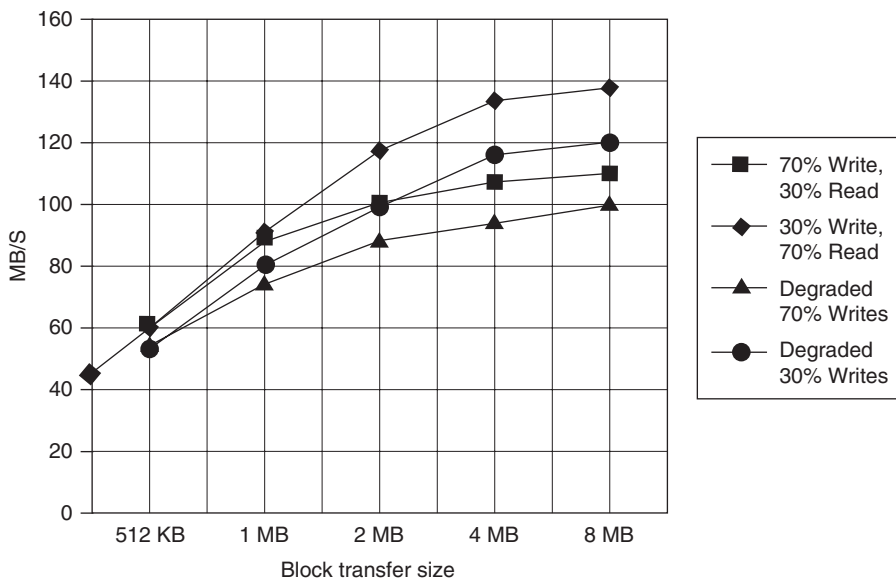
As a result, the average R/W bandwidth is strongly related to block size; bigger blocks yield higher data rates on average.

What about random versus sequential access? When planning your tour of Maui's best beaches, you could randomly visit five different beaches (Wailea, Hamoa, Kapalua, etc.) or plan to visit them by the most direct route, thereby saving travel time. This is the same as when doing a R/W access to a disc. If the R/W head does not seek to a new track (a sequential access) for each R/W transaction, then the overall throughput is increased compared to random R/W access. It is not always easy to place data sequentially (due to disc data layout strategies controlled by the OS), but A/V data lend themselves to this type of placement due to their large size and typical sequential access for a read. So let us celebrate long vacations with plenty of beach time.

These factors may be logically grouped into four specific access patterns:

1. Large blocks/sequential access (best A/V access rates)
2. Large blocks/random access
3. Small blocks/sequential access
4. Small blocks/random access (worst A/V access rates)

As per the reasoning thus far, the access bandwidth progressively improves from the worst case (bottom) item to the best case (top) item. It is arguable if items 2 and 3 are always positioned this way. For some cases they may in fact be reversed. There is a considerable performance gain between the top and bottom items. Figure 3A.11 shows the actual measured access bandwidth versus the block size for random R/W transactions.



**FIGURE 3A.11** Array performance versus block size.  
Source: Avid Technology.

The test configuration is four data drives plus one parity drive in two RAID 3 sets. So in all, there are eight data drives and two parity drives. RAID is fully explained in Chapter 5, but what you need to know now is that it is a reliability mechanism to recover all the data from one (or more) faulty drive. All user data are striped across all eight data drives (RAID 0), so an 8 MB user block is split into eight 1 MB blocks—one per data disc. In Figure 3A.11, the aggregate R/W bandwidth for the multidrive array is 140 MBps, with a mix of 70 percent reads and 30 percent writes and a block size of 8 MB. The same R/W mix reaches only 60 MBps, with a block size of 512 KB and much less for a 16 KB block that is common in many non-A/V transaction applications.

Notice that the mix of reads versus writes has a profound effect on the array throughput. The top axis in Figure 3A.9 is a measure of the R/W mix. For a 30 percent read and 70 percent write mix, the bandwidth is about 110 MBps. Why is this? In a RAIDed configuration, a write has a penalty because the parity disc needs updating for every write to the four data drives. If user data are not striped across all data drives but are written to only one, then the write penalty is much worse. The reasons for this are developed in Chapter 5. A read does not need to access the parity drive unless there is a faulty data drive that needs restoration. Some vendors' solutions activate RAID when a drive is late in returning read data, as this improves latency performance. Figure 3A.11 also shows data access rates under a "degraded mode." This is the case if one of the data drives is being rebuilt (RAID node) in the background during R/W transactions. Rebuilding a drive steals valuable bandwidth from user R/W activity.

The right axis of Figure 3A.9 is a measure of array utilization. As mentioned, the more array transactions per second, the longer the delay to complete a transaction. Because transaction delay is never good, many systems are designed to operate at less than ~80 percent utilization for a given latency.

Another contributor to storage command-response latency is a large R/W data block. This is especially true when an array or HDD has many requests in queue. Large R/W blocks require a longer time to execute, so a deep command-response queue can add significant delay if not well managed.

### 3A.2.2 Fragmentation, OS Caching, and Command Reordering

The image of the storage array data layout starts looking like Swiss cheese if the R/W and delete block sizes are small. The result is called fragmentation. There are two types of fragmentation: *file fragmentation* and *free space fragmentation*. File fragmentation refers to files that are not contiguous but rather are broken into scattered parts on the disc. Free space fragmentation refers to the empty space that is scattered about rather than being consolidated into one area of the disc. File fragmentation reduces read throughput, whereas free space fragmentation reduces write throughput. Interestingly, if data are formatted in large blocks, then the fragmentation is a nonissue. In fact, sequential A/V files and fragmented A/V (random access) files have about the same access

performance if the R/W block size is large according to the reasoning that has been developed. As a result, in many A/V systems, there is no need to defrag the arrays. This is good news because the defrag process is slow and might cause marked array performance problems for systems that operate 24/7.

In practice, it is the A/V equipment vendor's choice to fine-tune drive access and improve array throughput and latency. Many operating systems decide when (waiting adds latency for reads/writes) and with what block size (may be very small, 16KB) an array is accessed. For maximum performance, some A/V-centric applications bypass the OS services for managing the disc array access and rely on custom caching and R/W access timing.

The ideal model is for all access to be sequential (or random), large block transactions. However, it is often difficult to guarantee this, as small block transactions (metadata, proxy files, edit projects, etc.) and large A/V transactions are mixed for most applications. A mix of small and large block accesses can reduce the overall data throughput by up to 50 percent. For this reason, some vendors offer two different storage arrays for large A/V systems: one for large block A/V data and one for standard small block transactions.

HDD sequential R/W access has improved performance compared to purely random access for small block access. All modern drives have an internal cache for R/W queuing. So when a drive has, say, 10 random read commands in queue, then it can choose to reorder the access to optimize the read rate. A random command sequence of reads (or writes) is turned into a nearly sequential (approximates it) access operation. While it is true that read data may be returned out of order, the performance can be noticeably improved over the pure random access case. For most A/V applications, out-of-order returned data are resequenced easily by the application or other element. The extra logic needed to accomplish reordering is well worth the effort for the gains in performance.

The next section outlines common benchmarks for measuring storage system performance. Much of the reasoning that was developed in the previous sections is related to benchmarking. Figure 3A.9 is a way to visualize the factors that contribute to a performance benchmark.

### 3A.2.3 Storage System Benchmarks

Comparing storage system (SAN, DAS) performance and metrics can be a daunting task. By analyzing vendor data sheets and comparing specs, a reviewer may find a true apples-to-apples assessment to be nearly unattainable. This was the case until the Storage Performance Council ([www.storageperformance.org](http://www.storageperformance.org)) published its SPC-1 benchmark metric. The SPC-1 metric is a composite value of real-world environmental characteristics made up of the following:

- Demanding total I/O throughput requirements
- Sensitive I/O response times
- Dynamic workload behaviors



- Storage capacity requirements
- Diverse user populations and needs

SPC-1 is designed to demonstrate the performance and price/performance of storage systems in a server environment. The SPC-1 value is a measurement of I/Os per second (IOPS) that is typified by OLTP, email, and other business operations using random data requests to the system. The metric was designed to measure virtually any type of storage system from a single disk attached to a server to a massive SAN storage system.

Two classes of environments are dependent on storage system performance. The first is *systems based* with many users or simultaneous execution threads saturating the I/O request processing potential of the storage system. This type is typical of transaction processing applications such as airline reservations or Internet commerce. This spec is documented by SPC-1 IOPS. The second environment is one in which wall-clock time must be minimized (application based) for best performance. One such application is a massive backup of an array of data. In this case the total time needed to do the operation is crucial, so a minimum latency per I/O is crucial. This spec is documented by SPC-1 least response time (LRT) and is measured in milliseconds.

Of course, a storage system's performance is not the same as the performance of an individual HDD. It is good to review the basic performance metrics for a single HDD. As with an array, the key measures are access bandwidth, seek/latency, and capacity. A typical high-end HDD drive has the following specs:<sup>5</sup>

- 400 GB capacity (SCSI type) with a maximum sustained read transfer rate of 100 MBps. Maximum sustained rates are rarely achieved in the real world because they are measured on the outer tracks of the disc only. The inner track sustained data rate is about 70 percent of the maximum value. A sustained transfer requires a large file, continuously available without seeks or latency delays.
- Burst I/O rates >1 Gbps. This is the HDD internal buffer to/from the external bus transfer rate.

These are truly amazing specs and drive vendors keep pushing the speeds and capacity higher and higher. The usable, average transfer rates for large files may be 70 percent of the maximum value. Of course, for many A/V applications, system design demands specifying the worst case transfer rate, not the best or even average case. The HDD seek/latency spec is almost frozen in time (does not follow Moore's law), which affects the number of transactions per second that a drive can support.

---

<sup>5</sup> Note that *MBps* indicates megabytes per second, whereas *Mbps* indicates megabits per second. The *B/b* nomenclature is commonly misused in the technical literature, so beware.

## ACCESS DENSITY



Disk drive *performance* is improving at 10 percent annually despite storage *density* growing 50–60 percent annually. Raw disk drive performance is normally measured in total random I/Os per second and can surpass 100 I/Os per second.

Access density is the imbalance between storage density and performance. Access density is the ratio of

performance, measured in I/Os per second, to the capacity of the drive, usually measured in gigabytes. Access density = I/Os per second per gigabyte. For streaming video, I/Os per second is not always an important metric, because long I/Os are common.

### 3A.3 STORAGE SUBSYSTEMS

This section outlines several modern storage architectures. The following technologies are core to the modern AV/IT architecture. The following storage methods are discussed:

1. JBODs and RAID arrays
2. NAS servers
3. SAN storage
4. Object storage devices

In each case the technologies are examined in the light of A/V workflows. To fully appreciate storage requirements, check out the storage appetite for various video formats in Chapter 2. Fundamental to all storage subsystems are the actual storage devices. For our consideration, RAM, Flash, and HDD cover most cases, either alone or in combination. Optical is applied to some near-line and archive storage and is considered in Section 3A.7.3, “Archive Storage Choices.” See Appendix L for a review of the pros and cons of Solid State Disc technology.

#### 3A.3.1 HDD Capacity and Access Data Rate

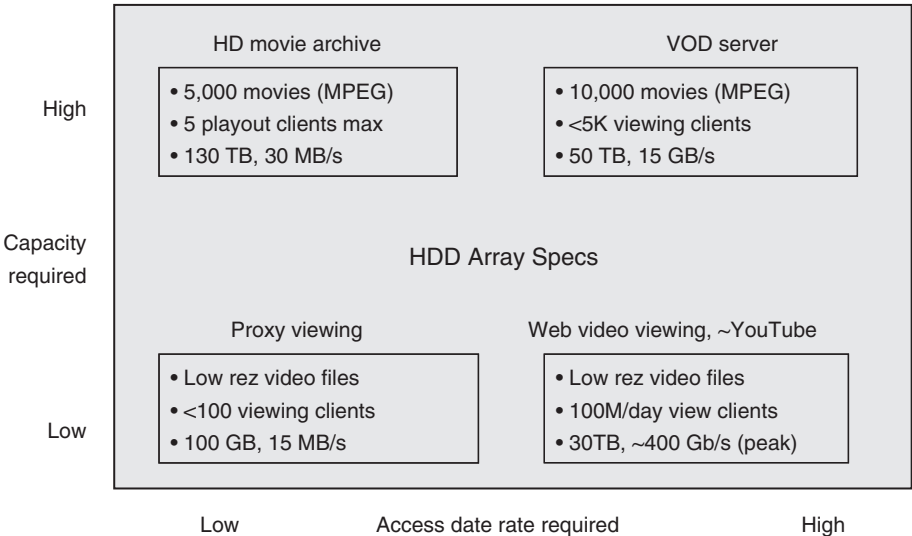
Every HDD provides storage capacity and an I/O delivery rate. These factors are both important and often misunderstood. We tend to think of a HDD as being defined by its capacity, as in “I just bought a 750GB hard drive at Fry’s Electronics.” That may be true, but that same drive also has an I/O spec. So, it is just as valid to say, “I just bought a 200 Mbps drive at Fry’s.” Disc bit density has increased about 60 percent annually since 1992, but storage device performance (random I/Os per second) is improving at only 10 percent per year. See the Snapshot on Access Density.

Drive vendors are focused on capacity more than increasing the I/Os per second or raw R/W rates. Some applications need capacity (a PVR), whereas

some need I/Os per second (a Web server with thousands of simultaneous clients). Here are some practical requirements for two applications:

- Centralized VOD server with the top 30 movies serving 1,000 individual viewers (cable TV premium service), each with PVR-like capabilities. Thirty 2-hr movies require only 135GB at 5 Mbps encode rate (one drive). The aggregate read access rate is 1,000 times 5 Mbps = 5 Gbps. If a single HDD has a spec of 15 MBps, then it takes 42 drives (all movies replicated on all drives) to meet the read access rates. For this case it is wise to use RAM storage. For 10,000 viewers, things get messy, and clever combinations of RAM and HDD are sometimes needed and/or some restrictions on PVR capability.
- At the other end of the scale is the case of 100,000 hr of 0.25 Mbps proxy video. Any of the content may be viewed by 200 desktop clients at once. To store the proxy files requires 11.25 TB total. If each HDD has a capacity of 250 GB (at 15 MBps), then it requires 45 drives to store 11.25 TB. The read bandwidth needs can be met with one drive (6.25 MBps total for 200 viewing clients).

Figure 3A.12 illustrates the two-dimensional nature of capacity versus access data rate. Four quadrants represent four application spaces each with their unique needs. As may be seen, applications range from requiring low capacity and low rate to those needing both high capacity and high rate. Knowing



**FIGURE 3A.12** Example HDD array capacity versus I/O rate needs per application.

what dimension, if not both, you may need to scale is vital for reduced upgrade headaches down the road. For each application the total HDD count needed depends on whether the storage capacity or access data rate is the driving factor.

So it is obvious that storage capacity is not the only HDD spec of importance when doing an A/V system design. The examples prove that the HDD disparity between required capacity and delivery rates can reach a factor of 40 or more for some designs. The extreme YouTube example in the figure requires ~40 HDD to support data delivery compared to one HDD for pure storage capacity needs. There are lively debates in the VOD design community over the best way to store the content. Some vendors use only HDDs, others use RAM only, and some offer a hybrid of HDD/RAM to meet the demanding needs of capacity and bandwidth.

### THE CASE OF THE MISSING 90GB



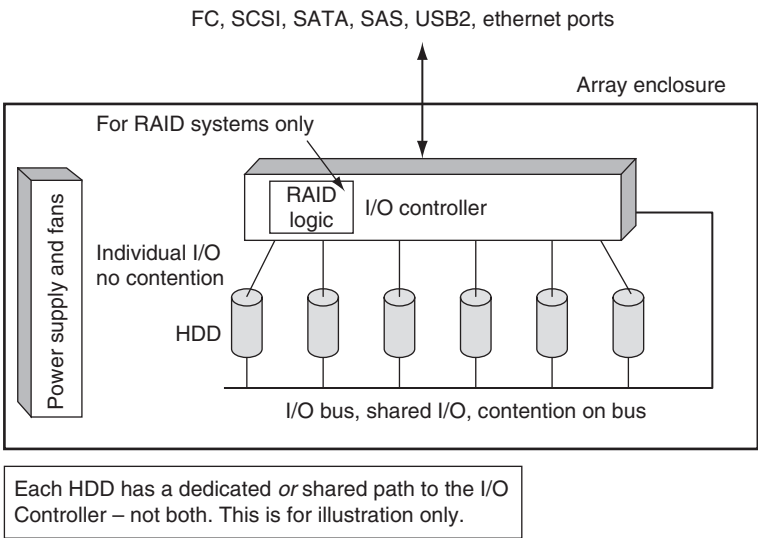
In computing, it is standard to use KB as representing 1024B. Likewise, MB is 1024KB, and GB is 1024MB. This is thinking<sup>6</sup> in a power of 2. So KB =  $2^{10}$  bytes, MB =  $2^{20}$ B, GB =  $2^{30}$ B, and TB =  $2^{40}$ B. Disc drive manufacturers, however, use KB to represent 1,000B, MB =  $10^6$ , and so on. A 300GB capacity HDD is exactly  $300 \times 10^9$  bytes, but when the drive is installed in most computer systems (Windows based, for example), its capacity is expressed using a K, M, or G based on 1,024B. Just a tad confusing.

For example, a 1TB HDD would show an installed capacity of 0.91 TB ( $091 \times 2^{40} = \sim 10^{12}$ ) so a 1TB (1,000B reference) HDD is the same as 0.91TB (1,024B reference). There is about a 90GB difference between the two methods. The “error” (–9 percent) is not trivial, but not an actual loss either. Do not confuse this with the difference between raw and formatted capacity—that is a true loss of useful capacity.

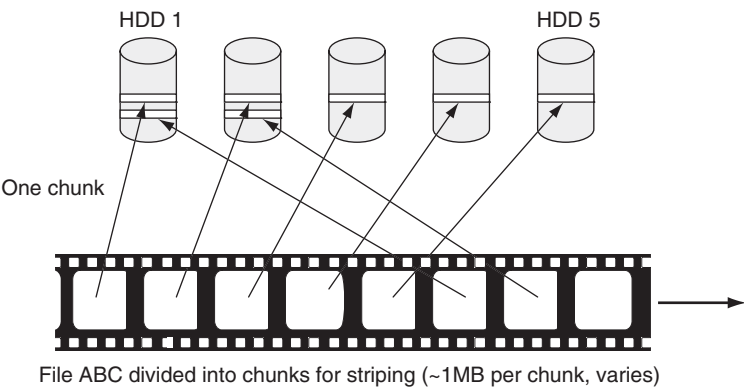
### 3A.3.2 Aggregate Array I/O Rates

An array filled with, say,  $N$  identical drives will deliver  $N$  times the storage capacity of one drive, but the aggregate I/O rate will not always be  $N$  times the I/O rate of one drive. Why? Drive I/O rate depends on how the I/O controller manages drive access. Is there an individual link from each HDD to the controller, or are all HDDs connected together on a common bus or arbitrated loop? Both methods are shown in Figure 3A.13. In the first case the drives have individual access to the I/O controller, so there is no contention among drives. In the second, all drives will share I/O bus resources; hence, simultaneous HDD access is impossible. Aggregate access rates depend on several factors, and file striping is one of them.

<sup>6</sup> See Appendix A for some  $2^N$  computing tricks.



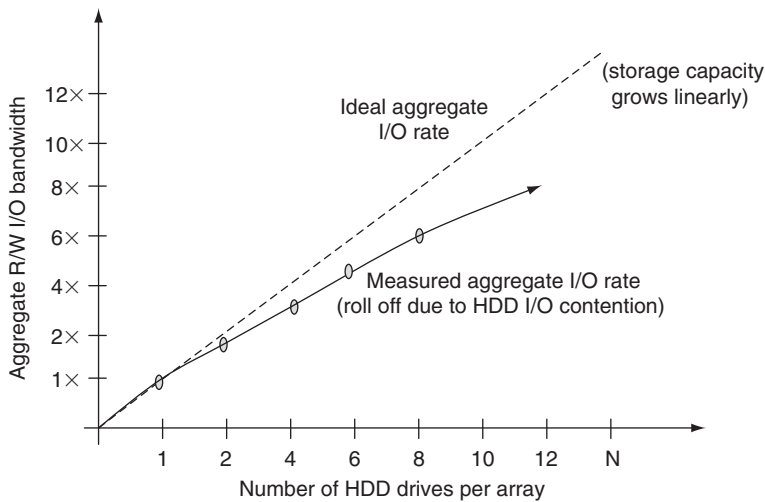
**FIGURE 3A.13** A simple JBOD or RAID array.



**FIGURE 3A.14** File striping example across five discs.

**3A.3.2.1 File Striping and Array Performance**

File striping increases access bandwidth by distributing the files across all drives in an array (or even across arrays). Figure 3A.14 shows one method of file striping to increase the aggregate access bandwidth. Because the file is divided among  $N$  drives, ideally, the file access rate is  $N$  times the case of the file being stored on one drive on average. Distributing a file across  $N$  drives allows more simultaneous clients to access the same file. One downside of this scheme is the vulnerability to file loss in the event of a HDD failure. If any one unprotected drive fails out of  $N$ , then the entire file is lost in practice. This increases the need to provide fault-resilient methods to store data (see Chapter 5). When



**FIGURE 3A.15** Storage array aggregate I/O bandwidth versus number of drives.

several users need simultaneous retrieval of the same file, accessing the discs out of order reduces HDD queuing delays. The methods used to avoid queuing delays are varied and beyond the scope of our discussion, but they can be managed to achieve good performance. Also, if the HDD I/O bus burst rate is much faster than its R/W platter rate, then some of the contention will be reduced.

Generally, if there is HDD I/O contention (I/O bus connection in Figure 3A.13), an array offers an access rate profile, as shown in Figure 3A.15. Figure 3A.15 is derived from measured results and shows a 25 percent reduction (75 percent of ideal) in aggregate I/O when there are eight drives sharing a common I/O bus. This phenomenon is also seen when files are striped across multiple arrays each with  $N$  internal discs. Your mileage may vary, since the rate reduction is a result of contention mechanisms that will differ among storage subsystems.

Optimizing and managing capacity and access rate at the HDD or array level are not child's play. Most A/V vendors tackle these issues in different ways. When evaluating a storage system, ask the right questions to fully comprehend what the real-world performance is.

The following summarizes the main points when adding capacity and/or I/O bandwidth to a system:

- Adding arrays or discs to an existing system adds storage capacity. However, the new capacity is made available to clients based on the configuration settings of SAN virtualization or a CFS if present.
- Adding arrays or discs to an existing system adds I/O capacity. However, the amount of R/W bandwidth available to clients depends on the presence of striping techniques and configuration of any SAN virtualization or CFS if present.

### 3A.3.3 General Storage Requirements

There are several factors to consider when selecting or defining a storage system.

- Usable capacity after accounting for redundancy (reliability) overhead.
- Usable R/W bandwidth considering the usage patterns: sequential, big block I/O, random or small block I/O. Remember, accessing small audio or proxy files (small block size usually) is markedly different from accessing high bit rate (large block rate usually) video files.
- Usable R/W bandwidth after accounting for RAID HDD rebuild methods.
- Reliability methods.
- Failover methods—done in A/V real time or not.
- Management methods—faults, warnings, alarms, configuration modes.
- A host of other features that may be significant depending on special needs.

When you are selecting a storage system, it is good policy to ask questions about these factors. There is no list that covers all user needs, so take the time and investigate to really understand the ins and outs of any system under investigation.

Many A/V facility owners/operators desire to use COTS storage (read cheaper) when configuring NLEs and video server nodes on shared storage. In general, however, these systems require A/V vendor-specified and tested storage systems (read expensive) to meet a demanding QoS. COTS storage vendors are happy to provide a generic product, but when it fails to meet the required QoS, they often will not make the needed upgrades. However, A/V vendors guarantee that their provided storage works well in demanding QoS environments. The bottom line is that COTS storage may work for some A/V workflows—if the shoe fits wear it—but A/V vendor-provided storage should always function as advertised. Next, let us examine the five basic storage subsystems found in various forms in most IT infrastructures: JBOD and RAID arrays, SAN, NAS, and object storage.

## 3A.4 JBOD AND RAID ARRAYS

There is nothing sophisticated about *just a bunch of discs* (JBOD), but you've got to love the name. It is the simplest form of a collection of disc drives in a single enclosure. Figure 3A.13 shows an example of a JBOD with either individual HDD links or a common I/O bus approach. It is really just an enclosure with power supplies, fans, drives, and an optional I/O controller board. The I/O controller may act as a gateway between the I/O ports (FC, SCSI, SATA, SAS, USB2, Ethernet ports) and the internal drive I/O. By definition, there are no

RAID functions. Advanced array features such as iSCSI I/O using Ethernet are possible but more often are found on RAID arrays. Admittedly, there is no precise definition of a JBOD, but simplicity and low cost reign. Pure JBODs lack the drive reliability that is often demanded, so their use is confined to areas where reliability is not of prime importance.

When JBODs grow up, they become RAIDs. No, this type of RAID is not bug spray. A RAID—Redundant Arrays of Inexpensive (or Independent) Discs—describes a family of techniques for improving the reliability and performance of a JBOD array. A RAIDed array can allow one (or two in some cases) of  $N$  discs to fail without affecting the storage availability, although the R/W performance may degrade. This concept is now accepted as *de rigueur*, and many hard disc arrays offer RAID functionality. See Chapter 5 for a detailed discussion of RAID techniques and categories.

### 3A.5 NAS AND SAN STORAGE

Network attached storage (NAS) and storage area networking (SAN) are discussed in Chapter 3B—a chapter dedicated to these technologies. In a nutshell, both technologies are used to provide storage to network attached nodes. A NAS provides remote resources for both sharing storage and sharing files. A SAN, in its most native configuration, provides only shared storage resources. With the use of a CFS, discussed earlier, a SAN may be configured to share files among attached nodes. Clients attach to a NAS using Ethernet usually. Clients and servers attach to SAN storage using Fibre Channel and, more recently, Ethernet. General-purpose servers (including NAS servers) often use SAN storage. They both can offer excellent A/V performance and are the bedrock of many AV/IT systems.

### 3A.6 OBJECT STORAGE

A novel class of storage is object based. It differs from traditional file (NAS) or block-based (SAN) storage in several ways. Object storage devices (OSDs) store data not as files or hard-addressed blocks, but as objects. For example, an object could be a single database record or table—or the entire database itself. An object may contain a file or just a portion of a file. An OSD is a content-addressable memory (CAM). If you provide it with an identifier (metadata fingerprint), it will return the content represented by the ID. The fingerprint is formed using a hashing algorithm for generating a unique 128 B (or similar) value that is used to identify and retrieve data.

Imagine that a 30 s video program has a fingerprint ID of value  $X$  and that this value uniquely represents it. If only one bit changes anywhere in the video, say due to a pixel change, the value of  $X$  changes. In practice, a requesting client asks for file ABC, which translates to a pointer of value  $X$ , and the OSD locates file ABC using this pointer. Depending on the implementation, clients



may access the storage using CIFS/NFS (discussed in the next chapter) or via a custom API provided by the OSD storage vendor. OSDs are finding applications as near-line or secondary storage, so they will not be replacing traditional high-performance RT storage any time soon.

Objects are stored on OSDs that contain processors, RAID logic, network interfaces, and storage hardware. Each OSD manages the objects as it deems necessary. The OSD hides the “file” layout, addressing, partitioning, and caching from a requesting client. Objects may reside on one OSD or be partitioned across a network of OSDs. This abstraction is practical for some file types such as X-rays, images, videos, email archives, and a myriad of documents that are normally recovered whole. Partial access is also allowed. Using metadata to track and manage files adds flexibility in scalability, performance, location independence, authenticity, and reliability compared to traditional addressable storage. The value of the metadata fingerprint also maintains the integrity of the file against changes.

Linux server clusters are one means to implement OSD systems. Among the vendors in this space are Panasas ([www.panasas.com](http://www.panasas.com)) and Permabit ([www.permabit.com](http://www.permabit.com)). Other companies are attacking the market with standalone storage systems (not based on Linux clusters). For example, EMC’s Centera and HP’s Integrated Archive Platform are object-based stores. Network Appliance ([www.netapp.com](http://www.netapp.com)) offers NearStore Appliance (not object based). NearStore is going after the same market segments (near-line, secondary storage) as Centera but without using object storage.

Object storage is a new model and will find some niche applications over time. A/V data types are ideal candidates to be treated as objects. Sample systems that may use OSDs are a post house or film studio with thousands of hours of digital content. Each piece of material (or any derivatives) will have a unique fingerprint that may be used for asset tracking. Because OSD systems can scale to >100 TB with NSPOF reliability, they are also ideally suited for mission-critical near-line storage.

The SNIA ([www.snia.org](http://www.snia.org)) has defined XAM (eXtensible Access Method) as the preferred way to access object stores. XAM provides an API for importing/exporting files to an object store. Note that access to an object store does not use the ubiquitous NFS or CIFS protocols. So, the XAM API enables a standard method to achieve cross-vendor compatibility for store access. XAM also enables users to associate metadata for each stored object. Traditional file systems don’t register file-associated metadata. Knowing about the stored object (who, what, when info; IPR rights windows; legal aspects; and so on) is a huge value to a media organization. Expect to see XAM used by enterprise media companies.

### 3A.6.1 Deduplication

One special case of object storage is called *deduplication*, or single instance storage technology. This is a method for reducing storage by eliminating redundant data found across multiple files. Only one unique instance of the data object is retained

in storage. Redundant data are replaced with a pointer to the unique data segment. For example, a typical email trail might contain many repeated textual segments. When you add new text to a received email and then send a reply, a new file is created that may contain 90 percent of the text from the previous email in the trail.

Squeezing out the repeated text (or other object) across many files can yield compression rates of  $20\times$  to  $50\times$  to in many business environments, according to the Enterprise Strategy Group ([www.enterprisestrategygroup.com](http://www.enterprisestrategygroup.com)). Don't expect any gain for single compressed A/V files; the air has already been expelled.

### 3A.7 HIERARCHICAL AND ARCHIVAL STORAGE

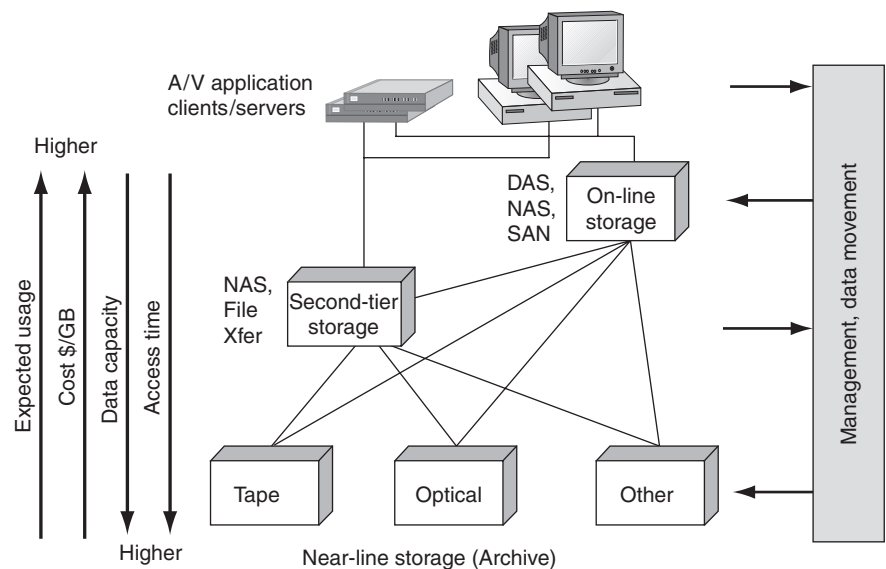
Installed digital storage is expected to grow 50–70 percent per year for the foreseeable future. (See Appendix D.) The economics of storing all this on RAM or even hard disc is untenable. As a result, IT has developed a hierarchy or pyramid of storage to balance the needs of users and owners of digital data. Each step in the hierarchy trades off access rates, capacity, and cost/GB. These three metrics are the key drivers behind the idea of tiered storage. Next, we consider a brief analysis of the trends in tiered storage.

Figure 3A.16 presents a simple view of a three-tier storage system. The IDC classifies hierarchical storage into five tiers for normal business systems. For A/V, we will consolidate into three layers. Some ranks may not be present in all systems (no near-line, for example), but our analysis considers the full-featured case. At the top are the application clients and servers that demand fast data access times with unlimited access to stored data. Online SAN or NAS storage is typical with speedy access times (in the A/V real-time sense—*mission critical*) but with limited storage. Lower in the chain is second-tier storage. The trade-off here is giving up low access times for less expensive (\$/GB) storage with more capacity. Second-tier storage may not guarantee RT access under all conditions. At the bottom is the archive layer and near-line storage. Here, excellent capacity and lower \$/GB are the key metrics, with access times being very slow compared to HDD arrays. It is not uncommon for mission-critical storage to cost  $10\times$  more in a \$/GB sense compared to slower offline (archive, many hour access time) storage.

In addition to the economic needs of trading off access rates, capacity, and cost, there are several business uses of tiered storage:

- Long-term archive
- Daily server backup/restore
- Disaster recovery copies
- Snapshots creating a point-in-time copy of data for quick recovery
- Data mirrors for regional access or improved reliability
- Replication—like mirroring but writes are asynchronous

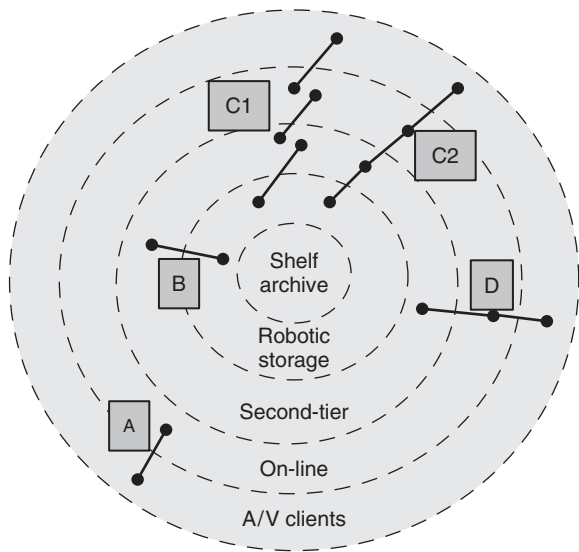
Most of these factors are discussed in Chapter 5.



**FIGURE 3A.16** *The hierarchy of storage.*

**3A.7.1 Data Flows Across Tiered Storage**

The flow of data between the different storage elements is outlined in Figure 3A.17, the storage onion. Four sample flows are identified. Flow “A” is the common client to DAS/NAS/SAN storage connection. In most cases the client controls access to the online storage pools. In case “B,” data move between the archive and a second-tier store. The client layer is unaffected by the transfer of content when



**FIGURE 3A.17** *The storage onion—sample data movement paths.*

it occurs. However, some director (controller) moved the files, perhaps based on a JITFT schedule, for later use by a client/server or transfer across a WAN.

In flow “C,” data are moved from archive to client (or in the opposite direction) in either separately completed steps (C1) or as part of a continuous data flow (C2). In the first case, a director moves data between stores in steps as needed by schedules or other requests. There may be a wait period (minutes to days) between steps in the chain. In the second case (C2), a director moves data from the archive to second-tier to online in one continuous step for use by the A/V client. One example of this second case is a request for an ASAP A/V playback of some archived material. The playback file is located on the archive and moved to the second-tier store. As the second-tier store is being loaded, the target file is moved to the RT online store. As the online store is being loaded, playout of the file may commence. Sure, it may be more logical to move the file from the archive directly online, but let us use the C2 flow as an example only.

Now, the C2 flow is tricky, but it can be implemented with proper attention to data streaming rates. Compared to C1, C2 requires a high-performance QoS for each element in the chain and strict attention to the timing of the flow. From request to client playback start may take 15 s to several minutes depending on the archive type (optical or tape). The flow of data along the C2 path must be continuous and never “get behind” the playout consumption data rate; if so, the playout data will dry up. The data flow across all devices in the chain must, on average, meet RT playback (or record) goals. Frankly, the continuous C2 flow is an unlikely workflow in the real world. The stepped version C1 is practical and done everyday in workflows like those at Turner Entertainment’s Cartoon Network (see Chapter 10). Of course, C1 may require ~15 min (or less with fast transfers) to transfer a 5-min file along the path. It is more likely to see an archive to online continuous transfer (skip the second-tier stage) if there is pressure to play out ASAP. The bottom line is that both C1- and C2-like flows are practical in the real world.

The last flow is task “D.” This is a traditional two-stage file transfer from second-tier to online to playout. Typically, automation software schedules the transfers (JITFT mode) between near-line and online and initiates the playout at the client. D may be either of types D1 or D2, as with the C flow. If D is implemented, D1 is the more common flow.

Some AV/IT workflows use all of these methods (A–D) in harmony. However, not all A/V facilities have need for an archive, but some do. For example, Turner Entertainment uses a Sun StorageTek PowderHorn tape archive just to store its 6,000+ cartoon library. Turner uses a true three-tier storage approach (C1) for the Cartoon Network and a two-tier approach (D1) for other networks.

The cost to manage data can be enormous, and it is one of the biggest issues in IT today. A/V workflows do not escape the management burden. Despite the nearly 70 percent increase in data annually, there is only a 30 percent increase in the ability to manage these data. So what are the best practices to manage data?

### 3A.7.2 Managing Storage

Over the years different strategies have emerged to manage storage. For many years hierarchical storage management (HSM) has been the mainstay for large enterprise data management. The most current thinking is centered on information life cycle management (ILM). ILM is the process of managing the placement and movement of data on storage devices as they are generated, replicated, distributed, protected, archived, and eventually retired. ILM seeks to understand the value of data and migrates them across storage systems for the most cost-effective access strategy. A report from Horison Information Strategies [Horison] states that enterprises need to

*Understand how data should be managed and where data should reside. In particular, the probability of reuse of data has historically been one of the most meaningful metrics for understanding optimal data placement. Understanding what happens to data throughout its lifetime is becoming an increasingly important aspect of effective data management.*

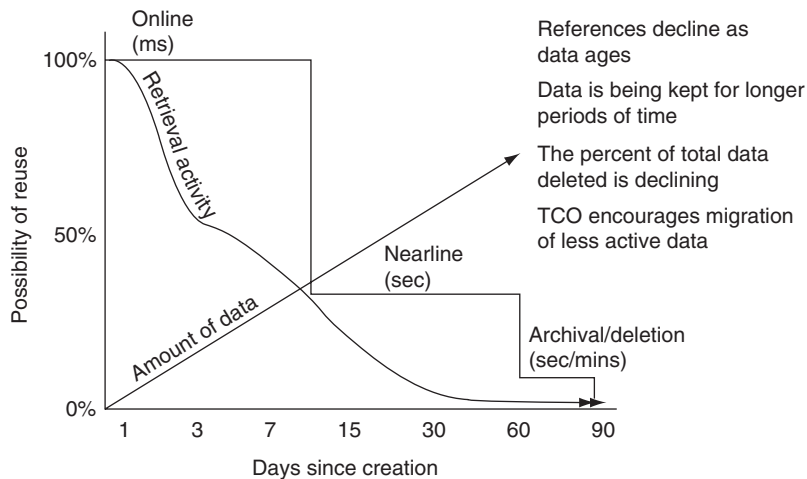
The concepts behind ILM are useful for any A/V facility design, expansion, or rethinking of current storage strategy (see [Glass]).

Implementing an ILM strategy requires a combination of process and technology. Some process-oriented questions are as follows:

- What is the value of data as they age?
- Do data become more or less valuable?
- How long should we keep data?
- What storage policies can we apply to our data?
- What is the right balance of our data across the hierarchy to optimize ROI and user experience?
- Do we understand the different data types that drive our business?

Figure 3A.18 posits one possible scenario of data reuse as they age. Figure 3A.18 is general, and various data types exhibit different access signatures. Knowing that access rates decline over the life of data allows automation logic to migrate data to less costly storage. A/V data typically fall into a similar pattern.

Several companies specialize in managing the A/V data archiving process. EMC (AVALONidm), Front Porch Digital (DIVArchive), Masstech Group (MassStore), SGL (FlashNet Archive Manager), and SGT are among the few that cater to the special needs of the A/V industry. They provide software to manage the migration across the hierarchy and R/W from storage robotics. What makes managing storage special in A/V facilities? A few aspects are A/V data flows involving very big files (15GB movies), partial file restore, appreciation for metadata, unusual A/V gear compared to enterprise devices, and time-critical delivery. Some traditional A/V equipment automation companies manage data movement between near-line and online storage as well.



**FIGURE 3A.18** Data access rates decline with age of data.  
Source: Horison

Most large-scale facilities need both storage management and control automation software solutions to meet all their A/V data flow needs. As online and offline storage capacity increases and prices drop, fewer designs demand a full three-tier solution.

Many large broadcasters use tape archive systems. Sometimes the content is owned by the broadcasters (movies, dramas, for example), and they expect very long-term storage. Alternatively, the archive may cache materials for several months or years for reuse later. One example of programming reuse is at KQED in San Francisco, a PBS member station. This station stores up to 12,000hr of PBS programming locally in a Quantum archive for reuse as needed. See Chapter 10 for a case study on KQED.

### 3A.7.3 Archive Storage Choices

Offline archive devices can be complex and usually involve some or all of the following:

1. Tape and/or optical removable media
2. Drives to control and R/W the media
3. Robotics to insert/remove media from drives
4. Housing for controllers, media, drives, and robotics

There has been a debate about what is a true archive media format. Some vendors tout 7–15 years media life as an archive format, whereas another will say that true archive needs a 35+ year media life. Although a worthwhile discussion, let us bypass this argument and treat any data tape or optical media as an archive format.

Archive devices are divided into at least three different camps: tape (helical and linear heads), optical disc, and holographic. In the mainstream, tape is the most popular, followed by optical. Some vendors manufacture all the components in the value chain, whereas others offer only pieces. For example:

- Sony manufactures media, robotics, and enclosure for the PetaSite tape storage system. The PetaSite supports up to 1.2 Petabytes ( $10^{15}$  bytes is one Petabyte). One Petabyte of capacity can store  $\sim 100,000$  movies at a 10 Mbps compressed rate.
- Spectra Logic manufactures the T-Series tape robotic system. It supports SAIT and LTO drives and media. This system scales from 50 to 680 tapes and 24 drives.
- Imation manufactures tape media, including format support for DLT, Ultrium (LTO), and SAIT media.

### 3A.7.3.1 *Magnetic Tape Systems*

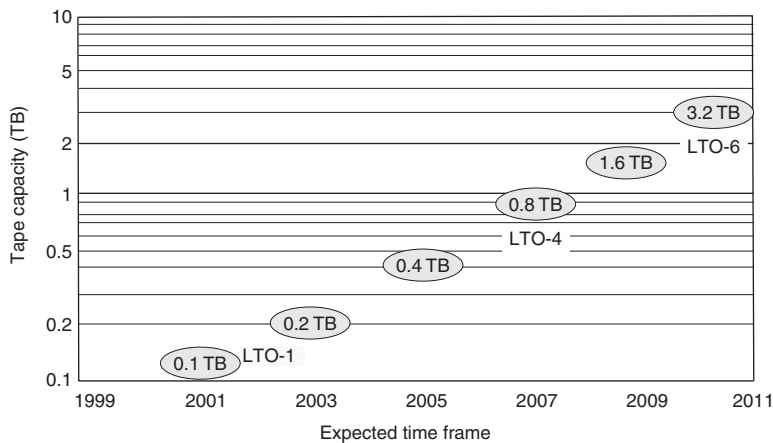
The magnetic tape industry produces more than 10 different recording formats with nearly 15 automated tape robotic suppliers. Single cartridge capacity is expected to approach  $\sim 3$  terabytes in  $\sim 2011$  based on planned improvements in recording density. The automated tape market is expected to grow as archive and disaster recovery requirements increase due to all things going digital (Moore 2002).

Out of the total, there are four top players in high-end archival tape formats. Sony offers AIT/SAIT (Super Advanced Intelligent Tape). A consortium of several vendors created the LTO (Linear Tape Open) format branded as Ultrium. Sun's StorageTek brand offers the  $9 \times 40$  tape format family. Quantum offered the DLT/SDLT (Super Digital Linear Tape) format for many years but discontinued it in 2007 in favor of LTO.

The current format leader is the LTO-4 with 800GB native capacity at 120MBps R/W rates. Figure 3A.19 shows the roadmap for the LTO format, the likely end winner in the format wars.

Also, both Sun and IBM have created "LTO-like" drives with native capacity at 1TB in 2009. LTO uses a "linear serpentine" recording format that stripes the data back and forth across the tape rather than the helical head scan format that was common for many years. For the economy minded, Sony and HP have co-developed a version of Digital Audio Tape (DAT) that has a native capacity of 320GB at 86GB/hour rates.

SMPTE is developing a tape layout format called the Archive Exchange Format (AXF). Any data tape that vendor A writes to using AXF can be read by vendor B. Before AXF, end users were locked into one vendor for all archive management needs. This is an untenable situation given the value of the media on the tape and possibility of the vendor defaulting.



**FIGURE 3A.19** LTO Roadmap (LTO-5,6 estimated dates).

In closing, a complete analysis of storage system performance involves metrics beyond tape formats. Other metrics include storage capacity per square foot and cubic foot of enclosure space, aggregate R/W throughput, tape retrieval time, search time, overall capacity, cost per TB (or per MBps), and line power needed per TB. Depending on your workflow, space, and power needs, these metrics have varying value.

### 3A.7.3.2 Optical Systems

Optical systems have always lagged behind magnetic tape in capacity and data rate, but recent products show promise for low-cost archives. Optical discs come in several flavors, including

- DVD with a single-sided, dual-layer capacity of 8.5 GB (single layer is 4.7 GB).
- Blu-ray disc is the crowned DVD successor format for HD. Secondly, the format serves as pure data storage. See Figure 3A.20. At least one vendor supports 8× burner speeds in 2009. Plasmon, HP, and others offer Ultra Density Optical (UDO) with up to 60GB density per disc.

Optical R/W rates are much less than for tape, but their access times are better by a factor of 50 or more. Also, the robotics for managing optical discs tends to be less expensive than tape robotics because the disc is very light and easy to maneuver. The optical-based archive is finding applications in A/V facilities worldwide. One example of an optical storage system is the TeraCart from Asaca Corp. It supports SAN and NAS connectivity and Blu-ray discs with an enclosure capacity of 9.8TB in only 4 square feet of footprint. More than 2,000 hr of 10Mbps programming can be stored in one relatively small unit.



Drive speed	Data rate		Write time for Blu-ray Disc (minutes)	
	Mbit/s	MB/s	Single layer	Dual layer
1×	36	4.5	90	180
2×	72	9	45	90
4×	144	18	23	45
6×	216	27	15	30
8×	288	36	12	23
12×	432	54	8	15

\* Theoretical

Video encoding: MPEG-2, AVCHD, VC-1

**FIGURE 3A.20** Key parameters of the Blu-ray disc format.

Up to eight libraries may be connected to act as one unit. Figure 3A.21 shows this unit, the AM420PD.

**3A.7.3.3 Other Archive Devices**

This section reviews some new and promising methods to archive data. One that is starting to gain momentum is Massive Array of Inactive Discs (MAID). The concept behind MAID is HDD-based storage designed specifically for write once, read occasionally (WORO) applications, where the focus is on infrequent access rather than I/Os per second. Applications that only occasionally access data permit the majority of discs not to spin, thereby conserving power, improving reliability, and simplifying data access. A typical MAID architecture may have only 1 percent of data spinning at any one time. Data access is measured in milliseconds to seconds—the time it takes to spin up a drive. MAID will have application in broadcast facilities where archived data are accessed only occasionally—say, to load a day’s playlist. MAID is not used for long-term, deep archive.

Holographic storage has been a promising technology for years. Until recently, it seemed as though it would languish forever in the laboratory. However, InPhase Technologies ([www.inphase-technologies.com](http://www.inphase-technologies.com)) has recently developed a WORM product (Tapestry) with 300 GB of capacity at 20 MBps transfer rates with a roadmap to 1.6 TB/120 MBps on a single platter. Several of the value propositions for holographic storage are leaders in density per cubic foot (32 TB/cubic foot), 50-year media life (magnetic tape is 8 to 20 years), and reading rates faster than Blu-ray (factor of ~2× in 2008). Not to be ignored is the cost of the media itself. Holographic discs promise to be the low-cost leader compared to tape or other optical storage systems. Of course, holographic storage is immature and needs industry experience before it takes any appreciable market share from tape or optical.



**FIGURE 3A.21** *Asaca TeraCart using ProDATA storage (9.8TB).*

A promising approach to achieve storage is the Pergamum project developed at the University of Santa Cruz (Pergamum). It uses a distributed network of intelligent, MAID-based, disk-based storage appliances (bricks) that store data reliably and energy-efficiently ( $\sim 2.5\text{W}$  per TB stored). Pergamum uses both intra-disk and inter-disk redundancy to guard against data loss. Each brick is Ethernet connected to create a unit that allows for obsolescence replacement with other commodity bricks.

### 3A.8 IT'S A WRAP: SOME FINAL WORDS

Storage technology is at the heart of all AV/IT systems. A good understanding of these concepts will help you better evaluate vendors' products. Plus, you will be able to decipher vendor speak when it comes to their storage offerings and connectivity. Expect to see the "pocket server" in a few years as HDD density continues to head into the stratosphere. Storing 500 hr of content on one drive is not far off, and small A/V media clients scattered throughout a facility may

become commonplace. It is not hard to imagine an A/V infrastructure with centralized storage (a mix of online and near-line) and media clients located across the LAN and WAN.

The next chapter outlines three common methods for clients and servers to connect to storage. The trilogy of DAS, SAN, and NAS are the basis of modern storage connectivity. By aggregating the information in this chapter with the next, you will develop a good foundation of the essentials of storage systems.

## REFERENCES

GlassHouse Technologies, White Paper, *Uncovering Best Practices for Storage Management*, [www.glasshousetech.com](http://www.glasshousetech.com), 2002.

*Information Lifecycle Management*, Horison Information Strategies, 2003, [www.horison.com](http://www.horison.com).

Moore, Fred (2002). *Storage Manifesto*. Boulder, Colorado: Spectra Logic Inc.

Mark W. Storer et al. Pergamum: Replacing Tape with Energy Efficient, Reliable, Disk-Based Archival Storage, *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST '08)*, February 2008, pages 1–16.

# Storage Access Methods

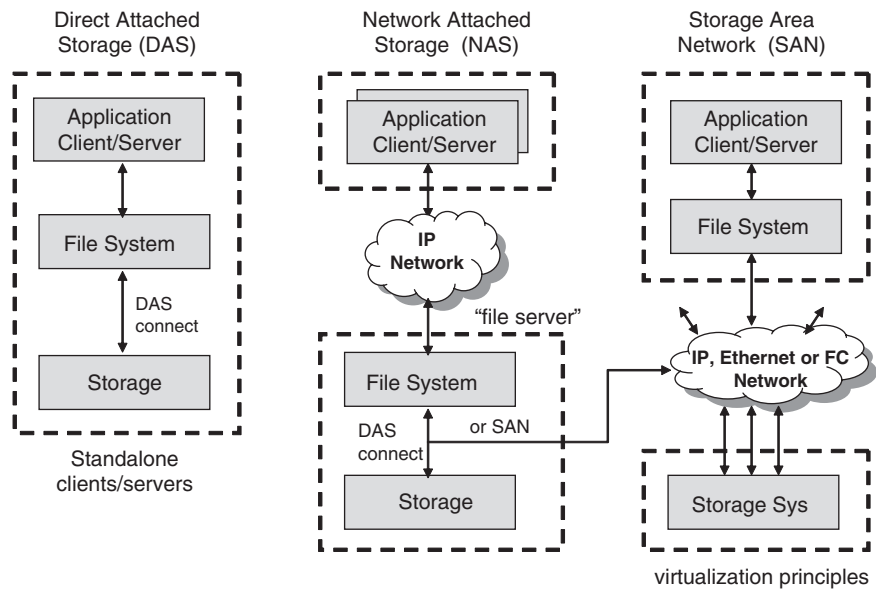
## CONTENTS

<b>3B.0</b>	<b>Storage Connectivity: DAS, SAN, and NAS</b>	<b>122</b>
<b>3B.1</b>	<b>Direct Attached Storage</b>	<b>123</b>
3B.1.1	Protocol Soup	124
3B.1.2	HDD I/O Connectivity and Drive Types	126
3B.1.3	ATA and SCSI I/O Convergence	128
3B.1.4	Remote DMA: The Next Frontier	130
<b>3B.2</b>	<b>Storage Area Networks</b>	<b>131</b>
3B.2.1	Form Follows Function	133
3B.2.2	The Fibre Channel-Based SAN	134
3B.2.3	Hybrid SANs: Merging FC and IP	137
3B.2.4	IP SAN Technology Choices	138
3B.2.5	TCP/IP SAN Performance	141
3B.2.6	SAN with Virtualization and Cluster File Systems	142
3B.2.7	SAN Vendor Overview	142
<b>3B.3</b>	<b>Network Attached Storage</b>	<b>143</b>
3B.3.1	NAS Attach Protocols	144
3B.3.2	NAS Vendors and Product Features	145
3B.3.3	A/V-Friendly NAS Connectivity	147
3B.3.4	NAS and Server Clustering	148
3B.3.5	NAS, SAN, and the Future	150
<b>3B.4</b>	<b>Caching Methods</b>	<b>153</b>
<b>3B.5</b>	<b>It's a Wrap: Some Final Words</b>	<b>155</b>
	<b>References</b>	<b>155</b>

3B.0 STORAGE CONNECTIVITY: DAS, SAN, AND NAS

Storage is a huge topic so the coverage is divided across two chapters, this one and Chapter 3A. The preceding chapter outlined the basics of storage systems, and this one adds the dimension of connectivity between clients/servers and storage. You may need to bounce between the two chapters because many of the concepts discussed are so intimately related. This chapter studies the three common methods for servers and clients<sup>1</sup> to connect to a storage subsystem. Common ways to illustrate the three key methods are shown in Figure 3B.1. This illustration is a bit poetic, as in reality some of the layers are very thin. But that is okay; it is a great place to start our discussion. In the final section of this chapter, Figure 3B.1 is revisited with some flesh added to more fully compare the three methods.

Of the three, direct attached storage (DAS) is the simplest but least flexible. It provides block-based storage access to a directly attached client/server. DAS is a common choice for applications that require high-performance local storage. The storage area network (SAN) replaces the simple connectivity of DAS with a switched fabric. Both clients and servers are connected to a fabric, allowing for scalable performance and capacity. It, too, provides



**FIGURE 3B.1** DAS, SAN, and NAS compared.

<sup>1</sup> Clients and servers are often collectively called the *host computer* when connecting to storage.

native, block-oriented access to storage. Third, network attached storage (NAS) uses networking to attach clients to file servers. NAS provides remote storage with an included file system. In NAS architecture, multiple hosts can share files. The distinction between SAN and NAS can be confusing at times, so the differences are clearly outlined in this chapter. The analysis emphasizes the features and aspects that are especially relevant to A/V systems. Let us start with DAS.

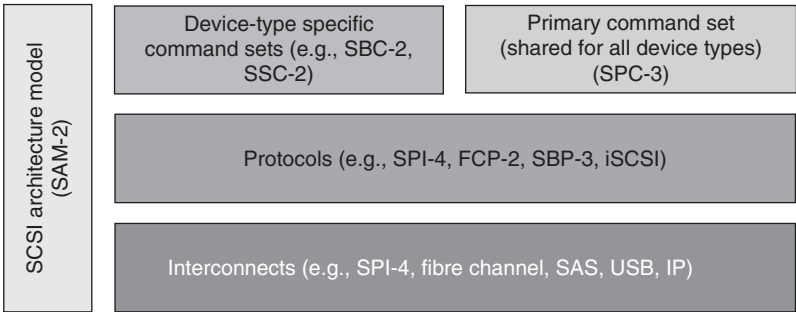
### **3B.1 DIRECT ATTACHED STORAGE**

This section provides a moderately detailed overview of direct attached storage interface technology. Depending on your interest level, this section may provide more information than you need. If your goal is to understand DAS buzz words and what is behind them, then this section is for you; if not, skip ahead to Section 3B.2 on SAN technology.

DAS is the most common form of external storage connectivity. In the recent past, the venerable SCSI connection was the most popular DAS method, and many servers and workstations have SCSI ports for connecting to external storage. A host computer connects directly to an array enclosure over a DAS link with little or no networking in between; it is a direct attach model as the DAS name implies.

The traditional SCSI interface was introduced as a method of connecting multiple peripherals to computers. It was developed by the T10/11 working groups of the InterNational Committee for Information Technology Standards (INCITS). It is based on a parallel bus structure, with each attached device having a unique ID (or address). The SCSI bus will support up to 15 devices plus the host controller and can transfer data at burst speeds of up to 320 MBps (Ultra 320). Because of the multiple device support, extended cable length (up to 6 m), and excellent transfer rate, the SCSI interface has been used to connect external devices such as scanners, CD duplicators, and HDD storage enclosures.

SCSI started life (1986) as a single parallel interface that included three closely connected elements: physical, protocol, and command specifications. Since then, it has matured into a family of standards, including specs for each of these independent levels. Figure 3B.2 outlines the essentials of the standards. At the top level are the command structures, including device-specific commands such as SCSI block commands (SBC) and SCSI stream commands (SSC) and the more generic SCSI primary commands (SPC). The mid-level contains protocols (that carry the upper-layer commands) for a variety of physical links. The bottom level lists the common physical links. Notice that Fibre Channel carries SCSI commands, as does USB. Note, too, that the physical level is mapped to select upper-layer protocols. As a result, Fibre Channel carries FCP, and USB carries SPB, and so on.



**FIGURE 3B.2** SCSI standards structure.  
Source: SCSI Trade association.

3B.1.1 Protocol Soup

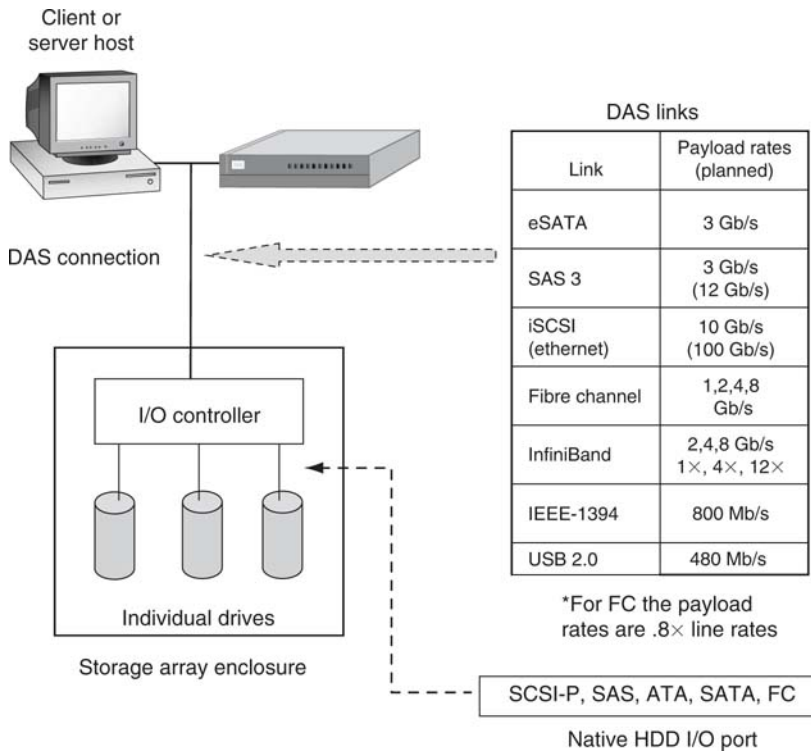
The following list outlines three legacy and three newer protocols that are key to the IT infrastructure. The first three are mature technologies with many vendors offering products for the market. Some expect that serial attached SCSI (SAS) and IP SCSI (iSCSI) will become the market leaders over the long run for enterprise applications. InfiniBand RDMA is finding niche use in high-speed device clustering applications.

- SCSI parallel interface (SPI-4 is the latest for support of Ultra 320)
- SBP-3 (serial bus protocol) for use over IEEE-1394 and USB-2 links (personal and small office use)
- Fibre Channel protocol (FCP) over Fibre Channel physical (more on this later in the chapter)
- iSCSI over Ethernet/IP physical (SAN)
- Serial attached SCSI (SAS), which is defined at the command, protocol, and physical layers
- SCSI RDMA protocol over InfiniBand (RDMA = Remote DMA)

In some cases, the physical links are defined by the SCSI standards group, as with traditional parallel SCSI or the newer serial attach SCSI links. In other cases, the physical link is defined by others (IEEE’s Ethernet, USB-2, IEEE-1394), but the protocol and command layers are SCSI standards. So when someone says that a link is *SCSI compatible*, you need to ask questions to confirm what is actually meant. Is it at the command, protocol, and physical level or some subset of these?

The older parallel SCSI-Parallel interface has lost favor due to several factors: short distance span, bulky cable/connectors, connector reliability, lack of networkability, cable cross-talk, and limited data throughput. SAS is replacing legacy SCSI links.

Figure 3B.3 shows the universe of DAS connectivity. In fact, there are more choices available (IBM offers several captive alternatives) but these will be the



**FIGURE 3B.3** DAS connectivity examples.

most common ones moving forward. Of note, InfiniBand is not yet applied for DAS but it may in the future. Too, eSATA (the “e” is external) is equivalent to SATA but with a special rugged connector and slightly tighter voltage tolerances to allow for a 2 meter span. A variety of vendors sell storage arrays with support for these link types.

Figure 3B.4 shows a USB, ATA-based personal storage array from Maxtor. Using USB-2 connectivity, you may DAS attach up to 129 drives enclosures to the USB serial path (needs bridges). This model is the OneTouch II.



**FIGURE 3B.4** Example of USB DAS attached storage appliance.  
Image courtesy of Maxtor.





**FIGURE 3B.5** Example of large disc array for DAS or SAN attach.  
*Image courtesy of DataDirect Networks.*

At the other end of the scale, Figure 3B.5 shows a DataDirect Networks S2A HDD-based storage array. A single unit can support 6GBps R/W, 1.2TB capacity, intermix of SAS and SATA drives, and RAID 6 reliability. This unit may be DAS attached, but for most applications it is shared with several servers and/or clients using SAN technology.

**3B.1.2 HDD I/O Connectivity and Drive Types**

Closely allied with DAS connectivity is the native HDD I/O interface port. This section outlines the common HDD ports and drive types. The native HDD I/O port type is not necessarily the same as the DAS link type shown in Figure 3B.3. Table 3B.1 outlines common native HDD I/O connectivity choices.

**Table 3B.1** Native HDD I/O Interface Connectivity

HDD I/O Port Type	Comments
SCSI-P	Also used as a DAS link <6 m
Serial attached SCSI (SAS)	Serial replacement for SCSI-P; Also a DAS link—10 m;
ATA-P (ATA commands)	Only for HDD I/O—<1 m
Serial ATA (SATA)	Serial replacement for ATA-P; Only for HDD I/O—<1 m
Fibre Channel (SCSI commands)	Also used as a DAS link—<10 km optical

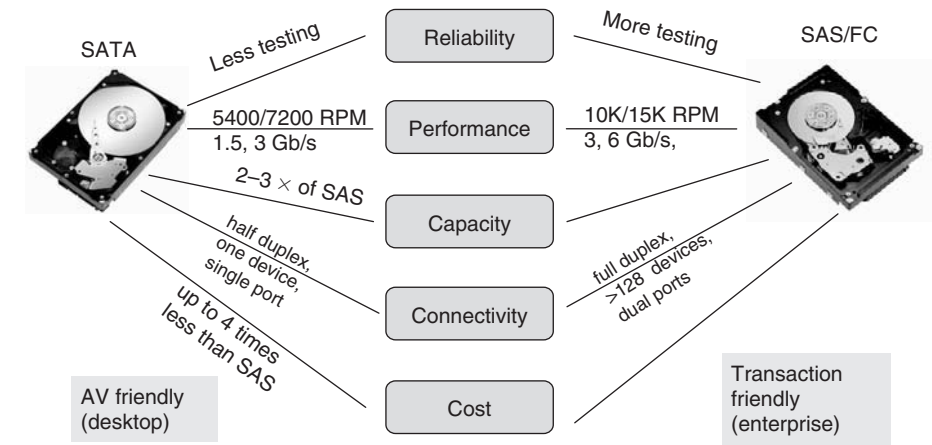
Drive -->	Serial ATA	Serial attached SCSI	Fibre channel AL
<b>Performance</b>	Half-duplex	Full-duplex	Full-duplex
	1.5 and 3 Gb/sec 6.0 Gb/s planned The 3 Gb/s speed is not called SATA II	3.0 and 6 Gb/sec 12 Gb/s planned	1, 2, 4 and 8 Gb/sec
<b>Connectivity</b>	1 m internal cable 2 m external eSATA	>6 m external cable	>15 m external cable
	One device	>128 devices	127 devices Loop or loop switch
	SATA only	SAS and SATA	Fibre channel only
<b>Availability</b>	Single port HDDs	Dual-port HDDs	Dual-port HDDs
	Single-host Point-to-point	Multi-initiator Point-to-point	Multi-initiator Shared media or point-to-point
<b>Drive model</b>	Software transparent with Parallel ATA	Software transparent with SCSI	Software transparent with SCSI

**FIGURE 3B.6** HDD I/O: Comparing three serial port types.

ATA (parallel form) and SATA are dedicated for HDD I/O only and are not normally deployed outside a storage array enclosure. The other three link types are used for native HDD I/O and DAS connectivity. An HDD I/O type and its array enclosure I/O type do not need to match in any way. It is possible to have internal SATA drives with Fibre Channel connectivity on the array enclosure. Of course, the I/O controller needs to translate across command, protocol, and physical domains for this scenario, so matching up the HDD I/O type and the enclosure I/O type is a simplifying advantage. Figure 3B.6 shows comparisons for the three serial drive I/O specs.

### 3B.1.2.1 ATA Versus SCSI Drives

In general, ATA (started life as IDE) connectivity is the alternative to SCSI-based connectivity. ATA drives have been used in PCs for many years. Because they offer lower performance than SCSI drives, ATA rules at the low end and SCSI at the high end. The commands, protocols, and physical connectors are completely different between SCSI-P and ATA-P. SATA is the serial equivalent of ATA-P just as SAS is the serial equivalent to SCSI-P. Fortunately, due to excellent cooperation among industry groups, there is now only one backplane connector type for both SATA and SAS drives. This allows storage array manufacturers to build one array enclosure, and it can be populated with either ATA or SCSI drives or a mix (if supported). Finally, SCSI and ATA are converging along parallel lines.



**FIGURE 3B.7** SCSI (SAS/FC) versus ATA (SATA) drives.

Some of the salient differences between SCSI and ATA drive types are seen in Figure 3B.7. Note that SCSI drives may support Fibre Channel or SAS I/O whereas ATA drives only support SATA I/O.

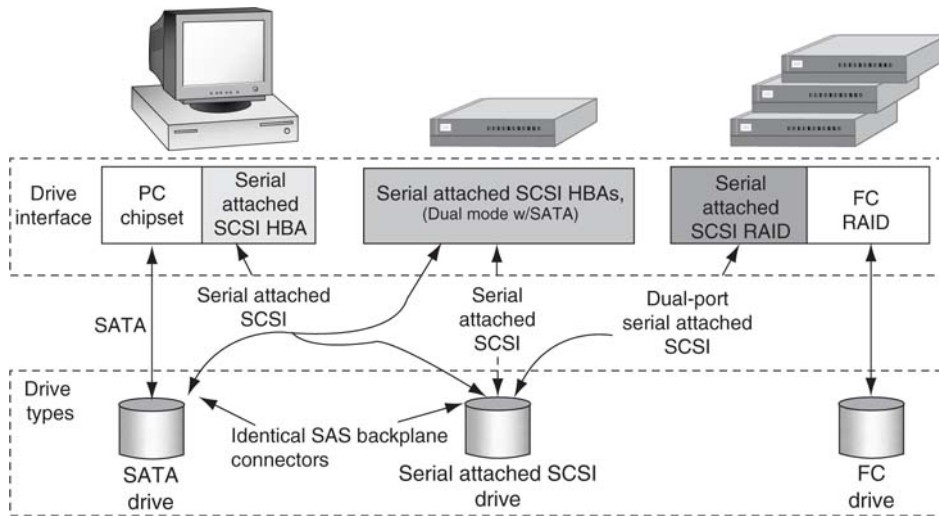
At first glance the SCSI drive appears to lead in performance and reliability but at the penalty of much higher prices. In practice, the reliability of an ATA drive is slightly less than for a SCSI drive. See Chapter 5 for some enlightening analysis on HDD reliability. Second, the performance is indeed better with SCSI for small block R/W transactions. However, as discussed in Chapter 3A, for large block R/W transactions, the rotational latency has marginal impact on R/W access rates. For A/V applications the ATA drive provides a big advantage due to its superior pricing and larger capacity. SCSI drives are marketed as the high-end choice, so drive vendors receive more profit margin than on lower-end ATA drives.

In general, IT server farms tend to use SCSI drives, while desktop and low-end servers tend to use ATA drives.

Figure 3B.8 illustrates drive connectivity for low-, mid-, and high-end systems for general use. For the low end, SATA drives are employed; for the mid range, SATA or SAS; and for the high end, SAS or FC SAN connectivity.

### 3B.1.3 ATA and SCSI I/O Convergence

As mentioned, the SCSI and ATA special-interest groups have collaborated and agreed on a common backplane connector and protocol for the serial versions of both ATA and SCSI. This was a formidable task because there is little in common between traditional SCSI and ATA parallel standards. The drive types will remain separate, but the serial I/O for each drive will converge. SAS standardizes



**FIGURE 3B.8** SATA, SAS, and FC for low-, mid-, and high-end systems.  
Source: Adaptec.

a combination of three protocols, each of which transports different information types over the serial interface:

- Serial SCSI protocol
- Serial ATA tunneling protocol (STP) passes through ATA commands
- SCSI management protocol (SMP) provides HDD management information

The SAS connector is form factor compatible with the SATA connector. SATA signals are a subset of SAS signals, enabling the compatibility of SATA devices and SAS controllers. SCSI/SAS drives will not operate on a SATA controller and are keyed to prevent any chance of plugging them in incorrectly.

In addition, the similar SAS and SATA physical interfaces enable a new universal SAS backplane that provides connectivity to both SAS drives and SATA drives, eliminating the need for separate SCSI and ATA drive backplanes. This consolidation of designs greatly benefits both backplane manufacturers and end users by reducing inventory and design costs.

Some of the benefits of this converged approach are as follows:

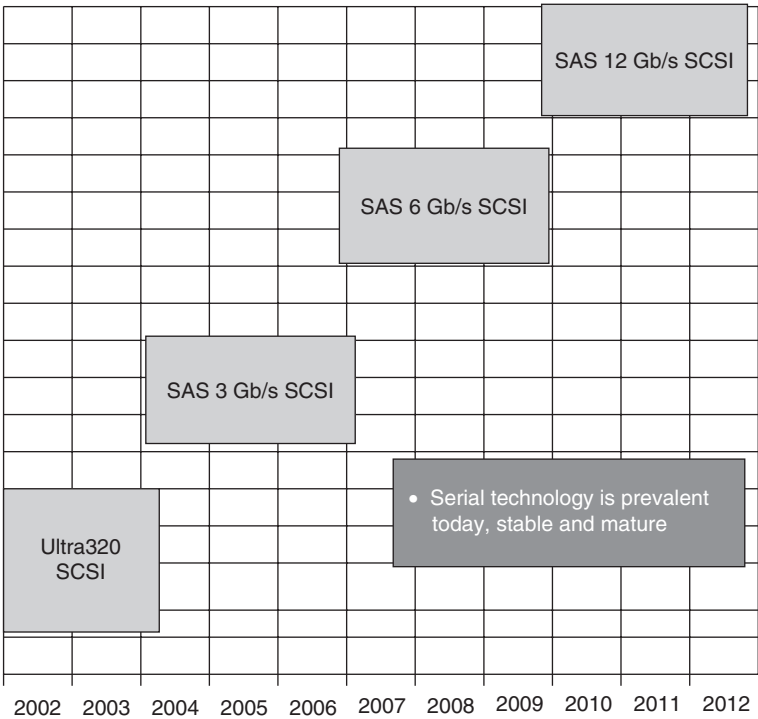
- Only one HDD I/O connector type is needed for both serial SCSI and serial ATA drives.
- A storage enclosure may support both serial ATA and serial SCSI drives.
- Host software drivers can include bundled support for SCSI and ATA if desired.
- Management commands and responses are common.

A SATA drive can plug into a SAS backplane connector, or a serial SCSI drive can plug into the same connector (Figure 3B.8). How does a host computer know what type of drive is installed and which protocol to use for communications? The SAS standard defines a method by which a host can identify the drive type and then correspond only with that drive type. Incidentally, the physical layer electrical signals (voltages, clocking) are different between SAS drives and SATA drives. Although the SAS spec defines maximum clocking speeds, the values are different for each drive type. The drives will evolve differently, but the interface spec will remain a common feature of each drive type.

SAS connectivity has a bright future, and therefore so do SAS and SATA drives. Figure 3B.9 shows a roadmap for the SAS standard with link speeds of 12 Gbps in ~2011. Small enclosures with gigabits per second of data flow and terabytes of capacity are just around the corner.

3B.1.4 Remote DMA: The Next Frontier

Designers are always looking for faster methods to move data across networks. In this light, emergence of the remote DMA technique is interesting. A direct memory access (DMA) operation is commonly used within a PC or server for



**FIGURE 3B.9** Serial technology has a bright future.  
Source: Adaptec.

internal memory-to-memory data transfers without the involvement of the CPU. It is always beneficial when the CPU is removed from the data transfer path. A DMA operation provides very low latency and super high transfer rates. The “memory” in a DMA operation is typically DRAM with HDD on the horizon. RDMA extends the idea to memory-to-memory data transfers across a network. Several techniques have been developed to this end. InfiniBand is the leading technology to implement RDMA.

InfiniBand is an ultra-low-latency, non-IP, communication, storage, and embedded interconnect. InfiniBand, based on an industry standard, provides a robust data center interconnect. With 30 Gbps and 60Gbps link products currently shipping, InfiniBand is at least a generation ahead of competing fabric technologies today. It was developed to cluster servers in data centers. It is considered exotic technology and is sometimes found at the high end of computing configurations. One of the leaders in InfiniBand-based products is Mellanox ([www.mellanox.com](http://www.mellanox.com)), which offers a single unit switch with 24 ports and a throughput of 60 Gbps per port. InfiniBand links use 8b/10b encoding, so the payload rates are 80 percent of the line rates. See Appendix E.

A competing RDMA method was defined by the RDMA Consortium ([www.rdmaconsortium.org](http://www.rdmaconsortium.org)) and uses TCP/IP (not InfiniBand) as the transport means over a variety of physical networks. Ten Gbps Ethernet will likely be the most common physical layer. Why invent yet another RDMA method when InfiniBand seems to do the job? Because InfiniBand is not IP networkable, it will never find the wide acceptance of Ethernet/IP. Although it is new, some storage vendors offer products with RDMA/Ethernet support. The RDMA/IP method bypasses the traditional CPU and TCP/IP software stack (TCP/IP is processed in the HBA card) and utilizes zero copies of data packets during the transfer of data. It is the ultimate in efficiency for moving data between two memory locations over IP. It remains to be seen how, if at all, RDMA technology will be used by A/V vendors. The first application may be film-related projects with uncompressed rates  $\sim 7$  Gbps.

## **3B.2 STORAGE AREA NETWORKS**

SAN is a mature technology and accepted way for multiple clients and/or servers to access storage pools over a network. SAN architectures are often chosen for applications that need highly scalable performance from storage. Fibre Channel SANs are installed today in a majority of companies showing \$1 billion or more in annual revenues (Chudnow 2002). According to Art Edmonds, previous Chair of the Fibre Channel Industry Association (FCIA),<sup>2</sup> “Fibre Channel technology continues to be the technology at the heart of SANs

---

<sup>2</sup> See [www.fcia.org](http://www.fcia.org) for more information on FC and associated technology.

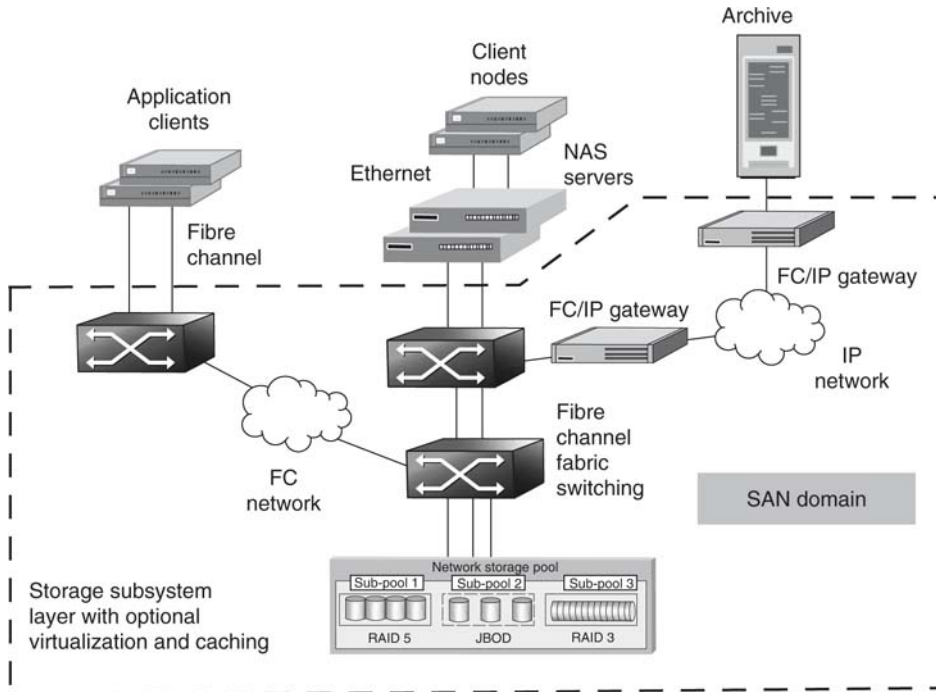
installed around the world. Conservative estimates place the number of SAN installations at 100,000 worldwide.” SANs are universally accepted as the preferred way to access storage in big deployments. This section studies SAN methods with emphasis on components, benefits, architectural requirements, protocols, and futures.

SAN replaces the link-based architecture of DAS with a switched fabric. Administering DAS islands is more expensive and complex than managing a SAN. Managing, say, 10 independent islands of servers connected to individual DAS storage is more costly than managing one shared pool. In effect, it is more efficient to allow  $N$  straws to draw from one huge pool (SAN) than to have  $N$  pools each with only one straw (DAS). See Figure 3A.3 in Chapter 3A for an illustration of this concept. According to one famous study by the Gartner Group, the cost to acquire a storage system is only 20 percent of the total cost of ownership (TCO) over its useful life. Nemertes Research (2007) found that over the life of a storage system 60 percent of the total cost of ownership (TCO) is management related.

As a result, there is a big motivation to concentrate a storage pool of homogeneous or heterogeneous devices. A SAN accomplishes this and hence the deployments of it in IT worldwide. However, the storage concentration also requires some way to manage and assign portions of the storage to the attached devices. One method is to use a clustered file system (CFS), and another method is to use virtualization. It is good to note that a SAN provides blocks of storage to attached clients, whereas a NAS provides direct file access. See Chapter 3A for a refresher on these concepts.

Let us use Figure 3B.10 as the basis of the SAN study. On the far left top are Fibre Channel (FC)-attached A/V clients (Ingest, Playout, NLE, processors, and other nodes) that connect through FC switch fabric to access storage. In the middle are NAS-attached servers. The servers access storage through the SAN switching fabric. On the far right is a remote archive that accesses the main SAN through an IP network. In this case the archive FC traffic is tunneled over TCP/IP and is converted back to FC to enter the switch fabric. This particular view of a SAN is Fibre Channel centric, but FC could be replaced with Ethernet and iSCSI, as discussed later. All the components inside the dotted box constitute the SAN. Note that virtualization and caching are optionally included in the network storage pool at the bottom of the diagram. There are, of course, many different ways to build a SAN, but the common defining high-level traits are as follows:

- Servers and other nodes access storage via Fibre Channel or Ethernet. Every attached node sees the storage as though it is local.
- The switching fabric allows any attached initiator to access homogeneous or heterogeneous block storage—not files. However, virtualization or CFS policies can partition the storage.



**FIGURE 3B.10** General view of FC-based SAN architecture.

- Management methods supervise one unified pool of storage rather than a disparate collection.

These SAN personality traits are not sufficiently specific to define the design. Other criteria are needed. The design should match the functional requirements. So what are these functions? Let us see.

### 3B.2.1 Form Follows Function

The famous architect Louis Sullivan is considered the inventor of the modern skyscraper. The Chicago buildings produced by Sullivan (and Dankmar Adler, his structural engineer) were at the leading edge of skyscraper design and were known for their gorgeous and tasteful architecture. One of his dictums was *form follows function*. When you are designing a storage system, be it SAN or NAS, the form of the architecture should follow its intended function. In a simple-minded view, the function of a SAN is to access data. However, when we look deeper, there are at least eight guiding principles (Glass) for defining the function of storage:

1. **Availability and performance**—Information access needs.
2. **Scalability**—Expected growth in capacity, access nodes, and transfer rates.
3. **Reliability**—May span from basic to mirrored with ~100 percent up time.



4. **Utilization**—Percentage of usage per population of attached nodes.
5. **Security**—Access rights and intrusion prevention.
6. **Connectivity**—Accessibility of the storage system.
7. **Backup and archive**—Various strategies to meet business needs.
8. **Cost (TCO)**—Does it meet budgetary needs over time?

The form or architecture of a SAN (or NAS) should be a strong factor of these eight functional requirements. Of these eight, performance, scalability, reliability, utilization, and connectivity are tied to the physical nature of a SAN. It is the protocols and configurations that define how to meet these needs. The next section considers these aspects of a SAN.

### 3B.2.2 The Fibre Channel-Based SAN

The three most important elements in legacy SANs are the Fibre Channel link, FC switches, and storage. Most installed SANs are FC based; however, Ethernet/IP is used in newer SANs. For now, let us concentrate on Fibre Channel; IP SANs are considered in the next section. What is Fibre Channel, what are its sweet spots, what are the defining configurations, and what is its future?

Fibre Channel is a serial, gigabit link technology designed to connect nodes to storage or to transfer data between nodes. Its speed ranges are 1, 2, 4, and 8 Gbps. Fibre Channel continues to provide the highest performance available for storage interconnects.

Fibre Channel is not a network link in the Ethernet sense, so its use is confined to local configurations. It may be configured in three ways: point to point, routed via a switch (or a mesh of switches), or in a serial arbitrated loop. Despite its name, it comes in both copper and optical flavors. The copper version is limited to about 20m (2G FC) but can reach 80km using single mode optical fiber.<sup>3</sup> Additional features of importance are as follows:

- **Guaranteed data delivery using hardware-related protocol handshakes.** This is a key feature of FC, as data reliability is assured by hardware on the host bus adaptor (HBA) card and not by a software stack running on the host CPU. Offloading data delivery to hardware operations rather than software can increase the throughput by an order of magnitude; plus, it frees the main CPU for other more important tasks.
- **TCP/IP has traditionally been very host CPU intensive, so FC has gained the edge.** This is changing, as will be seen when iSCSI is discussed.

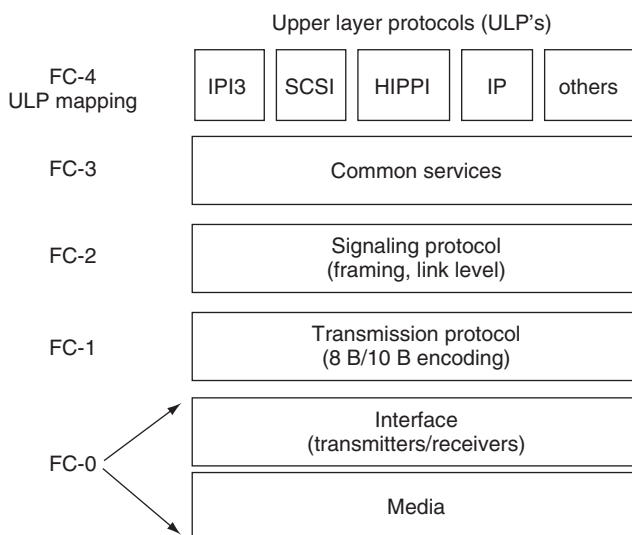
---

<sup>3</sup> See the Storage Networking Industry Association Web site at [www.snia.org](http://www.snia.org) and [www.snia-europe.org](http://www.snia-europe.org) for a wealth of information on SAN and NAS; look for *SNIA IP Storage Forum White Paper Internet Fibre Channel Protocol (iFCP): A Technical Overview*. See also the Fibre Channel Industry Association Web site at [www.fcia.org](http://www.fcia.org) for more details on FC.

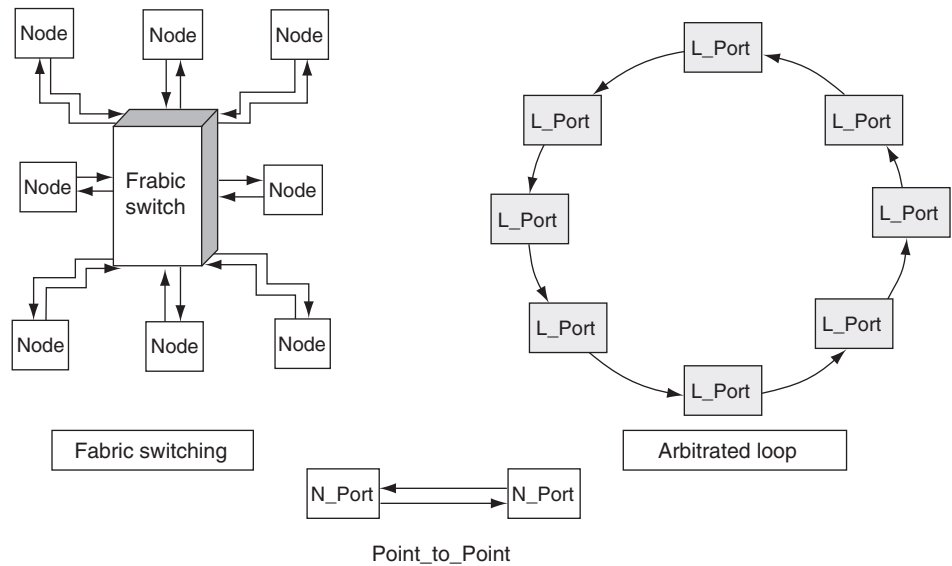
- Very efficient data packaging with 2 K frame sizes.
- Connection-oriented virtual circuits for excellent QoS.

The Fibre Channel protocol stack is shown in Figure 3B.11. The Fibre Channel standard defines a multilayered architecture for moving data. Five hierarchical layers are defined, named FC-0 to FC-4, with FC-4 being the highest layer. The layers are defined as follows:

- **FC-0**—The interface to physical media.
- **FC-1**—The encoding and decoding of data (8b/10b) and out-of-band physical link control information for transmission over the physical media. See Appendix E.
- **FC-2**—The transfer of frames, sequences, and exchanges comprising protocol information units.
- **FC-3**—Common services required for advanced features, such as striping, hunt group, and multicast.
- **FC-4**—This layer describes the interface between Fibre Channel and various upper-level protocols. A number of different protocols are supported as ULPs, including SCSI, IP, and high-performance protocols such as HIPPI and ATM. These protocols are encapsulated and carried over FC. SCSI command mapping is the most widespread, with the others used sparingly or not at all. SCSI mapping over FC is called FCP.



**FIGURE 3B.11** Fibre Channel protocol stack.



**FIGURE 3B.12** Fibre Channel connectivity choices.

Fibre Channel may be configured in three different ways, as Figure 3B.12 shows. Point to point is the most straightforward. Many FC systems used the arbitrated loop (FC-AL) method to move data. FC-AL is a topology in which nodes are connected together in series and share the aggregate bandwidth of a single FC link. Of course, the serial configuration is not ideal due to the obvious problems with a daisy chain; however, it is efficient for connecting together a few FC disc drives on the same link. Hubs eliminate some of the daisy chain problems, but switches are the preferred way to route FC frames.

From personal experience, hub-based FC-AL is problematic for mission-critical real-time A/V applications. Processing errors, disc failures, intermittent links, or simply rebooting a device can all incur a “LIP storm” (loop initialization primitive). LIP storms can cause multiple devices to send non-stop streams of initialization commands, requiring a great deal of hands-on attention by IT personnel until the problem is located and the loop stabilized.

With FC-AL, the available Fibre Channel bandwidth of 800Mbps (IG FC) is shared among all loop members. If four nodes are communicating with four separate storage devices on the loop, each pair would be able to sustain approximately 800 Mbps. Because of this sharing, devices must arbitrate for access with the loop before sending data.

The alternative to the loop is switching fabric. A fabric requires one or more switches to interconnect host computers with storage devices. With a fabric, the bandwidth is not shared. Each connection between two ports on the switch has a dedicated 800 Mbps, so a 16-port FC switch can support 8 paths (8 in,

8 out) and each path can support 800Mbps of payload (1G FC example). The switch's internal engine needs to handle at least  $8 \times 800\text{Mbps} = 6.4\text{Gbps}$  in one direction and 6.4Gbps in the other (FC is a full duplex). Also, switches are very resilient to problem nodes. Bad citizen nodes (intermittent, LIP storms, etc.) do not disturb the traffic on the other ports. This is another reason why a switch is the preferred interconnect (compared to FC-AL) when building a real-time A/V-centric FC network.

At the high end is the so-called Fibre Channel director. This is a glorified switch by another name. There is no industrywide definition of a director, but the common traits as seen from a survey of several FC directors by different vendors are

- Hot upgradeable firmware
- Dual elements: power supplies and control modules
- Hot pluggable elements (backplane is an exception): boards, power supplies, control modules, and GBIC I/O ports
- At least 64 ports

A FC director switch fits an enterprise that has the following needs:

- Very high SAN reliability
- Very small downtime for upgrades
- A large port count
- Limited personnel for managing many small switches

As you evaluate FC switches versus directors, the differences will center on brute-force switching versus sophistication. The legacy SAN is built out of FC-based elements. Does IP have a place in SAN architecture? This issue is considered in the following section.

### 3B.2.3 Hybrid SANs: Merging FC and IP

Within the last few years, there has been a lot of work to move SANs to TCP/IP-based networking. Why? FC-based SANs require specialized hardware, and FC is an expertise not common in many IT departments. Without a doubt, Ethernet/IP has won the networking wars. Managing Ethernet/IP is common knowledge and already a part of the day-to-day operations in most companies. To many IT managers, the move away from FC to all Ethernet would be welcome if the price/performance goals could be met, but Ethernet/IP is far from a shoe-in. The cost of deploying Fibre Channel, particularly the cost of the switches, has come down dramatically. Top-notch performance is still easier to achieve with FC. There is a huge engagement of deployed FC-based SANs, and these will not be converted to IP SANs overnight. Also, any scaling up of existing SANs will demand FC-compatible technology, not forklift upgrades. That being said, there is a strong case for IP SANs, as this section will show. Table 3B.2 outlines (SNIA) some of the benefits when using IP to extend a SAN.

**Table 3B.2** Comparing IP Versus Fibre Channel

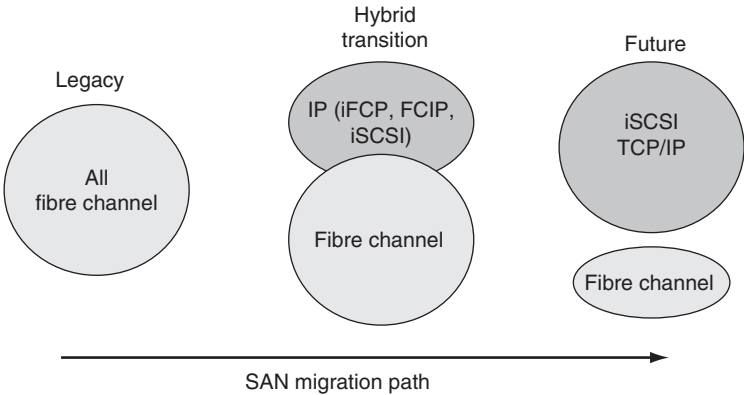
IP Networks	FC Networks
Global scale	Local data centers
Many switching nodes (1,000s)	Fewer switching nodes
Heterogeneous link support	Homogeneous links/switches
QoS a strong function of TCP and network reach	Excellent QoS, high data rates
Built-in recovery means routing around bad links or switches	Automatic rerouting not easy
Well understood by the IT staff	Less familiar to some IT staff

3B.2.4 IP SAN Technology Choices

Three technologies have been invented to meet the needs of the IP SAN coexisting with the FC SAN. They are iSCSI, iFCP, and FCIP. Each has a definite purpose and application area. In practice, there are three types of SANs: all FC (today's legacy), hybrid FC and TCP/IP, and all TCP/IP. iFCP and FCIP are designed for hybrid FC/IP SANs, and iSCSI is designed to be a full FC replacement technology. See Figure 3B.13 for a view of the migration from legacy FC to IP SANs. Fibre Channel will survive the onslaught of IP, but its market position will diminish for sure. Time will tell if IP completely swamps FC or FC holds its ground.

These three new protocols are defined as follows:

- **iSCSI**—A TCP/IP-based standard for accessing data storage devices over an IP network using SCSI as the storage access protocol. In relation to a traditional SAN, TCP (usually over Ethernet) carries the SCSI commands rather than Fibre Channel. There are no FC-related protocol layers.



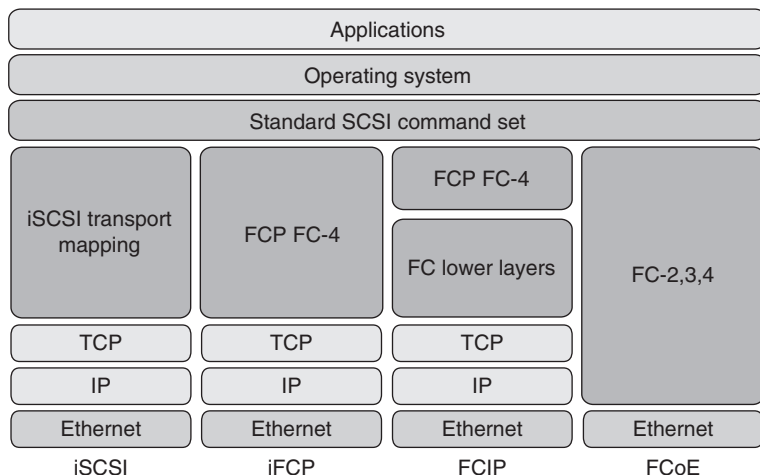
**FIGURE 3B.13** The migration of SAN to all IP.

iSCSI is a native way to map SCSI over IP. An iSCSI storage array should behave similarly to a FC array in terms of LUN masking and so on.

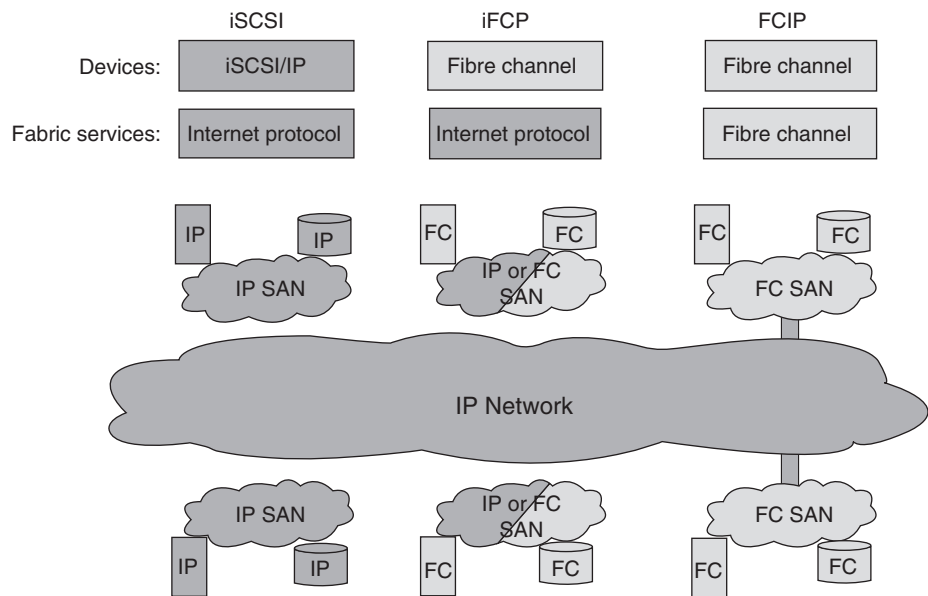
- **iFCP**—The iFCP specification (Internet FCP) is a gateway-to-gateway protocol for the implementation of the FCP layer (FC-4 SCSI layer) using TCP/IP switching and routing elements instead of Fibre Channel components. The transport, link, and physical layers (FC-0 to FC-3) in FC are replaced with a TCP/IP network and underlying physical link (usually Ethernet). Using TCP guarantees reliability over IP networks, and it replaces the reliability functionality inherent in FC's lower layers. iFCP's primary advantage is as a SAN gateway protocol that bridges both FC and IP domains yet has the SCSI FC-4 layer in common.
- **FCIP**—This protocol (FC over IP) standardizes a complete wrapping of all but the physical layers into TCP/IP packets. The IETF has defined similar encapsulations of FC over ATM and SONET. The primary function of FCIP is to forward FC frames. Think of FCIP as a FC extender protocol using IP. Two geographically separated FC SANs may be linked over an IP link using FCIP. This protocol is best used in point-to-point connections between SANs with no FC address routing.

These three similar yet different protocols are often a source of confusion. They all allow a sender to communicate with a receiver over IP using SCSI as the command language. Figure 3B.14 shows the three protocol stacks (left-most). TCP/IP maps SCSI commands in all three cases. For another view of the three protocols, consider Figure 3B.15. Some of the key features are

- iSCSI can be used to create a complete TCP/IP SAN with no FC needed. Given the presence of iSCSI ported storage arrays, a SAN may



**FIGURE 3B.14** Common SAN protocol stacks.



**FIGURE 3B.15** iSCSI, iFCP, and FCIP networking configurations.

be built without any FC technology. The possibility of a global SAN is enabled.

- iFCP is used to bridge existing FC systems across TCP/IP networks. The IP packets may be routed and managed using FC target addresses. This is a key feature, and iSNS is used to associate FC addresses with IP addresses.
- FCIP is used to extend a FC link over an Ethernet/IP link with no routing based on FC addressing. This will find limited use.

## iSNS



The Internet Storage Naming Service (iSNS) protocol is designed to facilitate the automated discovery, management, and configuration of iSCSI and Fibre Channel (iFCP) devices on a TCP/IP network. iSNS provides intel-

ligent storage discovery and management services comparable to those found in pure Fibre Channel networks, allowing a commodity IP network to function in a similar capacity as a SAN.

### 3B.2.4.1 Fibre Channel over Ethernet (FCoE)

Fibre Channel over Ethernet is the latest entry into the FC migration toolbox. The mapping of Fibre Channel frames over full-duplex IEEE 802.3 networks

leverages 10/40 gigabit Ethernet networks while preserving the Fibre Channel protocol, FC-4 the top level. (See Figure 3B.14, rightmost stack.) Importantly, most of the FC stack is eliminated, as is the TCP/IP layer. What are the pros and cons of this configuration?

- FCoE enables equipment room and building layer 2, Ethernet switching to carry FC traffic. Ethernet switches are less costly, are easier to manage, and offer higher data rates compared to FC switches. Properly configured switches will not drop data frames. So, a loss-free network is possible, much as with FC switching.
- Ethernet has a *pause* command that can be used to stop upstream data momentarily, thus preventing buffer overflows and loss of received data.
- FCoE consolidates all facility networking with Ethernet; there are fewer cables in total, fewer I/O cards, less power.
- Vendors offer switches that support both Ethernet/FCoE and FC to bridge the two domains.
- FCoE is not IP routable, as is iSCSI. However, for most SAN applications, Ethernet's reach is sufficient and more scalable than FC networks. For more information, see [www.T11.org/fcoe](http://www.T11.org/fcoe).

A related technology, ATA over Ethernet (AoE) has some limited success in small, switched LANs. It enables the possibility of building low-cost SANs. As with FCoE, the SAN is switched at layer 2. It is not a standardized protocol but is supported under Linux.

### 3B.2.5 TCP/IP SAN Performance

How will these protocols be used over the next few years? For upgrades to existing FC SANs, iFCP, FCIP, and FCoE will be used. Performance should be excellent and in line with pure FC configurations if the traffic engineering is modeled correctly. For new SAN installs, designers will choose between FC-based or IP (iSCSI)-based configurations. Given the super performance of FC solutions, will IP/Ethernet be able to equal it? Here are some of the considerations that limit iSCSI SAN performance.

1. TCP guarantees data delivery. In the general case, the host (and target) TCP stack is implemented in the CPU, which uses valuable CPU cycles and may limit the throughput under high data rates. Congestion in the IP network can seriously throttle throughput.
2. Long pipes between host and target can throttle throughput if the round trip time (RTT) exceeds some critical values.
3. Large data rates can "choke" the pipe and cause the data delivery to back off.




As it turns out, because all four of these protocols can be optimized for top performance, it is possible to achieve FC-like speeds and reliability in an IP environment. Chapter 6 covers each of these four points in detail in Section 6.3, “TCP/IP Performance.” A discussion of iSCSI accelerators using TCP off-load Engines (TOEs) is also covered there.

Many new SAN installs will use Ethernet/TCP/IP or FCoE instead of Fibre Channel. It will take years for iSCSI to replace FCP, but the roadmaps to IP are the clear future for mainline applications. In the meantime, hybrid solutions of IP and FC will coexist. Fibre Channel is not dead by any means, but it is stepping aside as the grand old man of SAN in favor of the new kid on the block, iSCSI.

**iSCSI TERMINOLOGY**

A host needs iSCSI *initiator* software to start an iSCSI transaction. Each transaction terminates at some iSCSI target in a storage system. The iSCSI initiator is analogous to a NFS client or a FCP initiator in a FC SAN. iSCSI software initiators/targets are available from vendors for

Windows or Linux and other operating systems. iSCSI storage devices are sold standalone, but it is also possible to roll your own by installing the iSCSI target onto a server unit with attached storage.



Snapshot

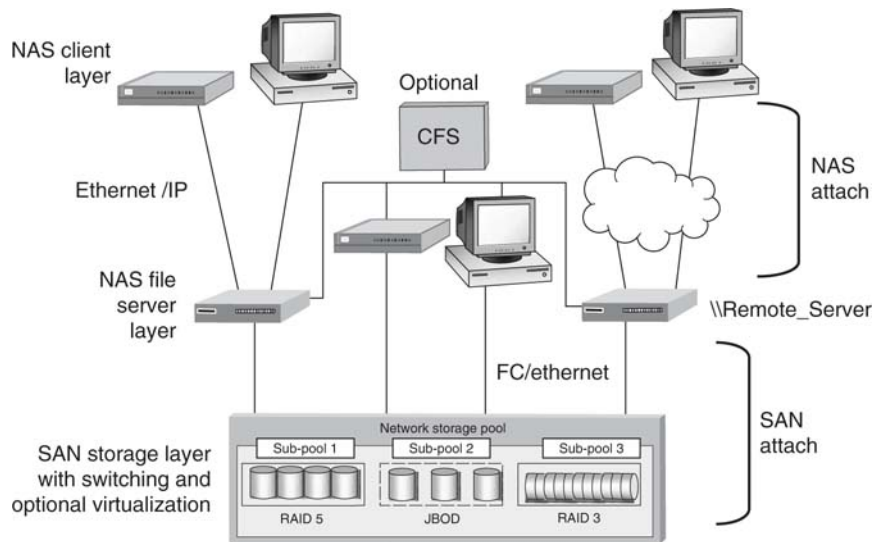
**3B.2.6    SAN with Virtualization and Cluster File Systems**

So far, the discussion on SAN technology has focused on connectivity and networking. Two other important aspects are virtualization and the cluster file system. These are discussed in detail in Chapter 3A. Virtualization is a method to carve up the storage into separate volumes assigned to clients and servers as part of the storage management process. However, the CFS gives SAN clients and servers access to a shared file system. Both may be applied to a SAN environment.

Figure 3B.16 illustrates how SAN clients and servers can share files (not just storage hardware) by including the CFS. It is good to remember that a SAN does not require a CFS—it is always optional. A SAN provides block-based storage in its most native form. However, if an A/V workflow demands file sharing, as in a news production or a file playout scenario, then a CFS is often included. Many A/V vendors provide a SAN storage with a CFS for editing clusters, A/V recording, and playout. In most cases, the CFS should be fault tolerant as well as the storage subsystem. Next, let us review the SAN vendor landscape and then focus on the third item in our trilogy: NAS.

**3B.2.7    SAN Vendor Overview**

There are many vendors of SAN storage systems from low to high end. Traditional SAN systems are composed of Fibre Channel switches, directors,



**FIGURE 3B.16** *SAN and NAS connectivity.*

RAID storage, storage management software, and optional clustered file systems. Some vendors provide for a complete turnkey solution, whereas others provide select pieces of the total puzzle. Some of the top SAN players are Brocade, Cisco, EMC, Hitachi Data Systems, HP, IBM, Sun, and Xyratex. There are countless others too, but studying these vendors' products will give you a good flavor of the landscape.

Of special note are the several A/V-specific vendors that offer SAN storage in support of A/V editors and other media functions. Among these are Archion, Avid, EditShare, and Facilis. There are also several new players entering the iSCSI SAN race. These will compete with the incumbents for this new storage space.

### 3B.3 NETWORK ATTACHED STORAGE

Network attached storage appears as a remote file server (X: drive or \\Remote\_Server) to any attached clients. NAS is a category of storage device that appears as a node on the IP network. A NAS file server does not provide any of the functions that an application server typically provides, such as email or Web page serving. However, the NAS server always provides a file system view of its storage with optional services such as file conversion, bandwidth management, automatic backup, synchronized mirroring (for transparent redundancy), and caching. A typical NAS environment is shown in Figure 3B.16.

For many server installations, the NAS server uses SAN-attached storage. In the simplest case, there is only one NAS server with DAS storage and no SAN. At the other end of the spectrum, there may be many NAS servers sharing storage

or using separate storage. The servers may also use clustering technology to create a huge virtual server. As part of a SAN, the servers may be configured with virtualization or a clustered file system, depending on functional needs.

3B.3.1 NAS Attach Protocols

Figure 3B.17 outlines the I/O and basic structure of a NAS server. The server may have SAN, DAS, or internal storage and offer one or more NAS attach protocols, such as NFS, CIFS/SMB, HTTP, or Apple Filing Protocol (AFP). In a nutshell, a NAS server is a remote file server with all the features that a file system offers. Additionally, some vendors will offer a general-purpose product that includes SAN (FC and/or iSCSI) or even RDMA attach points to create a one-size-fits-all product. Concentrating on the most general-purpose case, a NAS server is the storage and file system for a remote file server transaction. Clients communicate with the server using Ethernet and TCP/IP with an associated file access protocol. In general, the protocols enable a client to establish a session with a remote device, open/close files, R/W files, create and delete directories and files, search for files, and more. There are many more similarities than differences among the various NAS attach protocols.

One of the most common access protocols is the network file system (NFS), which was invented by Sun Microsystems in the 1980s. It was originally designed for file sharing between UNIX systems. Today, NFS is used primarily with machines running some variant of the UNIX OS and especially Linux. Common Internet file system (CIFS) is the nearest neighbor to NFS. It was invented by Microsoft and IBM in 1985 and started life as the server message

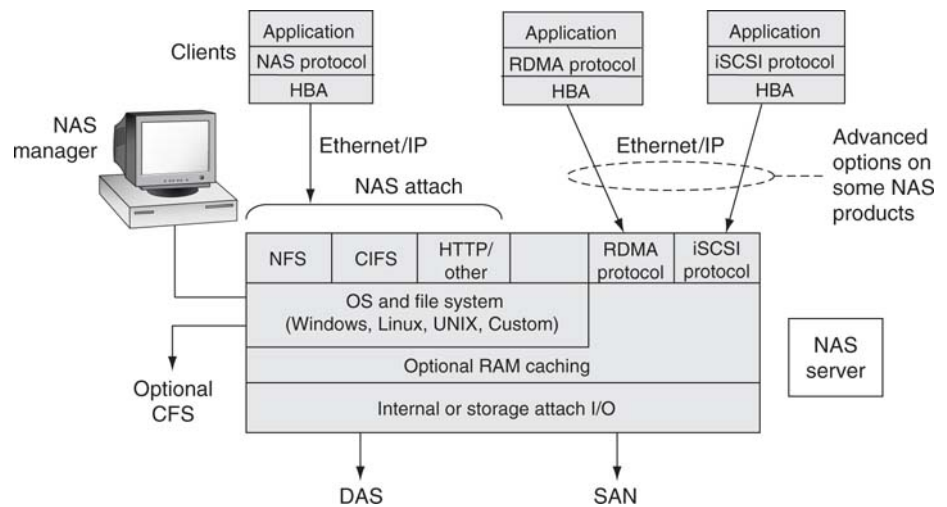


FIGURE 3B.17 Inside a NAS device.

block (SMB). Starting as a DOS networked file exchange, it has become a world-wide standard for client/server data sharing. SAMBA is the moniker for an open source implementation of the CIFS/SMB protocol and a Brazilian dance introduced in 1917. The non-dance version permits Linux/UNIX systems to masquerade as a Window server, thereby enabling cross-platform file exchange. Most NAS servers also support the HyperText Transfer Protocol (HTTP). This is used for Web-based administrative needs and file download to Web browsers. Finally, some NAS servers support Apple Filing Protocol, which is Apple's version of a file-sharing protocol. These protocols tend to coexist peacefully, and each has found its niche in the NAS world.

### CIFS, NFS, AND HTTP: DUCT TAPE FOR NETWORKS



Whenever a client connects to a remote server for storage access or Web content, one of three protocols is commonly used—CIFS, NFS, or HTTP. Without universally accepted access methods, a network (the Internet in particular) would be like a telephone system where every phone used its own brand of touchtone dialing—no one could connect to anyone. NFS and CIFS are more general purpose than HTTP, which was designed to retrieve Web pages. The minimal requirements for CIFS/NFS connectivity are

- Mount remote storage as an X: drive (or \\Remote\_Server) or equivalent at local client

- Open, close, R/W, delete, create directories, and so on for a remote storage file
- Provide file access restriction rights per client/user
- Support TCP/IP (some use UDP) for network routing and reliably transport data

Many legacy NFS connections use version 2 or 3. Version 4 was approved as an IETF standard (RFC 3010) and includes improved access security and byte-level locking (needed when multiple clients are writing to the same file). Finally, V4.1 is called Parallel NFS or pNFS. This latest version enables a CFS across several NAS servers.

### 3B.3.2 NAS Vendors and Product Features

NAS servers fall into several categories: small appliances, mid-size, and enterprise-size systems. The following is a representational list of vendors in this space. It is by no means exhaustive.

- **Low end:** Dell's PowerVault, Iomega's StoreCenter, Snap Appliance's Snap Server, Buffalo's TeraStation
- **High end:** BlueArc's Titan Silicon Server, EMC's Celerra family, IBRIX Fusion, Isilon's IQ X Series, Network Appliance's FAS series

There are countless others, but studying these vendors' products will give you a good flavor of the landscape. Each of the vendors has a value proposition that attempts to differentiate it from the others. The appliances and mid-size

servers are often a single server with embedded storage. The high-end enterprise units are built with scalability and load balancing and may include multiple servers clustered to appear as one. One way to cluster a NAS is with a CFS/DFS, as described in Chapter 3A.

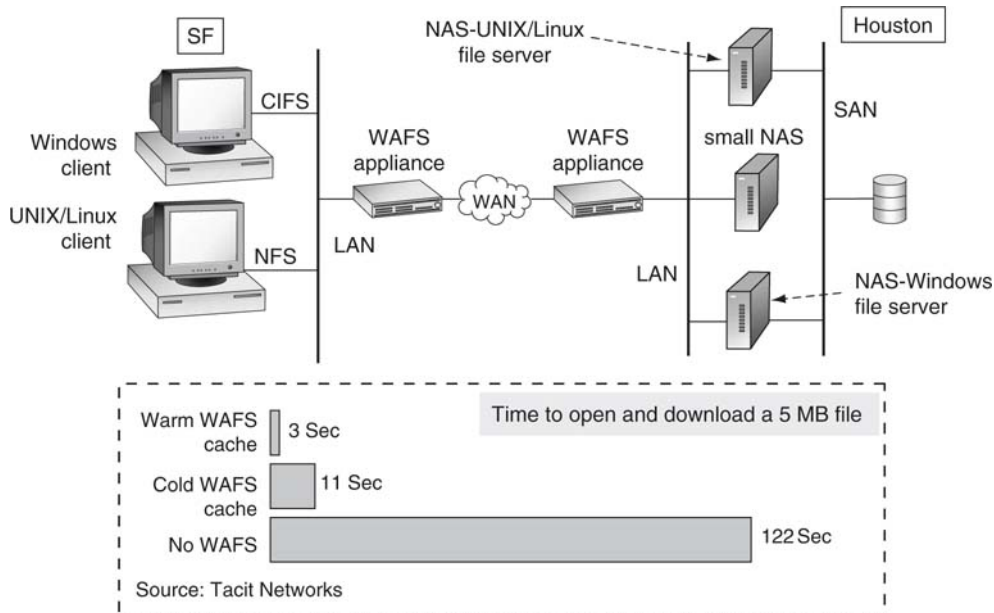
Note that most of these platforms are not fine-tuned to function as real-time HD storage for ingest, editing, and playout. Specialized platforms from Apple, Avid, Harris, Omneon, GVG, and others are often better suited for continuous, guaranteed real-time A/V operations. True, some NAS servers are used in the real-time space, but buyer beware and always understand the RT issues before purchase. There is no doubt that NAS is mature and well accepted in the modern enterprise and A/V facility.

The operating system of a NAS server may be a key selling point. Microsoft's storage server is a big draw because of its familiarity and features. It is an optimized OS for NAS servers. Some NAS vendors, such as Network Appliance (WAFL) and Isilon (OneFS), have designed proprietary file systems to add features and performance over non-specialized operating systems. In addition, the Linux OS has a major market share in this space. Remember, a NAS server is dedicated to behaving like a storage device, whereas a general-purpose server may run application programs of all types. However, most COTS NAS servers also offer auto-backup, restore, fault tolerance methods (RAID and more), specialized management, and more. To improve access performance as measured by latency and data throughput, some high-end systems support heavy caching. For some applications, a cache can improve the average storage access performance markedly. Even apparently random I/O performance can be improved using clever prediction with caching. See Section 3B.4, "Caching," for more details.

### **3B.3.2.1 NAS Acceleration over WANs**

At times, storage consolidation among disparate facilities or remote storage access is required to improve A/V workflows and/or a system's management efficiency. Under these scenarios, WAN data throughput can become the weak link in the operational chain. CIFS and NFS were designed for LAN networks and are chatty protocols with resulting poor performance over WANs. Wide area file services (WAFS) appliances are a new category of product to remedy this problem. These devices are protocol accelerators and use prediction and caching to dramatically improve CIFS and NFS performance. For example, the local appliance will return acknowledgments for routine session creation instead of waiting for a delayed remote machine response.

The performance improvements can be striking. Figure 3B.18 shows an example of a test case using a T1 line at ~1.54 Mbps from San Francisco to Houston with a roundtrip delay of 60Ms. Opening and downloading a 5 MB file takes only 11 s using WAFS, whereas the same file takes 122 s without the appliance. In fact, the data transfer takes only 3 s if the file is in the appliance cache.



**FIGURE 3B.18** WAFS performance acceleration example.

For more insight, study the product offerings and white papers from Cisco, Juniper Networks, or Packeteer. These are but a few of several WAFS product vendors. See, too, Cisco's wide area application services (WAAS) appliance that is a superset of a WAFS for general WAN acceleration.

### 3B.3.3 A/V-Friendly NAS Connectivity

The goal of NAS protocols is to provide for easy, reliable remote storage access with low latency and scalable bandwidth. They were designed for the general-purpose computing environment. It is not surprising that they lack some features needed for high-performance A/V applications, such as guaranteed access bandwidth QoS and large block ( $>>64$  KB) R/W to storage. A NAS connection without guaranteed QoS may result in A/V glitching when the stream is throttled due to server congestion. Also, NFS and CIFS have associated caching strategies for optimizing small block R/W storage access. Unfortunately, they do not optimize large block R/W access, and the caching algorithms leave a lot to be desired in terms of disc access performance.

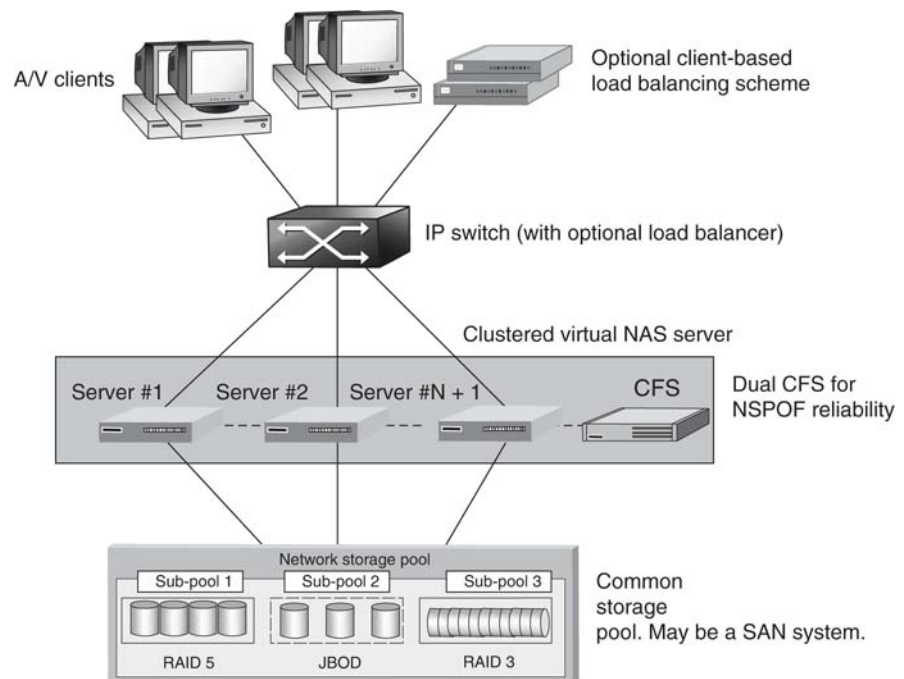
Because of the potentially poor performance of NFS/CIFS in the large block access A/V setting, some vendors have developed proprietary NAS augmentation protocols to meet their needs. Think of these protocols as assisting NFS and CIFS to improve R/W performance. The protocols are proprietary and are not advertised in general.

The basic idea is to provide the A/V client with a *data path* for all media-related transactions with a NAS server and use CIFS (for example) as the *control path* (file open, close, R/W, delete, etc.) only for the connection. The new data path bypasses CIFS's data caching methods; it optimizes both small and large data block storage access, thereby guaranteeing a  $\sim 2\times$  improvement compared to CIFS alone. Also, it may meter every data packet so that a maximum R/W rate may be set per client. This is crucial when many clients connect to the same NAS server.

Each client must be an excellent network citizen; otherwise, mayhem will rule. A single rogue client can ruin the neighborhood if it tries to hog all the bandwidth it can. Shame on bandwidth hogs! This does not mean that garden variety NFS/CIFS cannot be used for A/V applications—only that they may compromise performance due to their data caching methods.

### 3B.3.4 NAS and Server Clustering

It is possible to create a “super NAS” server (or some other server type) by clustering  $N$  smaller NAS servers. This concept is shown in Figure 3B.19, where a CFS unifies  $N$  servers into one. Any client seeking a server may be directed (by IP switching or a load-balancing DNS or other method) to any one server in



**FIGURE 3B.19** Clustered server and NAS connectivity.

the cluster based on loading, security, and reliability criteria. Generally, clients do not know the whereabouts or identity of the individual NAS server that is providing the storage. The servers may be heterogeneous devices and located anywhere on the network consistent with the QoS requirements for the cluster.

High-end NAS servers from some vendors use InfiniBand networking for peak performance. NetApp, for example, uses InfiniBand to cluster the processors of each file server for fault tolerance, thus improving I/O performance and reducing CPU utilization per NAS device.

Aside from the NAS application, server clusters are useful for general A/V computations. Animation rendering, 3D effects, cinema effects, and more are being done on Linux clusters. Each node in the cluster takes some of the computational burden. The nodes may or may not use a CFS.

Server clusters are mature and used every day for a variety of IT operations. They are commonly applied to non-real-time operations, but real-time clusters do exist. See Appendix C for more information on these leading technologies.

A NAS server cluster (or general server cluster) has several very compelling advantages, among which are

- **$N + 1$  reliability using external load balancer.** If one server fails in a cluster of  $N + 1$  servers (the 1 is the spare server), a router can direct new client requests to other working servers in the cluster. Seamlessly moving an *active client* from a failed server to a working server is not easy because the state of any R/W storage transaction must be moved to the new server as well. Most commercial-clustered NAS servers do not automatically migrate active clients to a new server without the current R/W transaction failing. Incidentally, large Web server farms are typically load balanced using an external router or other scheme, and a CFS is not normally used.
- **$N + 1$  reliability using client logic load balancer.** In a system where each client keeps tabs of its R/W transactions, the client may decide when storage access is below par and request another server in the cluster. This is accomplished by the client closing the existing connection to the cluster and opening one to another server. In this case, the client needs to know the identity of the servers so it can avoid the failed one. The load balancing is performed by assigning each client a list of preferred servers in the cluster, including which one to failover to. This scheme works well when the clients are part of a closed system where custom load-balancing software may be installed per client.

Several strategies may be used to guarantee that servers in the cluster are never overloaded. The most popular method is the so-called round-robin approach in which clients are progressively assigned the next server in a circular fashion. This assures approximately equal loading of all servers. Another method is the client-based one described earlier for  $N + 1$  failover.




Adding or removing servers to the cluster may be done “hot” without affecting the overall performance of the cluster. There is almost no limit on how many servers may be included in a cluster. Aggregate cluster bandwidth has been measured up to 12GBps in the IBM general parallel file system (GPFS) product. This system is an excellent example of a clustered NAS server.<sup>4</sup> There are several installations of the GPFS system with 25+ attached professional A/V editors sharing common storage. SGI also offers a Linux cluster using its CXFS A/V-friendly CFS. Linux Beowulf clusters are also in general use as Web server farms and NAS file servers.

SAN AND NAS

In its most basic form, a SAN provides *blocks* of storage to clients/servers, whereas a NAS provides *file* storage. If a clustered file system spans NAS servers, they appear as

one large file server. If a CFS spans SAN clients, they have a view of all stored files.

  
Snapshot

3B.3.5 NAS, SAN, and the Future

It is time to connect all the dots. What overall picture is formed? No doubt SAN and NAS systems are finding applications in all types of A/V solutions. As the foregoing sections discussed, SAN and NAS are complementary in many respects and will coexist for the foreseeable future. Figure 3B.20 outlines the

SAN		NAS	
<div><ul style="list-style-type: none"><li>• Native block storage access</li><li>• Storage virtualization</li><li>• Legacy FC connectivity</li><li>• IP for new systems and FC/IP hybrids</li></ul></div>	A	<div><ul style="list-style-type: none"><li>• Single file server</li><li>• Small systems, IP connect</li><li>• Limited throughput</li><li>• Not easily scalable</li><li>• CIFS/NFS centric</li></ul></div>	No CFS
	D	<div><ul style="list-style-type: none"><li>• Clustered file servers</li><li>• Big systems, IP</li><li>• High throughput</li><li>• Each server needs an IFS</li><li>• Usually NSPOF design</li></ul></div>	
		C	With CFS

FIGURE 3B.20 Sweet spots for SAN and NAS systems.

<sup>4</sup> See [www.ibm.com](http://www.ibm.com) and search for the title *An Introduction to GPFS for Linux*.

chief characteristics for each technology with and without a clustered file system. Each of these four solution spaces has been discussed separately earlier in this chapter or in Chapter 3A. Remember, inclusion of a CFS allows for every attached client/server to access all permitted storage *and* files.

Looking forward, when creating medium- to medium/large-scale collaborative A/V systems (>16 SD/HD edit, compositing stations, or I/O ports, for example), implementers will likely gravitate toward configurations with file sharing and with iSCSI RAID storage access (quadrant D in Figure 3B.20). This configuration enables scalable storage and bulletproof-reliable systems with an excellent price/performance ratio. The migration toward this configuration will not occur overnight, but there are compelling reasons for this direction. There are no servers or Fibre Channel HW to add complexity, and connectivity is based on an IP/Ethernet infrastructure.

The one roadblock to the widescale adoption of shared storage is lack of an A/V industry-accepted CFS. As discussed in other sections, implementing a CFS is non-trivial, and typically each A/V vendor offers one of its own design. As more vendors adopt the open systems approach using Linux, it is possible that a CFS may emerge that is A/V friendly and embraced by the A/V industry or that a particular CFS may rise to the occasion and become a darling of the A/V industry. At present, this seems a distant dream; however, things can change quickly in the IT world, so there is always hope. iSCSI SAN storage connectivity and vendor-specific CFS solutions will likely be the next step in medium/large-scale A/V (shared storage) systems design.

Another configuration that shares the spotlight is based on quadrant B (small NAS version) in Figure 3B.20. Several of these systems may coexist as islands and share A/V materials via file transfer. The quadrant B system may draw files from an NRT near-line storage, too. Building large systems out of many smaller ones adds comfort regarding security, reliability, and scalability. Due to the migration to iSCSI, the NAS-based system may take a backseat to IP SANs for large systems but not for small ones. Systems that require the services offered (data backup, for example) by a NAS server will require either a quad B or a quad C system. Of course, time is the great arbitrator in the ongoing contest of SAN versus NAS.

### **3B.3.5.1 A Sample SAN plus NAS A/V System**

A hybrid SAN + NAS A/V system is shown in Figure 3B.21. The traditional A/V routing infrastructure is not shown. It has the following characteristics:

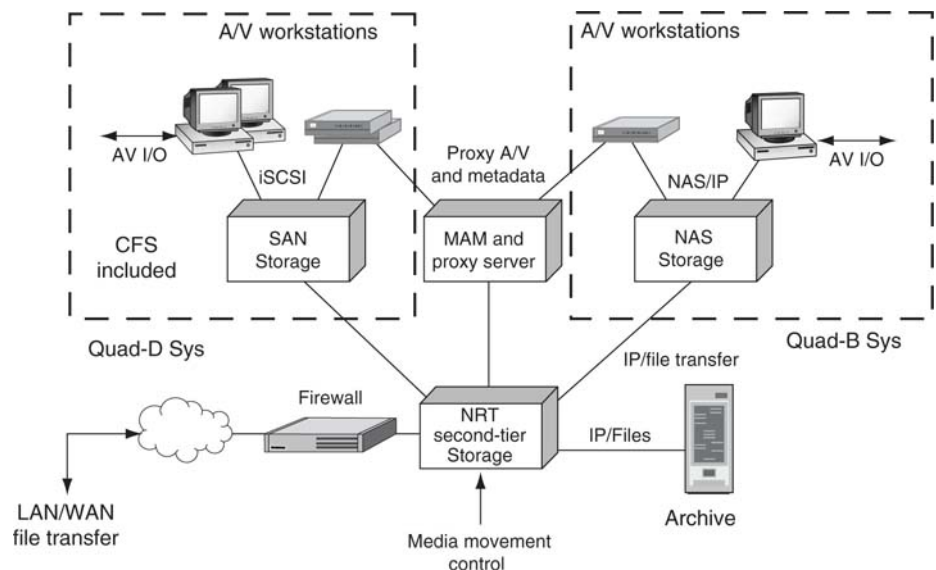
- Quad D IP SAN system offering shared storage and file access to all clients: May scale to many A/V clients and may offer excellent reliability.
- Quad B small NAS offering a simple island of A/V clients using a single file server: Normally limited number of clients. Moderate reliability without

additional NAS servers as alternates. May have more than one island if needed.

- IP/Ethernet connectivity at all levels.
- NRT second-tier storage and archive: The second-tier storage is a repository for all A/V materials needed by workflows. This reduces the amount of online storage needed.
- Industry-accepted systems' management schemas and protocols.
- Media asset management functions available to any client along with low bit rate proxy files.
- File import/export filtered by a firewall and possible digital rights management process.

Figure 3B.21 focuses on a representational view of what is possible using the most appropriate technologies for IT-based A/V systems. Of course, there are other configurations based on the four quadrants of Figure 3B.20, but quadrants B and D designs will become commonplace in systems architectures.

Type C will not find widespread acceptance but will be used in some large systems. The quad C server layer adds little functionality compared to the quad D (IP-based) configuration for most A/V applications. This statement may start a fight in some bars. Why? Well, there is a need to install a small IFS file redirector software component on every type D client/server node, whereas no IFS



**FIGURE 3B.21** Hybrid large/medium IP SAN and NAS systems.

is needed for type C clients. Yes, the type C “hides” the CFS from the clients because it is embedded in the NAS cluster. Plus, some type C NAS servers offer loads of backup/restore services. These are big advantages. Nonetheless, a quad D is a compelling solution and achieves excellent storage performance and file access to clients and servers. Finally, quad A will find only niche A/V applications (no file sharing support), although it is very popular in general IT business environments.

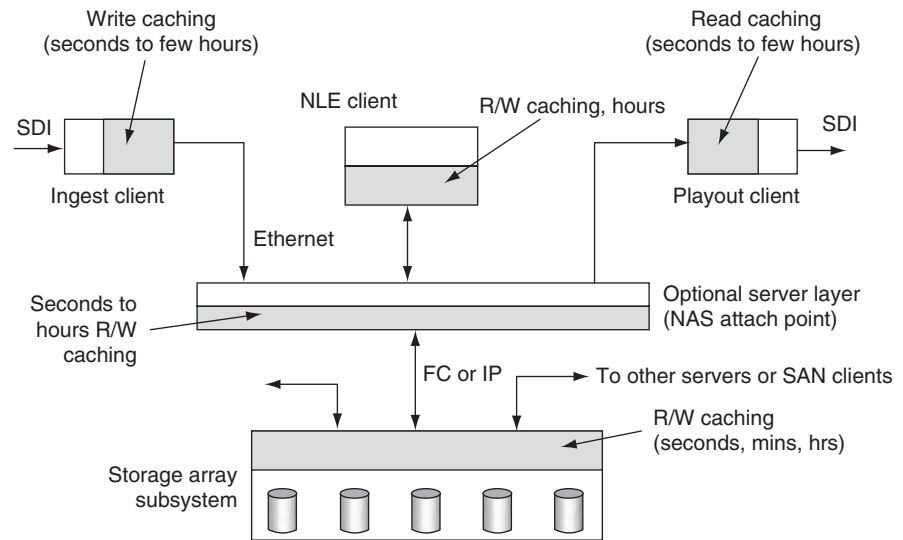
Of course, there are countless reductions and reconfigurations of Figure 3B.21 to meet specific workflows. Also, this discussion is focused on SAN/NAS access. There are other classes of systems based on streaming (over IT infrastructure) and file transfer. See Chapter 2 for a more complete discussion of methods for the accessing/transferring/streaming of A/V materials. When you are using DAS, SAN, or NAS, data flow may be improved significantly if caching is engaged. The following section outlines the advantages of caching.

### 3B.4 CACHING METHODS

A *cache* is defined as a block of memory (RAM or disc less frequently) reserved for temporary storage. Caches usually store data to/from slower disc arrays to make client access faster. A cache may decouple linear from random access. When an application references a data address in a storage array, the cache is checked to see whether it holds that address. If it does, data are returned to the application; if it does not, a regular memory access occurs. Caching is not the same as buffering. A buffer is a small (seconds usually) memory for smoothing out irregularities due to network I/O jitter or other reason. An A/V data cache can do the job of a buffer, but its main advantages aim for higher goals. A few definitions are in order before the advantages are outlined:

- **Cache hit**—The request for an external storage R/W operation is fulfilled by the local cache. If an I/O request is met 50 percent of the time by the local cache, then the network infrastructure is used 50 percent less and the external storage is accessed 50 percent less.
- **Cache coherency**—This is the property that accessing a cache gives the same values as the underlying main storage values. Cached data can become stale when other processes change main storage array data.
- **Cache location**—Caches may be in clients, in NAS servers, or in the storage subsystem. Each location has a particular advantage.

Figure 3B.22 shows a typical shared storage client-attached video system with all the caches clearly identified. This system exaggerates the use of caches, but for our purposes this is okay. Analyzing cache efficiency can quickly become a very complex problem due to the variables of hit rates, cache sizes/locations, I/O rates, and workflows. Instead of a quantitative view, let us have a



**FIGURE 3B.22** *Caching methods in a shared storage system.*

qualitative discussion. In general, A/V data caching will have advantages in the following areas:

1. Increase reliability of networked clients
  - Networked clients with an internal cache have increased reliability. The local cache allows for a client to lose contact with the external storage (due to temporary connectivity or storage failure) and still function as the cache is temporarily substituted for the external storage. Consider the case in which a playout client is prequeued with a playout file. The cache holds queued data. On command to play the file, the client reads from the local cache, so even if the external storage connection is disrupted, the local client outputs the A/V without a glitch.
2. Increase networked bandwidth utilization
  - Local cache hits effectively make an I/O appear faster. RAM-based cache R/W data rates are greater than that obtainable from external storage. As a result, a cached R/W operation reduces network traffic on average and appears to increase the application's read/write I/O rates.
3. Increase storage array data rate utilization
  - When a storage array has a cache in the main controller, hard disc R/W operations may be block optimized to increase the array's total available data rate.
  - When the HDD has cache, the R/W operation is noticeably faster.

#### 4. Decrease system storage access latency

- Whenever a cache hit bypasses external storage hard disc access, then the apparent R/W latency will decrease.

Cache efficiency is a strong function of data workflows. Ingest clients are write intensive with almost no cache hits for any cache in the system. Playout clients are read intensive with some cache hits likely for repeated plays. Editing clients are read intensive with some write activity. Many NLE editors never modify actual master A/V data but only generate compositional metadata lists to record the edit decisions. NLE clients show the most potential for cache hit advantages as users jog and shuttle around the timeline. Because a cache adds little or no performance gain if data streams have no read cache hits, the smart system designer must not count on cache gains for worst-case operational scenarios.

The use of caches is a vendor design decision. Some vendors may not include them, and another may design for their advantages. Good caching implementations will yield all or some of the advantages listed already, but only when the workflows and data access modes are favorable, which will rarely be 100 percent of the time.

### 3B.5 IT'S A WRAP: SOME FINAL WORDS

This chapter has put some flesh on the bones of DAS, SAN, and NAS. These three technologies are being used in A/V system design every day. Expect to see the iSCSI SAN and possibly FCoE become the next waves of networked storage connectivity. They offer compelling performance, Ethernet networking, and excellent management. Of course, a CFS is needed to extend a SAN to a shared file paradigm. Also, NAS clusters can take advantage of iSCSI as their storage access method. Storage is moving in the right direction for A/V systems, and its advancement will provide A/V configurations with new and compelling workflows at excellent price points.

## REFERENCES

- Cameron, D., & Regnier, G. (April 2002). *Virtual Interface Architecture*. Santa Clara, Ca, USA: Intel Press.
- Chudnow, C. T. (July 2002). Fibre Channel dukes it out with IP. *Computer Technology Review*.
- GlassHouse Technologies, White Paper, *Uncovering Best Practices for Storage Management*, [www.glasshousetech.com](http://www.glasshousetech.com), 2002.
- Gruener, J., Giganet (now owned by Emulex), *Building High-Performance Data Centers*, 2000.

This page intentionally left blank

# Software Technology for A/V Systems

## CONTENTS

4.0 Introduction	158
4.1 User Application Requirements	159
4.2 Software Architectures—The Four Types	160
4.2.1 Centralized Computing	161
4.2.2 Distributed Computing	162
4.2.3 Architectural Comparisons	179
4.3 Middleware Connectivity	179
4.3.1 Database Connectivity Protocols	182
4.4 Implementation Frameworks	183
4.4.1 The .NET Framework	183
4.4.2 The Java EE Framework	184
4.4.3 The Burden to Choose	186
4.5 Methods of Virtualization	186
4.5.1 Under the Hood: Servers	188
4.6 Open Source Software	190
4.7 High-Performance Real-Time Systems	191
4.7.1 Achieving Real-Time OS Performance	191
4.7.2 Multimedia Extensions and Graphics Processors	191
4.7.3 64-Bit Architectures and Beyond	192
4.8 Software Maintenance and System Evolution	192
4.8.1 Lehman's Laws	193
4.9 It's a Wrap—A Few Final Words	194
References	194

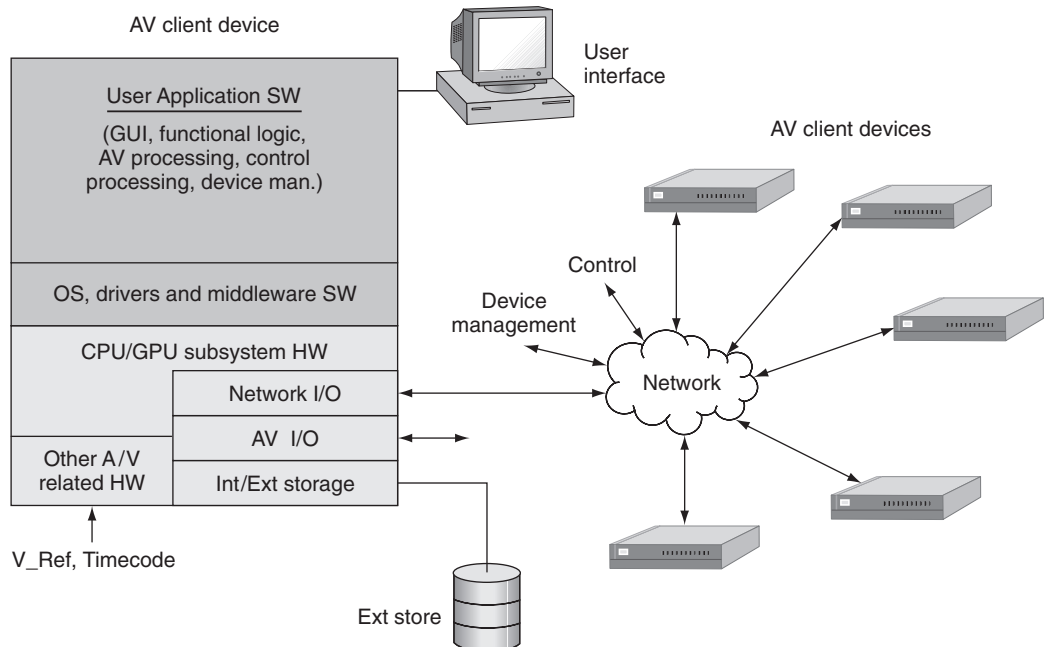


## 4.0 INTRODUCTION

Only a few years ago A/V devices were hardware centric with little software value. Video product engineers were experts in real-time, frame accurate, circuit, and system design. Today the roles have reversed, with software being the center of functional value for most devices. Software, coupled with sufficient CPU power, can perform almost any A/V operation in real time. Thanks to Moore's law, A/V-specific hardware is being relegated to I/O, live event switching, some real-time 2D/3D effects, and signal processing. Standard definition MPEG encoding and SD/HD decoding are done easily with a common CPU. HD MPEG2 encoding, especially MPEG4 Part 10 (H.264), still requires hardware support. When we look down the road, hardware A/V processing will become a rare commodity and software will rule. One important trend is to use the graphics processing unit (GPU, common in all PCs) to do real-time 2D/3D effects.

Figure 4.1 illustrates the software-centric nature of A/V systems. Of course, not every element has A/V I/O, but this only increases the saturation of software in the overall system. This chapter provides a working knowledge of the salient aspects of software as a system element. This includes

- User application requirements
- Software architectural models—four main types
- Software implementation frameworks—.NET, Java EE 5, Web services



**FIGURE 4.1** The software-centric AV/IT client.

- Open source systems
- Real-time systems
- SW maintenance and system evolution

Performance wise, one might think that Moore's law would always help speed up application execution. But due to software complexity and bloat, Wirth's Law comes into play ([http://en.wikipedia.org/wiki/Wirth's\\_Law](http://en.wikipedia.org/wiki/Wirth's_Law)). It states that software is decelerating faster than hardware is accelerating. Of course, this is not always true, but the law does have a profound effect for many applications. In jest, it has been said that "Intel giveth while Microsoft taketh away."

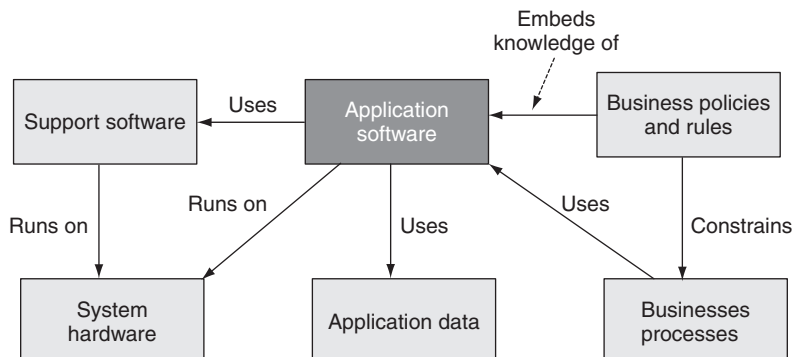
This chapter does not provide specific coverage of programming languages or practices, although we do touch on these subjects along the way. Let us get started.

## 4.1 USER APPLICATION REQUIREMENTS

Software user applications come in a variety of shapes and sizes. The two main user application types are server based and client based. Software "services" are used by applications but are not usually complete applications. Service concepts are covered later in this chapter. Figure 4.2 illustrates a *context diagram* for *user application software*. This high-level example shows the action verbs associated with an application. Context diagrams illuminate how a software component integrates into the bigger picture. When you are completing an application design, a context diagram is useful to locate missing or duplicate relationships.

The core application features and issues are as follows:

- **Functionality**—Does it meet your operational and business needs?
- **GUI look and feel**—How does the interface behave?
- **Ease of use**—Are operations intuitive? Training needed?
- **Performance**—Are benchmarks available to compare vendors' products?



**FIGURE 4.2** Context diagram for application software.

Table 4.1 User Interface Design Principles

Principle	Description
User familiarity	The interface should use terms and concepts familiar to most users. The interface should not break the good habits that most computer users have formed.
Consistency	Comparable actions should operate the same way. All commands and menus should follow the same format.
Minimal surprise	Users should never be surprised by behavior.
Recoverability	Users should be able to easily recover from errors—undo, for example.
User guidance	When errors are encountered, provide meaningful advice to recover. A dialog box displaying “Process Quit, OK?” is not acceptable.
User diversity	The interface should provide different levels of operation—novice to advanced users if appropriate.

- **Quality of result**—Does the A/V output quality meet business needs?
- **Standards**—Does it conform where applicable?
- **Reliability**—Is it stable when pushed to limits?
- **Network friendliness**—How well does it perform during network anomalies? Is there a readily available pool of trained users?

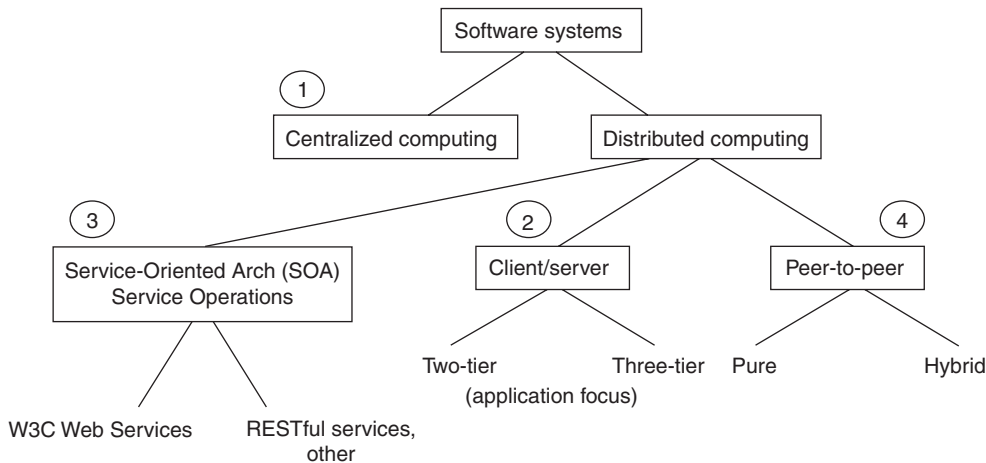
When evaluating a product, consider these minimal aspects of design. Some of the features are best evaluated by the technical staff and others by end users. Since the GUI is ever present, let us focus on six principles for good design, as outlined in Table 4.1.

Some well-meaning interface designers try to be too clever and break the first rule, believing that new user paradigms are a good idea. True, there is room for innovation, but designers should leverage users’ good habits when possible. Some applications are self-contained and do not depend on any networked resources, e.g., using a word processor on a PC. More and more, however, applications rely on available services across a network. The end goal is a great application that users find compelling from all angles. What software architectures are available to reach this goal? The next section reviews the four main types in common usage.

4.2 SOFTWARE ARCHITECTURES<sup>1</sup>—THE FOUR TYPES

At some point when users are discussing software, religion usually enters the picture. For sure, there are priests and disciples of programming methods and

<sup>1</sup> Although our focus is on software, the concepts discussed in this section relate equally to computer systems in general, so the terms *software system* and *computer system* are used interchangeably.



**FIGURE 4.3** *Software system's taxonomy.*

languages, operating systems, and architectures. In a non-denominational sense, this section reviews the central methods for solution construction using software.

What is a software architecture? Whether it is the humble microwave oven's embedded processor or Google's 500K+ searching servers, their software is constructed using a topology to best match the functional objectives. If done poorly, it is a pile of spaghetti code; if done well, it is more like an award-winning building or well-thought-out city plan.

The city plan is a useful analogy. For example, central Paris is constructed as a star centered on the Arc de Triomphe, whereas Manhattan is grid based. These are high-level plans and do not normally constrain how individual buildings look and feel. From this perspective, let us examine four different "city plan" architectures without detailed concern for the individual buildings. Figure 4.3 shows the taxonomy of the four plans under discussion. For sure, there are other ways to divide the pie. This illustration is not meant to be a rigid definition, but merely one way to segment the architectures under discussion. The first to be considered is the centralized architecture.

### 4.2.1 Centralized Computing

Centralized computing is a monolithic system ranging from a basic computer with a user interface (the ubiquitous standalone PC) to a mainframe computer system with hundreds of simultaneous users running multiple applications. This is the most basic plan and has been in use since the days of ENIAC, the world's first electronic, programmable computer. It filled an entire room, weighed 30 tons, and consumed 200kW of power when commissioned in 1945 (ENIAC). Oh, the joy of Moore's law.

Many A/V devices are monolithic in nature. The standalone video clip server, character generator, PC-based edit station, and audio workstation are all examples. Most of these are single-user applications. Older mainframe systems never found a niche in A/V applications and are less popular with the advent of client/server methods. Large, centralized systems can suffer from scalability and single-point-of-failure reliability problems. See Figure 4.4 for a view of these two types of design. Importantly, IBM, which pioneered OS virtualization for mainframes, has breathed new life into the machines. It is common to run 200 virtual Linux servers under one mainframe (System z) for large enterprise applications.

Moving on, another class of system is distributed computing. This main class is divided into three smaller classes. The following sections review these methods.

4.2.2 Distributed Computing

In the distributed computing category, there are three main classes (marked 2, 3, and 4 in Figure 4.3). All share the common attribute of using networking and distributed elements that, when combined, create a solution. The three classes are as follows:

- 1. Client/Server<sup>2</sup> (application focused)
  - a. Two-tier C/S (flat)
  - b. Three-tier C/S (hierarchical)

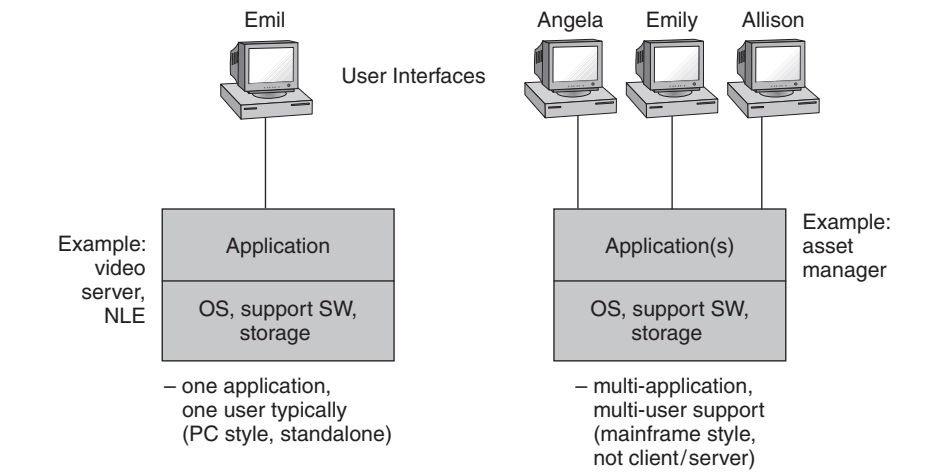


FIGURE 4.4 Single and multiuser centralized systems.

<sup>2</sup> In this chapter, the term C/S is used to represent client/server and the relationship between these two elements.

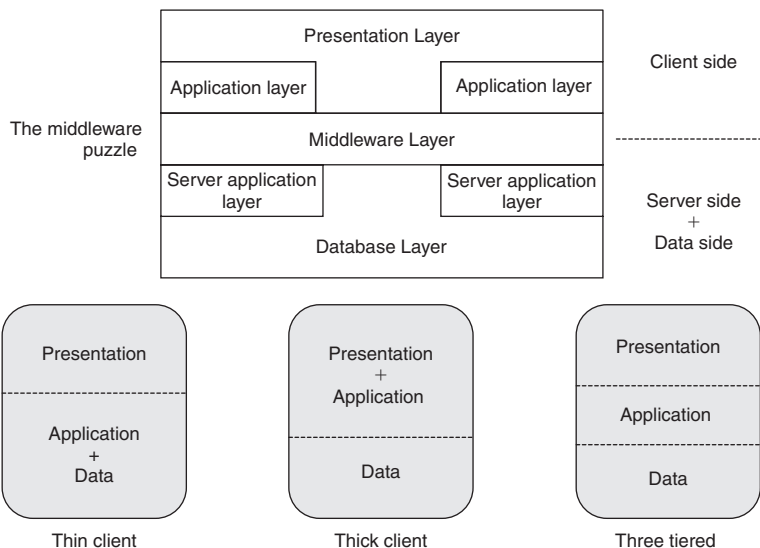
2. Service-Oriented Architecture (SOA) (service operations)
  - a. W3C Web services
  - b. RESTful Web services, other
3. Peer-to-Peer
  - a. Pure
  - b. Hybrid

Let us review each of them.

#### 4.2.2.1 The Client/Server Class

The Web as we know it rests on the bedrock of the client/server architecture. The client in this case is a Web browser, and the server is a Web server accessed via an address such as <http://www.smppte.org>. The C/S model is found in all aspects of modern software design, not just Web-related systems. There are three layers to a client/server dialog: presentation, application, and data. These layers are glued together using various middleware connectivity protocols. Figure 4.5 (top) shows middleware as a connector that links the various layers as needed. Middleware is discussed in more detail in Section 4.3. Figure 4.5 (bottom) shows three different but common C/S configurations.

- **Thin client:** The presentation layer is on the client, and the application logic and data are on the server. An example of this is a terminal with a simple Web browser connecting to a Web server. The client is called “thin” because it hosts only the browser and not the application or data functions. This is a two-tier model.



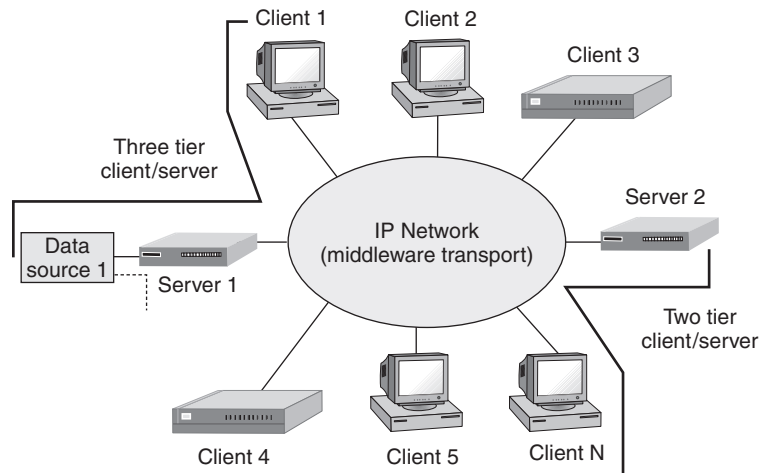
**FIGURE 4.5** Middleware and layering options.

- **Thick client:** Both the presentation and application logic are on the client, hence its thickness. An example of this is a sales report (spreadsheet) where data cells are filled in from a remote database. This is a two-tier model.
- **Three tiered:** In this case, the client supports the presentation, the server supports the application logic, and the database is the third layer. Examples of this configuration abound, and many Web servers are based on this model. The client is thin in this model.

Figure 4.6 illustrates examples of two- and three-tier models in a networked configuration.

As the naming implies, clients and servers are separate entities that work together over a network to perform a task(s). An exact definition of *client/server* is cumbersome, so let us define the term by its characteristics. The following definitions were loosely paraphrased from (Orfali):

- **Server:** The server process is a provider of services, and the client process is a consumer of services. The C/S method is a clean separation of function based on the idea of a service. The service may be a complete application (Web server, file server, email server, etc.). A service may also perform a simple function, such as a file converter or a currency value converter.
- **Asymmetrical:** There is a many-to-one relationship between clients and servers. *Clients* initiate a dialog by requesting a service. *Servers* respond to requests and provide a service. For example, one Web server can provide Web pages to thousands of simultaneous clients.



**FIGURE 4.6** Client/server in a two- and three-tier architecture.

- **Transparency of location:** The server may be anywhere on the network or even within the client platform. The client has no knowledge of the server's location. Client interfacing hides the connectivity to the server. In reality, servers that are distant from the client will offer a large response time if the network QoS is poor.
- **Mix and match:** The client and server platforms may be heterogeneous and allow for any combination of OS and programming languages on either side. This is why a PC or Mac Web browser can access the same Linux-based Web server without interoperability problems (ideally). This is a key feature and allows programmers to choose the client/server OS and programming language that best suits their needs.
- **Scalability:** C/S systems can be scaled horizontally or vertically. Horizontal scaling is adding/removing clients with little or no performance impact. Vertical scaling adds servers to meet loading and performance requirements.

These client/server characteristics permit systems to be built with distributed intelligence across a network. Clients may have human interfaces. As a result, there is no reason to be dogmatic in our definition of the C/S configurations, but rather embrace the openness and flexibility of the concepts.

Take another peek at Figure 4.6. The two- and three-tier nature of C/S is illustrated. The notion of three tiers adds a database layer. Instead of the database being integrated into the server, it becomes a separate element that may be shared by many servers. When the data storage aspects are separated from the service aspects, it is easier to scale and manage large systems. Hierarchical systems provide for scalability and data separation at the cost of complexity. Specialized versions of C/S are grid and cluster computing, which are discussed in Appendix C.

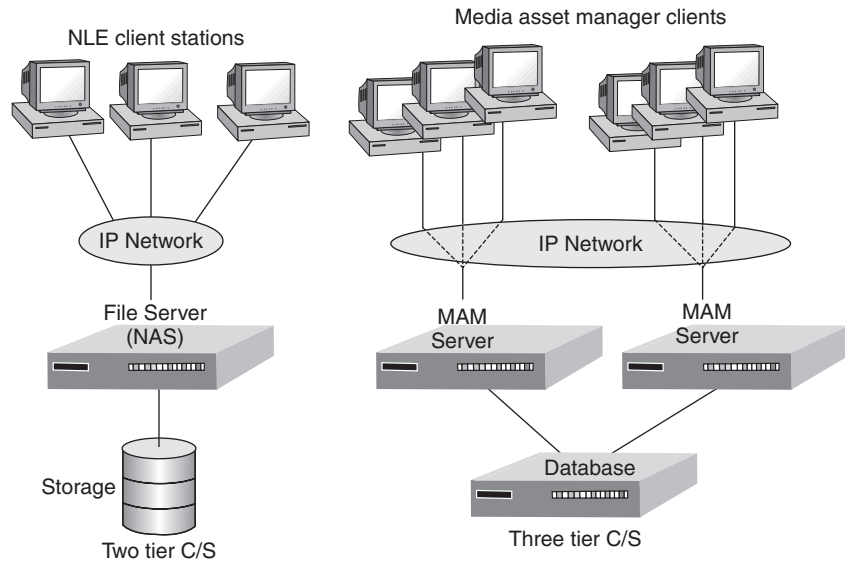
In the framework of A/V systems, the C/S configuration is quite popular. Figure 4.7 illustrates two examples. The two-tier NAS server example is discussed in Chapter 3A. With the three-tier model, media asset management (MAM) servers share a common database, which permits application server scaling independent of data. The three-tier model is a great way to scale a system. Of course, the networking QoS should support the C/S sustained data rates. Failover is not shown but could be included in the design. Next, let us look at a specialized version of C/S called the service-oriented architecture.

#### 4.2.2.2 The Service-Oriented Architecture (SOA)

The sage Bob Dylan penned these lines (1963):

*Don't stand in the doorway  
Don't block the hall  
For he that gets hurt will be the one who has stalled  
— For the times they are a-changin*



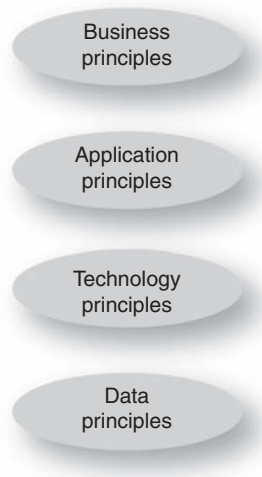


**FIGURE 4.7** Sample A/V applications using client/server.

Dylan's words are great advice to anyone who wants to constrain software to the comfort of application-focused client/server or standalone systems. These methods are quietly giving way to various forms of the service-oriented architecture (Figure 4.3, item 3).

Figure 4.8 shows one simplified landscape of a SOA. It starts from business principles. This is absolutely key to a true SOA implementation. From business needs, application principles are defined. Applications rely on technology (distributed software services, networking, servers, etc.) for implementation. Finally, the stack rests on data principles (values, formats, schemas, interfaces, metadata, etc.). A SOA is a powerful ally for creating business efficiencies for the media enterprise.

Consider an example of Figure 4.8 in operation. At the business level, there is a request to repurpose an archived program for the Web. This may include locating and viewing a proxy version, checking on usage rights and time windows, editing, transcoding, QA, and publishing. Appropriate applications are invoked, some automatically. These applications may be a mix of services (IP rights, auto-transcoding, and so on), a job request for some minor editing tweaks, and publishing to the Web. The entire stack is applied to meet the business need to publish media content. The more automated this process, the more cost effective and timely it will be. Incidentally, the lowest two layers may be time-aware for video frame accurate signal and control purposes. Video frame accuracy is a special need of the media enterprise.



**FIGURE 4.8** *The SOA principles stack.*

SOA provides the discipline to create scalable, reliable, managed business workflows. Using the technology principles of service reuse, granularity, modularity, and interoperability, a SOA provides a firm infrastructure for efficient applications deployment. Agility (easy reconfiguration for change) and visibility (dashboards, key performance indicators—KPI, see glossary) are also major benefits of a SOA.

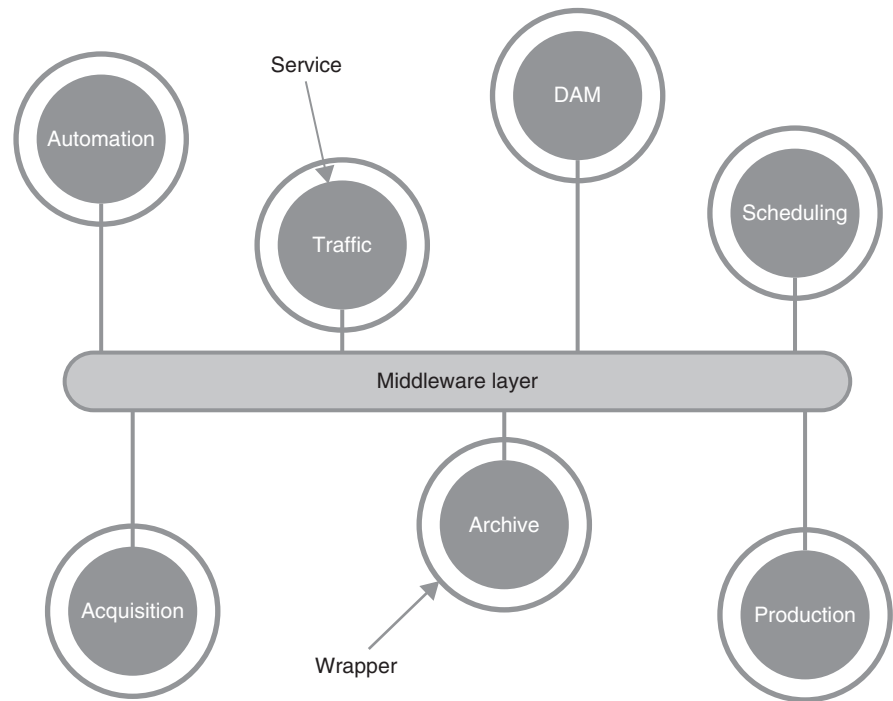
SOA is broadly defined as *an architecture of loosely coupled, wrapped services communicating via published interfaces over a common middleware layer* (Footen, Faust). First of all, this definition is not universal; there are others, but this will suit our needs and scope. This terse definition needs some explanation. The terms *services*, *middleware*, and *wrappers* will be explained. See Figure 4.9.

Before these terms are examined, a few words of caution: there are no strict term definitions that all industry gurus agree on. Some may quibble with the precision or scope of a definition. Given that, let's proceed.

Starting with *services*, the definition most often given is a loosely coupled component (software normally) capable of performing a task.<sup>3</sup> As an example, Amazon publishes its core system services (listings, searching, viewing products, etc.) so that its business logic can be used by external partners. Do a Web

---

<sup>3</sup> Don't confuse Web services with a Web server; they use completely different software models. The differences are outlined in the course of this chapter.



**FIGURE 4.9** A SOA's basic elements.  
Source: (Footen, Faust).

query for “Amazon Web services” to find examples. The services are defined by API interfaces. The services outlined in Figure 4.9 are generic examples and may be replaced with others ranging from simple transcoding to more sophisticated applications. Service technology is covered in the next section.

What is meant by loosely coupled? Basically, the service is independent, self-contained, stateless (usually), and isolated such that implementation (C++, C#, Java, and so on) and location are hidden. So, changing a service’s location (moved from server A to server B) and implementation language should not affect how the service performs within the limits of its QoS spec.

*Middleware* is defined in Section 4.3.

*Wrappers* sit between a service and the middleware layer and transform messages that pass through them. Wrappers provide a layer of abstraction necessary to connect applications or tools not originally designed as services. Wrappers can mask the technical differences between components, making them uniformly interoperable in the SOA environment. Wrappers are used only when needed.

### SOA in Context

Finally, SOA acceptance is growing in mid- to large-scale enterprise organizations. According to Randy Heffner of Forrester Research, at least 63 percent have implemented at least one SOA implementation by the end of 2008. So,

looking forward, many media organizations will be using SOA principles. This impacts equipment design, since vendors will be providing service interfaces on their products to better interface into SOA environments. In the end, this creates more flexible workflows and efficient facilities.

Of course, there is much more to learn to fully appreciate SOA. Consider researching topics such as the Enterprise Service Bus (ESB), management tools (HP's SOA Manager), frameworks (IBM WebSphere SOA), business process management methods (Business Process Management Initiative, BPMI.org), standards from OASIS ([www.oasis-open.org](http://www.oasis-open.org)), and the Advanced Media Workflow Association ([www.AMWA.tv](http://www.AMWA.tv)) for media-facility specific standards and best practices. Incidentally, this author is a board member of AMWA and contributes to document specifications for our industry. See, too, materials from the SOA Consortium for case studies ([www.soa-consortium.org/case-study.htm](http://www.soa-consortium.org/case-study.htm)).

Next, is a summary of the generic Web services model.

#### 4.2.2.3 *Web Services Model*

A service is a networkable component that provides well-defined functions (tasks) using a standardized data interface. An example is a service that converts currency from dollars to Euros. The input to the module is a dollar amount, the method is to convert to Euros using the current exchange rate data, and the returned value is in Euros. If the service is well defined, then any client can call on it for the conversion service. Importantly, the service may be called by a variety of unrelated user applications. It is not difficult to imagine a collection of individual services whose aggregated functional power is equivalent to, say, a standalone application server.

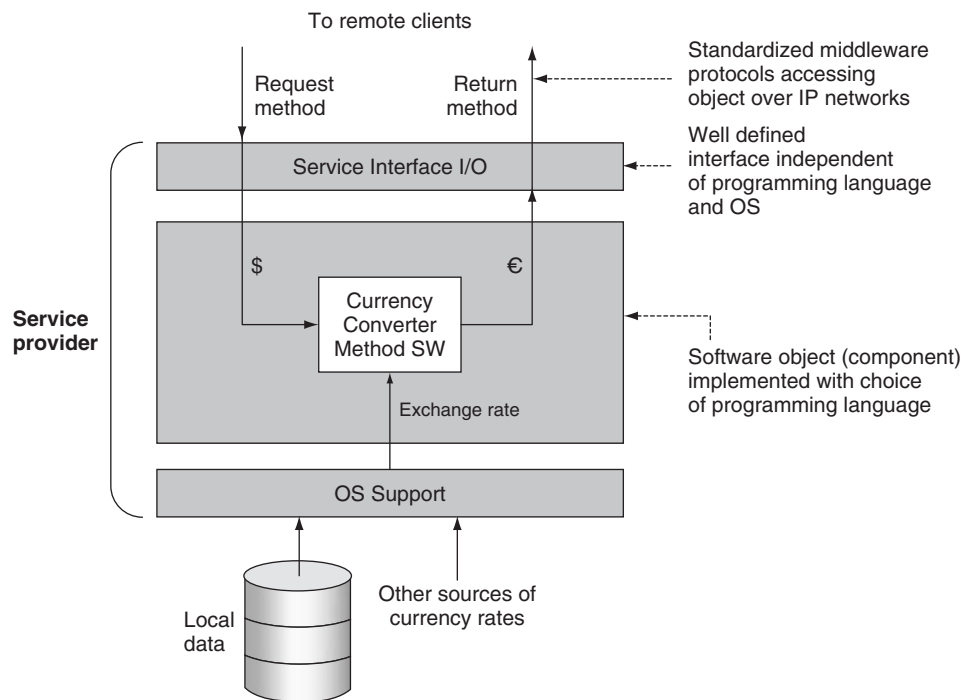
The term *Web service* is somewhat a misnomer and has generated no end of discussion. A service is not required to have a Web connection. This is a generic name and really implies network connectivity—Web based or otherwise.

Figure 4.10 shows a single service for converting dollars to Euros in a server environment. The component is invoked by some external client, and it returns the value in Euros based on the input value in dollars. Another example is a file metadata service; you provide a file name, and its structural metadata are returned—compression format, aspect ratio, bit rate, length, and so on. One of the more important aspects of service definition is its interface specification so that any heterogeneous client may use the service.

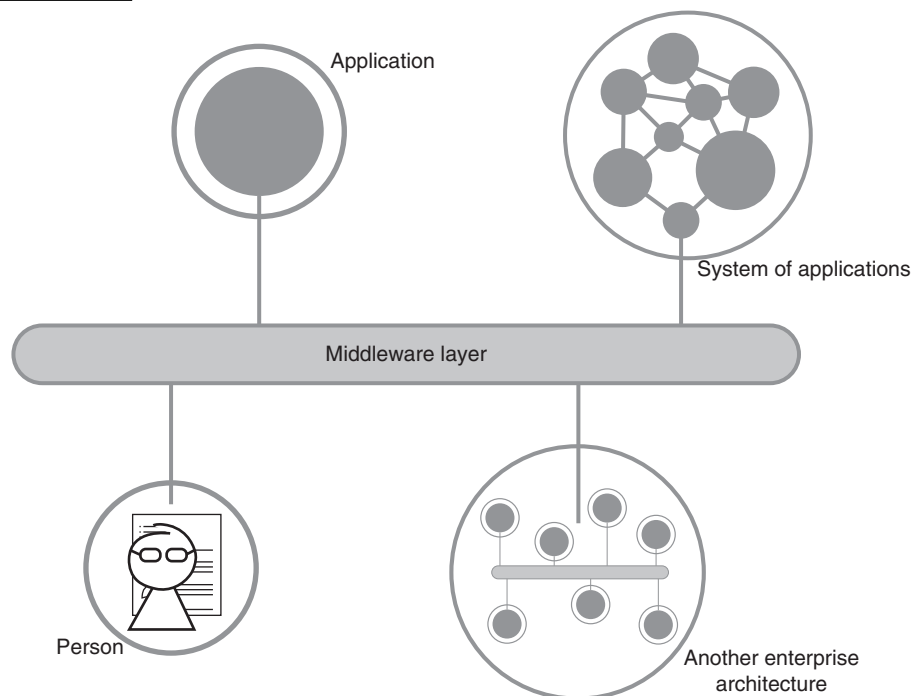
### Some Insights into the Services Landscape

Not all services fall into the neat categories shown in Figure 4.9. Most often a SOA system will require services to span people (yes, people can offer SOA services), simple application services, connection to legacy applications, and to other large systems. Figure 4.11 shows a landscape of services spanning these domains.

Unless a designer has a blank canvas to start with, the services will need to span across system components not always designed for services support. Often,



**FIGURE 4.10** Example of a software service.



**FIGURE 4.11** A landscape of service connectivity.  
Source: (Footen, Faust).

business-critical information is scattered among islands of data and stored in different formats on incompatible systems. Adapters can create a bridge to provide a consistent middleware interface and data formats. Companies such as Pervasive Software ([www2.pervasive.com](http://www2.pervasive.com)) provide service adapters for hundreds of legacy systems and cross-convert files, service interfaces, messages, applications, and database formats.

But what about the people interface? Today, work tasks may be assigned via emails, phone calls, and office memos. These methods are incompatible with SOA. A better way is to implement a to-do list as a network attached service with a UI. All tasks are formally documented and registered with this service application. Progress steps may be updated. When a task is complete, the “finished” button is hit—often to great satisfaction. Task progress and completeness contribute to business intelligence. If this new approach is not perceived as a positive business process, then users will bypass it and return to emails and phone calls.

In the previous section, you examined a content repurposing example. One of the process steps is to implement some creative editing tweaks. Ideally, this job should be recorded with a SOA-enabled task register. Then, once complete, the next step (QA review) can be automatically launched. The bottom line is a faster and more efficient workflow with good business intelligence visibility. For more information on the “people interface” for SOA, see, for example, BPEL4People at [www.wikipedia.org/wiki/BPEL4People](http://www.wikipedia.org/wiki/BPEL4People).

The service interface definition may take one of several forms. The two most popular are the W3C Web services model and the RESTful model. When experts compare these two service models, the discussions can quickly become a religious debate. They each have application spaces, and neither one is ideal for all use case scenarios. Next, let’s look at each one.

#### 4.2.2.4 *The W3C Web Services Model*

The World Wide Web Consortium has defined a version of Web services that is in wide usage. See (Erl), (Barry), (Graham), and [www.w3.org/2002/ws](http://www.w3.org/2002/ws). The W3C specification defines the interfaces and middleware but is silent about OS, programming language, and CPU choices. W3C methods fill out many of the details left open in the generic view of Figure 4.9. Designers have great latitude in how they implement services using tools defined by the W3C. This model has had wide acceptance in enterprise SOA applications development.

Web services<sup>4</sup> have the following characteristics:

- **Self-describing:** Web services have a well-defined interface.
- **Published, found, and invoked over a network:** A network is the communication media that Web services participants—requestors, brokers, and providers—use to send messages.

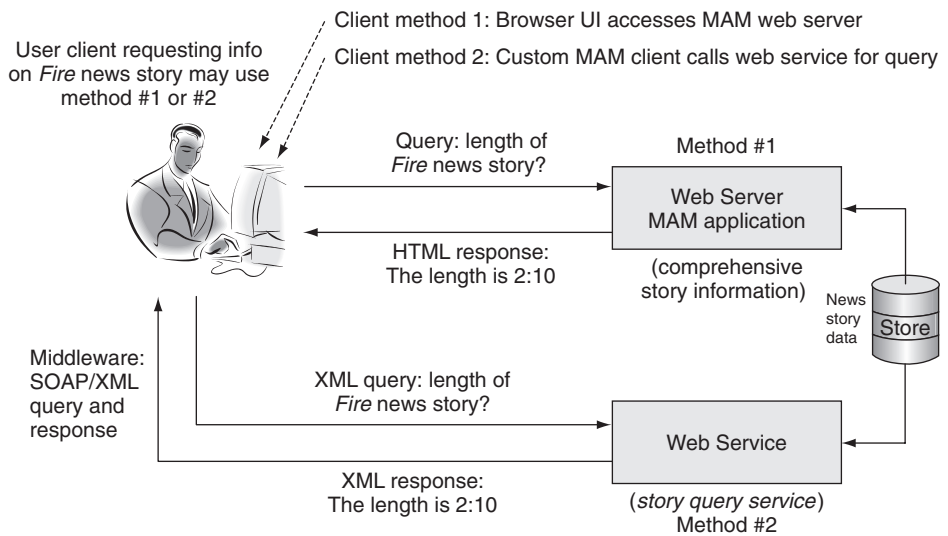
---

<sup>4</sup> Web services definitions were derived from a course presented by IBM.

- **Platform and language independent:** web services can be implemented on different platforms with a variety of programming languages.

Both Internet client/server and Web services use IP networking as a transport mechanism, but their purpose and implementation are different. Traditional Web servers return HTML text/graphics that are displayed in a browser. Web services, however, return XML messages/data to a client process that subsequently uses the information. For the most part, Internet client/server applications are process-to-human centric, whereas Web services are process-to-process centric.

Figure 4.12 shows two methods by which a client queries for the length of a news story. In the first case, the client uses the PC browser and accesses a Web server (with media asset management functionality) that provides the query/response operation. The client/server transaction is HTML/HTTP based. This is an example of a “thin client,” as shown in Figure 4.5. In the second scenario, the client runs a local MAM application installed on the PC. This application calls upon the Web services query/response operator when needed. The MAM client application formats the response for display. This is an example of the “thick client” shown in Figure 4.5. The Web service query/response interface is defined using a SOAP/XML methodology. Each of these two methods has trade-offs in terms of performance and user look-and-feel.



**FIGURE 4.12** Browser-based and Web services application examples.

**SOAP AND XML: THE CLEAN WAY TO COMMUNICATE**

Pass the SOAP and XML, please. SOAP has floated to the top as a preferred way to transport XML messaging. XML is widely used for packaging messaging and data elements.

SOAP/XML forms the foundation layer of the Web services stack, providing a basic process-to-process messaging framework that other layers can build on.

**Under the Hood of Web Services**

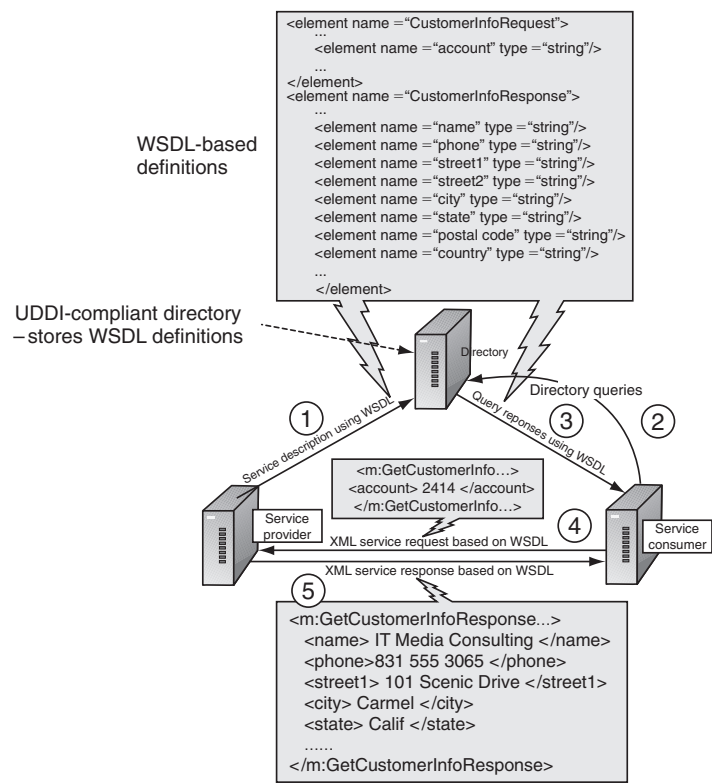
Web services may be advertised so that others can use them. For example, the “story query service” in Figure 4.12 can publish its service methods to a registry. Applications can then find the story query service and use it. Figure 4.13 shows the three players in the Web services dance: the service provider, the service consumer, and the service broker (a directory or registry). The dance goes like this:

1. The service provider registers its methods with the service broker. The broker knows the location and methods of every registered service.
2. Service consumers (clients) inquire of the broker to locate a service.
3. The broker returns the service address and methods.
4. Clients may then use the service as needed (transactions 4 and 5 in Figure 4.13).

The key tools in Figure 4.13 are SOAP, XML, WSDL, and UDDI. SOAP and XML are explored in Section 4.3. The Web Services Description Language (WSDL), expressed in XML, defines the methods and data structures employed by a service. It offers the first standard way for clients to know exactly what methods a service can perform. Consumers parse a WSDL document to determine the operations a Web service provides and how to execute them. UDDI is the Universal Description, Discovery, and Integration protocol. A service provider registers its service functionality, expressed in WSDL, with the directory using UDDI. The discovery broker in Figure 4.13 supports both UDDI as a registering mechanism and WSDL as a service descriptor. Web services can exist without UDDI, but the services must be advertised by other means. A study of Figure 4.13 shows the common use of XML/WSDL for message exchange.

The Web services concept holds great promise for A/V environments. There are many workflows used daily in TV stations, post-houses, and other A/V facilities worldwide that can benefit. Imagine a collection of Web services for logging, cataloging, querying, archiving, controlling, scheduling, notifying, transferring,





**FIGURE 4.13** W3C's Web services communication model.

converting, testing, analyzing, and so on. Vendors can develop a cache of these services and use them to create turnkey solutions based on customer need or a competent A/V facility staff programmer can assemble these services to perform useful applications. This may not be practical for small facilities but may be for larger ones. The possibilities are endless. Of course, the real-time capabilities of Web services depend on the QoS of the service definition.

Despite the promise of UDDI, it has not gained wide acceptance by industry or for the open Web. Its goal of a universal place to find services has remained a dream. On the surface, UDDI seems a great idea that providers would flock to. In reality, other methods have replaced it for the most part. In the enterprise, private service libraries have become the norm. An overseer of services maintains a list of WSDL services for all programmers to access.

For the general Web, a global index has never found a home either. Who would own it and maintain it? What is the business model behind it? Who will vouch for a service's reliability and security? Services for the public Web are largely offered by Amazon, Google, Microsoft, Yahoo!, and others. In addition, these services are not exclusively WSDL based, so UDDI would offer, at best,

a partial registry. The UDDI is not dead, and time will tell if it will find wider acceptance.

To promote best practices for service developers, the Web Services Interoperability Organization (WS-I) has defined 25+ standards and recommendations ([www.ws-i.org](http://www.ws-i.org)). Among the more notable ones are WS-Security, WS-Addressing, WS-Reliability, and WS-I Basic Profile. The abundance of documentation requires expert know-how to implement these. This is yet another reason to find expert guidance when starting a Web services-related project. Don't let the volume of standards scare you. This indicates that the SOA industry is putting muscle behind the bat. Best practices will help filter and prioritize.

#### 4.2.2.5 The RESTful Services Model

The main alternative to W3C's Web services is the so-called RESTful services. REpresentational State Transfer (the REST part) is a key design idea that embraces a stateless client/server architecture in which the Web services are implemented by named Uniform Resource Identifiers (URIs). Rather than using SOAP carrying WSDL messages, REST relies on the simple paradigm of create, read, update, delete (CRUD services) per URI. We can think of a URI, in this context, as a "Web server address" that implements CRUD operations. See [http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer). RESTful services are useful in SOA and pure client/server environments.

Some of the defining principles of RESTful services are

- Client/server style.
- Stateless. This feature allows for simple implementation. Different servers can handle initial and future service requests—large-scale enabled.
- Cacheable and layered. This feature permits gateways, proxies to intersect traffic without requiring protocol translation—large-scale enabled.
- A constrained set of content types and well-defined operations (CRUD). XML documents are frequently used as data structures.

RESTful services rely on HTTP—the ubiquitous protocol for accessing Web pages from a server. HTTP supports the four CRUD operations as follows:

HTTP Operation	CRUD Functions
POST	Create, Update, Delete
GET	Read
PUT	Update, Create
DELETE	Delete

Let's consider an example of a weather service. The URI processes weather requests. Each city name has a URI resource associated with it in the form: *http://weather.example/cityName*:

- **GET *http://weather.example/city***: Get the current weather at *city*. Simple XML document returned with weather data. Note that HTML data structures are *not* returned as with a typical Web page request.
- **POST *http://weather.example/city***: Set weather information for *city*. Simple XML document input.
- **DELETE *http://weather.example/city***: Delete a *city* from the system.
- **POST *http://weather.example/add/city***: Add a *city* to the system.

Note that the GET function does not modify the services' data structures. However, POST and DELETE modify, so these services need a special authentication step not described here. A garden variety Web server can implement the weather service.

The raw simplicity and worldwide scale of RESTful services have given them a huge boost in usage compared to SOAP/WSDL methods in non-enterprise applications. Amazon, Google, Microsoft, Yahoo!, and many others offer public RESTful services for all sorts of purposes. See <http://code.google.com/apis/maps> or <http://aws.amazon.com> for more examples.

It's easy to imagine a collection of media-related URIs for processing A/V data and metadata. For a representative example of video-related REST services, see Veoh's developer site ([www.veoh.com/restApiDoc/restReference.html](http://www.veoh.com/restApiDoc/restReference.html)).

So, Web services will be divided among SOAP/WSDL and RESTful methods although less common methods do exist. A decision to use one or the other will depend on many factors: scale, development tools, staff knowledge, existing infrastructure, public/private, standards, interop, reliability, cost, and more.

Next, let's consider item 4 in Figure 4.3.

## CLOUD COMPUTING AND SAAS

IBM president Thomas J. Watson (1952–1971) is known for his alleged 1943 statement: "I think there is a world market for maybe five computers." However, there is no evidence he ever said this. Assuming the quote is valid, Mr. Watson was in error by four computers; he should have said one. Why? Applications and services that commonly run on a desktop operating system are now running in the cloud: the unbounded, ever-changing, elusive collection of millions of servers that make up the Web.



Okay, it's not really one computer, but it appears to be so from the perspective of users of the network. Visit Google Maps or MySpace—most of Web 2.0—and you're using cloudware. For enterprise use, Software as a Service (SaaS) is a business model for leasing network available applications. See Appendix C for more information on "cloud-everything."

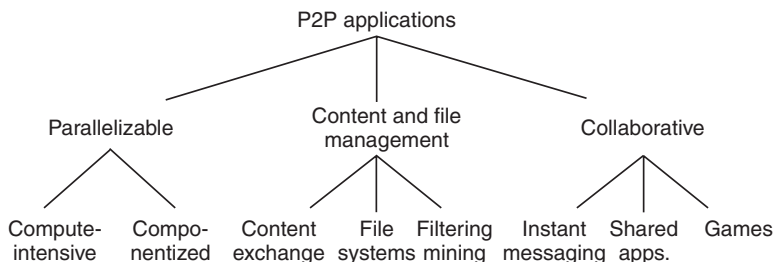
#### 4.2.2.6 Peer-to-Peer Computing

The term *P2P* refers to a class of systems and applications that use distributed resources to perform functions in a decentralized way. P2P systems may be classed as outlined in Figure 4.14 (Milojicic). Layer two of the diagram shows the three main application domains. These are further segmented into eight specific application areas. No doubt, P2P has become a household name with the likes of Napster, Kazaa, BitTorrent, Gnutella, and other file-sharing programs. However, we should not paint P2P as evil; it has plenty of legitimate applications. Let us look at the three main domains for P2P.

The first one is *parallel systems*. They split a large task into smaller pieces that execute in parallel over a number of independent peer nodes. The SETI@Home project has aggregated 1.8 million “client” computers to harness over 2 million years of equivalent computer time and counting. It is acknowledged by the Guinness World Records as the largest computation in history. True, this is an example of grid computing, but it uses P2P concepts. The second domain is that of *content and file management* and is mainly associated with file sharing, be it legal or illegal. Streaming, too, is supported by some applications. See Appendix I for a novel P2P streaming model. The third domain is P2P *collaboration*. Examples of this type are AOL and Yahoo!’s instant messenger (IM) and multiuser gaming (MUGs). Groove’s Virtual Office ([www.groove.net](http://www.groove.net)) is a P2P application suite for sharing files, workspace, and other information. There has been little professional A/V use of P2P to date, although some facilities have used it on internal, protected networks for file exchange.

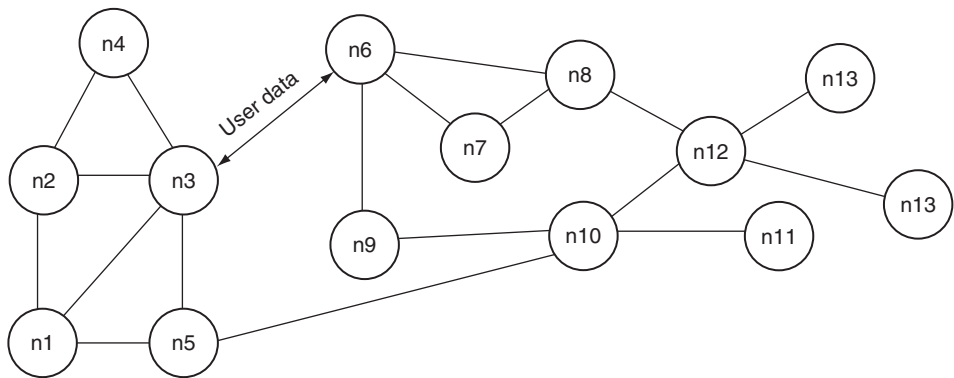
#### P2P Architectural Examples

It is not difficult to imagine the topology of a pure P2P network (see Figure 4.15 for an example). Each node is a computer with one or more network connections to other nodes. Nodes interconnect using 1:1 communication for the purpose of data exchange. However, because any given node can support more than one P2P conversation, the diagram shows a general 1:*N* connection space. In reality, the connections are networked and not hard-wired as the diagram may imply. There is no notion of a server or other hierarchy. This is true anarchy

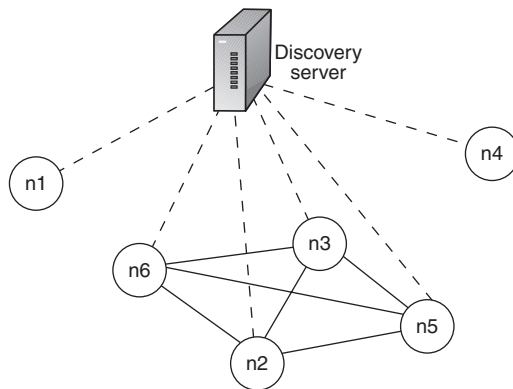


**FIGURE 4.14** P2P application classifications.

Source: HP Labs (Milojicic).



**FIGURE 4.15** Pure P2P architecture.



**FIGURE 4.16** Hybrid P2P architecture.  
Concept: Microsoft.

and one reason why it is virtually impossible to control and manage traffic between peers on the open Web. The Gnutella method relies on pure P2P, for example.

Aside from the pure P2P model, there is the hybrid form, as shown in Figure 4.16. Anarchy gives way to hierarchy with the introduction of a directory server. The original Napster system, among others, used the idea of a main server to provide a list of files (or other information) and locations to clients. Once the client locates a file, the server connection is broken and a true P2P relationship takes over. The server makes file location a snap but is a bottleneck in what is otherwise an infinitely scalable architecture. Of course, the directory server can be replicated, but this leads to other issues. This model is good for private networks, and the server adds management features. The Kazaa model uses the idea of a super-node to act as a server. Some nodes are dedicated to act as servers and to communicate to other super-nodes to create a single virtual server.

Performance of P2P networks is problematic. Because of its decentralized nature, performance is influenced by three types of resources: processing power and its availability, storage, and networking. In particular, low delivery rates can be a significant limiting factor. Data sourcing client disconnect is a major issue when transferring large files. On the plus side, P2P can scale to millions of users with a combined computing power of trillions of operations per second. In fact, Ellacoya Networks, a deep packet inspection product company, believes that ~40 percent of all Web traffic is P2P related.

### 4.2.3 Architectural Comparisons

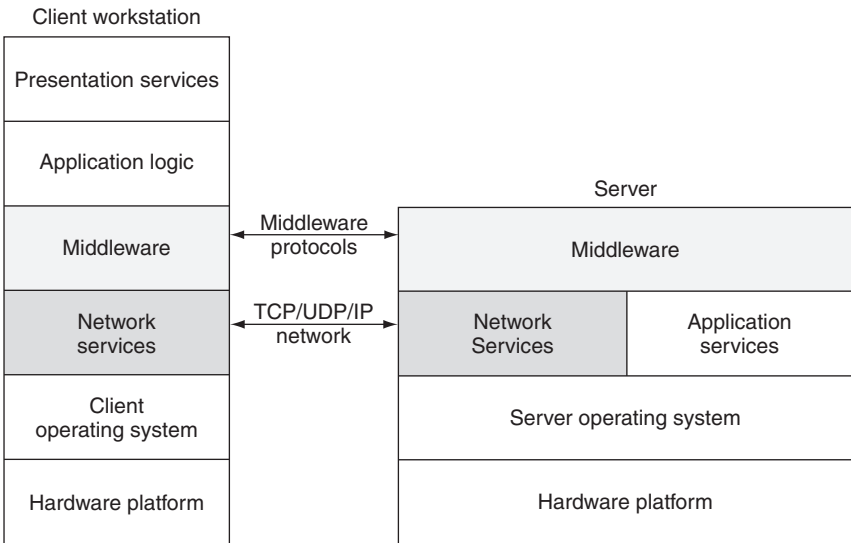
So, how does all this stack up? Which of the four architectural classes in Figure 4.3 has the edge? Well, the answer all depends on who the “user” is and what advantages and functionality are required. All four types find practical application use. Some key aspects of the four systems are as follows:

1. **Centralized.** Older, multiuser mainframe use is diminished greatly and defers to client/server methods. However, virtualization has given new life to this class. The single-user, ubiquitous standalone PC rules the client world. Most legacy A/V gear is standalone by design, such as VTRs, character generators, and NLEs. If connected to a network, the client becomes part of a C/S scenario in many cases. Dedicated devices such as A/V and IP routers are special cases of standalone systems.
2. **Client/Server.** It is the most powerful networked architecture in modern use. It is mature and offers advantages to users (inexpensive, accessible), application developers (tools galore, standards), and IT maintenance staff (management tools).
3. **Service-Oriented Architectures.** This space has been divided between two different Web service implementation frameworks and associated infrastructure methods. Its strengths are business process visibility, agility, scalability, aggregate reliability, manageability, and widely available development platforms and tools.
4. **Peer to Peer.** P2P is generally “out of control” in terms of IT’s management, QoS, and security needs. There are some exceptions to this, but for the most part, P2P is not popular in professional applications.

The distributed architectures (2, 3, and 4) rely on various forms of middleware to tie the pieces together. The following section reviews the essentials of middleware.

## 4.3 MIDDLEWARE CONNECTIVITY

Middleware is the layer of functionality that connects systems over networks for exchanging information between them or, put another way, is the “glue” that ties

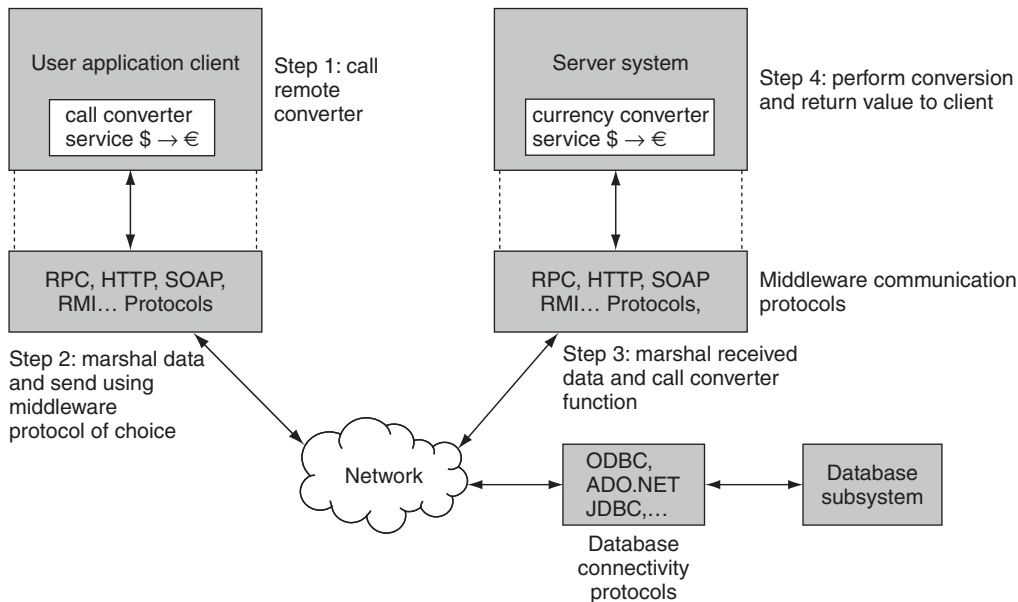


**FIGURE 4.17** *The role of middleware in a client/server transaction.*

various distributed software layers together. Most middleware technologies are platform independent, but some are vendor specific. Figure 4.17 demonstrates how middleware operates as a communication layer between client and server. In general, middleware is used in a variety of distributed computing architectures.

Figure 4.18 outlines a common scenario in which middleware ties the client(s) and server(s) together. For client/server interaction using middleware, the sequence of communication events is outlined in Figure 4.18 and later. The example illustrates a client calling a remote currency converter service. The sequence of events is as follows:

1. Client application formats user data structures and calls the currency converter service on the remote server.
2. Client middleware *marshals* the data into the select middleware format (RPC, RMI, HTTP/SOAP, etc.) and sends it over the network to the remote server. Importantly, the middleware data structures are independent of programming language (and OS) choices, which allows for true heterogeneous client/server communications.
3. The remote server receives the data structures and formats them for the target function (currency converter) in the local programming language.
4. The server application formats a response, marshals the data structures, and returns them to the client over the network. At this point a complete middleware transaction has occurred. In effect, steps 3, 2, 1 are repeated in reverse order to return the response to the client.



**FIGURE 4.18** *Process connectivity using middleware.*

Ideally, a remote service across the network behaves as a locally called function. However, due to network delay and other network anomalies, this may not be so. There is generally no QoS defined for these protocols, so moving real-time A/V with them is problematic. They are best used for non-real-time operations: query, simple control, user interface, database access, application services, and “just in time” services such as format converters, transcoders, and so on. With sufficient care and time-aware services, frame accurate control of A/V I/O is possible.

The following middleware protocol standards are in widespread use:

- **Remote Procedure Call (RPC)**. RPCs provide the means for one program to call the services (a procedure or other program) provided by a remote machine. The response is usually some data elements or status report. The RPC is the oldest protocol in this class. Millions of systems use the RPC library. See RFC 1831/1833 for more information.
- **HyperText Transfer Protocol (HTTP)**. This is the primary communication protocol for connecting browser-based clients to servers over the Web. In a secondary sense, it is also used as a generic connector between a client, not necessarily browser based, and a server.
- **Simple Object Access Protocol (SOAP)**. This is a basic means to move XML messaging between programs connected over a network. SOAP embeds XML data structures, and the package is transported using



HTTP. Typically, a SOAP message (the request) is sent to a remote receiver, and it counters with another SOAP message (the response). XML is a good means for exchanging business-related information. See [www.w3.org](http://www.w3.org) for more information on SOAP. The combination of SOAP/XML is the backbone of WSDL-based Web services.

- **Remote Method Invocation (RMI).** RMI is the basis of distributed object computing in a Java environment. It provides the tools necessary for one Java-based component to communicate with another Java component across a network. Think of this as a RPC for Java.
- **.NET Remoting.** This is a Microsoft concept for communicating between software objects in the .NET environment. .NET Remoting supports HTTP and SOAP (and other protocols) to transfer XML data structures between clients and servers, for example. This is roughly analogous to the RMI as used in a Java environment.

For SOA environments, the Enterprise Service Bus (ESB) is a special class of optional middleware. An ESB is software that sits between the services and enables reliable communication among them. See [www.sonicsoftware.com](http://www.sonicsoftware.com) for more insights on the ESB.

### 4.3.1 Database Connectivity Protocols

Another aspect of middleware is database connectivity (see Figure 4.18). What does this mean? For our discussion, this relates to heterogeneous clients or servers connecting to heterogeneous databases. In the context of media, a database may store metadata that describe A/V assets. Typical descriptors are title, length, owner/rights, trim points, format, descriptive key words, and so on. Edit stations, ingest control stations, browsers, traffic/programming, and device automation are among the types of clients that require database access.

The granddaddy of the database query is SQL. This is a language for accessing databases and doing adds/deletes and queries of the contents. By itself, SQL does not define how to connect to a database. Over the years several mature protocols have evolved to connect clients to databases. Highlighting the most significant, the following are in daily use:

- **ODBC (Open Database Connectivity).** This is an open standard defining how clients can connect to vendor-neutral databases. Microsoft defined it, but now it is an international standard and is used universally. Under the hood, ODBC uses SQL for operations on the database. Also, clients do not need to know where the database is located on a network or what brand of database is accessed. Of course, ODBC allows users to access vendor-specific features if needed.
- **ADO.NET (ActiveX Data Objects).** This is the cornerstone of Microsoft's .NET framework for database access. This is an ODBC-compliant API

that exposes the features of modern databases to client applications. It was designed for the .NET programming environment.

- **JDBC (Java Database Connectivity).** This provides database access to programs written in Java and JavaScript. It uses ODBC as the core of connectivity.

These database access methods are in wide use. For more information, see <http://java.sun.com/products/jdbc> for JDBC and [www.microsoft.com](http://www.microsoft.com) for ADO.NET particulars. In the bigger picture, middleware is an adhesive to create a variety of architectures made of heterogeneous, programming language-neutral components.

## 4.4 IMPLEMENTATION FRAMEWORKS

This section reviews the programming frameworks and major languages used for implementing the architectures discussed in the previous sections. In particular, the focus is on client/server, SOA, and Web services models. The frameworks and platforms in common use are as follows:

- Microsoft's .NET framework
- Sun Microsystems' Java Enterprise Edition—Java EE 5 framework

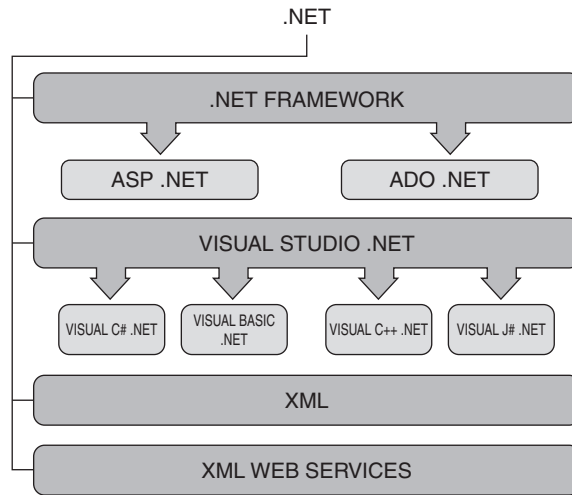
.NET is Microsoft's premier development platform and Web services software architecture. Sun Microsystems developed the Java EE for a similar purpose. Both of these are competing methods for developing client/server and SOA systems.

### 4.4.1 The .NET Framework

The .NET framework is an integral Windows component for building and running software applications. Figure 4.19 shows the main components in the .NET toolbox. The top level relates to Web server technology. Web servers are built with .NET's Active Server Pages (ASP) and ActiveX Data Objects (ADO). The middle layer is the Visual Studio program development environment. This enables programmers to design, write, and debug software applications. The bottom layers support the W3C model for Web services. These three main divisions are mutually independent aspects of .NET.

Key aspects of the .NET framework are:

- Supports over 20 different programming languages, mainly on X86 Intel/AMD CPUs. Visual C++, Visual Basic, and Visual C# are the most common.
- Works hand in hand with Microsoft's operating systems.
- Manages much of the plumbing involved in developing software, enabling developers to focus on the core business logic code.



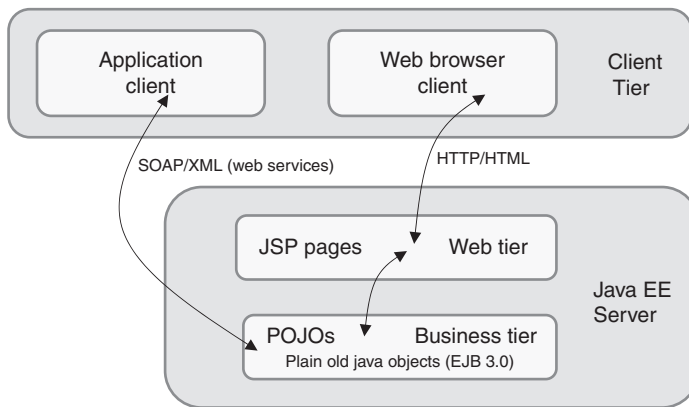
**FIGURE 4.19** *The .NET working environment.*  
Concept: Microsoft

.NET has found a home with standalone as well as distributed systems. In fact, all the architectural classes in Figure 4.3 can be implemented using .NET's tools, components, and connectivity. Many standalone A/V software applications use the .NET framework and development tools for program creation. Its tight integration with the Windows OS makes it the preferred framework for many software projects.

#### 4.4.2 The Java EE Framework

The Java Enterprise Edition is a programming framework for developing and running distributed multitier applications. Defined by Sun Microsystems, it is a Java-based, modular service-oriented architecture methodology. Java EE is defined by method specifications and corresponding interface definitions for creating individual modules for an enterprise-level computing architecture. It supports Web server functionality and true W3C-based Web services connectivity.

In many ways, Java EE is similar to .NET: it supports distributed services, uses professional development platforms, and uses various middleware protocols for client/server communications. In fact, the .NET functionality stack in Figure 4.19 has equivalent layers in the JAVA EE world. One big difference is this: JAVA EE supports only the Java programming language. However, while .NET supporters brag about its multiple language support, Java supporters brag about its multiple OS support, including the popular open source Linux. In principle, .NET is *write many* (language choices) *run once* (only on CPUs with a Microsoft OS), whereas Java is *write once* (Java only) *run many* (choice of CPU and OS). The two camps have taken up arms over the virtues of each platform.



**FIGURE 4.20** A simplified Java EE environment.

Figure 4.20 shows a simplified view of a hybrid JAVA EE environment: one browser based and the other Web services based. Browser-based clients (top right) interact directly with Java-based Web servers with Java Server Pages. This is a traditional client/server relationship in a Web environment. Non-browser-based clients (top left) interact with the Web services using W3C-defined or RESTful methods typically. The client executes local application code, which in turn calls the JAVA EE platform for services.

Why implement a non-browser-based application client? Client-based application programs (written in the C++ or Java language, for example) provide a richer user interface than what is available using HTML with a browser. Browsers offer limited UI functionality for security reasons, e.g., no drag and drop for the most part although there are exceptions. There are trade-offs galore between these two methods, and each has found application spaces.

Many vendors offer JAVA EE-compliant programming environments: BEA's WebLogic, IBM's WebSphere, JBoss (open source), and Sun's Application Server Platform, to name a few.

## NET AND JAVA EE INTEROPERABILITY: DREAM OR REALITY?



Integrating .NET and JAVA EE's Web services is feasible. However, there are pitfalls to cross-platform integration.

- Standards are sometimes interpreted differently by the two platforms, although the conflicts are usually minor.
- Web service functionality is a common subset of both platforms. However, other middleware and

messaging aspects do not have defined levels of interoperability. In this case, vendors supply bridges to cross the domains.

- Few developers are comfortable working in both environments.

### 4.4.3 The Burden to Choose

With so many frameworks and platforms, you might ask, “Which is the preferred one?” Well, the answer depends on many factors, some of which are application QoS performance, target HW, programmer experience/productivity, legacy IT infrastructure, choice of OS, cost of HW/SW, interoperability requirements, complexity, and available development tools. Any choice should factor in these considerations and more. The ideal platform should map onto your overall requirements list. Both of these programming paradigms exist in the much larger context of virtualized servers, networks, and storage. The next section will consider these leading-edge technologies.

## 4.5 METHODS OF VIRTUALIZATION

We are familiar with the world of physical clients, networking, servers, and storage. Given any configuration of user applications (SW and HW) and client loading, chances are some of the resources are underutilized. This implies wasted energy (not green) and inefficiently used compute-and-storage capacity. Is there a strategy to improve these metrics? Fortunately, there is: virtualization.

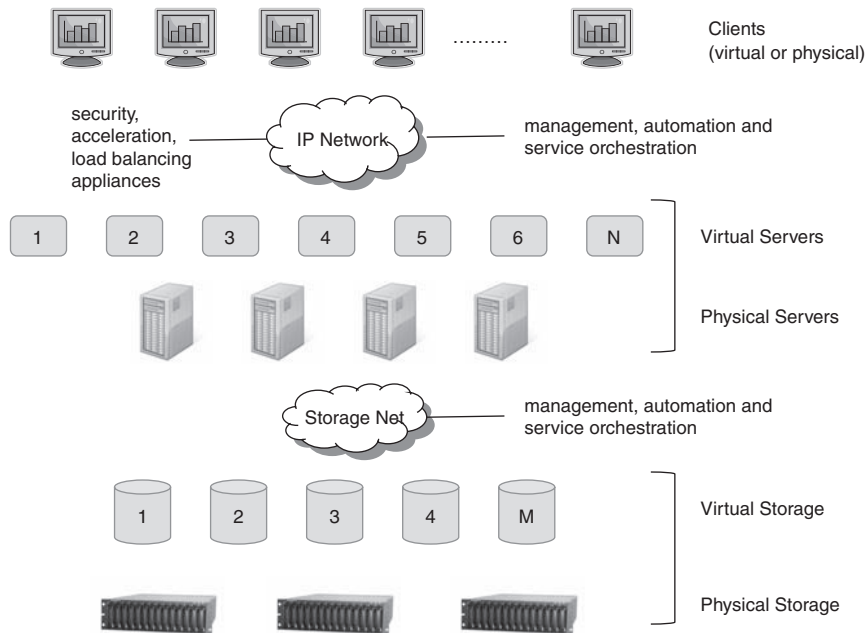
Virtualization is the creation of a virtual (rather than physical) version of something, such as an operating system, a storage partition, or a network resource. Virtualization hides the physical characteristics of a resource from its users. So, a server or storage device can appear as multiple virtual resources. Figure 4.21 shows layers of physical resources and the virtual versions that use these physical devices.

Some positive contributions provided by virtualization follow.<sup>5</sup>

- Virtualization provides for improved overall device utilization. The typical increase expected (servers) is 15 to 80 percent.
- Fewer physical devices are required (servers, storage, etc.). Virtualization enables you to reduce hardware and operating costs by as much as 50 percent.
- Floor space utilization efficiency is improved in the data center.
- It is greener. All electricity consumed by servers generates heat; heat is extracted with cooling that takes ~125 percent more electricity.
- Virtualization enables you to isolate applications within virtual servers. This improves QoS and update testing.
- It reduces the time to provision a new server by 70 percent.

---

<sup>5</sup> See various white papers at [www.vmware.com/solutions](http://www.vmware.com/solutions), <http://xen.org>, and [www.microsoft.com/virtualization](http://www.microsoft.com/virtualization).



**FIGURE 4.21** *The virtual data center.*

- Virtualization completely changes how the IT data center is configured and provisioned. The more automation, the more efficient the operations.
- Virtualization enables you to deploy multiple operating system technologies on a single hardware platform; you can run Microsoft Vista on top of Linux, for example.

The advantages are compelling, and virtualization is becoming the “must-have” technology for the data center. IDC forecasts that, in 2010, 1.7 million physical servers will be shipped to run virtual machines. To appreciate what virtualization can accomplish, experiment with VMware’s TCO/ROI calculator at [www.vmware.com/products/vi/calculator.html](http://www.vmware.com/products/vi/calculator.html).

The complete virtualized data center of Figure 4.21 is not often fully implemented. Virtual servers and associated management systems (automation) are the most mature elements in the picture. What about networking? Case in point: Cisco’s VFrame product provisions data center resources by integrating orchestration and control intelligence into the infrastructure that interconnects resources together. Storage virtualization is available using a SAN, for example, but more functionality is required to mesh with other virtual elements. See the Snapshot “Virtual Arrays Within an Array.”

For many real-time A/V applications, using virtualized elements is problematic. Why? Real-time performance (I/O, codecs, editing, etc.) is not always

guaranteed by the major providers of virtualization solutions. Time-insensitive applications (MAM, search, management, converters, and so on) are a sweet spot for virtualized servers, networking, and storage.

Detailed coverage of virtualized networking and storage is beyond the scope of this book. However, a short overview of server virtualization technology is of value.



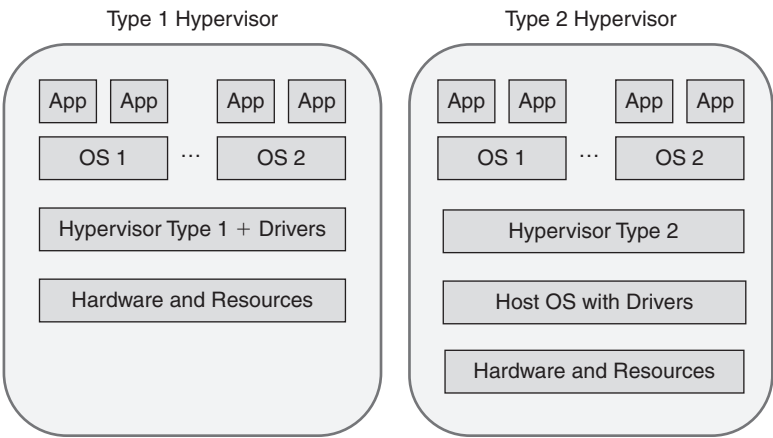
**VIRTUAL ARRAYS WITHIN AN ARRAY**

In some cases, dividing up SAN storage among separate clients/servers running applications offers subpar performance. Why? One storage attached server may hog the resources such that other connected devices receive a degraded storage QoS. Administrators sometimes solve this problem using a DAS attach per server or application. This yields an optimum storage QoS per server/application. Or, they divide up a SAN into physical domains yielding a carved-out equivalent DAS. Neither of these is ideal, since a DAS is always more management intensive and less flexible than a SAN.

One approach to this problem is to define one or more virtual arrays (VAs) or virtual domains within a storage system. Each VA offers application isolation and a QoS contract per application or server. So a SAN could be divided into multiple VAs, each with its own defined operational characteristics. Technically, creating a VA is not trivial while still maintaining all the goodness of a SAN. You might imagine a demanding video application sharing a SAN with, say, a business application with no resource conflicts. Companies such as 3PAR offer SAN storage with managed virtual domains.

**4.5.1 Under the Hood: Servers**

A virtual server (VS) is built with software and uses the physical resources of CPU, memory, and I/O to provide independent operating environments. Figure 4.22 shows the two main types of virtual server design. Both enable the



**FIGURE 4.22** Client/server virtualization methods.

equivalent of  $N$  virtual servers per hardware platform.  $N$  can be large, so systems with 20+ virtual servers are possible. The sweet spot has settled in around 5–7 per hardware platform. Applications don't know they are running in a virtual environment for the most part, although exceptions exist. Sharing I/O is a challenge, and this is one reason for two types of architecture.

The key technology in server virtualization is the hypervisor. This is a thin layer of software that runs directly on the hardware (type 1) or a host OS (type 2). This software layer needs to be lean; otherwise, it will adversely affect application performance. It consumes precious CPU cycles interpreting each application layer OS call. Hypervisors are resource managers and enable many upper-layer operating systems to run simultaneously on the same physical server hardware. Some important features of a virtual server are as follows:

- $N$  different user-level OSs (Windows, Mac OS, Linux, others) can run independently on the same hardware. Each of these OSs can support one or more user applications.
- The hypervisor is supported on X86 and Power PC architectures most commonly.
- A VS can be created or decommissioned at will. As user workloads change, IT staff (or automation) can configure the VS environment as needed. This is key to improving HW utilization and reducing the number of physical servers in a data center.

Type 1 hypervisors (a native virtual machine) provide higher performance efficiency, availability, and security than type 2 (a hosted virtual machine). Note that this method has fewer layers of software, and the hypervisor layer must provide the device drivers. Type 2 can be used on client systems with an existing OS, such as running Windows on a Mac OS. See [www.parallels.com](http://www.parallels.com) and Microsoft Desktop Virtualization for examples. Type 2 is also an advantage when a broad range of I/O devices is needed, since the host OS often has excellent driver support.

IBM, Microsoft, VMware, and Xen.org (Citrix) are among the leaders of server and client virtualization. See, too, [http://en.wikipedia.org/wiki/Comparison\\_of\\_virtual\\_machines](http://en.wikipedia.org/wiki/Comparison_of_virtual_machines) for a deep dive comparison of solutions.

What will the affect of virtualization be on the data center? A 2008 press release by Gartner stated, "Virtualization will be the highest-impact trend changing infrastructure and operations through 2012. Virtualization will transform how IT is managed, what is bought, how it is deployed, how companies plan and how they are charged." So, the media enterprise will be dragged into this future with the possible exception of some high-performance A/V appliances—video I/O, real-time graphics engines, editors, and so on. In the final



analysis, the virtualized media enterprise will offer operational efficiencies and cost savings of lasting value.

## 4.6 OPEN SOURCE SOFTWARE

The basic idea behind open source programs is simple. When masses of motivated freelance programmers are enabled to improve and modify code and then redistribute their changes, the software evolves. If the number of programmers is large, then the speed of software evolution is astonishing (Raymond). Open source provides free access to program source code. Licensing is required for most open source code, but the restrictions are not onerous. Changes are well controlled, tested, and folded into the next revision. The biggest open source project space may be found at SourceForge (<http://sourceforge.net>) with ~103 K registered projects. Many of these are developed by professional programmers, and the quality is excellent. Scan the SourceForge site for A/V tools and applications to see the variety of available software.

Gartner predicts that, by 2011, 80 percent of commercial software will contain open source code. Selecting an open source application depends on several factors. One, does it meet your needs? Seems obvious, but what this means is you should not select based on a staff zealot who hates vendor *X* in preference to open source. Two, is the distribution mature, is it sell supported, and does it meet your risk/reward ratio? Three, is your application mission critical? Open source can meet this need, but you need to be rigorous in testing before selecting.

Among open source code, LAMP is a set of programs commonly used together to run Web sites. LAMP is an acronym for Linux, the operating system, Apache, the Web server, MySQL, the database system, and the PHP server-side scripting language. Apache is used by ~65 percent of Web sites worldwide by some estimates. MySQL ([www.mysql.com](http://www.mysql.com)) is the most used open source database.

On the development front, Eclipse ([www.eclipse.org](http://www.eclipse.org)) is a popular Java/JAVA EE integrated development environment. JXTA ([www.jxta.org](http://www.jxta.org)) is a set of protocols for building P2P applications. Interestingly, JXTA is short for juxtapose, as in side by side. It is a recognition that peer to peer is juxtaposed to the client/server model. Of general interest is the OpenOffice Suite ([www.openoffice.org](http://www.openoffice.org)), which provides desktop productivity tools.

Expect to see more open source programming included with A/V vendor solutions. At present there are no “killer app” open source programs for the professional A/V space. If a critical mass develops, one day we may see something like MyTraffic, MyAutomation, or even MyVideoServer as an open source program. One interesting Linux-based, open source NLE application is Cinelerra (<http://cinelerra.org>).

## 4.7 HIGH-PERFORMANCE REAL-TIME SYSTEMS

For many A/V applications, real-time performance is the paramount feature. There are several ways to achieve this, and this section outlines the main aspects of RT systems for A/V. While it is true that not all C/S implementations are suitable for RT A/V applications, some are a perfect match. Some of the important themes are the RT OS, multimedia programming extensions, GPU acceleration, and 64-bit CPUs. Let us consider each of these next.

### 4.7.1 Achieving Real-Time OS Performance

The most common RT system implementation uses the standalone architecture with a dedicated OS. General-purpose, Windows-based (XP, Vista, Server 2008) products can achieve excellent RT performance, despite the bad rap they sometimes get for long-term reliability in normal use. This does not anoint the Win/OS as a real-time operating system (RTOS). Instead, for some selected applications, the OS meets the performance needs for A/V. For years vendors have built mission-critical, RT A/V applications with the Win/OS. What is the trick?

For one, you should run only the target A/V application—all others are *persona non grata*. Running unessential background applications (spyware, calendars, instant messaging, etc.) is a recipe for poor A/V client performance. In general, the more insular the application, the better its performance. Another trick is to set the OS priority of the target application for maximum utilization. Fine-tuning caching and networking also improves performance. With these precautions, the Win/OS (Linux and the Mac OS, too) supports A/V applications with excellent performance, long-term reliability, and delivery quality. An alternative is to base the application platform on a true RTOS, such as Wind River's VxWorks, LynxOS, Real-Time Linux, and Real-Time Specification for Java (RTSJ). An RTOS guarantees excellent OS response times for A/V applications. The RTOS environment runs as an embedded system and does not offer general-purpose computer resources. Embedded RTOS systems are single-minded and perform exceptionally well under the stress of RT workloads.

Consequently, some A/V product vendors have chosen the RTOS approach. For example, Chyron's Duet Graphics platform uses VxWorks. The choice of RTOS versus non-RTOS depends on many factors, and there are trade-offs for each camp.

### 4.7.2 Multimedia Extensions and Graphics Processors

Digital signal processors (DSPs) have long been a mainstay for compute-intensive operations. Just a few years ago, real-time A/V processing used DSP chips or dedicated hardware. Thanks to Moore's law, the momentum has shifted from the dedicated DSP to the general CPU for standard definition video and some

HD processing. Although DSP processors are still in demand for some applications, Intel/AMD processors, with DSP-like multimedia extensions (MMX), can perform RT A/V operations without resorting to special hardware. The PowerPC has also been optimized for multimedia operations with its Velocity Engine.

Some vendors are using software only for real-time SD MPEG encoding/decoding and HD decoding. For HD encoding (especially H.264), hardware acceleration is often needed. Today it is possible to build a four-channel video server with every A/V operation performed in software, including all SD/HD decoding and encoding using only one quad core CPU.

Some vendors are using high-performance video/graphics cards (designed for PCs) for video processing, 2D/3D imaging, and font shading. The key ingredient here is the embedded graphics processing unit (GPU). GPUs are being used as graphics accelerators for CPU-sited algorithms. The combination of CPU software algorithms plus GPU graphics acceleration provides amazing RT video processing power.

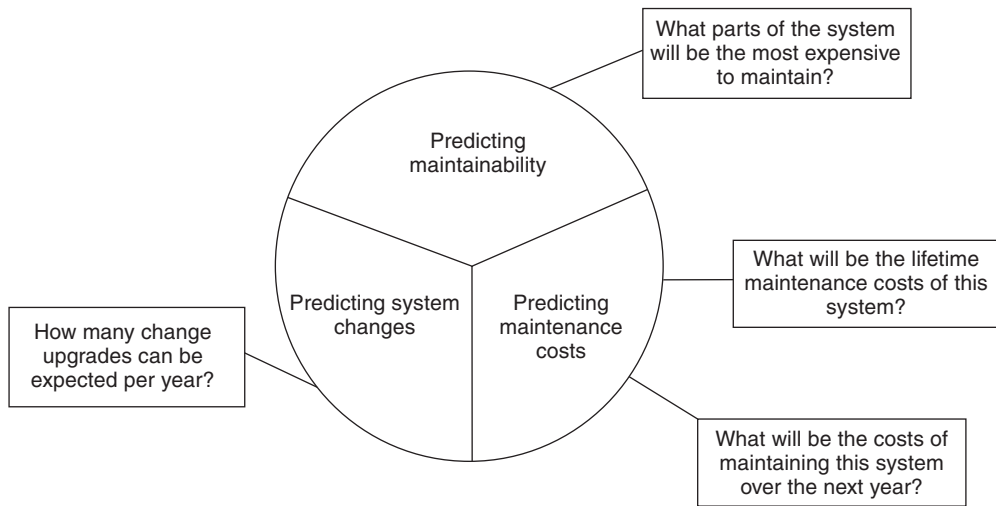
### 4.7.3 64-Bit Architectures and Beyond

The jump from 32- to 64-bit processing represents an exponential advance in computing power, not just a factor of two. With 32-bit registers, a processor has a dynamic range of  $2^{32}$ , or 4.3 billion. With 64-bit registers, the dynamic range leaps to  $2^{64}$ , or 18 billion billion. Compute speed and memory addressing range are improved. Many popular CPUs offer 64-bit modes. Microsoft has a 64-bit version of XP/Vista and Windows Server. These are joining the mature UNIX/64 and Linux/64 choices. Few A/V applications have been written to take advantage of 64-bit computing, but this will change as 64-bit computing becomes more mature. Porting a native 32-bit application to take advantage of 64 bits is a painful experience, so most vendors will not do it. However, new applications may well be written for the 64-bit mode.

Another way to improve compute performance is to use multiple processors. If  $N$  CPUs are ganged together, the overall compute power increases. Appendix C outlines grid and cluster computing concepts.

## 4.8 SOFTWARE MAINTENANCE AND SYSTEM EVOLUTION

It is inevitable that software-based applications and systems will need bug fixes and upgrades. The larger the application, the more likely it will take on a life of its own. Indeed, software needs maintenance just as hardware does. Figure 4.23 outlines several questions you should ask when purchasing vendor software. Do not underestimate the effort to keep the software running smoothly. Also always ask about hot upgrades. Large, mission-critical systems have many



**FIGURE 4.23** *Software maintenance factors.*  
 Source: *Software Engineering 7* (Sommerville).

**Table 4.2** Lehman's Laws of Software Evolution

Law	Description
1. Continual change	Programs become progressively less useful over time—they age.
2. Increasing complexity	Programs become more complex over time.
3. Large program evolution	Program size, time between releases, and number of reported errors are approximately invariant for each system.
4. Organizational stability	Over a program's lifetime, its rate of development is approximately constant and independent of the resources devoted to it.
5. Conservation of familiarity	Over the lifetime of a system, the incremental change in each release is approximately constant.
6. Continuing growth	The functionality has to increase continually to maintain user satisfaction.
7. Declining quality	The quality of a system will appear to be declining, unless it is adapted to the changes in its operational environment.

elements; make sure that any one element can be upgraded—while the system is running—without affecting the operation of the other pieces.

#### 4.8.1 Lehman's Laws

In 1972 Lehman and Belady (Lehman) published their laws of software evolution. These are reprinted in Table 4.2. They studied the history of successive releases of large programs. They found that the total number of modules increases linearly with release number but that the number of affected modules increases exponentially with release number. What does this mean? Repairs

and enhancements tend to destroy the structure of a program and to increase the entropy and disorder of the system. More time is spent repairing introduced flaws and less time on developing truly useful new features.

These laws are especially useful for program developers but also for end users. For example, law #6 adds a sense of realism to a product's functionality. Users will be disappointed if the vendor does not regularly add features to the product. Law #7 indicates that as a facility changes, the unmodified products will appear less capable unless they are updated to fit into the new environment. So, when you are selecting a product, it is wise to check with the vendor for the planned upgrade schedule. The larger the software effort behind a product, the more likely that Lehman's laws will apply. Note that these laws apply to large programs and do not necessarily apply in exactly the same way for small- or medium-size programs. Nonetheless, the principles have some applicability to most programming projects.

## 4.9 IT'S A WRAP—A FEW FINAL WORDS

A/V performance is tied to software performance—and increasingly so. While it is true that many contemporary A/V products use the standalone architectural model, expect to see distributed systems, especially Web services, applied to A/V systems. Non-RT designs have relaxed QoS levels and are easier to build than RT designs. However, RT distributed systems will become part of the A/V landscape as developers become more confident and experienced with service-oriented architectures.

Software evaluation, selection, configuration, and maintenance are key to a smoothly run media organization. An educated A/V staff will be an agile staff. The future of A/V systems' performance, scalability, reliability, manageability, and flexibility lies in software and networked systems. Keep your saw sharp in these areas.

## REFERENCES

- Barry, D. *Web Services and Service-Oriented Architectures: The Savvy Manager's Guide*: Morgan Kaufmann, 2003.
- Cummins, F. *Enterprise Integration* Chapter 10: Wiley, 2002.
- McCarthy, S. *ENIAC: The Triumphs and Tragedies of the World's First Computer*: Walker & Company, 1999.
- Erl, T. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*: Prentice Hall, 2004.
- Graham, S., et al. *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI* (2nd edition): Sams, 2004.
- Footen, J., & Faust, J. *The Service-Oriented Media Enterprise* Chapter 3: Focal Press, 2008.

- Lehman, M.M. & Belady, L.A. *An Introduction to Program Growth Dynamics in Statistical Computer Performance Evaluation*. W. Freiburger (ed.), Academic Press, New York, 1972, pp. 503–511.
- Milojicic, D., et al. *Peer-to-Peer Computing*, HP Labs Research Report, [www.hpl.hp.com/techreports/2002/HPL-2002-57R1.html](http://www.hpl.hp.com/techreports/2002/HPL-2002-57R1.html), March 8, 2002.
- Orfali, R., et al. *Essential Client/Server Survival Guide*: Wiley Press, 1994.
- Raymond, E. *The Cathedral and the Bazaar*: O'Reilly, 2001.
- TV Technology Magazine*, May 4, 2005, page 1.

This page intentionally left blank

# Reliability and Scalability Methods

## CONTENTS

<b>5.0</b>	<b>Introduction to High-Availability Systems</b>	<b>198</b>
5.0.1	Reliability Metrics	198
5.0.2	Detection and Repair of Faults	200
5.0.3	Failure Mechanisms	201
<b>5.1</b>	<b>HDD Reliability Metrics</b>	<b>205</b>
5.1.1	Failure Rate Analysis Domains	205
<b>5.2</b>	<b>Methods for High-Availability Design</b>	<b>208</b>
5.2.1	RAID Arrays	208
5.2.2	RAID Level Definitions	212
5.2.3	RAID Clusters	216
<b>5.3</b>	<b>Architectural Concepts for HA</b>	<b>219</b>
5.3.1	Single Point of Failure	220
5.3.2	No Single Point of Failure	220
5.3.3	$N + 1$ Sparing	221
5.3.4	Networking with High Availability	222
5.3.5	Mirroring Methods	223
5.3.6	Replication of Storage	225
5.3.7	Client Caching Buys Time	225
5.3.8	Other Topologies for HA	227
5.3.9	Concealment Methods	227
5.3.10	A Few Caveats	227
<b>5.4</b>	<b>Scaling and Upgrading System Components</b>	<b>228</b>
5.4.1	Upgrading Components	229
<b>5.5</b>	<b>It's a Wrap—Some Final Words</b>	<b>230</b>
	<b>References</b>	<b>230</b>



## 5.0 INTRODUCTION TO HIGH-AVAILABILITY SYSTEMS

Things do not always go from bad to worse, but you cannot count on it. When money is on the line, system uptime is vital. A media services company charges clients \$1K/day for editing and compositing services. A broadcaster is contracted to air the Super Bowl or World Cup finals along with \$100M worth of commercials. A home shopping channel sells \$10K worth of products per hour from on-air promotions. It is very easy to equate lost revenue to reliable equipment operations. How much time and money can you afford to lose in the event of downtime? Reliability is a business decision and must be funded on this basis. Therefore, it is straightforward to calculate the degree of reliability (and money to realize it) needed to secure continuous operations. For this reason, A/V facility operators often demand nearly 100 percent uptime for their equipment.

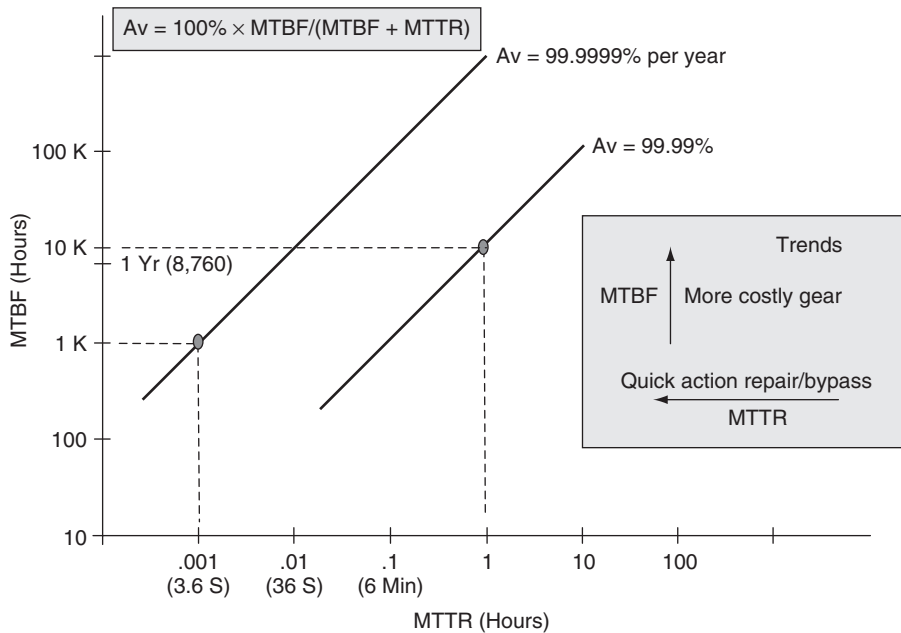
Of course, getting exactly 100 percent guaranteed uptime is impossible no matter how meticulous the operations and regardless of what backup equipment is available. Getting close to 100 percent is another matter. The classic “six sigma” measure is equivalent to 3.4 parts per million or an uptime of 99.9997 percent. In 1 year (8,760 hr), this equates to about 1.5 min of downtime. Adding another 9 results in 9.5 s of total downtime per year. In fact, some vendors estimate that adding a 9 may cost as much as a factor of 10 in equipment costs at the 99.999 percent level for some systems. Adding nines is costly for sure. This chapter investigates the techniques commonly used to achieve the nines required to support high-availability A/V systems. Also, it examines methods to scale systems in data rate, storage, and nodes.

### 5.0.1 Reliability Metrics

The well-known mean or average time between failures (MTBF) and mean time to repair (MTTR) are the most commonly used metrics for predicting up- and downtime of equipment. MTBF is not the same as equipment lifetime. A video server may have a useful life of 7 years, yet its MTBF may be much less, just as with a car, boat, or other product. An even more important metric is system *availability* (Av). How do MTBF and MTTR relate to availability?

Let us consider an example. If the MTBF (uptime) for a video server is 10,000 hr and the MTTR (downtime) is 1 hr, then the server is not usable for 1 hr every 1.15 y *on average*. (Because  $Av = \text{uptime} / (\text{uptime} + \text{downtime}) \times 100$  percent, then  $Av = 99.99$  percent availability for this example.) As MTTR increases, the availability is reduced. However, if the MTBF is small (1,000 hr) and if the MTTR is also small (3.6 s, auto failover methods), then Av may be excellent—99.9999 percent for this case. MTTR is an important metric when computing Av and for achieving highly available systems.

Figure 5.1 illustrates how availability is related to MTBF and MTTR. There are two significant trends. One is as MTBF is raised, the cost of equipment/systems usually raises too. Makes sense: more reliable gear requires better



**FIGURE 5.1** System device availability versus MTBF and MTTR.

components, design, and testing. However, excellent Av can be maintained even with inexpensive gear if the MTTR is reduced correspondingly. A design can trade off equipment cost against quick repair/reroute techniques and still achieve the desired Av. It is wise to keep this in mind when selecting equipment and system design styles. Even inexpensive gear has the potential of offering excellent Av if it can be replaced or bypassed quickly. Incidentally, for most TV channels, a MTTR of less than  $\sim 10$  s for the main broadcast signal is crucial to prevent viewers from switching away.

Sometimes the term *fault-tolerant system* is used to describe high availability. (Well, nothing is completely fault tolerant with an Av—100 percent.) In most so-called fault-tolerant systems, one (or more) component failure can occur without affecting normal operations. However, whenever a device fails, the system is vulnerable to complete failure if additional components fail during the MTTR time.

One example of this is the braking system in a car. Many cars have a dual-braking, fault-tolerant system such that either hydraulic circuit can suffer a complete breakdown and the passengers are in no danger. Now, if the faulty circuit is not repaired in a timely manner (MTTR) and the second braking system fails, then the occupants are immediately in jeopardy.

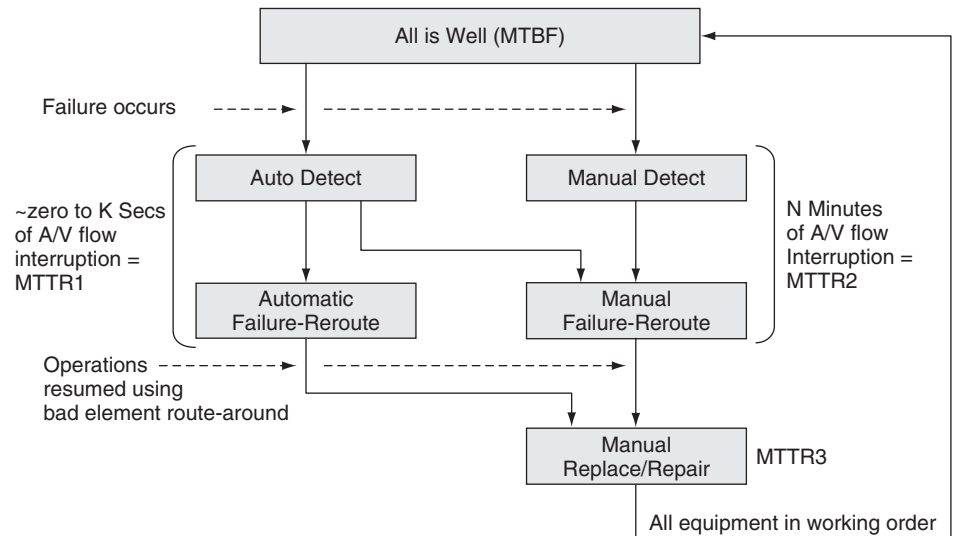
This analogy may be applied to an A/V system that offers “fault tolerance.” In practice, sometimes a single system component fails, but no one replaces the bad unit. Due to poor failure reporting alarms or inadequate staff training, some single failures go unnoticed until a *second* component fails months or

even years later—“Hey, why are we off the air?” Next, someone is called into the front office to explain why the fault-tolerant system failed.

### 5.0.2 Detection and Repair of Faults

Figure 5.2 shows a typical fault diagnosis and repair flow for a system with many components. There are two independent flows in the diagram: automated and manually based. Of course, all systems support manual repair, but some also support automatic self-healing. Let us focus on the automated path first on the left side of Figure 5.2. Automatic detection of a faulty component/path triggers the repair process. Detection may include HDDs, servers, switch ports, A/V links, and so on. Once the detection is made, then either self-healing occurs or a manual repair is required. Ideally, the system is self-healing and the faulty component is bypassed (implying alternate paths) or repaired in some automatic yet temporary way. The detection and repair should be transparent to the user community. In a traditional non-mission-critical IT environment, self-healing may take seconds to a minute(s) with few user complaints.

For many A/V applications, self-healing needs to be instantaneous or at least appear that way. Ideally, under a single component failure, no A/V flow glitching occurs. Quick failover is an art and not easy to design or test for. If done well, automated detection and healing occur in “zero” seconds (MTTR1 in Figure 5.2). In reality, most self-healing A/V systems can recover in a matter of a second or so. With proper A/V buffering along the way, the failure has no visual or user interface impact.



**FIGURE 5.2** Failure detection and repair process flow in an A/V system.

A no single point of failure (NSPOF) design allows for *any one* component to fail without operations interruption. Very few systems allow for multiple simultaneous failures without at least some performance hit. The economic cost to support two, three, or four crucial components failing without user impact goes up exponentially.

With an SPOF design, a crucial component will cause A/V interruption for a time MTTR2. Even with the automatic detection of an anomaly, the manual repair path must be taken as in Figure 5.2. Usually, MTTR2 is much greater than MTTR1. This can take seconds if someone is monitoring the A/V flow actively and is able to route around the failed component quickly. For example, Master Control operators in TV stations or playout control rooms are tasked to make quick decisions to bypass failed components. In other cases without a quick human response, MTTR2 may be minutes or even days. It should be apparent that availability ( $A_v$ ) is a strong function of MTTR. SPOF designs are used when the economic cost of some downtime is acceptable. Many broadcasters, network originators, and live event producers rely on NSPOF designs in the crucial paths at least.

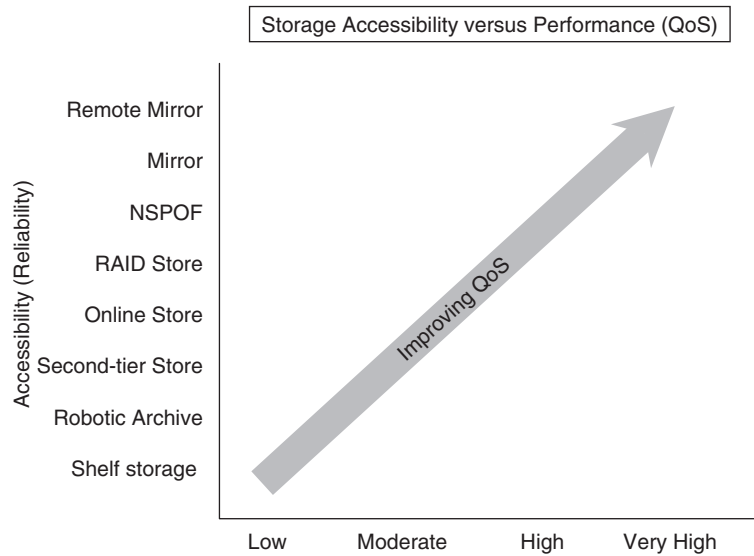
The worst-case scenario is the right side flow of Figure 5.2. Without automatic detection, it often takes someone to notice the problem. There are stories of TV channel viewers calling the station to complain of image-quality problems unnoticed by station personnel. Once failure is detected, the active signal path must be routed to bypass the faulty component. Next, the faulty part should be replaced/repared (MTTR3). All this takes time, and MTTR2 + MTTR3 can easily stretch into days.

In either case, manual or automatic detection, the faulty component eventually needs to be repaired or replaced. During this time, a NSPOF system is reduced to a SPOF one and is vulnerable to additional component failure. As a result, MTTR3 should be kept to a minimum with a diligent maintenance program. Examples of these system types are presented later in the chapter. The more you know about a system's health, the quicker a repair can be made. Chapter 9 covers monitoring and diagnostics—crucial steps in keeping the system availability high.

### 5.0.3 Failure Mechanisms

In general, A/V systems range from small SPOF types to full mirror systems with off-site recovery. Figure 5.3 plots the available system performance versus the degree of availability for storage components. Similar plots can be made for other system-level elements. Levels of reliability come in many flavors, from simple bit error correction to the wholesale remote mirroring of all online data. The increasing performance metric is qualitative but indicates a greater QoS in accessibility, reduced access latency, and speed.

Individual devices such as archives, servers, switches, near-line storage, or A/V components each have a MTBF metric. Common system elements and



**FIGURE 5.3** Storage accessibility versus performance (QoS).

influences (not strictly prioritized in the following list) that can contribute to faults are

- Individual device HW/SW and mechanical integrity
- Infrastructure wiring faults
- I/O, control, and management ports
- Middleware glue—communication between elements
- System-level software—spans several elements
- Error reporting and configuration settings
- Failover control
- Viruses, worms, and other security risks
- External control of system elements
- Network reliability
- Untested operational patterns
- Tape and optical media integrity
- Poorly conditioned or intermittent power, electrical noise
- Environmental effects—out of limits temperature, humidity, vibration, dust
- Required or preventative maintenance omitted
- Human operational errors
- Sabotage

In general, the three most common failure sources are human operational error and software- and hardware-related faults. Depending on the system configuration, either of these may dominate as failure modes. For non-human-related

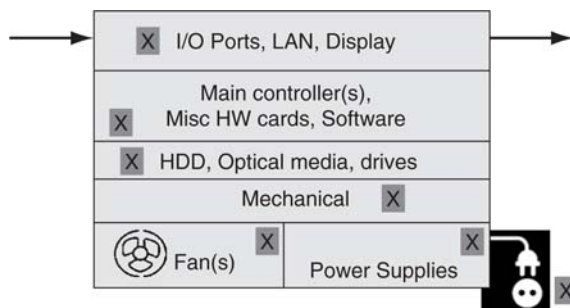
faults, Horison Information Strategies estimates that, on average, hardware accounts for 38 percent of faults, 23 percent is network based (including operational failures), 18 percent is software related, and 21 percent other.

Elements and influences are often interrelated, and complex relationships (protocol states) may exist between them. Device hardware MTBF may theoretically be computed based on the underlying design. Although in practice, it is difficult to calculate and is known more accurately after measuring the number of actual faulty units from a population over a period of time. A device's software MTBF is almost impossible to compute at product release time, and users/buyers should have some knowledge of the track record of a product or the company's reputation on quality before purchase. Often a recommendation from a trusted colleague who has experience with a product or its vendor is the best measure of quality and reliability.

In the list given earlier, the upper half items are roughly the responsibility of the supplying vendor(s) and system's integrator, whereas the lower half items are the responsibility of the user/owner. Much finger pointing can occur when the bottom half factors are at fault but the upper half factors are blamed for a failure. When defining a system, make sure that the all modes of operation are well tested by the providing vendor(s).

Naturally, one of the most common causes of failure is the individual system element. Figure 5.4 shows a typical internal configuration for a device with multiple internal points of failure. It is very difficult to design a stand-alone "box" to be fault tolerant. For most designs, this would require a duplication of most internal elements. This is cost prohibitive for many cases and adds complexity. In general, it is better to design for single unit failure with hot spares and other methods to be described. That being said, it is good practice to include at least some internal-redundant elements as budget permits.

The most likely internal components to fail are mechanically related: fans, connectors, HDD, and structure. The power supply is also vulnerable. Cooling is often designed to withstand at least one fan failure, and it is common to include a second power supply to share the load or as a hot spare. The most



**FIGURE 5.4** Standalone device with potential internal failure points.

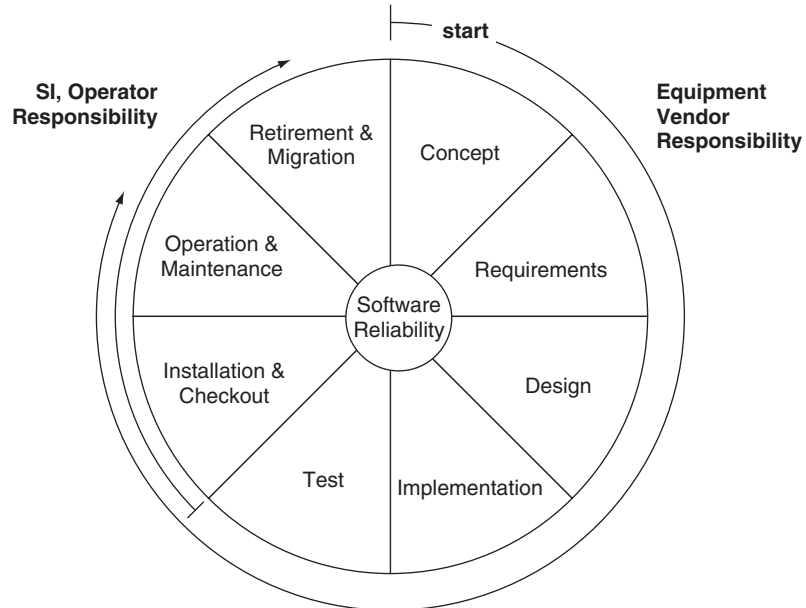
difficult aspect to duplicate is the main controller if there is one. With its internal memory and states of operations, seamlessly failing over to a second controller is tricky business.

Ah, then there is software reliability. This is without a doubt the most difficult aspect of any large system. Software will fail and users must live with this fact and design systems with this in mind. The software life cycle is shown in Figure 5.5. From concept to retirement/migration, each of the eight steps should include software design/test methodologies that result in reliable code. Note that some steps are the responsibility of the original equipment vendor, whereas others belong to the operator or installer/SI.

In the classic work *The Mythical Man-Month*, a surprising conclusion was learned from designing software for the IBM 360 mainframe: in complex systems the total number of bugs will not decrease over time, despite the best efforts of programmers. As they fix one problem or add new features, they create another problem in some unexpected way.

There are ways to reduce this condition, but bugs will never go to zero. The good news is that software does not make mistakes, and it does not wear out. See more on Lehman's laws of software evolution in Chapter 4.

One hot topic is software security in the age of viruses, worms, denial of service attacks, and so on. Although a security breach is not a failure mechanism in the traditional sense, the results can be even more devastating. Traditional A/V systems never contended with network security issues, but IT-based systems



**FIGURE 5.5** Software reliability and its life cycle.

must. Of course, A/V systems must run in real time, and virus checkers and other preventative measures can swamp a client or server CPU at the worst possible moment with resulting A/V glitches or worse. It takes careful design to assure system integrity against attacks and to keep all real-time processes running smoothly. One key idea is to reduce the surface of attack. When all holes are closed and exposure to foreign attacks is limited, systems are less vulnerable. This and other security-related concepts are covered in more detail in Chapter 8.

A potential cause of performance degradation—a failure by some accounting—is improper use of networked-attached A/V workstations. For example, a user may be told not to initiate file transfers from a certain station. If the advice is ignored, the network may become swamped with data, and other user's data are denied their rightful throughput. This is a case of a fault caused by user error.

Before starting the general discussion of configurations for high availability, let us investigate how reliability is measured for one of the most valuable elements in an A/V design: the hard disk drive.

## 5.1 HDD RELIABILITY METRICS

Because disk drives are core technology in networked media systems, it is of value to understand how drive manufacturers spec reliability. Disk drive manufacturers often state some amazing failure rates for their drives. For example, one popular SCSI HDD sports a MTBF of 1.2 million hours, which is equivalent to 137 years. The annual failure rate (AFR) is 0.73 percent. How should these values be interpreted? Does this mean that a typical HDD will last 137 years? It sounds ridiculous; would you believe it? What do MTBF and AFR really mean? The following sections investigate these questions. First, let us classify failure rate measurements into three different domains.

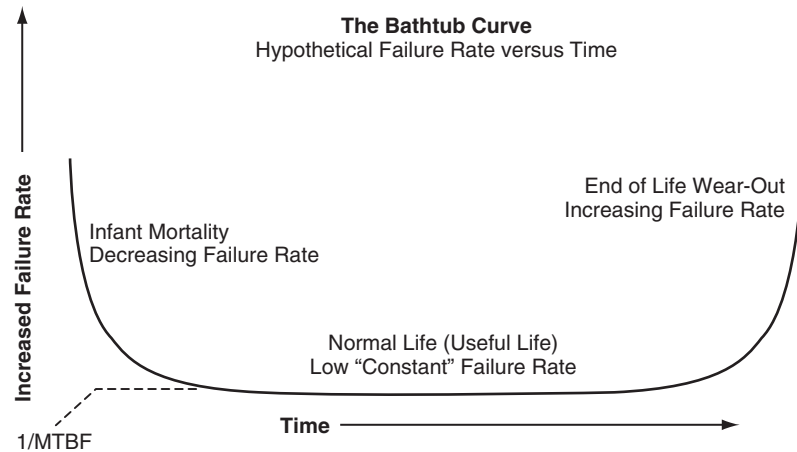
### 5.1.1 Failure Rate Analysis Domains

Measuring HDD failure rate is a little like the proverbial blind man describing an elephant—depending on his viewing perspective, the elephant looks quite a bit different. The three domains of HDD failure rate relevance are (1) *lab test domain*, (2) *field failure domain*, and (3) *financial domain*.

#### 5.1.1.1 Lab Test Domain

The *lab test domain* is a well-controlled environment where product engineers simultaneously test 500–1,000 same-vintage drives under known conditions: temperature, altitude, spindle on/off duty cycle, and drive stress R/W conditions. In this domain there are various accepted methods to measure HDD failure rates. The International Disk Drive Equipment and Materials Association (IDEMA, [www.idema.org](http://www.idema.org)) sets specs that most HDD manufacturers adhere to. The R2-98 spec describes a method that blocks reliability measurements into





**FIGURE 5.6** Component failure rate over time.

four periods. The intervals are 1–3 months, 4–6 months, 7–12 months, and 13 to end-of-design-life. In each period the failure rate expressed as  $X\%/1,000$  hours is measured. In reality, the period from 1 to 3 months is the most interesting. The results of these tests are not normally available on a drive spec sheet because evaluating a multivariable reliability metric is a knotty problem and may complicate a buyer's purchase decision.

Many electrical/mechanical components have a failure rate that is described by the bathtub curve as shown in Figure 5.6. A big percentage of "infant mortality" failures normally occur within the first 6 weeks of an HDD usage. If vendors exercise their product (burn-in) during these early weeks before shipment, the overall field failure rate decreases considerably. However, most cannot afford to run a burn-in cycle for more than a day or two, so failure rates in the field are dominated by those from the early part of the curve. In fact, most HDD vendors test SCSI drives 24/7 for 3 months during the design phase only. Correspondingly, ATA drives are often tested using stop/start and thermal cycling. This is yet another reason why SCSI is more costly than ATA drives.

However, about 10 times more ATA disk drives are manufactured today than SCSI and Fibre Channel drives combined. At these levels, ATA drive manufacturers are forced to meet very high process reliability requirements or else face extensive penalties for returned drives.

The IDEMA also publishes spec R3-98, which documents a *single*  $X\%/1,000$  hours failure rate metric. For this test, a manufacturer measures a collection of same-vintage drives for *only* 3 months of usage. For example, if an aggregate failure rate was measured to be  $0.2\%/1,000$  hours, we may extrapolate this value and expect an HDD failure before 500,000 hours ( $100/0.2 \times 1,000$ ) with almost 100 percent certainty. However, the R3-98 spec discourages using

MTBF in favor of failure rate expressed as  $X\%/1,000$  hours over a short measurement period. Nonetheless, back to the main question, how should MTBF be interpreted?

Used correctly, MTBF is better understood in relation to the useful *service life* of a drive. The service life is the point where failures increase noticeably. Drive MTBF applies to the aggregate measurement of a large number of drives, not to a single drive. If a drive has a specified MTBF of 500,000 hours (57 years), then a collection of identical drives will run for 500,000 *device hours* before a failure of *one* drive occurs. A test of this nature can be done using 500 drives running for 1,000 hours. Another way of looking at drive MTBF is this: run a single drive for 5 years (service life, see later) and replace it every 5 years with an identical drive and so on. In theory, a drive will not fail before 57 years on average.

#### 5.1.1.2 *Field Failure Domain*

Next, let us consider failure rate as derived from units returned from the field—the *field failure domain*. In this case, manufacturers measure the number of failed and returned units (normally under warranty) versus the number shipped for 1 year. The annual return rate or annual failure rate may be calculated from these real-world failures. Of course, the “test” conditions are almost completely unknown, with some units being in harsh environments and others rarely turned on. The return rate of bad drives is usually lower than the actual respective failure rate. Why? Some users do not return bad drives and some drives stay dormant on distributors’ shelves. So this metric is interesting but not sufficient to predict a drive’s failure rate.

#### 5.1.1.3 *Financial Domain*

The final domain of interest is what will be called the *financial domain*. The most useful statistic with regard to HDD reliability is the manufacturer’s warranty period. The warranty period is the only metric that has a financial impact on the manufacturer. Most vendors spec only a 5-year warranty—nowhere close to the 100+ years that the MTBF data sheet value may imply. For practical purposes, let us call this the *service life* of the drive. Of course the HDD manufacturer does not expect the drive to fail at 5 years and 1 day. Also, typical warranty costs for an HDD manufacturer are about 1 percent of yearly revenue. Because this value directly affects bottom-line profits, it is in the manufacturer’s best interest to keep returns as a very small percentage of revenue.

In reality, the actual drive lifetime is beyond the 5-year warranty period but considerably short of the so-called MTBF value. Conservatively built systems should be biased heavily toward the warranty period side of the two extremes. Of course, HDD reliability is only one small aspect of total system reliability. However, understanding these issues provides valuable insights for estimating overall system reliability and maintenance costs.

## 5.2 METHODS FOR HIGH-AVAILABILITY DESIGN

There are two kinds of equipment and infrastructure problems: those you know about and those you don't. The ones you don't know about are the most insidious. They hide and wait to express themselves at the worst time. Of those you know about, some can be repaired (corrected) and some can't. So, in summary, problems can be classified as follows:

1. Unknown problems (hidden failures, intermittent issues, etc.)
2. Known problems—detectable but not correctable
3. Known and correctable—detectable and fixable
4. Known and partially correctable—use concealment, if possible, for portion not correctable

Here are some simple examples of these four:

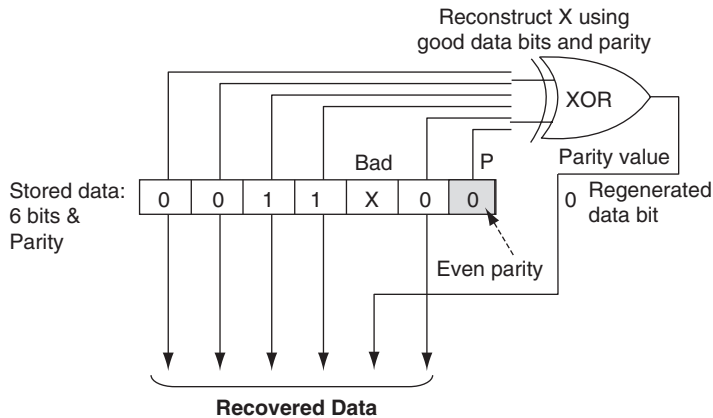
1. Two bits are bad in a field of 32. A simple checksum algorithm (error detection method) misses this case since it can detect only an odd number of bad bits.
2. A checksum result on a 32-bit word indicates an error. We know that at least one bit is bad but can't identify it. An error message or warning can be generated.
3. A full error correction (Reed/Solomon method, for example) method is applied to repair a bad bit (or more) out of 32.
4. A burst error of audio data samples has occurred. Reed/Solomon methods correct what they can (fix  $N$  bits out of  $M$  bad bits) and then conceal (mute if audio) what is not correctable. Some errors or faults are not concealable, but many are.

The set of four examples deals with bad bits. Other error types occur, such as failed hardware components, broken links, software bugs. Each error type may need a custom error detection scheme and/or associated error correction method. It is beyond the scope of this book to analyze every possible technique. Nonetheless, keep these high-level concepts in mind as this chapter develops various methods for creating reliable systems out of inherently unreliable devices.

The high-availability (HA) systems' methods under discussion are RAID for storage, storage clusters, NSPOF,  $N + 1$  sparing, dual operations, replication, mirroring, and disaster recovery.

### 5.2.1 RAID Arrays

The RAID [redundant arrays of inexpensive (independent, nowadays) disks] concept was invented at UC Berkeley in 1987. Patterson, Gibson, and Katz published a paper titled "A Case for Redundant Arrays of Inexpensive Disks (RAID)."

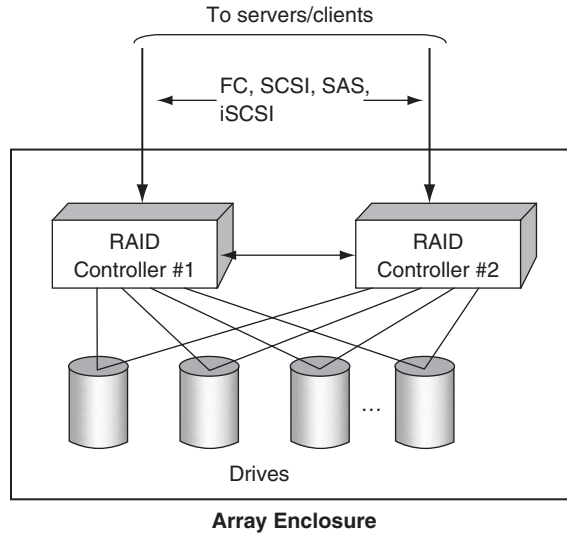


**FIGURE 5.7** RAID reconstruction example using parity.

This paper described various types of disk array configurations for offering better performance and reliability. Until then, the single disk was the rule, and arrays were just starting to gain acceptance. The authors developed seven levels of RAID. Before we define each technique formally, let us illustrate how RAID works in general terms.

The basic idea behind most RAID configurations is simple. Use a parity bit to correct for one missing (bad) bit out of  $K$  bits. Figure 5.7 shows a 6-bit data field and one parity bit. Parity measures the evenness or oddness of the 6-bit string. For example, the sequence 001100 has even parity ( $P = 0$ ), as there are an even number of ones, whereas the sequence 101010 has odd parity ( $P = 1$ ). If any single bit is in error, then when the stored parity bit is used, the bad bit may be reconstructed. Parity is generated using the simple XOR function. It is important to note that data bit reconstruction requires knowledge of the bad bit (or data block or array) position. If the  $0011 \times 0$  sequence in Figure 5.7 is given and  $P = 0$  (even number of ones), then it is obvious that  $X = 0$ ; otherwise,  $P$  could not equal zero. Of course, the parity idea can be extended to represent the parity of a byte, word, sector, entire HDD, or even an array. Hence, a faulty or intermittent HDD may be reconstructed. With an array of  $N$  data drives and one parity drive, one HDD can be completely dead and the missing data may be recovered completely. Most RAID configurations are designed to reconstruct at least one faulty HDD in real time.

Figure 5.8 is typical of an HDD array with RAID protection. In this case, the disks are protected by two RAID controllers. Either can R/W to a disc or do the needed RAID calculations. In the event of one controller going haywire, the second becomes responsible for all I/O. If designed correctly, with a passive backplane and dual-power supplies, this unit offers NSPOF reliability. Note too that each HDD has a direct link to each controller. A less reliable array may



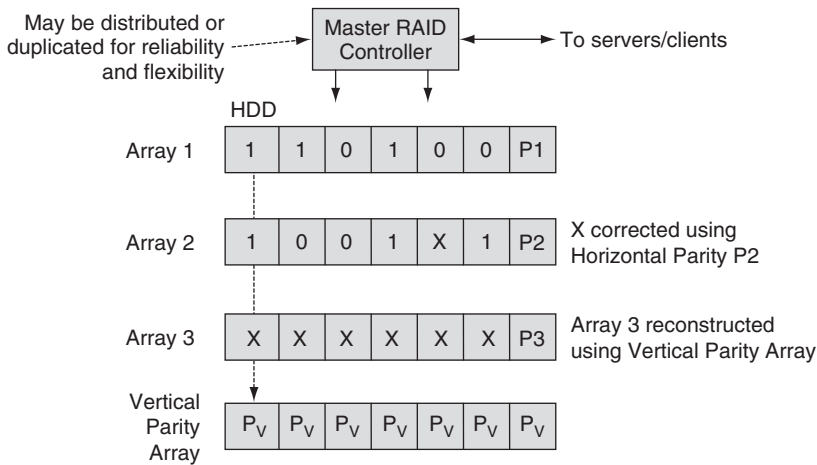
**FIGURE 5.8** HDD array with dual RAID controllers.

connect each HDD to one or two internal buses. In this case, one faulty HDD can hang a bus and all connected drives will become inaccessible. With care, an array can offer superior reliability. Several manufacturers offer NSPOF, dual-controller arrays ranging from a small 8-drive enclosure to an enormous array with 1,100 drives.

Incidentally, all clients or servers that access the storage array must manage the failover in the event of a controller failing or hanging. For example, if a client is doing a read request via controller #1 but with no response, it is the client's responsibility to abort the read transaction and retry via controller #2. As may be imagined, the level of testing to guarantee glitch-free failover is non-trivial. This level of failover is offered by a few high-end A/V systems' vendors.

### 5.2.1.1 Two-Dimensional (2D) Parity Methods

If we extend the RAID idea, it is possible to design a 2D array with two levels of parity. Figure 5.9 shows a 2D approach. One dimension implements horizontal RAID with parity for correcting data from a single faulty HDD. The second dimension implements vertical RAID and can correct for an *entire array* in the event of failure. The vertical parity method spans arrays, whereas the horizontal method is confined to a single array. The overhead in vertical parity can be excessive if the number of protected arrays is low. A four (three data + VP) array system has 25 percent overhead in vertical parity plus the horizontal parity overhead. Note that the parity value P3 is of no use when the entire array faults. Two-dimensional parity schemes can be complex, and they offer excellent reliability but are short of a complete mirror of all data.



**FIGURE 5.9** Two-dimensional horizontal and vertical parity methods.

Two-dimensional parity methods require some sort of master RAID controller to manage parity. For standard 1D parity, each array has its own internal (normally, as shown in Figure 5.8) controller(s) for managing the parity values and reconstructing data. However, for a 2D parity method as illustrated in Figure 5.9, no single internal array controller can easily and reliably manage both H and V parity across all arrays. However, there are several different controller configurations for managing and reconstructing missing data using 2D parity. Three of these are as follows:

1. A *master RAID controller* manages all parity (H and V) on all arrays. This may be a single physical external controller (or two) or distributed in some way to span one or more physical arrays.
2. Each array has an internal horizontal RAID controller *and* some external controller or distributed controllers to manage the vertical parity. The SeaChange Broadcast Media Cluster/Library<sup>1</sup> supports this form of 2D parity, although the vertical parity is distributed among arrays and is not concentrated in one array.
3. The vertical and horizontal parity schemes are both confined to the same array with an internal controller(s). This case allows for *any two drives* to fail per array without affecting operations. This variation is known as RAID-6 and is discussed later. Interestingly, this case is not as powerful as the other two, as it cannot recover a complete array failure.

<sup>1</sup> The SeaChange product does not reference “vertical” or “horizontal” parity in its documentation. However, in principle, it uses two levels of distributed parity to support any single HDD failure per array and any one complete array failure.

Judicious placement of the H/V controller(s) intelligence can provide for improved reliability with the same parity overhead.

### 5.2.1.2 Factors for Evaluating RAID Systems

Despite the relative conceptual simplicity of RAID using parity, here are some factors to be aware of:

- There is normally a RAID controller (or two) per array. It manages the R/W processes and RAID calculations in real time.
- The overhead for parity protection (1D) is normally one drive out of  $N$  drives per array. For some RAID configurations, the parity is distributed across the  $N$  drives.
- Some arrays use two parity drives but still rely on 1D correction. In this case, the layout is grouped  $(N + P)$  and  $(N + P)$ , with  $2N$  data drives protected by two parity drives. Each  $N + P$  group is called a RAID set. One drive may fail per RAID set, so the reliability is better than one parity drive for  $2N$  data drives. For example,  $5 + 1$  and  $7 + 1$  are common RAID set descriptors. This is not a 2D parity scheme.
- There is a HDD rebuild effort. In the event of a HDD failure, a replacement unit should be installed immediately. The missing data are rebuilt on the new HDD (or a standby spare). Array RAID controllers do the rebuild automatically. Rebuilding a drive may take many hours, and the process steals valuable bandwidth from user operations. Even at 80 Mbps rebuild rates, the reconstruction time takes 8.3 hr for a 300GB drive.
- All other array drives need to be read at this same rate to compute the lost data. Also, and this is key, if a second array HDD faults or becomes intermittent before the first bad HDD is replaced and rebuilt, then no data can be read (or written) from the entire array. Because RAID methods hide a failed drive from the user, the failure may go unnoticed for a time. Good operational processes are required to detect bad drives immediately after failure. If the HDD is not replaced immediately, the MTTR interval may become large.

## 5.2.2 RAID Level Definitions

The following sections outline the seven main RAID types. Following this, the RAID levels are compared in relation to the needs of A/V systems.

### 5.2.2.1 RAID-0

Because RAID level 0 is not a redundancy method, it does not truly fit the “RAID” acronym. At level 0, data blocks are striped (distributed) across  $N$  drives, resulting in higher data throughput. See Figure 3A.14 for an illustration of file striping. When a client requests a large block of data from a RAID-0

array, the array controller (or host software RAID controller) returns it by reading from all  $N$  drives. No redundant information is stored and performance is very good, but the failure of any disk in the array results in data loss, possibly irrecoverably. As a memory aid, think “RAID-0 means zero reliability improvement.”

#### 5.2.2.2 RAID-1

RAID level 1 provides redundancy by writing all data to two or more drives. The performance of a level 1 array tends to be faster on reads (read from one good disk) and slower on writes (write to two or more disks). This level is commonly referred to as mirroring. Of course, a mirror requires  $2N$  disks compared to a JBOD ( $N$  raw disks), but the logic to manage a mirror is simple compared to other RAID redundancy schemes.

#### 5.2.2.3 RAID-01 and RAID-10 Combos

Both RAID-01 and RAID-10 use a combination of RAID-0 and -1. The 01 form is a “mirror of stripes” (a mirror of a RAID-0 stripe set), and the 10 form is a “stripe of mirrors” (a stripe across a RAID-1 mirror set). There are subtle trade-offs between these two configurations, and performance depends on the distance the mirrors are separated.

#### 5.2.2.4 RAID-2

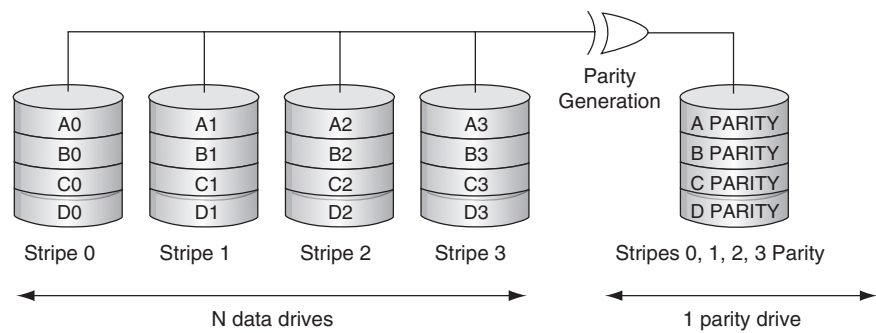
RAID level 2, which uses Hamming error correction codes, is intended for use with drives that do not have built-in error detection. Because all modern drives support built-in error detection and correction, this level is almost never used.

#### 5.2.2.5 RAID-3

RAID level 3 stripes *byte*-level data across  $N$  drives, with  $N$ -drive parity stored on one dedicated drive. Bytes are striped in a circular manner among the  $N$  drives. Byte striping requires hardware support for efficient use. The parity information allows recovery from the failure of any single drive. Any R/W operation involves all  $N$  drives. Parity must be updated for every write (see Figure 5.10). When many users are writing to the array, there is a parity drive write bottleneck, which hurts performance. The error correction overhead is  $100\% * [1/N]$  compared to 100 percent for the mirror case. A RAID-3 set is sometimes referred to as an “ $N + 1$  set,” where  $N$  is the data drive count and 1 is the parity drive.

Although not an official RAID level, RAID-30 is a combination of RAID-3 and RAID-0. RAID-30 provides high data transfer rates and excellent reliability. This combination can be achieved by defining two (or more) RAID sets within a single array (or different arrays) and stripe data *blocks* across the two sets. An array with  $K$  total drives may be segmented into two RAID-3 sets, each as  $N + 1$  drives. For example, with  $K = 10$  drives,  $4 + 1$  and  $4 + 1$  RAID-3 sets may be defined within a single array. User data may be block striped across the





**FIGURE 5.10** RAID level 3 with dedicated parity drive.  
Source: AC&NC.

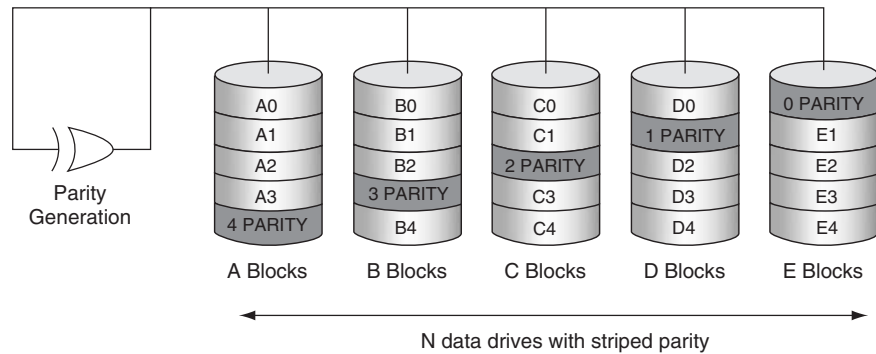
two sets using a RAID-0 layout. Note that any two drives may fail, without data loss, if they are each in a different set.

**5.2.2.6 RAID-4**

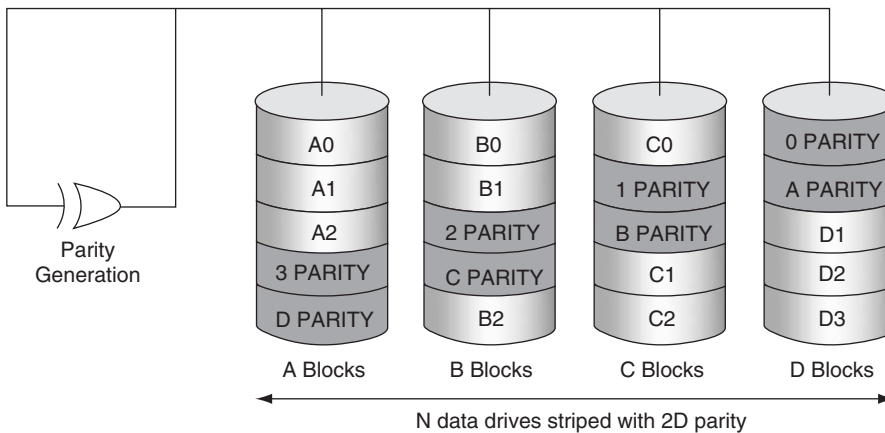
RAID level 4 stripes data at a *block* level, not byte level, across several drives, with parity stored on one drive. This is very similar to RAID-3 except the striping is block based, not byte based. The parity information allows recovery from the failure of any single drive. The performance approaches RAID-3 when the R/W blocks span all *N* disks. For small R/W blocks, level 4 has advantages over level 3 because only one data drive needs to be accessed. In practice, this level does not find much commercial use.

**5.2.2.7 RAID-5**

RAID level 5 is similar to level 4, but distributes parity among the drives (see Figure 5.11). This level avoids the single parity drive bottleneck that may occur with levels 3 and 4 when the activity is biased toward writes. The error correction overhead is the same as for RAID-3 and -4. RAID-50, similar in concept to RAID-30, defines a method that stripes data blocks across two or more RAID-5 sets.



**FIGURE 5.11** Raid level 5 with distributed parity.  
Source: AC&NC.



**FIGURE 5.12** RAID level 6 with 2D parity.  
Source: AC&NC.

### 5.2.2.8 RAID-6

RAID-6 is essentially an extension of RAID level 5, which allows for additional fault tolerance by using a second independent distributed parity scheme (two-dimensional, row/column computed parity) as illustrated in Figure 5.12. There are several ways to implement this level. In essence, with two parity values, two equations can be solved to find two unknowns. The unknowns are the data records from the two bad drives. This scheme supports two simultaneous drive failures (within a RAID set) without affecting performance (ideally). So, any two drives in Figure 5.12 could fail with no consequent loss of data. This mode is growing very popular, and several IT- and A/V-specific vendors offer storage with this protection level. Compare this to Figure 5.9 where the 2D parity is distributed across several arrays, not only across one array.

The salient RAID aspects are as follows:

- RAID-0 striping offers the highest throughput with large blocks but with no fault tolerance.
- RAID-1 mirroring is ideal for performance-critical, fault-tolerant environments. It is often combined with RAID-0 to create a RAID-10 or -01 configuration. These configurations are popular in A/V systems, despite the fact that RAID-3 and -5 are more efficient methods to protect data.
- RAID-2 is seldom used today because ECC is embedded in almost all modern disk drives.
- RAID-3 can be used in data intensive or single-user environments that access long sequential records to speed up data transfer. A/V access patterns often favor this form, especially the RAID-30 configuration.

- RAID-4 offers no advantages over RAID-5 for most operations.
- RAID-5 is the best choice for small block, multiuser environments that are write performance sensitive. With large block A/V applications, this class performs similarly to RAID-3. RAID-50 is also popular in A/V applications.
- RAID-6 offers two-drive failure reliability.

RAID calculations may be done in the array's I/O controller or in the attached client. In the first case, dedicated hardware is used to do the RAID calculations, or the RAID logic is performed by software in the CPU on the I/O controller. However, client-based RAID is normally limited to one client, as there is the possibility of conflicts when multiple client devices attempt to manage the RAID arrays. Microsoft supports so-called software RAID in some of its products.

At least one A/V vendor (Harris's Nexio server) has designed a system in which all RAID calculations are done by the client devices in software. This system uses backchannel control locking to assign one A/V client as the master RAID manager. The master client controls the disk rebuild process. If the master faults or goes offline, another client takes over. Each client reconstructs RAID read data as needed. Client-based RAID uses off-the-shelf JBOD storage in most cases.

#### **5.2.2.9 RAID and the Special Needs of A/V Workflows**

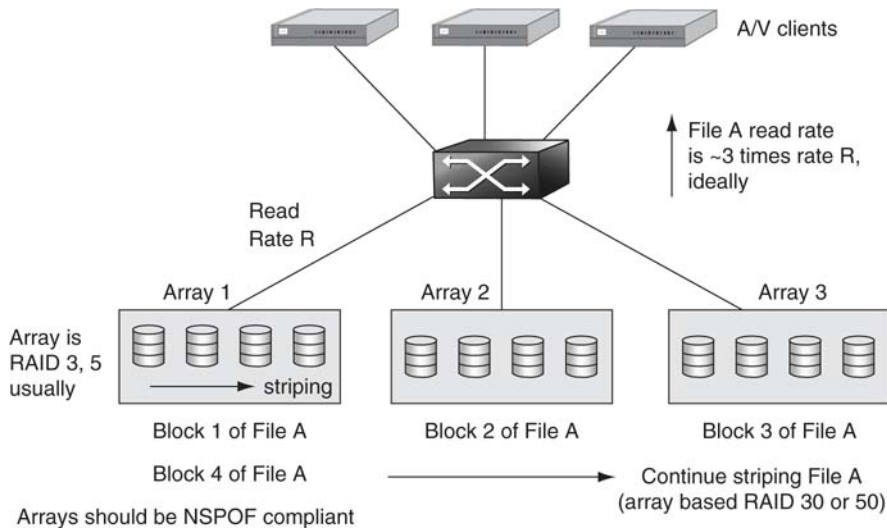
From the perspective of A/V read operations, RAID-3 and -5 offer about the same performance for large blocks. Write operations favor level 3 because there is only one parity-write operation, not  $N$  as with level 5. A/V systems are built using RAID levels across all available types except 2 and 4 for the most part.

For A/V operations, the following may be important:

- RAID data reconstruction operations should have no impact on the overall A/V workflow. Even if the data recovery operation results in throughput reduction, this must be factored into the system design. Design for the rainy day, not the sunny one.
- It is likely that the reconstruction phase will rob the system of up to 25 percent of the best-case throughput performance. Slow rebuilds (less priority assigned than for user I/O requests) of a new disk have less impact on the overall user I/O throughput. However, slow rebuilds expose the array to vulnerability, as there is no protection in the event of yet another drive failure.

### **5.2.3 RAID Clusters**

Another form of RAID-0 is to stripe across individual arrays (clustering). Say there are three arrays, each with RAID-3 support ( $N + 1$  drives per array). For maximum performance in an A/V environment, it is possible to stripe a single



**FIGURE 5.13** Array-level data striping increases throughput.

file across all the arrays. This increases the overall throughput to three times the throughput of an individual array on average. Typical striping could be such that data block 1 is on array 1, 2 on 2, 3 on 3, block 4 on array 1, and so on in a continuous cycle of striping. However, the entire storage is now three times as vulnerable compared to the same file stored on only one array. If any one of the three arrays faults, all stored files are lost, possibly forever. Hence, there is the need to guarantee that each array is NSPOF compliant and has a large MTBF metric (see Figure 5.13).

Is array striping really needed? For general-purpose enterprise data, probably not. However, if all clients need access to the same file (news footage of a hot story), then the need for simultaneous access is paramount. Take the case in which one array supports 400Mbps of I/O, and the files are stored as 50Mbps MPEG. If 15 clients need simultaneous dual-stream access, then the required throughput would be  $15 \times 2 \times 50 \text{ Mbps} = 1,500 \text{ Mbps}$ . If we want to get this data throughput, the hot file should ideally be striped across at least four different arrays, providing 1,600Mbps of read bandwidth. Many A/V systems vendors support array clustering to meet this requirement.

As it turns out, it is often easier to guarantee full availability to all files than to manage a few hot files. However, some storage vendors allow users to define the stripe width on a per-file basis. Of course, there are practical issues, such as upgrading for more storage and managing the wide stripes, but these issues can be resolved.

Striping across arrays is a multi-RAID schema. As discussed previously, RAID levels 30 and 50 are methods primarily for intra-array data and parity layout.

These RAID level names can be extended to include interarray striping as well (Figure 5.13). There is no official sanction for these names, but striping across RAID sets whether intra- or inter-based is commonly done for A/V applications.

In practice, as files are striped across  $M$  arrays (a cluster), the aggregate access rates do not increase perfectly linearly. This was the case for files striped across individual HDD devices, as shown in Chapter 3A. It was reasoned that striping files across  $N$  individual disks increases the access rates but not exactly proportional to  $N$ . Also, if files are striped across  $M$  arrays, the aggregate access rate is not exactly  $M$  times the access rate of one array. Access strategies vary, and it is possible to approach a linear rate increase using clever queuing tactics (at the cost of increased R/W latency), but a deeper discussion is beyond the scope of this book.

### 5.2.3.1 *Scaling Storage Clusters*

A cluster of arrays with striped files is difficult to scale. Imagine that the storage capacity of Figure 5.13 is increased by one array, from three to four. If all files are again striped across all arrays, then each file must be restriped across four arrays. Next, the old three-stripe file is deleted. This process must be automated and can take many hours to do, depending on the array size and available bandwidth to do the restriping.

If the arrays are in constant use, then the restriping process can be very involved. Files in use (R/W) cannot be moved easily until they are closed. Files in read-only usage may be moved, but with strict care of when to reassign the file to the new four-stripe location. If the cluster is put offline for a time, then the upgrade process will be much faster and simpler. There are other strategies to scale and restripe, but this example is typical. Note that the wider the stripe (more arrays), the more risk there is of losing file data in the event of any array failure.

A few A/V vendors offer live scaling and restriping on array clusters for selected products. It is always good to inquire about scalability issues when contemplating such a system. A few of the questions to ask a providing vendor are as follows:

- Live or offline restriping? Any user restrictions during restriping?
- Time delay to do the restriping?
- Scalability range—maximum number of arrays? Array (or node) increment size?
- For wide striping, what is the exposure to an array fault? Do we lose all the files if one array faults?
- Are the arrays NSPOF in design?

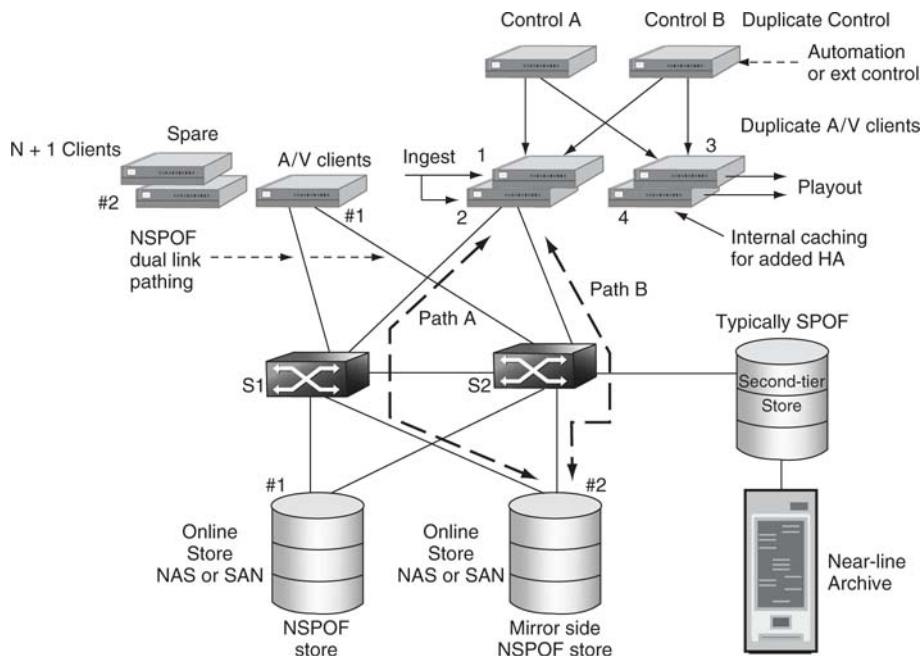
There are other methods of adding storage and bandwidth, but they tend to be less efficient. One method is to add mirrored storage. For read-only files (Web

servers, some video servers), this works well, but for file data that are often modified, synchronizing across several mirrored copies is very difficult and can easily lead to out-of-sync data. More on mirrored storage later.

### 5.3 ARCHITECTURAL CONCEPTS FOR HA

“Oh, it looks like a nail-biting finish for our game. During the timeout, let’s break for a commercial.” Is this the time for the ad playback server or automation controller to fail? What are the strategies to guarantee program continuity? Of course, there are many aspects to reliable operations, but equipment operational integrity ranks near the top of the list. Author Max Beerbohm said, “There is much to be said for failure. It is more interesting than success.” What is interesting about failure? How to avoid it!

Figure 5.14 illustrates several HA methods in one system. The design exemplifies SPOF, NSPOF,  $N + 1$  sparing, network self-healing, mirroring, and client caching. A practical design may use one or some mix of these methods. Let us examine each of them. Keep this in mind: reliability and budget go hand in hand. The more reliable the overall system, the more costly it is under normal circumstances. Paths of traditional A/V flows are not shown to simplify the diagram, but redundancy is the stock in trade to guarantee reliable operations.



**FIGURE 5.14** High availability using NSPOF,  $N + 1$ , mirroring, and caching.

### 5.3.1 Single Point of Failure

Why use SPOF when NSPOF is available? The answer is cost and simplicity. The cost of a NSPOF robotic archive, for example, is prohibitive for most operations. The need for reliable storage decreases as the device becomes more distant from online activities. In the case of archived A/V content, most often the need for materials is predicted days in advance. Near-line storage and servers may be SPOF for similar reasons. If business conditions allow for some downtime on occasion (consistent with the availability, MTBF, and MTTR numbers), then by all means use SPOF equipment.

### 5.3.2 No Single Point of Failure

NSPOF designs come in two flavors: standalone devices and systems. A standalone device needs duplicated internal elements (controllers, drives, power supplies, and so on). These can be very expensive and are rare. One notable example is the NonStop brand of mainframe from HP (invented by Tandem Computers in 1975). Also, some IP switchers/routers make the claim of NSPOF compliance. However, a system-centric NSPOF design relies on judicious SPOF element duplication to guarantee continuous operations.

It is not practical to design every system element to provide NSPOF operation. However, dual SPOF elements may be configured to act like a single NSPOF element. Critical path elements may be NSPOF in design. In Figure 5.14 there are several functionally equivalent NSPOF elements: the central switches are duplicated, some paths occur in pairs, control A and B elements are duplicated, and the ingest and playout A/V clients are duplicated. Here are a few of the ways that dual elements may be used to implement NSPOF:

- A duplicate element lies dormant until needed—a link, for example.
- Dual elements share the load until one faults, and then the other carries the burden—the switches S1 and S2, for example. Depending on the design, a single switch may be able to carry all the load or provide for at least some reduced performance.
- In the case of control elements, each performs identical operations. Devices Control A or Control B can each command the ingest and playout nodes. For example, if A is the active controller, then B runs identically with the exception that its control ports are deactivated until Control A faults. Alternatively, A may control ingest #1 and B may control ingest #2. If either ingest port or controller faults, there is still a means to record an input signal using the other controller or ingest port.

In some designs, A/V clients are responsible for implementing NSPOF failover functionality. Take the case of recording ingest port #1. Path A is taken to record to online storage #2. If switch S1 faults, a connecting link faults or

the online controller faults, then the client must abort the transaction and reinitiate using alternate path B. If path switching is done quickly (within buffering limits), then none of the input signal will be lost during the record process.

For bulletproof recording, an ingest client may record both to online stores #1 and #2 using a dual write scheme. This keeps the two stores in complete sync and guarantees that even if one fails, the other has the goods. Additionally, an input signal may be ingested into ports #1 and #2 for a dual record. Either port may fail, but at least one will record the incoming signal.

Because NSPOF designs are costly, is there a midpoint between SPOF and NSPOF? Yes, and it is called  $N + 1$  sparing.

### 5.3.3 $N + 1$ Sparing

While NSPOF designs normally require at least some  $2\times$  duplication of components at various stages,  $N + 1$  sparing requires only one extra, hot spare, out of  $N$  elements. It is obvious that this cuts down on capital cost, space, and overall complexity. So how does it work?

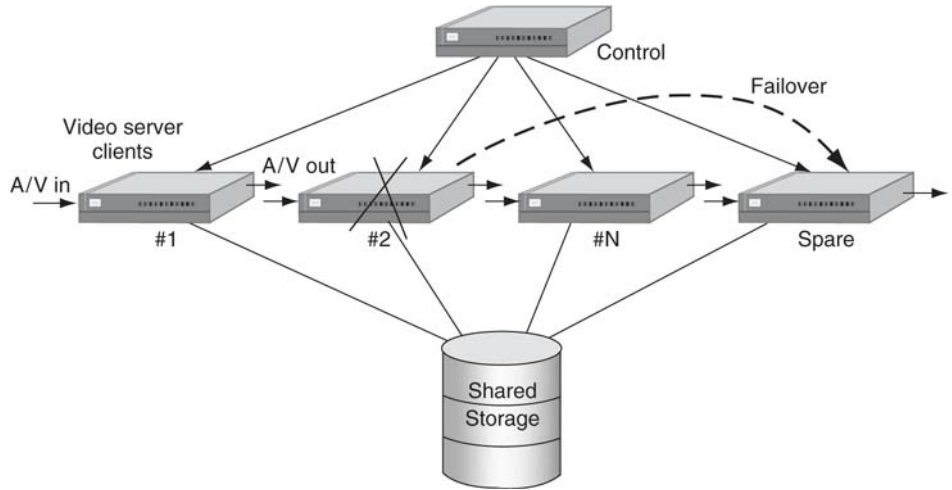
Let us consider an example. Say that  $N$  Web servers are actively serving a population of clients. An extra server (the  $+1$ ) is in hot standby, offline. If server # $K$  fails, the hot spare may be switched into action to replace the bad unit. The faulty unit is taken offline to be repaired. Of course, the detection of failure and subsequent switching to a new standby unit require some engineering. Another application relates to a cluster of NAS servers. In this case, if a server fails out of a population, it must be removed from service, and all R/W requests are directed to a hot standby. For example, if the clients accessing the NAS are A/V edit stations (NLEs), then they must have the necessary logic to monitor NAS response times and switch to the spare. Each client has access to a list of alternate servers to use in failover. Failover is rarely as smooth as with NSPOF designs, as there will be some downtime between fault detection and rerouting/switching to the spare. Also, any current transaction will likely be aborted and restarted on the spare. Most NLEs do not support client-based storage failover, but the principle may be applied to any A/V client.

Of course,  $N + 2$  sparing (or  $N + K$  in general) is better yet. The number of spares is dependent on the likelihood of an element failure and business needs. Some designs select “cheap” elements knowing that  $N + K$  sparing is available in the event of failure. This idea is not as daffy as it first sounds. In fact, some designs shun NSPOF in favor of  $N + K$  to cut equipment costs, as NSPOF designs can become costly. This is a good trade-off based on the trends shown in Figure 5.1.

Let us consider one more example. In Figure 5.15 there are  $N$  active video server I/O devices under control of automation for recording and playback. If device #2 fails, then by some means (automatic or manual) the recording and playout may be shifted to the spare unit.

Of course, this also requires coordination of input and output signal routing. Also, this works best when all stored files are available to all clients. In some





**FIGURE 5.15** *N + 1 sparing for a bank of video server nodes.*

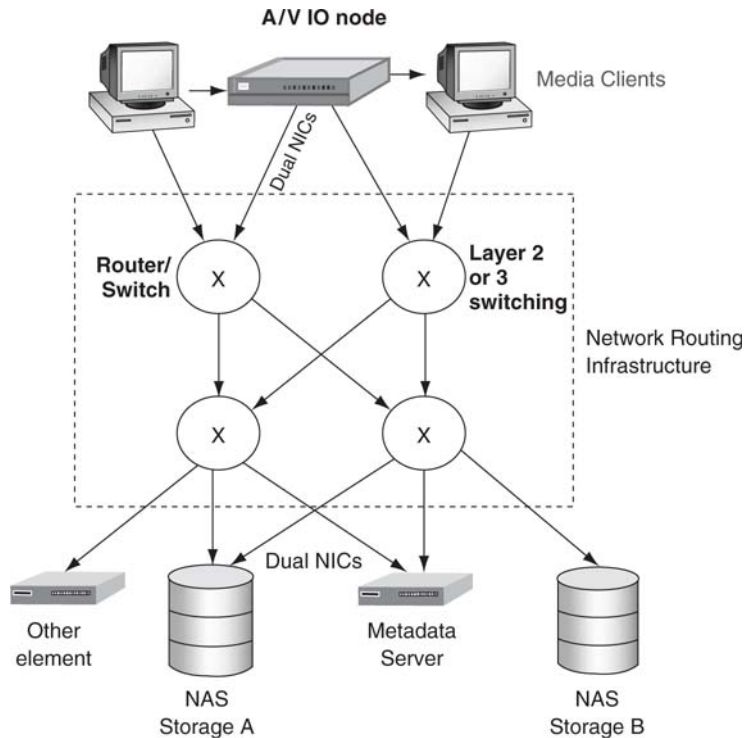
practical applications,  $N + 1$  failover occurs with a tolerable delay in detection and rerouting. We must mention that not all automation vendors support  $N + 1$  failover. However, all automation vendors do support using the brute-force method or running a complete mirror system in parallel, but this requires  $2N$  clients, not  $N + 1$ . Another vital aspect of AV/IT systems is network reliability. A failure of the network can take down all processes. HA networking is discussed next.

### 5.3.4 Networking with High Availability

Let us dig a little deeper into the methods for creating a high-availability networking infrastructure. Figure 5.16 illustrates a data routing network with alternate pathing and alternate switching. The concepts of layer 2 and layer 3 switching, routing protocols, and their failover characteristics are discussed in Chapter 6.

Figure 5.16 has some similarities to Figure 5.14. In addition to client-side route/path control, it is possible to create a network that has self-healing properties. It is always preferred for the network to transparently hide its failures than to invoke the client to deal with a network problem. For example, if one switch fails, packets may be routed over a different path/switch as needed. There are various routing and switching protocols, and each has its own specific automatic failover characteristics. Frankly, layer 2 switching is very practical for A/V workflow applications using small networks. Layer 3 is more advanced in general but can be used to route real-time A/V for more complex (including geographically separated) workflows.

Keep in mind that networked systems may introduce occasional delays that are not necessarily hard faults. For example, a local DNS cache may time out,



**FIGURE 5.16** Data routing with alternate pathing and switching.

requiring a delay in resolving a name to a numerical IP address. Routing tables too may need updating, causing a temporary and infrequent delay. These delays may occur at the most inopportune moment, so you need to plan for them or design them out.

Finally, there is a method (not shown) for creating a fault-tolerant “virtual router” using a pair of routers. One router is in standby, while the other is active. The Virtual Router Redundancy Protocol (VRRP, RFC 2338) performs an automatic failover between the two routers in the event that the active one fails. All connecting links attach to both routers, so there is always routed connectivity via one switch or the other. In effect, VRRP creates a NSPOF-configured router.

### 5.3.5 Mirroring Methods

In 1997 Alan Coleman and his geneticists amazed the world when his team cloned a sheep. Dolly became a household name. Cloning sheep holds little promise for A/V applications, but cloning data—now that is a different story. Cloned data are also called a *data mirror*. So how can mirroring juice up storage reliability?

If used in conjunction with  $N + 1$  and NSPOF methods, mirrors provide the belt-and-suspenders approach to systems design. Mirrors may include the following:

1. Exact duplicate store pools. Every file is duplicated on a second storage device. The mirror files are usually not accessed until there is a fault in the primary storage system. Ideally, both sides of the mirror are always 100 percent consistent. Figure 5.14 illustrates a storage mirror. Keeping mirrors perfectly in sync is non-trivial engineering. Also, after one side fails, they need to be resynced.
2. Mirrored playout.  $N$  active playout channels are mirrored by another  $N$  playout channels in lock step. The second set of channels may (one or all) be anointed as active whenever its mate(s) faults.
3. Mirrored record.  $N$  active ingest channels are mirrored by another  $N$  record channels in lock step. Assume a recording of one A/V signal via two inputs. Once the ingest is complete, there are two identical files in storage (using different names or different directories). One may then be deleted knowing that the other is safe, or the second file may be recorded onto a storage mirror, in which case both files are saved and likely with the same name.
4. Mirrored automation methods. As shown in Figure 5.14, some facilities run two automation systems in lock step. One system may control half the channels, thereby avoiding the case of a complete system failure, or each may control duplicate systems, one active and one in “active standby.”
5. Off-site mirror. A storage mirror (or even complete duplicate system) gives a measure of so-called disaster recovery. If the main system fails due to power failure, water damage, or some other ugly scenario, the off-site system can offer some degree of business continuity. In broadcast TV, a facility may transmit many channels out of one facility. Keeping the channel brands alive even with reduced quality levels (using low bit-rate materials) or fewer choices of programming is worthwhile for many large broadcasters. Off-site systems (broadcast applications) fall into the following categories.
  - a. Complete mirror system of primary system. This can get costly but is used when business conditions warrant. One example is the BBC Center in London. The main playout occurs from a location in London with a 48-hour mirror of all playout channels at another location 30 miles away. A WAN link provides the conduit for file transfer to sync the remote site.
  - b. Partial mirror of primary system. The partial system may be able to sustain a few of the most important channels for a time.

- c. “Keep alive” system. In this case the off-site provides canned programming on select channels for a time. The canned programming is selected to keep viewers from turning away even if it is not the advertised scheduled programming.

One of the main worries when using a storage mirror is keeping them 100 percent identical. If one side of mirrored storage fails, then the other unit will likely contain new files by the time the storage is repaired. If this happens, some resync operation needs to occur to guarantee true mirrored files. This should be automatic. Using mirrors is usually based on business needs. If they can be avoided, the overall system is simplified.

### 5.3.6 Replication of Storage

Storage replication is a poor man’s mirror. Replicated storage is a snapshot in time of the primary source. As the primary storage is modified, the secondary one is updated as fast as business needs allow for. If the secondary storage is connected via a low bandwidth WAN, then the syncing may lag considerably. For read-centric applications where the primary storage changes infrequently, replication is a better bet than a sophisticated mirror approach. The secondary storage may be of lesser performance and, hence, less costly. If the primary storage fails, then some logic needs to kick in to use the secondary files. Many A/V system vendors and integrators can provide storage replication functionality.

Storage backup (tape, optical) finds a place in A/V systems too. Inexpensive backup will have slow recovery times, so design accordingly. Backup is not the same as archive. Archive content lifetime may be 50+ years, whereas most backups last days or weeks. Some A/V systems keep a spare copy only as long as the near-line or online store has the working copy. As a result, the save time can vary depending on schedules and use patterns.

### 5.3.7 Client Caching Buys Time

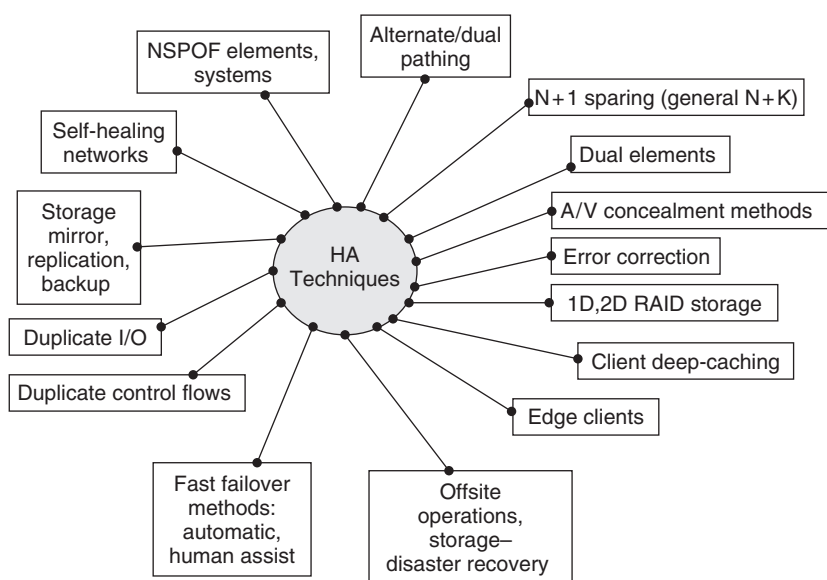
Client local caching (~buffering) also provides a measure of HA. Figure 5.14 shows several clients with A/V outputs. The main programming source files for playout come from either online or near-line storage. Usually, each playout client also has a measure of local storage. If the programming is known in advance (e.g., broadcast TV), then the local client cache may be prefilled (pre-queued) with target files. When playout starts, files are read from local memory and not directly from any external stores. With sufficient buffering, it is possible to cut the umbilical cord between the playout (or ingesting) client and external storage with no ill effects for some set time. In fact, a client may play out (or record) hours’ or days’ worth of A/V even though the main storage and connecting infrastructure are down. If the playout or record schedule is also cached in the client, then all client operations are independent of external control or storage. For sure, local caching increases the overall system reliability.

Refer to Figure 3B.22 and associated text for more insights into the advantages of caching.

Many control schemas do not support local caching, preferring to run directly from online storage. One issue is last-minute changes. Given a list of prequeued files, they may need to be flushed if the schedule changes at the last minute. This may happen in a live news or sports program. The control logic is more complex with caching but worth the effort to gain the added reliability. Also, running off local client-based storage relaxes the QoS of the external storage and connecting infrastructure, as the queuing step may be done in NRT under most circumstances. Also, the internal storage will need to support the required bandwidth (say for HD rates), not just the needed storage capacity. As A/V vendors make their products more IT friendly, expect to see more use of client caching.

One more use of local client caching is for remote operations. If the A/V client is remote from the main storage, providing a reliable WAN-based QoS can be costly. With proper client caching, remote clients may offer a high level of reliability, despite a low-grade link to main storage. This technique works best when the record/playout schedules are stable and known in advance. Remote clients are sometimes called *edge clients* because they are at the edge of a network or system boundary.

Figure 5.17 summarizes the main themes discussed so far. It is a good reminder of the choices that systems designers have at their disposal when configuring HA systems. There are other methods to build HA systems and the next section focuses on some novel approaches.



**FIGURE 5.17** Summary of high-availability techniques.

### 5.3.8 Other Topologies for HA

Currently, there is industry buzz about the virtual data center, virtual computing, and utility computing. Each of these describes a variant of a configuration where requests for services are routed to an available resource. Imagine a cluster of servers, clients, processors, and networks all offered as undedicated resources. A managing entity can assign “jobs” to server/client/network resources; when the job is complete, the resource is released and returned to the pool, ready for the next assignment. If one resource faults, another can take its place, although not always glitch-free in the A/V sense.

In enterprise IT, these new paradigms are capturing some mind share. The sense of a dedicated device statically assigned a fixed task is replaced by a pool of resources dynamically scheduled to perform as needed. Reliability, scalability, and flexibility are paramount in this concept. This idea has not yet been embraced by A/V systems designers, but the time will come when virtual computing will find some acceptance for select A/V (encoding, decoding, format conversing, conforming, etc.), control (scheduling, routing), and storage (NAS servers) processes. Although not yet applicable to A/V applications (other than searching methods), one particularly interesting no-fault design is that used by Google (Barroso). It has configured ~500,000 “unreliable” SPOF servers linked to form huge searching farms located worldwide. Query terms (video + IT + networking) are split into “shards,” and each shard is sent to an available search engine from a pool of thousands. The results of each shard query are intersected, and the target sites are returned to the client. The system scales beautifully and can withstand countless server failures with little impact on overall performance. There has been speculation that a Google-like architecture may be offered as a utility computing resource, ready for hire. This is an exciting area, ripe for innovation.

### 5.3.9 Concealment Methods

When all else fails, hide your mistakes. Sound familiar? Not only does this happen with human endeavors, A/V equipment does it as well. It is not a new idea; when a professional VTR has tape dropout problems, the output circuitry freezes the last good frame and mutes the audio until the signal returns. Concealing problems is preferred to viewing black or jerky video frames or hearing screeching audio. Similar techniques may be used when a device receives corrupted data over a link or data are late in arriving from storage. A good strategy is to hold the last good video frame and to output the audio if it has integrity or else mute it. When you are recording an A/V signal that becomes momentarily corrupt, it is good practice to record the last valid frame along with any good audio until signal integrity returns. It is preferable to conceal problems than to fault, display, or record garbage.

### 5.3.10 A Few Caveats

When specifying an HA design, always investigate what the performance is *after* a failure. Is it less than under normal operations, is the failover transparent,

and who controls the failover—man or machine? Also, what is the repair strategy when an element fails? Can the offending element be pulled without any reconfiguring steps? When a faulty element is replaced, will it automatically be recognized as a healthy member of the system? What type of element reporting is there? HA is complex, so ask plenty of questions when evaluating such a design.

## 5.4 SCALING AND UPGRADING SYSTEM COMPONENTS

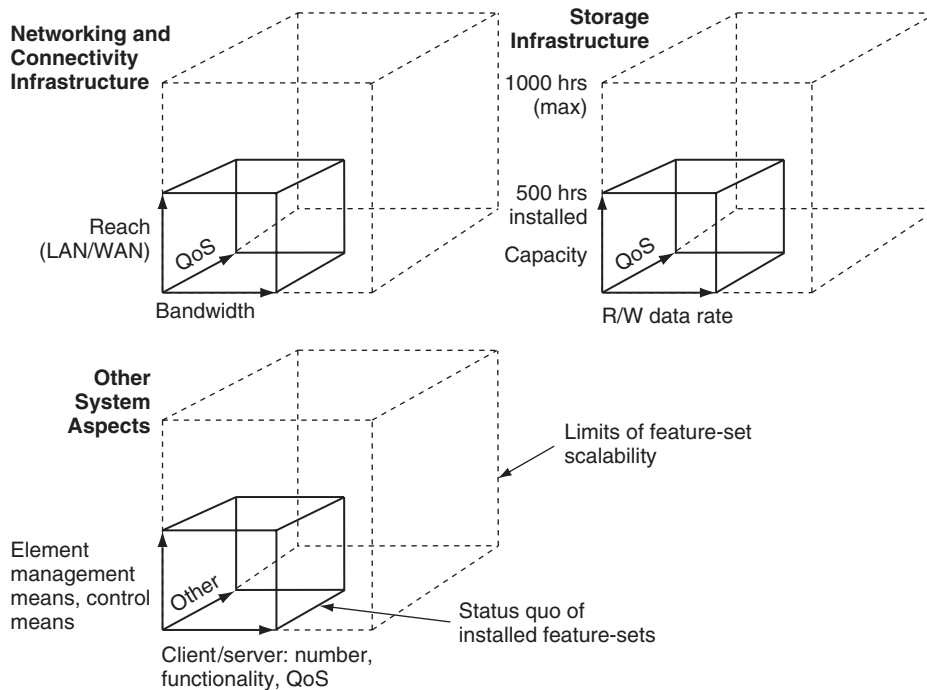
So, the system is working okay and everyone is happy. Then upper management decides to double the number of channels or production throughput or storage or whatever. The first question is, “Can our existing system be scaled to do the job?” The wise systems designer plans for days like this. What are some of the factors that are important in scaling AV/IT systems? Here is a list of the most common scale variables:

1. Scale networking infrastructure
  - a. Network reach—local and long distance
  - b. Routing bandwidth (data rate), number of supported links
  - c. QoS (packet loss, latency, reliability)
2. Scale storage infrastructure (online, near-line, archive)
  - a. Delivery R/W bandwidth
  - b. Capacity (hours of storage)
  - c. QoS (response time, reliability)
3. Scale number of clients/servers, performance, formats (ready for 3D video?), location of nodes
4. Scale control means—automation systems, manual operations, remote ops
5. Scale system management—alarms, configuration, diagnostics, notifications
6. Other—AC power, HVAC, floor space, etc.

Figure 5.18 illustrates each of the scale dimensions as legs of a cube.<sup>2</sup> The smaller volume cubes represent the current or status quo feature sets. The larger volume cubes represent the top limits of the same feature sets. Assume, for example, that the current storage capacity is 500 hr of A/V files but that the maximum practical, vendor-supported size is 1,000 hr. These values are indicated on the storage cubes in Figure 5.18. Generally, exceeding the maximum rate requires an expensive forklift upgrade of some kind. Knowing the maximum dimensions of each large cube sets limits on the realistic, ultimate system scalability. Incidentally, not every possible scale parameter is specifically cited in Figure 5.18. Hence, make sure you identify the key factors that are of value to you when planning overall system scalability.

---

<sup>2</sup> In reality, the shape may not be a cube, but it is a six-sided, 3D volume.



**FIGURE 5.18** Key dimensions of system scalability.

When selecting a system, always seek to understand how each of the dimensions should be scaled. Of course, guaranteeing a future option to scale in one area or another will often cost more up front. If 5TB of storage capacity is sufficient at system inauguration, how much more will it cost for the *option* to expand to 20TB in the future? This cost does not count the actual expansion of 15TB, only the rights to do it later. Paying for potential now may pay off down the road *if* the option is exercised. Sometimes it is worth the cost; sometimes it is not.

It is likely though, at some future time, that some change will be asked for, and the best way to guarantee success is to plan for it to a reasonable degree. When the boss says “Boy, are we lucky that our system is scalable,” you can think, “It is not really luck, it is good design.” Yes, luck is a residue of good design.

### 5.4.1 Upgrading Components

Independent of scaling issues, all systems will need to be upgraded at some point. At the very least, some software security patch will be required, or else you may live in fear of a catastrophe down the road. Then there are performance upgrades, HW improvements, bug fixes, mandatory upgrades, OS service patches, and so on. If system downtime can be tolerated during part of a day or week, then upgrades are not a big deal. However, what if the system



needs to run 24/7? Well, then the upgrade process depends on how the clients, infrastructure, and storage are configured and their level of reliability.

If the principles of NSPOF,  $N + 1$ , replication, and mirroring are characteristics of the system design, then “hot” upgrades (no system downtime, performance not degraded) are possible. It is easy to imagine taking the mirror side out of service temporarily to do an upgrade, removing a client from operation knowing that the spare ( $N + 1$ ) can do the job, or disabling an element while its dual (NSPOF) keeps the business running. In some cases, an element may be upgraded without any downtime. Increasing storage capacity may be done hot for some system designs, whereas others require some downtime to perform the surgery.

Always ask about hot upgrade ability before deciding on a particular system design. In a way, the ability to upgrade is more important than the option to scale. The reason is that upgrades will happen with 100 percent certainty except to the most trivial elements. However, future system scaling may never happen.

## 5.5 IT’S A WRAP—SOME FINAL WORDS

Yes, sometimes things do not always go from bad to worse, but you cannot count on it. Anyone who has been hit hard by Murphy’s Law will know the only way to outwit Mr. Murphy is to hire him as a consultant. Andy Grove of Intel famously said, “Only the paranoid survive.” When it comes to reliable operations, it is okay to be paranoid. Planning for the worst-case scenario is healthy thinking, especially if it leads to worry-free sleep and a system that just keeps humming along.

## References

- Barroso, L. A., et al. (March 2003). Web Searching for a Planet: The Google Cluster Architecture. *IEEE Micro Magazine*, 23(2), 22–28.
- The Mythical Man-Month: Essays on Software Engineering*, Fred Brooks, Addison-Wesley Professional, 2nd edition (August 12, 1995).
- Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, United States Pages: 109–116. Year of Publication: 1988. ISBN:0-89791-268-3.

# Networking Basics for A/V

## CONTENTS

6.0	Introduction	232
6.1	The Seven-Layer Stack	232
6.1.1	Physical and Link Layers	234
6.1.2	The IP Network Layer	237
6.1.3	The Transport Layer—TCP and UDP	244
6.2	Virtual LANS	249
6.2.1	VLAN Basics	249
6.3	TCP/IP Performance	251
6.3.1	Screaming Fast TCP/IP Methods	254
6.3.2	TCP Offload Engines (TOEs)	255
6.3.3	A Clever Shortcut to TCP Acceleration	255
6.4	The Wide Area Network (WAN)	256
6.4.1	WAN Connectivity Topologies	256
6.4.2	Network Choices	258
6.5	Carrier Ethernet (E-LAN and E-Line)	259
6.6	Understanding Quality of Service for Networks	260
6.6.1	Congestion Management	262
6.6.2	QoS Classification Techniques	262
6.6.3	QoS Reservation Techniques	263
6.6.4	The QoS Pyramid	264
6.6.5	MPLS in Action	265
6.7	It's a Wrap—Some Final Words	265
	References	266

6.0 INTRODUCTION

The kingpin of all of IT communication technologies is the network. Fortunately, we live at a time when there is one predominant network, and it is based on the Internet Protocol (IP). Gone are the days when IP and AppleTalk and IBM's SNA and Novell's IPX all competed for the same air. Gone are the days when clumsy protocol translators were required to move a file between two sites. Yes, gone are the days of network chaos and incompatibility. Welcome IP and its associated protocols as the world standard of communications networks. Without IP, there would be no Internet as we know it.

This chapter just touches on the basics of networking principles. The field is huge, and there are countless books and Web references (see [www.ietf.org](http://www.ietf.org) and [www.isoc.org](http://www.isoc.org)) dedicated to every nook and cranny of IP networking. See also (Stallings 2003) and (Stevens 1994) for information on IP and associated Internet protocols. The focus combines an overview of standard networking practices with special attention to A/V-specific needs, such as real-time data, live streaming over LAN and WAN, high bandwidth, and mission-critical QoS metrics. It is the intersection of A/V and networking that is most interesting to us. Let us begin with the classic seven-layer stack.

6.1 THE SEVEN-LAYER STACK

No discussion of networking is complete without the seven-layer stack. Figure 6.1 shows the different layers needed to create a complete end-to-end networking environment. The original Open Systems Interconnection (OSI) model is used

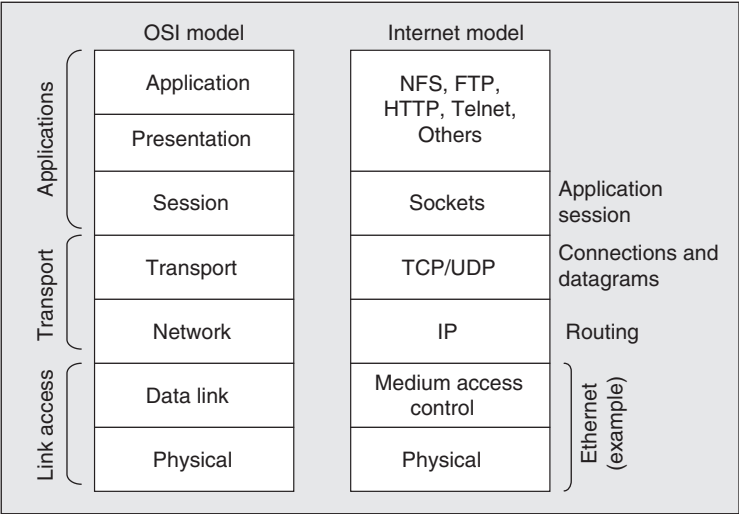


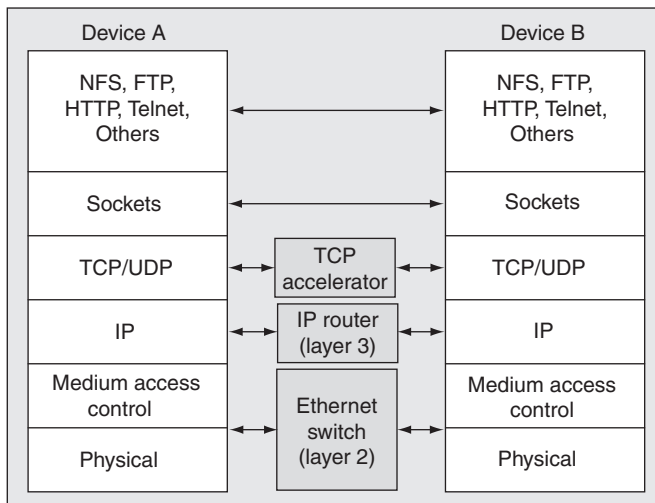
FIGURE 6.1 The OSI model and Internet stack.

as a reference by which to compare the Internet and other protocol stacks. The OSI stack itself may be split into three coarse levels: *link access layer*, *transport layer*, and *applications layer*.

The lowest level of the three coarse layers is the physical and link access (addressing, access methods) layer. The transport layer relates to moving data from one point in the network to another. The application layer is self-evident. The Internet stack does not have an exact 1:1 correspondence to every OSI layer, but there is no requirement for an alignment. Each layer is isolated from the others in a given stack. This is good and allows for implementations to be created on a per-layer basis without concern for the other strata.

Figure 6.2 shows the value of isolated layers. There is a peer-to-peer relationship between each layer on the corresponding stacks. A layer 2 switch (e.g., Ethernet I/O) operates with knowledge of the physical and data link levels only. It has no knowledge of any IP or transport layer activity. Another example is the IP router (or IP switch). It follows the rules of the Internet Protocol for switching packets but is unaware of the levels above or below. At the transport level, a TCP accelerator (HW-based TCP processor) can function without knowledge of the other layers as well.

Of course, some devices may need to comprehend several layers at once, such as a security firewall (or intrusion prevention device) that peeks into and filters data packets at all levels. However, each layer-process may still operate in isolation. The genius of peer-to-peer relationships has enabled stack processors to be designed and implemented independently with a resulting benefit to testing, integration, and overall simplicity.



**FIGURE 6.2** Layer processing using peer-to-peer relationships.



One quick way to remember the order and names of the layers is the acronym PLANTS. From the bottom up it is P (physical), L (link), A (okay, just say and), N (network), T (transport), and S (session). Of course, the presentation/application layers are always at the top.

Let us examine the layers in more detail. At the top of the stack are applications such as FTP, NFS, Web access via HTTP, time-of-day, and so on. The most common applications are available on desktops and servers as standard installs. For a list of all registered applications using well-known ports, see [www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers).

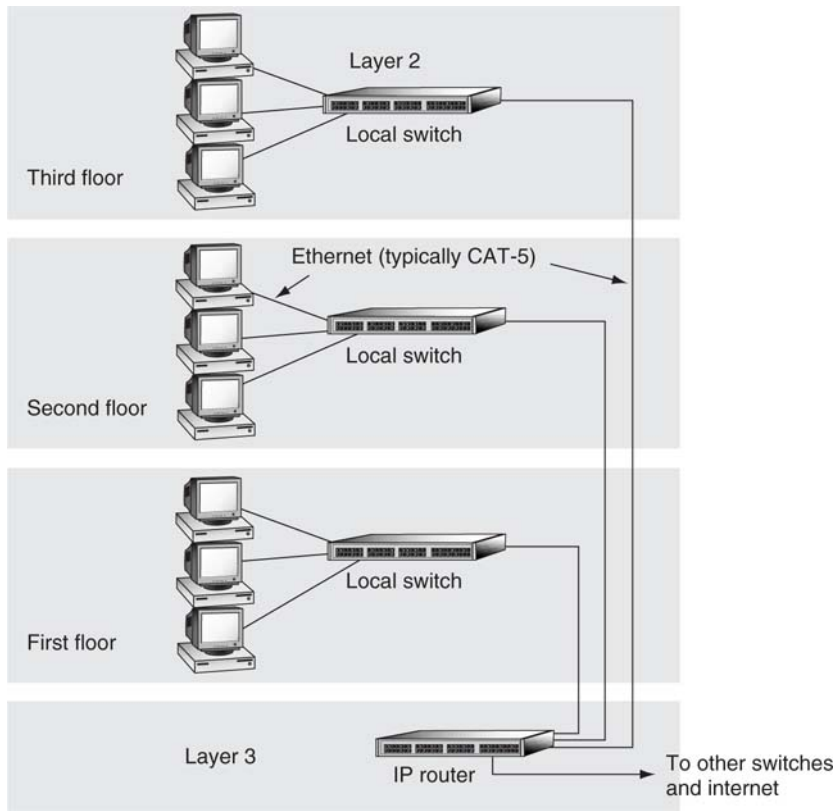
Immediately below the presentation/application layers is the session layer. A session is established between a local program and a remote program using sockets. Each end point program uses a socket transaction to connect to the network. Program data are transferred between end point programs using TCP/IP or UDP/IP. Sockets have their own protocol for setting up and managing the session. From here on, let us study the stack from the bottom up, starting with the physical and link layers.

### 6.1.1 Physical and Link Layers

Ethernet is the most common representation of the bottom two layers for local area network (LAN) environments and has won the war of connectivity in the enterprise IT space. Just a few years ago, it battled against Token Ring and other LAN contenders, but they were knocked out. In addition to LANs, WANs, MANs, satellite links, DSL, and cable modem service provide link layer functionality. WANs and MANs are considered later in this chapter.

One that has become a favorite in the Telco arena is called Packet Over SONET (POS). With POS, IP packets are carried over a SONET link. Another useful IP carrier is an MPEG Transport Stream as deployed in satellite and cable TV networks. Terrestrial transmissions using ATSC, DVB, and other standards also carry IP within the MPEG structure. The common DSL modem packages TCP/IP for carriage over phone lines. Ethernet too has been extended beyond the enterprise walls under the new name Transparent LAN (TLAN). This is a metropolitan area network (MAN) that is Ethernet based. TLANs and SONET are discussed in the WAN/MAN section later in this chapter. For now, let us concentrate on Ethernet as used in the enterprise LAN.

Ethernet was invented in 1973 by Robert Metcalfe of Xerox. It was designed as a serial, bidirectional, shared media system where many nodes (PCs, servers, others) connect to one cable. Sharing a snaked cable among several devices has merit, for sure. Unfortunately, because any one node can hog all the bandwidth, shared media have given way to central switching (star topology). This has become de rigueur for LANs, and each node has a direct line to a switch or



**FIGURE 6.3** Hierarchical switching example.

hub. Figure 6.3 shows a typical star configuration of Ethernet-connected nodes. The star is ideal for A/V networking, as it offers the best possible QoS, assuming that congestion (packet loss) in the switches is low or nonexistent.

Ethernet specs are controlled by the IEEE, and 802.3X is a series of standards for defining the range of links from 10 Mbps to the top of the line 10 Gbps. There are 12+ defined wire/fiber specifications. The most common spec is CAT-5 wiring, which supports 100Base-TX (100Mbps line rate) and 1000Base-T Ethernet. CAT-5 cable is four twisted and unshielded copper pairs. 100Base-TX uses two pairs, and 1000Base-T uses four pairs.

Ten Gbps rates demand fiber connections for the most part, although there is an implementation using twin-axial cable instead of the more common Category 5 cabling. 10G line coding borrows from Fibre Channel's 8B/10B scheme (see Appendix E) as do the first three links in the following list. Common IEEE-defined gigabit Ethernet links are

- 1000Base-LX from 500M (multimode fiber) to 5km (single mode fiber)
- 1000Base-SX from 220M to 550M (multimode fiber)

- 1000Base-CX at 25M copper (82 feet)
- 1000Base-T, copper Cat 5, 100M (327 feet) (five-level modulation)

10G links will not connect to PCs but rather as backbone transport between IP switchers and routers and some servers. There is no reason to stop at 10G rates. The IEEE is standardizing 40 and 100 Gbps rates using both fiber and 10 meter copper. Figure 6.4 gives a history of Ethernet progress. Note that the current development run rate is beating Moore’s law.

6.1.1.1 Ethernet Frames

The on-the-wire format to carry bits is the Ethernet frame, as shown in Figure 6.5. As with most packet protocols, there is a preamble and address field followed by the data payload (1,500 bytes max, nominal) field concluded by an error detection field. So-called jumbo frames carry payloads of sizes >1,500

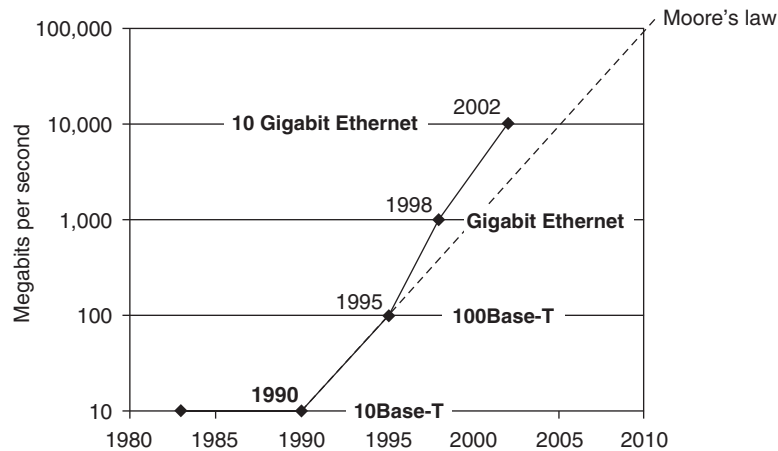
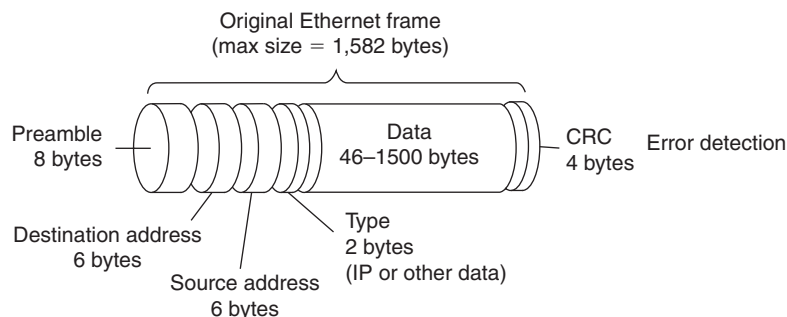


FIGURE 6.4 The evolution of Ethernet.



Destination/Source addresses are called MAC address – 256 Trillion address range  
Ethernet frames ride the wire and carry all upper level data

FIGURE 6.5 The Ethernet frame.

up to 9,000 bytes. Jumbo frames are processed more efficiently with less frame-handling overhead. Each Ethernet port has a worldwide 48-bit Medium Access Control (MAC) address that is unique per port. MAC addressing supports 256 trillion distinct ports. MAC addressing is used for many link types—not just Ethernet. Do not confuse this address with an IP address, which is discussed in the next section. Some network switches use the MAC address to forward frames. This mode of frame routing is often called layer 2 switching. Layer 2 switching is very limited in its reach compared to IP routing, as we will see.

Ethernet frames are sent asynchronously over the wire/link. There is no clock for synchronous switching as there is with a SDI signal. So any real-time streaming of A/V data must account for this. Several commercial attempts have tried to turn Ethernet into a time-synchronous TDMA medium, but they have not succeeded, despite the obvious advantages for A/V transport. See Chapter 2 for more information on using asynchronous links to stream synchronous A/V.

Let us move one layer higher in the stack to the network layer. This layer's entire data field is carried by the Ethernet frame.

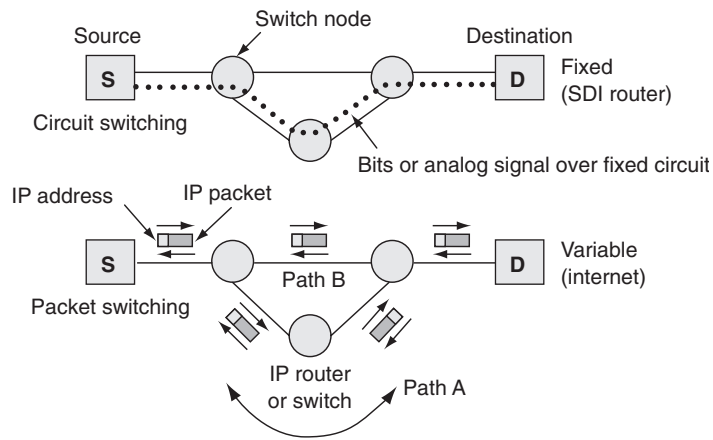
### 6.1.2 The IP Network Layer

The IP network layer is the IP routing layer. There are many ways to route signals (data) from source to destination. One way is called circuit switching, which is typified by the legacy telephone system. In this case, a telephone call is routed via switches to form a literal circuit from source to destination. The circuit stays intact for the duration of the call, and the QoS of the connection is excellent. The traditional SDI router is circuit switched (crossbar normally) and connects input ports to output ports with outstanding QoS. Circuits must be set up by some external control.

On the other hand is packet switching. As an example of packet routing, let us use the analogy of sending a picture postcard of the Chelsea Bridge from London to 385 Corbett Avenue, San Francisco, USA. The postcard (a packet) enters the post office system and is routed via various offices until it reaches its destination. A clerk, or machine, at a London post office examines the destination address and forwards the packet to Los Angeles, USA. Next, the LA post office forwards the packet to the main post office in San Francisco with subsequent forwarding to the local post office nearest Corbett Avenue. At each step the destination address is examined, and the packet is forwarded closer to the final address.

In the end, a letter carrier hand delivers the postcard packet to street address 385. If the destination address cannot be located, then a return message is sent ("return to sender"). It is not uncommon to simultaneously mail two postcards to the same address and have them arrive on different days. One card may traverse via New York City while another traverses via LA, so although the routes are different, the final destination is the same. Welcome to the world of packet routing. This type of routing is often called layer 3 routing. Figure 6.6 shows





**FIGURE 6.6** Example of circuit and packet routing.

examples of circuit and packet-switched methods. Layer 2, MAC switching, is similar to layer 3, but there are differences, as will be shown.

As with the Ethernet frame, each packet has an address field (the IP address) and payload field. Some of the features of a packet routing are as follows:

- Because each packet has a destination (and source) address field inspected at each routing point, packets are self-addressing—not true with circuit switching.
- Policies (routing protocols) decide how to route each packet—via LA or NYC?
- Congestion may cause some packets to be delayed or even dropped—the Christmas card syndrome.
- Packets are not error corrected (at higher stack levels they are, however).
- Packets associated with the same stream may take different routes (hops), resulting in out-of-order packet reception at the receiver (path A or B in Figure 6.6).
- Packets may exhibit jitter (variation in delay) during the life of the stream.
- The QoS is difficult to guarantee in large networks.
- Packets are carried by the lower two layers in the stack (e.g., Ethernet frames).

As a result, packet switching lacks some of the more A/V-friendly features of circuit switching. Despite this, packet switching offers self-addressing, resiliency to router failure, wide area routing, and IT-managed and relatively inexpensive

switches. The best success story for packet switching is the Internet. Imagine building the Internet from circuit-switched elements. Some entity would need to open/close every switch point, and this alone signals disaster for such a topology.

The biggest issue of using packets (compared to circuit switching) to move A/V data is a potentially low or unspecified QoS. As discussed in Chapter 2, there are clever ways to smooth out any packet jitter; using TCP (next layer in the stack), any packet errors may be 100 percent corrected. So the only real issue is overall latency, which may be hidden or managed in most real-world systems.

Over small department networks, the latency through several switch hops may be well controlled. It is possible to achieve a  $<30\text{-}\mu\text{s}$  end-to-end latency for such a network. This is less than one raster line of video in length. As a result, real-time streaming and device control using routed packets is practical. Of course, a  $30\text{-}\mu\text{s}$  end-to-end delay is not common for most networks, but the principle of low-latency networks is well established.

### 6.1.2.1 Comparing Layer 2 and Layer 3 Switching

This section sorts out some of the pros and cons of layer 2 and layer 3 switching. In many enterprise networks, switches route Ethernet frames using the MAC address (layer 2 switching) or packets using the IP address (layer 3 switching). Many commercial switches support both methods. See Figure 6.3 for a simple network using both layer 2 and 3 switching. Medium to large network domains use a mix of layers 2 and 3, whereas smaller networks or LAN subgroups use only layer 2 methods. There are trade-offs galore between these two methods, but the main aspects are as follows:

1. Layer 2 switching domain
  - Switching is based on MAC address in Ethernet frame.
  - It supports small/medium LAN groups or VLANs that confine broadcast messages but provides limited scalability.
  - It offers excellent per-port bandwidth control.
  - Path load balancing is not supported.
  - Spanning Tree Protocol (STP) supports path redundancy while preventing undesirable loops in a network that are created by multiple active paths between nodes. Alternate paths are sought only after the active one fails.
  - It is easy to configure and uses lower cost switches than layer 3.
2. Layer 3 switching (routing) domain
  - Switching is based on the IP address.
  - It scales to large networks—departments, campus networks, the Internet.
  - It routes IP packets between layer 2 domains.
  - Redundant pathing and load balancing are supported.

- It is able to choose the “least-cost” path to next switch hop using Open Shortest Path First (OSPF) or older Routing Information Protocol (RIP) routing protocols. OSPF has faster failover (<1 s possible) than STP.

Layer 2 and 3 switching can live together in complete harmony. It is quite common for portions of a network design to be based on layer 2 switching while other portions are based on layer 3. For A/V designs, layer 2 can offer excellent QoS at the cost of slow failover if a link or node fails. STP can take 30–45 s to find a new path after a failure is detected. For this reason, RSTP is sometimes used. Rapid STP, based on IEEE standard 802.1 W for ultrafast convergence, is ideal for A/V networks. See (Spohn 2002) for a good summary of layer 2 and 3 concepts and trade-offs. See also (DiMarzio 2002) for a quick summary of routing protocols of concepts or scour the Web for information.

Another aspect of layer 2 is deterministic frame forwarding. In most cases, Ethernet frames will traverse links according to the forwarding tables (built using STP) stored in each switch. The packet forwarding paths are static,<sup>1</sup> until a link or switch fails. Then, if possible, a new path will be discovered by STP, and packet forwarding continues. In a static network, for example, media clients accessing NAS storage should see a fixed end-to-end delay plus any switch-induced jitter for each Ethernet frame.

However, layer 3 can route over different paths to reach the same end point. Because routing occurs on a per-packet basis, this will likely introduce added jitter above what layer 2 switching introduces. See Figure 6.6 for an example of alternate pathing (path A or path B may be used as determined by routing protocols) using layer 3 routing. See Chapter 5 on how to build fault-tolerant IP routing networks. In the end, judicious network design is needed to guarantee a desired QoS and associated reliability.

The primary difference between a *layer 3 switch* and a *router* depends more on usage than features. Layer 3 switching is used effectively to segment a LAN rather than connect to a WAN. When segmenting a campus network,

for instance, use a router rather than a switch. When implementing layer 3 switching and routing, remember: *Route once, switch many.*



### 6.1.2.2 IP Addressing

Just as a house has a street address, networked devices have IP addresses. The IP (IPv4) address is a 4-byte value. The address range is split into three main classes: A, B, and C. Each class has a subdivision of *network ID* and *host ID*. What does this mean? A network ID is assigned to a company or organization

<sup>1</sup> It is possible that switch forwarding tables may change after a power failure or some network update, so there is no guarantee that a given L2 forwarding path between selected end points will be permanent.

depending on the number of host IDs (nodes on their network) (see Figure 6.7). Classes A, B, and C are for point-to-point addressing. A special class D address is reserved for point-to-multipoint data transfer.

For example, MIT has a class A network ID (netid) of 18, and the campus can directly support 16 million hosts ( $2^{24}$ ). A smaller organization may be assigned a class C address that supports only 256 hosts. Many addresses have an equivalent network name as well. For example, MIT's IP Web address is `http://18.7.22.83` (in so-called dotted decimal notation) and its equivalent name is `http://www.mit.edu`. A network service called a Domain Name Server (DNS) is available to look up an address based on a name. Numeric addresses, not names, are needed to route IP packets.

To better understand IP addressing, DNS address lookup, ping (addressed device response test), trace route (a list of hops), and other network-related concepts, visit [www.dnsstuff.com](http://www.dnsstuff.com) for a variety of easy-to-run tests. Ten minutes experimenting with these tests is worth the effort. Note that it takes time (<100Ms usually) to look up an IP address based on a name. In time-critical A/V applications, it is often wise to use numeric addresses and avoid the DNS lookup delay. As a result, `http://www.mit.edu` takes slightly longer to reach than `http://18.7.22.83` if the named address is not already cached for immediate use.

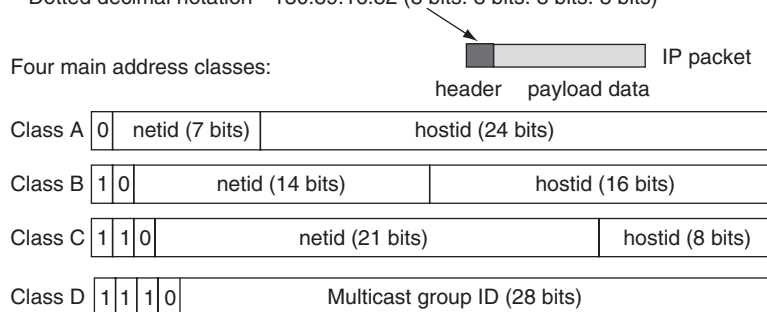
### 6.1.2.3 Subnets

A class A address space supports 16 million hosts. All the hosts on the same physical network share the *same broadcast traffic*; they are in the same broadcast domain. It is not practical to have 16 million nodes in the same broadcast

IP addressing;

- 32 bits, 4 billion hosts, very limited (IPv4)
- Dotted decimal notation --130.89.16.82 (8 bits. 8 bits. 8 bits. 8 bits)

Four main address classes:



Class A: Govt, HP, IBM – ½ of all IP addresses in this class (128 netids, 16 M hosts each)

Class B: campus, medium size companies (16 K netids, 64 K hosts per netid)

Class C: (2 million netids each with 254 host addresses)

Class D: Multicast groups (268 million groups)

**FIGURE 6.7** IP addressing concepts.

domain. Imagine 16 million hosts broadcasting data packets! Now that is a data storm. The result is that most of the 16 million host addresses are not usable and are wasted. Even a pure class B network with 64K hosts is impractical.

As a result, the goal is to create smaller broadcast domains—to wall off the broadcast traffic—and to better utilize the bits in the host ID (hostid) by subdividing the IP address into smaller host networks. The basic idea is to break up the available IP addresses into smaller subnets. So a class B host space may be divided into say 2,048 (11 bits of host ID) subnets each with  $\sim 32$  hosts (5 bits of host ID)—a very practical host size. In reality, the host ID cannot be perfectly subdivided to utilize every possible host address. Still, subnetting is a practical way to build efficient networks. Each subnet shares a common broadcast domain, and each is reachable via IP, layer 3, and switches/routers that bridge domains.



**Layer 2 Versus Layer 3 Switching:** Layer 2 switching uses the MAC address in the Ethernet frame to forward frames to the next node. Layer 3 switching uses the IP address in the IP packet to forward packets.

#### 6.1.2.4 *IPv6 and Private IP Addresses*

No doubt about it, the Internet is running out of IP addresses. The day is near when every PC, mobile phone, microwave oven, and light switch (or even light bulb) will require an IP address. There are two solutions to this problem. One is to migrate to the new and improved version of IP, IPv6 (RFC 2460). Among other valuable enhancements, each IP packet has a 128-bit address range, which is  $\sim 10^{38}$  addresses. This is equivalent to 100 undecillion<sup>2</sup> addresses. There are an estimated  $10^{28}$  atoms in the human body, so IPv6 should suffice for a while. IPv6 is slowly being adopted and will replace IPv4 over time. There are transition issues galore, as may be imagined. One transition scenario is to support dual stacks—IPv4 and IPv6—in all network equipment. This is not commonly done but may become so as IPv6 kicks into gear.

A more common solution for living with the limited IPv4 address space is to use the Network Address Translation (NAT) method. Several addresses have been set aside for private networks as listed in Table 6.1. These addresses are never routed on the open Internet but only in closed, private networks.

The NAT<sup>3</sup> function is similar to what a telephone receptionist does. The main office number is published (a public IP address), but the internal phone network has its own extension numbering plan (private IP addresses) not

<sup>2</sup> An undecillion is  $10^{36}$ .

<sup>3</sup> NAT is often referred to as IP masquerading.

**Table 6.1** Dedicated Private IP Address Ranges

Class	Private Start Address	Private Finish Address
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

directly accessible from the outside. The operator routes incoming calls to the correct extension by doing address and name translation. Because the private IP addresses are never routed on the open Internet, they may be reused as often as needed in private networks just as phone extension numbers are reused by other private phone systems.

NAT has effectively added billions of new virtual IP addresses, which has stalled the uptake of IPV6. Many companies use NAT services and rely on pools of internal, private IP addresses for network nodes. Many modern A/V systems (playout servers, news production systems, edit clusters) also use private IP addresses. NAT uses several methods to map internal private to external public IP addresses. To learn more, see <http://computer.howstuffworks.com/nat1.htm>.

The IP layer is replete with protocols to assist in routing packets over the open terrain of the Internet. For the most part, they do not influence A/V networking performance, so they are not covered. There is one exception: QoS. Network QoS is governed by several net protocols, which are reviewed later in this chapter. IP multicast is useful when streaming an IP broadcast to many end stations. The next section outlines the basics.

#### 6.1.2.5 IP Multicasting

Multicasting is a one-to-many transmission, whereas the Internet is founded on unicast, one-to-one communications. Plus, multicasting is normally unidirectional, not bidirectional, as with, say, Web access. Multicast file transfers are not common, but there are ways to do it, as discussed in Chapter 2. See also [www.tibco.com](http://www.tibco.com) for a variety of file distribution solutions to many simultaneous receivers.

The IP class D address is reserved for multicast use only. In this case, each host ID is a multicast domain, like the channel number on a TV. Any nodes associated with the domain may receive the IP broadcast. IP multicast is a suite of protocols defined by the IETF to set up, route, and manage multicast UDP packets. Most multicast is *best effort* packet delivery, although it is possible to achieve 100 percent transfer reliability. This is not common and becomes complex with a large number of receivers.

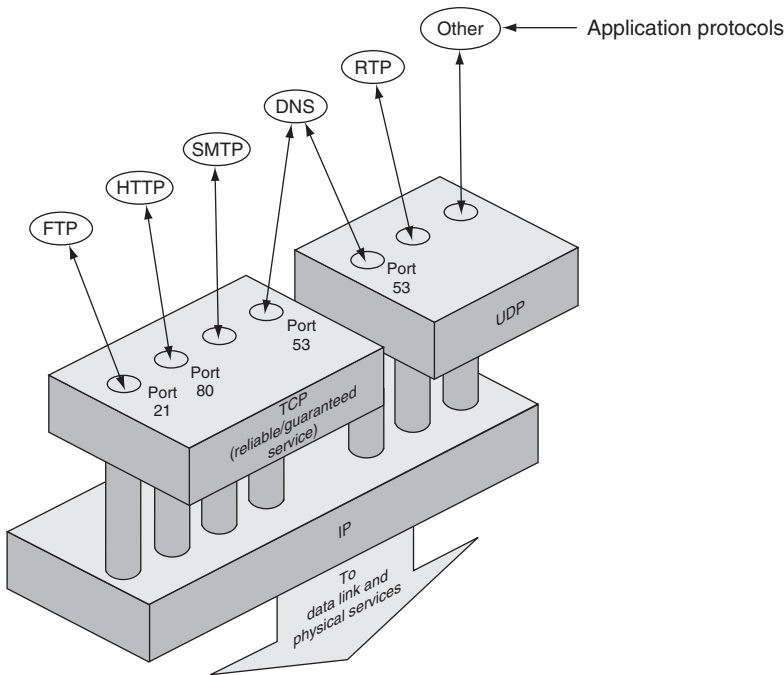
The key to multicasting is a multicast-enabled routing system. Each router in the network must understand IP multicast protocols and class D addressing. IP packets are routed to other multicast-enabled routers and to end point

receivers. Any node that tunes into an active class D host address will be able to receive the stream.

The Internet in general does not support IP multicasting for a variety of reasons. The protocols are complex, and there is no easy way to charge for multicast packet routing and bandwidth utilization. Imagine a sender who establishes a multicast stream to one million receivers that span 100 different Internet service providers. The business and technical challenges with this type of broadcast are intricate, so ISPs avoid offering the capability. However, campus-wide multicast networks are practical and in use for low bit rate streaming video applications. There is very little IP multicast used for professional A/V production. Next, let us focus next on the granddaddy of all protocols, TCP and its cousin UDP.

6.1.3 The Transport Layer—TCP and UDP

Transmission Control Protocol (TCP) is a subset of the Internet Protocol suite often abbreviated as TCP/IP. TCP sits at layer 4 in the seven-layer stack and is responsible for reliable data communications between two devices. Figure 6.8 provides a simple view of TCP’s relation to application-related protocols, UDP, and lower levels. Consistent with stack operations, TCP packets are completely carried as payload by IP packets. TCP supports full duplex, point-to-point communications.



**FIGURE 6.8** TCP and UDP in relation to application protocols.  
Concept: Encyclopedia of networking and telecommunications.

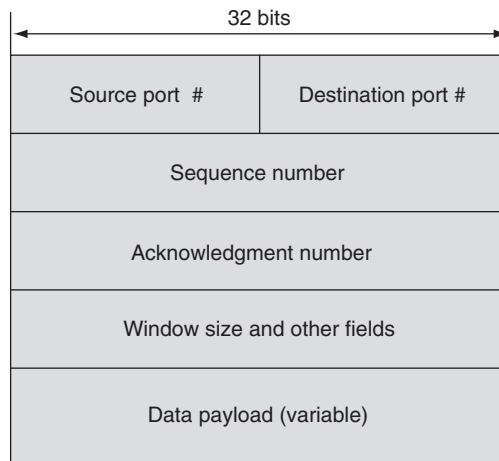
The TCP packet format is described in Figure 6.9. Some notable characteristics are as follows:

- No address field
- Port numbers used to distinguish application-layer services
- Sequence number
- Acknowledgment ID
- Window size
- Data payload—actual user data such as files

As Figure 6.8 shows, port numbers identify services. Many different TCP connections may exist simultaneously, each associated with a different application. For example, well-known port 21 is dedicated to FTP and 80 to HTTP for Web page access. There are 64K ports available, some assigned to specific services. Registered applications use what are called well-known ports for access.

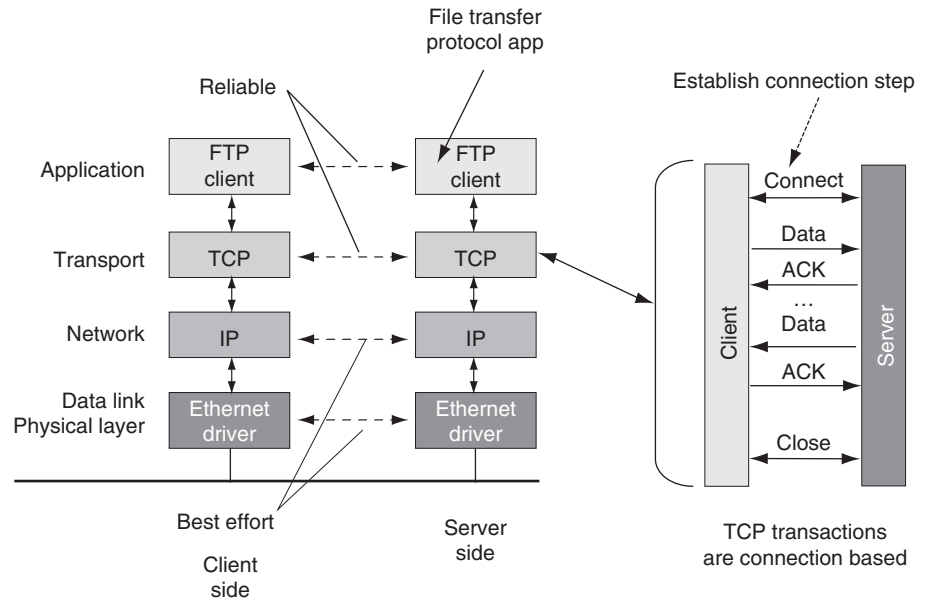
TCP is a connection-oriented protocol. This means that a handshake protocol is used to establish a formal communication between two devices *before* any payload data are exchanged. End point connections are called sockets. For example, when you are connecting to a Web server, a TCP connection is first established before any Web pages are downloaded. Figure 6.10 illustrates steps needed to move a file between a server and a client using FTP and TCP.

TCP connection establishment is a simple three-step sequence and is done only at the beginning of the call. Then file data are moved between the sides. Importantly, TCP requires that *every* packet be positively acknowledged so that the sender knows with certainty that a sent packet was received without error. Because the setup phase does consume a small amount of time, A/V-centric applications may decide to leave the connection established ready for future use. For short transactions, the setup/close can take >50 percent of the total connection time.



**FIGURE 6.9** General layout of a TCP packet.





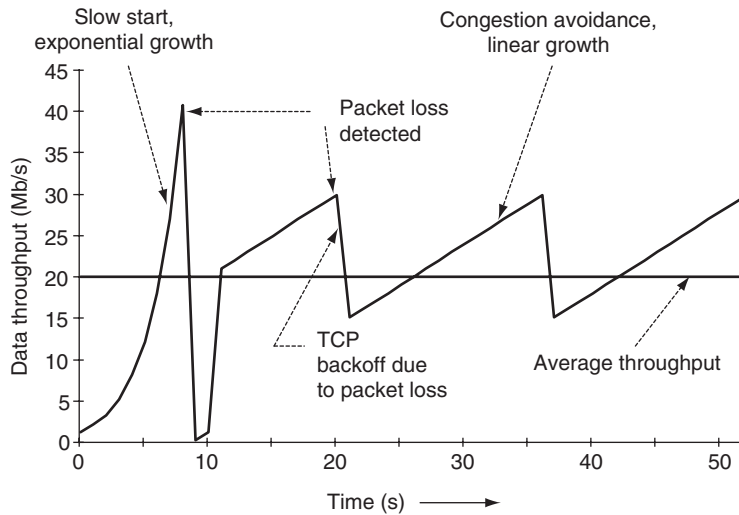
**FIGURE 6.10** TCP is connection based and 100 percent reliable.  
Concept: CISCO.

If a packet's acknowledgment ID is not received within a certain time period, then the suspect packet is re-sent. TCP will reduce its sending rate if too many packets are lost. This behavior helps reduce network congestion. TCP is a good network citizen. Positive acknowledgments and rate control are major features of TCP and have been both a curse and benefit to A/V data transfer performance. Figure 6.11 illustrates an example of sustained TCP performance in the presence of packet loss. Note the aggressive backoff and slow startup. The throughput would be constant only if packet loss was effectively zero. Although an IP packet can carry up to 64KB of payload data, when Ethernet is the underlying link layer, it is wise to limit IP data length to 1,500B—the Ethernet frame payload size. If a data bit error occurs at the frame or packet level, 1,500B of payload is lost for the general case.

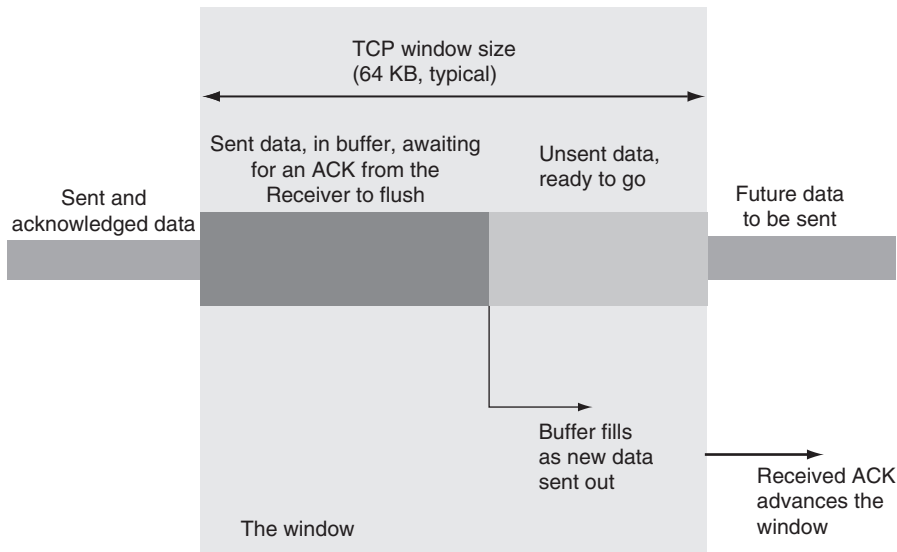
### 6.1.3.1 The Sliding Window

TCP uses what is called a sliding window approach to manage transmission reliability and avoid congestion, as illustrated in Figure 6.12. There are three kinds of payload data in the vocabulary of TCP:

1. Sent and acknowledged (ACK) data packets; the receiver has the data
2. Sent and awaiting an ACK from the receiver; the receiver may not yet have the data
3. Data packets not yet sent



**FIGURE 6.11** Behavior of TCP in the presence of packet loss.  
Concept: Cisco



**FIGURE 6.12** TCP's sliding data window.

Data that fall into case #2 are governed by the sliding window. In most cases, the TCP window is 64KB, although RFC 1323 allows for a much larger window with a corresponding increase in transfer rates over long-distance links. When the sender transmits a data packet, it waits for the receiver to acknowledge it. All unacknowledged sent data are considered “in the window.” If the window becomes full, 64KB of outstanding data, then the sender stops transmitting data until the next ACK is received.

For short-distance hops, the window does not impair performance because ACKs are received quickly. For long-distance transfers (across WANs, satellites), small windows contribute to slow FTP rates because the transmission pipe fills quickly with unacknowledged data. More on this later in the chapter.

Despite some performance problems with TCP, it is the king of the transport layer. How does it compare to User Datagram Protocol (UDP), its simpler cousin? Let us see.

### **6.1.3.2 UDP Transport**

In basic terms, UDP is a send-and-hope method of transmitting data. There are no connection dialogs, acknowledgments, sequence numbers, or rate control; UDP just carries payload data to a receiver port number. A UDP packet is launched over IP and, if all goes well, arrives at the receiver without corruption. UDP is a connectionless protocol compared to TCP being connection based.

Who would want to use UDP when TCP is available? Well, here are a few of UDP's advantages:

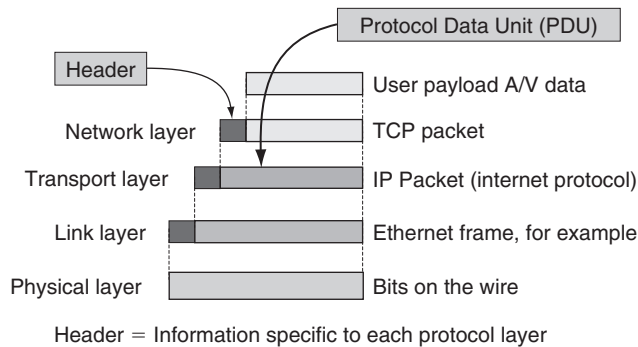
- UDP is very easy to implement compared to TCP.
- It has almost no software overhead and is very CPU efficient.
- It provides efficient A/V streaming (VoIP uses UDP and RTP to carry voice data for a call).
- There is no automatic rate control as with TCP; transmission metering can be set as needed.
- There is minimal delay from end to end.
- It supports point-to-multipoint packet forwarding (IP multicast).

If the network is not congested and application data are somewhat tolerant of an occasional packet loss, then UDP is an ideal transport mechanism. In fact, UDP is the basis for many real-time A/V streaming protocols. When you listen to streaming music at home over the Internet, UDP is often the payload carrier.

When UDP is coupled with custom rate control, it can outperform TCP. Some UDP-based file transfer protocols use TCP only to request a packet resend and set rates. Although TCP is part of the transaction, its use is infrequent and highly efficient. Some A/V streaming applications use error concealment to hide an infrequent missing packet. See Chapter 2 for examples of both UDP- and TCP-based file transfer applications.

### **6.1.3.3 Stacking It All Up**

The stack is a good way to organize the concepts and interactions of IP-related standards. The peer-to-peer relationship of the layers is an excellent way to divide and conquer a complex set of associations. Figure 6.13 summarizes how



**FIGURE 6.13** *Packet encapsulation.*

packets are encapsulated by the layer above. In the simplest form, each packet type is a header followed by a Protocol Data Unit (PDU). The IETF has set standards for each of these layers and their corresponding packet format. As trillions of packets transit the Internet and private networks each day, the IP stack has proven itself worthy of respect.

LANs are built out of the fabric of the stack, but not all LANs are created equal. Next, the VLAN is considered.

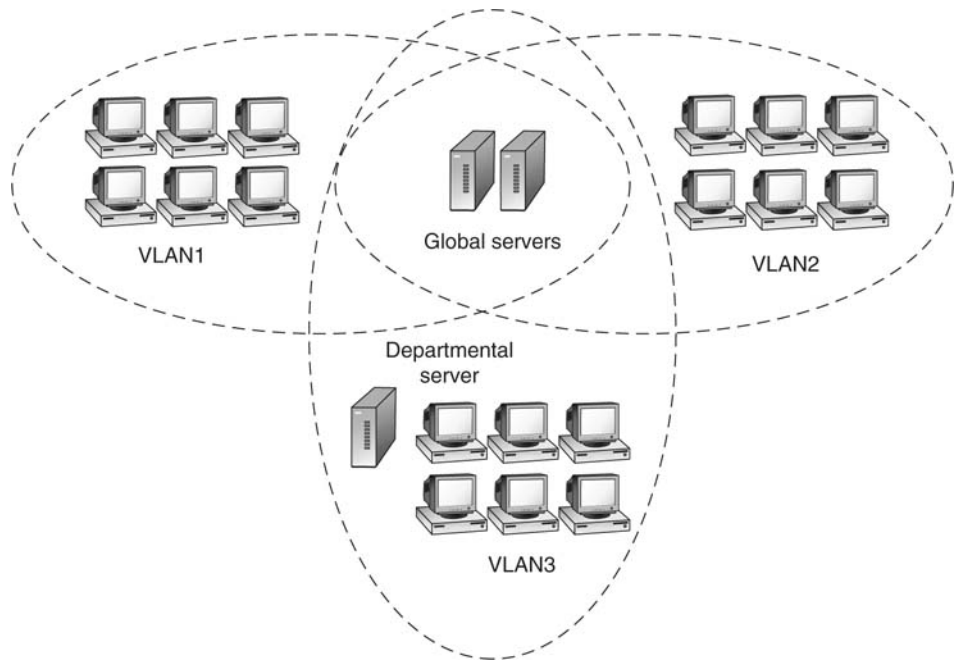
## 6.2 VIRTUAL LANS

One huge, flat network easily interconnects all attached nodes. At first blush, this may seem like the ideal topology. However, dividing it into smaller subnet domains offers better QoS, reliability, security, and management. Using a VLAN is a practical method to implement the segmentation. With a VLAN, the A/V domain may be on one LAN, sales on a second LAN, human resources on a third, and so on. Segmenting LANs is the ideal way to manage the network resources of each department or domain. Figure 6.14 illustrates the division of LANs. Especially important for A/V applications is the isolation between LANs afforded by VLANs.

During normal LAN operation, various layer 2 broadcast messages are sent to every member of a LAN. With VLANs, these broadcast messages are forwarded only to members of the VLAN. VLAN node isolation is a key to its performance gain. The IEEE has standardized 802.1Q for VLAN segmentation. The following section covers the layout and advantages of VLANs over traditional LANs.

### 6.2.1 VLAN Basics

In a traditional Ethernet LAN, nodes (PCs, servers, etc.) connected to the same layer 2 switch share a domain; every node sees all broadcast frames transmitted by every other node. The more nodes, the more contention and traffic overhead



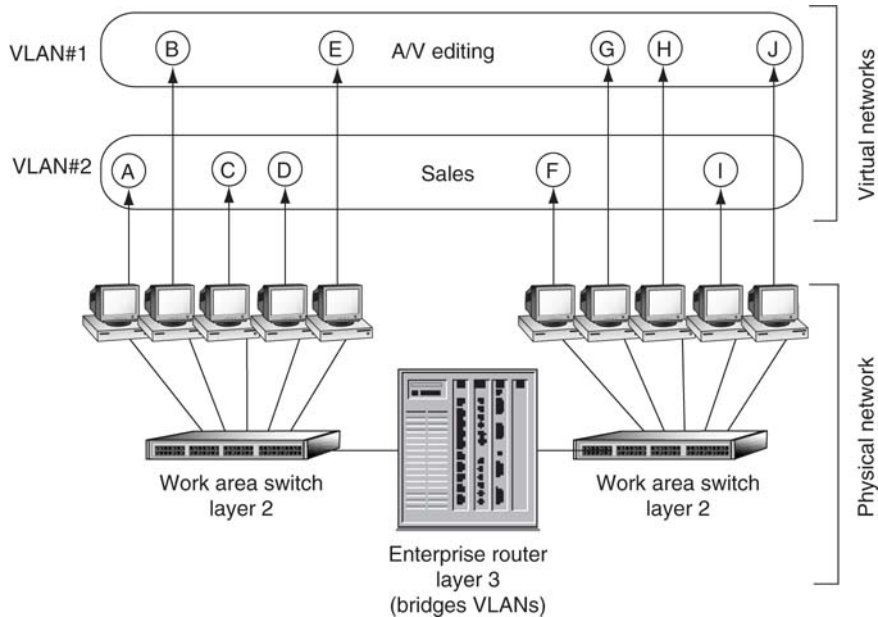
**FIGURE 6.14** *A network of isolated VLANs.*

are present. LAN QoS falls as the number of nodes increases. To avoid poor performance, the LAN must be decomposed into smaller pieces.

Nodal segmentation can be done by throwing hardware at the problem. Connect one set of stations to switch A, another to switch B, and so on and performance increases. This has the problem that associated nodes need to be in the same proximity. Is there a smarter way to segment a LAN? Yes.

VLANs provide logical isolation in place of physical segmentation. A VLAN is a set of nodes that are treated as one domain regardless of their physical location. A VLAN can span a campus or the world. Stations in VLAN #1 hear other stations' traffic in VLAN #1, but do not hear stations in other VLANs, including those connected to the same switch. This isolation is accomplished using VLAN tagging (see Figure 6.15). A VLAN tag is a 4-byte Ethernet frame extension (layer 2) used to separate and identify VLANs. Importantly, a VLAN's data traffic remains within that VLAN and can cross outside only with the aid of a layer 3 switch/router. Segmentation is especially valuable for critical A/V workflows where traffic isolation is needed for reliable networking and achieving a desired QoS.

For example, a layer 2 switch may be configured to know that ports 2, 4, and 6 belong to VLAN #1, whereas ports 3, 5, and 7 belong to VLAN #2, and so on. The switch sends out arriving broadcasts to all ports in the same VLAN, but never to members of other VLANs.



**FIGURE 6.15** VLAN segmentation example.  
*Concept: Encyclopedia of networking and telecommunications.*

The following are VLAN advantages for a domain of A/V application clients:

- QoS is improved for A/V VLAN segments.
- An A/V client may have two Ethernet ports, one per VLAN. With two VLAN attachments per device, it is possible to access VLAN #2 if VLAN #1 has failed. This is key to some HA dual pathing methods discussed in Chapter 5.
- Network problems on one VLAN do not necessarily affect a different VLAN. This is needed when the A/V network needs separation from, say, a business LAN.
- A VLAN has more geographical flexibility than with IP subnetting.

VLANs are not the only way to improve performance, security, and manageability. The next section outlines some protocols designed for setting and maintaining QoS levels.

## 6.3 TCP/IP PERFORMANCE

TCP has built-in congestion control and guaranteed reliability. These features limit the maximum achievable transfer rate between two sites as a function of the advertised link data rate, round trip latency, and packet loss. As discussed earlier, TCP's sliding window puts a boundary on transfer rates. TCP's average

data throughput is given by the following three principles; the one with the lowest value sets the data rate ceiling.

**TCP limitation 1.** You cannot go faster than your slowest link.

- If the slowest link in the chain between two communicating hosts is limited to  $R$  Kbps, then this is the maximum throughput.

**TCP limitation 2.** You cannot get more throughput than your window size divided by the link's round trip time (RTT).

- RFC 1323 does a good job of discussing this limitation, and TCP implementations that support RFC 1323 can achieve good throughput even on satellite links if there is very little packet loss. An RTT of 0.1 s and a window size of 100KB yield a maximum throughput of 1 MBps.

**TCP limitation 3.** Packet loss combined with long round trip time limits throughput.

- RFC 3155, "End-to-End Performance Implications of Links with Errors," provides a good summary of TCP throughput when RTT and packet loss are present. In this case the following approximate equation provides the limiting transfer rate in bytes per second.

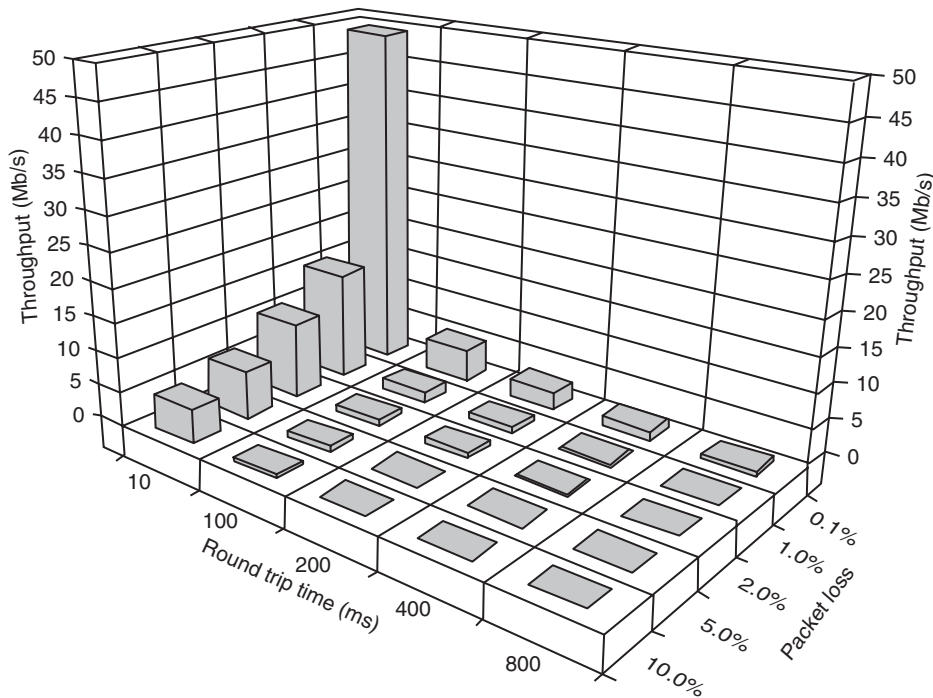
Throughput limit =  $1.2 \times (\text{Packet\_Size}) / (\text{RTT} \times \text{SQRT}(\text{packet loss probability}))$

Note that TCP's data throughput depends on link bandwidth only for rule #1. Rule #2 limits data throughput due to window size and round trip time (RTT), and rule #3 limits data rate due to RTT and packet loss. Knowing *why* a transfer is slow gives the hints needed to improve the transfer performance.

These TCP limitations are clearly illustrated in Figure 6.16. The bar graph shows the maximum continuous data throughput achievable under various RTT and packet loss conditions using a SONET OC-3 (155Mbps) link. The throughput sensitivity to RTT is obvious. Increasing the RTT from 10 to 100MS decreases the rate by more than 10×! This is precisely why throwing "bandwidth" at a slow file transfer is often a waste of money.

Sensitivity to loss is not as severe. Packet loss going from 0.1 to 1 percent decreases TCP throughput by about a factor of 3×. So, a transit path with 0.1 percent loss and 200MS of RTT (SF to/from London's [www.bbc.co.uk](http://www.bbc.co.uk)) would permit data rates ~1Mbps. This is <1 percent of the maximum achievable data rate. In fact, the transfer rate may be worse if, using the Internet, loss exceeds 0.1 percent or approximately 3Mbps if at 0.01 percent loss. Of course, the Internet has an undefined QoS, so user beware.

Speedy rates are attainable in local LANs with small RTT (<10MS) and loss <0.01 percent. So when you are doing system planning, knowing the RTT and loss characteristics of a transmission link will allow you to better predict throughput.



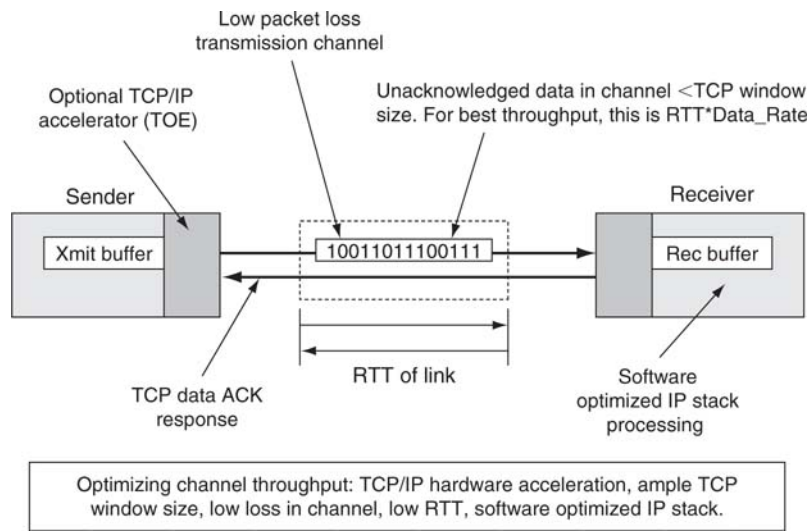
**FIGURE 6.16** TCP data throughput versus round trip time (RTT) and loss.

Techniques for improving TCP's throughput are numerous. Some methods use received error correction (FEC), and some use fine-tuning to adjust the myriad of TCP parameters for more throughput. Figure 6.16 was provided by Aspera, Inc. ([www.asperasoft.com](http://www.asperasoft.com)). This company offers a method (*asf*) using UDP and a return channel to achieve adaptive rate control and realize throughput rates approaching the line rate even in the presence of large RTT values and significant loss. Aspera's strategy does not use TCP, so both end points must use the non-standardized Aspera technology.

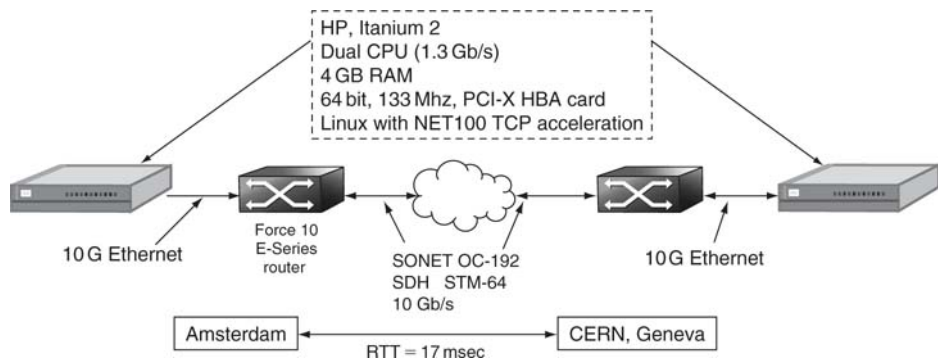
Naturally, the CPU stack processing of TCP/IP packets may also bottleneck performance, despite the efforts to optimize the three factors given earlier. There are countless ways to improve the stack's software performance, and many TCP processors use clever methods to avoid data copies and so on. Another way to improve TCP's compute performance is hardware acceleration with a TCP Offload Engine (TOE) card discussed later. A TOE card moves stack processing away from the main CPU to a secondary processor dedicated to TCP/IP processing. Figure 6.17 illustrates chief twiddle factors for improving TCP's performance.

So why do we use TCP at all if it has such limitations? All transfer protocols have less than ideal characteristics for some portion of their operational range.





**FIGURE 6.17** *Optimizing TCP/IP throughput.*



**FIGURE 6.18** *High-speed transmission test configuration.*

Some researchers have postulated that the relative stability of the Internet is at least partially attributed to TCP's aggressive back-off gentle slow startup under congestion (Akella 2002). So TCP is a good network citizen. See Chapter 2 for a list of methods to accomplish fast file transfer without using TCP.

### 6.3.1 Screaming Fast TCP/IP Methods

Researchers at the University of Amsterdam (Antony 2003) did an end-to-end file transfer experiment with the test conditions shown in Figure 6.18. They used high-end servers with 10G Intel Host Bus Adaptor (HBA) cards. These are not TOE cards. The servers ran Linux with specialized TCP stack software called TCP Vegas. One end point was in Amsterdam, and the other was in Geneva. Each 10G Ethernet LAN connected to a STM-64 10-Gbps WAN

(SONET) using a Force 10 router. The round trip delay between sites was only 17MS. Using only CPU TCP processing, the throughput reached 5.22 Gbps (about half of the 10G user payload) after proper tweaking of the TCP window size (socket buffer size was the adjustable parameter).

What limited the throughput performance? The researchers believe it was the internal PCI-X bus bandwidth. Also, if the WAN link had any congestion, the rate would have dropped precipitously. User data were R/W to RAM, not to HDD devices, so memory speeds were not an issue. The end device servers are high end and expensive. With a TOE card to accelerate the stack, the performance will increase. The next section discusses TOEs.

### 6.3.2 TCP Offload Engines (TOEs)

A TOE card is a server or workstation plug-in card (PCI-X or similar, or on a motherboard) that offloads the CPU-intensive TCP/IP stack processing. To obtain the fastest iSCSI, NAS, or file transfers, you may need a TOE card. The card has an Ethernet port, and all TCP/IP traffic passes through it. At 100Mbps Ethernet speeds, most CPUs can handle the processing overhead of TCP. A generally accepted rule of thumb is that a CPU clock of 1Hz can process the TCP overhead associated with transferring data at 1bps. With the advent of 10G Ethernet, server and host CPUs are suffocating while processing the TCP/IP data packets.

The research firm Enterprise Strategy Group ([www.enterprisestrategygroup.com](http://www.enterprisestrategygroup.com), Milford, MA) has concluded that “implementing TCP off-load in hardware is absolutely a requirement for iSCSI to become mainstream. TCP is required to guarantee sequence and deal with faults, two things block-oriented storage absolutely requires. Running TCP on the server CPU will cripple the server eventually, so bringing the function into hardware is a must.” TOE cards are used for some 1G and most 10G iSCSI host ports. Many iSCSI storage vendors use TOE cards.

### 6.3.3 A Clever Shortcut to TCP Acceleration

WAN transfer-speed acceleration is a proven concept given a WAFS appliance (see Chapter 3B), or similar, at each end point. But, can TCP be accelerated with a WAFS-like appliance located at only one end point? Surprisingly, yes. Researchers at Caltech, Pasadena (Cheng Jin 2004) have invented methods, collectively termed FastTCP, to accelerate TCP using only an appliance at the sending side. The receiving side(s) uses unmodified, industry standard TCP.

Using a combination of TCP-transmitted packet metering, round trip delay measurement, and strategies to deal with packet loss, a single appliance can achieve up to  $32\times$  throughput compared to no acceleration. The methods really shine when the end-to-end path has  $>100$ Ms delay,  $>0.1$  percent loss, and the pipe is large,  $>5$ Mbps. A link with these characteristics is often called a *long fat pipe*.

FastTCP has been implemented by FastSoft ([www.fastsoft.com](http://www.fastsoft.com)) in its Aria appliance. FastTCP currently holds the world's record for TCP transfer speeds with a sustained throughput of 101 Gbps. Imagine what FastTCP can do for transferring large video files over long distances using lossy Internet pipes. Plus, Aria supports up to 10,000 simultaneous connections: think Web servers.

## 6.4 THE WIDE AREA NETWORK (WAN)

A WAN is a physical or logical network that provides communication services between individual devices over a geographic area larger than that served by local area networks. Connectivity options range from plain-old telephone service (POTS) to optical networking at 160 Gbps rates (proposed). Terms such as *T1*, *E3*, *DS0*, and *OC-192* are often referred to in WAN literature, and frankly this alphabet soup of acronyms is confusing even to experts. There is no need to sweat like a stevedore when parsing these terms. See Appendix F for simple definitions and relationships of these widespread terms. Some of the links discussed in the appendix are used commonly to connect from a user's site to a Telco's office. Other links are dedicated to the generic Telco's internal switching and routing infrastructures. Usually, a WAN is controlled by commercial vendors (Telcos and the like), whereas a LAN is controlled by owners/operators of a facility or campus network. The QoS of the network depends not only on the type, but who controls it.

The four main criteria for segmenting wide area connectivity are

- **Topologies:** Switched and non-switched (point to point, mesh, ring)
- **Networks:** Private and public

Figure 6.19 segments these methods into four quadrants. An overview of topologies follows.

### 6.4.1 WAN Connectivity Topologies

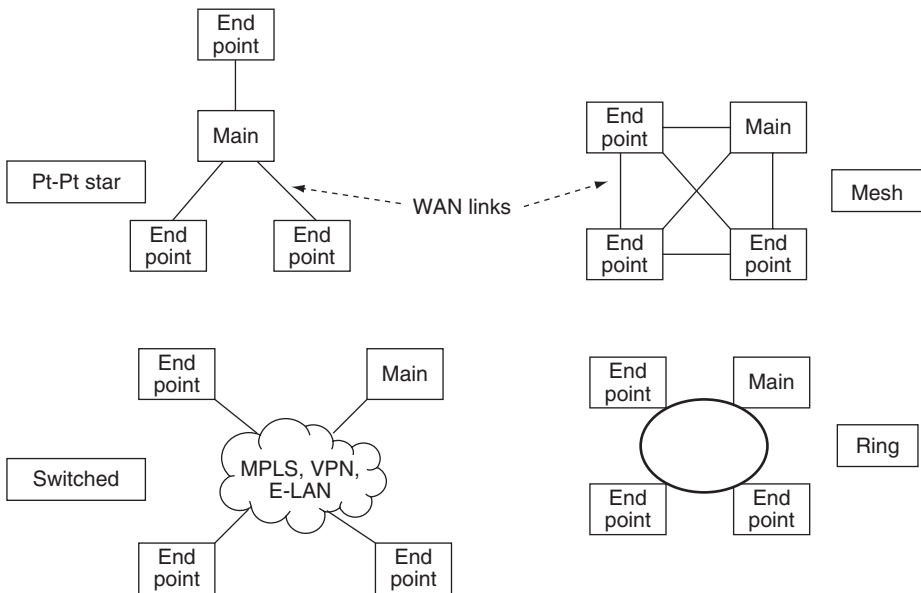
Each of the four types in Figure 6.20 may be used for general data communications, file transfer, storage access, and live A/V streaming. Figure 6.20 expands on Figure 6.19. Each has trade-offs in the areas of QoS, cost, reliability, security, and so on. The trade-offs are not covered in detail, but some consideration will be given to A/V-specific issues.

The point-to-point form is the most common type of connectivity. One example of this may be remote A/V sources (say, from three sports venues) all feeding live programming to a main receiver over terrestrial or satellite links. Many businesses have remote offices configured via a point-to-point means.

The mesh allows for peer-to-peer communications without going through a central office for routing. Depending on the geographic locations of the end points, a mesh may not make economic sense. In general, the mesh has been replaced by switched networks. The complexity of a mesh rises as the square of the number of nodes.

Switched	<ul style="list-style-type: none"><li>• Intranet</li><li>• Internet using VPN</li><li>• ATM SVC (legacy)</li><li>• MPLS</li><li>• E-LAN metro Ethernet</li></ul>	A	B
	<ul style="list-style-type: none"><li>• Campus fiber optic</li><li>• T1/E1,T3/E3</li><li>• SONET/SDH, ATM PVC</li><li>• E-Line metro Ethernet</li><li>• Wireless</li></ul>	D	C
Pt-Pt, mesh, ring			
	Private network	Public network	

**FIGURE 6.19** Wide area transport-type classifications.



**FIGURE 6.20** Wide area topologies.

The ring is a common configuration implemented by Telcos in a city or region. Using SONET, for example, a ring may pass by big offices or venues. Using short point-to-point links, the ring may be connected to nearby end points. Rings are often built with two counter rotating paths to provide for fault tolerance in the event that one ring dies. Many Telcos offer MAN services often based on ring technology.

Finally, there are switched topologies. The most common are

- Internet based (DSL, cable, other access), VPN or not
- Carrier Ethernet—E-LAN
- Multiprotocol Label Switching (MPLS)

Switched methods do not always offer as good a QoS as the other methods. Why? Switching introduces delay, loss, and jitter often not present in the others. Of course, WAN switching can exhibit excellent QoS, but only for selected methods such as MPLS and some Carrier Ethernet networks. MPLS is explained later in this chapter.

### 6.4.2 Network Choices

Figure 6.19 divides WAN network choices into public and private. WANs are available through Telcos and other providers. They are available to anyone who wants to buy a service connection. Normally, anyone on a given system can communicate to any other member. Quad A lists the most common enterprise WAN configurations. Nodes communicate over a managed service switched network. Users are offered a Service Level Agreement (SLA) that sets the QoS levels. Quadrant B shows the most common public switched networks. These are typically not managed, so the QoS level may be low. Quadrant D as a point-to-point system potentially offers excellent QoS for all types of A/V communications.

Whether a network is considered public or private, Telcos and other service providers can offer the equipment and links to build the system. The distinction between public and private is one of control, security, QoS, and access more than anything else. Given the right amount of packet reliability and accounting for delay through a network, any of the quadrants will find usage with A/V applications.

The Video Services Forum (VSF, [www.videoservicesforum.org](http://www.videoservicesforum.org)) is a user group dedicated to video transport technologies, interoperability, QoS metrics, and education. It publishes guidelines in the following areas:

- Multicarrier interfacing for 270Mbps SDI over SONET (OC-12)
- Video over IP networks
- Video-quality metrics for WANs
- Service requirements

The VSF sponsors VidTrans, an annual conference where users, Telcos, and equipment vendors gather to share ideas and demonstrate new A/V-networked products.

Another topic of interest to A/V network designers is the Carrier Ethernet. The next section outlines this method.

## 6.5 CARRIER ETHERNET (E-LAN AND E-LINE)

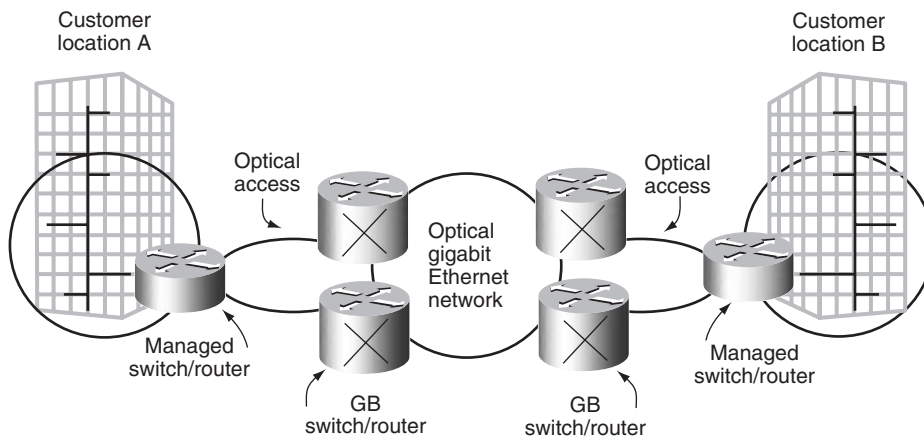
Ethernet has continually evolved to meet market needs. It was initially developed as a LAN. Over the years it has morphed from 10Mbps data rate to a soon-to-be 100 Gbps. Ethernet has staying power, and other layer 2 technologies have taken a back seat to its network dominance. Recently, Ethernet has been applied to metropolitan area networking (MAN) and even global level networking. This level of switching and access is termed *Carrier Ethernet*. This enables a seamless interface between a campus LAN and another campus LAN many miles away.

The Metro Ethernet Forum (MEF) has defined the following two standardized service types for Carrier Ethernet:

- **E-LAN:** This multipoint-to-multipoint transparent LAN service offers switched connectivity between nodes in a layer 2 Carrier Ethernet network. Figure 6.21 shows an example.
- **E-Line:** This is a virtual private line between two end points. There is no switching, and each line is statically provisioned.

Another dimension of value is geographical reach. The MEF is working to standardize the interconnection of Ethernet networks operated by different service providers, thus enabling a consistent user experience across vendors and distance. Carrier Ethernet provides the necessary QoS levels required for nodes to be seamlessly connected over a common infrastructure.

Regarding reliability, Carrier Ethernet can rapidly detect and recover from node, link, or service failures. Recovery from failures occurs in less than 50 milliseconds. This capability meets the most demanding availability requirements.



**FIGURE 6.21** Example of Carrier Ethernet E-LAN configuration.  
Concept: Reliance Globalcom.

For a video file transfer application, a 50 Ms reroute delay would not affect reliable delivery. Even for best-effort video streaming, only a few frames would be lost (not counting the time to resync the video). The missing frames may be concealed if needed.

Finally, Carrier Ethernet is centrally managed using standards-based vendor-independent tools. The advantages are as follows:

- Provision services rapidly
- Diagnose connectivity-related problems reported by a customer
- Diagnose faults in the network at any point
- Measure the performance characteristics of the service

Infonetics Research forecasts that worldwide Ethernet services revenue will be \$22.5 billion in 2009. Clearly, Carrier Ethernet is a viable switched network service for the media enterprise.

## 6.6 UNDERSTANDING QUALITY OF SERVICE FOR NETWORKS

The heart and soul of high-quality, digital A/V networking is the QoS<sup>4</sup> metric: low delay, low jitter, controlled bandwidth, low packet loss, and high reliability. Hand-wringing is common over maintaining QoS levels. Who sets them, how can they be guaranteed, and when are they out of limits? These are common concerns. Link QoS may be specified in a contract called a Service Level Agreement (SLA). Service suppliers provide SLAs whenever contracted for LAN or WAN provisioning. The elements of an SLA are useful criteria for any network design. QoS-related items are as follows:

- **Delay.** Also called latency, this is the time it takes a packet to cross the network through all switches, routers, and links. Link delays are never good, and the absolute value of an acceptable delay depends on use. Control signaling, storage access, file transfer, streaming (especially live interviews), and so on—each has different acceptable values. Delay may be masked by using A/V prequeuing and other techniques. Control has the strictest requirement for low delay and may be less than one line of video for some applications. However, for most applications, a control signal delay less than ~10 ms (less than half a frame of video) is sufficient. For LANs a 10 ms maximum delay is well within range of most systems.
- **Jitter.** This is the time variation of delay. Jitter is difficult to quantify, but knowing the maximum expected value is important.

---

<sup>4</sup> QoS can be applied to services of all types—networking, application serving, storage related, and so on. Each of these domains has a set of QoS metrics. For this section, networking QoS is the focus.

- **Controlled bandwidth.** Following are four common ways (some or all) to guarantee data rate:
  1. Overprovision the links with sufficient bandwidth headroom. Meter the ingress data rate to known values (e.g., 5 Mbps max).
  2. Do loading calculations for each link and switch to guarantee no switch congestion or link overflow at worst-case loading.
  3. Use reservation protocols to guarantee link and network QoS.
  4. Eliminate all IP traffic that does not have predictable data rates so that uncontrolled FTP downloads of huge files are not allowed over specified LAN segments. A rate shaping gateway may be used to tame unpredictable IP streams.
- **Packet loss.** The most common cause comes from congested switches and routers. A properly configured switch will not drop any packets, even with all ports at 100 percent capacity. Of course, traffic engineering must guarantee that ports are never overloaded. A/V clients that are good network citizens will always control their network I/O and thus help prevent packet congestion. It can be very difficult to eliminate data bursts that can cause congestion and packet loss downstream.
- **Reliability.** This topic is considered at length in Chapter 5 but is typically one of the most important elements of a SLA.

It is a good plan to work with A/V equipment providers that understand the subtleties of mission-critical networking and guaranteed QoS. See Chapter 2 for an illustration of network-related QoS metrics in action.

The Internet is a connectionless packet switched network, and all services are best effort. In contrast, leased lines and SONET are connection oriented, and data are delivered in predictable ways. Guaranteeing the QoS for a general Internet connection is nearly impossible. The Internet carriers do not agree on how to set and manage QoS criteria, someone has to pay for the extra level of service, and there is little motivation to change the status quo. There are specialized networks where the provider guarantees QoS using Multiprotocol Label Switching (MPLS) and Carrier Ethernet. This discussion does not make a distinction between class of service (CoS) and QoS, although they are different in principle. A CoS is a routing over a network path with a defined QoS. Incidentally, MPLS is designed to be network layer independent (hence, the name *multiprotocol*) because its techniques are applicable to *any* network layer protocol, but IP is the most common case.

QoS types can be broadly defined as soft or hard. A soft QoS has loose bandwidth guarantees, is statistically provisioned, and is defined in a hop-by-hop way. Hard QoS has (guaranteed rate rates,) bounded delay and jitter, deterministic provisioning, and is defined end to end. Hard QoS connections are best for



professional streamed video applications. What are the chief categories for QoS control? Here are some commonly accepted techniques:

- **Congestion management.** These methods reduce or prevent congestion from occurring.
- **QoS classification techniques.** IP packets are each marked (tagged) and directed to queues for forwarding. The queues are prioritized for service levels.
- **QoS reservation techniques.** Paths are reserved to guarantee bandwidth, delay, jitter, and loss from end to end.

Let us consider each one of these in brief.

### 6.6.1 Congestion Management

TCP has built-in congestion management by detecting packet loss and backing off by sending fewer packets and then slowly increasing the sending rate again (Figure 6.11). TCP is a major reason for the inherent stability and low congestion loss of the Internet. Within the network, routers sense congestion and may send messages to other IP routers to take alternate paths. Also, some routers may smooth out bursty traffic and reduce buffer overflows along the path. A router may use the Random Early Detection (RED) method to monitor internal buffer fullness and drop select packets before buffers overflow. Any congestion for critical A/V applications is bad news. High-quality streaming links cannot afford congestion reduction—they need congestion avoidance. File transfer can live with some congestion because TCP will correct for lost packets.

### 6.6.2 QoS Classification Techniques

Methods for QoS classification techniques are based on inspecting some parameter in the packet stream to differentiate and segment it to provide the desired level of service. For example, if the stream is going to a well-known UDP port address (say a video stream), then the router may decide to give this packet a high priority. Sorting on port numbers is not the preferred way to classify traffic, however; it breaks the law of independence of stack layers. The generally accepted classification methods in use today are based on tags. The three most popular means are as follows:

- **Ethernet frame tagging.** This is based on an IEEE standard (802.1D-1998) for prioritizing frames and therefore traffic flows. It has limited use because it is a layer 2 protocol and cannot easily span beyond a local LAN. For A/V use in small LANs, this type of segmentation is practical, and many routers and switches support it.
- **Network level ToS tagging.** The type of service (ToS) is an 8-bit field in every IP packet used to specify a flow priority. This layer 3 field has a

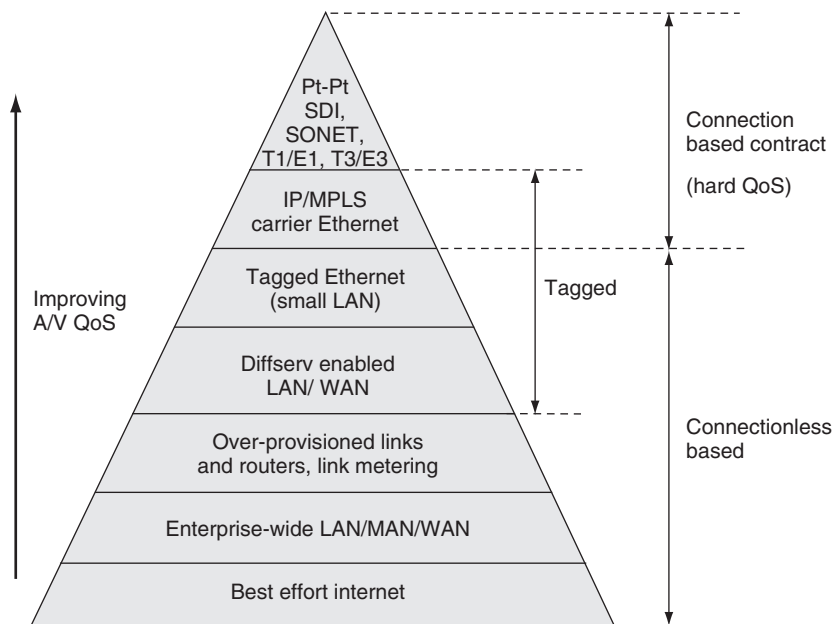
long history of misuse but was finally put to good use in 1998 with the introduction of IETF's Differentiated Services (Diffserv) model as specified by RFC 2475. Diffserv describes a method of setting the ToS bits (64 prioritized flow levels) at the edge of the network as a function of a desired flow priority, forwarding each prioritized IP packet within the network (at each router) based on the ToS value, and traffic shaping the streams so that all flows meet the aggregate QoS goals. Diffserv is a class of service method to manage data flows. It is a stateless methodology (connectionless) and does not enforce virtual paths as MPLS does.

- **MPLS tagging.** This technique builds virtual circuits (VCs) across select portions of an IP network. Virtual circuits appear as circuit switched paths, but they are still packet/cell switched. VCs are called label switched paths (LSPs). MPLS is an IETF-defined, connection-oriented protocol (see RFC 3031 and others). It defines a new protocol layer, let us call it "layer 2.5," and it carries the IP packets with a new 20-bit header, including a label field. The labels are like tracking slips on a pre-addressed envelope. Each router inspects the label tags and forwards the MPLS packet to the next router based on a forwarding table. Interestingly, the core MPLS routers do not examine the IP address, only the label. The label carries all the information needed to forward IP packets along a path across a MPLS-enabled network. Paths may be engineered to provide for varying QoS levels. For example, a path may be engineered for low delay and a guaranteed amount of bandwidth. MPLS operation is outlined in a later section.

These tagging methods are used in varying proportions in business environments and by Internet providers in the core of their networks. Several companies offer MPLS VPN services. A/V applications, including streaming, critical file transfers, storage access, and real-time control, can benefit from tag-enabled networks. Diffserv and MPLS are sophisticated protocols and require experts to maintain the configurations. Routers also need to be Diffserv and/or MPLS enabled. MPLS and Diffserv may indeed work together, as there is considerable synergy between the two methods. There is more discussion on these two methods in following sections.

### 6.6.3 QoS Reservation Techniques

Carrier Ethernet and MPLS virtual circuits can guarantee a QoS level while traversing across a broad network landscape. Each can carry IP packets as payload. Before routers pass any cells or packets, the QoS resources should be reserved. There are several ways to set up a virtual path with guarantees. One is to use the Resource Reservation Protocol (aptly named RSVP, RFC 2208, and others). RSVP is an out-of-band signaling protocol that communicates across a network to reserve bandwidth. Every router in the network needs to comprehend RSVP.



**FIGURE 6.22** *The QoS pyramid.*

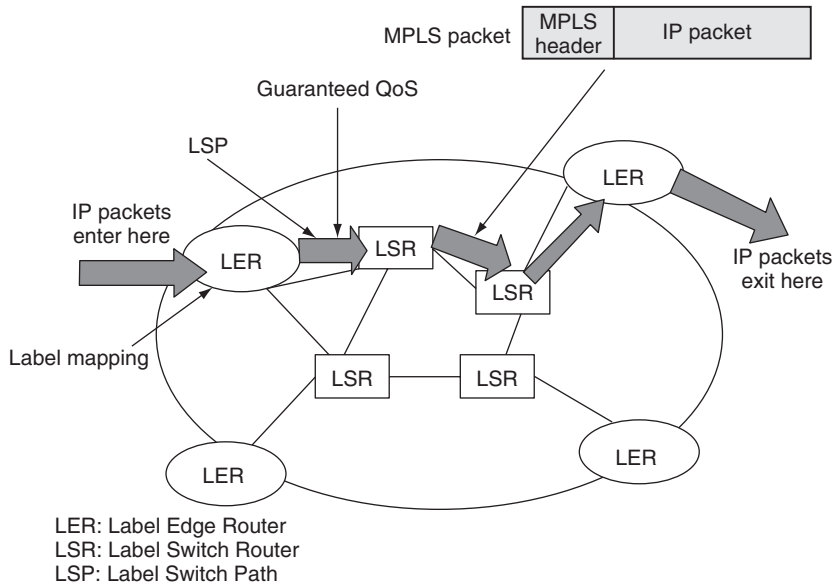
It finds application in enterprise intranets and in conjunction with MPLS and Carrier Ethernet to reserve path QoS.

Diffserv is a simpler, practical way to forward packets via what are called per-hop-behaviors (PHBs). Hops are defined (by the IETF) with different QoS metrics, such as minimum delay, low loss, or both. When a packet enters a router, its tag is inspected and processed according to the PHB it is assigned to. Admittedly, it is more concerned with class of service than QoS, a subtle distinction.

#### 6.6.4 The QoS Pyramid

Figure 6.22 illustrates the QoS pyramid. At the top of the pyramid is the trusty point-to-point link. There is no packet switching or sharing of bandwidth—it offers premium service at the sacrifice of self-addressed routing flexibility. The SDI (or equivalent) link falls here.

At the bottom is the Wild West of the Internet—the father of best effort service with routing (addressability) as its number one asset. All the other choices in the pyramid are specialized means to guarantee QoS to various degrees. Note that some of the methods are connection based, so a contract exists between end points; whereas others are connectionless with no state between end points. This is independent of the fact that layer 4 (TCP) may establish a connection over IP as needed. Some of the divisions may be arguable, but in general going up the pyramid provides improved QoS metrics.



**FIGURE 6.23** A MPLS routing environment.

### 6.6.5 MPLS in Action

Before we leave QoS, let us look at a simple MPLS-enabled network. As mentioned, MPLS is a connection-oriented protocol, so a contract exists between both ends of a MPLS network path. Figure 6.23 shows the chief elements of such a network. It is link layer independent. Standard IP packets enter the label edge routers (LER) for grooming, CoS classification (usually <8 classes defined, although more are available), and label attachment. MPLS packets traverse the network, routed by label switched routers (LSR).

The label is used to route the MPLS packets at each LSR and not the IP address. Packets follow a label switched path (LSP) to the designation LER where the label is stripped off as it enters a pure IP routed network. LSPs may be engineered for a range of QoS metrics. MPLS networks are becoming more common and are used in Internet carrier core networks, as offered by Telcos for private networks and for enterprise intranets. Expect to see MPLS applied to A/V applications, as it has a great combination of defined QoS levels and support for IP.

## 6.7 IT'S A WRAP—SOME FINAL WORDS

Networking is the heart and soul of IT-based media workflows. Just a few years ago, network performance was not sufficient to support professional A/V applications. Today, with proper care, LAN, WAN, and Carrier Ethernet are being used to transport A/V media and control messaging with ample fidelity. MPLS-, Carrier Ethernet-, and Diffserv-enabled connectivity offer good choices for

high-quality networking with performance guarantees. With IP networking performance and availability ever increasing, A/V transport is a common occurrence. True, dedicated video links will be with us for some years to come, but A/V-friendly networking is taking more and more of the business that was once the province of specialized A/V suppliers and technology.

## REFERENCES

- Akella, A., et al. (2002). Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP. *Proceedings of the 2002 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 117–130.
- Antony, A., et al. (October 23, 2003). *A New Look at Ethernet: Experiences from 10 Gigabit Ethernet End-to-End Networks between Amsterdam and Geneva*. The Netherlands: University of Amsterdam.
- Jin, C., Wei, D. X., & Low, S. H. (March 2004). TCP FAST: Motivation, Architecture, Algorithms, Performance <http://netlab.caltech.edu>. *Proceedings of IEEE Infocom*.
- DiMarzio, J. (April 2002). *Teach Yourself Routing in 24 Hours* Sams. New Jersey: Upper Saddle River.
- Jang, S. *Microsoft Chimney: The Answer to TOE Explosion!* Margalla Communications, [www.businessquest.com/margalla/](http://www.businessquest.com/margalla/), 8-19-03.
- Darren, Spohn. (September 2002). *Data Network Design* (3rd ed.). NYC, NY: McGraw-Hill Osborne Media.
- Stallings, W. (2003). *Computer Networking with Internet Protocols*. London, UK: Addison-Wesley.
- Stevens, R. (1994). *TCP/IP Illustrated, volume 1*. London UK: Addison-Wesley.

# Media Systems Integration

## CONTENTS

7.0	Introduction	268
7.1	The Three Planes	268
7.1.1	Examples of the Three Planes	270
7.1.2	The Control Plane	270
7.1.3	The Management Plane	277
7.1.4	The Data/User Plane	277
7.2	Wrapper Formats and MXF	278
7.2.1	Inside the MXF Wrapper	280
7.2.2	Working with MXF and Interoperability	285
7.3	Advanced Authoring Format (AAF)	288
7.3.1	Methods of AAF File Interchange	289
7.3.2	AAF Reference Implementation	290
7.4	XML and Metadata	291
7.4.1	Metadata Standards and Schemas for A/V	293
7.4.2	The UMID	294
7.4.3	ISAN and V-ISAN Content ID Tags	295
7.4.4	ISCI and Ad-ID Identification Codes	295
7.5	Media Asset Management	296
7.5.1	The MAM Landscape	297
7.5.2	MAM Functions and Examples	298
7.5.3	Using DRM as Part of a MAM Solution	301
7.5.4	Tastes Like Chicken	302
7.6	The Fundamental Elements of Media Workflows	303
7.6.1	The Design Element	304
7.6.2	The Process Orchestration Element	307
7.6.3	The <i>Operational</i> Element	310
7.6.4	The Workflow Agility Element	311

7.7 Broadcast Automation	314
7.8 It's a Wrap—A Few Final Words	315
References	315

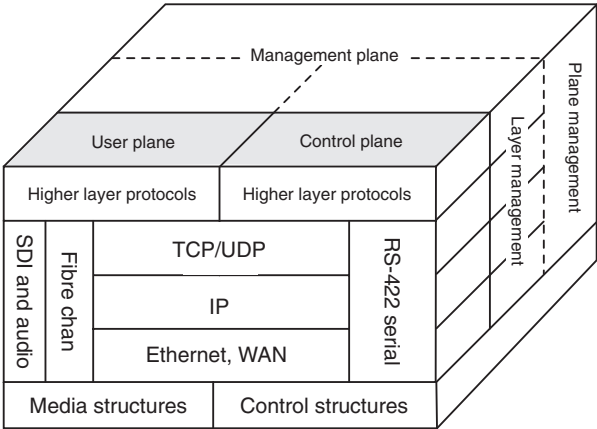
7.0 INTRODUCTION

The previous six chapters outlined the core elements of IT-based A/V systems design, including file-based technologies, streaming, storage, servers, HA methods, software platforms, and networking. This chapter ties these elements together to create world-class media workflow systems. Additionally, the foundations of media types, metadata, control methods, nodal management, and asset management are introduced to more fully describe networked media systems.

As a house is made of bricks, so a media system is composed of its constituents. But a pile of bricks does not make a house any more than a collection of servers and a network create a media system. It is the organization of the bricks that makes the house livable. So what are the organizational principles of A/V systems? Let us start by describing the three planes.

7.1 THE THREE PLANES

Is there a unified way to simply categorize all the disparate elements of an A/V system? Figure 7.1 is a pictorial of the three disciplines commonly used in most AV/IT systems: data/user, control, and management planes. Each plane has an associated protocol stack—LAN (TCP/IP), SDI, audio, or other as depicted. Figure 7.1 shows alternate stacks per plane depending on whether the system is based on traditional A/V or networked media. As a result, A/V (data plane) may be passed over an SDI link in one case or TCP/IP networking used in another. Due to the legacy of older control and management protocols, the RS-232/422 links will be in use for years to come. The stacks in Figure 7.1 are representational

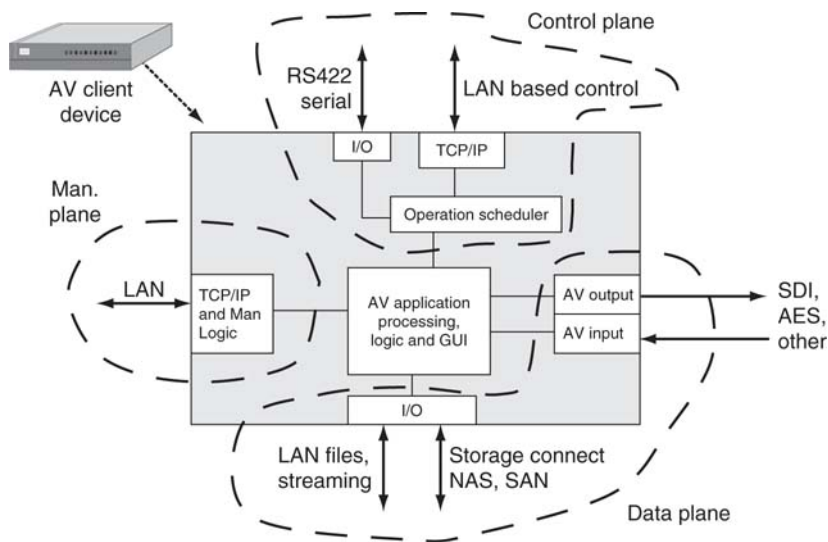


**FIGURE 7.1** The three planes: data/user, control, and management.

and not meant to document every possible stack for each plane. Although the planes are independent, they are often used together to accomplish a specific operation. For example, commanding a video server to play a file will involve both user and control planes. Here are brief descriptions of the three planes.

- **Data or user layer:** Moving A/V data across links in RT or NRT is a data plane operation. The data types may be all manner of audio, video, meta-data, and general user data. This plane is alternatively called data or user. One term describes the *data format* aspects of the plane, whereas the user handle denotes applications-related aspects, not shown, at the top of the stack. Editing a video that is stored on a remote networked device is a user plane operation.
- **Control layer:** This is the control aspect of a video system and may include automated and manual protocols for device operations (master control switcher, video server, video compositor, router, VTR, archive, etc.), live status, configuration settings, and other control aspects. This plane includes control applications, not shown, at the top of the stack.
- **Management layer:** This layer provides device management for alarms, warnings, status, diagnostics, self-test, parameter measurements, remote access, and other functions. This plane includes management applications, not shown, at the top of the stack.

Another way to view the three planes is illustrated in Figure 7.2. In this case, a general A/V device is divided into three functional domains. For sake of viewing, the LAN ports are repeated for data, control, and management, but in reality there may be only one LAN port and all three functional areas share the LAN.



**FIGURE 7.2** The three planes: physical view of client side.



However, in some cases the management LAN port would be a second port to completely isolate management from applications-related operations. Why do this? Device management operations should be non-intrusive and not affect the A/V operations in any way. The separate LAN port makes it easier to build and operate non-intrusive management operations. For example, blade servers typically have a dedicated Ethernet port for management use (see Appendix J).

In some cases, LAN isolation may apply to the control layer too. The choice of one, two, or three LAN ports is left up to the equipment manufacturer. Of course, choosing more than one LAN port can complicate the external network infrastructure if different QoS requirements are placed on each LAN connection.

### 7.1.1 Examples of the Three Planes

Complete industries exist to serve these layers. For example, automation companies such as Avid/Sundance, Florical Systems, Harris Broadcast Communications, Hitachi Systems, Masstech Group, Omnibus Systems, Pebble Beach Systems, Pro-Bel, SGT, and others sell products for the control layer (see Section 7.7). Traditional video equipment companies sell data/user (A/V equipment of all sorts) plane products. Device management solutions have traditionally been vendor specific, but Miranda (iControl) and Snell & Wilcox (RollCall), for example, offer general device management solutions, despite a lack of industry-wide standards. Let us consider a few examples in each plane.

### 7.1.2 The Control Plane

Traditionally, the control layer has been forged from custom solutions and lacks the open systems thinking that is prevalent in the general IT world. For example, many A/V devices still rely on RS-422-related control protocols and not LAN-based ones. For controlling video servers, the Video Disk Control Protocol (VDCP) has been used for many years over the RS-422 serial link, and many manufacturers are reluctant to move away from it, despite several vendor attempts to introduce LAN-based control protocols. The common Sony BVW-75 VTR control protocol is also in wide use and is RS-422 based. At present, there is no LAN-based A/V device control protocol sanctioned by SMPTE, although all automation and server companies have developed private LAN-based protocols. For example, some of the current *vendor-specific* LAN control protocols (and APIs) for networked A/V devices (especially servers) are as follows:

- Avid's Interplay Web services APIs
- Harris's VDCP over LAN
- Media Object Server (MOS) from Associated Press (AP) and the MOS user group
- Omneon's Server Control protocols: Player Control API and Media API
- Omnibus's G2/G3 Control protocols
- ClipNet protocol from Quantel

- Thomson/GVG Profile Server and K2 Server native control
- Sony, SeaChange, and others, which offer proprietary LAN-based control protocols

Vendors have developed device-frame-accurate, custom LAN-based protocols for controlling servers, file transfer, logo inserters, real-time compositors, A/V routers, character generators, format converters, and more.

For now, these incompatible protocols will coexist in AV/IT systems. Of course, this is not ideal and creates interoperability issues, but until SMPTE or some industry group standardizes a method(s) or a de facto one is selected by the market, there will be confusion and competition among protocols.

#### 7.1.2.1 The MOS Example

Media Object Server (MOS) is a protocol based on message passing and not one for direct device control. The MOS protocol was designed by the MOS Group for *story list management* in A/V devices. It has achieved excellent market acceptance as an IP-based protocol. The MOS Group is an industry body composed of representatives from many industry companies. The protocol is applied to news production for creating, organizing, deleting, and modifying the news “rundown list” of stories for a newscast. Video playback servers, character generators (CGs), video compositors, teleprompters, and even robotic cameras need to know what activity to do per-story entry. MOS manages and synchronizes the activity lists across devices.

The following is a sample list of device activities needed to run story #3 for the newscast:

- Story 3 needs a lower third text crawl, so the CG has a rundown story entry “Story 3, text crawl, ‘Snake River overflows banks ...’ “
- Story 3 requires an over-the-shoulder video clip of the swollen river, so the video server has a story entry “Story 3, play clip Snake-Flood.dv.”
- The teleprompter has a rundown entry “Story 3, file Snake-Flood.txt.”

The MOS protocol works in the background in non-real-time creating rundown activity lists in all equipment. It is not considered a real-time control protocol. At story time, a separate scheduling engine (or manual operation) triggers the individual devices to execute the list entry for story #3, thus creating a well-orchestrated harmony across all equipment. List management is an ideal activity for an IP-based protocol because no frame-accurate video control is required. See [www.mosprotocol.com](http://www.mosprotocol.com) for more information.

With the success of MOS, industry leaders are looking at ways to use the framework of the protocol (XML message passing) for general, real-time, frame-accurate, and device control over IP networks. Of course, new commands are needed, including the prequeuing methodology discussed here.

### 7.1.2.2 The Broadcast eXchange Format (BXF)

BXF is a message passing protocol and should not be confused with MXF (described later), an A/V essence and metadata wrapper format. Although the acronyms are similar, the formats are designed for completely different purposes. In most broadcast facilities MXF and BXF will happily coexist. So, what is BXF?

BXF is standardized as SMPTE 2021 and defines the communication of three basic types of data:

1. Schedule and “as-run” information
2. Content metadata
3. Content movement instructions

BXF is based on the XML data-interchange standard. It provides advantages to the broadcast TV facility (and similar operations), including the following:

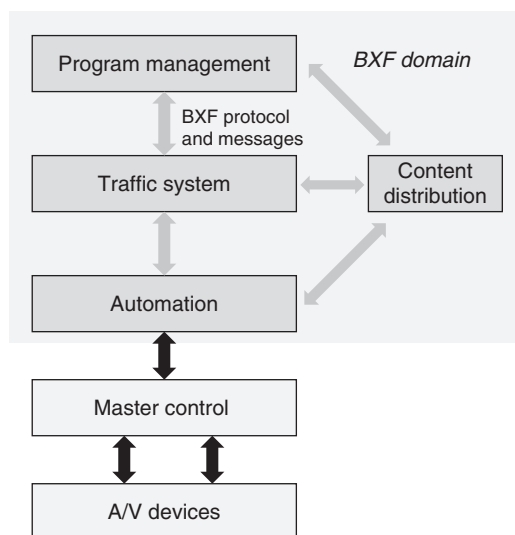
- A single method of exchanging data among these four domains: program management, traffic, automation, and content distribution
- Support for file- and message-based (dynamic) data exchange
- Increased integration of related systems
- Extended metadata set for data exchange

BXF is not a real-time device control protocol!

Before BXF, the four domains communicated with a hodgepodge of vendor-custom protocols. This resulted in incompatible message and data-passing methods and vendor lock-in. BXF is a breath of fresh air for interoperability and open systems; it will be required for all new installations. MOS and BXF have some overlap today, and there is potential for consolidation in the future (Figure 7.3).

#### BXF messaging

- Broadcast schedules
- “As run” information
- Content metadata, such as Content ID, title, duration
- Content management requests such as dub and purge requests
- Requests for transfer of content



**FIGURE 7.3** BXF messaging partners in broadcast operations.

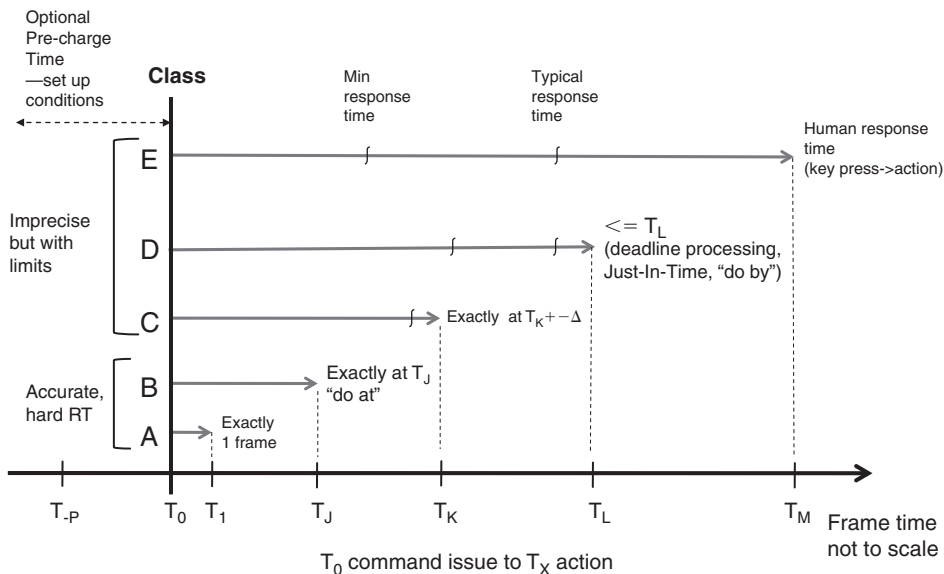
Next, let's consider what methods are needed to use a LAN and still achieve real-time control.

### 7.1.2.3 Command Execution Modes for Device Control

Let's start with a simple categorization of control accuracy and timing modes that are commonly used. Figure 7.4 illustrates five control events labeled A to E. The vertical axis is roughly related to event timing accuracy from the inception of a command at time  $T_0$  until command execution. The horizontal axis is time in increments of integer video frame counts. For example, a class A command issued at time  $T_0$  results in an event occurring (for example, video server playout starts) at  $T_1$ . This command type is "immediate," since the event occurs at the start of the next whole video (or audio) frame.

Control class B requires that the command issued at  $T_0$  is executed precisely at time  $T_J$ . This may be anywhere from 2 frames to thousands counting from  $T_0$ . This method is very useful when scheduling events to execute in the future: do-this-at- $T_X$ . This command type is typically underutilized in broadcast facilities. Class C is a version of B with a small allowable jitter in the command execution time.

Class D is a bounded control scenario. This class should be used when an operation's completion is required (convert file ABC to MPEG4) before some deadline time  $T_L$ . This class relaxes timing control considerably. Many facility operations can be designed to schedule operations to be complete before a deadline. File conversion, indexing, and transfers are a few operations that may



**FIGURE 7.4** Command/action timing models.

be scheduled by a class D event. In general, don't use classes A or B if D meets your needs.

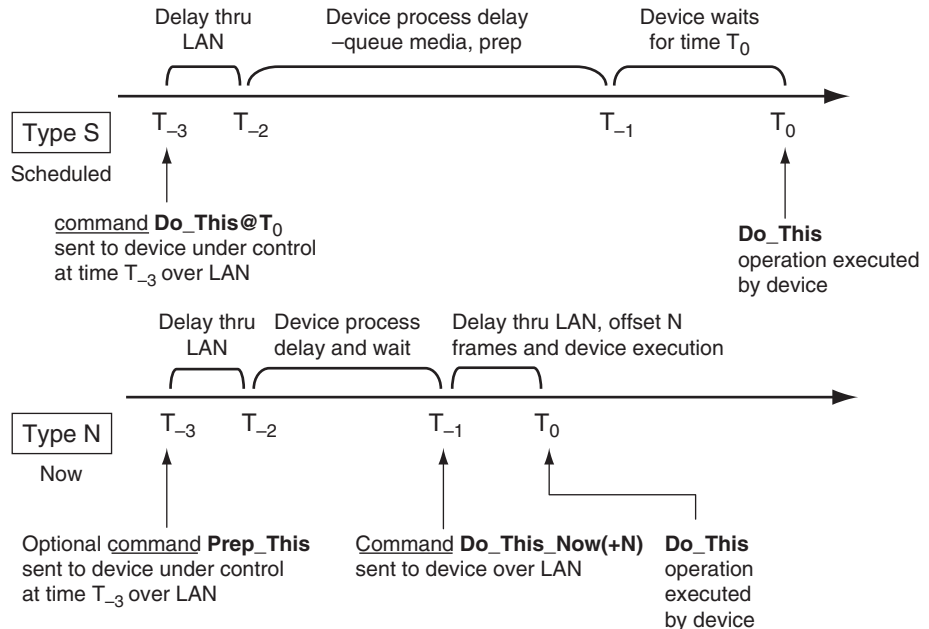
Finally, class E is human timed: "Ready on camera 3 ... take 3." It's not frame-accurate timing but is bounded by human reaction speeds. For the most part, we care about classes A and B when commands are sent over a LAN.

#### 7.1.2.4 Techniques for Control over IP

Why have LAN-based device control protocols been adopted so slowly? Most traditional RS-422 serial device control protocols are video frame accurate by the nature of the point-to-point wiring. There is never congestion or meaningful latency using a serial link. It is proven, it works, and it is still in wide use. Nonetheless, over time, LAN will replace dedicated RS-422 links. So, let's look at two strategies for using a LAN to achieve frame-accurate A/V device control. One is based on class A and the second on class B timing. Both control examples are shown in Figure 7.5. Let's call them type N (do this **now**, based on class A) and type S (**scheduled**, based on class B) methods. Note that the timing references in this figure are different from those in Figure 7.4;  $T_0$  is the desired *execute* time.

#### 7.1.2.5 Type S—LAN-Based Scheduled Real-Time Control Method

The type S model is based on prequeuing one or a list of actions in the target device. Queued items range from a simple command to play a clip to a complex

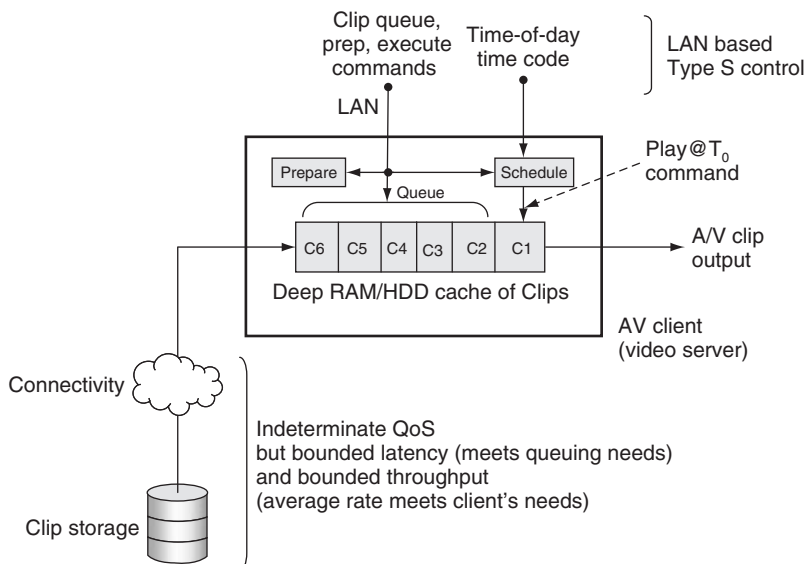


**FIGURE 7.5** Type S and N models for device control over LAN.

multielement video composite. Each queued item has an exact, frame-accurate, future execute time,  $T_0$ . As long as the device reliably receives the command instructions adequately before  $T_0$ , then any IP jitter or packet loss is of no consequence. When the device under control is allowed to, in effect, schedule the future operation, any small LAN delays will be absorbed. Of course, the controlling intelligence must cooperate by sending commands before they are to be executed at  $T_0$ .

Reviewing Figure 7.5, the type S timeline shows three critical periods from the initial command reception to command execution. A **Do\_This@ $T_0$**  command (with **This** implying some device operation) is sent to target device at  $T_{-3}$ . Delay through a small LAN network is normally  $<1$  ms (much less than one frame of video) if there is no router congestion. The target and sender device TCP/IP processing delay are both in series with the LAN delay, which can be significant if not managed. The next period ( $T_{-2}$  to  $T_{-j}$ ) is allocated for the device to queue any A/V media and prep for the desired action at  $T_0$ . This may take from one to  $N$  video frames based on traditional serial control methods. In the world of VTRs, this is called the *preroll time*. Next, there is a wait period until  $T_0$  occurs referenced to a time code clock (usually synced to the facility time-of-day clock). At  $T_0$ , the desired operation is executed, frame accurately. The  $T_{-j}$  until  $T_0$  delay can range from one frame to hours.

Figure 7.6 illustrates a video server with a deep queue using type S control. The external storage does not require a low latency response, as the server has ample time to queue the clips. Of course, the average storage access bandwidth



**FIGURE 7.6** Deep queuing with quasi-RT storage and type S control.

must meet the needs of the server. Type S does not require a deep queue (one level may be sufficient), but the deeper the queue, the more forgiving the overall system is to temporary LAN and storage access anomalies. In fact, a type S control schema considerably enhances overall system reliability if queuing is used judiciously.

A type S control has ideal frame-accurate characteristics as long as prequeuing timing is observed. The minimum practical time from command reception to execution is less than one frame of video. The maximum time depends on several factors, such as queuing time and the QoS of any external storage, and may require 7 s or more for reliable, repeated execution. Many typical operations (video server plays Clip\_C1, for example) need 3 s or less from  $T_{-3}$  to  $T_0$ . If the target device does not support scheduled operators or the application is not suited for this model, then type N may be used.

#### **7.1.2.6 Type N—LAN-Based Immediate Real-Time Control Method**

For the type N scenario, the concept is based on a **Do\_This\_Now** command plan. In some cases, such as selecting signal routing, immediate execution is required with no prequeuing or prescheduling possible as with a type S. A low LAN delay is crucial (less than one frame of video) for immediate execution of some commands. For other scenarios, prequeuing is required, so a **Prep\_This** command is issued *before* the **Do\_This\_Now** is executed. Figure 7.5 shows the prep command in the type N timeline. Command execution is not prescheduled as with a type S, but follows the **Do\_This\_Now** command being received by the target device. In general, the execute command needs to have an offset of  $N$  (0,1,2,3,4, ...) frames in the future to allow for the frame-accurate alignment of other coordinated A/V devices—each with a potentially different execute latency. As a result, **Do\_This\_Now(+N)** is a more general case where  $N$  is device specific. This is not a new problem and exists with RS-422 command control today.

There may be several prep commands issued before the corresponding **Do\_This** command. For example, the sequence of **Prep\_This\_1**, **Prep\_This\_2** may precede a **Do\_This\_2\_Now** followed by a **Do\_This\_1\_Now** execution sequence. The order of execution is not foreordained. The most crucial time period is from  $T_{-j}$  to  $T_0$  and should be less than one video frame ( $\sim 33$  ms with 525-line video). Modern LANs can meet this requirement.

For most type S and N control cases, the relaxing of storage access latency implies that the storage and connecting infrastructure is easier to build, test, and maintain. Plus quasi-real-time (latency may on occasion exceed some average value) storage is less expensive and more forgiving than pure RT storage. Of course, if the workflow and reliability demand immediate access and playout without the advantage of deep queuing, then the storage QoS will be rigid. There is no free lunch, as prequeuing clips in local client memory (RAM usually but disk is also possible for some cases) adds a small expense. A client cache that can hold 50 Mbps encoded clips for 1 min needs to be at least

375MB deep. This is not a huge penalty, but it is a burden. Also, the logic for deep queuing may be nontrivial, and some automation controllers may not be designed with deep queuing in mind. Also, if the playout sequence changes in the last seconds (news stories), then the queue needs to be flushed or reordered, which adds complex logic to the workflow. See the section on user data caching in Chapter 3B for more insight into the art of caching.

It is inevitable that LAN methods will replace legacy serial links. With certainty, the generic **Do\_This@T<sub>0</sub>** and **Do\_This\_Now** with associated deep queuing/prep commands will be implemented. Many industry observers predicted (hoped!) that LAN control would completely replace the serial link (RS-422) by 2009, but this has not happened yet. Next, let us consider the management plane.

### 7.1.3 The Management Plane

The management plane (see Figure 7.1) is the least mature of the three because there are too few A/V product management standards to gain the momentum needed to create a true business segment. The general IT device management solution space is very mature with hundreds of vendors selling to this domain. However, because A/V equipment manufacturers have been slow to develop standardized management plane functionality, many A/V-specific devices are managed in an ad hoc manner. SMPTE is encouraging all vendors to assist in contributing to common sets (general and per device class) of device status metrics, but the uptake has been slow. See Chapter 9 for a complete coverage of the management plane.

### 7.1.4 The Data/User Plane

The data/user plane is the most mature of the three, with many vendors offering IT-based NLE client stations, A/V servers, browsers, video processors, compositors, storage systems, and other devices. For example, Sony offers the XDCAM camera family (field news gathering) using the Professional Optical Disc (there is a Flash card version too), Panasonic offers the P2 camera family using removable Flash memory, and Ikegami offers the EditCam series with a removable HDD. These are a far cry from the videotape-centric cameras of just a few years ago. The P2 and XDCAM have LAN ports for offloading A/V essence,<sup>1</sup> usually wrapped by MXF with included metadata.

Generally, most recently developed A/V devices show a true hybrid personality with traditional A/V connectors, a LAN port, and other digital I/O ports such as IEEE-1394 or USB2. Incidentally, high-end P2 cameras support five 32GB removable Flash cards and a gigabit Ethernet port supporting download rates of 640Mbps. This storage is equivalent to 200 minutes of HD 1080/24P

---

<sup>1</sup> The term *essence* denotes underlying A/V data structures such as video (RGB, MPEG2, etc.), audio (AES/EBU, WAV, MP3, etc.), graphics (BMP, JPEG, etc.), and text in their base formats.



content. There are interesting and compelling trade-offs among these three camera styles.

The types of data plane A/V essence in use are varied from uncompressed digital cinema production quality (~7Gbps) to low bit rate compressed proxy video at 200Kbps. Audio too can range from multichannel, 24-bit uncompressed, 2.3Mbps per channel to MP3 (or a host of others) at 64Kbps. Due to the wide variety of A/V compression formats, video line rates, and H/V resolutions, achieving interoperability among different vendors' equipment can be a challenge. Although the data plane is standardized and mature in many aspects, creating workflows using different vendors' equipment can be a challenge.

The protocol aspects of this plane include network protocols such as TCP/IP, storage access protocols such as SCSI and iSCSI, and file server access protocols such as NFS and CIFS. The main goal of access protocols is to get to data—the A/V and metadata gold that resides on disk arrays and archive systems. These protocols are discussed in detail in Chapter 3B.

The data structures layer is rich in variety and detail. SMPTE and other standard bodies have devoted hundreds of standards to describing these structures. Chapter 11 specializes in the A/V data/user plane with coverage of the fundamentals of A/V signal formats, resolutions, interfaces, compression, transmission formats, and many more. You may want to read Chapter 11 first before continuing here if you are unfamiliar with the basics. If not, let us move on to wrapper formats, including MXF, AAF, and XML. See (SMPTE 2004) for a good tutorial series on MXF and AAF.

## 7.2 WRAPPER FORMATS AND MXF

The A/V industry is not lacking for file formats. A laundry list of formats confronts designers and users daily: MPEG1, 2, 4; H.264; VC-3; DV (25, 50, 100Mbps); Y'CrCb; RGB; audio formats; and the list continues. As shown previously, files are indispensable when acquiring, logging, browsing, storing, editing, converting, transferring, archiving, and distributing A/V materials. Is there a way to tame the format beast? Can we select one format that all users would support? If so, then interoperability would be a snap, and file exchange between users would rarely hit a snag. Additionally, A/V equipment interoperability, vendor-neutral solutions, and one archive format all follow when a universal file format is chosen. Despite the desire for interoperability, very few users would accept a one format policy. Why not? Each format has its strengths and weaknesses. Depending on business needs (acquisition format, cost, quality, bandwidth, simplicity, legacy use, etc.), format A may be a better choice than format B. In the end, the one format policy cannot be legislated, despite all its benefits.

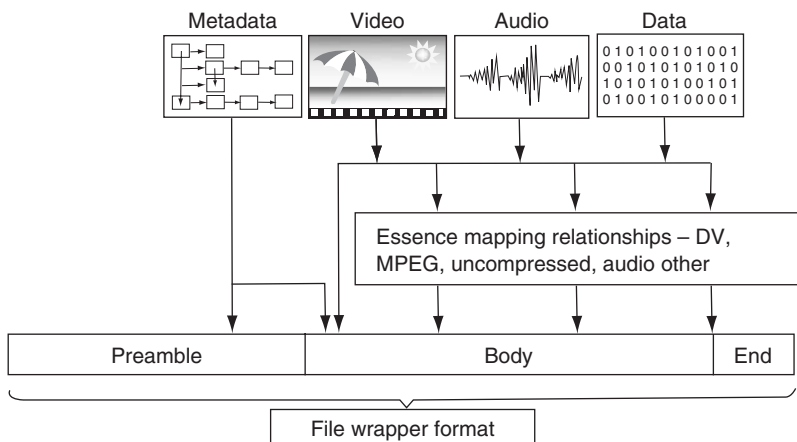
Fortunately, there is an acceptable trade-off: a standardized, universal, professional *wrapper format*. A wrapper does exactly what the name implies—it wraps something. In the broadest sense, that something may be A/V essence,

graphic essence, metadata, or generic data. Like peas inside a pea pod, a wrapper is a carrier of lower-level items. A universal wrapper fosters interoperability between users and equipment at several levels. Figure 7.7 illustrates the concept of a wrapper. Note the various essence mappings into the wrapper file.

A wrapper format is not the same as a compression format (e.g., MPEG2 Elementary Stream). In fact, many wrappers are compression agnostic even though they carry compressed essence. The QuickTime.mov format from Apple is a wrapper. The ubiquitous.avi file format is a wrapper. A File.mov or File.avi that carries MPEG essence or DV does not disclose which by its file extension, unlike File.dv, which is always a DV format. Many A/V essence formats have documented mappings into file wrappers. The term *essence agnostic* is often cited regarding MOV, AVI, or MXF but is only partially true. The wrapper must provide for the underlying essence mapping with supporting official documentation. An undocumented mapping is useless. For an excellent reference to MXF and other file formats, see (Gilmer 2004) and also the SMPTE Engineering Guidelines EG-41 and EG-42.

Despite the existence of A/V wrappers, all legacy formats fall short of the needs of professional A/V. The ideal wrapper requirements for our needs are

- Open and standardized (QuickTime is not open or standardized)
- Supportive of multiplexed time-based media
- Supportive of multiplexed metadata
- Targeted for file interchange
- Essence agnostic in principle
- OS and storage system independent
- Streamable
- Extensible



**FIGURE 7.7** Example of a file wrapper format.  
Concept: File interchange handbook, chapter 1.

Wrapper requirements were identified by the SMPTE/EBU Task Force report of April 1997. Following that, the ProMPEG Forum was formed to define and develop a wrapper format that met the requirement list given earlier. The initial work started in July 1999 and was called the Material eXchange Format, or MXF. After nearly 4 years of effort, the forum submitted its documents to SMPTE for standardization in 2003. In 2009, there are 31 MXF-related standards, proposed standards, and engineering guidelines. SMPTE 377M is the fundamental MXF format standard. MXF has been favorably embraced by the A/V industry worldwide. Of course, it will not replace existing wrappers or dedicated formats overnight. It will take time for MXF to gain enough steam to become the king of the professional A/V format hill. MXF is expected to have minimal impact on consumer product formats.

7.2.1 Inside the MXF Wrapper

There are two chief ways to view a MXF file: the physical layout and the logical layout. The physical view is considered first. Figure 7.8 shows the physical layout of a typical MXF file. The A/V essence, metadata, and optional index file (associates time code to a byte offset into the file) are all multiplexed together using basic key/length/value (KLV) blocking. KLV coding is a common way to separate continuous data elements and allow for quick identification of any element. The key is a SMPTE registered 16B Universal Label (SMPTE 336M) that uniquely identifies the data value to follow (audio, video, metadata, etc.). Length indicates the number of bytes in the value field. The value field carries the data payload, including audio samples, video essence, metadata, index tables, pointers, and more.

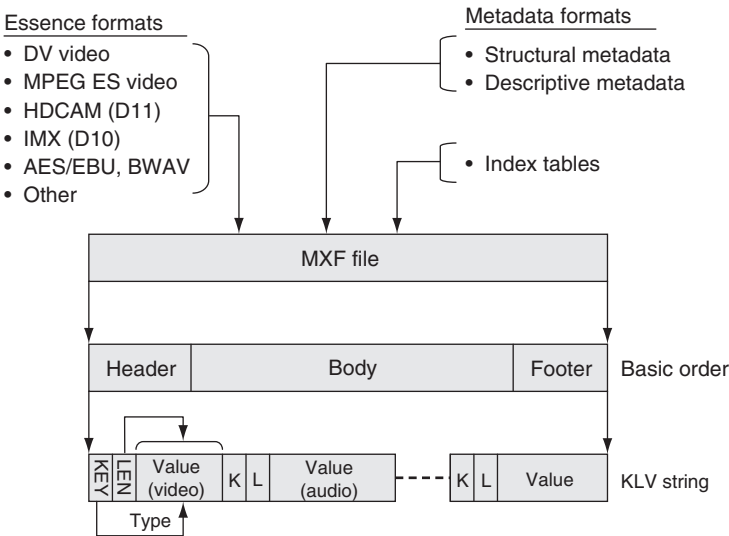


FIGURE 7.8 Physical views of a MXF file.

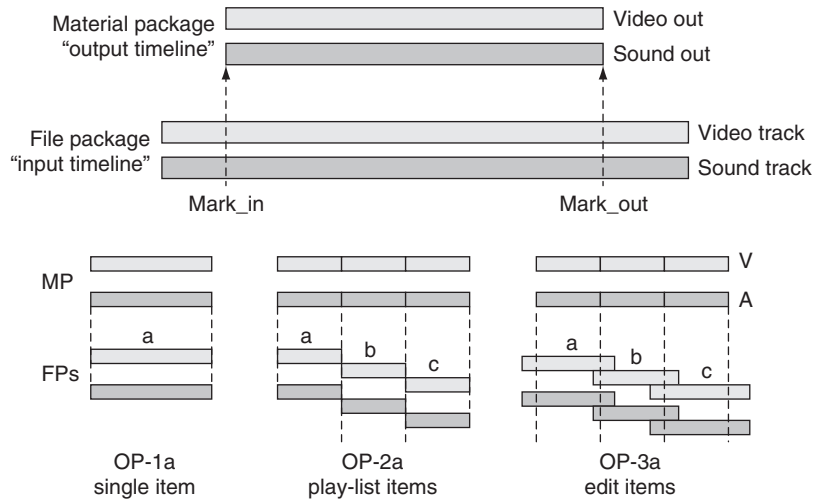
The KLV sequences are divided into three general groups: the header, body, and footer. The *header* contains information about the file, such as operational pattern (explained later), clip name, length, creator, aspect ratio, encoding rate, frame rate, and so on. This information is referred to as *structural* (in contrast to descriptive) metadata. The header also contains descriptive metadata that may be time synchronous with the A/V essence. The body contains A/V multiplexed essence. The A/V essence is mapped according to the rules per each data type. For example, MPEG (including MPEG1, 2, MPEG4 and H.264/AVC) has a mapping (SMPTE 381M), DV has a mapping (SMPTE 383M), AES/EBU audio has a mapping (382M), HD codec VC-3 has a mapping (2019-4), and so on. Most mappings locate each video frame on a KLV boundary for convenient, fast frame access, but there is no absolute requirement for this. Finally, the *footer* closes the file with optional metadata. Additionally, index tables (frame tables) are optionally included in the header, interleaved in the body, or stored in the footer.

#### 7.2.1.1 MXF: The Logical View

The second way to view a MXF file is logically. In this case, the focus is on the organization of the information, not how it is sequenced in the file. Figure 7.9 (top) illustrates a very simple MXF file with only sound and video essence tracks. Of course, data are stored as KLV sequences, but the organization shows a File Package (FP) and Material Package (MP). By analogy, the File Package is the “input timeline,” a collection of files in effect, whereas the Material Package is the “output timeline”—how the MXF internal files are to be read out. The example shows the output to be a portion of the stored essence. A small amount of internal metadata sets the `mark_in` and `mark_out` points and is changed easily.

This is only the tip of the organizational iceberg of MXF, and much of its documentation is devoted to describing its logical layout. It is not hard to imagine all sorts of ways to describe the output timeline based on simple rules between the File Package and the Material Package. These rules are called Operational Patterns. Consider the following in reference to Figure 7.9:

- Single File Package, Single Material Package. This is the most common case, and the FP is the same as the MP. This is called OP-1a in MXF speak and is referenced as SMPTE 378M. An example of this is DV essence, with interleaved audio and video, wrapped in a single MXF file.
- Multiple File Packages, Single Material Package. This is case OP-2a (SMPTE 392M) and defines a collection of internal files (a, b, and c) sequenced into one concatenated output.
- OP-3a (SMPTE 407M) is a variation of OP-2a with internal tracks a, b, and c each having `mark_in` and `mark_out` points.



**FIGURE 7.9** Logical views of a MXF file.

There are seven other operational patterns (2a, 2b, 2c, 3a, 3b, 3c, and OP-ATOM), each with its own particular FP to MP mapping. The simplest, OP-ATOM (SMPTE 390M), is a reduced form of OP-1a where only one essence type (A or V, not both) is carried. Many vendors will use this format for native on-disc storage but support one or more of the other OPs for file import/export. Frankly, the abundance of OPs makes interoperability a challenge, as will be shown.

### 7.2.1.2 Descriptive Metadata

A distinguishing feature of MXF is its ability to optionally carry time synchronous *descriptive metadata*. Other wrappers are not as full featured in this regard. An example of this type of metadata is the classic opening line of the novel *Paul Clifford*:

*It was a dark and stormy night; the rain fell in torrents—except at occasional intervals, when it was checked by a violent gust of wind which swept up the streets, rattling along the housetops, and fiercely agitating the scanty flame of the lamps that struggled against the darkness.*

—Edward George Bulwer-Lytton (1830)

It is not hard to imagine this text associated with a video of a rainy, night-time London street scene. As the video progresses, the descriptive metadata text is interleaved scene by scene in the MXF wrapper. Once available for query, the metadata may be searched for terms such as *dark and stormy* and the corresponding video timecode and frames retrieved. The promise of descriptive data is enticing to producers, authors, editors, and others. The entire value chain for descriptive metadata is nontrivial: authoring the text, carrying text, storing

searchable text separate from corresponding video, querying the text and retrieving corresponding video, editing the text, archiving it, and so on.

Our industry is struggling to develop applications that use metadata to its fullest potential. Because metadata span a wide range of user applications, no one vendor offers a comprehensive, end-to-end solution set. SMPTE has standardized several means to carry metadata inside a MXF wrapper and DMS-1 (Descriptive Metadata Schema, SMPTE 380M) is one such method. In addition, SMPTE has also defined a metadata dictionary (RP 210) with room for custom fields as needed. Interestingly, when Turner Entertainment documented its cartoon library, it invented 1,500 new terms to describe cartoon activities that rarely occur in daily life, such as stepping off a cliff, slowly realizing that the ground is far below, and then falling.

Several vendors support the budding metadata management world. A few tools in this space are MOG's Scribe and MXF Explorer ([www.mog-solutions.com](http://www.mog-solutions.com)), Metaglue's Diffuser ([www.metaglue.com](http://www.metaglue.com)), and OpenCube's MXF Toolkit ([www.opencube.fr](http://www.opencube.fr)). Do not confuse metadata management with digital asset management (DAM). DAM is considered later in this chapter. Incidentally, a free MXF SDK is available at [www.freemxf.org](http://www.freemxf.org).

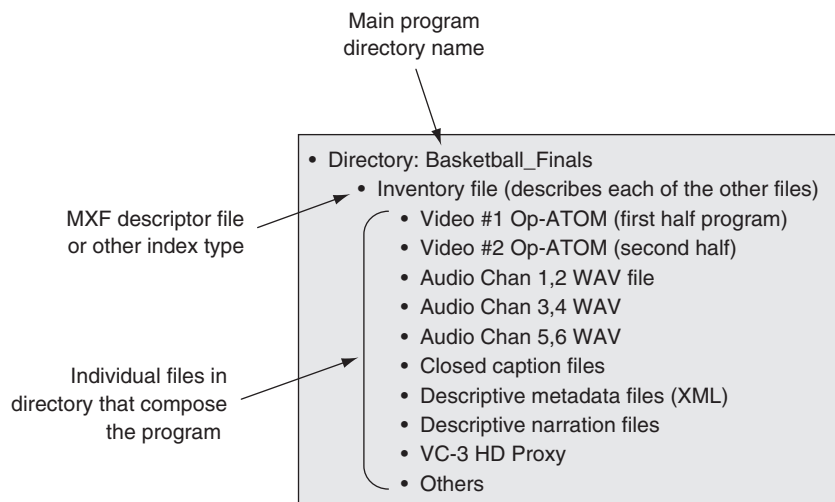
### 7.2.1.3 The Group of Linked Files (GOLF) Concept

The overarching goal of MXF is to package data formats of various flavors under one wrapper using operational patterns to define the packaging. An archived program may include some or all of the following data types:

- Video tracks—one or more, SD versions, HD versions
- Audio tracks—one or more
- Descriptive metadata
- Closed caption files (per language)
- Descriptive narration files (per language)
- Proxy files—VC-1, VC-3, or similar low bit rate (A + V)
- AAF compositional metadata file
- Others (rights use, etc.)

MXF defines the wrapping rules for many of these data types, but not all of them. For example, the proxy and closed caption files may remain separate, never to be wrapped by MXF. The audio and video could be wrapped into a single MXF file; however, there are reasons to keep them discrete.

While it is true that most of the file types in the list may be wrapped into a single MXF file, at times it is wise to keep *all* these files separate. Using the concept of a master inventory file, all referenced files become linked together. Let us call this a *group of linked files*. A GOLF uses an inventory list to document how all other individual files in the list are (time) related. For access purposes, a user may retrieve all or parts of the program, including partial access within an individual file/track. By referring to a named directory, the user can easily move all its parts in total to another location without fragmentation.



**FIGURE 7.10** Group of linked files (GOLF) example.

Figure 7.10 shows an example of a GOLF for the program title “Basketball\_Finals.” Notice that only the video tracks are wrapped in MXF using Op-ATOM, the simplest operational pattern. The inventory file is a key element and defines the contents of all the other files in the GOLF. The inventory file is MXF but does not carry any essence—only pointers to external essence and other data types. The inventory file creates a “smart directory” of sorts.

The GOLF files are accessed as a function of the needs of a workflow. For example, for A/V editing, the separate audio and video files are accessed as needed. For broadcast playout, the audio, video, and closed caption files are retrieved and sequenced together in time. For low-resolution browsing, only the proxy file is needed. As a result, a GOLF enables easy random access to target files. Using an all-encompassing MXF wrapper, all included tracks must be retrieved to access even one track.

The upper level applications need to assure proper A + V + data synchronization when combining files for playout. Incidentally, this is something that MXF does inherently well. The GOLF method has other advantages compared to a fully wrapped MXF file—less data wrapping and unwrapping to access and insert tracks.

Let us consider an example. Assume a 1-hour MXF program file with interleaved A + V (with 50Mbps MPEG essence). Accessing, modifying, and restoring an audio track requires these operations: the audio track inside the MXF file is demuxed/removed, modified by some audio edit operation, and remuxed back into the same MXF file. As might be imagined, these are data-intensive operations and may involve 45GB of R/W storage access even though the target audio track is about 650MB in size. When the GOLF is used, only the target

audio track is retrieved, modified, and restored with a huge savings in storage access time. Also, the inventory file may be updated to indicate a new version of the audio file.

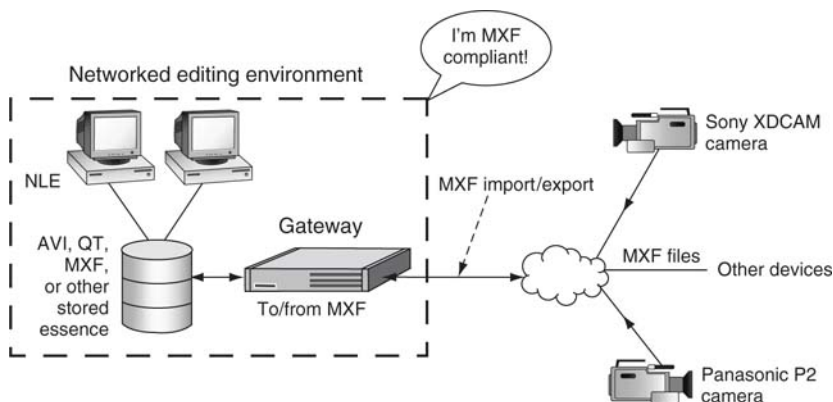
For some applications and workflows, working with a GOLF is simpler and more efficient than using a fully multiplexed, or “all-in-one,” MXF file. In general, access and restore are easier using the GOLF, especially partial file/track access. There is room for MXF-centric and GOLF-centric designs, and each will find its application space. The Advanced Media Workflow Association ([www.amwa.tv](http://www.amwa.tv)) is defining a specific implementation scenario for a GOLF. It is called MXF Versioning (AS-02). In the design all external essence, other data types, and the master inventory file are of MXF format.

### 7.2.2 Working with MXF and Interoperability

It’s interesting to note that one of the goals for MXF is to foster file interchange and interoperability between users/equipment and not to define an on-disc A/V format. What is the consequence of this decision?

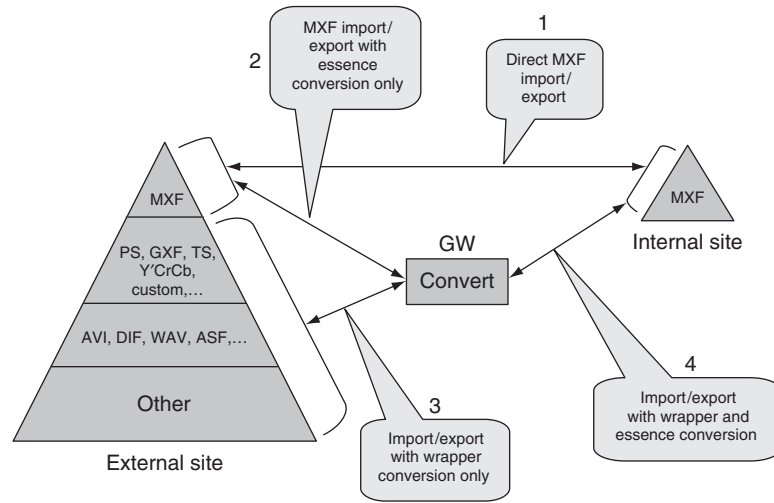
Equipment vendors are not required to store MXF as their *native file format*. Consider a networked editing system that stores all A/V essence in the AVI format. If external users expect to connect to this system via a NAS and directly access stored MXF files, they will be disappointed. However, if the system supports MXF file import/export using FTP, then these same users will be able to exchange MXF files, even though the storage format is natively AVI. Of course, this requires an AVI/MXF file conversion step as part of the import/export process. As a result, MXF does not guarantee interoperability for all system access modes, but it is a step in the right direction. Some vendors have chosen to work with natively stored MXF, but this is not a requirement of the standard.

One example of a system with mixed formats is illustrated in Figure 7.11. The networked editing system is “closed.” The attached NLEs have direct



**FIGURE 7.11** A sample MXF interchange environment.





**FIGURE 7.12** File conversion gateway operations.

access to stored A/V essence, which may be in AVI, QT, or even MXF formats. Outsiders have access to the stored essence via the gateway. If the internal format is AVI and the system advertises MXF compliancy, then the gateway is responsible for all format conversions. External users do not need to know that the internal format is AVI for the most part. As long as the gateway provides for the import/export function, then all is well. Or is it?

There is a world of difference between *MXF compliancy* and *MXF compatibility*. The system of Figure 7.11 is MXF import/export compliant, but it may not be MXF compatible. If it meets all the legal MXF specs—how it is formed—then it is compliant. However, in a two-sided transaction, both parties need to agree on exactly what will be transferred for compatibility. Assume that an external source has an MXF file that is OP-3a formatted, MPEG4 IBP essence with no index tables. If an internal location supports only OP-1a, MPEG2 with an index table, then there is a conflict. From the standpoint of the internal site, the external MXF file is not *compatible* even if it is *compliant*.

Figure 7.12 shows a stack of possible external formats, and MXF as the preferred internal format. The pyramid relates to the relative number of files in production today with MXF at the top because it is rare. A gateway (GW) sits between the external source/sink of files and the internal source/sink. The purpose of the GW is to massage the files and make them *MXF compliant* and *compatible* for import/export. Legacy files will always be with us, so the gateway is legacy's friend.

### 7.2.2.1 The File Conversion Gateway

The more choices MXF allows for (and there are plenty), the less likely that any two-party transaction will succeed without some format manipulation. The gateway performs at least four different kinds of operations per Figure 7.12:

- **Case 1.** MXF import/export is compatible and compliant to both sides. In this case the GW does no format changes. This is the trivial case.
- **Case 2.** MXF is compliant on both sides, but the MXF essence layers are not compatible. For example, the GW may need to transcode from MPEG to DV. Another possible change is from OP-3a to OP-1a. This step may cause quality degradation and delay the transfer due to slow transcoding.
- **Case 3.** The external wrapper is not MXF compliant (may be AVI), but the essence, say DV, is compatible. The GW unwraps the DV and rewraps it under MXF. This is a fast transaction, with no quality loss, but metadata may be lost when going from MXF to a non-MXF format.
- **Case 4.** The external wrapper layer is not MXF compliant, and the essence layer is not compatible. This is the worst case and costs in quality and time. Avoid if possible.

Several vendors are providing all-purpose file gateways, among them Anystream, Front Porch Digital, Masstech Group, and Telestream. Each of these vendors either offers MXF conversion or has plans to do so.

In general, gateway operations are governed by the following principles:

- Speed of file conversion;
  - Transparent RT is ideal, but few gateways operate in RT for all operations.
- No or minimal loss of essence quality during conversion;
  - As an example, transcoding a DV/25 file to MPEG2 at 10 Mbps will cause a generation loss of quality
- No or minimal loss of non A/V information;
  - Often some metadata is deliberately left on the floor during the conversion.
- Conversion robustness;
  - Testing all the conversion combinations is often not practical. For example, cross-conversion support among only 10 file format types leads to 90 conversion pairs that need to be tested and supported.

Gateways are a fact of life, but careful planning can reduce their heavy usage. With the advent of MXF, our industry will standardize on one wrapper format. Another use of a gateway is to create a proxy file from a higher resolution file. For

example, a gateway (really a conversion engine for this example) can watch a file folder for signs of any new files. When a new file arrives, the engine can encode to, say, a WM9 file for use by browsers facility-wide. Gateways will become more sophisticated in dealing with metadata too. The field of metadata mapping between A/V formats is unplowed ground for the most part. Also, Moore's law is on the side of the transcoding gateway as it becomes faster each year.

### 7.2.2.2 *You Only Get What You Define*

Okay, so you have decided all your time-based media will be MXF. If you want to reach this goal, it is best to publish an import/export specification to set the format ground rules. With adherence to these guidelines, format compatibility is all but guaranteed. Unfortunately, some suppliers will still provide MXF files that differ somewhat from what is desired. In many cases the imported file will be compatible. In other cases a gateway is needed to force compatibility. Some of the MXF specs that should be nailed down are as follows:

- SD and HD resolutions,  $4 \times 3$  or  $16 \times 9$ , 4:2:2, 4:2:0, duration
- Video essence layer—MPEG format and type (IBP or I-only), DV, other
- Video essence compression rate (e.g.,  $\leq 50$  Mbps)
- Audio essence layer—AES/EBU, Bwave, other, number of channels
- Operational patterns—OP-1a and OP-ATOM will be the most common for many years to come
- Use of metadata—DMS-1 (SMPTE 380M), other, or none
- Streamable or not—MXF streams are not in common use
- Frame-based edit units or other segmentation
- Advanced topics: length of partitions, alignment of internal fields, index table location(s), VBI carriage, other

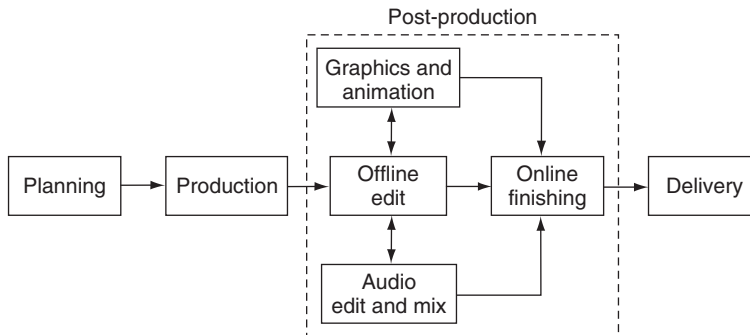
Early success with MXF depends on sticking to a formula for MXF file parameters. Without an interoperability document, MXF interchange quickly becomes a bad dream. See too the work that AMWA.TV is doing to specify an MXF Program Delivery specification (AS-03).

## 7.3 ADVANCED AUTHORING FORMAT (AAF)<sup>2</sup>

AAF is a specialized metadata file format designed for use in the postproduction of A/V content. Imagine a project where three different people collaborate on the same material. One person does the video edits, another does the audio edits and mix, and a third does the graphics. They all need to see the work of the others at different stages of the development. Figure 7.13 shows a typical

---

<sup>2</sup> This section is loosely modeled after and paraphrased from parts of Chapter 6 (AAF) by Phil Tudor, *File Interchange Handbook* (Gilmer).



**FIGURE 7.13** Workflow to create typical A/V program material.

workflow for such a production. The following is a list of common operations used in post workflows:

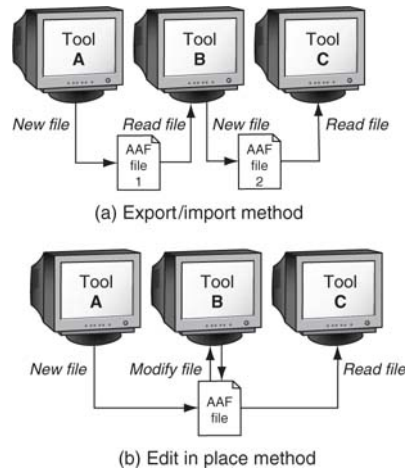
- Editing materials
- Visual effects
- Compositing
- Animations
- Mixing audio
- Audio effects
- Captions

It is obvious that a common language for material interchange is needed. At the essence level, MXF meets the need, but what is the best way to describe the assembly of the material? How are all the edits, mixes, compositions, effects, captions, and so on assembled to create the final program? This is where AAF comes in. At the most basic level, AAF is categorized as an edit decision list (EDL) format. Because it is a record of every edit operation, an EDL plus the essence completely defines the media project at any stage of development. Many proprietary EDLs exist with little interoperability and limited feature sets. There is a need for an open, extensible, full-featured “super EDL,” and AAF meets these needs.

AAF was developed as a response to the SMPTE/EBU Task Force’s recommendations for such a format. In time, the AAF Association (rebranded as the Advanced Media Workflow Association, AMWA) took up the mantle to manage AAF’s development as an open format although technically not a standard. The Association also promotes AAF through a series of awareness events.

### 7.3.1 Methods of AAF File Interchange

AAF supports two methods to interchange edit information between assembly tools. These are the import/export and the edit-in-place models. Figure 7.14 shows the two methods. With the import/export method (top of Figure 7.14), tool



**FIGURE 7.14** AAF export/import and in-place-edit interchange methods.

A creates a new AAF file that is read by tool B. Tool B creates a new AAF file that is read by tool C. The two interchanges are independent. This model is appropriate for simple exchanges between two tools but has limitations for moving data among multiple tools. However, the edit-in-place method allows for any tool to read/modify a common AAF file as needed. Any data created by tool A can be read by tool C with no dependence on tool B. When AAF files are created or modified, a record of each application's operations is made to leave an audit trail.

AAF supports internal or external A/V essence. Internal essence is practical for small projects. For larger ones, keeping the essence external (e.g., MXF) is desired. This is especially true if there are many essence files. Loading all essence into a single AAF file could easily create an impenetrable 50 GB file for just a few hours of raw video essence.

### 7.3.2 AAF Reference Implementation

The AMWA provides an open source software reference implementation of the AAF specification (the AAF SDK). It is distributed as C++ open source code with ports to several computer platforms. The reference implementation is recommended for use in products to reduce compatibility problems when crossing between different vendor implementations.

MXF and AAF share some common technology. MXF reuses a subset of the AAF object specification but maps it differently using KLV (SMPTE 336M) encoding. Parts of the object specification dealing with clips and source material are reused in MXF; parts dealing with compositions and effects are removed. When a common data model is used, metadata in an MXF file are directly compatible with AAF. This allows AAF and MXF to work in harmony across a broad range of applications.

**Table 7.1** Constrained Effects: Defined for Interoperability Using AAF

Dissolve effects	Layered 2D DVE effects
Wipe effects	Key effects
Motion effects	Alpha channel matte key definition
Frame repeat effects	Alpha key over video
Flip and flop effects	Luminance key
Spatial positioning and zooms	Chroma key effect
(2D DVE) including:	Audio gain effects
Moving the image	Audio clip gain and track gain
Cropping the image	Audio track pan effect
Scaling the image	Audio fade effect
Rotating the image	
Corner pinning	

There are virtually no limits to the types of effects that a composition may contain. Vendor A may offer a super 3D whiz bang effect that vendor B does not support. In this case, how does AAF help because the effect cannot be interchanged? While it is true that not every possible effect is transportable between tools, AAF supports a subset of effects that meets the needs of most creative workflows. The AAF edit protocol defines this practical subset.

The edit protocol is designed to codify best practices for the storage and exchange of AAF files (McDermid). It constrains the more general AAF to a subset of all possible operations to guarantee a predictable level of interoperability between tools. One area that requires constraint is effects. Interchanging effects is one of the most challenging aspects of AAF. Table 7.1 shows the classes of defined effects supported by the edit protocol. Other effects will need to be rendered into a video format before interchange. In the end, AAF is a life saver for program production across a collaborative group. AAF levels the playing field and gives users an opportunity to choose their tools and not be locked into one vendor's products.

## 7.4 XML AND METADATA

eXtensible Markup Language (XML) has become the lingua franca of the metadata world. When XML became a standard in 1998, it ushered in a new paradigm for distributed computing. Despite its hype, XML is simply a meta language—a language for describing other languages. For the first time, XML enabled a standard way to format a description language. Its use is evident in business systems worldwide, including AV/IT systems. XML makes it possible for users to interchange information (metadata, labels, values, etc.) using a

standard method to encode the contents. It is not a language in the sense of, say, C+ or Java but rather one to describe and enumerate information. Let us consider an example to get things started.

Your vacation to London is over. You took plenty of video footage and now it is time to describe the various scenes using textural descriptions (descriptive metadata: who, what, when, and where). When you use XML, it may look like the following:

```
<other XML header code here....>
<vacation_video>
<location> London, August, 2009 </location>
<scenes>
<time_code> 1:05:00:00 </time_code>
<action> "arriving at our South Kensington hotel" </action>
<action> "strolling down Pond St in Chelsea" </action>
<action> "walking along the King's Road with Squeak and Dave"
  </action>
</scenes>
<scenes>
<time_code> 1:10:12:20 </time_code>
<action> "visiting the Tate Modern Museum" </action> <action> ... and
  so on ...
</scenes>
</vacation_video>
<other XML footer code here...>
```

The syntax is obvious. All the information is easily contained in a small file, e.g., London-text.xml. Importantly, XML is human readable. The labels may take on many forms, and these are preferably standardized. Several groups have standardized the label fields (<scenes>), as described later. For example, one of the early standards (not A/V specific) is called the Dublin Core. The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems ([www.dublincore.org](http://www.dublincore.org)).

Due to the popularity of XML, there are tools galore to author, edit, view, validate, and do other operations.<sup>3</sup> Because editors and producers do not want to be burdened with the details of XML, the A/V industry is slowly creating high-level applications (authoring, querying, browsing) that use XML under the hood.

---

<sup>3</sup> For more information on some key XML definitions, learn about the XML schema and namespaces as defined by the W3C at [www.w3c.org/xml](http://www.w3c.org/xml).

Querying metadata is a very common operation. Let us assume a collection of 10,000 XML files each describing associated A/V essence files. What is the best way to query metadata to find a particular scene of video among all the essence? One customary method is to extract all metadata and load into a SQL database. A database query is supported by a variety of tools and is mature. Is it possible to query the 10K files directly without needing a SQL database? Yes, and one tool to assist is XQuery.

XQuery is a query language specification developed by the World Wide Web Consortium (W3C) that is designed to query collections of XML data or even files that have only some XML data. XQuery makes possible the exciting prospect of a single query that searches across an incoming A/V metadata file in native XML format, an archive of catalog data also in native XML format, and archived metadata held in a relational database. It will take some time for the A/V industry to appreciate the value of this important new query language.

Many professional video products offer some fashion of XML import/export. Descriptive metadata is the lifeblood of A/V production for documenting and finding materials. Expect XML and its associated metadata to touch every aspect of A/V workflow. From acquisition to ingest/logging, editing, browsing, archiving, and publishing, metadata is a key to managing media. Several industry players are defining how to use XML schemas to package metadata. The next section outlines some current efforts.

### 7.4.1 Metadata Standards and Schemas for A/V

We are at the cusp of standardized metadata that crosses tool and system boundaries. MXF supports SMPTE 380M for metadata descriptions. Also SMPTE supports the Metadata Dictionary, RP210. This is an extensible dictionary that may be augmented by public registry. SMPTE is also crafting XML versions of MXF metadata. In addition, the A/V industry has developed several metadata frameworks, each with its own strength.

The BBC has defined a Standard Media Exchange Framework (SMEF) to support and enable media asset management as an end-to-end process across its business areas, from production to delivery to the home. The SMEF Data Model (SMEF-DM) provides a set of definitions for the information required in production, distribution, and management of media assets, currently expressed as a data dictionary and set of entity relationship diagrams.

The EBU ([www.ebu.ch](http://www.ebu.ch)) project group, P/META, defines and represents the information requirements for the exchange of program content between the high-level business functions of EBU members: production, delivery, broadcast, and archive. The P/META scheme provides defined metadata to support the identification, description, discovery, and use of essence in business-to-business (B2B) transactions. Their work effort is based on an extension of SMEF.

MPEG7 is an established metadata standard for classifying various types of multimedia information. Despite its name, MPEG7 is not an A/V encoding



standard such as MPEG4. MPEG7 is formally called a “Multimedia Content Description Interface.” For an overview of MPEG-7 technologies, see (Hasegawa 2004). The standard supports a wide range of metadata features from video characteristics such as shape, size, color, and audio attributes such as tempo, mood, and key to descriptive elements such as who, what, when, and where. MPEG7 has found little use in professional A/V production so far. However, it has found application by the TV-Anytime Forum (personal video recorder products). Their defined metadata specification and XML schema are based on MPEG7’s description definition language and its description schemas.

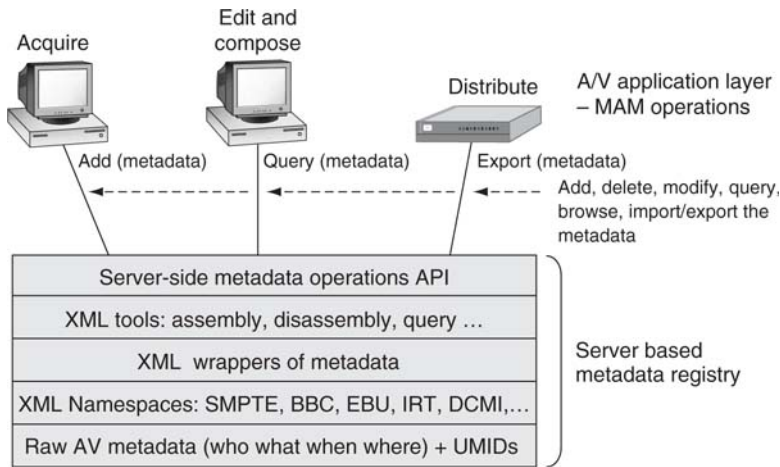
Finally, the Institut für Rundfunktechnik (IRT) in Munich and MOG Solutions ([www.mog-solutions.com](http://www.mog-solutions.com)) have codeveloped an XML mapping of the MXF metadata structures (DMS-1). The IRT has also developed an A/V metadata framework specific to A/V production applications. These are not yet standardized solutions but will likely find applications in some quarters.

### 7.4.2 The UMID

Without a way to identify metadata types and essence files explicitly, they quickly become lost in a sea of data. The Unique Material Identifier (UMID, SMPTE 330M and RP 205) is a global way to identify A/V materials. It is the kingpin in the quest for a universal way to unambiguously tag every piece of essence and metadata. UMIDs identify every component part of a program and provide a linkage between the essence and its associated metadata. The UMID is a 32-byte (64 in extended form) field with the potential to identify every A/V file with a granularity of frames if desired. For example, the UMID hex value of #A214F07C could represent the unique A/V essence of NASA’s original video of Neil Armstrong’s first step on the moon. MXF relies on UMIDs for content ID. Some of its characteristics are

- It is a globally unique identifier.
- It identifies any level of material granularity, from a single frame to a completed final package.
- It can be automatically and locally issued, which means that access to a central database or a registration authority is not needed.
- It may be used in different applications, i.e., not only as a global material identifier, but also as a local identifier with some specific local applications.

Figure 7.15 puts all the concepts together in a metadata registry example. It is server based, stores XML metadata, and provides for common client metadata operations. The different layers describe functions and aspects needed to implement a searchable metadata repository. There are no standardized and commercially available metadata application servers on the market. Each vendor offers something unique and fine-tuned for its products and supported workflows.



**FIGURE 7.15** XML-centric metadata registry: Server based.

Metadata management solutions range from hand-searched lists to federated networked systems with millions of metadata entries. No one architecture, schema, or vendor solution has won the hearts of all A/V users. Time will tell how metadata management solutions will pan out and what schema(s) becomes the king of the hill. Admittedly, several may rise to the top, as there is room for specialized schemas across the range of A/V businesses.

### 7.4.3 ISAN and V-ISAN Content ID Tags

The ISO has standardized the International Standard Audiovisual Number (ISAN) as a 64-bit value to identify a piece of programming. The ISAN goes beyond the UMID by providing fields for owner ID, content ID, and episode number. It should be embedded into the material (a watermark is one way) so that the ISAN value and content it points to are inseparable. Think of the ISAN value as representing a collection of A/V objects that are in total a program. V-ISAN is a 96-bit version that includes the version number, indicating language, edited for TV rating, and subtitles.

Metadata and their associated tools are only small cogs in the big wheel of media asset management (MAM). In what way is MAM part of the AV/IT revolution? Let us see.

### 7.4.4 ISCI and Ad-ID Identification Codes

The Industry Standard Commercial Identifier (ISCI) code has been used to identify commercials (aka “spots”) aired worldwide. It found application by TV stations, ad agencies, video post-production houses, radio stations, and other related entities to identify commercials for airing. The ISCI system is compact, allowing only 8 bytes to identify a commercial and its owner. This 30-year-old system is no longer adequate in a world of digitally addressable media.

ISCI is being replaced by the Ad-ID ([www.ad-id.org](http://www.ad-id.org)) code with 12 bytes, 4 alpha and 8 alphanumeric. The first 4-byte alpha field identifies a company (the producer), and the second 8-byte field identifies a unique spot. Ad-ID codes are computer generated through a secure, Web-accessible database. All existing ISCI prefixes are grandfathered into the Ad-ID system.

7.5 MEDIA ASSET MANAGEMENT

There is an old saying in the A/V business that goes something like this: “If you have it but can’t find it, then you don’t have it.” With a MAM solution, enabled users can—ideally—quickly and easily locate content they possess.

With the proliferation of media assets and Web pages with embedded A/V, MAM solutions are becoming commonplace in business. More generally, digital asset management (DAM) solutions (not media centric) are used to manage text documents with graphics. Think of MAM as DAM with the ability to manage time-based media. In the big picture, both MAM and DAM are content management (CM) concepts. According to a Frost & Sullivan report, the worldwide MAM market will grow to \$1.37 billion in 2010 at a compound annual growth rate estimated to be 20.2 percent. The overall market includes all types of media production and delivery, including Web based.

One definition of MAM is the process of digitizing, cataloging, querying, moving, managing, repurposing, and securely delivering time-based media and still graphics. It supports the workflow of information between users for the creation of new and modified media products.

But what is a media asset? On the surface, any media that sit in company storage may be considered an asset, but this is far from the truth in practice. Figure 7.16 shows the asset equation: an asset is the content *plus* the rights to use it. Many broadcasters have shelves full of videos that they cannot legally play to air because the use contract has expired. Also, the content is the essence *plus* the metadata that describe it. In the end, both metadata and rights are needed to fully qualify and manage a media asset. In fact, we need to modify the opening quote to reflect the true reality: “If you have it and can find it but with no rights to it, then you don’t have it.”

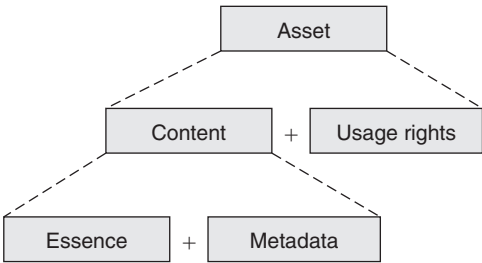


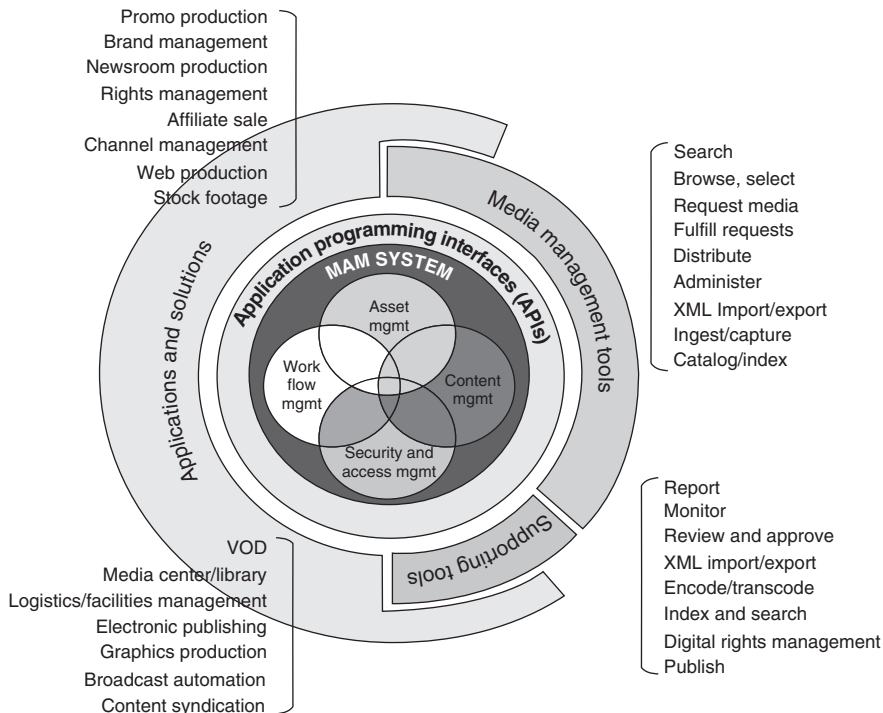
FIGURE 7.16 The asset equation.

Rights management is a complex topic. It involves aspects of copyright law, contracts, payments, and windows of use and reuse. A single program may have sequences each with its own rights clauses. Rights age, so what is legal today may not be tomorrow. All aspects of production require knowledge of media rights. See Section 7.5.3 for a primer on DRM.

### 7.5.1 The MAM Landscape

This section examines the major components in a MAM solution. Figure 7.17 outlines the MAM onion. The outer layer represents the applications and solutions needed by business processes. A/V-related ones are

- A/V production
- Broadcast automation
- News production
- Web production
- Rights management
- Video on demand
- Graphics production
- Content syndication



**FIGURE 7.17** The MAM onion.

Source: Perspective media group.

Each of these application areas may require a full-featured MAM system. The next layer comprises tools for ingesting, browsing, querying, and so on. Applications make use of these features as needed. Augmenting user functions are the support tools for reporting, reviewing/approving, publishing, and so on. Applications and tools are connected to the center of the diagram using defined APIs. Finally, in the center are the core processes as listed. One area not yet discussed so far is workflow management. This is a relatively new frontier for A/V production and provides methods to manage an entire project from concept to delivery. Workflow methods are examined later in the chapter.

No doubt, full-featured MAM solutions are complex. It is very unlikely that a shrink-wrapped MAM solution will meet the needs of any large real-world business.<sup>4</sup> Open market MAM solutions rely on customization to meet existing business process needs. Also, there are many vendor-specific aspects of these systems from video proxy formats (MPEG1, WM9/VC1, VC-3 for HD, MPEG4, etc.) to metadata formats (DCMI, SMEF, IRT, SMPTE, custom, etc.) to the APIs and middleware that connect all the pieces together. See (Cordeiro 2004) for insights into a unified API for MAM. Ideally, the MAM system should fit like a glove with the existing A/V systems architecture with its formats, workflows, control, and applications use. Unfortunately, the rather liberal use of “standard” formats prevents MAM systems from interoperating at even the most basic levels. Upgrading a MAM system from vendor A to B is a painful and often impossible task, so choose your MAM system wisely because you will live with it for a long, long time.

Choosing a commercial MAM system for a legacy business requires a large dose of compromise and realignment of internal processes to the abilities and functions of the MAM system. Many media operations have developed totally custom solutions because open market ones were not sufficient. Of course, when you are developing a new complex workflow from scratch, it is wise to base it on available MAM functionalities to enable the use of open market solutions.

### 7.5.2 MAM Functions and Examples

To fulfill the needs of a full-featured MAM system, the following functions (Abunu 2004) are required:

- One-time media capture and indexing of metadata (including rights, program information, usage, etc.) made accessible to all workflow participants.
- Standards for media assets for interoperability across the workflows.
- Implementation of a metadata set to support all workflow operations.

---

<sup>4</sup> Shrink-wrapped MAM solutions often meet the needs for simple workflows (Web page asset management) with a small number of users.

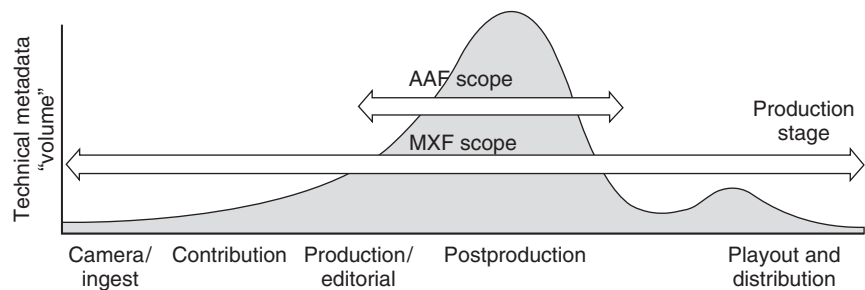
- Search functions to identify and locate A/V essence. This may range from a simple file name search to a query based on people, places, things, and activities.
- Shared views and access to media across an organization mediated by access control.
- Media life cycle management—from ingest to composing to converting to archiving.
- Workflow process support—assignments, approvals, releases, ownerships, usage history.
- Functionalities to package and distribute media according to business needs.

Exploring the intricacies of these items is beyond the scope of this book. However, to learn more about the details of MAM functionality (with support for time-based media and focus on broadcast and A/V production), study the representational offerings from companies such as Artesia Technologies, Avid Technology, Blue Order, Harris Broadcast, IBM (Content Manager), Microsoft (Interactive Media Manager), Omnibus, and Thomson. Although not media focused, Drupal (<http://drupal.org>) is a popular free software package that allows an individual or a community of users to easily publish, manage, and organize a wide variety of content for Web sites.

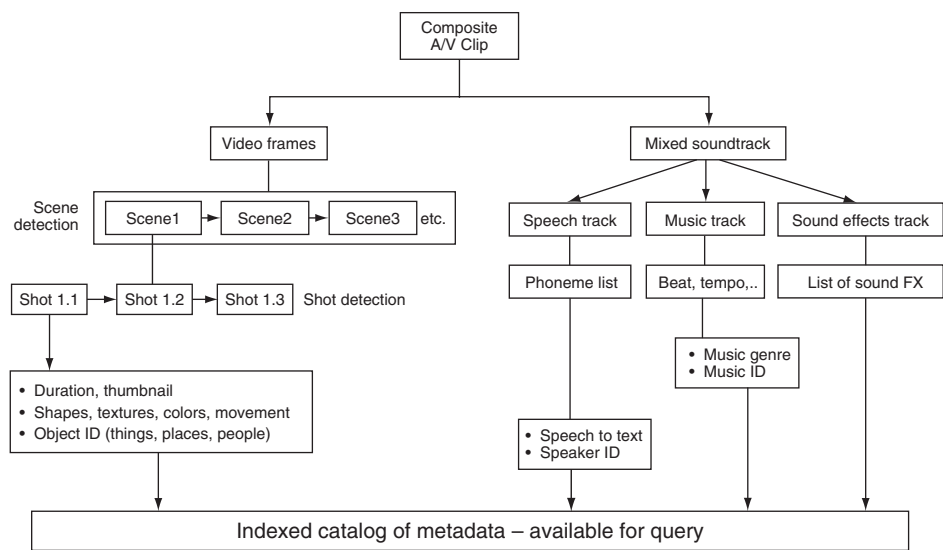
A classic case of integrating a MAM with A/V editing gear occurs in broadcast TV news production systems. Today many media companies from a large CNN to a small local TV station rely on IT-based news production systems for day-to-day operations. For example, Avid, Thomson/GVG, Harris, Quantel, and Sony all offer a range of IT-based news production systems incorporating a restricted MAM. These systems support end-to-end unified workflows from ingest to play-to-air of news stories. Metadata management plays a big role in these systems. Most of the traditional automation vendors also offer MAM as part of their overall product portfolio. It is paramount that the metadata and their format generated at video capture be usable throughout the workflow chain.

#### **7.5.2.1 Example of an Index/MAM Query Operation**

This section examines the indexing and querying operations. These are two common operators in any MAM system. Indexed and cataloged metadata are the lifeblood of any asset tracking system. Figure 7.18 illustrates the relative volume of metadata versus positions in the media workflow. As a project develops, the metadata volume increases to a peak during the editing and compositing stage. Little metadata are produced or consumed at either end of the production cycle. However, this trend may change as metadata methods become more mature. In the future it is likely that more descriptive information will be produced at image capture time.



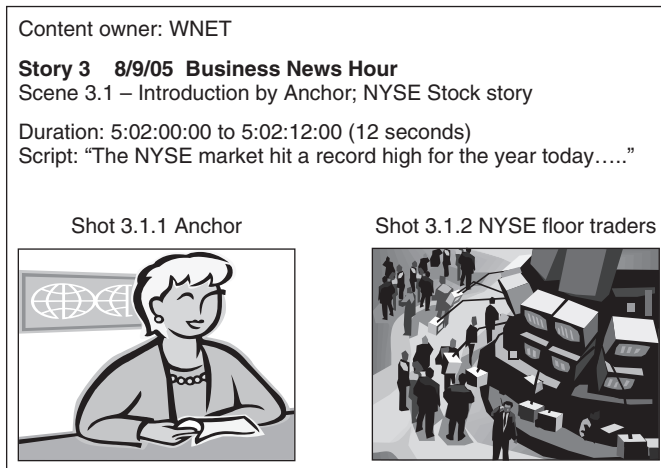
**FIGURE 7.18** Volume of technical metadata associated with a clip during production.  
Source: File interchange handbook, chapter 5.



**FIGURE 7.19** A/V media indexing hierarchy.

Human entry is often the most accurate and certainly the most detailed. It is also expensive and time-consuming. Ideally, an automatic indexer will identify some of the more common elements of a scene as outlined in Figure 7.19. It will be a long time before a machine can describe the subtle interaction among four people playing poker. Still, indexing technology is steadily improving and already generates a good deal of searchable metadata.

Figure 7.19 provides a divide-and-conquer approach to indexing an A/V clip. Some of the operations are straightforward and mature, such as shot detection, whereas others are state of the art, like speech to text in the presence of music. In the realm of science fiction is face recognition in a dense crowd. For less demanding scenes, such as TV news anchor ID, it is practical today. For each element there is a defined metadata type. The more powerful the indexer, the more valuable the searchable metadata. A/V indexing is a hot area of university research.



**FIGURE 7.20** Typical query response to: "Find: NYSE stock news, market high."

An example of a query response is illustrated in Figure 7.20. In this case the query was "Find: NYSE stock news, market high." Assuming the required metadata exist in a catalog, then the response is formatted as shown. Because there are no standards for query method or response format, these will remain custom methods for years to come, although an XML-formatted response would make sense. Once the material is located and the access rights determined, then it may be incorporated into a project.

### 7.5.3 Using DRM as Part of a MAM Solution

Digital rights management (DRM) is a double-edged sword. Content providers claim they need it to protect their files from illegal use. However, content consumers often find DRM a royal pain, and it limits their legitimate use of the files. Although not currently used in bulk for professional applications, DRM has a place in the broadcast and A/V workflow. Today, for the most part, contracts or custom solutions are used to define rights usage at the production level. However, traditional contracts tend not to be machine friendly, whereas DRM technology is machine friendly.

The following are common features of a DRM-protected file:

- **Encrypted file content**—Only users with an owner-provided key can open the media file.
- **Rights use**—Time windows, platforms (desktop, mobile, etc.), number of viewings, copyrights, sharing rights, and so on.
- **License granting**—Provided with file, on-demand, or silent background methods to obtain the license/key to use a file(s).



Think of DRM as a workflow element, not just a file use enforcer. A total DRM environment includes contract authoring, file encrypting, license and key distribution, and runtime contract enforcing. So why use it in the professional domain? If you cannot afford to lose control of your distributed media, then consider DRM as one way to manage it. A compromise to a full-featured DRM is to use only file encryption and manual key transfer. This achieves a level of protection without all of DRM's features.

One promising technology is from the Open Digital Rights Language initiative (<http://odrl.net>). This group has developed a rights expression language for general use, and it may find application in professional production MAM systems.

7.5.4 Tastes Like Chicken

The single most important factor in leveraging all things digital is smooth, efficient workflow. Nearly every component in a well-designed project workflow has the MAM stamp on it. MAM functionality is the glue that ties all the pieces together. Figure 7.21 outlines the various classes of MAM products, tools, and solution providers. There is no such thing as a one-size-fits-all product or solution.

When specifying MAM functionality for a project, think holistically. MAM should not be some add-on, plug-in, or attachment, but rather its presence should be felt systemically at all levels of the design. Imagine MAM as a personality feature of a well-designed workflow. For new designs, MAM functionality should be spelled out as part of the overall workflow, not only on a per component basis. Be specific as to what formats, operations, scale, and UI functionality are needed. Especially give care to the total interoperability among the various components. Also, be a realist. Your idea of the ideal workflow will not necessarily map into what is available commercially. It is often smarter to evaluate

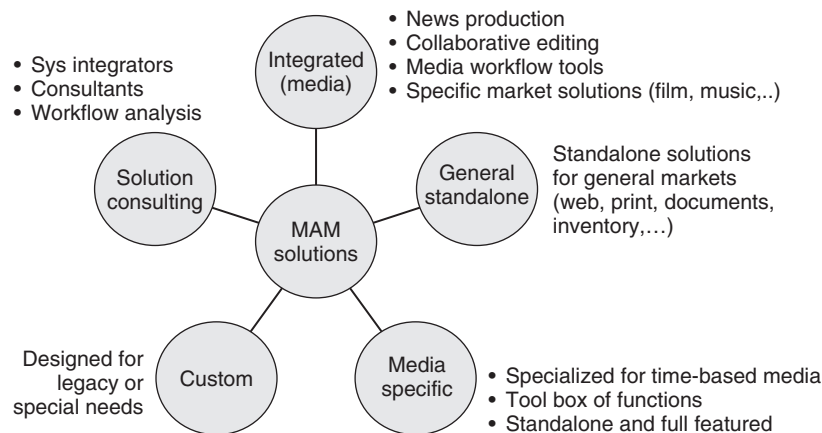


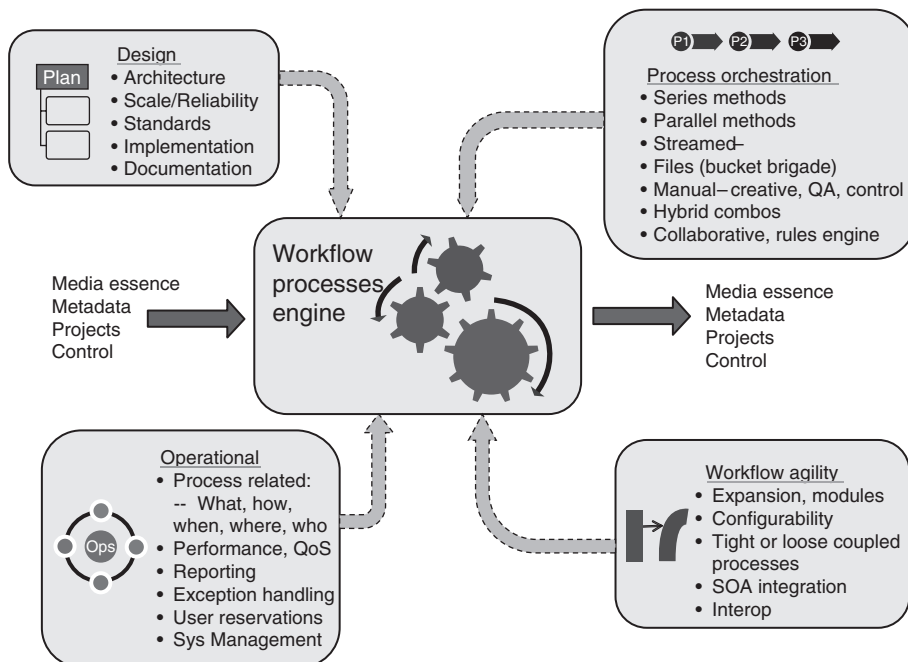
FIGURE 7.21 The MAM product and solution landscape.

what is available and then pattern your workflow accordingly. Workflows also deserve mention at this point, so let us consider some of these aspects.

## 7.6 THE FUNDAMENTAL ELEMENTS OF MEDIA WORKFLOWS

Workflows are found everywhere in life: cooking a meal, washing a car, planting a rose bush, producing a Hollywood movie. Basically, workflow is defined as “a set of sequential steps needed to complete a job.” Some workflows are easy to implement, needing only a few tools, whereas others demand mountains of infrastructure and human capital. This section concentrates on workflows for media systems from the small edit shop producing wedding videos to a large studio creating movies.

Figure 7.22 shows the five domains or precincts of interest for media workflows. The central box represents the actual workflow steps; do this, now do that, and so on. The other four domains help define the “what and how” of workflow functionality. Some of these are common to all workflows, such as operational elements (step design, timing, tool needs, resource availability, review, etc.), whereas others are specific to media systems such as A/V streams and file-related processes.



**FIGURE 7.22** The essential elements of media workflows.

Next, let's examine each of the elements in Figure 7.22: design, process orchestration, operational, and workflow agility. The constituent elements of these will be explored with examples. The end goal is to provide a simple high-level checklist to refer to when you are building a new workflow or modifying an existing one. For sure, this coverage is not exhaustive. Not every aspect will be examined; don't expect to become an expert. However, you will be versed in the language and concepts of media flows and their design.

7.6.1 The Design Element

Any viable workflow needs a design stage. The key elements of interest to us are as follows:

- **Architecture**—What solution space do you need?
- **Reliability/Scale**—14 methods for building reliable systems.
- **Standards and Interoperability**—SMPTE, IETF, IEEE, ITU-T, W3C, AMWA, and so on.
- **Implementation**—Choice of vendors, systems integrator, support.
- **Documentation**—Workflow design, not just wiring and layout!

The famed Chicago skyscraper architect Louis Sullivan said, "Form follows function." This simple yet powerful axiom applies to media workflows as well as skyscraper design. In essence, define your workspace and design to it. For our case, allow for growth, support agility, high availability as budget permits, and document not only layout/wiring but flows too. Figure 7.23 shows a typical generic flow for program production. This could be modified for broadcast, live event production, news, digital intermediates (DIs), or any number of flows.

This figure may be the first step in defining the workspace for a new project. The level of detail is intentionally high. The design architecture will support

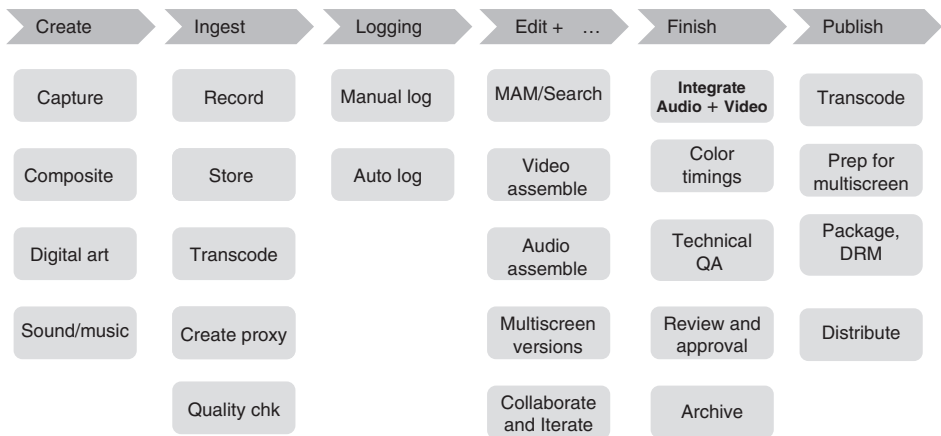


FIGURE 7.23 Generic program creation process flow.

these functions at a minimum. As the designer applies the divide-and-conquer rule, each process is implemented with the end goals in mind of one unified infrastructure, not islands of operations. Any given process may connect to central storage, application servers, a LAN/WAN, or traditional A/V router.

### 7.6.1.1 *Designing for Reliability*

Let's consider the aspect of designing for reliability, the second point in the earlier list. Given the acceptable downtime, a designer may select the options outlined in Chapter 5 covering the salient aspects of building an infrastructure for high availability (HA). For the ultimate doomsday, bulletproof system, up to 14 HA techniques (outlined in Chapter 5) could be applied at once. This is hardly practical, but some real-world, mission-critical systems come close. Normally, a few methods are applied and are determined by business needs for the most part.

Every design should have a *service availability* goal as a percentage of uptime. For example, 99.9999 percent uptime or ~32 seconds per year of downtime could be a goal. This value is achievable and allows for one (or more) serious failure(s) with 32 seconds available (or less) to route around the failed element. Another approach is to decide what length of time a system can afford to be down ("off air") and design from that value.

### 7.6.1.2 *Standards Ubiquity*

No practical system should be constructed without applying standards from a broad range of sources. Gone are the days when SMPTE standards were the only glue needed to create a video system. Today, in addition, we need the standards from IETF (Internet protocols related), W3C (XML, HTML, Web services, etc.), and the IEEE (Ethernet and real-time versions), plus others. User groups such as the Advanced Media Workflow Association ([www.amwa.tv](http://www.amwa.tv)) have a mission to create best practices and recommended technologies for implementing workflows in networked media environments. They are currently developing specifications for the Advanced Authoring Format (AAF), a MXF component inventory method, and a constrained MXF version for program delivery to broadcasters.

Another aspect related to standards is the selection of video formats for ingest, proxy, editing/compositing, distribution, and archive. A common case is that ingest, editing, and archive formats are all identical, with distribution almost certainly being different.

Finally, there is the need for common metadata formats across the workflow. This goal is not easy to achieve, and not all vendors support the same metadata formats and interpretations of their meaning. SMPTE is still grappling with defining the lingua franca for an industry-accepted metadata set. It has made good progress in defining the RP210 metadata dictionary (a glossary of terms), but more is needed. For example, some groups want a constrained, small set of media metadata (the equivalent of the famous Dublin Core for library use), whereas others want a much larger accepted set. Both groups have

valid concerns and business needs. If all goes well, both constituencies will have their needs meet.

### 7.6.1.3 Workflow Documentation Methods

Finally, under the design banner, let's consider documentation. For many years facility documentation was a collection of diagrams showing equipment layout and racking with detailed wiring and inventory diagrams using tools such as VidCAD. This level of documentation is still necessary but not sufficient to describe the workflow component of an installation. True, not every install will require a workflow diagram, but many will. If the workflow is complex with disparate processes and time-related dependencies, then the following methods should be of value.

Workflow stakeholders include the analysts who create and refine the processes, the technical developers responsible for implementing the processes, and the technical managers who monitor and manage the processes. Two diagramming methods are gaining acceptance to define process flow; one is based on Unified Modeling Language (UML) and another on Business Process Modeling Notation (BPMN).

The UML offers the following diagram types:

- Activity (do this, then do that if ...)
- Sequence (order of operations)
- Communication (messaging methods, participants)
- Timing (timelines, due dates, schedules, time dependencies, etc.)

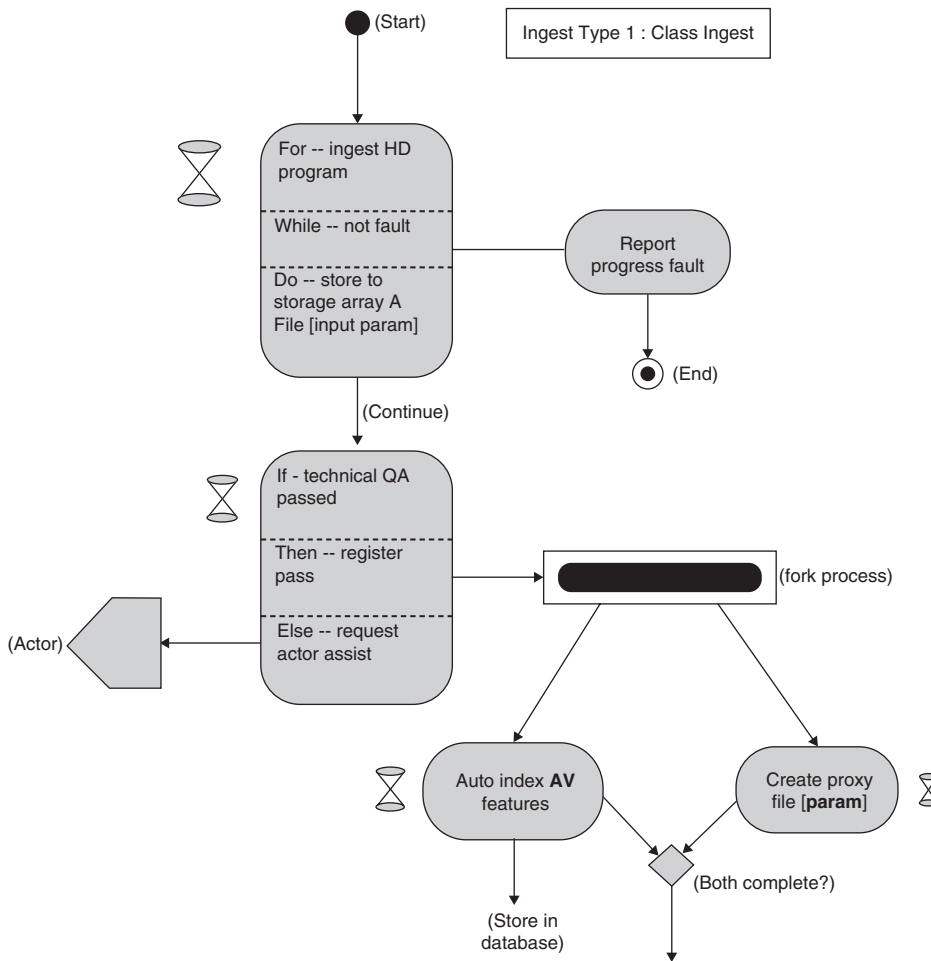
Which one should be used to describe a media workflow? Well, each has a special purpose, so depending on what aspects are to be communicated, the choice flexible. The activity diagram offers the most flexible layout for media workflow, but all will find usage. See [www.uml.org](http://www.uml.org) for more information and examples. See Figure 7.24 for an example of using an activity diagram.

Another graphical modeling tool is Business Process Modeling Notation. Don't be put off by the word *business* in the name. The tools translate to media workflows as proven by their usage by some systems integrators. The modeling in BPMN is made by simple diagrams with a small set of graphical elements. The four basic categories of elements are as follows:

- Flow Objects—Events, activities, gateways
- Connecting Objects—Sequence flow, message flow, association
- Swim-Lanes—Pool, lane
- Artifacts—Data objects, group, annotation

These four categories of elements enable designers to describe media flows. See [www.bpmn.org](http://www.bpmn.org) for more information and examples.

Fortunately, there are reasonably inexpensive layout tools for both UML and BPMN. Visio and SmartDraw can author UML diagrams. Process Modeler for



**FIGURE 7.24** UML workflow activity diagram.

Visio from ITP Commerce has received excellent reviews as a BPMN authoring tool. In addition, a plethora of free tools is available for both.

### 7.6.2 The Process Orchestration Element

This section discusses process orchestration referred to in the upper right box of Figure 7.22. There are three primary media transfer methods using networked media techniques. They are covered in some detail in Chapter 1. In review, the methods are (1) pure file transfer, (2) real-time streaming, and (3) storage access.

These three methods are the building blocks for the modern media facility. Designers must use wisdom when selecting one method over another. A big mistake when doing a new design is to mimic a videotape workflow using

networked media. Videotape flows are limited in many ways and networked media allow for many dimensions not permitted using only tape.

7.6.2.1 Comparing Flow Types

Next, let's compare three flow types: one using pure streaming and two using file transfer. Figure 7.25 illustrates these flows.

The general idea is to process an incoming video signal program as follows: ingest/record, apply an aspect ratio, convert side panels, add a lower third graphic, and finally output the result. The top flow is most commonly seen using SDI (or AES/EBU links for audio-related applications) connectivity. Of course, a process may be any transform or human-assisted means. For live events demanding a minimal in/out latency (few video frames), SDI streaming connectivity is king and used often.

The middle flow uses the bucket brigade method. First, the entire program is ingested and saved to storage. Then either by a file transfer between processes or via an intermediate "parking place" storage device, the program file is moved to the ARC process, then to the graphic composite overlay process, and then output. In each step, the entire file is imported to a process and then exported to the next process in the chain—hence, the bucket brigade nickname. The delay from *in* to *out* can be quite large, on the order of 10–20 minutes (total ARC and composite process delay) for a 60-minute program, not counting the time to ingest the incoming program (60 min). The faster each individual

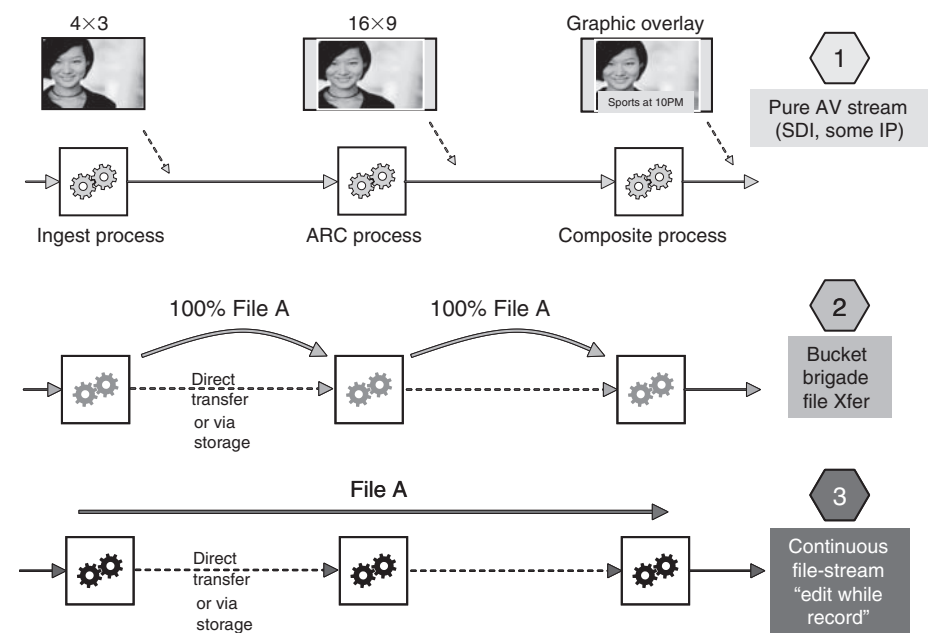


FIGURE 7.25 Three common media flow models.

process, the shorter the total in/out delay. A designer would choose this method when completion delay time is not a critical metric. For example, if the completed program is needed the following day, then by all means use this flow method. When timing is not critical, then low-speed transfers and slow processes may be used. Note, too, that individual processes don't need to work in real time, and this relaxes their design. In fact, most A/V processes can work much faster than real-time video.

The third flow is useful when the in/out delay needs to be low delay but not as short as the fully streamed flow. Basically, the target file is continuously streamed between processes. This style may be called "process-while-record," with "edit-while-record" being a common version. So, as process #1 finishes, say, the first 30 seconds of its task, process #2 can start its task and so on. No need to wait for an entire process to complete before another process in the chain can start, as with the bucket brigade. As long as the next process does not work ahead of the succeeding process, then all is well. Process timing is critical. Edit-while-record is used with some breaking-news story production. The editor can start his or her task even while the program material is being recorded with consequent faster time to air.

These flow methods may be combined in hybrid means to form serial/parallel versions. Decision steps are needed too, so a process branch may occur. A rules engine may be of use to automatically process tasks. Look for opportunities to automate common process chains. This not only saves manpower and speeds up a process chain, but also yields metrics to improve the flow.

### Flow Efficiencies

Most flows are based on moving compressed video along with uncompressed audio files and/or streams. For video, designers have a choice of pure intra-frame (short GOP, I-frame only) or inter-frame (long GOP) compression formats (see Chapter 11 for more on GOPs). Building a long GOP workflow is non-trivial since it requires great attention to frame accurate editing, effects compositing, recording, and playback. So why do it?

Long GOP required storage and data rates are typically 3–4 times less compared to I-frame only formats for equivalent visual quality. So, a 150 Mbps I-only production-quality format has roughly the same visual quality as a 50 Mbps long GOP format. One issue with long GOP formats is so-called *generational loss*. If repeated encoding and decoding cycles of the same file are needed, then the visual signal/noise ratio will decrease with consequently more artifacts as generations increase. If many repeated compression cycles are not required then quality is maintained and the long GOP workflow is very resource efficient.

#### 7.6.2.2 The Metric of Money

One common metric of interest to all facility managers is daily operational cost of a given process flow. If file transfer is involved between site locations, then



the cost of bandwidth may be a significant factor in workflow cost. We all know the proverb “Time equals money.” There are plenty of examples of this for both long (airline flights) and short (FedEx shipping) times. In the spirit of this analogy, there is a corollary to this:  $1/\text{Time} = \text{Money}$ . This is also true since  $1/T$  is rate (such as Mbps), and we pay for data transfer rates. A 1 Mbps WAN link costs substantially less than a 100 Mbps link.

One beauty of working with files is that transfer speed may be selected independently from the actual real-time nature of the video content. So a 10-minute video file may be transferred in 1 minute ( $\times 10$  speed up) or in real time at 10 minutes or 100 minutes ( $1/10$  slow down), with the actual rate selected based on need. In general, choose slow transfers over fast to save cost. Conscious effort to use slow file transfers may slash the operational costs of a workflow significantly. This variable is not available for SDI streaming. Incidentally, this metric applies to storage access rates too and not just WAN transfer rates.

### 7.6.3 The *Operational* Element

This section outlines the points in the lower left box of Figure 7.22. The main aspects relate to process-related concepts: what, how, when, where, and who. Let’s use the *create proxy* step in Figure 7.23 as an example. When you are defining a workflow process step, it’s a good practice to define each of these five characteristics for each step:

- **What** are the specs for the *create proxy* step? Encoder type, data rates, speed, file format, invoke means, input/output means, etc.
- **How** will *create proxy* be implemented? Use Vendor X’s encoder, running on Vendor Y’s server connected to storage Main. Reliability means, failover means, scale means, monitor means, Web services means, API means, etc.
- **When** will *create proxy* be invoked? Workload, duty cycle, peak/average jobs, etc.
- **Where** will *create proxy* be located? Local instantiation, remote service, contracted service, etc.
- **Who** will own and manage the *create proxy* service (A/V operations, IT operations, contracted operations), and **who** will use it (number of invokers, department, etc.)?

Resolving each of these interrogatives is good practice. The process of answering each question forces a designer to think deeply about each process. This way, hidden efficiencies may be uncovered; for example, we can share server Y with other processes, since the *create proxy* workload is small even with  $2\times$  more loading. Or, we can locate server Y in room M, since there is ample power and cooling available.

### 7.6.3.1 Performance QoS, Reservations, Exceptions

Although this topic is related to the five questions in the preceding section, it's of value to define the performance QoS, reservation methods, and error handling as separate aspects. Documenting application QoS is useful when scaling or changing a service in the future. When a service is shared (edit suite), then a reservation system with time used, billing, and resource usage may be required. Providing a systemwide reservation board (similar to a flight departure/arrival schedule display) available for all to see is often a good practice at a large facility.

Exception handling deals with HW/SW warnings, faults, errors, and status messaging. When something is out of order in one way or another, it should be registered and personnel notified to repair it. Exceptions may range from warnings of intermittent errors to out-of-spec signals (audio peaking) to resource status (90 percent full) to a complete device failure. High-end infrastructure monitoring systems are available from traditional IT vendors such as IBM, HP, CA, and others. A/V-specific solutions are available from several vendors. See, for example, the RollCall family from Snell and Wilcox and CCS Navigator from Harris.

Investing in monitoring and diagnostics products is a matter of ROI for the most part. If you are running a hobby shop, then equipment downtime may not be a big issue. If you are offering services—creative tools by-the-hour, commercial program payout—then money spent on keeping the facility up and running is worth the investment.

### 7.6.4 The Workflow Agility Element

The last major category is workflow agility, as shown in Figure 7.22. Agility is defined as “the ability to move in a quick and easy fashion, change directions quickly.” Typically, media workflows are purpose built: broadcast, post, animation, news, live events, DI, and so on. It's prudent, as discussed, for form to follow function. Nonetheless, within the bounds of scale, future changes, and reconfigurations, a workflow should be agile.

When your boss asks for a new feature set on an existing system, you want to say, “No problems. Can do.” Of course, no one can economically build a system with ultimate agility that anticipates all future requests. Still, designers can prudently craft workflows that are pliant to bounded future changes.

The most common dimensions of agility are as follows:

- **Expansion, scale**—Grow media clients, I/O, signal routing, connect bandwidth, storage, applications, reliability, reach, power, cooling, space
- **Secondary storage**—Backup, archive options
- **Configurability**—Ease in “re-wiring” a workflow to do something different and extend its functionality and life
- **Web services connectivity**—Usage, metrics, performance, reliability

- **Interoperability and reuse**—Between system components, between other systems, between partners
- **Business process interfacing**—Service-Oriented Architecture (SOA) ideals. This exposes the A/V technical layer to business processes for control, messaging, monitoring, and reporting. Too, it is created using a networked services model.

Approach every design with the idea of creating a flexible engine ready to move in a new, bounded direction as needed. Plan on allocating some project budget dollars to features that enable system flexibility.

#### 7.6.4.1 Loosely Coupled Designs

One key aspect of flexibility is the concept of *loose* versus *tight* coupled processes. Figure 7.26 illustrates this concept. Many legacy A/V systems are built on tightly coupled designs—“hard wired,” rigid systems with little flexibility for reordering or quick reuse of components. Sure, SDI and AES/EBU audio links are easily routable, but this does not constitute a loosely coupled system.

The concept of *loose coupling* is defined as follows:<sup>5</sup>

*The friction-free linking enabled by web services or equivalent. Loosely coupled services, even if they use incompatible system technologies, can be joined together on demand to create composite services, or disassembled just as easily into their functional components. Participants must establish a shared semantic framework (APIs) to ensure messages retain a consistent meaning across participating services.*

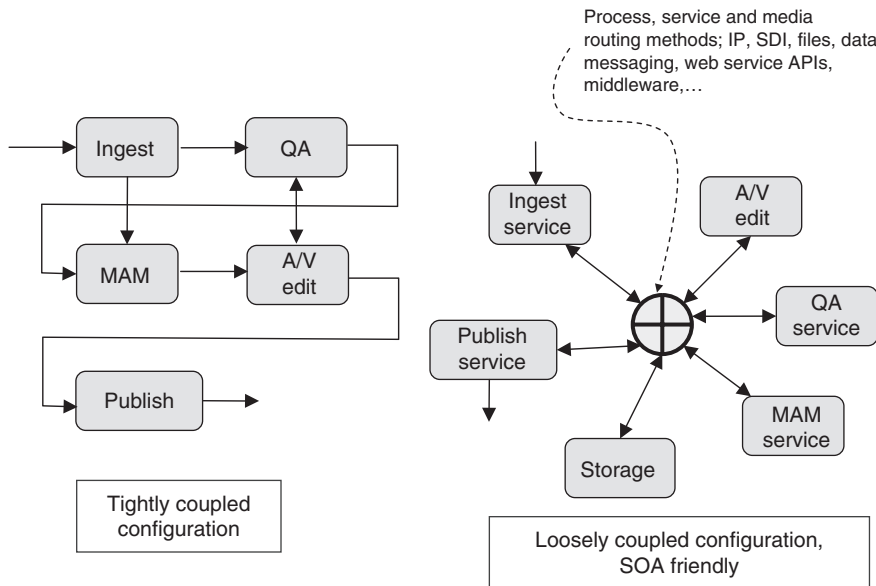
Loosely coupled designs rely on the aggregation of distributed services. A service is a networked application function available for use by others. Examples are video transcoders, media encoders, transfer engines, 3D render engines, a MAM searching engine, technical QA evaluation, and so on. Each function is available over a networked IP connection, is instantiated on an application server, provides a standardized access API, and defines a performance QoS for each client user.

Given a collection of relevant services, a designer can configure them to perform useful operations such as shown in Figure 7.26. These services are scalable, reliable (based on methods explained previously), reusable (the mantra of services design), and well defined in terms of API interfacing, data structures, and performance QoS.

One outstanding example of a services specification is defined by the W3C group. This group has defined what it calls *Web services*, and this includes specs for all aspects of interfacing to services, securing them, naming, addressing, finding, monitoring, and so on. Despite the name *Web services* (WS), these are not

---

<sup>5</sup> See glossary at <http://LooselyCoupled.com>.



**FIGURE 7.26** Comparing tightly and loosely coupled designs.

bound to the Web. They may be implemented across the Web or inside a secure enterprise environment. Google, Yahoo!, Salesforce.com, Amazon, MySpace, and many others offer Web services (or similar RESTful services) for public and private use. Some enterprise media facilities are using Web services for select applications. These topics and the concepts of SOA are discussed in more detail in Chapter 4.

Web services are not a panacea for all aspects of a media workflow. Real-time A/V operations often require dedicated devices to meet their QoS spec, and Web services fall short today. Still, when real time is not an issue, then workflows using Web services are practical and will see more light of day as our industry builds confidence in their practical value.

#### 7.6.4.2 Workflow Platform Examples

One popular platform with Web services interfacing is the Avid Interplay Platform. This workflow engine enables MAM, media capture, proxy creation, logging, transcoding, review, simple story layout, revision, collaboration, craft editing, and more. The built-in APIs enable third parties to control select features, import/export metadata, asset deletion, register new files, spawn file transfers, and access archives.

Another example of a workflow engine is Mediaflex from TransMedia Dynamics. TMD provides a reconfigurable, modular software product suite for managing the production, delivery, archiving, and repurpose of media content

in multiple formats. Its functionality can be reconfigured via a GUI to achieve agility.

The landscape is not barren. Flexible workflow solutions ranging from a “station-in-a-box” to complete news production systems are available. To learn more about this ocean of solutions, see Dalet’s DaletPlus, Thomson’s ContentShare 2, Cinegy’s Cinegy Workflow, Apple’s Final Cut Server, Pharos’s Mediator, and BitCentral’s Precis as a good representation of what is available.

#### 7.6.4.3 Workflow Takeaways

The definition of *workflow* as “a set of sequential steps needed to complete a job” is deceptively trivial. Yet, behind these few words lies a world of sophistication in terms of design, planning, documentation, implementation, and operational aspects. This chapter outlines the fundamental elements of any professional media workflow. By using these concepts and maxims as a guide or checklist, architects, designers, and engineers will have added confidence in the merits of their workflow solutions.

## 7.7 BROADCAST AUTOMATION

Automation techniques have a long history in TV station operations. Born from the pressure to replace human operators with automated processes, automation was initially used to frame accurately control (record, shuttle, play) videotape decks. From this humble beginning, automation can now control virtually all aspects of scheduled broadcast operations. Some TV stations, especially those without live events (news), can run with lights out for the most part, with automation software running technical operations.

Automation’s role in a typical broadcast facility is as follows:

- Control A/V devices and process frames accurately;
  - Video servers, tape decks, graphics/text engines, master control levels, signal routing, video transcoding, file transfer, secondary-event video compositing, recording, playout, satellite feed recording, news gathering, archiving, and more.
- Execute the provided schedule (from the traffic department) following a 24-hour timeline accurately timed to the SMPTE house clock. Every device operation may be controlled based on the common SMPTE clock. This assures all operations occur at precisely the correct frame time across all devices in a facility: queue file, play file, overlay scrolling text on video, set level, output to viewers. Log “as run” information to prove schedule execution. Incidentally, BXF plays a big role here. See Section 7.1.2.

Prompt, control, and monitor all human-assisted media prep operations.

- Ingest new materials, trim, handle QA.
- MAM support for registering new materials, metadata, searching, and indexing.

Broadcast automation is mature and supported by many vendors worldwide. Other types of automation are also popular, including news broadcast (with studio camera robotics control) with associated story production management. Virtually all aspects of automation's functional task list can be implemented using IT-based servers, networking, and storage. This is a *control plane* application for the most part (Section 7.1). Workflows of all types may be automated—partially or in full—if repeated, scheduled, or template operations are implemented. Expect to see automated operations gain more ground over the next few years as facility owners demand more ROI from their infrastructure investments.

## 7.8 IT'S A WRAP—A FEW FINAL WORDS

In this chapter and the preceding six, we have presented the fundamentals of media systems. Building or remodeling a media facility is a tall order, and you will likely require the skills of an expert systems integrator. When selecting an SI, look for knowledge and proven track record in IT and A/V systems. Use the insights, guidelines, and best practices outlined in these chapters to assist in making wise choices, especially during the design stage of a project.

## REFERENCES

- Abunu, D., et al., The BBC in the Digital Age: Defining a Corporation Wide MAM Approach, page 417, *Conference Publication of the IBC*, 2004.
- Bhaskaran, V., et al. (1997). *Image and Video Compression Standards* (2nd ed): Kluwer Press.
- Cordeiro, M., et al. (September 2004). The ASSET Architecture: Integrating Media Applications and Products through a Unified API. White Plains, NY. *SMPTE Motion Imaging Journal*.
- Gilmer, B. (Ed.). (2004). *File Interchange Handbook: For Professional Images, Audio and Metadata*: Focal Press.
- Hasegawa, F., & Hiki, H. (2004). *Content Production Technologies*. Hoboken, NJ: Wiley.
- Marpe, D., et al. (February 2004). Performance Evaluation of Motion-JPEG2000 in Comparison with H.264/AVC Operated in Pure Intra Coding Mode. *Proceedings of SPIE*, 5266, 129–137.
- McDermid, E. AAF Edit Protocol: Introduction and Overview, *SMPTE Motion Imaging Journal*, page 225, Issue: 2004 07/08 July/August
- Poynton, C. (2003). *Digital Video and HDTV—Algorithms and Interfaces*. San Francisco, CA: Morgan Kaufmann.
- AAF and MXF Tutorials. (2004) 07/08 July/August. *SMPTE Motion Imaging Journal*.
- Symes, P. (October 2003). *Digital Video Compression*. NYC, NY: McGraw Hill/TAB.
- Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (July 2003). Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*.

This page intentionally left blank

# Security for Networked A/V Systems

## CONTENTS

<b>8.0</b>	<b>Introduction and Scope</b>	<b>318</b>
<b>8.1</b>	<b>The Threat Matrix</b>	<b>320</b>
8.1.1	Viruses, Worms, Trojan Horses, and Malware	320
<b>8.2</b>	<b>Prevention Tactics</b>	<b>322</b>
8.2.1	Developing a Security Plan for System Elements	323
8.2.2	The Window of Vulnerability	325
<b>8.3</b>	<b>Prevention Technology</b>	<b>326</b>
8.3.1	The Main Firewall	326
8.3.2	Intrusion Prevention Systems	328
8.3.3	Intrusion Detection System	330
8.3.4	Antivirus and Client Shell Software	331
8.3.5	The Virtual Private Network	332
8.3.6	Securing the Media Enterprise	334
<b>8.4</b>	<b>Basics of Cryptography</b>	<b>335</b>
8.4.1	Modern Encryption Methods	336
8.4.2	Keys and Key Management	338
8.4.3	Kerberos	342
8.4.4	Digital Signatures (DS)	343
<b>8.5</b>	<b>It's a Wrap—Some Final Words</b>	<b>344</b>
	<b>References</b>	<b>344</b>

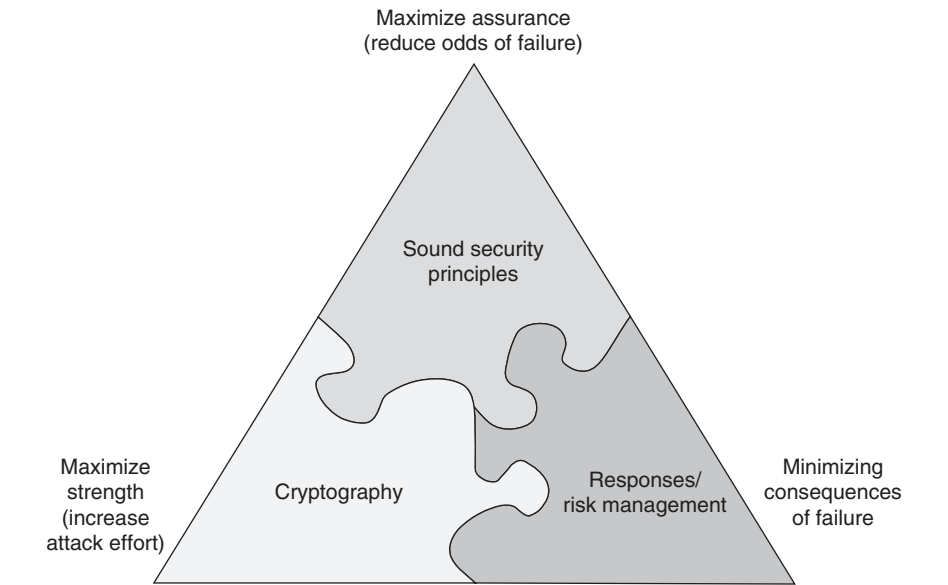


8.0 INTRODUCTION AND SCOPE

Information systems security has never been more critical.<sup>1</sup> At the same time, protection of information systems is increasingly complex. As A/V facilities reinvent their service infrastructure to meet business demands, traditional boundaries are disappearing. The cyber security threats lurking outside those traditional boundaries are real and well documented. Security by exclusion is both more necessary and more difficult, but at the same time not sufficient. The enterprise must also practice security by inclusion to allow access to the services that field offices, customers, suppliers, and business partners are demanding.

Information security is not only a technical issue, but also a business and governance challenge that involves risk management, reporting, and accountability. Effective security requires the active engagement of executive management to assess emerging threats and to provide strong cyber security leadership. Figure 8.1 provides a good look at the three main building blocks for a secure enterprise.

This chapter provides a summary of the basics of security in light of A/V systems. Our focus centers on the process of developing a sound security plan, what the threats are, and how to protect networked A/V systems against them.



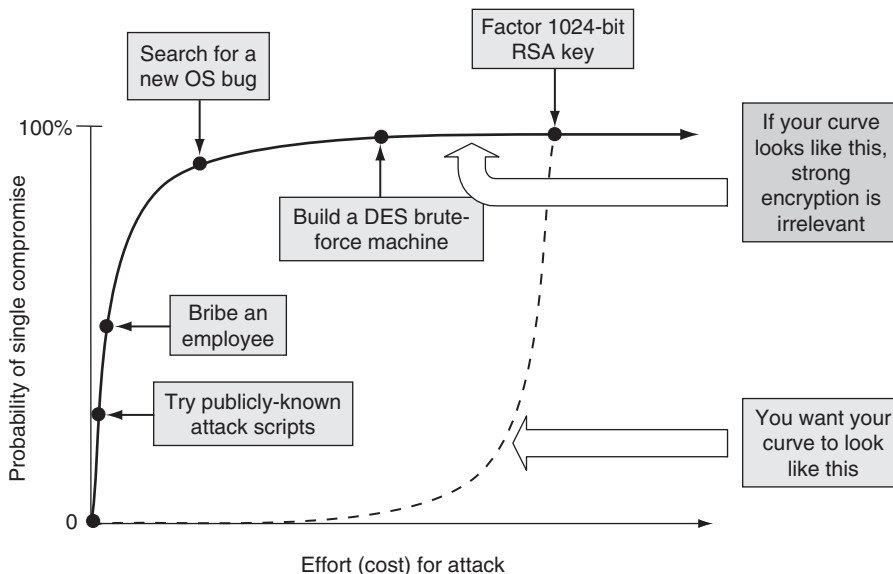
**FIGURE 8.1** *The building blocks of security.*  
*Concept: Cryprographic research, Inc.*

<sup>1</sup> This paragraph was paraphrased from (NAC).

Detailed examinations of all security aspects can be found in a number of books and Web resources. This chapter is organized around the following ideas:

- The threat matrix
- Intrusion prevention tactics and security planning
- Prevention technology
- Cryptography concepts

At one level, security is founded on the world's most sophisticated mathematics. At another, it is about bribery, loose lips, and poor processes. A decent percentage of security breaches occur from unhappy employees who have easy access to internal networks and systems. Figure 8.2 illustrates the two extremes of system security: plotting effort versus probability of attack success. The ideal curve is solely proportional to data encryption strength. In practice, the curve is composed of a mix of human, software, and process weaknesses. Most experts agree that encryption is rarely broken these days by hackers, although there are well-known examples—the breaking of the data scrambler for the DVD being a prime example. Rather, systems are compromised by other means. Let us look into what the threats are and then counter them with a summary of prevention and detection methods. An overview of cryptography is presented later in this chapter. The RSA key in Figure 8.2 is also explained later, giving you a chance to win \$200,000.



**FIGURE 8.2** Security should be proportional to the effort it takes to crack it.  
Concept: Cryptographic research, Inc.

## 8.1 THE THREAT MATRIX

Ideally, A/V equipment in mission-critical applications is 100 percent protected against all threats. Before examining how to prevent threats, let us look at the threat landscape. In general, a networked computer system threat has a life cycle. The following five-step sequence is typical of the life cycle:

1. **Probe the system.** This step may take place via port probes or studying for vulnerable access points. Outsiders often probe every known portal for access.
2. **Gain access.** This step allows a malicious entity (program, person, or machine) to gain access to a system or system element using the access point identified in step 1. If the access allows foreign code to execute, a virus, worm program, or code fragment is now free to execute. If unauthorized access is made (e.g., password compromise), then foreign or malicious internal users have complete access to all element resources.
3. **Execute malicious code or operation.** The execution of foreign code may be completely benign ("Hello!" message) at the low end of trouble up to deleting every file or stealing highly confidential information at the top end.
4. **Propagate.** Once a program executes, it can then propagate itself to other machines via network services, email, or other means.
5. **Remove the culprit.** Once the offender is identified, it can be removed from the compromised device, assuming it has not already destroyed itself in a Samson-like death. It is common for virus checkers to scan a disc and remove any harmful code, for example. If the threat was weak access security, then stronger entrance passwords are warranted possibly.

The life cycle may be cut short at any step if preventative measures are effective. For example, a port probe may be detected and the foreign data denied entry. Or, an antivirus program identifies the offender and removes it before execution.

Another threat is a denial of service (DOS). The goal is not to gain unauthorized access or run foreign code, but to deny legitimate users access to a service. Attackers may flood a network with large volumes of data or deliberately consume resources. Typical of such DOS attacks is to flood a TCP port (say, for the FTP service) with requests for a connection. One example is the well-known SYN flood associated with the initial steps in setting up a TCP connection. DOS attacks may consume a large share of main CPU clock cycles, thus preventing rightful use.

### 8.1.1 Viruses, Worms, Trojan Horses, and Malware

Four of the biggest offenders are viruses, worms, Trojan horses, and malware. These four sometimes-confusing terms are defined next.

#### 8.1.1.1 *Virus*

A *virus* is a program designed to infect a computer and spread via innocent human assistance. It often announces its presence to the user and may do damage to the target system's files or steal secrets. The virus attempts to spread via email attachments typically. In most cases a human unknowingly runs the virus program, thinking it is a friendly program or file. Always beware of executing unknown source files with .exe extensions. Plus, use caution with Microsoft Office macros, ActiveX, and some Java scripts.

#### 8.1.1.2 *Worm*

A *worm* enters a computer via a network connection and executes a downloaded program fragment—usually in data memory—without any human action. A worm is often described as a subclass of a virus, although the two are very different in how they infect a system. Worms can spread much faster than viruses by orders of magnitude. For example, the Code Red worm infected 359K networked machines in 12 hr. Simulations of worm spreading show potential rates of 7.5 to 30K infected machines per second! To make matters worse, the spreading is exponential in growth. A typical entry point is via a TCP/IP service port. It is common for Microsoft to publish 40–50 security vulnerabilities per year for its released OS.

#### 8.1.1.3 *Trojan Horse*

A *Trojan horse* is a program that initially looks benign but allows for backdoor, undesired actions. One example is free downloadable programs claiming to do some desired action (a drawing program) but also performing other undesired actions, such as installing pop-up advertising software. Beware of free programs!

#### 8.1.1.4 *Malware*

Malicious software, or *malware*, is designed to hijack some or all of the user experience and turn innocent grandmothers into porn providers. It includes viruses, worms, Trojan horses, spyware, and some pop-ups. Spyware is software that tracks usage and reports it to others, such as advertisers. Usually, the tracking is concealed from the user. A pop-up is a new browser window that usually appears unrequested on the screen, brandishing ads. Particularly maddening are those termed *exit pop-ups*—browser windows that launch when you leave a site or close a browser window. Within a scripting language, these are called “onUnload” and “onClose” events. Undesired pop-ups may turn the browsing experience into a sleazy carnival midway, complete with flashing lights and loud music.

Because of the prevalence of Microsoft OS and derivatives, many hackers target these systems for *worm* access. Other operating systems are not usually targets. Although Linux and the MAC OS have vulnerabilities, these are exploited infrequently. Several organizations document virus and worm threats.

The CERT Coordination Center ([www.cert.org](http://www.cert.org)) and SANS ([www.sans.org](http://www.sans.org)) are excellent resources for the early notification of security threats. Many IT managers subscribe to their services for early notification. If the threat is a virus, then antivirus vendors strive to find a prevention method ASAP. If the threat is a worm, then the OS provider usually offers a software patch.

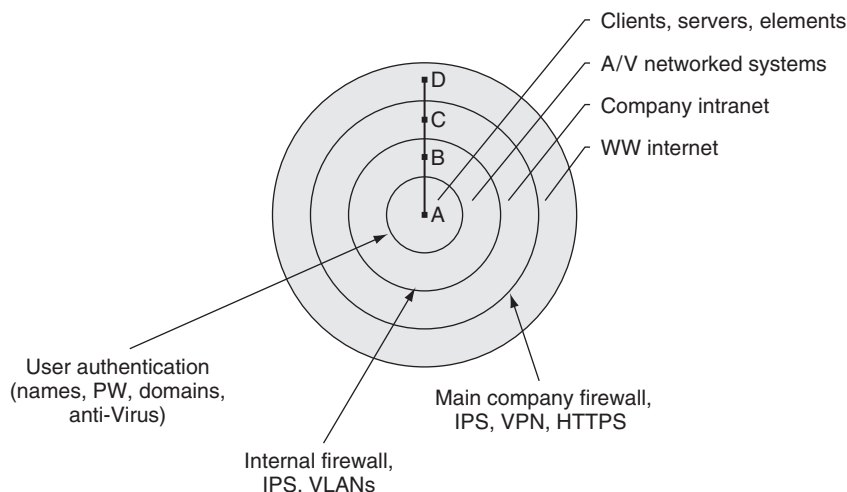
Knowing the types of threats is important, but stopping them cold is more important. The next section considers prevention tactics.

## 8.2 PREVENTION TACTICS

Many, if not most, computer security vulnerabilities can be eliminated if the systems are properly configured against attacks. Consistency is a key factor in security. A plan is needed for all devices that trade off usability against security. It is not difficult to wall off a computer to the extent it is practically unusable—no email access, no Web access, no internal network access, and so on. However, full and unfettered access to any and all network resources breeds trouble. Finding the balance between the two extremes is a science.

Developing a security process for an organization is the most important preventative measure. Passwords, antivirus software, firewalls, and so on are of value, but a cohesive prevention deployment process applied consistently across the organization is the most important aspect of a comprehensive security solution.

Let us put the regions of attack in perspective. Figure 8.3 shows typical boundaries between networks and systems for a large deployment. Four regions are identified, A–D, each with its own security needs. Region A is a typical element—server, desktop PC, and so on. B is a walled-off A/V system



**FIGURE 8.3** Security boundaries in large systems.

requiring very high security. This could be the news production system of a TV station, for example. Region C is the company intranet. Region D is the unbounded worldwide Internet. In most cases, either region A or B is the target for access. Regions C and D are the starting points of the attack. D is the most common starting location. However, internal attacks (C attacking A) can occur maliciously but at times out of ignorance too. For example, a virus may enter a building on a laptop or USB memory stick without the carrier even knowing it.

There are three security boundaries for this model. Figure 8.3 outlines typical methods to secure a boundary. For example, A, B, and C are protected from D by the use of firewalls, intrusion prevention systems, and VPNs. Each prevention method is explored in the discussion to follow.

### **8.2.1 Developing a Security Plan for System Elements**

This section outlines the questions to resolve as part of a security plan for system elements. The answer to each question should be incorporated into an overall security implementation plan. For each case ask what the implications are for RT A/V gear. Specific prevention technology is covered later. The following QA list was loosely paraphrased from information available on the CERT Web site.

1. Identify the purpose of each system element.
  - a. What is the nature of this element? What categories of information reside on it? Is it crucial to RT A/V production?
2. Determine network services provided by this element.
  - a. Email, Web access, Web services, file transfers, media converter, etc.
  - b. For each service, document whether the device will be configured as a client, server, or both.
  - c. Disable all other services per device. The list of active services should be well documented and no new ones added without permission. It is well known that rogue and poorly designed network services are a source of easy access into a device or network. Also, for A/V clients, unnecessary network activity can disrupt (cause glitches or worse) the RT nature of a client.
3. Identify the network service software to be loaded on this element.
  - a. Are the services bundled with the OS and/or are third-party services to be loaded?
  - b. Pay special attention to the security aspects of any third-party applications.
4. Identify the users or categories of users.
  - a. Roles of the users.
  - b. Actions performed, services needed.

5. Determine the file privileges for each category of user.
  - a. What file actions are allowed: read, write, delete files, directory access, and machine access.
6. Determine how users will be authenticated.
  - a. User names, passwords (and stale password change methods), secure ID cards, or other. Many networks rely on Microsoft's Active Directory (AD) for systemwide user/password registration. AD allows an administrator to enter a user name and password once, and this registry is consulted by all user applications across a range of products. AD enables "single logon" for users. AD also supports password aging and prevents breakable passwords from being used.
7. Determine access to information control.
  - a. Is all information available to all users? Is data encryption needed to protect sensitive information?
8. Develop intrusion detection strategies for select elements.
  - a. Decide what information is to be collected for login logs and audit methods.
9. Develop backup and recovery for devices that require it.
  - a. Document a method to restore a defective element from scratch.
  - b. Set up a backup method for devices that require it. This step is especially crucial for mission-critical devices.
10. Develop a documented plan for installing the OS of a device.
  - a. Include what security aspects should be turned on.
11. Determine how any system elements will be connected to the network.
  - a. Always, sometimes, or never connected. Determine if a device needs to reside in a DMZ (extra secure area of company intranet) or external hosting site for maximum security.
  - b. Define a clear policy when attaching foreign laptops to the internal network.
  - c. If remote computers need to use the internal network, define a VPN strategy.
12. Identify an antivirus strategy.
  - a. Do not install any software that may interfere with the RT nature of A/V system elements.
  - b. Work with your A/V vendor to find a way to provide for antivirus software. This is tricky, as every vendor's products have different requirements regarding when, at what level, and for how long a virus scan may operate.
13. Establish a plan to patch the OS as security alerts are announced.
  - a. This step is crucial, so make sure your A/V vendor has a good plan in this area.
  - b. Assure that your A/V vendor tests for the most crucial OS patches on mission-critical devices. Because vendor patch verification will always lag the availability of a patch, the latency needs to be defined.

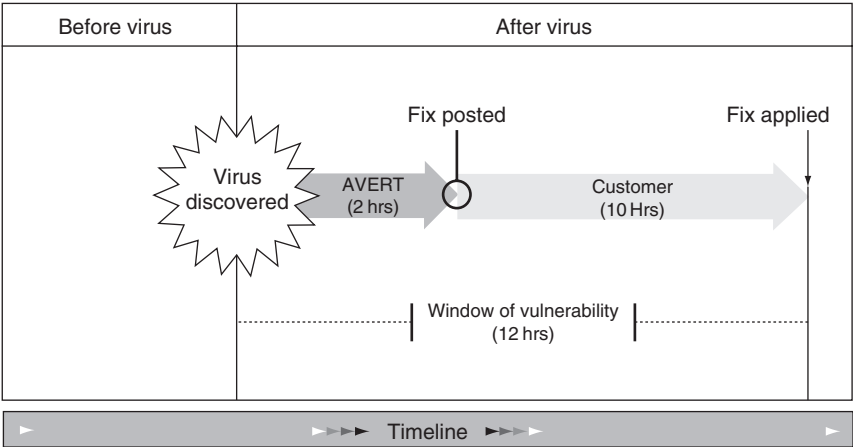
- 14. Keep the security plan current and promote awareness within the company.
  - a. Monitor and evaluate policy and control effectiveness.

Of course, there are other items needed to define the full constellation of a security solution. However, the 14 items just listed are the most common for any system. The deeper the device is located away from the Internet/email (Figure 8.3), the less likely it is to be attacked. Still, internal attacks are possible although unlikely. A security policy may not allow any Internet access on some elements. Some vendors rely on Internet access to diagnose and repair their equipment under a service contract. More on this topic in Chapter 9.

8.2.2 The Window of Vulnerability

Worms and viruses are quick acting. An IT manager may have only seconds to secure a device(s) against a new network-disseminated worm threat. Because an organization can react immediately, a firewall or software patch is relied on to prevent worm infections. More on firewalls in the next section. Viruses are slower acting than worms due to the need for human action to spread. It is common to get an email warning message from IT announcing something like: “Do not open any attachment named *I Love You*.” For viruses there is a window of vulnerability, as illustrated in Figure 8.4. Figure 8.4 also applies to worms, but the time period is much smaller.

The window shows four periods: one before a virus or a “worm hole” is detected and three after. The window of vulnerability is the time when devices are open to possible attack. The *avert* period is the time when IT sends messages to all users not to open named attachments for viruses or to patch their computers against worms. This period is sometimes called the *zero-day attack time*. The implication is that attacks can start on the heels of (or before) the



**FIGURE 8.4** The window of vulnerability.  
Concept: McAfee.



public announcement of a vulnerability and before a protection method is available.

Once an antivirus vaccine is produced, then all vulnerable systems should be updated with it. In most corporate environments, antivirus updates occur automatically and invisibly using antivirus management applications. The longer this takes (the *customer* period), the longer the period of vulnerability. Within hours of detection, an antivirus vaccine is normally available. Again, virus scanners must be installed on RT A/V equipment in cooperation with the equipment provider to guarantee RT performance when scans are active or scheduled for off-hours.

For RT gear it is especially important to have confidence that any software patch will not affect performance. Should IT install the patch, hurriedly test, and deploy, hoping that the patch works, or should IT wait until more is known about its ramifications? These are difficult decisions and keep the window of vulnerability open. For worms, OS providers often notify the community of the vulnerability and then provide the patch. The delay before the patch is installed gives attackers time to exploit defenseless system elements.

## 8.3 PREVENTION TECHNOLOGY

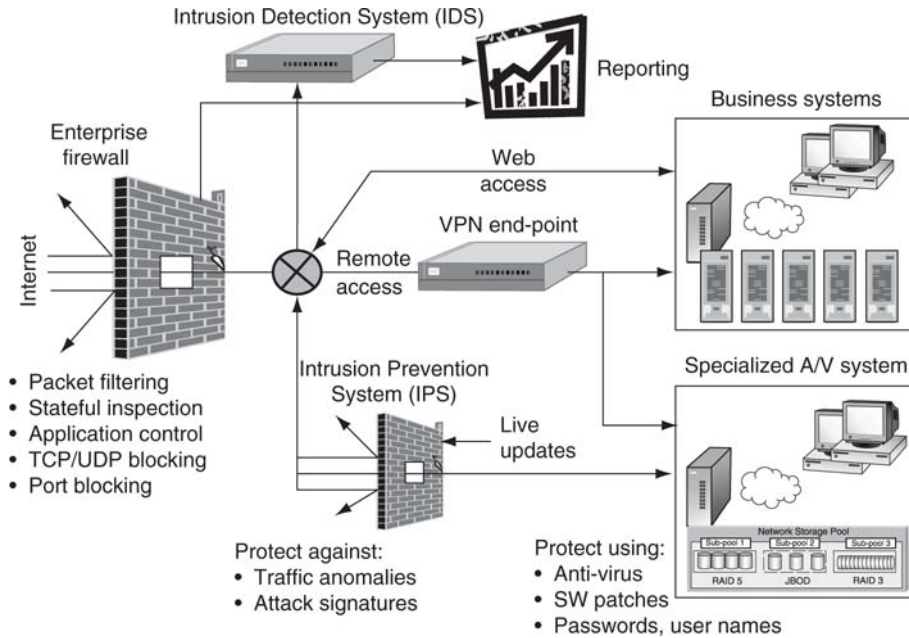
In the end, it is the prevention technology that will keep out the pirates. In general, there are five main means to prevent/discover attacks over a network:

1. Main firewall
2. Intrusion Prevention System (IPS)
3. Intrusion Detection System (IDS)
4. Antivirus methods
5. Virtual Private Network (VPN)

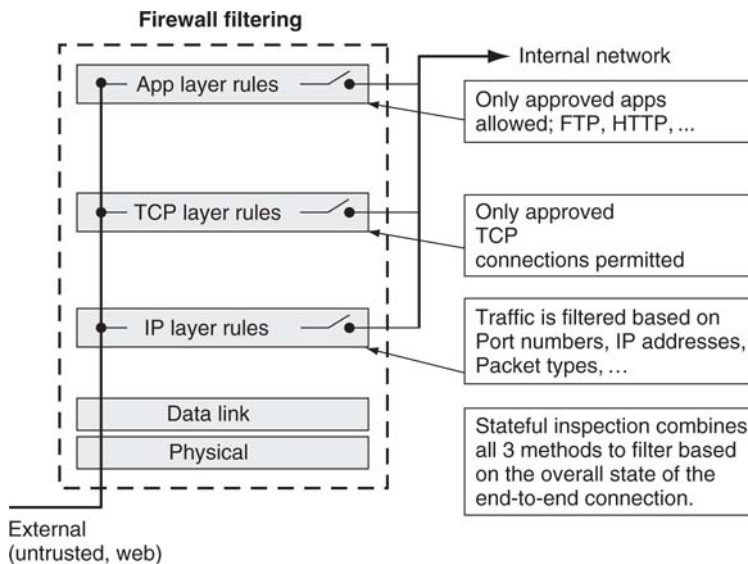
Figure 8.5 shows the overall landscape to prevent outside attacks against internal, private networks. Of course, there are other configurations, but let us use this one for the purposes of discussion.

### 8.3.1 The Main Firewall

The main firewall is the classic method used to protect against outside intrusion. Firewalls come in many flavors from a variety of companies. Some are host based and run on desktops, servers, and so on. Microsoft's Vista *Internet connection firewall* is a prime example. Others are network based, such as FireWall-1 from Check Point Software and Cisco's PIX Firewall family. Figure 8.6 illustrates the four main functions of a generic firewall. Some firewalls have added loads of other functions, such as NAT translation, VPN, and so on, but these are auxiliary to the main purpose behind a firewall, with the possible exception of VPN.



**FIGURE 8.5** Strategies for protecting business and A/V systems.



**FIGURE 8.6** Firewall filtering methods.

A firewall blocks malicious packets by using one or more of the following strategies:

- Packet inspection, filtering ports, packet types, IP addresses, etc.
- Transport layer filtering, TCP/UDP blocking, select connections permitted.
- Application layer logic, approved apps such as FTP, HTTP, and so on.
- Stateful inspection technology (also known as dynamic packet filtering), which tracks connection “state information.” It uses this intelligence to decide when to allow/disallow communication from remote computers.

Of course, the firewall should be configured by a competent IT security expert for the most effective blocking. There are several subtle settings that only a skilled expert would be aware of.

If A/V applications move data via a firewall, then attention must be given to bandwidth and other QoS needs for remote access. Does the remote user expect to transfer files at high rates? Will the transfer application be FTP? If so, it may be more practical to locate the FTP server in a third-party service center completely isolated from the enterprise network. Using off-site file transfer services allows for scalability, very high transfer rates, and complete isolation from business systems. For example, see [www.yousendit.com](http://www.yousendit.com) for a free basic service.

Also, it is not uncommon to have remote user access A/V files for proxy viewing and editing. It is problematic to allow direct access even via a firewall. In this case a secure VPN connection should be forged to guarantee secure access.

### 8.3.2 Intrusion Prevention Systems

The IPS has a higher level of blocking than a firewall. There is much industry debate about the need for this in addition to a firewall. Most of the IPS vendors do not attempt to duplicate all firewall functionality. However, some functions are indeed duplicated, along with many functions not found in a firewall. If one is installed, a firewall must also be used, as shown in Figure 8.5. The handwriting on the wall seems to indicate that eventually the IPS will likely assume firewall functions (or the firewall will assume IPS functions) as the product category matures. The IPS protects the A/V system but not the business system in Figure 8.5; however, it could also protect the entire internal network.

Higher levels of IPS security include the following:

- **Performance.** IPS reliably supports gigabit speeds and low latency. Ideally, it appears as a “bump on the wire” as installed. Some units sport throughput delays of  $<250\mu\text{sec}$ . The unit should be totally nonintrusive. It should filter only actual threats with no false positives (stopped a non-threat) or false negatives (did not stop a valid threat). Clustered IPS units

exceed 8 Gbps of throughput. The device needs to precisely discriminate between benign and attack traffic. It may also filter traffic in both directions.

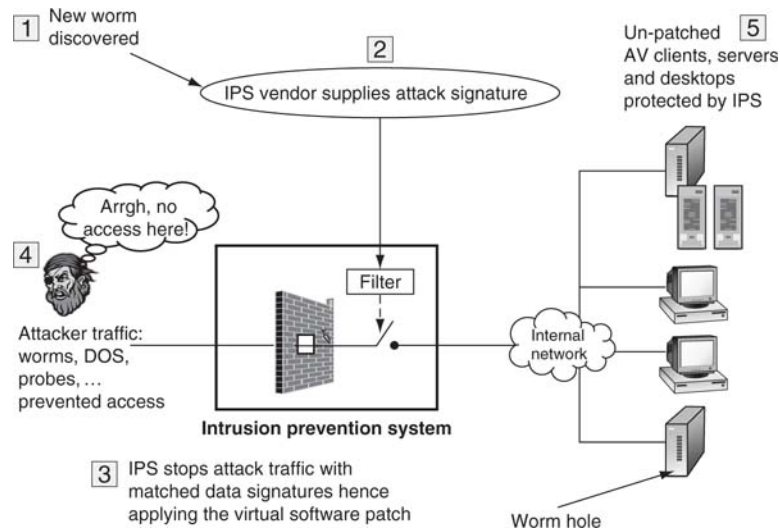
- **Protection.** Protocol analysis, anomalous behavior, identify and filter signatures of known methods of attack, statistical analysis, fragmentation attacks, usage patterns, port usage, more.
- **Near real-time updates of the IPS filter engine.** The vendor should supply updates to keep the protection engine “smart.” For example, when a new worm vulnerability is identified, the vendor updates the IPS to stop the worm from passing into the network.

The idea that an IPS appears as a bump on the wire is interesting. Most IPS units do not have a MAC or IP address and are simply inserted into the desired key path (Ethernet link) without any configuration. Their ability to analyze data streams for attack signatures and to stop any suspect data is a powerful feature. For example, some external malevolent probes look for device TCP port buffer overflows (and then take command of a CPU), but an IPS can detect this kind of activity and abort its operation.

The power to filter streams enables the *virtual software patch*. What is this? In most cases, the discovery of a worm vulnerability is accompanied by the release of a software patch for a program or OS. Ideally, an enterprise IT department will immediately install the patch on every target server and desktop. In reality, there may be a long delay between the availability of a patch and its installation on all vulnerable devices. It takes time to qualify a patch and then time to distribute the patch. Then, too, for real-time A/V equipment (on-air video server) the patch should be qualified by the A/V vendor *first* before the end user installs it. All this amounts to the window of vulnerability being stretched into days or even months in some cases.

Enter the virtual software patch. As soon as a vulnerability is announced, the IPS vendor can update its attack signature database to include this new threat. The vendor then downloads this new attack filter into every IPS under contract for real-time updates. In a matter of hours after a worm vulnerability is discovered, an entire network of computers can be protected against the new danger. This is a powerful feature and protects devices before they have installed the latest software patch for a particular exploit. Figure 8.7 illustrates this idea. No A/V vendor testing, no enterprise testing, and virtually immediate protection. That is the power of the virtual patch. Should the enterprise still install the recommended patch? Likely, but now time is not as critical, and a methodical plan to perform updates is in order. No rushing, no errors.

As a point of illustration, the UnityOne IPS from TippingPoint Technologies is one such product that provides for virtual software patches (see [www.tippingpoint.com](http://www.tippingpoint.com) for more information on these ideas). The IPS is a valuable



**FIGURE 8.7** *The IPS and the virtual software patch.*

product category and plays an important role in protecting mission-critical A/V gear.<sup>2</sup> Especially important is that device protection does not require the A/V vendor to qualify the suggested software patch immediately.

The IPS can become a lightning rod of blame when there are network problems. True, on occasion it can exhibit false positives if not tuned correctly. Albert Einstein once said, “Not everything that is counted counts, and not everything that counts can be counted.” Due to the complex nature of blocking threats, some false positives and false negatives will occur, so some enterprises are slow to install an IPS because it *may* block legitimate traffic. Then, too, if all traffic crosses through the IPS, it is a single point of failure, so a failover means is needed. Looking back 10 years, the newly introduced firewall was a punching bag when access problems occurred. Today the firewall is a necessary, non-controversial system element. The IPS is maturing, and over time it will likely reach a point of acceptance where it becomes *de rigueur* as a system element.

### 8.3.3 Intrusion Detection System

The IDS inspects data traffic and looks for problems. When irregularities occur, they are logged with optional notification to management. The IDS does not block traffic but indicates that traffic may be harmful. It is a fire alarm, not a fire extinguisher. The IDS signals attacks explicitly addressed by other security components (such as firewalls and even IPS) and also attempts to provide notification of new attacks unforeseen by other components. Intrusion detection

<sup>2</sup> Of value for insight is the TippingPoint white paper, “The Science of Vulnerability Filters,” by Victoria Irwin.

systems also provide forensic information that potentially allows organizations to discover the origins of an attack.

As the IPS matures, it will replace the IDS for many applications. In theory, a full-featured IPS is able to detect, report, *and* block threats, so investing in both technologies may be a waste of resources. Some consultants see the IDS as an essential component for sniffing at select parts of the network so it may well survive for some time. For example, when a virus or other threat is carried into the enterprise via an USB memory stick, the IDS should detect the traffic and send an alarm. Keep in mind that the IPS may not catch a threat that is released inside an organization. The demise of the IDS is a controversial conclusion, so time will be the arbiter.

In the open source world, Snort has become the de facto IDS standard (see [www.snort.org](http://www.snort.org) for information and downloads). This Web site has lots of information about threats and how to identify them. See, too, (Beale) for the complete bible on Snort.

Intrusion detection devices are not tuned for A/V gear or applications, so there is nothing special to expect from them in this regard. They are not in series with mission-critical data flows, so their failure will not immediately affect business operations. Looking forward, it seems that the firewall and full-featured IPS are the future of enterprise filtering and intrusion detection.

## NX NAILS WORMS

Worms commonly use data memory to execute their rogue program fragments. NX comes to the rescue—No eXecute—by blocking program execution from data areas. Before NX, CPU memory did not distinguish between permission to read and permission to execute instructions. NX

changes that by marking memory as executable or not. So, a worm may indeed enter a memory area, but NX stops it from running. AMD, Intel, others, and some operating systems now provide NX functionality.



### 8.3.4 Antivirus and Client Shell Software

Despite the power of the IPS, many will not block malicious virus attachments. As a result, it is good practice to keep up-to-date virus scanners on every desktop and server. Scanners are mature, and updates can occur in real time once an antivirus is produced. However, using antivirus programs on real-time A/V gear is problematic. Because most of these programs are not respectful of real-time devices and steal a good portion of CPU power to do a scan, it is prudent to schedule the scan for off-hours if possible. If the device is in service 24/7, then the A/V vendor needs to supply a solution that works in harmony with your needs. It is possible to place the scan at a low priority so as to only minimally steal CPU resources with no adverse affects on A/V performance.

The enterprise perimeter has become fluid as it passes through teleworkers, mobile employees, and contractors. Security policies should be enforced among

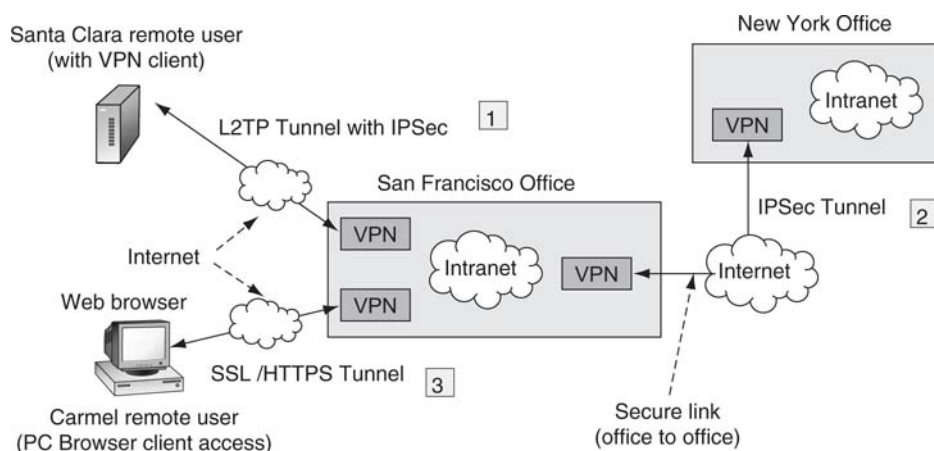
a decentralized user base as effectively as they are on the corporate network. Employees and contractors, whether deliberately or inadvertently, are in a position to seriously compromise existing security practices. The fast-changing pace of technology, such as the widening adoption of VPN and the rise of the “zero-day exploit” requires new methods to stop threats from spreading. One means to add security is via a client shell. This is an example of a host-based IPS.

Cisco’s Security Agent, NetIntelligence’s End Point, and other products are software shells that monitor all I/O network activity and user actions on end point devices such as a PC. Some shells can be configured to only allow for selected user operations. When a firewall, antivirus, filters, application restriction, and usage monitoring of all sorts are combined, these programs offer a high level of localized protection. However, they may even be more invasive than a vanilla virus scanner in terms of affecting A/V real-time operations, so do not install one of these on critical A/V gear unless the supplying A/V vendor guarantees performance.

### 8.3.5 The Virtual Private Network

The heart and soul of secure remote access are bound up in the VPN. The basic idea is to encrypt the TCP/IP data packets from external users to hide them from prying eyes as they traverse the Internet. Also, the logon process for these users is secure, using time-stamped passwords and other means to make unauthorized access almost impossible. The VPN is the vehicle to secure a tunnel from the external user to the internal network. For all practical purposes, remote users are an extension of a private network.

A VPN creates a secure “tunnel” through the public network, so the protocols used to establish the connection are called tunneling protocols. Figure 8.8 shows the three most common ways to use tunnels for modern VPN



**FIGURE 8.8** Using tunneling protocols to create secure VPNs.

implementations: L2TP with IPSec, IPSec alone, and SSL/HTTPS. The acronym soup will be explained next.

L2TP is the Layer Two Tunneling Protocol (RFC 2661) and can run over IP/UDP and others and carry various protocols in its tunnel. It also supports authentication means for user access. IP Security (IPSec) is a method used to secure IP packets. All packet data, except for the IP address header, are encrypted for passage over unsecured networks. As a result, it is obvious that the TCP and UDP payloads are also encrypted. IPSec is a suite of protocols (see RFC 2411 for a good summary). IPSec uses the Internet Key Exchange method (IKE, RFC 2409) to obtain the encryption keys per connection. The IKE uses Diffie-Hellman (DH) Public Key exchange. DH is described later.

The Secure Sockets Layer (SSL) is used most often to secure HTTP, and the combination is called HPPTS (as seen with <https://>). Virtually all common Internet browsers support SSL natively, which is a big convenience for “any client” (kiosks, wireless hotspots, Internet café, etc.) secure remote access. While SSL can add security to any protocol that uses TCP, it occurs most commonly with the HTTPS access method. HTTPS serves to secure Web page access. SSL uses public key cryptography and public key certificates to verify the identity of end points. The term *Transport Layer Security (TLS)* has somewhat replaced SSL, but both are used to describe essentially the same functionality.

### **8.3.5.1 Three Secure VPN Methods**

What are the differences and trade-offs among the three VPN methods as illustrated in Figure 8.8? The L2TP/IPSec method carries user data IP/UDP/TCP payloads by L2TP, which in turn are carried by IPSec. This is a common method for remote users who need complete secure access to a corporate network with strong authentication. Most often, specialized VPN software must be loaded on any remote machine to support this form of VPN.

For office-to-office connections, it is possible to only use IPSec, as there is no user authentication. The connections are permanent, and the tunnel provides transparent access for any IP payload.

The third method with only SSL usually supports HTTPS. This allows remote users to access HTTPS Web mail as supported by Microsoft's Exchange Server, for example. Importantly, even though SSL is secure, it is also application based. Remote users do not have access to the entire internal network, only to the device and application (e.g., Web server) that terminates the SSL connection. As a result, SSL adds security by tying a remote user to an application and not the entire internal network.

Some SSL-based VPN solutions permit access to more than Web applications using a proxy mapping method. However, each application must be “Webified” either natively or via the proxy mapper so that it may be accessed via a remote browser. SSL/VPN falls short of offering 100 percent complete internal



network access, but for most remote users this is preferred. There is currently a huge marketing battle between IPSec VPN vendors and SSL-based ones.

For A/V applications, L2TP/IPSec is a likely choice for remote proxy browsing, A/V editing, and file transfer. The remote users may have access to all the resources within the A/V system, but the IPS (remember, it is bidirectional in Figure 8.5) can limit access into the corporate network as needed. SSL/VPN providers claim equivalent performance to L2TP/IPSec, but each vendor will offer some slightly different model of operations, as the SSL/VPN is still browser based. Because SSL is TCP specific, UDP streaming applications are not supported, whereas with IPSec they are. With L2TP/IPSec, applications are browser independent.

Strictly speaking, VPN is a tunnel between end points over a network. Most references refer to VPN as secure (as our examples do), but some authors refer to VPN as a tunnel—secure or not. For example, Generic Routing Encapsulation (RFC 2784), IP-in-IP, and MPLS are tunneling protocols but are not encrypted. Often they are combined with an encryption method (such as IPSec) to create a secure VPN.

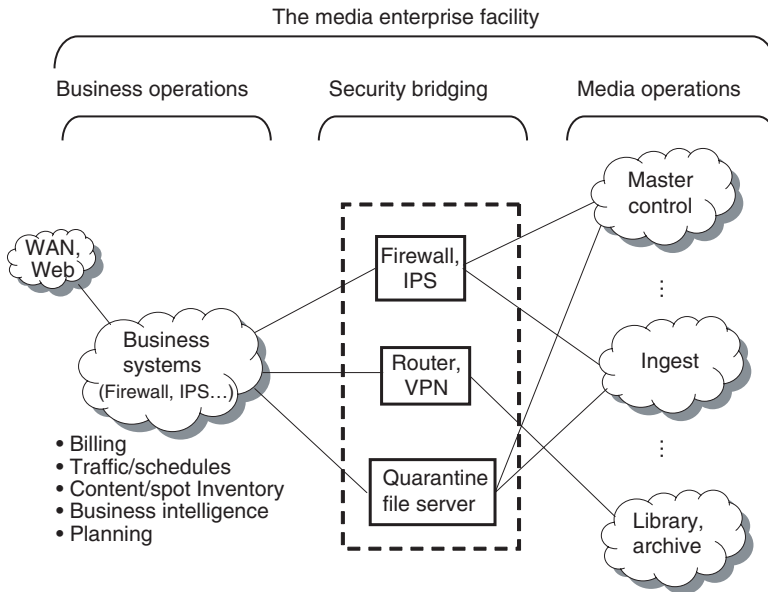
### 8.3.6 Securing the Media Enterprise

In any large media enterprise, there are two domains: business operations (finance, marketing, HR, traffic, planning, etc.) and media operations (ingest, master control, archive, etc.). For the most part, the media operations are considered mission critical and must be protected from any security threats that exist on the business side. This is not to say that business operations do not require security measures. However, the media operations must have the absolute highest level of security from unintentional actions or rogues originating from the business side. Imagine someone across the Internet gaining control of a TV station—shudders!

Figure 8.9 shows an example of how to protect the media operations domain. The basic idea is to insert security bridging between the two domains. The level of access into media operations depends on needs. Nothing travels between the two domains without strict security, checking, and permissions. Typically, access is granted to specific IP addresses or specific applications. Remote access may be permitted via VPN to enable vendor monitoring and maintenance of media equipment.

Note that the business operations domain has its own security mechanisms of firewalls, IPS, and so on (not shown in Figure 8.9). The security bridging area is *in addition* to the already existing security measures for normal business protection.

One method that several large U.S. broadcasters use is the quarantine file server. Any file destined for the media side is first stored on the quarantine server and checked for viruses and so on. Once the file is known to be clean, it is manually or automatically transferred to the end destination. Not all media domain PCs or servers may have access to the business side and associated



**FIGURE 8.9** Secure bridging methods of operational areas.

WAN (or Web) connectivity. If this seems draconian, then remember Andy Grove's comment—"Only the paranoid survive."

Next, a cornerstone of security is covered—cryptography. As mentioned earlier, total security is bound up in process and technology as per Figure 8.2, so hiding data behind mathematics is only a part of an overall way to secure the enterprise. This fascinating aspect of security is considered next.

## PROTECTING THE CROWN JEWELS

Stored data accessed via SAN and NAS have traditionally been protected with simple user name/password schemes. Enter a new class of storage security that combines secure access, authentication, and strong data encryption to

provide a new level of data safekeeping. One vendor supplying these services is Network Appliance. See [www.netapp.com](http://www.netapp.com) and search for "storage-security" to learn more.



## 8.4 BASICS OF CRYPTOGRAPHY

*Quiet, I've got a secret to tell you.* These words are whispered millions of times a day in a variety of networked transaction scenarios. Keeping secrets is one of the most important activities in an IT environment. No discussion of security can be complete without at least a cursory look at the basics of cryptography—the basics of keeping secrets. This section introduces four fundamental

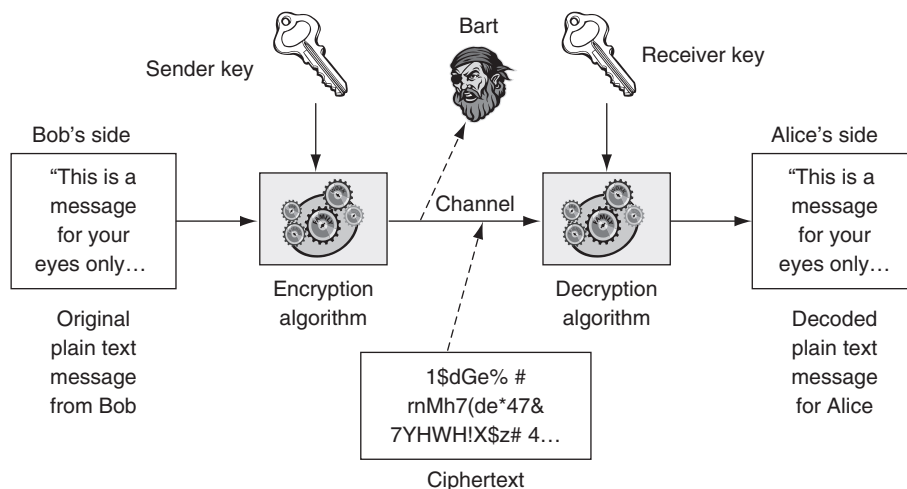
elements of secure transactions: **encryption**, **keys**, **key management**, and **digital signatures**. With the hacker's appetite to break into every nook and cranny of networked systems, cryptographic methods are applied to transactions of all kinds.

The history of secure transmissions is as old as mankind. One of the most famous persons of antiquity to employ coding of messages to foil eavesdroppers was Julius Caesar. In *The Gallic Wars*, Caesar describes using a substitution cipher to deliver a military message to Cicero, whose troops needed encouragement during a particularly difficult campaign. The author of the message substituted Greek letters for Roman letters, rendering the message inexplicable to the enemy. Caesar used three of our four themes: (1) encryption—the substitution cipher method; (2) a key—the “trick” to the cipher/decipher, swap Roman letters for Greek; and (3) key management—Cicero and only Cicero should know how to decode the message. Over the course of human development, thousands of ciphers have been developed to secure messages. So let us jump ahead two millennia and peek into the state of the art of these four themes.

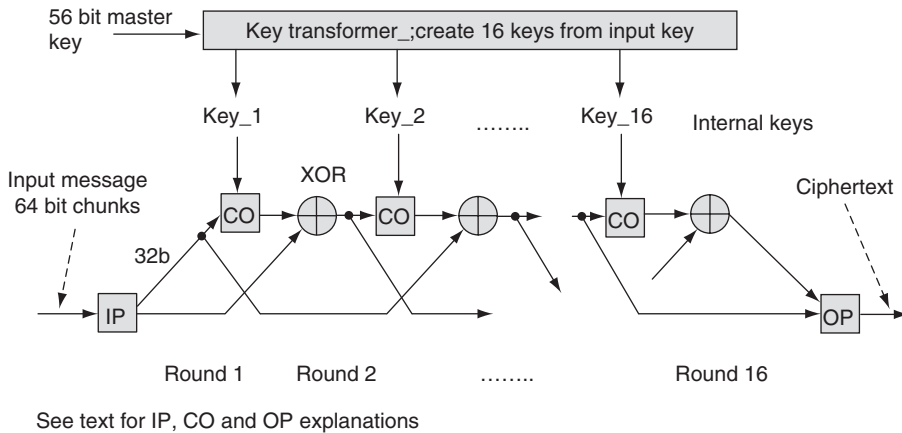
### 8.4.1 Modern Encryption Methods

There is an ocean of methods to encrypt/decrypt messages. This dialogue is limited to several of the most popular methods: DES, Triple DES, and IDEA. Figure 8.10 shows the simplest picture of a cryptosystem. Let us concentrate on the encryption/decryption methods—the keys used to uniquely code and decode the messages. For many systems, the two keys are identical. Our story has three players: sender Bob, receiver Alice, and eavesdropper Bart.

The Data Encryption Standard (DES) was developed in the 1970s by IBM for the U.S. government. This was the first “data mangler” to find wide acceptance;



**FIGURE 8.10** A basic crypto system.



**FIGURE 8.11** *Mangling numbers: Highly simplified DES design.*

it was free of licensing fees, simple to implement in hardware, well documented, and fast. It is a symmetrical design; i.e., the same algorithm is used to decrypt as is to encrypt. Considering the amount of data morphing that DES does, this is an amazing achievement. The basic idea is shown in Figure 8.11. A message (parsed into 64-bit chunks or blocks) enters at the left side, is permuted by the initial permutation (IP, a remapping of the input bits), and then is mangled 16 times by a combination of the all important cycle operator (CO), followed by an XOR function. At the final stage, the output permutation (OP) exactly undoes the IP operation. The CO function is a relatively involved combination of a bit permutation followed by an XOR (with an internal key as its second input) followed by yet another substitution/permutation of bits. Each cycle makes the input message more and more unrecognizable. After so much data morphing, it is nearly impossible to examine the output and thereby determine its input without knowing the key.

The internal keys Key\_1 to Key\_16 are derived from the 56-bit master key. Each internal key is a different circular bit shift and permuted version of the master key. Also, the decoder is amazingly the same device with the same 56-bit master key that the encoder used but with the 16 internal keys reversed; i.e., Key\_1 (decode) is Key\_16 (encode) and so on. Truly, DES is an object of beauty. To learn more about CO function, do an Internet search on “DES encryption” and feed to your heart’s desire or visit <http://csrc.nist.gov/publications/fips> to learn more about DES in general. See also (Stallings) for a good coverage of cryptography and DES.

At first, the DES algorithm was kept a national secret, as the thought of providing a recipe to nefarious Bart was abhorrent. It was finally published, which proved a good idea, as the “hacker/cracker” community could test its muscle. In fact, it did. A prize of \$10,000 was offered to crack DES by finding a particular 56-bit key value. In 1997, a worldwide user community of 14,000 PCs running

a cracker program for 4 months finally broke DES. Mind you, this was 4 months to discover the value of *one particular key*. However, the cat was out of the bag, and the security experts needed a better algorithm than DES. Interestingly, knowing the architecture of DES was of no help. The key was discovered by brute-force testing of all (or until the key was found) key possibilities.

What was DES's biggest sin? The master key length was too short at 56 bits. Let us assume that a 56-bit key could be discovered in 1s by a cracker. This is very optimistic, as the world record to crack DES is 22.5 hr using a massive array of PCs. For more information on how this was done, see [www.distributed.net](http://www.distributed.net). By simple scaling, it would take 4 min to crack a 64-bit length key; 76 bits takes 12.1 days, 112 bits takes a billion years, and 256 bits takes  $10^{52}$  years. So the trend is obvious and comforting to Bob and Alice and troubling to Bart. Clearly, Bob's message is safe for eons and then some as the key length becomes moderately large.

#### 8.4.1.1 Beyond DES

Since a single DES seems doomed, the crypto community tried a novel idea—triple DES. Consider a DES encryption (key1), followed by a DES decryption (key2), and then again by a DES encryption (key1 again). The overall effect is a 112-bit keyed cipher that is super secure. Not content and looking for efficiencies, two cryptographers in 1992 developed the International Data Encryption Algorithm encrypt engine. This works with a 128-length key, is twice as fast as DES methods, is more software friendly, and looks like a close cousin of DES in terms of operation. It has achieved the status of triple DES and is included in many cryptosystems in everyday use, despite the fact that it is patented and needs a license to use. Other coders in common use are the Advanced Encryption Standard and the stream cipher RC4 (which encodes a byte at a time, not a block at a time). Incidentally, these crypto methods can function in real time to process high-rate A/V data streams if needed.

#### 8.4.2 Keys and Key Management

Let us look again at Figure 8.10. Focus on the two secret keys. They must be the same for DES or the other common symmetrical encryption/decryption engines to operate properly, but how do both sides agree on a common key? Here is one way.

*Dear Alice,*

*Now that we have both purchased an EnigmaMan cryptosystem, let's start using it. My secret key is 831 408 625 989. To keep things simple, I plan to use this key for the next 10 years.*

*Cheers,*

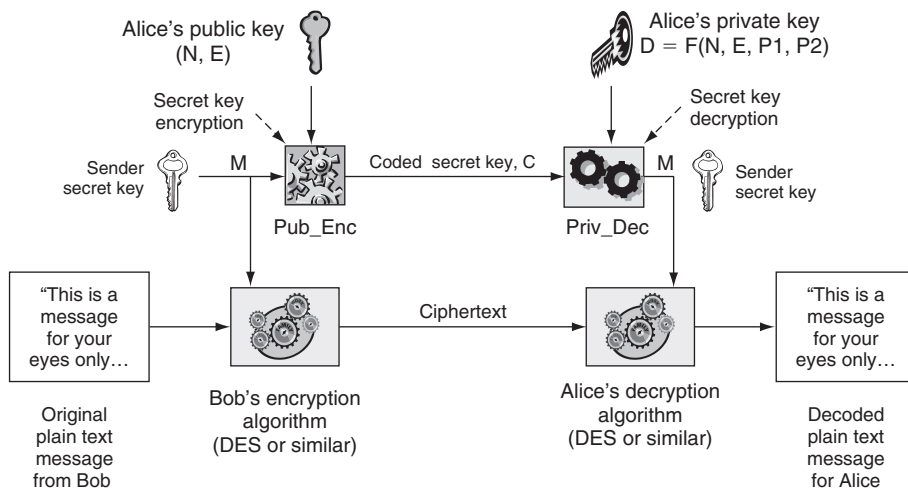
*Bob*

Eavesdropper Bart (who is always listening) loves this type of message, as he now knows the key's value and its lifetime. Whoopee! Imagine the living

nightmare of securely administering and distributing the keys for millions of multiparty coded conversations between Bobs and Alices. Simplifying this problem eluded cryptographers until 1976 when Whitfield Diffie and Martin Hellman published their seminal paper, “New Directions in Cryptography” (DiffHel). They introduced the idea of a public key and a private key instead of using two identical secret keys, as implied in Figure 8.10. The basic idea is that different keys are used for asymmetrical encryption and decryption (DES is a symmetrical algorithm). The public key is used for encryption, and the private key is used for decryption. The first key is publicly known, and the latter is kept strictly private.

A brief survey of the concepts behind the idea is worth our time, but a full discussion of this method is beyond the scope of this book. With the assistance of Figure 8.12, we can see the essence of the public/private key method. The lower portion of the diagram is essentially the same as Figure 8.10. The upper portion shows a “secret key” exchange method using a public key and a private key. Some of the salient aspects of Figure 8.12 are as follows:

- The plaintext message is encoded and decoded using the DES method or similar symmetrical algorithm. Coding and decoding use the same “secret key.”
- Alice’s public key is available to any Bob and is used to encrypt the secret key that is utilized by Alice to decode the plaintext message. Alice has a private key designed to decrypt any “message” that Bob sends. In this case the message is the secret key. The private key belongs to Alice and never leaves her possession, whereas the secret key must be used by both Bob and Alice. This is subtle and vitally important to the overall scheme.
- Pub\_Enc and Priv\_Dec are not DES or even similar to DES. Also, because these two algorithms are *not symmetrical*, the keys that operate them are not identical either. This is an important point of the public/private method.



**FIGURE 8.12** A hybrid method of coding using public/private keys.

Your first thought may be “This is complex with three different keys and three different crypto methods.” Think of it this way: in the big picture, this is the simplest way to solve the secret key distribution problem. One of the beautiful aspects of this method is the concept of a public key. It may seem that making a key available publicly as part of a cryptosystem is outlandish and just plain stupid. After all, why provide Bart with any hints? Despite publishing the public key, this method is extremely secure and has not been broken by any cracker.

A plaintext message is sent from Bob to Alice as follows:

1. Bob looks up Alice’s public key in a directory.
2. Bob encrypts his secret key using Alice’s public key and sends the encoded key to Alice.
3. Alice decrypts the secret key using her private key. At this point, both Bob and Alice have the secret key.
4. Bob now encodes his plaintext message using the secret key, and Alice decodes it using the same secret key.

Because the key exchange cryptosystem is used only for sending keys and not to encode the target plaintext message, the architectures of Pub\_Enc and Priv\_Dec need not be especially efficient or fast (on the order of 1,000 times slower than DES is okay), as they are used occasionally and only for short (<2,048 bits) keys. The magic of the public/private concept is embedded in the way in which the keys are created and how Pub\_Enc and Priv\_Dec function.

#### **8.4.2.1 The Public/Private Key Method**

Diffie and Hellman invented a creative and beautiful key exchange framework. Their main thesis is the following: What function  $Y = \text{Fun}(P)$  is easy to solve in one direction but fiendishly difficult to solve in the inverse?

An example of this is the deceptively simple one-way function  $Y = P1 \times P2$ , where  $P1$  and  $P2$  are prime numbers. Computing  $Y$  for values of  $P1$  and  $P2$  is easy even if the primes have hundreds of digits. For example, if  $P1 = 13$  and  $P2 = 19$ , then  $Y = 247$ . However, if asked to find  $P1$  and  $P2$  given the value 247, you would need to put in some effort.

Now if the  $P$ s are hundreds of digits long, imagine how difficult it would be to factor the resulting product if you had no hints. The current state of mathematics provides no fast methods to factor large numbers. Oh yes, there are tricks galore to speed up the factoring problem, but no quick fix methods exist. Given a factoring engine capable of  $\sim 10^{12}$  operations per second, it would take over 1,000 years to factor a 250-digit number (Stall) and about a million years for a 350-digit number. A composite number of 600 digits is currently beyond the scope of any machine to factor in a lifetime of the universe.

Diffie and Hellman did not utilize the factoring method but rather discovered another one-way function called the discrete logarithm. Finding secure

one-way functions is very challenging, and their method ranks among the truly great discoveries in cryptography. It turns out that by utilizing one-way functions, both public and private keys can be generated. Their paper inspired three other cryptographers to improve on the idea. In 1978, Ron Rivest, Adi Shamir, and Len Adleman of MIT proposed a method of public/private key generation now called the RSA algorithm. Their method proved so successful that RSA Security ([www.rsasecurity.com](http://www.rsasecurity.com)) was formed to commercialize the concepts. The RSA algorithm relies on the inability of machines to factor 200+ digit decimal numbers. The RSA algorithm is particularly complex, yet beautiful too. The basic idea is (refer to Figure 8.12) as follows:

- Alice produces a number  $N = P1 \times P2$ . She also chooses a value  $E$  (small, normally 3 or 7 or 65,537) that has a special relationship to  $P1$  and  $P2$ . She publishes her public key pair =  $(N, E)$  for Bob to use.  $N$  is normally hundreds of digits long. She keeps primes  $P1$  and  $P2$  private.
- Alice also needs to compute her private key ( $D$ ) and uses the function  $D = F(N, P1, P2, E)$  to do so. Note that she has the advantage of knowing  $P1$  and  $P2$ , whereas Bob and Bart do not.  $F()$  is defined by RSA and is not described here.
- Bob can access Alice's public key  $(N, E)$  whenever he wants to send Alice a secret key.
- Bob encrypts his message  $M$  (the secret key) by computing the value of  $C = M^E \pmod{N}$ .  $C$  is the cipher data that he sends to Alice. This RSA function is Pub\_Enc in Figure 8.12. The encryptor function raises  $M$  to the power of  $E$  and then applies Mod  $N$  of the result. Mod  $N$  is a remainder operator. For example, if  $X = A \text{ Mod } (B)$ , then  $A$  is the integer remainder when  $X$  is divided by  $B$ . If  $B = 9$  and  $X = 12$ , then  $A = 3$ , where  $A$ ,  $B$ , and  $X$  are integers.
- Alice receives the encrypted data stream and decrypts it using Priv\_Dec. RSA defines the decrypted value as  $M = C^D \pmod{N}$ , which is the recovered secret key in our example. Note the simple elegance of the RSA functions Pub\_Enc and Priv\_Dec.

Bob and Alice now have the same secret key, so they can proceed to send the actual message of interest using the lower part of Figure 8.12. Many books on cryptography spend 50+ pages to describe the public key method. Obviously, many details have been omitted here for the sake of simplicity. For an enjoyable and enlightening coverage of this method, see (Singh).

In practice, many real-world systems use either the Diffie-Hellman or the RSA Public/Private key exchange method. For example, IPSec uses Diffie-Hellman key exchange and SSL uses RSA methods. Even with the wonderful invention of the public/private key technique, there is a need for trusted agencies



to accept, store, and publish public keys. As you might imagine, there are all sorts of things that can go wrong unless someone manages the keys properly. An entire industry has been created to manage keys. Certificate Authorities (CAs) do the job of guaranteeing binding between the public key and the owner, thereby preventing masquerading. The Public Key Infrastructure (PKI) is a complete system for managing public keys and includes policies and procedures and digital certificates. A digital certificate is a short record that holds information about a person or organization, including any associated public key. A certificate binds a public key to its owner. For more information on CA, see, for example, [www.verisign.com](http://www.verisign.com) for practical solutions and white papers. Standard ITU-T X.509 specifies all the relevant details to implement a PKI system.

### 8.4.3 Kerberos

The much-used public key method is still a complex beast. As an alternative, the Kerberos (Greek spelling of the mythical three-headed dog that guards the entrance to the underworld) protocol is sometimes used. This method supplies identical private keys to Bob and Alice for their use in straightforward symmetrical DES encryption/decryption. It tends to be used at universities and within companies where a certain amount of trust can be placed in the operators. Over the raw Internet, the PKI is preferred over most private key exchange means. Still, Kerberos is widely used, and the open source code is available from MIT (RFC 1510).

The basic idea is that Kerberos uses two trusted third parties at the same time: the Kerberos server and the ticket-granting server. Bob and Alice transact with these servers to get a common secret key that they will subsequently use for exchanging their target coded information. The full transaction explanation is beyond the scope of this book (Wenstrom). Before leaving cryptography, we need to cover one more topic: digital signatures.

## FACTORIZING DIGITAL MONSTERS

Wars have been lost when a cipher was broken by the opposing side, so the interest in secure ciphers has led mathematicians to study factoring these long digit count monsters. Several organizations once offered challenge money of \$100 to \$200,000 to factor a given number. The amount of money is small compared to the effort expended. However, like climbing Mt. Everest, many will try to top it because "it is there." For many years, RSA offered (inactive now, but still unsolved) a \$100,000 challenge prize to find the two factors of  $N$  ( $P \times Q = N$ , find  $P$

and  $Q$ ) of the following 309-digit decimal number (1,024 bits). Happy computing!

```
1350664108659952233496032162788059699388
8147560566702752448514385152651060485953383
3940287150571909441798207282164471551373680
4197039641917430464965892742562393410208643
8320211037295872576235850964311056407350150
8187510676594629205563685529475213500852879
4163773285339061097505443349998111500569772
36890927563
```



#### 8.4.4 Digital Signatures (DS)

As the name suggests, a digital signature (DS) is a digital fingerprint of a data file. As with a traditional signature on a document, a digitally signed document attaches a person's acknowledgment or approval to the document. A DS is a form of checksum (or DNA strand by analogy) that may be used to validate a target data file's contents. Let us call a plaintext message PT and the DS of this as DS(PT). A DS is a hash function (also called a message digest) that "compresses" an entire PT no, matter how long, into a single value normally less than 256 bits. For example, the digitally represented signature of this book is a unique DS value. This value may be used to verify that an electronic file copy is indeed 100 percent identical to the digital master DS. Some of the aspects of a DS are as follows:

- DS(PT) is a value that uniquely identifies the PT. There is no other PT with the same signature ideally.
- Most signatures are relatively short, a la 160 bits.
- From the DS(PT) value, no one can reverse the function and produce the original PT. The function is secure.
- DS algorithms are published for open use.
- There are various hash functions in use today, with the most popular being SHA-1 (Secure Hash Algorithm), MD5 (RFC 1321), and RIPEMD-160.

As with most signatures, a digital signature may be used to verify who sent a PT message. Assume that Bob sends Alice a coded PT message, but Alice wants to know for sure that Bob sent it and not someone else (authentication). An example of this is the following:

- Bob calculates a single DS(PT) value for his plaintext message.
- Bob encrypts the value of DS(PT) using his private key and sends the encrypted value to Alice. Bob also sends Alice the PT as per the secure methods discussed earlier.
- Alice receives and decrypts Bob's PT coded stream, called PT\_Recovered.
- Alice also receives the encrypted value of DS(PT) and decrypts it using Bob's public key.
- Alice now computes DS(PT\_Recovered) and compares it to the received value of DS(PT). If the two match, then the received message was indeed sent by Bob (he is authenticated) and the message itself was unaltered.

The last step is the significant phase in authenticating that Bob is who he says he is and that the received PT is the correct message. Digital signatures are

used for a variety of applications, and this example is only one such instance. Message digests are well-accepted components in the big scheme of security, public key management, authentication, and the validity of messages.

In 2005 some university researchers discovered collisions for SHA-1. MD5 was shown to be vulnerable as well. This means that two different files produce the same signature. This is not good, and algorithm “fixes” are being studied. A collision is extremely rare, so for many applications, the problem is an academic issue.

The overall basics summarized in this section should provide you with sufficient understanding to plow through the minefield of cryptographic jargon. Don’t feel bad if these concepts don’t smoothly roll off your tongue; after all they are the results of hundreds of cryptographers’ efforts combined over 40 years. Fortunately, too, most applications hide the gory details from users. But it’s good to have a general knowledge of the main themes that are being employed daily in IT systems and the Internet.

Remember that cryptography is not the total solution to security. It’s really about process, not math. Encoding rules are only a small part of overall security methods.

## 8.5 IT’S A WRAP—SOME FINAL WORDS

Security methods will likely never become simpler than they are today. As threats increase, methods to protect against them will become increasingly more complex. This is a fact of life in the digital age. The networked requirements of A/V force security methods to account for real-time applications as never before. When deciding on security policy, factor in the real-time needs of A/V or pay the price of poor performance for applications.

## References

- Beale, J., et al. (May 2004). *Snort 2.1 Intrusion Detection* (2nd edition). Burlington, MA: Syngress Press.
- Diffie, W., & Hellman, M. (1976). New Directions in Cryptography, *IEEE Transactions of Information Theory*, IT 22/6.
- Networking, Analysis, Collaboration ([www.netapps.org](http://www.netapps.org)), Enterprise Security Architecture: A Framework and Template for Policy Driven Security, 2000, Executive Summary.
- Singh, S. (1999). *The Code Book*: Anchor Books ISBN-10: 0385495323.
- Stallings, W. (1995). *Network and Internetwork Security*. Saddle River, NJ: Prentice Hall.
- Wenstrom, M. (2001). *Managing Cisco Network Security*. San Jose, CA: Cisco Press.

# Systems Management and Monitoring

## CONTENTS

9.0	Introduction	346
9.1	The FCAPS Model	347
9.2	Traditional A/V Monitoring Methods	348
9.2.1	The Challenge	351
9.3	AV/IT Monitoring Environment	352
9.3.1	Traditional IT Device and Network Monitoring (Mon 1)	354
9.3.2	A/V IP Stream Monitors (Mon 2)	359
9.3.3	File Transfer Progress Monitor (Mon 3)	359
9.3.4	A/V File and Metadata Inspector (Mon 4)	360
9.4	Standards for Systems Management	361
9.4.1	The Management Information Base	362
9.4.2	The Simple Network Management Protocol	363
9.4.3	Web-Based Enterprise Management (WBEM)	365
9.4.4	Windows Management Instrumentation (WMI)	367
9.5	Service Diagnostics	368
9.5.1	Configuration Management	370
9.6	Futures—DCML	371
9.7	It's a Wrap—Some Final Words	372
	Reference	372

## 9.0 INTRODUCTION

Managing the IT infrastructure is one of the thorniest problems facing business executives today. It is not easy, vacuums up resources, and is ever changing. To some, the cost of managing IT is pure overhead with no apparent positive return on investment. However, when the gains of productivity, availability, and efficiency are counted, coaxing out every ounce of performance is wise and contributes to the bottom line.

This chapter reviews the fundamentals of systems management for pure A/V systems and hybrid AV/IT systems. Monitoring is a part of systems management and is covered with sufficient emphasis to be showcased in the chapter title. The following topics are discussed:

- Systems management 101: The FCAPS model
- A summary of traditional A/V monitoring
- IT monitoring methods
- An AV/IT system monitoring framework
- Systems management IT standards
- Methods for diagnosing problems

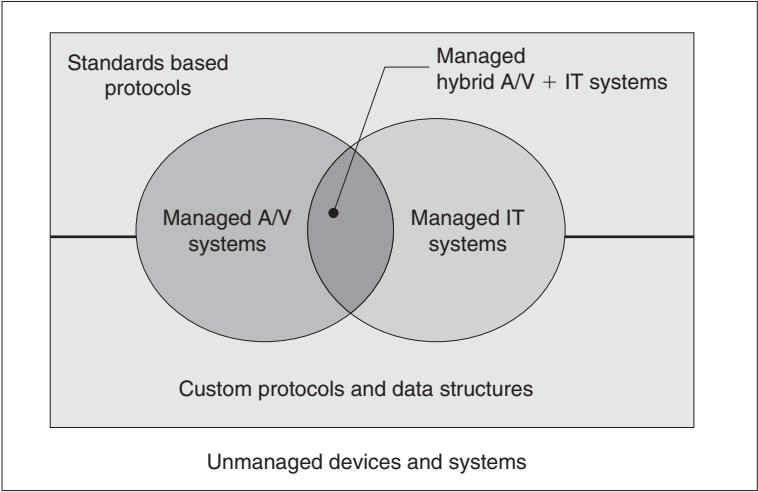
Remember, the management plane is part of the three-plane model discussed in Chapter 7, so it plays a pivotal role in IT systems. However it is the least developed of the three planes *for A/V systems*. Still, it is fitting to discuss the principles of the management plane, its developmental status, and how it can grow to meet the needs of A/V.

Who are the users of management systems? Broadly, they can be classed as

- Access to a device by *local service personnel*. They typically need the most detailed device reporting and diagnostics with hands-on access.
- Systems management for a campus or enterprise IT infrastructure—managing groups of devices from a central location by *IT staff*.
- Access to devices by *remote vendor service personnel*—ideally do any management operation remotely that can be done on-site. This challenges some A/V gear but enables remote troubleshooting.

The concepts discussed in this chapter apply to all three types of users.

Figure 9.1 illustrates the landscape of managed systems and devices. A “managed” device or system component (including software applications) is one that exposes its internal states to external observers over a network. A standalone device with only front panel error reporting does not fall into this category. This would be an unmanaged element. There are two general techniques to manage an element: using management standards or proprietary methods. Our focus is on standard methods. Also, there are three management domains for purposes of our discussion: traditional A/V, standard IT, and hybrid A/V + IT. All are considered. So let us get started.



**FIGURE 9.1** *The landscape of managed systems.*

9.1 THE FCAPS MODEL

In service of the telecoms industry, the ITU-T derived the FCAPS acronym to describe the salient aspects of systems management. It has been applied to managing IT as well. FCAPS (fault management, configuration, accounting, performance, and security) is a categorical model of the working objectives of network management. There are five levels: the fault management level (FM), the configuration level (CM), the accounting level (AM), the performance level (PM), and security level management (SM). Table 9.1 outlines the function of each level. Monitoring is an amalgam of select FM, AM, and PM functions.

The categories are self-describing, and Table 9.1 provides hints for each functional definition. This chapter focuses on fault management and performance management. But why not cover every column? Configuration management

Table 9.1 FCAPS Breakdown				
Fault Management	Configuration Management	Accounting Management	Performance Management	Security Management
Probe for measurements	System turn-up	Track service usage	Data collection	Control user access
Trouble detection	Device configuration	Bill for services	Report generation	Enable user functions
Alarm handling, logs access	Auto discovery		Test scripts	Access logs
Trouble correction with diagnostics tools	Back up and restore		Status reports	Secure devices
	Database handling			

is vital, of course, but there are precious few standards today. Although configuration management is fundamental, it also tends to be vendor specific, so it will be skipped for now. Accounting is also vendor specific and is applicable to facilities that rent equipment per hour or project for the most part. The security column was covered in Chapter 8. The lion's share of systems management, for our purposes, is related to monitoring, reporting, diagnosing, and repairing devices and systems.

Although not originally encompassed, the FCAPS model may be extended to cover Web applications management. It is not enough just to manage the IT infrastructure but ignore the applications layers. Modern systems management solutions can drill into applications as they run and report status, resource problems, performance, security issues, and so on.

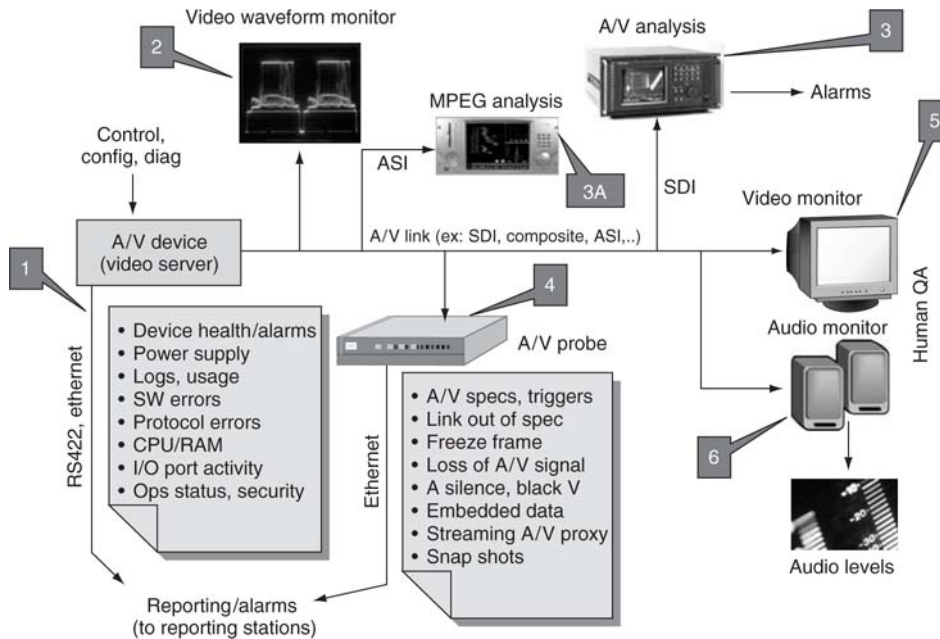
Ideally, all device/link/application-related FCAPS functions can be managed from a centralized station(s) connected to an IP network. Then hundreds of links, devices, and processes can all be monitored from one or more stations with integrated reporting across the entire system. This sure beats having to manage a multitude of devices individually with specialized protocols and methods.

If possible, every component, device, link, and network is 100 percent monitored, and faults (or preventative indicators) are reported with appropriate alarms. In some cases the cause of a fault is obvious, such as "Storage Array ABC, Disc 5 failed." There is no need to hunt down the bad element. The guilty party is replaced with a new disc drive. At the other end of the scale, the root cause of a fault may not be apparent. How should a technician respond to the reported fault "Protocol XYZ failure: audio level parameter L3 out of bounds"? Logs need to be scanned and interpreted and diagnostic tools run to discover the root cause(s) of the fault. These can be tricky problems to debug, and diagnostics tools are mandatory to avoid protracted troubleshooting. Of course, there is no magic bullet to resolve all problems, but applying the FCAPS model is a step in the right direction.

FCAPS implementation in the IT space has become an industry with many vendors supplying standards-based solutions. However, we are also interested in management solutions for hybrid A/V + IT systems, and this combination has been vendor specific with a only smidgen of IT-based FCAPS thrown in. Still, there is huge promise for FCAPS to be adopted by the A/V industry, and this chapter shows its potential. Of course, adopting the model is just a start. What is really needed is adoption of the *standards* that support the model. Before we explore this potential, we outline the status quo of A/V monitoring and problem resolution.

## 9.2 TRADITIONAL A/V MONITORING METHODS

The maturity of A/V standards has enabled a variety of monitoring techniques and methods. Figure 9.2 illustrates an exaggerated monitoring landscape. There are six monitor and reporting classes shown (the device class, the visual viewing



**FIGURE 9.2** Traditional A/V device and signal monitoring.<sup>1</sup>

class, and so on), and each is discussed in this section. The source of A/V is a video server for this example but could be any A/V source.

The most common monitoring method is to examine the internals of the device under test, point #1 in the diagram. This class enables the managed A/V device—the left lobe in Figure 9.1. Typical internal measured data points are listed. Monitored data are reported to remote observers over LAN or serial RS-422. There are several protocol means to report on the device health over IP networks. Many are proprietary, but one is based on an IT standard: the Simple Network Management Protocol (SNMP). This is an IETF standard for accessing measured device data over an IP network. SNMP is the workhorse for monitoring IT gear of all sorts. SNMP reads device internal data values arranged in a Management Information Base (MIB). MIB data reside inside the target device. A MIB is a tree-like data structure populated with measured values specific to the device under observation (alarms, power supply health, storage stats, I/O status, etc.). SNMP was designed to read this tree of data and return the values to a monitoring station. SNMP and MIBs are explained in more detail later in this chapter. Unfortunately, and this is key, there are very few standardized MIB definitions for A/V gear. Video servers, routers, switchers, and so on that have

<sup>1</sup> Element 3A reproduced with permission of Pixelmetrix, Inc. Element 3 reproduced with permission of Tektronix, Inc.



MIBs use proprietary ones for the most part. True, MIB data structures are usually published by the vendor to enable reporting, but they are still proprietary.

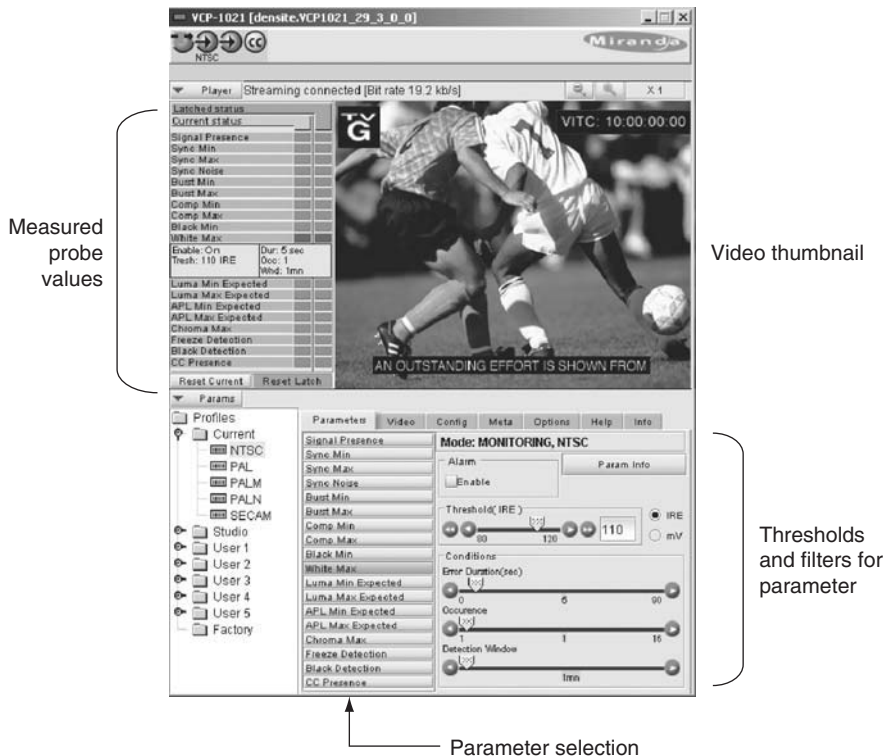
Device management is often based on vendor-specific technology and data structures. This frustrates plans for uniform facility monitoring as is available for IT systems. There is a nascent interest in establishing MIB A/V standards, but the work has been slow. However, MIBs for IT gear are plentiful and mature.

Monitor point #2 in Figure 9.2 is the time-honored video waveform monitor class. This time-based device displays one or more horizontal video lines. Video amplitude and timing-related specs are checked most commonly. A close cousin is the vectorscope used to measure color parameters. Even some non-technical users refer to the waveform display for a confidence check of min/max video luminance values. Some models can be configured to report over a network when illegal values are encountered. For a representative sample of signal monitors, refer to Tektronix ([www.tektronix.com](http://www.tektronix.com)), Harris ([www.harris.com](http://www.harris.com)), and Magni Systems ([www.magnisystems.com](http://www.magnisystems.com)).

Monitoring point #3 in Figure 9.2 shows the venerable VM-700 from Tektronix. This is the granddaddy of video measurement gear and can test every conceivable video spec. Some facilities use this as a quality and confidence monitor for master output feeds. Because these are expensive, simple-minded A/V probes are more practical for multilink monitoring. Point #3A in Figure 9.2 is a related method for monitoring MPEG TS streams over ASI links (and other links). The DVStation from Pixelmetrix ([www.pixelmetrix.com](http://www.pixelmetrix.com)) is such a device. Because DVB, ATSC, ISDB (Japan), and digital cable TV standards are all MPEG based, there is ample need to monitor/report on streaming MPEG signal integrity for these systems.

A relatively new class of monitor is the A/V probe (point #4 in Figure 9.2). These devices can be scattered throughout a facility and probe critical or suspect A/V links for signal integrity. They do not have an embedded display, such as a waveform monitor, but rather send a wide variety of real-time data points to centralized reporting stations. The method supports remote monitoring from virtually any Internet access point, which is powerful for multifacility monitoring from a central location. Each probe can be configured to report only when a trigger threshold has been reached. For example, say that the audio of a link is intermittent. A monitoring probe can be programmed to send a report only when the audio is silent (the trigger) for more than, say, 3 s. There is no end to the number of measured parameters and thresholds that can be programmed. Judicious configuration can find odd problems that rarely occur, which is a life saver for big system debugging. They can also be programmed to report continuously on select parameters for display on a central console.

Several vendors offer A/V probes with centralized reporting stations. Figure 9.3 shows an example PC screen shot from Miranda's iControl ([www.miranda.com](http://www.miranda.com)). It illustrates a full-featured reporting display that receives its data from Miranda



**FIGURE 9.3** A/V probe reported data.  
Image courtesy of Miranda Technologies.

A/V probes. All manner of stats, data points, alarms, trigger thresholds, latched triggers, meters, and proxies are displayed. Other providers include Evertz ([www.evertz.com](http://www.evertz.com)) with its VistaLINK series signal monitoring and reporting.

Monitor points #5 and #6 in Figure 9.2 are legacy visual and aural monitors. With standalone video, monitoring problems are detected only if someone is watching/listening at the moment of signal corruption. Modern control rooms and sports vans often use a monitor wall showing many channels simultaneously. They have embedded signal alarms and triggers based on A/V analysis. Barco's iStudio, Evertz's MVP, and Miranda's Kaleido are among several products that combine video images with overlaid data monitoring and alarms on the same display. See Figure 9.4 for a representative example.

### 9.2.1 The Challenge

Monitoring methods #2–6 are alive and well. They are based on existing A/V standards and signal links. A/V probes (#4) use custom data structures and reporting. Ideally, this could be standardized, but market pressure to do this is minimal at present. However, monitoring method #1 is problematic. A medium-sized facility may have 200+ devices from possibly 30 different vendors. Device



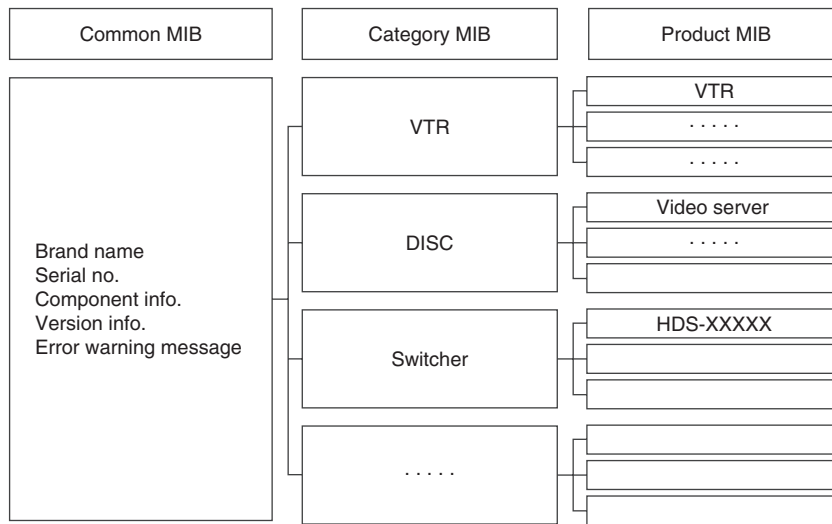
**FIGURE 9.4** Integrated wall display for A/V channel monitoring and stats.  
Image courtesy of Miranda Technologies.

management using 30 different protocols, data structures, and reporting screens is a nightmare. There are too few standards, and most solutions are ad hoc and vendor specific. Progress is needed to standardize device-specific MIBs with agreements on how to use Microsoft's Windows Management Instrumentation (WMI) for reporting OS, device resource, and application layer parameters. WMI is Microsoft's implementation of WEMB discussed in Section 9.4.4.

One exception to the status quo is a Sony effort to set a company-wide standard for professional A/V device MIBs. Called the Pro-AV MIB, it spans all of Sony's new A/V equipment. The MIB was first defined in 2001 and is now included with all new Sony professional products. Figure 9.5 shows a general outline of the structure of this MIB across product lines. Actually, the Pro-AV MIB is defined by combining MIBs from individual products. It is not yet a SMPTE standard, but it does have category MIBs for the most common A/V elements.

### 9.3 AV/IT MONITORING ENVIRONMENT

As systems migrate to hybrid A/V + IT technology, are strict A/V monitoring methods sufficient to meet users' needs? No, new monitoring methods are



As shown, the Pro-AV MIB is defined as a three-layered structure.

- Common MIB: Common information for all professional products
- Category MIB: Product category (i.e., VTR, camera, switcher)
- Product MIB: Product-specific information

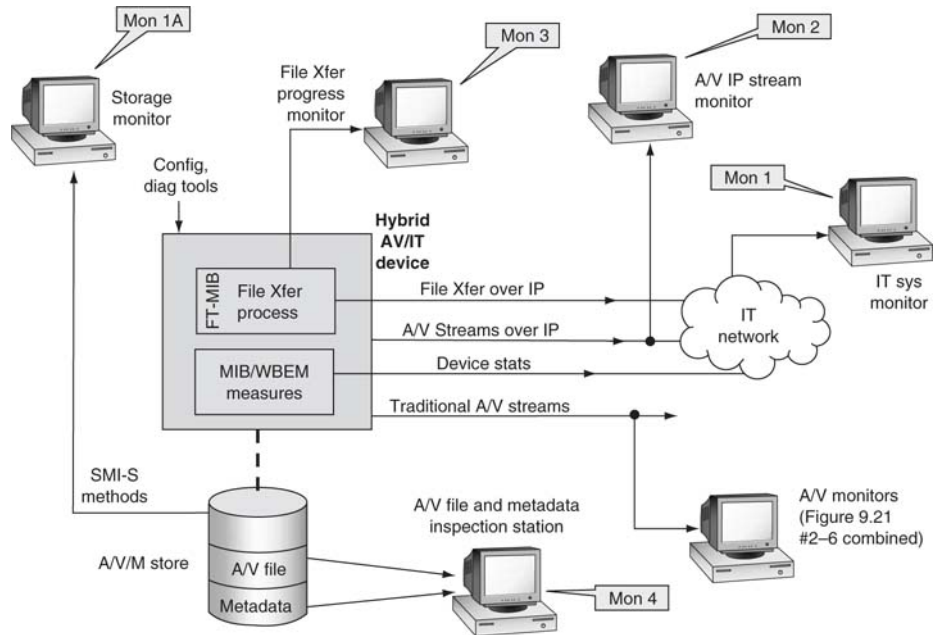
**FIGURE 9.5** Sony Pro-AV MIB general overview.

needed. Ideally, we need to add *functionality* to Figure 9.2 to include at least the following:

1. IT systems integrity—links, routers, rates, QoS measurements, alarms, performance, and usage. Storage system health, stats, reports (online, near-line, offline, and archive). (Mon 1, 1A)
2. File transfer progress reporting (Mon 3)
3. A/V file analysis, integrity, proxy viewing (Mon 4)
4. Metadata browsing (Mon 4)
5. A/V stream taps. Monitor an A/V stream over a network connection (possibly UDP/IP based) with a non-intrusive monitor probe (Mon 2)
6. Workflow reporting, job progress reporting

This domain is the merged overlap area shown in Figure 9.1. Mon 1–4 in the preceding list indicates reporting methods referenced in Figure 9.6.

All but the first item in the list are A/V specific. Standard IT means are sufficient to meet just about any network and storage measurement or monitoring scenario (#1 in list). However, functions #2–6 are not commonly available on centralized reporting displays, although some of these are available using vendor-specific tools.



**FIGURE 9.6** Full-featured AV/IT monitoring environment.

Figure 9.6 illustrates a full-featured AV/IT monitoring environment. It uses all the methods shown in Figure 9.2 and augments them to include the functions in the list given earlier. The added items are identified as Mon 1/A, 2, 3, and 4. They are shown separately but could be combined into one or more stations for convenience as needed. Each one will be outlined.

### 9.3.1 Traditional IT Device and Network Monitoring (Mon 1)

Mon 1 is the traditional IT infrastructure monitor means. This is the right lobe in Figure 9.1. Using MIBs (data structures populated with monitored and static values) and SNMP (the protocol used to retrieve MIB elements),<sup>2</sup> most IT elements may be monitored. Mon 1A in Figure 9.6 is dedicated to storage management, but is really just a subset of Mon 1. This area is well developed, with hundreds of vendors supplying reporting gear based on many standardized MIBs and other data structures. For example, the following grouping is of common network device-related MIBs as defined by the IETF:

RFC 1493 Bridge MIB; RFC 1213 MIB II; RFC 2096 IP Forwarding Table MIB; RFC 2737 Entity MIB; RFC 2665 Ethernet MIB; RFC2819 Four

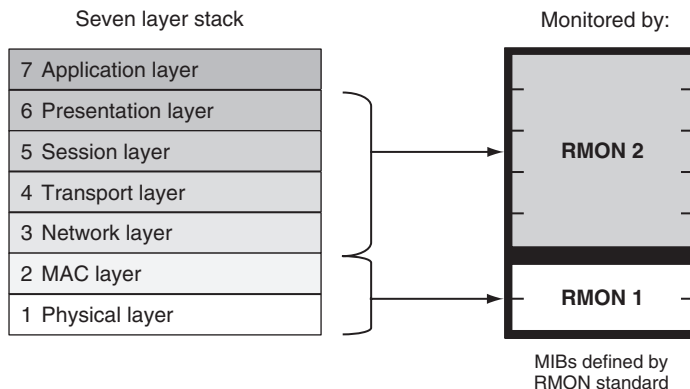
<sup>2</sup> MIBs, SNMP, and WBEM are described in more detail in Section 9.4. If you cannot wait, read ahead to learn more about these important IT-related specifications.

groups of RMON: 1 (statistics), 2 (history), 3 (alarm), and 9 (events);  
RFC 2021 RMON probe configuration; RFC 1850 IP Routing MIB

In fact, 145 official IETF RFCs in early 2009 referenced the word *MIB* in their title. In addition, there are thousands of proprietary MIBs. A few standards of noteworthy mention are MIB-I, MIB-II, and RMON MIBs. One of the first MIBs to be defined is called MIB-I (RFC 1156) and is used to manage the TCP/IP protocol suite. MIB-II is an updated version and is included in most devices that support TCP/IP. It is shown later in Section 9.4. The MIB is a tree structure with the upper part leading a path to the actual data elements in the lower part.

The RMON (versions 1 and 2) specification is an extension of MIB-II. RMON stands for remote monitoring and was designed to enable vendor-neutral IP stack network monitoring. A RMON-compliant probe (an IP stack parameter measurement device) can accumulate data and report when thresholds are reached. See Figure 9.7 for the breakdown of RMON 1 and 2 measurement domains. Because RMON defines a MIB standard, vendors can compete by selling probes and centralized reporting stations, knowing that the entire food chain is defined. The RMON equivalent for A/V-specific gear and links does not exist, yet. However, the Pro-AV MIB described earlier comes close.

In addition to MIBs and SNMP, the Web-Based Enterprise Management Initiative (WBEM) is a separate set of technologies developed to manage enterprise computing environments (integration platforms). WBEM provides the ability to deliver an integrated set of standard management tools that leverage Web techniques. More on this in Section 9.4. One special area is storage management. The Storage Networking Industry Association ([www.snia.org](http://www.snia.org)) has defined how to use WBEM to manage storage systems. Networked video relies heavily on storage, so let us review the development of SNIA.



**FIGURE 9.7** RMON MIB monitoring for the IP stack.

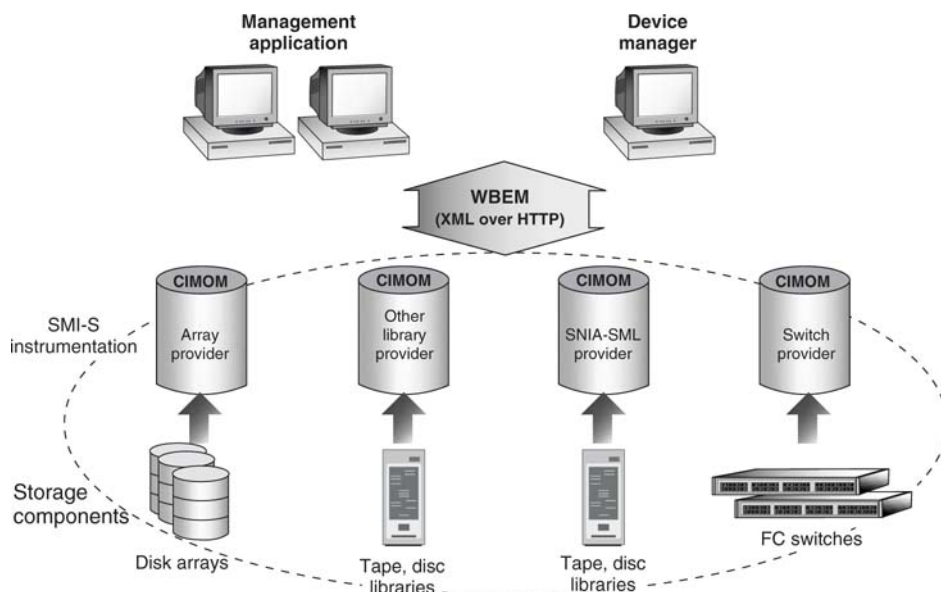
### 9.3.1.1 The Storage Management Initiative (Mon 1A)

The SMI was developed by the SNIA in response to the crying need for storage management across heterogeneous platforms. At its core, the SMI is composed of the following:

- Enabling and streamlining the integration of multivendor storage systems
- Leveraging established management methods
- Encouraging management consolidation (ending the nightmare of five management consoles for five different storage systems)
- Providing a common management interface for vendors to develop compatible products
- Offering interoperability and testing suites
- Providing SMI-S—the technical specification

SMI is more than a specification; it is an initiative spanning education, specs, interop testing, and marketing of the concepts. SMI-S is the specification portion, which is our focus. The SNIA states that the majority of new Fibre Channel SAN-based storage will be SMI-S compliant. Because virtually all storage vendors support this initiative, eventually most SAN storage systems will support it. The SNIA also supports NAS and iSCSI storage.

Figure 9.8 is a top-level view showing arrays, libraries, HBA, and FC switches being managed using WBEM. The basic idea is to leverage WBEM along with CIM (see Section 9.4 for details) for element, application, and system



**FIGURE 9.8** Overview of SMI-S management methods.  
Concept: SNIA.

management. CIM is an information model, a conceptual view of the managed environment that unifies and extends the existing management standards (SNMP, MIBs) using object-oriented constructs and design. The CIM model (CIM Object Model in Figure 9.8) is populated with reported data values that a management station may query to determine the state of a device or process.

The SMI-S 1.1 specification documents a secure and reliable interface that allows storage management systems to identify, monitor, provision, configure, and control physical and logical resources in a storage system. Importantly, SMI-S is much more than dumb element monitoring (reporting a bad fan) and includes configuration and a hierarchy of managed objects. The bottom line is this: administrators can manage heterogeneous storage platforms using a standardized management interface and vendor-neutral management stations. Storage administrators can use one application for many of the operations that traditionally take several vendor-specific management products. This brief coverage only scratches the surface of SMI-S (refer to [www.snia.org/forums/smi](http://www.snia.org/forums/smi) for a deeper look).

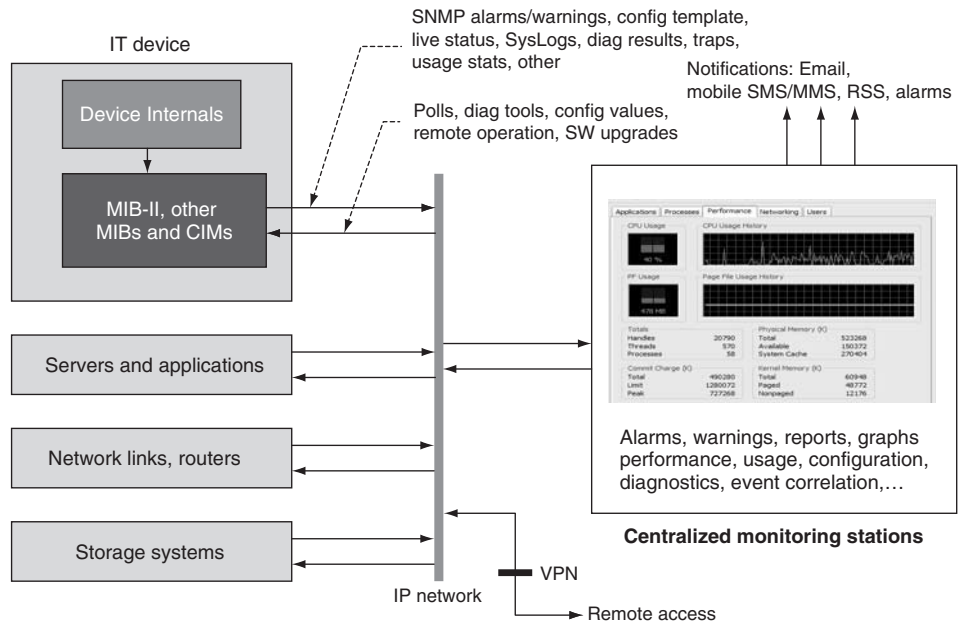
### **9.3.1.2 Centralized Enterprise Reporting Stations (Mon 1, 1A)**

At a higher level, there is an entire industry providing enterprise-wide management stations that hide the complexity of MIBs, SNMP, and WBEM. After all, what users really want are reports and notifications informing them of alarms and potential troubles and assisting with problem diagnostics and resolution. No one wants to read strings of MIB variables or CIM models to find a problem. Managers want the fewest people to administer the most system components. Figure 9.9 illustrates a simple monitoring configuration with a centralized station(s) monitoring potentially thousands of IT elements.

Some valuable performance attributes of the reporting station are as follows:

- General alarm reporting, including notifications to email, SMS/MMS, and so on. Support for all standard reporting means—standard MIBs, SNMP, WMI, WBEM, OMI, and so on.
- Resolving power—reporting the root cause of a problem if possible. Some problems can cause an alarm storm due to cascading errors. A good reporting tool will determine the root cause and give it priority over possibly hundreds of other cascading alarms.
- User-friendly screens—the ability to dig as deep into a system as needed. This may include embedded configuration diagrams for easy tracing of problems.
- Settable thresholds—this allows for a user to set trigger points in a device for select parameters and report only when the trigger has been reached; for example, report only when the storage capacity reaches 10 percent free space.





**FIGURE 9.9** *IT element monitoring environment.*

- Copious reporting choices—charts, graphs, and textual reporting of all sorts.
- Optional IT element diagnostics, configuration, performance, usage, event log viewing, and status are valuable features. Depending on the element(s) being managed, configuration, diagnostics, performance, and so on may be available only from the element's vendor as a standalone function and not integrated into a generic reporting station.

It is not easy to monitor A/V gear in real time, especially accessing internal, low-level parameters. The procedure of monitoring may interfere with the A/V operation. By way of analogy, this is related to Heisenberg's uncertainty principle. The more precisely a measurement is made, the more likely it is to interfere with process operations. It is left to A/V vendors to design their devices to be monitored non-invasively in real time. Non-real-time monitoring is easy but a nuisance, as it may miss critical events.

There is a world of choice when selecting enterprise systems management software. HP's OpenView (part of HP Software suite since 2007), IBM's Tivoli Management Software, and Computer Associate's Unicenter are the big three providers. Of course, there are 30+ smaller providers that often specialize to differentiate in some manner. The big three offer a complete line of management solutions, whereas others usually offer vertical solutions in select areas. None of these players specialize in A/V systems. However, several A/V vendors offer

entry-level, vertical management platforms dedicated to their own products and some third-party ones. Some sample offerings are Avid's Administration Tool for Unity, GVG's NetCentral, Harris's CCS Pilot, Miranda's iControl, Snell and Wilcox's Roll Call, and Sony's MMStation.

Look ahead to Figure 9.17, which demonstrates an example of a hierarchical system configuration as viewed on a management station. Users may drill deeper into any subsystem by selecting it for exposure. This example shows two levels, but more are likely for larger systems. At the lowest level, dashboards of device health are common. This example shows a video server the MIB with particular focus on the configuration details of the device. This aspect of Figure 9.17 is discussed later in the chapter.

### 9.3.2 A/V IP Stream Monitors (Mon 2)

The Mon 2 class of monitoring reports on A/V stream signal integrity over IP links. In a way, it is an extension of traditional A/V monitoring (class 3 in Figure 9.2) except for IP links. One such method uses UDP/IP and the RTP streaming protocol to carry MPEG, DV, uncompressed HD (RFC 3497), or other A/V payloads. Another model carries MPEG over IP directly. MXF has stream support, but there is little usage so far. Expect to see more IP usage as SDI is replaced by IP connectivity. However, as mentioned in Chapter 2, SDI will not replace IP streaming anytime soon for live event scenarios. The DVStation from Pixelmetrix is one such monitoring device, but this class is immature at present.

### 9.3.3 File Transfer Progress Monitor (Mon 3)

The usage of file transfer to move A/V assets has become commonplace. Many everyday operations, once based on SDI or composite links, are now performed using files. Anyone who has waited for a file transfer to complete knows the value of a monitoring display showing the stats of the transfer—time to go, percentage complete, stalled, failed, and so on. Frankly, there should be a federal law forcing vendors to always provide a meaningful progress monitor—the hourglass icon does not count!

As file transfers consume a larger part of overall A/V operations, it makes sense to have a unified way to monitor their status. One could argue that separate vendor-supplied custom monitors (usually embedded in an application) will always be acceptable, but then there is no consolidated view of systemwide transfer operations. Incidentally, if the file transfer is considered a stream, then the Mon 2 method may be used in some cases. Generally, however, monitoring the actual file as it moves across a link has little value and is difficult to implement.

One way to enable a universal view is to define a file transfer MIB populated with live stats. If 10 products from different vendors are all moving files independently and each supports this proposed FT-MIB, then a central station can easily monitor all transfers on one display. The FT-MIB would reside in the

box titled “File Xfer Process” in Figure 9.6. Some may reason that the individual transfers lose their meaning on a centralized report. However, a centralized knowledge of all transfer stalls/failures, completion of crucial scheduled transfers, and aggregate consumed bandwidth is valuable information for managing the overall facility. Defining the FT-MIB is a simple task, and possibly ~15 parameters would suffice. A suggested parameter list is as follows:

- File name, file size
- Transfer ID tag (tracking number)
- Source machine name, process name, and ID
- Destination machine(s) name, process name, and ID(s)
- Time of transfer start, estimated time of completion
- Percentage complete (updated every 2 seconds or so)
- Average Mbps transfer rate, peak, minimum
- Status: transfer in progress, stalled for  $N$  seconds, failed (failure code)

There is an old saying, “If you can’t measure it, then you can’t improve it.” The FT-MIB is a step in the right direction in improving file transfer efficiency and overall infrastructure utilization.

#### 9.3.4 A/V File and Metadata Inspector (Mon 4)

When you are inspecting a videotape, text labels are indispensable for identifying the contents. It is a snap to read a tape’s label and/or preview it on any active VTR. When you are using files, it may not be as straightforward to know what is in a file or to preview it. Enter the *file inspector*. This class is not strictly monitoring but rather examination. There are no alarms, warnings, or MIBs. Rather this class enables easy viewing of stored assets and associated metadata. It is a MAM element, as discussed in Chapter 7. A short list of file inspection criteria is as follows:

- Query for and locate one or more files.
- Browse file on a desktop client and view basic properties.
- Read the associated structural and descriptive metadata as appropriate.
- Move a file across the network as an option.
- (Optional) Test for validity and quality of a file; make corrections to file if needed.
- Perform validity checks for format legality—MPEG syntax, for example.

These steps are an analogy to the tape-based inspection case. Most MAM vendors offer file browsing with these features. Having access to inspection browsing reduces user anxiety in tapeless environments. Plus, with sufficient determination, it allows personnel to debug problems, find misplaced assets, and locate poorly indexed metadata. As it happens, several vendors provide utilities for testing file format (MPEG, MXF, XML metadata, etc.) legality. See the Harris QUIC product for file analysis and syntax correction.

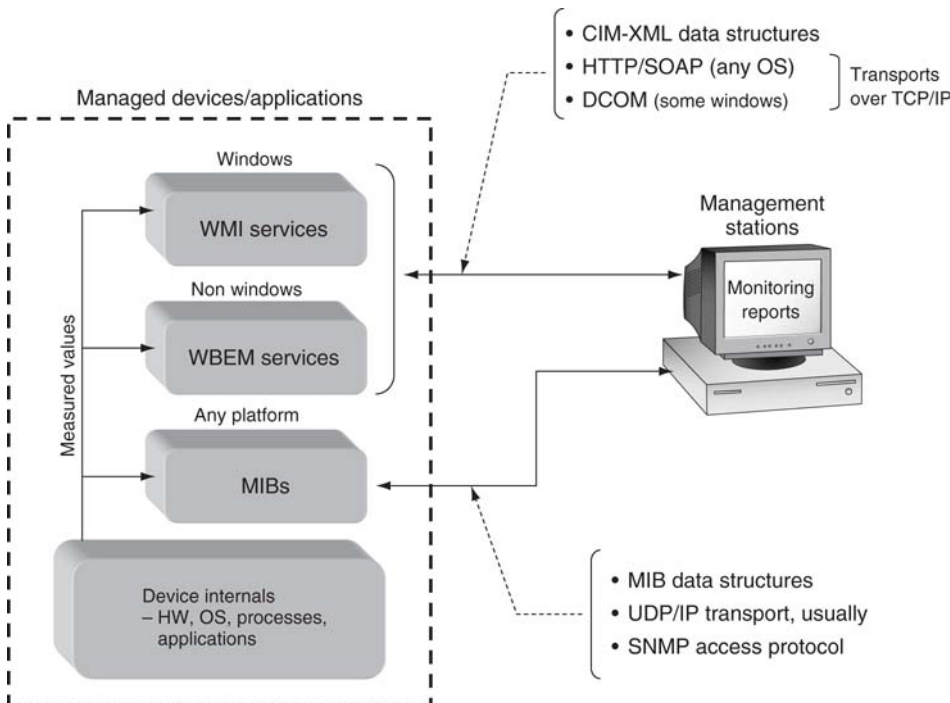
So, there it is. Figure 9.6 outlines the salient aspects of a full-featured monitoring system. Most systems will have some subset of this universal viewpoint. Those who ignore monitoring will suffer when troubles arise. Some of the pieces are in place today, some are coming on the scene, and some are long overdue. As IT moves deeper into A/V systems, expect to see more growth and standards in these areas. If you are a user, query any potential vendors for their response to the issues and features that make up the management plane.

## 9.4 STANDARDS FOR SYSTEMS MANAGEMENT

The world of managed systems is built on a strong foundation of standards. Actually, layers of standards. There are standards for data types, data structures, access methods, and transport. Figure 9.10 shows a high-level view of the relevant device and application management standards that impact IT and A/V systems. Of course, there are countless custom ways to create management systems, but this section focuses on industry standards and their functions.

The chief standards and controlling bodies are as follows:

- Management Information Base—IETF body
- Simple Network Management Protocol—IETF



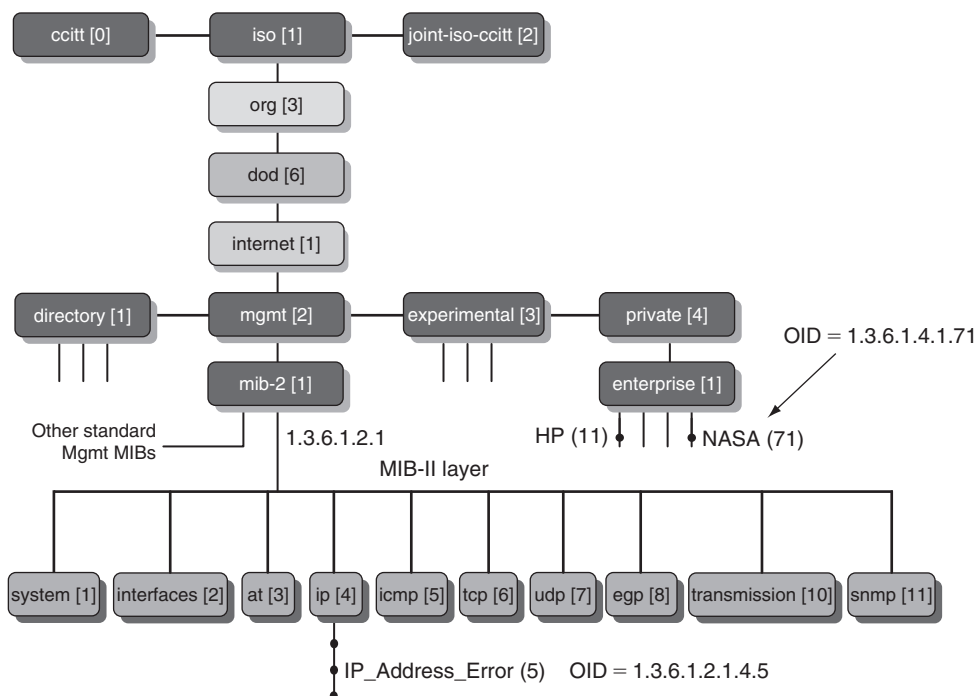
**FIGURE 9.10** System management standards.

- XML, SOAP, HTTP, TCP/IP data wrapping and transport standards—W3C, IETF
- Web-Based Enterprise Management Initiative (WBEM)—DMTF body
- Common Information Model (CIM)—DMTF
- Windows Measurement Instrumentation (WMI)—Microsoft implementation of WBEM
- CIM extended for Windows environment (de facto standard)

Each of these standards is explored in order to shed some light on their underlying function and place in the overall management food chain.

### 9.4.1 The Management Information Base

A MIB is a tree structure containing human-readable data values. Its base standard is RFC 1155. Each node in the tree contains an element value. In fact, all the values are ASCII textual values. Any text processor can display the contents of a MIB in much the same way that an XML file can be examined. An example of a MIB is shown in Figure 9.11. The tree structure is obvious. Each node is labeled with a number, the object ID (OID). The label identifies an associated value. The OID is a kind of address. To locate any element in the MIB tree is as simple as following the OID pointer. Some key nodal values are



**FIGURE 9.11** MIB-II definition.

- International Standards Organization (ISO); node OID = 1
- Organization; node OID = 3 (an ISO-recognized body)
- Department of Defense (DOD); node OID = 6
- Internet; node OID = 1
- Mgmt; node OID = 2 (all elements below this node are management data)
- MIB-II; node OID = 1 (standardized IP stack measures)

The full OID 1.3.6.1.2.1 points to the top of the MIB-II data structure. Just for fun, enter this number into an Internet search engine to appreciate the ubiquity of MIB-II in managed products. MIB-II nodal objects are defined by RCF 1213. One example of a MIB-II node element value is “IP\_Address\_Error” at OID = 1.3.6.1.2.1.4.5 descending from the node IP (4) in the MIB. This value is defined by MIB-II and is the count of received IP address errors. Of course, the full MIB-II contains several hundred nodes.

There are thousands of defined MIBs, some standardized and some private. The private MIB “enterprise” OID always begins with 1.3.6.1.4.1. For example, 1.3.6.1.4.1.11 is reserved for all of HP’s custom MIBs. Do an Internet search using this exact value to prove it. All companies with private enterprise MIBs need an assigned OID.

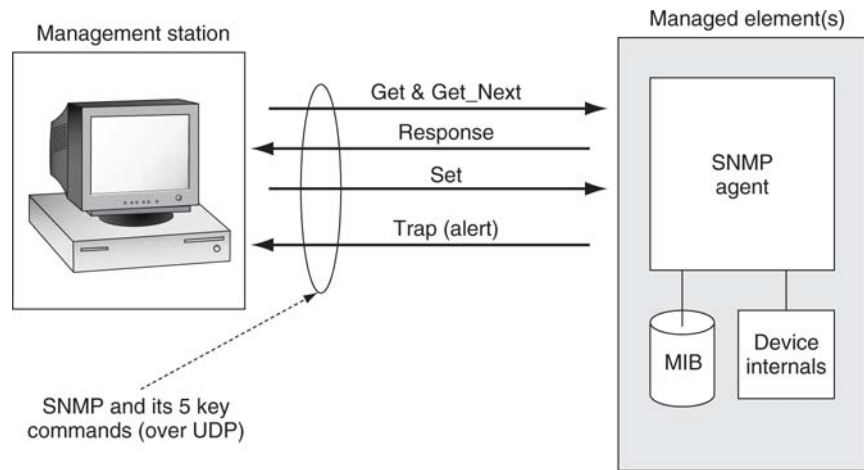
End users rarely have to worry about building MIBs; that is left up to the equipment vendors. So the gory details of how to compile a MIB and what variables to include should not be an issue for end users. However, viewing the MIB may be required if the management station does not recognize it. In that case, most stations will display (using a MIB browser) the values of the MIB without knowledge of their root meaning. It is not easy to understand raw MIB values, as the viewer needs a MIB dictionary (which maps OIDs to the value of a node) to make sense of the data. Make sure your equipment vendor supplies a dictionary for any MIBs that are not standardized. You never know when the dictionary will be needed when debugging odd problems.

The MIB structure is designed to be “crawled” to read out its values, and SNMP is designed specifically to do this. The next section provides the basics of this universal tool.

### 9.4.2 The Simple Network Management Protocol

SNMP is the foundation protocol for monitoring system elements. Because simplicity breeds ubiquity, SNMP is supported by nearly 100 percent of IT-based devices. Its main function is to interact with element MIBs. Figure 9.12 provides a glimpse of a monitoring environment and why SNMP is indeed simple: it has only five commands. See (Mauro 2001) for a good introduction.

Figure 9.12 shows four main components: the management station, the managed element with the SNMP agent and associated MIB, and the SNMP. Agents are software modules that reside in system elements. They collect and store management information (errors, stats, counters, etc.) in the MIB and



**FIGURE 9.12** Basic monitoring using SNMP.

provide SNMP support to R/W MIB variables. The five main commands shown in an illustrative way are

- **Get(1.3.6.1.2.1.4.5)**—read a variable from a MIB at node OID
- **Get\_Next(OID)**—read the next consecutive element in the MIB tree (OID + 0.1)
- **Response (returned value)**—value from Get or Get\_Next request
- **Set(OID, value)**—set the MIB element at node OID to value
- **Trap (returned value)**—trap sent in response to some threshold reached

It is a snap to read a MIB using the Get or Get\_Next commands. Sets can be useful for writing values into a MIB, but it is suggested that this feature not be overused because it quickly leads to poor data management and access rights issues. Get\_Next is useful for efficiently traversing the elements of a MIB. Traps are responses to some target MIB variable reaching a critical threshold. If too many traps are set or their trigger threshold is too low/high, trap storms can arise when some upstream problem occurs. Use traps and sets judiciously.

There are three versions of SNMP: SNMPv1 (RFC 1155), SNMPv2C, and SNMPv3, each in turn with more functionality. SNMPv1 and SNMPv2C define administrative relationships between SNMP entities called *communities*. Communities group SNMP agents that have similar access restrictions with the management entities that meet those restrictions. All entities that are in a community share the same *community name*. To prove you are part of a community, access is tested against the community name during the SNMP dialog. SNMPv3 defines the secure version of the SNMP protocol.

SNMPv3 addresses security by adding two new features: authentication via hashing and time stamps and confidentiality via encryption. A management application is authenticated via a SNMPv3 remote device before being

allowed to access the MIB variables. In addition, all of the requests/responses between the management station and the remote device are encrypted to prevent snooping.

Interestingly, SNMP runs over UDP, not TCP. Why? Because UDP requires low overhead, the impact on a network's performance is reduced. SNMP has been implemented over TCP, but this is more for special-case situations over long distances. It is the responsibility of the management station to guarantee data integrity, which can be done using time-outs and retries. Also, the set operation is problematic, as there is no guarantee that it will actually change the desired variable.

In A/V applications, many devices support SNMP. The real issue is whether the MIBs that are accessed are standards or vendor proprietary. If custom, then users are practically restricted as to their choice of management stations, usually vendor supplied. Of course, if all management is Web based (with a Web server in each managed device), then this is a moot point and SNMP is not involved. The plea for A/V MIB standards cannot be proclaimed loud enough, or else users will be forced into using closed, vendor-supplied, management stations.

MIBs and SNMP do not have a monopoly on management standards. The next section considers the other standards cited in Figure 9.10.

### 9.4.3 Web-Based Enterprise Management (WBEM)

The Distributed Management Task Force (DMTF) is the leading industry organization for the development of management standards and integration technology for enterprise and Internet environments. Its roster has reached 4,000 participants from 43 countries. Many of its participants provide IT devices enabled with WBEM technology.

The WBEM initiative was created to provide common management infrastructure components for instrumentation, control, and communication in a platform-independent and technology-neutral way. DMTF technologies include information models (CIM), communication/control protocols (WBEM), and core management services/utilities.<sup>3</sup>

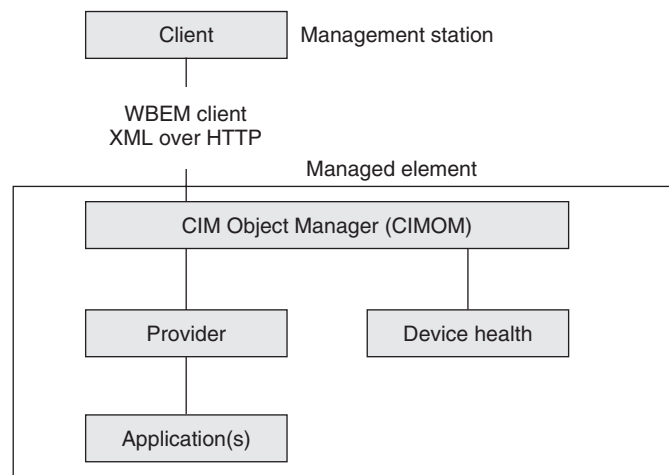
The goal of the DMTF was to create a set of standard methods to manage devices, applications, and infrastructure at a high level. The aim is more far reaching than monitoring a set of MIB variables.

A basic outline of the architecture is shown in Figure 9.13. A core piece of the puzzle is the Common Information Model (CIM). The CIM is based on a description language. This is an object-oriented model, describing an organization's computing and networking environments (its hardware, software, and services). All managed elements are positioned within this model, streamlining integration by enabling end-to-end multivendor interoperability in management

---

<sup>3</sup> Some of the information in this section is paraphrased from the [www.dmtf.org](http://www.dmtf.org) Web site.

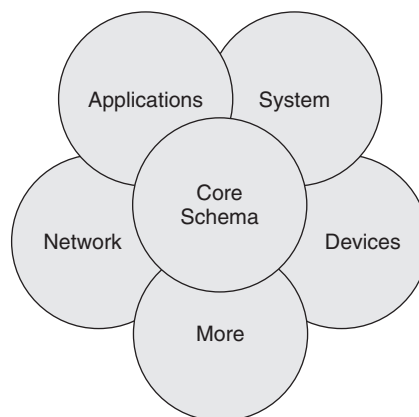




**FIGURE 9.13** WBEM-based device monitoring.

systems. As a crude analogy, its counterpart is the MIB. Although a MIB has no notion of object-oriented design, both concepts describe management information schemas.

Each management area, such as networks or applications, is represented in a CIM schema. Different management areas are worked on by different DMTF specialty groups. Figure 9.14 shows how existing CIM schemas are conceptually layered. A core schema is at the center, and schemas then build on each other to represent more specific management areas. At present there are 10 (not all shown in Figure 9.14) CIM management schemas. See [www.dmtf.org](http://www.dmtf.org) for full definitions of the CIM schemas.



**FIGURE 9.14** CIM management schemas.

The CIM data models are managed by the CIM Object Manager (CIMOM) component. It communicates with the management station using an XML mapping of the CIM data elements over HTTP. Think of the CIMOM as a way to access data that the CIM schema describes. XML/HTTP is a common method in Web environments to interchange data elements. Again, by way of simple analogy, this is similar to SNMP, but much more flexible in operation. One advantage is that it uses TCP transport compared to SNMP's unreliable UDP. This is especially valuable when management stations span the Web.

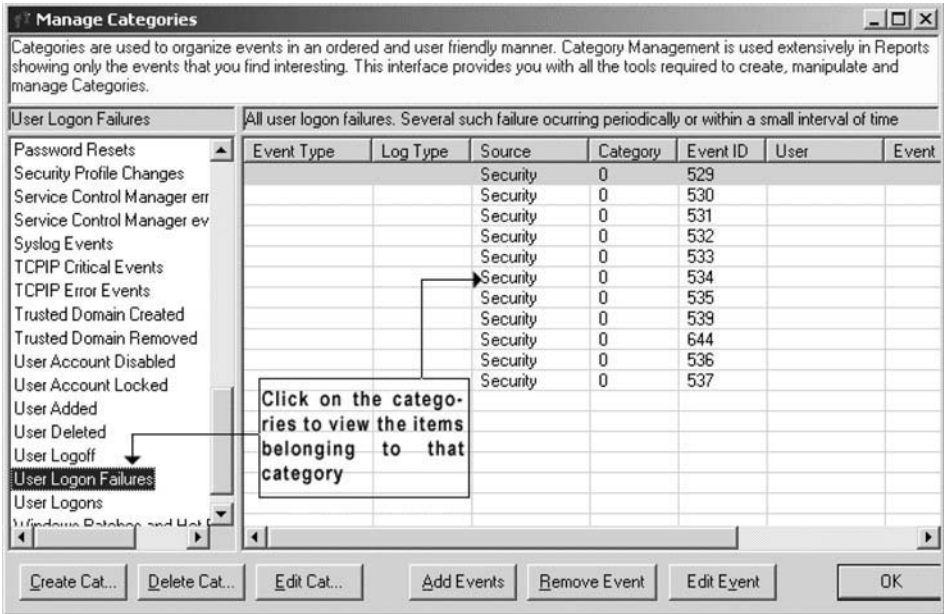
In theory, anything that can be managed using MIB/SNMP can be managed using WBEMs. However, the converse is not true due to the power of WBEM methodology. For example, monitoring of user application's performance, security, and status is rarely done using MIBs but is common using WBEM. WBEM encourages high-level monitoring of applications, devices, and networks simultaneously to get the most information of overall system performance and status. With the hegemony of Microsoft in enterprise IT, it is wise to ask if it supports WBMS. Well, yes and no. MS developed an implementation of WBEM called WMI.

#### 9.4.4 Windows Management Instrumentation (WMI)

Windows Management Instrumentation is a component of the Windows operating system that provides management information and control in an enterprise environment. Using CIM standards, managers can use WMI to query and set information on desktop systems, applications, networks, and other enterprise components. Developers can use WMI to create event monitoring applications that report when incidents occur. One area of incompatibility is how an application writes codes to WMI versus WBEM. WMI provides an API to access the CIMOM, and this API is not the same as, for example, a Linux implementation. As a result, this incompatibility complicates application porting. The Microsoft Web site offers many tutorials and white papers on WMI and its use in systems monitoring.

Because WMI is part of the Windows OS, practically any OS operating parameter may be monitored locally or by remote means. For example, Figure 9.15 shows a screen shot from the EventTracker product available from Prism Microsystems ([www.prismmicrosys.com](http://www.prismmicrosys.com)). This tool collects and consolidates all the Microsoft event logs from remote Windows devices. It can be configured to dig as deeply as needed to report on virtually any device-related parameter. User applications that are WMI compliant may be monitored with this tool. The Windows OS supports hundreds of WMI-monitored parameters. Many hardware-related stats may also be monitored using WMI.

To see what a local Windows XP/Vista machine event log looks like, examine the tools at Start, Control Panel, Admin Tools, Computer Management for a glance at some useful reporting screens.



**FIGURE 9.15** Console screen shot of EventTracker application (relies on WMI).  
Source: Prism Microsystems, Inc.

The bottom line is that more and more IT devices, networks, and applications will depend on WBEM/WMI for management. It is not yet common to use these methods for A/V-related applications. The usual mantra applies—as IT digs deeper into traditional A/V applications space, these methods will find use. So get ready! It is always good form for users to query A/V vendors on their product’s management ability, including support for application-level WBEM/WMI.

### 9.5 SERVICE DIAGNOSTICS

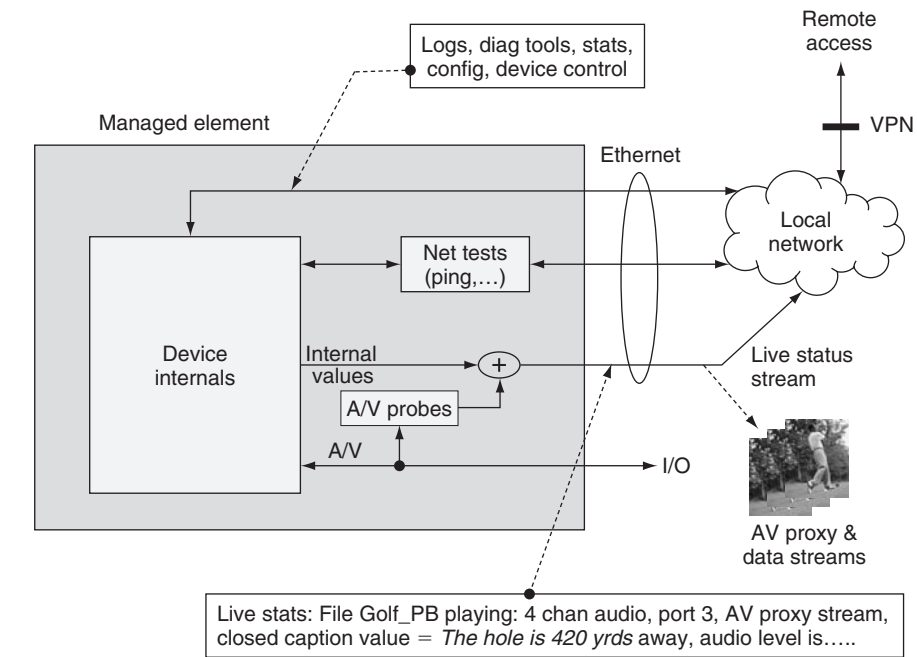
Reporting system events is only one aspect of overall management. Another is diagnosing the root cause of a problem. Some critical application servers can easily generate thousands of events a day, although this would be uncommon in most A/V systems. It is tough to comprehend this massive flow of data. Many of these events are also cryptic and make it difficult to pinpoint specific problems. As a result, some sort of event filtering is needed for big systems. Some vendors of IT management stations offer event correlation tools that make connections between hundreds of events to spot the likely cause.

Another scenario is finding the root cause for an event such as “Video Server application: Failure: Play File ABC: Port 4, 10:24 AM.” System logs need to be scanned and interpreted for clues. At times, diagnostics tools are needed to run tests to uncover the root cause. Vendor-supplied management stations should

offer some rudimentary tool set for debugging problems. Of course, there are no standards for troubleshooting, but some common techniques are available and widely used. The following is a list of general tools that *should* be made easily available on every modern A/V device to aid in troubleshooting:

- Easy access to the venerable Ping command. This polls a remote IP address for life. See Chapter 6 for ideas on simple network testing. Check out PingPlotter for a cool tool that reports on Ping results overtime.
- Log access. When a problem occurs, the first place to turn is the device event log. Applications that log events using WMI (or WBEM) or MIBs should be queried easily. Incidentally, RFC 3164 defines a SysLog format that should be adhered to by device designers or at least modified to suit instead of a totally proprietary log format.
- Remote access to read internal device logs, read/change configuration settings, and perform operational tests. Vendor access using a VPN along with tools such as PCAnywhere or VNC (a multiplatform product, available free) enables many device problems to be debugged and repaired remotely. Remote troubleshooting and device upgrades can cut downtime by orders of magnitude compared to on-site vendor visits.
- Configuration management (see later).
- Live status reporting for A/V gear. Instead of poll to read events (pull method), devices should also support a pushed “live status stream” over IP. The stream (maybe 30 Kbps) could contain select real-time monitored values, including small proxy snapshots of a target A/V I/O port. Think of the data stream as representing a configurable RT monitoring probe into device internals.
- If one were to dream, every I/O port should have the equivalent of an attached A/V probe as discussed in Section 9.2. The more non-intrusive eyes and ears watching and listening for anomalies, the easier it is to find and isolate problems. No real-world devices support this today, but it is a laudable goal.

Figure 9.16 outlines the ideal features of a system element that provides for the full gamut of testing and diagnostics tools in the preceding list. It is meant to be illustrative and would have one Ethernet port, and not three as shown. For the most part, diagnostics tools are a vendor afterthought as a product feature. It is always good to ask any potential vendor what tools are supported. Also, brace up to the fact that remote access (over secure VPN) for a vendor on a service contract is a good way to repair and upgrade equipment without the need for an on-site visit. It saves time and money.



**FIGURE 9.16** *The well-diagnosed A/V element.*

## CAN YOU HEAR ME?



Using the “Ping IP address” command is the simplest way to test a network connection. The command invokes a remote device to respond with “I’m alive.” Ping measures the round trip time to the remote server or device. Following are two examples of Ping in action. The first one shows a ~48 ms roundtrip delay between the sender and a Yahoo! server. The second test attempts to contact a HP server but with no response. In this case, HP decided to disable Ping responses to prevent rogues hammering their servers. Don’t be fooled; the connection path to the end server may be fine, but the end device may be silent on purpose.

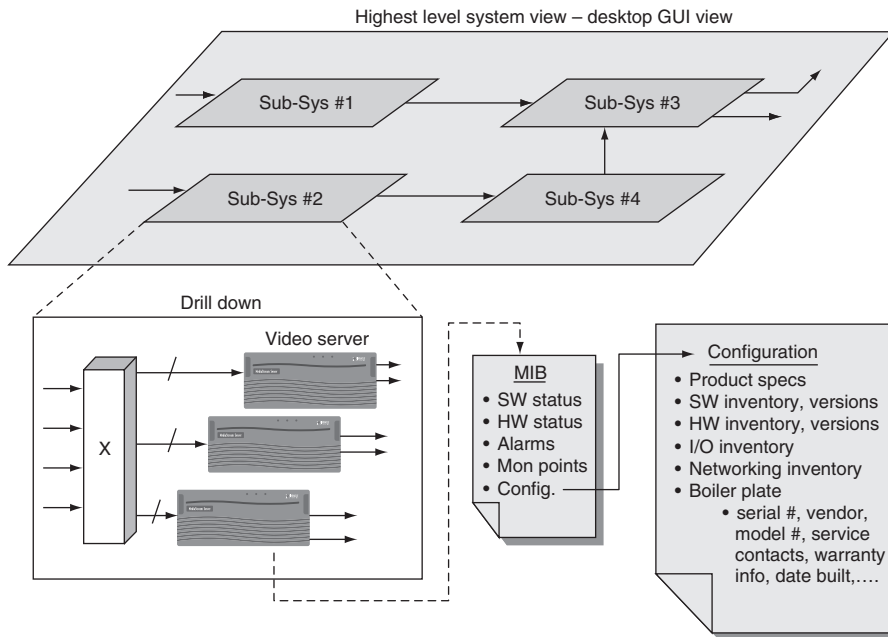
```
C:\Users\ping 209.131.36.158 (www.yahoo.com)
```

```
Reply from 209.131.36.158: bytes=32 time=48 ms
TTL=56
Reply from 209.131.36.158: bytes=32 time=47 ms
TTL=56
Reply from 209.131.36.158: bytes=32 time=49 ms
TTL=56
Reply from 209.131.36.158: bytes=32 time=48 ms
TTL=56
```

```
C:\Users\ping 15.200.30.22 (www.hp.com)
Request timed out
```

### 9.5.1 Configuration Management

Device configuration parameters have been on a long leash for many years and need to be tightened up. Figure 9.17 illustrates a suggestion for MIB-based configuration data structures. While it is true that some MIBs do support



**FIGURE 9.17** *Hierarchical system views.*

configuration data (see Figure 9.5, for example), the coverage is often ad hoc and not complete. The suggested parameters fall into the following areas:

- Product specs—summary of product specs; this is very useful when new personnel are doing troubleshooting
- HW, SW, I/O, and networking inventory along with versioning information
- Boilerplate information, as listed in Figure 9.17

This is offered in the spirit of a unified vendor configuration data structure. True, it is not complete, but with a little vendor cooperation, the details could be nailed down in short order. Some parameters need to be updatable to reflect changes in device SW and HW inventory and versioning. As a result, the update method needs to be obvious and easy to use. Well, that is it for the status quo of device and system management. What does the future portend? Let us see.

## 9.6 FUTURES—DCML

Managing more for less is the CFO's mantra. Only by using standards and leveraging the world of open systems will this happen. Managed IT sets the stage for how A/V should be managed. Progressive vendors see this, and others are following.

On the bleeding edge of managed systems is the work effort by Organization for the Advancement of Structured Information Standards (OASIS,

[www.oasis-open.org](http://www.oasis-open.org)). OASIS<sup>4</sup> is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. The consortium produces more Web service standards than any other organization, along with standards for security, e-business, and standardization efforts in the public sector and for application-specific markets. Its standards encompass much more than device management. At the center of its efforts is the Data Center Markup Language (DCML).

DCML provides the first specification that provides a structured model and encoding to describe, construct, replicate, and recover data center environments and elements. Using DCML, companies have a standard method to enable data center automation, utility computing, and system management solutions.

DCML provides the only open XML-based specification designed to do for the data center what HTML did for content and IP did for networking: achieve interoperability and reduce the need for proprietary approaches. It does this by providing a systematic, vendor-neutral way to describe the data center environment and policies governing the management of the environment.

The methodology is the first standard model to describe both a recipe and a blueprint of the data center environment. As a culinary recipe provides both the list of ingredients and the instructions for successfully combining them, DCML provides both an inventory of data center elements and the desired functional relationship between them. In this way, all of its component relationships, dependencies, configuration, operational policies, and management processes are well documented so that automated processes can take over the load of running and maintaining business processes.

## 9.7 IT'S A WRAP—SOME FINAL WORDS

"If you cannot measure it, then you cannot improve it." This chapter provides an overview of the big picture for monitoring and diagnostics of IT and AV/IT equipment and systems. Use this information to ask providing vendors what management solutions they offer, how they integrate with existing IT monitoring gear, and what standards are supported. You will not likely find the ideal solution because the A/V industry has only recently started providing standards-based management solutions. So, accept the immature status quo of current solutions and support industry efforts to create progress toward the nirvana of 100 percent managed AV/IT systems.

## Reference

Mauro, D., et al. [2001]. *Essential SNMP*. Sebastopol, CA: O'Reilly.

---

<sup>4</sup> Some of the information in this section is paraphrased from the OASIS Web site.

# The Transition to IT: Issues and Case Studies

## CONTENTS

10.0	Issues in the Transition to IT	374
10.0.1	Look Before You Leap	374
10.1	Organizational and Financial	375
10.2	Technical Operations and Support	377
10.2.1	Life Cycle Issues	378
10.2.2	Live Hardware/Software Upgrades and Repair	378
10.2.3	Integration with Legacy Equipment and Systems	378
10.2.4	Monitoring and Systems Management	379
10.2.5	Standards and Interoperability and Vendor Lock-In	379
10.3	Operations, Users, and Workflow	380
10.3.1	Managing Disruptions During the Transition	381
10.3.2	Know Thy Workflow	381
10.4	Case Studies	382
10.4.1	Case Study: KQED, Channel 9, San Francisco	382
10.4.2	Case Study: PBS, NGIS Project	385
10.4.3	Case Study: Turner Entertainment Networks— Centralized Broadcast Operations	389
10.5	Generic AV/IT System Diagram	393
10.6	The IT-Based Video System—Frequently Asked Questions	394
10.7	It's a Wrap—Some Final Words	397



## 10.0 ISSUES IN THE TRANSITION TO IT

The move to IT will have its share of pains and joys. Despite the rosy picture painted in Chapter 1 about the workflow benefits to hybrid A/V + IT infrastructures, there remain practical issues that can be daunting for some. Depending on your starting point and end goals, the road to IT may be fairly simple or dreadfully complex. In the latter case, the IT pill may kill the patient. How can we avoid this sorry end? As with most new adventures, proceed with caution and with eyes wide open. The section that follows outlines the key factors that should be considered before embarking on any infrastructure change.

Following the transition issues will be coverage of three case studies: KQED's digital plant, PBS's NGIS project, and Turner Entertainment's digital facility. Finally, a short FAQ will cover some common questions.

### 10.0.1 Look Before You Leap

Moving to AV/IT likely will not be a knife switch cutover for most users. Consider the following possible transition scenarios:

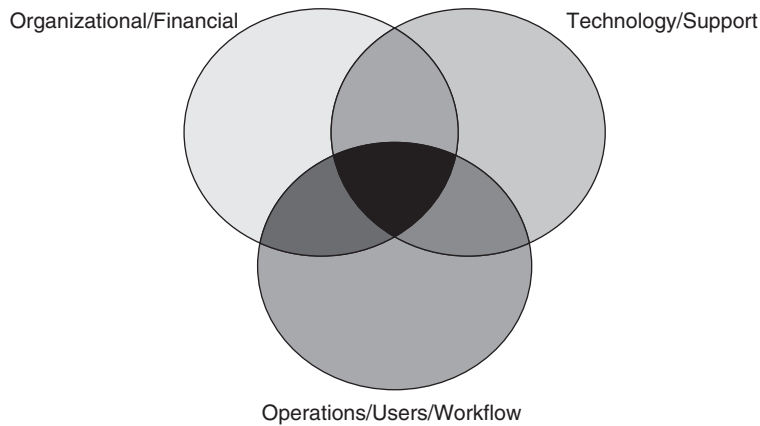
- New facility from the ground up—no or little legacy equipment or formats. This is the green-field case.
- Upgrade of select pieces of traditional A/V with the IT hybrid.
- Wholesale upgrade of existing facility—legacy issues galore.

In all but a few cases, the move to IT should improve media workflows, but will it? This depends on what the end design goals are. Here are some choices to consider:

1. Replace traditional VTR/tape workflow with IT/servers but keep tape-like workflow.
2. Replace portions of old workflows with improved workflows.
3. Perform wholesale upgrade to new workflows.

One error some planners make is to duplicate the old workflow but use new equipment (#1). It is tempting not to change what works. It is tempting to keep all the people and institutional experience in place even though new workflows have major advantages. One case in which the old workflows were completely discarded is the PBS engineering project called NGIS. This is one of our case studies and shows what wonderful benefits can be achieved by implementing #3.

Changing from an old but culturally established workflow to a new one (despite the proven advantages) can be a gut-wrenching experience for the operations and maintenance staff. Just the thought of changing from the comfortable to the new is difficult for many of us. So what factors need to be accounted for? Figure 10.1 shows the three domains that are normally affected by any new infrastructure or major workflow change. The stakeholders from each domain need to



**FIGURE 10.1** *Transition planning: Stakeholders of interest.*

be involved before any migration plan is in place. If not, there will be unhappiness and discord in many parts of your organization as they realize that the changes did not involve them even though they are affected in major ways. Let us see how each of these domain members are stakeholders in the process of change.

## 10.1 ORGANIZATIONAL AND FINANCIAL

When the horizon of change becomes apparent, we often hear reasons why the move is a bad idea. Consider a few classic lines:

- "It is too risky to change now."
- "It will not work in our market."
- "The union will never approve of this."
- "We cannot afford that."

How can an organization reduce the staff's anxiety level? It is understandable that some resistance will occur. Change management is an art. Cultural change does not occur overnight. Some of the steps needed to move forward with a project are as follows:

- Provide education for all those who will be affected—why, when, and how.
- Have a change leader in the organization to evangelize the new workflows and infrastructure.
- Sell the vision in small steps.
- Consider consolidating the media engineering and IT staffs or, at the very least, cross-pollinating these groups. This is a politically charged issue no matter which way it tips.

- Stay focused on short-term goals, but always keep the long-term vision as a guiding light.
- Secure visible support from executive management.
- Do not bite off more than you can chew.
- Make sure the transition strategy includes “what-if” scenario planning.

No project should move forward without a proper justification for the funding. For grand visions, the CEO and CFO will demand to see the return on investment (ROI) and total cost of ownership (TCO) analysis before proceeding. ROI can be a difficult metric to calculate, especially if the new workflows are unfamiliar. For more modest transitions, some of the questions may not be fitting, but they are still good reference points for consideration. Questions of interest follow:

Will it generate new revenue potential, how much, and when?

What are the most compelling reasons to make the change?

Will it lower the TCO compared to current operations? Fewer operational staff, lower maintenance costs, less ongoing capital spending, less floor space, less power usage?

What is the equipment lifetime?

Do we have the skill sets needed to be successful? Should we merge the video engineering and IT departments? Make changes in staff skill set? Redeploy some resources? Is training needed?

What is the initial capital cost for the equipment and installation?

What are the costs to upgrade the building for air conditioning, power, and security to support the new installation?

What is the timeline for the transition? Will it be done in phases? Will there be any disruption in our current operations and delivered product?

What is the overall ROI expected over the lifetime of the installation?

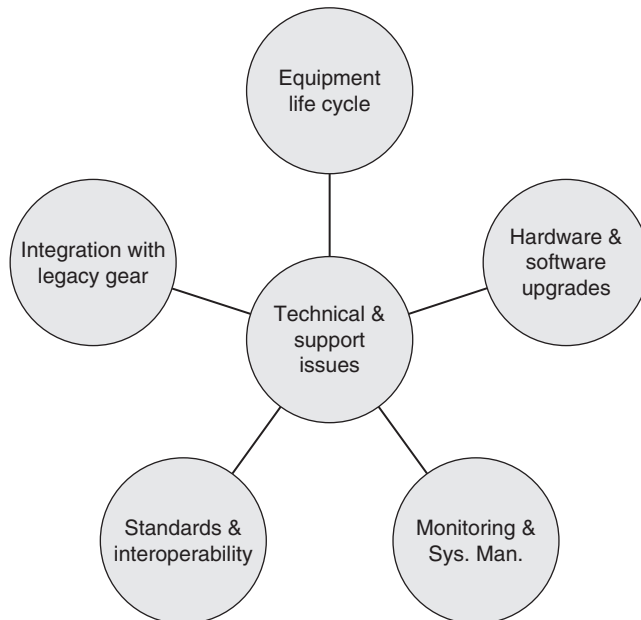
All these factors were important to the participants involved with the case studies considered in this chapter. For the Turner Entertainment and PBS cases, planning and execution were several-year events. The scale of the execution is enormous, as we shall see. It is beyond our scope to detail each of the aforementioned questions. However, if you are planning a transition, having solid answers to these questions is essential for your success.

## 10.2 TECHNICAL OPERATIONS AND SUPPORT

The *second* domain of stakeholders comprises engineering and ongoing technical support. In some ways, the change to AV/IT is the most challenging for this group. Why? Because traditional A/V experts are not always comfortable in the IT world. Also, the IT experts may lack a working A/V knowledge. *IT Media Engineer* is a new title, and few of the staff are fully comfortable taking on its mantle and working in both spheres. Here are some of the classic complaints:

- "IT networks cannot carry video in real time."
- "No one is trained to operate or repair it."
- "We have tried that before and it did not work."
- "The viruses and worms will kill us."

Of course, the eight technical advantages discussed in Chapter 1 are not all peaches and cream. Each has some counterpoint in reality. Every engineering choice is a verdict; "give me this for that" in effect. Very rarely is an engineering course set without weighing the pros and cons. So what are the main technical issues to weigh when contemplating the switch to the AV/IT world? Figure 10.2 outlines the five main topics for this discussion. The list is not exhaustive and assumes that equipment performance, reliability, and scalability are already factored into the system transition plan.



**FIGURE 10.2** System technical and support issues.

### 10.2.1 Life Cycle Issues

Life cycle issues are a red flag for many engineering managers. Some IT components, such as PCs, servers, some software, Ethernet cards, disk drives, and more, have notoriously short life spans. In a way, Moore's law is responsible for this. We might call it Moore's law of obsolescence. This is not just a headache for end users, but for equipment manufacturers too. They must support obsolete gear and stock a lifetime supply (5 years normally) of some spare parts or continue the product in production. Also, manufacturers often have to quickly redesign products due to lack of parts. This is a royal pain. Competent vendors account for this and deflect any obsolescence anguish away from the end user as much as possible. In the end, any purchase decision should factor life cycle into the plans.

### 10.2.2 Live Hardware/Software Upgrades and Repair

In the normal IT environment, components can be upgraded during off hours. Likely, we have all seen the message: "The email server will be down for 30 minutes tonight between 8 and 8:30 for a software upgrade. You will have no email access during that time." We learn to live with and work around the small inconvenience, but imagine a similar message, "The TV station's video server will be down tonight between 8 and 9 to install a software patch." This will not fly. The mantra for many media facilities is 24/7/365.

Knowledgeable vendors have designed much of their equipment to be upgradeable while in use. At first blush, this may seem nearly impossible. However, with a degree of equipment redundancy (to support fault tolerance), this is practical. For example, a video server may be configured as  $N + 1$  (see Chapter 5), and the spare unit may substitute for the component being upgraded. There are many clever methods to upgrade while "live." It is important that you query the vendor on these matters before a purchase decision. Incidentally, for some upgrades, both an A/V and an IT expert may need to be present. Depending on either alone may result in long delays when trouble is encountered, especially during off hours.

Another aspect is the initial transition from the old workflow to the new. How will this transition be managed? What sorts of interruptions will we encounter? What is the most intelligent way to manage this with minimal disruption to our operations? Should we run the new workflow in parallel with the old for a time to test for compliance and equivalence of functionality? Should the new workflow be enabled and tested in phases? For cases in which the system is composed of multiple versions of the same basic idea (e.g., station group implementation for 10 stations), should we implement one first and then follow with the others? These questions should be answered before making the leap to any new workflows.

### 10.2.3 Integration with Legacy Equipment and Systems

When a company or organization is building a new A/V facility, the issue of legacy integration may be a moot point. However, for most transitions, the

move from old to new cannot be done without accounting for legacy systems and media/metadata formats. Consider the case of CNN's digital feeds and edit system. In 2003 CNN initiated a phased installation to an AV/IT-based news production architecture from a tape-based one. As part of its New York City facilities, CNN installed 29 SD ingest encoders for recording news feeds from the field and elsewhere, 20 SD playout decoders, 15 professional NLEs, and 50 proxy editor/browsers. Media clients attach to ~950 hr of fault-tolerant, mirrored A/V storage. Most interconnectivity is based on Ethernet, mirrored IP switches, and some Fibre Channel. A similar system is also installed at CNN's Atlanta headquarters. These are not green-field installations, and on-air news production needed to continue as migration to the new system was phased in. Of course, the CNN integration is not typical of most transition scenarios due to its size and scope.

#### 10.2.4 Monitoring and Systems Management

Every system, whether traditional A/V or hybrid IT, needs some sort of fault, warning, and diagnostic reporting provision. The IT world offers a standardized, mature, and encompassing solution set as shown in Chapter 9. Compared to standard A/V gear reporting, IT systems management is very sophisticated and holistic. Most vendors provide their own user interface for access to the most common management operations.

#### 10.2.5 Standards and Interoperability and Vendor Lock-In

Most AV/IT systems are composed of different vendors' equipment in a tightly integrated configuration. Ideally, the end user has the choice to substitute one system element for another. However, regarding hybrid AV/IT systems, most providers require that no third-party "substitutes" [commercial off the shelf (COTS)] be used. For example, a data server element in a live-to-air environment may require very special operational specs to meet the needs of live failover or an element (Fibre Channel switch, IP router, etc.) is located in a critical data flow and requires guaranteed throughput. For most standard IT applications, a little slower or faster element may not impact workflow at all, whereas for AV/IT the element specs are crucial for operations. A two video frame (~65 ms) equivalent delivery delay in an email application will have zero impact on the end user. Compare that to the same delay for a critical A/V application where a lip-sync problem manifests itself. Even content-neutral elements such as IP routers and switches can have a major impact on system performance under corner case loading and failover scenarios. As a result, while the end user may be tempted to substitute COTS equipment in a configuration, the reality of guaranteed performance under all operating modes makes this a risky decision.

Product support is another aspect of using COTS in an end-user configuration. A vendor's system design is normally tested and validated with known

equipment. Each element may have specific error and status reporting methods that the system provider is depending on. The overall system status and health are strongly dependent on knowing the exact nature of each element. If a COTS element is substituted for a vendor-provided element, the vendor support contract may be breached. After all, configuration documentation is the heart of any big system, and changing components to suit the needs of the end user (normally to save money or repair time) will make support a nightmare.

All similarly specified IP routers are substitutable, right? Well, for some applications this may be true, but the performance and operational characteristics are never exactly equivalent between, say, a Cisco router and one from Foundry. Configuration methods are often completely different, internal components are rarely the same (power supply diagnostics will be different), and so on. Any substitute of a mission-critical system component will only cause long-term grief to the end user and the providing vendor.

With that preamble, it is not difficult to see that some vendor lock-in is inevitable. However, this comes with the upside of guarantees in performance and support. No end user wants to be completely locked into a vendor's products. That is why standards and interoperability are still of value. After all, most systems need to import files from external sources, so the file formats should be "open." Also, metadata need to be open in terms of access and formats. When evaluating a vendor's solution, make certain that the environment is open for file import and export. Because many systems have control points (record input A/V, playback, and so on), associated APIs and protocols should be well documented and open for third-party access. Ask many questions of any provider so you know which system domains are open and closed. Only then can you make intelligent buying decisions.

### 10.3 OPERATIONS, USERS, AND WORKFLOW

The *third* category of interest is where the rubber meets the road: the user experience. The end user usually gets the ear of the engineering and management staff when selecting new gear. Many purchase decisions are a direct result of what the operations staff demand. Changing workflows or the operational experience is never easy, and end users are often reluctant to venture into uncharted waters. Some operators (especially A/V editors) are comfortable with their favorite interface and can be intransigent when it comes to trying something new. This behavior comes from several motivations:

- "If it is not broken, do not fix it."
- "We do not know if the new way will work."
- "I like vendor XYZ's user interface, so do not ask me to learn a new one."
- "Changing products now will slow us down—too much to learn. If you want it done today, then let me use what I already know."

For sure, these are valid comments, but in the context of the greater good, they are inhibitors to more efficient workflows and all that IT promises to deliver. Let us look at some of the factors that may impede the adoption to AV/IT systems.

### 10.3.1 Managing Disruptions During the Transition

Managing disruptions is a crucial aspect of the move to any new workflow. Planning needs to be done to minimize any potential reduction in the quality and amount of output. Using a mirror training system is one way to reduce staff anxiety and opportunity for error. In the case of Turner Entertainment, they built a new facility and ran a new channel in parallel before decommissioning the old channel. This gave the staff some time to learn the new workflows and iron out the bugs. Of course, this was a challenge since the staff needed to operate both old and new simultaneously, but this proved a wise choice in the end.

Training is a key to smooth transitions. This may come from vendor-provided classes or in-house mentoring. As with most of us, once we are comfortable with a new process, then we are eager to spread our wings. Without proper training, we should expect the worst—problems at every turn. The poet Longfellow said, “A single conversation across the table with a wise man is better than ten years’ mere study of books.” How true. So get educated by attending industry events such as SMPTE Technical Conferences and Networkworld + Interop conferences. Meet the people driving AV/IT forward. Attending conference-related tutorials is a great way to learn. Subscribe to several A/V and IT industry trade magazines. Subscribe to free Web-based A/V and IT newsletters. Attend local A/V and IT industry events.

### 10.3.2 Know Thy Workflow

During the course of developing this chapter, I interviewed several facility engineers and end users. When contemplating the change to a new workflow, they acknowledged that there was the inevitable step of reviewing the documentation of the *existing* workflows. “Where is the documentation for our workflows?” was the refrain. Often, no one answered back. As it turns out, existing workflows were part of the culture of doing the job and not some well-thought-out and documented procedure. This was an eye opener to some of those about to embark on improving the old workflow, since they did not have a good handle on what they wanted to improve! This is a dangerous starting position for a new venture. Of course, every case is different, and each transition plan eventually comprehends the status quo to the extent needed. The lesson was clear to all: you cannot improve what you do not understand. If you are thinking of changing workflows, take the time to document what you have so the transition to new ones will not have as many surprises.



Chapter 7 reviews the fundamentals behind all A/V workflows. This is a high-level discussion but will give you some food for thought as you plan new workflows.

## 10.4 CASE STUDIES

The three short case studies to follow investigate the workflows, challenges, and successes of each new design. In all cases, the existing media enterprise planned and implemented (or using a systems integrator) the transition from traditional A/V to an AV/IT system. In one case the transition is a work in progress, whereas for others the job is done. For each case study, stakeholders were interviewed for their take on the success and issues with the undertaking. These three are representative of many similar facilities worldwide. One is a typical public TV station, one is a TV network facility, and one is a large multichannel broadcaster. The coverage is brief, and you may not find a direct parallel to a project you have in mind. Nonetheless, the general methods and lessons from these studies are applicable to just about any size AV/IT project. The following case studies are covered:

- **KQED San Francisco**—A flagship station of public broadcasting and a provider of PBS programming. It is one of 349 member stations serving commercial-free programming. In 2003, KQED replaced its tape-based master control chain with an IT-based one.
- **PBS**—The U.S.-based Public Broadcasting Service is executing a long-term plan to convert its member station A/V stream feeds over satellite to file transfers. Migrating from traditional video feeds to file transfer yields great economies. This transition is ongoing in 2009.
- **Turner Entertainment Networks**—In September 2003 TEN rebuilt its entire broadcast operations center in Atlanta. As one of the premier broadcasters worldwide, TEN has developed an IT-based, distributed operations center for 22+ channels.

### 10.4.1 Case Study: KQED, Channel 9, San Francisco

KQED<sup>1</sup> is a major public broadcaster and PBS affiliate. In 2003, on-air operations were based mainly on tapes and VTRs for time-delaying PBS programming. Yes, the station has some live local programming, but the lion's share of on-air content comes from stored materials. Most delays were anywhere from a few hours to a few years. The workflow is not identical to a commercial station, but there are many similarities.

---

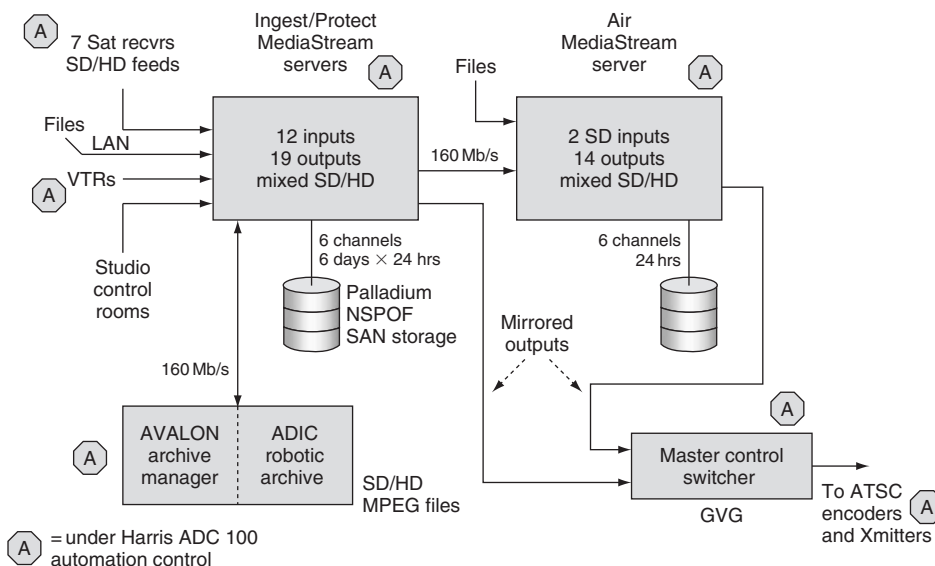
<sup>1</sup> This case study is based on information provided by Larry Reid, Sr., Director/Chief Broadcast Technology Officer of KQED.

KQED replaced most of its manual videotape chain with an automation system, a video server system, and a data tape archive. Figure 10.3 shows the major components in the system using IT-based control and stored program content. The legacy system had all the classic inefficiencies inherent in a tape-based system:

- Linear, non-networked material
- Tape-based, VTR maintenance headaches
- Little automation, no HD chain
- No support for upgrading to multichannel

A comprehensive plan for a new infrastructure should do more than eliminate the pains of the old one. In addition to moving beyond videotape, some of the goals of the new infrastructure were as follows:

- Moving from 20 hr per day to broadcasting 100 hr (six channels) without increasing staff.
- Maximizing the use of automation—ingest, file movement, storage management, playout, and switching.
- Providing support for HD along the entire chain.
- Providing support for eight audio channels per video channel.
- Addressing file transfer needs. Support for ingesting files (e.g., from PBS), archiving files, and playing out files. PBS file ingest is a future need, so the new system also supports recording and playing of traditional A/V streams.



**FIGURE 10.3** KQED digital on-air system.

### 10.4.1.1 The Configuration

Figure 10.3 shows the chief components of the new system. Central to all operations are two video servers using Avid's MediaStream. Each is attached to separate SAN storage. Two servers provide protection in the event of one failing. One is used primarily as an ingest server (it has 31 I/O), and the other as a playout air server (16 I/O). In the event of one malfunctioning, the other one can take over. Material comes in from satellite or tape and is recorded automatically into the ingest server. Once it is ingested, trimmed, and QA'd by personnel, it is pushed to archive and to air server if it airs within 24 hr.

The ADIC Scalar 1000 archive system is a major system component. It has a capacity of 1,000 tapes. AIT-3 tapes are used with a capacity of 100GB each, so each tape supports 18.5 hr of MPEG2 at 12 Mbps (SD) per tape for a total of 18,500 hr for the archive. A total of 12,000 hours of mixed SD/HD is a practical limit. HD is stored at 19.3 Mbps and some at 45 Mbps, so this also reduces total storage hours. When AIT-4 tapes are used, the capacity will double. EMC's AVALONidm storage management software automatically and transparently manages files in the ADIC tape system, optimizing the placement of data to match service levels.

Most of the stored materials are received from PBS. KQED has the rights to rebroadcast a show four times over a 3-year period for most of the programs. The automation is provided by Harris Automation, ADC100. It manages all A/V movement. Six days in advance of air time, the traffic department gives automation a list of programs for all channels. The Harris ADC100 locates the programs in the archive and moves them to the ingest server. Twenty-four hours before air time, they get copied to the air server from the ingest server's storage. Six days' notice is required for the "pull list"—that gives plenty of time to resolve any material location problems. The day before air time, interstitials are pulled from the archive and loaded into the ingest/protect and air servers. At air time, both servers play out in parallel all channel programming with fault tolerance and peace of mind.

When PBS initiates program delivery via file transfer, KQED will be ready for this new form of ingest. PBS has yet to define the exact A/V file format. It will likely be a constrained MXF file with MPEG2 essence. The Advanced Media Workflow Association (AMWA) is developing a best practice document to describe the MXF file constraints for program distribution. This is ongoing work in 2009. Operations are running smoothly, and the engineering staff and operators are gaining confidence in the new workflow. National TeleConsultants was the systems integrator for the project.

Of course, not every aspect of the migration went smoothly. KQED had issues with device configuration and establishing stability of some software elements. Some vendors overpromised their deliverables. The learning curve from traditional manual to full automation operations took time to negotiate.

### 10.4.2 Case Study: PBS, NGIS Project

PBS<sup>2</sup> is the main public broadcaster in the United States with 170 non-commercial, educational licensees operating 349 PBS member stations.

Headquartered in Alexandria, Virginia, PBS broadcasts several programming genres, each on a different channel, which in turn are consumed by the member stations. For many years, program distribution to member stations has been via satellite A/V streaming. Local stations recorded the feeds and scheduled playback based on their local needs, as discussed in the KQED case given earlier.

PBS is engaging the Next Generation Interconnect System (NGIS) project, which will completely revamp how it receives (from content providers) and distributes programming to member stations. This case study outlines the issues with the old workflow and the motivations and plans to upgrade to a file-based workflow nationwide. NGIS is a work in progress, but the ideas are instructive for review even though it is not complete. To put the project into perspective, Figure 10.4 illustrates media flow among providers, PBS, and member stations. NGIS targets the programming flow in and out of PBS.

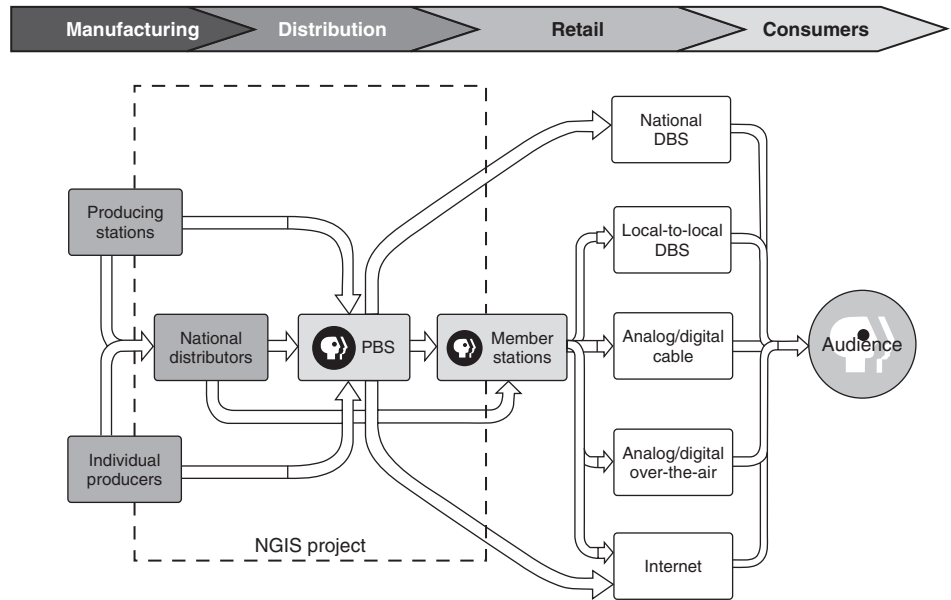
#### 10.4.2.1 Motivations to Move from the Status Quo

Let us start off by reviewing the current roadblocks to smooth program exchange between entities in Figure 10.4. The following list breaks the issues into three camps: programming inputs to PBS, PBS output streams, and member station operations.

1. Programming produced by outside sources:
  - All incoming content (files) enters a QA process.
  - Corrective measures are applied as needed.
  - Program and segment timings are checked and metadata are updated as necessary.
2. Real-time streaming distribution to member stations:
  - No metadata are included with programs. All program-related metadata are sent out-of-band—errors are common.
  - Programs are fed multiple times to cover U.S. time zones, thus wasted bandwidth.
  - Satellite rain fades impact end users.

---

<sup>2</sup> Much of this original material is based on a presentation (and private discussions) given at the NYC SMPTE Technical Conference on November 12, 2003, by Thomas Edwards, then Senior Manager, Interconnection Engineering, PBS. Thanks also to Jerry Butler for updates in 2008.



**FIGURE 10.4** *The PBS supply chain.*

### 3. Station side:

- 179 stations do basically the same thing:

They use direct-to-air transfer or a tape/server delay process.

Many stations record earlier time feeds to avoid a possible future rain fade during the correct playout time.

- As stations move to video servers, the feed recording/playout workflow is not efficient.

Table 10.1 outlines advantages to moving to a file-based ingest/playout workflow and away from tape ingest and A/V stream feeds to member stations. NGIS is aimed at removing the current inefficiencies and adding new performance features along with future proofing the design.

The last entry in Table 10.1 enables a big savings in satellite use and overall distribution bandwidth. By sending files instead of streams, NGIS increases the overall transmission efficiency by about a factor of six. Fortunately, because PBS member stations do not broadcast network programs live for the most part, transferring program files (especially HD) to stations using non-real-time delivery saves considerable satellite bandwidth and is a great way to cut costs.

#### 10.4.2.2 The NGIS System Outline

The following list outlines the salient features and technical advantages of the NGIS:

- All SD and HD files are compressed MPEG (or other) wrapped in MXF.
- Most content is distributed as files using IP over satellite.

**Table 10.1** Comparative Workflow Advantages of NGIS

Characteristic	Status Quo	With NGIS
Meets the needs of program distribution	▲	▲▲
Provides support for DTV and enhanced interactive services	▲	▲▲
Provides peer-to-peer connectivity		▲
Provides flexible workflows		▲
Optimizes the distribution resources (better satellite BW utilization, lower costs, other)		▲

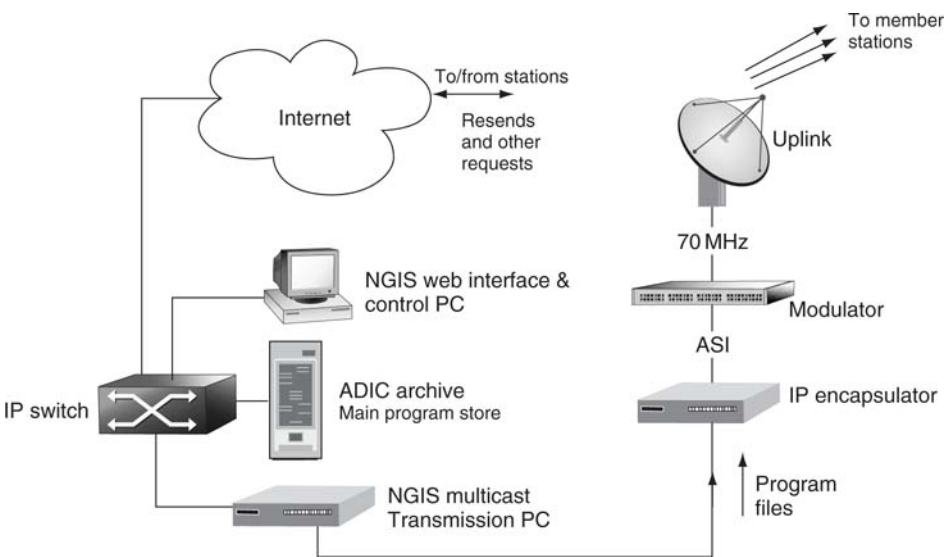
▲ Meets needs; ▲▲ Improved performance

- Content is sent once, providing accurate content distribution.
- Intermittent transmission outages do not impact file delivery.
- Operations are automated as much as possible.
- Received content is temporarily cached on an edge server at the station side.
- The number of transponders and costs is reduced.
- Missing packets are requested and re-sent using Internet connection.

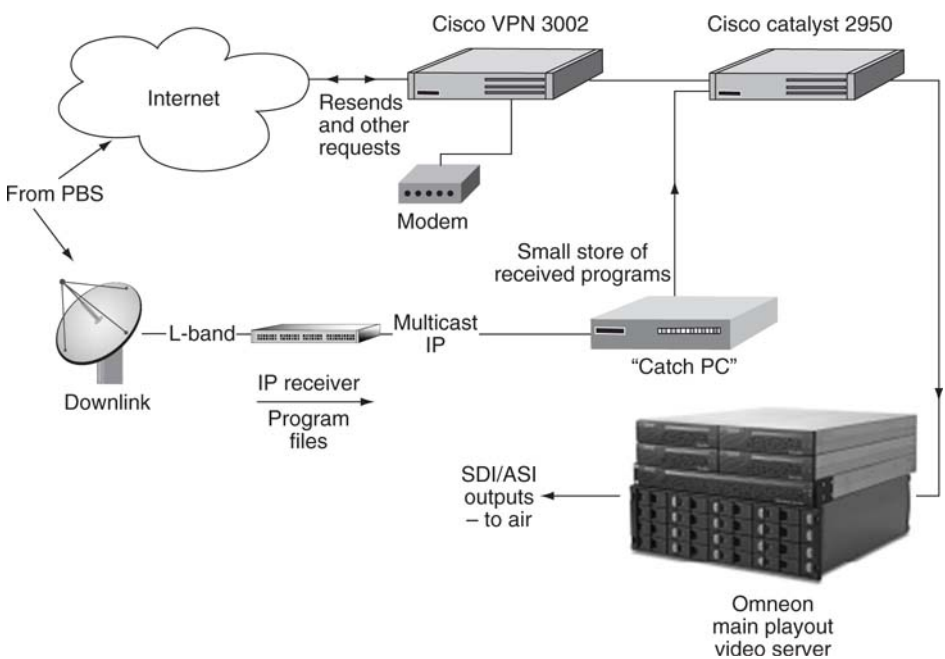
Sending files from point to multipoint over satellite is non-trivial. One missing bit can ruin nearly a second of MPEG video, so any loss is unacceptable. NGIS uses two strategies to guarantee 100 percent file delivery integrity. The first is a very robust forward error correction (FEC) means to correct up to 20 min of lost data! This requires sophisticated coding but is achievable. It is a necessity during a heavy rain fade. A 20 min loss at 80 Mbps (transponder rate) is equal to 12GB of recoverable data. Very impressive. See Chapter 2 for more information in FECs and error correction strategies.

A second strategy is to automatically ask for a retransmission of corrupt packets if the FEC cannot recover the errors. The backchannel request must be used sparingly; otherwise, retransmissions will kill the file transfer advantages. For example, a FEC that can recover only 0.1 s of data will force many retransmissions, and the satellite bandwidth usage will be worse than with legacy A/V streaming.

Figure 10.5 shows a NGIS test configuration for the PBS side of the equation. At a scheduled time, automation retrieves files from the ADIC archive, prepares them for transmission, and feeds them to member stations over satellite. Note the Internet connection. It is required for retransmission requests and for member station requests for nonscheduled programming materials. On the station side (Figure 10.6), IP packets are received by the “catch PC.” It applies the FEC to correct file data errors, asks for retransmission if needed, and optionally reformats the file for the member station legacy video server. Most stations already have video servers, supplied from the usual suspects, and it is not practical for PBS to send files formatted for each server type; that would waste bandwidth. As a result, the catch PC optionally reformats (lossless



**FIGURE 10.5** *Prototype NGIS PBS-side hardware.*



normally) the incoming PBS MXF file into a format usable by the installed legacy server. The prototype shows an Omneon Spectrum server. For the final rollout, various models will be used, depending on legacy installations on a per-station basis (see the KQED case given earlier).

NGIS also defines format requirements for vendors supplying programming to PBS. When WGBH produces a new *Frontline* episode for PBS, it would format the program within the NGIS guidelines for metadata, closed caption, MPEG format specs, and so on. Any errors inherent during PBS ingest may get propagated to all member stations. Hence, NGIS is keen to define a standard ingest file format and improve the quality of the total distribution process.

Upon completion, NGIS promises to be a very efficient and practical way for PBS to receive and distribute programming. NGIS sets a model for how syndicated programming, news, and commercials may be distributed to commercial stations. Of course, there is activity in this area. See, for example, the services and products from Bitcentral ([www.bitcentral.com](http://www.bitcentral.com)) and Pathfire ([www.pathfire.com](http://www.pathfire.com)) that are in general use and similar to the overall themes of NGIS.

Making the transition to NGIS is filled with challenges. Some of the issues needing to be resolved are the following:

- Member stations are currently using video servers from a host of vendors. The master PBS file format will likely be MXF with MPEG2 essence. However, because this format may not be compatible with all installed station-side video servers, file translation will be implemented as needed.
- PBS will package metadata with each program file. Local stations may decide to use these data to improve their operational workflow.
- PBS needs to coordinate with the 170 member stations so they have the proper file receive infrastructure, as discussed earlier. These stations need to start thinking files, not streams, as the new workflow.
- Funding from the Corporation of Public Broadcasting (CPB) is needed to complete NGIS. Writing proposals and working with the budget process are time-consuming and filled with iterations.

### 10.4.3 Case Study: Turner Entertainment Networks—Centralized Broadcast Operations

In January 1970, Ted Turner purchased UHF channel 17 (WJRJ) in Atlanta and renamed it WTCG. From that humble beginning, TEN<sup>3</sup> now operates 22 network feeds (plus time zone feeds), including three of the top five cable networks in the United States—downtime is not an option. Channels are fed to satellite and cable distributors in North and South America. In September 2003, TEN moved all broadcast operation to a new 198,000 square foot building (see Figure 10.7).

---

<sup>3</sup> The materials for this case study were provided by Clyde Smith, Senior VP, Broadcasting Engineering R&D, QA and Metrics of the Turner Broadcasting System; Naveed Aslam, Senior Director, Broadcast Technology and Engineering; Jack Gary, Director Projects and Integration Engineering; and Rick Ackermans, Director of Engineering, Turner Broadcasting Systems Network Operations.





**FIGURE 10.7** Broadcast operations building—part of the TEN campus in Atlanta.  
Image courtesy of TEN.

Migration from a videotape-based to an IT and file-based architecture was likely the biggest broadcast-related project of its kind at the time.

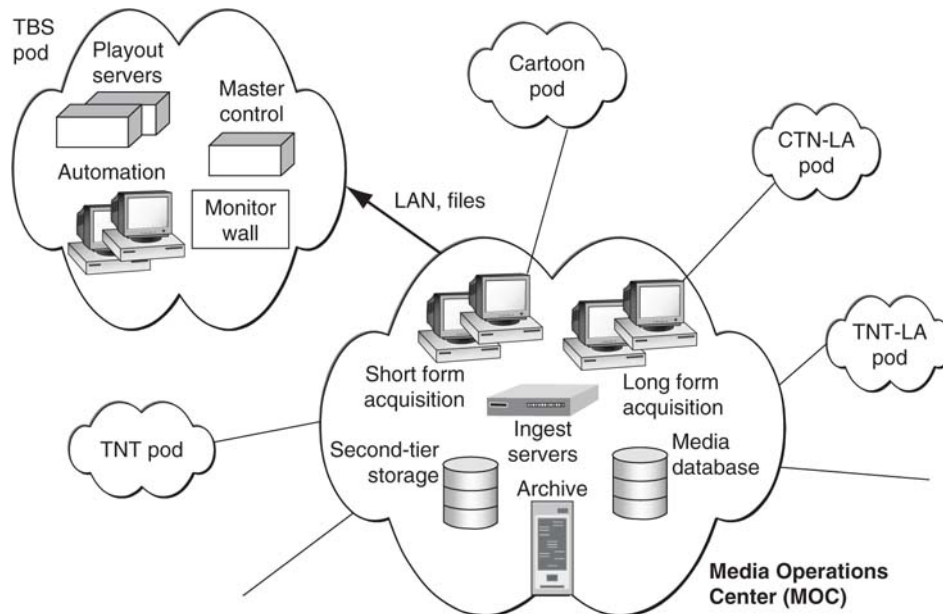
The following list enumerates the motivations behind the project.

- Provide for anticipated network growth
- Improve control and management of Turner physical and intellectual property assets
- Reduce current operational compromises and meet the growing demands of the Turner Entertainment Group
- Provide a platform for implementation of new media-based business initiatives
- Consolidate technical resources
- Implement a total digital infrastructure with fiber optic connectivity for file exchange
- Facilitate the implementation of HD digital TV
- Facilitate the migration from a videotape-based storage and workflow to a digital asset-based environment
- Enable upgrade of the traffic system and other key broadcast-specific operating systems

Now, that is a tall order. Figure 10.8 illustrates a high-level view of TEN's broadcast inventory management (BIM) system. At its heart is the media operations center (MOC). This portion is responsible for ingesting A/V materials from tape, utilizing six short form bays, four long form bays, and 20 cache engines. Tape ingest formats include IMX, D2, and SRW (Sony HD) tape formats. Ingest video servers are used to temporarily cache newly ingested commercial and promotional content. Next, the cached materials are transferred as files into EMC near-line storage arrays and the Asaca AM 1450 DVD (Blu-ray disc) backup libraries. Pro-Bel Automation's Sextant software moves the programming files via the AVALONidm (storage management software) from the EMC CLARiON FC4700 arrays to the playout pods as determined by channel scheduling needs. Program content is cached directly to the air playout servers.

Each pod is designed with parallel playout chain redundancy. Each channel chain (A and B) has separate automation control. Video servers (mirror of content) synchronously play out the channels under Pro-Bel control. GVG MC2100 master control switchers are used. There are 49 automation/air chains in all, with a total of 200+ associated computers.

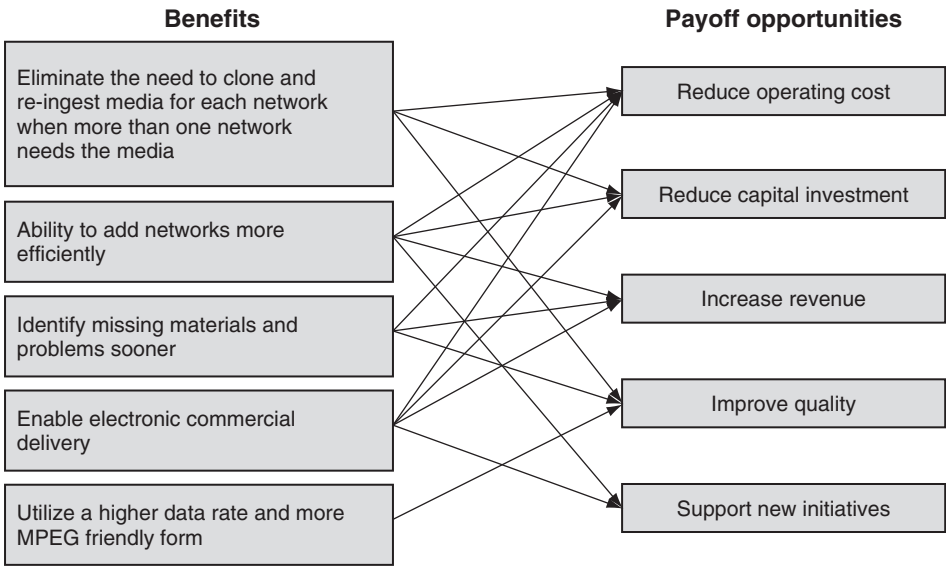
The Cartoon Network has two dedicated StorageTek Archives (Powderhorn 9310) for storing 6,000 cartoons. Total capacity for the archive is 240TB. While all other channels have near-line storage (with backup, not archive), Cartoon relies on deep archive due to the frequency of playback and total storage needed. Figure 10.9 shows a composite view of the 500-rack equipment room.



**FIGURE 10.8** Broadcast inventory management system.  
Image courtesy of TEN.



**FIGURE 10.9** Composite view of TEN's 500 racks of equipment.  
*Image courtesy of TEN.*



**FIGURE 10.10** Benefits and payoffs of the BIM project.  
*Image courtesy of TEN.*

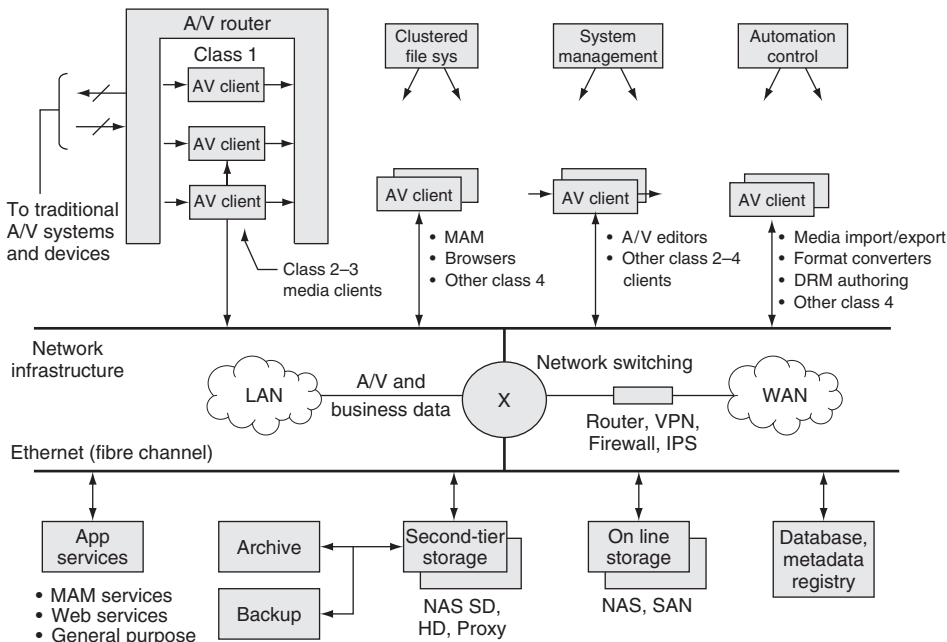
Figure 10.10 outlines the benefits and payoffs for the BIM project. The five payoffs on the right side are of universal value to any commercial media operation. It is obvious that migrating to IT has big advantages for TEN. Of course, there were challenges along the way. A few of the biggest obstacles were as follows:

- Building one of the first SMPTE 292M (serial HD link)-compliant facilities required unique installation methods to preserve the long-term integrity of the cabling.

- Developing the BIM system, which is not an off-the-shelf solution, required the cooperation of five major vendors in order to complete the system successfully.
- Painsstaking analysis of current workflows was required in order to best understand what BIM should provide.
- Work with many vendors over 12 to 24 months was required to develop new products in order to meet the vision of what the file-based, HD-compliant infrastructure needed to deliver.

## 10.5 GENERIC AV/IT SYSTEM DIAGRAM

In the final analysis, the three case studies and many others not outlined here are patterned after the generic converged AV/IT video system shown in Figure 10.11. Figure 10.11 illustrates the major architectural themes discussed in this book. By rearranging, factoring, scaling, and duplicating elements, a business or an organization can implement any AV/IT system, including the three case studies. For sure, this description is overly simplified, but it represents the new paradigm for IT-based video systems. Note the use of the four classes of media



**FIGURE 10.11** *The generic converged AV/IT video system.*

clients, as discussed in Chapter 2. Also note the use of three levels of storage including SAN/NAS, as outlined in Chapter 3. Prominent, too, is the use of IP networking, as discussed in Chapter 6. Ideally, all elements are tied into centralized systems management. Concepts such as media formats, protocols, security, software architectures, and reliability are implied but not distinctly shown. Of course, there is not a single central IP router; the one in Figure 10.11 represents an aggregate of network switches, including redundancy methods.

Figure 10.11 is useful as a touchstone when developing converged AV/IT systems. It is a reminder of the chief elements required for many designs.

Well, that concludes the three case studies. If space would permit, many more from Europe, Asia, and the United States could be reported on. Just about every new broadcast and professional installation has a large component of IT spread throughout. The FAQ to follow concludes the discussion about AV/IT systems. Chapter 11 is an overview of traditional A/V technology. The contents are a flash course in digital video basics.

## 10.6 THE IT-BASED VIDEO SYSTEM—FREQUENTLY ASKED QUESTIONS

Thinking of migrating to a converged AV/IT system? If so, the following FAQ will be of value to you. It covers select issues commonly encountered by facility planners. The FAQ is brief; likely, it will not answer all of your questions. However, it should give you a springboard to start your planning process.

**Q1:** What is a practical plan to convert to an IT-based A/V workflow?

**A:** Do not boil the ocean. Convert one workflow at a time. Start with ingest or playout or archive or editing or graphics. If you are starting from scratch, you may be able to do a complete IT design. For most designs, the move to IT will be incremental and must integrate with a legacy system. Remember that file-based technology and workflows are enabled using hybrid A/V + IT systems components.

**Q2:** Is there is downside to the migration to IT-based video?

**A:** Sure. Creating video systems using IT is not a panacea. There are hidden costs, operational issues, and implantation issues. Some of those issues are staff education, interop falling short of vendor promises, too many standards to track, proprietary aspects, the need to maintain the desired network QoS, the need to keep it secure, software upgrades, IT gear obsolescence, required software patches, unified element management, and consistent metadata and format use. Despite these issues, each can be overcome by applying the principles outlined in the book's chapters. Proof, too, is demonstrated by many successful AV/IT installations worldwide.

**Q3:** How should a small TV station or other A/V facility approach the move to IT?

**A:** Work with trusted vendors and system integrators who have a proven track record in offering and implementing IT-based systems. Educate yourself and the staff. Visit facilities that have made the move and interview the stakeholders. Make decisions based on the expertise of the IT and A/V staff; do not go it alone.

**Q4:** Where should I expect the most gains from the switch to IT?

**A:** Networked efficiencies and reach, content tracking and usage, less pure video to test and calibrate, flexible workflows, speed of content access, virtual facilities not bounded by distance, no videotape handling, the ability to ride the IT wave to lower costs, and higher performance. Remember not to duplicate a tape workflow with IT equipment but to leverage file transfer delivery times to save money and relax the demands on equipment performance and reliability.

**Q5:** What aspects of IT-based media client video flow will dominate: file transfer, IP streaming, or RT direct-to-storage access?

**A:** Based on current trends in facility design, file transfer is most common, followed by direct-to-storage access, followed by streaming. Low bit rate proxy video will be streamed to client stations. Your mileage may vary based on workflow needs.

**Q6:** What standards should we give special attention to?

**A:** For A/V file interop, focus on MXF and MPEG/DV; for storage access, CIFS, NFS, and iSCSI; for metadata, MXF, XML, SMEF, SMPTE, and EBU recommendations; for networking, IETF and IEEE; for editing, AAF and, of course, SDI and countless SMPTE standards for traditional A/V interfacing.

**Q7:** How should we approach asset management?

**A:** There is no simple answer. It depends on size of the installation, workflow needs, amount of programming, material repurposing plans, and other factors. Spend time to get this right, as it is the key to many workflow efficiencies. Again, learn from others who have already made the move. Do not try to boil the ocean.

**Q8:** In what format should we archive our A/V materials?

**A:** Your solution depends on several factors, so there are several answers. If you ingest materials from digital cameras, save in the camera's native

compressed format. If you ingest from legacy videotape or live video feeds, select a facility-wide encoding compression format. Choices range from DV 25/50/100, SMPTE's VC-3 HD format, and one of the many MPEG2 or MPEG4 formats at bit rates and line structures (HD, SD) from 10 to 450 Mbps or fully uncompressed video for very high-end work. Audio formats range from uncompressed 16/20/24 bits, Dolby-E, or a variety of other formats. Archive choice also depends on material repurposing plans. For SD/HD play-to-air only applications, lower video bit rates suffice. As MXF becomes mature, many facilities will choose to work and archive with it as a wrapper format.

**Q9:** What are the key AV/IT design parameters?

**A:** Give attention to the **Top 10**.

1. Functionality of the total workflow (includes MAM)
2. Interop with legacy systems and intersecting workflows
3. Essence and metadata formats
4. Element and system reliability (SPOF, NSPOF, MTTR)
5. System scalability (number of clients, storage, network size, bandwidth)
6. Network topology and QoS
7. Software architectures (Web services, middleware, Java EE, .NET, standalone)
8. Storage requirements (online, near-line, and archive, hours, QoS)
9. Security across all system elements
10. Element management (system management techniques)

**Q10:** We are thinking of developing new IT-based workflows for our A/V needs. How should we start?

**A:** Know thy current workflow first. Spend time to document how you do things now before you design replacement workflows. Time spent up front will enable you to design better workflows later. Make sure any providing vendors agree to your system requirements.

**Q11:** How should we qualify a new AV/IT installation?

**A:** Develop acceptance test criteria. Before signing the check for a completed systems installation, make sure it all works according to plan. Develop a list of test actions. A target list may contain items such as glitch-free A/V recording for 5 hours, glitch-free playout for 5 hours (or a lot more), file transfer speeds, storage access rates, component failure responses, device failover responses, security audit, media client responsiveness, power cycle responses, alarm reporting, MAM functionality,

and many more. The bottom line is: make sure you are buying a useful system that is not filled with holes.

**Q12:** Any parting words?

**A:** AV/IT is still maturing. Some workflows are mature and ready for prime time (news story production, ingest, playout, proxy browsing, cataloging, graphics playout, and production). Others are taking baby steps as with live production. Understand your workflow and map to stable technology. Do not bet on the latest unproven concepts for mission-critical applications. Traditional A/V technology is not dead. The fat lady is not singing, but she is practicing her scales.

**Q13:** What do you see in the crystal ball for technology?

**A:** Oh boy. Well, looking near the surface, IT costs will continue to decline, and element/link performance will continue to increase. That is an easy one. Traditional A/V will find applications for years, but AV/IT will continue to move into broadcast and other professional applications. Storage will support 1,000 “viewing quality” HD movies on a single 7TB disc drive by 2014. Open systems will take hold, and A/V vendors will offer (begrudgingly, for all but the pioneers) media clients to work with third-party online COTS storage for mission-critical applications. Green thinking and data center virtualization will drive purchasing decisions. Expect more and more software SD/HD processing with less and less video-specific hardware. Videotape will find shelf space at the Smithsonian. Looking deeper—now this is amazing—wait, it is getting cloudy. We will just have to wait and see.

**Q14:** What do you see in the crystal ball for A/V systems design?

**A:** Well, open systems for sure. Expect more use of Web services possibly coupled with SOA strategies. Expect to see rich media clients of all types, workflow management tools, more MAM, lots of creative tools for graphics, video and audio editing, and authoring. File transfer will rule the day, with A/V streaming taking a back seat. Live event production will continue to use traditional A/V methods for many years. It takes a pioneer to change the status quo of event production, and there is not one on the horizon. A/V devices will be managed using IT methods—at last. Oh yeah, and of course, IP- and Web-based everything.

## 10.7 IT'S A WRAP—SOME FINAL WORDS

Well, that is it—insights from the real world on the convergence of A/V and IT. Yes, the AV/IT train is moving down the track at high speed propelled by



Moore's law and the eight forces outlined in Chapter 1. The future looks bright for A/V system design, and it will be a fun ride over the next few years as the remaining transition issues get settled. Keep your saw sharp, and you will not be left behind as our industry moves forward. The next chapter outlines traditional A/V technology without special attention to IT. If you are new to A/V, then this chapter will be of interest.

# A Review of A/V Basics

## CONTENTS

11.0 Introduction to A/V Basics	400
11.1 A Digital View of an Analog World	400
11.2 Progressive and Interlace Images	401
11.3 Video Signal Timing	404
11.4 Video Resolutions and Aspect Ratios	405
11.4.1 Aspect Ratio Conversions	408
11.5 Video Signal Representations	409
11.5.1 The RGB and R'G'B' Signals	410
11.5.2 The Component Color Difference Signals	410
11.5.3 The Y'PrPb Component Analog Signal	411
11.5.4 The Y'CrCb Component Digital Signal	411
11.5.5 The Analog Composite Video Signal	413
11.5.6 The S_Video Signal	415
11.5.7 Analog and Digital Broadcast Standards	415
11.5.8 Professional Signal Formats—Some Conclusions	417
11.6 SDI Review—The Ubiquitous A/V Digital Link	418
11.6.1 The AES/EBU Audio Link	419
11.6.2 The Proteus Clip Server Example	420
11.7 Video Signal Processing and Its Applications	421
11.7.1 Interlace to Progressive Conversion—Deinterlacing	422
11.7.2 Standards Conversion	422
11.7.3 Compressed Domain Processing	423
11.8 A/V Bit Rate Reduction Techniques	424
11.8.1 Audio Bit Rate Reduction Techniques	426
11.8.2 Video Compression Overview	426
11.8.3 Summary of Lossy Video Compression Techniques	429

<b>11.9 Video Time Code Basics</b>	<b>433</b>
<b>11.9.1 Drop Frame Time Code</b>	<b>433</b>
<b>11.10 It's a Wrap—Some Final Words</b>	<b>434</b>
<b>References</b>	<b>434</b>

## 11.0 INTRODUCTION TO AV BASICS

Coverage in the other chapters has been purposely skewed toward the convergence of A/V + IT and the interrelationships of these for creating workflows. This chapter reviews traditional video and audio technology as standalone subjects without regard to IT. If you are savvy in A/V ways and means, then skip this chapter. However, if *chroma*, *luma*, *gamma*, and *sync* are foreign terms, then dig in for a working knowledge. Chapter 7 presented an overview of the three planes: user/data, control, and management. This chapter focuses on the A/V user/data plane and the nature of video signals in particular. Unfortunately, by necessity, explanations use acronyms that you may not be familiar with, so check the Glossary as needed.

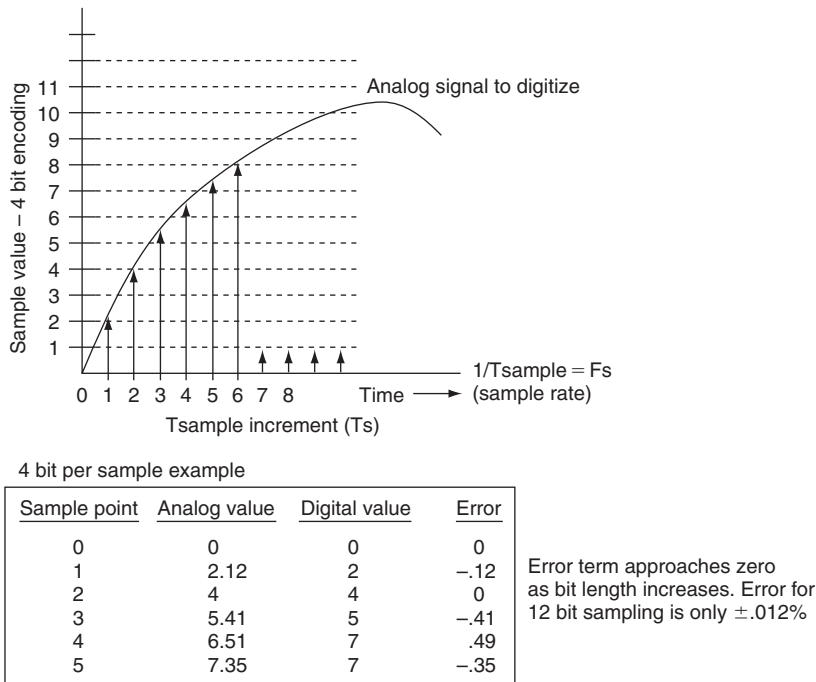
The plan of attack in this chapter is as follows: an overview of video fundamentals including signal formats, resolutions, A/V interfaces, signal processing, compression methods, and time code basics.

### 11.1 A DIGITAL VIEW OF AN ANALOG WORLD

Despite the trend for all things digital, we live in an analog world. In 1876, Bell invented his “electrical speech machine,” as he called it—the telephone—a great example of analog design. However, analog signals are susceptible to noise, are not easily processed, are not networkable, suffer loss during transmission, and are difficult to store. Digital representations of analog signals generally overcome these negatives. Sure, digital has its trade-offs, but in the balance it wins out for many applications. Figure 11.1 shows the classic transform from the analog domain to the digital. Transformation is performed by an A-to-D converter that outputs a digital value every sample period at rate  $F_s$ . Better yet, some digital video and audio signals are never in analog form but are natively created with software or hardware. Often, but not always, the digital signal is converted back to analog using a D-to-A operator.

There will always be analog diehards who claim that digital sampling “misses the pieces” between samples. Examining Figure 11.1 may imply this. The famous Nyquist sampling theorem states: “The signal sampling rate must be *greater than twice* the bandwidth of the signal to perfectly reconstruct the original from the sampled version using the appropriate reconstruction filter.” For example, an audio signal spanning 0 to 20 kHz and sampled at >40 kHz rates (48 or even 96 kHz is typical for professional audio) may be perfectly captured. It is one of those wonderful facts dependent on math and does not sacrifice “missing pieces” in any way.

Okay, there are two “pieces” involved here. One piece is along the horizontal time axis, as snapshot values only occur at sample points. However, according



**FIGURE 11.1** The analog digitization process.

to Nyquist, zero signal fidelity is lost if samples are spaced uniformly at a sufficiently high sample rate. The second piece is along the vertical (voltage) axis. In practice, the vertical domain must be digitized to  $N$ -bit precision. For audio signals, a 24-bit (16 and 20 also used) A-D is common, which yields vertical impression that cannot be detected aurally. For video, 12-bit precision is common for some applications, which yields noise that is well below visual acuity. The ubiquitous serial digital interface (SDI) link carries video signals at 10-bit resolution. Admittedly, 10-bit, especially 8-bit, resolution does produce some visual artifacts. Here is some sage signal-flow advice: digitize early, as in the camera, and convert to analog late, as for an audio speaker or not at all as for a flat panel display. Yes, core to AV/IT systems are digital audio and video.

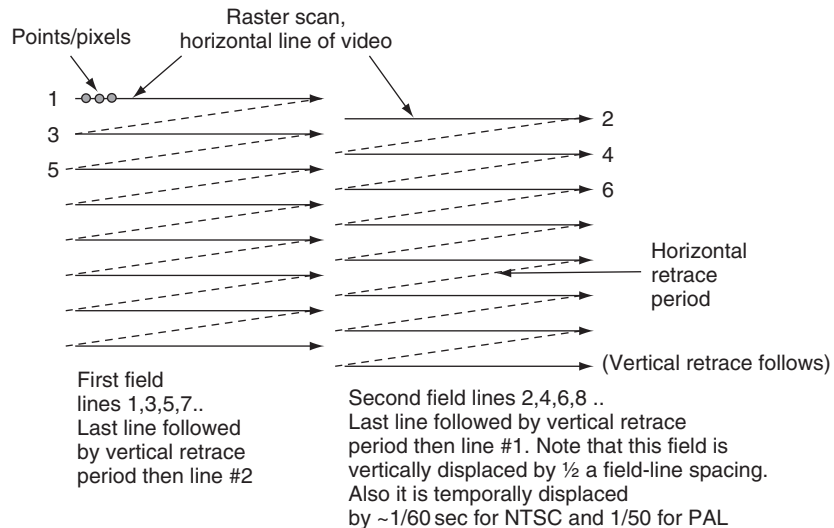
It should be mentioned that capturing an image with a sensor is a complex operation and that it is nearly impossible to avoid some sampling artifacts (aliasing) due to high-frequency image content. With audio, Nyquist sampling yields perfect fidelity, whereas with video there may be some image artifacts due to scanning parameter limitations with “random” image content.

## 11.2 PROGRESSIVE AND INTERLACE IMAGES

The display of the moving image is part art and part science. Film projects 24 distinct frames usually shuttered twice per frame to create a 48 image per second

sequence. The eye acts as a low-pass filter and integrates the action into a continuous stream of images without flicker. Video technology is based on the raster—a “beam” that paints the screen using lines that sweep across and down the display. The common tube-based VGA display uses a *progressive* raster scan; i.e., one frame is made of a continuous sequence of lines from top to bottom. Most of us are familiar with monitor resolutions such as  $800 \times 600$ ,  $1,024 \times 768$ , and so on. The VGA frame rate is normally 60 or 72 frames per second. Most VGA displays are capable of HD resolutions, albeit with a display size not large enough for the comfortable  $\sim 9$ -foot viewing distance of the living room. By analogy, a digital or film photo camera produces a progressive image, although it is not raster based.

The common analog TV display is based on an interlaced raster. Figure 11.2 illustrates the concept. A frame is made of two fields,<sup>1</sup> interleaved together. The NTSC frame rate is  $30^2$  frames per second (FPS) and it is 25 for PAL. The



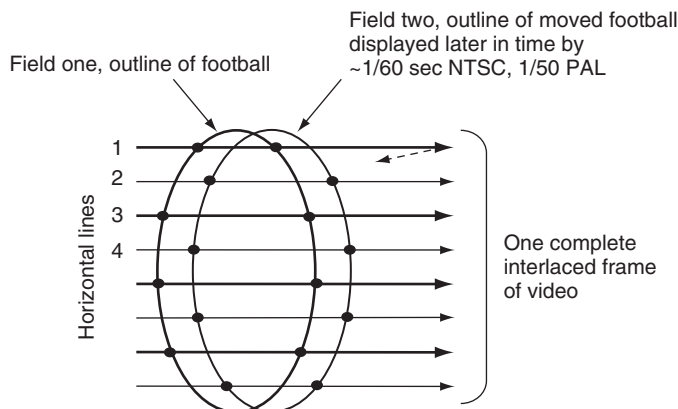
**FIGURE 11.2** An interlaced picture frame: First field followed by second field in time and space.

<sup>1</sup> In practice, field lines are numbered sequentially across both fields. To simplify explanations, however, they are numbered as odd and even in Figures 11.2 and 11.3.

<sup>2</sup> The frame rate of a NTSC signal is  $\sim 29.97$  frames per second, and the field rate is  $\sim 59.94$  fields per second. These strange values are explained later in this chapter. However, for simplicity, these values are often rounded in the text to 30 and 60, respectively. The PAL field rate is exactly 50, and the frame rate is 25 FPS. SECAM uses PAL production parameters and will not be explored further.

second field is displaced from the first in time by  $1/60$  (or  $1/50$ ) second and vertically offset by one-half field-line spacing. Each field has one-half the spatial resolution of the resultant frame for non-moving images. A field is painted from top to bottom and from left to right. Why go to all this trouble? Well, for a few reasons. For one, the eye integrates the 60 (50) field images into a continuous, flicker-free, moving picture without needing the faster frame scan rates of a progressive display. In the early days of TV, building a progressive 60 FPS display was not technically feasible. Second, the two fields combine to give an approximate effective spatial resolution equal to a progressive frame resolution. As a result, interlace is a brilliant compromise among image spatial quality, flicker avoidance, and high scanning rates. It is a form of video compression that saves transmission bandwidth at the cost of introducing artifacts when there is sufficient image motion.

Why does the interlace process introduce motion artifacts? Figure 11.3 shows the outline of a football as it is captured by two successive fields. Each field has half the resolution of the composite frame. Note that the ball has moved between field captures, so horizontal lines #1 and #2 display different moments in time but are adjacent in a spatial sense. A printout of a single frame (a frozen frame) with image motion looks ugly with jagged tears at the image edges. This can be seen in Figure 11.3. Fortunately, the eye is not highly sensitive to this oddity at 60 (50) fields per second. Also, when a captured image has detail on the order of a single display line height, then some obnoxious flicker will occur at the frame rate. With the advent of HD resolutions, both progressive and interlace formats are supported. Progressive frames appear more filmlike and do not exhibit image tearing. With the advent of flat panel displays at 60 and 120 Hz rates, interlaced formats will give way to all things progressive.



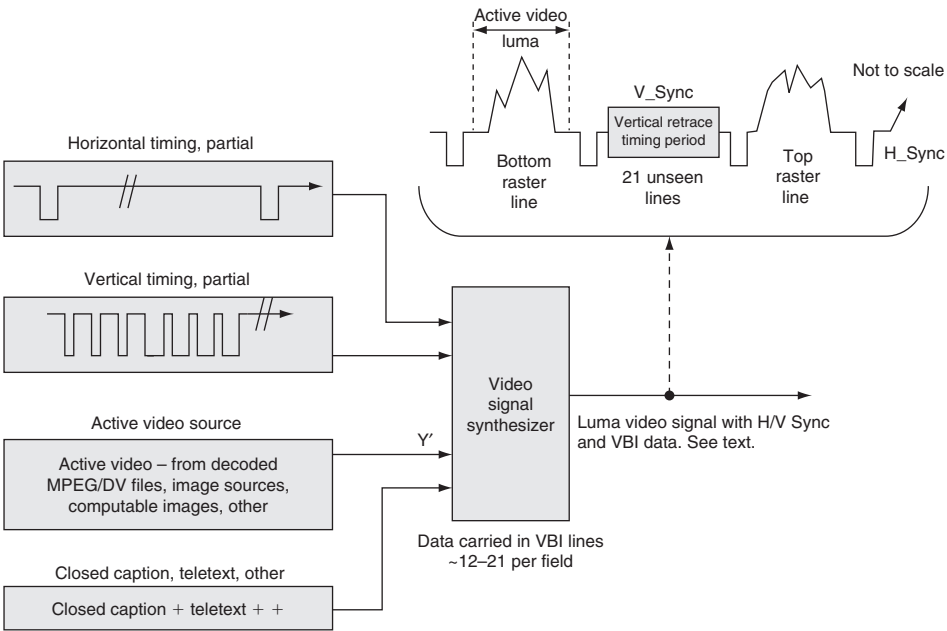
**FIGURE 11.3** Interlace offsets due to image movement between field scans.



Having trouble remembering the difference between a frame and a field? Think of this: a farmer took a picture of his two fields and framed it. There are two fields displayed in the frame.

11.3 VIDEO SIGNAL TIMING

In order to understand the basics of video timing, we have posited an example based on a monochrome (gray scale) signal. Color video signals are considered later, but the same timing principles apply. At each point along a horizontal line of video, the intensity (black to white) is relative to image brightness. To paint the screen either progressively or using an interlaced raster, we need a signal that can convey image brightness along with the timing to trigger the start of a line, the end of a line, and the vertical retrace to the top of the screen. The luma<sup>3</sup> signal in Figure 11.4 meets our needs. The luma signal represents the monochrome or lightness component of a scene. The active video portion of a line is sourced from file data, image sensors, computable values, and so on. It is of value



**FIGURE 11.4** *Synthesis of a monochrome video signal.*

<sup>3</sup> Technically, only the active picture portion of the signal is the luma value. However, for ease of description, let us call the entire signal luma. Luma (Y) and chroma are discussed in Section 11.5.

to note that most A/V files do not carry horizontal or vertical timing information. Timing must be added by hardware circuitry as needed by the delivery chain.

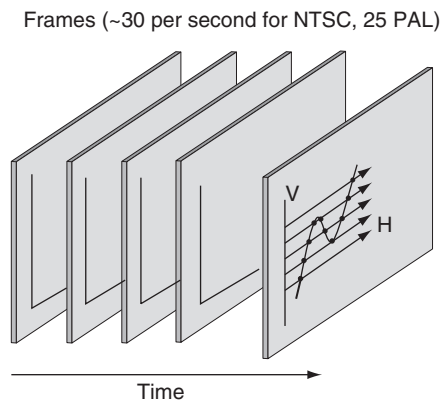
The receiver uses the horizontal timing period to blank the raster during the horizontal beam retrace and start the next line. The receiver uses the vertical timing to retrace the raster to the top of the image and blank the raster from view. The vertical period is about 21 unseen lines per field. Lines -12-21 (of both fields) are often used to carry additional information, such as closed caption text. This period is called the vertical blanking interval (VBI). The vertical synchronizing signal is complex, and all its glory is not shown in Figure 11.4. Its uniqueness allows the receiver to lock onto it in a foolproof way.

## 11.4 VIDEO RESOLUTIONS AND ASPECT RATIOS

We see a moving 2D image space on the TV or monitor screen, but it requires a 3D<sup>4</sup> signal space to produce it. Time is the third dimension needed to create the sensation of motion across frames. Figure 11.5 illustrates a sequence of complete frames (either progressive or interlace). There are four defining parameters for a color digital raster image:

1. **Number of discrete horizontal lines**—related to the vertical resolution. Not all of the lines are in the viewing window.
2. **Number of discrete sample “points” along a horizontal line and bit resolution per point**—both related to the horizontal resolution.

Not all of these points are in the viewing window. Each “point” is composed of three values from the signal set made from R, G, and B values (see Section 11.5).



**FIGURE 11.5** Video is a 3D spatiotemporal image.

<sup>4</sup> Video has the signal dimensions of horizontal, vertical, and time (HVT), so it may be called a 3D signal. Do not confuse this with a moving stereoscopic 3D image, which is a 4D signal.



- 3. **Number of frames per second**—related to the temporal resolution. This is  $\sim 29.97$  for NTSC and 25 for PAL, for example.
- 4. **Image aspect ratio**—AR is defined as the ratio of the picture width to its height. The traditional AR for analog TVs and some computer monitors is  $4 \times 3$ . So-called widescreen, or  $16 \times 9$ , is another popular choice.

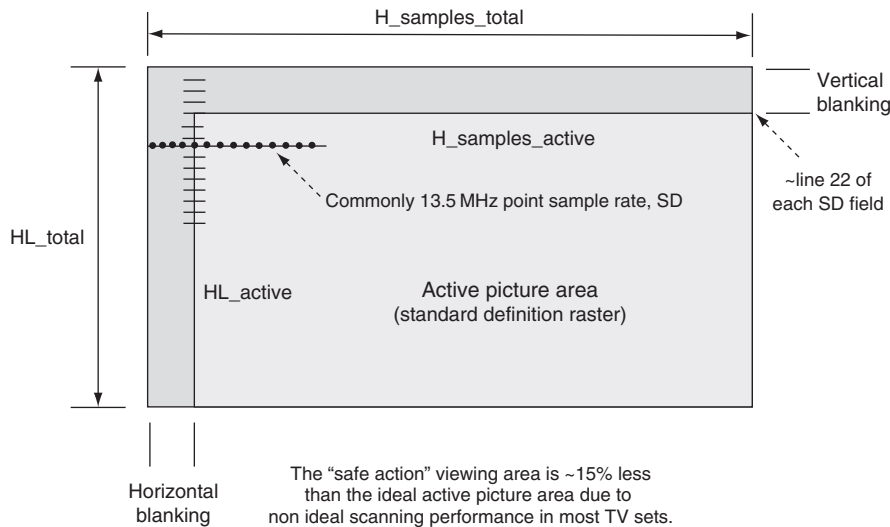
Next, let us look at the scanning parameters for the standard definition video image. There is a similar set of HD metrics as well, and some are covered in Table 11.1. Figure 11.6 outlines the key measures for a frame of SD video. They are as follows:

- **HL<sub>total</sub>**—total horizontal lines. For NTSC SD production, this is 525 lines and 625 for PAL.
- **HL<sub>active</sub>**—lines in the active picture area. For NTSC production, this is 480 lines and 576 for PAL. Only these lines are viewable. Usually, the first viewable line starts at the  $\sim 22$ nd line of each field.
- **H<sub>samples\_total</sub>**—digital sample points across one entire line, including the horizontal blanking period. There are 858 points (525 line system) and 864 points (625 line system) using a 13.5 MHz sample clock and a  $4 \times 3$  aspect ratio.

**Table 11.1** Common SDTV and HDTV Production Scanning Parameters

Common System Name Active Lines/Frames per Second (Fields per Second)	Active Picture Scanning Size: H <sub>samples_active</sub> $\times$ HL <sub>active</sub>	Total Picture Scanning Size: H <sub>samples_total</sub> $\times$ HL <sub>total</sub>	Display Aspect Ratio
480i/30 (60) or 525i/30 (60) SD	720 $\times$ 480	858 $\times$ 525	4 $\times$ 3
576i/25 (50) or 625i/25 (50) SD	720 $\times$ 576	864 $\times$ 625	4 $\times$ 3 and 16 $\times$ 9
480p/60 SD+	720 $\times$ 480	858 $\times$ 525	4 $\times$ 3
720p/60 HD	1,280 $\times$ 720	1,650 $\times$ 750	16 $\times$ 9
720p/50 HD	1,280 $\times$ 720	1,980 $\times$ 750	16 $\times$ 9
1,080i/30 (60) HD	1,920 $\times$ 1,080	2,200 $\times$ 1,125	16 $\times$ 9
1,080i/25 (50) HD	1,920 $\times$ 1,080	2,640 $\times$ 1,125	16 $\times$ 9
1,080p/30/60 HD	1,920 $\times$ 1,080	2,200 $\times$ 1,125	16 $\times$ 9
1,080p/25 HD	1,920 $\times$ 1,080	2,640 $\times$ 1,125	16 $\times$ 9
1,080p/24 HD	1,920 $\times$ 1,080	2,750 $\times$ 1,125	16 $\times$ 9

(1) The line count and sample metrics were defined earlier in this section (see Figure 11.6 for SD reference). The “i” term indicates interlace, the “p” indicates progressive scanning. It is common to refer to a 1080i system as either 1080i/30 or 1080i/60; the difference being the reference to frame versus field repeat rates. This reasoning applies to the other interlaced scanning standards as well. The 480p system falls between legacy SD and HD. It has twice the temporal resolution (60 frames per second) of SD formats, but because the horizontal resolution is still 720 points, it will be closed as SD+. Each of the frame rates 24, 30, and 60 has an associated system scaled by 1000/1001. NTSC uses the 525/30 system, and PAL uses the 625/25 system. A “point” along the horizontal line is composed of an RGB signal set.



**FIGURE 11.6** Image frame sizes and metrics.

- **H\_samples\_active**—digital sample points in viewing window. There are 720 active points for 525 and 625 line systems. This is a key metric for video processing and compression.

The horizontal sampling rate for SD production is 13.5 MHz. Other SD rates have been used, but this is widely accepted. For HD resolution, a common sampling rate is ~74.25 MHz. What is the total picture bit rate for an SD, 525,  $4 \times 3$  digital frame? Consider: **525\_bit\_rate** = 525 lines  $\times$  858 points/line  $\times$  3 samples/point  $\times$  29.97 FPS  $\times$  10 bits/sample = 405 Mbps. Another way to get to this same result is 13.5 MHz  $\times$  3 samples/point  $\times$  10 bits/sample = 405 Mbps.

A common data rate reduction trick in image processing is to reduce the color resolution without adversely affecting the image quality. One widely accepted method to do this, explained in Section 11.5.4, reduces the overall picture bit rate by one-third, which yields a 270 Mbps ( $= 405 \text{ Mbps} \times 2/3$ ) data rate. This value is a fundamental data rate in professional standard definition digital video systems.

Incidentally, 10 bits per sample is a popular sampling resolution, although other values are sometimes used. Surprisingly, for a 625 (25 FPS, same one-third bit rate reduction applied) system, the total picture bit rate is also exactly 270 Mbps. For HD 1080i/30 video the full frame sample rate is 1.485 Gbps. See Appendix G for more insight into this magic number. Note that this value is not the *active picture data* rate but the total frame payload per second. The active picture data rate is always less because the vertical and horizontal blanking areas are not part of the viewable picture.

There are a plethora of other video format standards, including one for Ultra High Definition Video (UHDV) with  $7,680 \times 4,320$  pixels supported by a

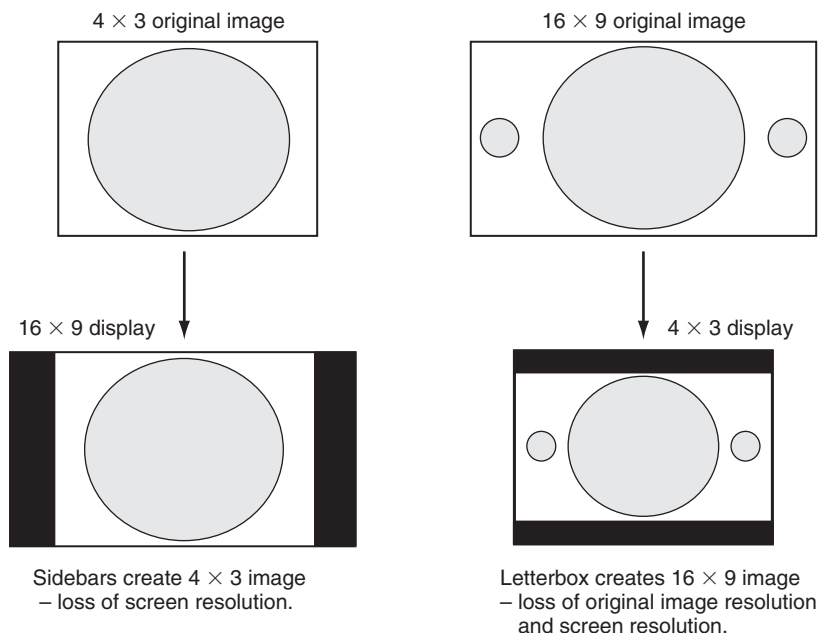
super surroundsound array with 22.2 speakers. An uncompressed 2-hour movie requires  $\sim 25\text{TB}$ ! Developed by Japan's NHK, this could appear circa 2016. Digital Cinema and associated production workflows (Digital Intermediate, DI) are often produced in 2 K (2,048 horizontal pixels per line) or even in 4 K (4,096 pixels) formats.

See SMPTE 274M and 296M for details on the sampling structures for HD interlace and progressive image formats.

Most professional gear—cameras, NLE's video servers, switchers, codecs, graphics compositors, and so on—spec their operational resolutions using one or more of the metrics in Table 11.1. In fact, there are other metrics needed to completely define the total resolution of a video signal, and Section 11.5 digs deeper into this topic.

### 11.4.1 Aspect Ratio Conversions

One of the issues with scanning formats is display mapping. Programming produced in  $16 \times 9$  and displayed on a  $4 \times 3$  display, and vice versa, needs some form of morphing to make it fit. Figure 11.7 shows the most popular methods to map one AR format into another for display. Another method includes panning and scanning the  $16 \times 9$  original image to select the “action areas” that fit into a  $4 \times 3$  space. This is used commonly to convert widescreen movies to  $4 \times 3$  without using letterbox bars. Still another method uses anamorphic squeezing to

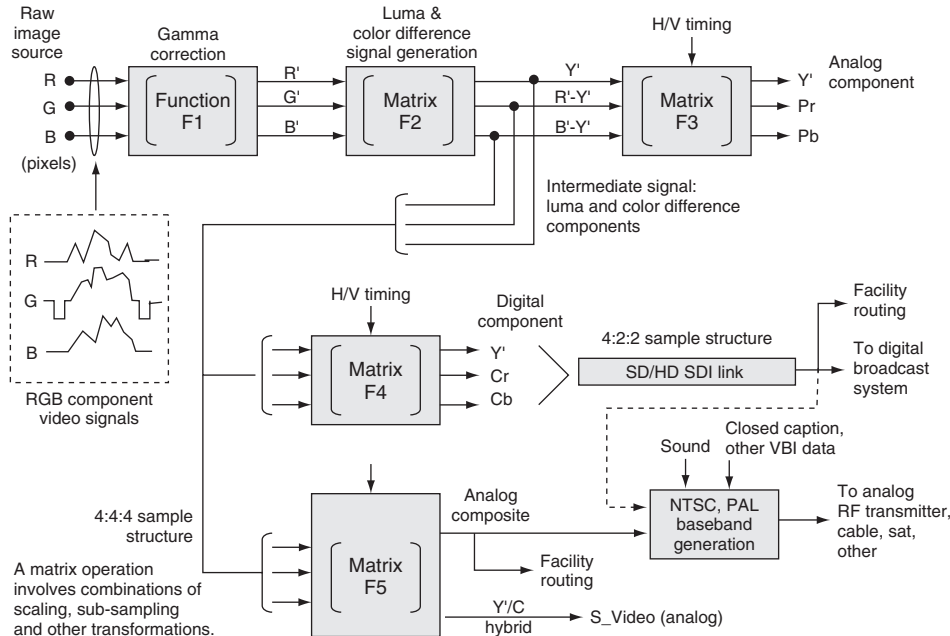


**FIGURE 11.7** Common aspect ratio conversions.

transform images from one AR to another. This technique always creates some small image distortion, but it utilizes the full-screen viewing area. Whatever the method, AR needs to be managed along the signal chain. Generally, both  $16 \times 9$  and  $4 \times 3$  formats are carried over the same type of video links.

## 11.5 VIDEO SIGNAL REPRESENTATIONS

There are many ways to represent a video signal, and each has some advantage in terms of quality or bandwidth in both analog and digital domains. Figure 11.8 outlines eight different signal formats. Figure 11.8 is conceptual, and no distinction is made between SD/HD formats. The H/V timing is loosely applied (or not applied) for concept only. It is not a design specification, but a high schematic level view of these important signal formats. In practice, some of the signals may be derived more directly via other paths, so consider Figure 11.8 as representational of the conversion processes. The *matrix Fx* term indicates mathematical operations for signal conversion and is different for HD and SD matrix operations. Keep in mind that all the operations are reversible; i.e., R'G'B' can be recovered for display from, say, Y'CrCb or S\_Video, but there is always some loss of fidelity due to the effects of the matrix math. Let us examine each signal and conversion path in Figure 11.8.



**FIGURE 11.8** Conceptual view of main video signal representations.

### 11.5.1 The RGB and R'G'B' Signals

At the start of the image chain is a camera imager or other means to generate a raw RGB signal. Red, green, and blue represent the three primary color components of the image; 0 percent RGB is pure black and 100 percent is pure white. The signal requires three channels for transport: one per pixel component. The three time-based signals are shown by example in the dotted box, with the green signal having horizontal syncs indicating the start and end of a line. This signal set is rarely used standalone but needs conversion to a gamma-corrected version,  $R'$ ,  $G'$ , and  $B'$ . Basically, gamma correction applies a power function to each of the three RGB components. The general function (F1) is of the approximate form  $A' = A^{0.45}$ , where the 0.45 is a typical gamma value. An apostrophe is used to represent a gamma-corrected signal. According to (Poynton 2003), "Gamma is a mysterious and confusing subject because it involves concepts from four disciplines: physics, perception, photography, and video. In video, gamma is applied at the camera for the dual purposes of precompensating the nonlinearity of the display's CRT and coding into perceptually uniform space." Charles Poynton does a masterful job of explaining the beauty of color science, gamma, luma, and chroma, so consult the reference for a world-class tour of these important parameters and much more. For our discussion, gamma correction is an exponential scaling of the RGB voltage levels. In the context of Figure 11.6, each "sample point" refers to the three pixels as a data set.

### 11.5.2 The Component Color Difference Signals

A full-bandwidth  $R'G'B'$  signal is a bandwidth hog and is used only for the most demanding applications where quality is the main concern (graphics creation, movies, and other high-end productions). Ideally, we want a reduced bandwidth signal set that still maintains most of the RGB image quality. We find this at the next stop in the signal chain: conversion to the  $Y'$ ,  $R'-Y'$ ,  $B'-Y'$  signal components. These are not RGB pixels but are derived from them. The  $Y'$  term is the luma signal (gray scale), and the other two are color difference signals often called chroma signals. Interestingly, statistically, 60–70 percent of scene luminance comprises green information. Accounting for this, and removing the "brightness" from the blue and red and scaling properly (matrix F2), yields two color difference signals along with the luma signal. What are the advantages of this signal form? There are several. For one, it requires less data rate to transmit and less storage capacity (compared to  $R'G'B'$ ) without a meaningful hit in image quality for most applications. Yes, there is some non-reversible image color detail loss, but the eye is largely insensitive to a high degree due to the nature of F2's scaling. In fact, there is a loss of 75 percent of the colors when going from an  $N$ -bit  $R'G'B'$  to an  $N$ -bit color difference format. If  $N$  is 8 bits, then some banding artifacts will be seen in either format. At 12 bits, artifacts are imperceptible. Despite the loss of image quality, the transformation to color difference signals is a worthwhile trade-off, as

will be shown in a moment. See (Poynton 2003) for more information on this transformation.

Giving a bit more detail, the following conversions are used by matrix F2 for SD systems:

$$Y' = 0.299R' + 0.587G' + 0.114B' \text{ (the luma signal)} \quad (11.1)$$

$$R' - Y' = 0.701R' - 0.587G' - 0.114B' \text{ (a chroma signal)} \quad (11.2)$$

$$B' - Y' = -0.299R' - 0.857G' + 0.886B' \text{ (a chroma signal)} \quad (11.3)$$

The coding method is slightly different for HD formats, as HD and SD use separately defined colorimetry definitions. Other than this, the principles remain the same.

### 11.5.3 The Y'PrPb Component Analog Signal

The color difference signals are intermediate forms and are not transported directly but are followed by secondary conversions to create useful and transportable signals. Moving along the signal chain, the Y'PrPb signal set is created by the application of matrix operation F3. Y'PrPb is called the analog component signal, and its application is mainly for legacy use. The *P* stands for parallel, as it requires three signals to represent a color image. Function F3 simply scales the two chroma input values ( $R' - Y'$  and  $B' - Y'$ ) to limit their excursions. The  $Y'$  signal (and sometimes all three) has H/V timing, and its form is shown as the middle trace of Figure 11.10.

### 11.5.4 The Y'CrCb Component Digital Signal

Matrix F4 produces the signal set Y'CrCb, the digital equivalent of Y'PrPb with associated scaling and offsets. The C term refers to chroma. Commonly, each component is digitized to 10-bit precision. This signal is the workhorse in the A/V facility. The three components are multiplexed sequentially and carried by the SDI interface. More on this later. One of the key advantages to this format is the ability to decrease chroma spatial resolution while maintaining image quality, thereby reducing signal storage and data rate requirements. Sometimes this signal is erroneously called YUV, but there is no such format. However, the U and V components are valuable and their usage is explained in Section 11.5.5.

#### 11.5.4.1 Chroma Decimation

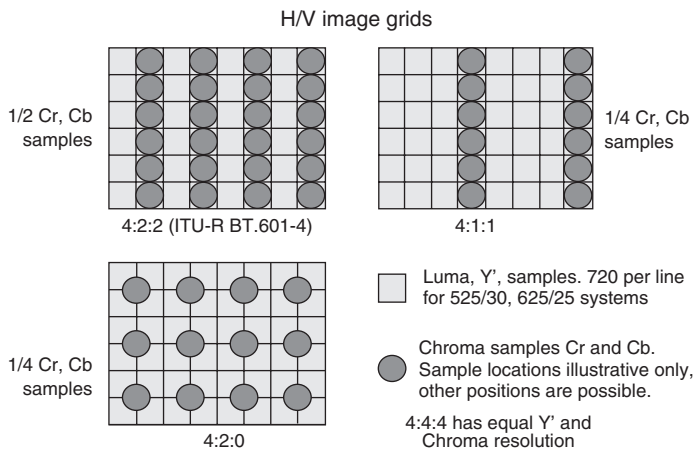
The Y'CrCb format allows for clever chroma decimations, thereby saving additional storage and bandwidth resources. In general, signals may be scaled spatially (horizontal and/or vertical) or temporally (frame rates) or decimated (fewer sample points). Let us focus on the last one. A common operation is to decimate the chroma samples but not the luma samples. Significantly, chroma

information is less visually important than luminance, so slight decimations have little practical effect for most applications. Notably, chroma-only decimation cannot be done with the RGB format—this is a key advantage of the color difference format.

Our industry has developed a short-form notation to describe digital chroma and luma decimation, and the format is A:B:C. The A is the luma horizontal sampling reference, B is the chroma horizontal sampling factor relative to A, and C is the same as B unless it is zero, and then Cr and Cb are subsampled vertically at 2:1 as well. It is confusing at best, so it is better not to look for any deep meaning in this shorthand. Rather, memorize a few of the more common values in daily use. The first digit (A) is relative to the actual luma pixel sample rate, and a value of 4 (undecimated luma) is the most common. Luma decimation, in contrast to chroma decimation, sacrifices some observable image quality for a lower overall bit rate. In most cases, it is the chroma that is decimated, not the luma. Some of the more common notations are as follow.

- 4:4:4—luma and 2 chroma components are sampled equally across H and V.
- 4:2:2—chroma (Cr and Cb) sampled at half of luma rate in the horizontal direction (1/2 chroma lost).
- 4:2:0—chroma sampled at half of luma rate in H *and* V directions (three-fourths chroma lost). The DV-(625/50) format uses this, for example.
- 4:1:1—chroma sampled at one-quarter of luma rate in the H direction (three-fourths chroma lost). The DV-(525/60) format uses this, for example.
- 3:1:1—as with 4:1:1 except one-fourth of luma samples are also discarded. A 1,920 sample point HD line is luma subsampled to 1,440 samples. Sony HDCAM format, for example.
- 4:4:4:4—the last digit indicates that an alpha channel (transparency signal) is present and sampled at the luma rate.

It turns out that 4:2:0 and 4:1:1 have an equal number of chroma samples per frame, but the samples are at different locations on the grid. Figure 11.9 shows examples of the Y', Cr, Cb sample grid for different methods. The common SDI studio link normally carries 4:2:2 SD/HD video at 10 bits per sample. Conceptually, the sequential order of carriage over SDI for 4:2:2 samples is Y' Cr Y' Cb Y' Cr Y' and so on. It is easy to see the one-half Cr and Cb chroma rate compared to the luma rate. For 4:2:0 there are one-half as many chroma samples compared to 4:2:2 and one-fourth compared to 4:4:4 (Y' Cr Cb Y' Cr Cb Y', etc.). In practice, the decimated Cr and Cb samples are derived by averaging several adjacent chroma values from the 4:4:4 signal. This gives a slightly better result than unceremoniously dropping chroma samples from the 4:4:4 source.



**FIGURE 11.9** A comparison of chroma sampling in 4:2:2, 4:1:1, and 4:2:0 formats.

The 4:2:2, 4:2:0, and 4:1:1 decimations are used as the source format for many video compression methods, including MPEG and DV. In fact, these codecs get an extra boost of compression when they decimate the chroma before the actual compression starts. Both DVB and ATSC digital transmissions systems utilize 4:2:0 decimated chroma at 8-bit resolution luma and chroma. The 4:4:4 HD R'G'B' format is used for high-end applications, and a dual-link HD-SDI method ( $2 \times 1.485$  Gbps data rate) is defined for carrying the three synchronous signals.

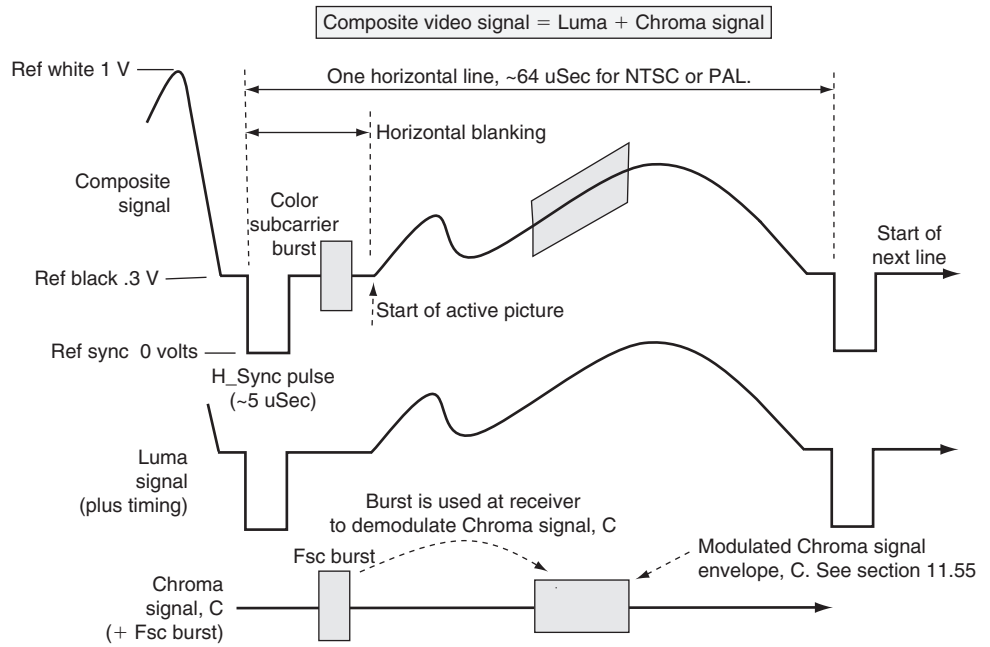
### 11.5.5 The Analog Composite Video Signal

The analog composite<sup>5</sup> video signal is output from matrix F5, as shown in Figure 11.8. This signal is illustrated in Figure 11.10 (top trace). The middle trace is the luma (plus H/V timing) signal. The bottom trace is the chroma signal (plus color burst, explained shortly). The top trace is composed of the sum of the bottom two signals and contains color and brightness information in one signal. A receiver can recover the H/V timing, brightness, and color details from the signal.

The composite signal is a wonderful invention, has a long history of use, and is the basis of the NTSC and PAL analog transmissions systems. Over the past 50 years, most TV stations and A/V facilities have used it as the backbone of their infrastructure, although the digital component SDI link is quickly replacing it. Most consumer A/V gear supports composite I/O as well. Also, there is a digital composite version of this often referred to as a “4F<sub>sc</sub> composite,” but it is used infrequently.

<sup>5</sup> CVBS, composite video burst and sync, is shorthand used to describe a composite signal.





**FIGURE 11.10** The making of a composite video signal.

Note: These are conceptual diagrams and some details are simplified (not to scale).

So what is the trick for carrying color and luma as one signal? Actually, there are several. First, let us define the U and V signals. Simply, these are scaled versions of the two color difference signals  $B' - Y'$  and  $R' - Y'$ , [Eqs. (11.3) and (11.2) (Section 11.5.2)].

$$U = 0.492 (B' - Y') \text{ and } V = 0.877 (R' - Y')$$

If we want to reduce the bandwidth further, U and V are bandwidth filtered and scaled. Next, they are combined into a single color signal, C, by quadrature modulating a subcarrier at frequency  $F_{sc}$ .

$$C = U \times \sin(\omega T) + V \times \cos(\omega T), \quad (11.4)^6$$

where<sup>6</sup>  $\omega = 2\pi F_{sc}$  and  $F_{sc}$  is the color subcarrier frequency. C is formed by AM modulating  $\sin(\omega T)$  with U and  $\cos(\omega T)$  with V both at the same frequency  $F_{sc}$ . C is shown as the bottom trace of Figure 11.10 along with the color burst reference signal, explained later.

<sup>6</sup> The sine  $\sin(\omega T)$  term generates a pure single wave at frequency  $F_{sc}$ . The cosine  $\cos(\omega T)$  term generates a wave at the same location but shifted by  $90^\circ$ . The resultant signal C can be demodulated with U and V recovered.

$F_{sc}$  is chosen at  $\sim 3.58$  MHz for NTSC and  $\sim 4.43$  MHz for most PAL systems. Now, the composite baseband video signal is

$$\text{Composite signal} = Y' + C + \text{burst} + \text{H/V timing} \quad (11.5)$$

When we select the value of the color subcarrier  $F_{sc}$  judiciously, the  $Y'$  and  $C$  terms mix like oil and water; they can be carried in the same bucket, but their identity remains distinct. Ideally, a receiver can separate them and, via reverse matrix operations, re-create the  $R'G'B'$  signals ready for display. See again Figure 11.10 and imagine the bottom trace ( $C$ ) and the middle trace  $Y'$  summed to create the top trace—this is Equation (11.5). In reality, due to a variety of factors,  $R'G'B'$  is not perfectly recovered but close enough in practice. Also, the  $C$  component is bandwidth limited by a filter operation before being summed with the luma signal; this limits the chroma resolution slightly.

In order for the receiver to demodulate  $C$  and recover  $U$  and  $V$ , it needs to know precisely the frequency and phase of the subcarrier. The “color burst” is an 8–10 cycle sample of  $\sin(\omega T)$  injected for this purpose.

It is a beautiful technique. See (Jack 2001) and (Poynton 2003) for more details on both composite and component signals and interfacing.

Composite video may be viewed as an early form of video compression. When luma and chroma are combined in clever ways, the bandwidth required to transport and store video is reduced significantly. The composite formulation makes acceptable compromises in image quality and has stood the test of time.

### 11.5.6 The S\_Video Signal

The final signal to discuss is the analog S\_Video signal, sometimes called Y/C or YC. This is just signals  $Y'$  and  $C$  carried on separate wires. Because there are never any  $Y'/C$  mixing effects as may occur in a pure composite signal,  $R'G'B'$  can be recovered with higher fidelity. The burst is carried on the  $C$  signal. S\_Video is rarely used in professional settings but is a popular SD consumer format.

### 11.5.7 Analog and Digital Broadcast Standards

All of the video standards reviewed in the preceding sections define baseband signals. These are not directly transmittable without further modification. Figure 11.8 shows the step of adding sound and other ancillary information to form a complete NTSC or PAL baseband signal. For analog broadcasts, the complete signal is upconverted to a RF channel frequency for ultimate reception by a TV. The input to this process may be a composite analog or digital component signal for added quality. Audio modulation uses FM and has a subcarrier of 4.5 MHz for NTSC. At the receiver, the RF signal is deconstructed and out pops  $R$ ,  $G$ ,  $B$ , and sound ready for viewing.

Three analog TV transmission systems span the world; they are NTSC, PAL, and SECAM (see the Glossary). Each has variations, with the color burst frequency and total baseband bandwidth being key variables. NTSC/M, J and PAL/B, D, G, H, I, M, N, and various SECAM systems are used worldwide and are adopted on a per country basis. See (Jack 2001) for a complete list of systems and adopted countries.

Before 1953, NTSC only defined B&W visuals with a temporal rate of exactly 30 FPS. When NTSC color was introduced, the committee had a conundrum to handle. If any non-linearity arose in the signal chain, color spectra information, centered approximately at 3.58 MHz ( $F_{sc}$ ), and the sound-modulated spectra, centered on 4.5 MHz, may interfere. The resulting intermodulation distortion would be apparent as visual and/or aural artifacts. The problem was avoided by changing the temporal frame rate. This is an overly simplified explanation, but the end result was to apply a scaling factor of 1,000/1,001 to the B&W frame rate. This resulted in a new frame rate of  $\sim 29.97$  and a corresponding field rate of  $\sim 59.94$ . No big deal it seems. However, this change introduced a timing discrepancy that is a pain to deal with, as the wall clock is now slightly faster than the field rate.

The 1,000/1,001 factor is awkwardly felt when a video signal or file is referenced by time code. Time code assigns a frame number to each video frame, but a  $\sim 29.97$  FPS rate yields a non-integer number of frames per second. A frame location based on a wall clock requires some time code gymnastics to locate it in the sequence. See Section 11.9 for more information on time code.

### **11.5.7.1 Digital Broadcast Standards**

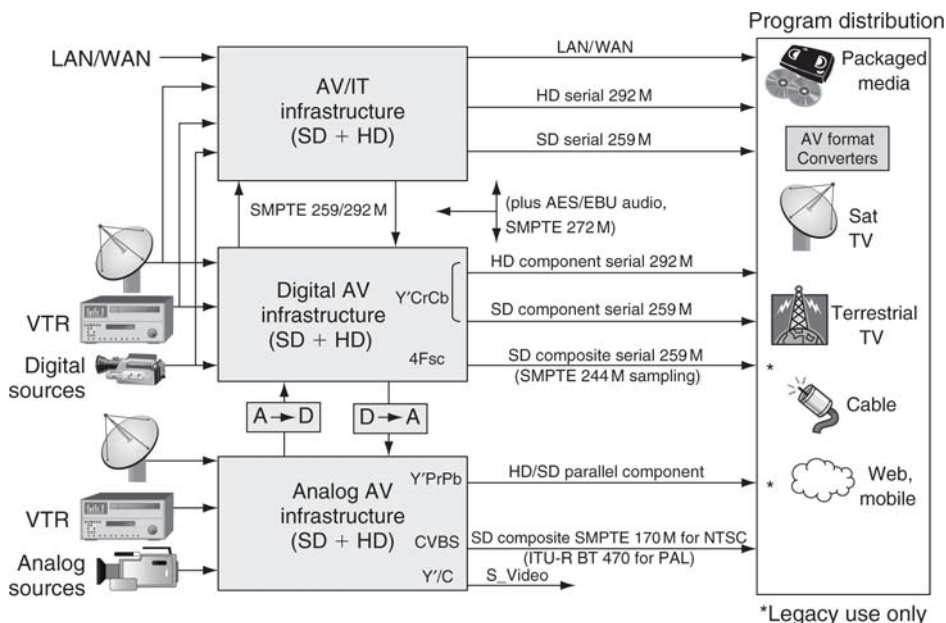
Analog TV broadcast systems are slowly being replaced with digital transmission systems worldwide. Figure 11.8 shows a SDI signal source feeding a digital broadcast system. There are four systems in general use. In Europe, Digital Video Broadcasting (DVB) standards began development in 1993 and are now implemented by 55+ countries over terrestrial, satellite, and cable. In the United States, the Advanced Television Systems Committee (ATSC) produced its defining documents in 1995 as the basis of the terrestrial DTV system. As of February 17, 2009, all full-power U.S. stations are transmitting using DTV formats. In Japan, the Integrated Services Digital Broadcasting (ISDB) system began life in 1999 and supports terrestrial, satellite, and cable. In China, DMB-T/H is specified with support for MPEG AVC and other compression standards. All four systems offer a selection of SD and HD video resolutions, use MPEG2 compression (or advanced versions), have six-channel surround sound, support non-A/V data streams, and use advanced modulation methods to squeeze the most bits per allocated channel bandwidth. The MPEG Transport Stream (TS) is a common data carrier for these systems. The TS is a multiplexing structure that packages A + V + data into a single bit stream ready for transmission. The TS method has found wide use. Many codecs have a mapping into the TS structure, including the popular H.264 method.

The typical transmitted maximum compressed HD data rate is  $\sim 20$  Mbps for 1080i/60 programming, although smaller values are common as broadcasters trade off image quality for extra bandwidth to send other programming or data streams. The main A/V program source to the encoding and transmission system is usually SDI or HD-SDI signals. No longer are composite signals used or desired. The new standards create a digital pipe to the viewer, and the broadcaster can use it in a wide variety of ways. The Web is filled with information on digital broadcast standards. See [www.atsc.org](http://www.atsc.org) or [www.dvb.org](http://www.dvb.org) for more information.

### 11.5.8 Professional Signal Formats—Some Conclusions

How should we rank these signal formats in terms of quality? Well, the purist is R'G'B', but due to its high bandwidth requirements, it is used only for very high-end A/V applications. The SD/HD Y'CrCb format is the workhorse of most facilities. It provides excellent quality; all components are separate, 4:2:2 resolution, 10 bits per component and all digital. Component video values are easily stored in a file. Plus, audio channels may be multiplexed into the SDI stream, creating an ideal carrier for A/V essence. All analog and composite formats are fading out. There are other formats, for sure, and some are specific to consumer use and others for very high-end HD use. Knowledge of the ones covered in this section will allow you to springboard to other formats with ease.

Figure 11.11 outlines a general selection of common interface standards across analog, digital, and AV/IT systems. Although R'G'B' is also transported



**FIGURE 11.11** Common professional A/V interface standards.

for high-end applications, it is not shown on the diagram. Most digital facilities use SDI links and AES/EBU links to carry A/V signals. The next two sections review these two important links. Together with LAN/WAN, they make up the lion's share of transport means in a converged AV/IT system.

In terms of video quality and transport convenience, digital trumps analog, component formats trump composite, and serial links trump parallel links.



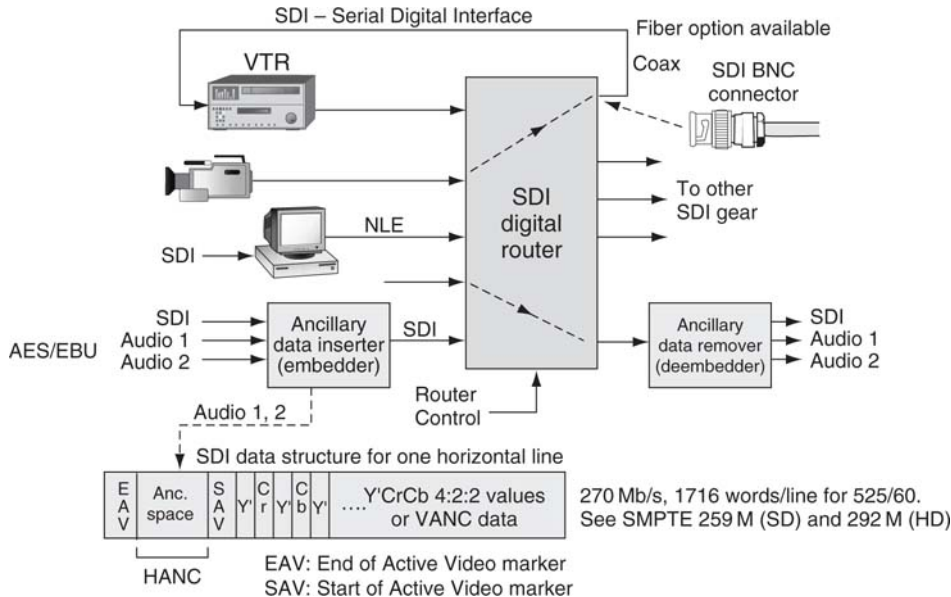
## 11.6 SDI REVIEW—THE UBIQUITOUS A/V DIGITAL LINK

The serial digital interface (SMPTE 259M for SD, 292M for HD) link has revolutionized the digital A/V facility. It has largely replaced the older analog composite link and digital parallel links, so it is worth considering this link for a moment to fully appreciate its power within the digital facility. SDI supports component video (Y'CrCb) SD and HD formats (supporting a variety of frame rate standards). The typical SD line bit rate is 270 Mbps (with support from 143 to 360 Mbps depending on the data format carriage) and is designed for point-to-point, unidirectional connections (see Appendix G).

SDI is not routable in the LAN/IP sense; however, an entire industry has been created to circuit switch SDI signals using video routers. Due to non-displayable areas in the raster scan, the active image data payload is less than the link rate of 270 Mbps. For example, the actual active picture data payload of a 525-line, 10 bits/pixel, 4:2:2 component signal is ~207.2 Mbps, not counting the vertical and horizontal blanking areas. Figure 11.12 shows the basic framing structure of a SDI signal and an example of SDI link routing. Routers range in size from a basic  $8 \times 1$  to mammoth  $512 \times 512$  I/O and larger. About 57 Mbps out of the total of 270 is available to carry non-video payloads, including audio. These non-video data are carried in both horizontal ancillary (HANC) and vertical ancillary (VANC) blanking areas. In a non-picture line, data between SAV and EAV markers carry ancillary VANC data.

Examples of payloads that SDI can carry are as follows:

- SD video with embedded digital audio. One channel of video and eight embedded uncompressed stereo pairs (16 audio channels) are a supported payload.
- Ancillary data carried in HANC and VANC areas. SMPTE has standardized several schemes for HANC and VANC non-video data carriage.
- Carriage of compressed formats in the SDTI-CP wrapper. SDTI-CP (Serial Digital Transport Interface—Content Package) is a means to carry compressed formats such as MPEG and DV over SDI links. This method



**FIGURE 11.12** Basic SDI use and framing data structure.

was popularized by Sony's IMX digital MPEG VTR and Panasonic's DVCPro VTRs. See SMPTE standards 326M and 331M for more information.

- HD-SDI (SMPTE 292M) uses a line bit rate 5.5 times higher than that of 259M (270 Mbps typical), clocking in at 1.485 Gbps to carry uncompressed HD video, embedded audio, and ancillary data.
- At the high end of production standards, SMPTE defines the 424M/425M serial link for carrying 1,080p/60 video at 3 Gbps. Also, SMPTE 435 defines a 10 Gbps (actually 10.692 nominal) link for carrying a 12 bits/pixel, 4:4:4, RGB format and other common formats in multiplexed style.

There is no H/V timing waveform embedded in a digital SDI link as with the analog luma or composite video signal. Rather, timing is marked using SAV, EAV words, and other sync words. These markers may be used to help create H/V timing as needed. Most professional AV/IT systems will be composed of a mix of SDI, HD-SDI, AES/EBU audio, and LAN/WAN links. The workhorse for audio is the AES/EBU link discussed next.

### 11.6.1 The AES/EBU Audio Link

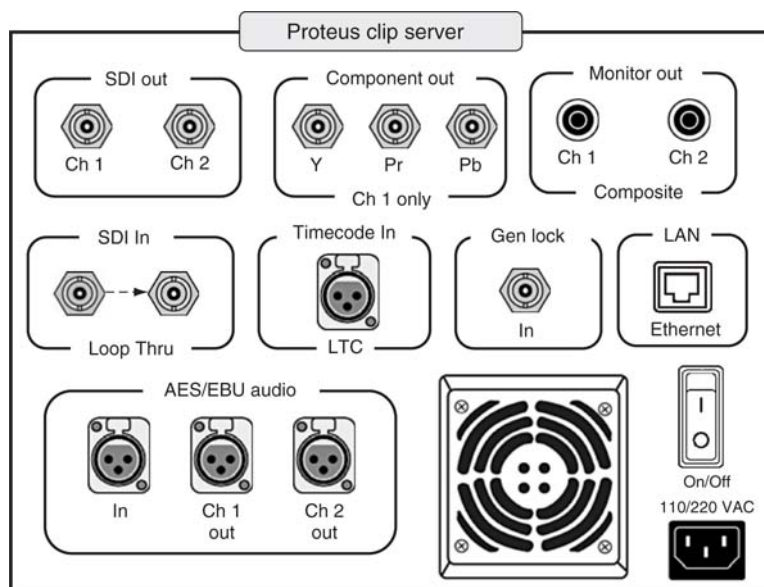
The AES/EBU (officially AES3) audio standard defines the data structures and physical serial link to stream professional digital audio. It came about as a

result of collaboration between the AES and the EBU and was later adopted by ANSI. The default AES/EBU uncompressed sample rate is 48 kHz, although 44.1, 96 kHz and others are also supported. An audio sample accuracy of 24 bits/sample (per channel) is supported, but not all equipment will use the full 24 bits. Most devices use 16 or 20 bits per sample. A full frame is 32 bits, but 8 bits of this are dedicated to non-audio user bits and synchronization.

A shielded, twisted-pair cable with XLR connectors on both ends is used to transfer two channels of audio and other data using a line data rate of about 3 Mbps for 48 kHz sampling. There is also a version that uses coax cable with BNC connectors. Both versions are in wide use. The link also supports a raw data mode for custom payloads. One example of this is the transport of compressed AC3 5.1 and Dolby-E audio formats. The AES/EBU data structure may be embedded into a SDI stream for the convenience of carrying up to 16 audio channels and video on one cable (see Figure 11.12).

### 11.6.2 The Proteus Clip Server Example

Announcing the new—drum roll, please—Proteus clip server: the newest IT-friendly video server with the ability to record (one channel in) and play back (two channels out) A/V files using networked storage. Okay, Figure 11.13 is fictional, but it exemplifies an AV/IT device with a respectable quota of rear panel I/O connectors. A little inspection reveals SDI video I/O (BNC connectors), AES/EBU audio I/O (XLR connector version), time code signal in, a Gen Lock



**FIGURE 11.13** *The Proteus clip server: Rear panel view.*

input, an analog signal output using Y'PrPb ports, composite monitoring ports, and a LAN connection for access to storage/files, management processes, device control, and other network-related functions.

The Gen Lock input signal (sometimes called *video reference*) is a super clean video source with black active video or possibly color bars. The Proteus server extracts the H/V timing information from the Gen Lock signal and uses this to perfectly align all outputs with the same H/V timing. Most A/V facilities use a common Gen Lock signal distributed to all devices. This assures that all video signals are H/V synced. Alignment is needed to assure clean switching and proper mixing of video signals. Imagine cross fading between two video signals each with different H/V timing; the resultant hodgepodge is illegal video. Sometimes it is unavoidable that video signals are out of sync. The services of a “frame sync” are used to align signals to a master video reference (see Appendix B).

Remember too that SDI signals may carry embedded audio channels on both inputs and outputs. Also, the SDI-In signal has a loop-through port. The input signal is repeated to the second port for routing convenience.

## 11.7 VIDEO SIGNAL PROCESSING AND ITS APPLICATIONS

Video is a signal with 3D spatiotemporal resolutions and has traditionally been processed using hardware means. With the advent of real-time software processing, hardware is playing a smaller role but still finds a place, especially for HD rates. Image handling may be split into two camps: raw image generation and image processing. The first is about the creation and synthesis of original images, whereas the second is about the processing of existing images. Of course, audio signal processing is also of value and widely used, but it will not be discussed here. Image processing is a weighty subject, but let us peek at a few of the more common operations.

1. Two- and three-dimensional effects, video keying, compositing operations, and video parameter adjustments
2. Interlaced to progressive conversion—deinterlacing
3. Standards conversion (converting from one set of H, V, T spatiotemporal samples to another set)
4. Linear and non-linear filtering (e.g., low-pass filtering)
5. Motion analysis, tracking (track an object across frames, used by MPEG)
6. Feature detection (find an object in a frame)
7. Noise reduction (reduce the noise across video time, 3D filtering)
8. Compressed domain processing (image manipulation using compressed values)

These operators are used by many video devices in everyday use. The first three deserve special mention. Item #1 is the workhorse of the video delivery



chain. Graphics products from a number of vendors can overlay moving 2D and 3D fonts onto live video, squeeze back images, composite animated images, transition to other sources, and more. Video editors apply 2D/3D effects of all manner during program editing. Real-time software-based processing has reached amazing levels for both SD and some HD effects operators. Video parameter adjustments of color, contrast, and brightness are run of the mill for most devices.

Video keying adds a bit of magic to the nightly TV news weather report. A “key” is a video signal used to “cut a hole” in a second video to allow for insertion of a third video signal (the fill) into that hole. Keying places the weather map (the fill) behind the presenter who stands in front of a blue/green screen. The solid color background is used to create the key signal. Keying without edge effects is a science, and video processing techniques are required for quality results.

### 11.7.1 Interlace to Progressive Conversion—Deinterlacing

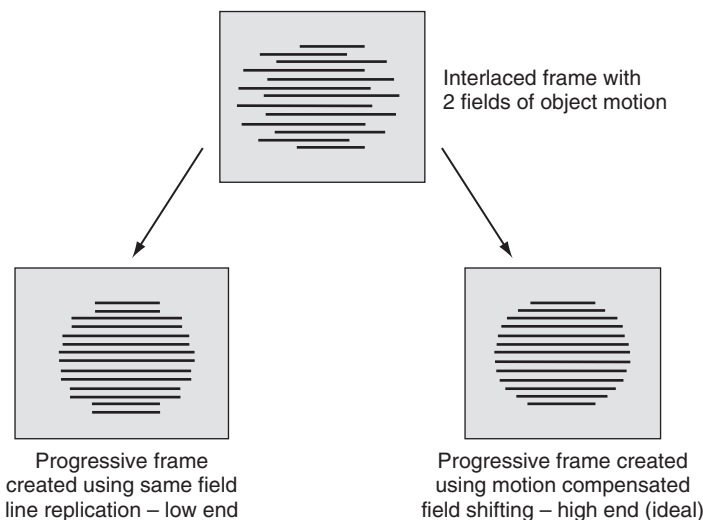
Recall that an interlaced image is composed of two woven fields, with the second one being time shifted compared to the first. Each field has half the spatial resolution of the complete frame as a first-order approximation. If the scene has object motion, the second field’s object will be displaced. Figure 11.3 shows the “tearing” between fields due to object displacement. Fortunately, the brain perceives an interlaced picture with a common amount of interfield image motion just fine. However, to print a single interlaced frame without obvious tearing artifacts or to convert between scanning formats, we often need to translate the image to a progressive one. This process is referred to as deinterlacing, as the woven nature of interlace is, in effect, unwoven. Ideally, the resulting progressive video imagery should have no interlaced artifact echoes. Unfortunately, this is rarely the case.

So what methods are used to unweave an interlaced image? A poor man’s converter may brute-force replicate or average select lines, fields, or frames to create the new image. These are not sophisticated techniques and often yield poor results. For example, Figure 11.14 (top) shows an interlaced image frame with field tearing. The bottom left image in Figure 11.14 is a converted progressive image using simple same-field line doubling. It is obvious that the spatial resolution has been cut in half.

At the high end of performance, motion-compensated deinterlacing measures the field-to-field motion and then aligns pixels between the two video fields to maximize the vertical frame resolution. The lower right image in Figure 11.14 illustrates an ideally deinterlaced image. There are countless variations on these themes for eking out the best conversion quality, but none guarantee perfect conversion under all circumstances. See (Biswas 2003) for a good summary of the methods.

### 11.7.2 Standards Conversion

The news department just received a breaking story in the PAL video format. It needs to be broadcast immediately over a NTSC system. Unfortunately, the



**FIGURE 11.14** *Interlaced to progressive conversion examples.*

two scanning standards are incompatible. Converting from one H/V/T scanning format to a second H/V/T format is a particularly vexing problem for video engineers (item #3 in the list given earlier). Translating the 25-frame PAL story to a 30-frame NTSC video sequence requires advanced video processing. This operation is sometimes referred to as a “standards conversion” because there is conversion between two different video-scanning standards. The PAL/NTSC conversion requires 5 new frames be manufactured per second along with per frame H/V resizing. Where will these new frames come from? Select lines, fields, or frames may be replicated or judiciously averaged to create the new 30 FPS video. Using motion tracking with line interpolation, new frames may be created with image positioning averaged between adjacent frames.

Reviewing Table 11.1, we see 10 different standards (plus the 1,000/1,001 rate variations) for a total of 15 distinct types. There are 210 different combinations of bidirectional conversions just for this short list. Yes, image-processing algorithms and implementation are becoming more important as scanning standards proliferate worldwide. In widespread use are SD-to-HD upconversion and HD-to-SD downconversion products. Several vendors offer standards converter products.

### 11.7.3 Compressed Domain Processing

Before leaving the theme of video processing, let us look at item #8 in the list given earlier: the hot topic of compressed domain processing (CDP). Video compression is covered in the next section, but the basic idea is to squeeze out any image redundancy, thereby reducing the bit rate. It is not uncommon for facilities to record, store, and play out raw MPEG streams. However, working natively

with MPEG and manipulating its core images are not easy. For example, the classic way for a TV station to add a flood watch warning message to the bottom of a MPEG broadcast video is to decode the MPEG source, composite the warning message, and then recode the new picture to MPEG. This flow adds another generation of encoding and introduces an image quality hit—often unacceptable. With CDP, there is no or very little quality hit in order to add the message. It is inserted directly into the MPEG data stream using complex algorithms that modify only the new overlay portion of the compressed image.

This is an overly simplified explanation for sure, but the bottom line is CDP is a powerful way to enhance MPEG streams at cable head ends and other pass-through locations. Several vendors are offering products with graphical insert (e.g., a logo) functions. Predictably, video squeeze backs and other operations will be available in a matter of time. Watch this space!

Well, that is the 1,000-foot view of some common video processing applications and tricks. Moving on, if there is any one technology that has revolutionized A/V, it is compression. The next section outlines the main methods used to substantially reduce A/V file and stream bit rates.

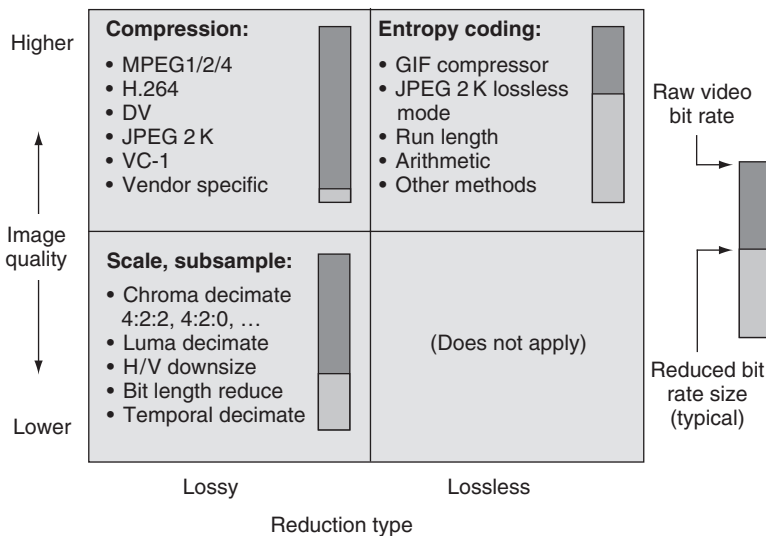
## 11.8 A/V BIT RATE REDUCTION TECHNIQUES

The data structures in a digital A/V system are many and varied. In addition to uncompressed video, there are several forms of scaled and compressed video. The following four categories help define the space:

1. Uncompressed A/V signals
2. Scaled uncompressed video in H/V/T dimensions, luma decimation, chroma decimation—lossy
3. Lossless compressed A/V
4. Lossy compressed A/V

Uncompressed video is a data hog from a bandwidth and storage perspective. For example, top-end, raw digital cinema resolution is  $\sim 4\text{ K} \times 2\text{ K}$  pixels, 24 FPS, yielding  $\sim 7$  Gbps data rate. A standard definition 4:4:4 R'G'B' signal has an active picture area bit rate of  $\sim 311$  Mbps. It is no wonder that researchers are always looking for ways to reduce the bit rate yet maintain image quality. Methods #2, #3, and #4 in the preceding list are bit rate reduction methods and are broadly rated against image quality in Figure 11.15. To better understand Figure 11.15, we need to define the term *lossy*.

Lossy, in contrast to lossless, is used to describe bit rate reduction methods that remove image structure in such a way that it can never be recovered. This may sound evil, but it is a good trick if the amount of image degradation is acceptable for business or artistic purposes. The lower left quadrant of Figure 11.15 rates quality versus bit reduction for method #2 given earlier. Chroma decimation is generally kind to the image and was discussed in Section 11.5.4.



**FIGURE 11.15** Video data rate reduction examples.

If luma decimation is used, it is much easier to notice artifacts. H/V scaling is done to reduce overall image size. For example, scaling an image to  $H/2 \times V/2$  saves an additional factor of 4 in bit rate, but the image size is also cut by 4. Another trick is to brute-force reduce the Y'CrCb bit length per component, say from 10 to 8 bits. This causes some visual banding artifacts in areas of low contrast. The last knob to tweak is frame rate. Reducing it saves a proportional amount of bits; keep going and the image gets the jitters.

The bit rate meter for this method shows a typical 3:1 savings for a modest application of scaling and decimation with virtually no loss of visual quality. Although these methods reduce bit rate, they are not classed as video compression methods. True compression techniques rely on mathematical cleverness, not just brute-force pixel dropping.

The next trick in the bit rate reduction toolbox is shown in the upper right quadrant of Figure 11.15. Lossless encoding squeezes out image redundancies, without sacrificing any quality. This class of rate reduction is called entropy coding after the idea that bit orderliness (low entropy) can be detected and coded with fewer bits. A good example of this is the common GIF still-image file compressor. It is based on the famous Lempel-Ziv-Welch (LZW) algorithm. GIF is not ideal for video but is the basis for similar methods that are applied to video. JPEG 2000 (JPEG2K) has a lossless mode supporting up to 12 bits/pixel encoding, resulting in outstanding quality. Run length coding is employed by MPEG encoders and other compressors to condense strings of repeating digits. Arithmetic coding is used by the H.264 video compressor and others to further pinch down the data rate by locating and squeezing out longer term value orderliness. Most lossless

coders achieve bit savings in the 25–50 percent range, but the results are strongly content dependent.

There is no guarantee that a pure lossless encoder will find redundancies, so the worst-case reduction factor may be zero. For example, losslessly coding a video of pure noise results in a bigger file due to coding overhead. As a result, in practice standalone lossless video coders are not common. However, lossless techniques are used liberally by traditional lossy video compressors (the upper left quadrant of Figure 11.15). A lossy compressor uses lossless techniques to gain a smidgen more of overall efficiency.

The real winner in bit rate reduction is true video compression (upper left quadrant of Figure 11.15). It has the potential to squeeze out a factor of  $\sim 100$  with passable image quality. Consumer satellite and digital cable systems compress some SD channels to  $\sim 2.5$  Mbps and exceed the 100:1 ratio of savings. Web video compression ratios reach  $>1,000:1$  but the quality loss is obvious. To get these big ratios, compressors use a combination of scaling, decimation, and lossy encoding.

However, some studio-quality codecs compress by factors as small as 3–6 with virtually lossless quality. A 20:1 reduction factor yields a  $\sim 15$  Mbps SD signal, which provides excellent quality for many TV station and A/V facility operations. Video compression is discussed in Section 11.8.2, but first let us look at some notes on audio bit rate reduction.

### 11.8.1 Audio Bit Rate Reduction Techniques

The four quadrants of Figure 11.15 have corresponding examples for audio bit rate reduction. Uncompressed audio formats are commonly based on the 16/20/24 bit AES/EBU (3.072 Mbps per stereo pair, including overhead bits) format or WAV format.

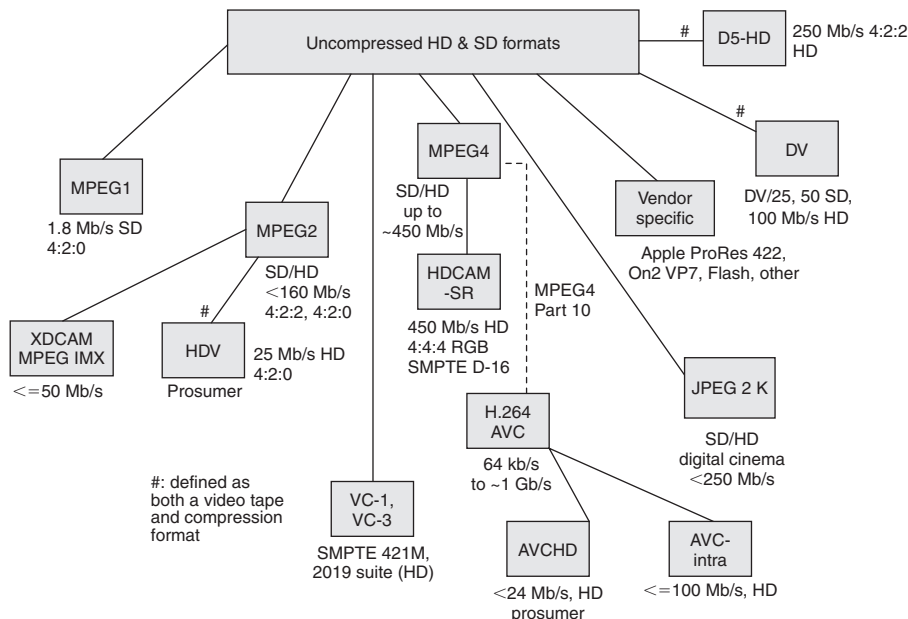
The EBU's Broadcast WAV format is a professional version of the ubiquitous WAV sound file format but contains additional time stamp and sync information. Lossless audio encoding is supported by the MPEG4 ALS standard, but it is not commonly used. One form of audio scaling reduces the upper end frequency limit, which enables a lower digital sample rate with a consequent bit rate reduction. This is a widespread practice in non-professional settings. Finally, audio compression is used in all forms of commercial distribution and some production. (Bosi 2002) gives a good overview of audio encoding and associated standards, including MP3, AAC, AC3, and other household names. These are distribution formats for the most part. Dolby-E is used in professional settings where quality is paramount. However, let us concentrate on video compression for the remainder of the discussion.

### 11.8.2 Video Compression Overview

When it comes to compression, one man's redundancy is another man's visual artifact, so the debate will always continue as to how much and what kind of

compression is needed to preserve the original material. There are three general classes of compression usage. At the top end are the program producers. Many in this class want the best possible quality and usually compress very lightly (slight bit rate reduction) or not at all. Long-term archive formats are often at high-quality levels. The next class relates to content distribution. In this area, program producers distribute materials to other partners and users. For example, network TV evening newscasts are sent to local affiliates using satellite means and compressed video. The level of compression needs to be kept reasonably high and is often referred to as mezzanine compression. Typically, HD MPEG2 programming sent at mezzanine rates range from 35 to 65 Mbps encoded bit rates. The third class is end-user consumption quality. This ranges from Web video at ~250 Kbps to SD-DVD at <10 Mbps to HDTV at 15–20 Mbps to digital cinema at <250 Mbps.

Figure 11.16 shows the landscape of the most common (there are many more) video compression formats in use. Why so many? Well, some are proprietary, whereas others are standardized. Some are designed for SD only, some for HD only, and some are both. A few are defined as a videotape and compression format, and some are improvements in a long line of formats. For example, MPEG1 begat MPEG2, which begat MPEG4, which begat MPEG4 part 10 (same as H.264, ITU-T spec, also called AVC, Advanced Video Codec). In Figure 11.16, the approximate maximum compressed data rate is given



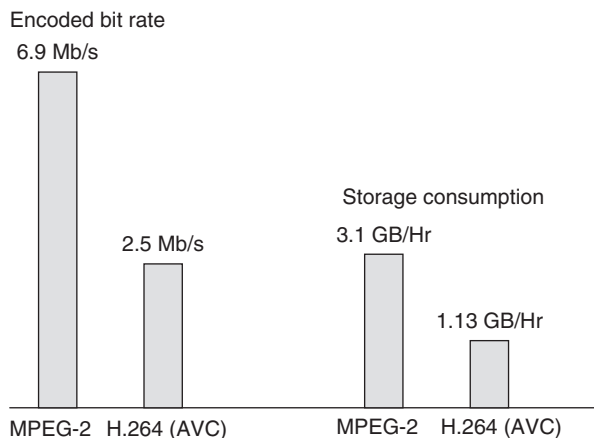
**FIGURE 11.16** Popular compressed video formats.

alongside each format box; the values are only a general guide. As a common rule, the compressors accept 8 bits of video component in the Y'CrCb format. However, several of the HD formats support 10-bit encoding. The fidelity range extensions (FRExt) of H.264 support encoding rates up to  $\sim 1$  Gbps for very high-end cinema productions. Most encoders apply some form of chroma and/or luma decimation before the actual compression begins. The compressors in Figure 11.16 are video-only formats for the most part. Audio encoding is treated separately.

Many of these formats are used in professional portable video cameras and/or tape decks. As such, many are editable and supported by video servers for playback. Most cameras in 2009 capture video on optical disk, HDD, or Flash memory, and 32 and 64 GB Flash cartridges are common. Moving forward, and as memory cost drops, Flash has the edge in terms of high recording rates, fast offload rates, ruggedness, and small size.

Figure 11.17 compares legacy MPEG2 against H.264 coding efficiency for the same SD source material (Sullivan 2004). This test demonstrates the superiority of H.264 for both bandwidth and storage usage. Here, we see an improvement of 2.75:1 for H.264, although this ratio is a function of source material; it is not a constant. Expect to see satellite TV, cable, and Telco operators (IPTV) using H.264 in their next generation of program delivery.

A best practice is to encode strictly and decode tolerantly. What does this mean? When encoding video strictly, apply to the standard in terms of what is legal. When decoding video, apply loosely to the standard. Of course, devices should decode materials that have been strictly encoded but also decode materials that may have deviated from the standard. If possible, rarely terminate a decode session (or generate useless video) based on some out-of-spec parameter. Decoders that do their best are always well respected.



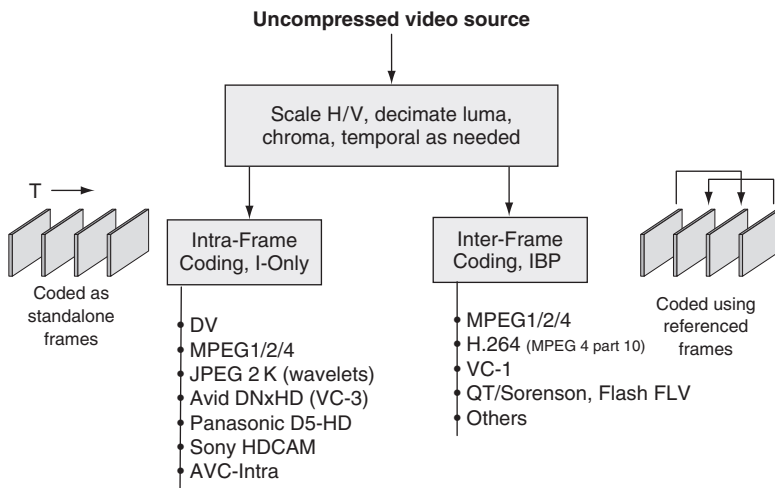
**FIGURE 11.17** Comparing codec efficiencies.

### 11.8.3 Summary of Lossy Video Compression Techniques

In general, there are two main classes of lossy compression: *intraframe* and *interframe* coding. Intraframe coding processes each frame of video as stand-alone and independent from past or future frames. This allows single frames to be edited, spliced, manipulated, and accessed without reference to adjacent frames. It is often used for production and videotape (e.g., DV) formats. However, interframe coding relies on exploiting temporal redundancies between frames to reduce the overall bit rate. Coding frame #*N* is done more efficiently using information from neighboring frames. Utilizing frame redundancies can squeeze a factor of two to three better compression compared to intra-only coding. For this reason, the DVD video format and ATSC/DVB transmission systems rely on intraframe coding compression. As might be imagined, editing and splicing interframes are thorny problems due to their interdependencies. Think of interformats as offering more quality per bit than intraformats.

Figure 11.18 illustrates a snapshot of common commercial video formats segregated according to type. This is representative of common formats in use, but not all methods are listed. Intraformats are sometimes called *I frame only*, signifying that only intraframes are coded.

Interformats are sometimes called *IBP* or *long GOP* (group of pictures), indicating the temporal association of the compression. The IBP and GOP concepts are described in short order. Note that some I frame-only formats such as JPEG 2000 and DV use only the intramode, whereas others such as MPEG are designed for either mode. In general, interformat compressors can also work in intramode when required. JPEG2K is unique because it has modes for both still picture and video compression. When a JPEG format codes video, it is sometimes called Motion-JPEG.



**FIGURE 11.18** Snapshot of compression methods and coding formats.



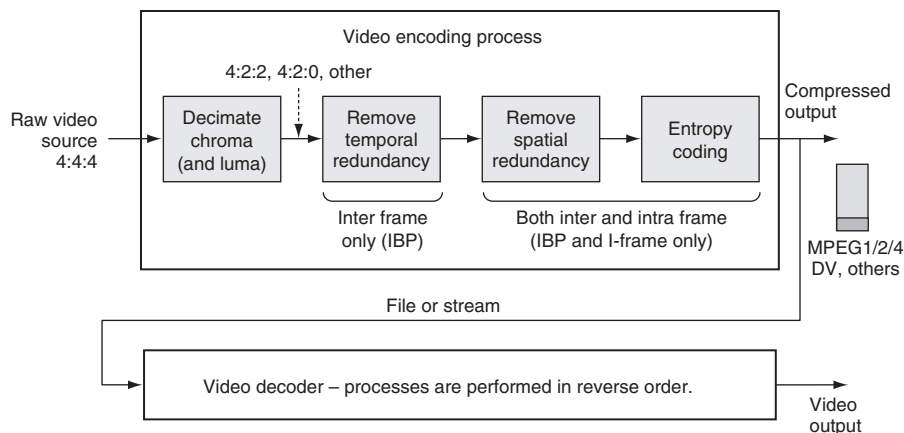
### 11.8.3.1 Intraframe Compression Techniques

Squeezing out image redundancies begs the question, “What is an image redundancy?” Researchers use principles from *visual psychophysics* to design compressors. They learn which image structures are discardable (redundant) and which are not without sacrificing quality. Despite its rather Zen-sounding name, visual psychophysics is hardcore science. It examines the eye/brain ability to detect contrast, brightness, and color and to make judgments about motion, size, distance, and depth. A good compressor reduces the bit rate with corresponding small losses—undetectable ideally—in quality.

Of all the parameters, reducing high-frequency detail and chroma decimation reap the most bit savings while maintaining image integrity. These techniques remove the spatial redundancy from the image. Scaling luma or reducing frame rates is usually bad business, and quality drops off suddenly. Figure 11.19 outlines the processes (ignore the second, temporal step for the moment) needed to compress an intraframe video sequence. Each process contributes a portion of the overall compressed bit savings. The order of compression is as follows:

- Decimate chroma and luma in some cases. This provides a lossy bit savings of 50 percent for 4:2:0 scaling.
- Remove high-frequency spatial detail using a filter-like operator and quantize the result. This step saves as much as needed. Pushing this lossy step too far results in obvious visual artifacts.
- Code for lossless entropy. This step saves an additional 30 percent on average.

For the middle step, two methods take the lead: transform based (discrete cosine transform, or DCT) and wavelet based. Both are filter-like operations



**FIGURE 11.19** The basic encoding and decoding processes.

for reducing or eliminating higher frequency image terms, thus shrinking the encoded bit rate. High-frequency terms arise due to lots of image structure. For example, a close-up image of a plot of grass has significantly more high-frequency terms than, say, an image of a blue wall. The DCT-based filter is used by nearly all modern video compressors. JPEG2K is one major exception; it uses the wavelet method. Mountains have been written about spatial compressing methods. For good overviews, see [Bhaskaran 1997], [Symes 2003], or [Watkinson 2004].

### THE DCT IN ACTION



The DCT is a mathematical device used to transform small pieces of the image domain into an approximation of their frequency domain representation. Once transformed, high-frequency spatial terms can be zeroed out. Plus, other significant spectral lines are quantized to reduce their bit consumption—these steps are lossy. The DCT

is tiled across an image frame repeatedly until all its parts have been processed. The resulting terms are entropy encoded and become the “compressed image bit stream.” At the decoder, an inverse DCT is performed, and the image domain is restored, albeit with some loss of resolution.

Intraframe methods encode each video frame in isolation with no regard to neighboring frames. Most video sequences have significant frame-to-frame redundancies, and intramethods do not take advantage of this; enter interframe encoding.

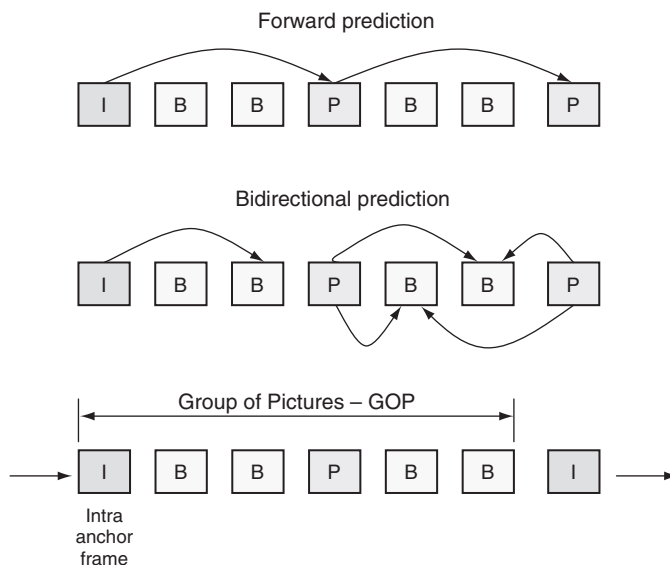
#### 11.8.3.2 Interframe Compression Techniques

Interframe compression exploits the similarities between adjacent frames, known as *temporal redundancy*, to further squeeze out the bits. Consider the talking head of a newscaster. How much does the head move from one video frame to the next? The idea is to track image motion from frame to frame and use this information to code more intelligently.

In a nutshell, the encoder tracks the motion of a block of pixels between frames using *motion estimation* and settles on the best match it can find. The new location is coded with motion vectors. The tracked pixel block will almost always have some residual prediction errors from the ideal, and these are computed (*motion compensation*). The encoder’s motion vectors and compressed prediction errors (lossy step) are packaged into the final compressed file or stream.

The decoder uses the received motion vectors and prediction errors and recreates the moved pixel block exactly—minus the loss when the error terms were compressed. Review Figure 11.19 and note the “remove temporal redundancy” step for interframe coding.

Various algorithms have been invented to track pixel group motion—not necessarily discernible objects. This step is one of the most computationally intensive for an encoding process. This form of encoding is sometimes referred



**FIGURE 11.20** *Interframe sequencing examples.*

to as IBP or long GOP encoding (see Figure 11.20). A GOP is a sequence of related pictures, usually from 12 to 15 frames in length for SD materials. Here, the I frame is a standalone anchor intraframe. It is coded without reference to any neighboring frames. The B frame is predicted bidirectionally from one or two of its I or P neighbors. A B frame has fewer coded bits than an I frame. The P frame is a forward-predicted frame from the last anchor frame (I or P) and has more bits than a B frame but fewer than an I frame.<sup>7</sup> Note that a B or P frame is not the poorer cousin of an I frame. In effect, all frames carry approximately equal image quality due to the clever use of motion estimation and compensation.

Most MPEG encoders use interframe techniques, but they can also be configured to encode using only intraframe methods. Users trade off better compression efficiency against easy editing, splicing, and frame manipulation when choosing between IBP and I-only formats. A general rule of thumb for production-quality compression is that IBP formats use about 3–4 times fewer bits compared to a pure I-only encoding. For example, an I-only VC-3 encoding at 140 Mbps has about the same visual quality as the Sony XDCAM HD 50 Mbps IBP format.

Researchers are constantly looking for ways to improve compression efficiency, and the future looks bright for yet another round of codecs. In fact, initial work has begun on the tentative H.265 compressor, which holds out the promise of another ~50 percent bit rate reduction within 4 to 5 years.

<sup>7</sup> These are general IBP bit rate allocations and may differ depending on image content.

## 11.9 VIDEO TIME CODE BASICS

Last but not least is the concept of time code. This “code” is used to link each frame of video to a time value. The common home VCR and DVD use displayed time code to show the progress of a program. The format for professional time code is HH:MM:SS:FF, where H is the hours value (00 to 23), M is the minutes value, S is the seconds value, and F is the frame value (00 to 29 for 525i). A time code of 01:12:30:15 indicates the position of the video at 1 hr, 12 min, 30 s, and 15 frames. If the first frame of the video is marked 00:00:00:00, then 01:12:30:15 is the actual running time to this point. A time code rollover example of 01:33:58:29 is followed by 01:33:59:00.

Accurate A/V timing is the lifeblood for most media facilities. For example, time code is indispensable when queuing up a VTR or video server to a given start point. NLEs use time code to locate and annotate edit points. At a TV station, program and commercial playback is strictly controlled by a playout schedule tied to time code values for every second of the day. SMPTE 12M defines two types of time code: linear (also called longitudinal) time code (LTC) and vertical interval time code (VITC). LTC is a simple digital stream representing the current time code value along with some additional bits. Historically, the LTC low-bit rate format is carried by a spare audio track on the videotape. VITC is a scheme in which time code bits are carried in the vertical blanking interval of the video. This convenient scheme allows the time code to always travel with the video signal and is readable even when the video is in pause mode. In both cases, the time code display format is the same.

Although LTC and VITC were designed with videotape in mind, non-tape-based devices (video servers) often support one or both formats. Time code is carried throughout a facility using an audio cable with XLR connectors. Many vendors offer time code generators that source LTC referenced to a GPS clock. As a result, it is possible to frame accurately sync videos from different parts of a campus or venue and guarantee they are referenced to a common time code value.

### 11.9.1 Drop Frame Time Code

With 25 FPS video (625i) there is an integer number of frames per second. However, the 525i frame rate is  $\sim 29.97$  FPS (see Section 11.5.7), and there is not an integer number of frames per second. Standard time code runs exactly 30 FPS for 525i video. A time code rate of 30 FPS instead of  $\sim 29.97$  creates a 3.6 s error (an extra 108 video frames) every 60 min. This is light years in video time, so we need a way to correct for this. One way to effectively slow down the 30 FPS time code signal is by skipping one time code value every 33.3333 s. This amounts to dropping 108 code points per hour. Now, the 30 FPS time code signal *appears* to run at  $\sim 29.97$  FPS. In reality, this is accomplished by dropping frame code numbers 00:00 and 00:01 at the beginning of every minute except for every 10th minute.

Importantly, no actual video frames are dropped, only the time code sequence is modified. If all this is confusing, at least remember that there are two forms of time code: *drop frame* and *non-drop frame*. In the 525i/29.97 world, drop frame is popular, whereas in the 625i/25 world, non-drop frame is popular. With 25 FPS material, there is no need to play tricks with the time code values.

Looking ahead, SMPTE is developing a new time code system using a Time Related Label (TRL). The TRL contains the time code field plus additional information (either implicit or easily computable), including frame count from start of material, instantaneous frame rate (FPS), date, and time of day (TOD). The new time label will eventually replace the current bare bones time code field specified by SMPTE 12M. In addition, the overall solution will replace existing methods to synchronize A/V signals across devices and geographies.

### 11.10 IT'S A WRAP—SOME FINAL WORDS

With this brief overview, you should be conversant with the basics of A/V technology. These themes form the quilt that touches most aspects of professional A/V systems. A key thesis in this book is leveraging IT to move A/V systems to new heights of performance, reliability, and flexibility. By applying the lessons learned in this chapter, plus lessons from the others, you should be well equipped to understand, evaluate, and forecast trends in the new world of converged AV/IT systems.

### References

- Bhaskaran, V., et al. (1997). *Image and Video Compression Standards* (2nd edition): Kluwer Press.
- Biswas, M., & Nguyen, T. (May 2003). *A Novel De-Interlacing Technique Based on Phase Plane Correlation Motion Estimation*: ISCAS, [http://videoprocessing.ucsd.edu/~mainak/pdf\\_files/asilomar.pdf](http://videoprocessing.ucsd.edu/~mainak/pdf_files/asilomar.pdf).
- Bosi, M., Goldberg, R. E., & Chiariglione, L. (2002). *Introduction to Digital Audio Coding and Standards*: Kluwer Press.
- Jack, K. (2001). *Video Demystified, A Handbook for the Digital Engineer* (3rd edition): Newnes.
- Poynton, C. (2003). *Digital Video and HDTV, Algorithms and Interfaces*. NYC, NY: Morgan Kaufmann.
- Sullivan, G. J., et al. (August 2004). The H264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extension. *SPIE Conference on Applications of Digital Image Processing XXVII*.
- Symes, P. (October 2003). *Digital Video Compression*. NYC, NY: McGraw Hill/TAB.
- Watkinson, J. (November 1, 2004). *The MPEG Handbook* (2nd edition). Burlington MA: Focal Press.

# Appendix A: Fast Shortcuts for Computing $2^N$

There are 10 kinds of people in this world: those who understand binary math and those who do not. Many of us live in a binary world despite our base ten arithmetic system. Often we need to convert a  $2^N$  notation to its decimal equivalent. Quick, what is the maximum addressable file size given a 32-bit number system? Well, that is  $2^{32}$ , and with a little calculator math, we come up with 4,294,967,296 elements, or approximately 4.3 billion. Is there a faster way to calculate the value? If we apply a few tricks, we can quickly approximate the value to within 7.5 percent worst case. This is good enough for many practical uses.

The basic idea is to split the exponent ( $N$ ) into two parts  $n1$  and  $n2$ , such that  $n1 + n2 = N$ . Let  $n1$  equal the tens value of  $N$ . So if  $N = 24$ , then  $n1 = 20$  and  $n2 = 4$ . Now the trick for speedy computation is to approximate the value of  $2^{n1}$  as a round number. We can do this because  $2^{10} = 1,024$  and  $2^{20} = 1,048,576$  for example. So let us approximate  $2^{10}$  to be 1,000 and  $2^{20}$  to be 1,000,000 (1 million, or  $10^6$ ) and  $2^{30}$  to be 1 billion. The max error is only 7.5 percent and often less.

To compute  $2^{24}$ , we set  $n1 = 20$  and  $n2 = 4$ . The approximated value is  $10^6$  times  $2^4$ , which is 16 million using  $2^N = 2^{n1} * 2^{n2}$  where  $2^{n1} = 1$  million for this example. The exact value is 16,777,216, so our approximation is within 5 percent. Of course the  $2^{n2}$  values should be exact, but there are only nine of them to memorize. Most of us likely already know these values (2, 4, 8, 16, 32, 64, 128, 256, 512), especially if we smiled after reading the first sentence of this appendix. Table A.1 shows some other examples. Happy approximating!

Table A.1		
<i>N</i>	$2^n$	Approx.
2	4	—
4	16	—
10	1,024	1,000 = 1K
12	4,096	4 * 1,000
20	1,048,576	1K * 1K = 1M
24	16.7M	1K * 1K * 16 = 16M
30	0.1B	1K * 1K * 1K = 1B
32	4.3B	1K * 1K * 1K * 4 = 4B

*For this table,  $K = 10^3$ ,  $M = 10^6$ , and  $B = 10^9$ .*

# Appendix B: Achieving Frame Accuracy in a Non-Frame Accurate World

Keeping video signals synchronized is a time-honored process. Ideally, all routed video signals in a facility are raster time-aligned horizontally and vertically and with a known frame time code. The basic method used to synchronize a target video signal is to apply a correcting time shift and thereby align it with a provided master reference signal. In most cases, the vertical and horizontal timing of the target signal is altered to agree with the provided video house reference. There is usually a one or more frame delay (input to output delay of corrector) to accomplish the time alignment.

It is not difficult to imagine how this process works. The input signal needing correction is fed into a “frame synchronizer” (standalone box or internal device process). It is written to a digital frame memory (sometimes called an elastic buffer) at the input rate and timing. Next, a separate process reads from the same memory to create a new signal for output. The output timing is aligned to the provided master reference signal. As long as the input data rate and output data are equal on average, then the buffer will not overflow or underflow. There is a clean separation of the two processes and any output H/V timing relation may be produced independent of the input signal timing.

Frame syncs are applied to many real-world problems. For example, a received satellite video feed will not naturally be time-aligned with a facility’s internal timing reference, so the received signal must be frame synced to align it. Incidentally, most video facilities have a video reference signal that is distributed to all A/V gear. When the reference signal is used, all device A/V I/O can be aligned to the master reference signal. In some cases, the input frame rate and output frame rate are not precisely equal. Over time (sometimes many hours or days) the frame sync needs to duplicate or drop a frame to keep the buffer from underflowing or overflowing. If this is done only on occasion, then the frame jumps are rarely noticed. With GPS worldwide timing, frame buffers rarely exceed their limits. Also, the audio associated with an aligned video must be delayed or resampled for lip-sync agreement. An infrequent audio click may be detected due to the frame drop/add operation.



When signals are streamed across digital networks, they will invariably drift in time due to the introduced delay and jitter and may need to be frame corrected at some point in the workflow. Frame synchronizing is an important process in the real world of A/V systems. For more information on new techniques for A/V signal synchronizing, do a Web search for documents published by the "SMPTE EBU Joint Task Force on Timing & Synchronization."

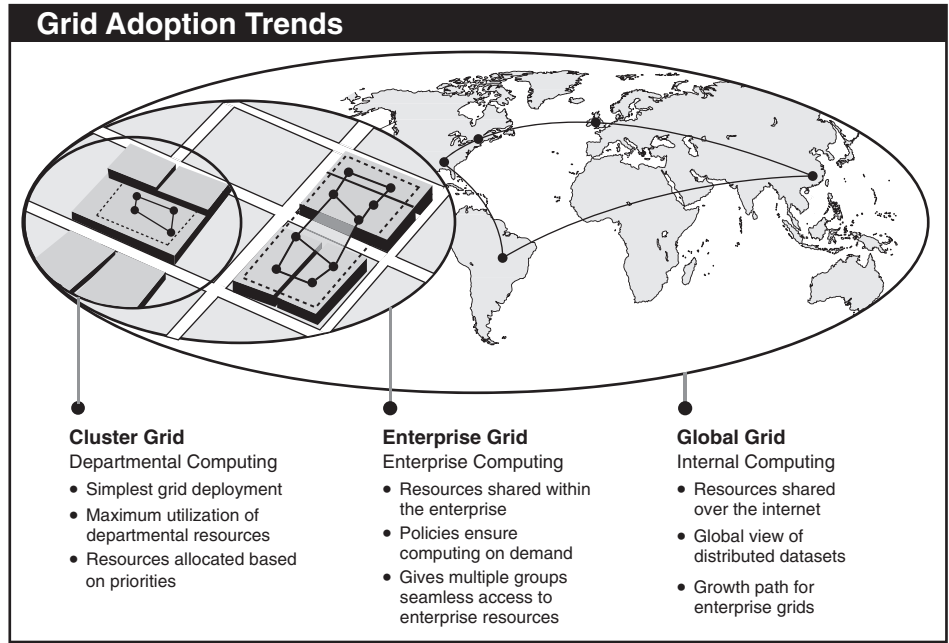
# Appendix C: Grid, Cluster, Utility, and Symmetric Multiprocessing Computing

*Grids* leverage underutilized CPU power from machines connected to the Internet (or a private network) to run various applications. *Clusters* are a formal collection of CPU resources (servers normally) connected to a private network for use in computing. At the most fundamental level, when two or more dedicated computers are used together to solve a problem, they are considered a cluster. *Utility computing* hides the complexity of resource (computers, networks, storage, etc.) management and provides what business wants: utilization on demand. Finally, *symmetric multiprocessing* (SMP) harnesses the power of  $N$  CPUs running in parallel. All of these techniques may be applied to A/V computational problems. The following sections review the four methods.

## C.0 GRID COMPUTING

Grid computing is a distributed environment composed of many (up to millions) heterogeneous computers (nodes), each sharing a small burden of the computational load. The power of the Internet enables thousands of cooperating PCs to be connected in a mesh of impressive computational power. By some estimates, most desktop PCs are busy only 5 percent of the time, so why not put these underutilized resources to better use? Even some servers are idle for a portion of each day, so a grid assembles and manages unused storage and CPU power on networked machines to run applications. The node application is run in the background and does not interfere with the primary user applications on the machine.

The potential of massive parallel CPU capacity is very attractive to a number of industries. In addition to pure scientific needs, bio-med, financial, oil exploration, and others are finding grids to be of value. If the computational solution is written with a grid in mind, it will run  $N$  times faster when  $N$  nodes are working in parallel on average. Figure C.1 shows trends ranging across local to enterprise to global grid computing. Grid.org ([www.grid.org](http://www.grid.org)) reports that 2.5 million computers from 200 countries are now tied into one grid for the



**FIGURE C.1** Grid computing environments. Concept: Sun.

purposes of cancer research, searching for extraterrestrial life (SETI project at <http://setiathome.ssl.berkeley.edu>), and other noble causes. For local or enterprise grids, it is possible to define a QoS for the computing power. In a global sense, a guaranteed QoS is problematic, as most of the Internet-connected nodes are voluntary citizens.

The Open Grid Forum (OGF) is a community of users, developers, and vendors leading the global standardization effort for grid computing. See [www.ogf.org](http://www.ogf.org) for a great collection of white papers and other references.

## C.1 GRID COMPUTING AND THE RIEMANN ZETA FUNCTION

One particularly interesting use of grid computing is related to finding the complex zeros of the Riemann zeta function. In 1859, Bernhard Riemann wrote a mathematical paper showing a formula for all the prime numbers less than  $N$ , the so-called prime number theorem (PNT). In this paper he asserted that all the zeros of the zeta function have a real part of  $1/2$ . This is the Riemann Conjecture. Since then, mathematicians have attempted to prove this, but without success. It is one of the most difficult problems facing mathematicians today. There is a \$1 million prize on the block if you can prove the conjecture (see [www.claymath.org/millennium](http://www.claymath.org/millennium)). Sebastian Wedeniwski of IBM used a grid

configuration of  $\sim 11,600$  nodes to find the first trillion complex zeros of the zeta function. Results show all zeros have a real part of  $1/2$ . This is not a proof of the Riemann Conjecture, but it provides tantalizing data to the affirmative.

To learn more about Riemann, the zeta function, complex zeros, the PNT, loads of interesting stories, and the quest to solve one of the greatest unsolved problems in mathematics, pick up a copy of the very entertaining book *Prime Obsession* by John Derbyshire. He treats the problem as a mystery and leads the reader through beautiful gardens of advanced, yet accessible, math to illuminate it.

## C.2 CLUSTER<sup>1</sup> COMPUTING

A *cluster* is a common term meaning independent computers combined into a unified system through software and networking. Clusters are typically used for high-availability (HA) or high-performance computing (HPC) to provide greater computational power than a single computer can provide. The Beowulf project began in 1994 as an experiment to duplicate the power of a supercomputer with commodity hardware and software (Linux).

Today, Beowulf cluster technology is mature and used worldwide for serious computational needs. The commodity hardware can be any of a number of mass-market, standalone compute nodes. This ranges from two networked computers sharing a file system to 1,024 nodes connected via a high-speed, low-latency network. Performance is improved proportionally by adding machines to the cluster.

Class I clusters are built entirely using commodity hardware and software using standard technology such as SCSI/ATA and Ethernet. They are typically less expensive than class II clusters, which may use specialized hardware to achieve higher performance. As Chapter 3B discussed, NAS clusters are very practical for A/V applications. Also, for high-end rendering and film effects work, a cluster is “without parallel.” Clusters are in daily use for computing the world’s most demanding scientific simulations.

## C.3 UTILITY AND CLOUD COMPUTING

Next, there is *utility computing*. In theory, utility computing gives companies greater utilization of data center resources more cost effectively. It is based on flexible computing, clusters, grids, storage, and network capacity that react automatically to changes in computing needs. Utility computing can be localized

---

<sup>1</sup> Some of the material in this section is paraphrased from the Beowulf Web site, [www.beowulf.org](http://www.beowulf.org).

(enterprise) or global (the Web). The data center of the future should be self-configuring, self-monitoring, self-healing, and invisible to end users.

In 2006, Amazon launched the EC2—Elastic Compute Cloud. This service sells “CPU cycles” over the Web. Users pay only for the instantaneous use of CPU cycles and consumed memory. There is no recurring subscription fee. The EC2 is being used by companies that have wide fluxes in computing needs but don’t desire to own/lease physical hardware. Scale and reliability are key attributes to EC2. Amazon also offers the S3 networked storage service, as do other companies. See <http://aws.amazon.com/ec2> for pricing and configuration.

3tera, Amazon, AT&T, BT, Google, Microsoft, and others have roadmaps to enable the “universal cloud computer.” Although not an authoritative fact, Google is rumored to have plans for ~2 million distributed servers with a combined storage capacity of ~5 million hard drives in 2009. Even if this estimate is high for 2009, it will likely be spot-on within a few years if Google continues to grow. It’s not difficult to imagine a global computer utility akin to the Edison electric grid for all to tap into on demand. Plug into it; use it without worries of where the resource came from or who manages its allocation and reliability.

The networked cloud computer would run cloudware (Software-as-a-Service, SaaS) to meet the needs of many business and some home users. Bye-bye desktop or enterprise server applications, hello cloudware. Will it happen? Yes—it is happening in 2009, but the scale is relatively small. For example, Salesforce.com currently offers over 700 SaaS applications. The SaaS phenomenon is in its early stages, analysts say. The research firm ITC predicts that SaaS companies, which earned \$3.6 billion in revenue in 2006, would earn \$14.8 billion by 2011.

For the cloud infrastructure, 3tera offers AppLogic, the first grid operating system designed for Web applications and optimized for transactional and I/O intensive workloads. AppLogic makes it easy, according to 3tera, to move existing Web applications onto a cloud grid without modifications.

The end game is “cloud-everything”—computing, storage, networking, applications, provisioning, management. This is our future, so be looking for it.

## C.4 SMP COMPUTING

The final method in our list to increase performance uses  $N$  tightly coupled CPUs. This is sometimes called symmetric multiprocessing. Commercial systems support 4 or 8 CPUs commonly, although 64 or more are possible. The programming model for a SMP system relies on dividing a large program into small threads spread out among the  $N$  CPUs. The CPUs share a common memory, so a computational speed up of  $N$  may occur under a best-case scenario. Linux and Windows support SMP as do other operating systems.

The most popular entry-level SMP systems use the x86 instruction set architecture and are based on Intel’s and AMD’s multi-CPU processors.

# Appendix D: The Information Flood—One Zettabyte of Data

It's no news that we are swimming in a flood of information. But, how much will we be inundated with? The Discovery Institute ([discover.org](http://discover.org)) published a white paper (1/2008) titled *Estimating the Exaflood*. In it, he predicted U.S. Internet traffic in 2015. To get a grasp of the results, we need to know what a Zettabyte is. Note that a Petabyte is  $10^{15}$  bytes, an Exabyte is  $10^{18}$  bytes, a Zettabyte is  $10^{21}$  bytes, and a Yottabyte is  $10^{24}$  bytes. The white paper estimates the following in 2015 for U.S. Internet traffic:

- Movie downloads and P2P file sharing could be 100 Exabytes.
- Video calling and virtual windows could generate 400 Exabytes.
- “Cloud” computing and remote backup could total 50 Exabytes.
- Internet video, gaming, and virtual worlds could produce 200 Exabytes.
- Non-Internet “IPTV” could reach 100 Exabytes and possibly much more.
- Business IP traffic will generate some 100 Exabytes.
- Other applications (phone, Web, email, photos, music) could be 50 Exabytes.

Note that video and rich media account for much of the totals. The sum is 1,000 Exabytes, or 1 Zettabyte. A close analog from chemistry is Avogadro's Number ( $6.023 \times 10^{23}$ , the number of atoms or units in 12 grams of carbon, a mole). So, a Zettabyte is about 0.0016 moles of bytes. Someday, the world-wide Internet will transport a mole of bytes per year. The U.S. Internet of 2015 will be at least 50 times larger than it was in 2006. Growth at these levels will require a “big bang” infusion of bandwidth, storage, and management capabilities throughout the Internet.

Outside Internet transport, according to IDC (Gantz), as of 2006 the total amount of digital data in existence was 0.16 Zettabytes. The same paper estimates that by 2010, the rate of digital data generated worldwide will be  $\sim 1$  Zettabyte per year. IDC predicts that this will be *twice* the amount of available

storage. By way of example, one single long-term experiment planned for the Large Hadron Collider in Geneva will create an amazing 300 Exabytes of data per year. So, as fast as vendors create storage, we will eat it up, and as Dickens's *Oliver Twist* said, we too will say, "Please, sir, I want some more."

## References

John F. Gantz et al, March 2007; *The expanding digital universe: A forecast of world-wide information growth through 2010*: IDC, Framingham MA 01701 USA.  
Gilder, George, *Estimating the Exaflood*, [www.discovery.org](http://www.discovery.org). Jan 29, 2008.

# Appendix E: 8B/10B Line Coding

When you are transmitting digital data across a LAN, WAN, or other link, it is important to design the link with the following features:

1. No average DC term (DC-free) when data are viewed over several bytes.  
So, only AC components are present. DC cannot pass through optical fiber.
2. Capability to allow for clock and bit recovery from received data stream.
3. Minimum frequency spectrum shape for greater cable reach.

Transmitting raw, uncoded user data bit streams will not meet any of these requirements. Why? It is likely for long strings of ones or zeros to occur, so conditions 1 and 2 are not met. Also, some data sequences have very high frequency components. Following are two different ways to meet the three conditions just given:

- Use a block code. One example encodes 8-bit user data as 10-bit valued code words. In its most basic form, 256 8-bit user values are mapped into 256 10-bit code words. This coding type, used by Fibre Channel, is called 8B/10B, although there are other popular ones, such as 8B/14B (used by the common CD) and 64B/66B (10 gigabit Ethernet). Using 8B/10B requires a 25 percent higher line bit rate due to the overhead of the code words. The 1G Ethernet line rate is 1.250 Gbps and the payload rate is 1 Gbps using 8B/10B coding. Each of the 256 code words has an equal number of ones and zeros for all but four cases, but a DC-free balance is maintained over several code words. Sometimes payload rates are quoted (as with GigE), and sometimes line bit rates are quoted (as with Fibre Channel). So, buyer beware. IBM patented this coding concept in 1984.
- Use a scrambler to randomize and statistically balance data. This approach does not expand the message as 8B/10B does, so it is more efficient. However, it does not guarantee a perfect DC-free balance, although blocks of reasonable size will be “nearly” balanced to high probability. Condition 3 is not strictly met, but scrambling does generate a predictable spectrum shape.



Both methods are in wide use. The common SDI link (SMPTE 259M) uses the scrambling method. Some pathological data patterns have been known to cause a poorly designed receiver to lose clock recovery and generate bands of colors instead of the payload image. Scrambling does not always eliminate certain problem sequences that 8B/10B, for example, can deal with effectively.

# Appendix F: Digital Hierarchies

Two main classes of telecom-based digital links are in worldwide use for transporting telephone and data signals. One system is called plesiochronous digital hierarchy (PDH) and is based on multiplexing numerous individual channels into higher rate channels. The PDH is not synchronous but “nearly synchronous.” The second link type depends on synchronous communications. The United States and Canada rely on SONET, whereas the rest of the world uses SDH. SONET stands for Synchronous Optical NETwork, and SDH is an acronym for Synchronous Digital Hierarchy. The ITU adopted SONET, with small variations, as the basis for the SDH. The SONET/SDH standards define a hierarchy of interface rates that permit different line rates to be multiplexed as with the PDH method.

The PDH uses clocks of different accuracies, whereas SONET transmits using a highly stable reference-clocking source for the entire SONET network. Both use different framing methods, line rates, line coding, and timing. Despite the many differences, there are common features to both systems:

- Each has a defined hierarchy of data rates (see later) for multiplexing lower rate signals of the same type.
- Each can carry telephone data payloads in multiples of DS0 (digital signal 0) at 64,000 bits per second.
- Many telecom carriers offer PDH (T1, T3) and SONET/SDH (OC-3, OC-12) links as WAN access connectivity to their network.
- Various data packaging methods are defined: IP packets over either, ATM over SONET/SDH, MPEG TS over either, and others.

The underlying technology of the PDH and SDH is complex and is not covered here. Table F.1 outlines the common naming and hierarchical relationships of the two methods.

North America, Europe, and Japan have chosen different basic PDH line rates, but the other aspects are identical. The J terminology is a colloquial naming scheme for Japan, and E is for Europe and T for North America. The physical links are copper based for T1/T3 and E1/E3 but are usually optical for higher rates. Rates beyond T3/E3/J3 are not in common use for WANs.

Table F.1 PDH Naming and Rate Relationships				
Line Speed	DSOs	North America	Europe	Japan
64 Kbps	1	—	—	—
1.544 Mbps	24	T-1	—	J-1
2.048 Mbps	32	—	E-1	—
6.312 Mbps	96	T-2	—	J-2
7.786 Mbps	120	—	—	—
8.448 Mbps	128	—	E-2	—
32.064 Mbps	480	—	—	J-3
34.368 Mbps	512	—	E-3	—
44.736 Mbps	672	T-3	—	—
97.728 Mbps	1,440	—	—	J-4
139.264 Mbps	2,016	—	—	—
139.268 Mbps	2,048	—	E4	—
274.176 Mbps	4,032	T-4	—	—
400.352 Mbps	5,760	T-5	—	—
565.148 Mbps	8,192	—	E-5	J-5

F.0 OPTICAL CARRIERS IN SONET AND THE SDH

In SONET and the SDH the optical standard is typically known by an OC-x number, where x is a multiple of the OC-1 rate of 51.84 Mbps. While the optical system is common, there are small differences in framing and bit rates between SDH and SONET. North America uses a STS-x (synchronous transport signal) format for frames (packets), whereas Europe uses a STM-x (synchronous transport module) format. Table F.2 lists the various SONET/SDH relationships.

Table F.2 SONET and SDH Relationships				
Optical Carrier	Line Data Rate	User Data Rate	SONET	SDH
OC-1	51.84 Mbps	49.536	STS-1	—
OC-3	155.52 Mbps	148.608	STS-3	STM-1
OC-9	466.56 Mbps	445.824	STS-9	STM-3
OC-12	622.08 Mbps	594.824	STS-12	STM-4
OC-18	933.12 Mbps	891.648	STS-18	STM-6
OC-24	1,244.16 Mbps	1,188.864	STS-24	STM-8
OC-36	1,866.24 Mbps	1,783.296	STS-36	STM-12
OC-48	2,488.32 Mbps	2,377.728	STS-48	STM-16
OC-192	9,953.28 Mbps	9,510.912	STS-192	STM-64
OC-768	40 Gbps	—	STS-768	STM-256
OC-3072	160 Gbps	—	STS-3072	STM-1024

## **F.1 CONCLUSION**

The PDH and SONET/SDH systems are the backbone of the telecom industry. Many telecom carriers offer A/V-friendly WAN services over T3/E3 and higher rates. IP data packaged over either method is a common offering. These methods plus pure Ethernet connectivity will continue to be the foundation of WAN technology worldwide.

This page intentionally left blank

# Appendix G: 270 Million—A Magic Number in Digital Video

The SDI link carries uncompressed SD video at 270 Mbps line rates. Interestingly, the SDI link can synchronously carry a digitally sampled 625/25 line structure signal (used in the PAL system) or the 525/29.97003 line structure signal (used in NTSC systems) at the same line rate of 270 Mbps. How is this possible, because the two TV systems have apparently unrelated sampling rates? History records that the 525 and 625 analog systems were developed independently of each other. However, the two digital systems have a common sampling rate of 270 Mbps. Let us see why.

In the 625 line system the horizontal line rate is  $625 \times 50/2 = 15,625$  lines per second (LPS, units in hertz). In the 525 line system, the LPS rate is  $525 \times 60/2 \times 1,000/1,001 = 15,734.2657734$ . Do the two line rates have a common relationship? By examining these two values, we get the following factorizations:

- $5^6 = 15,625$  (PAL)
- $(7 \times 25 \times 3) \times 30 \times (125 \times 8)/(7 \times 11 \times 13) = 15,734.265 \dots$  (NTSC)
- Dividing the second value by the first yields an exact integer ratio of 144/143. This relationship was not planned by the respective PAL/NTSC system designers, yet is of great benefit to us. Now,  $144 \times 15,625 \text{ Hz} = 143 \times 15,734.265 \text{ Hz} = 2.25 \text{ MHz}$  and this is the lowest common frequency from which both line rates can be derived. How may this value be used? All digitally sampled systems have a corresponding sample clock. This clock needs to be high enough (meet the Nyquist sample rate rules) to faithfully represent any analog video signal in digital form yet not be so high as to waste bandwidth and storage space. Given the bandwidth of SD video signals, a factor of 6 was deemed ideal, yielding a system sample clock of  $6 \times 2.25 \text{ MHz} = 13.5 \text{ MHz}$ . This is the basic sampling rate of SD digital video.

It turns out that each image RGB pixel set can be represented by a color difference component format. This digital format is referred to as Y'CrCb (luma and two chroma samples). Y'CrCb is derived from RGB pixels via mathematical

operations, albeit with some loss of image resolution. The SDI data structure format (component video mode) carries the Y'CrCb signal with 4:2:2 sampling. There is *one* Y' value and *one* chroma (either Cr or Cb) value per RGB sample set on average.<sup>1</sup> SDI supports 10-bit samples, so the luma/chroma sequence requires  $2 \times 10$  bits. The overall SDI line rate is  $13.5\text{MHz} \times 20 \text{ bits} = 270$  Mbps. This rate is usable for both 625 and 525 line TV systems. So indeed, 270,000,000 is a magic number in SD digital video systems.

One more point of interest: the ratio 144/143 between the 525 and the 625 systems is also a measure of the spatial/temporal information content difference between the two systems. One has more spatial resolution (625/25), whereas the other has more temporal resolution (525/29.97). As humans, we value both spatial and temporal quality when viewing a moving image. There are, of course, countless levels of nuance when evaluating image attributes, but at a high level it is apparent that the two systems' "image quality" differs by only 0.7 percent if the spatial and temporal resolutions are weighted equally.

---

<sup>1</sup> If this all sounds confusing, see Chapter 11 for more information on the Y'CrCb and 4:2:2 formats and their use in video systems.

# Appendix H: A Novel A/V Storage System

In high-end broadcast and postproduction environments, real-time, high-availability storage systems offer significant workflow advantages. This appendix reviews a new storage system architecture<sup>1</sup> designed specifically for the needs of A/V. The design blurs the lines between network attached storage (NAS) and storage area network (SAN) systems. It addresses a number of key needs, including support for a large number of real-time editing clients and I/O channels, high data availability, hot-swap components, heterogeneous clients (Windows and MacOS), and wide-range scalability in data rates and storage capacity.

Commercially available, enterprise-class NAS and SAN systems are hard pressed to meet these functional requirements. A NAS, in which client data pass through a single NAS server, does not scale easily and suffers from intrinsic data rate and latency constraints. Fibre Channel-based SAN topologies are also problematic and have limited client-attach flexibility and scalability. Finally, commercially available clustered file systems are not designed to support data requests with real-time QoS requirements.

## H.0 ARCHITECTURAL OVERVIEW

An entirely new networked file system and storage infrastructure was developed to overcome these deficiencies. The file system is unique in several ways. Every editing client and I/O channel requires consistent, glitch-free, A/V data streams delivered to/from storage. The file system arbitrates data transfer deadlines between clients and storage, ensuring clients complete transfers to storage within a prescribed time window. Without such functionality, a client's internal buffers would either overflow or starve. Further, the file system enables storage configurations to be modified dynamically, allowing system administrators to reassign, add, or remove storage from user groups without interrupting client operations.

To deliver the required performance and scalability, the file system is implemented as a clustered file system with intelligence distributed among the file

---

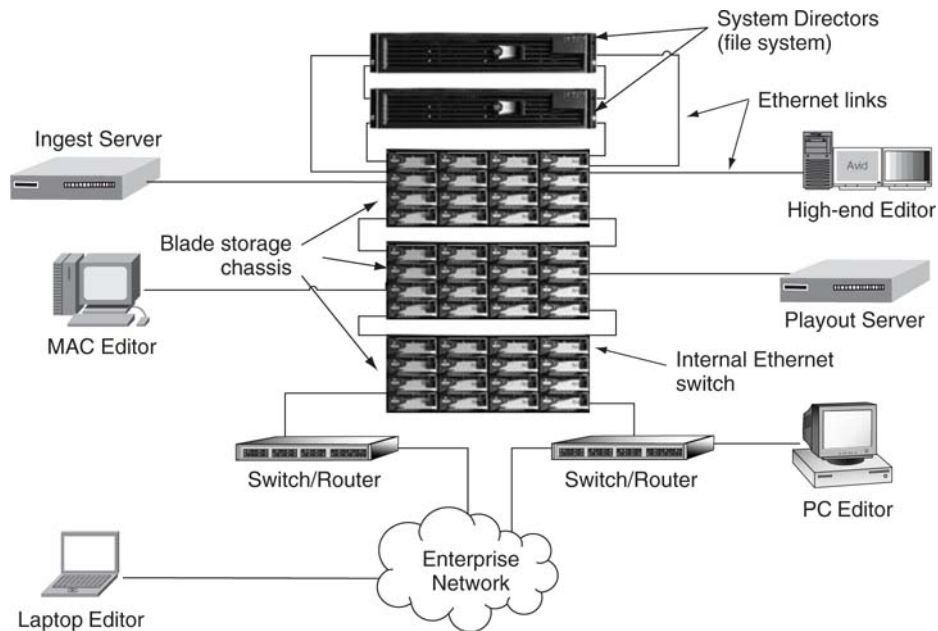
<sup>1</sup> This material summarizes the Avid Unity ISIS storage architecture. General storage system technologies are discussed in Chapters 3A and 3B.



system metadata managers (the system directors), the A/V clients, and intelligent storage blade servers. A/V clients access low data rate file metadata from the directors and access bulk file data via an Ethernet iSCSI-like connection to the storage blades. The director is mirrored offering a NSPOF design.

The storage infrastructure uses one or more blade chassis, each containing up to 16 intelligent storage blades (see Appendix K). Each blade contains two disks (250 or 500GB capacity each), a CPU, and dual gigabit Ethernet ports to connect to the backplane. Further, two independent Ethernet switch blades are integral components to each chassis, enabling clients to directly attach to the storage as well as interconnect multiple blade chassis. For the 24/7 world of broadcast, switch blades and power supplies are redundant and support hot swap. Figure H.1 illustrates a typical configuration of a small number of blade chassis and several real-time clients.

User data are protected using a redundant array of independent nodes (RAIN) rather than RAID methods. When a client writes data to a storage blade, the blade is responsible for making a redundant copy on other blades. If any blade fails, the system director notifies all blades of the failure. A new copy of any lost blade data is made immediately. The data replication process occurs in parallel across all the blades, resulting in exceedingly short rebuild times and improving overall reliability.



**FIGURE H.1** Sample unity ISIS configuration.

## **H.1 SUMMARY**

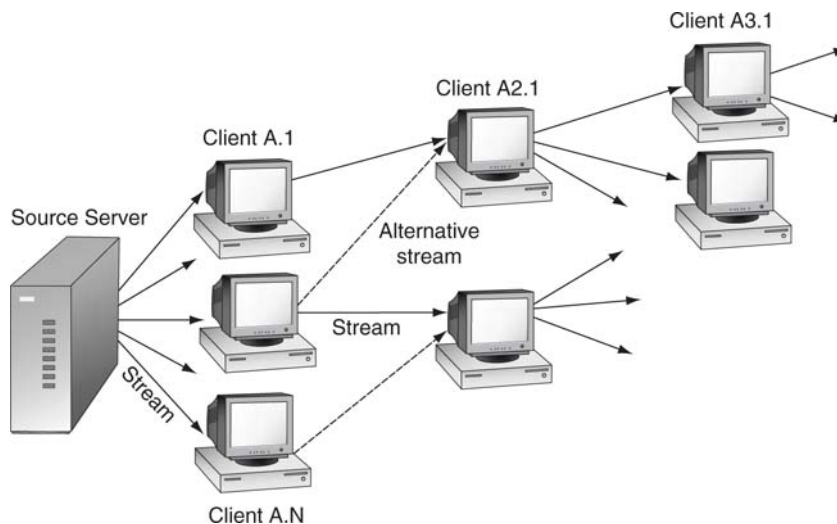
The methods discussed in this appendix illustrate a new approach to A/V real-time storage. The combination of Ethernet-based storage access, an intelligent file system director, and distributed storage creates a system designed specifically for A/V use. The system uses NSPOF design principles and clever use of blades to create a highly reliable and scalable real-time storage system.

This page intentionally left blank

# Appendix I: Is It Rabbits Multiplying or Is It Streaming?

Inventors have dreamed up all manner of methods to stream content. One novel approach is based on a mesh of peer-to-peer connections. The idea centers on distributing the master stream from the source to only a few clients and not the entire population, as is normally the case. Each client (e.g., a PC viewing or listening station) functions as a normal client but in addition provides a ministream server that other clients may draw on. The ministream server may source two to six or more streams, depending on the available link connection bandwidth. When deployed for home use over DSL or a cable modem, audio streaming bandwidths are low enough to allow the scheme to work well.

The aggregate connection bandwidth grows exponentially with the number of clients. With only 6 clients at the left-side head of the client population (assuming clients A.1 to A.6 in Figure I.1), 36 second-level clients may



**FIGURE I.1** *Architecture of a peer-to-peer RT streaming network.*

be fed. This assumes that each client sources 6 streams. With 36 second-level clients, 216 others may be sourced, which in turn can source others. Yes, this process is like rabbits multiplying. On first impression, it may seem that the system is a house of cards—if one client falters, then many others downstream will be deprived of a stream. However, each client is constantly monitoring its own active input and one or more other source streams for data integrity. If the active stream falters, then an alternate is switched to in real time as needed. This balancing act is delicate, but it works quite well even as member clients drop and join the mesh.

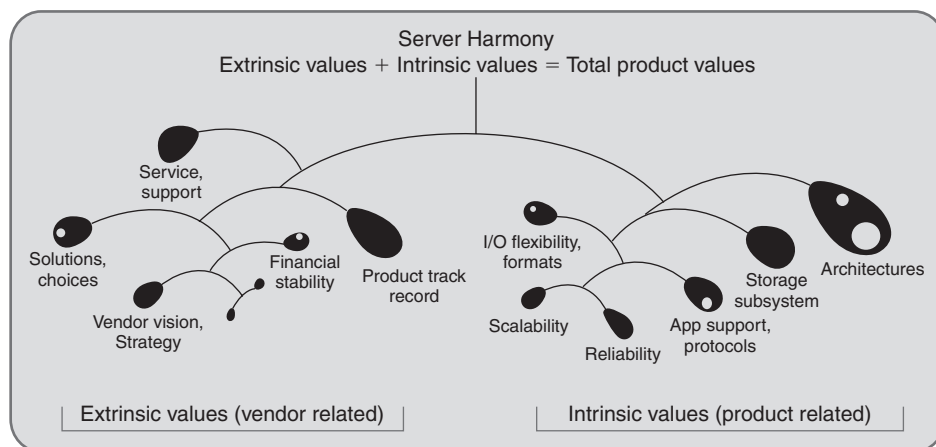
One vendor of this technology is Abacast ([www.abacast.com](http://www.abacast.com)), and it has several marquee clients using its solution. The beauty of this architecture is that the entire client population shares a little of the burden to distribute streams. Meanwhile, the main source server feeds only a few tens of streams, even though a total population of 100K or more clients may be actively attached to the mesh. Long live the rabbits. See Chapter 4 for more information on P2P architectures.

## Appendix J: How to Evaluate a Video Server

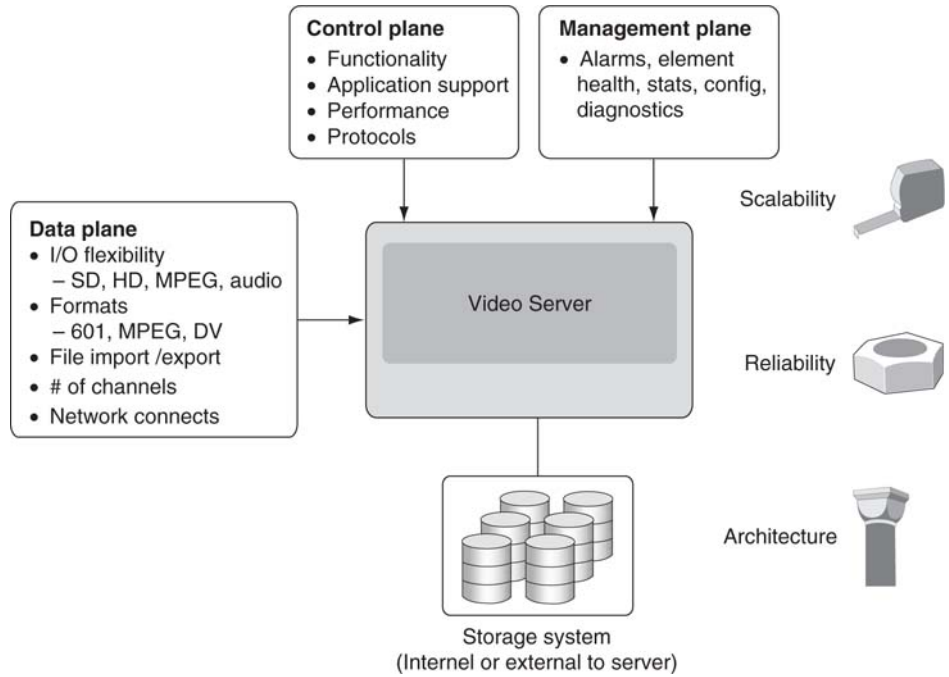
The mobile (Alexander Calder, 1898–1976) hangs in midair, suspended from a single point. Let us use the mobile as a guideline for evaluating broadcast products, particularly the video server.

Figure J.1 shows an example of a mobile illustrating the values of a video server. All the elements represent either product features (right side) or the vendor's values (left side). The left side represents the extrinsic (outside the product) value of a product, and the right side represents the intrinsic (inside the product) value of a product. The combination of the extrinsic plus intrinsic values is the total value of a product and should equate to the product's price.

The left side of the mobile illustrates a vendor's vision, product track record, financial stability, and service/support offerings. The right side notes aspects that are specific to the server's internal features. The discussion to follow focuses on these technical, intrinsic values and leaves the extrinsic vendor analysis to your better judgment.



**FIGURE J.1** *Server harmony.*



**FIGURE J.2** A server's intrinsic value.

There are five main intrinsic categories to consider; they are noted on the right side of the mobile and in Figure J.2. The topics are (1) the three planes, (2) architectures, (3) storage subsystems, (4) scalability, and (5) reliability. Next, let us examine the values associated with the three-plane model first.

## J.0 THE THREE PLANES

Traditional IT devices have been designed using the three-plane model (see Chapter 7). This model is ideal for describing the *data*, *control*, and *management* aspects of a device. Each plane offers specific functionality, as shown in Figure J.2.

The SDI interface on a server is a *data plane* component. File transfer using Ethernet is also a data plane component. It is worth mentioning the need for file exchange interoperability. For this to work smoothly, the file types and associated metadata must be standardized. MXF is becoming the standard of choice for this plane.

The *A/V control plane* mainly uses proprietary command sets (Sony VTR Protocol, VDCP, etc.) over RS422 links. This is changing, with most new protocols being IP based. However, machine control over standard LANs also uses proprietary methods as of 2009. Frankly, there are precious few true A/V control standards, so ask what is provided during your evaluation.

Industrywide, the *management plane* is the least mature of the three. Today, in some cases, this plane is completely absent from A/V devices. What is the purpose of this plane? It is a portal into a device to configure and monitor all aspects of its operations. Broadcast equipment suppliers are starting to include basic IT-based monitoring methods. See Chapter 9 for a summary of these concepts.

## J.1 THE THREE ARCHITECTURES

Video servers are founded on three architectures:

1. *Standalone server.* Sufficient internal storage for all streaming A/V I/O needs. Of course, external storage may be added to augment the internal. Video I/O ranges from 2 to 8 for most cases.
2. *Networked node; file-linked.* Server node with small internal storage, loaded (or unloaded) using the Just in Time File Transfer (JITFT) model. This architecture is mainly useful when the workflow is file centric. Typically, a single node supports 2 to 8 video I/O.
3. *Networked node; storage-linked.* Server nodes connect to external storage using SAN, NAS, or other methods. Typically, a single node supports 2 to 16 video I/O, but nodes may be ganged for larger aggregate I/O using the services of an optional clustered file system (CFS).

In brief, *standalone* servers are restricted because they have limited networked storage access. The *file-linked* method relies on some external controller (automation) to move A/V files to/from the server I/O node. The *storage-linked* method connects I/O nodes to external storage. This is a flexible architecture and allows nodes to share a common storage pool. Hybrid combinations of all three exist.

## J.2 STORAGE SUBSYSTEMS

Storage may be divided into internal and external types. Internal storage is normally one or more hard disc drives configured as just a bunch of discs (JBOD) or a RAIDed configuration.

External storage comes in three main forms: DAS, SAN, and NAS attached. They are discussed in detail in Chapter 3B. When you are selecting a server, ask lots of questions about storage performance (under failure mode especially), reliability, hot and cold upgrade methods, product lifetime, and expected future disc upgrades.

## J.3 SCALABILITY

So, you purchased a  $2 \times 4$  (2 in, 4 out) server 6 months ago. Today your boss asks whether the server can be upgraded to a  $2 \times 8$ . Can it be? What factors affect scalability?



For a small number of required I/O ports and storage capacity, the stand-alone server is the most practical. Its upgradeability is limited. File-linked systems are infinitely scalable because each node connects to a general-purpose IP network. The master source of media files is some offline repository that has its own scalability and reliability requirements. As a result, the problem of scalability is moved, not eliminated.

Storage-linked systems are also scalable, but ask about the maximum number of nodes and total I/O. Most practical systems of this genre support fewer than 100 A/V I/O ports.

## J.4 RELIABILITY ISSUES

Several factors affect the reliability of a server system. The following aspects are key:

- Software complexity and maturity.
- Security against network hazards—viruses, worms, etc.
- Storage protection strategy.
- Redundant components, hot upgrades: fans, power supplies, I/O, etc.
- Redundant nodes:  $N + N$  (true mirror) or  $N + 1$  strategies.  $jV + 1$  provides for a spare node to take over when any one of the  $N$ s fails. Ask the vendor about how the spare node is activated. This is not trivial and is the key to a smooth failover. Chapter 5 reviews failover modes.

Look for these basic factors: the track record of the server, flexibility of the design, and maturity of the solution, including automation control. When in doubt over a manufacturer's claims, decide based on these fundamentals.

## J.5 SUMMARY

The mobile illustration of the balanced server is useful for evaluating products of all types. Do not overlook any of the elements that make up the mobile, and you may indeed find the “perfect server” for your needs.

# Appendix K: Blade Servers

## K.0 TECHNOLOGY OVERVIEW<sup>1</sup>

A blade server plugs into the back or midplane of a chassis, like books slide into a bookshelf, sharing power, fans, floppy drives, and switches with other blade servers. The blades are literally servers on a card, containing processors, memory, integrated network controllers, an optional Fibre Channel HBA, and other I/O ports. Each blade typically comes with one or two local ATA or SCSI drives. For additional storage, blade servers connect to a storage pool facilitated by a NAS, Fibre Channel, or iSCSI SAN.

A high level of scale can be achieved with blade servers by simply adding books to the shelf. Blade management is a key feature offered by vendors. HP (OpenView) and others are extending their server management software to manage and provision blades. While chassis and blades are becoming commodity items, software is the big differentiator and should be at the top of the evaluation list when you are looking at blade server offerings.

## K.1 THE PLAYERS

To no one's surprise, HP, IBM, Dell, Sun, AMD, and Intel have embraced and are offering blade server technology. According to IDC, during 2007, more than 95 percent of all blade revenue was driven by x86 systems where blades represent 14.8 percent of all x86 server revenue. For the full year 2007, worldwide blade server revenue grew 40.9 percent year over year to \$3.9 billion.

In an attempt to seed the blade world, IBM introduced an open specification. Smaller companies can use this spec and create "IBM-compatible" blade servers. Does this sound familiar? If IBM's spec becomes a de facto standard, then the IBM blade server will be the "PC of the server world."

---

<sup>1</sup> The major part of this text was provided by Jacob Gsoedl, IT Director of a F500 company.

## K.2 BLADE SERVER MERITS AND CHALLENGES

Server and storage consolidation are an essential part of any IT cost reduction, and blade servers increasingly play an important role in it. Blade servers allow putting more processing power in less rack space, simplifying cabling, and reducing power consumption. Components such as the chassis, power supply, and fans can be shared across a mix of blades and blade generations, resulting in tangible savings and investment protection.

In combination with good management software, a blade server infrastructure is simpler to maintain and offers improved flexibility, manageability, and modular scalability by simply adding blades. The ability of system management software to dynamically assign blades and virtual machines to applications will give an additional boost to blade servers, bringing us ever closer to “on-demand” computing.

## K.3 OUTLOOK

The overall trend is toward higher density and more manageable systems and, as a result, blade servers will be favored over larger form factor servers. Research and Markets ([www.researchandmarkets.com](http://www.researchandmarkets.com)) estimates the worldwide total blade server market at \$5.2 billion in 2008 and is forecast to reach \$15.7 billion by 2013. Larger form factor servers will find a place where more expansion, additional redundancy options, higher processing power, more cooling, and in-box disk storage are required. As a result, the two server types will carve out individual market segments and will coexist for years to come.

# Appendix L: Solid State Discs Set Off Flash Flood

A Solid State Disc (SSD) mimics a HDD but with non-volatile flash memory replacing rotating platters. Will SSD replace HDD anytime soon? For some applications, it's happening now; for others, it may never happen. Comparing the two technologies is partially an "apples to oranges" scenario. True, they are both fruit—but the differences are significant even with identical I/O ports (SATA or SAS) and slot form factor.

The key SSD advantages over HDD are as follows: nearly instantaneous R/W access; lower active power consumption by 3–5× and ~40× for inactive state; no moving parts; robust, longer MTBF by a factor of ~3×; no noise; no file fragmentation. Surprisingly, SSD is not far behind HDD in terms of capacity; BitMICRO Networks offers a 2.5 inch SATA SSD with a capacity of 832GB.

Small data block, read-intensive applications show the most leverage compared to HDD. A report from Burleson Consulting – see [www.dba-oracle.com](http://www.dba-oracle.com) concluded that, for a query-intensive Oracle 9 database server benchmark, SSD-based storage outperformed HDD by the respectable factor of 176. For read-intensive A/V applications (VOD server, video server, field camera, etc.) SSD or pure Flash cards are replacing HDD now and will continue to do so as price points improve. For several years, Toshiba has offered the Flash-based ON-AIR MAX video server for broadcast and digital cinema operations.

According to Sun Microsystems, a 146GB disk drive with 15,000 RPM gets about 180 write IOPS and 320 read IOPS, while a 32GB Flash drive gets 5,000 or more write IOPS and at least 30,000 read IOPS. Disk drives ended up costing \$2.43 per IOPS (as of late 2008), and solid state drives using Flash cost \$0.08 per IOPS. When performance is needed, SSDs will be used. For an example of a hybrid SSD/HDD storage system that supports 288TB HDD and 600 GB SSD, see Sun's Amber Road Storage 7K series.

However, not all is rosy. SSDs suffer a ~100× cost/GB burden in 2008 compared to a SATA HDD and less for a SAS drive. Cost/GB will improve but likely never reach parity. Also, Flash has an upper limit on the number of write cycles. This number is always improving, but 5 million cycles

([www.mtron.com](http://www.mtron.com)) is state of the art in 2009. Most SSD file systems use *wear leveling* techniques to spread out writes for even distribution. Don't expect a Flash memory to drop dead after the write cycle spec is exceeded. Rather, write errors will slowly creep in and require strong read error correction methods to hide them. Some vendors use strong error correction to fend off this issue as long as possible.

The read/write operations mix for a SSD can yield drastically different access data rates. For example, a small block, random SSD read may be  $\sim 20\times$  faster than for a SAS HDD. On the other hand, for small random writes, SSD is slower by  $\sim 15\times$ . Importantly, for a 50/50 small block read/write mix, the HDD performance is  $\sim 8\times$  faster! So, the message is clear: small writes can ruin SSD read performance. See Easy Computer Company's white papers ([managedflash.com](http://managedflash.com)) for an excellent analysis of this problem and mitigation techniques. So, you should not choose a SSD in place of a HDD without knowing the application scenarios and R/W mix. Something as seemingly innocent as a frequent status log update written to a SSD can ruin the overall performance of a mission-critical application. Designer and user beware.

In the big picture, expect to see select application servers, including some video servers, use a mix of HDD and Flash. Flash offers the best IOPS performance for large block R/W with access speeds exceeding HDD.

## L.0 THE MEMRISTOR—A NEW APPROACH TO NON-VOLATILE MEMORY

In 1971 Leon Chua of UC Berkeley (Chua 1971) predicated the Memristor, a nonlinear, passive, two-terminal, circuit element. It is described as the fourth basic type of passive circuit element, in league with the capacitor, resistor, and inductor. It was a hypothetical device for 37 years, with no known physical examples.

In April 2008, a working device with similar characteristics to a Memristor was announced by a team of researchers at HP Labs ([www.spectrum.ieee.org/may08/6207](http://www.spectrum.ieee.org/may08/6207)). The Memristor may enable a new class of high-density, non-volatile digital memory. When a voltage is applied, the Memristor remembers how much was applied and for how long. Its resistance is a function of the applied voltage. This fact can be used to store one or more bits per element. It could surpass Flash memory as a new, high-density memory element.

## References

Chua, Leon O. (September 1971). Memristor—The Missing Circuit Element. *IEEE Transactions on Circuit Theory CT*, 18(5), 507–519.

# Appendix M: Will Ethernet Switches Ever Replace Traditional Video Routers?

No doubt, traditional video routers (SDI link I/O) form the backbone of many news, live event, post, and broadcast facilities. These routers offer non-blocking crossbar-like performance, enormous bandwidth, very small insertion delay/jitter, no loss, point-to-multipoint outputs, central or distributed connection control, and excellent reliability. So, why consider replacing them with Ethernet switches?

If Ethernet switches (MAC layer 2, not IP layer 3) can meet the strict user requirements just outlined, they will be used due to overall price, choice, and general IT advantages. But, can they actually “cut the mustard”? Well, consider the following features of the most modern carrier class Ethernet switches:

- They can be configured to support point-to-point trunking using the IEEE 802.1Qay and 802.1ah set of standards. This offers similar function as a connection-oriented SDI router system.
- Point to multipoint is supported. This is equivalent to a SDI router with multiple outputs per one input.
- Centralized route control is enabled. It's true that Ethernet and IP switching mainly use self-routing methods. However, standards also support layer 2 circuit emulation provisioning. This is equivalent to the out-of-band control of SDI routers.
- Ten gigabits per second I/O supports even the largest practical uncompressed video signals. One gigabit per second links will support multi-channel SD and compressed HD.
- They can switch with zero loss, short delay/jitter ( $<1\ \mu\text{s}$ ). With managed circuit provisioning, there are no Ethernet frames dropped during switch passthrough.
- They can scale to millions of trunks and switch failover in  $<50\ \text{Msec}$ .

It will take time, if ever, for these features to be fully applied and replace traditional video routers. Plus, all connecting devices must support 1 GB or 10 GB Ethernet ports rather than SDI links. Costwise, 10 GB Ethernet switches have a compelling price advantage over SDI routers (3 G SDI ports) by a factor of  $\sim 2\times$  in 2009.

Why not use IP layer 3 routers? Layer 2 switching offers lower delay/jitter, faster failover, central route control, and circuit emulation not found in IP routers. See the Metro Ethernet Forum ([www.metroethernetforum.org](http://www.metroethernetforum.org)) for more information. However, let's not be dogmatic; some professional video streaming applications are adequately supported by IP routers.

Another approach is to use the technology being developed by the IEEE 802.1 Audio/Video Bridging Task Group ([www.ieee802.org/1/pages/avbridges.html](http://www.ieee802.org/1/pages/avbridges.html)). A series of standards will enable time-synchronized, excellent QoS, low-latency A/V streaming services through 802 networks. This method, too, must become mature before having a chance to replace the SDI router.

In the final analysis, there are competitors waiting in the wings to replace the SDI router. Maturity of the technology and a willingness to apply these methods to video systems must prevail before SDI video routers fade into the sunset.

# A Glossary of AV/IT Terms

- 2K/4K (digital cinema)**—Shorthand for 2K video (most often resolutions of  $2,048 \times 1,536$  or  $2,048 \times 1,556$  pixels) and 4K video (most often resolutions of  $4,096 \times 2,160$  or  $4,096 \times 1,714$ ) usually at frame rates of 24p or multiples.
- 24p**—Video format at 24 progressive (not interlaced) frames per second. It is based on the traditional film rate of 24 FPS.
- 3:2 pull down**—The process of converting 24 frame per second film to 60 (59.94 actually) fields per second video by repeating one film frame as three video fields and then the next film frame as two fields. The actual order is 2, 3, so 2:3 is more accurate but the principle is the same.
- 4:2:2 (4:4:4, 4:1:1, 4:2:0)**—Common designations for pixel sampling relationships in a digital component video format. The first term relates to the luma (Y') sampling rate, and the other two relate to the chroma (Cr and Cb) sampling rates. See Chapter 11 to decipher the hidden codes.
- 480i**—Shorthand for a 480 active line, SD-interlaced scanning standard usually with 720 horizontal pixels at various frame rates. The progressive version is 480p. Total lines are 525. NTSC is based on this scanning structure.
- 50i/60i**—Shorthand for 50 or 60 interlaced fields per second video. Closely related is the shorthand 50p/60p, referring to the progressive frames per second version.
- 576i**—Shorthand for a 576 active line, SD-interlaced scanning standard usually with 720 horizontal pixels at various frame rates. Total lines are 625. PAL is based on this scanning structure.
- 601**—Formally CCIR 601. This is the original name of a standard published by the CCIR (now ITU-R) for converting analog video signals to digital form. The new name of the standard is ITU-R BT.601. It includes methods for converting 525/625 line analog video to digital.
- 720p**—Shorthand for a 720 active line, HD-progressive scanning standard with 1,280 horizontal pixels at various frame rates. Total lines are 750.



**1080i**—Shorthand for a 1,080 active line, HD-interlaced scanning standard with 1,920 horizontal pixels at various frame rates. The progressive version is 1,080p. Total lines are 1,125.

**AAF**—Advanced Authoring Format. A format for annotating the composition of an A/V edit project. The AAF Association (rebranded as AMWA) is responsible for its development.

**Active picture area**—The production area of the raster scan. For 525/NTSC systems, there are 480 active lines and 576 for 625/PAL systems. The *safe picture* is a slightly reduced area that is likely viewable on most TVs.

**AES**—Audio Engineering Society. This group sets standards and recommendations for audio technology.

**AES/EBU audio**—The joint-effort standard for packaging digital audio up to 24 bits/sample onto a serial link. There is also a mapping for compressed audio.

**Aliasing (Image)**—Visual distortions resulting from digitally sampling an image at a rate too low to capture all significant spatial or temporal detail. A result of temporal video aliasing can be seen as a car's wheels appearing to spin backward even though the car is moving forward.

**Alpha**—The measure of a pixel's opacity. A pixel with the maximum alpha value is solid, one with a value of zero is transparent, and one with an intermediate value is translucent.

**AMWA**—Advanced Media Workflow Association. An industry organization dedicated to creating technology for open specifications relating to file-based workflows (including AAF, MXF, and other formats), Service Oriented Architectures, and application specifications.

**Anamorphic**—An image system that optically compresses the picture horizontally (usually) during capture and then restores it to normal proportions for viewing. Typical application is to capture a  $16 \times 9$  image but compress it to  $4 \times 3$  for storage and transport.

**ASI**—Asynchronous Serial Interface. This is a 270 Mbps serial link used most often to carry MPEG Transport Streams. MPEG data can be a single program of A/V or many multiplexed programs. Most MPEG streams have data rates less than 100 Mbps. DVB defines this spec.

**Aspect ratio (AR)**—The ratio of a display's horizontal versus vertical size expressed as H:V or  $H \times V$ ;  $4 \times 3$  and  $16 \times 9$  are common. ARC is shorthand for Aspect Ratio Converter, a device that converts from one AR (for example,  $16 \times 9$ ) to another (for example,  $4 \times 3$ ).

**ATA**—Advanced technology attachment. A parallel link for connecting disc drives and other devices inside a PC or other product. Initially developed for low-end device attachments. See Chapter 3B.

- ATM**—Asynchronous Transfer Mode. A WAN service based on layer 2 switching. ATM cells carry upper layer payloads such as TCP/IP.
- ATSC**—Advanced Television Systems Committee. This group defined the digital terrestrial broadcast standard in use for North America and elsewhere. It is based on MPEG encoding and supports SD and HD resolutions.
- Automation**—The process of controlling A/V system operations with a hands-off scheduler. Facility routers, servers, VTRs, compositors, mixers, codecs, processors, and more are controlled by automation logic. The automation operations are often controlled by a time-based schedule.
- A/V (or AV)**—Audio/visual or audio/video. A generic term for describing audio, video, graphics, animations, and associated technology.
- AVC**—Advanced video codec. This term describes the compression format also referred to as MPEG4 part 10 and separately as H.264. AVC-Intra is a constrained form used as a portable camera capture format.
- AVI**—Audio video interleaved. An A/V wrapper or container format for multiplexing A/V essence. It is not a compression format.
- AVIT or AV/IT**—Shorthand for a A/V + IT hybrid system and related concepts.
- AXF**—Archive eXchange Format. A SMPTE-specified, data-block tape layout to enable cartridge exchange between archive systems.
- BWAV**—The EBU's broadcast audio WAV format.
- BXF**—Broadcast eXchange Format. BXF (SMPTE 2021) was developed to standardize methodologies for messaging and exchanging metadata between traffic, automation, content management, and workflow software systems.
- CCIE**—Cisco certified Internet work expert.
- CFS**—Clustered file system. A file system shared in common by more than one client or server. A CFS provides users with a shared or common view of all stored files, usually from one large pool. A CFS is a networkable service either configured as standalone or distributed among the nodes. In the latter case, a CFS is sometimes called a distributed file system (DFS).
- Chroma**—A value that conveys a color signal independent from the luma component.
- CIF**—Common intermediate format. This is a spatial resolution image format of 352 (H)  $\times$  288 (V) pixels, 4:2:0. See also *QCIF*. CIF also has a secondary meaning: common image format. This second use is defined by MPEG as either 720  $\times$  480 or 720  $\times$  576. Beware of this acronym. See also *SIF*.
- CIFS**—Common Internet File System. A Microsoft-developed remote file-sharing protocol. NAS servers often support this. See also *NFS*.

**CIM**—Common Information Model. This is a model for describing managed information.

**CIMOM**—CIM Object Manager. This is a software component for accessing data elements in the CIM schema.

**Closed captioning (CC)**—Textual captioning on a TV screen for the hearing impaired. CC data are carried on unseen line 21 using NTSC. Standard EIA-608 defines CC data structures for analog transmission, and EIA-708 defines CC for digital transmission systems. Teletext subtitling is a similar system used in PAL countries.

**Cloudware**—Software that runs from the Web rather than a local desktop or campus server. Google, Microsoft, and others offer networked applications that replace common desktop versions.

**Color burst**—A burst of 8–10 cycles of subcarrier inserted into a composite video signal after the H\_Sync pulse. It is used to synchronize the receiver's video color decoder circuitry.

**Colorimetry**—The science of defining and measuring color and color appearance.

**Component video**—A method of signal representation that maintains the original image elements separately rather than combined (encoded) as a single, composite signal. Video signal sets such as R'G'B' and Y'CrCb are component video signals.

**Composite video**—A standardized way to combine luma, chroma, and timing information into one signal. The NTSC and PAL standards define the methods to create a composite video signal. See Chapter 11.

**Content**—In the context of A/V media, content = essence + metadata.

**CORBA**—Common Object Request Broker Architecture.

**CoS**—Class of service. An indicator of performance or feature set associated with a flow of information. A CoS may be set to prioritize an A/V flow.

**COTS**—Commercial off-the-shelf. Products that can be purchased and integrated with little or no customization, thus facilitating customer infrastructure expansion and reducing costs. They are generic and not designed specifically to meet A/V requirements for the most part.

**CRUD**—Create, read, update, delete. CRUD describes the basic four functions needed to implement RESTful Web services as an example.

**CVBS**—Composite video burst and sync. Shorthand to describe a composite video signal.

**D1, D2, ... D16**—Various SMPTE-standardized videotape formats with D4, 8, and 13 not defined.

- DAS**—Direct attached storage. Storage that is local and dedicated to a device. Chapter 3B covers this in detail. Sometimes called direct access storage device (DASD).
- DCML**—Data Center Markup Language. A model that describes a data center environment.
- DES**—Data Encryption Standard. An international standard for encryption and decryption. The same key is used for both.
- DHCP**—Dynamic Host Configuration Protocol. A method to automatically assign an IP address to a newly attached network client. A DHCP server doles out IP addresses from a pool.
- DI**—Digital Intermediate. A process step in the digital cinema production chain, whereas the image is manipulated in digital form (as compared to pure optical) before being recorded to film or other media for display.
- Diffserv**—Differentiated services. A method defined by the IETF to segregate user data flows per class of service and associated QoS.
- DNS**—Domain Name Server. This network service translates between a named address (as in [www.ebay.com](http://www.ebay.com)) and its IP address (66.135.208.89). The DNS may select an IP address from a pool, thereby performing a type of load balancing.
- DPX**—Digital Picture Exchange. A file container format for digital cinema images. DPX is defined as SMPTE standard 268M.
- DRM**—Digital rights management. The processes and techniques to secure and control the use of digital content by users.
- DTMB**—Digital Terrestrial Multimedia Broadcast. This is the digital TV broadcast standard for China.
- DV**—Digital Video. This is a video compression format and tape format. It is in common use for news gathering, consumer cameras, and editing. The nominal video rate is 25 Mbps, but 50 and 100 are also standardized. See also *HDV*.
- DVB**—Digital Video Broadcasting. This is a family of standards for digital transmission over terrestrial, cable, and satellite. DVB standards are implemented by 55+ countries. It supports SD and HD resolutions.
- EBU**—European Broadcasting Union. A television broadcast users' group dedicated to education, setting policy, and recommendations for its members. Based in Geneva.
- EDL**—Edit decision list. A text file for annotating the composition of an edit project. See also *AAF*.
- Embedded audio**—The process of carrying audio and video on the same link; usually SDI as defined by SMPTE 259M or 292M.

**eSATA**—External Serial Advanced Technology Attachment. This is an external interface (connector type) for the SATA link. It competes with FireWire 800 and USB 2.0 to provide fast data transfer speeds for external storage devices.

**Essence**—Basic, low-level A/V data structures such as uncompressed audio or video, MPEG, DV, or WAV data. It is distinguished from “content” that normally has metadata and other non-A/V elements associated with it. A MXF file packages essence elements.

**FCoE**—Fibre Channel over Ethernet. FCoE is a proposed mapping of Fibre Channel frames over full-duplex IEEE 802.3 networks. Fibre Channel is able to leverage 10 gigabit Ethernet networks while preserving its higher level protocols.

**FCP**—Fibre Channel Protocol. This is a mapping protocol for carrying the SCSI command set over the Fibre Channel link.

**FEC**—Forward error correction. A method to correct for faulty transmitted data by applying error correction at the receiver. FEC needs overhead bits and can only correct for a maximum number of bad bits per sent payload.

**Fibre Channel (FC)**—A serial link for moving data to/from a storage element or system. It may be optical or even copper based despite the name. *Fibre* (British spelling) is used instead of *fiber* to distinguish it from other optical fiber links. One-, 2, 4, and 8 Gbps links are defined.

**Field**—With interlaced video, two fields are used to create a full frame. The first complete field (odd lines of the frame) is followed by the second field (even lines of the frame) in time. In practice, the lines are numbered consecutively across fields.

**FPS**—Frames per second.

**Frame**—Essentially, one complete picture. An interlaced frame is composed of two fields (two complete interlaced scans of the monitor screen). A frame consists of 525 interlaced horizontal lines in NSTC and 625 in PAL. A progressive frame is a sequential scan of lines without any interleaving.

**Frame accurate**—Actions on a video signal at a desired frame position.

**Gamma**—The exponent value applied to a linear-light signal to obtain an  $R'$ ,  $G'$ , or  $B'$  signal. For example,  $R' = R^{0.45}$  is a gamma-corrected red-valued video signal. The 0.45 is the gamma value. The apostrophe indicates a gamma-corrected variable. See Chapter 11.

**Gen Lock**—See *video reference*.

**GOLF**—Group of linked files. A directory of associated files that are part of the same program material.

- GOP**—Group of pictures. In MPEG, a GOP is a collection of sequential pictures (frames) bound by temporal associations. A short GOP is one I frame. The long GOP format is normally a 12- to 15-length IBP sequence for standard definition formats. See also *IBP*.
- H.264**—A video compression format also defined as MPEG4 part 10. It offers superior compression compared to the older MPEG2 methods.
- HA**—High availability. The ability of a device/system to withstand hardware or software failures. HA is achieved by using forms of element duplication. See Chapter 5.
- HANC**—Horizontal ANCillary data field. Both 292M and 259M SDI links contain ancillary data space in the horizontal and vertical dimensions. HANC is included in the portion of each scanning line outside the active picture area and may be used to carry embedded audio. The vertical ANCillary data space (VANC) corresponds to the analog vertical blanking interval. It encompasses much bigger chunks of data space than HANC. Metadata may be embedded in the VANC data space.
- HBA**—Host bus adaptor. An interface card that plugs into a computer's bus and provides network connectivity.
- HD**—High-definition video resolution. See Chapter 11.
- HDD**—Hard disc drive or hard disk drive.
- HDV**—High-definition video. This is an A/V tape and compression format using MPEG at 25 Mbps for 1080i and ~19 Mbps for 720p. The tape cartridge is the same as used for standard definition DV.
- Horizontal sync**—The portion of the video signal that triggers the receiver to start the next left-to-right raster scan.
- HSM**—Hierarchical storage management. The process of automatically moving/storing data to the lowest-cost devices commensurate with upper layer application needs.
- HTTP**—Hypertext Transfer Protocol. A protocol used to carry data between a client and a server over the Web. HTTPS is an encrypted version of HTTP. See also *SSL*.
- IBP**—Intraframe, Bidirectionally predicted, Predicted. This is MPEG shorthand for three different compressed video frame types. The I picture is a standalone compressed picture frame. The B picture is predicted from one or two of its I or P neighbors. The P picture is predicted from the previous I or P picture.
- IETF**—Internet Engineering Task Force. The body responsible for many of the Internet's standards.

**IFS**—Installable file system. This client or server software component redirects local file system calls to another internal or external file system (a CFS).

**IKE**—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services that require keys such as IPSec.

**ILM**—Information life cycle management.

**InfiniBand**—A switched-fabric I/O technology that ties together servers, storage devices, and network devices.

**Interlace scan**—A display image that is composed of time and spatially offset interleaved image fields. Two fields create a frame. Compare to *progressive scan*. See Chapter 11.

**Interoperability**—The capability to communicate and transfer data among various functional units without format or connectivity problems.

**IP**—Internet Protocol. The Internet Protocol, defined by RFC 791, is the network layer for the TCP/IP protocol suite. It is a connectionless, best-effort packet-switching protocol. This is most often referred to as IPV4. See Chapter 6.

**IPSec**—IP Security. A security protocol that provides for confidentiality and authentication of individual IP packets.

**IPV6**—A version upgrade of IPV4, including improved address space, quality of service, and data security.

**iSCSI**—SCSI commands over an IP-based link. It finds application in SAN environments and is a replacement technology for Fibre Channel. See Chapter 3B.

**ISDB**—Integrated Services Digital Broadcasting. The digital television broadcast standard for Japan.

**iSNS**—Internet Storage Naming Service. The iSNS protocol is designed to facilitate the automated discovery, management, and configuration of iSCSI and Fibre Channel devices on a TCP/IP network.

**Isochronous**—Signals that carry their own timing information imbedded as part of the signal. A SMPTE 259M SDI signal is an isochronous signal.

**IT**—Information technology. Related to technologies for the creation, storage, processing, consumption, and management of digital information.

**ITIL**—Information Technology Infrastructure Library. This is a framework of best practices that promote quality services in the IT sector. ITIL addresses the organizational and skill requirements for an organization to manage its IT operations.

**Java EE 5**—Java Enterprise Edition version 5. A Java-based, runtime platform for developing, deploying, and managing multitier, server-centric applications

on an enterprise-wide scale. Java Standard Edition 6 (Java SE 6) is a reduced form of the Java EE model.

**JBOD**—Just a bunch of disks. This informal term refers to a hard disk array that isn't configured according to RAID principles.

**JITFT**—Just in Time File Transfer. A concept of file exchange where the delivered file arrives at its destination “just in time” by some comfortable margin to be used for editing, playout, format conversion, or some other operation.

**JPEG**—Joint Photographic Experts Group. A digital image file format standard using image compression.

**JPEG2000**—A wavelet-based image compression standard used for both photos and moving images. It is the basis of the digital cinema specification as defined by the Digital Cinema Initiatives (DCI) group. It is often abbreviated as J2K.

**Key**—See *video key*.

**Key performance indicators (KPIs)**—Quantifiable measurements that reflect the critical success factors of an organization. Business dashboards (sales volume, inventory, units/hour, etc.) show typical KPIs.

**LAN**—Local area network. A data network covering a limited area. See Chapter 6.

**Long GOP**—See *GOP*.

**LTC**—Linear (longitudinal) time code. The SMPTE 12M time code standard historically recorded onto the audio track of a VTR or audio recorder. See also *VITC*.

**Luma**—A video signal related to the monochrome or lightness component of a scene. Often tagged as *Y'*.

**LUN**—Logical unit number. A LUN addresses a fixed amount of storage from a pool. The SCSI protocol uses LUNs to address portions of total storage.

**MAM**—Media asset management. These are the technologies used to index, catalog, search, browse, retrieve, manage, and archive specific media content objects. Also, more generally referred to as digital asset management when there are no time-based data types but mainly text and graphics.

**MAN**—Metropolitan area network. See Chapter 6.

**Media Dispatch Protocol (MDP)**—A transaction-oriented protocol for establishing the contract for a file transfer between two entities. It is standardized as SMPTE 2032.

**Metadata**—Literally defined as structured data about data. Metadata are descriptive information about an object or resource. *Dark metadata* are a value-set undefined to the current application but may be useful to another application.



**MIB**—Management Information Base. See Chapter 9.

**MOS Protocol**—Media Object Server Protocol. A protocol for managing the rundown list and associated operations per story for a news broadcast. See [www.mosprotocol.com](http://www.mosprotocol.com).

**MPEG**—Motion Picture Experts Group. This is an ISO/IEC standards body responsible for developing A/V compression formats (MPEG1, 2, 4) and other A/V-related standards (MPEG7, 21). MPEG2 and MPEG4 part 10 (H.264) are used as distribution formats for digital cable, satellite, over-the-air, and other applications.

**MPLS**—Multiprotocol Label Switching. See Chapter 6.

**MSCE**—Microsoft certified systems engineer.

**MSO**—Multiple system operator. A cable industry term describing a company that operates more than one cable TV system.

**MXF**—Material eXchange Format. A file wrapper or container format for A/V professional use. MXF encapsulates audio + video + metadata elements in a time-aligned manner. It also supports streaming. See Chapter 7 for more information.

**NAS**—Network attached storage. Typically, a data server on a network that provides file storage. See Chapter 3B.

**NAT**—Network address translation. This method maps a local area network private IP address to/from a globally unique IP address. The method conserves the precious, global IP address space.

**.NET**—Microsoft's programming framework for creating applications and services using combinations of servers and clients of all types. It relies on XML and Web services to implement solutions.

**NFS**—Network file system. A standardized protocol for networked file sharing. NAS file servers often support this. See also *CIFS*.

**NLE**—Nonlinear editor. The computer-assisted editing of A/V materials without the need to assemble them in a linear sequence. The visual equivalent of word processing. Tape-based editing is considered linear editing.

**NRCS**—News room computer system. A set of software applications for managing the editorial aspects of a news story.

**NRT**—Non-real-time. See also *RT*.

**NTP**—Network Time Protocol. A method for synchronizing remote clocks to a master clock over a packet-switched network with delay and jitter. RFC 1305 specifies the methods. A related protocol is IEEE 1588, the Precision Time Protocol.

**NSPOF**—No single point of failure. A system that tolerates a single component failure using fast bypass techniques. Performance should always be specified under a single failure mode.

**NTSC**—National Television Systems Committee. It describes the SD system of color analog TV used mainly in North America, Japan, and parts of South America. NTSC uses 525 lines per frame and 29.97 frames (59.94 fields) per second.

**OASIS**—Organization for the Advancement of Structured Information Standards. The group is a consortium that drives the development, convergence, and adoption of open standards for the global information society.

**OOP**—Object-oriented programming.

**PAL**—Phase alternate line. The name of the SD analog color television system used mainly in Europe, China, Malaysia, Australia, New Zealand, the Middle East, and parts of Africa. It uses 25 frames per second and 625 lines per frame.

**Plesiochronous**—Plesiochronous is derived from the Greek *plesio*, meaning near, and *chronos*, time. Plesiochronous systems run in a state in which different parts of the system are almost, but not quite perfectly, synchronized.

**POTS**—Plain old telephone service.

**Progressive scan**—An image that is scanned sequentially from top to bottom to create a single frame. Compare to *interlace scan*.

**PSTN**—Public Switched Telephone Network.

**QCIF**—Quarter common intermediate format. This is a spatial resolution image format of 176 (H)  $\times$  144 (V) pixels, 4:2:0. See also *CIF*.

**QoS**—Quality of service. A guarantee of predictable metrics for the data rate, latency, jitter, and loss for a network connection. It can also apply to other services with corresponding QoS metrics for that service. For example, typical QoS metrics of a storage system are transaction latency, R/W access rate, and availability.

**RAID**—Redundant array of independent (or inexpensive) discs. A method to improve the reliability of an array of discs. See Chapter 5.

**RDMA**—Remote direct memory access. A method to move block data between two memory systems where one is local and one remote.

**RESTful Services**—A model for Web services based on HTTP, CRUD functions, and URIs to implement the service calls. REST is derived from REpresentational State Transfer.

**RFC**—Request for comment. A specification developed by the IETF. The document series, begun in 1969, describes the Internet suite of protocols. Not all

RFCs describe Internet standards, but many do. Other bodies also contribute to the standards pool.

**RFP**—Request for proposal.

**RGB**—Red, green, and blue primary linear-light components. The exact color interpretation depends on the colorimetry scheme used.

**R'G'B'**—Red, green, and blue primary non-linear light components. The prime symbol denotes a gamma-corrected value. See also *gamma*.

**Router**—A device that buffers and forwards data packets across an internet-network toward their destinations. Routing occurs at layer 3 (the network layer, e.g., IP) of the protocol stack.

**RPC**—Remote procedure call. A protocol for connecting to and running individual processes on remote computers across a network. The client/server model may use RPC-style message passing.

**RT**—Real time. An activity that occurs in “A/V real time” such as live-streamed video/audio or A/V device control.

**RTP**—Real-Time Protocol. The IETF standard RFC 1889 for streaming A/V usually across IP/UDP networks. Most Web-based A/V streaming uses RTP. Professional IP-based video carriage often relies on RTP. Most A/V data types (MPEG, MP3 audio, others) have a mapping for carriage using RTP.

**Samba**—An open source implementation of Microsoft’s CIFS protocol for file and printer sharing. For example, a Linux computer using Samba appears as a Windows-networked file system.

**SAN**—Storage area network. This is a technology for sharing a pool of storage with many independent clients and servers. See Chapter 3B for more details.

**SAS**—Serial-attached SCSI. This a serial version of the venerable parallel SCSI link.

**SATA**—Serial ATA. The serialized form of the common ATA interface. SATA and SAS connectivity have converged; see Chapter 3B. The 3.0 Gbps speed has been widely referred to as SATA II or SATA2. This is a misnomer; there is only SATA.

**SCSI**—Small Computer System Interface. The standard parallel interface for disc drives and other devices for high-end use. The SCSI command layer is used in Fibre Channel and other serial links; see Chapter 3A.

**SD**—Standard definition video resolution. See Chapter 11.

**SDI**—Serial digital interface. A serial coaxial link used to move A/V digital data from point to point in a professional video system. The nominal line rate is

270 Mbps for SD video. It is defined by SMPTE 259M for SD and 292M for HD.

**SDTI**—Serial Digital Transport Interface (SMPTE 305M). This link uses SMPTE 259M (SDI) as an agnostic payload carrier. There are several defined data mappings onto the SDTI link with compressed payloads (MPEG, DV, VC-3) being the most common.

**SECAM**—A French acronym describing an analog color television system. It is closely related to PAL in line structure and rates.

**SI**—System's integrator.

**SIF**—Source input format. A 4:2:0,  $352 \times 288$  (25 FPS) or  $352 \times 240$  (29.97 FPS) image.

**SLA**—Service Level Agreement. A service contract between a customer and a LAN/WAN/MAN service provider that specifies the working QoS and reliability a customer should expect.

**SMB**—Server message block. The foundation protocol for Microsoft Windows file-server access, also described as the Common Internet File System (CIFS). See also *Samba*.

**SMEF**—Standard Media Exchange Framework. A BBC-developed XML schema for describing MAM metadata.

**SMI**—Storage Management Initiative. A project of the Storage Networking Industry Association (SNIA) to develop and standardize storage management methods.

**SMP**—Symmetric multiprocessing. See Appendix C.

**SMPTE**—Society of Motion Picture and Television Engineers. A professional engineering society tasked with developing educational forums and technical standards for motion pictures and television.

**SNMP**—Simple Network Management Protocol. See Chapter 9.

**SOA**—Service-oriented architecture. An architecture of distributed, loosely coupled services available over a network. Consumers (clients) access networked (using middleware) service providers (servers) to perform some well-defined task. SOA principles enable business agility and business process visibility.

**SOAP**—Simple Object Access Protocol. SOAP is a lightweight protocol for the exchange of information in a decentralized, distributed environment. It is XML based and consists of three parts: an envelope that defines the message and how to process it, a set of encoding rules for expressing the application-related data types, and a convention for representing remote calls and responses. SOAP is fundamental in W3C's Web services specs.

**SONET**—Synchronous Optical NETwork. See Appendix F for more information.

**SPOF**—Single point of failure. Compare to *NSPOF*.

**SQL**—Structured Query Language. A standard language for querying and modifying relational databases.

**SSD**—Solid State Disc. A SSD mimics a HDD but with non-volatile Flash memory replacing rotating platters. See Appendix L for a review of this technology.

**SSL**—Secure Sockets Layer. This is a method of encrypting networked data using a public key. HTTPS uses SSL.

**S\_Video**—A base-band analog video format in which the chroma and luma signals are carried separately to improve fidelity.

**TCO**—Total cost of ownership. This metric combines all the costs of a device from capital outlay to ongoing operational costs.

**TCP**—Transmission Control Protocol. This protocol provides payload multiplexing and end-to-end reliability for data transfers across a network. TCP packets are carried by IP packets. This is a layer 4, connection-based protocol. See Chapter 6.

**Timecode or time code**—A number of the form HH:MM:SS:FF (hours, minutes, seconds, frames) that defines the frame sequence in a video file/stream or film. For 29.97 frames per second systems, FF spans from 00 to 29, whereas for 25 frames per second systems, FF spans from 00 to 24. An example is 11:49:59:24, which is immediately followed by 11:50:00:00 one frame later. See Chapter 11.

**TLAN**—Transparent LAN. This is a MAN that is Ethernet based end to end.

**TOE**—TCP Offload Engine. A hardware accelerator that offloads TCP/IP stack processing from the main device CPU.

**Traffic system**—A software application for managing the precise scheduling of programming, commercials, and live events throughout a TV station broadcast day. The output of a traffic system is the on-air, second-by-second schedule.

**TS**—Transport Stream. This is a MPEG systems layer spec for carrying compressed audio, video, and user data. Some non-MPEG compression formats also have mappings into the TS wrapper.

**UDDI**—Universal Description, Discovery, and Integration protocol. UDDI is a specification for maintaining standardized directories of information about Web services, their capabilities, location, and requirements.

**UDP**—User Datagram Protocol. This protocol provides payload multiplexing and error detection (not correction) for end-to-end data transfers over IP. This is a layer 4, connectionless-based protocol. Compare to *TCP*. See Chapter 6.

**UHDV**—Ultra High Definition Video. An image system with a  $7,680 \times 4,320$  raster format that is being developed by the Japan Broadcasting Corp. (NHK). It has 16 times the resolution of HDTV at 1080i. It is expected to be in operation circa 2016.

**UMID**—Unique Material Identifier. A SMPTE standard, the UMID is an identifier for picture, audio, and data essence that is globally unique.

**URI**—Uniform Resource Identifier. In computing, a URI is a character string that names a resource. The most common URI is a URL—<http://www.google.com>, for example. RESTful Web services rely on URIs to identify each resource that enables CRUD-style manipulation.

**VANC**—Vertical ANCillary data field. See also *HANC*.

**VBI**—Vertical blanking interval. For NTSC, all the horizontal lines from 7 to 21 (field one) and from 270 to 284 (field two). These lines may carry nonvisual information such as time code, teletext, test signals, and closed caption text. For PAL, VBI spans lines 7–21 and 319–333.

**VC-1**—Video coding 1. A shorthand descriptor of the SMPTE 421M standard. It is based on Microsoft's WM9 video codec.

**VC-2**—Video coding 2. A shorthand descriptor of the tentative SMPTE 2042 standard. VC-2 defines a wavelet-based, intra frame video decoder for production applications. It provides coding at multiple resolutions including CIF, SDTV, and HDTV.

**VC-3**—Video coding 3. A shorthand descriptor for the SMPTE 2019 family of standards. Avid's DNxHD video production compression format is the basis of VC-3 supporting intra frame, 4:2:2, 10-bit sampling, and bitstream rates to 220 Mbps.

**VDCP**—Video Disc Control Protocol. This is commonly used to control video server operations.

**Vertical sync**—The portion of the video signal that triggers the receiver to start the vertical retrace, thereby bringing the raster in position to start the top line.

**Video key**—A video signal used to “cut a hole” in a second video signal to allow for insertion of a third video signal (the fill) into that hole.

**Video reference**—Typically, an analog composite or SDI video signal with a black image active area. It is also called “black burst.” It is distributed throughout a facility to any element that needs a common horizontal and vertical timing reference.

**Virtualization**—A technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources.

**VITC**—Vertical internal time code. A time code data structure described by SMPTE 12M and encoded in one or more lines of the VBI. See also *LTC*.

**VLAN**—Virtual LAN. A logical, not physical, group of networked devices. VLANs enable administrators to segment their networks (department or region) without physically rearranging the devices or network connections. VLANs are segmented at layer 2 in the protocol stack.

**VPN**—Virtual Private Network. A secure, end-to-end, private data tunnel across the public Internet.

**W3C**—World Wide Web Consortium. A vendor-neutral industry body that develops standards for the Web. Popular W3C standards include HTML, HTTP, XML, SOAP, Web services, and others.

**WAFS**—Wide area file services. WAFS products accelerate data transfers across WANs using caching and protocol emulation techniques.

**WAN**—Wide area network. A network that connects computers or systems over a large geographic area. See Chapter 6.

**WBEM**—Web-Based Enterprise Management Initiative. See Chapter 9.

**Web service**—A self-describing, self-contained unit of programming logic that provides functionality (the service) through a network connection. Applications access Web services using, for example, SOAP/XML without concern for how the Web service is implemented. Do not confuse Web services with the classic Web server; they rely on completely different software models.

**WMI**—Windows Management Instrumentation. See Chapter 9.

**WSDL**—Web Services Description Language. WSDL defines a Web service's functionality and data types. It is expressed using XML.

**XML**—eXtensible Markup Language. A data language for structured information exchange. Values are associated with tags, enabling the definition, validation, and interpretation of data elements between applications and systems.

**Y'**—See *Luma*.

**Y'CrCb**—Digital component signal set for uncompressed SD and HD video. See Chapter 11.

**Y'PrPb**—Analog component signal set for uncompressed SD and HD video. See Chapter 11.

# Index

Page numbers with “t” denote tables; those with “f” denote figures.

(Numerals<sup>1</sup>)

## A

AAF. *See* Advanced authoring format

Accounting management, 347t

ActiveX Data Objects, 184

Address mapping, for virtualization,  
89

Ad-ID, 295–296

ADO.NET, 183

Advanced authoring format, 41, 305  
definition of, 288

edit-in-place methods, 289–290,  
290f

file interchange, 289–290

import/export methods, 289–290,  
290f

MXF and, similarities between,  
290

operations provided by, 289

reference implementation,  
290–291

Advanced Media Workflow

Association (AMWA), 41

Advanced Televisions Systems

Committee (ATSC), 416

AES. *See* Audio Engineering Society

AES/EBU audio link, 308, 419–420

Aggregate array I/O rates, 105–107

Agility element, of media workflows,  
311–312

AMWA. *See* Advanced Media

Workflow Association

Analog broadcast standards, 415–416

Analog signals

composite video, 413–415

description of, 400

digitization of, 401f

Y'PrPb, 411

Analog-based systems

description of, 4, 400–401

illustration of, 24f

Ancillary data, 26, 27t

Antivirus software, 326, 331–332

AoE. *See* ATA over Ethernet

Application clients

definition of, 44

description of, 36, 83

Application functionality, 19

Archive exchange format, 116

Archive storage

devices for, 115

holographic, 118

magnetic tape systems for,

116–117

massive array of inactive discs,

118–119

optical systems for, 117–118

overview of, 115–116

Arrays

description of, 85

file striping effects on, 106–107

hard disk drive, 209–210, 210f

RAID, 208–212

storage, 97–100

striping, 106–107

virtual, 188

Aspect ratio, 406, 408–409

Asynchronous data links, 65

ATA hard disk drive

failure rate analysis for, 206

I/O convergence, 128–130

overview of, 15

SCSI hard disk drive vs., 127–128

service life of, 207

ATA over Ethernet, 141

Audio Engineering Society (AES), 41

Automatic telephone system, 10, 10f

Automation

broadcast, 314–315

definition of, 41

A/V application client, 44

A/V bit rate reduction

overview of, 424–426

techniques for, 426

video compression, 426–428

A/V connectivity, 147–148

A/V data

direct-to-storage real time access

to, 21

methods of moving, 21–23

streaming, 21

A/V delay, 26, 27t

A/V essence formats, 74–75

A/V lip-sync, 25

A/V media clients

class 1, 44

class 2, 45–46

class 3, 46–47

class 4, 47–48

description of, 44

A/V probe, 250

A/V processing, 26, 27t

A/V processor, 38–39

A/V signals, 78–80

A/V storage, 26, 27t

A/V stream, 21, 359

A/V systems

law of inertia applied to, 2

motivation toward, 5

performance metrics for, 25–28, 27t

traditional, 27t, 30

workflows for, 23, 25, 303

A/V timing, 23

Avert period, of security prevention,  
325

AV/IT system

advantages and disadvantages of,  
28–31

hybrid, 24f

infrastructure, forces that enable,  
6, 6f

<sup>1</sup>Numerals (i.e. 625) are indexed under their first digit (i.e. six).



AV/IT system (*Continued*)  
 performance metrics for, 25–28, 27t  
 schematic diagram of, 392–394, 393f  
 transition to, 23  
 AXF. *See* Archive exchange format

## B

Bandwidth  
 aggregate connection, 457–458  
 changes in, 8  
 controlled, 261  
 Fibre Channel, 136  
 Blade, 84  
 Blade servers, 463–464  
 Block size, 97–98  
 BPMN. *See* Business process modeling notation  
 Broadband Technology Committee, of SMPTE, 43  
 Broadcast automation, 314–315  
 Broadcast exchange format, 272–273  
 Broadcast inventory management system, 391  
 Buffer, 153  
 Buffering, 76–77, 153  
 Business process modeling notation, 306  
 BWAV format, 426  
 BXF. *See* Broadcast exchange format  
 Byte striping, 213

## C

CA. *See* Certificate authorities  
 Cache  
 A/V data, 154–155  
 buffer vs., 153  
 definition of, 153  
 description of, 146, 148  
 in shared storage system, 154f  
 Cache coherency, 153  
 Cache efficiency, 155  
 Cache hit, 153  
 Cache location, 153  
 Carrier Ethernet, 259–260  
 Case studies  
 KQED, 382–384  
 PBS, 385–389

Turner Entertainment Networks, 389–392  
 CDP. *See* Compressed domain processing  
 Centralized computing, 162, 179  
 Centralized enterprise reporting stations, 357–359  
 Certificate authorities, 342  
 CFS. *See* Clustered file systems  
 Chroma decimation, 411–413  
 Chua, Leon, 466  
 CIM. *See* Common information model  
 Circuit routing, 238f  
 Class 1 media client, 44  
 Class 2 media client, 45–46  
 Class 3 media client, 46–47  
 Class 4 media client, 47–48  
 Client(s)  
 application  
 definition of, 44  
 description of, 36, 83  
 edge, 226  
 remote, 226  
 servers vs., 164–165  
 Client local caching, 225–226  
 Client transactions, 96–97  
 Client-based servers, 16  
 Client/server systems  
 description of, 163–166, 179  
 features of, 164–165  
 scalability of, 165  
 Cluster(s)  
 definition of, 439  
 RAID, 216–219  
 scaling of, 218–219  
 Cluster computing, 84, 441  
 Clustered file systems (CFS)  
 captive, 95  
 description of, 68–69, 90–91, 132, 151  
 distributed file systems vs., 93  
 storage area networks with, 142  
 Storage Foundation, 94  
 virtualization vs., 94–95  
 Command execution modes, 273–274  
 Commercial off the shelf, 379–380  
 Common information model, 365–367, 366f

Common Internet file system, 144  
 Complex transaction, 96  
 Component color difference signals, 410–411  
 Composite video signal, 413–415  
 Compound growth rate of storage density, 13  
 Compressed domain processing, 423–424  
 Compression  
 methods of, 40  
 packing density affected by, 40  
 video. *See* Video compression  
 Computing  
 centralized, 162, 179  
 cluster, 84, 441  
 distributed. *See* Distributed computing  
 grid. *See* Grid computing  
 peer-to-peer, 177–179  
 SMP, 441  
 utility, 227, 439, 441–442  
 virtual, 227  
 Computing power  
 doubling trend in, 9  
 societal demand for, 9–10  
 for video processing, 10–11  
 Concealment, 59, 227  
 Configuration management, 347t, 370–371  
 Congestion management, 262  
 Connectivity  
 CIFS, 144–145  
 direct attached storage, 125f  
 HTTP, 145  
 I/O, 126t–127t, 126–128  
 NFS, 145  
 storage, 46  
 WAN  
 network, 258  
 overview of, 256  
 point-to-point form of, 256  
 topologies, 256–258  
 Content management, 296  
 Control layer, 71  
 Control plane, 269, 270–277, 315, 460–461  
 COTS. *See* Commercial off the shelf  
 CPUs  
 clock speed increases for, 11

- computing power of, 9–12
- doubling trend in, 9
- future of, 11–12
- parallel operation of, 11
- specialized types of, 11–12
- trillion operations per second by, 11
- workflow improvements secondary to advancements in, 11
- Cryptography
  - digital signatures, 343–344
  - encryption methods, 336–338
  - history of, 336
  - Kerberos, 342
  - keys, 338–343
  - overview of, 335–336
- Customer period, of security prevention, 326
- CXFS, 94
- D**
  - DAS. *See* Direct attached storage
  - DAT. *See* Digital audio tape
  - Data center markup language, 371–372
  - Data encryption standard, 336–338
  - Data layer, 71
  - Data mirror, 223
  - Data plane, 72, 460
  - Data routing, 222, 223f
  - Data wheel file transfer, 56
  - Database connectivity, 182–183
  - Data/user plane, 277–278
  - DAVIC, 56
  - DAWS. *See* Digital audio workstation
  - D-Cinema Technology Committee, of SMPTE, 43
  - DCML. *See* Data center markup language
  - DCT. *See* Discrete cosine transform
  - Deduplication, 110–111
  - Deinterlacing, 422
  - Delay, 260
  - Denial-of-service attacks, 320
  - DES. *See* Data encryption standard
  - Descriptive metadata, 282–283
  - Design element, 304–307
  - Deterministic control, 26, 27t
  - Deterministic frame forwarding, 240
  - Device management, 350
  - DFS. *See* Distributed file systems
  - Diffserv, 264
  - Digital asset management, 296
  - Digital audio tape, 116
  - Digital audio workstation, 47
  - Digital broadcast standards, 416–417
  - Digital hierarchies, 447–449
  - Digital raster image, 405–406
  - Digital rights management, 301–302
  - Digital signal processors, 192
  - Digital signatures, 343–344
  - Digitally based systems, 4
  - Direct attached storage
    - connectivity examples for, 125f
    - description of, 74, 123
    - protocols, 124–126
    - SCSI, 123–124
    - types of, 125–126
    - USB, 125, 125f
  - Direct memory access
    - definition of, 130–131
    - remote, 130–131
  - Direct R/W transaction, 96
  - Direct to storage
    - concepts regarding, 68–71
    - description of, 49
    - real time access, 21
  - Disc storage systems, 85
  - Discrete cosine transform (DCT), 430–431
  - Distributed computing
    - client/server class, 163–166
    - description of, 162–163
    - environment for, 170f
    - peer-to-peer computing, 177–179
    - service-oriented architecture, 166–169
    - Web services model. *See* Web services model
  - Distributed file systems
    - clustered file systems vs., 93
    - description of, 93
    - Microsoft, 93
  - Distributed management task force, 365
  - DMA. *See* Direct memory access
  - DMTF. *See* Distributed management task force
  - Domain name server, 241
  - Drop frame, 434
  - Drop frame time code, 433–434
  - E**
    - ECC. *See* Electronic correction code
    - Eclipse, 191
    - Edge clients, 226
    - Edge servers, 71
    - EFM. *See* Eight to fourteen modulation
    - Eight to fourteen modulation, 57
    - 8B/10B line coding, 445–446
    - E-LAN, 259–260
    - Electronic correction code, 58
    - Encryption, 336–338
    - Enterprise servers, 16
    - Entropy coding, 425
    - Erdos, Paul, 7
    - Error correction
      - forward, 56–59, 387
      - methods for, 58
    - eSATA, 125
    - Ethernet
      - ATA over, 141
      - bandwidth changes, 8
      - Carrier, 259–260
      - description of, 234
      - evolution of, 236f, 259
      - Fibre Channel over, 140–141
      - history of, 234–235, 236f
      - IEEE-defined Gigabit links, 235–236
      - LANs, 39
      - specifications, 235
      - time synchronous, 29
    - Ethernet frames
      - description of, 236f, 236–237
      - tagging, 262
    - Ethernet switch, 29, 37–38, 467–468
    - ETSI. *See* European Telecommunications Standards Institute
    - European Nuggets, 29
    - European Telecommunications Standards Institute, 41
    - Exabyte, 443
    - Extensible access method, 110
    - Extensible markup language. *See* XML
    - External storage, 461

**F**

Fabric application interface standard, 89

Fabric switches, 136–137

Failover, 200

Failure

- internal points of, 203f, 203–204
- mechanisms of, 201–205

FAIS. *See* Fabric application interface standard

FastTCP, 255–256

Fault(s)

- automatic detection of, 200
- detection of, 200–201
- repair of, 200–201
- self-healing of, 200

Fault management, 347t

Fault-tolerant system, 198–200

FCAPS model, 347–348

FCIP, 139

FCoE. *See* Fibre Channel over Ethernet

Fibre Channel

- arbitrated loop, 136
- arrays, 85
- bandwidth of, 136
- configuration of, 134–135
- definition of, 134
- description of, 46, 74
- director, 137
- features of, 134
- layers of, 135
- protocol stack, 135, 135f
- storage area network
  - characteristics of, 131–132, 134–137
  - IP storage area network vs., 138t
  - switching fabric, 84–85, 136–137

Fibre Channel over Ethernet (FCoE), 140–141

Field failure domain, 207

File-based operations, 3, 21–22

File conversion gateway, 287–288

File formats

- native, 285
- translation of, 74–75

File fragmentation, 100

File gateway, 39

File inspector, 360

File server, 38, 334–335

File sharing, 90

File striping

- array performance and, 106–107
- definition of, 92

File systems

- clustered. *See* Clustered file systems
- description of, 90–91
- distributed. *See* Distributed file systems
- general parallel, 94, 150
- installable, 91
- network, 92, 144–145
- parallel network, 92

File transfer

- audiovisual data moved via, 21
- characteristics of, 50–51, 52t
- concepts, 50–62
- data wheel, 56
- delivery chain, 57
- description of, 49
- general-purpose, 51–52, 52t
- information technology changes, 8
- interoperability domains, 73–74
- just in time, 52–53
- management information base, 359–360
- methods of, 54t
- progress monitoring of, 359–360
- reliable, 53–60
- shared storage model created
  - using, 70–71
- streaming vs., 62t
- UDP without FEC for, 59–60
- unidirectional links for, 57

File transfer input/output, 22

File wrapper

- description of, 74–75
- format, 279f

Files Structures Technology Committee, of SMPTE, 44

Film Technology Committee, of SMPTE, 43

Financial domain, 207

Firewalls

- description of, 38, 233, 326
- mechanism of action, 328
- worm threats prevented by, 325

Flash. *See* RAM/Flash

Forward error correction (FEC), 56–59, 387

4:1:1, 412

4:2:0, 412

4:2:2, 412

4:4:4, 412

4:4:4:4, 412

4F<sub>SC</sub>, 413

Fragmentation

file, 100

free space, 100

Frame(s)

Ethernet, 236f, 236–237, 262

jumbo, 236–237

Frame accuracy, 25, 437–438

Frame forwarding, deterministic, 240

Frame synchronizer, 437–438

Free space fragmentation, 100

FTP, 53–54

FTP/TCP method, 55–56

**G**

Gamma correction, 410

Gamma-corrected signal, 410

Gateway

definition of, 286

file conversion, 287–288

Gen Lock input signal, 420–421

General parallel file system, 94, 150

General-purpose file transfer, 51–52, 52t

Generational loss, 309

GIF, 425

Glitching, 26, 27t

GOP, 432

Graphical user interfaces, 19

Graphics processing unit, 158, 192

Grid, 439

Grid computing

description of, 84, 439–441

environments for, 440f

Riemann zeta function and, 440–441

GridFTP, 54t

Group of linked files concept, 283–285

**H**

H.264, 427–428

Hamming error correction codes, 213

- Hard disk drives
  - arrays, 209–210
  - ATA. *See* ATA hard disk drive
  - capacity of, 103–104
  - data access rate, 103–105
  - description of, 12
  - internal, 14
  - I/O connectivity, 126t–127t, 126–128
  - I/O delivery rate, 103
  - performance metrics for, 97–98
  - power requirements, 14
  - price reductions for, 13
  - read/write rates, 14
  - rebuild efforts, 212
  - reliability metrics for
    - failure rate analysis domains, 205–207
    - field failure domain, 207
    - financial domain, 207
    - lab test domain, 205–207
    - overview of, 205
  - SCSI. *See* SCSI hard disk drive
  - solid state disc vs., 465
  - storage density of, 13–14
- HD-SDI, 419
- Hierarchical storage, 111
- Hierarchical switching, 235f
- High-availability systems
  - architectural concepts for
    - $N + 1$  sparing, 219, 221–222
    - no single point of failure, 219f, 220–221
    - overview of, 219
    - single point of failure, 220
  - cluster computing for, 441
  - concealment methods for, 227
  - description of, 305
  - design methods
    - overview of, 208
    - RAID arrays, 208–219
  - failure mechanisms, 201–205
  - mirroring methods for, 223–225
  - networking with, 222–223
  - overview of, 198
  - scaling of components in, 228–230
  - summary of, 227
  - topologies for, 227
  - upgrading of components in, 229–230
- High-quality, high-rate streaming, 68
- Holographic storage, 118
- HTTP. *See* Hypertext transfer protocol
- H/V timing, 78, 80
- HyperMAP, 59
- Hypertext transfer protocol, 145, 176, 181
- Hypervisor, 189
- I**
  - IBP, 429
  - IEEE-1394, 2, 46
  - IEEE 802.1 Audio/Video Bridging (AVB) Task Group, 468
  - IEEE-1588 Precision Time Protocol, 66
  - IETF. *See* Internet Engineering Task Force
  - iFCP, 139
  - ILM. *See* Information life cycle management
  - Industry Standard Commercial Identifier (ISCI), 295–296
  - InfiniBand, 131
  - Information life cycle management, 114
  - Information security, 318–319. *See also* Security
  - Information technology. *See* IT
  - Ingest and playout system, 83
  - Installable file system, 91
  - Interactive streaming, 67–68
  - Interformats, 429
  - Interframe coding, 429
  - Interframe video compression, 431–432
  - Interlaced raster, 402
  - Internal storage, 461
  - International Data Encryption Algorithm encrypt engine, 338
  - International Disk Drive Equipment and Materials Association, 205–206
  - International Telecommunications Union, 41
  - Internet Engineering Task Force (IETF), 41
  - Internet FCP, 139
  - Internet firewall, 38
  - Interoperability
    - domains of, 72–75
    - MXF and, 285–288
  - Intraframe coding, 429
  - Intraframe video compression, 430–431
  - Intrusion detection systems, 330–331
  - Intrusion prevention systems, 38, 328–330
  - I/O
    - for ATA hard disk drive, 128–130
    - description of, 22
    - hard disk drive connectivity, 126–128
    - for SCSI hard disk drive, 128–130
  - IP
    - description of, 232
    - techniques for control over, 274
  - IP addressing, 240–241, 241f
  - IP multicasting, 243–244
  - IP router, 233
  - IP routing layer
    - IP addressing, 240–241, 241f
    - IPV6, 242–243
    - layer 2 and layer 3 switching, 239–240, 242
    - multicasting, 243–244
    - overview of, 237–239
    - private IP addresses, 242–243
    - subnets, 241–242
  - IP storage area network, 138–141
  - IP switching, 83
  - IPS. *See* Intrusion prevention systems
  - IPV6, 242–243
  - ISAN, 295
  - ISCI, 295–296
  - iSCSI
    - description of, 85, 124
    - storage area network systems, 138–139, 141
    - terminology associated with, 142
  - Isochronous links, 64
  - IT
    - architecture of, 34–35, 35f
    - components of
      - description of, 75–76
      - repair of, 378
      - upgrading of, 378

IT (*Continued*)

- infrastructure, 346
  - life cycle of, 378
  - media
    - business-related factors for
      - using, 5–6
    - motivation toward, 5–7
    - technology-related factors for
      - using, 5
  - objections to, 75–76
  - six-tier architecture of, 34–35, 35f
  - streaming using, 73–74
  - systems
    - interoperability of, 17–19
    - manageability of, 15–16
  - transition to
    - case studies of. *See* Case studies
    - disruptions during, 381
    - financial issues, 376
    - issues in, 374–375
    - organizational issues, 375–376
    - technical support issues, 377
    - user issues, 380–381
  - video system, 394–397
- IT/AV system
- hybrid, 24f
  - infrastructure, forces that enable,
    - 5, 6f
  - performance metrics for, 25–28,
    - 27t, 28–31
  - schematic diagram of, 392–394,
    - 393f
  - transition to, 23
- ITU. *See* International Telecommunications Union

**J**

- Java database connectivity, 183
- Java EE, 16, 185–186
- JBOD, 108–109
- JDBC, 183
- Jitter, 260
- JPEG2K, 431
- Jumbo frames, 236–237
- Just in time file transfer, 52–53, 461
- JXTA, 191

**K**

- Kerberos, 342
- Kettering, Charles, 31

- Key/length/value blocking, 280–281
- Keys, 338–343
- KQED case study, 382–384

**L**

- Lab test domain, 205–207
- Label edge routers, 265
- Label switched paths, 262
- LAMP, 190
- LANs
  - configuration of, 39
  - description of, 2
  - Ethernet, 39
  - file transfer using, 21
  - real time control, 274–276
  - RS422 serial linking replaced with,
    - 28–29
  - streaming, 64
  - streaming audiovisual using, 21
  - transparent, 234
  - virtual, 249–251
- Law of inertia, 2
- Layer 2 switching, 239–240, 242
- Layer 3 switching, 239–240, 242
- Layer two tunneling protocol. *See* L2TP
- Legacy equipment and systems,
  - 378–379
- Lehman's laws, 193–194, 194t
- Lempel–Ziv–Welch algorithm, 425
- Linear time code, 433
- Link delays, 260
- Linux
  - marketshare by, 17
  - server clusters, 110
- Live A/V switching, 26, 27t
- Load balancing, 149
- Long fat pipe, 255
- Long group of pictures, 429
- Long-term archive, 86
- Look ahead buffer, 76, 78
- Look around buffer, 76, 78
- Look behind buffer, 76, 77
- Loosely coupled designs, 312–313
- Lossless encoding, 425–426
- Lossy
  - definition of, 424
  - video compression techniques
    - interframe, 431–432
    - intraframe, 430–431

- overview of, 429
- LTC. *See* Linear time code
- LTO, 116
- L2TP, 333
- L2TP/IPSec, 333–334
- Lustre FS, 95

**M**

- Magnetic tape systems, for archive
  - storage, 116–117
- MAID. *See* Massive array of inactive discs
- Malware, 321–322
- MAN. *See* Metropolitan area networking
- Management information base
  - (MIB), 349–350, 352, 354–355, 362–363
- Management layer, 71
- Management plane, 269, 277, 461
- Management systems
  - broadcast inventory, 391
  - illustration of, 347f
  - users of, 346
- Massive array of inactive discs,
  - 118–119
- Mean time between failures (MTBF)
  - definition of, 198, 201
  - for hard disk drives, 205
- Mean time to repair (MTTR),
  - 198–199, 200–201
- Media asset, 296
- Media asset management
  - with A/V editing gear, 299
  - components used in, 297–298
  - description of, 39, 166, 296–297
  - digital rights management used in,
    - 301–302
  - examples of, 298–301
  - functionality of, 302–303
  - functions of, 298–299
- Media object server protocol, 270
- Media system
  - overview of, 268
  - planes of
    - control, 269, 270–277
    - data, 269, 277–278
    - management, 269, 277
    - overview of, 268–270
- Media workflows

- agility element, 311–312
- design element, 304–307
- documentation methods, 306–307
- elements of, 303–314
- flow efficiencies, 309
- flow types, 308–309
- loosely coupled designs, 312–313
- operational costs, 309–310
- operational element, 310
- platform examples, 313–314
- process orchestration element of, 307–310
- reliability, 305
- standards, 305–306
- takeaways, 314
- MediaGrid, 95
- Medium access control (MAC), 237
- MEF. *See* Metro Ethernet Forum
- Melio FS, 95
- Memristor, 466
- Metadata, 293–294
- Metadata inspector, 360–361
- Metadata Technology Committee, of SMPTE, 44
- Metcalfe's law, 18–19
- Metric drifting, 26, 27t
- Metro area network. *See* MAN
- Metro Ethernet Forum, 259
- Metropolitan area networking, 259
- Mezzanine compression, 427
- MIB. *See* Management information base
- Middleware
  - connectivity using
    - database, 182–183
    - example of, 180f
    - overview of, 180–183
  - definition of, 180
  - illustration of, 164f
  - protocol standards for, 182
- Mirrored playout, 224
- Mirrored record, 224
- Mirroring, 223–225
- Monitoring
  - A/V IP stream, 359
  - AV/IT environment, 352–361
  - description of, 379
  - file transfer progress, 359–360
  - IT device, 354–355
  - methods of, 348–352
  - network, 354–355
  - Moore's law, 9, 9f, 158
  - Motion artifacts, 403
  - Motion compensation, 431
  - Motion-compensated deinterlacing, 422
  - MPEG, 8, 192, 281, 423–424
  - MPEG-1, 427
  - MPEG-2, 427
  - MPEG-4, 427
  - MPEG-7, 293–294
  - MPLS
    - description of, 265
    - tagging, 263
  - MTBF. *See* Mean time between failures
  - MTTR. *See* Mean time to repair
  - Multicasting, 243–244
  - MXF
    - advanced authoring format and, similarities between, 290
    - compatibility, 286
    - compliance, 286
    - description of, 64
    - descriptive metadata, 282–283
    - File Package, 281
    - gateway, 286
    - group of linked files concept, 283–285
    - interchange environment, 285f
    - interoperability and, 285–288
    - logical view of, 281–282
    - Material Package, 281
    - specifications, 288
    - wrappers, 280–285
  - MySQL, 191

**N**

  - $N + 1$  reliability, 149
  - $N + 1$  sparing, 219, 221–222
  - $N + 2$  sparing, 221
  - NAS. *See* Network attached storage
  - NAT. *See* Network address translation
  - Near-line storage, 151
  - .NET
    - Active Server Pages, 184
    - ActiveX Data Objects, 184
    - description of, 16, 184
  - .NET Remoting, 182
  - Network
    - bandwidth of, 7–8
    - digitally based, 4
    - infrastructure of, 7–8
    - monitoring of, 354–355
    - quality of service for
      - classification techniques, 262–263
      - congestion management, 262
      - delay, 260
      - management techniques, 260–265
      - overview of, 260–261
      - pyramid, 264, 264f
      - reservation techniques, 263–264
    - WAN, 258
  - Network address translation, 242–243
  - Network attached storage (NAS)
    - attach protocols for, 144–145
    - A/V-friendly connectivity, 147–148
    - client access to, 83
    - commercial systems, 453
    - connectivity of
      - clustered server and, 148f
      - illustration of, 143f
    - definition of, 109
    - description of, 74, 123, 143
    - device components, 144f
    - future of, 150–153
    - $N + 1$  reliability, 149
    - operating system of, 146
    - security issues, 335
    - server clustering and, 148–150
    - storage area network and, 123, 143–144, 150, 151–153
    - vendors of, 145–146
    - WANs and, 146–147
  - Network file system (NFS), 92, 144–145
  - Network ID, 241
  - Network level type of service tagging, 262–263
  - Network path quality of service, 61f
  - Networked media
    - core elements of
      - application client, 36
      - Ethernet switch, 37–38
      - firewall, 38
      - intrusion prevention system, 38

- Networked media (*Continued*)
    - networking infrastructure, 39–40
    - overview of, 35
    - router, 36–37
    - servers, 38–39
    - software, 41
    - storage subsystems, 39
  - definition of, 2
  - examples of, 3
  - motivation toward, 5–7
  - systems of, 4
  - Networked-based systems
    - description of, 4
    - performance metrics for, 25–28, 27t
  - Network/Facilities Infrastructure Technology Committee, of SMPTE, 44
  - Networking
    - high-availability systems, 222–223
    - infrastructure for, 64
    - overview of, 232
    - seven-layer stack
      - application layer, 233
      - design of, 233f
      - IP routing layer. *See* IP routing layer
    - overview of, 232–234, 233f
    - peer-to-peer relationships, 233, 233f
    - physical layer, 234–237
    - transport layer, 233, 244–249
  - Newton, Sir Isaac, 2
  - Next generation interconnect system, 386, 388–389
  - NFS. *See* Network file system
  - NGIS. *See* Next generation interconnect system
  - No single point of failure (NSPOF)
    - description of, 201
    - dual single point of failure elements for, 220
    - high availability system design using, 219f, 220–221
    - stand-alone devices, 220
  - Node(s)
    - file system access by, 91
    - redundant, 462
    - redundant array of independent, 454
  - Non-drop frame, 434
  - Nonlinear editors, 36, 46–47
  - Nonreal time
    - control of, 275
    - definition of, 22
    - real time vs., 47
    - transfers, 52t
  - Nonreal time storage pool, 46
  - NRT. *See* Nonreal time
  - NX, 331
  - Nyquist sampling theorem, 400
- ## O
- OASIS. *See* Organization for the Advancement of Structured Information Standards
  - Object ID, 362
  - Object storage, 109–111
  - ODBC, 183
  - Offsite mirror, 224
  - Open database connectivity. *See* ODBC
  - Open source software, 190–191
  - Open systems interconnection, 232–233
  - Operating systems
    - real time performance, 191–193
    - Windows management instrumentation, 367–368
  - Operational element, of media workflows, 310
  - Optical systems, for archive storage, 117–118
  - Organization for the Advancement of Structured Information Standards, 371–372
  - OSI. *See* Open systems interconnection
- ## P
- Packet encapsulation, 249f
  - Packet loss, 261
  - Packet Over SONET, 234
  - Packet routing, 237, 238f
  - Packet switches, 37
  - Packet switching, 237, 239
  - Packing density, 40
  - Parallel network file system, 92
  - PBS case study, 385–389
  - PCI Express bus, 12
  - PDH. *See* Plesiochronous digital hierarchy
  - PDU. *See* Protocol data unit
  - Peer-to-peer computing, 177–179
  - Peer-to-peer relationship, 248–249
  - Performance management, 347t
  - Pergamum, 119
  - Petabyte, 443
  - Physical layer, 234–237
  - Physical links, 124
  - Pixel group motion, 431
  - PKI. *See* Public key infrastructure
  - Plesiochronous digital hierarchy, 447–448
  - Plesiochronous links, 65
  - pNFS. *See* Parallel network file system
  - Point to multipoint, 80
  - Pop-up, 321
  - Power consumption, 12
  - Power supply failure, 203
  - Precision Time Protocol, 66
  - Preroll time, 275
  - Prime number theorem, 440
  - Private IP addresses, 242–243
  - Private keys, 339–343
  - Professional signal formats, 417–418
  - Progressive raster scan, 402
  - Proteus clip server, 420–421
  - Protocol data unit, 249
  - Proxy client, 56
  - Public key infrastructure, 342
  - Public keys, 339–343
  - Public Switched Telephone Network, 72
  - Push-and-pull streaming, 67
- ## Q
- QoS. *See* Quality of service
  - Quality of service
    - concepts for, 61–62
    - definition of, 61–62
    - network
      - classification techniques, 262–263
      - congestion management, 262
      - delay, 260
      - management techniques, 260–265

- overview of, 260–261
- pyramid, 264, 264f
- reservation techniques, 263–264
- network path, 61f
- parameters for, 61
- performance, 311

Quarantine file server, 334

## R

### RAID

- arrays, 208–212
- A/V workflows and, 216
- calculations, 216
- clusters, 216–219
- configuration of, 209–210
- controllers, 211–212
- description of, 100, 109, 146
- evaluative factors for, 212
- hard disk drive arrays for, 209–210, 210f
- history of, 208–209
- level 0, 212–213, 215
- level 01, 213
- level 1, 213, 215
- level 2, 213, 215
- level 3, 213–214, 215
- level 4, 214
- level 5, 214
- level 6, 215
- level 10, 213
- level 30, 213
- master controller, 211
- two-dimensional parity methods, 210–212

RAIN. *See* Redundant array of independent nodes

RAM/Flash, 12

Random early detection, 262

RDMA. *See* Remote direct memory access

Read/write operations, for solid state disc, 466

### Real time

- definition of, 22
- LAN-based immediate control of, 276–277
- LAN-based scheduled control of, 274–276
- MPEG encoding/decoding in, 192
- nonreal time vs., 47

- operating system performance in, 191–193

### Real-time storage

- access to, 70t
- description of, 39
- users of, 47

Real-time storage pool, 46

Real-time streaming protocol (RTSP), 67–68

Redundant array of independent nodes (RAIN), 454

Redundant arrays of inexpensive disks. *See* RAID

Redundant nodes, 462

Reed-Solomon coding, 57, 58

Registries Technology Committee, of SMPTE, 44

### Reliability

- hard disk drive, metrics for
  - failure rate analysis domains, 205–207
  - field failure domain, 207
  - financial domain, 207
  - lab test domain, 205–207
  - overview of, 205
- media workflow designs for, 305
- metrics of, 198–200
- $N + 1$ , 149
- software, 19–20, 204, 204f
- video server, 462

Reliable file transfer, 53–60

Remote access, 334

Remote clients, 226

Remote direct memory access, 130–131

Remote method invocation, 182

Remote procedure call, 182

Reporting stations, 357–359

Resource reservation protocol, 263–264

RESTful services model, 175–176

Return on investment, 376

RGB signal, 410

R'G'B' signal, 410, 415

Riemann Conjecture, 441

Riemann zeta function, 440–441

Rights management, 297

RMI. *See* Remote method invocation

### Router

- as firewall, 38

- function of, 36–37
- layer 3 switch vs., 240
- video, 467–468
- WAN interfaces, 37

RP210 metadata dictionary, 305

RPC. *See* Remote procedure call

RS422 serial link, 28

RSA algorithm, 341

RT. *See* Real time

## S

SAIT, 116

SAN. *See* Storage area network

SAS connective, 129

### Scalability

- of client/server systems, 165
- of high-availability systems, 228–230
- of video server, 462

Scaling of clusters, 218–219

### SCSI

- block commands, 123
- description of, 2, 123–124
- history of, 123
- primary commands, 123
- standards, 124f
- stream commands, 123

### SCSI hard disk drive

- ATA hard disk drive vs., 127–128
- description of, 15
- failure rate analysis for, 205–206
- I/O convergence, 128–130
- service life of, 207

SDH. *See* Synchronous digital hierarchy

### SDI link

- description of, 66, 401, 418–421
- line rates for, 451–452

SeaChange Broadcast MediaCluster, 95

Secure sockets layer, 333–334

### Security

- boundaries for, 322f
- elements of, 318
- foundations of, 319
- media enterprise, 334–335, 335f
- overview of, 318–319
- software, 204–205

Security management, 347t

Security plan, 323–325



- Security prevention
  - antivirus software, 326, 331–332
  - firewalls, 326–328
  - intrusion detection systems, 330–331
  - intrusion prevention systems, 328–330
  - tactics for, 322–326
  - virtual private network, 332–334
- Security threats
  - denial-of-service, 320
  - life cycle of, 320
  - malware, 321
  - overview of, 320
  - pop-ups, 321
  - prevention of. *See* Security prevention
  - spyware, 321
  - Trojan horse, 321
  - viruses
    - antivirus software for, 326, 331–332
    - description of, 321
    - window of vulnerability for, 325–326
  - window of vulnerability for, 325–326
  - worms
    - description of, 321
    - firewall protection against, 325
    - NX for, 331
    - virtual software patch for, 329–330
- Self-healing systems, 200
- Serial attached SCSI, 124
- Server(s)
  - blade, 84, 463–464
  - client-based, 16
  - clients vs., 164–165
  - definition of, 165
  - description of, 38, 83–84
  - edge, 71
  - file, 38
  - quarantine file, 334
  - transparency of location, 165
  - video. *See* Video server
  - virtual, 188–189, 189f
  - Web service hosted by, 169
- Server clustering, 148–150
- Server subsystem, 83–84
- Service
  - definition of, 169
  - example of, 171f
  - Web services model. *See* Web services model
- Service diagnostics, 368–371
- Service level agreement (SLA), 260
- Service-oriented architecture (SOA), 166–169, 180
- Shared storage
  - barriers to industry adoption of, 151
  - caching in, 154f
  - file transfer used to create model of, 70–71
  - model for, 69
  - real time, 70t
  - virtual model, 70–71
- Signaling System protocol, 72
- Simple network management protocol, 349, 363–365
- Simple object access protocol, 173, 182
- Single instance storage, 110–111
- Single point of failure, 220
- Single storage pool, 39
- 64-bit architectures, 192–193
- SLA. *See* Service level agreement
- Sliding window approach, 246–248
- SMI. *See* Storage management initiative
- SMP. *See* Symmetric multiprocessing
- SMPTE. *See* Society of Motion Picture and Television Engineers
- SNIA, 110
- SNMP. *See* Simple network management protocol
- SOA. *See* Service-oriented architecture
- SOAP. *See* Simple object access protocol
- Society of Motion Picture and Television Engineers
  - archive exchange format, 116
  - description of, 43
  - standards of, 43
  - technology committees, 43–44
- Sockets, 245
- Software
  - classification of, 41
  - Lehman's laws of evolution for, 193–194, 194t
  - maintenance of, 193f, 193–194
  - open source, 190–191
  - overview of, 158–159
  - reliability of, 19–20, 204, 204f
  - scalability of, 19–20
  - security of, 204–205
  - storage, 86
  - user application requirements, 159–160
  - user interface design principles for, 160t
- Software architecture
  - centralized computing, 162, 179
  - definition of, 161
  - description of, 16–17, 160–180
  - distributed computing
    - client/server class, 163–166
    - description of, 162–163
    - environment for, 170f
    - peer-to-peer computing, 177–179
    - service-oriented architecture, 166–169
  - Web services model. *See* Web services model
- implementation frameworks for
  - Java EE, 185–186
  - .NET, 184
  - overview of, 183
- 64-bit, 192–193
- 32-bit, 192
- Software patch, for worm threats, 329–330
- Software service, 171f
- Solid state disc, 13, 465–466
- SONET, 65, 257, 447–449
- SPC-1, 101–102
- Spyware, 321
- SQL, 183
- SSD, 13, 465
- SSL. *See* Secure sockets layer
- Stack, 248–249
- Stand-alone servers, 461
- Standard(s)
  - analog broadcast, 415–416
  - digital broadcast, 416–417
  - purpose of, 41–44
- SCSI, 124f

- Society of Motion Picture and Television Engineers, 43
- systems management
  - management information base, 362–363
  - overview of, 361–362
  - simple network management protocol, 363–365
  - Web-based enterprise management, 365–367
  - Windows management instrumentation, 367–368
- Standard media exchange framework, 293
- Storage
  - archive. *See* Archive storage
  - commercial off the shelf, 108
  - direct attached. *See* Direct attached storage
  - external, 461
  - functions of, 133–134
  - hierarchical, 111
  - holographic, 118
  - internal, 461
  - management of, 114–115
  - methods of, 12
  - object, 109–111
  - replication of, 225
  - shared. *See* Shared storage
  - software for, 86
- Storage area network (SAN)
  - building of, 132–133
  - with cluster file systems, 142
  - commercial systems, 453
  - description of, 122, 131–133
- Fibre Channel
  - characteristics of, 131–132, 134–137
  - IP storage area network vs., 138t
- function of, 133–134
- future of, 150–153
- hybrid, 137
- IP, 138–141
- iSCSI, 138–139, 141–142
- network attached storage (NAS)
  - and, 123, 150, 151–153
- security issues, 335
- TCP/IP, 138–139
- vendors of, 143
- with virtualization, 142
- Storage area networking, 109
- Storage array data, 97–100
- Storage bandwidth, 12
- Storage connectivity, 46
- Storage density
  - compound growth rate of, 13
  - definition of, 12–13
- Storage layer, 85–86
- Storage management initiative, 356–357
- Storage mirror, 224
- Storage Networking Industry Association, 355–356
- Storage Performance Council, 101–102
- Storage subsystems
  - description of, 39
  - interfacing of, 74
  - JBOD, 108–109
  - networked attached storage, 109
  - object storage, 109–111
  - RAID, 109
  - requirements for, 108
  - storage area networking, 109
  - tiered, 111–113
  - types of, 103
  - video server, 461
- Storage switching layer, 84–85
- Storage systems
  - benchmarks for, 101–102
  - disc-based, 85
  - example of, 453–455
  - infrastructure of, 82f
  - overview of, 82–86
  - performance factors, 96–97
  - RAM-based, 98
  - requirements for, 108
  - for transactions, 96–97
- Storage transactions, 96–97
- Storage virtualization, 87–89
- Store and forward, 51
- Streaming
  - A/V, 21, 49, 66
  - best effort for, 63
  - delivery methods, 64–65
  - examples of, 62
  - file transfer vs., 62t
  - high-quality, high-rate, 68
  - information technology for, 73–74
  - interactive, 67–68
  - LAN-based, 64
  - pathways for, 64–65
  - peer-to-peer connections for, 457–458
  - push-and-pull, 67
  - real time protocol, 67–68
- Striping
  - array, 106–107, 217f, 217–218
  - byte, 213
  - definition of, 92
- Subnets, 241–242
- S\_video signal, 415
- “Swarm,” 60
- Switches
  - Ethernet, 29, 37–38, 467–468
  - fabric, 136–137
  - internal data routing structure, 37
  - packet, 37
  - simplicity of, 37–38
- Switching
  - layer, 2, 239–240, 242
  - layer, 3, 239–240, 242
  - packet, 237, 239
- Switching fabric, 84–85, 136–137
- Symmetric multiprocessing, 439, 442
- SYN flood, 320
- Synchronous communications, 65–66
- Synchronous digital hierarchy (SDH), 447–449
- Synchronous links, 65, 66
- Systems management
  - description of, 26, 27t, 379
  - device control and, 40–41
  - FCAPS model of, 347–348
  - overview of, 346
  - service diagnostics, 368–371
  - standards for
    - management information base, 362–363
    - overview of, 361–362
    - simple network management protocol, 363–365
    - Web-based enterprise management, 365–367
    - Windows management instrumentation, 367–368
  - users of, 346

**T**

## Tagging

- Ethernet frame, 262
- MPLS, 263
- network level type of service, 262–263

## Tape

- archive storage uses of, 116–117
- description of, 13

## TCP

- connection establishment, 245–246
- definition of, 53, 244
- denial-of-service attacks, 320
- description of, 245–246
- FastTCP, 255–256
- FTP/TCP method, 55–56
- limitations, 252
- offload engines, 255
- sliding window approach, 246–248
- SNMP vs., 365
- UDP and, 244f, 248
- TCP accelerator, 233
- TCP packet, 245, 245f

## TCP/IP

- congestion control, 251
- CPU stack processing of, 252
- performance of, 252–256
- speed of, 254–255
- storage area network, 141–142
- throughput of, 253f, 255
- Television Technology Committee, of SMPTE, 43
- Thick client, 163
- Thin client, 163
- 32-bit architectures, 192
- Threats. *See* Security threats
- 3:1:1, 412
- Time code, 433–434
- Time code accuracy, 26, 27t
- Time related label, 434
- Timing
  - audiovisual, 23
  - video signal, 404–405
- Training, 381
- Transactions, 96–97
- Transparent LANs, 234
- TRL. *See* Time related label
- Trojan horse, 321

- Turner Entertainment Networks case study, 389–392
- Two-dimensional parity, 210–212
- 2N, 435–436

**U**

## UDDI, 174

## UDP

- advantages of, 248
- definition of, 53
- description of, 248
- file transfers using, 59–60
- SNMP vs., 365
- TCP and, 244f, 248
- without FEC, 59–60
- UHDV. *See* Ultra high definition video
- Ultra high definition video, 407–408
- UMID. *See* Unique material identifier
- UML. *See* Unified modeling language
- Unidirectional links, 57
- Unified modeling language, 306, 307f

- Uniform resource identifiers, 175
- Unique material identifier, 294–295
- Upgrades/upgrading, 229–230, 378
- URIs. *See* Uniform resource identifiers
- USB, 2, 125, 125f
- User access rights, 68
- User application
  - functionality of, 19
  - requirements for, 159–160
  - software, 159–160
- User datagram protocol. *See* UDP
- User layer, 71
- User metadata, 26, 27t
- User training, 381
- Utility computing, 227, 439, 441–442

**V**

- VBI. *See* Vertical blanking interval
- Vendor(s)
  - lock-in of, 380
  - networked attached storage, 145–146
  - storage area networks, 143

- Vertical blanking interval, 405
- Vertical interval time code, 433
- Video compression
  - description of, 426–428
  - lossy
    - interframe techniques, 431–432
    - intraframe techniques, 430–431
    - overview of, 429
- Video disk control protocol, 270
- Video keying, 25, 422
- Video processing, 10–11
- Video reference, 419
- Video reference accuracy, 26, 27t
- Video resolutions, 405–409
- Video routers, 467–468
- Video server
  - architectures of, 461
  - description of, 459
  - reliability of, 462
  - scalability of, 462
  - stand-alone, 461
  - storage subsystems, 461
  - three planes of, 460–461
- Video Services Forum, 258
- Video signal(s)
  - component color difference signals, 410–411
  - processing of
    - compressed domain processing, 423–424
    - deinterlacing, 422
    - interlaced to progressive conversion, 422, 423f
    - overview of, 421–422
    - standards conversion, 422–423
  - professional formats, 417–418
  - representations, 409–418
  - RGB, 410
  - R'G'B', 410, 415
  - S\_video signal, 415
  - timing, 404–405
  - Y'CrCb component digital signal, 411–413, 452
  - Y'PrPb component analog signal, 411
- Video system, information
  - technology-based, 394–397
- Video time code, 433–434
- Virtual arrays, 188
- Virtual circuits, 263

Virtual computing, 227  
 Virtual data center, 227  
 Virtual LANs, 249–251  
 Virtual private network, 37, 332–334  
 Virtual router redundancy protocol, 223  
 Virtual servers, 188–189, 189f  
 Virtual software patch, 329–330  
 Virtualization  
   advantages of, 187–188  
   clustered file systems vs., 94–95  
   data center effects, 190  
   definition of, 186  
   methods of, 186–190  
   principles of, 87–89  
   storage area networks with, 142  
 Viruses  
   antivirus software for, 326, 331–332  
   description of, 321  
   window of vulnerability for, 325–326  
 V-ISAN, 295  
 VITC. *See* Vertical interval time code  
 Voice over IP (VoIP), 49, 64  
 Volume management, 92–93  
 Volume organization, 92f  
 VPN. *See* Virtual private network  
 VRRP. *See* Virtual router redundancy protocol

## W

WAAS. *See* Wide area application services  
 WAN  
   connectivity  
     network, 258  
     overview of, 256

    point-to-point form of, 256  
     topologies, 256–258  
   definition of, 256  
   networked attached storage  
     acceleration over, 146–147  
   private, 258  
   public, 258  
   router interfaces with, 37  
   transport type classifications, 257f  
 WAN accelerator, 60  
 Watson, Thomas J., 177  
 WBEM. *See* Web-based enterprise management  
 W3C Web services model, 172–175, 174f  
 Wear leveling, 466  
 Web services  
   for A/V environments, 174–175  
   characteristics of, 171  
   definition of, 312  
   description of, 19, 173–174  
   example of, 173f  
 Web Services Description Language, 174  
 Web services model  
   description of, 169–172  
   W3C, 172–175, 174f  
 Web-based enterprise management, 355, 365–367  
 Wide area application services, 147  
 Wide area file services (WAFS), 60, 146–147, 147f  
 Wide area network. *See* WAN  
 Wide geography metric, 26, 27t  
 Windows  
   description of, 17  
   marketshare by, 17  
 Windows management  
   instrumentation, 367–368

“Wire speed” packet forwarding, 37  
 Workflow(s)  
   definition of, 303, 314  
   description of, 23, 25  
   media. *See* Media workflows  
   preexisting, 381  
 WORM, 118  
 Worm(s)  
   description of, 321  
   firewall protection against, 325  
   NX for, 331  
   virtual software patch for, 329–330  
 Wrappers  
   description of, 74–75  
   formats, 278–280  
   MXF, 280–285  
   SDTI-CP, 418  
 WSDL. *See* Web Services Description Language

## X

XAM, 110  
 XML  
   description of, 173, 291–293  
   metadata standards and schemas, 293–294  
 XQuery, 293  
 Xsan CFS, 95

## Y

Y’CrCb component digital signal, 411–413, 452  
 Y’PrPb signal, 411

## Z

Zero-day attack time, 325