DIGITAL VIDEO DISTRIBUTION IN BROADBAND, TELEVISION, MOBILE AND CONVERGED NETWORKS

TRENDS, CHALLENGES AND SOLUTIONS

Sanjoy Paul, Ph.D

Formerly of Bell Labs and WINLAB, Rutgers University, USA, now of Infosys Technologies Limited, India

A John Wiley and Sons, Ltd., Publication

DIGITAL VIDEO DISTRIBUTION IN BROADBAND, TELEVISION, MOBILE AND CONVERGED NETWORKS

DIGITAL VIDEO DISTRIBUTION IN BROADBAND, TELEVISION, MOBILE AND CONVERGED NETWORKS

TRENDS, CHALLENGES AND SOLUTIONS

Sanjoy Paul, Ph.D

Formerly of Bell Labs and WINLAB, Rutgers University, USA, now of Infosys Technologies Limited, India

A John Wiley and Sons, Ltd., Publication

This edition first published 2011

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Paul, Sanjoy.
Digital video distribution in broadband, television, mobile, and converged networks : trends, challenges, and solutions / Sanjoy Paul.
p. cm.
Includes index.
ISBN 978-0-470-74628-8 (cloth)
1. Multimedia communications. 2. Digital video. 3. Multicasting (Computer networks) I. Title.
TK5105.15.P38 2010
006.7–dc22

2010019472

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

Print ISBN: 9780470746288 (H/B)
ePDF ISBN: 9780470972922
oBook ISBN: 9780470972915

Typeset in 10/12pt Times by Aptara Inc., New Delhi, India

Contents

About the Author

Sanjoy Paul is Associate Vice President, General Manager-Research and Head of the Convergence Lab at Infosys Technologies Limited, where he heads research and innovation in the field of communications, media and entertainment. Previously, he was a research professor at WINLAB, Rutgers University and founder of RelevantAd Technologies Inc, before which Sanjoy spent five years as the Director of Wireless Networking Research at Bell Labs, Lucent Technologies, and as the CTO of two start-up companies (Edgix and WhenU) based in New York. In a previous tenure at Bell Laboratories as a distinguished member of technical staff, Sanjoy was the chief architect of Lucent's IPWorX caching and content distribution product line.

Sanjoy has over 20 years of technology expertise, specifically in the areas of end-to-end protocol design and analysis, mobile wireless networking, quality of service, multicasting, content distribution, media streaming, intelligent caching and secure commerce. He served as an editor of *IEEE/ACM Transactions on Networking*, a guest editor of *IEEE Network Special Issue on Multicasting*, Steering Committee member of IEEE COMSNETS, General Chair and Technical Program Committee Chair of IEEE/ICST COMSWARE 2007 and 2006 respectively, and as a technical program committee member of several IEEE and ACM international conferences.

Sanjoy has authored a book on multicasting, published over 100 papers in international journals, refereed conference proceedings, authored over 80 US patents (28 granted, 50+ pending), and is the co-recipient of 1997 William R. Bennett award from IEEE Communications Society for the best original paper in IEEE/ACM transactions on networking.

He holds a Bachelor of Technology degree from IIT Kharagpur, India, an M.S and a Ph.D. degree from the University of Maryland, College Park, and an MBA from the Wharton Business School, University of Pennsylvania. He is a Fellow of IEEE and a Member of the ACM.

Preface

This book is the result of my experience working on multiple video- and content-related projects in many companies and universities over the last decade. It started to take shape about five years ago when, together with Katie Guo, a colleague at Bell Labs, I tried to put our experiences in the form of a tutorial for IEEE/ACM conferences. Interestingly enough, a major transformation was happening in the industry at the same time. Applications, such as YouTube, started to make video mainstream on the Internet, and devices like iPhone, was making video ubiquitous on handheld devices.

The first decade of the twenty-first century saw the transformation of the voice industry from a specialized industry with its own ecosystem to becoming an application on the Internet but the second decade is going to see yet another transformation. This time the transformation is going to affect the mainstream media and television industry as the cost of bandwidth plummets and storage per dollar increases exponentially, making the Internet infrastructure capable of transporting video traffic in a cost-effective manner without compromising the quality of experience. However, as video traffic increases over the Internet, the pipes, regardless of how fat they are, will start to get clogged, particularly with the onset of high definition (HD) video. Furthermore, the expectations of end users will increase in terms of wanting to access video from anywhere, using any device, thereby stressing the capacity of mobile wireless networks and the capability of hand-held devices. As a result, the industry has to devise innovative architectures, clever algorithms, better encoding and compression techniques for distributing video across converged (broadband, television, mobile) networks. This book attempts to expose the problems and challenges in multiple dimensions, whether it is for open networks like the Internet or closed networks like the managed networks of the communication service providers. Moreover, some existing solutions and some around-the-corner solutions that are being worked on to address the above challenges are also discussed in order to encourage the reader to think about how to approach the challenges at hand. Technical and business trends that are going to impact the distribution of video in converged network of the future are also described, to give a holistic view of the topic.

There are several people who deserve credit for shaping my thoughts and ideas in this book and let me thank them not in any special order. Katie Guo, my colleague at Bell Labs, worked with me on several video-related topics including video-on-demand, video streaming cache, video multicast over cellular networks and video over Internet for a long time. Many of the ideas presented in this book owe their origin to her. Professor Injong Rhee from North Carolina State University worked with us (Katie and myself) on video-on-demand and helped us better understand the challenges and come up with innovative solutions. Professor Lixin

Gao's work with Professor Jim Kurose and Professor Don Towsley significantly increased my understanding of the challenges in providing video-on-demand services. Sarit Mukherjee, my colleague at Bell Labs and also at Edgix, was influential in shaping my thoughts and ideas in the area of video streaming cache and content distribution networks. I gained a lot of technical knowledge from Neeraj Prasad, Debopam Bandyopadhyay and Prabal Sengupta at Alumnus Software, with whom I worked on an exciting project on video streaming. Arindam Mukherjee, a good friend, was instrumental in making my interactions with the technical folks at Alumnus possible and I thank him for that. Bill Goers, Jose De Francisco Lopez and Sathya Sankaranarayanan from Alcatel-Lucent also contributed significantly to clarify my thoughts and understanding while designing a video streaming system for mobile wireless networks. It was because of Kumar Ramaswamy at Thomson Multimedia that I learned a lot about the economic and technical tradeoff between traditional content distribution networks (CDNs) and peer-to-peer (P2P) networks in the context of distributing standard definition (SD) and high definition (HD) video over the Internet. Dipankar Raychoudhury, one of my mentors from WINLAB, Rutgers University, helped me think of the complexities of wireless networks for video and content distribution and also encouraged me to design a novel architecture (cache and forward architecture) for distributing video in intermittently connected networks. In the context of the project I interacted with Lijun Dong and Professor Yanyang Zhang, who helped shape my thought process significantly. Sumathi Gopal did much of the hard work in exposing the limitations of wireless networks for transporting bits and devised techniques for overcoming them. In addition, I have leveraged results from the research conducted by Shweta Jain and Ayesha Saleem at WINLAB, Professor Brian Levine at UMASS, Amherst, Professor Srinivasan Keshav at the University of Waterloo, Rajesh Krishnan at BBN and Professor Reza Rajaie at University of Oregon. I owe a lot to my distinguished colleagues at Infosys: Manish Jain (Convergence Gateway), Jayraj Ugarkar (Virtual Private Multimedia Library), Rajarathnam Nallusamy (UCOMAP), Tobby Thomas (QoE monitoring), P.N Manikandan (transcoders) who have helped me gain insights into practical systems, especially, in the context of managed (closed) networks of communication service providers.

Finally, a book of this size cannot be written without the constant sacrifice of family members. My wife Sutapa suffered the most as I took time away from her to write this book. Without her constant encouragement and support this book would not have been possible. Our children Prakriti and Sohom were also denied some family time because of my writing during holidays. I thank them, too, for their sacrifice! My parents and parents-in-law have been the constant source of inspiration for me all my life, particularly during writing of this book. I also thank my friends and family members who have encouraged me through the entire eighteen months of writing.

Part One

Technology Trends

1

Convergence

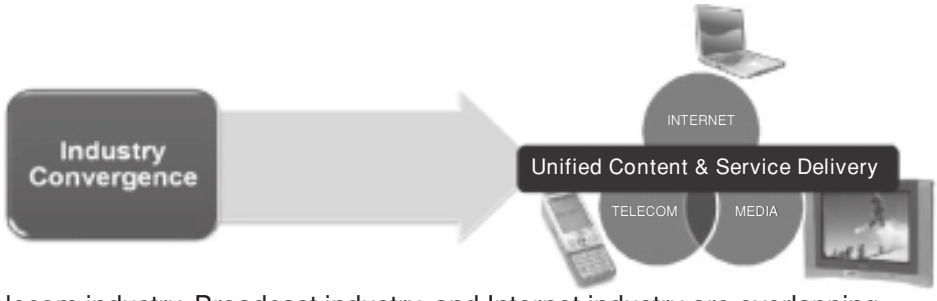
Convergence means different things to different people depending on the context. However, for the purpose of this book, we define four kinds of convergence: (i) industry convergence, (ii) device convergence, (iii) network convergence and (iv) service convergence.

1.1 Industry Convergence

The telephone (telecommunication) industry, the television (media/broadcast) industry and the Internet industry once existed separately with specialized infrastructure to deliver their respective services. For example, the telecom (voice) industry was built on specialized circuit-switched network technology, a multibillion dollar telecom switch and equipment industry, to deliver "telephony" or "voice" services to consumers [1, 3]. Consumers wrote a check to a telephone company, such as AT&T, for the monthly telephony service they received from it. The television (media/broadcast) industry used specialized broadcast network technology to deliver television (video broadcast) services to consumers. As with the telephony service, consumers made a monthly payment to a cable/satellite/television service provider, such as Comcast or DirectTV, for television services. Broadband access to the Internet was also offered as an independent service by broadband service providers, such as AOL, and consumers paid them for Internet (data) services.

However, with technological advances, these apparently independent industries are converging and are contending in the same digital content distribution space (Figure 1.1) [2]. There are two major reasons behind this transition. First, analog content is being replaced with digital content. As a result, content, no matter what industry it belongs to, is converted from analog to digital and then packaged into small units called packets. Second, the network infrastructure is converging into a common packet-switched Internet protocol (IP)-based network technology, which is capable of carrying packets in an efficient manner. Naturally, all content, voice, video and data, are being transported over the common network. Telephony has become an "application" on the Internet (voice over IP). Television is also becoming an "application" on the Internet (Internet television) and the Internet itself, which used to be non-friendly to real-time traffic, is also morphing to support real-time traffic, preserving quality of service. As a result, the challenges being faced by these industries are almost identical, except

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul



Telecom industry, Broadcast industry, and Internet industry are overlapping in the Digital Content Distribution space leading to new Business Models

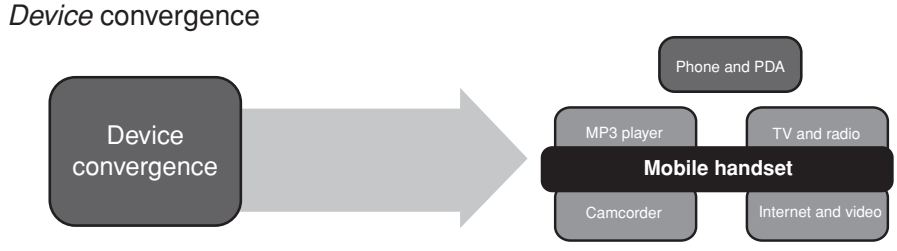
	Before	***Now and Future***
Telecom (Voice)	Specialized Circuit-switched Network for Voice	Voice is an Application on the Internet
Broadcast (Television)	Specialized Broadcast Network for Television	Television is an Application on the Internet
Internet (Data)	Packet-switched Network designed for non Real-time Applications: Email & Web	Network enhanced to support Real-time Applications: Voice & Video

Figure 1.1 Industry convergence.

for whatever business challenges they have specific to their domain. Moreover, each of these industries is expanding the boundaries of its business, thereby treading in so-called unfamiliar territory. This is leading to challenges but also to opportunities that we discuss in detail in later parts of the book.

1.2 Device Convergence

Consumer electronics and communications functionality are converging onto consumer devices. For example, laptops are being equipped with microphone, speakers, cameras and other consumer electronics to enable new capabilities like telephony and video conferencing (using Skype [5], Yahoo! Messenger [6], GTalk [7] etc.) across the Internet in addition to the traditional applications, such as Web surfing, instant messaging and e-mail. What used to be just a mobile phone a few years ago is today a camera, a video recorder, an MP3 player, an AM/FM radio, an electronic organizer, a gaming controller, a phone, a device for surfing the Web, a device for sending instant messages, and in some cases, a device for watching television (Figure 1.2). Consumers' ownership of such powerful handheld devices opens the door for communications service providers (CSPs) to deliver a variety of content embodied in text, images, audio and video to the end user. However, the fact that an end user can store the delivered multimedia content and share it with the rest of the world with a single push of a button may lead to unprecedented illegal sharing of content, making it content owners' worst nightmare. Thus the benefits of convergence come with challenges of security and privacy.



	Before	***Now and Future***
Laptop	Used for *Web* and *Email*	Used for Web and Email + *voice, video*
Mobile phone	Used for making *phone* calls	Used for phone + *Web, Email, IM, camera, video, MP3 player, AM/FM radio, PDA* etc.

Figure 1.2 Device convergence.

1.3 Network Convergence

Network infrastructures used by the telephone (telecommunication) industry, the television (media/broadcast) industry and the Internet industry have traditionally been very different. The telecommunications industry has been using circuit-switched network elements; the television industry has been using broadcast network equipment and the Internet industry has been using packet-switched network elements. Packet-switched networks have been built using different technologies as well. For example, asynchronous transfer mode (ATM), frame relay (FR) and IP are all technologies that have been used and are still being used in CSPs' networks. One way of reducing capital expenses (capex) and operational expenses (opex) would be to choose a common technology for the network infrastructure. This would assist CSPs in their need to contain expenses by training and employing technical people skilled in only the chosen type of technology. The fact of the matter is, the CSPs are converging onto using only IP/MPLS-based networks for transport and IP multimedia subsystem (IMS)-based infrastructure for session/service and blended (voice, video, data) applications (Figure 1.3) . This transition in the industry to converge on to a common network for applications and services is referred to as network convergence and this has far-reaching consequences for the industry.

1.4 Service Convergence

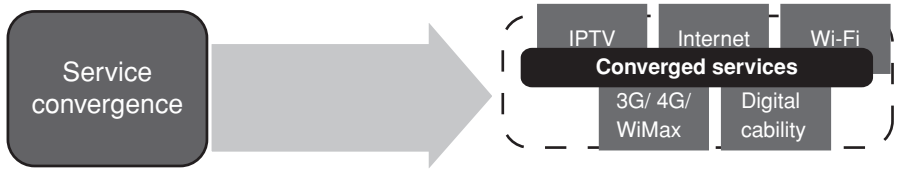
Services offered by the telecommunications industry, the television industry, the Internet industry and the wireless services industry have been independent of one another. However, with the introduction of new technology enabling unified communications across these networks, consumers expect to access the same services (voice, e-mail, messaging and so forth) and content (Web, video, audio) anytime from anywhere using any device (laptop, TV set, cellphone) with consistent quality of experience (Figure 1.4) [4]. An example of service convergence

Network convergence

	Before	***Now and future***
Networking technology	ATM network Frame relay network DSL/cable network IP network Proprietary session control	*IP/MPLS* network is the common end-to-end technology *IMS-based* common session layer enables blending of applications

Figure 1.3 Network convergence.

Service convergence



People want to *access their services and content* anytime anywhere using any device

	Before	***Now and future***
Communication	*Voice* only from residential phones *Email* only from *Internet/PC* *Messaging* only by *SMS/mobile* *Separate Address Book* for residential phone, email and cell-phones	*Voice, Email, messaging from anywhere using any device* (cell-phone, laptop, TV) *Common address book* for communication *Seamless service mobility* from one network to another
Content	*Web* only on *internet/PC* *Video* only on *TV* set *Audio* only on *radio, CD player*	Web, video, audio on any device (laptop, TV set, cell-phone) from anywhere *Seamless content mobility* from one device/network to another

Figure 1.4 Service convergence.

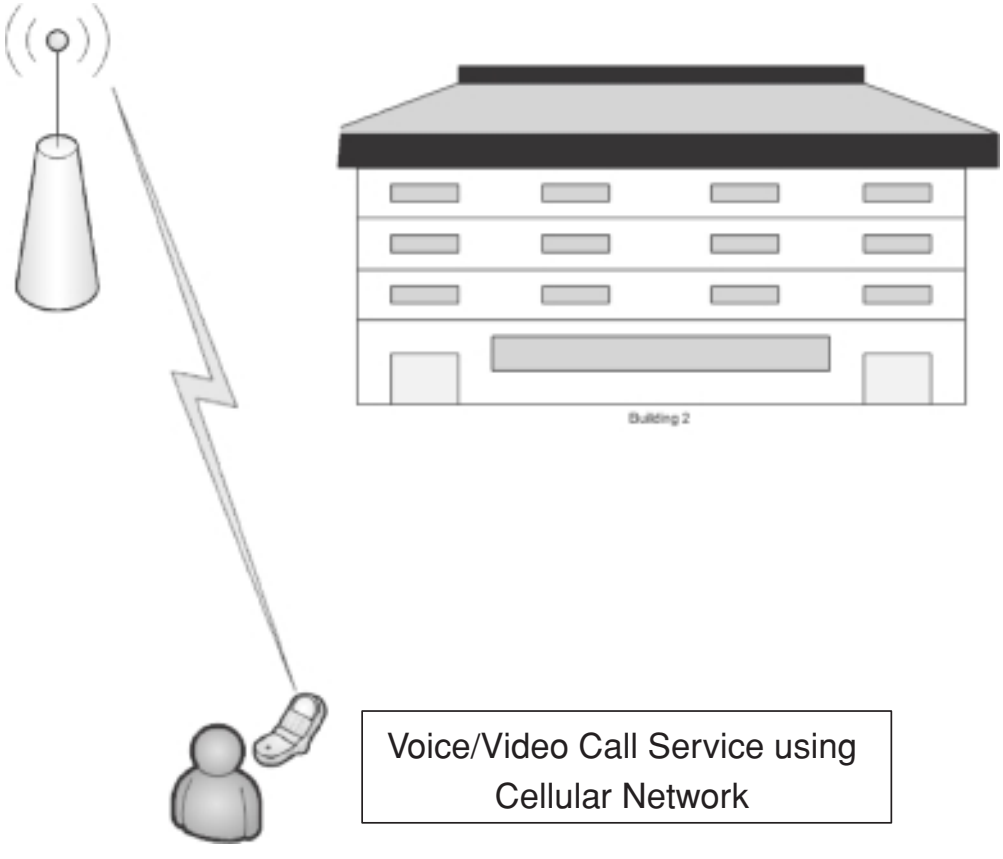
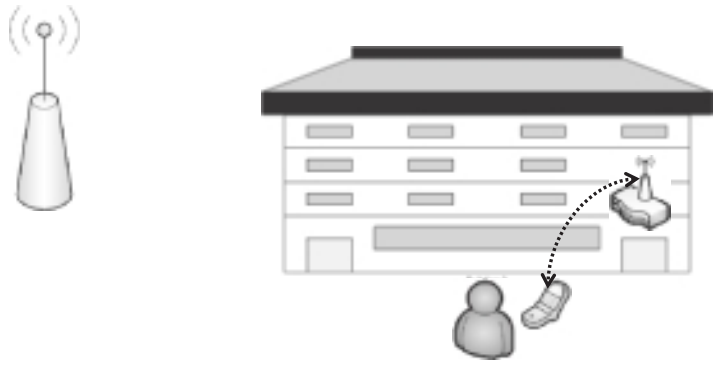


Figure 1.5 Voice/video call using cellular network.

would be for CSPs to offer a service that would enable their customers to take part in social networking using any device from anywhere. Moreover, customers not only expect to be able to use the services from anywhere using any device, but they also expect to move content/services seamlessly from network to network without compromising quality of experience.

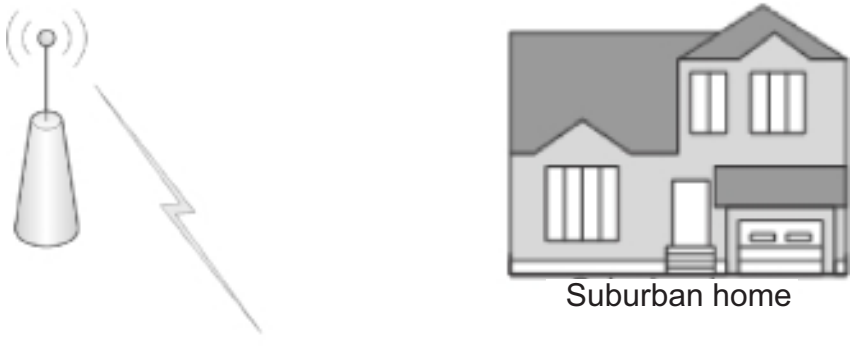
For example, as shown in Figures 1.5 and 1.6, a phone call uses the cellular network when that is the only network available for connectivity and uses the WiFi network when that is available in addition to the cellular network. In fact, the transition from cellular network to WiFi network happens seamlessly without interrupting the phone call.

Figures 1.7 and 1.8 show how video being watched on a small-screen cellphone in a train is seamlessly transitioned to a large-screen TV set when the user enters home. This is an example of seamless mobility of content. While service convergence opens up unprecedented opportunities of offering novel value-added blended services for the CSPs, it also makes the content providers worried that what used to be *protected* content in their network may not be protected any more due to lack of a comprehensive security solution spanning multiple networks.



Voice/Video Call Service continues uninterrupted while the underlying network changes from cellular outdoors to wifi indoors

Figure 1.6 Voice/video call seamlessly moves to WiFi network.



Watching a movie on a mobile handset while commuting on a train

Figure 1.7 Watching movie on mobile handset.

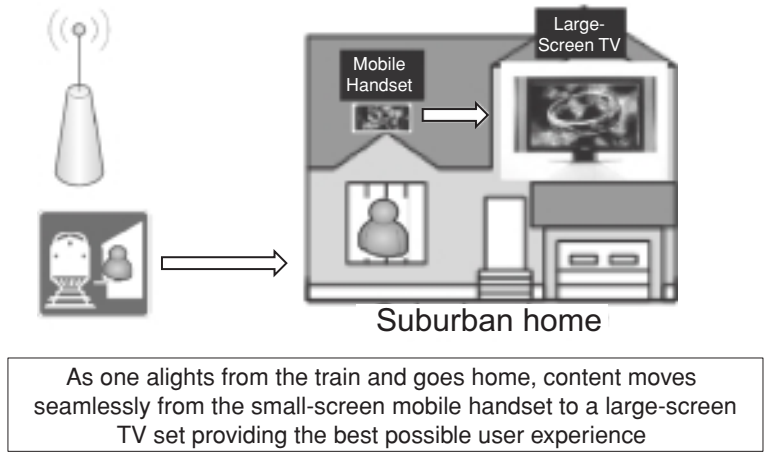


Figure 1.8 Movie transitions seamlessly from small screen of mobile handset to large screen of TV.

1.5 Summary

Digital convergence is already happening in the industry. With digitization of content, the distinction between voice, video, images and text is blurring as everything is being treated uniformly as data and transported over a common IP network as opposed to using specialized networks for transporting voice, video and data. Furthermore, everything is becoming an application on the IP network leading to overlapping of what used to be distinct industry segments, namely, telecom (voice), broadcast and media (video) and Internet (data). In order to provide access to these applications from anywhere and at any time, devices (PC/laptop, mobile handsets, TV sets) are becoming more and more powerful with multiple consumer electronic features being built into them, leading to what is known in the industry as device convergence. A case in point is a smart handset with features, such as, AM/FM radio, mobile TV, phone, browser, digital camera, video recorder, MP3 player, calendar, office applications and a host of other features. A variety of network technologies are converging into IP-based technology, leading to mixing and matching of applications and features in any service from the end-user perspective and lower capital expense (capex) and operational expense (opex) from the service provider perspective. *Service convergence* refers to the capability of end users to avail the same service regardless of the network over which it is accessed and the ability of end users to access the same content over multiple devices in a seamless manner. Digital convergence is leading to new applications and services that were not possible before and opening up new possibilities both from the service provider perspective as well as from the end-user standpoint.

References

[1] Hudson H.E. (1997) Converging technologies and changing realities: towards universal access to telecommunication in developing world, in *Telecom Reform: Principles, Politics and Regulation* (ed. W.H. Melody), Technical University of Denmark, Lyngby.

[2] Lamberton, D.M. (1995) Technology, information and institution, in *Beyond Competition: The Future of Telecommunications* (ed. D.M. Lamberton), Elsevier, Amsterdam.
[3] Mitchell, J. (1997) Convergent communication, fragmented regulation and consumer needs, in *Telecom Reform: Principles, Politics and Regulation* (ed. W.H. Melody), Technical University of Denmark, Lyngby.
[4] Service convergence: bringing it all together; Telecom Asia, April, 2005.
[5] Skype. http://www.skype.com/ (accessed June 9, 2010).
[6] Yahoo Messenger. http://in.messenger.yahoo.com/ (accessed June 9, 2010).
[7] Google Talk. http://www.google.com/talk./ (accessed June 9, 2010).

2

Video Compression, Encoding and Transport

Video is nothing but a sequence of still images. One way of compressing and encoding video is to compress each individual still image and encode it independently of the other images in the sequence. The Joint Photographic Experts Group (JPEG) format is one way of compressing still images. When individual still images in a sequence are independently compressed and encoded using JPEG, the video-encoding format is called Motion JPEG (or MJPEG). However, as will subsequently be discussed, there are better ways of compressing and encoding video, which result in far fewer bits than MJPEG for representing the same video in digital form.

In any case, techniques used for compressing still images form the foundation of video compression [1–3]. So we will start by understanding how still images are compressed.

2.1 Still Image Compression

2.1.1 Block Transform

Each image is usually divided into many blocks, each of size 8 pixel × 8 pixel. These 64 pixels are then transformed into frequency domain representation by using what is called discrete cosine transform (DCT). The frequency domain transformation clearly separates out the low-frequency components from high-frequency components. Conceptually, the low-frequency components capture visually important components whereas the high-frequency components capture visually less striking components. The goal is to represent the low-frequency (or, visually more important) coefficients with higher precision or with more bits and the high-frequency (or visually less important) coefficients with lower resolution or with fewer bits. Since the high-frequency coefficients are encoded with fewer bits, some information is lost during compression, and hence this is referred to as "lossy" compression. When inverse DCT (IDCT) is performed on the coefficients to reconstruct the image, it is not exactly the same as the original image but the difference between them is not perceptible to the human eye.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

2.1.2 Quantization

As mentioned in the earlier section, the DCT coefficients of each 8 × 8 pixel block are encoded with more bits for highly perceptible low-frequency components and fewer bits for less perceptible high-frequency components. This is achieved in two steps. First step is quantization, which eliminates perceptibly less significant information, and the second step is encoding which minimizes the number of bits needed for representing the quantized DCT coefficients.

Quantization is a technique by which real numbers are mapped into integers within a range where the integer represents a level or quantum. The mapping is done by rounding a real number up to the nearest high integer, so some information is lost in the process. At the end of quantization, each 8 × 8 block will be represented by a set of integers, many of which are zeroes because the high-frequency coefficients are usually small and they end up being mapped to 0.

2.1.3 Encoding

The goal of encoding is to represent the coefficients using as few bits as possible. This is accomplished in two steps. Run length coding (RLC) is used to do the first level of compression. Variable length coding (VLC) does the next level of compression.

In fact, after quantization, the majority of high-frequency DCT coefficients become zeroes. Run length coding takes advantage of this by coding the high-frequency DCT coefficients before coding the low-frequency DCT coefficients so that consecutive number of zeroes is maximized. This is accomplished by scanning the 8 × 8 matrix in a diagonal zigzag manner. Run length coding encodes consecutive identical coefficients by using two numbers. The first number is the "run" (the number that occurs consecutively) and the second number is "length" (the number of consecutive occurrences). Thus if there are N consecutive zeroes, instead of coding each zero separately, RLC will represent the string of N zeroes as [0, N].

After RLC, there will be a sequence of numbers and VLC encodes these numbers using minimum number of bits. The technique is to use the minimum number of bits for the most commonly occurring number and use more bits for less common numbers. Since a variable number of bits are used for coding, it is referred to as variable length coding.

2.1.4 Compressing Even Further

The techniques described so far focused on the optimal way of compressing an 8 × 8 pixel block. However, there is significant correlation between neighboring blocks in a frame. Thus instead of coding each block independently, a prediction component is introduced before quantization. Essentially, the coefficients of a given block are used to predict the coefficients of an adjacent block. Since the predicted coefficients are very close to the actual coefficients, instead of quantizing and coding the actual coefficients, the difference is quantized and coded. Naturally, that requires less number of bits compared to the case of coding the actual coefficients. This technique is referred to as "intraframe" coding.

2.1.5 Adding Color to an Image

There are two ways of encoding color. The first one represents the same image in three planes, where each plane represents a color (red, green and blue) and then encodes each plane exactly as described above. The second approach is also to use three planes but the first plane is for "luminance" (or brightness of the pixels) and the other two planes are for "chrominance" (or color). The human eye is more sensitive to brightness than to color. As a result, luminance needs to be encoded with higher resolution or more bits and chrominance can be encoded with lower resolution or fewer bits. In fact, video compression uses the second approach and for every "macro block" (16 $\times$16 pixel) in a frame, there are four 8 $\times$ 8 blocks of luminance, and two 8 $\times$ 8 blocks of chrominance. This is known as "subsampling" and contributes to further compression.

2.2 Video Compression

Video is nothing but a sequence of still image frames. The previous section described how still images are compressed and encoded. One possible way to encode video is to encode individual frames independently as described in the section above. With the most sophisticated techniques, the maximum compression possible without compromising quality is 30:1. However, there are more efficient ways of compressing video that can provide as much as 200:1 compression.

One major observation in case of video is temporal correlation – there is significant correlation between consecutive frames. In slow-moving videos, macro blocks of the next frame can be predicted from corresponding macro blocks of the current frame. Then, instead of coding the macro block of the next frame, the difference between the predicted value and the original value of macro block is coded. The difference is usually small, so fewer bits can be used to code it. This provides significant compression.

However, if the camera pans or a large object moves in the video, "motion estimation" becomes very important from the encoding point of view.

2.2.1 Motion Estimation and Compensation

Motion estimation involves finding a matching macro block in a previously coded frame referred to as "reference" frame with the macro block in the frame under consideration. Once the reference frame is found, the motion estimator computes what is called a "motion vector", which captures the horizontal and vertical offset between the two frames. Then the macro block of the target frame is predicted using the macro block of the reference frame and the difference is coded using the still-image coding techniques described above. Once again this requires fewer bits as the prediction error is small.

In video encoding, the order in which frames are coded is not the same as the order in which they are played. Thus, it is not necessarily true that the reference frame for a future frame is the one just before it in the playing sequence. In fact, the video encoder jumps ahead from the currently displayed frame and encodes a future frame and then jumps back to encode the next frame in display order. Sometimes, the video encoder uses two "reference" frames: the currently displayed frame and a future encoded frame to encode the next frame in the display sequence. In this case, the future encoded frame is referred to as P-frame ("predicted" frame) and the frame encoded using two reference frames is referred to as B-frame ("bi-directionally" predictive frame). However, there is a problem with this approach because an error in one of

the frames would propagate through the encoding process forever. To avoid this problem, the video encoders typically encode one video frame (periodically) using still-image techniques and such frames are referred to as "intra" frames or I-frames.

2.2.2 *Group of Pictures (GOP)*

A sequence of frames from one I-frame to another I-frame is referred to as a group of pictures (GOP). A GOP is best represented using a two-dimensional table (see Table 2.1). Note that the order of encoding and the order of display are not the same for the frames in a GOP.

Table 2.1 Display order versus encoding order in a group of pictures (GOP).

Encoding Order (top to bottom)	Display Order (Left to Right)													
	I													
					P									
		B												
			B											
				B										
									P					
						B								
							B							
								B						
													P	
										B				
											B			
												B		
														I

2.3 Video Transport

The previous section showed how video is compressed and digitally encoded. Encoded video can then be stored as files just as any other type of content is stored in the file system. When delivery is requested, the file is deconstructed into frames and the frames are broken down into packets and transported over the network. For live video feeds, video is captured, compressed and encoded on the fly into the same GOP structure, and one frame is transported at a time where each frame, depending on the type and size, may need to be broken into multiple packets and transported over the network.

2.4 Summary

In this chapter we first discussed how images are digitized, quantized, encoded and compressed. We also looked at the additional dimension of encoding color. In fact, there are two ways of encoding color. First one represents the same image in three planes where each plane represents a color (red, green and blue). The second approach also uses three planes but the first plane is for "luminance" (or brightness of pixels) and the other two planes are for "chrominance" (or color). Video encoding was discussed next as an extension of image encoding with motion estimation. Typically videos have three types of frames: I-frames, P-frames and B-frames, where I-frames capture most of the information in a scene followed by P-frames and B-frames, which capture the "deltas" from the original frame (captured in I-frame) resulting from motion. Finally, I, P and B frames can be packetized and transported over an IP network just like any other type of data.

References

[1] Richardson, Iain E.R. (2002) *Video Codec Design: Developing Image and Video Compression System*, John Wiley & Sons, Ltd. ISBN: 0471485535, 9780471485537.

[2] Bhaskaran, V. (1996) and Konstantinides, K. *Image and Video Compression Standards: Algorithms and Architectures*, 2nd edn, Kluwer Academic Publishers. ISBN: 0-7923-9952-8.

[3] Shi, Yun Q. and Sun, Huifang. (1999) *Image and Video Compression for Multimedia engineering: Fundamentals, Algorithms and Standards*, CRC Press. ISBN: 0-8493-3491-8.

3

Internet Protocol Television (IPTV) versus Internet Television

Internet Protocol television (IPTV) is fundamentally different from Internet television [1, 3]. Whereas IPTV (also called telco TV) is closed and controlled by an operator or telecom service provider, Internet television is open, exactly like the Web [2]. Consumers interact with the operator in IPTV whereas with Internet television they interact with the content publisher independent of any specific operator. Internet Protocol television is fundamentally geographically bound, meaning that deployment infrastructure is based in regions. By contrast, Internet television uses a *global reach business model* meaning that video and television services offered in one region can be accessed from any other location. Programs on IPTV are similar to those offered by digital Cable/Satellite service providers, whereas programs on Internet television can be anything, including user-generated content. On-demand services in IPTV are similar to the on-demand services in cable/satellite, so one can choose from a preselected list of items. By contrast, on-demand service in Internet television includes videos of independent rights holders who can be individuals creating videos for a small audience or traditional publishers publishing for large audience. The above description is summarized in Figure 3.1.

3.1 Internet Television and Video over IP

In 1995, voice conversation was first transported over a packet-switched network using Internet Protocol (IP), rather than over traditional circuit-switched telephone networks, by an Israeli company called Vocaltec [4]. Transporting telephony over IP dealt a serious blow to telecom service providers. Moreover, peer-to-peer (P2P) technology, which allowed file swapping over the Internet, led to the electronic distribution of music over the Internet with serious business consequences for the music industry. The question is whether similar developments will take place in the context of video.

Video is more difficult to transport over the Internet because it requires significantly higher bandwidth compared to voice. Even if the required bandwidth is available end-to-end for some time, sustaining that over a longer period of time is extremely difficult, if not impossible. In the

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

IPTV (or Telco IPTV)	Internet television
Operator or carrier controlled Physical pipes and infrastructure controlled by IP-TV operator Internet cannot normally be accessed	OPEN just like the Web! Anyone can create an endpoint and publish that on a global basis. Internet is the medium!
Consumers interact with IP-TV operator	Consumers interact with content Publisher using multiple devices Independent of any specific operator.
Fundamentally geographically bound Deployment infrastructure is based in regions	Uses a global reach business model Video and television services offered in one geography can be accessed from any other global geography
Programs similar to digital cable/ satellite providers.	Programs limited only by imagination!
Includes on-demand/pay per view similar to cable/satellite	Includes on-demand/pay per view for any rights holder who can be an individual creating a video for a small audience or a traditional publisher publishing for large audience
IPTV is fundamentally different from Internet TV: Think Web for Internet TV	

Figure 3.1 IPTV versus Internet television.

context of video file distribution, the size of video files is huge (GB) compared with audio files, which are about three orders of magnitude smaller. Bandwidth costs have not been economical for transporting DVDs (approximately $5/video for transport only). An end user would rather go to a video store to pick up a video of choice than download it over the Internet. However, most recently, with the advancement and maturity of technology, bandwidth costs have fallen significantly (approximately $0.30/GB of transport), storage costs have nosedived ($0.35–$1/GB), and compression technology has developed to compress a DVD quality video into a much smaller file. The bottom line is that the economics of delivering video is changing so that its distribution is becoming a viable business.

3.1.1 Content

There are three trends in video content. First, several companies provide Internet TV focusing mainly on niche content. The Internet TV companies offer TV channels specialized for niche content (such as cycling or rock climbing) and for ethnic TV content brought from the country of origin of immigrants for consumption in their country of residence. These videos are broadcast in real time over the Internet. Second, many companies, including some of the Internet TV companies, offer movies/videos on demand. These videos are delivered on demand and are mostly unicast. Finally, there are companies that allow users to upload their videos and share them with others.

3.1.2 Distribution

Video distribution is a challenge mainly because it requires significantly higher bandwidth (a few orders of magnitude more) than any other media on the Internet. If a video needs to be

stored in the network instead of being streamed it would require significantly higher storage than any other type of media on the Internet. There are usually two different approaches to video distribution: (i) infrastructure based and (ii) noninfrastructure based. Example of infrastructure-based video delivery networks are content distribution networks (CDN) such as, Akamai [5] and Limelight Networks [6]. Noninfrastructure-based video delivery networks are based on P2P networking technology [7].

3.1.3 Search

Search is as big in the context of video as it is in the context of data. However, there are two different approaches to video search. The first one is based on meta information in which videos are described using a set of keywords and users search them using keywords [8, 9]. This approach is similar to data search. The other approach is based on video scene analysis [10, 11, 12]. In this approach, users could search for similar videos that are identified based on the analysis of video content.

3.2 Summary

The main goal of this chapter was to clarify the distinction between the terms Internet television and IPTV, which many people use synonymously. Internet television is the equivalent of the Web for videos giving the ability to anyone with an Internet connection to be able to publish any video content and consume any video content published by someone else. It has no geographic boundaries and from the business perspective, the end user is a customer of the publisher of the video content. Internet Protocol television, on the other hand, is the equivalent of the CableTV/SatelliteTV, except that it is handled by a telecom service provider. Thus the content available over IPTV is controlled by the telecom service provider and is out of reach for individuals from video content publication point of view. Moreover, just like CableTV, IPTV has geographic boundaries over which the content is accessible and from the business standpoint, the end-user is a customer of the telecom service provider. Finally, distribution of video over the Internet, and searching for video content were highlighted as the major challenges for Internet television.

References

[1] IPTV vs. Internet Television: Key Differences; http://www.masternewmedia.org/2005/06/04/iptv_vs_internet_television_key.htm (June 9, 2010).
[2] Internet Television Is An Open Platform: Jeremy Allaire; http://www.masternewmedia.org/news/2005/05/17/internet_television_is_an_open.htm (June 9, 2010).
[3] IPTV vs. Me-Too TV; http://www.lightreading.com/document.asp?doc_id=74576&site=lightreading (June 9, 2010).
[4] History of Voice over IP Phones and VoIP Technology. http://www.voipusa.com/history-voip.php. (June 9, 2010).
[5] Akamai Technologies. http://www.akami.com/.
[6] Limelight Networks. http://www.limelightnetworks.com/. (June 9, 2010).
[7] Peer to Peer Overlay Networks. http://en.wikipedia.org/wiki/Peer-to-peer. (June 9, 2010).
[8] Video Search Engine: Blinkx. http://www.blinkx.com/. (June 9, 2010).
[9] Truveo Video Search. http://in.truveo.com/. (June 9, 2010).

[10] Israël, M., van den Broek, E.L. and van der Putten, P. (2004) *Automating the Construction of Scene Classifiers for ContentBased Video Retrieval*. Proccedings of MDM/KDD'04, August 22, 2004, Seattle, WA, USA.

[11] Flickner, M., Sawhney, H., Niblack, W., *et al.* (1995) Query by image and video content: The QBIC system. *IEEE Computer*, 28 (9), 23–32.

[12] Van Der Putten, P. (1999) *Vicar Video Navigator: Content Based Video Search Engines Become a Reality*, IBC edition, Broadcast Hardware International.

4

Multicast

Multicast refers to a technology in which the sender (source) transmits content once regardless of the number of receivers (destinations). Multicasting content over a broadcast network (such as a satellite network or the air interface of a wireless network) is relatively straightforward as the underlying physical network supports the concept of sending content once irrespective of the number of recipients. However, multicasting over point-to-point networks (such as wired Internet or IPTV) is more complex as it requires additional protocol-level support to construct a multicast distribution tree spanning the sender (source) and the receivers (destinations). Figure 4.1 shows how the sender transmits the same content eight times for eight receivers in the absence of multicast technology, resulting in a significant load over the network infrastructure.

Figure 4.2, in contrast, shows that the sender sends the content once for eight receivers and the network replicates content only as many times as is necessary to reach the final destinations.

4.1 Multicast in IPTV Networks

Digital television content is usually distributed using a broadcast network such as a satellite network, as in direct-to-home (or DTH) systems. The competing technology uses satellite networks to distribute television content over a large coverage area and then uses coaxial cables to distribute it in the local areas. Internet Protocol television poses the unique challenge of distributing digital television content over point-to-point networks rather than over broadcast networks. Digital television requires significantly high bandwidth (2–20 mbps per channel) for distribution. Regular TV channels, such as ABC, NBC, CBS and Fox and/or specialized cable channels, such as ESPN, HBO and CNN, have several hundred thousand viewers tuning in simultaneously. Typically, the video content of these channels is sourced into a central headend in a telco network (as shown in Figure 4.3) and that eventually becomes the point of distribution for IPTV. Just as there is a central headend, there are also regional headends that source video content for so-called regional channels and distribute it in a region spanning multiple cities [1]. Regardless of whether it is a central headend or a regional headend, high-bandwidth video content needs to be distributed to hundreds of thousands of viewers simultaneously.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

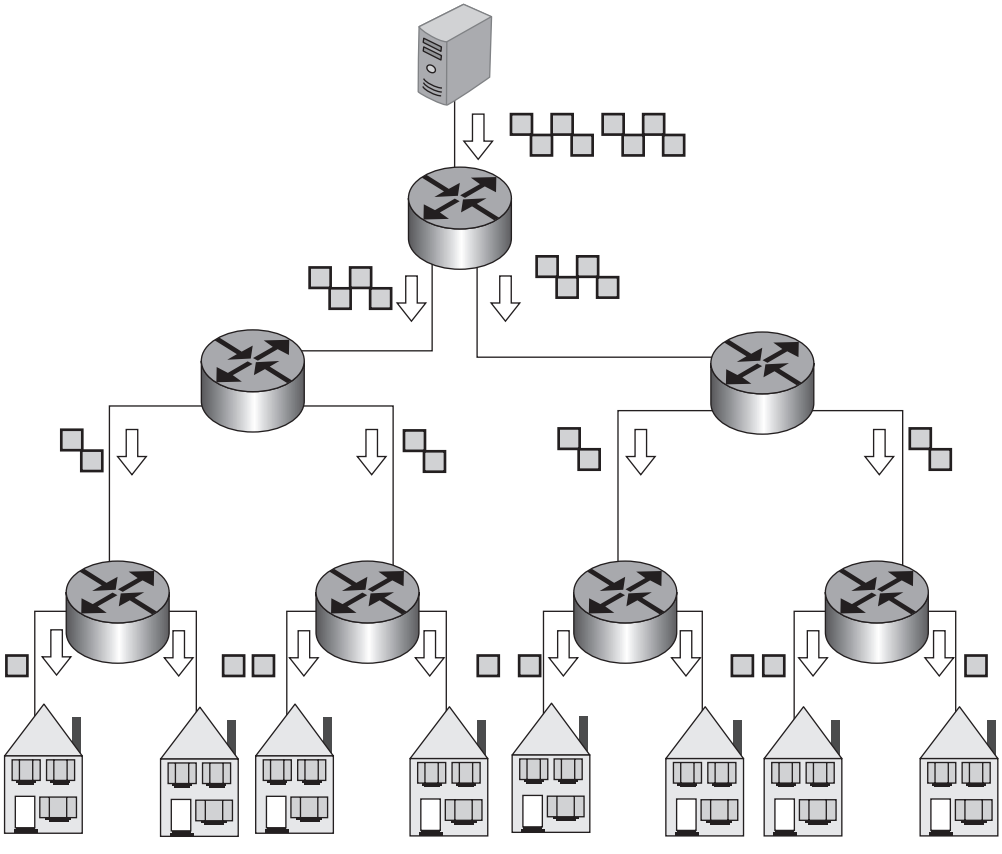


Figure 4.1 Multiple unicasts from sender to receivers.

This would require a bandwidth of hundreds of gigabits per second if the content were sent to each viewer separately using a point-to-point distribution mechanism. This is certainly not a scalable solution. A much more efficient way of distributing video content is by using multicast where the content of each channel is sent by the headend only once regardless of the number of viewers [2]. Internet Protocol multicast [3,4,10] is a technology that was developed to meet exactly this need. Figure 4.3 shows both network-wide multicast and region-specific multicast for distributing TV content nationwide and region wide respectively. The IP multicast tree is rooted at the headend and it spans the set top boxes (STBs) that subscribe to the corresponding IP multicast group.

4.2 Multicast in Mobile Networks

Live television on a mobile handset has been in commercial use in Korea since 2005, and in the US and Europe since 2006. Korea has embraced digital multimedia broadcast (DMB) technology [11] whereas the US has adopted Forward Link Only (FLO) [8, 9] technology

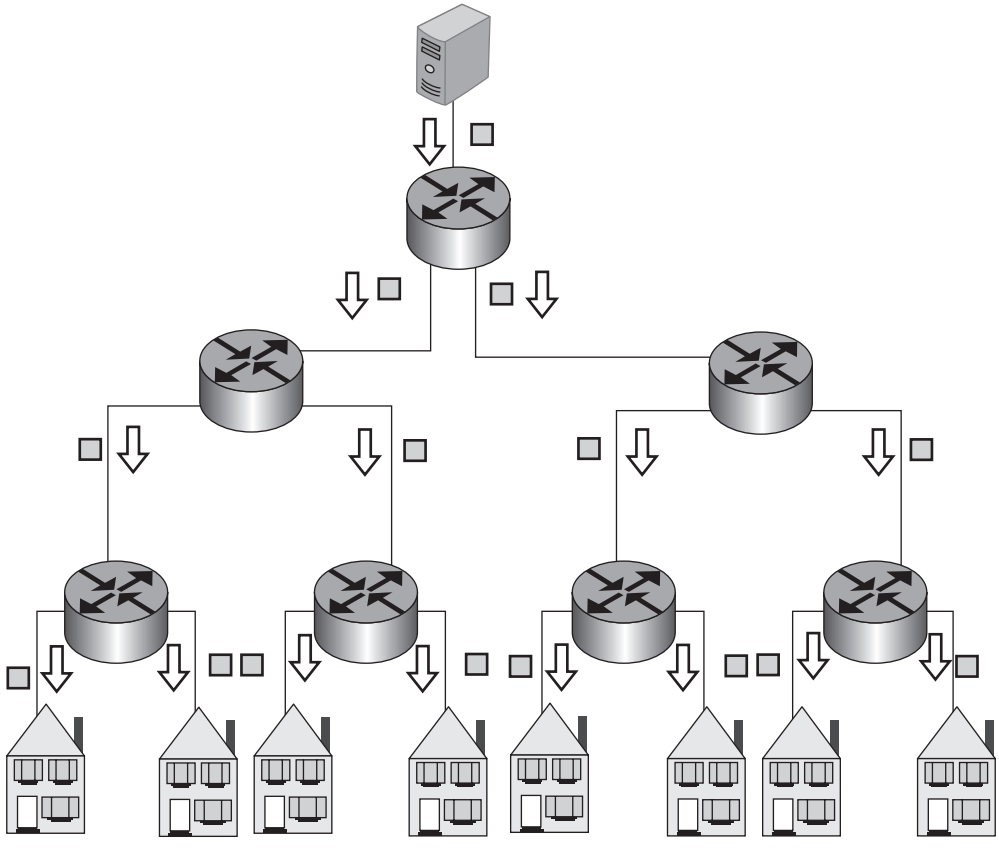


Figure 4.2 Multicast from sender to receivers.

and some parts of Europe have adopted Digital Video Broadcast for Handheld (DVB-H) [5,6] technology. All of these technologies use one-way broadcast networks. The other alternative is to use two-way cellular networks, such as high-speed packet access (HSPA) networks [12,13] with multimedia broadcast multicast service (MBMS) [14,15]. Regardless of how mobile TV content is distributed [7] (via dedicated one-way broadcast network or via two-way cellular networks), the key technology that makes mobile TV economically feasible is multicast. While multicast comes for free in dedicated one-way broadcast networks, the protocol support for multicast needs to be implemented in the network elements of cellular networks. Without multicast, live television broadcast would not be possible. However a combination of unicast and multicast is most likely to be used in a mobile network, where multicast will be used for distributing popular programs and live TV channels, whereas unicast will be used for distributing additional not-so-popular content and on-demand content (see Figure 4.4.)

Figure 4.5 shows the network system-level view of hybrid multicast where the mobile handset receives high-bandwidth multicast video content from FLO/DVB-H/DMB networks and low bandwidth multicast video content from the cellular network [13–15].

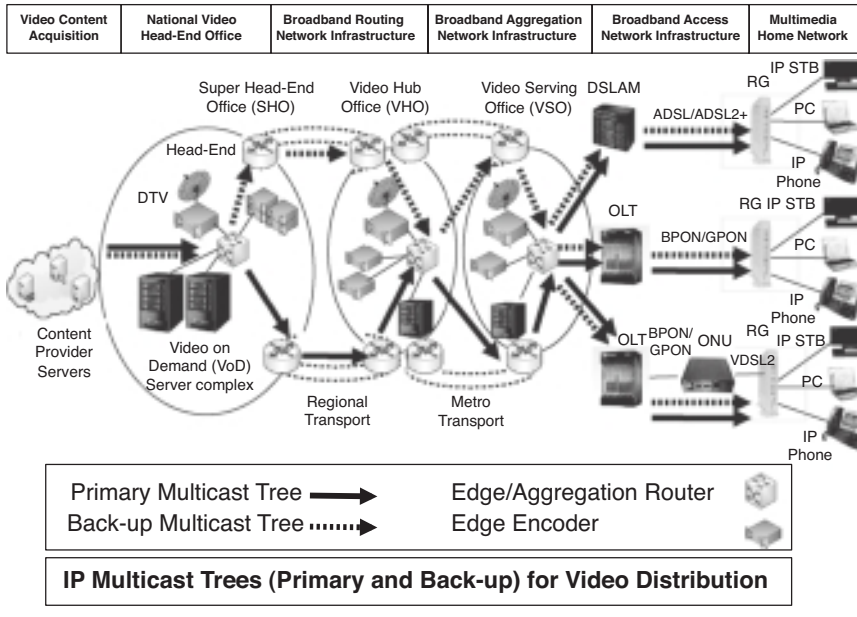


Figure 4.3 IPTV distribution network.

4.3 Summary

This chapter highlighted the importance of multicast in live video distribution, especially for broadcast television-type services over the terrestrial network (such as an IPTV network) as well as over the mobile network. Multicast technology enables the video server (source of video content) to transmit the content once regardless of the number of the receivers. Multicast technology for a point-to-point network such as IPTV was described, followed by the use of

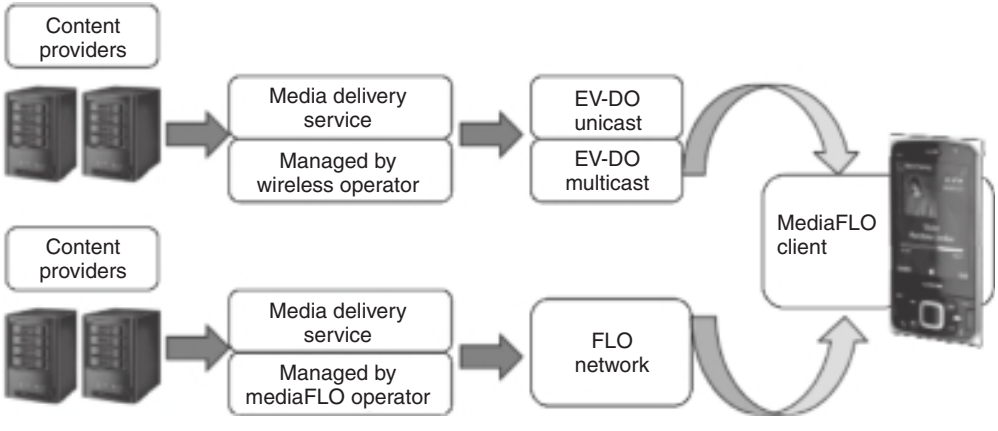


Figure 4.4 Hybrid multicast in mobile networks.

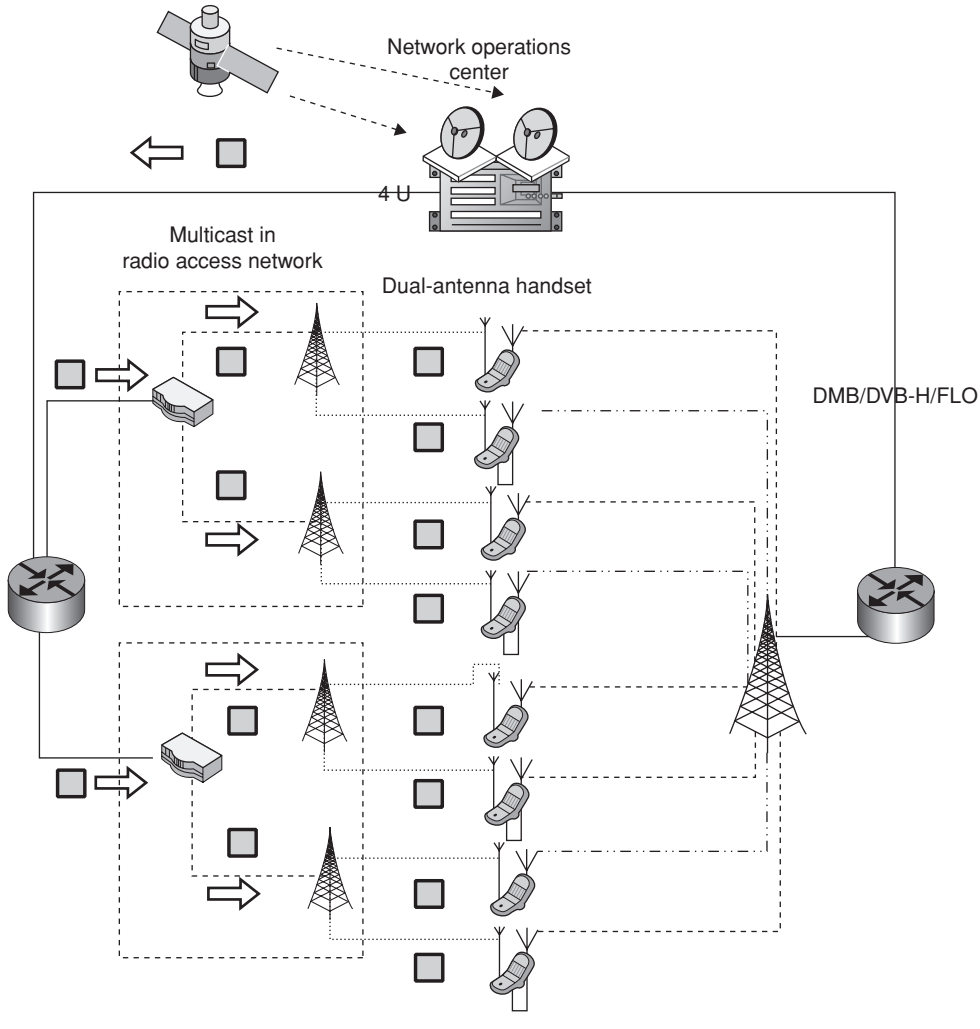


Figure 4.5 Network system-level view of hybrid multicast in mobile networks.

multicast in mobile wireless networks. Specifically, the concept of hybrid multicast (using a broadcast network such as FLO together with a point-to-point network such as the core and radio access network in a 3G/4G wireless network) was introduced in the context of delivering live and popular video content over a mobile network at a price point acceptable to the market.

References

[1] High-quality and resilient IPTV multicast architecture, http://www.juniper.net/solutions/literature/white_papers/iptv_multicast.pdf. (accessed June 9, 2010).

[2] Zhang, C., Liu, D., Zhang, L. and Wu, G. (2009) *Controllable Multicast for IPTV Over EPON*, Higher Education Press, co-published with Springer-Verlag GmbH; Vol. **2**, No. 2, pp. 222–228.

[3] IP Multicast Explained, http://www.klicktv.co.uk/tv-distribution-solutions/iptv/multicasting.html. (accessed June 9, 2010).
[4] Minoli, D. (2008) *IP Multicast with Applications to IPTV and Mobile DVB-H*. John Wiley & Sons, Ltd. ISBN 0470258152, 9780470258156.
[5] Kornfeld, M. and Reimers, U. DVB-H – the emerging standard for mobile data communication. http://www.ebu.ch/en/technical/trev/trev_301-dvb-h.pdf. (accessed June 9, 2010).
[6] Television on a handheld receiver – broadcasting with DVB-H. http://www.digitag.org/DTTResources/DVBHandbook.pdf. (accessed June 9, 2010).
[7] Fitchard, K. TV wars go wireless. http://connectedplanetonline.com/wireless/technology/telecom_tv_wars_go/. (accessed June 9, 2010).
[8] Flotm Technology Overview. http://www.qualcomm.com/common/documents/brochures/tech_overview.pdf. (accessed June 9, 2010).
[9] Data Delivery using FLO™ Technology. http://www.qualcomm.com/blog/2010/02/03/data-delivery-using-flo-technology. (accessed June 9, 2010).
[10] Paul, S. (1998) *Multicasting over the Internet and its Applications*, Kluwer Academic Publishers. ISBN: 0792382005.
[11] Digital Multimedia Broadcasting. http://en.wikipedia.org/wiki/Digital_Multimedia_Broadcasting. (accessed June 9, 2010).
[12] 3GPP – HSPA. http://www.3gpp.org/HSPA. (accessed June 9, 2010).
[13] HSPA evolution brings mobile broadband to consumer mass markets. Nokia Siemens Network whitepaper. http://www.nokiasiemensnetworks.com/sites/default/files/document/HSPA_for_the_massmarket_WP.pdf. (accessed June 9, 2010).
[14] 3GPP TS 26.346. Multimedia Broadcast/Multicast Service (MBMS): Protocols and codecs. http://www.3gpp.org/ftp/Specs/html-info/26346.htm. (accessed June 9, 2010).
[15] MBMS: Spreading the content. http://voicendata.ciol.com/content/GOLDBOOK2008/108030530.asp. (accessed June 9, 2010).

5

Technology Trend and its Impact on Video on Demand Service over Internet

Moore's Law [7, 8] describes a long-term trend in computing hardware in which the number of transistors that can be incorporated inexpensively in an integrated circuit is expected to double approximately every two years. The capabilities of many digital electronic devices – processing speed, memory capacity, number and size of pixels in digital cameras – are strongly linked to Moore's law. Moore's law continues to hold given that the speed of processors has been doubling every 18 months.

Storage capacity per dollar has also been increasing at the same rate [5, 6]. The same trend continues with bandwidth of broadband networks at home, wireless bandwidth indoors and mobile bandwidth outdoors. Specifically, 2.5 GHz/MP processors, 2GB/$ storage, 20 mbps broadband bandwidth to home, 100 mbps indoor wireless bandwidth and 2 mbps outdoor wireless bandwidth are available today (Figure 5.1). Video compression technology has also been improving tremendously, thereby providing equivalent quality of experience for a video with half the encoded file size that it required a few years ago. A combination of higher bandwidth (5×) and better compression (2×) is providing a 10× improvement in cost-performance, resulting in an economically viable business model for delivering video over the Internet. It is not just the favorable economics of transporting bits that is making video delivery over Internet a reality – the fact that the time needed to download a video over the Internet is almost comparable to the time needed to procure a video from a store is also contributing to the same cause.

A medium-size movie (2GB) takes approximately nine hours to download with a broadband connection of 500 kbps whereas the same movie with better compression (1GB in size) takes less than half an hour to download with a broadband connection of 5 mbps (Figure 5.2). Users are certainly willing to wait for half an hour for a movie to download as it would take them approximately the same time to physically go to a video rental store, rent the video and come back home.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

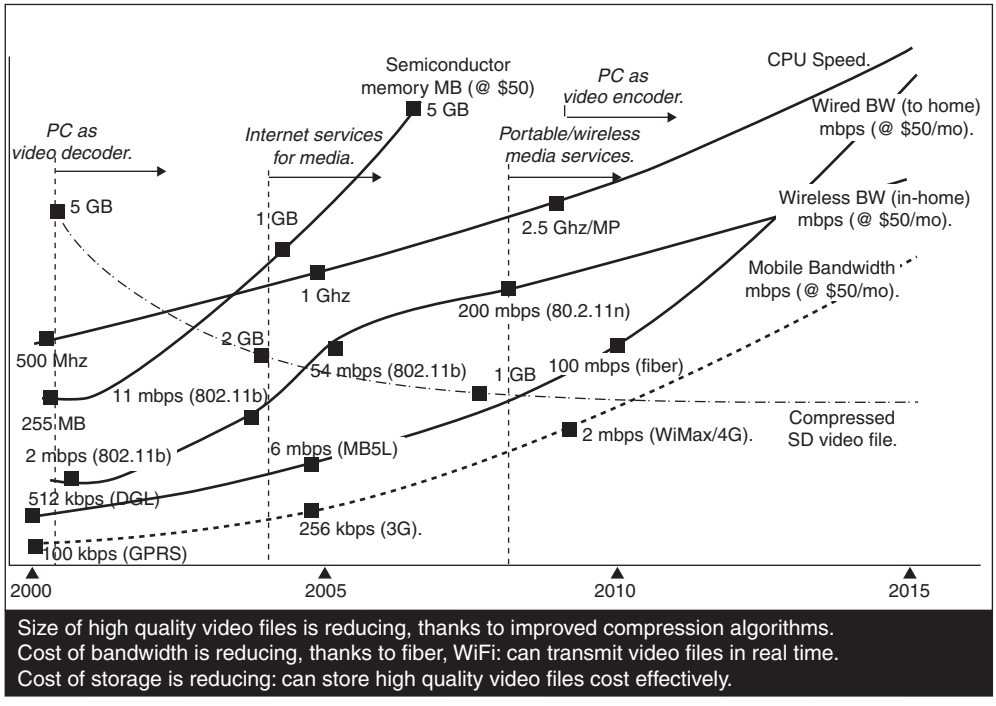
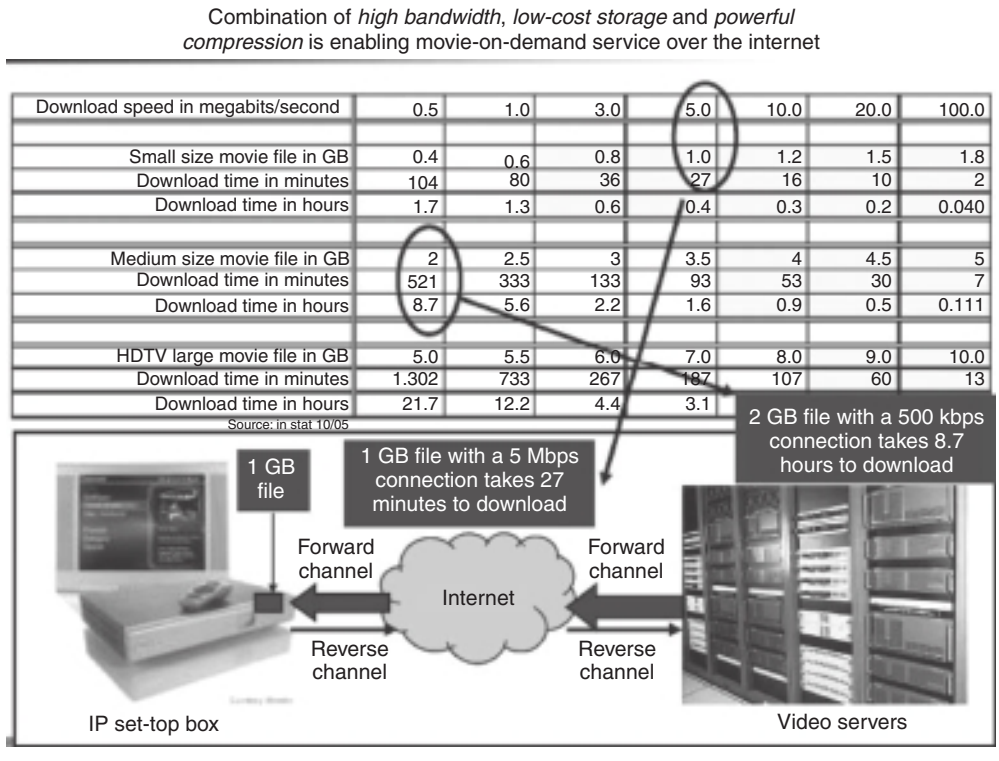


Figure 5.1 Moore's law enablers for digital convergence.

5.1 Open versus Closed Networks

Challenges of distributing video, whether by progressive download or by streaming, are the same regardless of whether the network used for distribution belongs to the distributor or not [1–4]. However, the techniques used for mitigating the challenges differ based on whether the network is owned or not owned by the distributor. When the distribution network is not owned by the distributor, it is called an "open" network and when the distributor owns the network, it is referred to as a "closed" network. For example, if a Communication Service Provider (or CSP) wants to distribute video using its own network then that is an example of a "closed" network, as the CSP is the only one who has full access to the network and has complete control over how the network is used. Specifically, the CSP can control the amount of traffic going through its network and can decide how much traffic will be sent over which paths in its network. On the other hand, if some company X other than a CSP wants to distribute video over the Internet, then that is a case of "open" network as the company X can only "use" the Internet but not "control" the traffic going through it.

One pertinent analogy with an "open" network would be the Interstate highways in the US, which can be used by anyone for transporting goods. But the transporter would have no control on traffic generated by others, and hence would be delayed from time to time based on the level of congestion on the highways. In contrast, there are walled military campuses with wide roads inside that have restricted traffic. The military is in full control of what traffic gets



Download speed in megabits/second	0.5	1.0	3.0	5.0	10.0	20.0	100.0
Small size movie file in GB	0.4	0.6	0.8	1.0	1.2	1.5	1.8
Download time in minutes	104	80	36	27	16	10	2
Download time in hours	1.7	1.3	0.6	0.4	0.3	0.2	0.040
Medium size movie file in GB	2	2.5	3	3.5	4	4.5	5
Download time in minutes	521	333	133	93	53	30	7
Download time in hours	8.7	5.6	2.2	1.6	0.9	0.5	0.111
HDTV large movie file in GB	5.0	5.5	6.0	7.0	8.0	9.0	10.0
Download time in minutes	1.302	733	267	187	107	60	13
Download time in hours	21.7	12.2	4.4	3.1			

Figure 5.2 Movie-on-demand is becoming economically feasible.

into its campus and which traffic takes which route and, as a result, delays can be kept under control. This is an analogy with a "closed" network.

Let us see some of the challenges in transporting video by streaming over a network. First, video files and/or streams are broken down into small packets that form the unit of transportation. This is the same process as for transporting any other content over packet-switched networks, such as the Internet. These packets are independently routed over the network through a series of routers towards the destination. The intermediate routers have finite buffer size. When the traffic flowing into the buffer exceeds the traffic flowing out of the buffer, packets are dropped in the network (Figure 5.3).

Loss of packets degrades the quality of experience. Therefore, the first challenge is to reduce, if not eliminate, the loss of packets in the network.

What else can impair the quality of video? One property of real-time traffic such as audio and video is that the frames/packets are generated at regular intervals. As a result, they need to be played out at exactly the same interval in which they are generated. After the video frames are generated at the source, they are broken down into packets and routed through the network as explained above. However, in the presence of many other streams in the network, packets from multiple streams are mixed at intermediate routers and they have to wait in the queue until the ones that came in before get out of the buffer (assuming First Come First Served – FCFS, scheduling at the routers). Thus, it is fairly conceivable that a large number of packets from

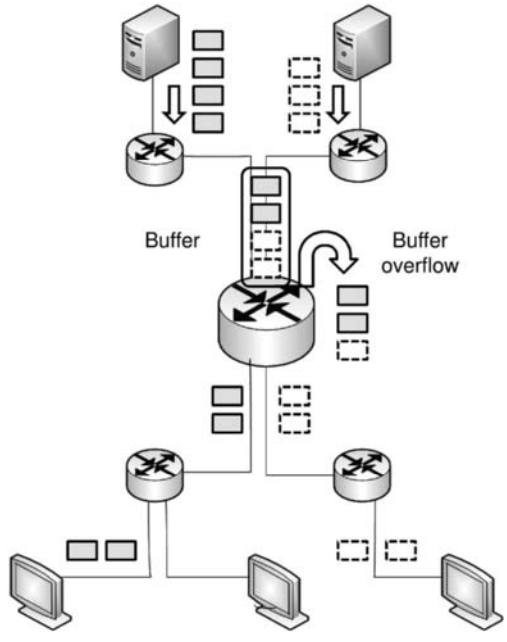


Figure 5.3 Buffer overflow leads to packet loss impacting quality of video.

other streams are injected between two consecutive packets of a given stream in the buffer of an intermediate router. This process may repeat at various intermediate routers leading to significant separation in time between two consecutive packets belonging to a given stream. Technically, the maximum separation in time between two consecutive packets is referred to as "jitter". Jitter in the network leads to uneven separation in playout time between two consecutive frames and that leads to degradation of quality of experience as well (Figure 5.4). Therefore, the second challenge is to reduce, if not eliminate jitter from the network.

5.2 Open Networks

The basic "open" network model consists of video servers that host and serve video to clients who are distributed on the Internet. More advanced model of "open" network consists of an "overlay" network of mirror servers distributed in the Internet and serving video to clients from the mirror server closest to the client. Such an overlay network is commonly known as a content distribution network (CDN) and will be covered in detail later in the book.

Let us briefly see how the video distribution challenges described above are handled in "open" networks. In this model, the clients provide feedback to the (mirror) server about the quality of video they are receiving. Specifically, the clients inform servers about the packet loss rate. Servers slow down injection of packets in the network if the error rate exceeds a threshold. This can be done in a variety of ways, one of which is to reduce the number of frames generated at the source. Note that if every video server that sees high packet loss slows down, the overall number of packets in the network reduces. Consequently, the number of

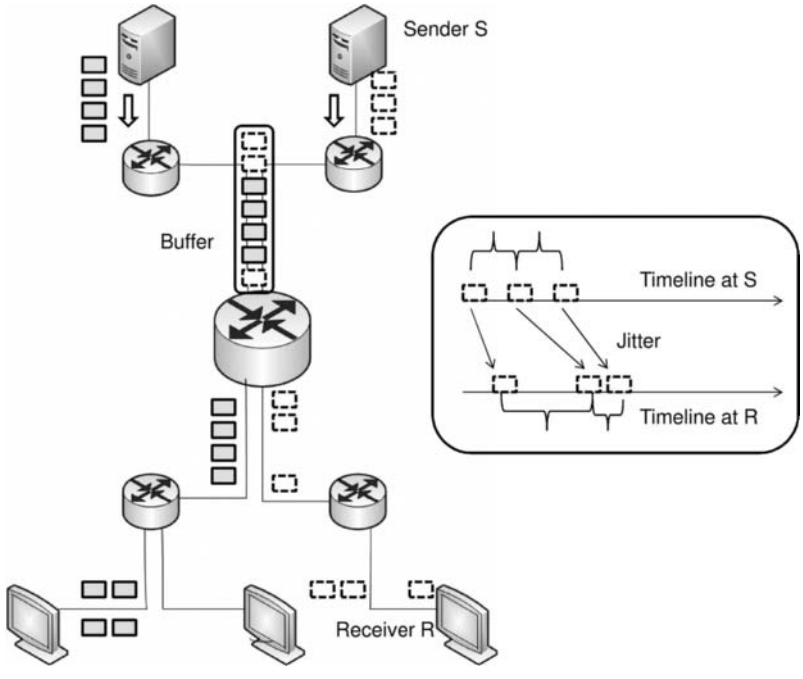


Figure 5.4 Jitter leads to uneven inter-packet gap impacting quality of video.

packets contending for buffer space in the intermediate routers also reduces resulting in lower packet loss (Figure 5.5). Note that the server "estimates" how much it should slow down to eliminate packet loss and this estimate is arrived at without knowing the actual state of the buffers in the network. Hence there is no guarantee that packet loss will be eliminated but it does guarantee that packet loss will be reduced.

To address the jitter issue, the open network architecture assumes the existence of a buffer at the client (referred to as a "jitter" buffer) that holds packets for some time (approximately 5 s) after their arrival at the client, before playing them out. By holding packets in the buffer for a short time and briefly delaying the play out, jitter introduced in the network can be absorbed (Figure 5.6).

5.3 Closed Networks

In case of the "closed" network, the service provider has complete knowledge about the topology of the network, the capacity of various pipes connecting the routers and the capacity of the routers. In addition, as video servers are also deployed in its network, it knows the capacity of the video servers in terms of the number of concurrent streams it can serve and the aggregate bandwidth it can support.

Typically, service providers do "traffic engineering", meaning that they decide how much traffic will be allowed to flow along which paths in their network so that there is no buffer overflow in intermediate routers (Figure 5.7). To prevent excess traffic from being injected

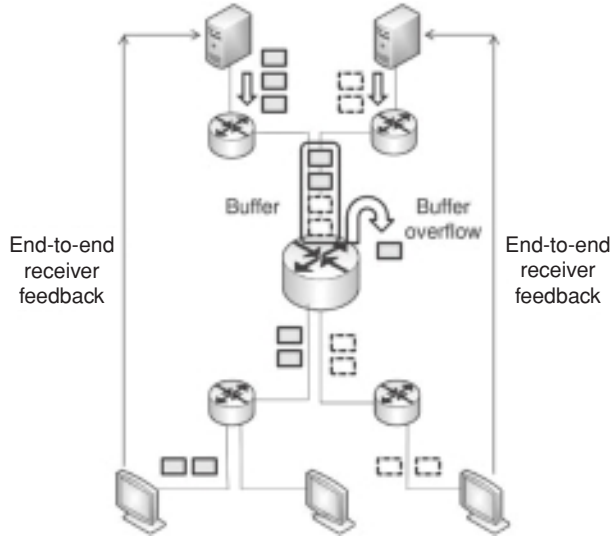


Figure 5.5 Open network solution to packet loss.

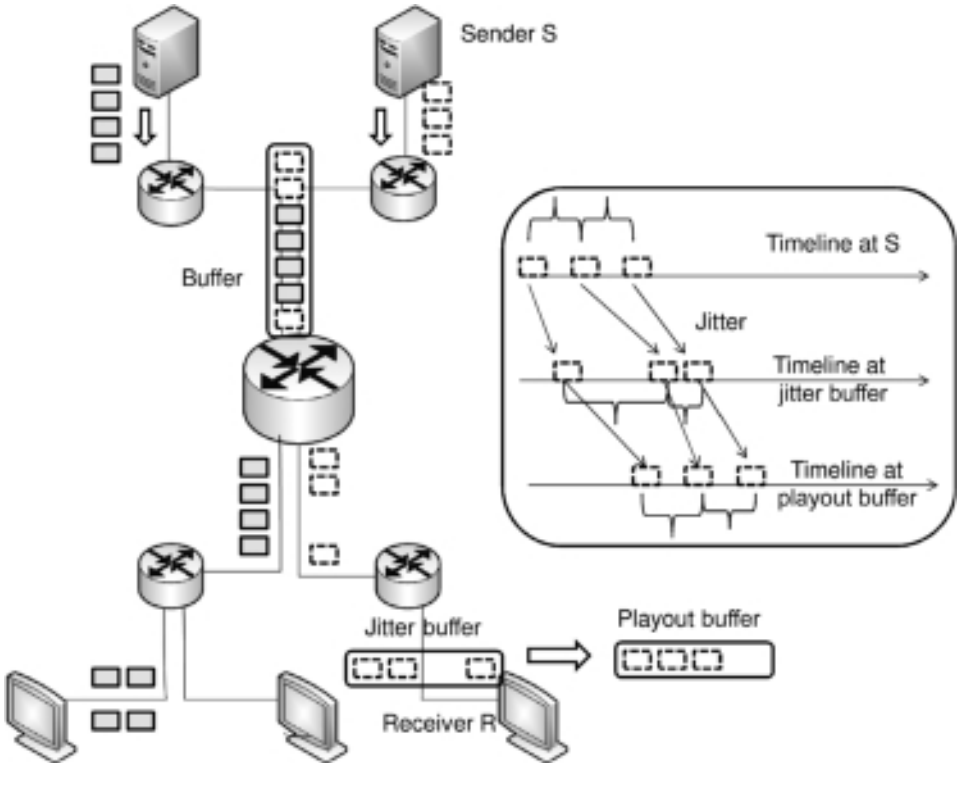


Figure 5.6 Open network solution to jitter.

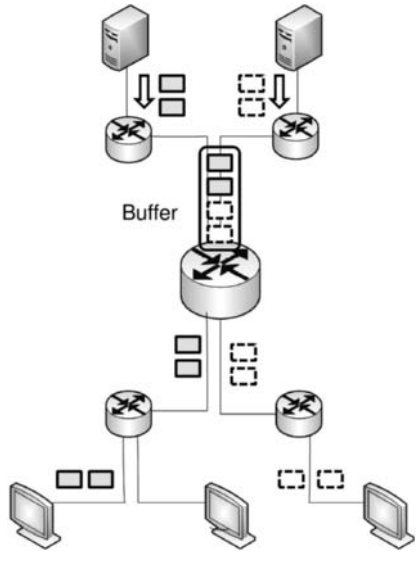


Figure 5.7 Closed network solution to packet loss.

in the network, traffic policing is done at the edge of the network and the traffic sources are constrained to the agreed upon traffic profile. With such a disciplined approach, buffers do not overflow at intermediate routers and packet loss is prevented. Note the radically different approach taken in the "closed" network compared to the one taken in "open" network to mitigate the same problem of packet loss.

Just as traffic engineering is done to avoid buffer overflows in the intermediate routers, scheduling policies are implemented in the routers to serve packets from multiple streams in a fair way (Figure 5.8). Thus packets from multiple streams in the same router buffer effectively get served in a round robin manner and as a result, the time difference between consecutive packets of the same stream is maintained within reasonable limits. Fair scheduling policy thus significantly reduces jitter compared to a first-come-first-served scheduling policy at the routers. Once again it is interesting to see that the approach taken by the "closed" network is radically different from the approach taken by the "open" network to mitigate the problem of "jitter".

5.4 Summary

This chapter claimed that the technology trend of higher bandwidth ($5\times$) and better compression ($2\times$) is providing a $10\times$ improvement in cost-performance, and is also enabling the download of a video over the Internet in a time almost comparable to the time needed to procure a video from a store, resulting in an economically viable business model for delivering video over the Internet. The challenges for smooth delivery of video over the Internet are packet loss and jitter. Open networks that do not have access to the transport network solve the problem of packet loss by slowing down the sender, based on the loss statistics obtained from the receiver(s), and address the jitter problem by using a relatively large playout buffer at the receiver. Closed networks, on the other hand, are able to eliminate packet loss by admitting

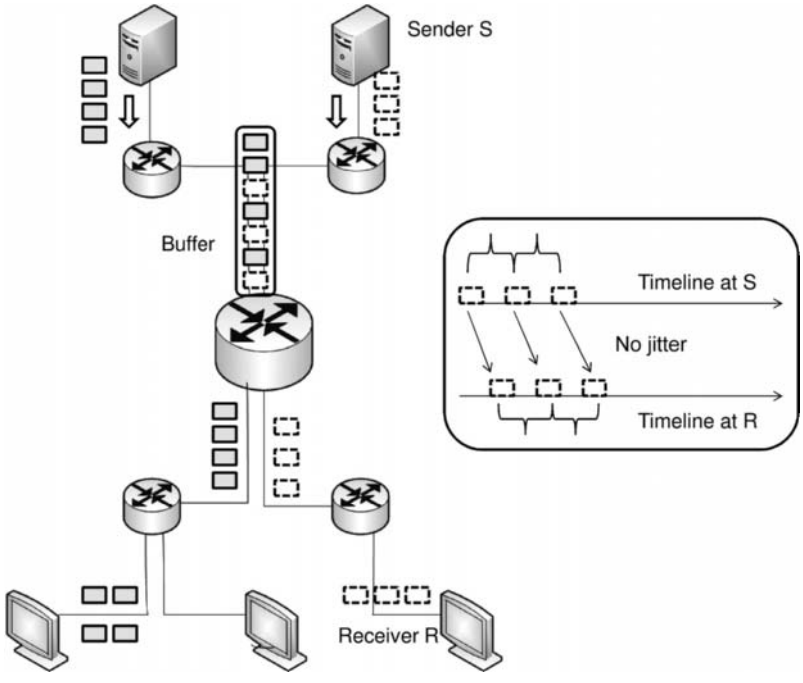


Figure 5.8 Closed network solution to jitter.

only as much traffic as can be handled by the network. They deal with jitter by scheduling at the routing elements inside the network.

References

[1] Rojo, D. and Oreggia, E. Video Streaming across wide area networks. Research on technologies for content distribution. http://xname.cc/text/video-streaming-on-wan.pdf (accessed June 9, 2010).

[2] Cooper, W. (2007) *Delivering Broadband Video*. IPTV Conference 2007 - Deployment and Service Delivery, IET, 13 Dec. 2007, pp. 1–41.

[3] Apostolopoulos, J.G., Tan, W.-T. and Wee, S.J. Video Streaming: Concepts, Algorithms, and Systems. http://www.hpl.hp.com/techreports/2002/HPL-2002-260.html (accessed June 9, 2010).

[4] McNamee, D., Krasic, C., Li, K. *et al.* (2000) Control Challenges in Multi-level ADAPTIVE VIDEO Streaming. Proceedings of the 39th IEEE Conference on Decision and Control, 2000, Vol. 3, pp. 2228–2233.

[5] SAN vs. DAS: A Cost Analysis of Storage in the Enterprise. http://capitalhead.com/articles/san-vs-das-a-cost-analysis-of-storage-in-the-enterprise.aspx (accessed June 9, 2010).

[6] Flash Memory vs. Hard Disk Drives - Which Will Win?. http://www.storagesearch.com/semico-art1.html. (accessed June 9, 2010).

[7] Keyes, R.W. The Impact of Moore's Law. (September 2006); Solid State Circuits. http://www.ieee.org/portal/site/sscs/menuitem.f07ee9e3b2a01d06bb9305765bac26c8/index.jsp?&pName=sscs_level1_article&TheCat=2165&path=sscs/06Sept&file=Liddle.xml (accessed June 9, 2010).

[8] Brock, D.C. (ed.) (2006) *Understanding Moore's Law: Four Decades of Innovation*, Chemical Heritage Press, Philadelphia. ISBN-10: 0941901416.

6

Summary of Part One

Part One of the book was dedicated to exploring trends in the industry and their impact on video distribution.

Chapter 1 focused on the megatrend of digital convergence sweeping the industry. With digitization of content, the distinction between voice, video, images and text is blurring as everything is being treated uniformly as data and transported over a common IP network. This is making specialized networks for transporting voice, video and data obsolete. Telecom (voice), broadcast and media (video) and Internet (data) are all overlapping in content distribution space. In order to provide access to these applications from anywhere and at any time, the devices (PC/laptop, mobile handsets, TV sets) are becoming more and more powerful with multiple consumer electronic features being built into them, leading to what is known in the industry as "device convergence". A variety of network technologies are converging into Internet protocol (IP)-based technology leading to mixing and matching of applications and features in any service from the end-user perspective and lower capital expense (capex) and operational expense (opex) from the service-provider perspective. Service convergence is enabling end users to obtain the same service regardless of the network over which it is accessed and the ability of end-users to access the same content over multiple devices in a seamless manner. Digital convergence is enabling new applications and services that were not possible before and opening up new opportunities both from the service-provider perspective as well as from the end-user standpoint.

Chapter 2 was dedicated to video encoding and compression. Specifically, we discussed how images are digitized, quantized, encoded and compressed, and we looked at the additional dimension of encoding color. Video encoding was discussed next as an extension of image encoding with motion estimation. Typically videos have three types of frames: I-frame, P-frame and B-frame where I-frame captures most of the information in a scene followed by P-frames and B-frames, which capture the "deltas" from the original frame (captured in I-frame) resulting from motion. Finally, I, P and B frames can be packetized and transported over an IP network just like any other type of data.

Chapter 3 was aimed at clarifying the distinction between the terms "Internet television" and "IPTV", which many people use synonymously. Internet television is the equivalent of the Web for videos, giving anyone with an Internet connection the ability to publish any video

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

content and consume any video content published by someone else. On the other hand, IPTV is the equivalent of the cable TV/satellite TV except handled by a telecom service provider. Thus the content available over IPTV is controlled by the telecom service provider and is out of reach for individuals from the video content publication point of view. Moreover, just like CableTV, IPTV has geographic boundaries over which the content is accessible and, from the business standpoint, the end user is a customer of the telecom service provider.

Chapter 4 focused on the importance of multicast in live video distribution, especially for broadcast television-type services over the terrestrial network (such as, IPTV network) as well as over the mobile network. Multicast technology enables the video server to transmit video content once regardless of the number of recipients. Multicast technology for a point-to-point network, such as, IPTV was described followed by the use of multicast in mobile wireless networks. Specifically, the concept of hybrid multicast (using a broadcast network, such as, FLO together with a point-to-point network, such as, the core and radio access network in a 3G/4G wireless network) was introduced in the context of delivering live television content over a mobile network at a price point acceptable to market.

Chapter 5 introduced the concept of video delivery over "open" and "closed" networks. Open network means "no access" while closed network means "full access" to the underlying networking infrastructure. The main obstacles to delivering video over the Internet are packet loss and jitter. Open networks that do not have access to the transport network solve the problem of packet loss by slowing down the sender based on the loss statistics obtained from the receiver(s) and address the jitter problem by using a relatively large playout buffer at the receiver. Closed networks, on the other hand, are able to eliminate packet loss by admitting only as much traffic as can be handled by the network. They deal with jitter by scheduling at the routing elements inside the network. Most importantly, the chapter provided the reason for video delivery over the Internet becoming a viable business at this point of time. Specifically, the technology trend of higher bandwidth ($5\times$) and better compression ($2\times$) is providing a $10\times$ improvement in cost-performance, and also enables a video to be downloaded over the Internet in a time almost comparable to the time needed to procure a video from a store. As video distribution is becoming economically viable, new business models are emerging and they are described in the remainder of the book.

In summary, digital convergence is happening in the industry, in the device, in the network, as well as in the realm of services. As a result, consumers will soon be expecting to access video content and video-related applications from anywhere using any device at any time. Video encoding and compression technology have advanced to a stage whereby video can be transmitted in packetized form over an IP network without losing quality using significantly lower bandwidth compared to what was the case several years ago. This has been possible due to the advances in video encoding and compression technology. Furthermore, due to the availability of higher bandwidth in residences, video download time over the Internet is becoming increasingly comparable to that of making a trip to the nearest video rental store and renting the video of choice. The combination of improved user experience and lower cost is leading to the possibility of Internet television, which is the equivalent of the Web for video content. Anyone would be able to publish any video content and anyone would be able to consume any video content with the desirable quality of experience. When it includes the "broadcast" TV content as well, television becomes an application over the Internet just as phone services have become an Internet application (voice over IP). It is important to understand and appreciate the difference between Internet television and IPTV where the latter

is the "equivalent" of the traditional television service as provided by the cable and (direct-to-home) (DTH) satellite service providers except that the provider of IPTV is a telecom operator. Focusing on the challenges of delivering of video over the Internet, we looked at how the same problems of packet loss and jitter are dealt with differently in open and closed networks due to the different levels of access they have to the transport network.

The rest of the book describes technology and business models for video distribution in "open" and "closed" networks.

Part Two

Challenges of Distributing Video in OPEN Networks

Probably the best way to understand and appreciate the challenges of video distribution in "open" networks is to study the problem from the perspective of an independent company trying to enter the business of providing video distribution from scratch. Assume that the company wants to provide the following specific services:

- Movie-on-demand over the Internet.
- Internet television (independent rights holders).
- Broadcast television (cable/satellite television equivalent service) over the Internet.

We will look into the specific challenges that arise when a company that does not own or manage the underlying transport network tries to offer the above services.

7

Movie-on-Demand over the Internet

7.1 Resource Estimation

To offer a movie-on-demand (MoD) service, it is necessary to understand the resources needed to operate such a service. Based on resource needs, cost of operations can be estimated for various alternative ways of providing the service. Primary resources in offering a MoD service are storage and bandwidth. When it comes to data download, which happens in bursts, rather than media streaming, which happens continuously, the charging happens by the amount of data download rather than by the bandwidth of the pipe. Therefore, in addition to storage and bandwidth, it is important to estimate the amount of data download.

7.1.1 Storage

Approximately 500 movies are produced per year in the US and 5000 movies are produced per year across the globe. To calculate storage needs, it is necessary to know how much storage is needed per movie. A standard definition (SD) MPEG-2 encoded movie of duration 1.5 hour typically takes 6.75GB of storage and a high definition (HD) version takes 10GB of storage. Thus if the goal is to make available all movies produced in the US for the last fifty years, assuming 500 movies per year, the required storage space is $500 \times 6.75 \times 50 = 168.75$TB. Depending on the scope of the service, storage requirements can be computed accordingly.

7.1.2 Bandwidth

Assume that the service provider wants to provide service only to customers in the US. The number of households in the US with broadband data connection at home was approximately 55 million in 2009. Out of this pool, suppose the service provider is able to attract 10%, meaning that the customer base would be 5.5 million. Not all customers will be using the MoD service at the same time. Assuming 10% simultaneous requests and a mean bandwidth of

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

1 mbps, the aggregate bandwidth need would be 550 gbps which is extremely high! To reduce load on the network, an important strategy would be to prepopulate the set top box (STB) with the most popular movies or better still with the movies that fit the customer's profile. The process of populating the STB with the "most likely to be watched" movies would have to be done periodically to keep the STB's cache updated. If "caching" the right content on the STB can serve 90% of requests, then only 10% of the requests would have to be served by the servers belonging to the service provider. This will bring down the bandwidth requirement by an order of magnitude to 55 gbps.

7.1.3 Download

To estimate the amount of data that would be downloaded by the customers, it is necessary to estimate the number and size of requests. Assume 20% of customers download four movies per month. Each movie is 1.5 hours of standard definition (SD) MPEG-2 and is of size 6.75GB. Furthermore, assume smart caching at STB provides a 90% hit rate. Based on the above assumptions (5.5 million customers), the total amount of data downloaded by customers from the service providers network would be $5\ 500\ 000 \times 0.2 \times 4 \times 6.75 \times 0.1\text{GB} = 2.970\text{PB/month}$.

7.2 Alternative Distribution Models

7.2.1 Content Distribution Network (CDN)

Content on the Internet is accessed from everywhere in the world. All it requires is a PC that is connected to the Internet. For example, content available in the US would be accessed by people on the other side of the world, such as, India or China. Due to the physical distance between the server in the US hosting the content and the clients in India and/or China, there is a significant amount of **propagation delay** which is computed by the physical distance divided by the speed of light. Furthermore, the time needed for physical transmission of bits over a transmission line depends on the speed (bandwidth) of the transmission line. This is referred to as **transmission delay**. In addition to the propagation delay and the transmission delay, there is another component of delay, which contributes to the overall end-to-end delay in transmitting content from a server to the client. This delay component is referred to as **queuing delay**. Queuing delay arises from the fact that the packets containing content, while traversing several networks from the server en route to the client are queued at the intermediate routers from time to time. The more the number of networks and routers the packets have to traverse, the higher is the probability of longer queuing delay. The sum total of these delay components manifests in the end-to-end delay that a client from India or China sees when accessing content from a server in the US.

When considering streaming of video under the exact same circumstances, **jitter** becomes a big issue in addition to delay. In fact, the likelihood of jitter being greater increases as the packets traverse more networks and routers from the server en route to the client.

Packet loss would also have a tremendous impact on the quality of video streaming, especially if the packet loss results in losing key frames, such as, I-frames in MPEG. If the packets traverse many networks and routers, likelihood of packet loss increases and recovering the lost

packets by retransmission mechanism does not work because by the time the retransmitted packets arrive at the client, the deadline for playing out the packet might be past.

From a content provider's perspective, such deterioration of quality of experience for its potential customers is not acceptable.

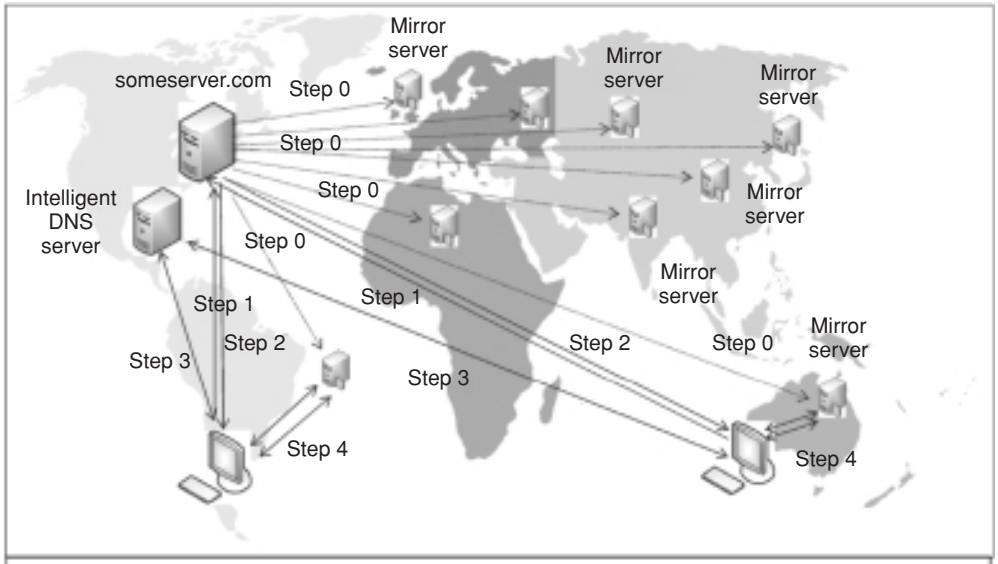
In order to address this issue faced by many content providers, a new breed of companies came into existence and they started to provide a service called content distribution network (CDN) service [1–9]. Content distribution network service providers create an **overlay network** of servers that are strategically deployed in the Internet. Content distribution networks are overlay networks because these networks are composed of mirror servers (also called replica servers) which are nothing but end systems (that terminate TCP/IP connections). Mirror servers sitting on top of the Internet leverage the network-level service of the Internet. Mirror servers are placed in the data centers of the Internet service providers (ISPs). Thus a CDN consists of an overlay network of large number of mirror servers distributed over the Internet.

The objective of having geographically distributed mirror servers is to be able to serve the clients from the "nearest" mirror server in order to mitigate the effects of propagation delay, transmission delay and queuing delay. In the context of video streaming, jitter is likely to be reduced as the number of routers that are instrumental in injecting jitter is significantly reduced between the client and the mirror server from which video is streamed. For exactly the same reason the probability of packet loss is also reduced. The overall user experience is thus significantly improved.

There are some subtle points that need to be understood and appreciated in the context of CDN. First, CDN is a "shared" infrastructure in that the mirror servers belonging to a CDN service provider are usually "shared" between multiple content providers. These content providers are the customers of the CDN service providers. Second, the value provided to the content providers by the CDN service providers is measured by the amount of content (GB) downloaded from the mirror servers and the peak bandwidth needed to serve the content from the mirror servers, because in the absence of CDN service provider, their own infrastructure would have to do the same.

A representative deployment of CDN is shown in Figure 7.1. Note that the origin server is in the US and there are several mirror servers all over the globe. Content from the origin server is replicated on to the mirror servers. Typically, large and static content, such as images and videos, are replicated in the mirror servers as shown in step 0 of the diagram. When the client requests content from the origin server (step 1), the origin server returns a web page (step 2) in which the embedded links referring to static content (images and videos) contain "symbolic" names of mirror servers. The "symbolic" name of the mirror server (from which images/videos will be eventually retrieved) is then resolved using DNS server of the CDN service provider (step 3) to the IP address of the mirror server "nearest" to the client. The client then establishes a connection with the "nearest" mirror server and retrieves the images and videos (step 4).

One of the main challenges for CDN service providers is to be able to find the "nearest" mirror server corresponding to a client. Typically, the IP address of the client is used to identify the location of the client. Once the location of the client is found, finding the location of the "nearest" mirror server is relatively straightforward. The exact algorithm used by different CDN service providers to find the nearest mirror server for a client is proprietary.



Step 0: Push all image/video files to the mirror servers
Step 1: Request index.html file from someserver.com
Step 2: Respond with index.html in which the links to image/video files are replaces with symbolic links. For example, http://cdnmirror01.net/someserver.com/images/image1.gif
Step 3: Contact the "Intelligent DNS server" to resolve the server name cdnmirror01.net into the IP address of the "closest" mirror server
Step 4: Retrieve the image/video files from the "closest" mirror server

Figure 7.1 How content distribution network (CDN) works.

7.2.2 Hosting

When a content provider decides to host, manage and maintain its own content on a server or bank of servers without depending on any third party, such as, a CDN service provider, it is referred to as hosting [28–30]. Typically content hosting is done by renting rack spaces in a data center and placing the servers there and by leasing network connectivity from the data center service provider. However, if the hosting is done in one location, all requests will be served from that location leading to exactly the same problems discussed in the previous section. Specifically, the servers would have to serve millions of requests per day if the site is popular; this leads to poor performance resulting in inferior quality of experience for the end users. In the context of video, the servers would have to serve multiple concurrent requests for streaming, which would consume significant bandwidth and may impact end-user quality of experience. Furthermore, if the hosting center is in the US and the requests for streaming video have to be served to clients from India, the quality of experience will suffer significantly. The reasons have been discussed in the last section.

Essentially, the problems with serving content, especially video, are the same from the perspective of content provider irrespective of whether the content provider decides to go for hosting or for CDN. Content distribution network service providers have a scalable distributed infrastructure in place that content provider can use at a price or can build a similar geographically distributed infrastructure by renting rack spaces, placing their servers and leasing bandwidth at multiple data centers across the globe.

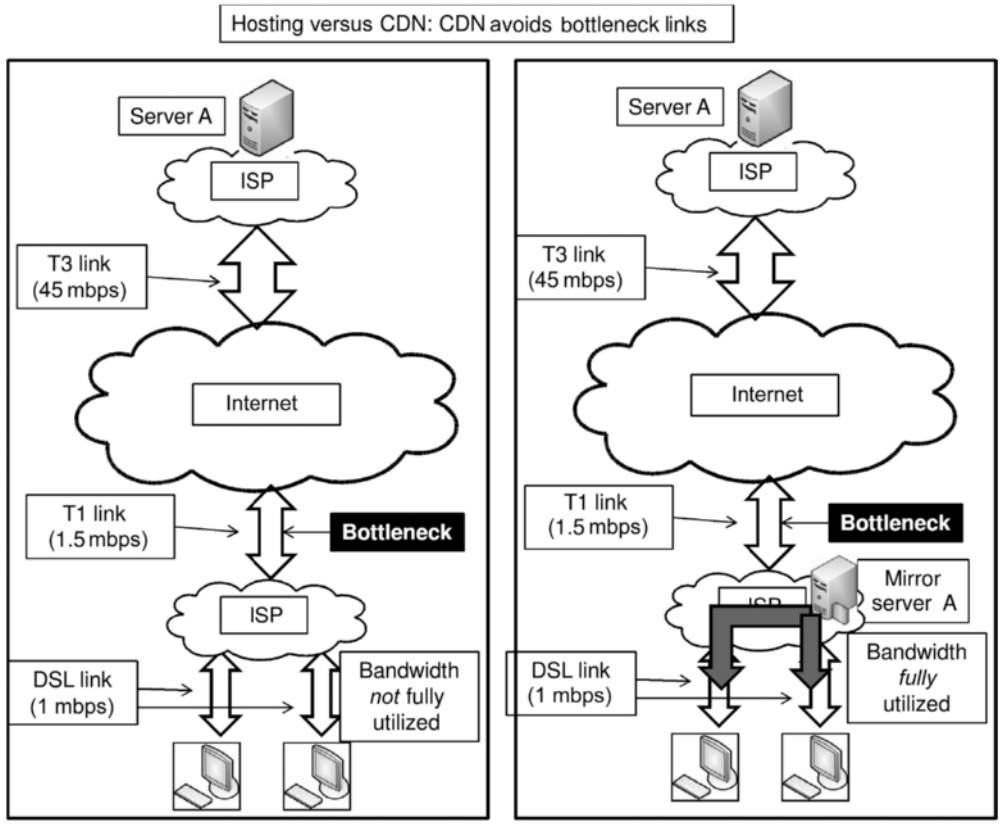


Figure 7.2 Content distribution network avoids bottleneck links.

When we explore the economics of CDN versus hosting in the context of an Internet-based video service provider, it will become clear that CDN has an advantage over hosting mainly because a CDN service provider is able to amortize the cost of such a huge globally distributed infrastructure over all its customers and is able to keep the utilization of such a capex- and opex-intensive infrastructure fairly high by statistically multiplexing traffic from multiple content providers.

7.2.3 Hosting versus CDN

Hosting provides higher control on the content as the content provider determines what content should be replicated where. On the other hand, CDNs provide several advantages over hosting:

- Content distribution networks avoid bottleneck links (Figure 7.2).
- They provide higher availability (Figure 7.3).
- They reduce the likelihood of bottleneck links between the client and the mirror server (Figure 7.4).

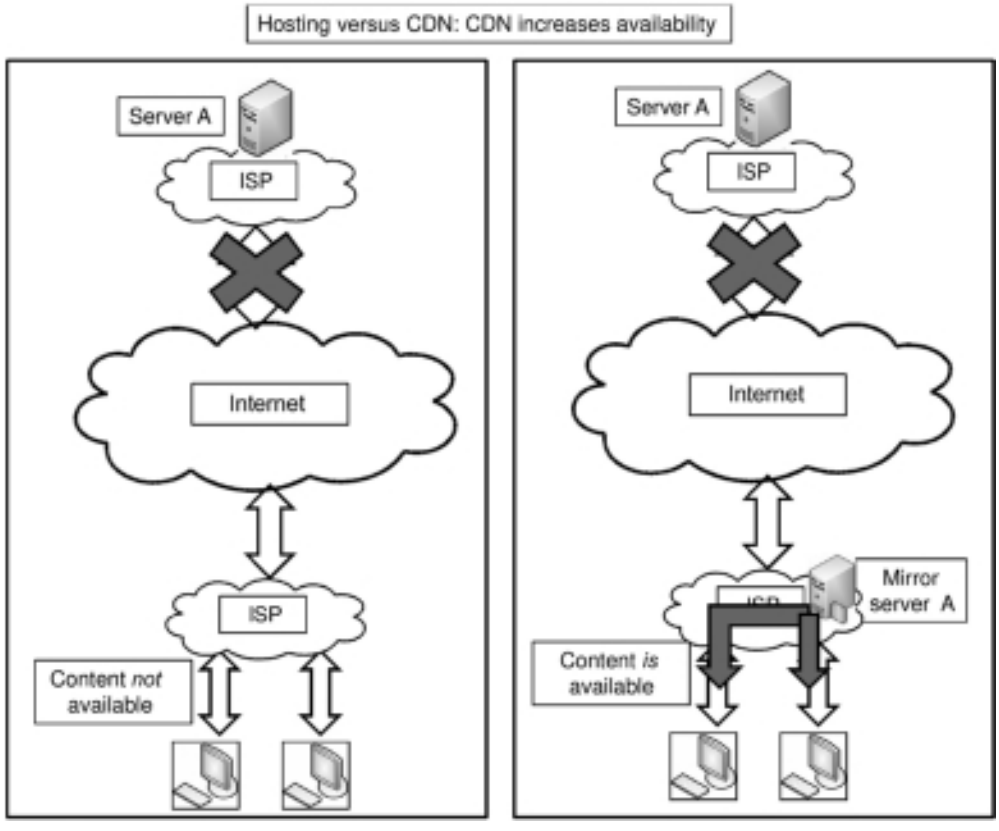


Figure 7.3 Content distribution network increases availability.

In Figure 7.2, hosting is shown on the left while CDN is shown on the right. Server A hosts the content with a link of bandwidth 45 mbps connecting its ISP to the Internet. Last mile links are shown to be 1 mbps but the upstream link of the client's ISP has a T1 (1.5 mbps) link, which is the bottleneck in the end-to-end path between the server and the client. In the hosting model where content is hosted on server A only, the last-mile bandwidth of the client is not fully utilized as the bottleneck is the upstream link of the client's ISP. In the CDN model where a mirror server is placed in the client's ISP, the last-mile bandwidth of the client is fully utilized. Note that, even in the hosting model, one could imagine placing a mirror server in the client's ISP but the cost of deploying mirror servers for a single content provider would be prohibitively high compared to that of a CDN who is able to amortize the cost of mirror servers by replicating the contents of "multiple" content providers in the same server.

Figure 7.3 is used to illustrate the point of higher "availability" in the CDN model compared to that in the hosting model. Note that when the ISP corresponding to the hosting server is disconnected, the hosting model fails to deliver content to the requesting clients. However, in case of the CDN model, the mirror server located in the client's ISP is able to deliver content under exactly the same condition.

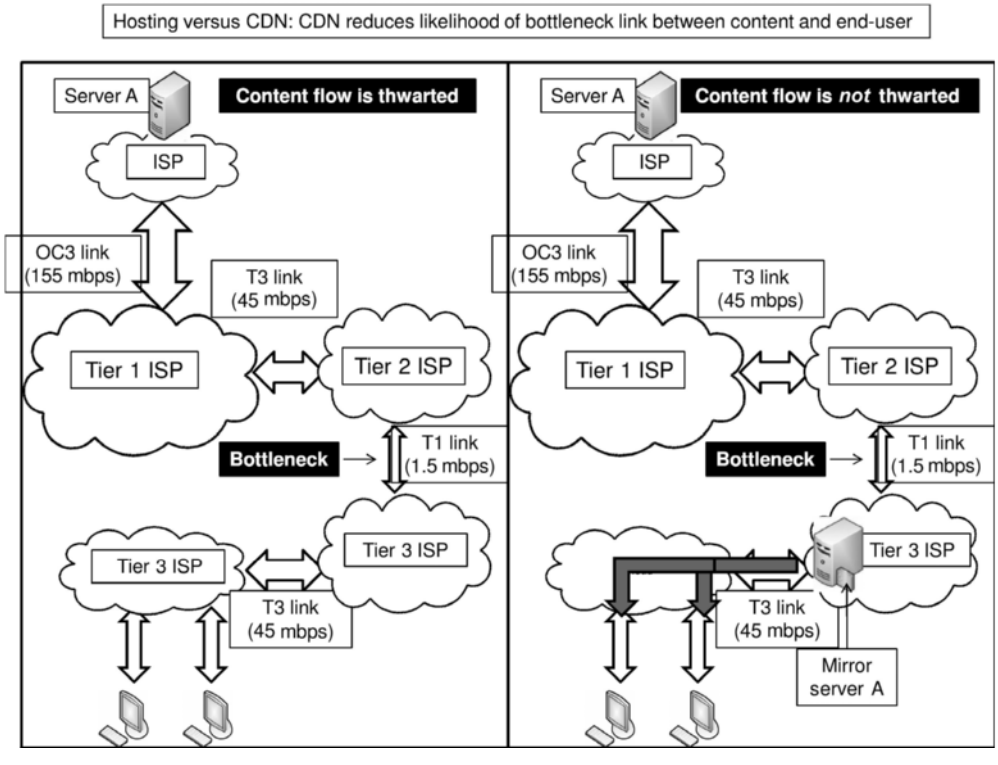


Figure 7.4 Content distribution network reduces the likelihood of bottleneck Links.

Figure 7.4 shows a chain of ISPs linking the hosting server A and a client requesting content. In general, there would be many hops between the networks of the server and the client, especially if the client is located in a remote country. As more and more clients try to access content from remote places, the likelihood of having a bottleneck link in the end-to-end path increases. In the context of video streaming, such a bottleneck will significantly worsen the quality of experience. However, due to the placement of large number of mirror servers in the CDN, the chances are higher that the content will be served from a "nearby" mirror server, thereby improving the quality of experience.

7.2.4 Peer-to-Peer (P2P) Networks

It is clear that to make content available to end users in an acceptable manner there needs to be enough storage and enough bandwidth. Both hosting and CDN require "dedicated" storage to physically store content and "dedicated" servers with "dedicated" bandwidth to serve them from the storage. In case of content hosting these dedicated resources are owned and paid for by the content provider, while, in the case of CDN, these dedicated resources are owned and partially paid for by the CDN service provider, and partially paid for by the content provider through the price of the CDN service.

An alternative approach to content distribution is to harness the storage and bandwidth of the end-user systems and create a globally distributed pool of resources that together would be able to serve the end users without compromising on the quality of experience. Furthermore, because this approach is a "grounds-up" community approach, end users are willing to "lend" their resources in exchange for receiving the desired content from other end users (referred to as peers) with the desired quality of experience. This approach in which peers exchange content and provide the implicit infrastructure for content distribution is referred to as a peer-to-peer (P2P) network.

From the perspective of content owners, P2P network-based distribution of content is the most economical option because the content owner does not have to spend either on the cost of infrastructure or on paying the CDN service providers for their service. Instead it would be able to distribute its content for "free" by leveraging the resources (storage and bandwidth) already "paid for" by the end users.

However, there are several drawbacks of P2P networks, mainly from the security and copyright angle. The fact that P2P network-based content distribution depends on the end users, who may not be trusted partners, opens up security holes and raises concerns. Furthermore, P2P networks [31, 32] acquired a bad name due to illegal distribution of copyrighted content.

7.2.5 P2P Networks for Content Download

This is an area that has been covered at great length in the literature [14–20]. In this book, the basic ideas are touched upon while references are given for details on any one of the topics of interest in P2P networks. There are two distinct approaches to P2P networks: (i) structured P2P networks and (ii) unstructured P2P networks.

The main idea in "structured" P2P networks is to organize the end systems (or peers) in a structured manner following specific rules and also to distribute content among the end systems (peers) in a "structured" manner following prespecified rules. Because of this structured organization of content, when a peer looks for specific content, it is easy to follow the prespecified rules to locate the exact peer that has the desired content. The structured P2P network is based on the key concept of a distributed hash table (DHT) [33–36], which provides the rules for organizing the peers, adding and deleting peers, distributing content among the peers and locating content given a request from a peer. There are several popular structured P2P networks, such as, CHORD [34], PASTRY [37], RON [38] and so on.

In contrast to structured P2P networks, in which content is distributed and retrieved in accordance with specific rules, in unstructured P2P networks content is distributed and retrieved in a dynamic manner. The concept of unstructured P2P networks is embodied in Gnutella [18] in which, when a peer wants to retrieve any content, it floods the P2P overlay network with queries for the content, and whichever peer has the content responds to the query followed by retrieval of content from that peer. While Gnutella is an example of an early version of P2P network, BitTorrent [16, 17] is an example of a later version of unstructured P2P network, which is much more sophisticated. In a nutshell, BitTorrent works as follows (Figure 7.5).

When a user wants to retrieve content from a P2P network, it contacts a Web server and requests a file with a .torrent extension (FILE.torrent in Figure 7.6). The web server returns the requested file which contains the IP address and port number of what is called the tracker (also known as AppTracker). The user (peer) then registers with the tracker, which in turn

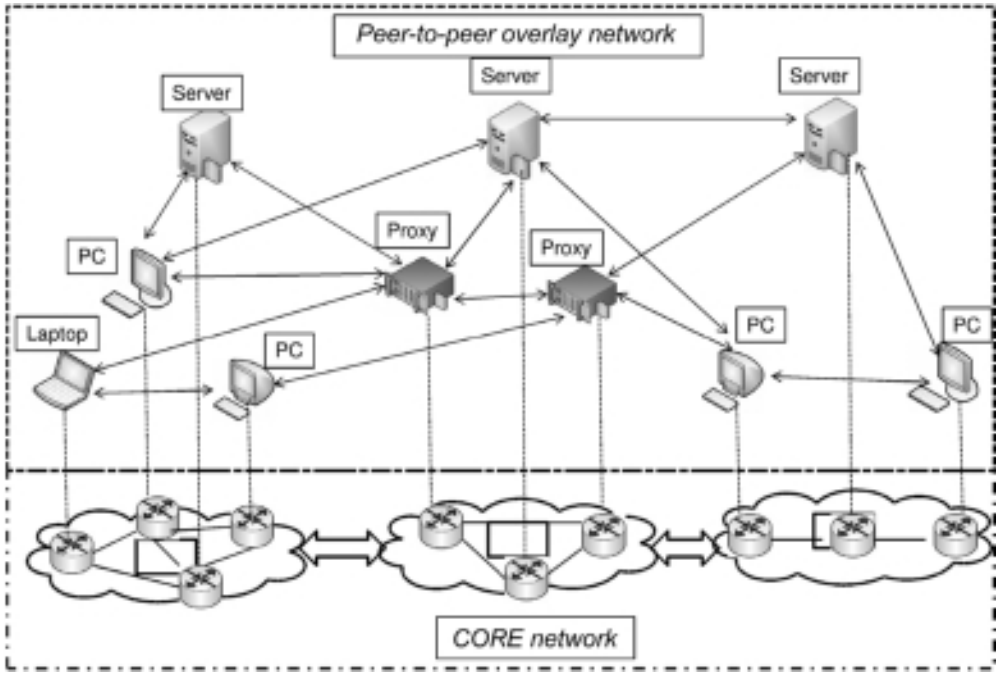


Figure 7.5 Peer-to-peer (P2P) network.

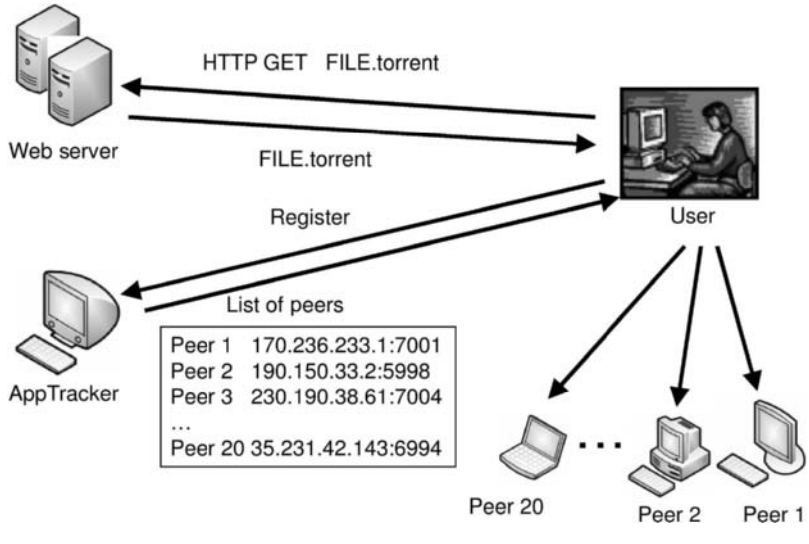


Figure 7.6 How BitTorrent P2P network works.

provides the requesting peer with a list of peers (IP address and Port Number) that are either looking for the same content or are already loaded with the same content (referred to as seeds). Content is usually divided into smaller segments and distributed in the P2P network. At any point of time, a specific peer interested in a specific content has a list of segments available with it. As this list is shared among the peers interested in the content, they use some algorithm to decide which fragments should be retrieved from which peers and proceed accordingly. In fact, for the sake of fairness, BitTorrent follows a "tit-for -tat" rule in which a participating peer receives segments from other peers based on its contribution [16, 17]. For example, if a peer does not contribute much, either because of bandwidth limitations or for some other reason, it is penalized by other peers by slowing down delivery. On the other hand, peers that contribute generously to the community receive desired segments quickly from other peers.

7.2.6 CDN vs. Peer-to-Peer (P2P) Networks

Both CDNs and P2P networks help distribute content. However, CDNs require dedicated infrastructure, such as mirror servers, which are connected to the Internet at strategic locations. Peer-to-peer networks, on the other hand, ride on the resources provided by the end systems. Essentially, P2P networks depend on the principle of helping one another in terms of providing resources: storage, processing and bandwidth. Let's try to understand the economics of CDN and P2P networks to appreciate which one of the two is less expensive from the viewpoint of the content providers.

As shown in Figure 7.7, the bandwidth of the ISP's link connecting Server A to the Internet is partially paid for by the content provider, albeit indirectly through the CDN service provider. Payment is "partial" for a specific content provider as the payment for the bandwidth is shared by all the content providers. In addition, the bandwidth needed for connecting mirror servers to the Internet is paid for by content providers indirectly through the CDN service providers.

As shown in Figure 7.8, similar to the CDN case, the bandwidth of the ISP's link connecting Server A to the Internet is partially paid for by the content provider. However, the content provider, unlike in the CDN case, does not have to pay for the bandwidth of the link connecting the end-user's ISPs to the Internet. That is paid for by the ISP. In addition, when user Y retrieves content A from X, the uplink bandwidth is paid for by X and the downlink bandwidth is paid for by Y. Similarly, when user Z retrieves content A from user W, the uplink bandwidth is paid for by W and the downlink bandwidth is paid for by Z. Thus, comparing apples to apples, the cost of distributing content A is less in case of P2P networks compared to that in the case of CDNs.

7.2.7 CDN versus Caching

Content distribution networks use mirror servers, which are nothing but local storage devices as part of their infrastructure. Caching is a technology that also depends on local storage devices but they are used by ISPs to reduce their operational expenses. In this section, we take a critical look at the two technologies and try to understand similarilities and differences. First of all, CDNs provide content distribution service to content providers by replicating their content on the mirror servers. By contrast, proxy caches are used by the ISPs to "cache" all

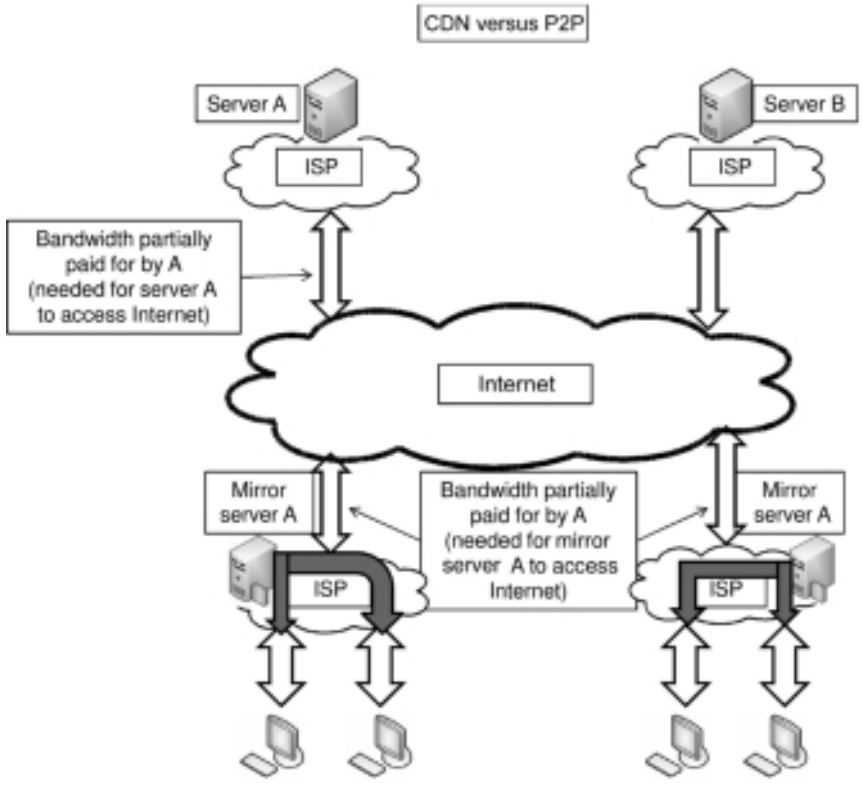


Figure 7.7 In CDN, bandwidth is paid for by content provider.

contents (not specific content of specific content providers as in the case of CDNs). Proxy caches can serve content to the end users without retrieving content from the origin servers, thereby saving bandwidth for ISPs.

Figure 7.9 shows two content providers, A and B. A is a customer of a CDN service provider that has deployed mirror servers in ISP X and ISP Y. Content distribution network service provider replicates content of server A in the two mirror servers. As a result, when an end user tries to access content A, it retrieves it from the mirror server as opposed to from the origin server (server A). As a result, content provider A does not need to have as fat a pipe at the source as it would need otherwise. This saves bandwidth cost for content provider A.

From the perspective of ISP X and ISP Y, bandwidth is saved "only" for content A as the mirror servers store only content A and not content B.

Figure 7.10 is exactly the same as Figure 7.9 except that the mirror servers are replaced with proxy caches in ISP X and ISP Y. Note that the proxy caches all contents (A as well as B) that have been requested at least once by one of the end users in ISP X (Y). That means, unlike in CDNs, no content is "proactively" stored in proxy caches. Because of content-provider-agnostic caching of content, proxy caches save bandwidth for ISPs. Moreover, due to "reactive" caching of content, proxy caches do not save as much bandwidth for content providers as is saved by the CDN service providers.

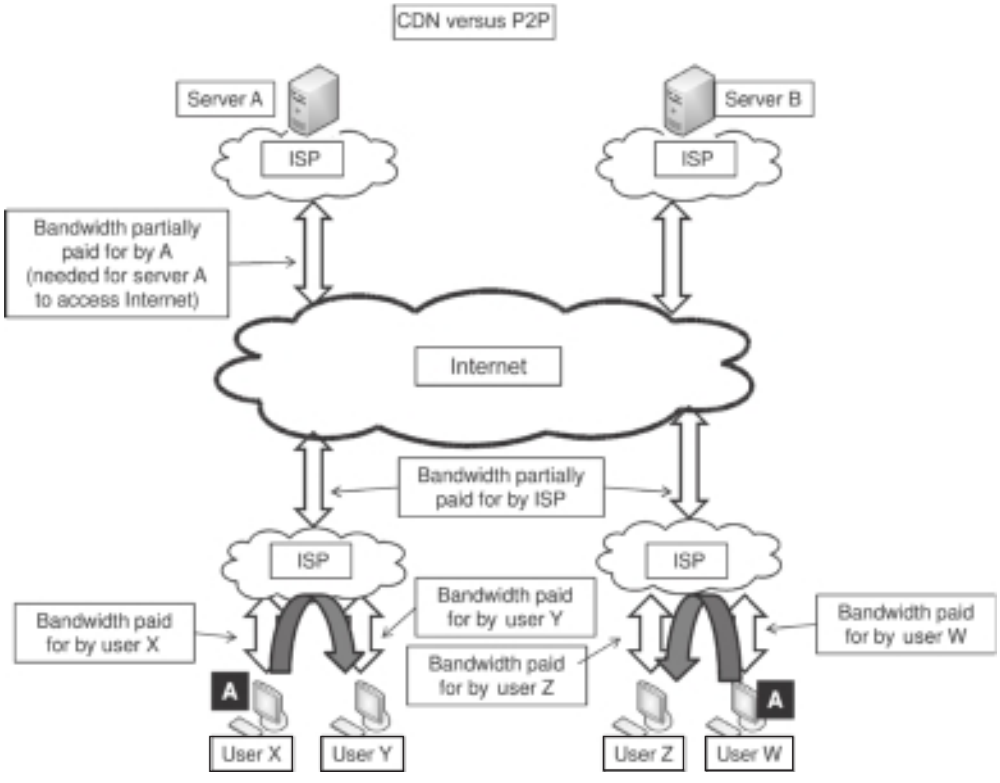


Figure 7.8 In P2P, bandwidth is paid for by ISP and users.

7.2.8 Hybrid Networks

On one hand, the CDNs with dedicated infrastructure comprising mirror servers provide a controlled environment for distributing content, while on the other hand, P2P networks provide the ultimate scalability. Naturally, the optimal solution would be to combine the best of both worlds. One such approach is to have a limited infrastructure of a number of "trusted" and "fully controlled" dedicated servers and use them for authenticating the individual peer machines that would form the major part of the P2P distribution network.

In the above hybrid system, the "trusted" servers are responsible for:

- Authenticating individual peers.
- Doing a traceroute to individual peers to form a topological tree.
- Providing to each peer a prioritized list of peers (approximately 10–12) for sourcing content.

The peers in the hybrid system are responsible for:

- Opening multiple simultaneous connections with peers in the list provided by the "trusted" server. The typical number of connections is seven or eight.

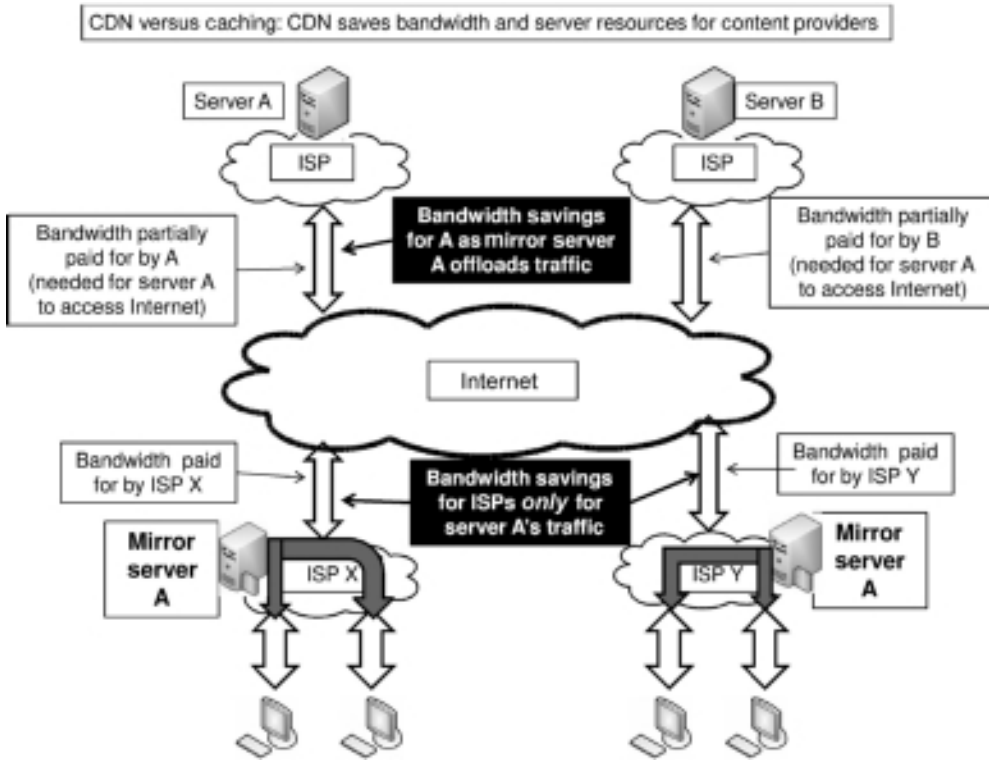


Figure 7.9 Content distribution network versus caching: CDN saves bandwidth for content providers.

- Downloading different chunks of a file from different peers in the list.
- Monitoring rate of download from individual peers and saturating the incoming bandwidth with flows from multiple peers. In a typical broadband connection, the ratio of downlink to uplink bandwidth is 5:1 and hence it requires about five peers to fill the pipe.

7.2.9 Combining Caching and P2P

Peer-to-peer as well as CDN are solutions for content providers. As described in the last couple of sections, the goal of content providers is to be able to reach out to end users in a scalable way and provide them with the best possible quality of experience. In the process, content providers would also like to have enough redundancy to survive server failures and to be able to balance load across multiple mirror servers. Both P2P and CDN help them meet their objectives.

It is to be noted that when a mirror server is deployed by a CDN provider in an ISP's point of presence (PoP), it *improves user experience only for those sites who are the customers* of the CDN. This is because the CDN mirrors content only of its customers. This is in sharp contrast to a proxy cache, which when deployed by an ISP, caches content from all sites visited by the ISP customers, and hence improves *user experience for all sites on the Internet*.

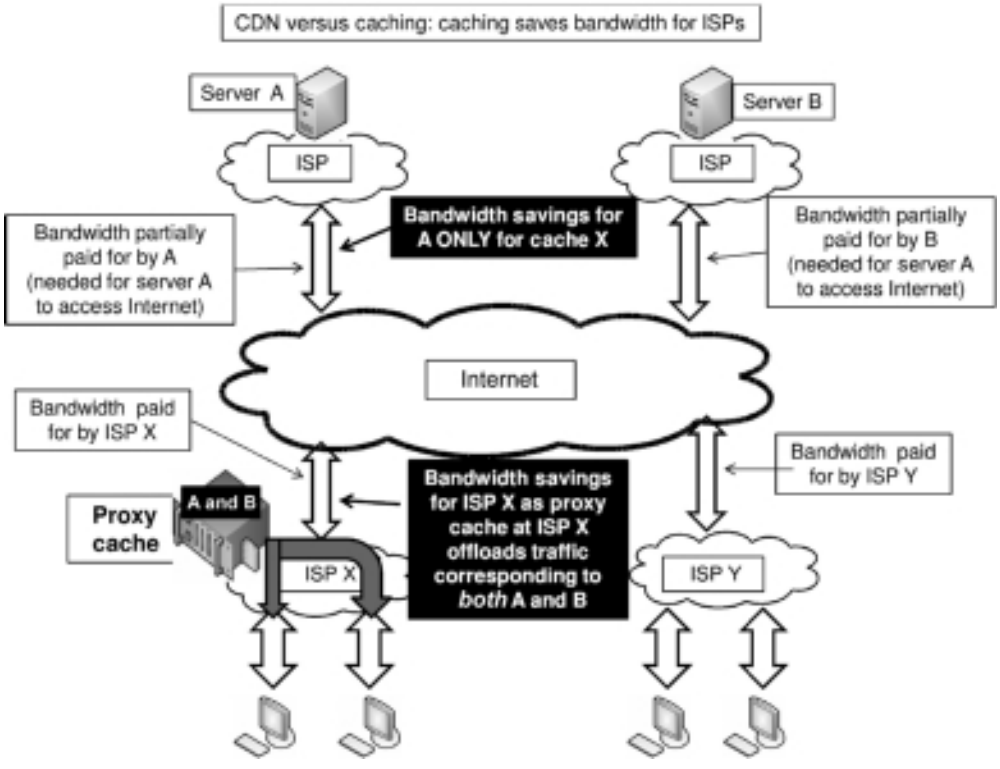


Figure 7.10 Content distribution network versus caching: Caching saves bandwidth for ISPs.

In the context of P2P networks, the peers are connected in the "logical" plane through TCP/IP connections and hence it is possible that two peers that are "logically" adjacent are actually "physically" far (in two different ISP networks). As a result, when a peer (in ISP X) tries to retrieve content from a "logically" adjacent peer (in ISP Y), traffic actually flows through the uplink of ISP Y (which is paid for ISP Y) and the downlink of ISP X (which is paid for by ISP X) as shown in Figure 7.11. Most interestingly, the beneficiary of this traffic flow is the "content provider" who is able to distribute content for "free" as the traffic generated by the peers in the P2P network for distributing content rides on the last mile bandwidths paid for by the peers and the uplink ISP bandwidths paid for by the ISPs. One subtle point to understand here is that when the ISPs decide on the capacity of the uplink in their network, they do it based on the traffic generated *explicitly* by the end users and not on the traffic generated *implicitly* by the end users as a result of participating in P2P networks. As a result, ISPs' uplink bandwidth becomes saturated leading to poor performance for their customers. In order to alleviate the poor user experience of their customers, ISPs would have to invest in upgrading their uplink bandwidth. Unfortunately, the beneficiary of this upgrade becomes the "content providers" again as they will now push more traffic through the ISPs' uplink. Essentially, this is a vicious cycle in which the ISP becomes the victim and never realizes the return on investment (ROI) on the expenses incurred in upgrading the uplink bandwidth.

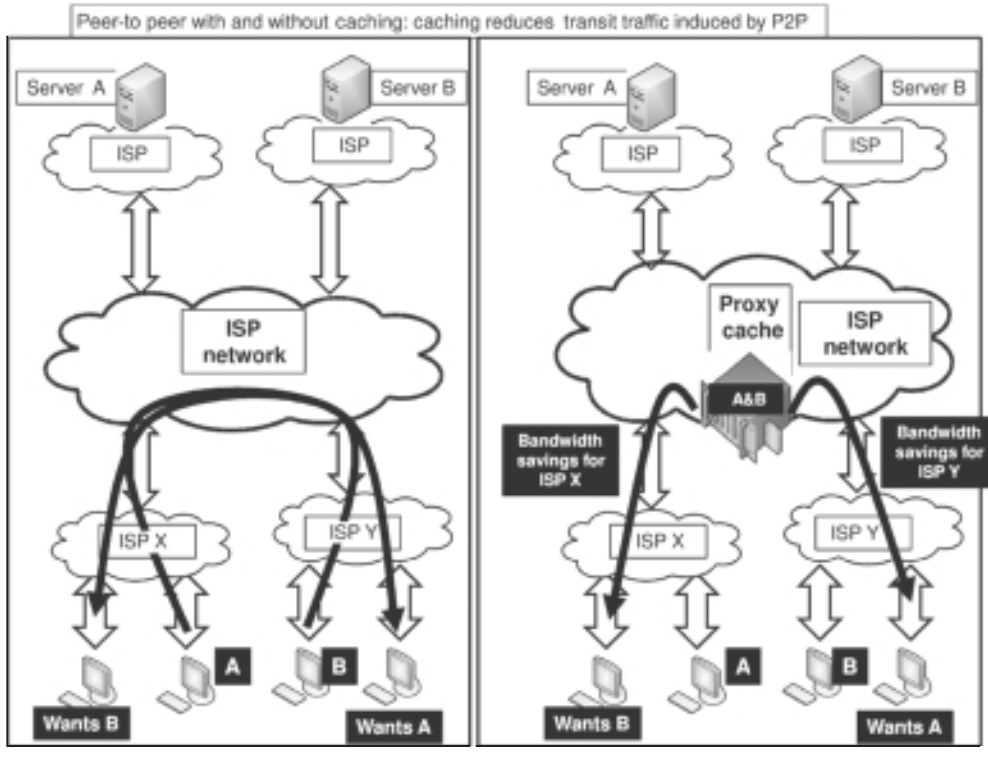


Figure 7.11 (a) P2P traffic pattern without proxy cache; (b) P2P traffic pattern with proxy cache.

There are two possible solutions to this problem: (i) identify the P2P traffic and throttle it explicitly and (ii) introduce caching to alleviate the impact of P2P traffic. The first solution hinges on the fact that the ISP (the provider of bandwidth and connectivity) treats traffic *differentially* based on the type of traffic and/or the source of traffic. This is the subject of an ongoing debate referred to as the *net neutrality* debate [39–41]. In general, this is not necessarily the politically correct thing to do. That leaves the ISPs with the second option, namely to introduce caching.

Proxy caches need to be deployed by the ISPs at their PoP. Note that to benefit all ISPs without thwarting P2P networking, proxy caches need to be deployed by ISPs at every level of the hierarchy, starting with the lowest level ISPs (tier-3 or tier-4) all the way up to the highest level ISPs (tier-2 and tier-1) as shown in Figure 7.12.

The role of a proxy cache would be to cache all transit P2P traffic and effectively serve as an aggregate proxy peer meaning that it can serve any P2P content that has been retrieved by the peers located in all the downstream ISPs.

As an illustration, in Figure 7.11(a), note that a client in ISP X has content from server A and a client in ISP Y has content from server B. If another end system in ISP X wants content from server B, the content will be served from the peer in ISP Y with the desired content resulting in traffic traversing the uplinks of both ISP X and that of ISP Y. However, the proxy cache in the upstream ISP would have cached contents from both A and B (when content from server

Figure 7.12 Proxy cache deployment at multiple tiers of ISP.

A was downloaded by the end system in ISP X and the content from server B was downloaded by the end system in ISP Y). Therefore, the request for content B in ISP X can be served by the proxy cache and hence the traffic on the uplink of ISP Y will be alleviated (Figure 7.11(b)). Similarly, when the end system in ISP Y requests content B and the content is served off the proxy cache, the traffic on ISP X's uplink will be alleviated. Stretching the scenario a little more, if a proxy cache is deployed in ISP Y, it will cache both A and B (Figure 7.13) if an end-system in ISP Y had requested the contents earlier. Then if another end system requests content A in ISP Y, the content will be served off the proxy cache in ISP Y and alleviate traffic in the link upstream of ISP Y.

To summarize the above discussion, the best way for ISPs to alleviate the impact of P2P networks would be to deploy proxy caches that would cache the P2P traffic and effectively act as peers, thereby offloading the uplink trunks. This will avoid the controversial alternative of selectively identifying P2P traffic and throttling it.

7.3 Summary

Movie-on-demand (or in general, video-on-demand) would be the most popular service expected from video-over-Internet service providers. The goal of this chapter was to explore alternative architectures both from technology and business perspectives for providing such a service. As a first step in that direction, we first computed the resource requirements (storage and bandwidth) for a service provider to be able to serve on demand any movie made in the US in the last 50 years. Then three fundamentally different architectures were compared: CDN, hosting and P2P. The benefits of CDN over hosting are avoidance of bottlenecks in the network, increased availability of content and better user experience in terms of faster

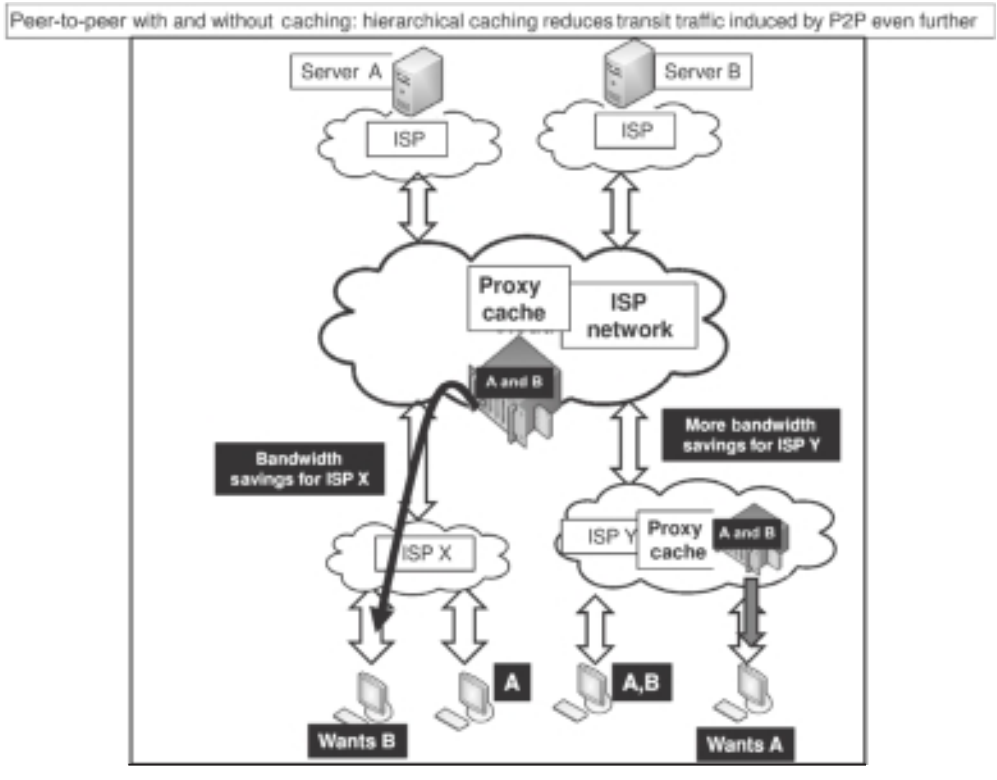


Figure 7.13 P2P network with and without caching.

download and lower response time. Peer-to-peer scales far better than CDN and has a lower cost of operation because the resources (storage and bandwidth) already paid for by the end users are leveraged in this model. Although P2P is economically the best option, there are challenges, such as security, reliability, quality of service, and digital rights management for copyrighted content that are not properly addressed in P2P model. The industry has been trying a combined CDN-P2P model to obtain the best of both worlds. In order to eliminate the confusion between caching and CDN, several examples were given to show how caching works for ISPs and how CDNs work for content providers. Furthermore, it was shown how even P2P networks would benefit from the use of caches in the network.

References

[1] Akamai: http://www.akamai.com. (accessed June 9, 2010).
[2] Akamai Technologies: http://en.wikipedia.org/wiki/Akamai_Technologies. (accessed June 9, 2010).
[3] Dilley, J., Maggs, B., Parikh, J. *et al.* Globally Distributed Content Delivery. http://www.akamai.com/dl/technical_publications/GloballyDistributedContentDelivery.pdf. (accessed June 9, 2010).
[4] Limelight Networks: http://www.limelightnetworks.com/. (accessed June 9, 2010).
[5] Edgecast: http://www.edgecast.com/. (accessed June 9, 2010).
[6] Buyya, R., Pathan, M. and Vakali, A. (eds) (2008) *Content Delivery Networks*, Springer, XVI, 418 p. 120 illus., Hardcover, ISBN: 978-3-540-77886-8.

[7] Content Delivery and Distribution Services. http://www.web-caching.com/cdns.html. (accessed June 9, 2010).
[8] Content Delivery Network (CDN) Research Directory. http://ww2.cs.mu.oz.au/~apathan/CDNs.html. (accessed June 9, 2010).
[9] A Taxonomy and Survey of Content Delivery Networks. http://www.gridbus.org/reports/CDN-Taxonomy.pdf. (accessed June 9, 2010).
[10] Caching Tutorial. http://www.mnot.net/cache_docs/. (accessed June 9, 2010).
[11] Wessels, D. (2001) Web Caching. June, O'Reilly 1-56592-536-X, Order Number: 536×318 pages. http://oreilly.com/catalog/webcaching/chapter/ch05.html. (accessed June 9, 2010).
[12] How Caching Works. http://www.enfoldsystems.com/software/proxy/docs/4.0/cachetutorial.html. (accessed June 9, 2010).
[13] Web Cache Communication Protocol v2. http://www.ciscosystems.com/en/US/docs/ios/12_0t/12_0t3/feature/guide/wccp.html. (accessed June 9, 2010).
[14] Peer to Peer File Sharing – P2P Networking. http://compnetworking.about.com/od/p2ppeertopeer/Peer_to_Peer_File_Sharing_P2P_Networking.htm. (accessed June 9, 2010).
[15] Mitchell, B. P2P Networking and P2P Software: Introduction to Peer to Peer (P2P) Networks and Software Systems. About.com Guide. http://compnetworking.about.com/od/p2ppeertopeer/a/p2pintroduction.htm. (accessed June 9, 2010).
[16] BitTorrent: http://www.bittorrent.com. (accessed June 9, 2010).
[17] Cohen, B. (2003) Incentives Build Robustness in BitTorrent. www.bitconjurer.org/, BitTorrent, May. (accessed June 9, 2010).
[18] Adar, E. and Huberman, B. (2000) Free riding on Gnutella, Technical report, Xerox PARC, 10, Aug. (accessed June 9, 2010).
[19] Backx, P., Wauters, T., Dhoedt, B. and Demeester, P. (2002) A comparison of peer-to-peer architectures, Eurescom Summit 2002, Heidelberg, Germany.
[20] Leibowitz, N., Ripeanu, M. and Wierzbicki, A. (2003) *Deconstructing the Kazaa Network*. 3rd IEEE Workshop on Internet Applications (WIAPP'03), June 23–24, 2003, San Jose, CA.
[21] Bhagwan, R., Savage, S. and Voelker, G.M. (2003) *Understanding Availability*. Int. Workshop on Peer to Peer Systems, Berkeley, CA, Feb. 2003.
[22] Saroiu, S., Gummadi, P.K. and Gribble, S.D. (2002) A Measurement Study of Peer-to-Peer File Sharing Systems, Multimedia Computing and Networking 2002 (MMCN '02).
[23] Gummadi, K.P. et al. (2003) Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. 19-th ACM Symposium on Operating Systems Principles (SOSP'03), Oct. 2003.
[24] Karagiannis, T., Broido, A., Brownlee, N. *et al.* (2003) File-sharing in the Internet: A characterization of P2P traffic in the backbone. Technical Report, UC Riverside.
[25] Chu, J., Labonte, K. and Levine, B. (2002) Availability and locality measurements of peer-to-peer file systems. ITCom: Scalability and Traffic Control in IP Networks, July 2002.
[26] Sen, S. and Wang, J. (2004) Analyzing peer-to-peer traffic across large networks. *ACM/IEEE Transactions on Networking*, **12**(2), 137–150.
[27] Oh-ishi, T., Sakai, K., Iwata, T. and Kurokawa, A. (2003) The Deployment of Cache Servers in P2P Networks for Improved Performance in Content-Delivery. Third International Conference on Peer-to-Peer Computing (P2P'03), Sep. 2003.
[28] Savvis, Inc. – Savvis.Net: http://www.savvis.net/en-US/Pages/Home.aspx. (accessed June 9, 2010).
[29] Cable and Wireless Hosting Solutions. www.cw.com. (accessed June 9, 2010).
[30] AT&T Hosting and Application Services. http://www.business.att.com/enterprise/Portfolio/application-hosting-enterprise/. (accessed June 9, 2010).
[31] Napster. http://www.napster.com/. (accessed June 9, 2010).
[32] Napster. http://en.wikipedia.org/wiki/Napster. (accessed June 9, 2010).
[33] Distributed Hash Tables. http://p2pfoundation.net/Distributed_Hash_Table. (accessed June 9, 2010).
[34] Chord. http://pdos.csail.mit.edu/chord/. (accessed June 9, 2010).
[35] Manku, G.S. Routing Networks for Distributed Hash Tables. http://infolab.stanford.edu/~manku//papers/03podc-dht.pdf. (accessed June 9, 2010).
[36] Xie, M. P2P Systems Based on Distributed Hash Table. Research Report, University of Ottawa. http://wpage.unina.it/cotroneo/dwnd/P2P/P2P_DHT.pdf. (accessed June 9, 2010).

[37] Rowstron, A. and Druschel, P. (2001) *Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems*. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, November, 2001, pp. 329–350.
[38] Resilient Overlay Network (RON). http://nms.csail.mit.edu/ron/. (accessed June 9, 2010).
[39] Net Neutrality. http://googlepublicpolicy.blogspot.com/search/label/Net%20Neutrality. (accessed June 9, 2010).
[40] Say no to Net Neutrality. http://www.reasonforliberty.com/reason/say-no-to-net-neutrality.html. (accessed June 9, 2010).
[41] Net Neutrality: A Complex Topic Made Simple. http://www.mynews.in/News/Net_neutrality_A_complex_topic_made_simple_N38645.html. (accessed June 9, 2010).

8

Internet Television

Internet television is very different from broadcast television. Broadcast television, as described in the previous section, consists of professionally produced content that is usually distributed over the air or via cable networks or via direct-to-home (DTH) satellite networks. In sharp contrast, the content in Internet television can be thought of as *Web content* in that it can be produced by anybody. As a result, some of the content would be professional (just as some web sites have professionally produced content) and some would be amateurish (just as some web sites have user-generated content). The first commercial incarnations of Internet television service providers have been those that distribute specialized content to certain special interest groups. For example, some service providers, such as Narrowstep [27] provide channels targeted to cyclists or to rock climbers and some others, such as World Wide Internet TV [28], leverage the pent-up demand of immigrants in foreign lands to consume content from their country and provide TV channels on the Internet from around the world. However, providing television-quality content to end users over the Internet is still a challenge and several technical issues need to be resolved to fill the gap between expectation and reality. Once these challenges are overcome, Internet television will become as normal as Web hosting and there would be equivalents of YouTube that, instead of asking users to upload video clips on to a web site, will enable users to broadcast their personal video content, such as the video of a wedding or of a local football match.

8.1 Resource Estimation

Estimating resources for Internet television is challenging. As mentioned in the previous paragraph, the resource needs would evolve with the number of users broadcasting personal content over the Internet. Let's assume that there would be 1.5 million users/month that would be generating content for Internet television. However, these contents would not fill the entire day for a given channel of Internet television. Suppose the duration of these user- generated videos is 30 minutes. Simple calculation would show that 48 (approximately 50) users will fill a day's worth of content for a channel and approximately 1500 users would fill a full month's worth of content for a channel. This implies that there would be content for 1000

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

channels generated per month. Once the content is generated, let us assume there would be an aggregator (service provider) who would pack the content into Internet television channels, just as a multi service operator (MSO) or DTH satellite service provider does today, and broadcast the content over the Internet. Thus that service provider would have a pipeline of content to fill up its channels on a continuous basis after the initial ramp-up phase.

8.1.1 Bandwidth

Assuming that the content would be encoded at 1 mbps and there would be 1000 channels that would be broadcast simultaneously to all end users, the service provider would require a bandwidth of 1 gbps. As mentioned earlier, as the technical challenges are overcome, more and more users will generate content and it will not be too long before the bandwidth requirement shoots up by a factor of 10 and then by a factor of 100 as Internet television explodes just as the Web did.

8.1.2 Storage

An aggregate bandwidth of 1–10 gbps translates to 332–3320TB/month assuming continuous transmission 24/7 for a month (30 days) at the aggregate rate. This implies that the service provider interested in distributing television content to the end users would require storage at the rate of 332–3320TB/month. For the sake of simplicity, let's use 0.3–3PB/month as the storage requirement number. However, the service provider may decide never to redistribute certain content after it is broadcast once and hence may not need to store such content. This will reduce the storage requirement from growing at the astounding rate of 0.3–3PB/month. In addition, the service provider may also choose to keep only select content (say 10% of the content) forever and discard the rest after N number of years. That will also put on a cap on the storage needs of the service provider.

Interestingly, new business models will emerge in which Internet television service providers may enable individual content providers to store their content in the storage infrastructure of the service provider for X number of years for $Y/month.

8.2 P2P Networks for Streaming

Peer-to-peer networks have traditionally been used for distributing files. Files are usually broken down into smaller pieces called chunks and these chunks are then distributed through a P2P network. Specifically, when a peer interested in a given file contacts what is called a *tracker* in a BitTorrent system [1], the *tracker* provides it with a list of peers to contact. The peer node then contacts the peers and collects information about who has what chunks. Then it uses an algorithm to decide what chunk to pull from which peer. This process continues until all the chunks belonging to a file are retrieved by the peer. It is to be noted that in the context of file retrieval, it does not matter in which order the chunks are retrieved by the peer. For example, it is conceivable that the *first* chunk retrieved by the peer is sequentially the *last* chunk of the file. The goal is to retrieve all the chunks, which can then be assembled to constitute the entire file.

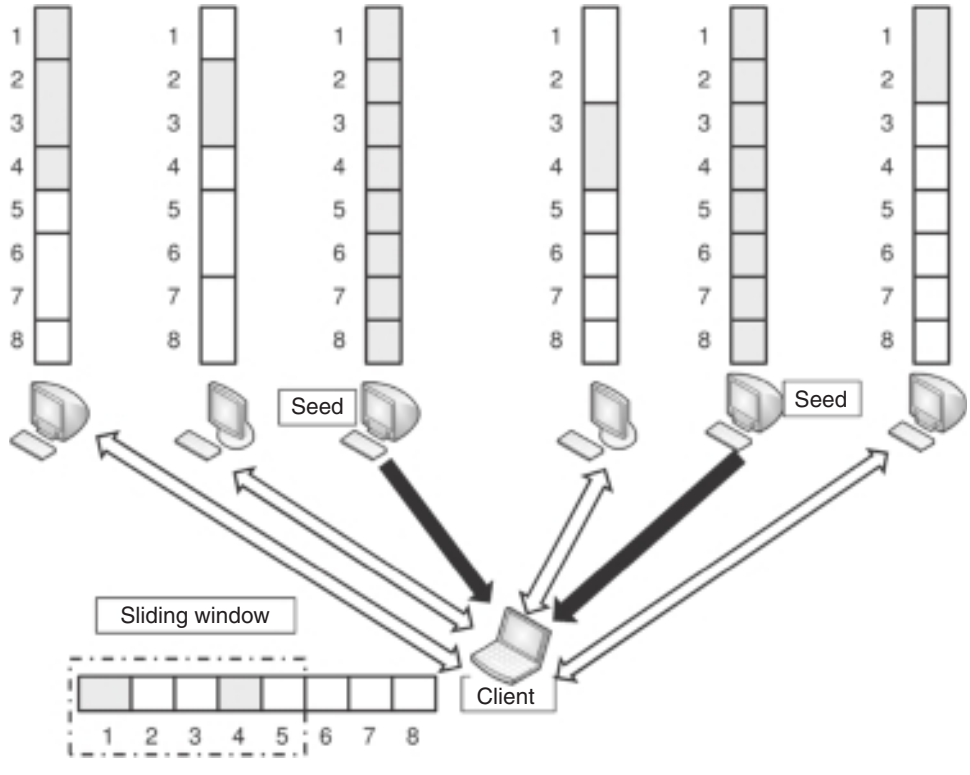


Figure 8.1 Sliding window for P2P streaming.

In the context of *streaming*, the situation is very different. It does matter in which *order* the chunks are retrieved because each chunk has a deadline for playback and if a chunk is retrieved after its playback time is over it would be of no use. Thus, it is unlikely that in case of on-demand P2P streaming (stored video as opposed to live video) a peer would retrieve the last chunk of the video file first. Moreover, in the context of *live* streaming, the chunks are generated on the fly and the peers involved in streaming live video have access to only a subset of chunks in a sliding window (window sliding with time as new video content is being generated) rather than the chunks in entire file (Figure 8.1).

The goal in P2P live streaming [7–14, 16–26] is to deliver live video to peers in a scalable manner with the optimized quality of experience while accommodating the heterogeneity and asymmetry of access link bandwidth and churn among participating peers in a session. Scalability with the number of participating peers in a session can be achieved by efficiently utilizing the contributed resources, namely the outgoing bandwidth, of the participating peers.

Having established the fundamental differences between P2P file download and P2P streaming, and the goal of P2P streaming, let us first introduce the notion of *adaptive P2P streaming* and then explore two different architectures for P2P streaming. These are tree-based P2P streaming and mesh-based P2P streaming.

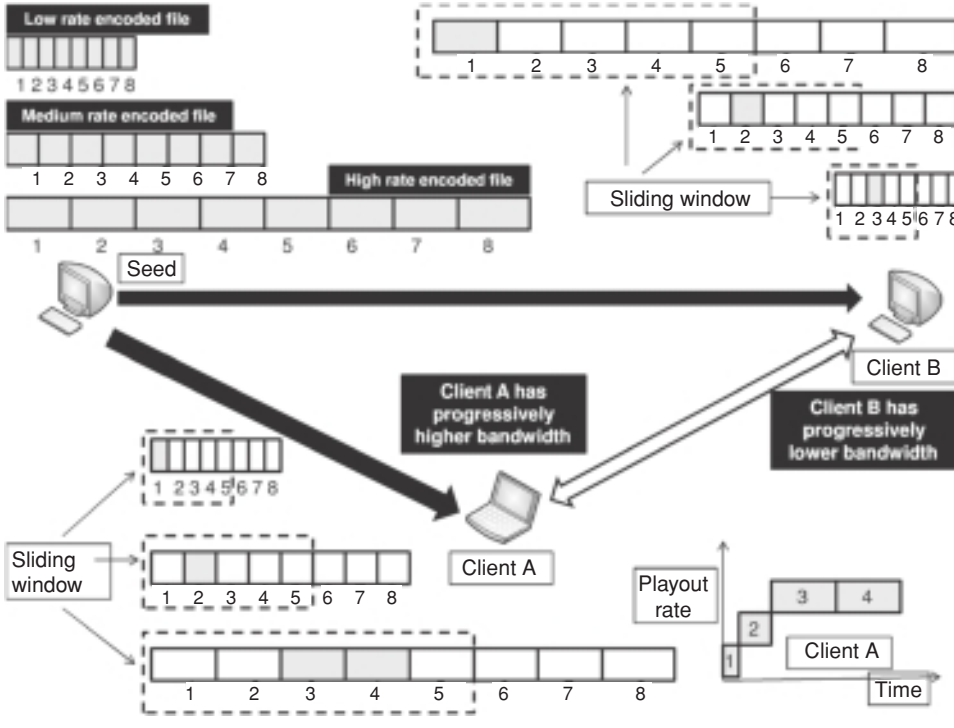


Figure 8.2 Dynamics of adaptive P2P streaming.

8.2.1 Adaptive P2P Streaming

Peer-to-peer streaming is more challenging than P2P file sharing when multiple bitrate encoded video is used for adapting the video streaming rate to the available bandwidth of the client [26]. As shown in Figure 8.2, client A starts off with low bandwidth and hence is served chunks of the video file from the low-rate encoded video file. The corresponding sliding window shows smaller chunks (corresponding to low bitrate video) being retrieved and stored. As the available bandwidth at client A increases, the corresponding sliding windows at client A start to retrieve and store bigger and bigger chunks, first for medium-rate encoded video and then for high-rate encoded video. The situation is reversed for client B, which starts with high available bandwidth and slowly moves to lower available bandwidth (Figure 8.2).

8.2.2 Tree-Based P2P Streaming

In tree-based P2P streaming, the peers are organized into multiple tree-based overlays, such that each peer is an internal node in one of the trees and is a leaf node in the other trees. Once the trees are formed, a subset of chunks, referred to as a sub-stream, is pushed down each tree separately to the intended recipients. If the aggregate outgoing bandwidth of a node in the tree is same as the aggregate incoming bandwidth of the node, then tree-based P2P streaming works very well. However, if the aggregate outgoing bandwidth of a node is smaller than the

aggregate incoming bandwidth of the node, then the node will not be able to push the chunks down the tree as fast as they arrive at the node leading to what is referred to as "bandwidth" bottleneck. In fact, in tree-based streaming, the rate of content delivery through a tree to a peer is limited by the minimum throughput of the upstream connections.

Given that the quality of streaming is dependent on the bandwidth of the upstream connections in a tree, the construction of the tree is very important. Here's an algorithm for the construction of tree in a system called End System Multicast (or ESM) [15, 29].

8.2.2.1 Single Tree Construction Algorithm

1. When a node A wants to join a tree, it needs to decide whom to choose as its parent. In order to do that, it probes a random subset of nodes, biased towards those not probed before.
2. Each node B responds with:
   a. Throughput it's currently receiving in the tree.
   b. Delay from source of the tree.
   c. Degree-saturated or not.
   d. Whether it is a descendant of A.
3. Time elapsed between sending the probe and receiving the response gives the round-trip time between A and B.
   A waits for a duration T (typically T = 1 s) to maximize the number of replies.
4. Based on the received replies, A eliminates those:
   a. That are descendants of A.
   b. That have their out-degree saturated.
5. For each node B that is not eliminated, A evaluates throughput and delay it expects to receive if B were chosen as parent. Here are the steps:
   a. Expected throughput = min (B's throughput, Average bandwidth of path AB).
   b. Choose the node with the highest expected throughput.
   c. If bandwidth to B is not known, A picks parent based on delay.

Thus a new node A picks as parent a node B if either:

1. the estimated application throughput is high enough for A to receive a higher quality stream; or
2. if B maintains the same bandwidth as A's current parent but improves throughput.

While the above algorithm describes how a single tree can be constructed in a P2P network, in general a node joins multiple such trees for resiliency and better quality of service. For example, in Figure 8.3, note that peer 5 joins two trees.

There are several algorithms in the literature to accomplish the above task. However, the one described below is one of the best known [27].

8.2.2.2 Multiple Tree Construction Algorithm

1. When a node A wants to join multiple trees, it contacts the bootstrapping node and indicates the number (N) of trees it wants to join.
2. The bootstrapping node, in trying to balance the number of internal nodes in the trees:

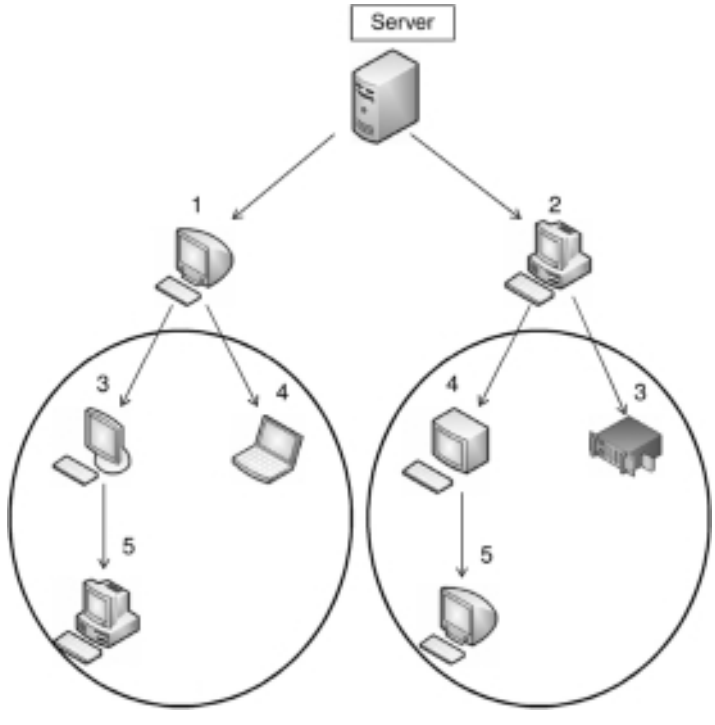


Figure 8.3 Same node (peer) joins multiple trees (trees are shown within the circles).

   a. Places the new node A as an internal node in that tree that has the minimum number of internal nodes.
   b. Does a breadth-first search in the chosen tree, and picks the node with the lowest depth as the parent of node A. This is done to maintain minimum depth of each tree.
3. When an internal node departs, each one of the child nodes and the subtrees rooted at them are partitioned and hence need to be reattached:
   a. Each node in the respective sub-trees wait for a time-out period to allow their respective root nodes (child nodes of the departed node) to rejoin the tree. If that happens, reattachment of the partitioned nodes is accomplished. If not,
   b. Each node of a subtree independently tries to rejoin the tree in the same position (either as an internal node or as a leaf node).

There are several tree-based P2P streaming techniques that have been used for multicasting live content over the Internet. One such technique is CoolStreaming [3, 4], which extends the swarming technique of BitTorrent [1] for distributing live video. Essentially, the peers in CoolStreaming do exactly what the peers in BitTorrent would do except that due to the *real-time* nature of the video content, the peers cannot afford to pull segments (chunks) of the live video feed in any order. Specifically, the peers maintain what is called a sliding window such that the segments within the sliding window are pulled before segments outside of the

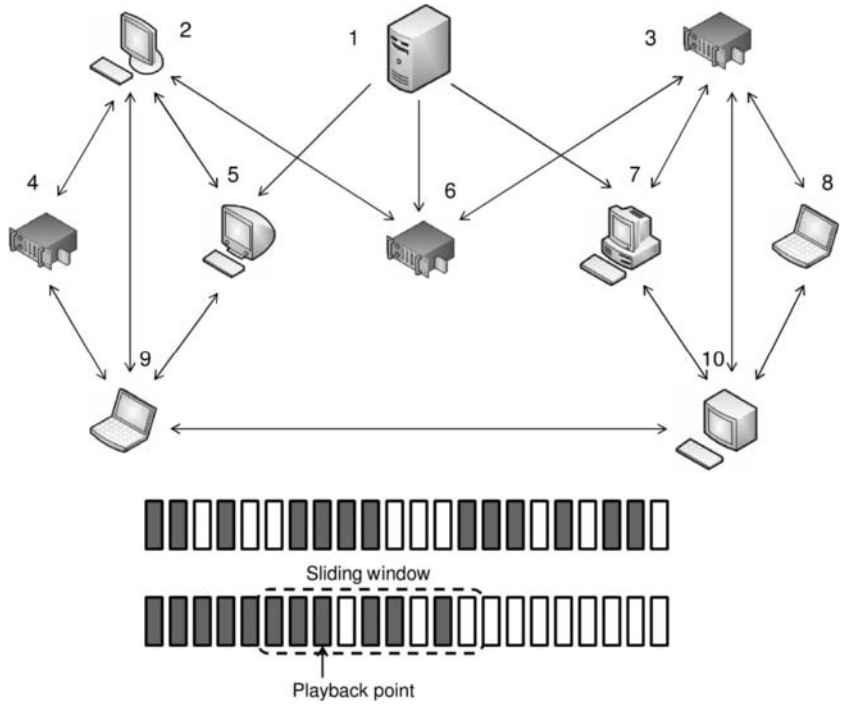


Figure 8.4 Peer 6 is simultaneously fetching content from peers 1, 2 and 3. Segment map of BitTorrent peer followed by segment map of CoolStream peer showing sliding window and playback point.

sliding window. This is to ensure that the segments with earlier playback time are retrieved before the segments with later playback time. The system can afford to wait a little longer for the retrieval of segments with later playback times compared to the ones with earlier playback times. In addition, *all* the segments with playback times before the current *system playback time* need to be pulled in order for the video to be played with the best possible quality of experience. The above description is illustrated in Figure 8.4, in which peer node 6 receives content from nodes 1, 2 and 3. However, it maintains a sliding window of segments as shown in the diagram (lower part) in which the segments outside the sliding window are not fetched. The shaded segments indicate that they have been fetched and the blank ones indicate that they have not yet been fetched. This is in contrast to the segment map of a regular BitTorrent peer shown in the diagram right above the sliding window based segment map where the segments are retrieved in any order. Typical size of sliding window for P2P video streaming is 120 segments (2 min worth of video) and each segment is usually 1 s long.

8.2.3 Mesh-Based P2P Streaming

In mesh-based P2P streaming, the peers are organized in a randomized mesh such that they utilize their bandwidth in the optimal way to deliver the highest quality of video content possible

given the topology and the bandwidth. PRIME P2P streaming described in [22] showed how to think about mesh-based streaming in a systematic way. Specifically, mesh-based streaming can be decomposed into two phases: (i) diffusion and (ii) swarming.

The goal of the diffusion phase is very quickly to distribute the chunks generated at the source to the peers collectively. A few terms need to be defined to avoid confusion in the description that follows. A *segment* is defined as the video content generated at the source every Δ seconds. Assuming that all peers have the same bandwidth b, each can *pull* at most $b * \Delta$ units of data from its parent. Let's refer to $b * \Delta$ as a *data unit*. In general, a *segment* will consist of a number of *data units* and each *data unit* will consist of a number of *chunks* and each chunk will consist of a number of *packets*.

Diffusion phase is similar to tree-based streaming in that the peers are arranged in a tree except that this tree is not formed explicitly as in the case of tree-based streaming rather it is formed implicitly based on the scheduling algorithm run at each peer to decide on what chunks to pull from which peers. Conceptually, the source (which is assumed to be at *level 0*) will have some children at *level 1* and each child at level 1 will have some children at *level 2* and so on until *level d* where $d =$ depth of the tree that covers all the peers.

Every Δ seconds, a new segment is generated at the source, and the data units belonging to the segment are collectively distributed to the peers at level 1. For simplicity, assume that a segment consists of six data units, D_1 through D_6, and also assume that the source has three children (P_1, P_2 and P_3) in level 1. Ideally each of the three peers at level 1 would pull 2 data units each. Say P_1 pulls D_1 and D_2, P_2 pulls D_2 and D_3 and P_3 pulls D_5 and D_6 at time $t = \Delta$ assuming that the segment was generated at the source at time t = 0. Each peer is assumed to have the same bandwidth b, so all peers at level n belonging to the subtree rooted at P_1 will pull D_1 and D_2 at time $t = n^* \Delta$. Similarly, all peers at level n belonging to the subtree rooted at P_2 will pull D_3 and D_4 at time $t = n^* \Delta$ and all peers at level n belonging to the subtree rooted at P_3 will pull D_5 and D_6 at time $t = n^* \Delta$. Thus at $t = depth^* \Delta$, the segment would be collectively disseminated among the peers. This phase is referred to as the *diffusion* phase because the segment is quickly disseminated among the peers in a collective way. This is shown diagrammatically in Figure 8.5.

After the diffusion phase comes the swarming phase. Note that, at the end of diffusion phase, peers in each subtree rooted at the *level 1 peers* have different data units. For example, in the above example, all peers belonging to the subtree rooted at P_1 would have data units D_1 and D_2, all peers belonging to the subtree rooted at P_2 would have data units D_3 and D4, and all peers belonging to the subtree rooted at P_3 would have data units D_5 and D_6 at time $t = depth^*\Delta$. Right after that, in order to fetch data units that a peer does not have, it generates a request for the missing data units from the peers in other subtrees. In this example, peers in subtree rooted at P_1 would generate requests for data units D_3, D_4, D_5 and D_6; peers in subtree rooted at P_2 would generate requests for data units D_1, D_2, D_5 and D_6; and peers in subtree rooted at P_3 would generate requests for data units D_1, D_2, D_3 and D_4.This is exactly like the swarming used in BitTorrent, where peers pull missing chunks from other peers and simultaneously multiple peers gather all the chunks belonging to the desired file. Figure 8.6 shows how the data units are exchanged among peers at time $t = 3\Delta + \delta$.

Peers pull missing data units from other peers at time $t = 3\Delta + 2\delta$ as shown in Figure 8.7.

The diagram combining diffusion and swarming is shown in Figure 8.8.

Table 8.1 shows the progression of data unit acquisition in an alternative format.

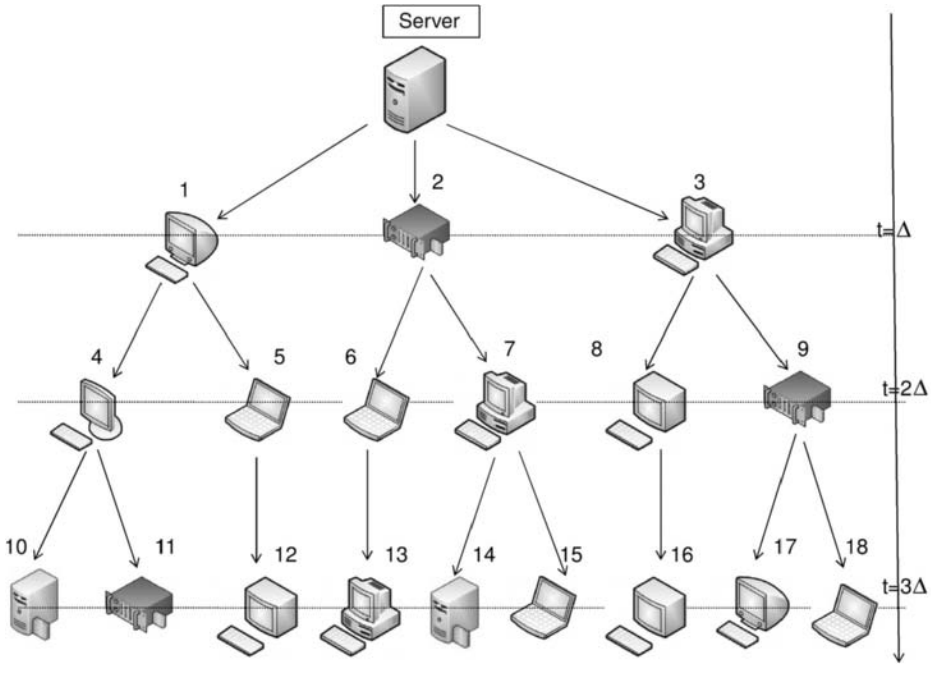


Figure 8.5 Diffusion phase of P2P streaming.

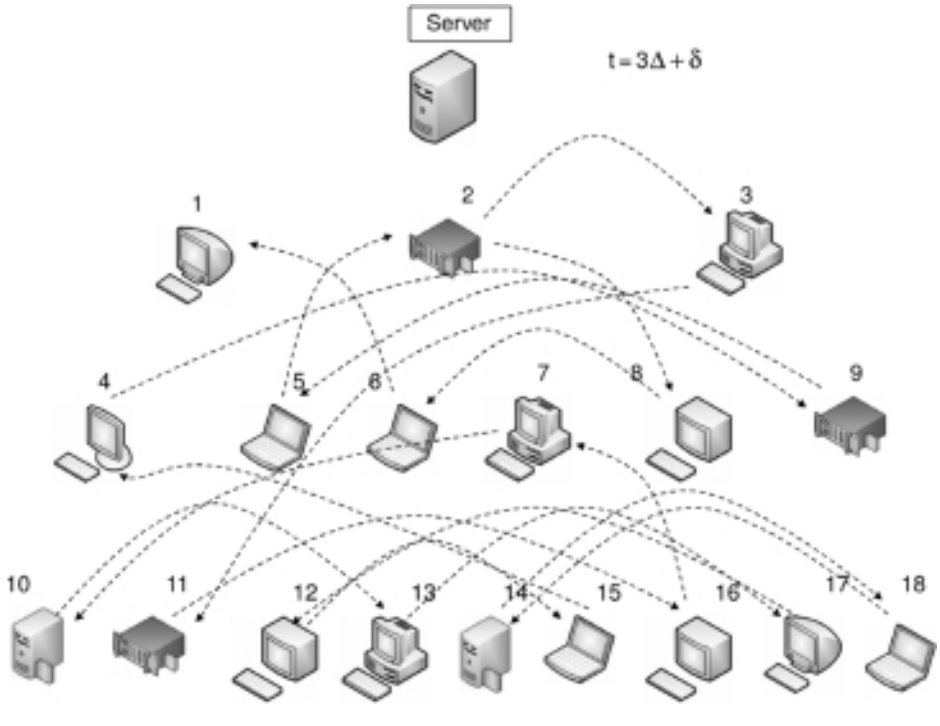


Figure 8.6 Swarming phase of P2P streaming (time $t = 3\Delta + \delta$).

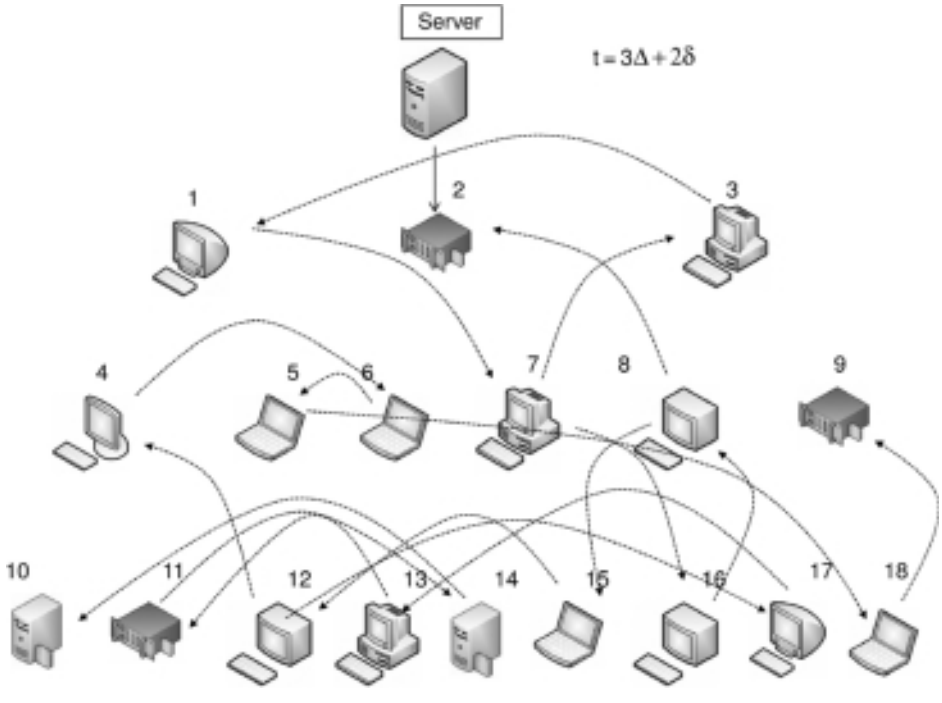


Figure 8.7 Swarming phase of P2P streaming (time $t = 3$ delta $+ 2$ delta).

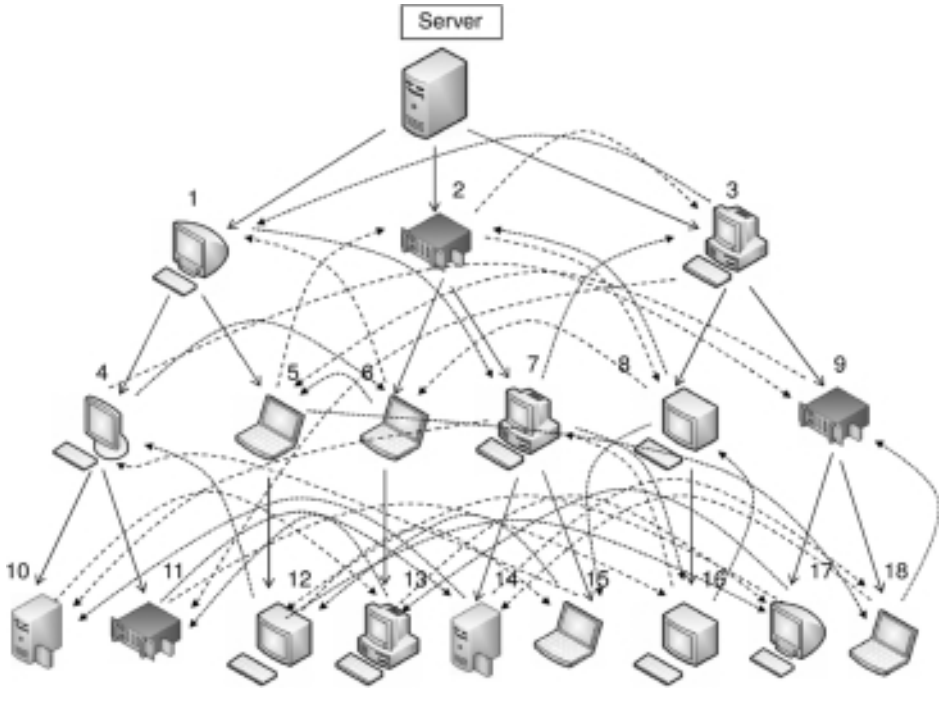


Figure 8.8 Combined diffusion and swarming in P2P streaming.

Table 8.1 Progression of data unit acquisition.

	$T = \Delta$	$T = 2\Delta$	$T = 3\Delta$	$T = 3\Delta + \delta$	$T = 3\Delta + 2\delta$
P_1	D_1, D_2	D_1, D_2	D_1, D_2	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_2	D_3, D_4	D_3, D_4	D_3, D_4	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_3	D_5, D_6	D_5, D_6	D_5, D_6	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_4		D_1, D_2	D_1, D_2	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_5		D_1, D_2	D_1, D_2	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_6		D_3, D_4	D_3, D_4	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_7		D_3, D_4	D_3, D_4	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_8		D_5, D_6	D_5, D_6	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_9		D_5, D_6	D_5, D_6	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{10}			D_1, D_2	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{11}			D_1, D_2	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{12}			D_1, D_2	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{13}			D_3, D_4	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{14}			D_3, D_4	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{15}			D_3, D_4	D_1, D_2, D_3, D_4	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{16}			D_5, D_6	D_1, D_2, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{17}			D_5, D_6	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$
P_{18}			D_5, D_6	D_3, D_4, D_5, D_6	$D_1, D_2, D_3, D_4, D_5, D_6$

8.2.4 Scalability of P2P Networks

Peer-to-peer networking was a topic of research in academia until it started to become mainstream. Large-scale deployment of P2P networks became real in early 2000, and people from Asia and Africa started retrieving their favorite music and video from people in North America and vice versa. There are two main reasons behind the wild success of P2P networks. First, content distribution and retrieval using P2P networks doesn't cost any money. While this statement is not strictly correct, because the cost is actually borne by the end users (peers), the peers don't pay anything extra for P2P networks on top of what they already pay for connecting to the Internet. The other resource that the peers contribute to the P2P networks is the storage in their PC, which they have paid for anyway. In short, there is no "additional" expense incurred by the peers to gain the benefits of P2P networks. The second reason behind the success of P2P networks is the tremendous scalability of the system. Specifically, as new peers join an existing P2P network, the capacity of the P2P network increases exponentially.

Here is a simple illustration to explain scalability of P2P networks. In Figure 8.9, we assume a P2P network that distributes media files by breaking them into pieces which area referred to as chunks. In general, a file is broken into C chunks. In the illustration, $C = 2$. In addition, each peer feeds N peers. In the illustration, $N = 2$. Further, each peer is assumed to have an outgoing bandwidth of B.

The server divides a media file of size S into two chunks and sends the first chunk to peer 1 and the second chunk to peer 2 and splits its outgoing bandwidth between the two peers. Thus peers 1 and 7 will respectively have chunks 1 and 2 at the same time. Let's refer to that time instant as $t = 0$. Then peer 1 distributes chunk 1 to its peers 2 and 3 and peer 7 distributes chunk 2 to its peers 6 and 5. Note that it will take [(S/C)/(B/2)] time units for the chunks to reach the respective destinations. Therefore, peers 2 and 3 will have chunk 1 and peers 6 and 5

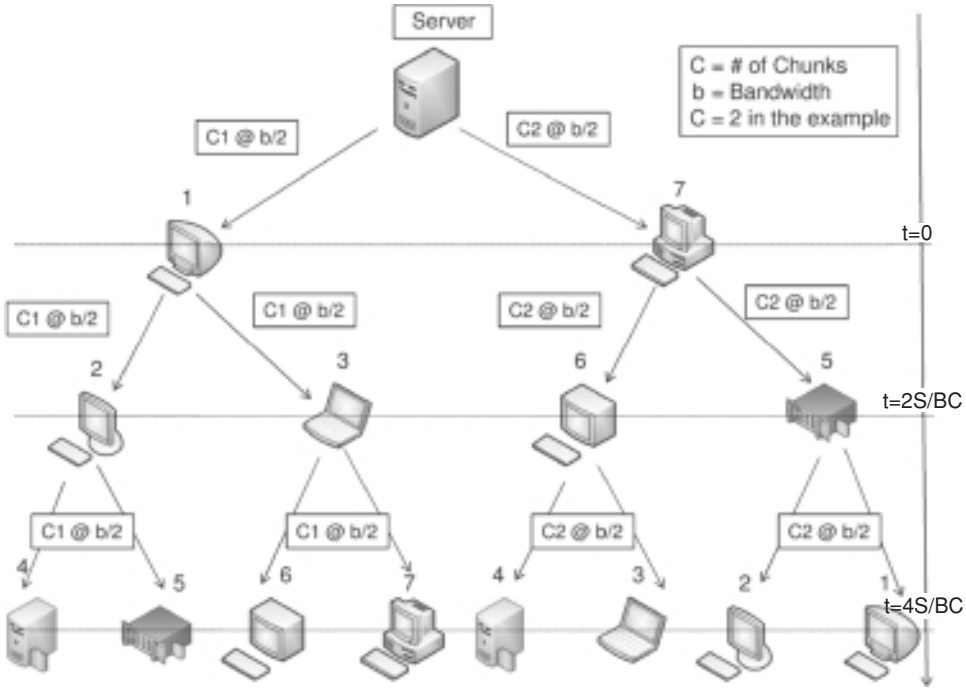


Figure 8.9 Scalability calculation of P2P streaming.

will have chunk 2 at time $t = 2$S/BC. The snapshot of which peer has what chunk is shown in Table 8.2 in column $T = 2$S/BC. Then peers 2 and 3 distribute chunk 1 to peers 4, 5, 6 and 7 while peers 6 and 5 distribute chunk 2 to peers 4, 3, 2 and 1. This process also takes the same time 2S/BC and hence at time $t = 4$S/BC, all the peers (1 through 8) have both chunks 1 and 2 and the media file distribution is complete.

It is to be noted that it takes log (N) time to distribute a file to N peers. This also implies that the capacity of P2P networks increases exponentially with the number of peers.

Technically, for N peers each with an out-degree of k (meaning that each peer feeds into k other peers), the time T (units) to distribute a file to all peers by dividing it into C chunks, can

Table 8.2 Snapshot of chunks with peers.

	$T = 0$	$T = 2$S/BC	$T = 4$S/BC
1	C1	C1	C1, C2
2		C1	C1, C2
3		C1	C1, C2
4			C1, C2
5		C2	C1, C2
6		C2	C1, C2
7	C2	C2	C1, C2

be expressed as:

$$T \sim [1 + (\log_k N) * (k/C)]$$

where the time to download to one peer is 1 unit.

Thus, for $k = 10$, $N = 10\,000$ and $C = 20$, $T = 3$ units.

Equivalently, the total number of peers (N) that complete upload of a file in t units (assuming time to download to one peer in 1 unit) given that each peer has an out-degree of k and each file is divided into C chunks can be expressed as:

$$N \sim k^{[(t-1)*(C/k)]}$$

Thus, for $k = 10$, $t = 3$ and $C = 20$, $N = 10\,000$.

8.2.5 *Comparison of Tree-Based and Mesh-Based P2P Streaming*

There are several similarities and differences between tree-based and mesh-based P2P streaming (see Tables 8.3 and 8.4). First we look at the ways they are similar and then we try to understand where they differ.

Table 8.3 Similarities between tree-based and mesh-based P2P streaming.

	Tree-based P2P streaming	Mesh-based P2P streaming
Similarities		
Structure of Overlay	Uses multiple diverse trees	Random mesh is also a collection multiple diverse trees
Content received by Peers	Peers receive different chunks from different trees	Peers receive different chunks from different trees
Role of Peers	Each peer receives chunks from multiple peers and distributes chunks to multiple peers	Each peer receives chunks from multiple peers and distributes chunks to multiple peers, especially in the *swarming* phase
Tree structure	The set of edges used for distributing a single packet from the source to all the peers forms a tree rooted at the source	The set of edges used for distributing a single packet from the source to all the peers forms a tree rooted at the source, especially in the *diffusion* phase
Buffering at each Peer	Each node maintains a loosely synchronized playout time behind the source's playout time and that requires each peer to maintain a buffer where packets can be received out of order as long as they are received before their corresponding playout time	Each peer maintains a buffer of size $\omega^*\Delta$ where Δ is the interval for the generation of new segments at the source and ω is a constant multiple to account for the loosely synchronized playout time. This buffer can receive packets in out of order fashion during the *swarming* phase

Table 8.4 Differences between tree-based and mesh-based P2P streaming.

	Tree-based P2P streaming	Mesh-based P2P streaming
Differences		
Formation of Delivery Tree for individual packets	The delivery path of all packets belonging to a particular description is fixed and it is determined by the *static* tree mapping of descriptions to trees	The delivery path of all packets belonging to a particular description is *dynamically* determined as the packets flow through the overlay and it is determined by the *swarming* algorithm
Quality of streaming	When the bandwidth of an upstream connection in the overlay tree is less than what is needed for streaming a particular description, the quality of experience for streaming downstream of the bottleneck connection suffers	When the bandwidth of an upstream connection in the overlay tree is less than what is needed for streaming a particular description, the downstream peers still receive the packets through alternative paths from other parents, thereby providing the best possible streaming quality of experience

8.3 Provider Portal for P2P (P4P)

Based on the above sections, it is quite convincing that P2P technology is the way to scale distribution of video on the Internet. However, we also argued that a hybrid CDN-P2P model is best for "controlled" distribution of content. The focus so far has been to look at what's best from the perspective of "content distributor". What about the Internet service providers (ISPs)? Briefly we looked at the interest of ISPs when we discussed the benefits of combining caching with P2P. In this section, we look at a relatively new development in IETF where the goal is to optimize the combined interest of P2P-based content providers and the ISPs. The work that started off as Provider Portal for P2P (P4P) is currently driven as an effort called application level traffic optimization or ALTO in IETF [30].

Given that P2P is becoming a key component of content delivery infrastructure for legal content through projects like iPlayer (BBC) [31], Joost [32], Pando (NBC Direct) [33], PPLive [5,6], Zattoo [34], BT (Linux) [35], Verizon P2P [36], the usage statistics for P2P legal content distribution is also soaring. For example, there are close to 20 million simultaneous users on average and 100 million licensed transactions/month.

8.3.1 Some Statistics of P2P Traffic

Figure 8.10 shows that 70% of the Internet traffic is due to P2P applications, out of which 65% is from video (Figure 8.11). Video traffic comes both from streaming as well as from download of video files using P2P.

While the popularity of P2P applications is on the rise, P2P traffic is largely network-oblivious and as a result, it is not network efficient. Here are some relatively recent (2008) statistics from Verizon, a leading global Internet Service Provider. On an average, a bit distributed through P2P applications traverses 1000 miles and 5.5 metro-hops on Verizon's

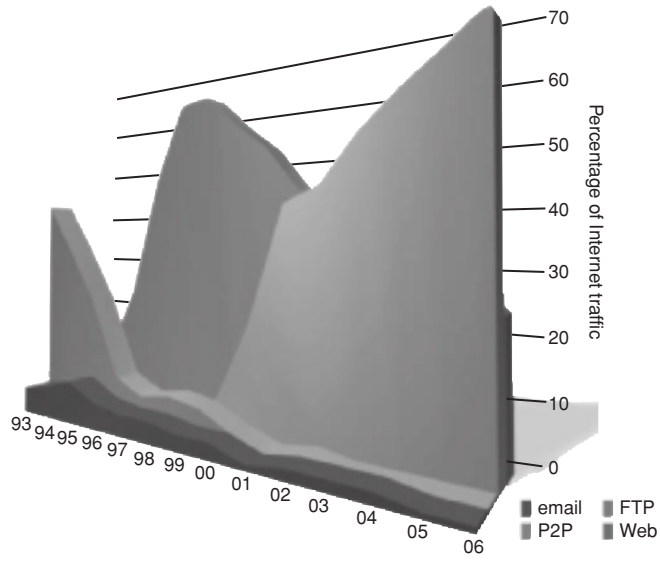


Figure 8.10 Internet traffic breakdown (1993–2006). [Courtesy: CacheLogic].

network! In addition, due to network-agnostic nature of P2P applications, 50–90% of locally resident (resident on university network) content is retrieved externally (outside the university network) by P2P applications [37]. Retrieving content externally means increased usage of the upstream bandwidth paid for by the ISP or a university campus, leading to higher operational expenses for the ISP/university.

8.3.2 Alternative Techniques to Deal with P2P Traffic in ISPs Network

In order to deal with this mounting pressure of P2P traffic, ISPs have been trying some alternative strategies:

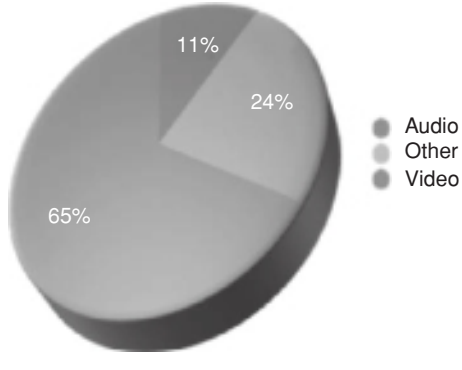


Figure 8.11 File types: major P2P networks (2006). [Courtesy: Velocix].

- upgrade infrastructure;
- charge based on usage;
- limit bandwidth and/or terminate P2P service;
- P2P caching.

Obviously, upgrading infrastructure is a capex-intensive proposition. Charging based on usage or limiting bandwidth usage for P2P are not necessarily best things to do given the net-neutrality controversy. Peer-to-peer caching is a partially viable solution but seems like a tactical as opposed to a strategic long-term solution. It is extremely difficult, if not impossible, for ISPs alone to effectively address network efficiency problems.

8.3.3 Adverse Interaction between ISP Traffic Engineering and P2P Optimization

From the perspective of P2P service providers, the goal is to improve the quality of experience by reducing download time. To that effect, some solutions have been proposed in the literature [37–39] where the P2P applications try to infer the network topology and adapt the traffic pattern accordingly. If the ISP did not do any traffic engineering, adaptive P2P solutions would perform reasonably well in terms of reducing average latency and decreasing maximum network utilization for ISPs. Similarly, if P2P solutions did not adapt traffic patterns, average latency would have decreased and the maximum network utilization would have reduced. However, when both P2P service providers adapt the traffic pattern and the ISPs do traffic engineering to shift traffic away from highly utilized links, the result is negative! Average latency goes up (even worse, fluctuates about a high mean) and so does maximum network utilization. This specific problem exposes a fundamental limitation of the Internet architecture, namely that of limited feedback to the network applications.

8.3.4 P4P Framework

In order to avoid optimizing in an uncoordinated manner, the P4P proposal is to design an open framework to enable better cooperation between network providers and network applications. The P4P framework consists of:

- data plane;
- management plane;
- control plane.

In the data plane, applications mark the importance of traffic while routers mark packets to provide faster, fine-grained feedbacks.

In the management plane, it is important to monitor compliance of application traffic to the contract set forth with the ISP about the traffic profile and the marking of traffic according to importance.

In the control plane, the application needs to have a mechanism to obtain information about the network. In fact, the service providers publish an API for the applications to obtain network

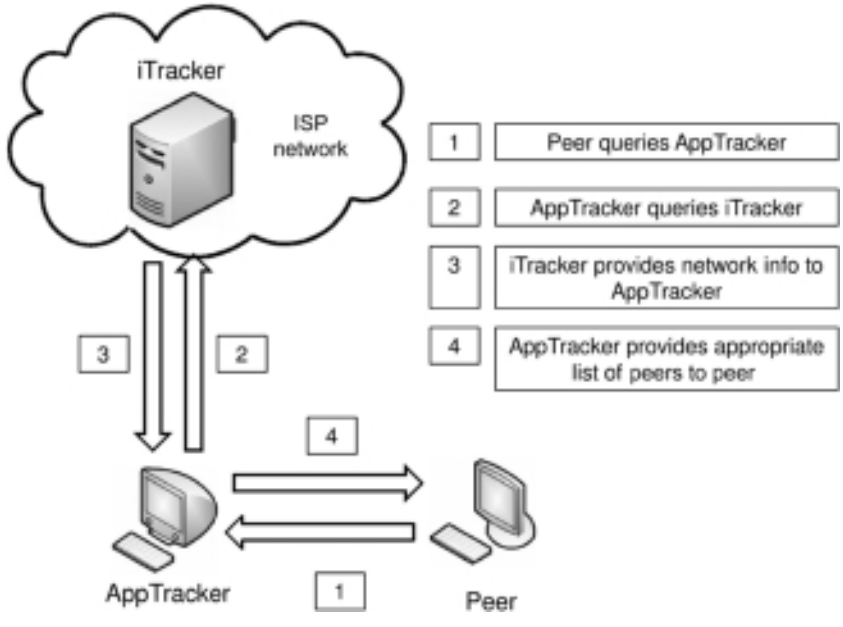


Figure 8.12 P4P Control Plane in the context of P2P traffic.

information. The applications query the interface and based on the retrieved information, adjust the traffic communications pattern.

An application of the control plane to optimize P2P traffic is shown in Figure 8.12. In the absence of this framework, when a peer contacts the AppTracker, the AppTracker randomly chooses a set of peers and returns the set to the peer. In sharp contrast, in the context of P4P framework, when a peer contacts the AppTracker, the AppTracker *intelligently* selects the list of peers based on the feedback it receives from the ISPs.

8.4 Summary

Internet television is what differentiates traditional cable/satellite television service from an Internet-based television service from the end-user perspective. In other words, video over Internet service providers would be able to source much more independent content compared to that done by cable/satellite service providers and be able to stream them as television shows just as traditional shows are televised in cable/satellite networks. The goal of this chapter was to explore alternative architectures, from both technology and business perspectives, for providing such a service. As a first step in that direction, we computed the resource requirements (storage and bandwidth) for a service provider to be able to serve independent content as television shows. Then three fundamentally different architectures were compared: P2P streaming, tree-based streaming and mesh-based streaming. The main benefit of tree-based and mesh-based streaming over one-to-one P2P streaming is scalability, the ability to stream in real-time the same content to multiple recipients with acceptable quality of user experience. Then the similarities and differences between tree-based and mesh-based streaming were discussed and

the superiority of mesh-based streaming was established. Finally, it was shown why P2P traffic is detrimental to ISPs' networks and how ISPs are trying to use traffic engineering techniques to reduce the impact of P2P traffic in their networks, only to see the problem becoming worse due to unwanted interaction between traffic engineering done by the ISPs and the P2P traffic optimization done by the P2P network. Next, we discussed a relatively new development in IETF, namely, provider portal for P2P (P4P). The P4P framework formalizes the interaction between P2P networks and service providers by enabling P2P networks to choose appropriate peers without adversely affecting traffic in service provider networks.

References

[1] BitTorrent Protocol, http://bittorrent.org/beps/bep_0003.html (accessed June 9, 2010).

[2] Cohen, B. (2003) Incentives Build Robustness in BitTorrent, In IPTPS, February.

[3] Zhang, X., Liu, J., Li, B., and Yum, T.-S.P. (2005) *DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming*. Proceedings of the IEEE INFOCOM'05, Miami, FL, USA, March 2005.

[4] Zhang, X., Liu, J. and Li, B. (2005) *On Large Scale Peer-to-Peer Live Video Distribution: CoolStreaming and Its Preliminary Experimental Results*. IEEE International Workshop on Multimedia Signal Processing (MMSP), October 2005.

[5] PPLive, http://www.pplive.com/en/about.html. (accessed June 9, 2010).

[6] Hei, X., Liang, C., Liang, J. *et al.* (2006) Insights into PPLive: A measurement study of a large-scale P2P IPTV system. Workshop on Internet Protocol TV (IPTV) services over World Wide Web in conjunction with WWW2006, May 2006.

[7] Liu, Z., Shen, Y., Ross, K.W. *et al.* (2008) *Substream Trading: Towards an Open P2P Live Streaming System*. Proceedings of IEEE International Conference on Network Protocols (ICNP), 2008.

[8] Huang, C., Li, J. and Ross, K.W. (2007) *Can Internet Video-on-Demand be Profitable?* Proceedings of ACM SIGCOMM, 2007.

[9] Cui, Y., Li, B., and Nahrstedt, K. (2004) *oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks*. IEEE Journal on Selected Areas in Communications, 2004.

[10] Hua Chu, Y., Chuang, J. and Zhang, H. (2004) *A Case for Taxation in Peerto- Peer Streaming Broadcast*. Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems, 2004.

[11] Zhang, M., Zhang, Q., Sun, L. and Yang, S. (2007) Understanding the power of pull-based streaming protocol: can we do better? *IEEE Journal on Selected Areas in Communications*, **25**.

[12] Tewari, S. and Kleinrock, L. (2006) *Proportional Replication in Peer-to-Peer Networks*. Proceedings of the IEEE INFOCOM 2006, April 2006.

[13] Qiu, D. and Srikant, R. (2004) *Modeling and Performance Analysis of BitTorrent-Like Peer-Peer Networks*. Proceedings of the SIGCOMM 2004, September 2004.

[14] Dana, C., Li, D., Harrison, D., and Chuah, C.N. (2005) *BASS: BitTorrent Assisted Streaming System for Video-on-Demand*. IEEE International Workshop on Multimedia Signal Processing (MMSP), October 2005.

[15] Chu, Y.H., Rao, S.G., Seshan, S. and Zhang, H. (2002) A case for end system multicast. *IEEE Journal on Selected Areas in Communication (JSAC)*, Special Issue on Networking Support for Multicast, **20**(8): 1456–1471.

[16] Magharei, N. and Rejaie, R. (2006) *Understanding Mesh-based Peer-to-Peer Streaming*. Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video, May 2006.

[17] Castro, M., Druschel, P., Kermarrec, A.-M. *et al.* (2003) *SplitStream: High-Bandwidth Multicast in a Cooperative Environment*. Proceedings of SOSP'03, October 2003.

[18] Padmanabhan, V.N., Wang, H.J. and Chou, P.A. (2003) *Resilient Peer-to-Peer Streaming*. Proceedings of IEEE ICNP, November 2003.

[19] Jin, X., Cheng, K.-L. and Chan, S.-H. (2006) *SIM: Scalable Island Multicast for Peer-to-Peer Media Streaming*. Proceedings of the IEEE International Conference on Multimedia Expo (ICME), July 2006.

[20] Li, J. (2004) PeerStreaming: A Practical Receiver-Driven Peer-to-Peer Media Streaming System. Microsoft Research TR-2004-101, Sept.

[21] Magharei, N. and Rejaie, R. (2009) ISP-friendly P2P streaming. *IEEE Multimedia Communications Technical Committee e-Letter*.
[22] Magharei, N. and Rejaie, R. (2009) PRIME: Peer-to-peer receiver-Driven MEsh-based Streaming. *IEEE/ACM Transactions on Networking*, **17**(4): 1415–1423.
[23] Incorporating Contribution-Awareness into Mesh-based Peer-to-Peer Streaming Services.
[24] Rejaie, R. (2006) Anyone can broadcast video over the internet. *Communications of the ACM*, special issue on Entertainment Networking, **49**(11), 55–57.
[25] Magharei, N. and Rejaie, R. (2006) Adaptive receiver-driven streaming from multiple senders. *ACM/SPIE Multimedia Systems Journal*, 11(6), 550–567. Springer-Verlag.
[26] Rejaie, R., Handley, M. and Estrin, D. (2000) Layered quality adaptation for internet video streaming. *IEEE Journal on Selected Areas of Communications (JSAC), Special issue on Internet QoS*, 18: 2530–2543.
[27] Narrowstep Inc. http://www.narrowstep.com/aboutus.aspx. (accessed June 9, 2010).
[28] World Wide Internet TV. http://wwitv.com/ (accessed June 9, 2010).
[29] End System Multicast (ESM): Streaming for the Masses. http://esm.cs.cmu.edu/ (accessed June 9, 2010).
[30] Application Level Traffic Optimization. http://www.ietf.org/dyn/wg/charter/alto-charter.html. (accessed June 9, 2010).
[31] BBC iPlayer. http://www.bbc.co.uk/iplayer/ (accessed June 9, 2010).
[32] Joost. http://www.joost.com/ (accessed June 9, 2010).
[33] Pando. http://www.pando.com/ (accessed June 9, 2010).
[34] Zattoo. http://www.zattoo.com/ (accessed June 9, 2010).
[35] BitTorrents for Ubuntu Linux. http://techcityinc.com/2009/01/23/top-5-bittorrent-clients-for-ubuntu-linux/ (accessed June 9, 2010).
[36] Verizon P2P Effort Supports Efficient File Sharing. http://www.crn.com/software/206903773;jsessionid=HIXKKH2OLWEBZQE1GHPSKH4ATMY32JVN. (accessed June 9, 2010).
[37] Karagiannis, T., Rodriguez, P. and Papagiannaki, D. (2005) *Should Internet Service Providers Fear Peer-Assisted Content Distribution?* ACM/USENIX Internet Measurement Conference, Oct. 2005
[38] Bindal, R., Cao, P., Chan, W. *et al.* (2006) *Improving Traffic Locality in BitTorrent via Biased Neighbor Selection*. Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, 2006.
[39] Choffnes, D.R. and Bustamante, F.E. (2008) *Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in P2P Systems*. Proceedings of the ACM SIGCOMM 2008, August 2008.

9

Broadcast Television over the Internet

Broadcast television for an open network refers to the distribution of television channels over the Internet. In order to avoid potential confusion in terminology, let us discuss different types of content and how they are currently distributed. First of all, the term "broadcast television" was coined because the television signals used to be "broadcast" over the air just like the radio signals and the television sets at home used to have antennas (just like in radios) to catch the television signals and display on the TV sets. There used to be a number of such channels. However, the television industry evolved with the introduction of cables. With the introduction of cables, it was possible for specialized content providers (other than those that broadcast television content over the air) to broadcast content using satellites and for the cable service providers to capture that content using a satellite dish receiver. Once that content is captured, the cable service providers (traditionally known as multi service operators or MSOs) started distributing them to households using their *cable infrastructure*. The MSOs started distributing not only the specialized content but also the traditional broadcast TV content and with a higher quality of experience compared to over-the-air broadcast television service. With the success of the cable industry in the residential markets, a new segment emerged – the digital satellite television or direct-to-home (also called DTH) television service providers. Direct-to-home providers started to distribute a range of TV channels using their *satellite infrastructure* and allowing residential users to receive the television signals *directly* as opposed to *indirectly* receiving them via the cable infrastructure of MSOs. Regardless of how the television signals are received, whether over the air, or via cable or via satellite, we refer to the content received by residential users as *broadcast television* content. In order to estimate the cost of distributing this content over the Internet we need to estimate the resources needed.

Note that there are their legal implications in distributing television content over the Internet, but we do not address such issues here. For example, to be able to distribute television content over the Internet [1,2,13–48] one has to store the content before distribution and such storing of content by anyone other than the end consumer is not permissible under the current regulations. In addition, the content providers (owners) have the final say about the quality of experience with which their content is expected to be consumed. Thus a service provider distributing

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

television content and not maintaining the desired quality of experience is liable to be sued by the content providers (owners). As mentioned earlier, we avoid that discussion here and assume that these legal issues have been taken care of by the service provider who is interested in distributing television content over the Internet. Let us now look at the resource needs of a service provider.

9.1 Resource Estimation

Distribution of television content, just as in the case of movie of demand, would require storage hours of professionally produced television content across all the channels, and the bandwidth to distribute that content to the consumers over the broadband networks. We consider the bandwidth needs first followed by storage needs.

9.1.1 Bandwidth

Analog television channels require a bandwidth of 8MHz. Each such channel can accommodate five digital television channels. Thus, if there are 100 8MHz analog channels, they can accommodate 500 digital television channels. Each digital television channel (standard definition) would require 4–6 mbps and the bandwidth requirements would approximately double for high definition television channels. Five hundred digital TV channels would thus require an aggregate bandwidth of approximately 2–3 gbps. Most importantly, these channels would be multicast to all consumers simultaneously just as in broadcast television and, as a result, neither the source nor the network would have to send the same content multiple times as would be the case for one-to-one transmission of content. Note that, with multicast, the sender sends the content once regardless of the number of end users while in case of one-to-one transmission the sender sends the same content as many times as there are end users. From the network point of view, in the case of multicast, the network replicates content only at the point where the multicast tree branches and hence multicast uses the network in the most efficient manner. In sharp contrast, for one-to-one transmission, since the number of times that the source replicates content equals the number of end users, the network has to transport the same content as many times as there are end users and hence its resources are not efficiently used.

9.1.2 Storage

An aggregate bandwidth of 2–3 gbps translates to 648–972TB/month assuming continuous transmission 24/7 for a month (30 days) at the aggregate rate. This implies that the service provider interested in distributing television content to the end users would require storage at the rate of 648–972TB/month. For the sake of simplicity, let us use 1PB/month as the storage requirement number. However, the service provider may decide never to redistribute certain content after it is broadcast once and hence may not need to store such content. This will reduce the storage requirement from growing at the astounding rate of 1PB/month. In addition, the service provider may also choose to keep only select content (say 10% of the content) for ever and discard the rest after a number of years. That will also put a cap on the storage needs of the service provider.

Another way of looking at the storage need would be to fix the storage size (to say, 100 PB) and use policies similar to those used in *cache replacement* to decide how best to use the fixed amount of storage space.

9.2 Technology

9.2.1 CoolStreaming

As discussed in the previous chapter, many overlay construction algorithms have been proposed in the literature for constructing a tree structure for delivering data using multicast. While this works well with dedicated infrastructure, such as in IP multicast, it often fails in the case of application-level overlay with dynamic nodes as the overlay nodes may crash or leave unpredictably making the tree highly unstable. Instability of the overlay multicast tree becomes a bigger issue with streaming applications that have high bandwidth and stringent continuity demands. Sophisticated structures, such as mesh, can partially solve the problem, but they are much more complex and often less scalable. Migrating multicast functionalities to application layer leads to greater flexibilities because the nodes have strong buffering capabilities and can adaptively and intelligently determine the data forwarding directions. These observations led the inventors of CoolStreaming [3, 11, 41] to envision a *datacentric* design for a streaming overlay. In this design, a node always forwards data to others that are expecting the data, with no prescribed roles of nodes, such as upstream/downstream. In fact, it is the availability of data that guides the flow directions, and not a specific overlay structure with restrictive flow directions. This datacentric design is more suitable for overlay with highly dynamic nodes, leading to the design of DONet, a data-driven overlay network. Basically, each node in DONet periodically exchanges data availability information with a set of partners, retrieves unavailable data from one or more partners, and/or supplies available data to partners, very much like the P2P streaming described earlier in this book.

There are three key features of the data-driven design in DONet:

- *Ease of implementation*: there is no need to construct and maintain a complex global structure.
- *Efficiency*: data forwarding is dynamically determined according to data availability and not restricted by specific directions.
- *Robustness*: periodically updated data availability information enables adaptive and quick switching among multiple sources of data.

Furthermore, it has been shown, using analytical results [41], that a logarithmic relationship exists between the overlay radius and its size, implying that DONet can scale to large networks with limited delay.

9.2.2 Design of DONet

Figure 9.1 depicts the system diagram of a DONet node. There are three key modules:

- **Membership manager**: helps the node maintain a partial view of other overlay nodes.
- **Partnership manager**: establishes and maintains the partnership with other nodes.
- **Scheduler**: schedules the transmission of video data.

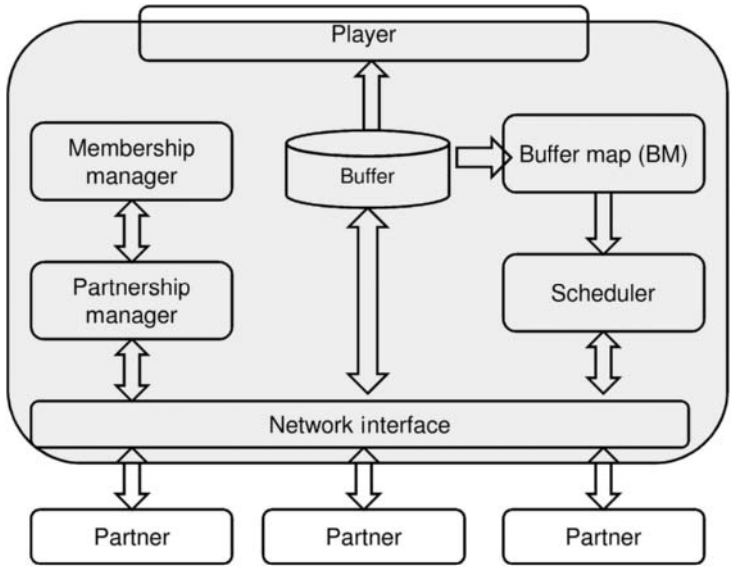


Figure 9.1 Functional diagram of a DONet node.

For each segment of a video stream, a DONet node can be either a receiver or a supplier, or both, depending on the dynamic availability information for the segment. The availability information is periodically exchanged between the node and its partners. An exception is the source node, which is always a supplier, and is referred to as the *origin node*. It could be a dedicated video server, or simply an overlay node that has a live video program to distribute. The interactions among the modules and their design issues are described next.

9.2.2.1 Node Join and Membership Management

Each DONet node has a unique identifier, such as its IP address, and maintains a membership cache (*mCache*) containing a partial list of the identifiers for the active nodes in the DONet.

In a basic node-joining algorithm, a newly joined node first contacts the origin node, which randomly selects a proxy (also called a "deputy") node from its mCache and redirects the new node to the proxy. The new node can then obtain a list of partner candidates from the proxy, and contacts these candidates to establish its partners in the overlay. Proxy thus plays the role of load balancer, evenly distributing load among the partners.

This process is generally viable because the origin node persists during the lifetime of streaming and its identifier/address is universally known. The redirection enables more uniform partner selections for newly joined nodes, and greatly minimizes the load of the origin.

A key practical issue here is how to create and update the mCache. To accommodate overlay dynamics, each node periodically generates a membership message to announce its existence; each message is a 4-tuple $<$*seq num*, *id*, *num partner*, *time to live*$>$, where *seq num* is a sequence number of the message, *id* is the node's identifier, *num partner* is its current number of partners, and *time to live* records the remaining valid time of the message. Scalable gossip

membership protocol (SCAM) [32], is used to distribute membership messages among DONet nodes. Scalable gossip membership is scalable, lightweight, and provides a uniform partial view at each node. Upon receiving a message from a new *seq num*, the DONet node updates its mCache entry for node *id*, or creates the entry if it does not exist. The entry is a 5-tuple <*seq num*, *id*, *num partner*, *time to live*, *last update time*>, where the first four components are copied from the received membership message, and the fifth is the local time of the last update for the entry.

9.2.2.2 Buffer Map Representation and Exchange

Neither the partnerships nor the data transmission directions are fixed in DONet. More explicitly, a video stream is divided into segments of uniform length, and the availability of the segments in the buffer of a node can be represented by a buffer map (BM). Each node continuously exchanges its BM with the partners and then schedules which segment is to be fetched from which partner accordingly.

As live media streaming has deadlines to meet, the filling sequence of the playout buffers of the DONet nodes are semi-synchronized. Analytical results demonstrate that the average segment delivery latency is bounded in DONet, and the experimental results further suggest that the time lags between nodes are not likely to exceed to 1 minute. Assume that each segment contains 1 s of video; a sliding window of 120-segment can effectively represent the buffer map of a node. The BM can be represented by 120 bits, where a bit 1 indicates that a segment is available and bit 0 indicates otherwise.

9.2.2.3 Scheduling Algorithm

Given the BMs of a node and that of its partners, a schedule is to be generated for fetching the expected segments from the partners. For a homogenous and static network, a simple round-robin scheduler may work well, but for a dynamic and heterogeneous network, a more intelligent scheduler is necessary. Specifically, the scheduling algorithm is needed to meet two constraints: the playback deadline for each segment, and the heterogeneous streaming bandwidth from the partners. If the first constraint cannot be satisfied, then the number of segments missing deadlines should be kept at a minimum. This being an NP-hard problem, it is not easy to find an optimal solution, particularly considering that the algorithm must quickly adapt to the highly dynamic network conditions. A simple heuristic can be used for fast response time.

1. **Step 1**: Calculate the number of potential suppliers for each segment (i.e., the partners containing the segment in their buffers).
2. **Step 2**: As a segment with less potential suppliers is more difficult to meet the deadline constraints, the algorithm determines the supplier of each segment starting from those with only one potential supplier, then those with two, and so forth.
3. **Step 3**: Among the multiple potential suppliers, the one with the highest bandwidth and enough available time is selected.

The algorithm steps taken from [41] are shown in Figure 9.2 for a more accurate description:

Input:

band[k] :	bandwidth from partner *k*;
bm[k] :	buffer map of partner *k*;
deadline[i] :	deadline of segment *i*;
seg_size :	segment size;
num_partners :	number of partners of the node;
set_partners :	set of partners of the node;
expected_set :	set of segments to be fetched.

Scheduling:

```
for segment i ε expected_set do
        n ← 0
        for j to num_partners do
                T[j,i] ←deadline[i] – current_time;
                        //available time for transmitting segment till i;
                n ← n + bm[j, i];
                        //number of potential suppliers for segment i;
        end for j ;
        If n = 1 then       //segments with only one potential supplier;
                k ← arg_r {bm[r, i] = 1};
                supplier[i] ← k;
                for j ε expected_set, j > k do
                        t[k, j] ← t[k, j] – seg_size/band[k];
                end for j;
        else
                dup_set[n] ← dup_set[n] U {i};
                supplier[n] ← null;
        end if ;
end for i ;

for n = 2 to num_partners do
        for each i ε dup_set[n] do
                        //segments with n potential suppliers;
                k ←     arg_r {band[r] > band[r'] | t[r, i] > seg_size/band[r],
                            t[r', i] > seg_size/band[r'], r, r' ε set_partners};
                if k =/= null then
                        supplier[i] ← k;
                        for j ε expected_set, j > k do
                                t[k, j] ← t[k, j] – seg_size/band[k];
                        end for j ;
                end if ;
        end for i ;
end for n;
```

Output:

supplier[i] : supplier for unavailable segment *i* ε *expected_set.*

Figure 9.2 Scheduling algorithm at a DONet node.

Each execution requires approximately 15 ms, implying that the computation overhead is quite low. The algorithm can thus be executed frequently to update the schedule.

Given a schedule, the segments to be fetched from the same supplier are marked in a BM-like bit sequence, which is sent to that supplier, and these segments are then delivered in order through a real-time transport protocol. DONet does not specify a particular protocol. However, the TCP-friendly rate control (TFRC) protocol [42] is used in many practical systems.

The BM and scheduling results can also be piggybacked on the data packets to achieve fast and low-overhead updates. Note that the origin node servers act as suppliers only, and they always have all the segments available. An adaptive scheduling algorithm enables an origin server to schedule requests from its partners in such a way that it does not receive more than a handful number of requests within a given duration. As a result, the origin server can maintain performance and does not get slowed down with increasing number of peers (partners).

If needed, it can also proactively control its load by advertising conservative buffer maps.

9.2.2.4 Failure Recovery and Partnership Refinement

Node departure in DONet can be either graceful (letting others know about its departure ahead of time) or accidental (without any prior notice) due to a crash. In both cases, the departure can be easily detected after an idle time of TFRC [42] or BM exchange. Once detected, an affected node can quickly react through rescheduling using the BM information of the remaining partners. In addition to this built-in recovery mechanism, there are additional techniques to enhance resilience even more:

1. Graceful departure: the departing node issues a departure message, which has the same format as the membership message, except that the *num partner* filed is set to -1.
2. Node failure: a partner that detects the failure issues the departure message on behalf of the failed node. The departure message is gossiped very much like the membership message. In the node failure case, duplicate departure messages may be generated by different partners, but only the first received will be gossiped by a node and others will be suppressed. Each node receiving the message flushes the entry for the departing node, if available, from its mCache.
3. *Random partner selection*: each node periodically establishes new partnerships with nodes randomly selected from its mCache. As a result, each node is able to maintain a stable number of partners (M) in the presence of node departures, and is also able to explore partners of better quality.

9.2.3 Evaluation of DONet

The performance of DONet was explored in the PlanetLab [43] setting with overlay nodes spread across all continents, with the majority being in North America and Europe. While details can be obtained from [41], the most important observations are summarized in this section. Experiments were conducted for two different environments: (i) stable and (ii) dynamic.

9.2.3.1 Performance under Stable Environment

In this case, all the nodes join in an initialization period (around 1 min) and then stay on through the entire lifetime of the streaming (120 min, a typical length for a movie). The

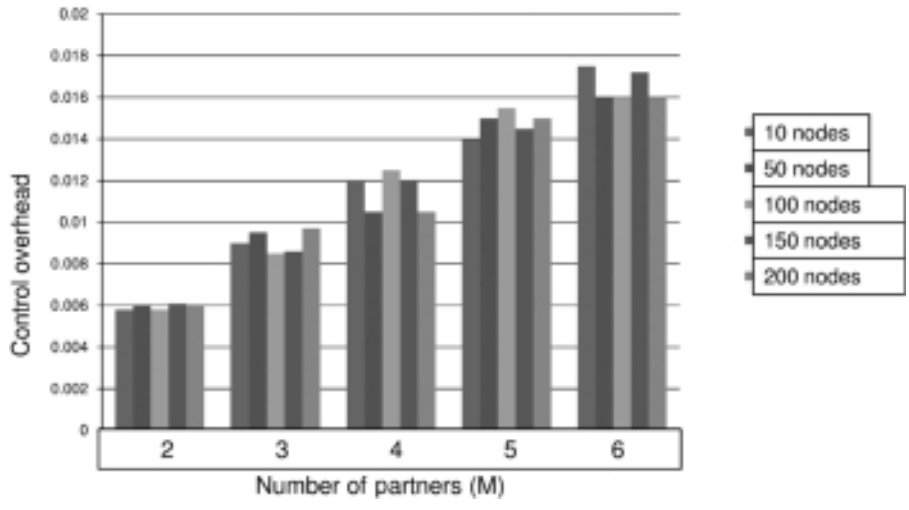


Figure 9.3 Control overhead as a function of the number.

default streaming rate is 500 kbps and each segment contains 1 s of the stream. Each DONet node maintains a sliding window of 60 segments, or 60 seconds of the streaming data, and the playback starts 10 s after receiving the first segment.

Control overhead: As the membership management employs a light-weight gossip protocol, most control messages in DONet are for exchanging data availability information. Naturally, as the number of partners increases, control traffic also increases. Control overhead, which is defined as the ratio of control traffic volume to video traffic volume, is a proxy for control traffic and increases with number of partners as shown in Figure 9.3. Compared to video traffic, the control traffic is essentially minor, even with over five or six partners (less than 2% of the total traffic). This is intuitive given that the availability of each video segment is represented by a single bit only.

Playback continuity: Maintaining continuous playback is the main objective of streaming. To evaluate continuity, a new parameter called the *continuity index* is needed. The *continuity index* is defined as the ratio of the number of segments that arrive before or on playback deadlines to the total number segments. Figure 9.4 shows the continuity index as a function of M, the number of partners. Continuity improves with increasing M because each node has more choices for suppliers. The improvements with more than four partners are marginal. The continuity index is plotted as a function of different streaming rates in Figure 9.5. Again, the use of four partners is reasonably good, even under high rates. Considering that the control overhead increases with more partners, $M = 4$ is a good practical choice.

Scalability: It can be seen from Figure 9.6 that the control overhead at each node is almost independent of the overlay size. This is because the availability information (BM) is only locally exchanged. In addition, as shown in Figure 9.7, the continuity index is maintained high even with large overlay sizes. In fact, a larger overlay often leads to better playback continuity due to the increasing degree of cooperation. In summary, DONet is scalable in terms of both overlay size and streaming rate.

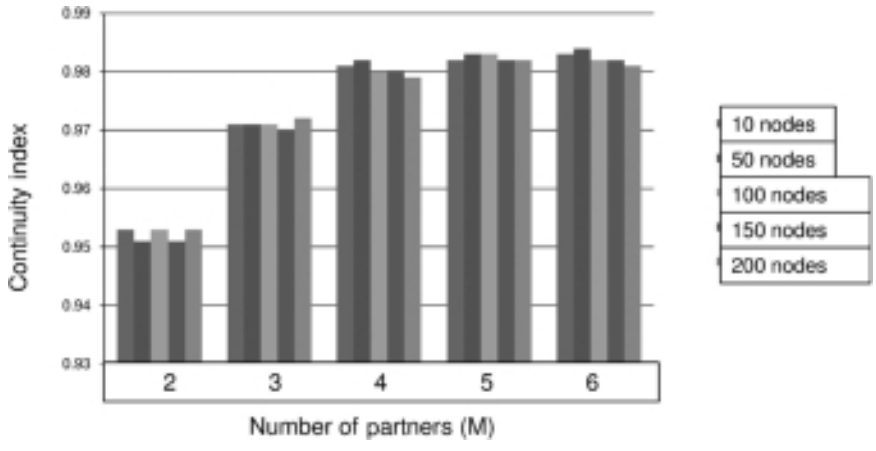


Figure 9.4 Continuity index as a function of the number of partners.

9.2.3.2 Performance under Dynamic Environment

In the dynamic DONet environment, nodes join, leave and fail. Most parameter settings are similar to those in the stable environment, except that each node changes its status following an ON/OFF model. The ON state means that a node actively participates in the overlay, while the OFF state means that a node leaves (or fails). Both ON and OFF periods are exponentially distributed with an average of T seconds.

Figure 9.6 shows the control overhead as a function of the ON/OFF period for different overlay size. Note that the control overhead decreases as dynamism decreases or the ON/OFF

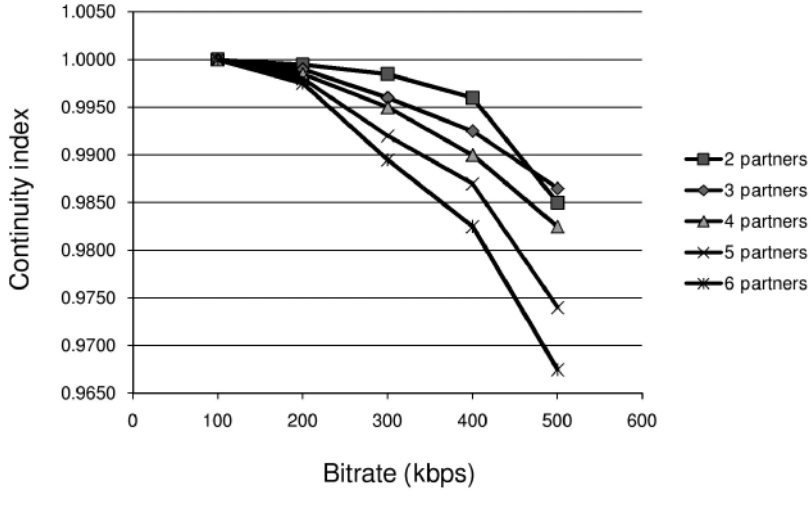


Figure 9.5 Continuity index as a function of the streaming rate. Overlay size = 200 nodes.

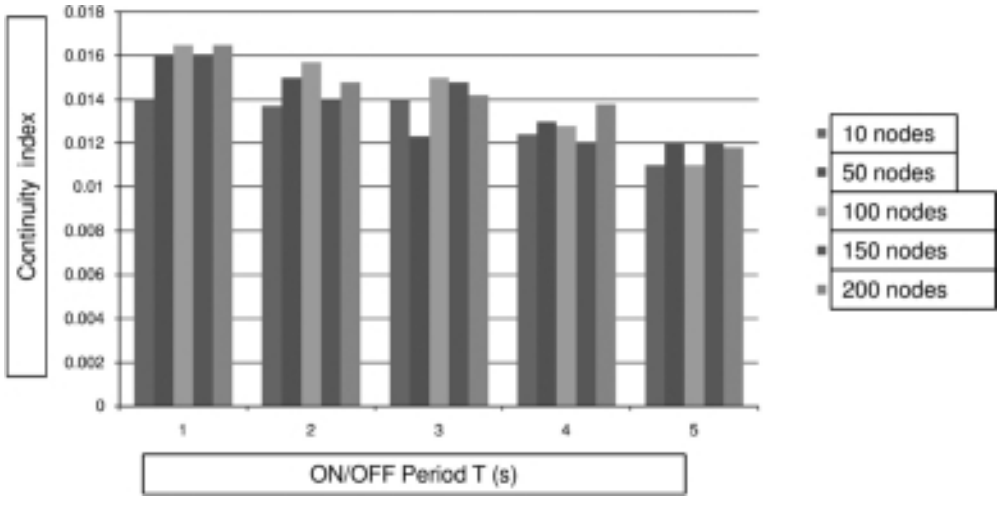


Figure 9.6 Control overhead as a function of the average ON/OFF period for different overlay sizes.

period increases. The extra control traffic results from the leave/failure notifications, which forms a small part of the overall control traffic.

Figure 9.7 shows the impact of dynamism on the continuity index. Specifically, the continuity index increases (or a smaller number of packets miss their playout deadline) as dynamism decreases or the ON/OFF period increases. On the other hand, a shorter ON/OFF period leads to poorer continuity but the drop is minimal. With the intrinsic recovery mechanism, the continuity index of DONet remains acceptable even under highly dynamic networks (alternating in less than 1 min).

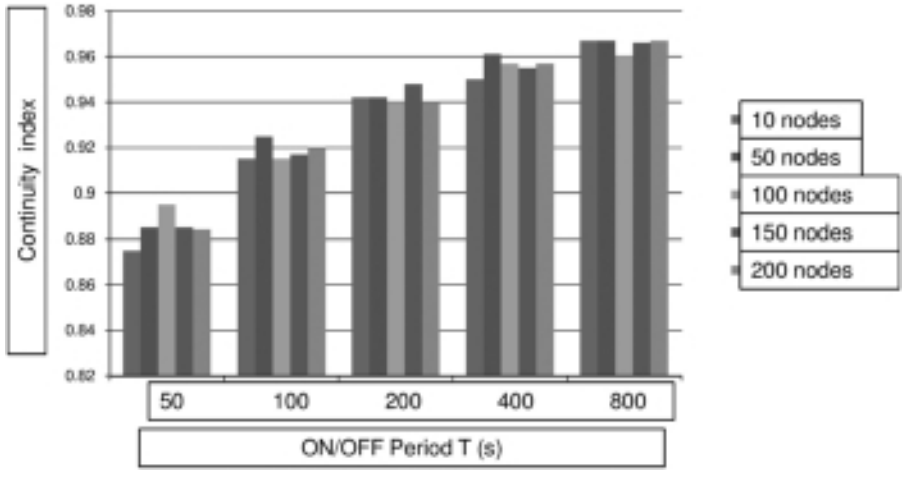


Figure 9.7 Continuity index as a function of the average ON/OFF period for different overlay sizes.

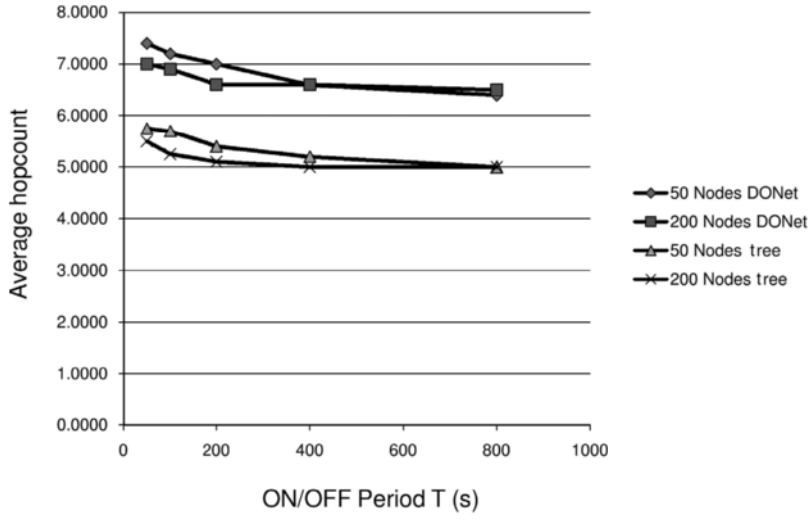


Figure 9.8 Average overlay hop-count of DONet and tree-based overlay.

9.2.3.3 Comparison with Tree-Based Overlay

Given the popularity of tree-based overlays, it makes sense to compare their performance with that of DONet. Each internal tree node is limited to three children in addition to one parent node. The total degree is thus four. The only exception is the origin node, which can have four children. These limits are set to have a fair comparison with DONet where each node can have four partner nodes (degree of four). First the end-to-end delays of DONet and the tree-based overlay are compared. However, for simplicity, hop count is used as a proxy for end-to-end delay.

As shown in Figure 9.8, contrary to the popular belief that a tree-based overlay achieves shorter delay, DONet does better than the tree-based overlay when it comes to end-to-end delay under both stable and dynamic environments. In fact, the outbound bandwidth constraints can noticeably increase the height of the tree, leading to worse end-to-end delay compared to DONet.

In addition, the continuity index of the tree topology is remarkably lower than that of DONet, particularly with dynamic and large overlays as shown in Figure 9.9. This happens because a tree structure is very vulnerable to internal node failures. In fact, some internal nodes are really crucial in the tree, and if any one of those nodes fails for whatever reason, it may cause buffer underflow in all the downstream nodes. The more the number of downstream nodes is dependent on the critical internal nodes, the higher is the impact on end-to-end delay. To further illustrate the vulnerability of the tree topology, the continuity index over time is shown in an experiment of 200 nodes in Figure 9.10.

Note that the continuity index of the tree overlay is not only lower than that of the DONet, but is also highly fluctuating. As an example, between 800 s and 900 s, the continuity index for tree-based overlay drops to 0.4 when a child of the root leaves. Such issues do not arise in DONet, as the loads of the nodes are evenly distributed and delivering paths are dynamically set according to data availability. In fact, it can be theoretically proven that even if the tree is

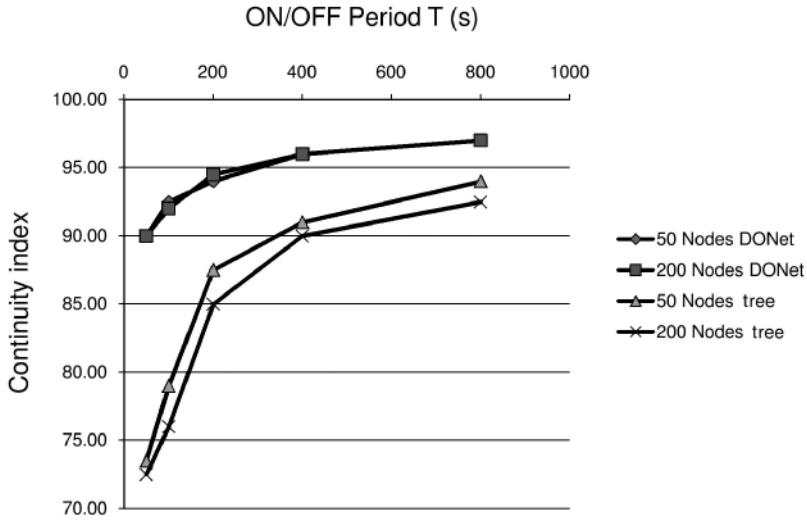


Figure 9.9 Comparison of continuity indices for DONet and the tree-based overlay.

full and balanced, a tree-based overlay is still more vulnerable in a dynamic environment as compared to DONet.

9.2.4 GridMedia

GridMedia [44] is the technology behind an unstructured P2P network for large-scale live media streaming through global Internet. It uses gossip-based protocol to organize end nodes

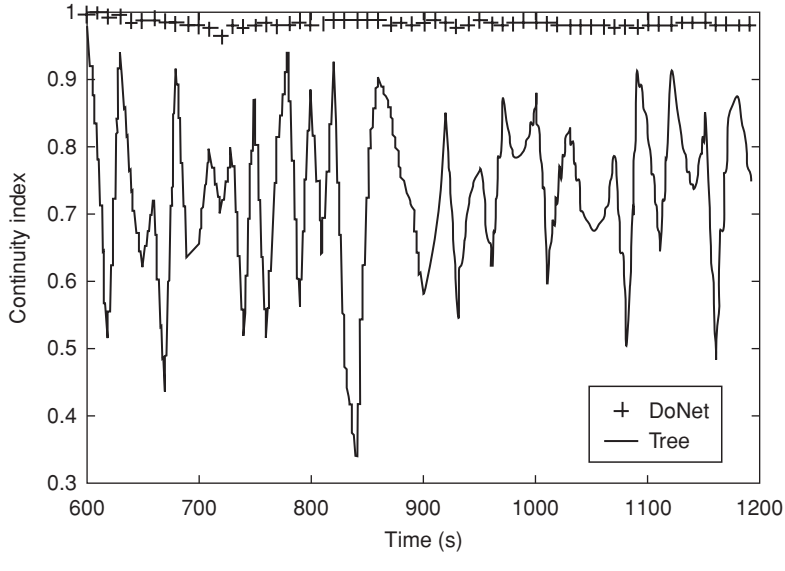


Figure 9.10 Samples of continuity indices for DONet and a tree-based overlay in a experiment (from 10 min to 20 min).

into an application layer overlay. Each node in GridMedia independently selects its neighbors and uses a novel and efficient push-pull streaming mechanism to fetch data from neighbors with low latency and little redundancy. The traditional pull mode in the unstructured overlay (for example, DONet described in the previous section) has inherent robustness to high churn rate, which is common in the P2P environment, while the push mode could efficiently diminish the accumulated latency observed at end users. A practical system based on this architecture has been developed and its performance has been evaluated on PlanetLab [42] in various rigorous conditions. All the results demonstrate that the proposed push-pull method in GridMedia achieves good performance even with high group change rate and very low upload bandwidth limitation.

9.2.4.1 GridMedia Overview

GridMedia technology is based on an unstructured P2P overlay. This is particularly useful in making live media streaming robust against high churn rate in P2P live media streaming applications. Each GridMedia node has two components: (1) an overlay manager and (2) a streaming scheduler. These components are described next.

9.2.4.1.1 Overlay Manager

The overlay manager component on each node uses gossip protocol to discover the appropriate neighbors in the P2P overlay. GridMedia uses a well-known *rendezvous point* (RP) for bootstrapping the formation of P2P overlay. In fact, a new node first contacts the RP to get a "candidates list", which consists of nodes that are already in the P2P network. A subset of nodes from the "candidates list" is chosen as the initial neighbors. The new node first measures the round-trip time (RTT) to each node in the "candidates list" and then chooses some nodes with the minimum RTT as one part of its initial neighbors. In order to avoid partitioning of the overlay network, the other part of its initial neighbors is selected randomly. As this process is repeated for all nodes, the nodes self-organize into an unstructured mesh.

Each node maintains a member table initially obtained from the RP, and later updated based on exchange of information among neighboring nodes. In fact, the information of member tables is encapsulated into a UDP packet and exchanged among neighbors periodically using gossip protocol. In order to reduce control overhead of the gossip protocol, not all the items in the member table are included in the information packets exchanged. Each node updates its member table in accordance with the member table sent by its neighbors. Note that each item in the table has a field called *life-time.* It denotes the elapsed time since the latest message was received from this member. When the life-time of a node exceeds the predefined threshold, it is removed from member table. Once a node quits, it broadcasts a "quit message" to all its neighbors. Further, this message will be flooded within limited hop counts. The nodes who receive this message will delete the corresponding node from its member table. In addition, each node delivers an "alive message" to all its neighbors periodically to declare its existence. Thus, the failure of a neighbor can be detected when the node does not receive any message from a neighbor for a while, this neighbor then will be erased from member table. Figure 9.11 illustrates an instance of the unstructured mesh in GridMedia. To calculate the life-time of each node in member table and to deal with the synchronization issue among nodes more lightly, the local clock of the newly joined node should be synchronized with that in the RP. In fact, the participating node sends a UDP packet containing its local time. Then the RP returns the difference between that clock time and the clock on the RP immediately. If those UDP

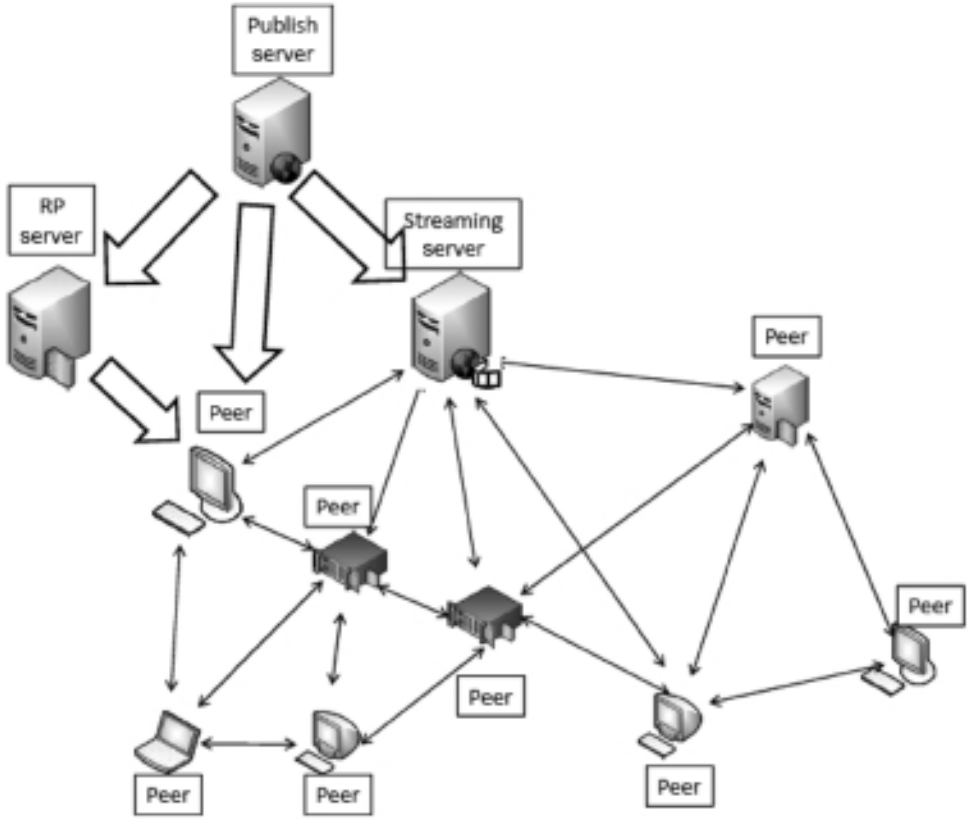


Figure 9.11 The structure of GridMedia.

packets are lost, this process will be repeated. Although the synchronization method is not extremely precise, it is clear that the error will not exceed the RTT value between the node and the RP, which is accurate enough for this system.

If some neighbors of a node are quitting or failing, the node will retrieve alternative neighbors in its member table. The selection of substitutes is similar to the selection procedure of initial neighbors. Moreover, every node will refine its neighbors iteratively. Each neighbor will obtain an evaluation index in one term. The evaluation index of a neighbor here means the sum of packets received from this neighbor and the packets sent to this neighbor, that is, the total bidirectional traffic between the node and its neighbor in one bout. If the minimum evaluation index of the neighbors is less than a threshold, the poorest neighbor is discarded, and in the meantime, a new neighbor is probed from the member table.

9.2.4.1.2 Streaming Scheduler

The streaming scheduler component of each node is responsible for fetching packets (that it does not have) from its neighbors and in delivering packets requested by the neighbors. GridMedia uses a combination of "pull" and "push" methods. The pull mode is similar to the

data-driven approach in DONet where one UDP/RTP packet is used as a transmission unit rather than a segment containing one second media content. The sequence number field in the RTP packet header can be used for buffer management, while the timestamp field can be used for establishing synchronization among nodes since the packet rate is variable in general. Just as in DONet, buffer maps are swapped between neighbors continuously. A buffer map in GridMedia has three fields: (i) the maximum sequence number of buffer map *sBM*, (ii) the length of buffer map *lBM*, and (iii) a bit vector representing the availability of all packets with the sequence number from *sBM* -*lBM*+1 to *sBM*. Every node periodically sends a request to its neighbor to fetch packets that it does not have. When a node receives a request from one of its neighbors, it tries to deliver the requested packets to this neighbor. In GridMedia, the transmission protocol between neighbors can be UDP, TCP or TFRC [41]. However, UDP is the default protocol due to real-time characteristics in live media streaming. Moreover, UDP has no congestion control mechanism. Since the maximum download bandwidth of many hosts over the Internet is not very high (such as the hosts accessing Internet by DSL or cable), any bursty arrival of packets to those hosts would most likely lead to packet loss. Therefore, it is important to have a traffic shaper between each pair of neighbors to smooth the traffic. Once a request is received from one neighbor, requested packets are delivered at a uniform rate rather than being delivered in a burst.

9.2.4.1.3 Drawbacks of the Pull Method

Each node in a pure pull-based method (such as DONet) periodically exchanges a buffer map (BM) of media segments with partners, and then retrieves the missing segments from partners that possess them. Here are few more definitions that are useful for the discussion that follows:

- Absolute delay: end-to-end latency (delay) between the sampling time at the server and the playback time at the receiver.
- Absolute playback deadline: playback time of a packet.
- Delivery ratio: ratio of number of packets that arrive at each node before or right on absolute playback deadline to the total number of packets. With increasing absolute delay, higher delivery ratio can be achieved.
- α-playback-time: minimum absolute delay at which the delivery ratio is larger than α, where $0 <= \alpha <= 1$.

The goal is to investigate the delivery ratio as a function of absolute delay for a pure pull method.

9.2.4.1.4 Simple Analysis of the Pull Method

A pull method has three steps (see Figure 9.12):

1. The sender informs the receiver about the existence of a packet in its buffer.
2. If the receiver needs the packet, it sends a request for the packet to the sender.
3. The sender delivers all requested packets to the receiver at a smooth rate.

Note that each step incurs a one-way delay (OWD) between sender and receiver. Thus, there are at least three one-way delays. However, there are additional components of delay. For example, for efficiency reasons, buffer map and request packets are only sent in a certain time

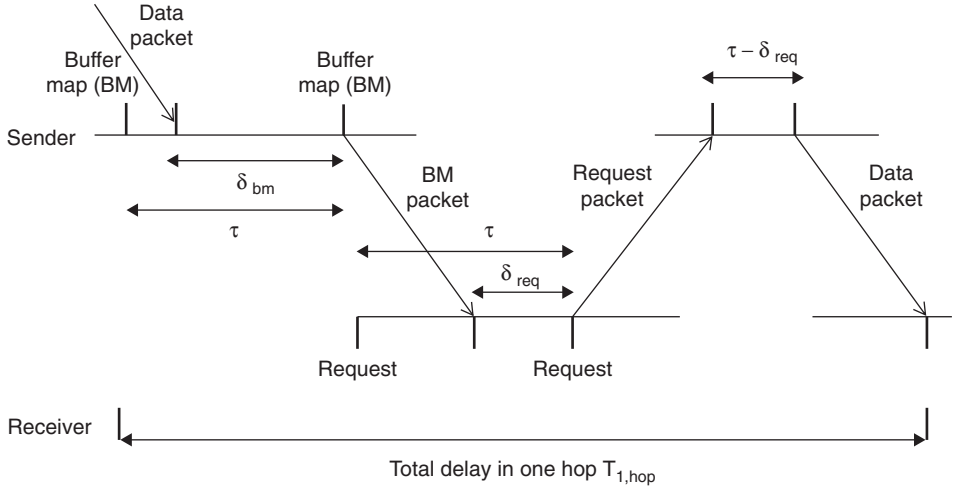


Figure 9.12 Delay of a Packet Transmitted in one Hop using Pure Pull Method.

interval so that multiple packets can be mapped into one single packet. As a result, the delivery of most packets will have extra delays in one hop. Let us denote the interval between two buffer map (request) packets by τ and the waiting time between the arrival of a data packet and the sending of the next buffer map packet by δbm to denote this waiting time. Similarly, because the request packet is not sent as soon as the buffer map packet is received, there is a waiting time and we refer to that waiting time by δreq . Finally, data packets are not sent as soon as the request packet is received. In fact, packets have to wait due to pacing. As the packet with the largest waiting time (d_{req}) in the second step will be sent first and the transmission of the requested packets should be finished in one cycle (t) in the presence of enough bandwidth, the waiting time is τ-δreq. $\Delta\, req$ and δbm are independent, and their average values are both $\tau/2$ if the packet rate is invariable. Thus, the total average latency for a packet transmitted in one hop $T1hop$ is approximately: $\delta bm + \delta req + (\tau\text{-}\delta req) + 3\ OWD = 3t/2 + 3OWD$.

9.2.4.1.5 Experiment of the Pull Method

An experiment over PlanetLab was performed by [44] in a static environment (that is, all the nodes were continuously online during the experiment) with the following parameters:

- Group size of 310 ($N = 310$).
- Each node with at most five neighbors ($n = 5$).
- Period of exchanging buffer map packet and request packet is 1 second ($\tau = 1$ s).
- Average packet rate is 30 packets/s.
- Average packet size is 1300 bytes. Thus, the bit rate is about 310 kbps.
- Average RTT between nodes is approximately 120 ms or OWD = 60 ms.

Delivery ratios at various absolute delays are plotted in Figure 9.13.

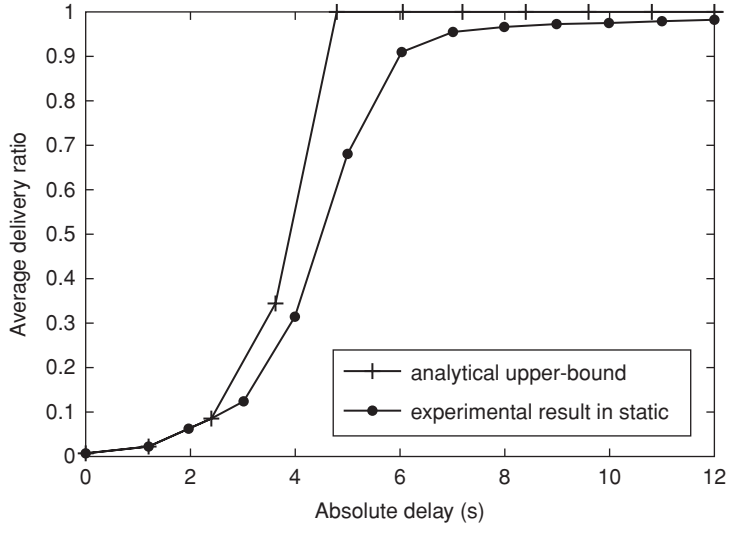


Figure 9.13 Delivery ratio as a function of absolute delay by analysis and experiment, group size 310.

Figure 9.13 shows the analytical upper bound and experimental result of the average delivery ratio as a function of absolute delay in static environment. Both analytical and experimental results reveal that the pure pull method results in significantly high absolute delay. Further, the α-playback-time ($\alpha = 0.97$) of the experimental result is about 10 s.

9.2.4.1.6 Push-Pull Method used in GridMedia

Although the traditional pull or data-driven method works well with high churn rate and dynamic network conditions, it may not meet the demands of delay-sensitive applications because of the huge latency accumulated hop by hop. A push-pull streaming mechanism greatly reduces the absolute delay while inheriting the simplicity and robustness of traditional pull method. Specifically, the push-pull mechanism can obtain the same delivery ratio as a pure pull method but with much smaller absolute delay.

9.2.4.1.7 Packet Scheduling

In a push-pull streaming mechanism (Figure 9.14), each node uses the pull method in the startup phase, and after that each node relays packets to its neighbors as soon as the packet arrives without any explicit request from the neighbors.

Essentially, the streaming packets are classified into pull packets and push packets. A pull packet is delivered by a neighbor only when the packet is requested, while a push packet is relayed by a neighbor as soon as it is received. Each node initially works in pull mode and then, based on the traffic from each neighbor, subscribes to the push packets from its neighbors at the end of each time interval.

In order to eliminate duplication, each neighbor of a node should stream different packets. The list of push packets to be obtained from a neighbor in the next time interval

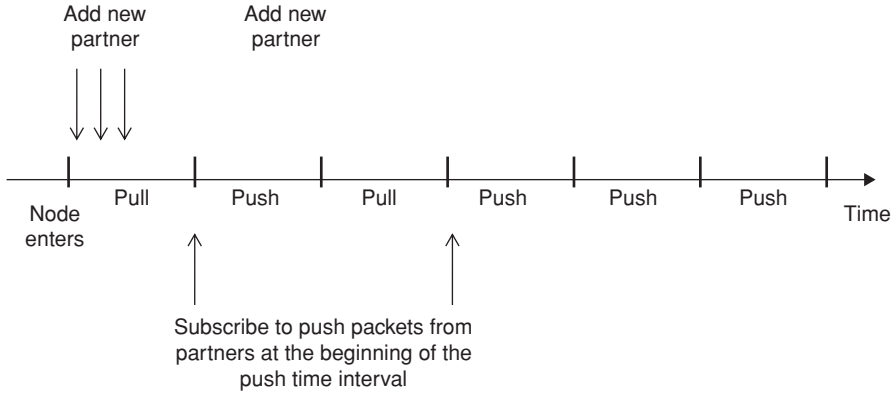


Figure 9.14 Push-pull mechanism in GridMedia.

is called a *pushing packets map* (PPMAP). Here is how a PPMAP is constructed for a neighbor:

1. **Step 1**: Partition a stream into P parts (numbered by $0 \ldots P-1$).
2. **Step 2**: Hash each packet into only one of the P parts where the hashing function (mod P function) is applied on the RTP sequence number. Note that every bit in PPMAP represents one part of the stream.
3. **Step 3**: Prepare two PPMAPs: (i) incoming PPMAP and (ii) outgoing PPMAP. Incoming PPMAP represents the push packets to be *received* from a neighbor while outgoing PPMAP represents push packets to be *sent* to a neighbor.

Push packets are allocated to each neighbor at the next time interval. However, the selection probability of each neighbor is equal to the percentage of traffic from that neighbor in the previous time interval. That is, more packets are pushed from those neighbors that pull more packets from a given node. This concept is similar to Bitorrent's tit-for-tat principle.

Note that when packets are pushed, they might be lost due to the unreliability of the network link or the failure of neighbors. However, such packets are retrieved in GridMedia using pull just as in a traditional pull method. Thus, most of the packets received will be push packets from the second time interval onwards. To keep the process simple, the start of time interval on each node should be synchronized. Therefore in the GridMedia scheme each node synchronizess with RP right at the time of joining.

9.2.4.2 Performance Evaluation of GridMedia over PlanetLab

The authors in [44] performed extensive performance evaluation of GridMedia over the world-wide overlay network of PlanetLab [43]. The results below are taken from [44] to illustrate the key technical contribution of GridMedia's push-pull technology vis-à-vis DONet's pull-based technology.

Table 9.1 Experimental parameters without upload bandwidth limitation.

Parameter	Value
Group Size	300–340
Number of Neighbors	5
Streaming Bit Rate	310 Kbps
Packet Rate	30 Packets/sec
Buffer Map Interval	1 Second
Request Interval	1 Second
Push Packet Subscription Interval	10 Second
Average online time in Dynamic environment	100 Second
Average offline time in Dynamic environment	10 Second

9.2.4.2.1 Control Overhead of the Gossip Protocol in GridMedia

Control overhead is defined as the ratio of the control traffic to total traffic at each node. Control traffic results from the messages used to probe and keep neighbors, the exchanged member table packets, buffer map packets, request packets, and PPMAP packets in push-pull method. As expected, most of the control traffic is generated from the exchange of buffer map packets among neighbors. Since every control message has a scope constraint, the average control overhead of each node does not increase with the overlay size. The main parameters are shown in Table 9.1. The experiments are repeated with neighbor sizes of three and five to explore the impact of number of neighbors on control overhead.

As illustrated in Figure 9.15, the two solid lines show the control overhead with different overlay sizes when each node has five neighbors while the two dotted lines represent the control overhead when each node has three neighbors. It is clear that the overhead has no relationship

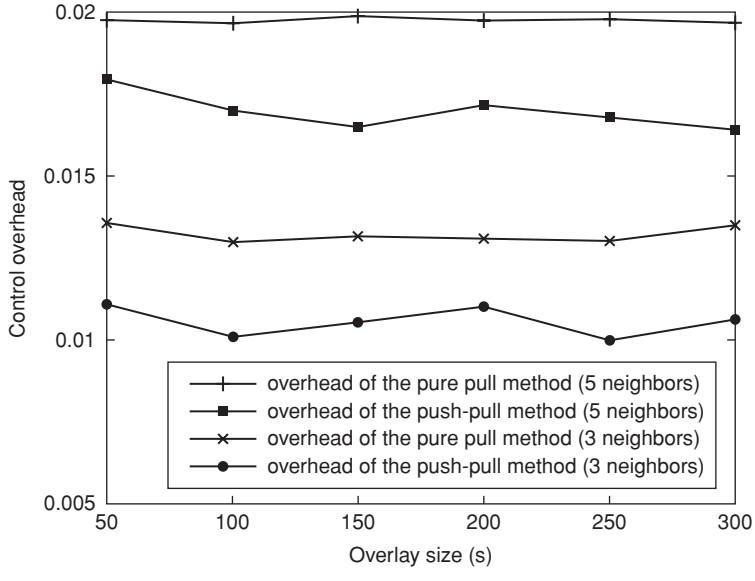


Figure 9.15 Control overhead of the gossip protocol in GridMedia.

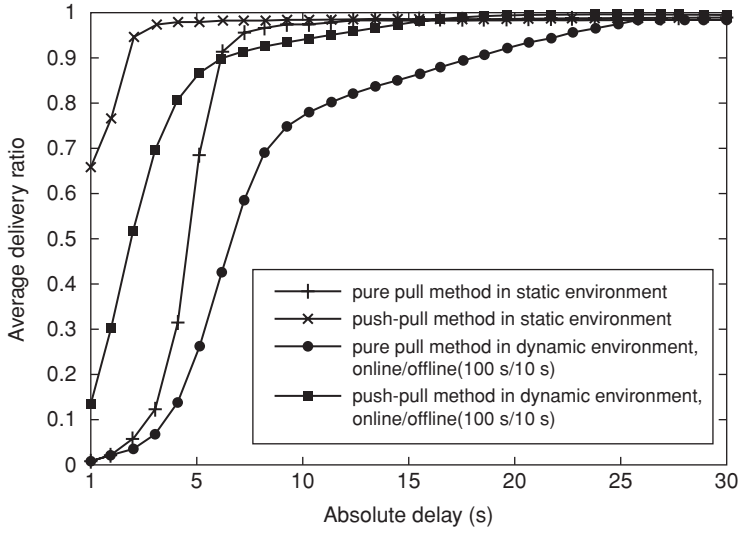


Figure 9.16 Comparison between pull and push-pull methods in both static and dynamic environments (no limit on upload bandwidth).

with the overlay size. In fact, the key factor affecting the control overhead is the number of neighbors. As expected, the more neighbors each node has, the more control messages should be exchanged between them. Additionally, it should be noted that the control overhead of pull method is always a little greater than of the push-pull method. This is because in pure pull method, each node continuously sends request packets (one per second). In push-pull method, each node mostly works under push mode, and the request packet per second is replaced by the aggregate PPMAP exchange every 10 s. Hence the overhead in push-pull method becomes smaller.

9.2.4.2.2 Performance Evaluation without Upload Bandwidth Limitation

The results presented in this subsection assume on limitation on upload bandwidth of the peer nodes. The parameters of the experiments are summarized in Table 9.1.

Figure 9.16 demonstrates the comparison between the traditional pull method and the push-pull method in static and dynamic environments. In a dynamic environment, 335 nodes on PlanetLab are used where each node continuously joins and departs from the group. The online and offline durations of each node are exponentially distributed with mean values of 100 and 10 s respectively. This represents very high group dynamics leading to 300–320 nodes being online at any instant of time. Note that the push-pull method can reach an equivalent

Table 9.2 Summary of results (no limit on upload bandwidth).

	Push-Pull	Pull
Static 97%-playback-times	4 sec	11 sec
Dynamic 95%-playback-times	13 sec	22 sec

Table 9.3 Experimental parameters with upload bandwidth limitation.

Parameter	Value
Group Size	290–320
Number of Neighbors	5
Streaming Bit Rate	310 kbps
Packet Rate	30 Packets/sec
Buffer Map Interval	1 Second
Request Interval	1 Second
Push Packet subscription Interval	10 Second
Upload Bandwidth limitation at each node	500 Kbps

α-playback-time at a much smaller absolute delay than the pure pull method as shown in Table 9.2. The two solid lines in Figure 9.16 show that the α-playback-times ($\alpha = 0.97$) of the pure pull method and the push-pull method are 11 s and 4 s respectively in a static environment while the two dotted lines illustrate that the α-playback-times ($\alpha = 0.95$) of the two methods in a dynamic environment are 22 and 13 s respectively. These results validate the design principles of GridMedia and the value added by push techniques to the traditional data-driven pull methods.

9.2.4.2.3 Performance Evaluation with Upload Bandwidth Limitation

The results presented in this subsection assume a limitation of 500 kbps on upload bandwidth of the peer nodes. The parameters of the experiments are summarized in Table 9.3.

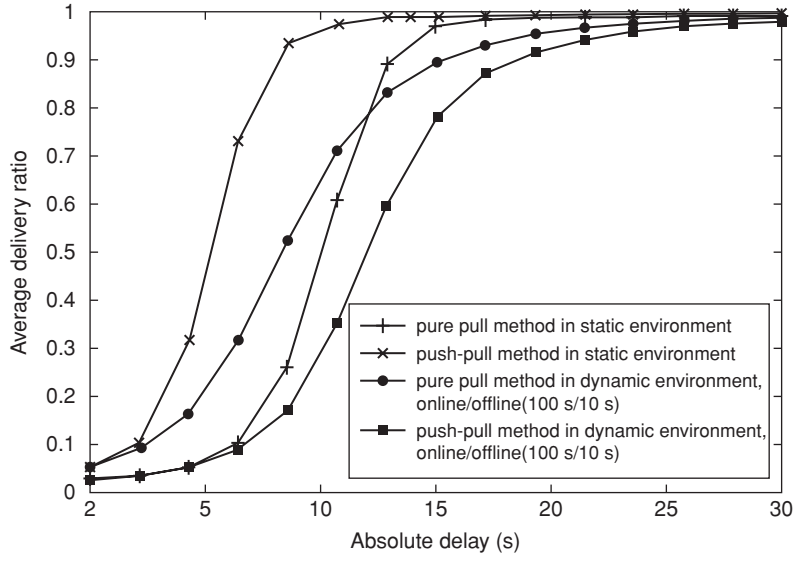


Figure 9.17 Comparison between pull and push-pull methods in both static and dynamic environments (limited upload bandwith).

Table 9.4 Summary of results (limited upload bandwidth).

	Push-Pull	Pull
Static 97%-playback-times	13 sec	18 sec
Dynamic 95%-playback-times	20 sec	24 sec

Figure 9.17 illustrates the comparison between the pull method and the push-pull method with 500 kbps uploading bandwidth limitation on each node in both static and dynamic environments. On PlanetLab, 335 nodes are used where each node continuously joins and departs from the group. The online and offline durations of each node are exponentially distributed with mean values of 100 s and 10 s respectively. This represents very high group dynamics leading to 290–310 nodes being online at any instant of time. Note that the push-pull method can reach an equivalent α-playback time at a much smaller absolute delay than the pure pull method as shown in Table 9.4. The two solid lines in Figure 9.17 show that the α-playback times ($\alpha = 0.97$) of the pure pull method and the push-pull method are 18 and 13 s respectively in a static environment while the two dotted lines illustrate that the α-playback-times ($\alpha = 0.95$) of the two methods in a dynamic environment are 24 and 20 s respectively. These results validate the design principles of GridMedia and the value added by push techniques to the traditional data-driven pull methods.

9.2.4.3 Summary of GridMedia

Through a simple analysis of the traditional pull method in P2P streaming multicast, it was shown that user nodes participating in the pull method can experience significant latency. An unstructured P2P protocol combining push and pull mechanisms, called GridMedia, was proposed to greatly reduce the latency while retaining the positive features (such as simplicity and robustness) of the pull method. Performance evaluation of GridMedia over PlanetLab demonstrated that it is an efficient technique for distributing live multimedia content, including Broadcast Television, over the Internet.

9.3 Products

9.3.1 Satellite Direct

With satellite direct software on a PC [4], one can watch live television channels, such as National Football League (NFL) and a host of other TV stations spanning films, music, news, cartoons and documentaries, all streamed to the PC connected to a broadband Internet connection. For example, one can watch baseball, world cup soccer, basketball, racing, ESPN news, EuroSport TV, Games Sports TV in sports category and BBC News, CNN, CNN-IBN, NBC News, ABC News, Eyewitness News, C-SPAN and so on in the news category. There is no need for a satellite dish, receiver or any other DVI TV equipment to Satellite Direct. There are over 3500 channels available worldwide.

9.3.2 Download Dish TV for PC Internet Streaming

DishTvforPC [5] uses P2P streaming technology to allow one to watch TV over the Internet on PC or laptop. More than 3000 channels from around the world can be streamed directly to

PC. Dish TV for PC offers live sports like boxing, football and soccer on ESPN or the latest live news from news agencies like BBC, CNBC, CNN, Fox, Bloomberg, NBC. Even the latest episodes of popular TV shows can also be watched as can movies from HBO or the shopping channels like HSN and QVC.

9.3.3 PPMate Streaming TV

PPMate software [6] lets one watch TV online, including sports, live football, news, movies, live TV and VOD TV series. It offers over 940 channels to choose from, all streamed directly to PC. PPMate also provides the option to record anything that is being watched.

9.3.4 SopCast TV Streaming

SoP is the abbreviation for Streaming over P2P. Sopcast is a Streaming Direct Broadcasting System based on P2P. Watching TV online has evolved with the latest version of Sopcast [7] which allows superfast channel changing and almost immediate start-up. All the major TV channels such as Star Sports, NBA LIVE, and so on, can be watched from the PC.

SopCast includes four components, Sop Server, Sop Player, VoD Server and Web Player. Sop Server is for the broadcaster whereas Sop Player is for the viewer. Similarly, VoD Server provides movies over P2P and Web Player is used by the end users for Web viewer.

9.3.5 3webTotal Tv and Radio Tuner

3 Web Total Tv and Radio [8] provides access to more than 900 TV channels from 104 countries and more than 4000 radio stations from 148 countries for less than one month of basic cable service. It allows one to watch channels such as HBO, Bravo, Paramount Comedy 1, Sci-fi, Men and Motors, Channel 4, E4, More4 and many more.

9.3.6 Free Internet TV Streams

Free Internet TV [9] is a software program that provides users with their own Internet television tuner. The software allows them to view nearly 2000 streaming TV stations from around the globe. There is no need for a satellite or cable tuner card because the TV channels are streamed directly down the broadband Internet connection. This is a good choice for one who wants to watch TV from all over the world in many different languages.

9.3.7 Online TV Live

Online TV Live [10] is a great new piece of software that lets one watch over 3500 free Internet TV channels and on-demand streamed videos, and lets one listen to hundreds of free Internet radio stations on a PC. Online TV Live has an attractive and intuitive user interface with a simple menu on the left-hand side of the program listing the different categories and stations. One can also easily search for TV and radio channels with Online TV Live search

tab. Online TV Live boasts one of the largest databases of online live Internet TV stations, which is constantly being updated as new channels are brought out. Online TV live requires no additional computer equipment to play back Internet TV channels – all one needs is a PC with Windows Media Player installed.

Like some of the other software described earlier, Online TV Live lets one watch a multitude of programs and events. From the latest TV shows and sitcoms, to live sporting events including football and soccer, live basketball and major league baseball.

9.3.8 CoolStreaming

CoolStreaming [11] is based on P2PTV (peer-to-peer Television) technology, which makes it possible for the users to divide and share the bandwidth of the television channel being streamed with an Internet connection. The technology behind CoolStreaming is similar to that of BitTorrent. Channel content is both viewed and uploaded simultaneously by the viewers of the stream. CoolStreaming creates a local "stream" on the localhost and this stream is then shown through Windows Media Player, RealPlayer or any other media player. CoolStreaming is a data-centric design of peer-to-peer streaming overlay.

CoolStreaming uses intelligent scheduling algorithm that copes well with the bandwidth differences of uploading clients and thus minimizes skipping during playback. Moreover, it uses a swarm-style architecture that uses a directed graph based on gossip algorithms to broadcast content availability.

9.3.9 PPLive

PPlive [12] is another P2P streaming video network similar to CoolStreaming. PPLive software was written at Huazhong University's Science and Technology department in China. It combines P2P technologies and Internet TV, leading to what is called P2PTV. PPLive provides European and English Premiership football games, in addition to American football, basketball and baseball. One can also watch a huge number of other TV channels and stations through the PPlive software. Most of them are categorized under a variety of channel groups such as Movies, Music, TV series and live TV streaming.

9.4 Summary

Consumers would expect to see the same television channels that are available in cable and/or satellite television networks in addition to being able to access the video on demand service. The goal of this chapter was to estimate the resource requirements (storage and bandwidth) for a service provider to be able to provide broadcast television service. Analog television channels require a bandwidth of 8 MHz and each such channel can accommodate five digital television channels. Thus, to compete with a digital cable/satellite television service provider that replaces 100 analog TV channels with digital channels, a video-over-Internet service provider would have to serve 500 digital channels over the Internet, leading to an aggregate bandwidth requirement of approximately 2–3 gbps. Without using multicast technology, this bandwidth requirement is difficult, if not impossible, to meet. Storage requirements were also computed

for broadcast television service and it turned out to be approximately 1PB/month leading to requirements for techniques for periodic archiving or periodic purging of content. Technology to broadcast live TV stations over the Internet, namely CoolStreaming and GridMedia, based on a data-driven "pull" method and a hybrid "push-pull" method was described followed by some software products that need to be downloaded for watching live TV over broadband Internet connection.

References

[1] Cherry, S. (2005) The battle for broadband Internet protocol television. *IEEE Spectrum*, **42**(1), 24–29.

[2] Jain, R. (2005) I want my IPTV. *IEEE MultiMedia*, **12**(3), 96.

[3] Zhang, X., Liu, J., Li, B. and Yum, T.-S.P. (2005) *DONet/CoolStreaming: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming*. Proceedings of INFOCOM, Vol. 3, Miami, FL, USA, 13–17 March 2005, pp. 2102–2111.

[4] SatelliteDirect: http://site.mynet.com/burdurum/ (accessed June 9, 2010).

[5] DishTVforPC: http://www.dishtvforpc.com/ (accessed June 9, 2010).

[6] PPMate Streaming TV: http://ppmate-nettv.en.softonic.com/. (accessed June 9, 2010).

[7] Sopcast: http://www.sopcast.com/ (accessed June 9, 2010).

[8] 3webTotalTV and Radio Tuner: http://3webtotal-tv-radio-tuner.software.informer.com/4.1/ (accessed June 9, 2010).

[9] wwiTV: http://wwitv.com/ (accessed June 9, 2010).

[10] OnLine TV Live: http://www.live-online-tv.com/ (accessed June 9, 2010).

[11] CoolStreaming: http://webtv.CoolStreaming.com/ (accessed June 9, 2010).

[12] PPLive: http://pplive.en.softonic.com/ (accessed June 9, 2010).

[13] Guo, L., Chen, S., Ren, S. *et al.* (2004) *PROP: a Scalable and Reliable P2P Assisted Proxy Streaming System*. Proceedings of the ICDCS'04, Tokyo, Japan, Mar. 2004.

[14] Padmanabhan, V.N., Wang, H.J., Chou, P.A. and Sripanidkulchai, K. (2002) *Distributing Streaming Media Content Using Cooperative Networking*. Proceedings of the NOSSDAV'02, USA, May 2002.

[15] Jin, S. and Bestavros, A. (2002) *Cache-and-Relay Streaming Media Delivery for Asynchronous Clients*. Proceedings of the International Workshop on Networked Group Communication (NGC'02), Boston, MA, USA, Oct. 2002.

[16] Xu, D., Hefeeda, M., Hambrusch, S. and Bhargava, B. (2002) *On Peer-to-Peer Media Streaming*. Proceedings of the ICDCS'02, Wien, Austria, Jul. 2002.

[17] Cui, Y., Li, B. and Nahrstedt, K. (2004) oStream: asynchronous streaming multicast. *IEEE J. Select. Areas Comm.*, **22**: 91–106.

[18] Cui, Y. and Nahrstedt, K. (2003) *Layered Peer-to-Peer Streaming*. Proceedings of the NOSSDAV'03, Monterey, CA, Jun. 2003.

[19] Heffeeda, M., Habib, A., Botev, B. *et al.* (2003) *PROMISE: Peer-to-Peer Media Streaming Using CollectCast*. Proceedings of the ACM Multimedia (MM'03), Berkeley, CA, Nov., 2003.

[20] Guo, Y., Suh, K., Kurose, J. and Towsley, D. (2003) *P2Cast: Peer-to-Peer Patching Scheme for VoD Service*. Proceedings of the WWW'03, Budapest, Hungary, May 2003.

[21] Deshpande, H., Bawa, M. and Garcia-Molina, H. (2001) Streaming live media over peer-to-peer network. Technical Report, Stanford University.

[22] Tran, D.A., Hua, K.A. and Do, T.T. (2004) A peer-to-peer architecture for media streaming. *IEEE J. Select. Areas in Comm.*, **22**: 121–133.

[23] Banerjee, S., Bhattacharjee, B. and Kommareddy, C. (2002) *Scalable Application Layer Multicast*. Proceedings of the ACM SIGCOMM'02, Pittsburgh, PA, Aug. 2002.

[24] Banerjee, S., Lee, S., Bhattacharjee, B. and Srinivasan, A. (2003) *Resilient Multicast Using Overlays*. Proceedings of the ACM SIGMETRICS'03, San Diego, CA, USA, Jun. 2003.

[25] Chu, Y.-H., Rao, S.G. and Zhang, H. (2000) *A Case for End System Multicast*. Proceedings of the SIGMETRICS'00, Santa Clara, CA, USA, Jun. 2000.

[26] Ratnasamy, S., Handley, M., Karp, R. and Shenker, S. (2002) *Topologicallyaware Overlay Construction and Server Selection*. Proceedings of the INFOCOM' 02, New York, USA, Jun. 2002.

[27] Chu, Y.-H., Ganjam, A., Ng, T.S.E. *et al.* (2004) *Early Deployment Experience with an OVERLAY BASED INTERNET Broadcasting System*. USENIX Annual Technical Conference, Jun. 2004.
[28] Zhang, X., Zhang, Q., Zhang, Z. *et al.* (2004) A construction of locality-aware overlay network: mOverlay and its performance. *IEEE J. Select. Areas in Comm.*, **22**: 18–28.
[29] Liu, J., Li, B. and Zhang, Y.-Q. (2003) Adaptive video multicast over the Internet. *IEEE Multimedia*, **10**(1), 22–31.
[30] Castro, M., Druschel, P., Kermarrec, A.-M. *et al.* (2003) *SplitStream: High-Bandwidth Multicast in Cooperative Environments*. Proceedings of the ACM SOSP'03, New York, USA, Oct. 2003.
[31] Kostic, D., Rodriguez, A., Albrecht, J. and Vahdat, A. (2003) *Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh*. Proceedings of the ACM SOSP'03, New York, USA, Oct. 2003.
[32] Ganesh, A.J., Kermarrec, A.-M. and Massoulie, L. (2003) Peer-to-Peer Membership Management for Gossip-Based Protocols. *IEEE Transactions on Computers*, **52**(2): 139–149.
[33] Eugster, P., Guerraoui, R., Kermarrec, A.-M. and Massoulie, L. (2004) From epidemics to distributed computing. To appear in *IEEE Computer Magazine*.
[34] Yang, M. and Fei, Z. (2004) *A Proactive Approach to Reconstructing Overlay Multicast Trees*. Proceedings of the INFOCOM'04, Hong Kong, Mar. 2004.
[35] Hefeeda, M., Bhargava, B. and Yau, D.K.-Y. (2004) A Hybrid Architecture for Cost-Effective On-demand Media Streaming. *Computer Networks*, **44**(3).
[36] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001) *Introduction to Algorithms*, 2nd edn, MIT Press, Cambridge, MA.
[37] Shi, S. and Turner, J. (2002) *Routing in Overlay Multicast Networks*. Proceedings of the INFOCOM'02, New York, Jun. 2002.
[38] Banerjee, S. and Bhattacharjee, B. A comparative study of application layer multicast protocols. http://www.cs.wisc.edu/∼suman/pubs/compare.ps.gz (accessed June 9, 2010).
[39] Chawathe, Y. (2000) Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service. Ph.D. thesis. University of California, Berkeley. http://research.chawathe.com/people/yatin/publications/docs/thesis-single.ps.gz (accessed June 9, 2010).
[40] Rejaie, R. and Ortega, A. (2003) *PALS: Peer to Peer Adaptive Layered Streaming*. Proceedings of the NOSSDAV'03, Monterey, CA, USA, Jun. 2003.
[41] Zhang, X., Liuy, J., Liz, B. and Peter Yum, T.-S. (2005) *CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming*. Proceedings of IEEE Infocom 2005.
[42] Handley, M., Floyd, S., Pahdye, J. and Widmer, J. (2003) TCP Friendly Rate Control (TFRC): Protocol Specification. *RFC 3448*, January 2003.
[43] PlanetLab Website: http://www.planet-lab.org/ (accessed June 9, 2010).
[44] Zhang, M., Zhao, L., Tang, Y. *et al.* (2005) *Large-Scale Live Media Streaming over Peer-to-Peer Networks through Global Internet*. Proceeding of P2PMMS'05, November 11, 2005, Singapore.

10

Digital Rights Management (DRM)

The need for digital rights management (DRM) is paramount in a world where content (literature, music, videos, movies, artwork etc.) is available in digital form and hence can be distributed within seconds to any device anywhere around the globe. If copyrighted content were distributed without constraints, it would certainly benefit the consumers but would seriously affect the content creators/owners as they would not get the royalty they are entitled to. In order to address this fundamental need, DRM has been conceived and is being implemented [2–18]. The next section describes the functional architecture of DRM [1].

10.1 DRM Functional Architecture

Digital rights management functional architecture is shown in Figure 10.1 where the main functional blocks are:

- Intellectual property (IP) asset creation and capture.
- Intellectual property asset management.
- Intellectual property asset usage.

10.1.1 Intellectual Property Asset Creation and Capture

This functional block is responsible for determining how to manage the creation of digital content so that it can be easily traded. Specifically, it helps with asserting rights when content is first created (or reused and extended with proper rights) by various content creators. The steps involved in IP asset creation and capture are:

1. **Rights validation**: this is needed to ensure that content created from existing content includes rights to do so. For example, this would apply to remix songs that are derived from the original songs.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

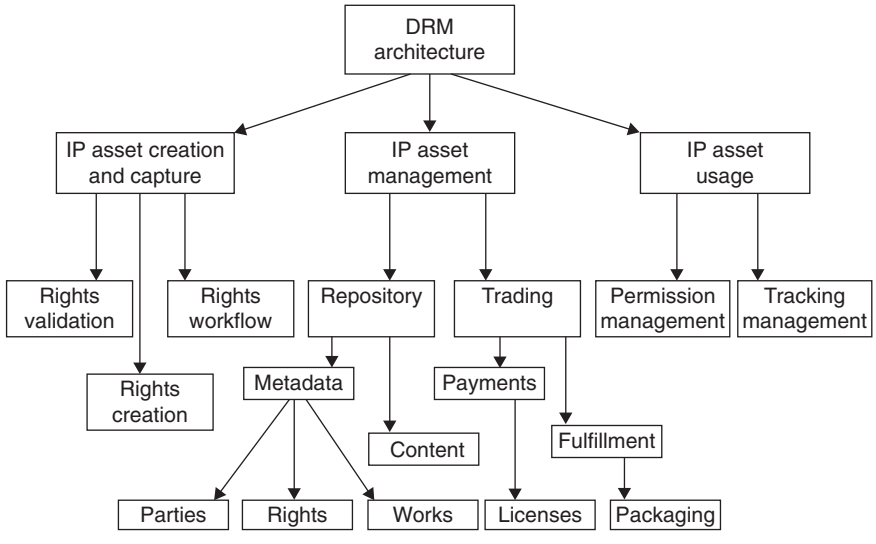


Figure 10.1 DRM functional architecture.

2. **Rights creation**: this is needed to assign rights to new content. For example, this would apply to newly created music video or a movie where the rights owners and allowable usage permissions need to be specified.
3. **Rights workflow**: this is needed to review and approve rights to content and the goal is achieved by allowing content to be processed through a series of workflow steps.

10.1.2 Intellectual Property Asset Management

This functional block is responsible for determining how to manage and enable trade of digital content. Specifically, it helps with accepting content from content creators into an asset management system and managing descriptive metadata and rights metadata. The main functions involved in IP asset management are:

1. **Repository functions**: these are needed to enable access to and retrieval of content and associated rights metadata from potentially distributed databases. Metadata consists of the parties involved, rights for each party and the works, which are nothing but various incarnation or instantiation of the original work. For example, the same movie (original work) may be instantiated in different languages and may be packaged in different ways, such as into a video CD or a DVD or a video cassette or even just a file on the computer. These instances are referred to as "works".
2. **Trading functions**: these are needed to enable the assignment of licenses to parties who have traded agreements for rights over content. They also ensure payment of royalties from licensees to the rights holders. Trading functions also manage rights for various instantiations of the digital media, namely, video CD or a DVD or a video cassette or even just a file on the computer.

10.1.3 Intellectual Property Asset Usage

This functional block is responsible for managing the usage of content once it has been traded. Specifically, it helps with supporting constraints/restrictions over traded content in specific devices/systems. The main functions involved in IP asset management are:

1. **Permission management**: this is needed to ensure that the rights associated with a digital media are honored. For example, if a user has the right to play a video on a specific mobile device, the permission management systems would ensure that the user is not able to play the video in a different mobile device or able to make a copy of the video.
2. **Tracking management**: this is needed to track the usage of digital media in order to ensure that the user abides by the rules agreed to as part of the license agreement. For example, if a user has a license to play a video five times, the tracking management systems would monitor the usage of the video and ensure that the user is not able to play the video more than five times.

10.2 Modeling Content in DRM Functional Architecture

From the content creator's perspective, the same content is usually instantiated in various items. Rights need to be assigned to each entity that can be traded and consumed by the users. Hence it is important to model content in the DRM functional architecture. Figure 10.2 shows that the original "work" can be realized or recreated in various expressions. Each expression can then be embodied in various "manifestations" which again can be instantiated in a variety of "items". Figure 10.3 shows a simple example to illustrate the concept. In the example, the original "work" is the movie *Titanic*. The "expressions" for the same original movie are "original movie in English", "Spanish version of the movie" and "Hindi version of the movie". The "original movie in English" is manifested in a "DVD" and a "digital file". Similarly the "Spanish version of the movie" is manifested in a "digital file" and the "Hindi version of the movie" is manifested in a "DVD", a "video cassette" and a "digital file". Furthermore, the "digital file" version of the "original movie in English" can be obtained

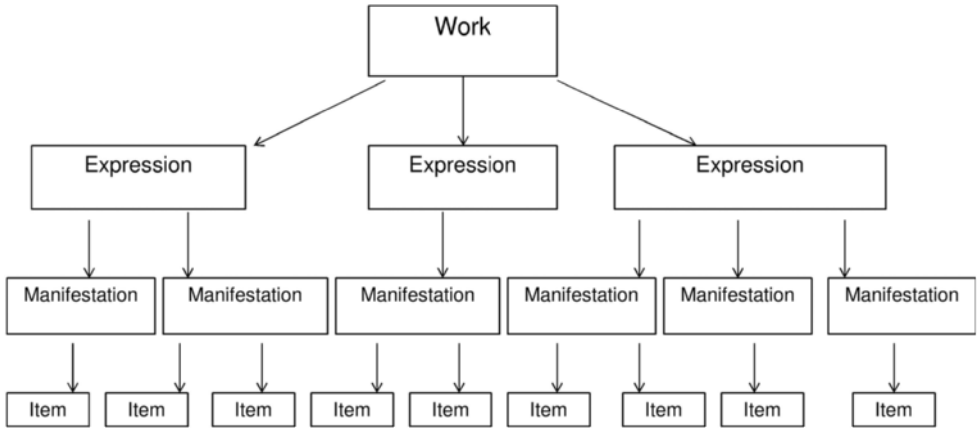


Figure 10.2 Content modeling in DRM functional architecture.

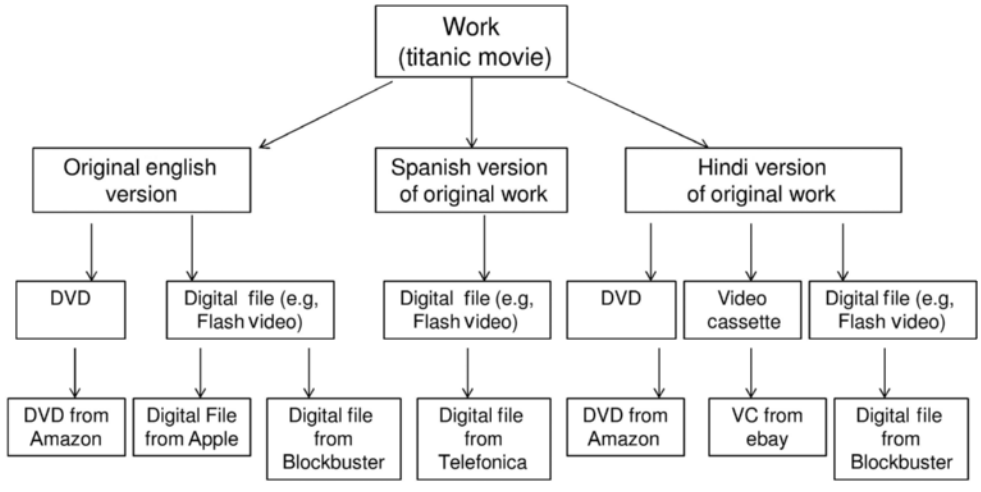


Figure 10.3 Example of content modeling.

either from Apple or from Blockbuster. Each entity (item) in the model should have rights associated with it and the DRM system should monitor the rights as the entity is either traded or consumed.

10.3 Modeling Rights Expression in DRM Functional Architecture

Rights are modeled using several parameters in order to capture various dimensions that characterize them. Specifically, rights expression consists of the following components (shown in Figure 10.4):

1. **Permissions**: this component specifies what one is allowed to do with the content. For example, one may be allowed to play a video but not allowed to copy.
2. **Constraints**: this component specifies the restrictions on the permission. For example, one may be allowed to play a video at most 10 times starting a given day on a specific device. Furthermore, the usage of the video may be restricted to the US only.

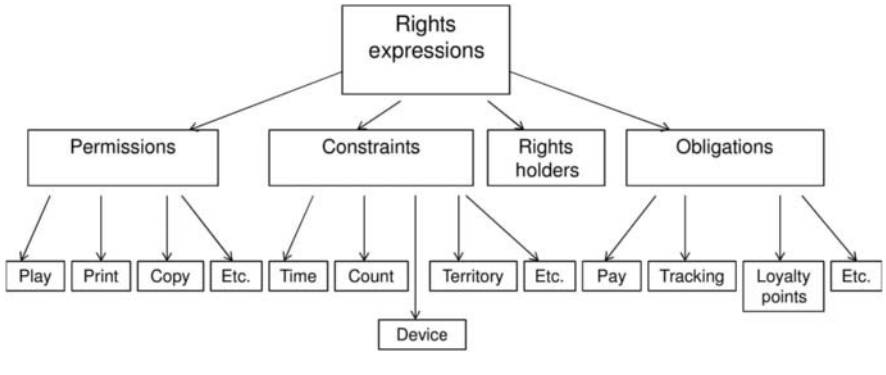


Figure 10.4 Rights expressions modeling in DRM functional architecture.

3. **Obligations**: this component specifies what one has to do or provide in order to have the permission to use the content. For example, one may have to pay $4.99 to play a video up to 10 times on his TV and agree to be tracked.
4. **Rights holders**: this component specifies the rights holders for a specific content and the distribution of royalty among them. For example, the royalty for a specific video may be shared among three rights holders, one entitled to 20%, the second one entitled to 30% and the third one entitled to 50%.

10.4 How DRM works

Digital rights management (DRM) involves various stakeholders: (i) content owner, (ii) content distributor and (iii) content consumer. Content owner's interest is to ensure that (s)he gets royalty when the content is consumed. In order to drive that interest, content owner does the following things: (i) encrypts content with a key, (ii) specifies to the content distributor the usage restrictions and corresponding payment terms for the content and (iii) provides the content encryption key to the content distributor. The content distributor, on receiving the above information from the content owner, creates a license containing the usage restrictions, payment terms and the key. Depending on the business model, it would be either the content owner or the content distributor who would upload the encrypted content on to a Web server (for download) or to a streaming server (for streaming). Content consumer is responsible for obtaining the license for the desired content before being able to play it. Next section describes some of these ideas in ore details.

10.4.1 Content Packaging

It is the responsibility of the digital rights manager to package digital content for distribution. The package consists of the encrypted content, the URL of the license server, and a partial encryption key. The URL of the license server is provided so that the media player knows where to obtain the license from for playing the content. The partial encryption key is provided so that the license server can compute the encryption key based on a secret it shares with the content owner and the partial encryption key obtained from the packaged digital content. This is shown in step 1 of Figure 10.5.

10.4.2 Content Distribution

The encrypted file would be placed in a Web server for download or a streaming server for streaming. This is shown in step 2 of Figure 10.5. Since the encrypted content and the license (which includes the partial encryption key) are decoupled, it is possible to distribute the encrypted content to anyone. The content cannot be played without acquiring the license and this mechanism safeguards the interest of the content owner.

10.4.3 License Distribution

The license for playing the media file is uploaded into a license server, which may be operated by an independent license clearing house. This is shown in step 3 of Figure 10.5. The role of

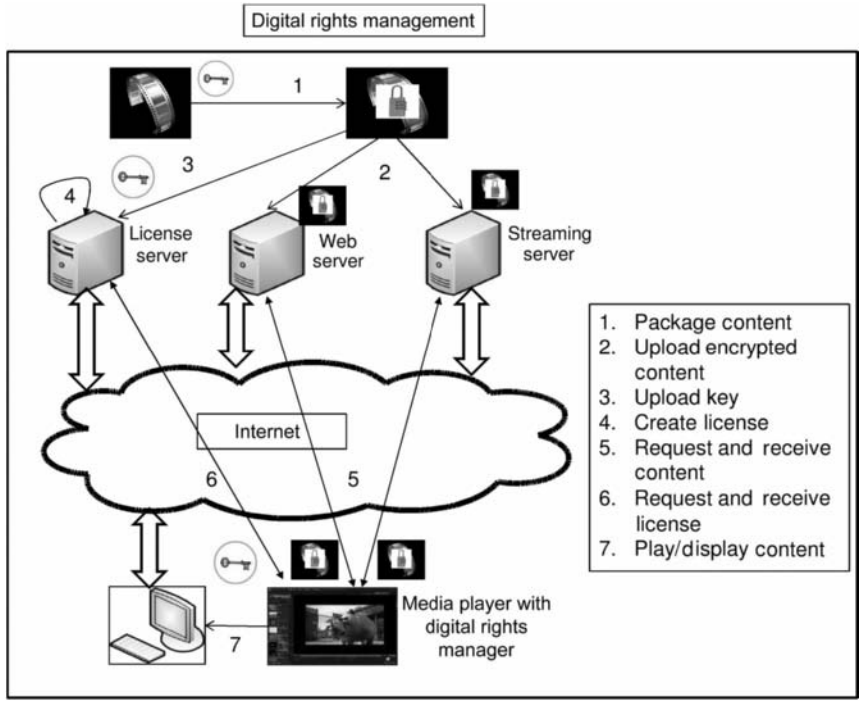


Figure 10.5 How DRM works.

the license server is to validate the request from a media player (client) before allowing it to download the license. In fact, the license server uses the partial encryption key from the media player together with the shared secret (between license server and content owner) to generate the "key" needed by the media player to decrypt the encrypted media file.

In addition to the decryption key, each license contains the rights and policies that govern the usage of the media file. For example, the license may contain a variety of business rules, including those addressing the following issues:

- On which devices the file can be played.
- To which devices the file can be transferred.
- If the file can be copied onto a CD.
- Starting time and ending (expiry) time for playing the file.
- How many times the file can be played.

10.4.4 License Creation and Assignment

Typically a merchant who sells content needs to create and assign licenses to each piece of content that he wants to sell. This process kicks in right after the merchant receives the encryption key from the content owner/creator (step 4 in Figure 10.5). Here are the steps:

1. **License definition creation**. License definition is created for content with multiple properties, such as, duration, count, copy allowed or not, expiration time and authentication module. If the authentication module specifies "remote authentication", then the user will be asked to provide credentials, such as, username/password and upon verification, will be issued the license.
2. **License Assignment**. After license definition is created, merchant can assign title and license to encrypted media. Contents are usually grouped into categories, such as, "trailers" or "advertisements" or "promotions" and are assigned different types of license. For example, trailers usually do not require user authentication and hence, the license to play the trailers is usually delivered "silently" to the user in the background. The reason for having the license in this case is to track usage of the content.

10.4.5 License Acquisition

In order to play a packaged digital media file, the media player must first acquire a license key to decrypt the encrypted media file. The process of acquiring a license begins automatically when the consumer attempts to acquire the protected content. Since the packaged content contains the URL of the license server, the media player knows exactly where to go to fetch the license. There may be different ways of handling the request for license. For example, the license may be automatically downloaded the first time the user tries to play the content. This may be prompted by a promotional use of the content. In other cases, the user may be redirected to a registration page where information is requested or payment is required, before retrieving the license from the license server. This is shown in step 6 of Figure 10.5.

10.4.6 Playing the Media File

After acquiring the license, the media player on the consumer's device can play the digital media file according to the rules and policies specified in the license. As described earlier, the rights of the consumer would be specified in the license. Typically, the license would specify the start times and dates, duration and the number of times the media may be played. Moreover, it may also specify the devices in which the media file may be played. In general, default rights may allow the consumer to play the digital media file on a specific computer and copy the file to a portable device. This is shown in step7 of Figure 10.5.

10.5 Summary

In this chapter, we covered a very important concept, namely digital rights management, which is driven mostly by the content owners to preserve the "ownership" of content and ensuring payment of "royalty" to the content creators based on use of respective content. Functional architecture of DRM was explained with its three main aspects: intellectual property, asset creation and capture; IP asset management and IP asset usage. Modeling of content and digital rights expression was also covered. Finally, the steps involving license creation, license distribution and license acquisition together with content packaging, content distribution and content display (media playing) were described to explain how DRM works in real life.

References

[1] Digital Rights Management. http://www.microsoft.com/windows/windowsmedia/forpros/drm/default.mspx (accessed June 9, 2010).

[2] Becker, E., Buhse, W., Günnewig, D. and Rump, N. (eds) (2003) Digital rights management: technological, economic, legal and political aspects. *Technological, Economic, Legal and Political Aspects Series: Lecture Notes in Computer Science*, **2770** (XI), 805. ISBN: 978-3-540-40465-1.

[3] Sander, T. (ed.) (2002) Security and privacy in digital rights management. ACM CCS-8 workshop DRM 2001, Philadelphia, PA, USA, November 5, 2001. *Revised Papers Series: Lecture Notes in Computer Science*, **2320** (X), 245. ISBN: 978-3-540-43677-5.

[4] Van Tassel, J.M. (2006) *Digital Rights Management: Monetizing and Protecting Content*, Focal Press, Elsevier Inc. ISBN: 978-0-240-80722-5.

[5] Zeng, W., Yu, H.H. and Lin, C.-Y. (eds) (2006) *Multimedia Security Technologies for Digital Rights Management*, Academic Press, Elsevier Inc. ISBN: 978-0-12-369476-8.

[6] Coyle, K. The Technology of Rights: Digital Rights Management. http://www.kcoyle.net/drm_basics.pdf (accessed June 9, 2010).

[7] Erickson, J.S. *et al.* (2001) Principles for Standardization and Interoperability in Web-based Digital Rights Management. A Position Paper for the W3C Workshop on Digital Rights Management (January 2001).

[8] Roscheisen, R.M. (1997) A Network-Centric Design For Relationship-Based Rights Management. Ph.D. Dissertation (Stanford University).

[9] Rust, G. and Bide, M. (2000) The Metadata Framework (June). http://www.doi.org/topics/indecs/indecs_framework_2000.pdf (accessed June 9, 2010) (accessed June 9, 2010).

[10] Kahn, R. and Wilensky, R. (1995) A Framework for Distributed Digital Object Services. http://www.cnri.reston.va.us/home/cstr/arch/k-w.html (accessed June 9, 2010).

[11] Iverson, V. et.al. (2001) MPEG-21 Digital Item Declaration WD (v1.0). ISO/IEC JTC 1/SC 29/WG 11/N3825 (January, Pisa, IT).

[12] MPEG (2001) Call for Requirements for Rights Data Dictionary and Rights Expression Language. http://xml.coverpages.org/RLTC-Reuters-Reqs.pdf (accessed June 9, 2010).

[13] Iannella, R. Open Digital Rights Language Specification v0.8. http://odrl.net/ODRL-08.pdf (accessed June 9, 2010).

[14] Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/ (accessed June 9, 2010).

[15] Resource Description Framework (RDF) Model and Syntax Specification, 1999. http://www.w3.org/TR/REC-rdf-syntax/ (accessed June 9, 2010).

[16] Erickson, J.S. (2001) Information objects and rights management: A mediation-based approach to DRM interoperability. *D-Lib Magazine*, 7 (4). ISSN 1082-9873.

[17] Gunter, C.A., Weeks, S.T. and Wright, A. (2001) *Models and Languages for Digital Rights*. InterTrust Star Lab Technical Report STAR-TR-01-04, March. http://www.star-lab.com/tr/star-tr-01-04.pdf (accessed June 9, 2010).

[18] Iannella, R. (2001) Digital rights management (DRM) architectures. *D-Lib Magazine*, 7 (6). ISSN 1082-9873.

11

Quality of Experience (QoE)

Quality of experience (QoE) is the most important aspect of video distribution and delivery to end users. In the context of open networks there are several techniques to improve QoE.

For a content delivery network (CDN) service provider, QoE can be improved by: (i) increasing the number of mirror sites such that end user has higher likelihood of accessing video from a nearby mirror server and/or (ii) increasing the caching storage per mirror server so that it increases the likelihood of the requested video to be present at the mirror site closest to the end-user.

For a P2P service the provider perspective, QoE can be improved by: (i) picking appropriate peers such that the peers are closer to the requesting end-user and have the best possible bandwidth for delivering video, (ii) smart load balancing such that peers with fewer feeds are chosen to avoid overloading, (iii) creating multirate encoded videos, dividing them into chunks (similar to techniques described earlier) and enabling clients to pick up the right chunks depending on their available bandwidth.

As a concrete example of how quality of experience can be improved for video streaming in wireless networks, the design of a caching system is presented next as a case study.

11.1 QoE Cache: Designing a QoE-Aware Edge Caching System

The QoE cache can be thought of as the building block of a CDN, which is an overlay network designed to improve QoE for end users when accessing rich media portals either for downloading content or for streaming video over the cellular network.

In order to improve QoE of mobile users, the QoE cache system performs several functions:

- It moves content closer to the mobile user by caching content locally (Web proxy/cache).
- It optimizes TCP to deal with variable round-trip time, time-varying packet error rate and potentially high packet error rate during handoff or poor RF condition (TCP optimizer).
- It optimizes streaming by adapting media stream transmission to the condition of the wireless network (streaming optimizer).
- It eliminates DNS queries over the air link (DNS Optimizer).

A schematic diagram of the QoE cache is shown in Figure 11.1.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

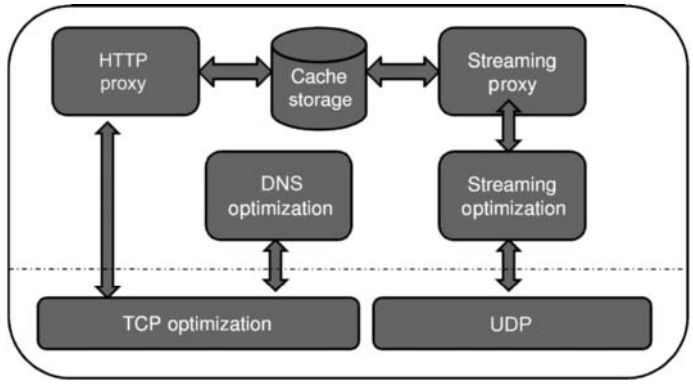


Figure 11.1 The QoE cache functional diagram.

11.1.1 TCP Optimizer

The TCP optimizer (T-Opt) is a kernel data-path component. It implements a version of TCP that uses the packet flow between the TCP end points to estimate the end-to-end available bandwidth and adjust its sending rate (congestion window) according to that [45–48]. The TCP performance improvement is obtained using a variety of techniques, such as:

- Reducing the round-trip time between the mobile handset and the content server.
- Decoupling error control from flow control, meaning that the congestion window of TCP will not be blindly cut to half on packet loss rather will be set based on the estimated bandwidth delay product.
- Increasing slow start threshold to allow exponential growth of window for a longer period of time than usual, and delay congestion avoidance phase, which equates to linear increase rather than exponential increase of congestion window.

11.1.2 Streaming Optimizer

Streaming optimizer (S-Opt), a Linux User Space service, will implement rate adaptation of video based on the packet loss rate at the receiver as indicated by the RTCP feedback messages [1–4, 7, 11]. Essentially, S-Opt acts as an intermediary in the data path handling RTP/UDP [31] packets rather than in the control path. Streaming performance improvement will be obtained using a variety of techniques, such as:

- Switching to a lower transmission rate when packet loss rate at a receiver exceeds a high threshold (T_h) and switching to a higher transmission rate when packet loss rate at the receiver falls below a low threshold (T_l).
- Maintaining the transmission rate when packet loss rate is in between T_h and T_l.
- Retransmitting lost packets if the probability of the retransmitted packet reaching the mobile before the deadline is high.
- Using "optimized" TCP to fill the playout buffer quickly at mobile to improve start-up latency.

- In addition, S-Opt implements a TCP-friendly congestion control mechanism that shares the available bandwidth fairly with competing TCP connections. In fact, the TCP-friendly congestion control mechanism (such as TFRC) is used to determine the data rate for the media stream.

11.1.3 Web Proxy/Cache

The Web proxy/cache (WPC) acts as an intermediary between the mobile handset-based Web browser and the origin server. In effect, WPC implements HTTP 1.0 protocol including all the rules and regulations specified in the HTTP 1.0 protocol to deal with caching directives. For example, if there is a "no cache" directive in the HTTP header from the origin server, WPC should not cache the content contained in the payload of the HTTP packet. In general, a WPC works as follows:

1. The WPC receives the Web browser's HTTP request message. Note that there are both transparent and nontransparent ways of redirecting client request to WPC.
2. The WPC checks for availability of requested content in its local cache, and if available, serves it from local cache as opposed to going back to the origin server to serve the content.
3. If the content is not stored locally and/or the content stored locally is not up to date, the WPC goes to the origin server and fetches and stores the content locally, while at the same time it forwards the content to the browser on the mobile handset.

11.1.4 Streaming Proxy/Cache

The streaming proxy/caches (SPCs) implement a media proxy, which acts as an intermediary between the mobile handset and the origin streaming server. The SPC performs the following functions:

1. Proxies the RTSP protocol [32] between the media client and the streaming server.
2. Caches media streams at the local storage and retrieves cached content on an as-needed basis. Note that the same media object may be available at the origin server encoded in multiple bitrates, and in that case, the SPC fetches multiple versions of the same media object and caches it locally so that it can serve off the right version based on the estimated available bandwidth at the mobile access terminal (AT).
3. Traps the RTCP [31] feedback packets and passes them over to the streaming optimizer (S-Opt) to decide on the optimal transmission rate.
4. Uses the S-Opt on the data path for adapting the media transmission rate.

11.1.5 DNS Optimizer

DNS optimizers (D-Opts) eliminate DNS queries over the air, thereby reducing the web-page download time. The DNS optimizer also helps improve streaming QoE by reducing the start-up delay of a video clip. The D-Opt works in unison with the WPC and SPC as follows:

1. The WPC (SPC) receives the initial page (index.html) from the origin server and passes it on to the D-Opt.

2. The D-Opt replaces each embedded URL in the index.html page with a new URL that has the IP address of the WPC (SPC) as the prefix and sends it back to WPC (SPC).

   Upon receiving the modified index.html, WPC (SPC) sends it to the requesting client on the mobile handset.

   Since the embedded URLs in the index.html page have an IP address as a prefix, the DNS client at the handset will not generate any request over the air.
3. The D-Opt, in the background, resolves the DNS names into the corresponding IP addresses and sends the original index.html page to WPC (SPC) where the embedded URLs will have the names of the origin servers replaced with their corresponding IP addresses.

   Thus the DNS resolution is done over the wired network by D-Opt and relieves the DNS client at the mobile handset from doing the same over the air.
4. The WPC/SPC, on receiving, the modified index.html page, establishes TCP connections with the servers identified by the IP addresses in the URL, retrieves content from those servers, and stores them in the local cache.
5. When the client request arrives for the embedded URLs at the WPC/SPC, WPC/SPC will serve the content (retrieved in step 4) from the local cache.

There are two key ideas at work. First, response time is reduced by eliminating time-consuming DNS requests over the air. Such requests are resolved over the wired network. Second, response time is further reduced by serving requested content from the cache. Content is fetched by WPC/SPC from multiple origin servers in parallel and stored in the local cache.

11.1.6 TCP Optimizer (Details)

A wireless-link-aware TCP protocol provides lower-latency and higher TCP throughput for a user by using information contained in the so-called acknowledgment (ACK) packets to modify TCP slow start and congestion avoidance/flow control mechanisms. This would result in faster downloads of Web pages, ring tones, short video clips, images, and so on.

TCP is an end-to-end protocol that uses trial and error to estimate the available bandwidth in a connection. The amount of data that the TCP sender pumps into a connection during a round trip time (RTT) depends on the minimum of the buffer availability at the TCP Receiver (commonly known as the available window, or "awnd") and the capacity of the network (commonly known as the congestion window, or "cwnd"). In the beginning of its transmissions, TCP goes through a process called "Slow Start" in which the TCP sender probes the network capacity by sending one segment worth of data (i.e., the initial value of cwnd) to the receiver. With each positive acknowledgment obtained from the receiver, TCP increments cwnd, effectively sending twice the amount of data, and keeps repeating the process until cwnd reaches what is called the "slow start threshold" (ssthresh), at which point TCP enters what is called the congestion avoidance phase. In the congestion avoidance phase, cwnd grows linearly as opposed to the exponential growth in the slow start phase. In other words, a higher slow start threshold (ssthresh) would mean a longer duration for the exponential (faster) growth of the TCP's window resulting in transfer of larger amount of data within a given duration of time. Furthermore, when the TCP detects congestion, it blindly cuts the cwnd to half of what it was before the detection of congestion. This is done as a conservative measure in the absence of specific information about the state of end-to-end connection. However, in wireless networks, many a time, packets are lost due to spurious reasons rather than due to congestion and as a result cutting the congestion

window blindly by a factor of two is not necessarily the right thing to do. Ideally, the cwnd should be adjusted to the estimated available bandwidth and that would mean the ability to push more bytes within a given duration of time, resulting in higher TCP throughput and hence faster download.

Thus, if it takes the TCP "n" round trips to reach the slow-start threshold (ssthresh), the amount of data that would be pushed during this time would be $2^{(n-1)}$ segments. Naturally, for each additional round trip time it takes the TCP to reach ssthresh, the TCP can push double the amount of data than it would otherwise. In addition, for each random loss, the TCP would halve the amount of data it pushes during a round trip time, while a wireless-link-aware variation of the TCP would hardly change the amount of data it pushes during the round-trip time. This implies, a wireless-link-aware variation of TCP would push more data during a given duration improving TCP's throughput and hence making downloads much faster. Assuming that the packet losses are random, the benefit of optimizing TCP along the lines described above would be significant.

These TCP enhancements are described below in a step-by-step manner.

11.1.6.1 Drawback of TCP (New) Reno

TCP (New) Reno "blindly" cuts Congestion Window (cwnd) to half on receiving 3 duplicate ACKs. This is too aggressive for wireless networks where the loss is spurious rather than due to buffer overflow. As a result of this aggressive adjustment of cwnd, available bandwidth is wasted reducing TCP throughput slowing down "downloads".

Therefore, the goal is to estimate "available" bandwidth using TCP ACK arrival rate at the sender and use that info to set the TCP's congestion-control parameters.

11.1.6.2 Key Ideas in TCP Westwood[45–48]

- Modify "congestion control" algorithm of TCP (New) Reno.
- **Step-1:** Estimate bandwidth available:
  - Available Bandwidth Estimate (ABE) is based on the amount of data transferred in between successive ACK reception at the sender.
- **Step-2:** Use ABE to set congestion control parameters of TCP:
  - Slow Start Threshold (ssthresh) and Congestion Window (cwnd).
  - TCP is able to utilize available bandwidth of the pipe, thereby resulting in higher TCP throughput and faster "downloads".
- Everything else remains unchanged in TCP (New) Reno.

11.1.6.3 Congestion Avoidance Algorithm

```
•  if (3 Duplicate ACKs are received)
   -     ssthresh = (ABE * RTTmin) / seg_size;
   -     if (cwnd > ssthresh)
         •     cwnd = ssthresh;
   -     endif
•  endif
•  seg_size refers to the length of TCP segment in bits
```

11.1.6.4 Slow Start Algorithm (On Timeout)

- if (timeout)
  - ssthresh = (ABE * RTT_{min}) / seg_size;
  - if (ssthresh < 2)
    - ssthresh = 2;
  - endif
  - cwnd = 1
- endif
- seg_size refers to the length of a TCP segment in bits

11.1.6.5 Estimation of Available Bandwidth

- cumul_ack = cur_ack_seqno - prev_ack_seqno;
- if (cumul_ack = 0)
  - accounted = accounted + 1;
  - cumul_ack = 1;
- endif
- if (cumul_ack > 1)
  - If (accounted >= cumul_ack)
    - accounted = accounted - cumul_ack;
    - cumul_ack = 1;
  - else if (accounted < cumul_ack)
    - cumul_ack = cumul_ack - accounted;
    - accounted = 0;
- endif
- prev_ack_seqno = cur_ack_seqno;
- acked = cumul_ack;
- prev_bw = cur_bw;
- prev_filtered_bw = cur_filtered_bw;
- delta_timestamp = cur_timestamp - prev_timestamp;
- cur_bw = acked / delta_timestamp;
- tau = 1; alpha = [2 * tau/delta_timestamp]
- cur_filtered_bw = [(alpha-1)/(alpha+1)] * prev_filtered_bw
  + [1/(alpha+1)] * (cur_bw + prev_bw)
- **ABE = cur_filtered_bw**

11.1.7 Streaming Optimizer (Details)

Streaming optimizer, a Linux User Space service, (S-Opt) will implement rate adaptation of video based on the packet loss rate at the receiver as indicated by the RTCP feedback messages [31]. Essentially, S-Opt acts as an intermediary in the data path handling RTP/UDP packets rather than in the control path. Streaming performance improvement can be obtained by using various algorithms. Two representative algorithms are described below:

11.1.7.1 Algorithm-1 (Simple)

1. The streaming optimizer calculates the optimal encoded rate to use from the current rate-set (higher, lower, or keep the same) based on packet error rate (PER) thresholds:

a. If the PER is below a minimum provisioned threshold (T1), select the next higher encoded rate from the current rate-set. If the highest encoded rate is already selected, maintain the current encoded rate.
b. If the PER is above a maximum provisioned threshold (T2), select the next lowest encoded rate from the current rate-set. If the lowest encoded rate is already selected, maintain the current encoded rate.
c. If the PER is between the minimum and maximum provisioned thresholds, maintain the current encoded rate.

2. Repeat step 1 as long as the streaming session is active.

11.1.7.2 Algorithm-2 (Complex)

1. The streaming optimizer calculates the optimal encoded rate to use from the current rate-set (higher, lower, or keep the same) based on packet error rate (PER) thresholds:
   a. If the PER is below a minimum provisioned threshold (T1), select the next higher encoded rate from the current rate-set. If the highest encoded rate is already selected, maintain the current encoded rate.
   b. If the PER is above a maximum provisioned threshold (T2), select the next lowest encoded rate from the current rate-set. If the lowest encoded rate is already selected, maintain the current encoded rate.
   c. If the PER is between the minimum and maximum provisioned thresholds, compare the rate-of-change in the PER with a third or fourth provisioned threshold, depending on the direction of the rate-of-change:
      i. If a positive rate-of-change occurs in the PER (i.e., if $(PER_{tnow} - PER_{tlast}) > 0$) and if this rate-of-change fails to exceed a third provisioned threshold (T3), maintain the current encoded rate. That is, if:
         1. $(PER_{tnow} - PER_{tlast}) < T3$, maintain the current encoded rate.
      ii. If the positive rate-of-change in the PER equals or exceeds the third provisioned threshold (T3), select the next lowest encoded rate from the current rate-set. That is, if
      iii. $(PER_{tnow} - PER_{tlast}) >= T3$, select the next-lowest encoded rate from the current rate-set.
      iv. If a negative rate-of-change occurs in the PER (i.e., if $(PER_{tnow} - PER_{tlast}) < 0$) and if this rate-of-change fails to exceed a fourth provisioned threshold (T4), maintain the current encoded rate. That is, if
      v. $(PER_{tlast} - PER_{tnow}) < T4$, maintain the current encoded threshold.
   d. If the negative rate-of-change in the PER equals or exceeds the fourth provisioned threshold (T4), select the next highest encoded rate from the current rate-set. that is, if
      i. $(PER_{tlast} - PER_{tnow}) >= T4$, select the next-highest encoded rate from the current rate-set.
2. The streaming optimizer makes a decision on the rate-set (higher, lower, or keep the same) to use based on the long-term average of the optimal encode rate. The long-term average is calculated using the current value of the selected rate and the previous value of the LTA, plus a fifth provisioned parameter, T5. The long-term average is calculated as:
   a. $LTA_{tnow} = (T5)^*(current\_rate) + (1\text{-}T5)^*(\%LTA_{tlast})$.

b. As the long-term average approaches the maximum encoded rate within the current rate-set, select the next rate-set with a higher set of rates. If the rate-set with the highest rates is already selected, maintain the current rate-set. Let the upper_rate_set_proximity_threshold be T6. Then

```
i.    if (LTA_tnow > T6*currentRateSet[maxRate]) {
      1.    if (there is a higher rate set) {
            a.    currentRateSet =
                  next_higher_rate_set;
            b.    current_rate =
                  next_higher_rate_set[minRate];
      2. }
ii.   }
```

c. As the long-term average approaches the minimum encoded rate within the current rate-set, select the rate-set with the next- lowest rates. If the rate-set with the lowest set of rates is already selected, maintain the current rate-set. Let the lower_rate_set_proximity_threshold be T7. Then,

```
i.    if (LTA_tnow < (1 + T7)*currentRateSet[minRate]) {
      1.    if (there is a lower rate set) {
            a. currentRateSet = next_lower_rate_set;
            b. current_rate =
               next_lower_rate_set[maxRate];
      2.    }
ii.   }
```

3. If it is desirable to have the same proximity deviation for the upper and lower tests, then T7 = (1 – T6).
4. Repeat steps 1 and 2 as long as the streaming session is active.

11.1.8 Web Proxy/Cache (Details)

The Web proxy/cache (WPC) will act as an intermediary between the mobile handset-based Web browser and the origin server. In effect, the WPC will implement HTTP 1.0 protocol including all the rules and regulations specified in the HTTP 1.0 protocol to deal with caching directives. For example, if there is a "no cache" directive in the HTTP header from the origin server, the WPC will not cache the content contained in the payload of the HTTP packet. In general, the WPC will work as follows:

1. The mobile client generates a request to access a site http://xyz.com. DNS client resolves xyz.com into its corresponding IP address 123.456.789.123 and the http request is directed to the web server at IP address 123.456.789.123.
2. The TCP SYN packet represented by the 5-tuple [src_ip = mobile client's ip addr, src_port = *, dst_ip = 123.456.789.123, dst_port = 80, protocol = TCP] matches the filter at the L4 switch and the L4 switch GRE tunnels the packet to the QOE-CACHE platform, specifically to the WPC component of the QOE-CACHE platform.

3. WPC de-tunnels the packet and responds with a SYN ACK packet in which it fakes the src_ip address to be 123.456.789.123 and establishes the TCP connection with the mobile client.
4. After the TCP connection is established between the mobile client and the WPC, the mobile client sends an HTTP GET. The WPC checks if the requested object is in its cache and if so, it returns the requested object to the mobile client. If not, it sends the request to the origin server 123.456.789.123, retrieves the content, caches it and sends the response back to the mobile client.

   Note that every packet sent by the WPC to the mobile client has the origin server's IP address (123.456.789.123) as the source IP address. In addition, that IP packet will be encapsulated in a GRE tunnel with the IP address of the WPC as the source IP and the L4 switch's IP as the destination IP address for the tunnel.
5. L4 switch de-tunnels the packet and forwards the response from the WPC to the mobile client.

11.1.9 Streaming Proxy/Cache (Details)

The Streaming Proxy/Cache (SPC) will implement a Windows Media Proxy [11–30, 33–36], which will act as an intermediary between the mobile handset and the origin streaming server [1–9]. The SPC will perform the following functions:

1. The mobile client generates a request to access a media object rtsp://xyz.com. DNS client resolves xyz.com into its corresponding IP address 123.456.789.123 and the rtsp request is directed to the media server at IP address 123.456.789.123.
2. The TCP SYN packet represented by the 5-tuple [src_ip = mobile client's ip addr, src_port = *, dst_ip = 123.456.789.123, dst_port = 554, protocol = TCP] matches the filter at the L4 switch and the L4 switch GRE tunnels the packet to the QOE-CACHE platform, specifically to the SPC component of the QOE-CACHE platform.
3. SPC de-tunnels the packet and responds with a SYN ACK packet in which it fakes the src_ip address to be 123.456.789.123 and establishes the TCP connection with the mobile client.
4. After the TCP connection is established between the mobile client and the SPC, the mobile client sends an RTSP DESCRIBE. WPC checks if the requested object is in its cache and if so, it returns the Description of the media object to the mobile client. If not, it sends the request to the origin server 123.456.789.123, retrieves the content encoded in multiple bitrates, caches them and sends the Description of the media object back to the mobile client.

   Note that every packet sent by the SPC to the mobile client has the origin server's IP address (123.456.789.123) as the source IP address. In addition, that IP packet will be encapsulated in a GRE tunnel with the IP address of the SPC as the source IP and the L4 switch's IP as the destination IP address for the tunnel.
5. The L4 switch de-tunnels the packet and forwards the response from the SPC to the mobile client.
6. The SPC responds to subsequent RTSP commands, such as SETUP and PLAY, and uses the streaming optimizer to decide at what rate it should stream the requested content to the mobile client.

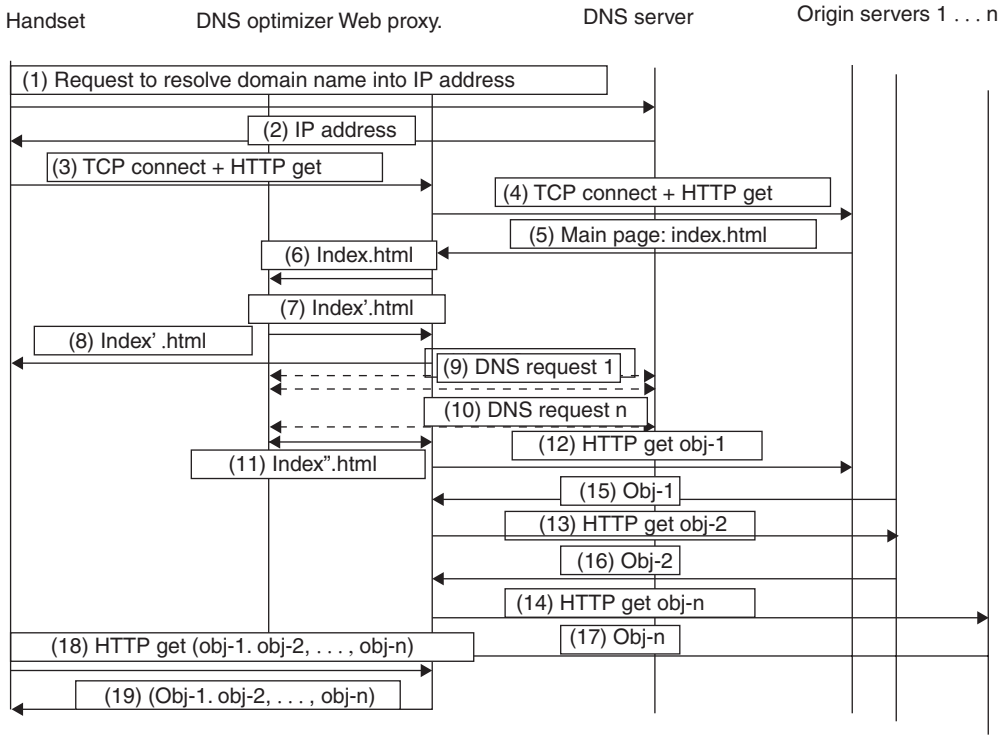


Figure 11.2 Message flow for DNS optimization.

7. The SPC will also trap RTCP packets from the mobile client and pass that information to the S-Opt so that the S-Opt can determine the optimal streaming rate based on the packet loss information contained in the RTCP feedback packets (using the algorithm described in Section 11.1.7).

11.1.10 DNS Optimizer (Details)

The DNS optimizer (D-Opt) will eliminate DNS queries over the air thereby reducing the web-page download time [44]. The DNS optimizer will also help improve streaming QoE by reducing the start-up delay of a video clip. D-Opt will work in unison with the Web proxy/cache (WPC) and streaming proxy cache (SPC) as follows (see Figure 11.2):

1. The DNS resolver in the handset sends a DNS query to DNS Server to resolve the origin server's name to its IP address.
2. The DNS server returns the IP address of the corresponding origin server.
3. The Web browser tries to establish a TCP connection with the origin server, but the L4 switch transparently sends the TCP connection request (TCP SYN packet) to the Web proxy. The Web proxy responds to the TCP connection, thereby establishing a TCP connection between the web browser and the Web proxy. The Web browser then sends an HTTP GET for the main page to the Web proxy.

4. The Web proxy checks if the requested object is in its cache and, if not, establishes a TCP connection with the origin server and sends an HTTP GET for the main page to the origin server.
5. The origin server returns the index.html page to the Web proxy.
6. The Web proxy forwards the index.html page to the DNS optimizer.
7. The DNS optimizer replaces each embedded URL in the index.html page with a new URL that will have the IP address of the Web proxy as the prefix and send it back to the Web proxy.

   For example, an embedded URL http://xyz.com/images/image1.jpg will be replaced with http://mmm.mmm.mmm.mmm/xyz.com/images/image1.jpg where mmm.mmm.mmm.mmm is the IP address of the Web proxy:
8. Upon receiving the modified index.html, the Web proxy will send it to the Web browser on the mobile handset.
9. The DNS optimizer, in the background, resolves the DNS names into the corresponding IP addresses.
10. Same as (9).
11. The DNS optimizer sends the original index.html page to the Web proxy where the embedded URLs will have the names of the origin servers replaced with their corresponding IP addresses.

    For example, an embedded URL http://xyz.com/images/image1.jpg will be replaced with http://nnn.nnn.nnn.nnn/images/image1.jpg where nnn.nnn.nnn.nnn is the IP address corresponding to the origin server xyz.com.
12. The Web proxy, on receiving the modified index.html page, establishes TCP connections with the servers identified by the IP addresses in the URL and sends HTTP GET to fetch the corresponding content.
13. Same as (12), happening in parallel.
14. Same as (13), happening in parallel.
15. The Web proxy retrieves content from those servers and stores them in the local cache.
16. Same as (15) happening in parallel.
17. Same as (16) happening in parallel.
18. The Web browser at the handset sends HTTP request for the embedded URLs to the Web proxy.
19. The Web proxy retrieves the content fetched in steps (15)–(17) from the cache and serves the content to the Web browser.

The message flow between the various components of the QOE-CACHE platform is shown in diagram below.

11.2 Further Insights and Optimizations for Video Streaming over Wireless

Multimedia streaming (based on the RTSP/RTP protocol) is likely to become one of the most important services for mobile terminals in the near future. Providing satisfactory quality of experience to end users is a challenge because of the time-varying bandwidth, error rate and round-trip time of the air interface and the limited capabilities of the mobile terminal as have been described in earlier sections.

The main reason behind impairment of video quality of experience in mobile networks involves buffers within the mobile operator network (e.g. SGSN/RNC buffers). Unlike Internet routers that allocate shared buffer space for multiple users, mobile networks reserve separate storage capacity for each user connection. Therefore congestion due to traffic generated by other users is not a concern. However, since the wireless link is typically the bottleneck in an end-to-end connection, packets tend to gather in this buffer leading to an overflow condition, especially in extreme situations, such as handoffs. Such problems are usually dealt with by a properly designed transmission control at the server that adapts transmission rate based on both play-out status at the client and the channel conditions, thereby providing efficient client buffer management in terms of avoiding underflow and overflow, and doing effective network buffer congestion control.

11.2.1 QoE Cache Enhancement Insights

The solution focuses on solving the above mentioned problem using a three-step approach:

1. **Step 1**: Estimates network condition based on RTCP feedback. Essentially ABE (available bandwidth estimate), PER (packet error rate) and RTT (round trip time) are estimated. This helps the network estimation to be client agnostic, as long as the client conforms to IETF RFC 3550.
2. **Step 2**: Source encodes the media file into MBR (multiple bit rate) format. It essentially means that the file is composed of more than one CBR (constant bit rate) streams for each media track. For simplicity it is assumed that media file will contain total of two tracks. One will be video track and other will be audio track. The term "source encoding" is used to mean that media file is encoded into MBR format even before receiving a request to stream the media.
3. **Step 3**: Provides a framework to seamlessly switch from one CBR stream to another on detection of change in network condition.

11.2.2 Functional Enhancements to the Basic QoE-Cache

The basic QoE Cache is an implementation instance of the design (Section 11.1) focused on leveraging proxy, cache and network-optimized streaming capability for Windows Media Streams.

Detailed analysis of Windows Media streaming reveals that Windows Media Server uses a modified version of RTSP and RTP for streaming, which is understood only by Windows Media Player.

Also, there exists a mechanism for Video Rate Adaptation between Windows Media Player and Windows Media Server. The Windows Media Service Framework provides capability for Step 1 and Step 3 mentioned in Section 11.2.1. The mechanism works only if the media in cache has already been source encoded into MBR format.

In case the media is encoded in MBR format, the mechanism works as follows:

1. The Windows Media Server provides details of all CBR streams in media description. An RTSP Streaming Server will send the media description in response to RTSP DESCRIBE request from the RTSP Streaming Client.

2. The Windows Media Client has a proprietary algorithm to estimate network condition.
3. Based on the estimated network condition and set of available CBR streams (notified in the media description), the Windows Media Client requests a specific CBR stream for each media track (audio and video).
4. The Windows Media Server, on receipt of this request switches to the newly requested CBR stream.

To benefit from this already existing video rate adaptation mechanism, the streaming proxy cache (SPC) component in the QoE cache for Windows Media Streams can be enhanced to have media transcoding capability. The Windows Media Encoder can be used for the transcoding process.

The following events happen in the process of transcoding cached media to the MBR format:

1. SPC (based on Windows Media Service Framework) generates a 'CACHE COMPLETE' event to signify the completion of storing a media file in the cache.
2. The above mentioned event is caught and a transcoding process is started to convert the cached media to MBR format. A transcoding profile is also created with properly chosen bit rate to cover broad range of network scenario (from 1xRTT to EVDO Rev-A).
3. When the MBR encoding of the specified media is complete, the cache manager is notified to use the transcoded file for serving the client.

Windows Media Player requests for "stream acceleration" for initial few seconds of streaming. This is to ensure that the Windows Media Player quickly fills up the jitter buffer. However, this may result in poor quality if available network throughput is lower than the accelerated stream bandwidth.

To remove this drawback the following technique can be added in the streaming optimizer (SOPT) component of the QoE cache platform for Windows Media Streams:

1. Calculate RTT for packets being exchanged between Windows Media Server and Windows Media Client during packet pair experiment. This experiment takes place during RTSP negotiation before RTSP SETUP is sent by the Windows Media Player.
2. If this RTT is higher than a threshold (calculated based on empirical data), SOPT modifies the RTSP PLAY request (going from Windows Media Stream to Windows Media Server) to remove the acceleration parameter.
3. Windows Media Server then transmits the stream at rate it was encoded.

11.2.3 Benefits Due to Basic QoE Cache

In the case of '**CACHE-HIT**' the following benefits were measured statistically in an implementation instance of the QoE cache:

- around 20% reduction on PER on client side;
- around 40% reduction in the initial startup time;
- around 20% improvement in Average Stream Bit Rate (this means that client received more number of packets successfully) observed at the client;
- visible QoE improvements when media streamed through the QoE cache.

11.2.4 Functional Enhancement to Generic QoE Cache

The generic QoE cache is an implementation instance of the QoE cache (including streaming proxy, streaming cache and network optimized streaming capability) for non-Windows Media Streams.

To provide network optimized streaming all the steps mentioned in Section 11.2.1 need to be designed and developed in the QoE cache so that together they provide visible QoE improvements in RTSP-based streaming.

11.2.4.1 Network Estimation

The following parameters need to be estimated from the RTCP feedback.

11.2.4.1.1 Available Bandwidth Estimate (ABE)

The value of this parameter is useful to select appropriate CBR stream from the available CBR stream set in situations when actual network throughput has gone lower than streaming rate. In other words, the client is receiving less than what is transmitted from SPC.

Each of the RTP packets transmitted from the SPC contains a sequence number (specified in IETF RFC 3550 Section 6.4.1) which is consecutively increasing for each packet transmitted. At the SOPT the RTP packets are intercepted and the sequence number and the corresponding size are stored.

The ABE calculation is done at each network estimation interval as follows:

1. Let T1 be the time of receipt of first RTCP RR packet and T2 be the time of receipt of the next RTCP RR. The value is T2 is greater than T1.
2. Let EHSN1 be the EHSN reported in the first RTCP RR and EHSN2 be the EHSN reported in the second RTCP RR.
3. Let C1 be the cumulative lost packets in the first RTCP RR and C2 be the cumulative packet lost reported in the second RTCP RR.
4. Let APS be the average payload size of the data being transmitted to the client.
5. So, ABE = (((EHSN2– EHSN1) – (Num Packet Lost))*APS) / (T2-T1); where, Num Packet Lost = C2–C1.

11.2.4.1.2 Network Buffer Fill Level (NBFL)

This is measure of amount of bytes that have been transmitted from the SPC but have still not reached the client.

A positive rate of change in the value of NWFL over time denotes a possible upcoming congestion, whereas a negative rate of change in the value of NWFL denotes that network is clear.

The network fill level at the measurement time T1 is calculated as follows:

1. Let, say at time T1, EHSN1 be the sequence number of packet received by the client, and let EHSN2 be the sequence number of packet transmitted by the SPC.
2. Let APS be the average payload size of the RTP packets being transmitted by the SPC.
3. So, NWFL = (EHSN2-EHSN1) $\times$ APS.

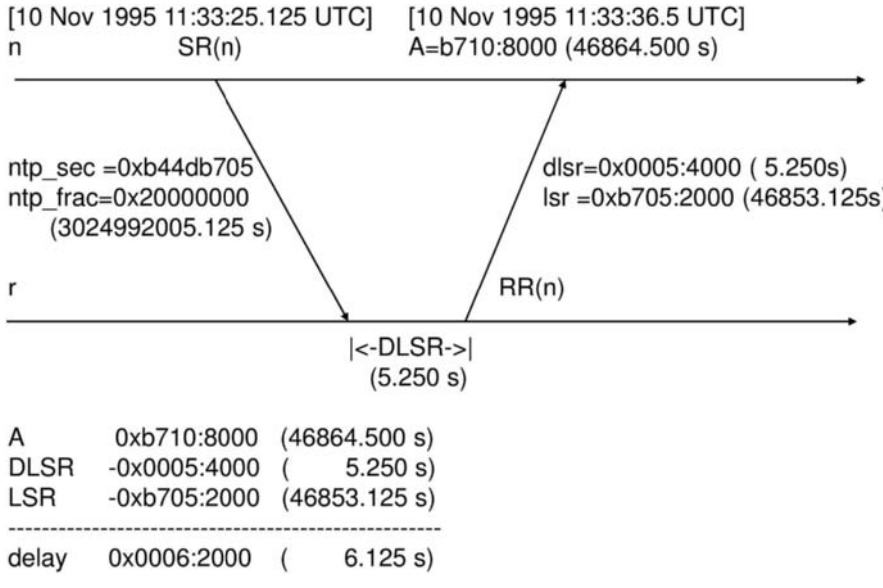


Figure 11.3 Example round-trip computation.

11.2.4.1.3 Packet Error Rate (PER)

The PER is specified directly in the RTCP RR reports in the cumulative and lost fraction field.

11.2.4.1.4 Round Trip Time (RTT)

The calculation provided below has been borrowed from IETF RFC 3550 section 6.4.1, p. 40.

Let SSRC_r denote the receiver issuing this receiver report. Source SSRC_n can compute the round-trip propagation delay to SSRC_r by recording the time A when this reception report block is received. It calculates the total round-trip time A-LSR using the last SR timestamp (LSR) field, and then subtracting this field to leave the round-trip propagation delay as (A – LSR – DLSR). This is illustrated in Figure 11.3. Times are shown in both a hexadecimal representation of the 32-bit fields and the equivalent floating-point decimal representation. Colons indicate a 32-bit field divided into a 16-bit integer part and 16-bit fraction part. This is to be used as an approximate measure of distance to cluster receivers, as wireless links have asymmetric delays.

11.2.4.2 Media Cache Transcoding[37–41]

In addition to the transcoding of the cached media into several CBR streams, the process packs the streams into a single container file. The packing is done as such to ensure that media switching framework has very low overhead in switching from one stream to another stream. A switching marker is also placed in the stream to help switching subsystem decide that it is appropriate to switch to new CBR if needed.

It uses popular FOSS [42] encoder decoder FFMPEG [43] to perform the transcoding job.

11.2.4.3 Bit Stream Switching Framework

This performs two very important tasks:

- The selection of the proper CBR stream based on network condition estimated by the network estimation system.
- It performs the actual switch from one CBR stream to another CBR stream seamlessly.

11.2.4.3.1 Selection of Proper CBR Stream

On a very high level the decision if switching is needed is taken as follows:

1. On detection of *poor network condition* switch to CBR stream whose bit-rate is closest (lower bound) to the ABE.
2. On detection of *good network condition* switch to CBR stream which the next higher than the current stream.
3. On detection of *average network condition* continue with the current CBR stream.
4. A network condition is termed *poor* if the measured PER is higher than a predetermined threshold, or the rate of change in NBFL is positive.
5. A network condition is termed *average* if PER is nonzero and is below the predetermined threshold interval.
6. A network condition is termed *good* if the measured PER is zero and the rate of change in network fill level is negative.

Note: 80% of ABE is provisioned for video and 20% is provisioned for audio

11.2.4.3.2 Switching to proper CBR stream

On a very high level the following steps are taken for switching to a new CBR stream:

1. At start of the stream the SPC reads the container metadata to get information about the CBR stream. The metadata contain the following information:
   a. Bit rate of the available CBR streams.
   b. Position of switching markers in each of the CBR stream (as bytes offset).
2. The metadata is stored in a proper data structure for fast lookup.
3. Start with the stream whose bit-rate is closest (higher bound) to average of all the available bit-rates.
4. The SPC keeps a counter which increments by one when it encounters a switching marker while streaming from one of the CBR stream.
5. When SPC encounters a switching marker, it checks if a switch to a new CBR stream is needed. If needed it looks up the new bytes offset position of the switching maker 'N' (where N is the absolute count of the total number switching marker encountered in the stream) using the metadata.
6. It seeks to the new byte offset position and start sending packet from that stream.

11.3 Performance of the QoE Cache

Table 11.1 shows the performance of the QoE cache.

Table 11.1 Performance of QoE cache.

Problem effect	Problem cause	Our solution	Test result
TCP congestion control negatively impacting the Mobile web browsing	Variable Round Trip time (RTT), time varying Packet Error Rate (PER) and potentially high PER during handoff or poor RF condition	**TCP Optimization** *faster web browsing and download*	Improvement factor ranging from 2 to 7 depending on the mobility environment
Slow web page download time over the air interface	DNS queries over the relatively high latency air link	**DNS optimization** *faster startup*	
Slow delivery of content over air interface from origin server	Need to transverse the end to end network to download/access the content	**Web Proxy and Cache** *caching content locally enabling faster content downloads*	
Increased re-buffering events, black screens, pixelation, jerky motion and blurriness, during content streaming	Packet loss associated with fast changing radio conditions in mobility	**Streaming Optimizer** *bit rate switching enabling faster startup and steady streaming without intermittent media artifacts*	
Slow delivery of content for streaming	Need to transverse the end to end network to download/access the content	**Streaming proxy and cache** *caching content locally with efficient buffer management enabling faster and smoother streaming*	

11.3.1 Web Browsing

Table 11.2 shows the Web browsing performance of the QoE cache.

11.3.2 Streaming

In streaming, for each one of sunny day, average day and rainy day, two conditions are tested:

- video encoded in single bit rate at the origin server;
- video encoded in multiple bit rates at the origin server.

The idea is that, when content is encoded in multiple bit rates, the QoE cache fetches *all* the encoded versions of the same content and lets the client choose the most appropriate bit rate

Table 11.2 Performance of the QoE cache (web browsing).

	QoE indicator			QoS indicator
Scenario (Static)	User A QoE Cache (Cache Miss)	User A QoE Cache (Cache hit)	User B (Baseline)	Performance improvement (Factor) Cache Miss/Cache hit
Network is perfect	26 sec	23 sec	34 sec	**1.3/1.5**
Network has reasonable load	58 sec	60 sec	226 sec	**3.9/3.8**
Network is in bad condition	183 sec	123 sec	820 sec	**4.5/6.7**

based on network conditions. When content is encoded in single bit rate at the origin server, the QoE cache fetches the content and transcodes it into multiple bit rates. As a result of transcoding performed at the QoE cache, the same content is *always available* in multiple bit rates regardless of whether the origin server makes the content available in multiple bit rates or not.

In addition, testing is done on four different types of video:

- high motion;
- medium motion;
- low motion;
- animated videos.

Furthermore, the metrics (available from Windows Media Player) considered for comparing the videos with and without the QoE cache are:

- QualityFactor, which is inversely proportional to packet loss rate.
- AvgStreamBitRateFactor, which is the average rate at which video is streamed.
- BufferFactor, which is the product of the number of times rebuffering is done and the duration of rebuffering.
- LossFactor, which captures packet loss rate. In general, higher values of QualityFactor and AvgStreamBitRateFactor and lower values of BufferFactor and LossFactor imply better quality of video.

11.3.2.1 Improved Jump Start of the video using QoE-Cache

Overall the QoE cache provides improved jump start (time taken to start the video once the user clicks the PLAY button; which includes connecting to media server and buffer time.

Table 11.3 shows that the QoE cache provides an improved jump start in the average day as well as the rainy-day scenario.

Table 11.3 Performance of the QoE cache (jumpstart improvement).

Scenario	Baseline (sec)	QoE Cache (sec)
Typical Day	16.2	9
Bad Day	28	14

11.3.3 Performance on a Typical Day

11.3.3.1 High Motion Video

In the case of high motion video, it is clear from Figure 11.4 that the *BufferFactor* is significantly higher (three times) for the baseline (no QoE cache) compared to the one with the QoE cache (cache hit). This happens because Windows Media believes in recovering packets that are lost in transit from server to client and it does so by freezing the display of video at the client while recovering the lost packets through retransmission. The packet loss is reduced in the presence of the QoE cache because the client has the option of switching to a lower bit rate when the network condition is bad, thereby reducing packet loss. This also means that the QoE cache has to rebuffer fewer times (if at all) compared to the case without the QoE cache. The BufferFactor for the QoE cache miss is 2.3 times higher than that for the case without the QoE cache. This happens because the QoE cache SPC fetches the single bit-rate video from the origin server, and transcodes it, resulting in longer buffering time.

The *LossFactor* without the QoE cache is significantly higher (three times) compared to that in the case of the QoE cache hit. The reason for the lower packet loss rate is because the client can switch to a lower bit rate video under bad network conditions even when the video at the origin server is available in single bit rate. Packet loss in case of the QoE cache miss is half of that in the case without QoE cache and packet loss in case of the QoE cache hit is one-fourth of that in the case of no QoE cache. This can be explained by the fact that in case of cache miss, the QoE cache proxy fetches content from the origin server using TCP and hence the packet loss in the WAN (1% in the average day scenario) does not get reflected in the Windows Media Player.

The *AvgStreamBitRateFactor* in case of the QoE cache hit is 2.8 times the case without the QoE cache. Even the cache miss scenario provides 1.5 times the average streaming rate

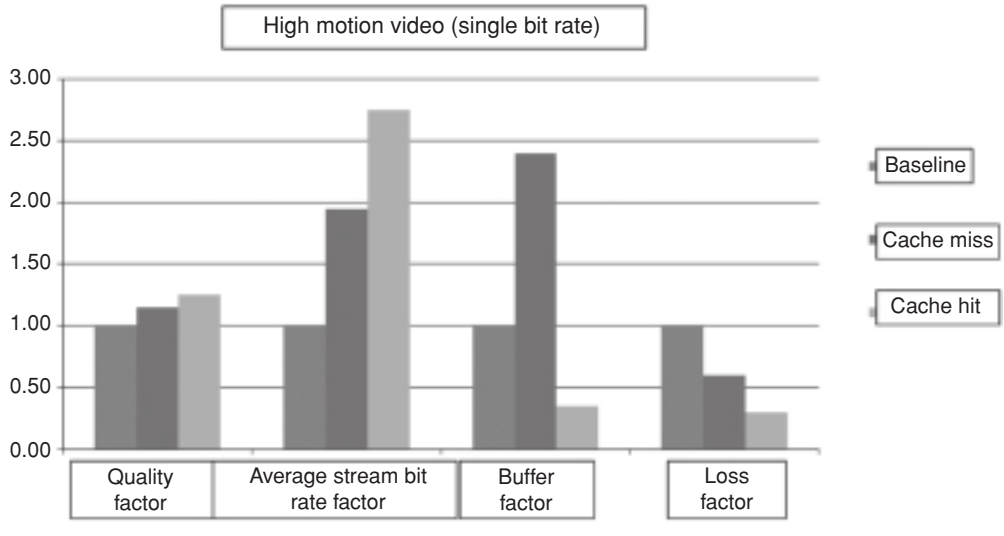


Figure 11.4 Performance of the QoE cache (high motion video).

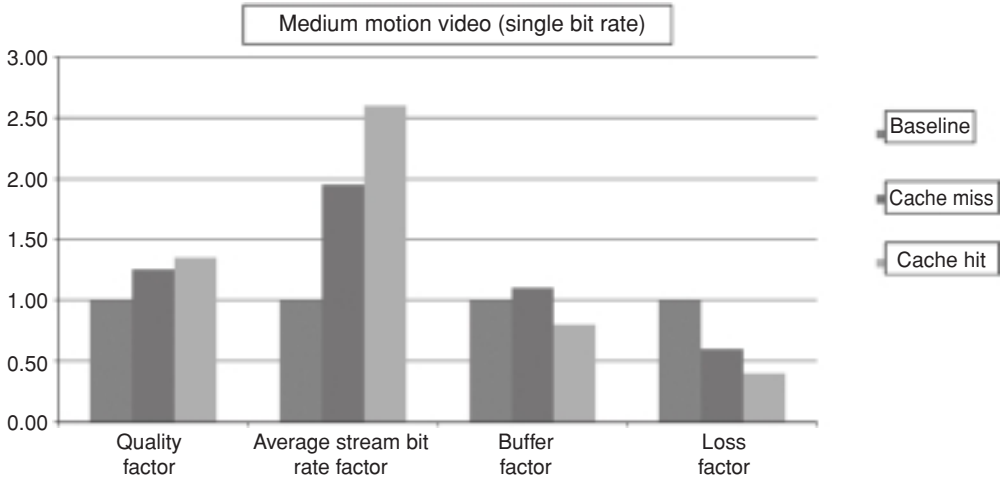


Figure 11.5 Performance of the QoE cache (medium motion video).

compared to the case without the QoE cache. This benefit results from the fact that the same video is available in multiple bitrates at the QoE cache, enabling the media client to adapt to the available bandwidth in a much better way.

The *QualityFactor* is a reflection of the LossFactor and hence the QualityFactor for QoE cache is 1.1 and 1.2 times that of the no QoE cache case for cache miss and cache hit respectively. Thus, the QoE cache improves the overall performance of single bit-rate high-motion video streaming in the average case.

11.3.3.2 Medium Motion Video

In the case of medium motion video (Figure 11.5), the *BufferFactor* without the QoE cache is about 1.2 times that of the QoE cache hit. This implies more rebuffering and/or longer duration of rebuffering, compromising the quality of experience. Note, however, that in the case of the QoE cache miss scenario, the BufferFactor is higher compared to that of the no QoE cache case because the QoE cache, after fetching the video from the origin server, transcodes the video into multiple bitrates leading to longer delay and longer buffering.

The *LossFactor* in case of the QoE cache miss is half of that for the case without the QoE cache and packet loss in case of QoE cache hit is less than half of that in the case of no QoE cache. This can be explained by the fact that in case of cache miss, the QoE cache proxy fetches content from the origin server using TCP and hence the packet loss in the WAN (1% in the average day scenario) is not reflected in the Windows media player.

AvgStreamBitRateFactor in the case of the QoE cache hit is 2.6 times compared to the case without the QoE cache. Even the cache-miss scenario provides twice the average streaming rate compared to the case without the QoE cache. This benefit results from the fact that the same video is available in multiple bitrates at the QoE cache enabling the media client to adapt to the available bandwidth in a much better way. The QualityFactor is a reflection of the

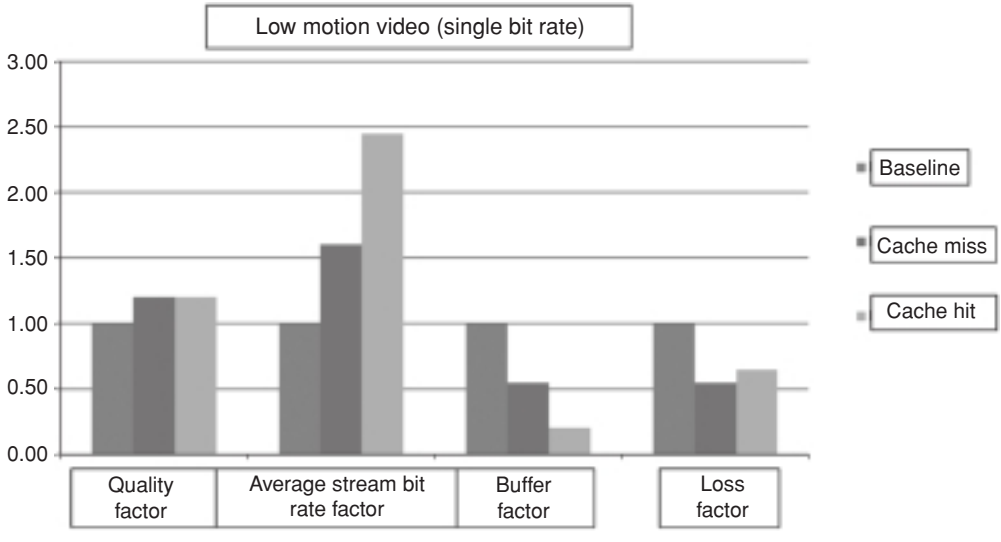


Figure 11.6 Performance of the QoE cache (low motion video).

LossFactor and hence the QualityFactor for QOE-CACHE is 1.2 and 1.3 times that of no QoE cache case for cache miss and cache hit respectively. Thus, the QoE cache indeed improves the overall performance of single bit rate medium motion video streaming in the average case.

11.3.3.3 Low Motion Video

In the case of low motion video (Figure 11.6), the buffer factor is still the highest for the case without the QoE cache and is progressively lower for the QoE cache miss and the QoE cache hit. The packet-loss rate continues to be similar to that for medium motion video.

11.4 Additional Features and Optimizations Possible for QoE-Cache

The following features can be further added to the QoE cache to make it scalable, more functional and more useful in a wider range of scenarios.

11.4.1 Capability of handling Live Streams in addition to Video-on-Demand

Live streams, unlike video on demand (VOD) streams, are transient in nature, hence they cannot be cached, source-encoded and stored for future use. So we need a mechanism to transcode the incoming live media in real time and to be able to serve the transcoded live media to one or more requesting clients, which may be present in different network conditions.

Also, in the case of streaming live media (as opposed to VOD) all the clients streaming the same live media will be on same timeline.

Therefore the following approach can be taken:

1. On the first request for a live stream, the SPC will start streaming down the live stream from the origin server.
2. It will divide the incoming live stream in chunks and forward the chunks to the transcoding module.
3. The transcoding module transcodes the chunk into MBR format (containing all bit rates). At the end of transcoding it will transfer the transcoded chunk to an output stream buffer.
4. The length of the chunk (in milliseconds) is such that the transcoding of the chunk data and copying into output buffer completes before a new chunk is encountered.
5. All the clients (which requested for same live media) will be served from the same output buffer reserved for this live channel.
6. The chunking, transcoding of the live media stream and copying it into the output buffer will continue as long as there is an active client streaming the live stream.

11.4.2 Hardware-Based Transcoding

Software based transcoding works in a prototype scenario or for a small-scale system. However, to meet the requirements of a real system and be able to handle transcoding of live video streams, transcoding must be performed using hardware. This will enable more number of streams to be processed in parallel.

11.4.3 Video Bit Rate Adaptation with RTP over TCP Streaming

RTP over TCP is useful when the client is behind some proxy and UDP ports are blocked.

Since TCP itself has congestion control and packet retransmission capability the idea was to use this capability in selecting appropriate CBR stream which is closest to TCP transmission rate. This would help in keeping the stream bit rate close to TCP transmission rate, in conditions where available network throughput is lower than highest available bit rate in the MBR transcoded file.

Currently it is not clear on how to get the value of TCP transmission rate to the application layer so that appropriate bit stream can be selected. However, once this problem is solved, RTP over TCP Streaming can be optimized as per the description above.

11.4.4 Video Bit Rate Adaptation for HTTP-Based Progressive Download[10]

HTTP based Progressive Download will benefit from the idea described in Section 11.4.3 because that will reduce the buffering requirement.

In addition to that, since most of the HTTP streaming are flash based streaming (where the player gets downloaded from the server side before start of the stream), we can add capability in player to interact will server to provide network condition statistics. Based on the network

information provided by the client, WPC will select the appropriate CBR stream (assuming content is source encoded into MBR format) to transmit to the client. This would help in CDN model, where we may be able to provide our own flash client.

11.4.5 Adaptation of Video Based on Client Device Capabilities

There are many different types of User Equipment (UE) in use in wireless networks. Each UE may have different media codec capability, resolution and bit rate support. The idea is to source encode the content in variety of codec, resolution format.

Approach:

1. On receipt of streaming request from the client, the SPC will check the type of UE from which the request came (determined by the User Agent field and XWAP profile).
2. The QoE cache system needs to maintain a database containing the capabilities of most of the popular UE. An open source database called WURFL (http://wurfl.sourceforge.net/) provides UE capabilities for most of UE available for wireless access.
3. The SPC will query the database for knowing the capabilities of the UE that has requested the media.
4. The SPC will then select appropriate media stream matching the capabilities of the UE.

For example, we can have video track encoded into H.264 and MPEG4 format. The H.264 is an advance video codec format providing better quality at less bit-rate (around 30% saving in bit-rate over MPEG4 for same quality stream). However H.264 needs 4 times the processing power for decoding as compared to MPEG4 for same bit-rate. Therefore, only very modern UE can support decoding H.264.

Thus, depending on UE capability, the SPC can select the appropriate video stream to be transmitted to the client.

11.5 Summary

The goal of this chapter was to stress the importance of quality of experience for video streaming, and describe the design of a QoE cache that can be deployed in a wireless network to improve that experience. Specifically, the QoE cache is based on four principles. First, content needs to be moved closer to the end user by caching it locally (Web proxy/cache) so that the probability of its being subjected to congestion or packet loss is reduced significantly. Second, in the context of video streaming to a mobile user, TCP can be further optimized to deal with variable round-trip time, time-varying packet error rate and potentially high packet error rate during handoff or poor RF condition leading to reduction of start-up time for streaming and reduction of download time for download and play. Third, by adapting media stream transmission to the condition of the wireless network, streaming quality can be improved. Finally, the response time for streaming video can be further improved in wireless networks by eliminating DNS queries over the air link. An implementation instance of QoE cache based on the Windows Media server was described, drawbacks pointed out and a potential solution was prescribed. Further insights were provided about how to further enhance the performance of the QoE cache and algorithms were given in fair amount of detail to incorporate those

enhancements. Detailed performance evaluation of the QoE cache was done based on an implementation instance and the benefits were highlighted. Finally, additional features and optimizations that are possible for the QoE cache were described.

References

[1] Open Source Streaming Server. http://developer.apple.com/opensource/server/streaming/index.html (accessed June 9, 2010).

[2] QuickTime Streaming Server: Delivering Digital Media Standards. http://www.apple.com/quicktime/streamingserver/ (accessed June 9, 2010).

[3] Darwin Streaming Server. http://dss.macosforge.org/ (accessed June 9, 2010).

[4] Microsoft Windows Media – Web Server vs. Streaming Server. http://www.microsoft.com/windows/windowsmedia/compare/WebServVStreamServ.aspx (accessed June 9, 2010).

[5] Adobe Flash Media Streaming Server 3.5. http://www.adobe.com/products/flashmediastreaming/ (accessed June 9, 2010).

[6] Streaming from a Web Server. Bill Birney, June 2003. http://www.microsoft.com/windows/windowsmedia/howto/articles/webserver.aspx (accessed June 9, 2010).

[7] Red 5: Open Source Flash Server. http://osflash.org/red5 (accessed June 9, 2010).

[8] Understanding Video Streaming. http://www.deskshare.com/Resources/articles/vc_StreamingMediaFormats.aspx (accessed June 9, 2010).

[9] VideoLAN – VLC Media Player. http://www.videolan.org/ (accessed June 9, 2010).

[10] Krasic, C. and Li, K. and Walpole, J. (2001) The case for streaming multimedia with TCP. Lecture Notes in Computer Science, pp. 213–218, Springer (accessed June 9, 2010).

[11] Helix Proxy and Proxy. http://www.realnetworks.com/products-services/helix/media/server_proxy.aspx (accessed June 9, 2010).

[12] Chen, S., Shen, B., Wee, S. and Zhang, X. (2003) *Adaptive and Lazy Segmentation Based Proxy Caching for Streaming Media Delivery*. Proceedings of ACM NOSSDAV, Monterey, CA, June 2003.

[13] Bommaiah, E., Guo, K., Hofmann, M. and Paul, S. (2000) *Design and Implementation of a Caching System for Streaming Media over the Internet*. Proceedings of IEEE Real Time Technology and Applications Symposium, Washington, DC, May 2000.

[14] Chiu, M.Y. and Yeung, K.H. (1997) *Partial Video Sequence Caching Scheme for Vod Systems with Heteroeneous Clients*. Proceedings of the 13th International Conference on Data Engineering, Birmingham, United Kingdom, April 1997.

[15] Kangasharju, J., Hartanto, F., Reisslein, M. and Ross, K.W. (2001) *Distributing Layered Encoded Video Through Caches*. Proceedings of IEEE Inforcom, Anchorage, AK, April 2001.

[16] Miao, Z. and Ortega, A. (2002) Scalable proxy caching of video under storage constraints. *IEEE Journal on Selected Areas in Communications*, **20**(7): 1315–1327.

[17] Reisslein, M., Hartanto, F. and Ross, K.W. (2000) *Interactive Video Streaming with Proxy Servers*. Proceedings of the First International Workshop on Intelligent Multimedia Computing and Networking, Atlantic City, NJ, February 2000.

[18] Rejaie, R., Handely, M. and Estrin, D. (1999) *Quality Adaptation for Congestion Controlled Video Playback Over the Internet*. Proceedings of ACM SIGCOMM, Cambridge, MA, September 1999.

[19] Sen, S., Rexford, J. and Towsley, D. (1999) *Proxy Prefix Caching for Multimedia Streams*. Proceedings of IEEE INFOCOM, New York City, NY, March 1999.

[20] Zhang, Z., Wang, Y., Du, D. and Su, D. (2000) Video staging: A proxy-server based approach to end-to-end video delivery over wide-area networks. *IEEE Transactions on Networking*, **8**, 429–442.

[21] Rejaie, R., Handley, M., Yu, H. and Estrin, D. (1999) *Proxy Caching Mechanism for Multimedia Playback Streams in the Internet*. Proceedings of International Web Caching Workshop, San Diego, CA, March 1999.

[22] Rejaie, R., Yu, M.H.H. and Estrin, D. (2000) *Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet*. Proceedings of IEEE INFOCOM, Tel-Aviv, Israel, March 2000.

[23] Wu, K., Yu, P.S. and Wolf, J. (2001) *Segment-based Proxy Caching of Multimedia Streams*. Proceedings of WWW, Hongkong, China, May 2001.

[24] Chae, Y., Guo, K., Buddhikot, M. *et al.* (2002) Silo, rainbow, and caching token: Schemes for scalable fault tolerant stream caching. *IEEE Journal on Selected Areas in Communications*, **20**(7): 1328–1324.

[25] Chen, S., Shen, B., Wee, S. and Zhang, X. (2004) *Investigating Performance Insights of Segment-Based Proxy Caching of Streaming Media Strategies*. Proceedings of ACM/SPIE Conference on Multimedia Computing and Networking, San Jose, CA, January 2004.

[26] Cherkasova, L. and Gupta, M. (2002) *Characterizing Locality, Evolution, and Life Span of Accesses in Enterprise Media Server Workloads*. Proceedings of ACM NOSSDAV, Miami, FL, May 2002.

[27] Chesire, M., Wolman, A., Voelker, G. and Levy, H. (2001) *Measurement and Analysis of a Streaming Media Workload*. Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, CA, March 2001.

[28] Wang, B., Sen, S., Adler, M. and Towsley, D. (2002) *Proxy-Based Distribution of Streaming Video Over Unicast/Multicast Connections*. Proceedings of IEEE INFOCOM, New York City, NY, June 2002.

[29] Khan, J.I. and Tao, Q. (2001) *Partial Prefetch for Faster Surfing in Composite Hypermedia*. Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, CA, March 2001.

[30] Jung, J., Lee, D. and Chon, K. (2000) *Proactive Web Caching with Cumulative Prefetching for Large Multimedia Data*. Proceedings of WWW, Amsterdam, Netherland, May 2000.

[31] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (1996) Rtp: A transport protocol for real-time applications. http://www.ietf.org/rfc/rfc1889.txt, January 1996.

[32] Schulzrinne, H., Rao, A. and Lanphier, R. (1998) Real time streaming protocol (rtsp). http://www.ietf.org/rfc/rfc2326.txt (accessed April 1998).

[33] Chen, S., Shen, B., Wee, S. and Zhang, X. (2003) *Streaming flow Analyses for Prefetching in Segment-Based Proxy Caching Strategies to Improve Media Delivery Quality*. Proceedings of the 8th International Workshop on Web Content Caching and Distribution, Hawthorne, NY, September 2003.

[34] Ma, W. and Du, H. (2000) *Reducing Bandwidth Requirement for Delivering Video Over Wide Area Networks with Proxy Server*. Proceedings of International Conferences on Multimeida and Expo., New York City, NY, July 2000.

[35] Tewari, R., Vin, A.D.H. and Sitaram, D. (1998) *Resource-Based Caching for Web Servers*. Proceedings ACM/SPIE Conference on Multimeida Computing and Networking, San Jose, CA, January 1998.

[36] Guo, H., Shen, G., Wang, Z. and Li, S. (2007) Optimized streaming media proxy and its applications. *Journal of Network and Computer Applications*, **30**(1), Special Issue on Network and Information Security, pp. 265–281, ISSN: 1084-8045.

[37] Vetro, A., Christopoulos, C. and Sun, H. (2003) Video transcoding architecture and techniques: an overview. *IEEE Signal Processing Magazine*, **20**(2), 18–29, ISSN: 1053-5888.

[38] Chan, Yui-Lam, Cheung, Hoi-Kin and Siu, Wan-Chi (2009) Compressed-domain techniques for error-resilient video transcoding using RPS. *IEEE Transactions on Image Pprocessing*, **18**(2), 357–370, ISSN: 1057-7149.

[39] Dick, M., Brandt, J., Kahmann, V. and Wolf, L. Adaptive Transcoding Proxy Architecture for Video Streaming in Mobile Networks. http://www.ibr.cs.tu-bs.de/papers/brandt-icip05.pdf (accessed June 9, 2010).

[40] Ahmad, I., Wei, X., Sun, Y. and Zhang, Y.-Q. (2005) Video transcoding: An overview of varioustechniques and research issues. *IEEE Transactions on Multimedia*, **7**(5), 793.

[41] Universal Media Transcoding. http://www.rhozet.com/ (accessed June 9, 2010).

[42] Free and Open Source Software (FOSS). http://en.wikipedia.org/wiki/Free_and_open_source_software (accessed June 9, 2010).

[43] FFMPEG: http://ffmpeg.org/ (accessed June 9, 2010).

[44] Rodrguez, P. (2004) Sarit Mukherjee and Sampath Rangarajan. Session level techniques for improving web browsing performance on wireless links. Proceedings of the 13th International Conference on World Wide Web, New York, NY, USA, 2004. pp. 121–130, ISBN: 1-58113-844-X.

[45] TCP Westwood. http://www.cs.ucla.edu/NRL/hpi/tcpw/ (accessed June 9, 2010).

[46] Mascolo, S., Casetti, C., Gerla, M. *et al.* (2001) TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. ACM Mobicom.

[47] Casetti, C., Gerla, M., Mascolo, S. *et al.* (2002) TCP westwood: end-to-end congestion control for wired/wireless networks. *Wireless Networks Journal*, **8**(5), 467–479.

[48] Wang, R., Valla, M., Sanadidi, M.Y. and Gerla, M. (2002) Adaptive Bandwidth Share Estimation in TCP Westwood, Globecom.

12

Opportunistic Video Delivery Services in Delay Tolerant Networks

The number of endpoints connected wirelessly to the Internet has long overtaken the number of wired endpoints, and the difference between the two is widening. Wireless mesh networks, sensor networks, and vehicular networks represent some of the new growth segments in wireless networking in addition to mobile data networks, which are currently the fastest growing segment in the wireless industry. Wireless networks with time-varying bandwidth, error rate, and connectivity beg for opportunistic transport, especially when the link bandwidth is high, error rate is low, and the endpoint is connected to the network in contrast to when the link bandwidth is low, error rate is high and the endpoint is not connected to the network. "Connected" is a binary attribute in TCP/IP meaning one is either part of the Internet and can talk to everything or one is isolated. In addition, connecting requires a globally unique IP address that is topologically stable on routing timescale (minutes to hours). This makes it difficult and inefficient to handle mobility and opportunistic transport in the Internet. Clearly we need a new networking paradigm that avoids a heavyweight operation like end-to-end connection, and enables opportunistic transport. In addition to the above scenarios, given that the predominant use of the Internet today is for content distribution and content retrieval, there is a need for handling dissemination of content, especially video, in an efficient manner. This chapter describes a network architecture that addresses the above-mentioned unique requirements.

12.1 Introduction

Caching [1, 2] and content distribution networkss (CDNs) [3, 4] have proven to be extremely useful on the Internet today. However, the mechanisms used to leverage the usage of caches on the Internet today are not very clean. For example, to use an institutional proxy cache, typically, the browsers have to be *configured* to point to the proxy cache or a special device

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

like a Layer-4 switch has to be used to transparently redirect web requests to the institutional cache or some automated scripts are run to identify the proxy cache for the corresponding browser. Multiple mechanisms exist because each has its own advantages and disadvantages and none of these techniques is a clear winner. Similarly, to redirect a user request to the nearest mirror server of a CDN, different CDN vendors use different mechanisms again. Moreover, the details of the mechanism and signaling used by a CDN vendor like Akamai [3] and/or Limelight Networks [4] are *proprietary* even though we know it is most likely based on a domain name system (DNS). While the DNS-based redirection is best for CDN vendors like Akamai and Limelight networks who do not *own* the network, it may not be the best way out for network service providers like AT&T, who *own* their network, for building a CDN. Once again, just as in the case of caching, multiple techniques are used in CDNs to redirect an end-user request to the "nearest" mirror server. In summary, *several* complex parallel *control and signaling* infrastructures have been built on top of the Internet to make use of the caches (or storage nodes). The question is, *if* we had the luxury of building a *clean slate* next-generation Internet, would it make sense to maintain status quo or to design a simpler unified mechanism to leverage the well-proven benefits of caching (storage) in the network.

A parallel development has been happening in the Internet community in the context of delay/disruption tolerant networking (DTN) [5] whose objective is to deal with *disruption* or *intermittent* connections on the Internet that the traditional TCP/IP paradigm cannot handle efficiently. Interestingly enough, the DTN community recognized the need for hop-by-hop transport combined with *caching* as a way of mitigating the effect of *disruption* in communication. The DTN community has proposed a different *control and signaling* mechanism on top of the Internet.

Yet another community, mostly driven by the researchers in the field of mobile communications and networking [6, 7, 23] have realized the benefit of hop-by-hop transport in multi-hop wireless communications to improve performance of content delivery and once again *caching* plays a central role. To take advantage of caching, this community is designing *yet another* control and signaling mechanism.

Given that *caching* is so central to multiple communities and that it is being used to serve a variety of needs, and given that due to the limitations of the current Internet design, each community has to come up with its own control and signaling mechanism, and also given the luxury of designing the next-generation Internet from scratch, there is a tremendous benefit in designing a unified protocol for leveraging the *caches* to meet the needs of these diverse communities.

The architecture proposed in this chapter is not an alternative to what the CDN community has deployed or to what the DTN community or mobility community has proposed – rather it is an attempt to leverage the best ideas from these communities and to put them together into a unified framework. In the context of the current Internet, this framework can be thought of as an overlay network on top of Internet. In a clean-state design of the next-generation Internet, the unified framework may very well be integrated into the network itself.

12.2 Design Principles

There are several reasons why a new architecture is needed for opportunistic transport and delay-tolerant networking. First, the Internet architecture assumes that there exists an end-to-end path between the endpoints that need to communicate and exchange information. This is certainly not true for mobile endpoints, which may not be within the range to communicate or

for sensor nodes that wake up intermittently to communicate. Second, the Internet architecture computes a single path from the source to the destination for routing packets between the two endpoints. However, there are several scenarios where computing a path from the source to the destination is not possible ahead of time, especially when the source or the destination is not connected to the network. In addition, in the event of congestion along the precomputed path, packets become delayed. It may be a better approach to decide on the route dynamically as opposed to statically before the communication begins. Third, packet switching is assumed to be the most appropriate abstraction for interconnecting heterogeneous systems. However, when the end-users are mostly interested in content, the appropriate switching entities need not be packets – rather they could be messages or contents themselves. Fourth, the Internet architecture assumes that packet loss rate is small and the lost packets can be recovered through end-to-end retransmissions. However, when such assumptions fail, as in time-varying wireless links where packet loss rate could be significantly high from time to time, or in systems where an end-to-end path does not exist, the end-to-end performance suffers badly. In general, these shortcomings of the Internet architecture need to be addressed for the following types of networks:

- hybrid fixed and mobile networks;
- military ad-hoc networks;
- vehicular networks;
- mobile wireless networks;
- media distribution networks;
- sensor networks.

All the above factors lead to the design of a new architecture for opportunistic communication and delay-tolerant networking with the following characteristics:

- Network elements should have *persistent memory* or *storage (cache)* integrated in them. This is important because the intended destination may be out of reach and the message may need to be stored at an intermediate network element until the intended destination is connected. The intermediary carrying the message to the final destination could also be mobile and hence may need to hold on to the message until it gets back into the connected network or gets a chance to hand over the message to the destination. The side effect of storing content in the network is the efficient delivery that can be achieved by virtue of delivering the content from the network itself as opposed to from a server outside the network.
- The network should not be built on packet-switching technology but rather on message-switching technology where a message could be as big as the entire content file itself.
- Messages should be transmitted between two successive intermediate network elements using a reliable *virtual link layer* protocol. The link between two successive network elements is called *virtual* because it consists of multiple hops in the underlying physical network but behaves as a single link between two nodes in the overlay network. The link layer protocol should be configurable so that it can be tuned to the characteristics of the virtual link.
- *Routing* decisions should not be made at the source at the time of transmission – rather, it should be made at each intermediate network element as the message is transmitted hop by hop.
- In addition to address-based routing, there is a need for *content based* routing.

- A *network layer* should support multiple classes of service so that some messages are treated with higher priority compared to others based on the urgency of message delivery.
- *Naming and late binding* should be two of the most important support services in the network. Late binding is useful because resolving names upfront makes sense only when the routing needs to be decided at the source. However, when the destination may not even be connected to the network or the exact location of the destination is not known ahead of time, it makes sense to resolve names to addresses towards the end of delivery process.
- The semantics of *multicasting* needs to be defined differently because the members of a multicast group may not be online when the multicast session starts or ends. Moreover, the source and/or destinations may be mobile leading to dynamic formation of the multicast tree.
- The *transport layer* becomes minimal in this case because the network itself provides reliable transmission between network elements. Moreover, since the final destination may not be connected, it may be difficult, if not impossible, to have a timely end-to-end acknowledgment as in case of TCP in the Internet.
- Acknowledgment continues to make sense for the *Application layer* protocol. However, the semantics may vary depending on the circumstances.

12.3 Alternative Architectures

Several network architectures and associated protocols have been proposed to handle disruptive communication. However, the driving factors behind these architectures have been different and hence, despite significant functional commonality, these architectures evolved slightly differently as described next.

12.3.1 Delay and Disruption Tolerant Networking (RFC 4838)

Delay and disruption tolerant networking [5, 30] was the result of combining research in the fields of mobile and ad hoc networking (MANET), vehicular ad hoc networking, and the DARPA-funded research on the Interplanetary Internet (IPN). The IPN architecture that was developed to cope with significant delays and packet corruption of deep-space communications laid the foundation of DTN architecture. However, it evolved significantly from the initial IPN architecture as the focus shifted from just Interplanetary Internet to more general concept of delay and disruption tolerant networking.

12.3.1.1 Architecture

Delay and disruption tolerant networking (RFC 4838) architecture (Figure 12.1) consists of endpoints (source and destination) and intermediate nodes, some of which merely forward bundles (bundles are equivalent of packets in DTN architecture) and some of which, in addition to forwarding bundles, also store them for forwarding at an opportunistic moment some time in future (such nodes are referred to as custodians). All nodes in the architecture have a common protocol layer, namely the bundle protocol layer that binds together all components of DTN architecture. The bundle protocol layer, as described later, is the equivalent of TCP/IP in the Internet architecture. The architectural highlights of DTN are presented next.

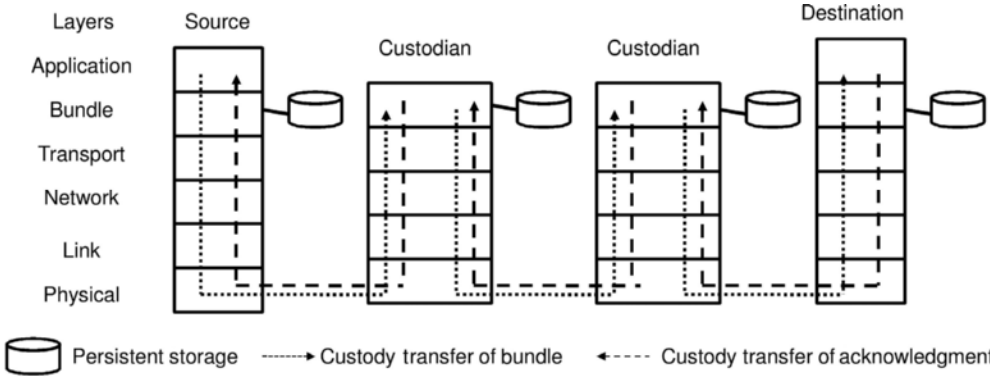


Figure 12.1 DTN Architecture (RFC 4838).

12.3.1.1.1 Hop-by-Hop Delivery

A fundamental paradigm used in DTN networks is "'store and forward" where storing is persistent and not transient as in IP networks. Furthermore the unit of storage and forwarding in DTN networks is a "bundle" as opposed to a "packet". A bundle is formed by adding relevant header information to an application data unit (ADU) so that the ADU can be routed to the right destination by the bundle layer. A bundle header consists of the original source and final destination endpoint identifier (EID) so that each intermediate node in the DTN network knows where the bundle originated from and where it is headed. Each intermediate node forwards the bundle based on the EID. However, all intermediate nodes are not the same. Some of them simply forward the bundle towards the final destination, while some others take on custody of the bundle. Taking custody of a bundle means taking on the responsibility for "reliably" transferring the bundle to the next custodian or to the final destination whichever may be closer. Reliable transmission requires the custodian to figure out if the bundle has been successfully delivered to the next custodian or not and, if not, retransmit it until the bundle reaches the desired custodian and/or the final destination.

12.3.1.1.2 Naming and Late Binding

Endpoints in DTN architecture are identified using an EID, which follows the syntax of a uniform resource identifier or URI (RFC 3956). Each EID may refer to either a single destination endpoint or a set of destination endpoints. The latter is applicable to anycast and multicast.

Binding refers to mapping an EID to the next-hop EID or the lower layer address for transmission. For example, in the context of the Internet, the binding happens at the source where the name is mapped into an IP address using DNS. However, in case of DTN architecture, EIDs may be reinterpreted at each intermediate node as the final destination may not be connected to the network or its location in the network may not be known. Thus, DTN nodes perform "name-based" routing with late binding as opposed to "address-based" routing.

12.3.1.2 Protocols

12.3.1.2.1 Virtual Link (Bundle Delivery) Layer

In DTN networks, a "virtual" link (bundle delivery) layer protocol is responsible for transferring a "bundle" from one DTN node to the next DTN node just as the link layer protocol is responsible for transferring a packet from one router (host) to the next router (host) in the

TCP/IP protocol stack.. The "virtual" link (bundle delivery) layer in DTN networks rides on top of traditional transport layer protocols (TCP and UDP).

In contrast to the TCP/IP protocol stack where the link layer is usually best effort (no guarantee of delivery, for example in Ethernet), the bundle layer in DTN supports both best effort as well as reliable delivery mechanisms. Best effort delivery happens between two nodes when the next-hop DTN node is not a "custodian". However, between two "custodian" nodes, the delivery is expected to be "reliable".

12.3.1.2.2 Virtual Network (Bundle Forwarding and Routing) Layer

In DTN networks, "virtual" network (bundle forwarding and routing) layer protocol is responsible for computing the route of a "bundle" from the original source to the final destination. DTN node does the forwarding of a "bundle" to the next-hop node. In fact, the "virtual" network (bundle forwarding and routing) layer resides on top of traditional transport layer protocols (TCP and UDP).

The bundle header contains the original source EID, final destination EID, current custodian EID and report-to endpoint EID in addition to some other fields. Forwarding decisions are made based on the final destination EID and reports, such as, return receipt and so on are sent to the report-to EID.

Routing is tricky in DTN because the capacity and delay in DTN links vary with time. If link characteristics are known ahead of time, forwarding decisions can be made in an intelligent manner. However, often such information is not available and then routing becomes challenging. In general, the links could be persistent (DSL line), on demand (dial-up modem), scheduled intermittent (low-orbiting satellite), opportunistic intermittent (unscheduled low-flying aircraft) or predictive intermittent (based on previously observed pattern). Different routing protocols are appropriate for different types of links.

The DTN architecture supports routing and forwarding of anycast and multicast traffic in addition to that of unicast traffic. However, the semantics of multicast routing in DTN is tricky as a member of the multicast group might express interest in content that might have already been delivered to other members of the multicast group. This requires support for storage and forwarding at intermediate nodes for delivery at a later point of time.

12.3.1.2.3 Virtual Transport (Bundle Flow Control and Congestion Control) Layer

In DTN networks, "virtual" transport (bundle flow control and congestion control) layer protocol is responsible for ensuring that the average rate at which a sending node transmits data to a receiving node does not exceed the average rate at which the receiving node is prepared to receive data (flow control) and the aggregate rate at which the senders inject traffic into the network does not exceed the maximum aggregate rate at which the network can deliver data to the destinations over time (congestion control). In addition, there are various acknowledgment schemes to guarantee end-to-end delivery. As the "virtual" transport protocol for DTN network rides on top of transport layer protocols (TCP and UDP), it can leverage both the flow/congestion control of TCP and the acknowledgment scheme of TCP for its own "equivalent" functions at a higher level.

12.3.1.2.4 Application Layer

Applications interface with the DTN architecture asynchronously and that is the most appropriate mechanism in long/variable delay environments. Usually the applications register

callback actions when certain triggering events occur (such as, arrival of an application data unit or ADU). The application layer protocol generates ADUs and uses the bundle layer for forwarding and delivery.

12.3.2 BBN's SPINDLE

BBN's SPINDLE [8] program was driven by DARPA with the objective of transforming the US military into an agile distributed network-centric force. In order to achieve that goal, it was critically important to have access to mission-related information even under temporary disruptions to connectivity in the global information grid (GIG). DARPA's DTN program, with the above goal in mind, has been developing technologies that enable access to information when stable end-to-end paths do not exist and infrastructure access cannot be assured. The DTN technology makes use of persistence within network nodes, along with the opportunistic use of mobility, to overcome disruptions to connectivity. That is the genesis of BBN's SPINDLE architecture.

BBN's SPINDLE architecture is designed on the principle of extensibility with the goal of leveraging the same architecture for serving a variety of next-generation networking needs. A DTN application which focuses on delivering a bundle to the destination in an intermittently connected network would have different needs compared to a content discovery and retrieval solution. However, BBN's SPINDLE network is designed to meet the needs of both of these seemingly disparate types of applications through its extensible architecture. The details of the architecture are described below.

12.3.2.1 Architecture

The core of SPINDLE architecture (Figure 12.2) consists of a bundle protocol agent (BPA), which implements the main functionality of bundle protocol (RFC 4838). For example, BPA implements forwarding a bundle to the next-hop DTN node, performs delivery of bundle to the applications, implements custody-transfer mechanism and so on. However, the routing and forwarding functions, the implementation of reliable delivery of bundle, and so forth, are decoupled from the basic forwarding functionality of the bundle protocol and are designed as separate components. In fact, the other components of the SPINDLE architecture are: data store (DS), decision plane (DP), convergence layer adapter (CLA) and application/middleware (A/M). These components are coupled with the core BPA component through inter component communication protocol (ICCP).

12.3.2.1.1 Bundle Protocol Agent

The BPA offers the services of the bundle protocol (BP). It executes the procedures of the BP and that of bundle security protocol (BSP) in cooperation with other components of the architecture. For example, while the BPA is responsible for implementing the mechanisms of the BP, such as reading, creating and updating the fields in the bundle header, it has the flexibility of leveraging the DP component of the SPINDLE architecture for any key decisions, such as those related to policy or optimization.

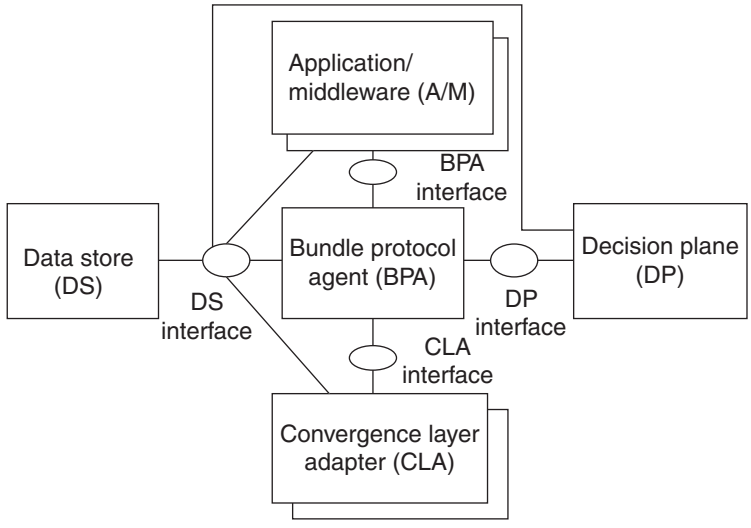


Figure 12.2 BBN's SPINDLE architecture.

The main functions of the BPA are:

- Forwarding a bundle to the next-hop DTN node, whether it is for unicast, anycast or multicast. However, the next-hop computation is done by the DP and passed on to the BPA.
- Doing fragmentation and reassembly of bundle payload as needed to adapt the delivery of payload over a link with time-varying capacity
- Implementing custody-transfer mechanisms in the bundle header, such as, sending custody acknowledgment. However, whether to accept or reject custody is determined by the DP again.
- Delivering a bundle to a "registered" application.
- Discarding and deleting a bundle. Once again, it is the DP that decides whether a bundle should be discarded or not.
- Implementing all security functions, such as, authentication, confidentiality, and data integrity.

In addition to the above functionality, the BPA implements agent interface that can be accessed by applications and it uses the interfaces exposed by other components of the architecture.

12.3.2.1.2 Data Store

The DS implements persistent storage that is used to store not only the bundles, but also the bundle metadata, network state information and application state information. Network state information includes routing tables, content metadata, policies, and so on, while application state information includes registration information, application metadata, and so forth.

The data store implements a full data base management system (DBMS) to enable basic database functions such as query processing.

Knowledge-based (KB) systems can also be integrated with the DS to enable advanced inferencing based on execution of rules.

12.3.2.1.3 Decision Plane

If BPA is the heart of the system, the DP is the brain. Specifically, the decision plane is responsible for routing information dissemination, route computation, routing table updates, late binding and name resolution, policy handling, content caching and replication decision, content search and other decisions. The DP consists of several modules:

1. *Routing information dissemination module*. This module is responsible for exchanging routing-related information among the network elements. Specifically, this module decides what information will be shared with whom and when. In addition to disseminating the information, this module also collects the routing-related information in incoming bundles and updates the relevant entries in the knowledge base.
2. *Routing module*. This module is responsible for computing routes for unicast and multicast, for updating the routing table entries, for generating next hops for bundles, for scheduling the bundles, for making decisions about whether to take custody of a bundle or not, and so on.
3. *Policy module*. This module is responsible for interpreting policies, enforcing policies, dispatching events based on policies, enabling users to add/delete policies and for subjecting bundles to policies as they pass through the DTN node.
4. *Naming and late binding module*. This module is responsible for resolving names of DTN nodes and feeding the information to the router module so that the right decision about forwarding a bundle can be taken. Usually, this module is invoked and used when the bundle is close to the final destination or close to the care-of address of the final destination where it will be stored for opportunistic delivery to the final destination.
5. *Content module*. This module is used for content-based access, specifically for content search, content caching and replication, content routing and other content-related functionality.

12.3.2.1.4 Convergence Layer Adapter

The CLA is responsible for actual transport of the bundles. It leverages whatever transport functionality is available from the underlying network. The status of links (available or not), schedule (for opportunistic delivery) and quality of service parameters are all monitored by CLA and the relevant information is passed on to the relevant modules of the architecture for their efficient functioning.

12.3.2.1.5 Application/Middleware

The A/M module is responsible for sending and receiving bundles based on application needs. This module leverages the services exposed by BPA.

12.3.2.2 Protocols

12.3.2.2.1 Virtual Link Layer

In BBN's SPINDLE network, "virtual" link layer functionality is implemented by the CLA. The beauty of CLA is that it is not limited to using TCP and UDP – rather it can potentially use any custom protocol (such as CLAP) that might be available at the corresponding DTN node for use in a specific type of network (for example, CLAP may be available at a DTN node and it may be the best suited protocol for wireless links with highly time-varying bandwidth, delay and error characteristics).

12.3.2.2.2 Virtual Network Layer

In the SPINDLE architecture, the "virtual network layer" functionality is implemented by the DP. The beauty of DP is that it is not limited to using any specific protocol – rather it allows use of any protocol to disseminate and assimilate routing information. Moreover, the routing information is also customizable, meaning that the information that will be distributed will depend on the type of routing. For example, routing information to be disseminated for content-based routing could be very different from the routing information needed for traditional address-based routing. The SPINDLE architecture supports this flexibility. Specifically, policy-based routing, content-based routing, late-binding, rich naming, and so on, are all supported in an extensible manner by DP in SPINDLE architecture.

12.3.2.2.3 Virtual Transport Layer

The "virtual" transport layer protocol is responsible for ensuring flow control and congestion control and is implemented by the DP. This is because the network state information, including congestion is available to and disseminated by the DP module.

12.3.2.2.4 Application Layer

Applications interface with BBN's SPINDLE network architecture asynchronously and that is the most appropriate mechanism in long/variable delay environments. In the SPINDLE architecture, the application-layer functionality is implemented by the A/M module. The A/M module also provides multiplexed DTN communication service to non-DTN applications running on the node.

12.3.3 KioskNet

KioskNet [9, 10] started at the University of Waterloo with the goal of providing very low cost Internet access to rural villages in developing countries using the principles of Delay Tolerant Networking. KioskNet system uses vehicles, such as buses, to ferry data between village kiosks and Internet gateways in nearby urban centers. The data carried by the buses from the rural areas are reassembled at an intermediary (or proxy server) for interaction with legacy servers.

12.3.3.1 Architecture

KioskNet (Figure 12.3) consists of a set of kiosks from which *ferries* (buses) carry data to a set of *gateways* that communicate with a *proxy* on the Internet. The ferries not only carry data from the kiosks, but they also carry data to the kiosks. The main architectural components of KioskNet are described below in more detail.

12.3.3.1.1 Kiosks

Each kiosk is equipped with a server referred to as kiosk controller from which one or more PCs can boot off. Kiosk controllers have WiFi connectivity to allow users to connect to them wirelessly. In addition, although Kiosk controllers could have different types of backhaul, such as, dial-up, GPRS, VSAT, the most interesting one from the perspective of DTN is the mechanical backhaul, such as, ferries (buses, cars, motorbikes, and so forth).

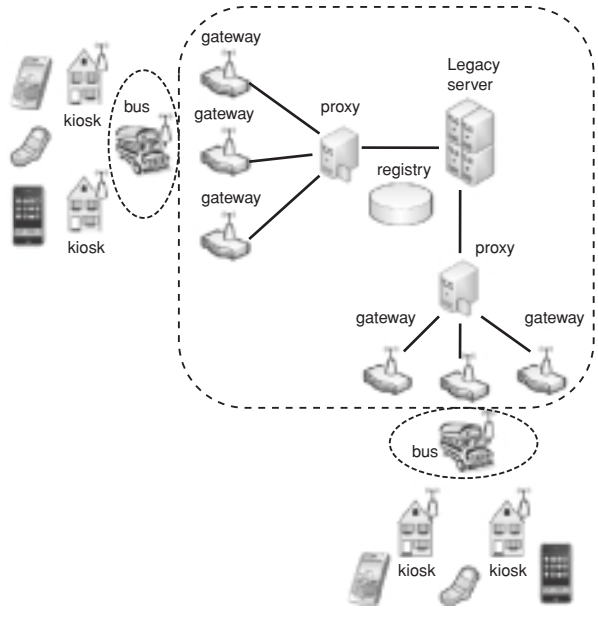


Figure 12.3 KioskNet architecture.

The kiosk is expected to be used by two types of users. First type of users use a PC that boots over the network from the kiosk controller and can then access and execute application binaries provided by the kiosk controller. The second type of users use their own devices, such as smart phones, PDAs and laptops to connect to one or more kiosk controllers, or a bus directly and use them as wireless hotspots that provide store-and-forward access to the Internet.

A KioskNet *region* consists of a set of kiosks in the same geographic area administered by the same entity. This means that all entities within the region are certified by the same certificate authority. In addition, from the networking perspective, all data bundles are flooded within a region. Figure 12.3 shows a system with two regions, which could be managed either by different administrative entities or by a single administrative entity.

12.3.3.1.2 Ferries

Ferries provide Internet connectivity to the kiosks via a mechanical backhaul. Examples of ferries include cars, buses, motorbikes, or trains that pass by a kiosk and an Internet gateway. A ferry has a PC with 20–40GB of storage and a WiFi network interface and is powered by the vehicle's own battery. The PC communicates opportunistically with the kiosk controllers and Internet gateways when it comes within their coverage area. During an opportunistic communication session, which may last up to several minutes, hundreds of MB of data can be transferred in each direction. This data is stored and forwarded in the form of self-identifying *bundles*. Ferries upload and download bundles opportunistically to and from an Internet gateway.

12.3.3.1.3 Gateways

A gateway is a nothing but a PC with WiFi network interface, and a broadband (DSL or cable) Internet access. A gateway collects data opportunistically from a ferry and holds it in local storage before uploading it to the Internet through the proxy. A region may have one or more gateways.

12.3.3.1.4 Proxy

Most communication between a kiosk user and the Internet would probably be for existing services such as e-mail, or for accessing back-end systems that provide government-to-citizen services. Legacy servers that provide such services typically are not designed to handle either long delays or disconnections and, most importantly, they cannot be easily modified. Therefore, the architecture requires a disconnection-aware proxy that hides end-user disconnection from legacy servers. A proxy is assumed to exist in every region.

The proxy is resident in the Internet and has two halves. One half establishes disconnection-tolerant connection sessions with applications running on the kiosk controller or on mobile users' devices. The other half communicates with legacy servers on behalf of disconnected users.

When a proxy receives application data from a legacy server, it transfers the data to an appropriate gateway, which eventually forwards it to a passing ferry. The ferry delivers the data to a kiosk, which in turn passes it to kiosk users.

In the opposite direction, when a kiosk user wants to send data to the Internet, it uses a ferry to transport the data to a gateway, which in turn transfers it to a proxy. The proxy passes received data to the legacy Internet servers.

12.3.3.2 Protocols

The protocol stack in KioskNet is shown in Figure 12.4.

12.3.3.2.1 Virtual Link Layer

In KioskNet, TCP is used as the "virtual" link-layer protocol and is responsible for transferring a "bundle" from one DTN node to the next DTN node. Note that the mobile device (cell phone), kiosk controller, ferry, gateway as well as the proxy are considered DTN nodes in KioskNet architecture.

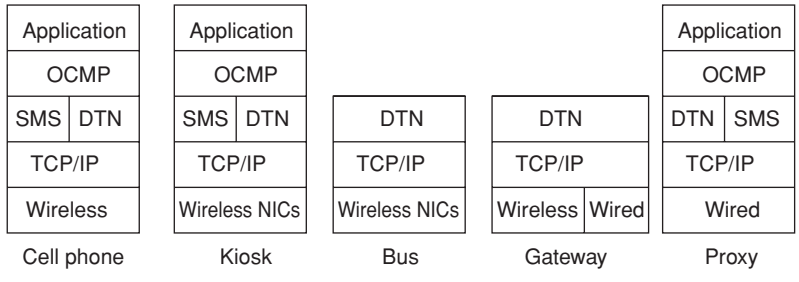


Figure 12.4 KioskNet protocol stack.

12.3.3.2.2 Virtual Network Layer

In KioskNet architecture, the "virtual" network layer protocol is responsible for routing a "bundle" from the original source to the final destination. Routing within a disconnected region of KioskNet is different from routing from Internet to Kiosk.

12.3.3.2.3 Routing Within a Disconnected Region

A routing algorithm allows a kiosk to decide which ferry to use to send data to the Internet, and for a gateway to decide which ferry to use to communicate with a particular kiosk. However, ferries may fail and ferry trajectories are not always known beforehand. Therefore, routing in KioskNet is a hard problem. Fortunately, a ferry can transfer several tens of megabytes of data to and from a kiosk as it passes by, and it can store tens of gigabytes of data on its hard drive. Based on these observations, routing is done using flooding, thereby trading off over-the-air bandwidth and storage for reliability and ease of routing. That means, in KioskNet, a kiosk or a gateway transfers all its data to every ferry that passes by, and accepts data from every ferry. Clearly, this redundancy maximizes the probability of bundle delivery while eliminating routing decisions altogether. An added benefit is that, with flooding, communication between kiosk users in the same region does not require a bundle to go to the proxy. Finally, flooding requires less configuration at deployment time, making KioskNet easier to deploy.

KioskNet eliminates the inefficiencies commonly associated with naive flooding using two optimization techniques. First, before any data is transferred from a kiosk controller to a ferry and *vice versa*, bundle metadata is exchanged so that each side knows what bundles the other side has, and as a result can avoid accepting bundles it already has.

Bundle metadata exchange happens as follows:

1. First, the kiosk controller tells the ferry the user GUIDs registered at the kiosk.
2. Second, the ferry informs the kiosk controller of the bundle IDs on the ferry belonging to these users.
3. Third, the kiosk controller determines the missing bundles, and requests them from the ferry.
4. Finally, the ferry transfers these bundles to the kiosk controller. No metadata exchange is required in the other direction: a kiosk transfers all its bundles to every passing ferry.

In addition, although bundles sent from a kiosk destined to legacy servers on the Internet are flooded to all reachable gateways in the same region, and these gateways accept all bundles from all kiosks, these gateways coordinate with each other to make sure that each bundle will be sent to the proxy by one and only one gateway. This avoids wasting bandwidth on the link between the gateways and the proxy.

With these two optimizations, despite flooding, KioskNet resources, namely kiosk-to-bus communication link, and the gateway-to-proxy link, are not unnecessarily wasted.

12.3.3.2.4 Routing of Internet-to-Kiosk Bundles

Data from legacy servers destined to kiosk users is first buffered at the responsible proxy, then sent to gateways that transfer bundles to ferries. After a bundle is sent to a gateway, it is flooded to reach its destination kiosk (handed to all ferries passing by that gateway).

Proxies are located in bandwidth-rich data centers, but gateways are connected to the Internet typically using slow dialup or DSL links. Given that the link between the gateways and the

proxy is the bottleneck, ideally the proxy should choose only *one* gateway in the region to send each bundle to, rather than flooding it to all the gateways in the region.

If the schedules of ferries are known to the proxy, a routing and scheduling algorithm can be used at the proxy that can choose the best gateway for each bundle and decide the order in which they are sent so as to minimize the overall delay. Moreover, this algorithm can also enforce arbitrary bandwidth allocation among kiosks. If bus schedules are not known, then the proxy has no choice but to flood it to all the gateways.

12.3.3.2.5 Virtual Transport Layer

In KioskNet, these capabilities are provided by the opportunistic connection management protocol (OCMP) which runs on top of DTN and other available network connections. OCMP can be viewed as a disconnection-tolerant and policy-driven session layer that runs over both DTN and standard links. Each type of available communication path is modeled as a connection object (CO) within OCMP. For instance, the DTN mechanical backhaul path is a CO, just as a TCP connection over WiMAX or dialup is.

The OCMP allows a policy manager to assign bundles arbitrarily to transmission opportunities on COs. This scheduling problem is complex because it has to manage many competing interests: reducing end-to-end delay, while not incurring excessive cost, and maximizing transmission reliability.

12.3.3.2.6 Application Layer

Applications, residing on mobile device (cell phone), kiosk controller and the Proxy, interface with the KioskNet architecture asynchronously and that is the most appropriate mechanism in long/variable delay environments. Usually the applications layer protocol generates ADUs and uses the bundle layer for forwarding and delivery.

12.4 Converged Architecture

The last section described several alternative architectures for dealing with disruption tolerant networking. However, each architecture was designed to solve a slightly different problem and hence they evolved differently. For example, the DTN architecture (RFC 4838) has been designed primarily to deal with significant delays, including long interruptions, in communications. BBN's SPINDLE architecture evolved from the need of providing access to information to military field force where stable end-to-end paths do not exist and infrastructure access cannot be assured. KioskNet was designed with the goal of providing very low cost Internet access to rural areas. DieselNet's [11–13] goal has been primarily to deal with the challenges of vehicular DTN. PocketNet [14, 15] was designed to enable communications via storage and networking purely at the end-hosts.

Shortcomings of Internet architecture need to be addressed for a variety of networks including hybrid fixed and mobile networks, military ad-hoc networks, vehicular networks, mobile wireless networks, media distribution networks, and sensor networks.

While DTN architecture primarily addresses the requirements of hybrid fixed and mobile networks, BBN's SPINDLE mostly focuses on military ad-hoc networks, KioskNet deals with hybrid fixed and mobile networks, and DieselNet primarily explores DTN in vehicular networks. There are isolated efforts to deal with the time-varying characteristics of mobile

wireless networks [23, 31, 32] and the existence of CDNs to address the needs of media distribution [3, 4].

A deeper look into the entire problem space exposes an underlying commonality in the basic building blocks of a converged network architecture that addresses all the above problems in a uniform manner. The converged network architecture is referred to as cache and forward (CNF) network architecture.

12.4.1 Cache and Forward Network Design Goals

Cache and forward architecture [16, 17], evolved at WINLAB, Rutgers University in order to solve four main problems: (i) efficient delivery and retrieval of video, (ii) improving throughput in multihop wireless network, (iii) improving content delivery in a mobile network where the mobile nodes may be intermittently connected to the wired infrastructure, and (iv) improving communication in sensor networks. These issues are briefly discussed below:

1. Efficient delivery and retrieval of video (*challenges of media distribution networks*). Video will be driving the need for improved communications infrastructure in the foreseeable future, as is evident from the phenomenal rise of YouTube [18], Revver [19], and other video-sharing sites in addition to the rise of Internet television where specialized sites provide niche television content over the Internet. The uniqueness of video as content is the huge size of files that are several orders of magnitude larger than music (audio) files. While peer-to-peer (P2P) networking is helping scale the distribution of video, the P2P delivery mechanism, by itself, cannot optimize the bandwidth usage in the underlying network. Moreover, the existing TCP/IP networking paradigm is not exactly suitable for video retrieval, because the TCP/IP paradigm expects the application to figure out through an out-of-band mechanism (such as, a search engine) the name/IP address of the server where a given video is hosted and then connect to the server to fetch the desired content as opposed to allowing the application to query the network for a given video and retrieve it from the network, all operations being done in-band.
2. Improving throughput in multihop wireless networks (*challenges of mobile wireless networks*). When TCP/IP is used over wireless links, performance is often degraded due to transport layer timeouts. In-network solutions such as indirect TCP have been proposed in earlier work [20]. In addition, when TCP is used over multiple wireless hops (an increasingly common scenario), the so-called "self-interference" effect in which packets from the same flow (specifically, the data and acknowledgment packets belonging to the same flow but traveling in opposite directions), contend for the same radio resources, can further degrade end-to-end performance [21, 22]. For multi-hop wireless networks, the probability of impairment or disconnection in at least one radio link can be quite high as the number of hops, n, increases. It can be shown that the probability of failure before the file transfer is completed is increased by a factor of n^2 over the probability of a single hop failure. This is almost an order-of-magnitude increase for $n = 3$ hops and is two orders of magnitude increase for $n = 10$ hops.
3. Improving content delivery in mobile networks where mobile nodes may be intermittently connected to wired network (*challenges of hybrid fixed and mobile networks, military ad-hoc and vehicular networks*). The existing TCP/IP architecture embraced the concept of

mobile IP to reach mobile hosts when the point of attachment of the mobile host (with the wired network) changes due to its mobility. However, the scope of mobile IP is limited to the case when mobile node is not disconnected from the wired network for a significant amount of time (longer than the lifetime of a typical Internet session). At the same time, research has shown that if content is temporarily stored in the network when the destination node is not connected to the wired network, and is ferried via "mobile nodes" to the destination node, the capacity of the wireless network increases substantially [24, 25].

4. Improving communication in sensor networks (*challenges of sensor networks*). Internet applications involving sensors are expected to grow rapidly in the next 10 years. Sensor scenarios have unique networking requirements [33] including the ability to deal with disconnections due to wireless channel impairments as well as sensor hardware sleep modes. In addition, sensor applications tend to be data-centric and are thus more interested in content-aware services (for example, querying data) rather than in connecting to a specific IP address.

Cache and forward architecture was designed to address the above issues in an efficient manner.

12.4.2 Architecture

The main concepts of CNF architecture (Figure 12.5) are listed below:

Post Office (PO). The CNF architecture is based on the model of a postal network designed to transport large objects and provide a range of delivery services. Keeping in mind that the sender and/or receiver of an object may be mobile and may not be connected to the network, we introduce the concept of PO, which serves as an indirection (rendezvous) point for senders and receivers. A sender deposits the object to be delivered in its PO and the network routes it to the receiver's PO, which holds the object until it is delivered to the final destination. Each sender

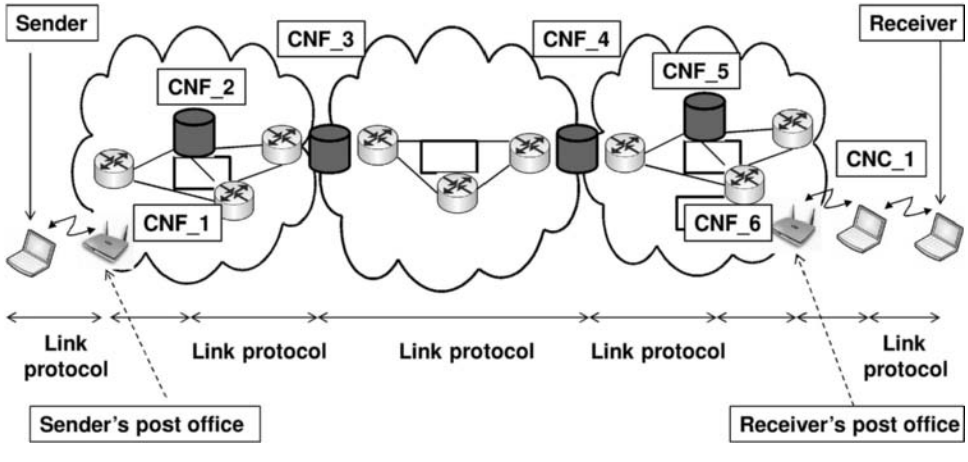


Figure 12.5 Cache and forward (CNF) architecture.

and receiver may have multiple POs, where each PO is associated with a point of attachment in the wired network for a mobile endpoint (sender/receiver). In the context of DTN network and BBN's SPINDLE, a PO is nothing but a special type of custodian node, while in the context of KioskNet, PO is equivalent of a Gateway.

Cache and Forward (CNF) Router. The CNF router is a network element with persistent storage and is responsible for routing packages within the CNF network. Packages are forwarded hop-by-hop (where a hop refers to a CNF hop and not an IP hop) from the sender's PO towards the receiver's PO using forwarding tables updated by a routing protocol running either in the background (proactive) or on demand (reactive). In the context of DTN network and BBN's SPINDLE, a CNF router is nothing but a DTN node that may or may not be a custodian node, while in the context of KioskNet, a kiosk as well as a gateway is a CNF router.

Cache and Carry (CNC) Router. The CNC Router is a mobile network element that has persistent storage exactly as in a CNF Router, but is additionally mobile. Thus a CNC router can pick up a package from a CNF router, another CNC router or from a PO and carry it along. The CNC router may deliver the package to the intended receiver or to another CNC router that might have a better chance of delivering the package to the desired receiver. In the context of DTN network and BBN's SPINDLE, a CNC Router is nothing but a mobile DTN node that may or may not be a custodian node, while in the context of KioskNet, a Ferry is a CNC Router.

Content Identifier (CID). To make content a first class entity in the network, we introduce the notion of persistent and globally unique content identifiers. Thus if a content is stored in multiple locations within the CNF network, it will be referred to by the *same* content identifier. The notion of a CID is in contrast to identifiers in the Internet, where content is identified by a URL whose prefix consists of a string identifying the *location* of the content. CNF endpoints will request content from the network using content identifiers. Since none of the described alternative architectures considered content as a first-class citizen of the network, there was no need to have a specific content ID which is a "network" level ID; rather they continued to use the "application" level ID, such as, URLs as in the case of traditional Internet. However, CID is a fundamentally important concept in the converged network architecture.

Content Discovery. Since copies of the same content can be cached in multiple CNF routers in the network, discovering the CNF router with the desired content that is "closest" to the requesting endpoint must be designed into the architecture. We discuss this in more detail in the next section. Once again, since none of the described alternative architectures in Section 3 considered content as a first-class citizen of the network, there was no need to discover content within the "network" rather they continued with the traditional Internet model whereby a search engine is expected to be used to locate the node holding the content and once the node is located, traditional network-based routing is used to access the content. One exception is BBN's SPINDLE architecture where the concept of

content module has been conceived as a part of the decision plane module for enabling content-based access, specifically for content search, content caching and replication, content routing, and other content-related functionality. However, in the converged architecture, as content is a first-class citizen of the network, content discovery is part of the network layer functionality.

Type of Service. In order to differentiate between packages with different service delivery requirements (high priority, medium priority, low priority), a type of service (ToS) byte will be used in the package header. The ToS byte can be used in selecting the cache replacement policy and in determining the delivery schedule of packages at the CNF routers. This concept exists with both DTN architecture as well as BBN's SPINDLE architecture.

Multiple Delivery Mechanisms. A package destined for a receiver would be first delivered to, and stored in the receiver's PO. There are several ways in which the package can be delivered from the PO to the receiver:

- A PO can inform the receiver that there is a package waiting for it at the PO and it (the receiver) should arrange to pick it up. The receiver can pick up the package when in range of that PO. Otherwise, it may ask its new PO and/or a CNC router to pick up the package on its behalf.
- A receiver can poll the PO to find out if there is a package waiting for pick up. If it is and the receiver is within range of the PO, it can pick up the package itself. Otherwise, it may ask its new PO and/or a CNC router to pick up the package on its behalf.
- A PO can proactively *push* the package to the receiver either directly or via CNC routers.

Routing mechanisms are not prescribed in either DTN architecture or in BBN's SPINDLE architecture as the architecture is expected to provide flexibility for choosing variety of routing techniques, especially at the edge of the wired network. KioskNet, however, does talk about intelligent flooding from the Gateway (equivalent of a PO) to the final destinations (PCs and/or mobile phones) as a way of routing. Nonetheless, the above-mentioned techniques in the converged architecture cover the entire gamut of routing from the wired edge node to the mobile or wirelessly connected end nodes (or final destinations).

Details of protocols used in CNF network are described next.

12.4.3 Protocols

The CNF architecture represents a set of new protocols (Figure 12.6) that can be implemented either as a "clean-slate" implementation or on top of IP.

Virtual Link Layer. The virtual link layer in CNF architecture uses a reliable link layer protocol referred to as CNF LL in Figure 12.6. CNF LL protocol is used for reliable delivery of packages (bundles) between two adjacent CNF nodes which could be either CNF/CNC routers or CNF hosts. In the traditional TCP/IP network paradigm, two adjacent CNF nodes could be separated by multiple IP router hops or could be connected by a wireless link with highly time-varying bandwidth, delay, loss characteristics. Because of this diversity, the link

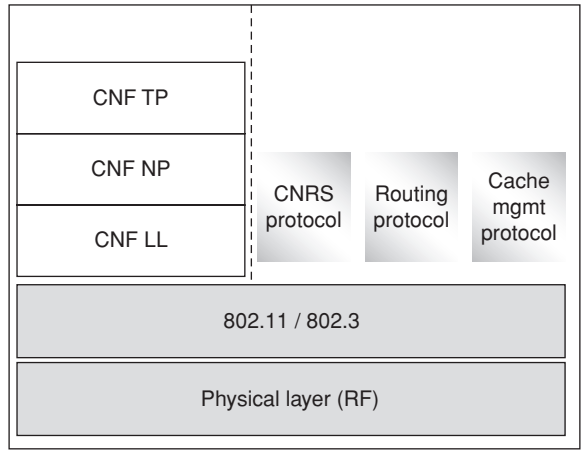


Figure 12.6 Cache and forward protocol stack.

layer protocol used in CNF is "configurable" to suit the characteristics on the underlying link. For example, if the virtual link in CNF consists of a few wired IP router hops, TCP may be the best virtual link layer protocol for reliable delivery between two adjacent CNF nodes. On the other hand, if the virtual link in CNF consists of a wireless link with highly time-varying bandwidth, delay, loss characteristics, then some proprietary protocol (such as, CLAP) may be the best virtual link layer protocol for reliable delivery between two adjacent CNF nodes.

Virtual Network Layer. The virtual network layer in CNF architecture uses a network layer protocol referred to as CNF NP in Figure 12.6. Each node in the CNF network, as described earlier, is assumed to have a large storage cache (∼TB) that can be used to store packages (files/file segments) in transit, as well as to offer in-network caching of popular content. CNF routers may either be wired or wireless, and some wireless routers may also be mobile. The basic service provided by the network is that of file delivery either in "push" or "pull" mode, that is, a mobile end user may request a specific piece of content, or the content provider may push the content to one (unicast) or more (multicast) end users. Each query and content file transported on the CNF network is carried as a CNF packet data unit or *package* in a strictly hop-by-hop fashion. The package is transported reliably between data stores at each CNF router before being prepared for the next hop towards its destination. The CNF network assumes the existence of a reliable link-layer protocol between any pair of CNF routers, and this protocol can be customized to the requirements of each wireless or wired link in the network. Packages are forwarded from node to node using opportunistic, short-horizon routing and scheduling policies at CNF nodes that take into consideration factors such as package size, link quality and buffer occupancy. Alternative routing techniques may also be used opportunistically to deal with congestion or link failure. Caches in the network can create more complex scenarios. To retrieve any content, a host would send a query to the network with the location-independent content ID (CID), and the query would then be routed to the "nearest" CNF router using a *content routing* procedure, and the content would then be routed back to the host using the conventional routing capability mentioned earlier.

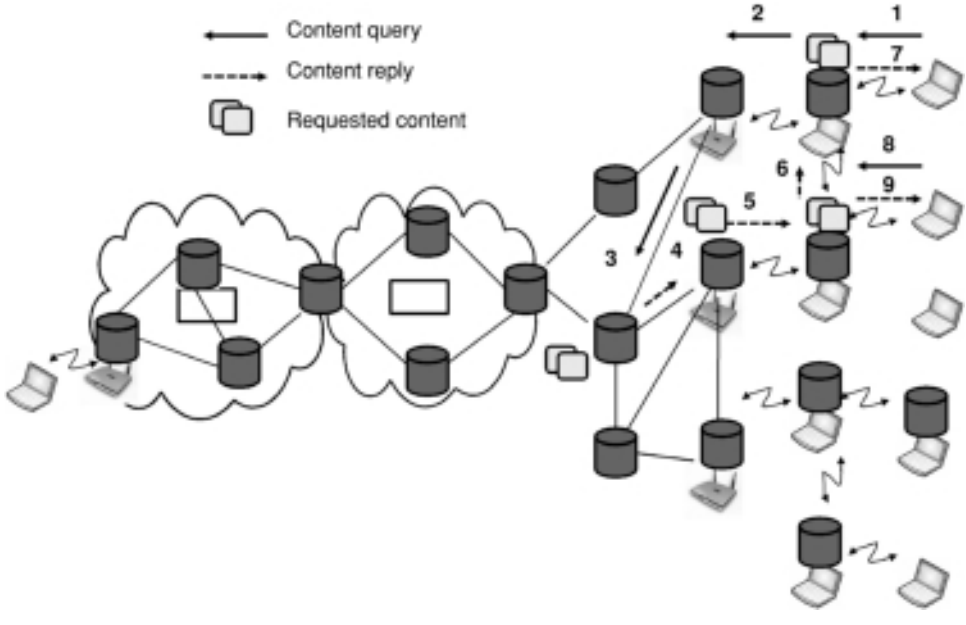


Figure 12.7 Routing of queries and content in CNF network.

One unique aspect of CNF network layer protocol is the concept of "query routing" whereby an application may trigger query for a specific content object that may be stored "within" the CNF network. Note that this is possible as content is cached "within" the network itself. The query is routed from the originating node through the network. Just as a traditional router in the Internet maintains a routing table with IP addresses of destination networks/hosts, a CNF router maintains a routing table with CID and indicating next hops to follow in order to reach the desired content object stored within the network.

When the query is routed to the CNF node that has the desired content cached, the network layer triggers what is called "response routing". Response is routed to the node that originated the query and note that the response forwarding is similar to traditional TCP/IP routing where packets are routed towards a given destination IP address. Perhaps the only difference is that the response (desired content object) is cached at each CNF router en route to the destination.

Figure 12.7 shows how content queries and responses are routed. Specifically, the content query originated at the top right laptop is routed through the CNF network (steps 1–3) until the query reached a CNF node with the desired content. Content is routed back (steps 4–7) where steps 5–7 are over wireless links that may or may not be mobile (if the link is a mobile wireless link, the corresponding node would be considered a CNC router. As the content is routed through the CNF network, it is cached at intermediate nodes and the benefit of caching shows up when another CNF host queries for the same object (step 8) which is now cached at the first hop (thanks to the previous query and content response routing and caching) and the content is immediately sent back to the originating CNF host (step 9).

Virtual Transport Layer. The virtual transport layer in CNF architecture uses a transport layer protocol referred to as CNF TP in Figure 12.6. The purpose of virtual transport layer

protocol is to provide an end-to-end acknowledgment or notification for the delivery of content where the ends are defined as the original source and final destination. Because of reliable link-layer delivery between CNF nodes, transport layer also includes "intermediate" level acknowledgment and notification which helps diagnose delivery problems in the same way as the tracking system does in Fedex or UPS delivery networks. In addition, it is the virtual transport layer that needs to deal with congestion and flow control as contents are transported across the CNF network from multiple sources to multiple destinations.

12.4.3.1 Support Services

Content name resolution service (CNRS). Since the CNF network is designed to support efficient distribution and retrieval of content and it allows applications to "query" for content cached in the network (see the virtual network layer), it is useful to have the IDs of CNF routers corresponding to a given file (or Content ID) that have the corresponding content cached. Since there would be potentially millions of objects, constantly updating the CNRS server for all the content would not scale. Hence the idea in CNF is to update the CNRS server with the IDs of caches where a "popular" is cached. This information may be used by the CNF hosts (when originating a query) and/or by the CNF routers when forwarding the query in order to optimize routing.

12.4.4 Performance of Protocols in CNF Architecture

12.4.4.1 CNF and TCP/IP Based Internet in Mobile Content Delivery

The goal of this section is to compare the performance of the proposed converged architecture (referred to as CNF architecture) with that of the traditional TCP/IP-based Internet architecture when it comes to delivery of content, especially large-size content, such as, video files when the sender or the receiver or both are mobile. To compare the performances of these two networks, a 24-node transit-stub network is considered, and the time taken in transferring a file from a source to a destination is computed under varying offered load where offered load = arrival rate × file size × number of source nodes. Specifically, three scenarios are considered: (i) client and server nodes are wired, (ii) client nodes are wireless but server nodes are wired and (iii) both client and server node are wireless [26,29]. The results are shown in [8] and are taken from [26].

For CNF traffic, the transmission delay depends on the number of hops and the bandwidth of each hop. For TCP traffic, the transmission delay depends on the bandwidth of the bottleneck links. If all the links have the same bandwidth, TCP based data transfer performs better than CNF based data transfer because there is no store and forward delay in TCP and it is able to take full advantage of "streaming" data. However, if the bottleneck bandwidth is much smaller than that of the other links, TCP throughput is significantly reduced as it is limited by the bottleneck bandwidth. From the plots in Figures 12.8, 12.9 and 12.10, it is clear that the file delivery time (referred to as file transfer delay in the figures) for TCP/IP network shoots up at much lower offered load compared to the case of CNF networks. Specifically, if the throughput is defined as the offered load with delay limit of 100s, the throughput of TCP is less than 2 mbps in the case where both clients and servers are wirelessly connected, while CNF throughput is 2–7 mbps.

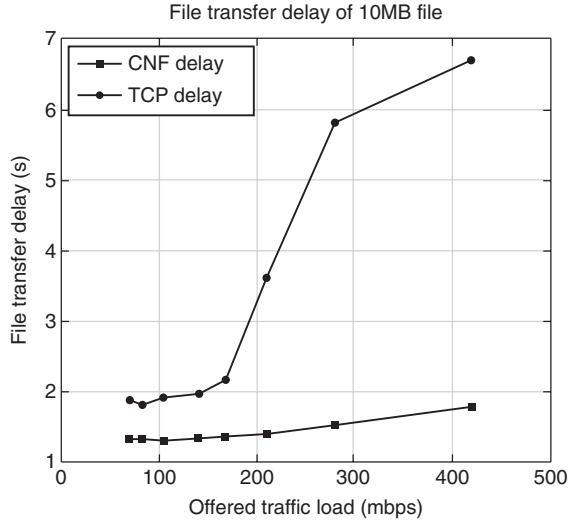


Figure 12.8 Both clients and servers are wired nodes.

12.4.4.2 CNF and TCP/IP-Based Internet in Content Retrieval

The goal of this section is to compare the performance of the proposed converged architecture (referred to as CNF architecture) with that of the traditional TCP/IP-based Internet architecture when it comes to content retrieval. To compare the performances of these two networks, a 12-node transit-stub network is considered, the time taken to retrieve a specific content is

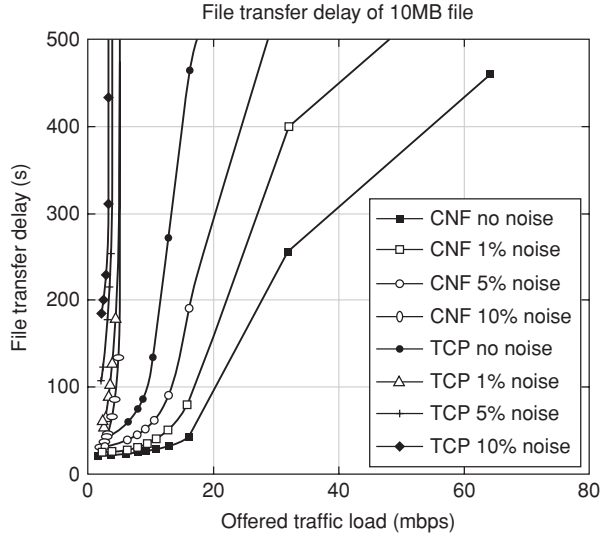


Figure 12.9 Clients are wireless, servers are wired nodes.

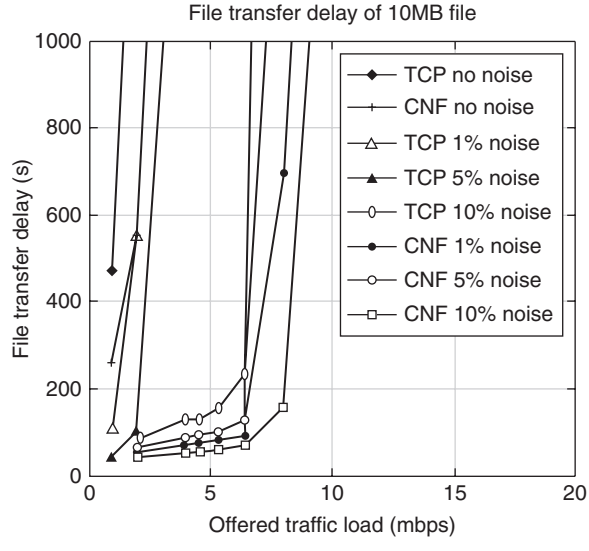


Figure 12.10 Both clients and servers are wireless nodes.

computed when the network has the intelligence (as in the converged network architecture) versus when the network is dumb (as in the traditional Internet) and the intelligence of locating content resides outside the network at the application level.

Two schemes are compared in [27]:

- Server Only or SO (meaning that the content resides only on the servers).
- Cache and capture or CC (meaning content gets cached in the network as it transits through the network and hence can be retrieved from the network as opposed to from the server only) under three different scenarios: (i) small network, many requests, (ii) large network few requests, (iii) large network, many requests [27].

It is clear from Table 12.1 that while caching helps in every scenario, request number has a bigger impact on the caching effect than network size. In both scenarios (a) and (c) where a node makes a large number of requests, integrated caching can improve the performance by more than 52%, while in scenario (b), the improvement is only 24%. This is because more requests are served off the cache in cases (a) and (c) compared to that in case (b).

Table 12.1 Comparison of content retrieval schemes in CNF networks.

Scenario	(a)	(b)	(c)
SO (seconds)	0.369	0.737	0.635
CC (seconds)	0.175	0.561	0.304
Improvement	52.6%	23.9%	52.4%

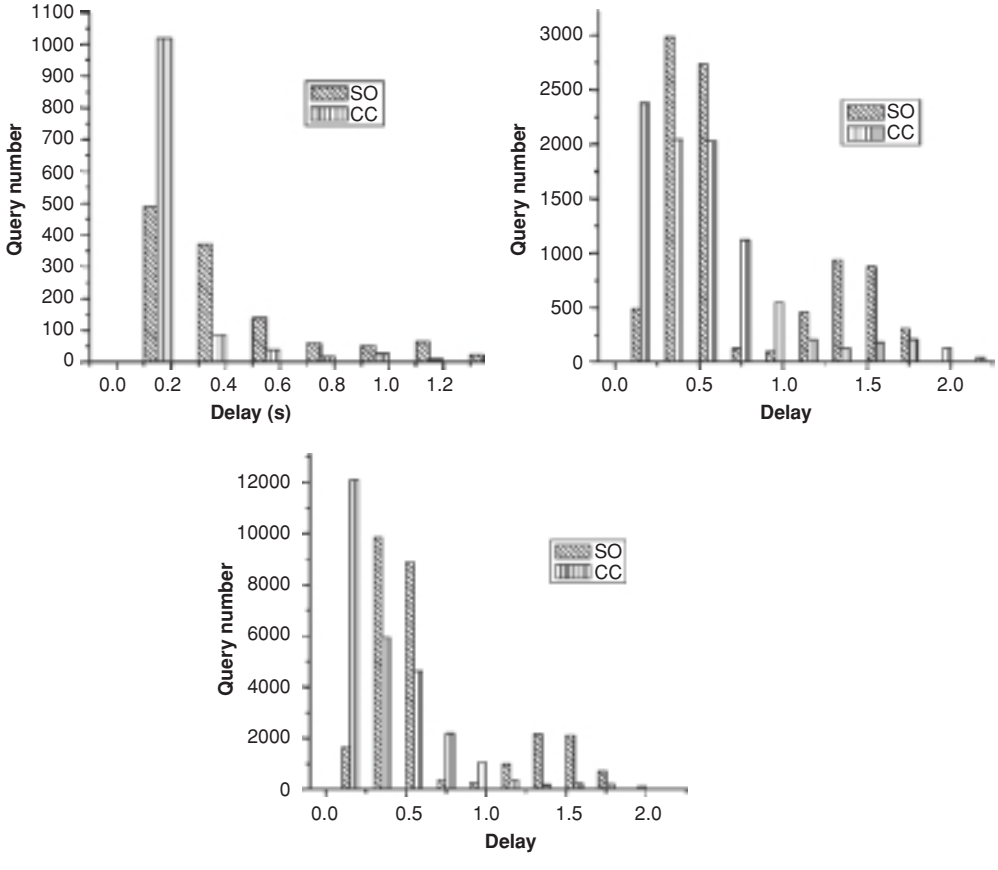


Figure 12.11 Histogram of content retrieval latency (scenarios a, b and c).

In the histograms shown in Figure 12.11, the first bin corresponds to the number of requests that were satisfied by a one-hop neighbor of the requester, either because that neighbor is the server hosting the content, or because the neighbor has a copy of the named content in its cache and the same explanation holds for the i^{th} bin. Figure 12.11 (a)–(c) show that caching and capture (CC) as proposed in the CNF architecture can satisfy many more requests by nodes that are within a small number of hops from the requester than server only (SO) where the content is located only at the server outside the network as in the traditional TCP/IP based Internet architecture. This clearly demonstrates the benefit of integrated caching and routing that is a core part of the proposed converged architecture.

12.4.4.3 CNF and TCP/IP-Based Internet in Routing

The goal of this section is to compare the performance of the proposed converged architecture (referred to as CNF architecture) with that of the traditional TCP/IP-based Internet architecture when the converged architecture uses storage-aware intelligent routing. To compare

Table 12.2 Static network with intermittently failing links.

Protocol	OLSR			STAR		
Mean Inter-arrival time (seconds)	50	10	1	50	10	1
Files transmitted/Delivered	149/147	320/265	1016/691	165/164	468/438	1147/862
Average File Delay (seconds)	0.678	1.047	2.936	1.634	10.883	5.518
File Stream throughput (Mbps)	3.868	3.187	2.305	3.546	3.755	3.281
Network throughput (Mbps)	0.302	0.548	1.417	0.337	0.897	1.765

the performances of these two networks, storage aware routing (STAR) scheme proposed for CNF architecture in [28] is compared with the traditional OLSR routing protocol in a 25-node network in two cases: (i) a static network in a 500 m $\times$ 500 m grid with intermittently failing links and (2) a network in which the nodes move according to Truncated Levy Walk in 2500 $\times$ 2500 area [28].

Tables 12.2 and 12.3 show the number of files transmitted and delivered, average file transfer delays, file streaming throughput and overall network throughput. File streaming throughput represents the average physical data rate used to transfer the file at every hop. Unlike the network throughput, streaming throughput does not include delays incurred in queues and storage.

From Table 12.2 showing results for the static case (case 1), it can be observed that the STAR protocol is able to deliver a larger percentage of files that are admitted into the network, and this is because STAR selects faster paths for transmission. The average file delays as well as streaming throughput in STAR are larger showing the preference for storage over using slow transmission channels. Although file delays are high, the cache-and-forward concept improves the overall network throughput.

From Table 12.3 showing results for the mobile case (case 2), it can be observed that the file delivery fraction achieved by STAR is 17 and 33% more when the mean inter-arrival times are 10 and 50 s respectively. The average file delays are much lower (or equivalently, the file stream throughputs are much higher) in STAR compared to OLSR. These results indeed justify the benefits of STAR in DTN like mobility models.

12.4.4.4 CNF and TCP/IP Based Internet in Multi-Hop Mobile Wireless Networks

The goal of this section is to compare the performance of the proposed converged architecture (referred to as CNF architecture) with that of the traditional TCP/IP-based Internet architecture

Table 12.3 Performance with levy mobility model.

Protocol	OLSR		STAR	
Mean Inter-arrival time (seconds)	10	50	10	50
File Delivery Fraction	72.34	66.67	89.66	79.17
Average File Delay (seconds)	100.65	109.65	92.28	38.66
File Stream throughput (Mbps)	1.09	1.39	1.7	1.59

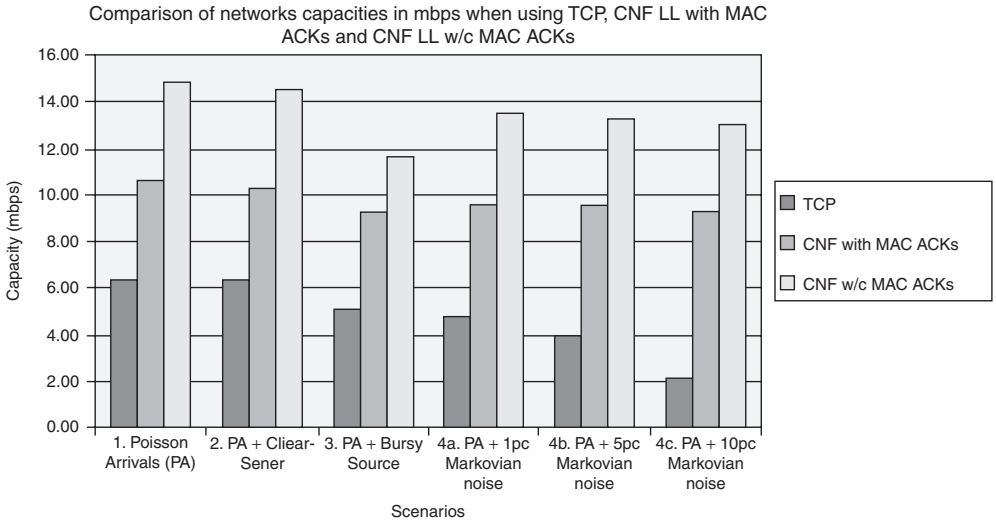


Figure 12.12 Comparison of network capacity (TCP versus CNF LL).

when the network consist of multiple wireless hops in an end-to-end connection. To compare the performances of these two networks, a 49-node grid topology was considered under variety of traffic and noise patterns as shown in Figure 12.12 to observe the effect on average file-transfer delay [7].

Since the TCP's performance over wireless without MAC layer reliability only becomes worse, the network capacity achieved by TCP/IP without MAC level reliability is not shown. It can be seen that for case 1, the network capacity offered by CNF link layer (LL) with MAC-level ACKs is about 70% higher than that offered by TCP. Disabling the MAC-level ACKs increases the CNF capacity gains to 140% over TCP. For case 2, the client server model, the capacities achieved remain approximately the same for all three strategies, that is TCP, CNF LL with MAC ACKs and CNF LL without MAC ACKs. Hence the capacity gains of CNF over TCP also remain the same. For case 3, the bursty source model, it can be seen that noticeable reductions occur in the capacity achieved by the CNF LL protocol with and without MAC ACKs. This is because burstiness in traffic causes congestion at nodes. Link Layer queue sizes increase and the average file delay becomes longer. Capacity gains of 85 and 130% can be seen for CNF LL with MAC ACKs and CNF LL without MAC ACKs respectively over TCP. For case 4, where Markovian noise was introduced in the links, it should be noted that CNF is much more noise resilient as compared to TCP. For example, in 10% Markovian noise TCP shows a two-thirds reduction in capacity while CNF LL without MAC ACKs suffers from 13% reduction in capacity. The capacity gains achieved by CNF LL over TCP in this case become about 650%.

12.5 Summary

This chapter was motivated by networking scenarios driven by intermittent connectivity, content, and mobility, which are not effectively handled by the traditional TCP/IP-based

Internet. The proposed converged architecture to deal with these networking scenarios consists of in-network persistent storage, and hop-by-hop reliable transport in the data plane; name resolution, late binding, and routing in the control plane. While these architectural components can be built as an overlay on the core networking (TCP/IP in the Internet) infrastructure, in a clean-slate network, these should be built into the core network itself. Given the exponentially dropping cost of processing/MIPS and storage/GB, it is highly conceivable that the network itself will consist of network elements (or future routers) with a significant amount of persistent storage and a significant amount of processing power, so that the route would be computed in real-time at each node (rather than being computed in the background and the forwarding table used in real-time for forwarding packets) based on multiple dimensions, such as congestion in the network, available storage in the network elements, availability of cached content, and so on. Thus it makes a lot of sense for the next-generation clean-slate network architecture to encompass the support for intermittent connectivity, content, and mobility right in the network fabric as these themes together have a very broad scope in the overall area of networking.

References

[1] SQUID Cache: http://www.squid-cache.org (accessed June 9, 2010).

[2] Appliansys: Server Appliances: http://www.appliansys.com (accessed June 9, 2010).

[3] Akamai: http:///www.akamai.com (accessed June 9, 2010).

[4] Limelight Networks: http://www.limelightnetworks.com (accessed June 9, 2010).

[5] Delay-Tolerant Networking Architecture, IETF RFC 4838.

[6] Gopal, S. and Paul, S. (2007) *TCP Dynamics in 802.11 Wireless Local Area Networks*. IEEE International Conference on Communications, June 2007.

[7] Saleem, A. (2008) Performance Evaluation of the Cache and Forward Link Layer Protocol in Multihop Wireless Subnetworks. Master's Thesis, Rutgers University, WINLAB.

[8] Krishnan, R., Basu, P., Mikkelson, J.M. *et al.* (2007) The SPINDLE Disruption Tolerant Networking System. Proceedings of IEEE Milcom, Orlando, FL, October 29–31.

[9] Guo, S., Falaki, M.H., Oliver, E.A. et al. (2007) Very low-cost internet access using KioskNet. *ACM Computer Communication Review*, October.

[10] Guo, S., Falaki, M.H., Oliver, E.A. *et al.* (2007) *Design and Implementation of the KioskNet System*. International Conference on Information Technologies and Development, December 2007.

[11] Balasubramanian, A., Mahajan, R., Venkataramani, A. *et al.* (2008) *Interactive WiFi Connectivity for Moving Vehicles*. Proceedings of the ACM SIGCOMM, pp. 427–438, August 2008.

[12] Banerjee, N., Corner, M.D., Towsley, D. and Levine, B.N. (2008) *Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure*. Proceedings of the ACM Mobicom, pp. 81–91, San Francisco, CA, USA, September 2008.

[13] Soroush, H., Banerjee, N., Balasubramanian, A. *et al.* (2009) *DOME: A Diverse Outdoor Mobile Testbed*. Proceedings of the ACM Intl. Workshop on Hot Topics of Planet-Scale Mobility Measurements (HotPlanet), June 2009.

[14] Hui, P., Chaintreau, A., Gass, R. et al. (2005) *Pocket Switched Networking: Challenges, Feasibility and Implementation Issues*. Proceedings of the Second IFIP Workshop on Autonomic Communications 2005, Athens, Greece, October 2005.

[15] Hui, P., Chaintreau, A., Scott, J. et al. (2005) *Pocket Switched Networks and the Consequences of Human Mobility in Conference Environments*. Proceedings of the SIGCOMM 2005 Workshop on Delay Tolerant Networking, Philadelphia, USA, August 2005.

[16] Paul, S., Yates, R., Raychaudhuri, D. and Kurose, J. (2008) *The Cache-And-Forward Network Architecture for Efficient Mobile Content Delivery Services in the Future Internet*. Proceedings of the First ITU-T Kaleidoscope Academic Conference on Innovations in NGN: Future Network and Services, 2008.

[17] Dong, L., Liu, H., Zhang, Y. *et al.* (2009) *On the Cache-and-Forward Network Architecture*. D. IEEE International Conference on Communications, 2009, Volume, Issue, 14–18 June 2009, pp. 1–5.

[18] YouTube. http://www.youtube.com (accessed June 9, 2010).
[19] Revver. http://www.revver.com (accessed June 9, 2010).
[20] Bakre, A. and Badrinath, B.R. (1995) *I-TCP: Indirect TCP for Mobile Hosts*. Proceedings of the 15th Int'l Conference on Distributed Computing Systems, May 1995.
[21] Gopal, S. and Raychaudhuri, D. (2005) *Experimental Evaluation of the TCP Simultaneous-Send problem in 802.11 Wireless Local Area Networks*. ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND), Conference held in Philadelphia, USA in August 2005.
[22] Gopal, S., Paul, S. and Raychaudhuri, D. (2005) *Investigation of the TCP Simultaneous-Send problem in802.11 Wireless Local Area Networks*. Proceedings of the IEEE Computer and Communications Conference (ICC)2005, Vol. 5. Conference held in Seoul, South Korea, 16–20 May 2005, pp. 3594-3598.
[23] Gopal, S., Paul, S. and Raychaudhuri, D. (2007) *Leveraging MAC-layer Information for Single-hop Wireless Transport in the Cache and Forward Architecture of the Future Internet*. The Second International Workshop on Wireless Personal and Local Area Networks (WILLOPAN) held in conjunction with COMSWARE 2007, Bangalore, India, 12 January 2007.
[24] Zhao, W., Ammar, M. and Zegura, E. (2004) *A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks*. Proceedings of ACM Mobihoc 2004, Tokyo Japan, May 2004.
[25] Zhao, W. and Ammar, M. (2003) *Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks*. Proceedings of the IEEE Workshop on Future Trends in Distributed Computing Systems, Puerto Rico, May 2003.
[26] Liu, H., Zhang, Y. and Raychaudhuri, D. (2009) *Performance Evaluation of the "Cache-and-Forward (CNF)" Network for Mobile Content Delivery Services*. IEEE International Conference on Communications (ICC) Workshops 2009, 14–18 June 2009.
[27] Dong, L., Zhang, Y. and Paul, S. Performance Evaluation of In-network Integrated Caching. Submitted for publication.
[28] Shinkuma, R., Jain, S. and Yates, R. (2009) Network Caching Strategies for Intermittently Connected Mobile Users, IEEE PIMRC, Sept 2009.
[29] Jain, S., Saleem, A., Liu, H. *et al.* (2009) *Design of Link and Routing Protocols for Cache-and-Forward Networks*. IEEE Sarnoff Symposium 2009, Princeton, NJ.
[30] Cerf, V., Burleigh, S., Hooke, A. *et al.* (2007) Delay-tolerant network architecture. RFC 4838, April, MILCOM 2007.
[31] Acharya, A., Ganu, S. and Misra, A. (2006) *DCMA: A Label Switching MAC for Efficient Packet Forwarding in Multihop Wireless Networks*. IEEE JSAC Special Issue on Wireless Mesh Networks, November 2006.
[32] Wu, Z., Ganu, S. and Raychaudhuri, D. (2006) IRMA: Integrated Routing and MAC Scheduling in Wireless Mesh Networks. Proceedings of the Second IEEE Workshop on Wireless Mesh Networks, (WiMesh), Reston, 2006.
[33] Akyildyz, I., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) A survey on wireless sensor networks. *IEEE Communications Magazine*, August.

13

Summary of Part Two

Part Two of the book explored the challenges in delivering video over an "open" network where the video-over-Internet service provider has no access or control of the underlying network infrastructure.

Chapter 7 was dedicated to movie- (video)-on-demand services over the Internet. Specifically, various alternative architectures from technology and business perspectives for providing movie-on-demand (or in general, video-on-demand) services were explored from the viewpoint of video-over-Internet service providers. After computing the resource requirements (storage and bandwidth) for a service provider for providing movie-on-demand services, three fundamentally different architectures were compared: the content distribution network (CDN), hosting and peer-to-peer (P2P). It was shown that P2P scales better than CDN and has a lower cost of operation because the resources (storage and bandwidth) already paid for by the end users are leveraged in this model. While P2P is economically the best option, there are challenges, such as security, reliability, quality of service, and digital rights management for copyrighted content, which are not properly addressed in this model. Industry has been trying a combined CDN-P2P model to obtain the best of both worlds. It was shown how even P2P networks would benefit from the usage of caches in the network.

Chapter 8 focused on Internet television, which enables streaming of nontraditional video content (meaning the content not typically beamed over satellite or cable TV networks) to end-users' PCs. Video-over-Internet service providers would be able to source much more independent content compared to that done by cable/satellite service providers and would be able to stream them as television shows just as traditional shows are televised in cable/satellite networks. Alternative architectures encompassing P2P streaming, tree-based streaming and mesh-based streaming for providing such a service were discussed after computing the resource requirements (storage and bandwidth) from the viewpoint of video-over-Internet service providers. The main benefit of tree-based and mesh-based streaming over one-to-one P2P streaming is scalability, the ability to stream in real time the same content to multiple recipients with acceptable quality of user experience. Finally, it was shown how P2P traffic, which is detrimental to the ISPs' network, can be transported over a traffic-engineered ISP network using the P4P framework.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

Chapter 9 was dedicated to broadcast television over Internet. Consumers would expect to see the same television channels as are available in cable and/or satellite television networks in addition to using the video-on-demand service. It was shown that to compete with a digital cable/satellite television service provider that replaces 100 analog TV channels with digital channels, a video-over-Internet service provider would have to serve 500 digital channels over the Internet leading to aggregate bandwidth requirement of approximately 2–3 gbps. Without using multicast technology, this bandwidth requirement is difficult, if not impossible, to meet. Technology to broadcast live TV stations over the Internet was also described followed by some software products that need to be downloaded for watching live TV over a broadband Internet connection.

Chapter 10 covered digital rights management (DRM). Specifically, we saw how DRM is primarily driven by the content owners to preserve the "ownership" of content and ensuring payment of a "royalty" to the content creators based on use of respective content. The functional architecture of DRM was explained along with modeling of content and digital rights expression. In addition, the steps involving license creation, license distribution and license acquisition together with content packaging, content distribution and content display (media playing) were described to explain how DRM works in real life.

Chapter 11 focused on quality of experience (QoE). The entire chapter was dedicated to detailed design of a QoE cache that can be deployed in a wireless network to improve quality of experience for video. Four main design principles of the QoE-Cache are: (i) content needs to be moved closer to the end user by caching it locally (Web proxy/cache) so that the probability of its being subjected to congestion or packet loss is reduced significantly; (ii) TCP can be further optimized in cellular wireless networks to deal with variable round-trip time, time-varying packet error rate and potentially high packet error rate during handoff or poor RF condition; (iii) media stream transmission needs to be adapted to the condition of the wireless network and (iv) DNS queries need to be eliminated over the air link. An implementation instance of a QoE cache based on Windows Media Server was described, drawbacks pointed out and a potential solution was prescribed. Then algorithms based on fundamental insights were provided to further enhance the performance of the QoE cache and the performance was evaluated. The chapter concluded with additional features and optimizations that are possible for the QoE cache.

Chapter 12 is dedicated to a new networking paradigm for the efficient distribution of video. The new paradigm of cache-and-forward networking is motivated by scenarios driven by intermittent connectivity resulting mostly from mobility and nonavailability of network connectivity. Such situations are not effectively handled by the traditional TCP/IP-based Internet. The converged architecture proposed to deal with these networking scenarios consists of in-network persistent storage, and hop-by-hop reliable transport in the data plane; name resolution, late binding and routing in the control plane. While these architectural components can be built as an overlay on the core networking (TCP/IP in the Internet) infrastructure, in a clean-slate network, these should be built into the core network itself. Given the exponentially dropping cost of processing/MIPS and storage/GB, it is highly conceivable that the network itself will consist of network elements (or future routers) with a significant amount of persistent storage and a significant amount of processing power so that the route would be computed in real time at each node (rather than being computed in the background and the forwarding table used in real time for forwarding packets) based on multiple dimensions such as congestion in the network, available storage in the network elements, availability of cached content and

so on. Thus it makes a lot of sense for the next-generation clean-slate network architecture to encompass the support for intermittent connectivity, content and mobility right in the network fabric as these themes together have a very broad scope in the overall area of networking.

In summary, several models of video delivery including CDNs, P2P networks, hybrid CDN and P2P networks, cache-based P2P networks and P4P networks were explored in fair amount of details. Resource (bandwidth and storage) needs for supporting various models, such as movie on demand, Internet television and broadcast television were computed and variations of P2P technology for streaming, such as tree-based P2P streaming, mesh-based P2P streaming, CoolStreaming and GridMedia were covered, identifying the advantages and disadvantages of the various techniques. Real-world software systems enabling live television on broadband-connected PCs were described, exposing what can and cannot be done with such systems. Given the importance of QoE in video streaming, the design of a system called the QoE cache was presented in details covering each component of the system individually. New insights were discussed based on an implementation instance of the QoE cache and the details of how those insights can be incorporated in the design of an enhanced QoE cache were highlighted. Given that video distribution is going to dominate the use of networks, as is evident from the success of YouTube and other video-sharing applications, the networks need to have special characteristics, such as support of storage within the network and the capability of delivering video from the point nearest to the requestor. Furthermore, traditional TCP/IP-based networks would not be able to deliver video content when the users (either the originator or the recipients or both) are mobile and are not connected to the network for the entire duration. Based on the above observations, architectures, such as, cache-and-forward networks will become mainstream and opportunistic delivery will be important.

Part Three

Challenges for Distributing Video in CLOSED Networks

Challenges faced in "open" networks for video distribution are mostly centered around the "video servers", where the goal is to improve the quality of experience by adjusting parameters, such as, encoding rate, frame rate and quantization steps, and by increasing the availability of service via content distribution networks where the video servers are cloned to serve requests locally in a distributed manner.

In sharp contrast, the challenges in a "closed" network are all about optimizing the "network" resources and leveraging the power of the network to offer better quality of experience. Specifically, the telecom operators who own the networks can engineer their network in a way that video traffic gets "special" treatment compared to standard data traffic and the "special" treatment could be in the form of "reserved" bandwidth and resources, priority scheduling at the network elements, leveraging IP multicast or point-to-multipoint MPLS connections and so on. In addition to all of these optimizations, telecom operators can leverage everything that an "open" network service provider can do, such as adjusting the parameters at the video servers (encoding rate, frame rate, etc.) and cloning the video servers in geographically distributed locations in order to improve availability and serve requests locally without overloading the core network.

The goal of a telecom operator (owner of a "closed" network) is to offer the following services to its customers:

- video on demand;
- broadcast television (cable/satellite television-equivalent service);
- same video and television programs on any device (TV, PC, Mobile) at any place (home, work or in transit) at any time.

Furthermore, the networks available to the telecom operator to optimize and deliver video are:

- wired networks, such as, DSL, Ethernet, MPLs;
- wireless networks, such as, 3G, 4G, WiFi, DTV;
- a combination of these.

While these networks used to exist and operate in silos, most recently, they are converging into a single "converged" network as described next. It is because of the "converged" network architecture, telecom operators are able to provide the above-mentioned services to their customers in a seamless manner.

14

Network Architecture Evolution

Traditionally, the networks providing communication services (phone), entertainment services (television), Internet services (data) and mobile services (cellular) have existed in silos in that there have been separate engineering and operations support in addition to separate service platforms for providing value-added services in their own networks.

However, the telecom (cable) operators realized that maintaining separate network and service infrastructures for each service is expensive both from a capital expenses perspective as well as from operational expenses standpoint. Furthermore, due to the segregation of the networks, providing value-added services spanning multiple networks is either not possible or is too expensive and time consuming from go-to-market perspective.

Furthermore, there is a huge threat to the telecom operators from the over the top (OTT) players, such as, Google, Yahoo, Amazon, eBay and others who have been nimble enough to roll out innovative services very quickly for the end users, leaving the telecom operators just providing connectivity services.

In other words, telecom operators, whose core business was "voice" services, transformed themselves to offer "data" services via broadband DSL rollout. Moreover, to stay competitive with Cable operators, they have been further transforming their network and service infrastructure to provide video and mobile services as well, leading to what is popularly referred to as quadruple play (voice, video, data and mobile).

However, the irony is, while the telecom operators have been spending significantly on the upgrade of their networking infrastructure, the benefits are being reaped in a disproportionate manner by the OTT players, leaving just the broadband access revenue with the telecom operators.

Therefore, to compete with the OTT players, telecom operators have been spending significant resources to develop a "converged" network for providing "quadruple" play services in a seamless manner. Furthermore, such "converged" network architecture would enable the telecom operators to blend services across networks very quickly in creative ways so as to extract some value from the end-to-end value chain beyond just connectivity revenue. Convergence in network is possible because of Internet protocol (IP) that is gluing together presumably disparate networks. Convergence at the network level is not enough. There has to be a common

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

infrastructure for “policies”, “sessions”, and “services”. In the parlance of telecom, IP multimedia subsystem (IMS) provides that common infrastructure.

Once the converged network and service infrastructure is in place, telecom operators can start to provide the video services, such as, video-on-demand, broadcast television and interactive services across multiple networks over multiple devices. The next section describes the core video service from telecom operators, popularly known as IP Television (IPTV).

15

IP Television (IPTV)

Internet protocol television (popularly known as IP television or IPTV), is the offering of television services by telecommunication companies (telcos) [1–3]. As indicated earlier, it is the equivalent of cable/satellite TV operated by the telcos. The main difference is that the telcos offer IPTV services over an IP-based transport network, as opposed to proprietary transport networks of the cable/satellite operators. Due to the bidirectional characteristics of the IP network, Telcos can not only offer the traditional broadcast television services, but can also offer interactive video services. This chapter describes in detail the architecture and protocols used in IPTV networks in addition to alternative ways of distributing live video content, one using traffic engineered MPLS network and the other using a combination of best effort network and forward error correction.

15.1 IPTV Service Classifications

IPTV offers three types of services: (i) basic channel service, which is equivalent to broadcast services, (ii) selective channel service, which is equivalent to on-demand video services and (iii) interactive data services, which are equivalent to Internet services in principle (Table 15.1).

Broadcast service is not limited to video. Audio broadcasts (radio-like services) as well as data broadcasts (local information) are supported by basic channel service. Video-on-demand also comes in multiple flavors. For example, there could be real video-on-demand or near video-on-demand (similar to pay per view with scheduled broadcast) or personal video recording (PVR) or electronic program guide (EPG)-based recording services. Interactive services are expected to enable TV viewers to interact with the services on television for gathering information (weather, traffic etc.), for doing transactions (home shopping, banking etc.), for communication (messaging, email, chatting), for entertainment (sharing photo album, playing games) and/or for learning (education courses). These are captured in Table 15.1.

15.2 Requirements for Providing IPTV Services

There are standards for IPTV terminology in order to promote a common understanding among IPTV equipment manufacturers, service providers and all vendors in the ecosystem.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

Table 15.1 IPTV service classifications.

Classifications	Services
Basic Channel Service (*Broadcast*)	Audio and video (for SD/HD) Audio only Audio, video and data
Selective Channel Service (*On Demand Video*)	Near VoD broadcasting Real VoD EPG PVR Multi-angle service
Interactive *Data* Service	T-information (news, weather, traffic and advertisement) T-commerce (security, banking, shopping, auction and ordered delivery) T-communication (mail, messaging, SMS, channel chatting, video conference and video phone) T-entertainment (photo album, game, karaoke and blog) T-learning (education for children, elementary, middle and high school student, languages and estate)

The requirements with respect to display quality, such as, bandwidth, audio/video compression formats, resolution are also specified in the standards. Transport mechanisms defined in the standards encompass data encapsulation, transmission protocols, quality of service, service delivery platform, content distribution and forward error correction (FEC). Metadata formats, transport and signaling are keys to providing advanced services, such as, search capabilities in television. Table 15.2 summarizes the requirements for providing IPTV services.

15.3 Displayed Quality Requirements

15.3.1 Bandwidth

To provide a very high quality of experience for end users, IPTV services require high transmission rates and hence high bandwidth capacities from the underlying network. The

Table 15.2 Requirements for providing IPTV services.

Displayed Quality	Bandwidth Audio/video compression formats Resolution (picture quality)
Transport	Data encapsulation Transmission protocols Quality of service Service platform design for QoS Content distribution Forward error Correction (FEC)
Middleware	Integrated middleware
Metadata	Metadata formats Metadata transport and signaling

transmission rate depends on the compression and coding technology used. For example, for MPEG-2 coded standard definition (SD) video on demand (VoD) stream or IPTV stream per one TV channel, 3.5 Mbits/s to 5 Mbits/s is desirable. For H.264 (or MPEG-4 part 10), SD VoD or IPTV stream per one channel, the desired bandwidth is 2 Mbits/s. High definition (HD) video streams using H.264 coding requires 8–12 Mbits/s.

15.3.2 Audio/Video Compression Formats

Video compression techniques used in IPTV are MPEG-2, MPEG-4 or H.264 in both SD and HD. Windows Media using VC1 is also used as a popular video codec.

H.264, also known as MPEG-4/AVC (advanced video coding), is a video compression standard created by the Joint Video Team (JVT), which consists of the International Telecommunication Union (ITU) and the Moving Picture Experts Group (MPEG). The ITU has adopted the name H.264 for the standard, whereas ISO/IEC refers to it as the MPEG-4/AVC (part 10) standard (document ISO/IEC 14 496-10).

MPEG-4/AVC is largely based on the two predecessor families MPEG-1 and H.261. Major changes have been made to deliver significantly higher efficiency, for example by taking special measures to make decoding faster. The resulting process makes cost-effective implementation possible for terminal equipment. To improve interoperability, the concepts of profile and level have been introduced. A profile defines the supported features, while a level specifies a limit for a variable such as picture resolution.

Audio compression techniques used in IPTV are advanced audio codec (AAC), high efficiency advanced audio codec (HE-AAC), low complexity AAC (LC-AAC), advanced audio coding scalable sampling rate (AAC-SSR), bit sliced arithmetic coding (BSAC).

Advanced audio codec is a lossy compression method for digital audio data that was developed by the MPEG working group. It supports up to 48 channels with full bandwidth as well as additional channels for data and low frequency effects (LFE) in one stream. AAC is used in digital TV, online music sales, as well as for mobile audio players and game consoles.

High efficiency advanced audio codec is a lossy compression method for digital audio data that requires a license. HE-AAC is a next-stage development of the AAC standard and supports up to 48 channels with full bandwidth as well as additional data. It also delivers relatively good results at low bit rates, making it especially well suited for streaming applications and mobile telephony.

Two versions of HE-AAC are now available: HE-AAC v1 and HE-AAC v2. In contrast to version 1, version 2 additionally uses a method named parametric stereo (PS) when compressing stereo signals by means of which stereo signals can be compressed more effectively. Parametric stereo therefore does not offer any advantage for mono signals.

15.3.3 Resolution

Standard definition television (SDTV) has the following resolution:

- 480i (NTSC uses this; 480i/60 split into two interlaced fields of 243 lines).
- 576i (PAL uses this; 720 × 576 is split into two interlaced fields of 288 lines).

Enhanced definition television (EDTV) has the following resolution:

- 480p (720 × 480 progressive scan).
- 576p (720 × 576 progressive scan).

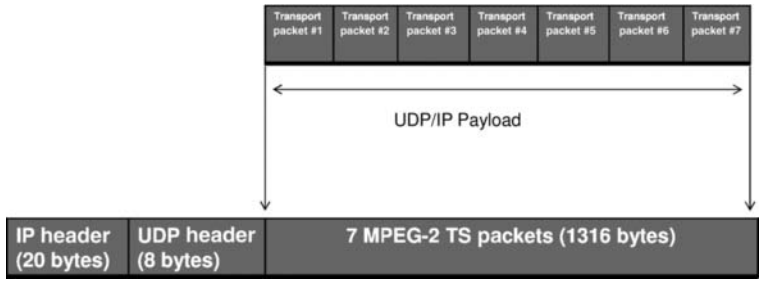


Figure 15.1 Encapsulation of MPEG-2 transport stream packet in UDP/IP packet.

High definition television (HDTV) has the following resolution:

- 720p (1280 × 720 progressive scan).
- 1080i (1920 × 1080 split into two interlaced fields of 540 lines).
- 1080p (1920 × 1080 progressive scan).

15.4 Transport Requirements

15.4.1 Data Encapsulation

Digital media can be encapsulated and transported in two different ways:

1. Encapsulation of MPEG-2 transport stream (TS) packets in UDP/IP: MPEG-2 TS consists of a sequence of 188 byte transport packets (as described in the next subsection) [38–40]. The simplest way of transporting these packets would be to pack seven of these packets into the payload of an IP packet. Ethernet's MTU size is 1500 bytes and hence seven MPEG-2 TS packets can fit into a single Ethernet frame as shown in Figure 15.1.

   This approach works well in a "closed" network where traffic can be controlled and sufficient bandwidth can be provided for maintaining Quality of Service. However, in an "open" network, such as the Internet, MPEG-2 TS packets need to be encapsulated in RTP packets and then transported over the IP network.
2. Encapsulation of MPEG-2 TS packets in RTP/UDP/IP: MPEG-2 TS packets are encapsulated in RTP packets and carried over IP networks as shown in Figure 15.2.

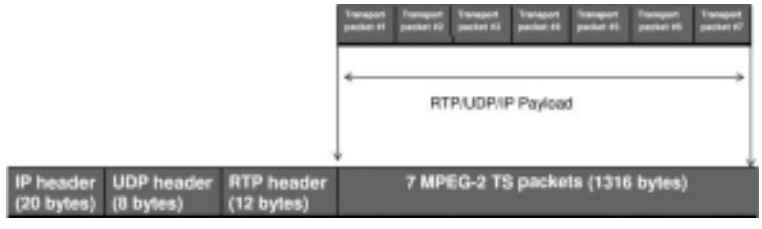


Figure 15.2 Encapsulation of MPEG-2 transport stream packet in RTP/UDP/IP packet.

This approach is needed for transporting video over an "uncontrolled" network, such as the Internet.

15.4.2 Transmission Protocols

There are several ways of transporting video over IPTV networks (Figure 15.3). Physical transport can happen over variety of media, such as, fiber, wireless, DSL, satellite, cable and so on. However, Internet protocol is used at the network layer across all types of physical media. For point-to-point communication, such as for video on demand, IP is used in unicast mode while for point-to-multipoint communication, such as, for live TV, IP is used in multicast mode. Furthermore, video could be transported either using TCP or UDP. Typically, progressive download or download-and-play mode of IPTV uses HTTP/TCP as the transport layer protocol. An alternative reliable delivery mechanism for video over multicast networks is to use file delivery over unidirectional transport or FLUTE [12] protocol, which leverages asynchronous layered coding (ALC) for scalable reliable multicast. Most commonly, video is transported in IPTV in "streaming" mode. Audio/video data is encapsulated in MPEG-2 TS, which could either directly go over UDP/IP or leverage RTP/UDP/IP. Note that streaming mode of transport could happen either in unicast mode or in multicast mode. Internet protocol multicast is used for point-to-multipoint streaming and that requires the receiving end-points of IPTV transmission to join relevant IP multicast groups using Internet group management protocol (IGMP) [13–19]. The receivers of the IPTV stream provide feedback about the quality of reception to the server using RTP control protocol (RTCP) [22, 41]. Based on the feedback, the servers can adjust the rate of streaming either by changing encoding rate or frame rate and thereby improve the quality of experience for end users. In the case of video on demand service, there is a need to support trick-play (fast forward, rewind, pause etc.) and the signaling

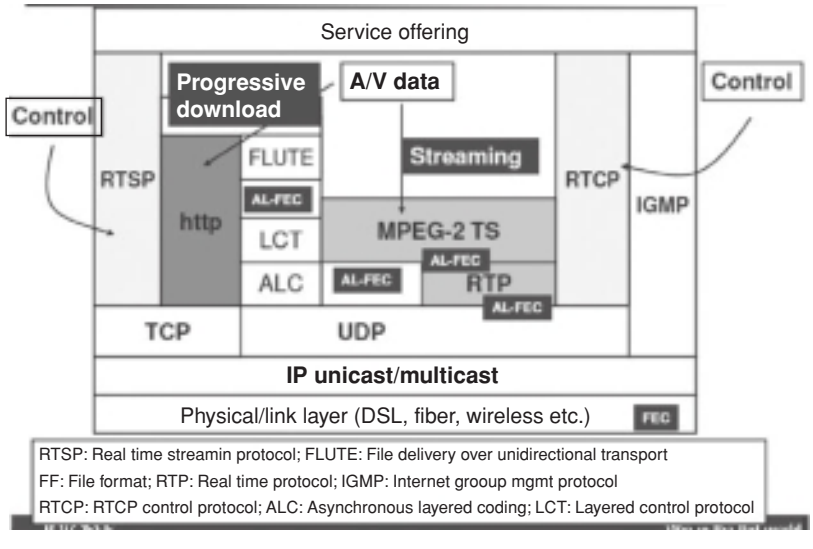


Figure 15.3 Transmission protocols.

to control the stream is done using real-time streaming protocol (RTSP) [23]. Commonly used protocols in IPTV are briefly introduced next.

15.4.2.1 IP Multicast and Internet Group Management Protocol (IGMP)

Internet protocol multicast [20, 42] is a scalable and efficient of way of distributing high-bandwidth content, such as, video over an IP-based network. The beauty of IP multicast is that the replication of content (packets) is done optimally in order to avoid unnecessary traffic generation. Essentially, IP multicast sets up a distribution tree from the source to the destinations or from an aggregation point (commonly referred to as rendezvous point or RP) to the destinations. The most commonly used protocol for setting up a source-based multicast tree is PIM SSM [21] in which the IP routers, starting from the ones closest to the destination hosts, send JOIN messages corresponding to a source-group pair (S,G) towards the source using the reverse shortest path, and each router along the way creates a routing table entry corresponding to (S,G) identifying the incoming interface of the JOIN message as the outgoing interface for the multicast traffic flowing downstream from the source via the corresponding routers. In some situations, the source corresponding to a multicast group may not be known in advance and hence PIM SSM cannot be used for setting up the multicast tree. In that case, PIM SM [21] is used as the multicast routing protocol. In PIM SM, the IP routers, starting from the ones closest to the destination hosts, send JOIN messages corresponding to an IP multicast group (*,G) towards the Rendezvous Point (RP) using the reverse shortest path, and each router along the way creates or augments (if the entry already exists) a routing table entry corresponding to (*,G) identifying the incoming interface of the JOIN message as the outgoing interface for the multicast traffic flowing downstream from the source via the RP through the corresponding routers. Just as the destinations (receivers) join the IP multicast group by sending JOIN messages through the routers towards the RP, the sources (senders) via their nearest router register to the IP multicast group by sending REGISTER messages to the RP, in the process creating state for (*,G) along the route from the sender to the RP. Thus RP indeed becomes the rendezvous point for the senders and receivers.

While PIM SSM and PIM SM set up the IP multicast tree in the network spanning the routers at the core and the routers at the edge connecting the senders and receivers, Internet group management protocol (IGMP) is used by the multicast group members (receivers or destinations) to intimate their nearest routers about their interest in joining a specific group (G). IGMP v1/2 allowed the members to issue IGMP JOIN (G) messages to their nearest router while IGMP v3 allows the members to specify the source in addition to the group of interest via IGMP JOI (S, G) message. Once the routers attached to the members are intimated about the multicast group join, the routers initiate the PIM JOIN (*, G) message corresponding to IGMP v1/2 for PIM SM and PIM JOIN (S, G) message corresponding to IGMP v3 for PIM SSM.

15.4.2.2 Real Time Transport Protocol (RTP) and RTP Control Protocol (RTCP)

Real-time transport protocol (RTP) was developed by the Audio/Video Transport working group in the Internet Engineering Task Force (IETF). RTP standard [22] defines a pair of

protocols, RTP and Real-Time Transport Control Protocol (RTCP). While RTP is used for transporting multimedia data, RTCP is used for periodically sending quality of service control parameters.

Real-time transport protocol is designed for end-to-end, real-time transport of audio and video data taking into account unique requirements of audio/video transport. For example, RTP allows the recipients of audio/video data to compensate for the jitter and gaps in sequence that may occur during the transport of packets over an IP network. It also supports point-to-multipoint distribution of audio/video data using IP multicast, and in the true spirit of IP networks, does not provide any delivery guarantee (best effort delivery). However, by defining media payload type, incorporating sequence numbers for loss detection, and timestamps for enabling timing recovery, both reliability and error recovery can be built into the application layer. Depending on the profile, and payload format in use, rules for timestamps and sequence number usage would be different in each case.

RTP control protocol (RTCP) provides reception quality, feedback, participant identification and synchronization between media streams. While RTP packets are sent every few milliseconds, the control protocol operates on the scale of seconds. RTCP traffic is small compared to the RTP traffic, typically around 5% [22].

An RTP session consists of an IP address and a pair of destination ports (one for RTP and the other for RTCP). In a typical multimedia session, audio and video streams are carried in a separate RTP session, and each session has its own RTCP packets reporting the reception quality for that session. Usually, session description protocol (SDP) is used to define RTP sessions and to negotiate session parameters between the involved parties.

There are several different formats in which audio and video can be encoded. RTP provides sufficient information in its header to allow applications to interpret the payload. Specifically, the RTP header contains a "payload type" that indicates the format of the payload, and determines its interpretation by the application. In general, each payload type identifies a "profile" and a "format", some of which are statically defined in RFC 3551.

The RTP packets are usually carried in UDP/IP envelope, and hence the RTP packet looks like:

IP Header	UDP Header	RTP Header	RTP Payload

RTP captures the unique aspects of audio/video transport via the use of timestamps, and the type of payload reflecting the specifics of audio/video codecs used in compressing and encoding the audio/video data included in the payload. The details of RTP packet header are given below:

Bit offset	0-1	2	3	4-7	8	9-15	16-31
0	V	P	X	CC	M	PT	Sequence number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifier (optional)						

- Version (V): 2 bits. This field identifies the version of RTP. The version is 2 up to RFC 1889.
- Padding (P): 1 bit. If the padding bit is set, the packet contains one or more additional padding octets at the end, which are not part of the payload. Padding is required by some encryption algorithms with fixed block sizes. It may also be needed for carrying multiple RTP packets in a lower-layer protocol data unit.
- Extension (X): 1 bit. If the extension bit is set, the fixed header is followed by exactly one header extension.
- CSRC count (CC): 4 bits. The CSRC count contains the number of contributing source (CSRC) identifiers that follow the fixed header.
- Marker (M): 1 bit. Used at the application level and defined by a profile. If it is set, it means that the current data has some special relevance for the application.
- Payload type (PT): 7 bits. This field identifies the format (e.g. encoding) of the RTP payload and determines its interpretation by the application. This field is not intended for multiplexing separate media.
- Sequence number: 16 bits. The sequence number increments by one for each RTP data packet sent and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable).
- Timestamp: 32 bits. Timestamp is used to enable the receiver playback the received samples at appropriate intervals. When several media streams are present, the timestamps are independent in each stream, and may not be relied upon for media synchronization. The granularity of the timing is application specific. For example, an audio application that samples data once every 125 μs could use that value as its clock resolution. The clock granularity is one of the details that is specified in the RTP profile or payload format for an application.
- SSRC: 32 bits. The synchronization source identifier uniquely identifies the source of a stream. The synchronization sources within the same RTP session will be unique.
- CSRC list: 0 to 15 items, 32 bits each. The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources.
- Extension header (optional): the first 32-bit word contains a profile specific identifier (16 bits) and a length specifier (16 bits) that indicates the length of the extension (EHL: Extension Header Length) in 32-bit units, excluding the 32-bits of the extension header.

15.4.2.3 Real-Time Streaming Protocol

Real-Time Streaming Protocol [23] was developed by the Multiparty Multimedia Working Group (MMUSIC WG) of the IETF and published as RFC 2326.

It is an application-level protocol for control over the delivery of data with real-time properties. It is used to control streaming media servers in cases where the media players issue VCR-like commands, such as, “play” “rewind”, “fast forward” and “pause”.

Actual transport of streaming data happens over RTP/UDP/IP and is not a task of RTSP. However, controlling the stream is indeed a function of RTSP. There is a lot of similarity between HTTP and RTSP, especially in syntax. However, while HTTP is a stateless protocol, RTSP is not. RTSP uses a session identifier to keep track of sessions. Usually, the RTSP messages are sent from the client to the server, particularly for VCR-like commands, however, there are some exceptions when the server sends messages to the client. Default port number of RTSP is 554.

Basic RTSP messages are listed below:

- OPTIONS
  This request can be issued at any time by the client.

  Example:

```
C->S: OPTIONS * RTSP/1.0
      CSeq: 1
      Require: implicit-play
      Proxy-Require: gzipped-messages

S->C: RTSP/1.0 200 OK
      CSeq: 1
      Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

- DESCRIBE
  This request is issued by the client to retrieve the description of a presentation or media object identified by the URL. In response to the request, server responds with a description of the requested resource. DESCRIBE request-response pair constitutes the initialization phase of RTSP.

  Example:

```
          C->S: DESCRIBE rtsp://generic.example.com/jumble/zoo
RTSP/1.0
          CSeq: 529
       Accept: application/sdp, application/mheg

        S->C: RTSP/1.0 200 OK
        CSeq: 529
        Date: 24 June 2009 21:13:12 GMT
        Content-Type: application/sdp
        Content-Length: 376

             v=0
             o=spaul 2345674526 2345672807 IN IP4 133.44.55.6
             s=Superbowl 2008
             i=Streaming video of last year's Superbowl
             u=http://www.mystreamingserver.com/Jan26-
2008/sdp.01.ps
             e=spaul@mycompany.com (Sanjoy Paul)
             c=IN IP4 230.1.22.33/127
             t=1234397496 12343404696
             a=recvonly
             m=audio 4567 RTP/AVP 0
             m=video 4466 RTP/AVP 31
```

- SETUP
  This request specifies how a single media steam must be transported. This must be done before a PLAY request is sent. The SETUP request contains the media stream URL and a

transport specifier including the local port numbers for receiving RTP and RTCP data/control packets. The reply from server confirms the chosen parameters, and fills in the server's chosen port numbers. Each media stream needs to be separately configures using SETUP before a PLAY request is sent for the multimedia stream.

Example:

```
       C->S: SETUP rtsp://generic.example.com/jumble/zoo/zoo.rm
RTSP/1.0
             CSeq: 519
             Transport: RTP/AVP;unicast;client_port=6677-6678

             S->C: RTSP/1.0 200 OK
             CSeq: 519
             Date: 24 June 2009 21:13:12 GMT
             Session: 12345678
             Transport: RTP/AVP;unicast;
                   client_port=6677-6678;server_port=7788-7789
```

- PLAY
  This request causes one or all media streams to be played. The URL may be an aggregate URL that plays all media streams or a single media stream URL that plays only that stream. A range may be specified with the request to indicate what portions of the specified stream should be played. In the absence of the range header, the stream will be played from the beginning to the end. If the PLAY request is issued after a media stream has been paused then the media plays from the paused instant.

  Example to show how PLAY requests can be sent for the server to first play seconds 20 through 30, then, immediately following, seconds 40 to 55, and finally seconds 65 through the end.

```
     C->S: PLAY rtsp://generic.example.com/audio RTSP/1.0
            CSeq: 123
            Session: 12345678
            Range: npt=20-30

     C->S: PLAY rtsp://generic.example.com/audio RTSP/1.0
            CSeq: 124
            Session: 12345678
            Range: npt=40-55

     C->S: PLAY rtsp://generic.example.com/audio RTSP/1.0
            CSeq: 125
            Session: 12345678
            Range: npt=65-
```

- PAUSE
  As the name suggests, this request temporarily halts one or all media streams. The paused media stream(s) can be resumed later with a PLAY request. The time to pause a media

stream is indicated using a range header. In the absence of a range header, pause happens immediately.

Example:

```
       C->S: PAUSE rtsp://generic.example.com/jumble/zoo/zoo.rm
RTSP/1.0
             CSeq: 567
             Session: 12345678

       S->C: RTSP/1.0 200 OK
             CSeq: 567
            Date: 24 June 2009 21:13:12 GMT
```

- RECORD
  This message initiates recording a range of media data according to the presentation description. The timestamp reflects start and end time (UTC). If no time range is given, the start or end time provided in the presentation description is used. If the session has already started, recording commences immediately.

  Example shows recording a conference session to which the media server was previously invited.

```
    C->S: RECORD rtsp://generic.example.com/meeting/audio.en
RTSP/1.0
          CSeq: 897
           Session: 12345678
           Conference: 130.11.22.33/45678910
```

- TEARDOWN
  This request is used to terminate a session. The TEARDOWN request stops all media streams and frees all session-related information for the media server.

  Example:

```
        C->S: TEARDOWN rtsp://generic.example.com/jumble/zoo
RTSP/1.0
              CSeq: 596
              Session: 12345678

        S->C:  RTSP/1.0 200 OK
               CSeq: 596
```

15.4.2.4 MPEG-2 Transport Stream (MPEG-2 TS)

Transport stream (TS) is a standard defined in MPEG-2 Part 1, Systems (ISO/IEC standard 13818-1) with the objective of transporting multiplexed audio/video/data streams over unreliable media [38–40]. There are built-in mechanisms for correcting errors resulting from

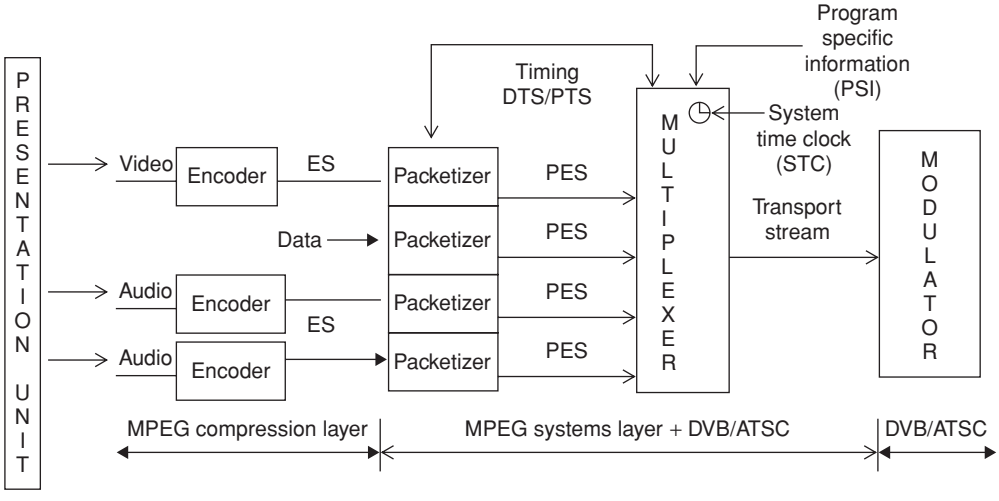


Figure 15.4 MPEG-2 transport stream creation.

transport over unreliable media, and for synchronization of audio, video and data streams at the receiver. Transport stream is used in broadcast applications such as DVB and ATSC.

Steps involved in creating a transport stream are shown in Figure 15.4:

1. Creating an Elementary Stream (ES)
   Presentation Units (audio and video) are first compressed to form access units (AUs). A sequence of AUs constitutes an elementary stream (ES). In general, there would be one video ES, and one or more audio ES.
2. Packetizing an Elementary Stream:
   a. Elementary streams are packetized to yield packetized elementary stream (PES).
   b. Packetized elementary stream content is then distributed among a number of fixed-size Transport Packets.
   c. Timing information is added to PES and Transport Packets for the purpose of synchronization.
   d. Program specific information (PSI) is added to transport packets to enable demultiplexing of packets into corresponding PES at the receiving side for original program reconstruction.
   e. The sequence of transport packets forms a Constant Bit Rate (CBR) MPEG-2 Transport Stream.

A PES, as described above, contains audio, video or data AUs. A PES packet is variable in size with a maximum size of 64 kb. Among many things in the PES packet header (Figure 15.5), the most important elements are: (i) decoder time stamp (DTS), which indicates when to decode a video AU, and (ii) presentation time stamp (PTS), which indicates when to present a decoded video or audio AU to the viewer. For audio, PTS refers to first AU in packet. For video DTS/PTS refers to the AU containing the first picture start code commencing in the packet.

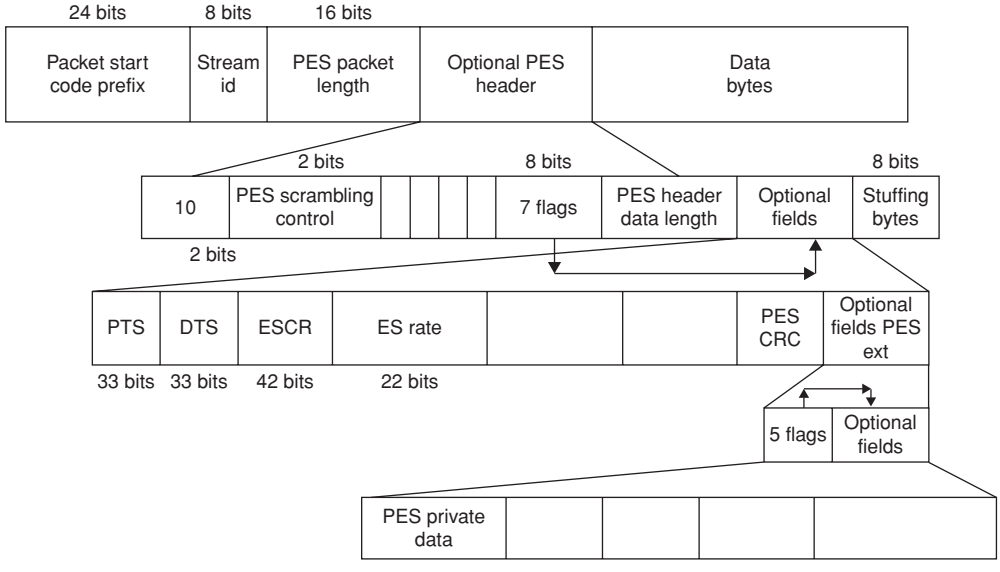


Figure 15.5 Packetized elementary stream (PES) packet.

The PES packet is eventually carried in fixed-size transport stream packets as shown in Figure 15.6.

A packet is the basic unit of data in a transport stream. Transport packets are of fixed length of 188 bytes and each packet contains only one type of data, namely audio, video, data or program guide information. These packets may also carry timing information (program clock reference or PCR) needed for synchronization at the receiver. Each 4 byte header (shown in Figure 15.7) contains (i) sync byte 0×47, (ii) transport error indicator, (iii) payload unit start indicator, (iv) transport priority, (v) 13-bit packet ID (PID), (vi) scrambling control, (vii) adaptation field control, and (viii) continuity counter, which usually increments with each subsequent packet of a frame, and can be used to detect missing packets. The adaptation field in the header may be used either for stuffing or for information. The rest of the packet consists of payload. While most transport packets are 188 bytes in length, some are 204 bytes where the last 16 bytes are Reed-Solomon forward error correction data.

Packet ID is one of the most important fields in the transport stream packet header as it helps a demultiplexer extract elementary streams from the transport stream. Related PIDs can be grouped into what are called programs. Program-specific information (PSI) contains

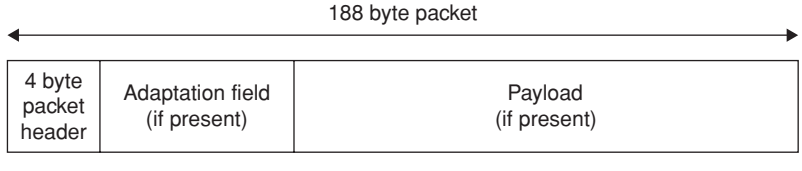


Figure 15.6 Transport stream packet.

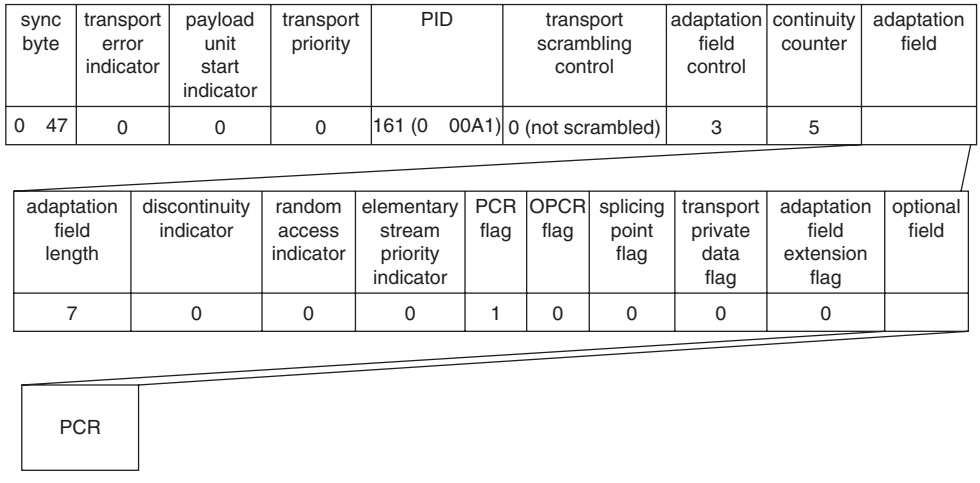


Figure 15.7 Transport stream packet header.

information specific to programs (think television channels). Program-specific information, in turn consists of the program association table (PAT) and program map table (PMT) among other things. The PAT lists all programs available in the transport stream. Each program is identified using a 16-bit program number. Each program listed in PAT consists of a list of PIDs defining the PMTs in the stream. Each PMT lists PIDs containing data relevant to the program, and also provides metadata (such as, the type of video for a video stream identified by a PID) about the streams in the corresponding PIDs. The relationship among PAT, PMT and PIDs related to individual streams within a specific program is shown in Figure 15.8. Note that the PID corresponding to the PAT is 0.

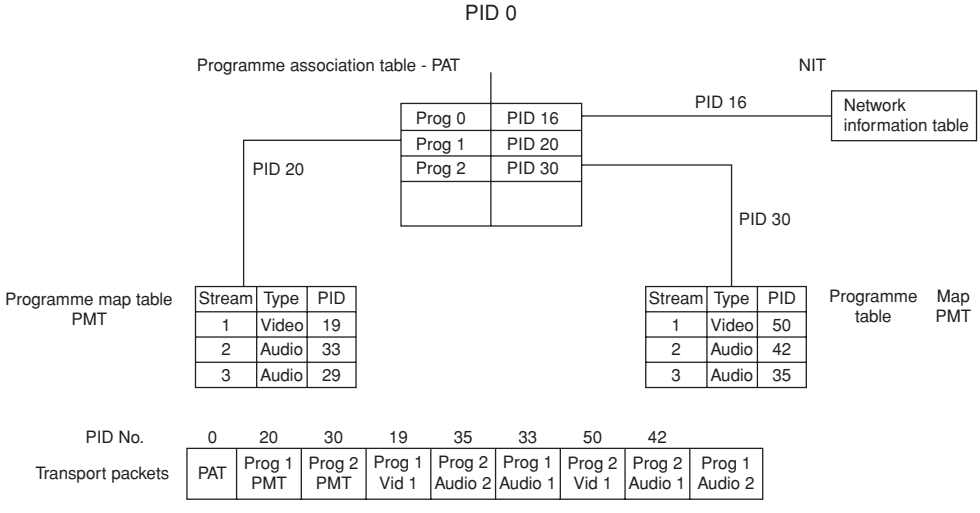


Figure 15.8 Relationship between PAT, PMT and PIDs.

15.4.2.5 File Delivery over Unidirectional Transport (FLUTE)

FLUTE [12] is a protocol for unidirectional delivery of files over the Internet. It is built on asynchronous layered coding (ALC) (RRC 3450) [31], the base protocol designed for massively scalable multicast distribution. Asynchronous layered coding defines transport of arbitrary binary objects, while FLUTE defines a mechanism for signaling and mapping the properties of files to concepts of ALC in a way that allows receivers to assign those parameters for received objects. Furthermore, ALC is a protocol instantiation of layered coding transport building block (LCT) (RFC 3451) [30].

Specifically, FLUTE:

- Defines a file delivery session on top of ALC, including transport details and timing constraints.
- Provides in-band signaling of transport parameters of the ALC session.
- Provides in-band signaling of the properties of delivered files.
- Provides details associated with the multiplexing of multiple files within a session.

A *file delivery session* in FLUTE is defined by an ALC/LCT session which in turn consists of a set of logically grouped ALC/LCT channels associated with a single sender sending packets with ALC/LCT headers for one or more objects. An ALC/LCT channel is defined by the combination of a sender and an address associated with the channel by the sender. A receiver joins a channel to start receiving the data packets sent to the channel by the sender, and a receiver leaves a channel to stop receiving data packets from the channel.

One of the fields carried in the ALC/LCT header is the transport session identifier (TSI). The (source IP address, TSI) pair uniquely identifies a session, that is the receiver uses this pair carried in each packet to uniquely identify from which session the packet was received. In case multiple objects are carried within a session, the transport object identifier (TOI) field within the ALC/LCT header identifies from which object/file the data in the packet was generated.

Another important concept in FLUTE is the file delivery table (FDT), which describes various attributes associated with files that are to be delivered within the file delivery session. There are two types of attributes:

- Attributes related to the *delivery* of file. These attributes include TOI, FEC, transmission information, size of file, aggregate rate of sending packets to all channels.
- Attributes related to the file *itself*: These attributes include name, identification and location of file (specified by the URI), MIME media type of file, size of file, encoding of file, message digest of file.

Logically, the FDT is a set of file description entries for files to be delivered in the session.

Within the file delivery session the FDT is delivered as FDT instances. An FDT instance contains one or more file description entries of the FDT. A receiver of the file delivery session keeps an FDT database for received file description entries. The receiver maintains the database, for example, upon reception of FDT Instances. Thus, at any given time the contents of the FDT database represent the receiver's current view of the FDT of the file delivery session.

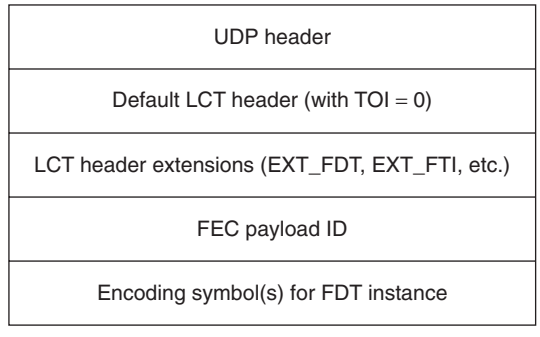


Figure 15.9 Format of ALC/LCT packet carrying an FDT Instance.

FDT Instances are carried in ALC packets and with an LCT Header extension called the FDT instance header. The FDT instance header (EXT_FDT) contains the FDT instance ID that uniquely identifies FDT instances within a file delivery session. The LCT extension headers are followed by the FEC payload ID, and finally the encoding symbols for the FDT instance. The overall format of ALC/LCT packets carrying an FDT instance is depicted in Figure 15.9. All ALC/LCT packets are sent using UDP.

15.4.2.5.1 Describing File Delivery Sessions

To start receiving a file delivery session, the receiver needs to know transport parameters associated with the session. Specifically, it needs to know: (i) the source IP address, (ii) the number of channels in the session, (iii) the destination IP address and port number for each channel in the session, (iv) the transport session identifier (TSI) of the session, (v) an indication that the session is a FLUTE session. There may also be some optional parameters, such as (i) the start time and end time of the session, (ii) FEC Encoding ID and FEC Instance ID, (iii) content encoding format, (iv) some information that tells receiver that the session contains files that are of interest.

These parameters are usually described using XML and held in a file that would be acquired by the receiver before the FLUTE session begins by means of some transport protocol (such as email, HTTP, SIP, manual pre-configuration, etc.).

Forward error correction object transmission information is usually delivered within a file delivery session using an LCT extension header EXT_FTI in ALC/LCT packets carrying an FDT instance as shown in Figure 15.9.

15.5 Modes of Transport

15.5.1 Unicast Transport for Video-on-Demand (VoD)

Video-on-demand is based on two-way communication between a video player and a video server. The video player requests a video and the VoD server serves the requested video using p-t-p communication (unicast) (Figure 15.10). More details are given in Chapter 20.

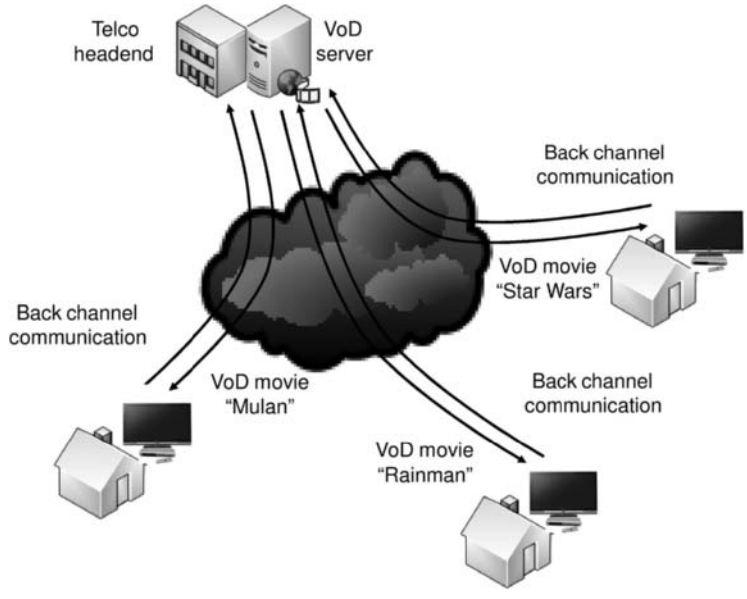


Figure 15.10 Video on demand in IPTV.

15.5.2 Multicast Transport for Live TV

There are two distinctly different approaches to delivering Live TV or broadcast content in an IPTV network (Figure 15.11). The first approach is based on the idea of the best effort delivery paradigm of the Internet while the second approach is based on the idea of delivering live TV content over traffic-engineered network paths. In the best effort paradigm, the live TV content is distributed using IP multicast [20,42] over a shared network infrastructure where it is statistically multiplexed with other types of traffic. Lost packets are recovered at the end hosts using forward error correction (FEC) [29–37]. In the alternative approach, point-to-multipoint label switched paths (LSPs) are used to set up traffic-engineered paths from the source to the destinations (edge routers in general) whereby live TV traffic is delivered with a guaranteed quality of service over the IPTV network. Specific details of these approaches are described next.

15.5.2.1 IP Multicast-Based Live TV Distribution

There are two different types of cases in IP multicast, one in which the source for an IP multicast group is known and the other in which the source for the IP multicast group is not known. Typically for live TV, the source is known, as the content is uniquely tied to a source. In such situations, a source-specific multicast (SSM) variation of protocol independent multicast (PIM) routing protocol commonly referred to as PIM-SSM [21] is used. The steps involved in using PIM-SSM for distributing live TV content are described next.

1. A set top box (STB) for a TV tuned to a given channel (mapped to IP multicast group G) sends an IGMP join message to the receiver designated router (DR) corresponding to the

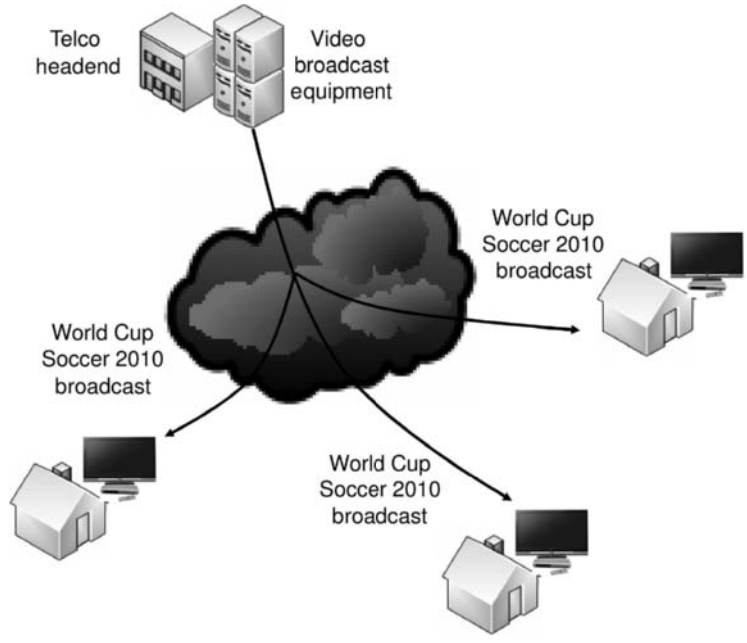


Figure 15.11 Live TV distribution in IPTV.

STB. If the STB uses IGMP v3, then it specifies the source S corresponding to multicast group G and provides that information to the Receiver DR. In case the STB uses IGMP v1/2, then the receiver DR uses a (G → S) mapping to identify the source S corresponding to multicast group G. Once the receiver DR knows the source S, it sends a PIM join message towards the source using the shortest reverse path as specified by interior gateway protocol (IGP). This is shown in Figure 15.12.

2. As the PIM join (S,G) message propagates through the routers towards the source, the routers create routing table entries corresponding to (S,G) specifying the outgoing interfaces leading to the corresponding receiver DR all the way from the source S. If the PIM join message hits a router that already has an entry created for (S,G), the (S,G) entry is updated with the corresponding outgoing interface and the propagation of the PIM Join message stops. As this process is repeated, the IP multicast tree corresponding to (S,G) is dynamically built and the tree is rooted at the source. In fact, the PIM-SSM multicast tree is nothing but the union of reverse shortest paths from the receiver DRs to the source where the shortest paths are computed based on IGP. The resultant multicast tree with DR-1A, DR-2A and DR-3A as leaf nodes is shown in Figure 15.13.
3. In order to add resilience to the distribution of live TV content, two disjoint multicast trees involving redundant routers need to be set up. Therefore the multicast receivers 1, 2 and 3, each have two routers DR-1A(B), DR-2A(B), DR-3A(B) to which they send IGMP messages and hence have redundant paths for receiving content. In addition, the redundancy extends all the way to the source as at each step of the multicast tree, there are two routers, one belonging to the primary multicast tree and the other belonging to the redundant

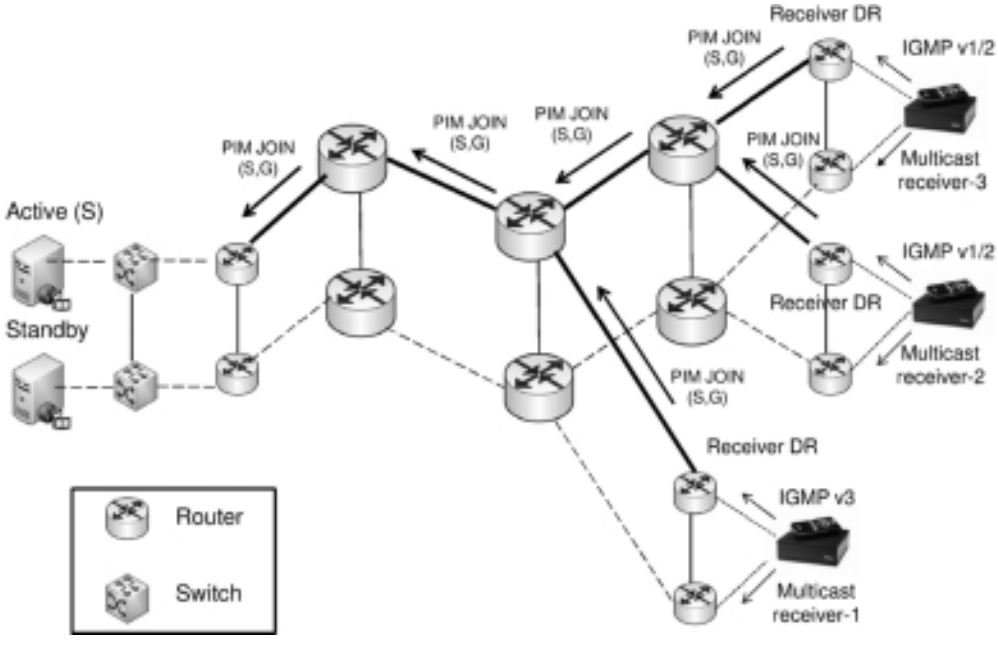


Figure 15.12 IGMP and PIM join message propagation.

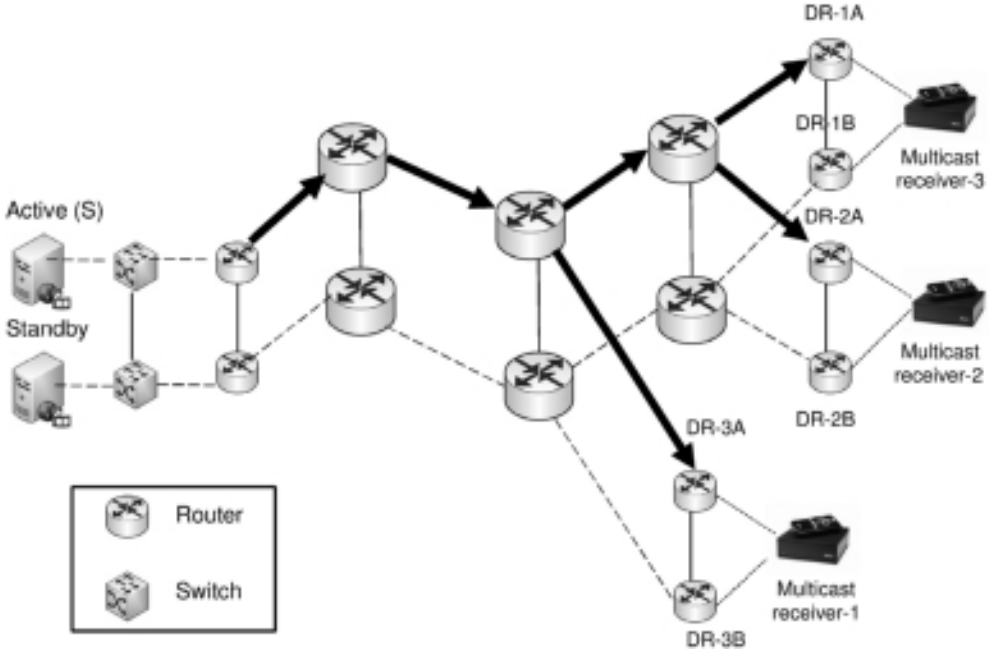


Figure 15.13 Primary multicast tree.

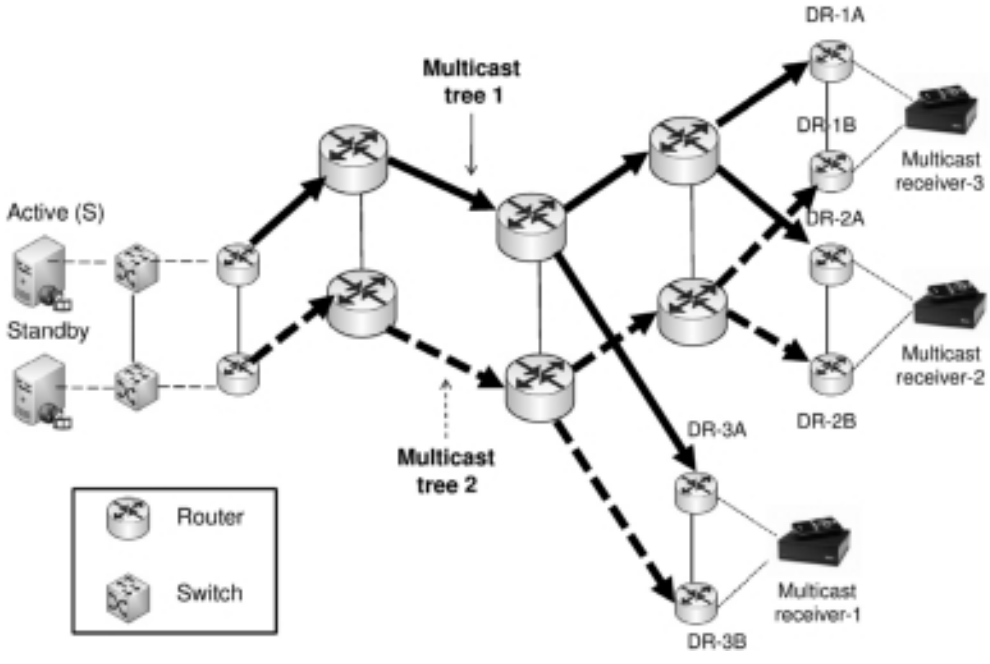


Figure 15.14 Primary and secondary multicast trees.

(or secondary) multicast tree leading to the same set of receiver DRs. The secondary multicast tree with DR-1B, DR-2B and DR-3B as leaf nodes is formed in a similar manner as shown in Figure 15.14.

4. Once the multicast tree is formed, live TV content for multicast group G is distributed along the multicast tree from the source to the Receiver DRs that have STBs interested in receiving content belonging to group G.
5. As shown in Figure 15.15, some video packets flow through the primary multicast tree and some through the secondary multicast tree reaching the final destinations.

15.5.2.2 Using Forward Error Correction (FEC) on Top of IP Multicast

Since IP multicast has best effort delivery semantics, if video packets are dropped during transport from the source to the final destinations (set top boxes and hence the TV sets) for whatever reason, they will not be recovered by IP multicast. However, the effect of packet loss on video quality of experience could be really bad, especially if the packets belong to an I-frame. Furthermore, in an MPEG group of pictures (GOP), I frame is the largest frame and hence the number of packets belonging on an I-frame is significantly more than that in other frames. Therefore, it's important that there be a mechanism to recover lost packets built on top of IP multicast transport. Forward error correction is used for exactly that purpose.

The main idea of FEC is captured in Figure 15.16. Erasure code is a type of FEC in which K source (original data) packets are transported together with N-K repair (redundant) packets

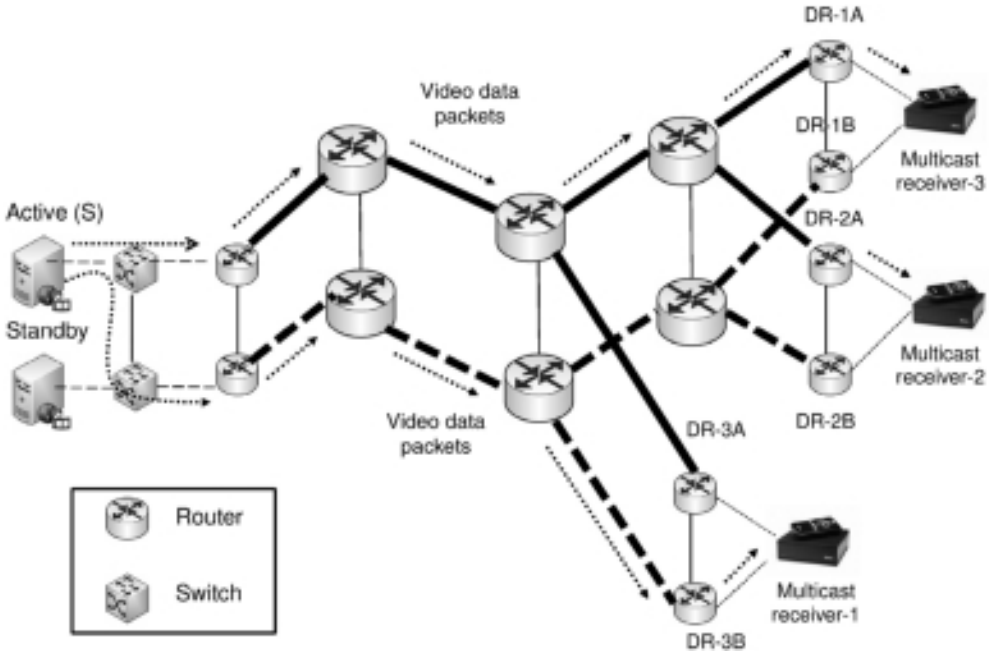


Figure 15.15 Flow of video data packets along primary and secondary multicast trees.

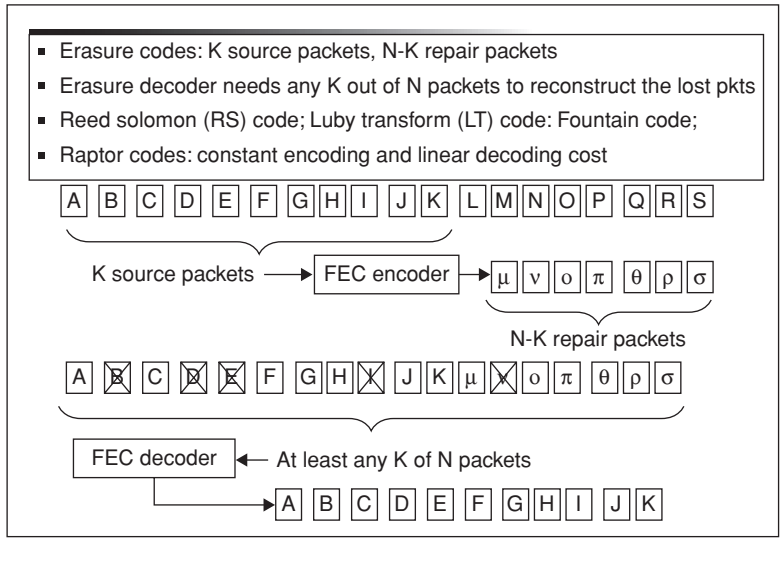


Figure 15.16 Forward error correction.

where the repair packets are computed based on the source packets in such a way that as long as the receiver (decoder) receives any K out of N packets, it can reconstruct the original K source packets. An illustration of erasure code is shown in Figure 15.16.

As shown in Figure 15.16, K source packets, A through K, are fed into an FEC encoder which uses one of many algorithms, such as, Reed Solomon (RS) code, to generate N-K repair packets μ<σψ μβολ> </σψ μβολ>through <σψ μβολ≥σ</σψ μβολ>. During transmission of these N packets, five packets, namely, B, D, E and I (four out of K source packets) and <σψ μβολ>ν</σψ μβολ> (one out of N-K repair packets), get lost. However, when the received packets (N-5) are passed through the FEC Decoder (assuming N-5 > K), the decoder can reconstruct all the lost packets, especially, the four source packets B, D, E and I.

Thus the beauty of FEC is that it does not matter what type of packet, original or redundant, is dropped; as long as the receiver receives any K packets (whatever be the mix between original and redundant), original K packets can be reconstructed. This gives tremendous robustness to the transport of video packets over best effort IP multicast transport. As shown in Figure 15.3, application layer FEC (AL-FEC) [36, 37] can be layered either on top of UDP/IP-multicast, or on top of RTP/UDP/IP-multicast.

15.5.2.3 Layered AL-FEC in IPTV Broadcast over IP Multicast

Internet protocol TV uses a variety of physical media (such as, Fiber To The Home or FTTH, Short-Loop DSL, Long-Loop DSL etc.) to distribute TV signals to people's homes. However, these physical media differ significantly in terms of propagation and other physical-layer characteristics resulting in different error characteristics. For example, FTTH links are virtually error free while Long-Loop DSL lines have relatively higher packet error rate compared to either Short-Loop DSL or FTTH. In order to multicast video packets over links with varying error rates, the architecture shown in Figure 15.17 is used. As shown in the figure, all receivers (set top boxes: STBs) whether they are connected over FTTH, short-loop DSL or long-loop

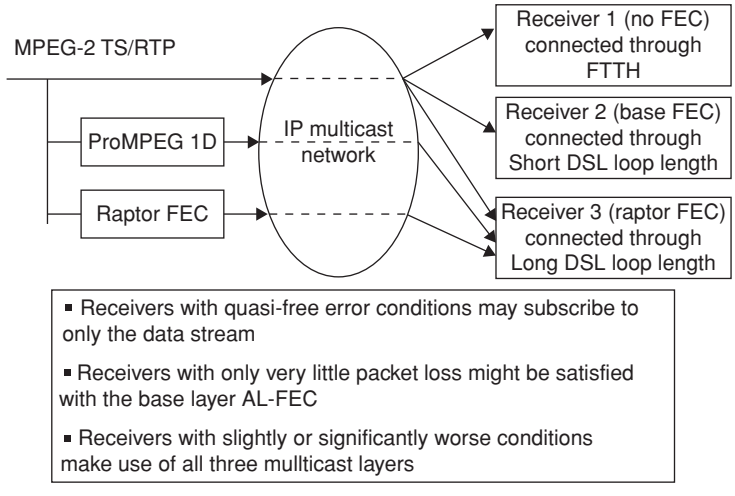


Figure 15.17 Layered AL-FEC used in IPTV broadcast.

DSL, subscribe to the IP multicast group carrying the original video stream. In addition to that, the receivers (STBs) that are connected over short-loop DSL and long-loop DSL subscribe to another multicast group carrying FEC packets (referred to as ProMPEG 1D in the figure). Furthermore, the receivers (STBs) that are connected over long-loop DSL lines subscribe to the third multicast group carrying even more FEC packets (referred to as Raptor FEC in Figure 15.17). Essentially, in the context of the FEC discussion in the previous section, this means that the number of redundant packets (K) received by receivers connected over long-loop DSL lines is more than the number of redundant packets received by receivers over short-loop DSL lines and hence can recover from higher packet-loss rates.

15.5.2.4 AL-FEC in Video File Distribution over IP Multicast

Just as AL-FEC can be used on top of IP Multicast to improve the quality of experience for Live TV viewing, it can also be used for efficiently distributing video files from the video server at the head-end to the STBs at the residences. This kind of service would be useful for distributing "hot" movies ahead of time to the STBs or for progressive download. In progressive download, a video file is usually divided into chunks, and the first chunks (say, the first 5 minutes of the video) are pushed to the STBs interested in the video and while those chunks are viewed, the next chunks (say, the next 5 minutes of the video) are pushed to the STBs. In other words, progressive download is a way of distributing video to the STBs in near real-time such that the files are pushed to the STBs in a pipelined manner. In this mode of video distribution, the receiving STB has to wait at most for the first chunk of video to arrive before playing the video. There is a trade off between the initial waiting time and the quality of experience. In fact, how much delay variation (or jitter) in the network/system can be absorbed by the STB will depend on the jitter buffer at the STB, which in turn depends on the product of bandwidth of the delivery channel and the initial waiting time. In typical Web-based video streaming systems, the jitter buffer is sized for 5–10 seconds of initial waiting time.

Regardless of whether the video files are distributed using progressive download or complete file download, reliability of delivery is important. It is important to realize that unlike in unicast (point-to-point) delivery where TCP can be used over IP for reliable delivery of content, there is no standardized mechanism for reliable multicast delivery. However, one of the most efficient ways of delivering files in a point-to-multipoint connection, as in the case described above, is to leverage FEC over best-effort IP multicast delivery. Earlier, we described the protocol called FLUTE that enables exactly that. In this subsection, the focus is not on the protocol rather on the clever usage of FEC for efficient file delivery over multicast.

The main idea is captured in Figure 15.18 in which a "carrousel"-based file delivery mechanism is compared against fountain AL-FEC-based file delivery [32,34]. In Figure 15.8 a server is trying to distribute a file consisting of five packets (A through E) by repeatedly transmitting the file to all receivers in a one-way broadcast channel, such as the satellite network. In the first repetition, packets B and E are dropped requiring a second repetition and in the second repetition, packets A and E are dropped. Since packet E has been lost in both repetitions, it requires a third repetition and in the third repetition, packet C is dropped. However, because packet C has been received in the earlier repetitions, loss of packet C in the third repetition is not important and the delivery is complete after three repetitions! It is interesting to observe that the receivers received 10 packets (A two times, B two times, C two times, D three times,

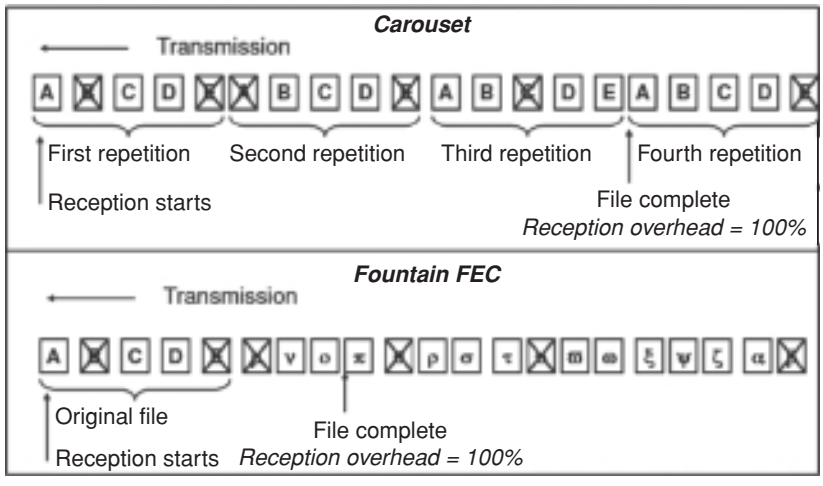


Figure 15.18 Comparing carrousel and fountain AL-FEC based file download.

E one time) when they needed only five packets. This implies 100% reception overhead. Furthermore, the delivery time is stretched to three repetitions.

In the fountain AL-FEC based system, the transmission sequence includes the five packets (A through E) of the original file followed by a "fountain" of FEC packets (μ, ν, o, π, θ, ρ, σ, τ, υ, $\bar{\omega}$, ω, ξ, ψ, ζ, α, β, . . .). What is shown in Figure 15.18 is that the *exact same* sequence of packets is lost in both cases (carrousel based file distribution and Fountain AL-FEC based file distribution). It so happens that in the fountain AL-FEC based system, the lost packets correspond to original data packets B and E and FEC packets μ, θ, υ, β. However, in case of Fountain AL-FEC, the file distribution is complete immediately after five packets (*any* five packets) are received as the lost original data packets can be re-constructed from the received five packets. As shown in Figure 15.18, file delivery is complete right after the FEC packet o is received. Note that, in sharp contrast to the carrousel scheme, the receivers receive exactly five packets, implying a reception overhead of 0%. Furthermore, the delivery time is less than two repetitions.

15.5.2.5 Point to Multi-Point (P2MP) Multi Protocol Label Switching (MPLS)-Based Live TV Distribution

15.5.2.5.1 Primer on Multi-Protocol Label Switching (MPLS)

Multi-protocol label switching (MPLS) [24, 25] is a technique that is used for efficient forwarding of data packets over a routed network, such as the Internet. The main idea in MPLS is to set up what is called a label switched path (LSP) between two edge routers referred to as label edge routers (LERs), one at the source and the other at the destination, with label switching routers (LSRs) in between (Figure 15.19) and to inject an IP packet encapsulated in

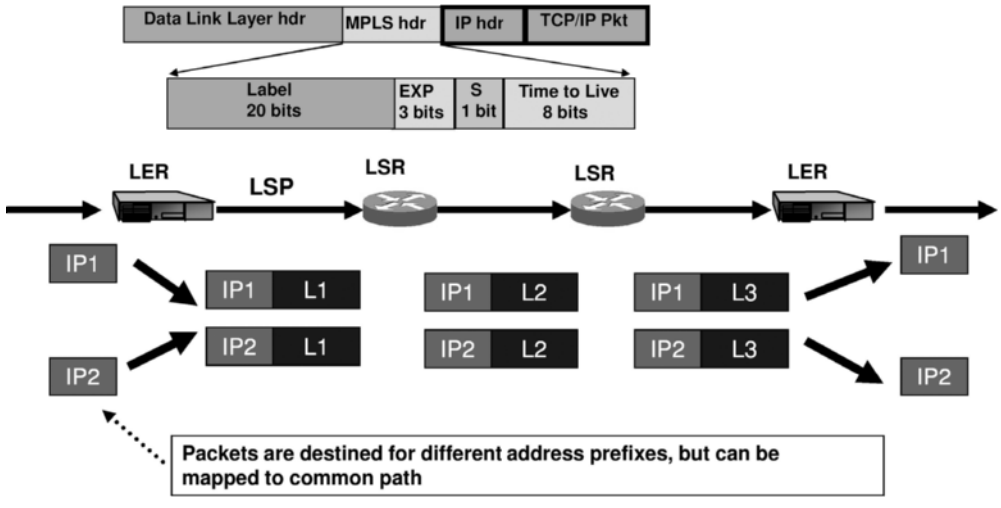


Figure 15.19 MPLS encapsulates IP packets and switches them through a label switched path (LSP).

a MPLS header with a label at the source LER and switch the label at each intermediate LSR as the packet is forwarded from LSR to LSR, finally reaching the destination LER.

In Figure 15.19, two IP packets with destination IP addresses IP1 and IP2 are encapsulated at the source LER with MPLS label L1 and they are forwarded through two LSRs as the label is switched from L1 to L2 and then from L2 to L3, finally reaching the destination LER. Once the IP packets reach the destination LER, they are routed through an IP network based on their destination IP addresses IP1 and IP2 respectively. Note that the LSP was set up between the two LERs and once the packet is in the LSP, it is simply forwarded by label switching technique. This makes the transport of packets extremely fast and efficient.

15.5.2.5.2 MPLS Technology Benefits for Service Providers

First of all, MPLS technology provides control to Service Providers to determine traffic route between a pair of source and destination in their network.

For example, in Figure 15.20, traditional IP routing would have routed a packet from R1 to R9 along the shortest path R1-R2-R3-R4-R9. However, for a variety of reasons, the service provider might want to route the traffic along a different path R1-R2-R6-R7-R4-R9 which happens to be a longer route. The MPLS technology provides the service provider with that capability. All that is needed is to set up an LSP from R1 to R9 as shown in the Figure 15.20. In fact, to set up an LSP from R1 to R9, labels have to be distributed to all the LSRs and LERs in between the source LER and the destination LER for the given LSP so that any packet injected into the LSP with a starting label can get forwarded along the LSP by switching the label at each intermediate LSR according to the labels distributed at the LSP set-up time. There are two popular protocols for label distribution, namely label distribution protocol (LDP) [43] and resource reservation protocol (RSVP) [27].

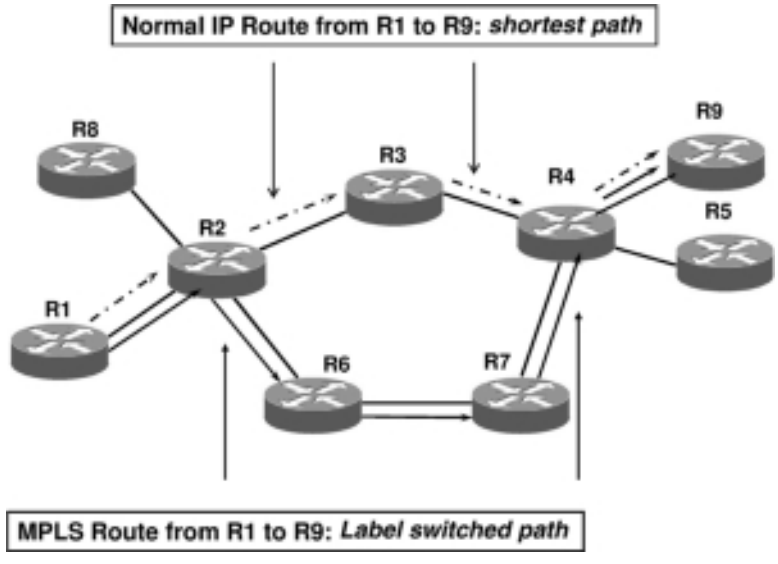


Figure 15.20 MPLS provides control over traffic route to service providers.

In fact, in RSVP protocol, the source LER sends an RSVP PATH message indicating the path to be followed by the MPLS packet when the desired LSP is set up (as shown in Figure 15.21).

As the PATH message is routed through the LSRs, a path state is set up at each LSR including the IP address of the previous router so that when the destination LER sends back RESV message with the label inside, the RESV message can follow the exact reverse path followed by the PATH message, distributing the appropriate labels. In Figure 15.22, the labels 13, 32, 22, 17 and 49 are distributed to LSRs R4, R7, R6, R2 and LER R1 respectively.

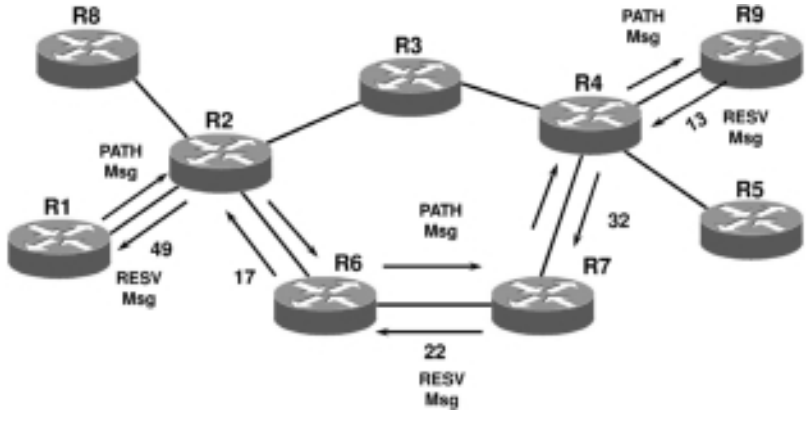


Figure 15.21 Label switched path (LSP) set up using PATH and RESV msgs.

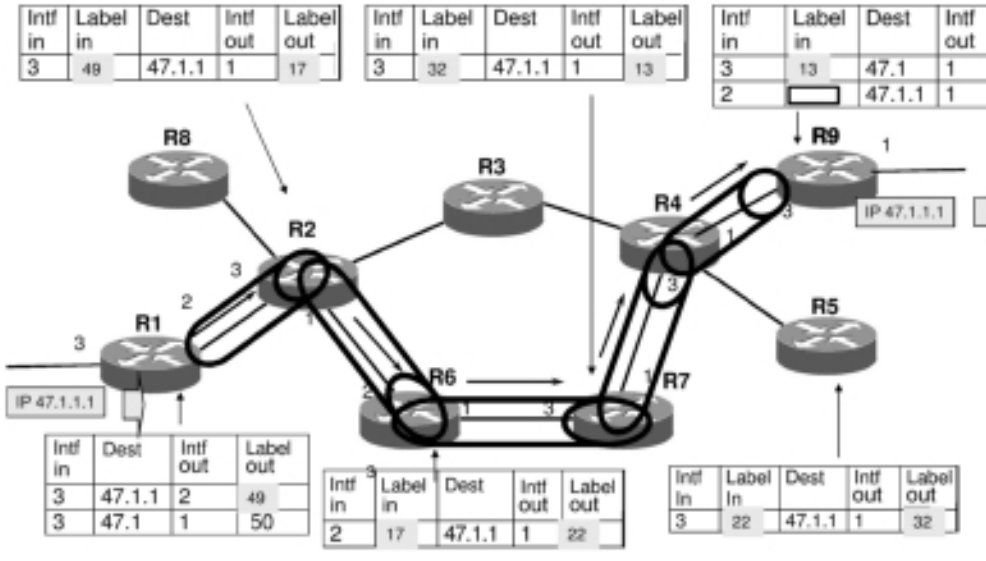


Figure 15.22 Switching of MPLS packets through a label switched path (LSP).

Going forward, any MPLS packet injected at R1 and destined for R5 will have the label 49 starting out at LER R1 and when it reaches R2, R2 consults the routing table (shown in Figure 15.22) and finds a match with the entry corresponding to incoming interface 3, incoming label 49 and forwards the packet on outgoing interface 1 with the label 17. The process is repeated at R6 where the label is switched to 22 followed by switching the label to 32 at R7, and 13 at R4 until the packet reaches the destination LER R9 where the MPLS header is stripped to take out the original IP packet and route it based on the destination address.

In fact, Figures 15.23 and 15.24 show the difference between IP Routing and MPLS Switching and the relationship between the two. For example, in Figure 15.23, the IP packet is routed based on the destination IP address D and in Figure 15.24, the same IP packet is

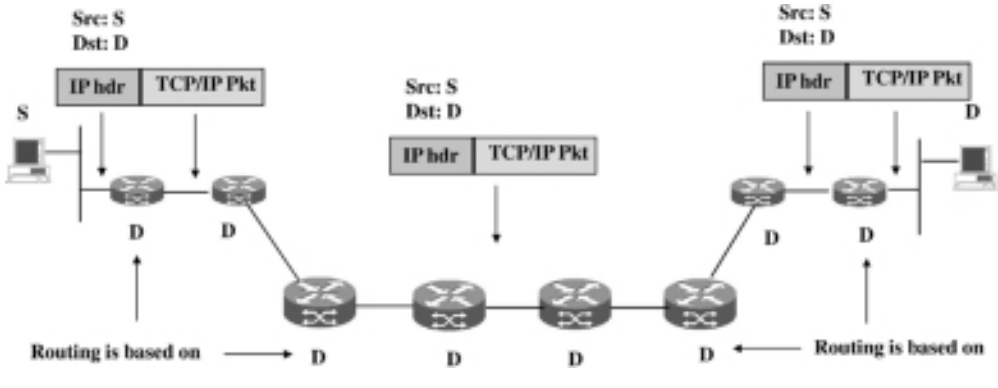


Figure 15.23 IP routing is based on destination IP address in the IP header of a packet.

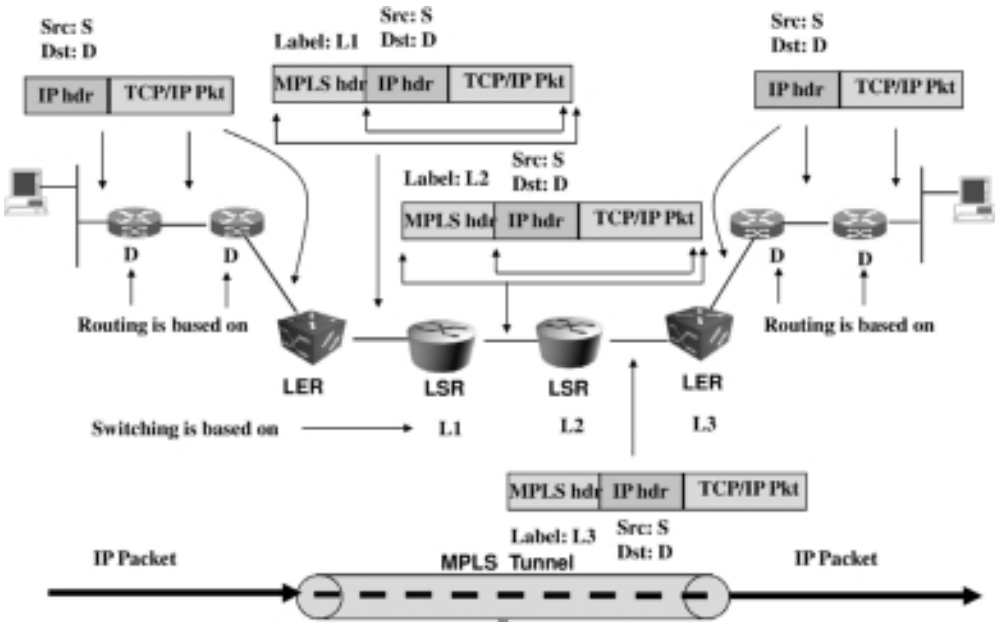


Figure 15.24 MPLS switching is based on labels in the MPLS header of a packet.

encapsulated in an MPLS header at the LER with Label L1 which is switched at subsequent LSRs based on the label. In fact, the label is switched from L1 to L2 and then from L2 to L3 until the IP packet reaches LER where the MPLS header is stripped and IP packet is routed beyond LER using the destination IP address D.

15.5.2.5.3 Traffic Engineering in MPLS Networks [25]

Multi-protocol label switching not only enables service providers to choose the path for traffic flow, but also enables them to decide how much traffic should flow through a given LSP. In addition, it enables service providers to set up back up paths for each LSP between a source and a destination, thereby providing redundancy. Specifically, the traffic engineering version of signaling protocol RSVP, popularly known as RSVP-TE [27], enables requests for reserving resources along an LSP. Using RSVP-TE, it is possible to set up LSPs with guaranteed bandwidth in an MPLS network. For example, in Figure 15.25, there are two LSPs, one from PE1 to PE4 with a bandwidth of 20 mbps and the second one from PE2 to PE5 with a bandwidth of 10 mbps.

15.5.2.5.4 Point to Multi-Point (P2MP) Label Switched Paths in MPLS Networks

P2MP MPLS is driven by three factors:

- IP multicast routing may take seconds to converge on route failure which is unacceptable for live video distribution.
- IP multicast does not lend itself easily to traffic engineering which is essential for high quality video distribution.
- IP multicast does not provide control on the multicast route path selection.

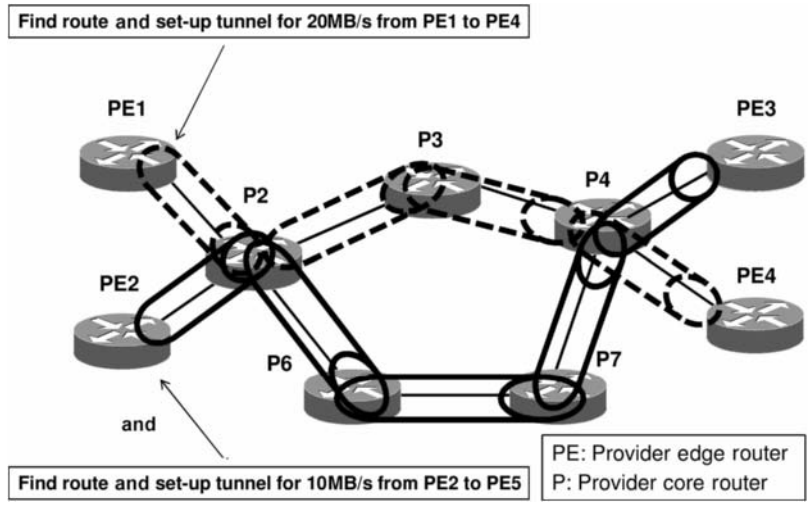


Figure 15.25 Traffic engineered MPLS label switched paths (LSPs).

P2MP MPLS, by virtue of the characteristics of MPLS networks exactly avoids the above drawbacks of IP multicast.

Essentially, P2MP traffic engineered LSPs [26, 28] provide an efficient way of traffic replication in the data path (this functionality is similar to IP multicast) as shown in Figure 15.26 where traffic is multicast from source PE1 and is distributed to destinations PE2, PE3, PE4 and PE5 in such a way that traffic replication happens at P1and P3. In addition to that, RSVP-TE

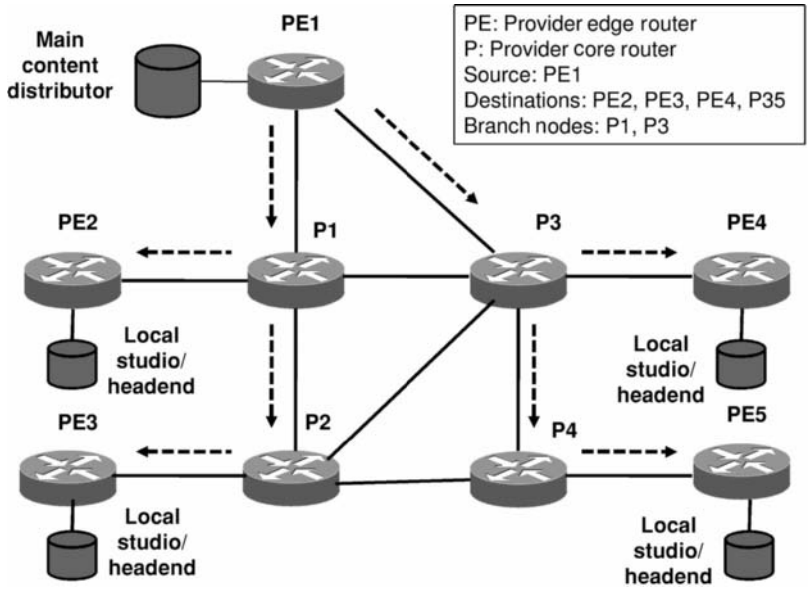


Figure 15.26 Point to multipoint (P2MP) MPLS label switched path (P2MP LSP).

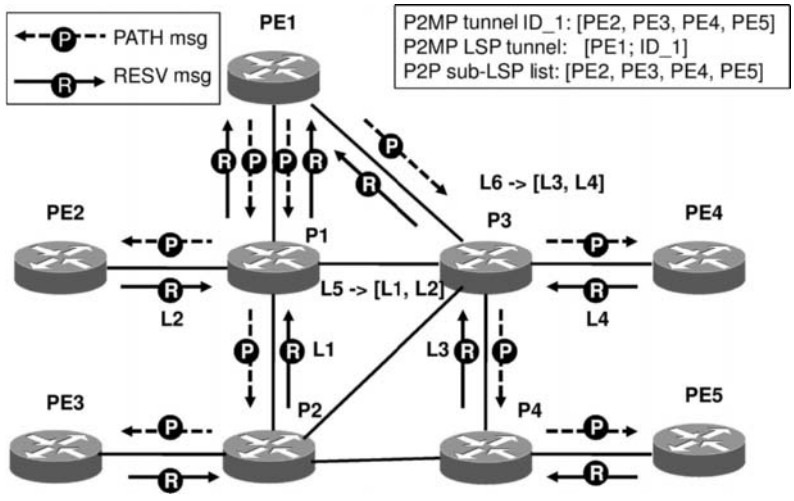


Figure 15.27 Point to multipoint (P2MP) MPLS LSP tunnel set up using multiple PATH messages.

signaling enables reserving bandwidth on the P2MP LSP such that the source may pump traffic through the network at the prenegotiated rate. This is very important for video traffic where QoE expectations are set at high standards by existing technology.

Typically a P2MP tunnel consists of multiple P2MP LSP tunnels and is identified using a 32-bit P2MP ID and a 16-bit tunnel ID. A P2MP LSP tunnel consists of multiple Point to Point (P2P) sub-LSPs and is identified by the M2MP Tunnel Session ID and P2MP Sender_Template which identifies the source via IPv4/IPv6 source IP address and LSP ID. A P2P sub-LSP is identified using the Egress LSR destination address and P2P sub-LSP ID. Sub-LSPs are set up using explicit routing from ingress to egress routers.

Usually multiple PATH messages are used to signal a single P2MP tunnel LSP as shown in Figure 15.27. Each PATH message can signal single P2P sub-LSP, multiple P2P sub-LSPs, or all P2P sub-LSPs.

In Figure 15.27, RESV messages from P2 and PE2 with labels L1 and L2 respectively are combined to have a common label L5 and, similarly, the labels L3 and L4 from P4 and PE4 respectively are combined to have a common label L6. Labels L5 and L6 are carried to PE1 in RESV messages from P1 and P3 respectively. Note that, in addition to distributing labels, RESV messages are also used to reserve bandwidth in the P2MP LSP tunnel. Thus, a P2MP LSP tunnel provides a traffic-engineered point-to-multipoint distribution channel that can be used effectively for video traffic distribution in a packet-switched network.

The beauty of P2MP LSP is that once such an LSP is set up, a variety of traffic, such as layer-2 multicast traffic, layer-3 (IP) multicast traffic, multicast virtual private network (VPN) traffic as well as virtual path label switching (VPLS) multicast traffic can be carried over it in a seamless manner.

Figure 15.28 shows that, in addition to traffic engineering for guaranteed bandwidth, P2MP LSPs can be set up with redundant paths for a quick switch over (less than 50 ms) in case of route failure.

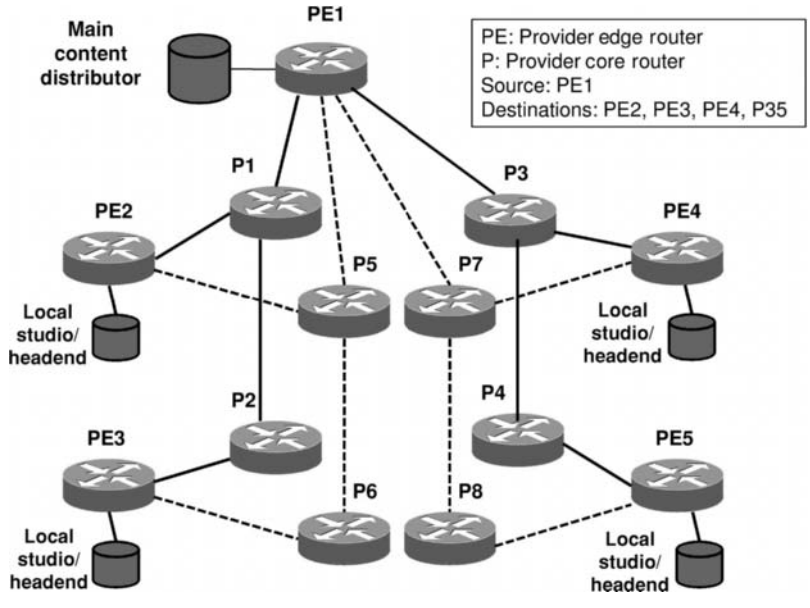


Figure 15.28 Point to multipoint (P2MP) MPLS LSP with redundant paths.

15.5.2.6 Point to Multi-Point (P2MP) MPLS vs. IP Multicast Routing

As described above, P2MP MPLS LSPs in the core network of a service provider provides several benefits over IP multicast routing as the core transport mechanism. The main benefits are summarized in Table 15.3.

Table 15.3 Comparison of IP multicast and point-to-multipoint MPLS.

	Point to multi-point MPLS LSPs	IP multicast (E.g, PIM-SSM, PIM-SM)
Protection	Strict Service Level Agreement Link Protection Graceful Restart (GR) Fast Reroute (FRR) 'Make before Break' capabilities	No such capabilities exist as IP multicast routing protocols are not traffic engineering protocols
Resource Reservation	Strict SLA Resource reservation for each multicast yree	No resource reservation mechanism
Explicit Path Routing	Supports explicit routing along paths different from hop-by-hop IP routing	No equivalent support exists
Multicast Tree	P2MP LSP is signaled by the root and hence flexible multicast tree computation algorithms can be used	IP multicast tree selection is receiver-driven and hence does not have the flexibility of computing say a minimum cost (Steiner) tree

15.6 Summary

This chapter focused on IPTV services. Internet protocol TV offers three types of services: (i) basic channel service which is equivalent to broadcast services, (ii) selective channel service which is equivalent to on-demand video services and (iii) interactive data services, which are equivalent to Internet services in principle.

Broadcast services are not limited to video only. Audio broadcast (radio-like services) and data broadcast (local information) are supported by basic channel service. Video-on-demand also comes in multiple flavors: (i) real video-on-demand, (ii) near video on demand, (iii) personal video recording (PVR) or (iv) electronic program guide (EPG)-based recording service. Interactive services are expected to enable TV viewers to interact with the services on television for gathering information, doing transactions, for communication, for entertainment and/or for learning.

Internet protocol TV standards specify requirements with respect to display quality, such as bandwidth, audio/video compression formats and resolution. Transport mechanisms defined in the standards encompass data encapsulation, transmission protocols, quality of service, service delivery platform, content distribution, and FEC. Metadata formats, transport and signaling are keys to providing advanced services, such as search capabilities in television.

There are several ways of transporting video over IPTV networks. Physical transport can happen over variety of media, such as, fiber, wireless, DSL, satellite, cable and so on. However, IP is used at the network layer across all types of physical media. For point-to-point communication, such as for video-on-demand, IP is used in unicast mode while for point-to-multipoint communication, such as for live TV, IP is used in multicast mode. Furthermore, video could be transported either using TCP or UDP. Typically, the progressive download or download-and-play mode of IPTV uses HTTP/TCP as the transport layer protocol. An alternative reliable delivery mechanism for video over multicast networks is to use file delivery over unidirectional transport or FLUTE protocol, which leverages ALC for scalable reliable multicast. Most commonly, video is transported in IPTV in "streaming" mode. Audio/video data is encapsulated in MPEG-2 TS, which could either directly go over UDP/IP or leverage RTP/UDP/IP. The streaming mode of transport can happen either in unicast mode or in multicast mode. Internet protocol multicast is used for point-to-multipoint streaming, and that requires the receiving end-points of IPTV transmission to join relevant IP multicast groups using IGMP. The receivers of IPTV stream provide feedback about the quality of reception to the server using RTCP. Based on the feedback, the servers can adjust the rate of streaming either by changing encoding rate or frame rate and thereby improve the quality of experience for end users. In the case of the video-on-demand service, there is a need to support trick-play (fast forward, rewind, pause etc.) and the signaling to control the stream is done using RTSP.

There are two distinctly different approaches to delivering Live TV or broadcast content in an IPTV network. The first approach is based on the idea of best-effort delivery paradigm of the Internet while the second approach is based on the idea of delivering live TV content over traffic-engineered network paths. In the best effort paradigm, the live TV content is distributed using IP multicast over a shared network infrastructure where it is statistically multiplexed with other types of traffic. Lost packets are recovered at the end-hosts using FEC In the alternative approach, point-to-multipoint LSPs are used to set up traffic-engineered paths from the source to the destinations whereby live TV traffic is delivered with a guaranteed quality of service over the IPTV network.

References

[1] IPTV: http://en.wikipedia.org/wiki/IPTV (accessed June 9, 2010).

[2] Guide to Open IPTV Standards: http://www.lightreading.com/document.asp?doc_id=157934 (accessed June 9, 2010).

[3] IPTV: Internet Protocol Television Architecture and Technologies.

[4] TR-058: DSL Forum Technical Report – Multi-Service Architecture and Framework Requirements.

[5] TR-069: DSL Forum Technical Report C CPE WAN Management Protocol.

[6] TR-094: DSL Forum Technical Report C Multi-Service Delivery Framework for Home Networks.

[7] MPEG 2 description: http://mpeg.chiariglione.org/standards/mpeg-2/mpeg-2.htm (accessed June 9, 2010).

[8] International Organization for Standardization (ed.) (1999). International Standard ISO/IEC 138181. Information Technology – Generic coding of moving pictures and associated audio information. International Organization for Standardization.

[9] International Organization for Standardization (Ed.) (2005). International Standard ISO/IEC 14496-10: Infrastructure of audiovisual services – Coding of moving video: Advanced video coding for generic audiovisual services. International Organization for Standardization.

[10] European Telecommunications Standards Institute (Ed.) (2001). ETSI TR 101 290 Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems. Sophia Antipolis Cedex, France: European Telecommunications Standards Institute.

[11] ITU-T Recommendations: http://www.itu.int/rec/T-REC/e

[12] FLUTE - File Delivery over Unidirectional Transport. RFC 3926

[13] Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD)-Based Multicast Forwarding("IGMP/MLD Proxying", RFC 4605.

[14] Cain, B., Deering, S., Kouvelas, I. *et al.* (2002) Internet Group Management Protocol, Version 3, RFC 3376, October.

[15] Fenner, W. (1997) Internet Group Management Protocol, Version 2, RFC 2236, November.

[16] Deering, S. (1989) Host extensions for IP multicasting, STD 5, RFC 1112, August.

[17] Deering, S., Fenner, W. and Haberman, B. (1999) Multicast Listener Discovery (MLD) for IPv6, RFC 2710, October.

[18] Vida, R. and Costa, L. (2004) Multicast listener discovery Version 2 (MLDv2) for IPv6, RFC 3810, June.

[19] Holbrook, H., Cain, B. and Haberman, B. (2006) Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast, RFC 4604, August 2006.

[20] Deering, S. (1991) Multicast Routing in a Datagram Internetwork. Ph.D. Thesis, Stanford University.

[21] Fenner, B., Handley, M., Holbrook, H. and Kouvelas, I. (2006) Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601, August, 2006.

[22] Real-Time Transport Protocol (RTP), RFC 3267.

[23] Real Time Streaming Protocol (RTSP), RFC 2326.

[24] Multiprotocol Label Switching Architecture, RFC 3031.

[25] MPLS Traffic Engineering: http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s5/mpls_te.htm (accessed June 9, 2010).

[26] Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs), RFC 4461 (accessed June 9, 2010).

[27] Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs), RFC 4975.

[28] Availability and Performance Challenges in P2MP MPLS. http://www.metaswitch.com/download/MPLS-Ethernet-WC-Feb2009-P2MP.pdf (accessed June 9, 2010).

[29] Watson, M., Luby, M. and Vicisano, L. (2007) Forward Error Correction (FEC) Building Block, RFC 5052, August.

[30] Luby, M., Watson, M. and Vicisano, L. (2009) Layered Coding Transport (LCT) Building Block, RFC 5651, October.

[31] Luby, M., Watson, M. and Vicisano, L. Asynchronous Layered Coding (ALC) Protocol Instantiation. draft-ietf-rmt-pi-alc-revised-10.

[32] Shokrollahi, A. Fountain Code. http://www.dss.uwaterloo.ca/presentations_files/DSS_Shokrollahi.pdf (accessed June 9, 2010).

[33] Tracey Ho. Summary of Raptor Codes. http://web.mit.edu/6.454/www/www_fall_2003/trace/raptor2.pdf (accessed June 9, 2010).
[34] Digital Fountain Technology: http://www.digitalfountain.com/technology.html (accessed June 9, 2010).
[35] Amin Shokrollahi. Raptor Codes. http://www.cs.huji.ac.il/labs/danss/p2p/resources/raptor.pdf.
[36] DVB-IPTV Application-Layer Hybrid FEC Protection. draft-ietf-fecframe-dvb-al-fec-03.
[37] Gómez-Barquero, D. and Bria, A. Forward Error Correction for File Delivery in DVB-H. http://www.cos.ict.kth.se/publications/publications/2007/2590.pdf (accessed June 9, 2010).
[38] http://en.wikipedia.org/wiki/MPEG_transport_stream (accessed June 9, 2010).
[39] MPEG 2: http://en.wikipedia.org/wiki/MPEG-2 (accessed June 9, 2010).
[40] DVB Transport Stream: http://users.abo.fi/jbjorkqv/digitv/lect4.pdf (accessed June 9, 2010).
[41] Ott, J., Chesterfield, J. and Schooler, E. RFC5760: RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback.
[42] Paul, S. (1998) *Multicasting over the Internet and its Applications*, Kluwer Academic Publishers. ISBN: 0792382005.
[43] Andersson, L., Doolan, P., Feldman, N. *et al.* Label Distribution Protocol. RFC3036.

16

Video Distribution in Converged Networks

Communication service providers (CSPs) who own multiple delivery networks, such as, broadband, mobile (Cellular, WiMax etc.) and television networks (IPTV, Digital Cable, Digital Satellite etc.) can reach out to their customers in multiple ways. However, businesses within CSPs have been running in silos, meaning that each network, whether broadband, mobile or television, has been developing its own applications, acquiring its own content and running its own profit and loss without considering the possible gains that might incur from synergizing their businesses.

16.1 Impact of Treating Each Network as an Independent Entity

Table 16.1 captures the impact of running each network as independent business even though they are part of a bigger umbrella.

16.2 Challenges in Synergizing the Networks and Avoiding Duplication

In order to overcome the drawbacks mentioned in Table 16.1, it is important that the application is written once and used across multiple networks. That will eliminate the duplication of effort in writing the same application multiple times, once for each network; it will reduce time to market new applications and services and improve the return on investment (ROI) on licensed content. Specifically, if a CSP can make its licensed content available to its customers through a variety of networks regardless of the device used by the customer, that would increase the monetization possibilities of the content [1].

However, there are many challenges in being able to write an application once and use it in multiple networks. Figures 16.1, 16.2 and 16.3 capture some of the challenges.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

Table 16.1 Impact of confining applications and content to isolated channels (networks).

Application development in isolation for each channel	Impact on business
Result in separate applications, each specific to a channel/device, for accessing same content/service	• Developing applications for different channels/devices leads to significant duplication of effort • Long time to develop and market new applications/services
Content and services are not shared across channels/devices	• ROI on content not fully realized
User experience across channels/devices is inconsistent	• Lower brand recognition • Sub-optimal quality of experience

When the same content (video) is provided to end-users in different channels (networks) on different devices, the way they are expected to interact with the content is different due to the limitations of user interface. Naturally, writing an application once and transforming it in a way that it can be used seamlessly across devices and networks requires "transformation of user interface".

In addition to heterogeneity of user interface, there is tremendous challenge in "transforming content" to suit the heterogeneity in screen size, resolution, and digital aspect ratio as shown in Figure 16.2.

	PC	TV	Mobile
Input device	Keyboard	Remote	Keyboard / numeric keypad
Type of keyboard	Qwerty	Triple tap	Qwerty / triple tap
Navigation	Mouse and keyboard	Arrow buttons in remote control	Arrow buttons
Selection	Mouse click or enter button in keyboard	Select button in remote control	Select button

Figure 16.1 Heterogeneity of user interface and navigational techniques.

	PC	TV		Mobile
Screen size	15 inch to 17 inch	32 inch to 46 inch		1 inch to 3.5 inch
Viewing distance	2 ft. to 3 ft.	10 ft. to 15 ft.		1 ft. to 2 ft.
Screen display resolution	800 600 1024 768 1280 800	SDTV 480i	640 480 704 480 852 480	128 128 128 160 160 160 176 220 160 240 240 240 240 320
		720p (HDTV)	1280 720	
		1080p, 1010i (HDTV)	1920 1080	
DAR (digital aspect ratio)	4:3 5:4 16:9	SDTV 480i	4:3 or 16:9	4:3 15:9
		720p (HDTV)	16:9	
		1080p, 1010i (HDTV)	16:9	

Figure 16.2 Heterogeneity of screen size, resolution and digital aspect ratio.

Different devices have different screen size, different resolution and different aspect ratios, so unless content (video) is transformed properly, user experience is going to suffer.

In addition, the codecs used in PC, TV and mobile are different, and so are the containers, image formats, media players, media delivery mechanisms and the network itself as shown in Figure 16.3. Therefore, transforming media to suit the entire gamut of heterogeneous networks and devices is indeed a big challenge.

	PC	TV	Mobile
Codecs	WMA, WMV, RealAudio, RealVideo, H.264, Sorenson (flash)	MJPEG, MPEG-1, MPEG-2, MPEG-4	H.263, H.264, MPEG-4
Containers	AVI, ASF, RealMedia, Quicktime (MOV), FLV, MP4	MPEG-TS	3GP, MP4
Image formats	All known image formats are supported by all browsers	IPTV browser specific support	Mobile browser specific support
Media players	Windows Media Player, RealPlayer, QuickTime	Windows media player, THEO IPTV media player elecard IPTV player	Windows Media Player, RealPlayer, Adobe FlashLite Player
Media delivery	Progressive download, streaming	Download and play, streaming	Streaming
Network	Ethernet, TCP/IP	DSL, fiber	GPRS, EvDO, HSPA

Figure 16.3 Heterogeneity of media codecs, formats, players and delivery.

16.3 Potential Approach to Address Multi-Channel Heterogeneity

Some of the key ideas to address the challenges mentioned before now follow.

16.3.1 Rule-Based Transformation of Media

Different mobile and TV terminals have different form factors, different resolutions, and they support different codecs and image formats. The idea of media transformation is to use a media transformation engine that transforms media from one format/codec to another based on the capabilities of the display machine.

16.3.2 Static versus Dynamic Transformation

There are potentially millions of videos (movies, sports, music, UGC etc.) and a significant number of them are available on the Web. However, transforming videos "on the fly" from a Web-supported format to a mobile and/or TV-supported format would not be possible with the increasing number of requests. Another alternative is to transcode each video into multiple formats and store these instances in a shared storage and pick the right format based on the capabilities of the requesting device. This approach is not scalable either because the amount of storage needed would be enormous! One possible approach would be to use a combination of the static and dynamic approach where the top 20% of accessed videos would be transcoded ahead of time into top 20% formats and cached locally at the media transformation engine. Thus the most frequently accessed videos will be instantly served from the cache. However, if the request falls outside of the popular window, then the transcoding will be done "on the fly" by the media transformation engine and served to the requesting device.

16.3.3 Dynamic Selection of Top 20% Videos and Top 20% Formats

In order to satisfy the above requirements of classifying content as "popular", one approach is to choose the top 20% of videos and top 20% of formats based on some static market analysis. The other approach is to dynamically track the popularity of videos and the popularity of the formats (devices) and constantly update the set of videos to be cached at the media transformation engine. The approach based on dynamic tracking of videos is more scalable as the number of videos in the system increases.

16.3.4 Template for Applications

There are thousands of applications on the web and each might have its own user interface, so the first step to creating a contextually relevant user interface is "standardization". That is, it would make sense to classify applications into "categories" and for each category of application; to generate a "standard" template (layout). For example, there are different templates for the following application categories: (i) digital content retailing, (ii) social networking, (iii) virtual reality and so on.

Note that this list is extensible and as new categories of applications need to be added, a new template can be generated. Templates are used to define the layout of the window where the static text and result sets will be displayed.

16.4 Commercial Transcoders

There are several commercial transcoders available in the market [2, 11–15]. In this section, we provide an overview of the existing solutions in the market and go into the detailed description of one of the commercial transcoders to provide a complete understanding of how these systems work in practice. While we focus on one [2] of the many products in the market, it should be understood that the goal is not to exclude other products which are similar in functionality, rather to illustrate the general principles behind these products.

Typically, commercial transcoders have the capability to transcode both live and stored content and to make the content available for multiple channels (TV, mobile and Web) to consume. Figure 16.4 shows a live broadcast television feed captured using a satellite dish and processed through the bank of encoders to prepare the content for consumption via different channels (Cable TV, IPTV, mobile TV, Web TV). In the figure, all of the above channels have been represented as "broadcast TV" to indicate that the content is live and is being broadcast. In addition, video stored in tapes and DVDs can be ingested and processed through the bank of encoders for multi-format encodings. Further, ingested content kept in network storage can also be transcoded to generate multirate, multiformat video that can played through Web servers to PCs, mobiles and TV sets with broadband data connection.

Here are some of the leading transcoding products in the market:

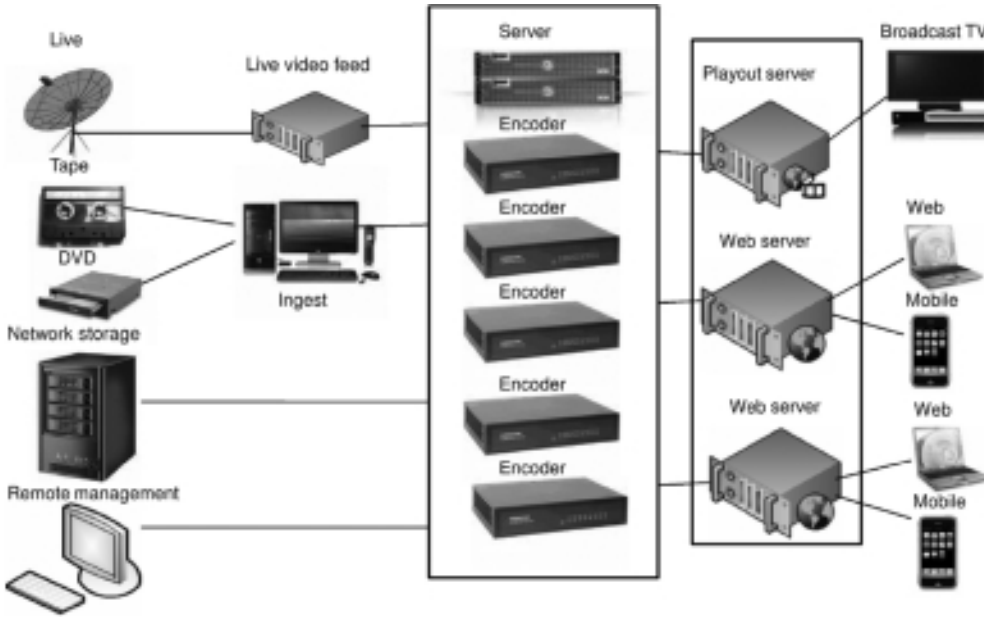


Figure 16.4 Transcoding of live and stored video and delivery into multiple channels.

- Rhozet Carbon Coder.
- AnyStream Agility.
- Telestream FlipFactory.
- Agnostic Media Gridcoder.

To provide a practical feel for what kind of capabilities these products have, how they can be used, and integrated in a solution, one of the products, namely, Rhozet Carbon Coder is described next.

16.4.1 Rhozet Carbon Coder – Usage Mechanism

Carbon Coder can be used in four different ways:

1. **Manual Processing**
   Manual processing mode requires the files that need to be converted, and the desired target formats in which the conversion should happen. On selecting the convert option files are converted to the request format.
2. **Automatic Processing**
   Automatic processing requires the watch folder and target folder to be preconfigured in addition to preconfiguring the conversion details. Whenever content is added to the watch folder it is transcoded and the output file is placed in the target folder.
3. **XML Based**
   An XML-based technique is useful for integrating the transcoder with the end-to-end flow of an application. Essentially, this feature enables application to be programmatically controlled. All functions can be fully controlled via SDK
4. **Network Based**
   Network-based usage enables multiple transcoding services to be connected to a carbon server through network (see Figure 16.4). Server manages load balancing, prioritization, FTP transfers, status monitoring and job notification. In this system, the content that needs to be transcoded can be divided into multiple pieces, transcoded independently by different transcoding services and, once all the parts are converted, they can be spliced together. This feature is extremely useful for reducing the transcoding time.

16.4.2 Important Features

1. **Predefined Configuration Types**
   Rhozet provides various configuration types that help in tailoring content conversion to suit variety of needs. For example, in applications where quality can be compromised for speed, one can select the video target as “Web” in Rhozet GUI. When “Web” is selected as the video target, Rhozet dynamically provides the options to select “server type” (streaming server or a Web server), and “average connection bandwidth” and prepares the content appropriate for the selected environment.
   Table 16.2 lists the format types and dynamic options supported by Rhozet.

Table 16.2 Video formats supported by Rhozet.

Video target	Rhozet provided video format type	Rhozet provided dynamic options (content will be prepared accordingly)
Web Video	Quick Time, Real Media, Windows Media	Software asking the type of servers (web server, streaming server) and the targeted connection speed.
HD	MPEG-2, Windows Media	Asking frame
CDROM Video	AVI, MPEG-1, Quick Time, Window Media	Asking the frame size in which we want the content (320*240, 640*480), High/Medium/Low quality
VideoCD	NTSC, PAL	Providing options like Optimized for speed / Optimized for quality
Super VideoCD	NTSC, PAL	Constant bit rate or two-pass variable bit rate Optimized for speed / Optimized for quality
DVD	NTSC, PAL	Constant bit rate or two-pass variable bit rate Optimized for speed / Optimized for 1uality
Video Editing	NTSC, PAL	Asking various video editing formats

2. **Auto Conversion of Single Content to Multiple Preconfigured file Formats**
   Administrative interface provides a range of functionality and significant flexibility to transcode content. Multiple target files corresponding to a single source file can be created with different configurations. Whenever new content is added to the so-called "watch" folder, it is converted to multiple target file formats simultaneously. For example, if one is planning to support mobile devices with 3gp, iPhone and web with flv, windows media and quick time, it could be done with the setup shown in Figure 16.5.
3. **Preset Functionality**
   Preset helps users set up the configuration for conversion by providing predefined settings as shown in Figure 16.6. This helps reduce the preparation time for conversions and provides a good idea about what types of content are supported by different categories of devices. Figure 16.6 shows the list of presets category and their properties, which are pre-configured in Rhozet admin.
4. **List of Features**
   Table 16.3, although shown in the context of a specific product, namely, Rhozet carbon coder, is a typical feature list of most transcoding solutions in the market.
5. **Other Salient Features of Carbon Coder**
   There are several powerful features of Carbon Coder that can be leveraged for building applications around it. Here are some examples:
   a. During transcoding of content, Rhozet Carbon coder enables:
      i. extraction of the audio component from video;
      ii. extraction of thumbnails;
      iii. extraction of the images to create story boards.

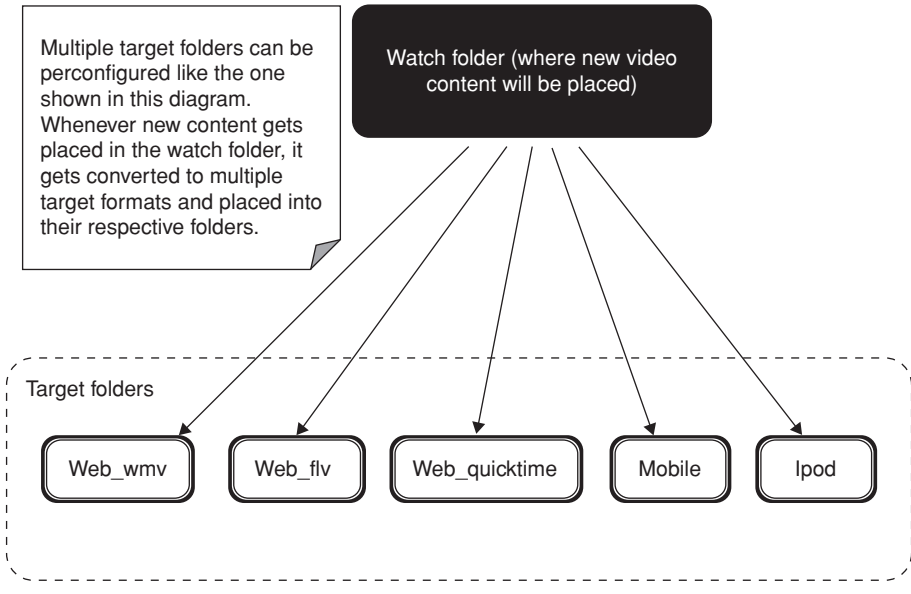


Figure 16.5 Watch folder and target folders in Rhozet.

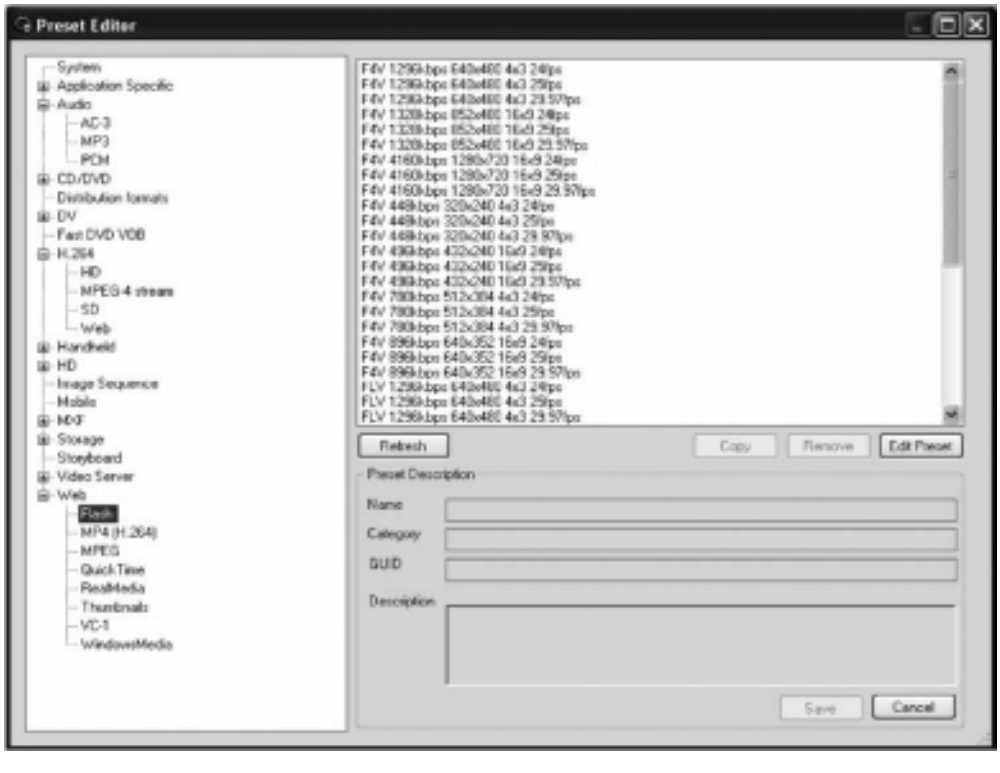


Figure 16.6 Rhozet admin interface showing preset configurations.

Table 16.3 Feature list of Rhozet.

Features

Supported Video Codecs
- MPEG-1, MPEG-2, MPEG-4
- H.263, H.264, JPEG-2000, VC-1
- Windows Media, RealVideo, Flash
- DV25, DV50, DVCPro, HDV
- DPX, DivX, Image Sequences

Supported Containers
- AVI, QuickTime
- ASF, WMA, WMV
- MXF (including D-10/IMX)
- MPEG-2 PS, MPEG-2 TS, VOB
- Windows Media Audio, RealAudio

Supported Systems
- Leitch VR, Nexio
- Grass Valley Profile, K2
- Omneon Spectrum, Quantel sQ
- Avid Editing Systems, Apple Final Cut Pro, Adobe Premiere Pro, Canopus Edius

Basic Video Operations
- Frame size conversion
- Frame rate conversion
- Color space conversion
- Aspect ratio conversion
- Interlace/Deinterlace conversion
- Inverse telecine
- PAL/NTSC conversion
- SD/HD conversion
- Cropping

Video Processing
- Fade in/out
- Black/white correction
- Blur
- Color correction
- Gamma correction
- NTSC-safe
- Median
- Rotate
- Sharpen
- Temporal noise reduction

Audio Processing
- Normalize
- Fade in/out
- Low-pass
- Volume
- Dynamic range compressor

Additional Operations
- Timecode imprint
- Subtitle/CC imprint
- XML controllable titler
- Metadata transport and conversion
- Line 21/CC preservation/conversion
- Watermarking
- Logo insertion
- Native precessing in 4:2:2 YCbCr or RGB
- 601/709 color space support
- Video capture support for supported capture devices which include the Viewcast Osprey 230, 240, 540 and 560.
- Multiple simultaneous target outputs
- Unlimited number of encoding passes supported
- Remote job submission
- Batch processing
- Watch folder automation
- Segment extraction/insertion
- Poster frame extraction

b. Leading clip and trailing clip can be added along with the content. This helps to place advertisements at the leading and trailing space.
c. Rhozet provides preset configuration for various types. For example, if content needs to be displayed on ipod, then it provides a preset for ipod, which holds all the relevant information related to ipod supported file format. It also provides flexibility to change the parameters available in the preset.

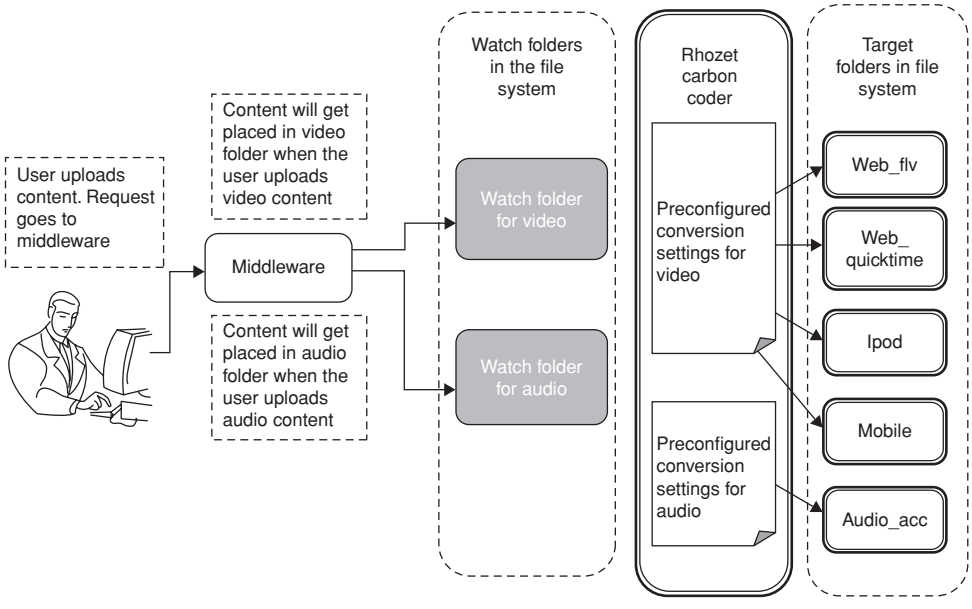


Figure 16.7 User uploading content: Preparing the content for pre-defined formats with the help of Rhozet.

d. Watch Folder need not be available in the same server. It could be available in a different server location. Carbon Admin provides functionality to connect to the folder through FTP mechanism or through remote path.
e. If various watch folders are provided and lot of transcoding is done, then the folders can be prioritized. Accordingly, Rhozet gives preference to the content to be transcoded.

16.4.3 Rhozet in a Representative Solution

If YouTube content needs to be shown on TV and mobile in addition to Web/PC the architecture in Figures 16.7 and 16.8 may be used.

If the requirement is for real-time processing, then Rhozet could be used to prepare the content at the time of fetching, rather than the time of uploading as is used in the batch processing mechanism (see Figure 16.8).

16.4.4 Rhozet in Personal Multimedia Content Distribution Solution

Table 16.4 gives the list of requirements for personal multimedia content distribution across multiple networks and devices as expected from Rhozet and its contribution.

16.4.5 Rhozet: Summary

Rhozet is a powerful tool for transcoding video contents. It provides extensive support for various popular video formats. It also provides lot of flexibility and dynamic functionality to transcode video content. In addition, Rhozet takes care of nonfunctional requirements like

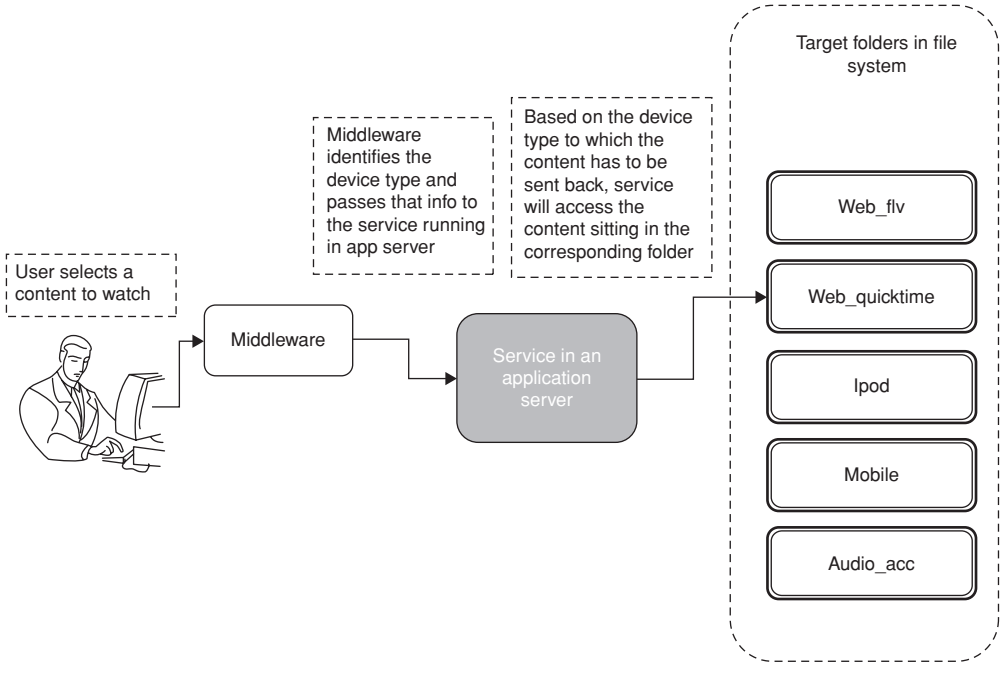


Figure 16.8 User playing content (Web with flv support is considered for this scenario).

Table 16.4 Requirements expected from Rhozet and its contribution.

S.No	Requirements	Rhozet
1	Multiple device rendering – images	Rhozet is not for suitable for image transcoding
2	Multiple device rendering – audios	Rhozet provides support for audio codecs like AAC, WAV, MP3 and so on
3	Multiple device rendering – videos	Extensive support for video content conversion. Rhozet provides lot of intelligent mechanism in this area
4	Real time transcoding	Could be achieved with the help of Carbon coder server and Micro Gridding mechanism, where content gets split into multiple pieces and then get transcoded simultaneously. This mechanism reduces transcoding time
5	Batch processing	Good support for this with the help of watch folders and queuing systems
6	Integrating with other applications	Rhozet can be controlled by other applications easily. It provides an XML based integration mechanism where other applications, irrespective of the technology can interact with Rhozet

performance, scalability and interoperability with its robust architecture. Considering audio content, format supported by Rhozet is limited to AAC, WAV, MP3 and PCM. It did not provide an image transcoding solution at the time of writing this book.

16.5 Architecture of a System that Embodies the Above Concepts

16.5.1 Solution Architecture Diagram

Figure 16.9 shows the functional architecture diagram of Convergence Gateway, designed and developed at Infosys [16].

16.5.1.1 Core Engine

The objective of the "core engine" is to generate the user experience for browsing a particular service. For example in the case of a "Digital Content Retailing" service, it will present the user with a catalog of available videos (movies, music, sports etc.) and in the case of "Banking" it will present the user with a listing of possible transactions. So the role of "Core Engine" is essentially to present textual / graphical information to the user.

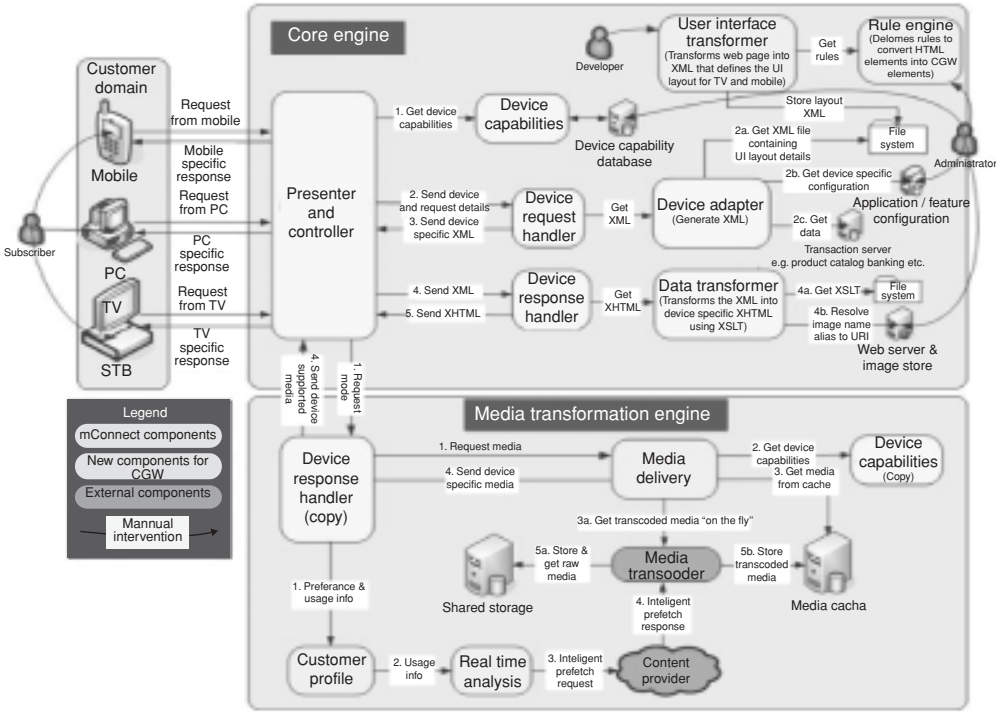


Figure 16.9 Architecture of a system that transforms content for suitable viewing in different channels.

Following is a short description of various subsystems that are part of "Core Engine":

The "Presenter and Controller" Subsystem receives the request from a device (PC, mobile or TV) along with the details of the device. It then processes the request in the sequence mentioned in the diagram and sends the response to requesting device in the presentation format that is supported by the device.

The "Device Capabilities" Subsystem has an exhaustive list of the devices (mobile and set top box) and their capabilities, including but not limited to presentation standards and supported interfaces.

The "Device Request Handler" Subsystem amalgamates the "standardized" XML templates predefined for a particular device with the information from the transaction server of client device and generates an XML. The combined XML is sent to the "presenter and controller" subsystem and contains information to construct the device-specific user interface.

The "Transformer" Subsystem gets the XSLT (extensible stylesheet language transformations) from the "file system" corresponding to the device. It then transforms the XML from "device request handler" using the XSLT obtained previously into device specific "human readable" document that is generally in XHTML format. While doing this transformation it takes care of specific transformations like replacing the names / alias in web / mobile interface with an image URI (uniform resource locator) and adds a virtual keyboard whenever there is a text entry box in web interface.

The "Device Response Handler" Subsystem gets the XHTML page from the "transformer" subsystem and sends it to the "presenter and controller" subsystem to respond to the device request.

16.5.1.2 Media Transformation Engine

The objective of the "media transformation engine" is to transform media from one format/codec to another based on the capabilities of the display machine [3–10].

A short description follows of various subsystems that are part of the "media transformation engine".

The "media delivery" subsystem delivers the requested media. It first gets the capabilities of the requesting device from the "device capability" subsystem. It then checks the "media cache" for the availability of the requested media in the device supported format. If the media is available in the media cache, it gets the media and delivers the same to the requesting device. Otherwise, it requests the "media transcoder" subsystem to get the desired media from "shared storage" and transcode it "on the fly". The transcoded media are then delivered to the requesting device.

To deliver requested media in shortest possible time, customer usage information is dynamically tracked. Most popular videos in most widely used formats are transcoded ahead of time and stored in the "media cache". The "customer profile" subsystem receives the customer preferences and usage information from the "media purchase and delivery" subsystem. It passes

Table 16.5 Benefits of an architecture that enables reuse of applications and content across channels.

Features	Benefits
Flexible platform to enable rapid development and deployment of services across multiple screens with consistent user experience	• Shorter time to develop and market • Elimination of duplicate effort • Sharing leads to higher ROI on content • Cross-Channel Advertising opportunity
Single intuitive home page for all applications regardless of channel/device	• Greater customer satisfaction leading to longer time spent on the portal, more monetizing opportunity and reduced churn
Consistent user experience across multiple screens (TV, mobile, PC)	• Stronger brand recognition
Personalized user experience across multiple screens	• Greater customer satisfaction leading to higher customer loyalty
Flexible rule-based configuration of applications and features per screen and per user profile	• Better control and operation of services leading to significant operating efficiency

this information to the "real time analysis" subsystem, which constructs a list of media objects that are most likely to be requested in the near future based upon the usage pattern. Such media objects are pre-fetched from the "content provider" and given to media trans-coder subsystem. The "media transcoder" subsystem stores the raw media in the "shared storage". The media objects are also trans-coded in most popular formats and stored in the "media cache". The entire process is executed in the background.

16.6 Benefits of the Proposed Architecture

The proposed platform overcomes the drawbacks of the existing way of doing business in silos and enables CSPs to avoid duplication of effort across business units dedicated to each silo, bring new products and services to market quickly, generate high ROI on content and improve user satisfaction. Table 16.5 summarizes the benefits of the proposed architecture.

16.7 Case Study: Virtual Personal Multimedia Library

Virtual Personal Multimedia Library (VPML) is an application that enables uploading and organizing all personal multimedia content such as photos, videos and songs in the network such that the content can be accessed by anyone from anywhere using any device (TV, mobile, PC). The concept is shown in Figure 16.10 where a tourist takes a photo/video using his mobile handset and the photo/video is automatically uploaded into the VPML via a WiFi or a 3G connection. Once the content is in the VPML system, it can be accessed using a TV, mobile or PC.

While uploading the content into the VPML system is straightforward, the challenges lie in making the content available for various devices (TV, mobile, PC) to consume. This is exactly the multichannel media delivery problem that the CSPs are interested in.

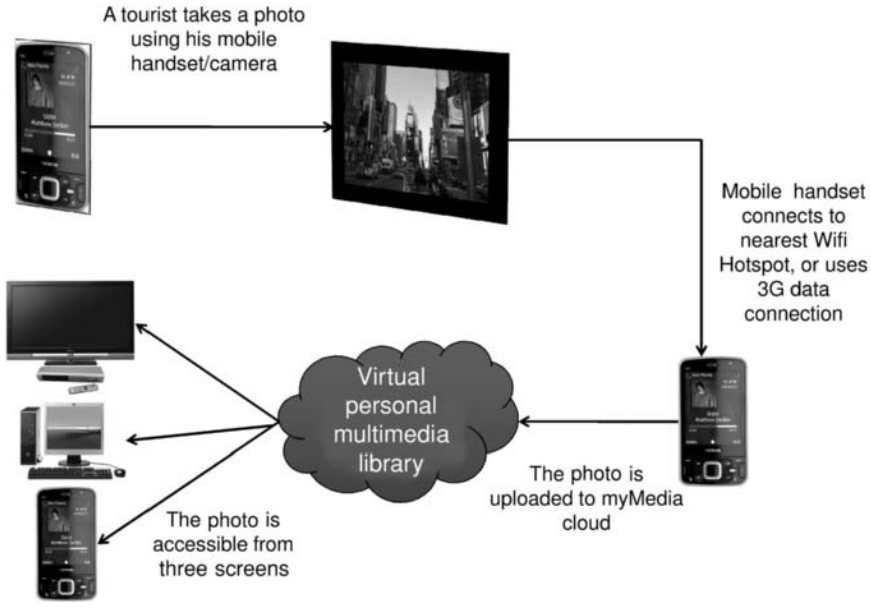


Figure 16.10 Virtual personal multimedia library use case.

In order to make this happen, the VPML transcodes the uploaded content into various formats and bitrates. Depending on the device that wants to access the content (TV, mobile, PC) and its capability, the appropriate version of the content is served to the requesting device as described in the solution architecture diagram.

In converged applications, such as VPML, it is important that the user experience is delivered optimally. For example, due to the flexibility of having a qwerty keyboard and mouse, one can do more complex functions, such as organizing the multimedia assets into albums, naming the albums and searching for content on the PC. However, the functionality is limited on the TV and navigation is more user-friendly as shown in Figure 16.11. Photos/videos can be displayed on full screen with high resolution on TV but due to the limitation of both screen size and the wireless bandwidth, only a lower resolution version of photos/videos can be displayed on the mobile handsets. Furthermore, due to the limited screen size, a limited amount of content can be fitted in one screen and pagination is needed to accommodate the same amount of information that usually fits in a single screen of PC. Notice that, in Figure 16.11, the mobile handset does not display the thumbnails of photos/videos on the screen while the PC and TV do.

16.8 Summary

Communication Service Providers who own multiple delivery networks, such as, broadband, mobile (Cellular, WiMax etc.), and television networks (IPTV, Digital Cable, Digital Satellite etc.) can reach out to their customers in multiple ways. However, businesses within CSPs have been running in silos, meaning that each network, whether broadband, mobile or television, has

Figure 16.11 Virtual personal multimedia library user experience in different screens.

been developing its own applications, acquiring its own content and running its own profit and loss without considering the possible gains that might incur from synergizing their businesses. In order to overcome the above-mentioned drawbacks, it is important that the application is written once and used across multiple networks. That will eliminate the duplication of effort in writing the same application multiple times, once for each network; it will reduce time to market new applications and services and improve the ROI on licensed content.

However, the heterogeneity in terms of screen size, viewing distance, aspect ratio, resolution, user interface, codecs, media formats and bandwidth availability for TV, mobile and Web/PC makes it challenging to transform an application written for one screen to be made available to other screens. Approaches to address the challenges of multi-channel content/application rendering include rule-based media transformation, which enables transformation of media from one format/codec to another based on the capabilities of the display device, static versus dynamic transformation, which transcodes the more "popular" content ahead of time and uses "on-the-fly" transcoding for less frequently accessed content, and use of "standardized" templates for different categories of applications.

A commercial transcoder called Rhozet was presented in detail to provide a complete understanding of how transcoding systems work in practice. In fact, Rhozet is a powerful tool for transcoding video and it provides extensive support for various formats of video commonly used in the market. Rhozet offers a lot of flexibility while also taking care of nonfunctional requirements like performance, scalability and interoperability with its robust architecture.

The chapter ended with a case study of VPML. This is an application that enables an end user automatically to upload, and organize personal multimedia content, such as, photos, videos, songs, and so on, in the network such that they can be accessed by anyone from anywhere using any device (TV, mobile, PC). While uploading the content into the VPML system is straightforward, the challenges lie in making the content available for and accessible to various devices (TV, mobile, PC) with an optimal user experience. In order to make this happen, VPML trans-codes the uploaded content into various formats and bitrates. Depending on the device that wants to access the content (TV, mobile, PC) and its capability, the appropriate version of the content is served to the requesting device.

References

[1] Technological convergence: http://en.wikipedia.org/wiki/Technological_convergence (accessed June 9, 2010).

[2] Rhozet: Universal Media Transcoding: http://www.rhozet.com (accessed June 9, 2010).

[3] White Paper, IBM Transcoding Solution and Services. http://www.research.ibm.com/networked_data_systems/transcoding/transcodef.pdf (accessed June 9, 2010).

[4] Han, R. and Bhagwat, P. (1998) Dynamic adaptation in an image transcoding proxy for mobile web browsing. *IEEE Personal Communications Magazine*, **5**(6): 8–17.

[5] Mohan, R., Smith, J. and Li, C.-S. (1999) Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, **5**(1): 104–114.

[6] Smith, J.R., Mohan, R. and Li, C.-S. (1998) *Content-Based Transcoding of Images in the Internet*. Proceedings of the International Conference on Image Processing (ICIP), 1998.

[7] Smith, J.R., Mohan, R. and Li, C.-S. (1998) *Transcoding Internet Content for Heterogenous Client Devices*. Proceedings of the IEEE Inter. Symp. on Circuits, Syst. (ISCAS), Special session on Next Generation Internet, June, 1998 PDF.

[8] Smith, J.R. (1999) Digital video libraries and the internet. *IEEE Communications Mag*, Special issue on the Next Generation Internet, January.

[9] Han, R. and Smith, J.R. (1999) Internet transcoding for universal access, in *Multimedia Communications Handbook* (ed. J. Gibson). CRC Press.

[10] Anystream: http://www.anystream.com/ (accessed June 9, 2010).

[11] K2 coder: http://www.grassvalley.com/docs/DataSheets/servers/k2_coder/SER-3030D.pdf (accessed June 9, 2010).

[12] Envivio 4Caster C4: http://www.envivio.com/products/cgs_4caster_c4.php (accessed June 9, 2010).

[13] Envivio Products: http://www.envivio.com/products/ (accessed June 9, 2010).

[14] Telestream: http://www.telestream.net/ (accessed June 9, 2010).

[15] Ripcode: http://www.ripcode.com/ (accessed June 9, 2010).

[16] Paul, S. and Jain, M. (2010) Convergence gateway for a multichannel viewing experience. *Annual Review of Communications*, **61**, 221–229.

17

Quality of Service (QoS) in IPTV

Providing television services over IP networks to households comes with significant responsibility in terms of ensuring that customers receive at least as good a user experience as in existing cable/satellite networks and the service is at least as reliable as the existing alternatives (cable and satellite TV) in terms of availability and robustness. A typical IPTV network deployment is shown in Figure 17.1.

Quality of service requirements for IPTV networks need to be specified at multiple layers: application layer, transport layer and network layer. While application-layer requirements focus on the end-to-end application-level bit rates, transport-layer requirements focus on latency, jitter and packet loss rate, and network-layer requirements focus on packet loss rate as a function of bit rate and the interval between two consecutive uncorrected packet loss events. The next sections provide a detailed description of these requirements [1–20].

17.1 QoS Requirements: Application Layer

17.1.1 Standard-Definition TV (SDTV): Minimum Objectives

Table 17.1 lists the recommended minimum video application layer performance objectives at the MPEG level, prior to IP encapsulation for broadcast SDTV (480i/576i) [3, 9, 15, 17, 21]. The audio stream bit rates are additional and specified separately below.

The numbers specified in Tables 17.1–17.3 assume 4 : 3 aspect ratio, 720 pixels × 480 lines (North America), 720 pixels × 576 lines (Europe), 30 frames/s (North America), 25 frames/s (Europe) and two interlaced fields per frame.

17.1.1.1 Broadcast TV

The MPEG-2 bit rates shown in Table 17.1 are the minimum required to provide adequate quality over a range of broadcast program material complexity. Note that many competing services (e.g., digital cable and satellite) use higher MPEG-2 bit rates and often VBR encoding [11]. Where access link bandwidth permits, service providers should use higher bit rates, particularly for broadcast materials with highly complex image content, such as sports.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

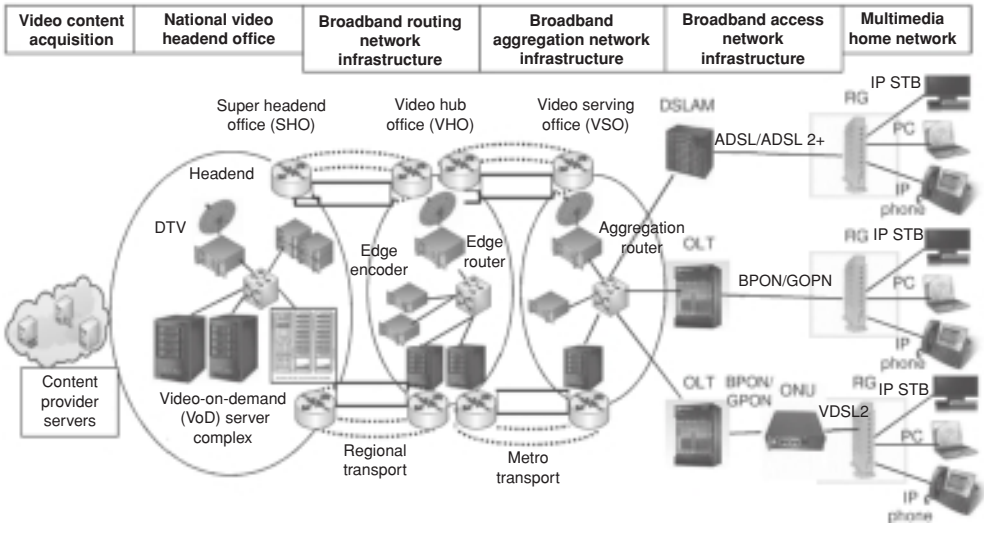


Figure 17.1 Representative IPTV network.

Table 17.1 Recommended minimum application layer performance for standard definition broadcast program sources.

Video codec standard	Minimum bit rate (video only)
MPEG-2 – Main profile at Main level	2.5 Mbits/s CBR
MPEG-4 AVC (Main profile at Level 3.0)	1.75 Mbits/s CBR
SMPTE VC-1	1.75 Mbits/s CBR
AVS	1.75 Mbits/s CBR

Table 17.2 Recommended minimum audio application layer performance for standard definition sources.

Audio codec standard	Number of channels	Minimum bit rate (kbps)
MPEG Layer II	Mono or stereo	128 for stereo
Dolby Digital (AC-3)	5.1 if available, else left/right stereo pair	384 for 5.1 / 128 for stereo
AAC	Stereo	96 for stereo
MP3 (MPEG-1, Layer 3)	Stereo	128

Table 17.3 SDTV audio – video synchronization requirements.

Audio – video	Audio lead video	Audio lag video
Synchronization	15 ms maximum	45 ms maximum

Table 17.4 Recommended minimum application layer performance for standard definition VoD.

Video codec standard	Minimum bit rate (video only)
MPEG-2 – Main profile at Main level	3.18 Mbits/s CBR
MPEG-4 AVC (Main profile at Level 3)	2.1 Mbits/s CBR
SMPTE VC-1	2.1 Mbits/s CBR
AVS	2.1 Mbits/s CBR

17.1.1.2 Video-on-Demand (VoD)

Application layer performance for video-on-demand (VoD) and other premium content such as pay per view in standard definition format is expected to be similar to that of regular broadcast material. However, subscriber expectation may be higher because of additional fees paid to access the content and comparison to alternative delivery options. In the case of VoD, users may compare to VoD materials delivered over digital cable systems or even DVDs.

Video-on-demand application layer parameters in North America are defined by Cable Labs [22], and since existing VoD content is aligned with these parameters used by cable providers, consumers will compare the quality levels. Therefore, IPTV services should follow these as the minimum guidelines. Recommendations for MPEG-4 AVC or VC-1 and AVS encoded VoD materials assume a 1.5 × improvement in bit rate, aligned with the state of commercial deployments of these encoders.

The numbers specified in Tables 17.4 and 17.5 assume the source material in NTSC (North America), PAL/SECAM (Europe/Asia Pacific), 4 : 3 aspect ratio, 352 pixels × 480 lines (North America), 352 pixels × 576 lines (Europe), 30 frames/s (North America), 25 frames/s (Europe) and two interlaced fields per frame [3, 9, 15, 17, 21].

17.1.2 High-Definition TV (HDTV): Minimum Objectives

Table 17.6 lists the recommended minimum video application layer performance objectives for broadcast HDTV (720p / 1080i).

The numbers specified in Table 17.6 assume the source material in ATSC (North America), DVB (Europe), 16 : 9 aspect ratio, 1280 pixels × 720 lines with 50, 59.94, 60 progressive scan frames/sec (North America), 1920 pixels × 1080 lines with 29.97, 30 interlaced frames/s and two fields per frame (Europe), 29.97 (59.94i), 30 (60i) interlaced frames per second, two fields per frame.

Table 17.5 Recommended minimum audio application layer performance for VoD.

Audio codec standard	Number of channels	Minimum bit rate (kbit/s)
Dolby Digital (AC-3)	5.1 if available, else left/right stereo pair	384 for 5.1 / 192 for stereo

Table 17.6 Recommended minimum application layer performance for high definition (HD) broadcast program sources.

Video codec standard	Minimum bit rate (video only)
MPEG-2 – Main profile at Main level	15 Mbits/s CBR
MPEG-4 AVC (Main profile at Level 4)	10 Mbits/s CBR
SMPTE VC-1	10 Mbits/s CBR
AVS	10 Mbits/s CBR

17.1.2.1 Broadcast TV

The MPEG-2 bit rates shown in Table 17.6 are the minimum required to provide adequate quality over a range of broadcast program material complexity. Note that many competing services (for example, digital cable and satellite) use higher MPEG-2 bit rates and often VBR encoding. Where access link bandwidth permits, service providers should use higher bit rates, particularly for broadcast materials with highly complex image content, such as sports.

The numbers specified in Table 17.7 assume the source material in ATSC (North America), DVB (Europe), more than one audio track to support multiple languages, multichannel audio for surround sound effects, Dolby 5.1 (used for special events like concerts and sporting events), audio sample rate of 48 KHz for Dolby digital, 16 KHz to 44.1 KHz for MP3, 32 KHz, 44.1 KHz or 48 KHz for DVB audio sources.

17.1.2.2 Video-on-Demand

Application layer performance for VoD and other premium content such as pay per view in high definition format is expected to be similar to that of regular broadcast material (as shown in Tables 17.8 and 17.9).

17.2 QoS Requirements: Transport Layer

The main elements in transport layer QoS requirement include latency, jitter and packet loss [6, 12, 13, 21]. If the jitter buffer at the set top box (STB) is provisioned to match network and video element performance, reasonable end-to-end delay and jitter can be tolerated. However, quality of experience (QoE) for video is highly sensitive to information loss, and all losses

Table 17.7 Recommended minimum audio application layer performance for high definition sources.

Audio codec standard	Number of channels	Minimum bit rate (kbits/s)
MPEG Layer II	Mono or stereo	128 for stereo
Dolby Digital (AC-3)	5.1 if available, else left/right stereo pair	384 for 5.1/128 for stereo
AAC	Stereo	96 for stereo
MP3 (MPEG-1, Layer 3)	Stereo	128

Table 17.8 Recommended minimum application layer performance for high definition (HD) VoD.

Video codec standard	Minimum bit rate (video only)
MPEG-2 – Main profile at Main level	15 Mbits/s CBR
MPEG-4 AVC (Main profile at Level 4)	10 Mbits/s CBR
SMPTE VC-1	10 Mbits/s CBR
AVS	10 Mbits/s CBR

are not equal. Impact on QoE due to system information and header losses is different from the losses resulting in impairment of I and P frames, which in turn is different from losses that affect B frames. Furthermore, the impact on QoE is dependent on codec, transport stream packetization, loss distance and loss profile. High-encoding bit rates make the stream more vulnerable to packet loss impairments. Specifically, impairments due to loss on a higher rate video stream occur more frequently for the same packet loss ratio simply because there are more packets per second transmitted and each one has the same probability to be affected. Set-top boxes usually deploy error concealment algorithms that can help mitigate perceptual impact of some losses.

Here is an example of how video can become impaired due to a loss of single IP packet that contains seven MPEG-2 packets. Figure 17.2 shows the impact if the lost information is from a B frame (on the left) or from an I-frame (on the right). Note that the impact in case of I-frame is significant and noticeable as it happens to be a key frame used in the compression of subsequent P and B frames, and the impairment propagates in time across 14 frames of video or almost a half second (assuming 33 ms per frame). On the other hand, if the lost packet impacted a B frame, the impairment impacted only that frame with duration of 33 ms and the impairment is barely noticeable. This example assumes that the decoder is not running any error concealment algorithm.

In order to meet the quality of experience of end-users and hence the application-layer minimum guidelines, network transport has to meet certain packet loss, delay and jitter requirements. This section describes those transport-layer requirements.

There is a trade-off between loss and jitter and the desired target numbers for these parameters are highly dependent on the design of set top box (STB). Typical jitter buffer size for STBs is 100–500 ms of SDTV video and as long as the jitter is within these limits, quality of experience will not suffer. If jitter or delay variation exceeds this limit, it will manifest as

Table 17.9 Recommended minimum audio application layer performance for high definition VoD.

Audio codec standard	Number of channels	Minimum bit rate (kbits/s)
MPEG Layer II	Mono or stereo	128 for stereo
Dolby Digital (AC-3)	5.1 if available, else left/right stereo pair	384 for 5.1/128 for stereo
AAC	Stereo	96 for stereo
MP3 (MPEG-1, Layer 3)	Stereo	128

Figure 17.2 Impact of single IP packet loss (B Frame and I Frame).

packet loss. While increasing jitter buffer would reduce packet loss, it would increase latency and hence the start-up time of video.

Packet loss is characterized by loss distance, loss duration and loss rate. Loss distance is defined as the spacing in time between two consecutive error events. Loss duration is the number of consecutive packet losses during an error event. Loss rate refers to the percentage of packets lost. If the network is not able to meet the specified performance objectives with respect to packet loss, forward error correction (FEC) or interleaving techniques can be used at the network layer and error concealment, application-layer FEC and/or automatic repeat request (ARQ) can be used at the application layer to achieve the required performance level.

In order to balance the interleaver depth needed to protect against impulse noise induced by xDSL errors and the delay added to applications due to interleaving (note the trade-off between delay and loss), while limiting visual impairments to an average of one per 60 minutes of SDTV resolution video stream, the loss period is recommended to be less than 16 ms. Actual number of packets lost during this time period will depend on the bit rate of the video stream.

In DSL access lines, a packet loss outage can last as long as 10–20 seconds, which is certainly not acceptable for IPTV services. Using SONET/SDH protection switching mechanism, the packet loss duration can be reduced to 50 ms. This duration may be stretched to 250 ms in the IP networks where MPLS fast reroute and fast IGMP convergence techniques are used.

Packet loss bursts of up to 30 s may happen when the IP routing tables need a complete reconvergence. Certainly such long bursts should be treated as service outage rather than an in-service quality defect.

The overall goal for IPTV service provider would be to minimize visible artifacts to as few as possible using a combination of network-layer and application-layer performance enhancements, such as loss recovery mechanisms (for example, FEC, interleaver) and loss-mitigation mechanisms (for example, decoder loss concealment).

Table 17.10 Recommended minimum transport layer parameters for satisfactory QoE for MPEG-2 encoded SDTV services.

Transport stream bit rate (Mbits/s)	Latency	Jitter	Maximum duration of a single error	Corresponding loss period in IP packets	Loss distance	Corresponding average IP video stream packet loss rate
3.0	<200 ms	<50 ms	≤16 ms	6 IP packets	1 error event/hr	≤5.85E-06
3.75	<200 ms	<50 ms	≤16 ms	7 IP packets	1 error event/hr	≤5.46E-06
5.0	<200 ms	<50 ms	≤16 ms	9 IP packets	1 error event/hr	≤5.26E-06

17.2.1 Standard-Definition Video: Minimum Objectives

Table 17.10 lists the recommended minimum transport layer performance objectives for satisfactory quality of experience (QoE) for MPEG-2 encoded SDTV services.

17.2.1.1 Broadcast TV

The numbers specified in Table 17.10 apply to IP flows containing video streams only, after using any application layer protection mechanisms at the customer premise equipment, and assume MPEG-2 codec, MPEG-2 transport stream with seven 188-byte packets per IP packet, and no or minimal loss concealment at STB.

The numbers specified in Table 17.11 apply to IP flows containing video streams only, after using any application layer protection mechanisms at the customer premise, and assume MPEG-4 AVC or VC-1 codec, MPEG-2 transport stream with seven 188-byte packets per IP packet, and no or minimal loss concealment at STB.

17.2.1.2 Video-on-Demand (VoD)

Application layer performance for VoD and other premium content such as pay per view in standard definition format is expected to be similar to that of regular broadcast material. However, to meet subscriber expectations that may be higher because of additional fees paid to access the content and comparison to alternative delivery options, the transport-level requirements are expected to be more stringent.

Table 17.11 Recommended minimum transport layer parameters for satisfactory QoE for MPEG-4 AVC or VC-1 and AVS encoded SDTV services.

Transport stream bit rate (Mbits/s)	Latency	Jitter	Maximum duration of a single error	Corresponding loss period in IP packets	Loss distance	Corresponding average IP Video stream packet loss rate
1.75	<200 ms	<50 ms	≤16 ms	4 IP packets	1 error event/hr	≤6.68E-06
2.0	<200 ms	<50 ms	≤16 ms	5 IP packets	1 error event/hr	≤7.31E-06
2.5	<200 ms	<50 ms	≤16 ms	5 IP packets	1 error event/hr	≤5.85E-06
3.0	<200 ms	<50 ms	≤16 ms	6 IP packets	1 error event/hr	≤5.85E-06

Table 17.12 Recommended minimum transport layer parameters for satisfactory QoE for MPEG-2 encoded HDTV services.

Transport stream bit rate (Mbits/s)	Latency	Jitter	Maximum duration of a single error	Corresponding loss period in IP packets	Loss distance	Corresponding average IP video stream packet loss rate
15.0	<200 ms	<50 ms	≤16 ms	24 IP packets	1 error event per 4 hours	≤1.17E-06
17	<200 ms	<50 ms	≤16 ms	27 IP packets	1 error event per 4 hours	≤1.16E-06
18.1	<200 ms	<50 ms	≤16 ms	29 IP packets	1 error event per 4 hours	≤1.17E-06

17.2.2 High-Definition Video: Minimum Objectives

High-definition television (HDTV) services are expected to have at most one visible impairment event per 12 hours. Assuming that not all errors result in a visible impairment, transport-layer requirements specify a value of four hours as the minimum loss distance for HDTV services. This assumption is based on the fact that loss of B-frame information does not necessarily result in visual impairment and furthermore, HDTV decoders use error concealment.

17.2.2.1 Broadcast TV

The numbers specified in Table 17.12 apply to IP flows containing video streams only, after using any application layer protection mechanisms at the customer premise, and assume MPEG-2 codec, MPEG-2 transport stream with seven 188-byte packets per IP packet, and some level of loss concealment at STB.

The numbers specified in Table 17.13 apply to IP flows containing video streams only, after using any application layer protection mechanisms at the customer premise, and assume MPEG-4 AVC or VC-1 and AVS codec, MPEG-2 transport stream with seven 188-byte packets per IP packet, and some level of concealment at STB.

To achieve a packet loss rate in the range of 10^{-6} recommended for video services may require special error-control techniques to achieve the target.

Table 17.13 Recommended minimum transport layer parameters for satisfactory QoE for MPEG-4 AVC or VC-1 and AVS encoded HDTV services.

Transport stream bit rate (Mbits/s)	Latency	Jitter	Maximum duration of a single error	Corresponding loss period in IP packets	Loss distance	Corresponding average IP Video stream packet loss rate
8	<200 ms	<50 ms	≤16 ms	14 IP packets	1 error event per 4 hours	≤1.28E-06
10	<200 ms	<50 ms	≤16 ms	17 IP packets	1 error event per 4 hours	≤1.24E-06
12	<200 ms	<50 ms	≤16 ms	20 IP packets	1 error event per 4 hours	≤1.22E-06

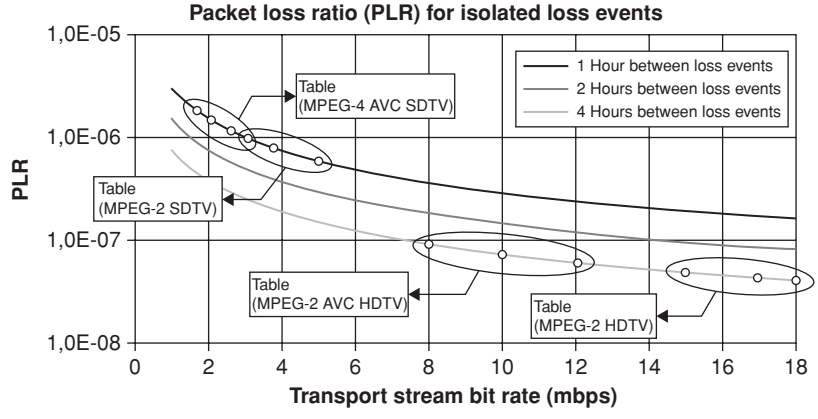


Figure 17.3 PLR required to meet average time between loss events of 1, 2 and 4 hours assuming isolated lost packets (Source [21]).

17.3 QoS Requirements: Network Layer

The main QoS requirement at the network layer is about maintaining packet loss ratio below a threshold in order to meet the transport layer requirements [12, 13]. For example, if the transport layer requires that the number of loss events be limited to "n" within a period of "m" hours, then based on the transport stream bit rate and the duration of a loss event, there would be a limit to the packet loss ratio at the network layer.

The network layer performance objectives are summarized in the figures below. Assuming $n = 1$ and m = 1, 2 and 4 hours, the packet loss ratios are plotted as a function of bit rate in Figure 17.3 for uncorrected loss events for isolated packet loss events. Specific points shown in Figure 17.3 correspond to SDTV video with a loss distance of 1 hour between packet loss events. Also shown are points corresponding to HDTV video with a loss distance of 4 hours between packet loss events. The figure assumes that each IP packet carries seven MPEG data packets, each 188 bytes long.

Figure 17.4 shows packet-loss ratios as a function of bit rate and time between uncorrected loss events for a typical DSL burst loss event of 16 ms. Rounding to an integer number of lost/corrupted IP packets causes the so-called ripples in the plot, Here's a sample calculation of packet loss ratio corresponding to an MPEG-2 transport stream at a bit rate of 2 Mbits/s for a loss event of 16 ms every second plotted in Figure 17.4.

MPEG-2 transport stream at a bit rate of 2 Mbits/s translates to:

$$= 2 \text{ Mbits/s / 8 bits per byte / 188 bytes per MPEG pkt}$$
$$= 1329.8 \text{ MPEG packets per second.}$$

Total IP packets per second = 1329.8 / 7 MPEG packets per IP packet:

$$= 190 \text{ IP packets per second.}$$

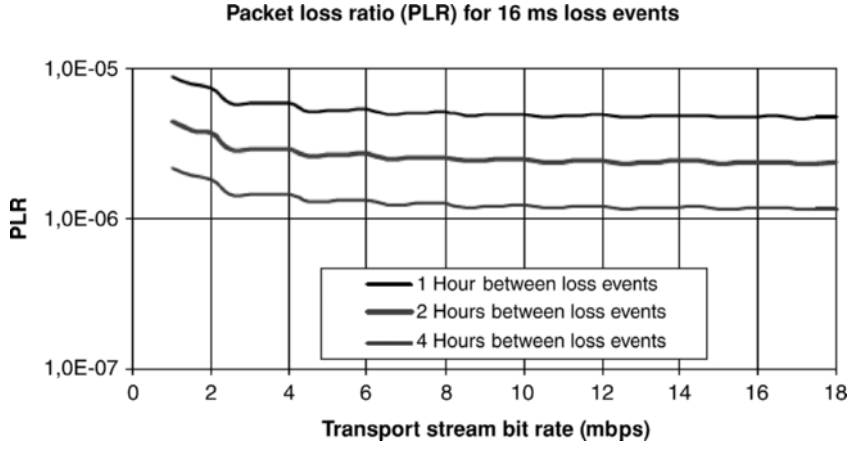


Figure 17.4 PLR required to meet average time between loss events of 1, 2 and 4 hours assuming each event is an uncorrectable DSL error that loses 16 milliseconds of contiguous data (Source [21]).

A loss of 16 ms corresponds to = 190 IP packets per second $\times$ 0.016 seconds:

$$= 3.04 \text{ IP packets lost.}$$

Because an entire IP packet is lost if a part of a packet is lost, this is rounded to the next integer = 4 IP packets. And because the lost bytes are not necessarily aligned to IP packet boundaries, this would be further rounded to 5 IP packets.

Note that there are other sources of impairment at the network layer in addition to the above-mentioned packet-loss ratio. Specifically, there could be gross impairments resulting from video frame drops, frame repetitions (freeze frames), or short duration loss of intelligible audio or video or control that may result from protection switching. In general, requirements on such events are usually specified by frequency of the error event per time unit – for example, a maximum of one severe error per day and the duration of the impairment.

17.4 QoE Requirements: Control Functions

17.4.1 QoE Requirements for Channel Zapping Time

End-user experience is heavily influenced by channel zapping time (channel switching time). Channel zapping time is primarily determined by the time required to have a proper frame at the STB in order to start the decoding process for the new channel. Furthermore, channel zapping requests can occur under the following circumstances:

- Random channel selection by entering channel number using remote control.
- Channel selection using channel up/down button in remote control.
- Channel selection using channel up/down button in STB.
- Channel selection on interactive program guide (IPG) menu.

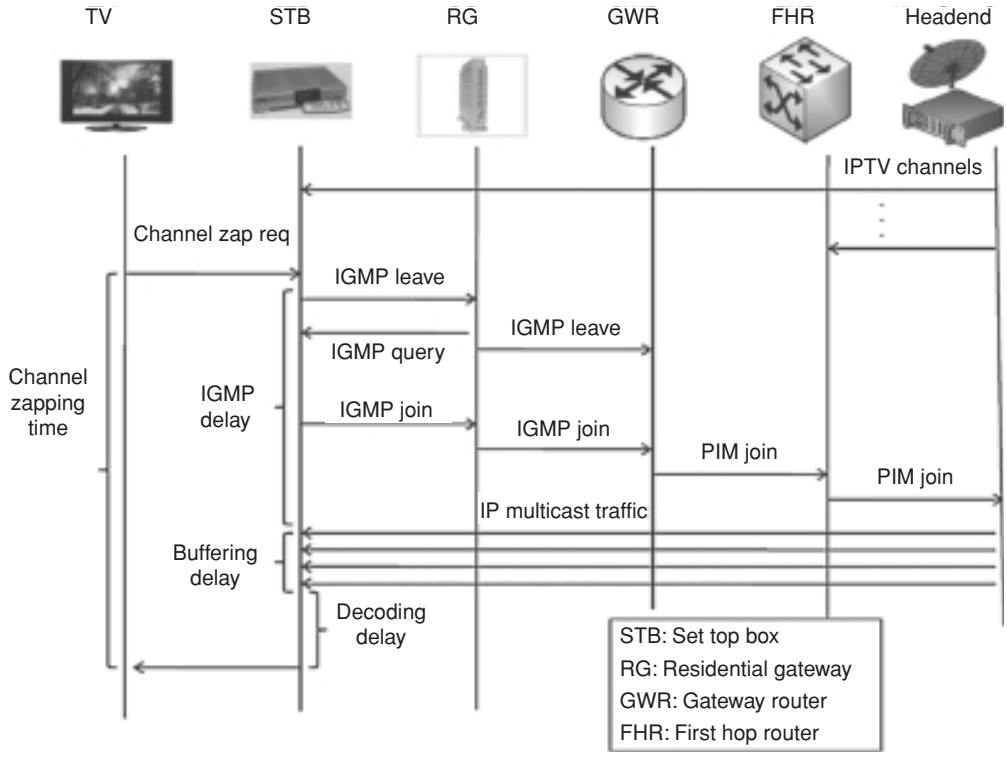


Figure 17.5 An example of channel zapping process.

- Requesting metadata in electronic program guide (EPG) or interactive program guide (IPG).
- Powering on STB/TV and tuning to initial channel assigned by IPG.

Channel zapping time depends on three components (Figure 17.5):

- IGMP delay;
- buffering delay;
- decoding delay.

17.4.1.1 IGMP Delay

A channel zap request is triggered when a channel is changed. Each channel is mapped by STB to a multicast group address carried in the IGMP message. Thus, leaving the previous channel triggers an IGMP Leave message with the multicast address corresponding to the previous channel, while joining the new channel generates an IGMP Join message with the multicast address corresponding to the new channel. IGMP Leave message is sent to the residential gateway (RG), which in turn, sends it to the gateway router (GWR), which may generate a PIM Prune message if there is no receiver downstream of first hop router (FHR). An IGMP

Join message is also sent to the RG, which, in turn, sends it to the GWR, which may generate a PIM Join message towards the FHR. The FHR may forward the PIM Join message to the source of the multicast tree in PIM-SSM. In fact, RG plays an IGMP proxy role, and processes the IGMP messages and sends the corresponding IGMP request to the GWR. After an IGMP message is sent toward the source or the rendezvous point by the GWR, corresponding channel data is delivered to the end point.

17.4.1.2 Buffering Delay

As the packets corresponding to an IPTV stream are delivered to the STB, the STB places them in a buffer. Buffering delay is the duration between the arrival time of the first multicast packet in the buffer and the time when enough data packets are available in the STB for playback.

17.4.1.3 Decoding Delay

In order to decode the data stored in its buffer, the STB requires certain video frames (such as, an I-frame, as opposed to a B-frame) to be available. When the relevant data packets are available in the buffer, the STB starts the decoding process. This delay in processing buffered data and rendering them to TV screen is referred to as decoding delay. This type of delay includes both codec decoding delay, and I-frame acquisition delay.

The IPTV architecture is recommended to support means to provide channel zapping with sufficient QoE.

17.5 QoE Requirements: VoD Trick Mode

Video-on-demand trick mode provides VCR-like features in VoD services. For example, this mode provides the control mechanism to pause, play, rewind, fast forward and stop the video from the VoD server.

17.5.1 Trick Latency

Corresponding to each control function (video selection, play, pause, rewind, FFW, stop), there is a delay. Therefore, the QoE metrics for VoD transaction quality are expressed using the following parameters [4, 5, 14, 20]:

- **Video selection process delay:** duration between the time when the subject is selected to the time when content is displayed.
- **Play delay:** duration from the time the Play entry is selected to the time the content is displayed.
- **Stop delay:** duration between the time the Stop play video entry is selected to the time the content is stopped playing as indicated by video content display.
- **Rewind delay:** duration between the time the Rewind video entry is selected to the time the rewind action is executed as indicated on display device.

- **Pause delay:** duration between the time the Pause video entry was selected to the time the pause action is executed as indicated on display device.
- **Fast forward delay:** duration between the time the Fast Forward video entry is selected to the time the FFW action is executed as indicated on display device.

17.5.2 Requirements for VoD Trick Features

Since each trick feature latency directly affects QoE, it is expected to be sufficiently low to meet the user's requirement for QoE relating to VoD trick features.

17.6 IPTV QoS Requirements at a Glance

The QoS requirements of IPTV can be summarized as follows [4, 5, 12, 14, 20, 21]:

1. **Resiliency:** IPTV network should be able to recover from network failure rapidly in order to reduce network outage time. Expected failover time live video transmission should be less than 50 ms.
2. **Packet loss:** IPTV network should be able to avoid packet loss due to traffic bursts resulting from aggregation at the Edge and Aggregate networks. Video is highly compressed and more fragile than audio, and hence a single packet drop can cause a range of missing video frames leading to poor quality of experience.
3. **Packet reordering:** IPTV network should be able to avoid packet re-ordering resulting from scheduling of packets in the intermediate routers. Video is highly compressed and hence out of order packets can cause a range of corrupted video frames leading to poor quality of experience.
4. **Packet jitter:** IPTV network should be able to maintain packet jitter under 200 ms as most STBs come with 200 ms of jitter buffer. Some IP STBs provide up to 2 second of jitter buffer allowing the IPTV network some additional slack in terms of jitter. Any jitter beyond the prescribed limit will result in missing the playout deadline for the corresponding packet resulting in poor user experience.
5. **CBR and VBR traffic:** IPTV system should be able to support both constant bit rate (CBR) and variable bit rate (VBR) traffic. Simple encapsulation of standard MPEG-2 will result in CBR traffic while MPEG_2 Transport Stream over RTP will result in VBR traffic due to removal of null stuffing packets. Unless the STB is capable of supporting both variations, there might be issues with playing the delivered video affecting user experience.
6. **Channel Changing:** The IPTV network should be able to switch channels at user request in less than 1 s in order to maintain the same user experience as in cable/satellite television. The IPTV server delivers video to STBs using IP multicast except for on-demand video. Switching channel requires the STB to leave a multicast group and join a new multicast group. Furthermore, the STB has to wait for the I-frame in a group of pictures (GoP) to arrive so that it can start decoding the video. The latter component may result in a delay of as much as half a second (480 ms) while the former may contribute to as much delay, if not more.
7. **Differentiated QoS:** IPTV service providers should be able to provide the desired QoS to the IPTV traffic in the presence of other types of traffic in the same network. Furthermore,

IPTV will have different types of traffic, such as VoD traffic, broadcast TV traffic, network PVR traffic, trick play traffic and customers of different category with their own QoS requirements. All of these dimensions have to be covered via differentiated QoS support in the network.

8. **Deterministic QoS:** It is not enough for the service provider to offer differentiated service quality to customers, but it is of paramount importance to guarantee the agreed-upon QoS (via service level agreement or SLA contracts) to both the content providers as well as to end customers. QoS parameters consist of packet loss, delay, jitter, throughput, channel change time, failover time, availability and other parameters relevant to specific type of IPTV service.

17.7 Summary

Quality of service requirements for IPTV networks need to be specified at multiple layers: the application layer, the transport layer and the network layer [1–20]. While application-layer requirements focus on the end-to-end application level bit-rates, transport-layer requirements focus on latency, jitter and packet loss rate, and network-layer requirements focus on packet loss rate as a function of bit rate and the interval between two consecutive uncorrected packet loss events.

From the application-layer point of view, we looked at the requirements for both SDTV and HDTV and, in each case, we considered the requirements for broadcast TV and video-on-demand service. Quality of service requirements at the transport layer are captured by latency, jitter and packet loss, where packet loss is characterized by loss distance, loss duration and loss rate. Loss distance is defined as the spacing in time between two consecutive error events. Loss duration is the number of consecutive packet losses during an error event. Loss rate refers to the percentage of packets lost. The QoS requirements for transport layer require the underlying network to meet specific requirements. Network layer QoS is characterized by packet loss ratio. The main QoS requirement at the network layer is about maintaining packet loss ratio below a threshold in order to meet the transport layer requirements. For example, if the transport layer requires that the number of loss events be limited to “n” within a period of “m” hours, then based on the transport stream bit rate and the duration of a loss event, there would be a limit to the packet loss ratio at the network layer. If the network is not able to meet the specified performance objectives with respect to packet loss, FEC or interleaving techniques can be used at the network layer and error concealment, application-layer FEC and/or ARQ can be used at the application layer to achieve the required performance level. In addition to bearer path QoS requirements, we also discussed the QoS requirements for the control functions including limiting the channel zapping time under a threshold and reducing video on demand trick-mode latency as well. Finally, IPTV QoS requirements across layers were presented at a glance.

References

[1] ITU-T Recommendation E.800 (1994) Terms and Definitions Related to Quality of Service and Network Performance Including Dependability.

[2] ITU-T F.700 (2000) Framework recommendation for multimedia services, November.

[3] ITU T Recommendation H.264, Series H: audiovisual and multimedia systems: infrastructure of audiovisual services – Coding of moving video. (2009) Advanced video coding for generic audiovisual services.
[4] ITU-T Recommendation P.10/G.100 Appendix I (2007) Definition of Quality of Experience (QoE), January
[5] ITU-T Recommendation P.800 (1996) Methods for Subjective Determination of Transmission Quality, August.
[6] ITU-T Recommendation Y.1541 (2006) Network Performance Objectives for IP-based Services.
[7] ITU-T Recommendation T.140 (1998) Protocol for Multimedia Application Text Conversation, February.
[8] ITU-R Recommendation ITU-R BT.500-11. (2005) Methodology for the subjective assessment of the quality of television pictures.
[9] ITU-R Recommendation ITU-R BT.601-5. (1995) Studio Encoding Parameters of Digital Television for Standard 4:3 And Wide-Screen 16:9 Aspect Ratios.
[10] ITU-R Recommendation ITU-R BT.1359-1 (1998) Relative Timing of Sound and Vision for Broadcasting, November.
[11] Digital Video Broadcasting (DVB). (2009) Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications.
[12] IETF RFC 3393 (2002) IP Packet Delay Variation Metric for IP Performance Metrics (IPPM).
[13] IETF RFC 3357 (2002) One-Way Loss Pattern Sample Metric.
[14] DSL Forum TR-126 (2006) Triple-play Services Quality of Experience (QoE) Requirements.
[15] GB/T20090.2 (2006) Information technology - Advanced coding of audio and video - Part 2: Video.
[16] SMPTE 421M-2006. Television - VC-1 Compressed Video Bitstream Format and Decoding Process.
[17] ISO/IEC 14496-3. Information technology – Coding of audio-visual objects – Part 3: Audio.
[18] FG IPTV-DOC-0114. Working Document: IPTV Services Requirements, p. 41.
[19] CableLabs® (2006) CableLabs® Headend VoD Metadata Content 2.0 Specification, May 22.
[20] ATIS (2006) A Framework for QoS Metrics and Measurements Supporting IPTV Services, December, 2006F.
[21] Quality of Experience Requirements for IPTV. (2007) ITU FG IPTV-DOC-0151.
[22] CableLabs. http://www.cablelabs.com (accessed June 9, 2010).

18

Quality of Service (QoS) Monitoring and Assurance

In order to meet QoS requirements, an IPTV service provider would have to face several challenges. Here is a list of those challenges:

1. **Network and Service Capacity Planning**: In order to deliver high-quality video, it is absolutely essential for a service provider to predict traffic growth as accurately as possible and engineer the network with enough capacity to enable congestion-free transport in order to ensure minimal packet loss, packet reordering and packet jitter. Further, the network should have enough redundancy to ensure fast failover of circuits to minimize outage in service delivery. Note that traffic engineering in MPLS network as discussed earlier will play a crucial role in meeting these challenges.

   Network capacity planning is aimed at avoiding network congestion by minimizing resource contention in the network and whenever there is contention, resolve it with proper allocation of resources using appropriate fair scheduling techniques at the network nodes. Note that these are the benefits of a "closed" network over an "open" network as the network service provider has full control on how to route traffic (traffic engineering) and how to prioritize and allocate resources (scheduling in the network) in order to meet the requirements with respect to packet loss, packet reordering and packet jitter.
2. **Network and Service Provisioning to Increase Efficiency**: In order to keep in check the growth of traffic, an IPTV service provider would have to leverage IP multicast for replication, distributed caching for serving content locally, flexible content insertion and storage at most economical points in order to avoid high cost of transmission, and do video admission control in order to prevent injection of new traffic in the network when it is congested. These techniques go hand-in-hand with network capacity planning as they help in further optimizing network usage.
3. **Video Admission Control**: In order to prevent overloading of network with video traffic especially when there is sudden surge in requests (such as, release of a new blockbuster movie, hosting of the Olympics or FIFA World Cup football tournament), it is important to

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

(dis)allow certain requests. Disallowing certain requests from being served leads to lower availability of the service but ensures proper quality of service for the requests that are currently being served by the network. Naturally, there is a tradeoff between availability and quality of service. Note that consumers are already used to the concept of requests being temporarily denied by the mobile wireless service providers when the network is overloaded. The same concept will apply even for IPTV services. However, the goal would be to minimize "blocking" and hence clever techniques, such as predictive analytics to forecast capacity requirements and proactively reserving resources in the network and/or dynamically adding resources in the network to handle additional traffic would play a critical role in ensuring best possible user experience for customers.

Usually, video on demand requests generate spikes in the network as these requests result in high-bandwidth unicast traffic as opposed to broadcast TV in which the traffic is multicast. However, techniques, such as, batching [12–14] and patching [15] can be used to combine requests in clever ways and serve them using significantly lower number of streams compared to the number of requests that need to be served.

Multicast video admission control (M-VAC) applies to broadcast channels and deals with the selection of channels that would be admitted into the network. In many cases, it is necessary to multicast an unwatched channel with the assumption that someone might switch to that channel and the content should be made available instantaneously (see the channel-changing requirement above). This is based on the fact that if the multicast group corresponding to a traffic channel is available, switching to that channel is quicker compared to starting multicast of the requested channel and then allowing the user to switch. Naturally, the prediction of popular channels will play a role here as it would make sense to proactively broadcast only the popular channels as the likelihood of end users requesting such content is higher. Given finite network capacity, certain less popular channels would have to be denied admission, leading to lower quality of user experience for less popular content.

4. **End-to-end QoE Monitoring and Assurance**: Network capacity planning, Service provisioning for increased efficiency and video admission control would certainly help in optimizing traffic in the network and providing high quality of experience (QoE) to end users. However, despite every attempt in engineering the network, there is a finite probability of congestion and contention for resources in the network leading to degraded quality of experience. The only way that can be eliminated would be constantly to monitor the video and audio quality at multiple points of the network, starting from the video headend equipment, such as MPEG encoders, and streaming servers to the residential gateway and IP set top boxes with multiple intermediate points in the delivery network, and take action when the quality degrades beyond a threshold. Network diagnostics, and reporting; network performance and fault management; MPEG 2/4 analysers and video monitors are some of the key challenges from end-to-end service management perspective.

18.1 A Representative Architecture for End-to-End QoE Assurance

Maintaining QoE on an end-to-end basis for broadcast TV as well as for on-demand video sessions is challenging [1–11]. The immediate solution of reserving per-session bandwidth to offer guaranteed quality of service is difficult for video traffic, which is very bursty and

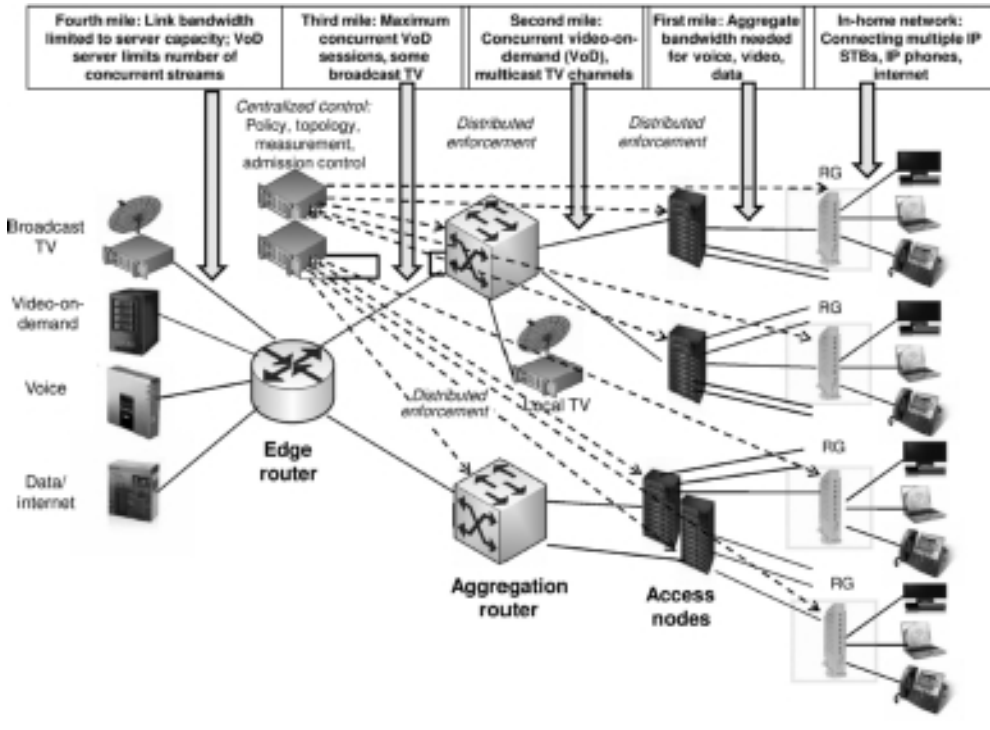


Figure 18.1 Representative end-to-end architecture for QoE assurance.

dynamic in nature. Moreover, per-session resource reservation techniques have not worked in wide area data networks despite multiple attempts. An alternative approach is to monitor bandwidth at each link of the network in a distributed manner, and enforce quality through VAC while leveraging the benefits of statistical multiplexing. The architecture schematically shown in Figure 18.1 assumes centralized control but distributed enforcement. Centralized control does not mean "physically" centralized; rather it means "logically" centralized. Conceptually, the centralized servers collect information from various elements in the network to get a near real-time view of the traffic load at each link and use that information to make decisions about enforcing admission control at the relevant entry points of video traffic into the network. In Figure 18.1 the dashed lines from the centralized servers to the various network elements (such as, routers, switches and residential gateways) show "distributed" enforcement of video admission control. Admission control, in real networks, is highly influenced by policy. Specifically, there are policy issues about which traffic gets priority and which traffic gets blocked/dropped in case of congestion in the network. These policies depend on various factors, such as, type of application, class of traffic, level of subscription and entitlements. In general, a service provider would like to maximize revenue by allowing traffic belonging to premium customers while blocking traffic from basic customers. All these policies are also expected to be logically centralized.

Having established the importance of monitoring traffic at each link and node of the network, we next take a look at a network monitoring tool specifically focused on IPTV networks.

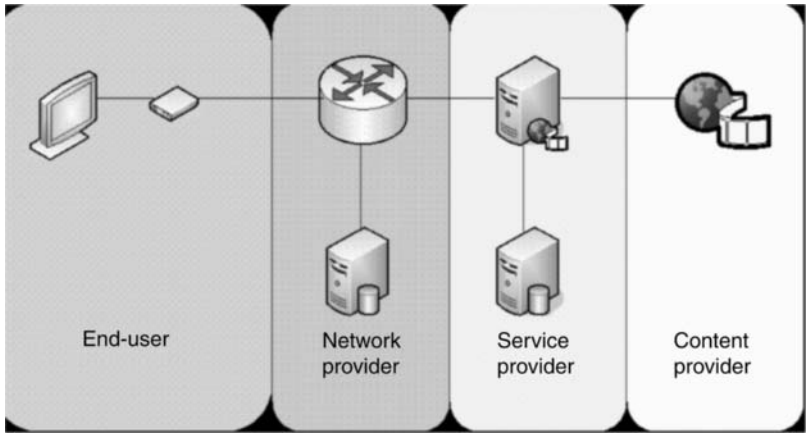


Figure 18.2 Internet protocol TV domains.

18.2 IPTV QoE Monitoring

18.2.1 Monitoring Points

IPTV delivery chain can be divided into multiple *domains* as shown in Figure 18.2.

Operators at their respective domain borders can perform monitoring. When these are taken together, it provides an end-to-end monitoring topology. Monitored performance characteristics, across a single domain or multiple domains, can be integrated with existing or new operations support systems (OSS) and/or network management systems (NMSs).

An example topology with generalized domain boundaries is shown in Figure 18.2. These domains are further divided into specific monitoring domains in Figure 18.3. Within each domain, different aspects can be monitored at each domain boundary as outlined below.

The management platform entities manage some domains and collect parameters from monitoring points, make performance analysis and generate reports.

18.2.2 Monitoring Point Definitions

Point 1 (PT1) demarcates the domain border between content provider and IPTV service provider. At this point, source video quality monitoring, source audio quality monitoring and metadata verification are most important.

Point 2 (PT2) demarcates the domain border between service provider and network provider. At this point, original streaming quality monitoring, such as audio-visual quality monitoring, IPTV service attribute monitoring and metadata verification are most important.

Point 3 (PT3) demarcates the IP core and IP edge networks where monitoring of IP-related performance parameters, such as network monitoring, network performance monitoring are most important.

Point 4 (PT4) is closest to the user where monitoring the quality of streaming, audio-visual quality and IPTV service attribute monitoring are most important.

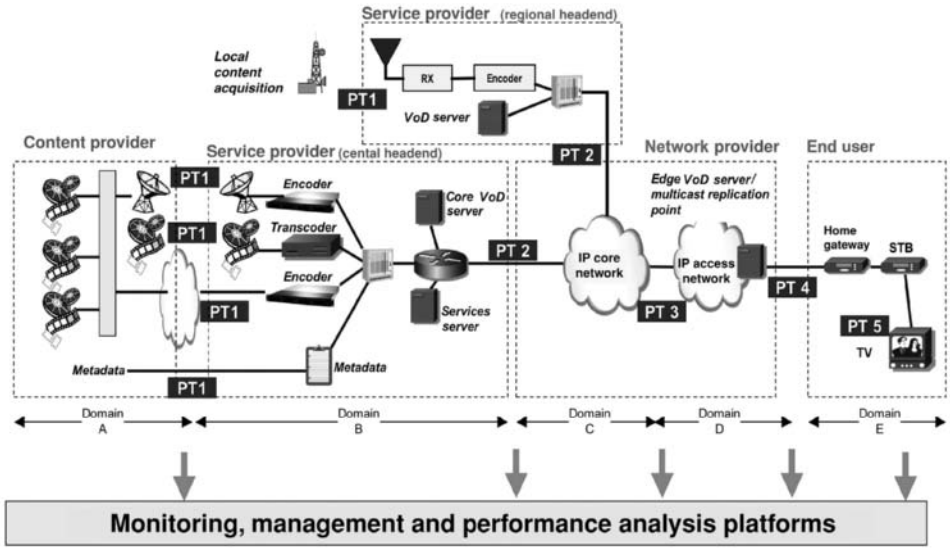


Figure 18.3 Monitoring points.

Point 5 (PT5) is at the final end point and directly relates to end-user QoE. Monitoring audiovisual quality and IPTV service attribute monitoring are most important.

18.2.3 Monitoring Parameters

Different parameters need to be monitored at different stages of an end-to-end IPTV delivery chain. With reference to Figure 18.3, which identifies monitoring points and domains, specific parameters are applicable to specific domains. However, all parameters taken together are equally significant for monitoring the true performance of IPTV delivery.

Usually in the context of an IP television service, a physical interface (e.g. Ethernet) carries one or many IP flows. Each IP flow, in turn, carries single or multiple transport streams, each of which has multiple channels/programs within. Each channel/program has multiple attributes defining the content (Figure 18.4).

With respect to the monitoring points shown in Figure 18.3, if a specific monitoring parameter is applicable, the corresponding cell in Table 18.1 is marked with "1", or else it is marked with "0".

18.2.3.1 Physical Layer Parameters

The main attributes to be monitored at the physical layer are RF signal level, signal to noise ratio (SNR), modulation error ratio (MER) and bit error rate (BER). These measurements can be used to characterize the state of the RF signal.

18.2.3.2 Network (IP) Layer Parameters

The main attributes to be monitored at the network layer are bandwidth, packet loss and jitter.

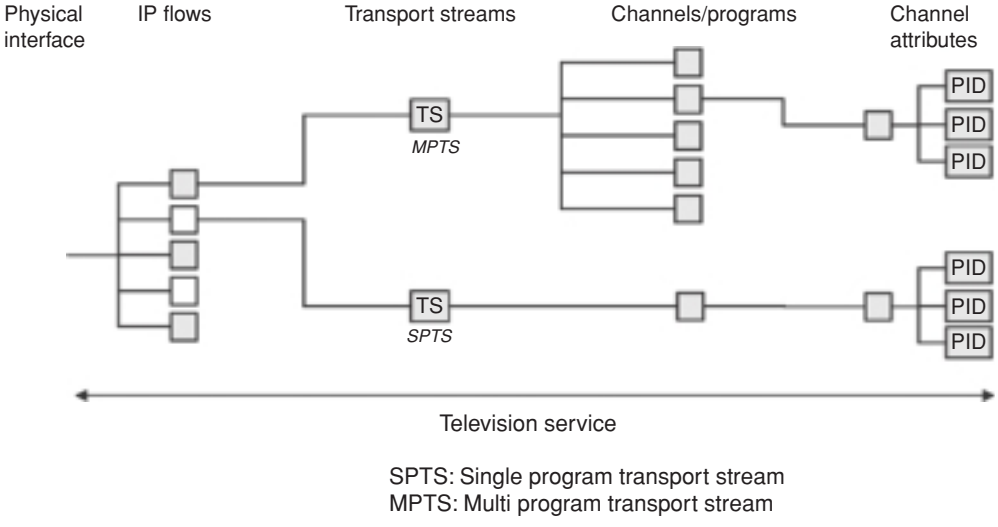


Figure 18.4 Television service hierarchy.

18.2.3.2.1 Bandwidth

Link IP Layer Used Bandwidth is defined as the sum of the IP layer bandwidth for all IP packet flows within in a link.

Link IP Layer Available Bandwidth is defined as the maximum IP layer bandwidth which the link can provide without influencing other existing flows in the link.

With the knowledge of the above two parameters, the network providers can determine the bandwidth utilization ratio of a link.

End-to-end IP Layer Bandwidth is defined as the maximum IP layer bandwidth an end-to-end path can provide given no background flows exist along that end-to-end path. It can also be understood as equal to the lowest Link IP Layer Bandwidth along that end-to-end path, and hence the bottleneck along that path.

End-to-end IP Layer Available Bandwidth is defined as the maximum IP layer bandwidth that an end-to-end path can provide without influencing other existing flows along that path.

18.2.3.2.2 Packet Loss

Network performance requirements define loss distance, loss duration, and loss rate as described in detail in an earlier section. In addition, impairments may occur in a number of burst patterns. There are three main kinds of loss patterns:

- **Sparse Bursts** are periods of high packet loss (several seconds) that begin and end with a lost (or discarded) packet and often caused by network congestion, random early drop (RED) in routers, or other related events.

Table 18.1 Monitoring points and parameters.

Monitoring Parameter	PT1	PT2	PT3	PT4	PT5
Physical Layer Parameters					
RF Integrity	1	0	0	0	0
IP Layer Parameters					
Bandwidth	0	1	1	1	1
Packet Loss	1 (only for IP)	0	1	1	1
Jitter	1 (only for IP)	0	1	1	1
IGMP Join/Leave	0	0	0	1	1
IP Flow List	1	1	1	1	1
Transport Layer Parameters					
Packet Loss	1	0	1	1	1
Jitter	1	0	1	1	1
Basic Monitoring (TR 101 290: Priority 1)	1	1	1	1	1
Periodic Monitoring (TR 101 290: Priority 2)	1	1	0	0	0
Application-Dependant Monitoring (TR 101 290: Priority 3)	1	0	0	0	0
Service Line-up Parameters					
Metadata Verification (Parental Control/EPG/Subtitles)	1	1	0	0	1
Metadata Validity	1	1	0	0	1
Metadata Integrity	1	1	0	0	1
Channel Zap time	0	0	0	0	1
Channel Line Up Verification	0	1	0	0	1
Correctness Rate	0	0	0	0	1
Connection Success Rate	0	0	0	0	1
Connection Time	0	0	0	0	1
Channel Attribute Parameters					
Video/Audio Bitrate	1	1	1	1	1
Video Quality	1	1	0	0	1
Audio Quality	1	1	0	0	1
Blackout Detection	1	1	0	0	1
Freeze Frame Detection	1	1	0	0	1
Audio/Video Loss/Presence	1	1	0	0	1
VoD Parameters					
VoD Request Performance					
VoD Request Accuracy	0	0	0	0	1
Connection Success Rate	0	1	0	1	0
Connection Time	0	1	0	1	1
Other Parameters					
AAA Success Rate	0	1	0	0	1
STB Booting Time	0	0	0	1	0

- **Continuous Bursts** are periods during which every packet is lost, and are often caused due to packetization (packing several transport packets within an IP packet), link failures within the IP network or other phenomena.
- **Isolated Lost Packets** are also possible and they are typically caused by bit errors in transmission or excessive collisions on local area networks.

Parameters of interest are:

Loss Run Length Distribution: this describes the distribution of run lengths of consecutive lost IP packets.

Let l_i, $i = 1, 2, \ldots, n - 1$ denote the number of loss bursts of length i, where n-1 is the longest loss burst. L denotes the number of total lost packets ($L > 0$). The loss run length distribution can be calculated as l_i/L, $i = 1, 2, \ldots, n - 1$.

Error-free Interval Distribution: this describes the distribution of run lengths of consecutive correctly received IP packets.

Let f_i, $i = 1, 2, \ldots, n - 1$ denote the number of error-free intervals having length i, where n-1 is the longest error-free interval. F denotes the number of total received packets ($F > 0$). The error-free interval distribution can be calculated as f_i/F, $i = 1, 2, \ldots, n - 1$.

Packet Loss Rate: this specifies the proportion of IP or RTP packets lost in the network or discarded by layer 1 or 2, or lost after correction by FEC or retransmission. This is computed by dividing the total packets lost by total packets expected for a given measurement interval.

Out of Order Packet Rate: this specifies the proportion of (IP or RTP) packets arriving out of order. This is computed by dividing total packets out of order by the total packets expected for a given measurement interval.

Burst Loss Rate: this specifies the proportion of IP or RTP packets lost within a (sparse) burst. This is computed by dividing the burst loss by burst length.

This parameter indicates the severity of transient problems affecting video quality, and hence is useful for configuration of FEC and retransmission.

Gap Loss Rate: this specifies the proportion of IP or RTP packets lost within gap periods. It is computed by dividing the gap loss by total packets expected in the gap period.

Mean Gap Length: this indicates the mean length of gaps between bursts of IP or RTP packets. This indicates loss conditions during good quality periods.

Mean RTP Burst Length: this indicates the mean length of (sparse) burst periods or IP or RTP packets, which shows the severity of transient problems affecting video quality.

Loss Period Count: this is the count of the number of times the loss period exceeded the Loss Period Threshold. This indicates the frequency of very severe loss conditions where the video screen will go blank.

IP Maximum Loss Period: this is the maximum number of IP packets in an excess loss event (max hole size). This metric is used for the watching of a program. It will report the maximum time a screen will freeze or go blank. This parameter is useful for setting error correction or retransmission parameters

Retransmissions: this refers to the number of retransmitted RTP/UDP packets in a Measurement Interval. This metric indicates error conditions and bandwidth usage.

18.2.3.2.3 Jitter

The parameter of interest is: streaming jitter (SJ) which represents the maximum and minimal bit-rate of the streaming server output. Streaming jitter is from the network's viewpoint and it is a vital metric when monitoring the performance.

18.2.3.3 Transport Stream Parameters

In broadcast environments, MPEG-2 transport stream protocol is used and as a result, the exchange of audio/video services with associated system information tables is facilitated between compatible equipment. If the transport stream packet headers are not encrypted, monitoring of transport stream parameters is possible. However, some encryption schemes might scramble all data after the IP header. In such cases, no monitoring of transport stream parameters is possible. Only IP related statistics would be relevant. The transport stream parameters are categorized according to severity as:

- Priority 1: basic monitoring necessary for decodability.
- Priority 2: periodic monitoring.
- Priority 3: application-dependent monitoring.

18.2.3.4 Service Line-Up Parameters

The main parameters that need to be monitored in this case are:

18.2.3.4.1 Channel Line Up

This refers to the parameters applicable to the channels/programs offered by the IPTV service provider. Monitoring these parameters is essential as it details the presence and correctness of channels intended to be delivered over the service. Some important parameters are:

- number of channels on the service;
- channel identity (ID) number;
- the names of the services;
- source of content for the services (content provider/aggregator);
- number of interactive channels;
- number of free to air/pay channels;
- genre of channel content and number in each genre.

18.2.3.4.2 Service Metadata

Service metadata carries additional information aside from the video, audio, and subtitles comprising the main components of the television service. Incorrect or nonexistent metadata might severely affect overall service quality or result in legal liabilities for the operator. Some important parameters are:

- parental or age rating;
- source provider of metadata;
- language sets correctness;
- correlation of EPG with actual content;
- correctness of subtitling;
- size of metadata;
- availability of metadata.

18.2.3.4.3 Metadata Validity

Metadata validity is for checking whether the metadata files from content providers comply with the syntax and semantics as per the specification. For example, unless the IPTV metadata file uses XML, it is flagged as an error.

18.2.3.4.4 Metadata Integrity

Metadata integrity is for checking whether the metadata files from content providers contain all the necessary information and comply with the metadata specification. For example, when a metadata file of a movie for VOD purpose is received, all the attributes about director, actor, and so on, need to be checked.

18.2.3.4.5 Channel Zap Time

Channel zap time (measured in ms) is defined as the time it takes for the channel to change from the moment the subscriber presses the button on the remote.

18.2.3.4.6 Correctness Rate

Correctness rate is defined as the number of times the "correct" content expected by the user is played as a percentage of the total number of requests attempted by users in a pre-defined time interval.

18.2.3.4.7 Connection Success Rate

Connection Success Rate is defined as the number of connections which were successfully established with the streaming servers as a percentage of the total number of attempted connections in a unit of time.

18.2.3.4.8 Connection Time (CT)

Connect time is defined as the amount of time elapsed between the initial request by the media player or STB and the start of buffering. Connect time has many components:

- Domain name system (DNS) lookup and resolution;
- metafile actions;
- RTSP handshakes; and
- transport of the first byte of data to the player.

18.2.3.5 Channel Attribute Parameters

Specific attributes of each channel are embedded as part of the transport stream carrying the channels. The program ID (PID) defines the identity of the program being carried by the transport stream (TS).

18.2.3.5.1 Channel Attributes

- Program ID;
- type: audio/video/data;
- aspect ratio;
- conditional access enabled/disabled;
- age rating;
- encoding format of the channel;
- bit rate of the channel;
- bit rate of each component of the channel in case it carries more than one type of information (e.g. audio and video).

18.2.3.5.2 Video Quality

Quality of video can be assessed either subjectively or objectively. User perception of video quality can only be approximated by using objective estimation models and represented as a mean opinion score (MOS).

18.2.3.5.3 Audio Quality

Audio quality can be assessed using either subjective assessment, or objective assessment. The user perception of video quality can only be approximated by using objective estimation models and represented as a MOS.

18.2.3.5.4 Presence and Status of Audio/Video Output

Presence indicates whether any audio/video content is produced. Status indicates whether the video has freeze frames or is in the blackout condition.

18.2.3.6 Video-on-Demand (VoD) and Other Parameters

The parameters of interest are as follows.

18.2.3.6.1 Video-on-Demand (VoD) Request Performance

Performance of on-demand content request is measured as the time elapsed between the user pressing the remote control button to request content to the time the video first arrives on the screen.

18.2.3.6.2 AAA Success Rate

User experience depends on the success rate of requests from end users, and hence authentication, authorization and accounting (AAA) success rate needs to be monitored.

18.2.3.6.3 Set-Top-Box Booting Time

The set-top box (STB) turns on when the power button is clicked on by the user. Its boot up time is defined as the time it takes for the it to start up and be ready for service consumption.

18.2.4 Monitoring Methods

Network performance monitoring largely belongs to three categories:

- active monitoring;
- passive monitoring; and
- hybrid (active and passive) monitoring.

Active Monitoring: in this method, a test device injects test packets into the network, and some other device measures test packets at some other points within the network. While this method increases traffic load within the network, it provides control of traffic generation based on variant scenario.

Passive Monitoring: in this method, a test device just observes characteristics of packets on a network link. The observed characteristics can be used for flow analysis. While this method does not generate any extra traffic, thereby measuring real network status, there are limitations in observing characteristics of all packets and in estimating trouble scenarios.

Hybrid Monitoring: in this method, an active test device injects some probe packets into the network and a passive test device, on recognizing the active probe packets, measures network level parameters, such as delay and jitter from the customer's terminal to the passive measurement point. The main advantage of this method, compared to active monitoring, is significantly less number of active probes in the middle of the managed networks.

18.2.5 Multi-Layer Monitoring

18.2.5.1 Physical-Layer Monitoring

Demodulation of RF signals involves: (i) tuning and (ii) demodulation. The RF signal level is extracted from the tuning stage, while the remaining measurements are taken from the demodulator module. signal-to-noise ratio (SNR) and the modulation error ratio (MER) can be derived from the constellation of the RF signal. Both SNR and bit error rate (BER) are measured in dB. Bit error rate can be derived from the forward error correction stage. Depending on the RF modulation, different forward error correction schemes are used.

18.2.5.2 Network-Layer Monitoring

Network resource and operational status monitoring including link and bandwidth characteristics, flow direction, inter-node time delay, jitter and packet loss rate are expected to be performed during network operation. When a monitored network parameter exceeds a threshold value defined by the operator the network operator needs to be notified.

18.2.5.3 Application-Layer Monitoring

There are several things that need to be monitored at the application layer including IPTV service attributes, channel line-up, service metadata and channel zap time as described below.

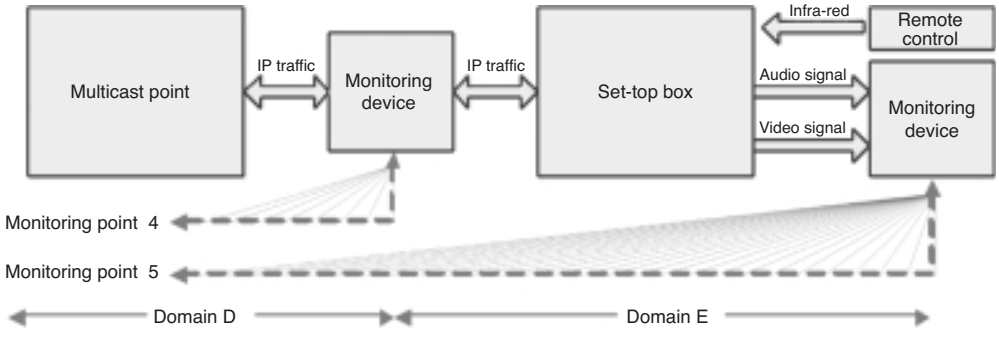


Figure 18.5 Direct observation monitoring model for domain E.

18.2.5.3.1 IPTV Service Attribute Monitoring

A direct observation approach to monitoring via the use of a dedicated measurement devices or functionality built into a device deployed at appropriate monitoring points in the IPTV delivery network presents a true metric of service performance. Figure 18.5 shows a deployment scenario that involves monitoring the direct user experience before and after the STB. This setup is applicable for Domain E (Figure 18.5) and monitoring points 4 and 5.

18.2.5.3.2 Channel Line Up Validation

Every service plan comes with a basic set of channels. Validating all the channels in each service requires taking into account the various service plans and the region specific channels.

18.2.5.3.2.1 Channel Line-Up Validation for Domain B

Channel line-up validation at domain B involves checking the various unicast and multicast IP flows prior to delivery from the head end. The unicast and multicast addresses of all the outgoing IP flows are checked and correlated with the channel line up. This ensures correct configuration of the IP flows.

18.2.5.3.2.2 Channel Line-Up Validation for Domain E

Channel line-up validation at domain E involves examining the IP traffic entering the STB as well as the audio/video output of the STB. Set-top boxes' incoming IP traffic contains either the unicast or multicast address. Each broadcast channel is typically assigned a multicast address. When a remote control is used to invoke channel change on the STB, a correlation needs to be made between the remote control's channel-change parameter and the multicast address of the IP traffic. This test also ascertains whether an STB has received the correct channel multicast address-mapping information. The presence and status of the audio/video output of the STB should be checked whenever a new channel change command is issued by the remote control. Presence indicates whether any audio/video content is produced. Status indicates whether the video is frozen or is in the blackout condition. In the case of audio, silence and tone should be checked. This test is applicable to all the available audio/video outputs of the STB.

18.2.5.3.3 Service Metadata Validation

Metadata is typically delivered via tables in the transport stream, or by other means, comprising of XML tables, HTTP traffic, encrypted data and out-of-band channel delivery.

18.2.5.3.3.1 Service Metadata Validation at Domains A and B

Metadata that are generated and subsequently embedded in the IP flows or delivered by other means should be decoded and validated against the operator's service offering. Multiple metadata sets can be checked simultaneously.

18.2.5.3.3.2 Service Metadata Validation at Domain E

Metadata is usually presented to the user in the form of an electronic program guide. The STB's incoming IP traffic should be decoded appropriately at monitoring point 4, depending on the method of delivering the metadata. This metadata should be checked against the operator's service offering. Only one metadata set can be checked at any one time.

18.2.5.3.4 Channel Zap Time

Channel zap time can be measured by taking the time difference between the channel change commands from the remote control and the time the new channel's video and audio are presented on the display. A device to which the audio/video outputs of the STB are connected can be used to detect the presence of the new channel's video and audio content.

18.2.6 Video Quality Monitoring

Video quality assessment can be carried out in two ways:

- Subjective assessment.
- Objective assessment.

18.2.6.1 Subjective Quality Monitoring

There are many subjective video quality measurement methods, the output of which is an average of the MOS quality ratings, usually given on a five-grade impairment scale:

5 – imperceptible.

4 – perceptible, but not annoying.

3 – slightly annoying.

2 – annoying.

1 – very annoying.

The grading of video quality can also be expressed as a percentage from 0 to 100, with 0 indicating no distortion. The higher the value, the greater is the distortion.

18.2.6.2 Objective Quality Monitoring

Many advances have been made in objective video quality-monitoring techniques. Although not as accurate as subjective quality measurement, objective assessment can provide quick support to fine-tuning network variables. Since subjective video monitoring is complex and time consuming there has been significant development of perceptual video quality measurement (PVQM) algorithms in recent years in an attempt to replicate the scores given by subjects in an objective tool. Objective quality monitoring can be classified into following categories:

- Techniques based on models of human video perception.
- Techniques based on video signal parameters.

Mean square error (MSE) and mean absolute error (MAE) calculate the frame difference between corresponding streams. The difference can also be assessed by the peak signal-to-noise ratio (PSNR), which is the log of the ratio of the peak signal squared to the MSE.

There are two different models for objective quality monitoring, namely, packet layer model and bit-stream layer model.

18.2.6.2.1 Packet-Layer Model

The packet-layer model estimates video quality using only IP-layer information. Since it does not include video-related payload information, its estimation of video quality is limited compared to other models, such as, the bit stream layer model (described next) which has more video-specific information. Naturally, its computational load is very light as processing is limited.

18.2.6.2.2 Bit Stream Layer Model

It is possible that an objective model may have access to bit stream data from which the model can obtain additional information on transmission errors (for example, delay, packet loss), codec parameters (for example, type, bit rates, frame rates, codec parameters), and so on. This kind of information is easily available from bit-stream data at the receiver. It is expected that such models may provide improved performance in terms of accuracy and speed compared to objective video quality models which use only processed video sequences.

The bit stream layer model can estimate the quality of each video sequence using IP information that includes video-related payload information, as shown in Figure 18.6. That is, it can take into account the effects of video contents.

In order to improve the accuracy of objective models, it is also possible to transmit video quality scores of the compressed video data which are transmitted. Periodic video quality measurements should be made such that sufficient information is available on the video quality of the processed video quality sequence. If there are no transmission errors, the video quality at the receiver would be the same as that of the transmitted video sequence. If transmission errors occur, the received video sequence suffers from both compression impairments and transmission error impairments. With video quality scores available, an objective model which measures the video quality of the received video sequence may be improved. The video quality scores can be transmitted as metadata.

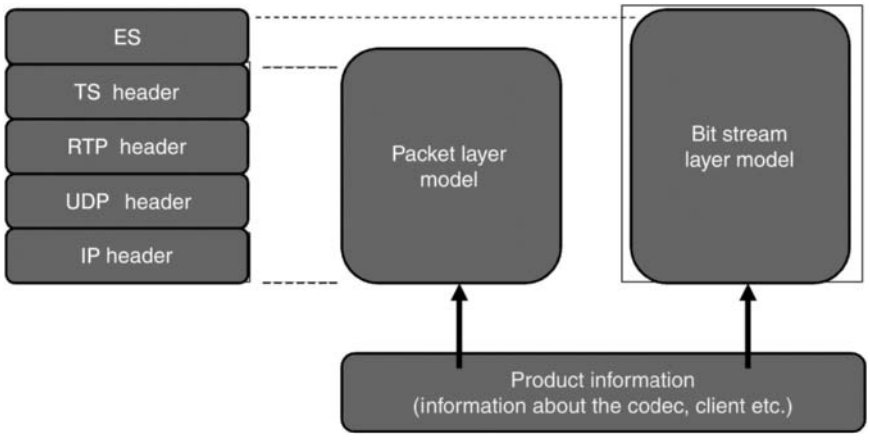


Figure 18.6 Relationship between packet-layer model and bit stream layer model Hybrid perceptual/bit-stream models.

18.2.6.3 Back Channel Requirements

Internet protocol TV terminals may have video-quality evaluation functions and reporting functions for the video quality scores to IPTV servers. This helps in gathering and analyzing video quality reports from multiple IPTV terminals.

Quality measurements can be made at various measuring points including a number of reference points and terminals and can be used for various purposes, such as, for quality-based billing, quality-based codec parameter optimization at streaming servers, or for network management. Depending on applications, the quality measurements must be computed in a given unit and the quality scores sent to the corresponding destination. If the quality scores are used for quality-based billing, the measuring unit should send the information to the service provider (Figure 18.7) and if they are used for quality-based codec parameter optimization, the

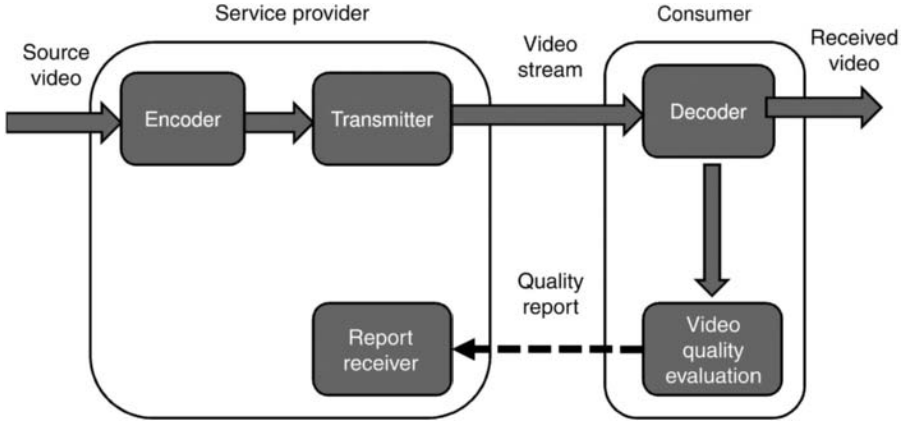


Figure 18.7 Back channel for quality reporting.

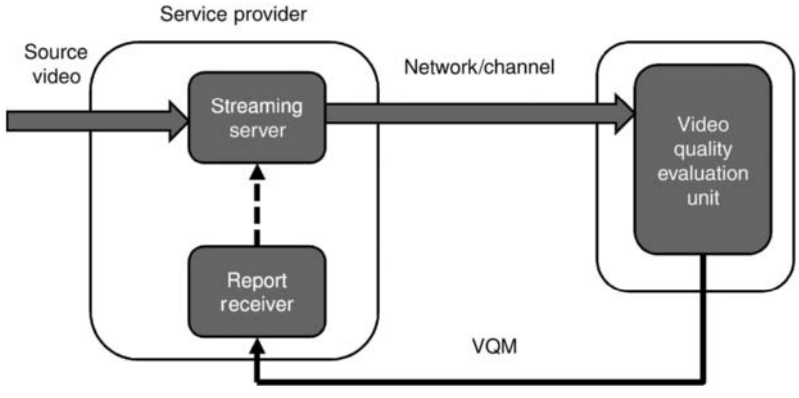


Figure 18.8 A streaming server that optimizes codec parameters based on the perceptual quality.

measuring unit needs to send the information to a streaming server (see Figure 18.8). On the other hand, if the quality scores are used for network management, the measurement results need to be sent to the network provider (see Figure 18.9).

18.2.7 Audio Quality Monitoring

Audio quality monitoring can be carried out in two ways:

- Subjective assessment.
- Objective assessment.

18.2.7.1 Subjective Quality Monitoring

There are a number of subjective audio quality measurement methods, the output of which is often an average of the MOS quality ratings. Five-grade scales may be used for the subjective

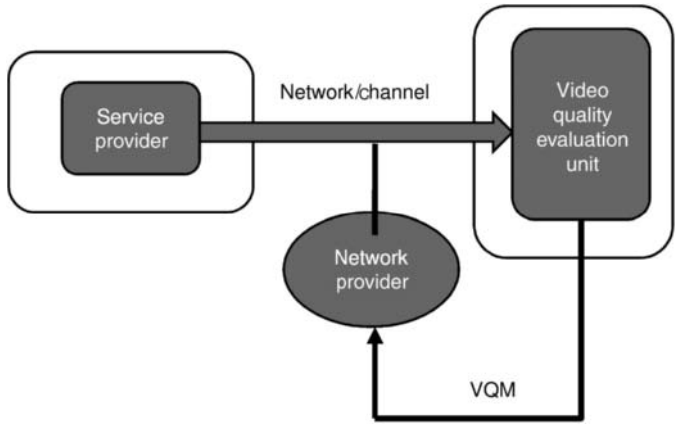


Figure 18.9 A network provider that uses perceptual quality information.

Table 18.2 Audio quality and impairment scales.

Quality		Impairment	
5	Excellent	5	Imperceptible
4	Good	4	Perceptible, but not annoying
3	Fair	3	Slightly annoying
2	Poor	2	Annoying
1	Bad	1	Very annoying

assessment of sound quality or impairment as shown in Table 18.2. Depending on the intended purpose, different methods need to be used for the subjective assessment of sound quality and of the performance of audio systems.

18.2.7.2 Objective Quality Monitoring

Objective measurement methods need to be developed for estimating the audio quality because subjective techniques are time-consuming and expensive. Techniques, such as, perceptual estimation audio quality (PEAQ) and perceptual estimation of speech quality (PESQ) are commonly used for making objective perceptual measurements of perceived audio quality. However, it should be kept in mind that objective measurement methods, in general, are not a substitute for formal subjective listening tests.

18.3 Internet Protocol TV QoE Monitoring Tools

There are several products in the market that focus on IPTV QoE monitoring. However, we focus on a specific tool from a vendor called Ineoquest [6] to understand what kind of monitoring support is available from a commercial tool.

18.3.1 IQ Pinpoint – Multidimensional Video Quality Management

The IQ Pinpoint solution [7] is a mix of hardware probes and analysers located at intelligent points across the end-to-end network of IPTV services. The iVMS software product, a video management software, aggregates data received in real-time from these hardware probes and analysers to deliver a business-driven, multifaceted, proactive network-monitoring solution. The uniqueness of the solution is captured in the Veriframe enabled last-mile technology (Figure 18.10).

The Veriframe enabled technology uses a family of "Slingbox"-type hardware devices across the network that can work at an:

- individual subscriber STB point level;
- service group level; and
- video head end level.

The industry differentiator of the IQ Pinpoint solution is that it can aggregate massive amounts of data into a single, five-nines (99.999%) measurement that gives IPTV providers an instant QoS gage of service delivery. By enabling five-nines metric on a per-channel basis, service

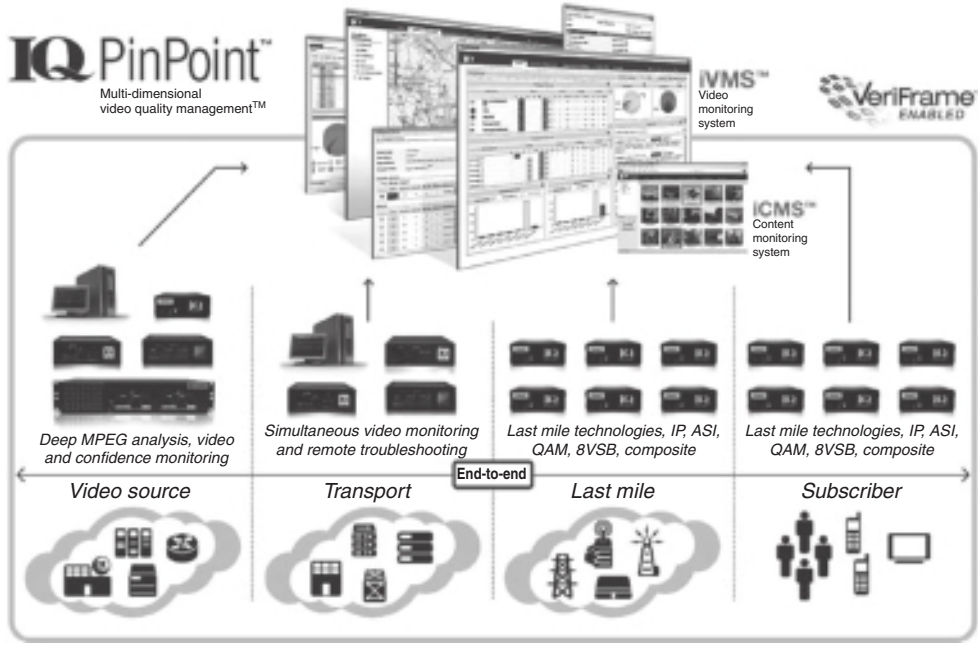


Figure 18.10 Multidimensional video quality management.

providers can fine tune their network and operations to achieve total control over the service(s) they are providing, rather than basing their performance on the number of calls to their customer service center.

18.3.1.1 IQPinPoint Applications

IQPinPoint solution can be used for:

- Rapidly isolating faults by proactive monitoring of all video flows at all locations at all times.
- Correlating Video QoS monitoring with actual subscriber experience.

18.3.1.2 IQPinPoint Benefits

IQPinPoint solution provides a business solution that can audit, monitor, analyze, and troubleshoot from a multidimensional approach.

Specifically, it provides:

- Per-channel five-nines program availability reporting.
- A scalable solution that perfectly matches the bandwidth required at the appropriate places in the network.
- The broadest physical transport coverage: ASI, Ethernet, QAM, 8-VSB and more.
- complete end-to-end QoS and QoE solution when integrated with IQ *VeriFrame*™ technology.

Figure 18.11 iVMS video management system.

18.3.1.3 IQPinPoint Supported Software: iVMS™ Video Management System

- iVMS (Figure 18.11) provides a complete view into the health of the digital video distribution system.
- Proactive five-nines (99.999%) program availability monitoring and analysis of all content simultaneously and continuously.
- Rapid fault isolation.
- An intelligent single management dashboard.
- Real-time alerts and executive reports.
- Industry standard measurements including MDI.
- Proactive monitoring of all content simultaneously.
- Intuitive Web client.
- Integrated with VeriFrame technology enabled iCMS product.

18.3.1.4 IQPinPoint Supported Probes

There are several probes available depending on the position of the network where the information needs to be tapped. Here are some examples:

- Geminus family (dual 1 GigE or 10 GigE modules; Up to 20GB of IP video generation).
- IQMediaMonitor (Copper 10/100/1000 and fiber port; 256 simultaneous TS with 1500 PID, 80MB capture).

- Singulus G1-T (Copper 10/100/1000Mb and fiber port, 256+ simultaneous flows, 1500 PIDs, 80MB capture, 2 × GE generation).
- Singulus G10 (10Gb line-rate traffic generation; 10 GigE core monitoring analysis and test, 1000+ simultaneous TS, 128 000 PIDs, 200MB capture, 10GE generation).
- Cricket family (monitoring from headend to home, subscriber feedback controls, remote fault isolation probe, up to 10 flows simultaneous monitoring; ASI, IP, 8-VSB, QAM).

18.3.2 Headend Confidence Monitoring

The headend confidence monitoring solution can monitor hundreds of channels which can automatically test content integrity 24/7 and generate alerts (via email and SNMP) for channels that do not meet specified limits.

IPTV operators simply connect a feed's composite video and audio output into the Cricket FrameGrabber (Figure 18.12) hardware probe which can:

- tune to any channel;
- scan all the channels and monitor the video and audio for any loss; and
- auto-detect black screen, freeze frame, and other impairments and immediately alert the operator through e-mail or SNMP via the iCMS content monitoring system.

The Cricket FrameGrabber captures multiple thumbnails every second and uploads them to the iCMS content management system. The iCMS provides a mosaic view, along with other views, of all video thumbnails from the Cricket FrameGrabbers in the network. This provides a cost-effective centralized monitoring solution of the headend video input and output points. Via a simple Web browser, the operator can visually inspect the mosaic thumbnail view to validate the content at the headend location. The iCMS stores all the thumbnails in a database for reporting and trending applications

The headend confidence monitoring system:

- Provides a Mosaic thumbnail view of multiple sources for immediate visual verification.
- Includes long-term data storage and video thumbnail archiving for reporting and trending.

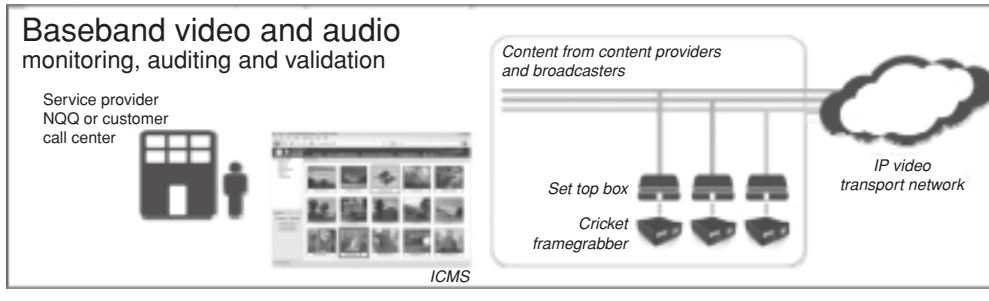


Figure 18.12 Cricket FrameGrabber.

- Automatically detects video and audio signal loss, black screen, freeze frame, and alerts the operator via SNMP or e-mail.
- Automatically monitors the entire channel lineup.
- Provides for the identification, evaluation and failure analysis of video source errors.
- Generates alerts when faults are detected and confirmed.
- Logs and allows review of faults via the iCMS or Web interface.

18.3.3 Field Analysis and Troubleshooting

IneoQuest offers solutions for ensuring quality of video over IP, by supporting multiple video and IP formats. There are portable systems that can be quickly implemented to reliably monitor, troubleshoot and field-test network traffic. The portable troubleshooting solutions range from 10 GbE for core network to 1 GbE for metro/hub/CO to 10/100 for customer premises. A set of software applications that provide live and deferred time IP and video analysis work over these portable systems.

The field analysis and troubleshooting solution can monitor and analyze all video streams simultaneously on a fully loaded gigabit ethernet network. This capability provides network operators the visibility they need into live video traffic in real time. Automatic notification means that streaming issues can be resolved before they reach the customer.

18.3.4 Product Lifecycle Test and Measurement

These are IPTV/IP video measurement solutions that test devices with hundreds of live IPTV flows, simultaneously analyzed with high-volume/live-video ASIC technology.

The product lifecycle test and measurement solution can be used for:

- Transport stream analysis and generation.
- Remote debugging.
- Video network design, emulation and deployment.
- Real-time network analysis and fault isolation.
- Real-time 10 GigE solution.

18.4 Summary

In order to meet the QoS requirements, an IPTV service provider would have to face several challenges [1–11]. First, it is absolutely essential for a service provider to predict traffic growth as accurately as possible and engineer the network with enough capacity to enable congestion-free transport in order to ensure minimal packet loss, packet reordering and packet jitter. Further, the network should have enough redundancy to ensure fast failover of circuits to minimize outage in service delivery. Second, an IPTV service providers would have to leverage IP multicast for replication, distributed caching for serving content locally, flexible content insertion and storage at most economical points in order to avoid high cost of transmission, and carry out video admission control in order to prevent injection of new traffic in the network when it is congested. Third, in order to prevent overloading of network with video traffic, especially when there is sudden surge in requests, it is important to (dis)allow certain requests. Disallowing certain requests from being served leads to lower availability of the service, but ensure proper quality of service for the requests that are currently being served by the network.

The same concept will apply even for IPTV services. However, the goal would be to minimize "blocking" and hence clever techniques, such as predictive analytics to forecast capacity requirements and proactively reserving resources in the network and/or dynamically adding resources in the network to handle additional traffic, would play a critical role in ensuring the best possible user experience for customers. Finally, despite every attempt in engineering the network, there is a finite probability of congestion and contention for resources in the network leading to degraded quality of experience. The only way that this can be eliminated would be to constantly monitor the video and audio quality at multiple points of the network, starting from the video headend equipment, such as MPEG encoders, and streaming servers to the residential gateway and IP STBs, with multiple intermediate points in the delivery network, and take action when the quality degrades beyond a threshold. Network diagnostics and reporting, network performance and fault management, MPEG 2/4 analysers and video monitors are some of the key challenges from end-to-end service management perspective.

A representative architecture for end-to-end QoE assurance was presented to highlight the components of such a system. Then IPTV monitoring was discussed with its various domains and monitoring points. Operators at their respective domain borders can perform monitoring. When these are taken together, it provides an end-to-end monitoring topology. Monitored performance characteristics, across a single domain or multiple domains, can be integrated with existing or new operations support systems (OSS) and/or network management system (NMS) systems.

Quality of experience monitoring for IPTV can be done using "active" monitoring in which a test device injects test packets into the network and some other device measures test packets at some other points within the network. While this method increases traffic load within the network, it provides control of traffic generation based on variant scenario. Passive monitoring can also be used in which a test device just observes characteristics of packets on a network link. The observed characteristics can be used for flow analysis. While this method does not generate any extra traffic, thereby measuring real network status, there are limitations in observing characteristics of all packets and in estimating trouble scenarios. Finally, both the above-mentioned monitoring techniques can be combined with the goal of eliminating their disadvantages and building on their advantages. This led to "hybrid" monitoring, in which case an active test device injects some probe packets into the network and a passive test device, on recognizing the active probe packets, measure network level parameters, such as delay and jitter from the customer's terminal to the passive measurement point. The main advantage of this method, compared to active monitoring, is significantly less number of active probes in the middle of the managed networks.

Both the video and audio qualities can be measured with subjective and objective methods. There are several products in the market that focus on IPTV QoE monitoring. However, we focused on a specific tool from a vendor called Ineoquest to understand what kind of monitoring support is available from a commercial tool.

References

[1] O'Driscoll, G. (2008) *Next Generation IPTV Services and Technologies*, John Wiley & Sons, Ltd. ISBN: 978-0-470-16372-6, January.

[2] Zapater, M.N. and Bressan, G.E. A proposed approach for quality of experience assurance of IPTV. ISBN: 0-7695-2760-4.

[3] Video QoS Measurement for IPTV Networks. http://www.castify.net/white_papers/pdf/qos_measurement_whitepaper_cbn.pdf (accessed June 9, 2010).

[4] Winkler, S. (2005) *Digital Video Quality - Vision Models and Metrics*, John Wiley & Sons, Ltd.
[5] Wolf, S. and Pinso, M. (2002) Video quality measurement techniques NTIA Report 02-392, June.
[6] Extending Video Quality Management and QoS Assurance from Source to Subscriber. http://www.ineoquest.com/video_monitoring-QoS_assurance-Cricket_Family (accessed June 9, 2010).
[7] Multi-Dimensional Video Quality Management. http://www.ineoquest.com/iqpinpoint (accessed June 9, 2010).
[8] Chernock, R. (2008) Stream Quality Assurance for IPTV. IEEE BTS Tutorial at IBC'2008. http://www.ieee.org/organizations/society/bt/08iptv4.pdf (accessed June 9, 2010).
[9] Agama Technologies. http://www.agama.tv/ (accessed June 9, 2010).
[10] Measuring IPTV QoS performance at the box. http://www.networksystemsdesignline.com/howto/ipnetworking/180206240 (accessed June 9, 2010).
[11] The Next Generation QoS Assurance. http://www.atsweb.it/fileadmin/user_upload/Brochure_Mutech/QX ManagerQoS.pdf (accessed June 9, 2010).
[12] *On Optimal Batching Policies for Video-on-Demand Storage Servers*. Proceedings of the International Conference Multimedia Systems'96, June 1996.
[13] Dan, A., Sitaram, D. and Shahabuddin, P. (1994) *Scheduling Policies for an On-demand Video Server with Batching*. Proceedings of the ACM Multimedia Conference, Oct. 1994.
[14] Sitaram, D. (1996) Dynamic batching policies for an on-demand video server. *Multimedia Syst.*, **4**(3), 51–58.
[15] Hua, K.A., Cai, Y. and Sheu, S. (1998) *Patching: A Multicast Technique for True Video-on-Demand Services*. Proceedings of the ACM Multimedia Conference, Bristol, U.K., Sept. 1998.

19

Security of Video in Converged Networks

Typically video content is produced in a studio [40]. There are several stages involved in the creation of video content. The process typically starts with a cameraman shooting a scene using a video camera/camcorder. The video is then stored and edited in a computer using editing software and packaged into a final product owned by a media and entertainment company, which becomes the owner of that video content. The final video is then licensed to broadcasters who archive (store) the content before distributing it in real time using the broadcast network of a service provider. Another route for distributing video is to distribute it in non-real-time by storing the content in a CD/DVD and distributing it through a retail channel. Video is finally consumed using an end-user device, such as a TV, PC or mobile. Thus, video content, during its lifecycle, undergoes creation, transformation, storage, broadcast/distribution and consumption. There are different entities involved in various stages of its lifecycle as shown in Figure 19.1.

The infrastructure required to create and deliver video to consumers can be categorized into: (i) video production system, (ii) video distribution system, (iii) gateway to consumer and (iv) video consumer system.

A video production system usually involves capturing the raw content using a digital video camera followed by processing and multiple rounds of editing. The video distribution system has two aspects: (i) broadcast networks, such as ABC, NBC, CBS, Fox, Star, BBC, Sony and others who own the rights to distribute the content and (ii) distribution network infrastructure providers, such as cable operators, satellite TV service providers, telecom service providers or distribution network for the recorded media, who provide the infrastructure to distribute the content. Broadcast networks make the content available to the end users under the terms of the contract they enter into with the content owners. Service providers such as cable operators distribute the content from broadcasters to the consumers through their networks. Prerecorded music and video CDs and DVDs are distributed through the traditional distributions networks and made available to the end customers through retail stores. Gateway to Consumer could be any system, such as a set-top box, which is used as a gateway system at the end customer premises. A *video consumer system* is the equipment where video is actually displayed. The

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

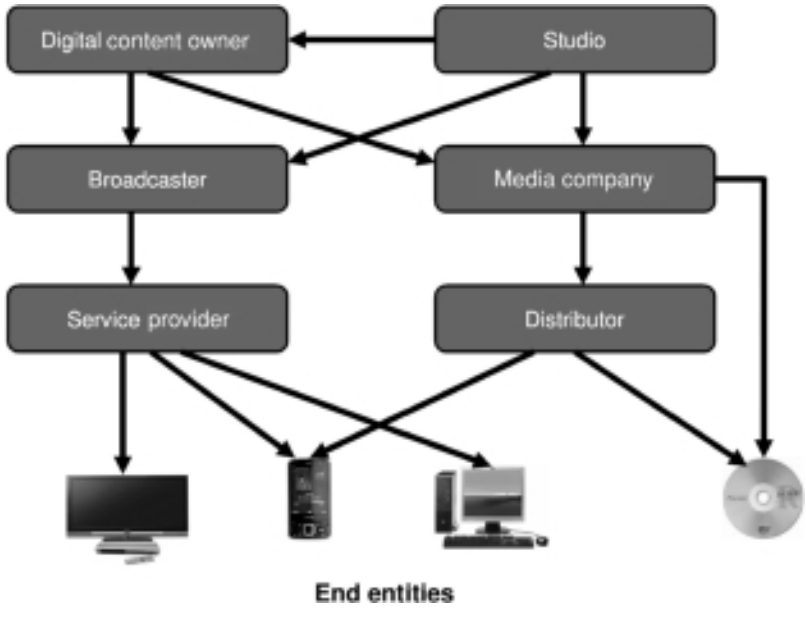


Figure 19.1 Entities involved in various stages of digital video lifecycle.

system could be a high-definition TV (HDTV), mobile phone, a computer, media player and so on. Depending upon the type of content, one or more of the above systems are involved in the entire lifecycle as shown in Figure 19.1.

For comprehensive protection, the content needs to be protected at all stages and any violation at any stage should be traceable to the user and/or device responsible for the violation. The protection mechanism should not be biased towards any party, whether it is the owner or the consumer and should address the concerns of all relevant parties. The comprehensive security system should not assume that threats to digital content exist only during its transmission or distribution. Rather the threats exist at every stage of the content lifecycle from production to consumption.

19.1 Threats to Digital Video Content

Among many threats to digital video content, the following are the most important:

- Unauthorized access to listen, view, modify, and copy.
- Unauthorized distribution of content.
- Denial of access to legitimate users by tampering with the associated security controls.
- Removal of security controls to make it available freely over the network or by copying to CDs and DVDs, destroying the economic potential of the content.

To protect digital video content from such threats it is essential to build a comprehensive security system that can provide confidentiality, integrity, and availability, and protect copyright

through all stages of its lifecycle. To achieve an effective protection, security mechanisms need to be enforced right from the time of capture /creation extending through all stages of its lifecycle.

19.2 Existing Video Content Protection Technologies

Content, whether digital or in any other form, is a valuable asset with the rights held by an individual or an organization. The value and returns from the asset depend on how well it is protected. The annual worldwide loss due to piracy of copyrighted material, excluding Internet piracy, is estimated to be over $20 billion. Traditionally, video content in the broadcast industry is transmitted over closed networks, such as, cable or satellite network. With video distribution over IP networks becoming a reality, risks associated with the content have increased significantly due to the inherent vulnerabilities associated with the TCP/IP networks [2].

Currently, the best form of protection provided to content is by digital rights management (DRM) in combination with other technologies such as, digital watermarking and encryption [20, 21, 33]. However, DRM suffers from consumer acceptance and legal limitations because of the interoperability issues, limited playability and channels of availability, and excessively restrictive copying rights. Sometimes DRMs do not permit even legitimate copying of the content for the purpose of backup or the use of newer technology to view the content.

19.2.1 DRM Systems

Digital rights management, also referred to as digital restrictions management, is a system used to manage the usage of digital content by enforcing proper usage with the help of some policies, tools and techniques [34, 35]. Its functionalities include:

- packaging content for easy distribution and tracking;
- protecting content during transmission;
- specifying suitable rights for authorized use and modes of consumption;
- facilitating protected distribution of content using CDs, DVDs and over networks;
- providing mechanisms to authenticate proper content, devices and users;
- tracking content usage and secure payment; and
- managing security and privacy issues of consumers.

In addition to the digital content, DRM systems also maintain *metadata*. Metadata of a content include

- author information, date of creation, data format, and so on, which are referred to as *content-descriptive metadata;*
- type of content, keywords, and so forth, pertaining to actual content, which are referred to as *content-dependent metadata*.

Digital rights management systems associate rights with the content, specifying the manner in which the content can be used by the consumer or device. Syntax and semantics of the

rights are expressed using rights expression languages (REL) such as, Open Digital Rights Language (ODRL) [15], eXtensible Rights Markup Language (XrML) [38], and so on. Content management information (CMI), which is carried with the content, indicates conditions and requirements regarding the use of content. It may include copy control information *(CCI)*, which indicates constraints specific to making copies of the content. Protection of the content is achieved by enforcing *compliant devices* with respect to the CMI of the content in an appropriate manner. This enforcement of the compliant devices is achieved through encryption. Encryption, digital watermarking (DWM) and fingerprinting are the major security controls used in DRM systems for different purposes [21, 33].

Encryption offers protection against unauthorized access to digital content. Encryption of content is like applying a secure wrapper over the content making it possible to get the original content only by removing the wrapper by using a key which is available only to legitimate users. When only encryption is used to protect the content, the content available in clear once the wrapper is removed is vulnerable to copying. This is where watermarking plays an important role in protecting the content which is available in clear. While watermarking complements encryption by offering forensic analysis capability to detect violations of copyright in digital content, it is an effective protection method for analog content.

Digital watermarking is the process of embedding some copyright information into a digital signal, which may be audio, pictures or video [21, 33]. If a copy of the signal is made, the embedded information is also copied, making it possible to trace to the source from which it is copied. The watermarking could be visible or invisible. In visible watermarking, the embedded information is visible in the picture or video. An example is the logo of television broadcaster appearing on broadcasted video. In invisible watermarking, information is added as digital data to the audio, picture or video, which cannot be perceived as such. Annotation of digital photographs with information such as date and time, camera settings, and so on, is an example of invisible watermarking [36].

Watermarks can generally be classified as (i) *robust watermarks* and (ii) fragile watermarks. Robust watermarks are resistant to attacks and are detected for verifying the content copyright information. *Fragile watermarks* change when there is tampering with the content carrying them. Robust watermarks are used for broadcast monitoring, copyright protection, fingerprinting and copy control. Fragile watermarks are used for content authentication and integrity verification and medical records.

Fingerprinting is a form of digital watermarking by which unique watermark is dynamically embedded for each user even if the content remains the same.

Watermarking is also considered as a *last line of defense* against unauthorized distribution of digital content when other protection layers such as encryption is not in effect. Digital watermarking and fingerprinting do not prevent illegal copying of content but they facilitate tracing back to the source of the illegal copy using the embedded data thereby acting only as a deterrent.

19.2.1.1 Drawbacks of DRM Systems

Most of the DRM systems in use are proprietary and are not interoperable. They have also failed to prevent piracy and have put restrictions even on fair use by consumers [25]. Though

the issue of interoperability of DRMs is being addressed, the present day DRMs still suffer from the following drawbacks:

- Digital rights management-enforced rules sometimes contravene the rights and privileges granted to the public under copyright law [31]. However, circumventing any DRM rules may be unlawful under the DMCA [37], which makes it a crime to circumvent anti-piracy measures built into most commercial software.
- Some DRM techniques have the potential to expose users to security and privacy risks [9, 12].
- Lack of interoperability locks the users with selected CE devices and reduces the competition.
- Users face increased complexity due to additional software tools that enforce DRM rules and bear additional cost of the tools which are beneficial to content owners and DRM system vendors.
- Protection of content by restricting its use only in specific hardware devices or software has raised concerns of interoperability.

19.2.2 Content Scrambling System (CSS)

A Content Scrambling System (CSS*)* is an encryption technology used to protect content in DVD-video discs from piracy and apply region-based viewing restrictions. The CSS is a two-part player-host mutual authentication and encryption system for which content owners and manufacturers of hardware or software client system purchase licenses. The information on DVD discs is encrypted. The DVD players, either a computer drive with necessary software or a home video player, have technology to authenticate and decrypt the information for viewers. A DVD copy control association (DVD CCA*)* [6] is a not-for-profit corporation that is responsible for licensing CSS to manufacturers of DVD hardware, discs and related products. Licensees of CSS include the owners and manufacturers of the content of DVD discs, creators of encryption engines, hardware and software decryption systems, and manufacturers of DVD players and DVD-ROM drives.

19.2.3 Content Protection for Recordable Media and Pre-Recorded Media (CPRM/CPPM)

Content protection for recordable media and prerecorded media *(*CPRM/CPPM*)* is a renewable cryptographic method for protecting entertainment content when recorded on physical media and used to protect DVD-audio discs. The system is defined by the specification, which relies on key management for interchangeable media, content encryption and media-based renewability. Content protection for recordable media uses a C2 cipher [17] and a media key block (MKB). A C2 cipher is used to encrypt and decrypt content and also as the basis of one-way and hash functions. *Media key blocks* are tables of cryptographic values that implement a form of broadcast key distribution, and they provide for renewability in content protection solutions. Media key blocks are generated by the 4C entity, LLC, and enable compliant licensed products to calculate a common "media key." Each licensed product is given a set of "device keys" when manufactured (also provided by LLC), which are used to process the MKB to calculate the media key [29].

Content protection for prerecorded media [18] defines a renewable method for protecting content distributed on prerecorded (read-only) media types. It is applicable for both audio and video content in different read-only media types and is suitable for implementation on PCs and CE devices. The system is based on key management for interchangeable media, content encryption and media based renewability. Common cryptographic functions used for CPPM are also based on a C2 block cipher.

19.2.4 Conditional Access System (CAS)

A conditional access system *(CAS)* [7] is another system for protecting content in broadcast domain by enabling only legitimate subscribers to view the transmitted content. The system uses set-top boxes (STBb) and smart cards to protect the content. Content is encrypted at the source and decrypted at the STBs using keys stored in the smart card.

19.2.5 Advanced Access Content System

An advanced access content system [16] is a system for managing content, including next-generation high-definition content, stored on the next generation of pre-recorded and recorded optical media for consumer use with PCs and CE devices. An AACS is based on: (i) Encryption of protected content using AES cipher and (ii) Key management and revocation using Media Key Block technology.

19.2.6 Content Protection System Architecture

Content protection system architecture *(CPSA)* [19] provides a framework of 11 axioms that describe how compliant devices handle CMI and CCI and identify the content protection obligations of compliant modules such as playback, output and recording. It is an overall framework that accommodates the needs of various technologies such as CPRM, CPPM, CSS, DTCP, and so on.

19.2.7 Digital Transmission Content Protection (DTCP)

A DTCP protects audio/video entertainment content from illegal copying, intercepting and tampering as it traverses high performance digital buses, such as the IEEE 1394 standard [14]. Only legitimate entertainment content delivered to a source device via another approved copy protection system (such as the DVD content scrambling system) will be protected by this copy protection system. DTCP relies on strong cryptographic technologies to provide flexible and robust copy protection across digital buses. Content encoded for DTCP protection is encrypted and securely transmitted only to recording and display devices that implement DTCP. The content carries information indicating whether and to what extent the content may be copied.

19.2.8 High-Bandwidth Digital Content Protection (HDCP)

The high-bandwidth digital content protection (HDCP) system [5] was designed to protect the transmission of audiovisual content between an HDCP Transmitter and an HDCP Receiver. Each HDCP device is provided with a unique set of 40, 56-bit secret keys, referred to as the

device private keys, and a corresponding identifier known as the key selection vector (KSV), which is a 40-bit binary value, from the digital content protection LLC. Protection is achieved by encrypting the HDCP content between the two devices using the shared secrets established during the authentication. Further, an HDCP transmitter can identify compromised receiver devices and prevent transmission of HDCP content.

Although HDCP scheme is fast and easy to implement it has been broken because of the design mistake in creating the shared secret linearly [3]. This flaw makes it possible to decrypt eavesdropped communications, spoof the identity of other devices and even forge new device keys as if the keys are from the trusted center.

19.3 Comparison of Content Protection Technologies

From Table 19.1, it is clear that there is no unified framework that can protect content through its entire lifecycle. Therefore, there is a definite need for a unified content protection framework that would meet the requirements as specified in Section 19.5 and would protect the content stored in media or transmitted through device interconnections and in multi-play converged networks.

19.4 Threats in Traditional and Converged Networks

Network is the medium through which service providers around the world reach their customers. Different networks existed for different services like PSTN for voice, Internet for data and Cable network for television. They were originally designed and developed in different domains to cater to specific requirements. Protocols and technologies were developed to specifically address these requirements. Asynchronous Transfer Mode (ATM) was the first network technology that allowed convergence of data, voice and video over wide area networks. But, it is the IP/MPLS (multiprotocol label switching) technology that made the convergence widely deployable through its support for various access technologies such as TDM, ATM, and frame relay [24].

The next generation network (NGN), a concept backed by international telecommunication union's telecommunication standardization sector (ITU-T) among several others to transport information and services (voice, data, and all sorts of media such as video) through a single network by encapsulating these into packets, as in the Internet, is a single converged network that enables broader range of revenue generating service offerings with reduced complexity. It supports triple-play and quad-play service offerings such as video-over-IP, voice-over-IP (VoIP), data services for music and video downloading. With the convergence of networks and services, customers can access *any service, from anywhere, using any device, at anytime*.

19.4.1 Content in Converged Networks

As of June 30, 2009, of the estimated world population of 6.767 billion, 1.668 billion were Internet users [23]. There are about 3.72 billion mobile wireless subscriptions worldwide [1, 41]. Telecom and television services are also witnessing a sea change with cable operators providing broadband data and voice services over television broadcast service [27] and wireline

Table 19.1 Comparison of content protection technologies.

S. No.	Description	Developed by	Techniques used	Protects	Stage in lifecycle	Major drawback(s)
1.	DRM	Apple, IBM Sony, Microsoft, Real Networks, . . .	Rights Expression Language, CMI, CCI, cryptography, watermarking, fingerprinting	CD, DVD, Online Digital Content, portable and networked audio and video devices	Packaging, Broadcast, distribution, consumption	Refer to *Drawbacks of DRM Systems* under Section 19.2.1
2.	CAS	NDS Group ple, ETU, Scientific Atlanta and various telecom agencies	Set-Top Box, Smart cards, Encryption, digital signature	TV Broadcast broadcast	Broadcast, Distribution, Consumption	Limited to television
3.	CSS	DVD CCA	Encryption Digital Signature	DVD Video Disks DVD Players	Packaging, Distribution, Consumption	Limited to DVD discs
4.	CPHM/ CPPM	IBM, Intel, Matsushita and Toshiba	Cryptography Protected CMI, bit-by-bit copy protection	DVD media	Packaging, Distribution, Consumption	Limited to DVD discs and flash media
5.	AACS	Intel, IBM, Matsushita, Microsoft, Sony, Toshiba, Walt Disney and Warner Bros.	AES Encryption, Key Management and digital signature	HD DVD, Blue-ray discs	Packaging, Distribution, Consumption	Limited to HD DVD, Blue-ray discs
6.	HDCP	Intel	Custom Indentity-based Cryptosytem	DVI HDMI, UDI, GVIF, DisplayPort interfaces	Broacast, Distribution, Consumption	Proprietary system
7.	CSPA	Intel, IBM, Matsushita and Toshiba	CMI, CCI, Eneryption	Computers and Consumer Electronic devices	Packaging and Storage, Broadcast, Distribution, Consumption	Based towards content owners
8.	DTCP	Hitachi, Intel, Matsushita, Sony and Toshiba	CCI, Device Authentication and Encryption	Interconnection between devices, USB, IP, WiFi, Bluetooth, FireWire and MOST	Distribution, Consumption	Limited to device interconnections

telecom service providers providing television service in addition to their traditional voice and data services. One such example is IPTV, which is the convergence of communications, computing and content – and slated to be the killer application in the near future [39]. There is an ever increasing demand for packet-based service provider network traffic as compared to circuit-switched transport network due to the potential capex, transport capacity efficiency and opex advantages [11, 13].

The magnitude of the size of the converged network, the multitude of devices attached to it, the threats from the coexisting services, and newer threats warrant innovative security solutions and framework. Existing security solutions cannot address likely newer threats because of newer possibilities. However, existing security technologies can be judiciously combined to create a framework that will form the basis for protecting the content from the point of creation to consumption and storage for future use.

19.4.2 Threats in Traditional Networks

In the traditional voice networks, the physical security to the PBXs and the media provided adequate security to the users' voice traffic. In the traditional data network, the threats were mainly unauthorized access to content such as eavesdropping or hacking in to a system or network, denial of service (DoS) attacks and distributed DoS attacks, malware attacks such as virus, worm and trojans, phishing attacks, exploiting the vulnerabilities in the unpatched and misconfigured systems, and so on. However, security technologies have also matured and tools such as firewalls, IDS and IPS, VPN, content filtering systems, antivirus and antispam solutions have been effectively used to counter threats and protect against those vulnerabilities.

19.4.3 Threats in Converged Networks

Security provided by the physical separation of the transport media in conventional networks is lost when different networks converge. With the content availability over IP, each device attached to the network is potentially a source of copying and redistribution of the content. Content can easily be traded over the Internet without any regard to the IPR. Content owners have little control over unprotected content being distributed over the Internet.

For example, consider broadcast of a mega budget movie through the television network for the first time. The cable and broadcast network is tightly controlled and is not accessible to the unsubscribed public. However, if the broadcast is accessible through Internet using computers and CE devices, the following additional threats are introduced providing possibilities for malicious users to distribute the movie illegally depriving content owners of their royalty:

- Users can have more control, like installing additional software tools for copying the content locally.
- Users can use hacking tools to remove DRM controls embedded in the content.
- Copied content may be re-distributed to unsubscribed users.
- Content purchased and downloaded to DVRs connected to the network is susceptible to retrieval by hackers getting unauthorized access to the system.

All the threats associated with the data, voice, video and wireless services will coexist in a converged network. In addition to the conventional threats associated with the individual services, the following factors contribute to additional threats:

- Additional access doors are opened because of other coexisting services.
- Vulnerabilities in one service can destabilize other services leading to theft of service and denial of service attacks.
- Additional devices that are attached to the network add another dimension to the problem as these devices can potentially become entry points for the intruders in to the converged networks to stage newer attacks.
- Resource overheads, because of encoding/decoding and compression/decompression of voice and video traffic, may limit the implementation strength of the algorithms in CE devices because of limited processing power and available memory.

The SysAdmin, Audit, Network, Security (SANS) Institute listed risks posed by "VoIP Servers and Phones" among the SANS Top-20 2007 security risks [32]. By leveraging the vulnerabilities found in products such as Cisco Unified Call Manager and Asterisk, along with VoIP phones from multiple vendors, attackers can carry out VoIP phishing scams, eavesdropping, toll fraud, or denial-of-service attacks. IP video technology will have to face spam over IP Video (SPIV) and will require device-adequate protections including suppression of unsolicited pop-up advertisements [28].

The threat scenario combined with the drawbacks of the existing content protection technologies underline the need for a unified content management and protection framework that addresses the protection requirements of digital content over its entire lifecycle including protection over converged networks.

19.5 Requirements of a Comprehensive Content Protection System

Illegal access to content and piracy are perceived to be very likely occurrences during network transmission and other forms of distribution. Analogous to the *insider attack,* which is a major information security threat, content can also be illegally accessed and pirated at any stage in its lifecycle. For example, the videographer can have an illegal copy of the video content he captures, where he has been employed by an agency or an individual to cover some event. The videographer has been employed and is providing paid professional services and the actual owner of the content is one who has employed the videographer. A broadcaster is responsible for the safe custody of the content received from a media agency and can only broadcast the content under the terms and conditions of a contractual agreement. However, unless the content is adequately protected by the broadcaster, it is vulnerable to insider theft within the broadcasting agency itself. Hence, it is necessary to protect the content from illegal access and piracy at all stages of its lifecycle.

The flexibility provided by convergence for offering many innovative services to the consumer is also prone to misuse unless there is a fine balance between usability and security. For comprehensive protection of content traversing through various devices and handled by different users across all stages of its lifecycle, the content protection system should meet some basic requirements, not only in terms of confidentiality and integrity of content but also in terms of security and privacy of the users. These requirements are listed below:

- All proprietary content either stored or in transit should be accessible only by authorized users for authorized purposes.

- All proprietary content that are transmitted through any channel/ medium should be protected from unauthorized access by network intruders by suitable encryption.
- All devices and/or users requesting access to protected content should be authenticated using appropriate access control mechanism.
- All users and/or devices that are part of the content delivery network should be registered with a central data repository.
- The data repository should be adequately protected from unauthorized access and modification to data.
- All devices in the content delivery network should be capable of encryption, decryption, digital signature and verification for the purpose of authentication and content protection.
- Each device in the content delivery network should identify itself with the immediate upstream and downstream devices for content delivery.
- The security system should include provisions to permit each user to have multiple devices, such as, PC, mobile phone, IPTV and each device may be operated by multiple users, such as a home user operating set-top box, IPTV and home computer.
- The framework should allow seamless mobility of users and/or devices across geographies and service providers as in the case of mobile phones.
- At any point of time, the content should be identifiable with its creator, owner, distributor and the current rights holder.
- The security framework should allow the concept of family users.
- The content protection mechanism should provide capability for the content consumer to assign rights to the content purchased to legal heir(s) as is permitted in civil laws.

19.6 Unified Content Management and Protection (UCOMAP) Framework

Unified content management and protection[40] is a framework that attempts to address the requirements of video content protection mentioned above in a converged network.

There are several technologies and frameworks currently existing for content protection. As can be seen from Table 19.1, no single the technology or framework provides comprehensive protection of the content from creation stage through the complete lifecycle. An attempt has been made to give a comprehensive protection to content by the proposed UCOMAP framework.

19.6.1 Technical Assumptions

In the context of computer networking, media access control address (MAC address) of network adapters (or NICs) is used not only for the purpose of networking but also for security. Media access control binding in wireless networks and DHCP are examples. A MAC address is a quasi-unique identifier that serves as an identifier for a particular network adapter. In network, security can be enhanced by imposing that only computers with particular MAC addresses could be connected. In the UCOMAP framework, the device identification number *(*DIN*)*, something akin to MAC address in the case of network adapters and IMEI number in the case of mobile phones, is proposed for all the devices that connect to a content delivery network. Each device should have the capability to make a digital certificate request, digitally sign and verify, encrypt and decrypt, and store digital certificates and private keys securely.

digital certificates should contain the DIN as subject name and the owner's name as one of its attributes. It should be legally permissible for a user to export signed and encrypted digital content from one device to another device that he/she or his/her family member owns. In some cases it may be restricted to individuals rather than the family members.

All the devices that connect to the network of UCOMAP devices should have basic cryptographic capabilities for key generation, encryption, decryption, digital signature and signature verification. The UCOMAP framework is designed to address all the requirements as listed above.

19.6.2 Major Components of UCOMAP

UCOMAP has the following major systems and subsystems:

- Cryptography and PKI system.
- User and device data repository.
- Content delivery infrastructure.
- Rights management system.

19.6.2.1 Cryptography and PKI System

Innovations in securing communications over an insecure public channel [4, 22] using public key distribution systems and digital signatures for electronic communications [10, 30] lead to what is currently known as public key infrastructure *(PKI)*.

With the Internet gaining popularity, online transactions became a reality with the use of digital certificates. Digital certificates are issued, managed, renewed and revoked using PKI. Public key infrastructure is also used to bind the digital certificate securely with a consumer and the container of the digital certificate. A PKI consists of the following:

- Certificate authority, which includes the CA server and CA management interface for the designated certificate authority.
- A directory server, which acts as the repository for the issued certificates and CRLs.
- Registration authority, which acts as an intermediary between the CA and the users.
- Online certificate status protocol (OCSP), a component that acts as the gateway for the certificate status verification process.

19.6.2.2 User and Device Data Repository

In UCOMAP, when the user switches on his HDTV or any other device for receiving digital TV content, it authenticates itself to the immediate upstream device using digital certificate and securely exchange a secret key for decrypting content in an agreed fashion, time-based or program-based. On a daily basis the validity of the certificate can be verified using an OCSP certificate verification process. To establish trust and verify certificate validity directory servers can also be established in a hierarchical fashion. Since the trust has to be established on a global scale with billions of end user systems, there has to be many levels of replication

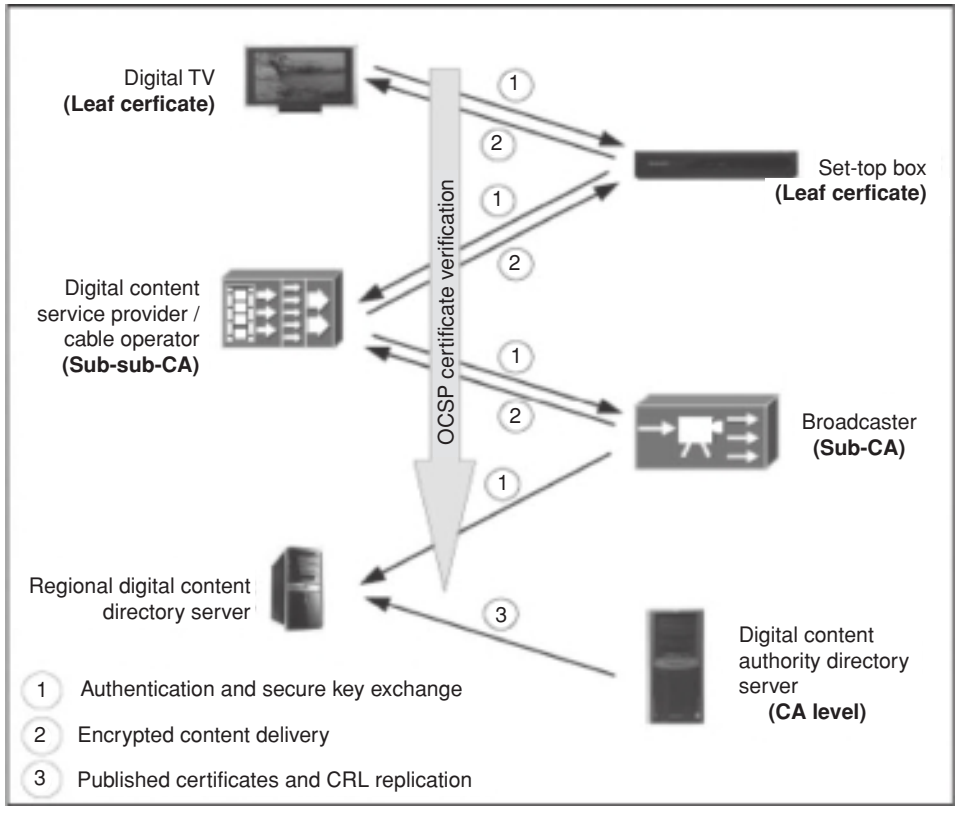


Figure 19.2 Authentication and delivery of encrypted content.

for directory servers – global, regional, national, broadcaster and may be service provider, to reduce the latency in authentication and verification. The two lower most levels of directory servers may only be read-only. A typical operation of the system is depicted in Figure 19.2.

19.6.2.3 Content Delivery Infrastructure

The content delivery infrastructure for digital content has been described earlier. UCOMAP framework protects content undergoing various processes in different stages, such as production, transformations, replication, distribution and storage and retrieval. The content is protected by storing the content encrypted using the public key of the device where it is stored or the public key of the entity to which the content is transmitted.

19.6.2.4 Rights-Management System

In addition to the controls to protect the confidentiality, integrity and authenticity of digital content, sometimes there is a need to specify what the customer can do with the copy of the

content that has been purchased. Flexible customer-driven pricing models can be implemented by including a suitable rights-management system into UCOMAP by applying restrictions based on customers' choice of some rights. UCOMAP does not introduce yet another DRM system. Instead, it leverages existing DRMs, which are specific to individual channels, such as DRM for Internet-based content delivery, DRM for television broadcast, DRM for mobile content delivery or DRM for device or software specific content delivery. It can seamlessly integrate with existing DRMs through APIs and make use of the existing DRM controls or propagate additional DRM controls to the underlying delivery system.

19.6.2.5 Protection for Transmission

Digitally protected content should be transmitted, encrypted and signed to a downstream device only upon authentication. Each registered device must authenticate itself to the corresponding upstream device using digital certificate unless the certificate has been revoked prior to the authentication. The encryption should be made using a secret key exchanged during authentication. Subsequently, the secret key can be renewed in a predetermined manner.

19.6.2.6 Anti-Piracy Protection

Content production systems should be designed to store the content encrypted and signed. The content can be decrypted and viewed by the same content production system without any verification as in the case of viewing the video locally in a camcorder or by using registered devices associated with that particular user. Secured digital watermarking and fingerprinting are used for copyright protection.

19.6.3 Other Advantages of UCOMAP Framework

All devices conforming to the UCOMAP framework will have the following added advantages:

- Online firmware update by the manufacturer or authorized agency without recalling the devices to the workbench.
- Online troubleshooting.

19.7 Case Study: Secure Video Store

Secure video store (SVS) is an application accessible from computer, IPTV, and mobile phone as shown in Figure 19.3. It is an electronic storefront of videos organized by genre and themes so that users can search, browse through and select specific videos. Users can choose from a menu of choices stating the number of plays and/or duration together with the screen on which the video has to be played. For example, the user can buy a video by paying $4.99 to play it ten times in a year on the IPTV or buy it for $8.99 to play it 20 times within a year either on IPTV or mobile phone. The purchase can be made from any of the three registered devices – mobile, TV or the computer. Secure video store is *built once but deployed for multiple channels*. The access to this video store from the three types of end-user devices is protected

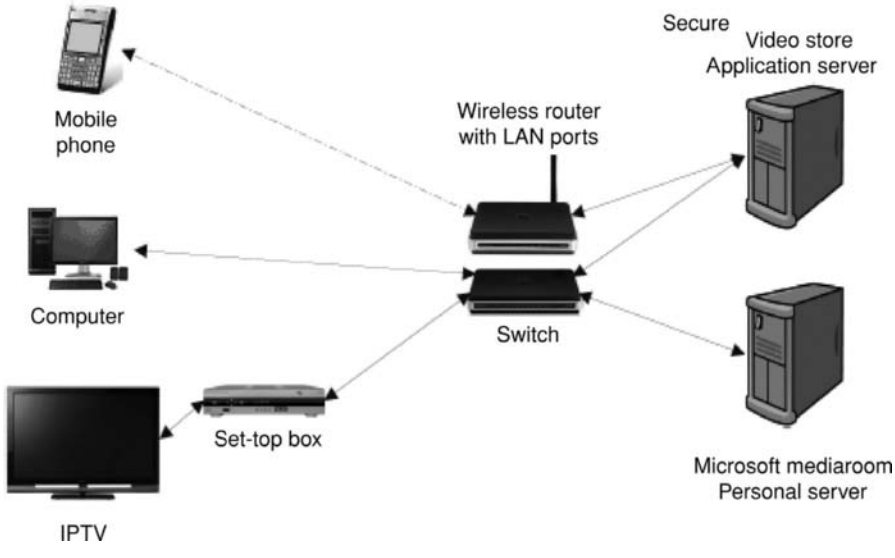


Figure 19.3 Secure video store: Case study.

by the UCOMAP multichannel security framework. A media transcoding engine (MTE) [26] transforms the media files suitable for consumption by different types of end-user devices. Technical architecture of SVS is depicted in Figure 19.4.

While registering on the SVS portal, in addition to the regular login userid/password creation, the user also needs to register the devices, by supplying the device IDs of devices such as the unique DeviceID of STB, the IMEI of mobile phone, and the MAC address of the computer, which will be used to access the video content of this portal. The device IDs of all the devices that users register are mapped against their user ID in the back-end database. After completing the registration process, the user needs to download and install digital certificate for each registered device. Each certificate contains the respective device ID included as part of the subject name of the CommonName (CN) of the certificate. As an added security mechanism, the device should be capable of accepting only the digital certificates that contain its DeviceID in the subject name or the device ID should be extracted from the device and securely compared with the device ID on the certificate before authenticating the device.

An authorized user can access this video store using one of the end-user devices that the user has already registered. Both the user and the device are securely authenticated before the user can purchase or play any content. The user can purchase the rights to the content for playing on any one or more of the devices registered and for a specific duration or number of plays. When a user requests access to a purchased content using a registered device, the user's entitlement to the specific content is verified by the entitlement manager. This verification is done after authenticating the user and the device. For the IPTV, the entitlements are propagated to the Microsoft Mediaroom [42] DRM from the UCOMAP Entitlement Manager using the Mediaroom SDK. After access, the entitlement is updated along with access details using time stamping. Subsequent versions will be enabled with watermarking and fingerprinting for each instance of access.

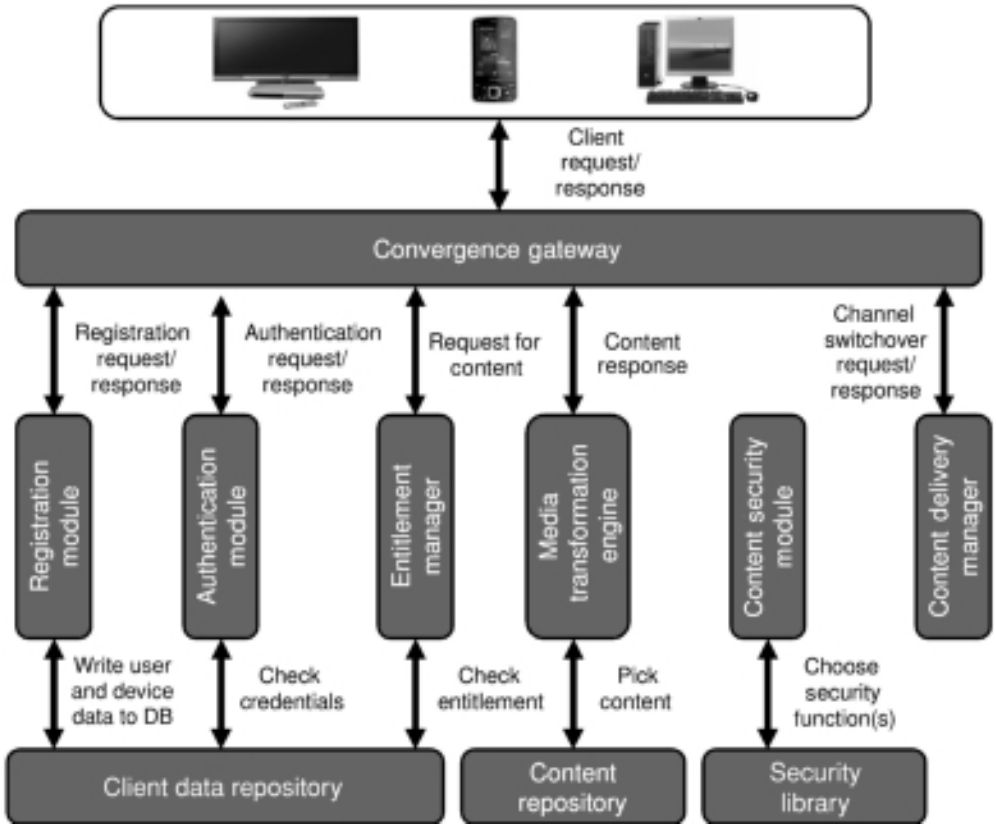


Figure 19.4 Technical architecture of secure video store with UCOMAP.

19.8 Summary

Digital content is vulnerable to unauthorized access and piracy at all stages of its lifecycle, not only during broadcast and distribution but also from insider threats such as employees of production studios, media and broadcast agencies. In a converged network, wired or wireless, which carry data, voice and video, the magnitude and scope of content and its handlers and consumers are enormous. Such a vast content network and the content traversing across the network cannot be protected using the traditional security solutions. The existing content protection technologies and frameworks do not provide lifecycle protection for the content. A security framework referred to as UCOMAP is proposed in this section for the protection of digital content throughout its complete lifecycle in a converged network. The proposed framework is based on trusted and proven PKI technology, which is not known to have been breached. Each device, agency or individual that create, handle, or consume content will have to carry a digital certificate and private keys. All content, in whatever form, raw or processed, will have to be stored or transmitted encrypted and signed. Only certified devices associated with consumers and capable of encrypt, decrypt, sign and verify content will be able to receive, process, store and transmit copyrighted digital content. Copyright protection of the

decrypted content will be provided by secure digital watermarking and fingerprinting. The SVS application based on the UCOMAP framework has been described as a proof-of-concept for the protection of content in a converged network.

References

[1] 3g.co.uk (2008) GSM/HSPA Subscriptions Now 3 Billion Worldwide. URL http://www.3g.co.uk/PR/April2008/5961.htm (accessed June 9, 2010).

[2] van der Beek, K., Swatman, P.M.C. and Krueger, C. (2005) *Creating Value from Digital Content: eBusiness Model Evolution in Online News and Music*. Hawaii International Conference on System Sciences 7, 206a. DOI http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.175.

[3] Crosby, S., Goldberg, I., Johnson, R. *et al.* (2002) *A Cryptanalysis of the High-Bandwidth Digital Content Protection System*. DRM '01: Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management. Springer-Verlag, London, UK, pp. 192–200.

[4] Diffie, W. and Hellman, M. (1976) New directions in cryptography. Information Theory, *IEEE Transactions on*, **22**(6), 644–654.

[5] Digital Content Protection LLC (2006) High-bandwidth Digital Content Protection System, Revision 1.3. Specification.

[6] DVD Copy Control Association: Content Scrambling System. Online. URL http://www.dvdcca.org/css/ (accessed June 9, 2010).

[7] EBU Project Group B/CA (1995) Functional model of a conditional access system. EBU Technical Review, pp. 64–77. URL http://www.ebu.ch/trev_266-ca.pdf (accessed June 9, 2010).

[8] Encryptonite CA/DRM System. http://www.securemdia.com/wp-content/uploads/2010/04/brochure.pdf (accessed June 16, 2010).

[9] Felten, E.W. and Halderman, J.A. (2006) Digital rights management, spyware, and security. *IEEE Security and Privacy*, **4**(1), 18. DOI http://dx.doi.org/10.1109/MSP.2006.12 (accessed June 9, 2010).

[10] Ford, W. (1995) Advances in public-key certificate standards. *SIGSAC Rev.*, **13**(3), 9–15. DOI http://doi.acm.org/10.1145/219618.219714.

[11] Fujitsu Network Communications Inc (2006) The Role of Emerging Broadband Technologies on the Converged Packet-based Network. Whitepaper. URL: http://www.fujitsu.com/downloads/TEL/fnc/whitepapers/ Emerging-Broadband.pdf (accessed June 9, 2010).

[12] Halderman, J.A. and Felten, E.W. (2006) *Lessons from the Sony CD DRM Episode.* USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium. USENIX Association, Berkeley, CA, USA, pp. 6–6.

[13] Han, S., O'Connor, D., Yue, W. and Havala, P. (2006) *Next-Generation Packet-Based Transport Networks Economic Study*. Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference. OFC 2006, 10 p.

[14] Hitachi, Intel, Matsushita, Sony, Hitachi (1998) 5C Digital Transmission Content Protection, Revision 1.0. URL http://www.dtcp.com/data/wp_spec.pdf (accessed June 9, 2010).

[15] Iannella, R. (2002) Open Digital Rights Language (ODRL) Version 1.1. W3C Note. http://www.w3.org/TR/odrl/ (accessed June 9, 2010).

[16] Intel, IBM, Matsushita, Microsoft, Sony, Toshiba, Walt Disney, Warner Bros. (2004) Advance Access Content System (AACS), Technical Overview (informative). URL http://www.aacsla.com/marketplace/overview/aacs_technical_overview_040721.pdf (accessed June 9, 2010).

[17] Intel, IBM, Matsushita, Toshiba (2003) C2 Block Cipher Specification Revision 1.0.

[18] Intel, IBM, Matsushita, Toshiba (2003) CPPM Specification: Introduction and Common Cryptographic Elements, Revision 1.0. URL http://www.aacsla.com/marketplace/overview/aacs_technical_overview_040721.pdf (accessed June 9, 2010).

[19] Intel and IBM and Matsushita and Toshiba (2000) Content Protection System Architecture - A Comprehensive Framework for Content Protection. URL http://www.4centity.com/docs/CPSA_081.pdf (accessed June 9, 2010).

[20] Lin, E., Eskicioglu, A., Lagendijk, R. and Delp, E. (2005) Advances in digital video content protection. *Proceedings of the IEEE*, 93 (1), 171–183. DOI 10.1109/JPROC.2004.839623.

[21] Liu, J. and He, X. (2005) *A Review Study on Digital Watermarking.* Information and Communication Technologies, 2005. ICICT 2005. First International Conference on pp. 337–341.

[22] Merkle, R.C. (1978) Secure communications over insecure channels. *Communications of the ACM*, **21**(4), 294–299. DOI http://doi.acm.org/10.1145/359460.359473.

[23] Miniwatts Marketing Group (2008) Internet Usage Statistics: The Internet Big Picture -World Internet Users and Population Stats. URL http://www.internetworldstats.com/stats.htm (accessed June 9, 2010).

[24] Mohapatra, S.K. and Mortensen, M.H. (2007) A solution framework for next-generation network planning. *Annual Review of Communications*, **60**: 205–210. URL www.iec.org/newsletter/nov07_1/analyst_corner.pdf (accessed June 9, 2010).

[25] Mulligan, D.K., Han, J. and Burstein, A.J. (2003) How DRM-based content delivery systems disrupt expectations of "personal use. DRM '03: Proceedings of the 3rd ACM workshop on Digital rights management. ACM, New York, NY, USA, pp. 77–89. DOI http://doi.acm.org/10.1145/947380.947391. (accessed June 9, 2010).

[26] Paul, S. and Jain, M. (2008) Convergence gateway for a multichannel viewing experience. *Annual Review of Communications*, **61**, 221–230.

[27] Pawlowski, C. and Song, S. (2005) New developments in cable TV networks: data over cable services and PacketCable. Electrical and Computer Engineering, 2005. Canadian Conference on pp. 1634–1637. DOI 10.1109/CCECE.2005.1557231.

[28] Ramirez, D. (2005) Converged Video Network Security - How service providers can counter the various security risks associated with implementing IP Video. White Paper by Lucent Technologies.

[29] Ripley, M., Traw, C.B.S., Balogh, S. and Reed, M. (2002) Content protection in the digital home. *Intel Technology Journal*, **6**, 49–56.

[30] Rivest, R.L., Shamir, A. and Adleman, L. (1978) A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, **21** (2), 120–126. DOI http://doi.acm.org/10.1145/359340.359342. (accessed June 9, 2010).

[31] Samuelson, P. (2003) DRM {and, or, vs.} the law. *Communications of the ACM*, **46**(4), 41–45. DOI http://doi.acm.org/10.1145/641205.641229. (accessed June 9, 2010).

[32] SANS (2007) SANS Top-20 2007 Security Risks (Annual Update) URL http://www.sans.org/top20/ (accessed June 9, 2010).

[33] Su, J.K., Hartung, F. and Girod, B. (1998) Digital watermarking of text, image, and video documents. *Computers & Graphics*, **22**(6), 687–695.

[34] Subramanya, S. and Yi, B. (2006) Digital rights management. *Potentials, IEEE*, **25**(2), 31–34. DOI 10.1109/MP.2006.1649008.

[35] Taban, G., Cárdenas, A.A. and Gligor, V.D. (2006) Towards a Secure and Interoperable DRM Architecture. DRM '06: Proceedings of the ACM workshop on Digital rights management. ACM, New York, NY, USA, pp. 69–78. DOI http://doi.acm.org/10.1145/1179509.117952423. (accessed June 9, 2010).

[36] Tian, L. and Tai, H.M. (2006) Secure Images Captured by Digital Camera. Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on pp. 341–342. DOI 10.1109/ICCE.2006.1598450.

[37] U.S. Copyright Office (1998) *The Digital Millennium Copyright Act of 1998. U.S. Copyright Office Summary*. URL http://www.copyright.gov/legislation/dmca.pdf. (accessed June 9, 2010).

[38] Wang, X., Lao, G., DeMartini, T. *et al.* (2002) XrML – eXtensible rights Markup Language. XMLSEC '02: Proceedings of the 2002 ACM workshop on XML security. ACM, New York, NY, USA, pp. 71–79. DOI http://doi.acm.org/10.1145/764792.764803. (accessed June 9, 2010).

[39] Xiao, Y., Du, X., Zhang, J. *et al.* (2007) Internet protocol Television (IPTV): the killer application for the next-generation internet. *IEEE Communications Magazine*, pp. 126–134.

[40] Nallusamy, R. and Paul, S. (2009) Unified Content Management and Protection. Infosys Whitepaper.

[41] Global Mobile Suppliers Association (GSA) (2009) GSM/3G Market Update. URL http://www.gsacom.com/gsm_3g/market_update.php4 (accessed June 9, 2010).

[42] Microsoft Mediaroom. http://www.microsoft.com/mediaroom/ (accessed June 9, 2010).

20

Challenges for Providing Scalable Video-on-Demand (VoD) Service

When providing video-on-demand (VoD) service to IPTV users, two distinct infrastructure components come into play: (i) the network and (ii) the VoD server. So far, the focus has been on the first part, namely the network infrastructure. In this section, the focus shifts to the second part, namely, the VoD server. Several video-serving strategies will be presented next, gradually approaching a solution that is fairly close to the best possible solution.

A typical video delivery system consists of a video server, a high-speed network and clients (Figure 20.1). A client requests a video from the video server and the server delivers the video stream via the high-speed network. The client request is transmitted via a control channel to the scheduler on the server side, which determines when and on which channel it will deliver the requested video stream to the client. This information is sent to the client via the control channel. The server and network resources required for delivering one video stream are referred to as a data channel in Figure 20.1. The video server retrieves the desired video from the storage and transfers it to each data channel in an order determined by the scheduler.

Each client contains a set-top box, a disk and a display monitor. A client is connected to the network via a set-top box, which selects one or more network channels to receive video data according to instructions from the server. The received video is either stored on the disk or sent to the display monitor for immediate playback. The display monitor can either retrieve stored data from the disk or receive data directly from a data channel. We assume that the client can store minutes of video data. The waiting time experienced by a client is the amount of time that the client has to wait to start the playback once he/she requests a video.

The server network bandwidth is divided into control channels and data channels. Each data channel is capable of delivering a video stream at its playback rate. The server delivers video to clients through either a unicast or a multicast connection. With a unicast connection, the server and network dedicate a single video stream to each client. A multicast connection permits the server to deliver one video stream to a group of clients in the network using a single data channel. For example, each data channel can be implemented using an IP multicast group. A client that needs to retrieve data can join the corresponding multicast group. Since

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

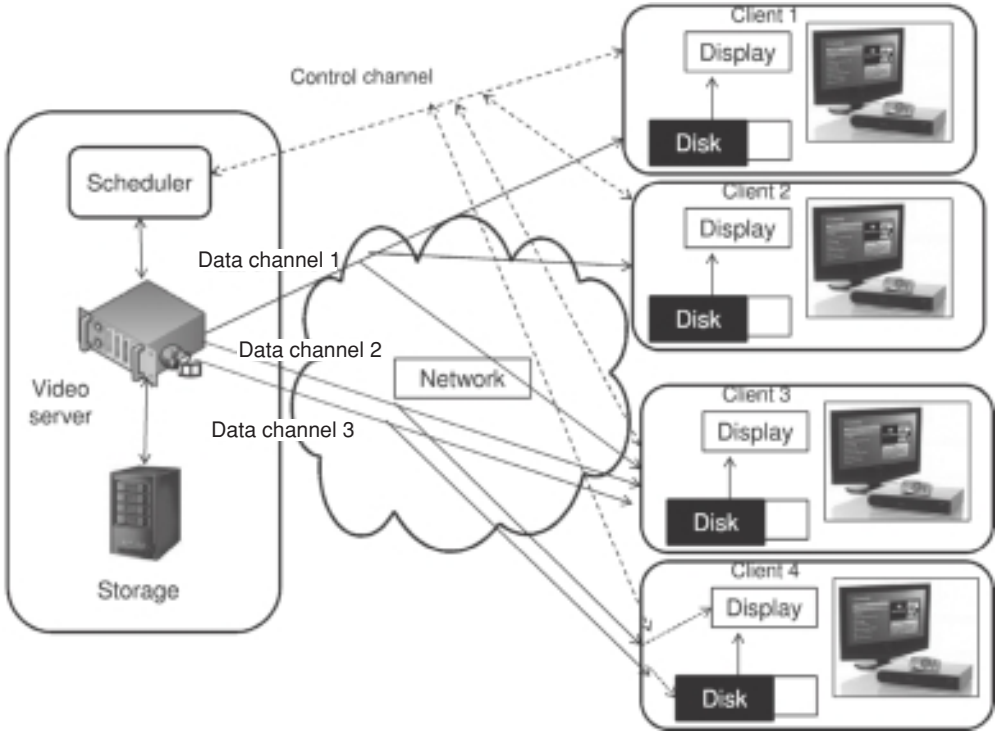


Figure 20.1 Architecture of video on demand system.

data delivery requires the most network and server resources, the goal is to minimize the network and server resources required for video delivery.

Existing VoD schemes fall in two categories: (i) closed-loop (client-initiated) schemes and (ii) open-loop (server-initiated) schemes.

20.1 Closed-Loop Schemes

In most closed-loop schemes [1–8], the server allocates channels and schedules transmission of video streams based on client requests. Basic functioning of a closed-loop system is shown in Figure 20.1. Closed-loop schemes belong to two categories:

- Client initiated (CI).
- Client initiated with prefetching (CIWP).

In client-initiated multicast scheme, a client makes a request for a video and waits until that video is eventually multicast. When a channel becomes available, the server selects a batch of pending requests for a video according to some scheduling policy.

In a client-initiated with prefetching (CIWP) scheme, two clients that request the same video at different times can share a multicast stream; the earlier arrival client is not delayed.

For example, consider two requests spaced 10 minutes apart for a 120 minute-long video. The complete video is multicast by the server in response to the first request. The second request is satisfied by transmitting first ten minutes of the video while requiring the client to prefetch the remaining 110 minutes of the video from the first multicast group. Because the second client is ten minutes behind, it continually buffers ten minutes of the video for much of the time. Client-initiated with prefetching takes advantage of the resources (such as disk storage space and network bandwidth) at the client side to save server network-I/O bandwidth by multicasting segments of video. Several CIWP schemes have been proposed in [15, 18].

Some of the popular closed-loop systems are described next.

20.1.1 Batching

In order to understand the challenges, let us assume a perfect network between the video server and the video clients. The clients send requests to the server at different points of time. In order to reduce response time, the server responds to the requests immediately as shown in the bottom left corner of Figure 20.2. The round-trip time between sending the request and receiving the response is assumed to be T. It is interesting to observe that if the clients request the same video, the server will be streaming the same video multiple times, thereby consuming n times the bandwidth compared to streaming the video only once where n is the number of clients requesting the video. In Figure 20.2, there are three clients, each requesting the same video leading to three times consumption of bandwidth compared with serving a single client

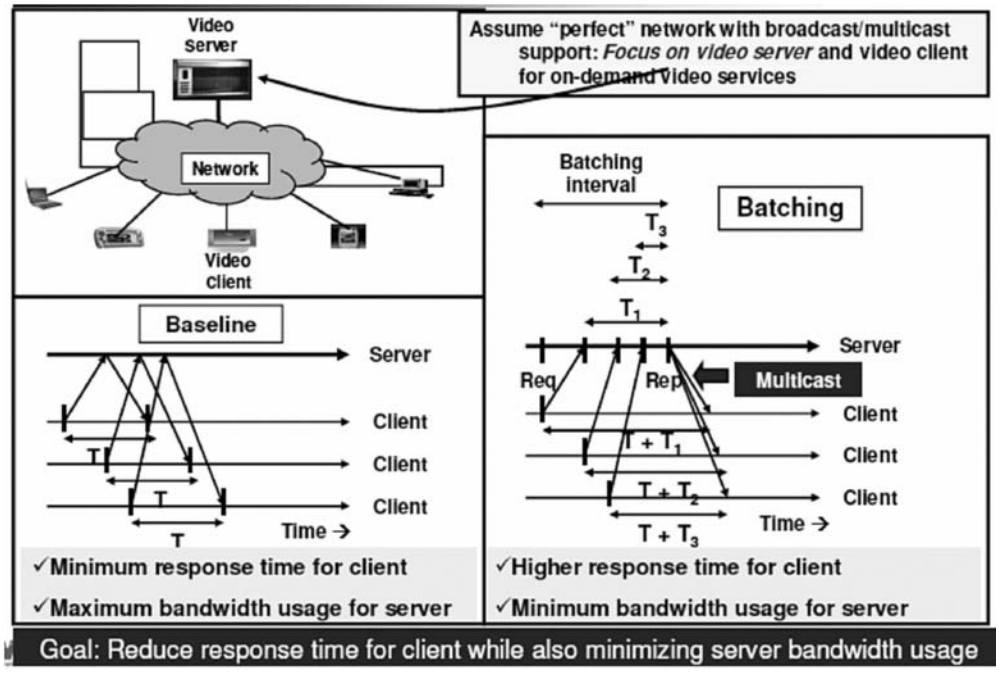


Figure 20.2 Comparing batching of requests to the baseline scenario of sequentially serving requests.

request. This scenario is referred to as the *baseline*. An alternative strategy followed by the video server would be to collect requests over a period of time and multicast the video stream to the interested clients. This approach is shown on the bottom right-hand corner where the response to the first request is delayed by T_1, the response to the second request is delayed by T_2 and the response to the third request is delayed by T_3. In this approach, it is obvious that response time increases compared to the baseline, as it is highly unlikely that the server will respond to a request right away. However, there is significant savings of server bandwidth as the server streams a video once regardless of the number of clients requesting the same video. Since the server "batches" requests over a period before "serving" it, this scheme is referred to as *batching* [5–7, 14].

The ideal solution would minimize the response time for each client while also minimizing the consumption of server bandwidth. Keeping that objective in mind, we discuss the next scheme, which is popularly known as *patching*.

20.1.2 Patching

The notion of patching is based on "patching" missing portions of a video stream for a client [15–17]. As shown in the baseline part of Figure 20.3 on the left, there are four clients requesting the same video from the server and the server serving those requests as soon as they arrive in a sequential manner.

Exactly as before, the response time is minimized in the baseline and server bandwidth consumption is maximized. The idea in *patching* is for the server to wait for a fixed time to batch requests (exactly as in batching) before streaming the video using multicast, and to stream the missing portion of video using unicast (unlike batching). The right hand side of

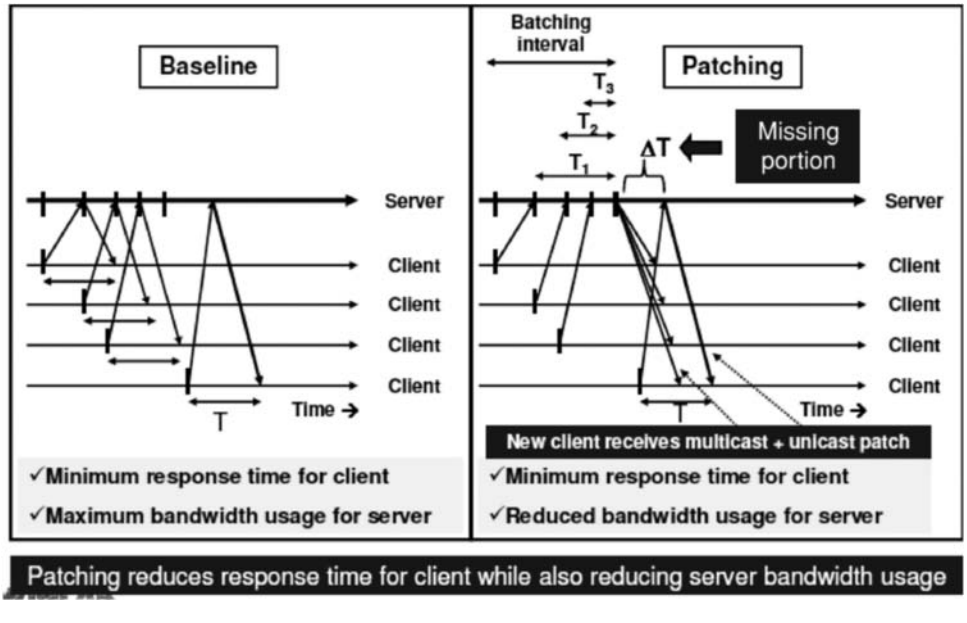


Figure 20.3 Comparing *batching* of requests to the baseline scenario of sequentially serving requests.

Figure 20.3 shows that the server waits to receive three requests before starting to multicast the video stream. The fourth request arrives ΔT time after the server has started multicasting the requested video stream. In patching, the fourth client joins the ongoing multicast stream ΔT time after the streaming has started and hence has missed the first ΔT amount of the video. Therefore, the client sends a request to the server for a "patch"' consisting of the first ΔT time of the video it has missed. Thus, the client opens two sockets, the first to receive a multicast common stream and the second to receive the unicast patch stream. This is shown on the right hand side of the Figure 20.3. There are two things to observe: (i) the response time for patching is better than that for batching, as the client does not always have to wait and (ii) server bandwidth is slightly higher than that needed for multicast-only streaming as the server has to serve an "incremental" amount of video for helping the late-joining client catch up with rest of the clients.

Note, however, as the number of clients sending requests for patches increases, problems similar to batching start to crop up. This leads to the next technique that leverages multicast for streaming "patches".

20.1.3 Batched Patching

This scheme is similar to "patching" from the client perspective except that it will open two multicast sockets. The first is for tuning in to the original multicast stream and the second one is to tune in to the multicast stream containing the requested patch. The concept is shown in Figure 20.4.

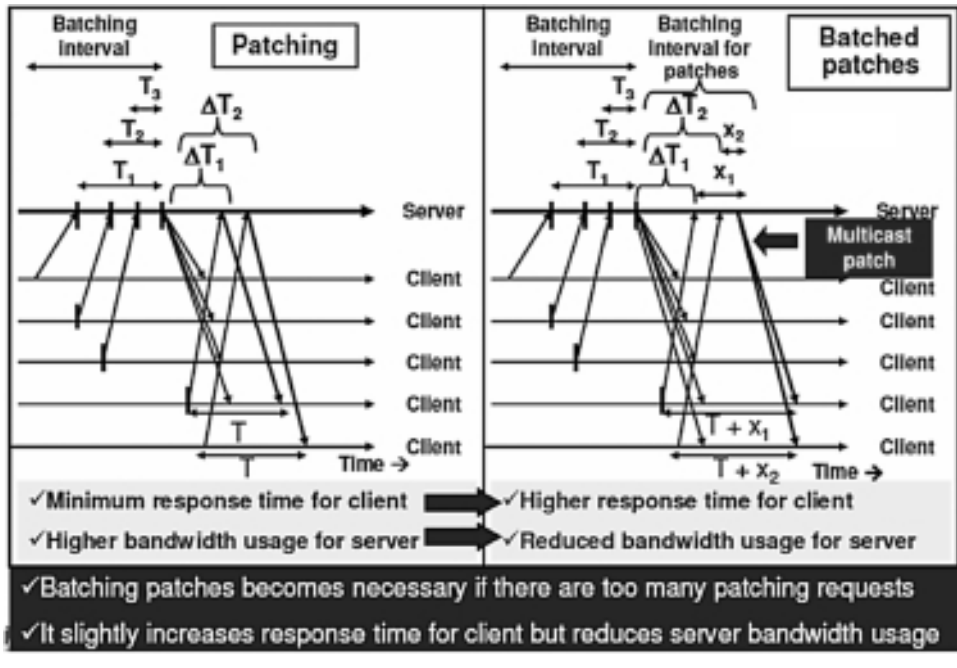


Figure 20.4 Comparing *batched patching* to patching.

In the left part of Figure 20.4, three clients request the same video and the server serves them using a multicast stream. Then there are two more clients requesting the same video ΔT_1 and ΔT_2 times after the server started multicasting the requested video stream. In patching, the server will have two unicast patch steams, one containing ΔT_1 amount of video missed by the first late-joining client and the second containing ΔT_2 amount of video missed by the second late-joining client. The right-hand side of Figure 20.4 shows batched patching in which the server multicasts a patch containing ΔT_2 amount of video out of which the first late-joining client picks only the $\Delta T_1 < \Delta T_2$ amount of video that it missed while the second late-joining client picks up the entire ΔT_2 amount of video. Certainly, batched patching reduces the server bandwidth consumption compared to *patching* while increasing the delay due to batching of patches. The next section investigates another scheme to reduce the response time for the clients while also reducing server bandwidth requirement.

20.1.4 Controlled (Threshold-Based) Multicast

Threshold-based multicast scheme improves on the basic CIWP algorithm in [15,18] by introducing a threshold to control the frequency with which a complete video stream is delivered. In other words, threshold-based multicast does not always allow a client to share a segment of a complete video stream as in "patching" and "batched patching". Instead, if it has been some time since the most recent complete transmission of the video, the server will initiate a new complete video transmission. Here's the rationale. Suppose there are four clients. The first client requests the video at time 0 and is indeed the first client requesting the video. The server will multicast a complete video stream at time 0. The request from client 2 for the same video arrives at time t_1, say. The server will transmit the first t_1 minutes of video for client 2 to catch up (patching). Client 2 will not only tune in to the multicast stream started for client 1 but also to the stream containing the patch. Now, suppose clients 3 and 4 request the same video at time $L/2 + t_2$ and $L/2 + t_3$ respectively where L is the length of the complete video. If the server continues to send patches, the total length of the two patches would be $L + t_2 + t_3$ which is more than the total length of the video. A better strategy for the server would be to multicast the complete video to client 3 and stream a patch of length $t_3 - t_2$ to client 4. In this case, the total length of video transmitted by the sender would be $L + t_3 - t_2$, which consumes less server bandwidth. This implies the existence of a threshold beyond which transmitting patches (partial video streams) to clients becomes expensive so far as server bandwidth is concerned. The main idea of threshold-based multicast is described below.

Let T denote a threshold that controls the frequency at which a complete video is multicast. When the first request arrives, the complete video is immediately multicast on a specific channel. All subsequent requests for the same video will tune in to the same channel to prefetch content as long as they arrive within T minutes of starting the multicast. Requests arriving beyond T minutes will be served by a new multicast session for the complete video. This process is repeated for all videos and all requests.

Going into the next level of detail, when a client request arrives for a video, there are four potential ways the requests can be satisfied:

- Batch the requests to be served by a complete video multicast session to start later. This is used when the current time (t_{cur}) is before the latest starting time of the complete video multicast (t).

- Prefetch from an ongoing multicast session of the complete video and serve the missing portion using a second multicast session of the batched "patches" to start later. This is used when the current time (t_{cur}) is after the latest starting time of the complete video multicast (t) but is before the latest starting time of the "patch" multicast (t_p) and the difference between t_p and t is less than the threshold T.
- Prefetch from an ongoing multicast session of the complete video and serve the missing portion using an immediate transmission of "patch". This is used when the current time (t_{cur}) is after the latest starting time of the complete video multicast (t) and is also after the latest starting time of the "patch" multicast (t_p) and the difference between t_p and t is less than the threshold T.
- Serve the request using an immediate multicast session of the complete video. This is used when the current time (t_{cur}) is after the latest starting time of the complete video multicast (t) and is also after the latest starting time of the "patch" multicast (t_p) and the difference between t_p and t is greater than or equal to the threshold T.

Results show that the server network-I/O bandwidth requirement is reduced significantly by using the optimal threshold. Specifically, in the basic CIWP, the server network-I/O bandwidth requirement for a video is a linear function of the request rate times the video length, whereas, in the threshold-based multicast, the server network-I/O bandwidth requirement for a video is a linear function of the square root of the request rate times the video length. Simulation results presented in [40] show that even if the client disk storage is as small as 150 MB, threshold-based multicast outperforms the basic CIWP significantly.

20.1.5 Batched Patching with Prefix Caching

The client response time has two components: (1) time between receiving a client request and serving it and (2) round-trip time between the client and the server. In Figure 20.5, these two components are shown as x_i and T respectively. The schemes described so far focused on reducing x_i. Another approach for reducing the response time is to decrease T. This can be done by caching the "prefix" of the requested video in an intermediary (such as, a regional cache), which is significantly closer to the client compared to the origin server (which could be an entity at the national level). How the system will identify the would-be requested videos ahead of time and cache their prefixes is beyond the scope of this discussion. There are several techniques proposed in the literature for pre-fetching. A simpler way is to select the "popular" videos (80–20 rule: 80% requests are for 20% videos) and cache their prefixes at the intermediary so that a near-instant start-up time can be guaranteed when a client requests one of those popular videos. This approach is referred to as multicast with cache (Mcache) [22].

The main idea of Mcache is to combine the best of "batching", "patching" and "prefix caching" with multicasting to compensate for the weaknesses of one scheme with the strengths of others to provide the best possible performance in a closed-loop scheme.

Going into the next level of detail, a client sends its request simultaneously to the server and the cache. The Mcache technique works as follows:

- The request is immediately served by the cache for the prefix of the requested clip (prefix caching).

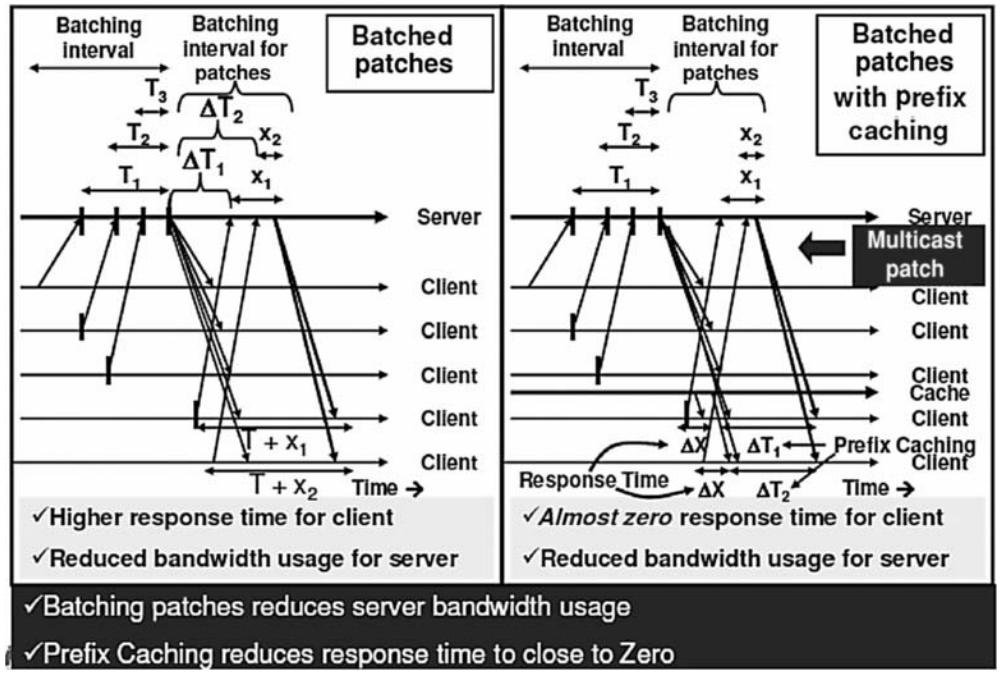


Figure 20.5 Prefix caching helps reduce response time for batched patching.

- As other requests arrive when the prefix is being served, the server batches them and serves them using a single multicast (batching and multicasting).
- Client requests arriving after the multicast session for the video has started are served using "patches". While clients receive the prefix from the cache, patches are batched and served using a single multicast (batching, patching and multicast) from the server.

The objective of the Mcache scheme, as is apparent from the above description, is to reduce bandwidth usage (whether at the server or at the cache) by sharing multicast channels for both the complete video as well as for the patches among as many clients as possible. One interesting side effect of providing instantaneous playback using prefix caching is that both the server and the cache get enough time to delay the start time of multicast without actually delaying the client's playback time and hence, can use this time to batch many more client requests than would be possible without prefix caching. This enables Mcache scheme to use the server bandwidth very effectively.

As shown on the right hand side of Figure 20.5, the round-trip time is reduced from T to Δx and hence the client response time is reduced.

Note that if the amount of video "prefix" cached in the intermediary is more than the amounts of video missed (ΔT_1, ΔT_2) by the late-joining clients, then the entire patch can be served from the intermediary, thereby reducing client response time and also reducing server bandwidth (as the video is served from the intermediary as opposed to from the server). However, if the prefix cached in the intermediary is less than the requested patches from the

clients, the client requests for the remaining amount as the desired "patch" from the server. The server batches those patch requests and serves them using a multicast stream. In this case, the server bandwidth consumption is more than in the case when the entire patch is served off the intermediary but is less than in the case when the entire patch is served off the origin server.

Although this technique is better than what we have seen so far, the bandwidth consumption from the intermediary could be high as the patches are still unicast from the intermediary. Another optimization that can be done would be to batch requests at the intermediary and use a single multicast stream to supply all the "patches" in one transmission.

Note, however, that as the number of clients sending requests for patches increases, it is not clear how long the server (or the intermediary in case of prefix caching) should wait for batching the patches. A longer wait for batching would reduce server (intermediary) bandwidth but increase the client response time, while a shorter wait for batching would reduce client response time while increasing the server (intermediary) bandwidth. It is therefore important to choose the batching interval judiciously.

Assuming a Poisson arrival of rate λ, denoting the batching interval by y, prefix length by x and length of a clip by L, the optimal batching interval can be obtained by solving the equation for y and rounding it off to the nearest integer [22]:

$$y^2 * \quad (1 - \alpha) + 2y + 1 + 2\alpha - 2\mathrm{L} = 0$$

where $\alpha = \mathrm{e}^{-\lambda}$.

The benefits of Mcache scheme may be summarized as follows:

- Under very low arrival rates, the mean server bandwidth tends to λL where λ is the arrival rate and L is the length of the complete video and for high arrival rate, the mean server bandwidth is bounded by O(log L) which is independent of the arrival rate. The linear relationship between server bandwidth and λ at a low arrival rate can be explained by the fact that each request triggers a separate transmission from the server. At a high arrival rate, as multiple requests are batched and served by a single multicast session, the server bandwidth is independent of the request arrival rate λ.
- Clients experience instant playback of requested video, thanks to prefix caching. Further, the client bandwidth is constant and is limited to usage of at most two channels (one for retrieving the prefix from the cache, and the other for the rest from the server) and is independent of the length of the video.
- The performance of the Mcache system does not depend as much on the disk space of the client (needed for storing prefetched content) and the cache (needed for storing prefix of video clips) as most of the other closed-loop systems. In fact, Mcache trades off bandwidth and storage at the server and at the cache much better than most if not all closed-loop approaches.
- Unlike many other schemes, Mcache does not need to know how much disk space is available at the clients.

One other possible dimension of optimization is segmenting the video and serving the video segments from the server using separate multicast channels as opposed to serving the entire video using a single multicast channel. This is exploited in the next scheme.

20.1.6 Segmented Multicast with Cache (SMcache)

The main intuition behind SMcache algorithm [22] is to "iteratively" apply the Mcache algorithm for each segment of a complete video and derive the benefits of Mcache for each segment leading to further saving of bandwidth at the server. In fact, in the case of Mcache, the threshold for controlling the frequency of multicast is in the order of SQRT(L) and hence the mean server bandwidth for Mcache is O(SQRT(L)), which is worse than that of some open-loop schemes (described in the next section) that require O(log L)) bandwidth at the server.

However, the above limitation of Mcache scheme can be overcome without sacrificing the benefit of lower latency of clients by segmenting a video of length L into n segments of lengths L_1, L_2,.., L_n respectively and treat each segment as if it were a complete video in Mcache scheme. Here is how the SMcache scheme would work:

1. L_1 is multicast by the server exactly as in Mcache scheme at time 0.
2. A request from a client arrives at time u_1. That means, the client needs to receive a patch of length $u_1 + x_1$ where x_1 is the size of prefix for L_1. Note that transmission time (t_p) of this patch should be before time $u_1 + x_1$ because the client would have completed reception of the patch by that time, and also before time y_1 (where y_1 is the threshold for controlling multicast of segment L_1) because a new multicast would start after time y_1. Moreover, the transmission of the patch must end before $u_1 + x_1 + L_1$ because the client would have completed playback of segment L_1 by that time. Therefore during time interval ($y_1 + u_1 + x_1$, $L_1 + u_1 + x_1$), the client simply receives a single stream (patch) of size $L_1 - y_1$. This is as if the client is generating a "virtual request" for segment L_2 with a prefix size of $x_2 = L_1 - y_1$ at time $u_2 = y_1 + u_1 + x_1$ with a threshold of y_2 for segment L_2.
3. L_2 is multicast and the client joins the multicast channel after it has completed reception of the prefix x_2 from the server.
4. The algorithm is iteratively applied for segment $i + 1$ where the client generates a "virtual request" for segment $i + 1$ with a prefix size of $xi_{+1} = L_i - y_i$ at time $ui_{+1} = y_i + u_i + x_i$ with a threshold of y_{i+1} for segment L_{i+1}.

In summary, SMcache is a generalized version of Mcache with improved performance. Essentially, by partitioning a video clip into smaller segments, it becomes possible to avail two receiving channels at the client to a far greater extent compared to Mcache. SMcache enables a closed-loop scheme to have the same efficiency as the most optimal open-loop scheme in terms of server bandwidth usage (O(log L)) when the request rates are high and the prefix ratio is large (that is, video clips are significantly larger than the prefix). In addition, SMcache maintains lower server bandwidth usage (typical of closed-loop schemes) compared to open-loop schemes when the request rates are low. Furthermore, SMcache maintains the lower latency at the clients provided by Mcache and outperforms open-loop schemes in that regard.

20.2 Open-Loop Schemes

Open-loop schemes [9–13] require constant bandwidth regardless of request rate as the server's transmission is not triggered by client requests. In fact, the control channel in Figure 20.1 does not exist in open loop systems. Open-loop schemes belong to two categories:

- Server-initiated (SI).
- Server-initiated with prefetching (SIWP).

In a server-initiated multicast scheme, the server multicasts video objects periodically via a dedicated server network and I/O resources. Clients join an appropriate multicast group to receive the desired video stream. The server-initiated scheme can guarantee a maximum service latency independent of the arrival time of the request. For example, the simplest server-initiated scheme involves starting multicast sessions at fixed time intervals (say every x minutes) and transmit video clips regardless of whether there are clients interested in the video or not. This scheme guarantees a maximum service latency of x minutes independent of the arrival time of the requests. Therefore, sever-initiated schemes are more efficient for hot videos that receive more requests than for cold videos that receive fewer requests.

In a server-initiated with prefetching (SIWP) scheme, a video object is divided into segments, each of which is multicast periodically via a dedicated multicast group. The client prefetches data from one or more multicast groups for later playback. Prefetching is done in such a way as to ensure the continuous playback of the video. A number of server-initiated schemes [11,12,20,21] have been proposed. A SIWP scheme takes advantage of resources (such as disk storage space and network bandwidth) at the client end, and therefore significantly reduces the server network and I/O resources required. Like server-initiated schemes, such a scheme guarantees a maximum service latency independent of the arrival time of the request and performs better for hot videos than for cold videos.

20.2.1 Equally Spaced Interval Broadcasting

In this scheme, a given video is broadcast at equally spaced intervals. Hence service latency can only be improved linearly with the increase of server bandwidth.

20.2.2 Staggered Broadcasting

In this scheme, multiple copies of the same video are broadcast except staggered (overlapping) in time. Service latency can be improved by reducing the interval between successive broadcasts. This will require higher bandwidth at the server compared to interval broadcasting as multiple copies of the same video are broadcast simultaneously.

20.2.3 Harmonic Broadcasting

In Harmonic Broadcasting [23], each video is equally divided into N segments S_1, S_2, .., S_N. Then each segment S_i is further divided into i segments denoted by $S_{i,1}$, $S_{i,2}$, .., $S_{i,i}$ of equal size. Thus, segment S_1 has only one subsegment $S_{1,1}$, S_2 has two subsegments $S_{2,1}$ and $S_{2,2}$ and so on as shown in Figure 20.6.

N channels C_1, C_2, ...,C_N are allocated where channel C_i broadcasts all subsegments of S_i sequentially and periodically. Thus, if the bandwidth required for channel C1 is the data consumption rate b of the video, the bandwidth required for channel Ci is b/i. Therefore, total bandwidth required for N channels $= \sum_{1}^{N} \frac{b}{i}$.

It was further shown in [24] that harmonic broadcast is bandwidth optimal for a constant bit-rate video broadcast.

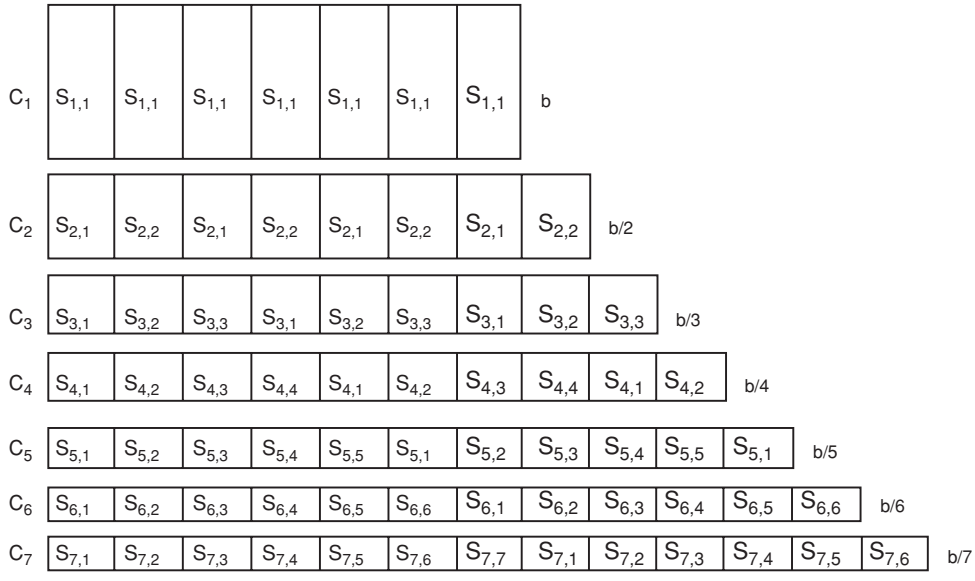


Figure 20.6 Harmonic broadcast for 7 channels.

20.2.4 Pyramid Broadcasting

In pyramid broadcasting [10], each video is partitioned into segments of geometrically increasing sizes, and the server bandwidth is divided into K data channels. The i^{th} channel is used to broadcast the i^{th} segment of all videos sequentially.

Let us denote the server bandwidth by B, number of videos by M, length of a video by D, number of data channels by K, and display rate of each video by b. In addition, let us represent i^{th} fragment of video v by S_i^v, and size of the i^{th} fragment of a video by D_i. In pyramid broadcasting, the size of subsequent segments are chosen in such a way that $D_1, D_2, \ldots, D_K$ of each video form a geometric series with the factor α such that $\sum_{1}^{K} Di = D$. In other words, $D_{i+1} = \alpha.D_i = D_1.\alpha^i$ for i=/=0 and $D_{i+1} = D.(\alpha - 1)/\alpha^{(K-1)}$ for $i = 0$.

In pyramid broadcasting, the entire bandwidth is divided into K logical channels with a bandwidth of B/K per channel. The i^{th} channel will periodically broadcast only the i^{th} segments of each video, namely $S_i^1, S_i^2, \ldots, S_i^M$. No other segments are transmitted through this channel. The client begins downloading the first data segment of the requested video at the first occurrence and starts playing it back concurrently. For the subsequent fragments, it downloads the next fragment at the earliest possible time after beginning to play back the current segment. Thus at any point of time, the client downloads from at most two consecutive channels and consumes the data segment from one of them in parallel. The parameter α must be chosen in such a way as to ensure that the play back duration of the current segment eclipses the worst latency in downloading the next segment. Mathematically, $\frac{D_i}{b} \geq \frac{D_{i+1}.M}{\frac{B}{K}}$.

Substituting $\alpha.D_i$ for D_{i+1}, we have $\alpha \leq \frac{B}{b.M.K}$. Since the first segments are very small, they can be broadcast more frequently than others using the first channel. This ensures a small waiting time for each video. In fact, since the access time of a video is equal to the access time of the first data segment which is multicast on channel 1, the worst case wait time can be

computed as: $\frac{M.D_1.b}{B/K} = \frac{D.M.K.b.(\alpha-1)}{B.(\alpha^k-1)}$. This means by letting K increase as B does, the access time (latency) can be exponentially improved with B. In other words, higher server bandwidth (B) and more segments (K) reduce latency in pyramid broadcast scheme.

The main drawback of this scheme is the large disk size at the client to hold approximately 70% of the video. In addition, because the client receives video on multiple channels, a very high I/O bandwidth is needed to write data to the disk as quickly as it receives the video.

An alternative scheme called permutation-based pyramid broadcasting (PPB) [11] has been proposed to partially overcome the limitations of pyramid broadcasting (PB) scheme. In PPB, each logical channel is further divided in P.M sub-channels where P is an arbitrary number and M is the number of videos in the system. Each subchannel has bandwidth $\frac{B}{K.P.M}$. A replica of each fragment S_i^v of size D_i is now broadcast on each of the P logical subchannels with a phase delay of $\frac{D_i}{P}$. On each subchannel, one segment is repeatedly broadcast in its entirety. Since smaller bandwidths are used for transmitting the video segments and the same video segments are repeatedly broadcast with a stagger, the disk space requirement at the client is significantly reduced compared to pyramid broadcast. Furthermore, the service latency is reduced to the time needed to access the first segment, which is $\frac{D_1.M.K.b}{B} = \frac{D_1}{P+\alpha}$.

Therefore, by virtue of each channel multiplexing its own segments instead of transmitting them sequentially, and starting a new stream within a shorter duration, I/O bandwidth is reduced and so is disk-space requirement at the client. However, because the size of data segments increases exponentially, the last segment becomes very large (larger than 50% of the video size) and hence the disk size requirement at the client still remains large. One other drawback of PPB scheme is the higher client-side bandwidth requirement as it needs the client to tune into multiple logical subchannels at the same time to collect data for a given segment. This synchronization mechanism is difficult to implement because the tunings must be done at the right moment during a broadcast.

While PPB tries to address some drawbacks of PB but the complexity involved in achieving those gains seem to outweigh the benefits. Furthermore, the main painpoint of PB, namely large disk space at the client still remains as an important issue that need to be dealt with. This leads to the next scheme, which is as simple as PB but overcomes the drawbacks of both PB and PPB.

20.2.5 Skyscraper Broadcasting

In Skyscraper Broadcasting [1, 12], the server bandwidth B is divided into $\frac{B}{b}$ logical channels. These channels are then allocated evenly among the M videos such that there are $K = \frac{B}{b.M}$ channels for each video. To broadcast a video over the K dedicated channels, each video file is partitioned into K segments using the data segmentation scheme described next. Each of these segments is repeatedly broadcast on its dedicated channel at the playback rate.

In the skyscraper broadcasting (SB) scheme, instead of fragmenting a video file according to a geometric series [1, α, α^2, α^3, ...] as in pyramid broadcasting scheme, a series generated by the following recursive function is used:

$$f(n) = \begin{cases} 1 & n = 1 \\ 2 & n = 2, 3 \\ 2f(n-1)+1 & n \bmod 4 = 0 \\ f(n-1) & n \bmod 4 = 1 \\ 2f(n-1)+2 & n \bmod 4 = 2 \\ f(n-1) & n \bmod 4 = 3 \end{cases}$$

An illustration of how the series unfolds, here is an example: {1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52,. . .}.

This means the size of the second fragment (D_2) is twice that of the first segment. Or $D_2 = 2 \times D_1 = D_3$.

Similarly, $D_4 = D_5 = 5 \times D_1$ and so on. Additionally, another parameter called width (W) is used to restrict the segments from becoming too large. If some segment is larger than W times the size of the first segment, it is restricted to W units. The reason behind this design choice is that a bigger K^{th} fragment will result in a larger disk requirement at the client. This scheme is referred to as skyscraper broadcasting because the data segments stack up very tall instead of a short and very wide pyramid in pyramid broadcasting.

The width W can be controlled to achieve the desired access latency. Note that the number of videos (M) determines the parameter K (recall K = B/(b.M)). Given K, the size of the first fragment (D_1) can be controlled using W. A smaller W leads to a larger D_1 and conversely, a larger W leads to a smaller D_1. Since the maximum access latency is D_1, the access latency can be reduced by increasing W.

The server multiplexes among K.M logical channels where each channel is used to repeatedly broadcast one of the K.M data fragments. The reception at the client end is done in terms of a transmission group, which is defined as consecutive segments having the same sizes. For instance, in the example series {1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52,. . .}, the first segment forms the first transmission group, the second and the third constitute the second transmission group (2,2), the fourth and the fifth constitute the third transmission group (5,5) and so on. A transmission group (X,X) is called an odd or even transmission group depending on whether X is odd or even. The odd and even groups are interleaved in the transmission side. In order to receive and play back these data segments, a client takes turns in tuning into an odd group and an even group and when it tunes in to any group, it downloads the corresponding segments in their entirety, and in the order they occur in the video file. As the loaders fill up the client buffer, the video player consumes data at the rate of the broadcast channel and plays back the content.

The SB reduces the storage-I/O bandwidth to 3.*b,* and access latency to the time needed to download the first segment of size D_1. In addition, it limits the client disk requirement to the product of D_1 and W.

20.2.6 Comparison of PB, PPB and SB

The comparison among PB, PPB and SB schemes is captured in Table 20.1.

Assuming playback rate b = 1.5 Mbits/s, size of video D = 120 minutes and number of popular videos M = 10, and computing the parameters K, P and α as per the corresponding schemes, are extensive simulations are carried out to get some intuitions as to how the above three schemes compare in terms of performance.

Table 20.1 comparison among PB, PPB and SB schemes.

Technique	I/O Bandwidth (Mbits/s)	Access Latency (min)	Buffer Space (Mbit)
PB	b+2B/K	DMKb $(\alpha - 1)/[B(\alpha^K - 1)$	$60.b(D_K - bKD_K/B + D_{K-1})$
PPB	B+B/KPM	D_1 MKb/B	$60.bDMK\ (\alpha^K - \alpha^{K-1})/[B(\alpha^K-1)$
SB	2b or 3b	$D/[\Sigma^{K}_{i=1} \min(f(i), W)]$	$60.bD_1\ (W-1)$

Disk I/O Bandwidth: PPB and SB have similar disk bandwidth requirements except that the requirements for SB can be reduced by selecting W = 2. In fact a smaller W leads to lower disk I/O bandwidth requirement and lower disk space requirement at a client. SB requires only 3.b disk I/O bandwidth to guarantee a jitter-free playback. In contrast, PB has large disk I/O bandwidth requirement. For example, the mean disk I/O bandwidth required in PB can be as high as 50.b.

Access latency: PPB can have significant access latency. In fact to keep access latency below 30 seconds, PPB requires a very high server I/O bandwidth (300 Mbits/sec). Access latency in SB can be controlled by choosing W judiciously. Larger values of W lead to lower access latency while increasing client disk space requirement. If keeping client disk space small is important, W needs to be decreased and that leads to higher access latency. In contrast, PB provides very low access latency at the cost of higher client disk space requirement.

Client buffer (disk) space: PB requires very high buffer (disk) space at the client. In fact, each client is expected to have about 75% of size of video as the buffer space. For a two-hour MPEG-1 encoded video with playback rate of 1.5 Mbits/sec. The client buffer requirement is higher than 1.25 GB in PB. PPB reduces buffer size requirement at the cost of higher access latency to about 250MB which is 20% of PBs. Under similar conditions, the client buffer requirement in SB is about 30MB with W = 2 and that is about 12% of PPB's.

In summary, SB is a generalized broadcasting technique parameterized by W, the width of the skyscraper. W can be controlled to obtain the desirable trade off between client buffer (disk), disk I/O bandwidth and access latency. SB can offer low access latency comparable to that provided by PB at a significantly lower client buffer (disk) compared to what PB requires. Compared to PPB, SB is much simpler, does not require complex synchronization, requires much lower client buffer (disk) and provides lower access latency. Thus, while PB and PPB must make a tradeoff between access latency, client buffer (disk) size, and disk I/O bandwidth, SB allows flexibility via parameter W to win on all three metrics.

20.2.7 Greedy Disk-Conserving Broadcast (GDB)

A Greedy disk-conserving broadcast (GDB) scheme [21], like PB and SB, partitions a video into segments. Each segment is periodically broadcast using a dedicated multicast channel. The size of the segments is carefully designed so that the clients who wish to receive the video can join the appropriate channels at the scheduled times to ensure continuous playback.

A simple example (Figure 20.7) is chosen to illustrate the concept. In this case, the video is partitioned into four segments, namely, A, B, C and D of lengths d, 2d, 2d and 5d respectively. Each channel is periodically broadcast. Note that A, being a smaller segment, its broadcast is repeated more often compared to that of either B (C) or D. Clients prefetch data according to a schedule that ensures existence of video data in the playout buffer for the entire duration of video playback. Note how client buffer (disk) builds up for Clients 1 and 2 in the figure. Interestingly, using 4 dedicated channels, the latency is capped at d minutes.

The partition function used in GDB is given below:

$f_{GDB}(n) = 1$	for $n = 1$
$= 2$	for $n = 2, 3$
$= 5$	for $n = 4, 5$
$= 12$	for $n = 6, 7$
$= 5\ f(n-4)$	for $n > 7$

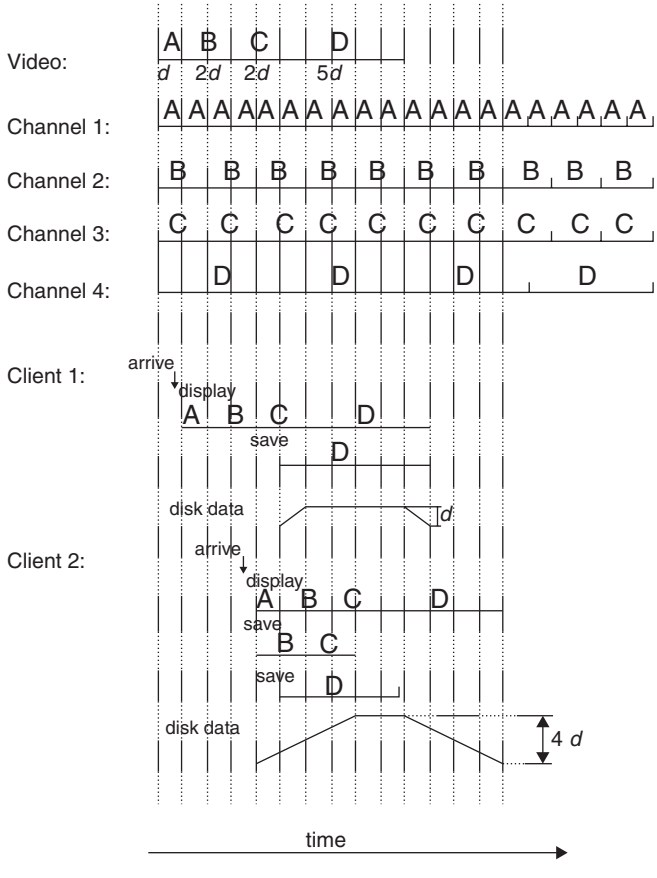


Figure 20.7 An example of GDB scheme.

The GDB scheme requires minimal server bandwidth O(log L) and provides an access latency of $L/\left(2.\sum_{m=1}^{K} f\,(m)\right)$ where K is the number of segments in which the video of length L is divided.

20.3 Hybrid Scheme

While open-loop schemes, such as, GDB provide the best possible server bandwidth O(log L) for hot videos, they perform miserably for so-called cold videos. On the other hand, closed-loop schemes, such as, Threshold-based multicast provide the best possible server bandwidth O (λL) for cold videos with very low arrival rate λ.

Ideally, an adaptive scheme is needed that will switch to GDB for hot videos and switch to Controlled Multicast for cold videos. The authors in [19] proposed a scheme that classifies a video as hot if GDB uses fewer multicast channels at the server for a video compared to Threshold-based multicast. Otherwise, the video is classified as cold. As an example, if the

desired goal is to have an access latency of 0.35 minutes, GDB requires nine channels for each video using the partition sequence 1, 2, 4, 4, 10, 10, 24, 24, 50. A threshold-based multicast scheme requires a server bandwidth of ($\sqrt{(2Ll+1)}-1$)) b) where l is the arrival rate of requests. Therefore, videos for which ($\sqrt{(2Ll+1)}-1$) b) > 9 are classified as cold and are scheduled using threshold-based multicast and videos for which ($\sqrt{(2Ll+1)}-1$) b) $<=9$ are classified as hot and are scheduled using GDB.

A hybrid scheme systematically combining the best of the breed from open-loop schemes (GDB) and closed-loop schemes (controlled multicast) and adaptively switching between the two based on whether the video is hot or cold provides the best performance for a generic video delivery system.

20.4 Summary

Video-on-demand (VoD) is poised to be one of the most lucrative services offered by a service provider to its customers. However, in order to offer the VoD service in a cost-effective manner, a service provider has to be cognizant of several resources including the I/O bandwidth at the video server, network bandwidth, I/O bandwidth at the client, storage space at the client while providing an optimal user experience in terms of quick start-up time and uninterrupted high-quality of video. In this section, we looked at closed-loop and open-loop systems of VoD where closed-loop systems enable video servers to schedule video streams based on client requests while in open-loop systems, video servers schedule video streams without taking into account the client requests. There are two broad categories of closed-loop systems: (i) client-initiated and (ii) client-initiated with prefetching (CIWP).

In the client-initiated multicast scheme, a client makes a request for a video and waits until that video is eventually multicast. When a channel becomes available, the server selects a batch of pending requests for a video according to some scheduling policy.

In a CIWP scheme, two clients that request the same video at different times can share a multicast stream; the earlier arrival client is not delayed. For two requests spaced a few minutes apart, the complete video is multicast by the server in response to the first request. The second request is satisfied by transmitting the first few minutes of the video while requiring the client to prefetch the remainder of the video from the first multicast group. Because the second client is few minutes behind, it continually buffers those few minutes of the video for much of the time. A CIWP scheme takes advantage of the resources (such as disk storage space and network bandwidth) at the client side to save server network-I/O bandwidth by multicasting segments of video. Several CIWP schemes have been described in this chapter.

Open-loop systems can also be categorized into broad classes: (i) server-initiated and (ii) server-initiated with prefetching.

In a server-initiated multicast scheme, the server multicasts video objects periodically via dedicated server network- I/O resources. Clients join an appropriate multicast group to receive the desired video stream. The simplest server-initiated scheme involves starting multicast sessions at fixed time intervals and transmit video clips regardless of whether there are clients interested in the video or not. Thus, this scheme guarantees a maximum service latency of intersession interval independent of the arrival time of the requests. Therefore, sever-initiated schemes are more efficient for hot videos that receive more requests than for cold videos that receive fewer requests.

In a server-initiated with prefetching (SIWP) scheme, a video object is divided into segments, each of which is multicast periodically via a dedicated multicast group. The client prefetches data from one or more multicast groups for later playback. Prefetching is done in such a way as to ensure the continuous playback of the video. A number of server-initiated schemes have been proposed. A SIWP scheme takes advantage of resources (for instance, disk storage space and network bandwidth) at the client end, and therefore significantly reduces the server network-I/O resources required. These schemes also guarantee a maximum service latency independent of the arrival time of the requests and perform better for hot videos than for cold videos.

Usually, the closed-loop systems perform well for low client request rates while the open-loop systems perform better for high client request rates. We also looked at a hybrid scheme that combines the best closed-loop scheme with the best open-loop scheme and applies the open-loop scheme for the "hot" videos and the closed-loop scheme for "not-so-hot" videos for an optimal overall system performance.

References

[1] Eager, D.L. and Vernon, M.K. (1998) *Dynamic Skyscraper Broadcasts for Video-on-Demand*. Proceedings of the 4th Int. Workshop Multimedia Information Systems (MIS'98) Istanbul, Turkey, Sept. 1998.

[2] Eager, D.L., Ferris, M.C. and Vernon, M.K. (1999) *Optimized Regional Caching for On-demand Data Delivery*. Proceedings of the Multimedia Computing and Networking (MMCN'99), San Jose, CA, Jan. 1999.

[3] Dan, A., Heights, Y. and Sitaram, D. (1996) *Generalized Interval Caching Policy for Mixed Interactive and Long Video Workloads*. Proceedings of the SPIE Conf. Multimedia Computing and Networking, San Jose, CA, Jan. 1996, pp. 344–351.

[4] Gao, L. and Towsley, D. (1999) *Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast*. Proceedings of the IEEE Int. Conf. Multimedia Computing and Systems, Florence, Italy, Jun. 1999.

[5] Aggarwai, C., Wolf, J. and Yu, P.S. (1996) *On Optimal Batching Policies for Video-on-Demand Storage Servers*. Proceedings of the International Conference Multimedia Systems'96, June.

[6] Dan, A., Sitaram, D. and Shahabuddin, P. (1994) *Scheduling Policies for an On-demand Video Server with Batching*. Proceedings of the ACM Multimedia Conference, Oct. 1994.

[7] Dan, A., Sitarami, D. and Shahabuddin, P. (1996) Dynamic batching policies for an on-demand video server. *Multimedia Syst*., **4**(3), 51–58.

[8] Golubchik, L., Liu, L.C.-S. and Muntz, R.R. (1995) *Reducing (I/O) Demand in Video-On-demand Storage Servers*. Proceedings of the ACM Sigmetrics 1995, pp. 25–36.

[9] De Bey, H.C. (1995) Program transmission optimization, Mar. 1995.

[10] Viswanathan, S. and Imielinski, T. (1996) Metropolitan area video-on-demand service using pyramid broadcasting. *ACM Multimedia Syst*, **4**(4): 197–208.

[11] Aggarwal, C.C., Wolf, J.L. and Yu, P.S. (1996) *A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand Systems*. Proceedings of the IEEE Int. Multimedia Computing and Systems, June 1996, pp. 118–126.

[12] Hua, K.A. and Sheu, S. (1997) *Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems*. Proceedings of the ACM SIGCOMM, Sept. 1997.

[13] Birk, Y. and Mondri, R. (1999) *Tailored transmissions for efficient near-video-on-demand service*. Proceedings of the IEEE International Conference Multi- media Computing and Systems, Florence, Italy, June 1999.

[14] Sheu, S., Hua, K.A. and Hu, T.H. (1997) *Virtual Batching: A New Scheduling Technique for Video-on-Demand Servers*. Proceedings of the 5th DASFAA Conf. Melbourne, Australia, Apr. 1997.

[15] Hua, K.A., Cai, Y. and Sheu, S. (1998) *Patching: A Multicast Technique for True Video-on-Demand Services*. Proceedings of the ACM Multimedia Conference, Bristol, U.K., Sept. 1998.

[16] Gao, L. and Towsley, D. (1999) *Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast*. Proceedings of the IEEE International Conference Multimedia Computing and Systems, Florence, Italy, Jun. 1999.

[17] Sen, S., Gao, L., Rexford, J. and Towsley, D. (1999) *Optimal Patching Scheme for Efficient Multimedia Streaming*. Proceedings of the NOSSDAV, June 1999.

[18] Carter, S.W. and Long, D.D.E. (1997) *Improving Video-on-Demand Server Efficiency Through Stream Tapping*. Proceedings of the International Conference on Computer Communication and Networks (ICCCN '97), IEEE, Las Vegas, September 1997, pp. 200–207.

[19] Gao, L. and Towsley, D. Threshold-based multicast for continuous media delivery. *IEEE Transactions on Multimedia*, **3**: 405–414.

[20] Viswanathan, S. and Imielinski, T. (1996) Metopolitan area video-on-demand service using pyramid broadcasting. *IEEE Multimedia Systems*, **4**, 197–208.

[21] Gao, L., Kurose, J. and Towsley, D. (1998) *Efficient Schemes for Broadcasting Popular Videos*. Proceedings of NOSSDAV, Cambridge, UK, July 1998.

[22] Ramesh, S., Rhee, I. and Guo, K. (2001) Multicast with cache (mcache): An adaptive zero-delay video-on-demand service. *IEEE Transactions on Circuits And Systems for Video Technology*, **11**(3), 440–456.

[23] Juhn, L.-S. and Tseng, L.-M. (1997) Harmonic Broadcasting for video-omdemand service. *IEEE Transactions on Broadcasting*, **43**(3), 268–271.

[24] Engebretsen, L. and Sudan, M. (2002) *Harmonic Broadcasting is Bandwidth-Optimal Assuming Constant Bit Rate*. Proceedings of Annual ACM-SIAM Symposium Discrete Algorithms 2002, san Francisco, CA, Jan 2002, pp. 431–432.

21

Challenges of Distributing Video in Mobile Wireless Networks

Video can be distributed in cellular networks in one of three ways: (i) full download, (ii) progressive download and (iii) streaming. Full download has the disadvantage of the initial wait time as downloading a video in its entirety takes time. However, it has the advantage of high-quality playback as the video is stored in the local disk and is played from there and is not subject to network impairments. Full download is suitable for non-real-time video. Streaming, on the other hand, works by filling out a playout buffer at the client (usually a few seconds) and playing back from the buffer as soon as the buffer fills up. The advantage of streaming is almost instant playback as only the playout buffer needs to be filled. However, because the content is streamed from the server across a network to the client, network impairments, such as, packet loss and jitter have significant impact on the quality of experience. Specifically, if the rate of transmission of the video across the network is lower than the playback rate, the playout buffer runs out resulting in freezing of the video until the streaming rate picks up and fills up the playout buffer again. Streaming is useful for live and/or real-time video delivery. A nice compromise between the two extremes (full download and play) and streaming is progressive download in which a portion of the video is first downloaded and stored and played back while the next segment of video is downloaded. Progressive download is suitable for real-time video delivery, and can potentially be used for live video delivery. Segments are downloaded in their entirety, so the initial wait time is limited by the time needed to download that segment. Moreover, since the video is played from the local disk, playback quality is excellent.

When multiple people simultaneously request the same video to be played, point-to-point (p-t-p) delivery of video is not economical in terms of bandwidth consumption. A better approach is to use point-to-multipoint (p-t-m) delivery whereby the source transmits the video once regardless of the number of recipients.

Conceptually, Figures 21.1 and 21.2 show the difference between unicast (one to one) and multicast (one to many) scenarios.

There is a subtle difference between multicast and broadcast. Broadcast means one to all while multicast means one to many. In case of broadcast, video is meant for all clients whereas in case of multicast, video is meant only for those who expressed interest in the given video.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

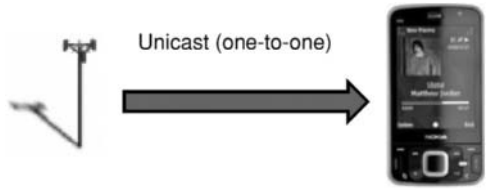


Figure 21.1 Unicast.

To support multicast in the cellular network, there is a need to define the architecture and protocols to set up multicast trees across the cellular network spanning the core as well as the radio access network. 3GPP defined multimedia broadcast multicast service (MBMS) in Release 6 [1–11, 13–24]. The architecture and protocols of MBMS are described next.

21.1 Multimedia Broadcast Multicast Service (MBMS)

Multimedia broadcast multicast service defines two service modes:

- Multicast: this is subscription-based meaning that the service is provided only to those subscribed to the service using the p-t-m delivery mechanism.
- Broadcast: this requires no subscription meaning that the service is provided to all users in a given service area without any subscription using the p-t-m delivery mechanism.

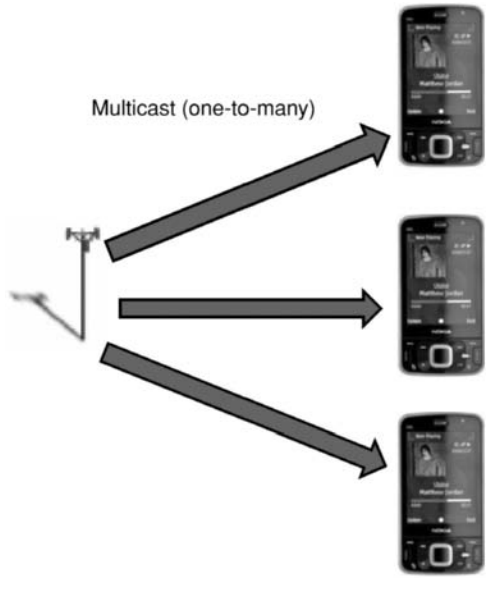


Figure 21.2 Multicast.

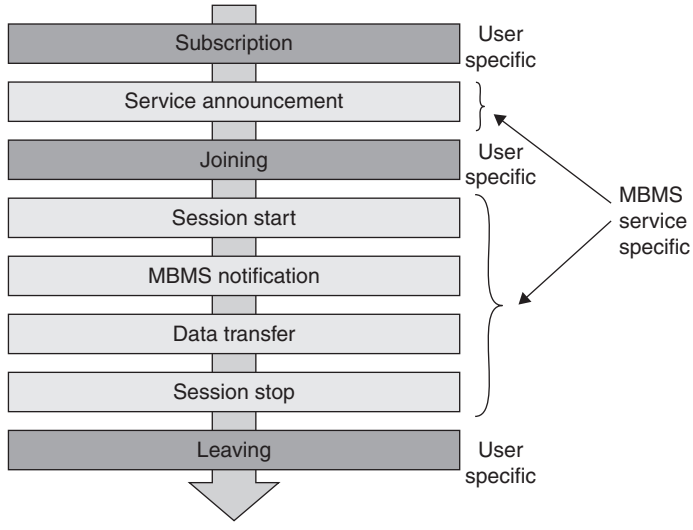


Figure 21.3 Phases of multicast service provision.

Multicast service provisioning happens in the following phases (Figure 21.3).

Note that among the steps shown in Figure 21.4, the steps of "Subscription", "Joining" and "Leaving" are user-specific meaning that these steps need to be followed only if a user is subscribed to a given multicast service.

In contrast, a broadcast service provisioning avoids the above three steps as shown in Figure 21.4.

The differences between broadcast and multicast are captured in Table 21.1.

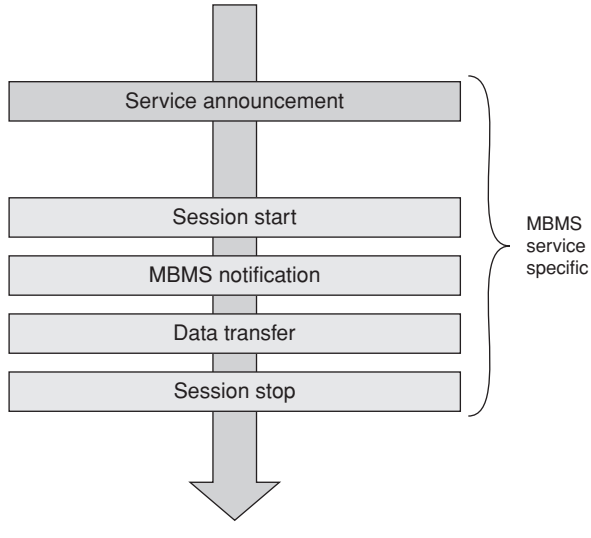


Figure 21.4 Phases of broadcast service provision.

Table 21.1 Broadcast and multicast compared.

Broadcast	Multicast
Broadcast service is a unidirectional point-to-multipoint service in which data is transmitted from a single sender to multiple receivers (user equipments) in the associated broadcast area	Multicast service is a unidirectional point-to-multipoint service in which data is transmitted from a single sender to multiple receivers (user equipments) that belong to a Multicast group in the associated Multicast area
Broadcast is a 'push' service. End users do not have to subscribe to be part of broadcast group	Multicast is a 'pull' service. End users have to subscribe to a Multicast group to receive data
Broadcast is free	Multicast can either be free or may involve subscription fee for the channel

21.1.1 MBMS User Services

21.1.1.1 Delivery Methods

MBMS user services support two types of delivery methods:

- streaming; and
- download.

Streaming uses IETF-specified RTP protocol with limited codec and RTP payload selection. Specifically, the following codes are supported:

- **Speech:**
  - The AMR codec will be supported for narrow-band speech.
  - The AMR wideband speech codec will be supported when wideband speech working at 16 KHz sampling frequency is supported.
- **Audio:**
  - MPEG-4 AAC low complexity object type should be supported. The maximum sampling rate to be supported by the decoder is 48 KHz.
  - The channel configurations to be supported are mono (1/0) and stereo (2/0).
  - In addition, the MPEG-4 AAC long-term prediction object type may be supported.
- **Still Image:**
  - ISO/IEC JPEG together with JFIF will be supported. The support for ISO/IEC JPEG only apply to the following two modes:
    - mandatory: baseline DCT, non-differential, Huffman coding;
    - optional: progressive DCT, non-differential, Huffman coding.
- **Video:**
  - For terminals supporting media type video, ITU-T Recommendation H.263 profile 0 level 10 will be supported.
  - This is the mandatory video codec for the MMS.
  - In addition, MMS should support:
    - H.263 Profile 3 Level 10.
    - MPEG-4 visual simple profile level 0.

For streaming, FEC code usage is recommended (Reed Solomon, low-density parity check (LPDC) and raptor codes).

For download, IETF-specified FLUTE protocol (see section 15.4.2.5) is used. File types are not limited. FEC code usage is recommended (Reed Solomon, low-density parity check (LPDC) and raptor codes).

21.1.1.2 Service Announcements

In order for users in MBMS to subscribe to and tune in to specific content channels, the available services need to be announced. This section describes how service announcements are done.

FLUTE is used for transporting a metadata envelope which:

- uses minimalist XML schema;
- provides versioning and time validity of metadata fragment;
- references metadata fragment;
- embeds metadata fragment of multiple syntaxes;
- uses application/sdp and application/xml.

A metadata fragment is a well defined "block" of metadata that is "uniquely" identified. It can have any number of attributes and be of any size. Furthermore, a metadata fragment contains:

- **user service description** (XML) for service identification and linkage to sessions / delivery methods;
- **session description protocol** (SDP) from IETF for FLUTE/RTP commonality and reuse;
- **associated delivery procedure description** (XML) for file repair and reception acknowledgment parameters.

User service description in a metadata fragment contains a unique service identifier and specifies:

- service type: streaming, messaging, and so on, so that the right application is launched in the UE;
- service-language(s), session identification;
- whether associated delivery procedure is on or off, and if on, it specifies the related parameters: for file repair and delivery reporting;
- whether security is on or off and if on, specifies the related parameters.

A **Session description protocol** (SDP) in metadata fragment specifies:

- QoS, data rates, UE MBMS bearer capability requirements;
- user service session start/stop time;
- destination port number, source and destination IP addresses, protocol, (delivery method);
- service-language (per medium);

- media types and codecs;
- whether FEC is on or off, and if on, specifies the related parameters;
- mode of bearer(s): broadcast/multicast.

Associated delivery procedure description in metadata fragment specifies:

- postdelivery file repair, which can be either p-t-p (interactive) repair or p-t-m repair;
- postdelivery reception reporting, which contains for both streaming and download:
  - statistical reports (for QoS reporting);
  - reception acknowledgment.

21.1.2 MBMS Architecture

Multimedia broadcast multicast service architecture was designed with the objective of maintaining compatibility with the broader Internet while not compromising the core functionalities of the cellular network:

- Multimedia broadcast multicast service architecture must enable efficient usage of network (radio-network and core-network) resources as well as radio resources. Specifically, multiple users should be able to share common resources when receiving identical traffic.
- Multimedia broadcast multicast service architecture should support both multicast and broadcast modes in a uniform manner, for example both modes shall preferably use the same low-layer bearer for data transport over the radio interface.
- Multimedia broadcast multicast service architecture does not describe the means by which broadcast/multicast service data is obtained by the operator. The data source may be external or internal to the operator's network. For example, the content servers could be in the Internet or could be hosted inside the operator's network.
- Multimedia broadcast multicast service architecture should be incremental to the existing 3GPP network architecture, meaning that the architecture should re-use, as much as possible, the existing 3GPP network components and protocol elements. The design goal was to minimize changes necessary to the existing infrastructure and provide a solution based on well-known concepts.
- Multimedia broadcast multicast service should provide a point-to-multipoint bearer (data plane) service for IP packets in the packet switched (PS) domain and not affect the circuit switched (CS) domain.
- Multimedia broadcast multicast service must interoperate with IP multicast and consequently support IP multicast addressing.
- User equipments (UEs) should be able to receive MBMS data in parallel to other services and signaling (for instance, paging, voice call).
- Multimedia broadcast multicast service areas should have the physical granularity of cells and logical granularity of individual service within the cell.
- Charging data for billing purposes must be provided per subscriber for MBMS multicast mode.

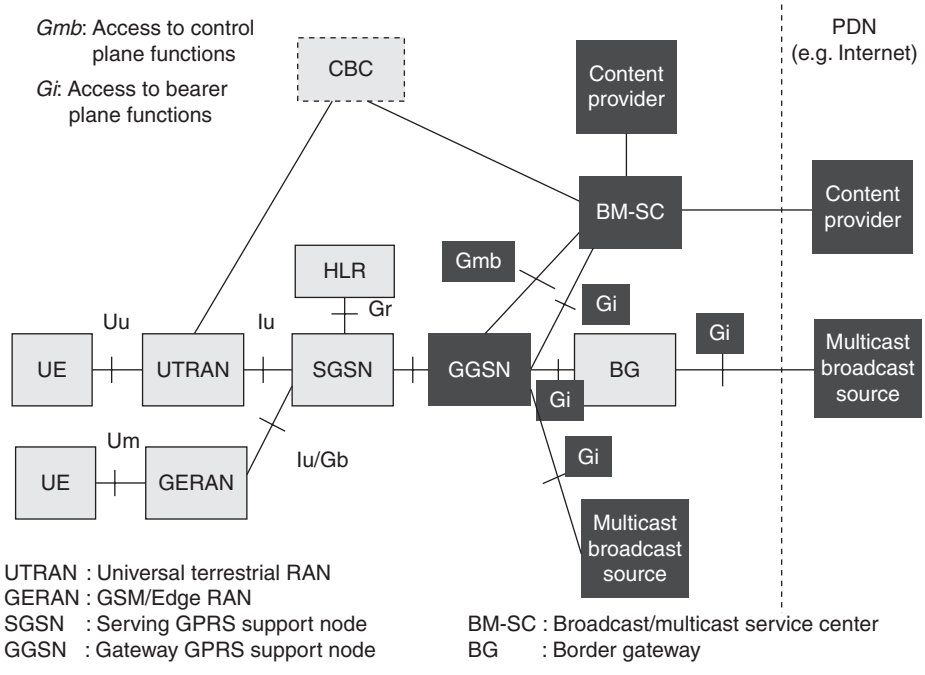


Figure 21.5 MBMS architecture.

Multimedia broadcast multicast services should not generate excessive signaling load on the network. MBMS architecture as defined in 3GPP Release 6 is shown in Figure 21.5.

Two new interfaces are introduced in a 3GPP network to support broadcast/multicast services: (i) Gmb: this is for accessing the control plane and (ii) Gi: this is for accessing the bearer plane. Existing packet-switched network entities, namely, GGSN, SGSN, UTRAN and user equipment (UE) need to be enhanced to provide the MBMS bearer (data path) service. Specifically, signaling is done through the Gmb interface to set up a multicast tree from the multicast broadcast source through the GGSN spanning the SGSNs and the Node Bs in the UTRAN. Once the multicast tree is set up, IP multicast datagrams are delivered from the multicast broadcast source from the Gi reference point to the UEs via the multicast tree.

The key components of the MBMS architecture are:

1. **The Broadcast Multicast Service Center (BM-SC)**: This is the entry point for content providers in a cellular network for providing broadcast/multicast service. Specifically, the BM-SC performs the following functions (Figure 21.6):
   a. User broadcast/multicast service announcements; media description using standard IETF protocols over MBMS bearer services.
   b. User service discovery.
   c. Authorization and initiation of MBMS bearer services in the operator's network.
   d. Multimedia broadcast multicast services delivery method functions:

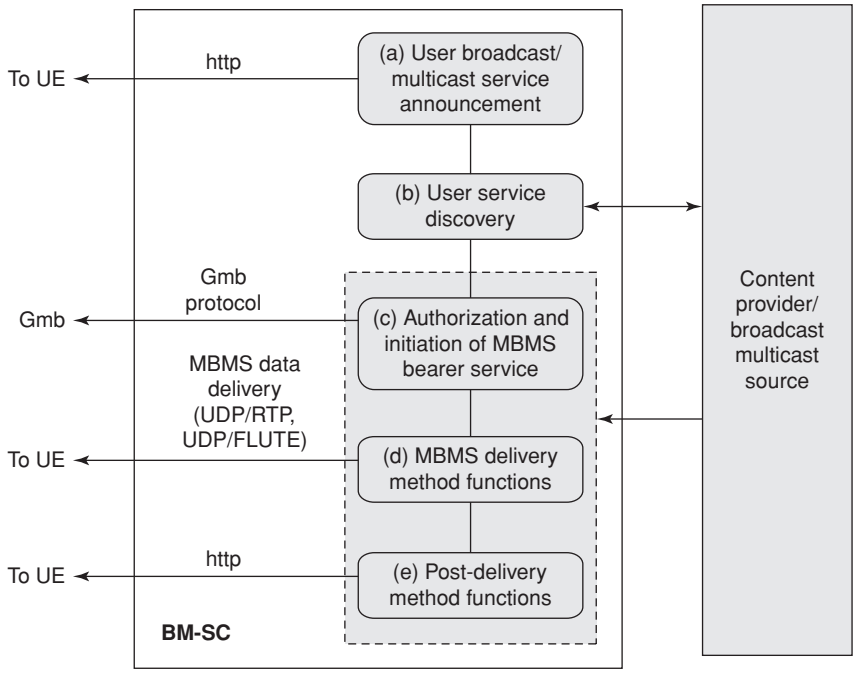


Figure 21.6 Functions of BM-SC.

i. Scheduling and delivery of MBMS transmissions.
ii. Scheduling session retransmissions; labeling sessions; doing error-resilient coding and transcoding.

e. Postdelivery functions:
i. Content provider charging by generating charging records.

2. **GGSN**: GGSN is not unique to MBMS rather a network element in the core network of a cellular operator. However, its functionality needs to be enhanced to support MBMS services. Specifically, GGSN terminates the MBMS GTP tunnels from the SGSN and links these tunnels via IP multicast with the MBMS data source. The GTP tunnels constitute the branches of the multicast tree in the radio access network and the core network of the cellular operator. Specifically, GGSN performs the following functions:
   a. Provides an entry point for IP multicast traffic that is MBMS data.
   b. Establishes/tears down MBMS bearer service upon getting notification from BM-SC.
   c. Maintains MBMS bearer and UE contexts.
3. **SGSN**: Just like GGSN, SGSN is not unique to MBMS rather a network element in the core network of a cellular operator. However, its functionality needs to be enhanced to support MBMS services. In the MBMS architecture, SGSN multiplexes all individual users of the same MBMS service into a single MBMS service. In addition, user–specific service control functions in the context of MBMS service are also performed by SGSN. SGSN maintains a single connection with the source of the MBMS data for a given

multicast session, thereby providing the desired efficiency in network resource utilization. Specifically, SGSN performs the following functions:
   a. Provides support for Inter/Intra SGSN mobility via MBMS context transfer.
   b. Transmits MBMS data to UTRANs.
   c. Performs user MBMS bearer control functions.
4. **UTRAN**: UTRAN's functionality needs to be enhanced to support MBMS services. UTRAN's main responsibility is to ensure radio-interface-efficient delivery of MBMS data to the designated service area. Specifically, UTRAN performs the following functions in the context of MBMS service:
   a. Counts users in a cell to determine whether to use p-t-p or p-t-m transmission mode.
   b. Provides support for Intra/Inter UTRAN mobility. Intra-UTRAN mobility involves change of Node Bs within the same UTRAN while inter-UTRAN mobility involves change of node Bs across UTRANs.
   c. Provides service announcements to UEs.
   d. Supports simultaneous MBMS data and voice service.
5. **MBMS user equipment (UE)**: this needs to support functions for activation/deactivation of MBMS bearer service. Specifically, UE performs the following functions in the context of MBMS service:
   a. Supports MBMS security functions.
   b. Receives MBMS user service announcements and paging information.
   c. Supports simultaneous services: voice and MBMS data.
   d. Stores MBMS data. However, this may involve DRM issues.
6. **MBMS-specific reference points**: two new reference points are created in the 3GPP architecture to support MBMS as shown in Figure 21.7:
   a. Gi: provides access to MBMS *bearer* plane. It defines the network side boundary of the MBMS bearer service from the data plane perspective.

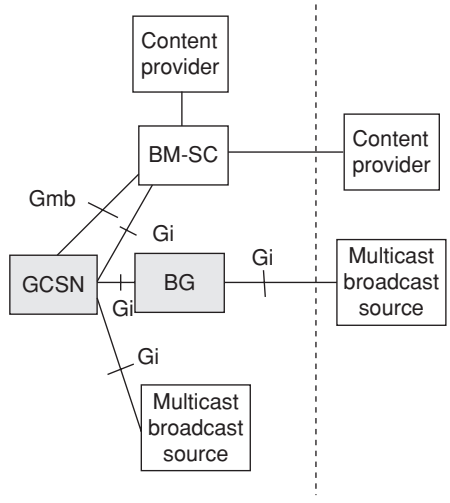


Figure 21.7 New reference points for supporting MBMS service in 3GPP architecture.

b. Gmb: provides access to MBMS *control* plane functions. It defines the Network side boundary of the MBMS bearer service from the control plane perspective. Specifically, it is through the Gmb interface that the following signaling functions are achieved:
   i. Signaling between BM-SC and GGSN.
   ii. MBMS bearer service-specific signaling.

21.1.3 MBMS Attributes and Parameters

In order to support MBMS services in a 3GPP cellular network, various information need to be maintained in the network in order to identify multicast groups uniquely and to offer the appropriate level of service to the respective multicast groups. These information elements are referred to as MBMS attributes and parameters and are categorized into three classes:

1. **TMGI (temporary mobile group identity)** is a globally unique identity allocated by BM-SC per MBMS bearer service. It is used for:
   a. MBMS notification purpose and communicated to UEs either during service activation (multicast service) or during service announcements (broadcast service).
   b. Radio resource efficient MBMS bearer service identification.
2. **MBMS UE context** contains UE-specific information related to a specific MBMS bearer service that user has joined; one UE context per MBMS bearer service that the user has joined:
   a. A MBMS UE context is created in UE, SGSN, GGSN and BM-SC when a user joins a MBMS bearer service.
   b. A MBMS UE context contains:
      i. IP multicast address.
      ii. APN (access point name).
      iii. TMGI (temporary mobile group identity).
      iv. International mobile subscriber identity (IMSI).
      v. Linked NSAPI (networks service access point identity).
      vi. MBMS-NSAPI.
3. **MBMS bearer context** contains all information regarding a MBMS bearer service:
   a. A MBMS bearer context is created in each node involved in the delivery of MBMS data.
   b. A MBMS bearer context contains:
      i. IP multicast address.
      ii. APN (access point name).
      iii. TMGI.
      iv. Required bearer capabilities.
      v. QoS.
      vi. MBMS service area.
      vii. List of downstream nodes.
      viii. Number of UEs.

ix. List of RAs (routing areas).
x. List of UEs in PMM-IDLE mode per RA.
xi. State of bearer plane (active/standby).

21.1.4 Multicast Tree in Cellular Network

The MBMS architecture as defined above enables a "continuation" of the IP multicast tree (from the Internet) into the network of the cellular operator. Figure 21.8 shows that existence of IP multicast tree originating from the broadcast/multicast source and extending up to the GGSN via the border gateway. The purview of MBMS begins beyond the GGSN extending all the way up to the UTRAN. The MBMS signaling helps set up the multicast tree in the core and access network of a cellular operator.

Other than setting up the MBMS tree over the point-to-point physical network architecture of the core and access networks, MBMS also defines efficient usage of the air-interface. The next section describes some of the extensions in the channels structure needed to accommodate the requirements of MBMS service.

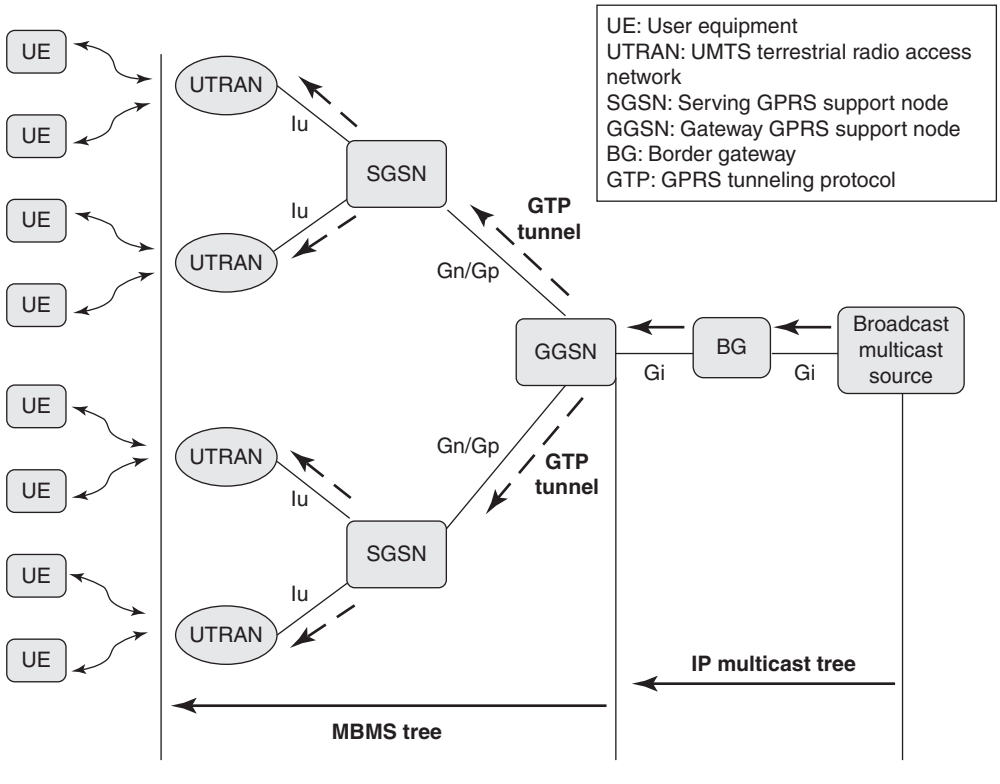


Figure 21.8 Multicast tree from the Internet extended through Cellular Network through MBMS.

21.1.5 MBMS Procedures

The following steps describe how an MBMS session is initiated and provisioned and what roles are played by the various components of the MBMS architecture described in the previous section:

a. The UE sends the IGMP/MLD a join request to the GGSN over a default PDP context where IGMP is Internet group management protocol and MLD is multicast listener discovery.
b. The GGSN contacts BM-SC to find whether the user (UE) is authorized to receive the bearer service.
c. For authorized users, appropriate signaling (GTP signaling) is used to create MBMS distribution tree and MBMS user and bearer contexts are established at the appropriate nodes. MBMS security procedures are used to authenticate the user.
d. Resources are allocated in the bearer plane when the MBMS session starts.

A typical timeline for MBMS service is shown in Figure 21.9 where two user equipments (UE1 and UE2) are shown to subscribe to multicast service 1 at different points in time. The service 1 announcement is shown to start right after UE1 subscribes to it. The UE1 listens to the service announcement, and joins the service. Once session for service 1 starts, after an idle period of few seconds, data transfer to UE1 begins. During that time, UE2 also joins the ongoing session and hence both UE1 and UE2 receive the data. After an idle period again, data transfer begins and since both UE1 and UE2 are in the session, they both receive the

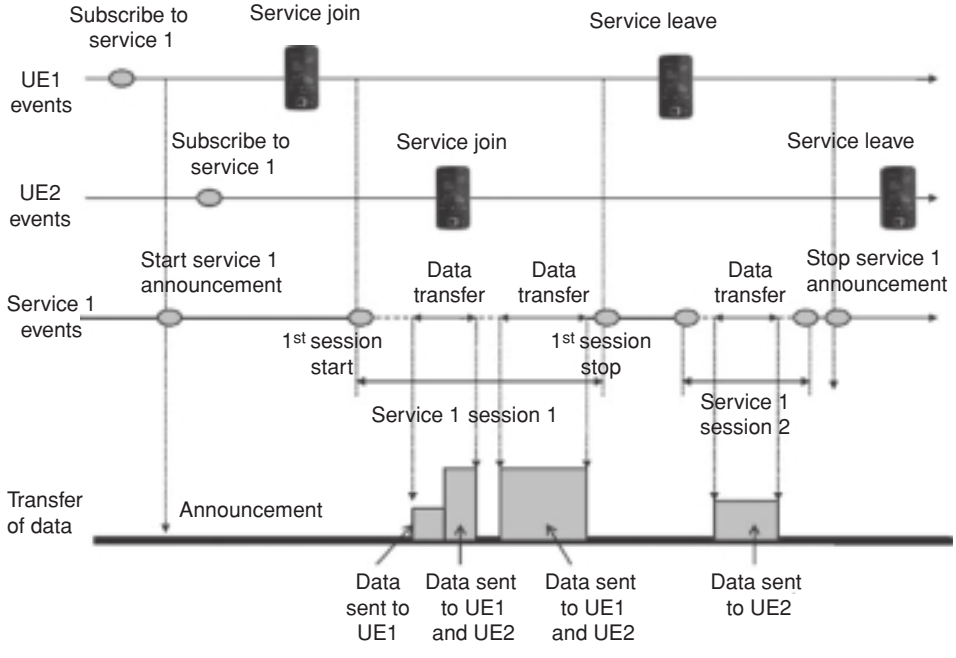


Figure 21.9 Timeline for a typical multicast service.

data. Session 1 stops and then after some time, UE1 leaves the service. Note that UE2 is still subscribed to service 1 and hence when a new session (session 2 for service 1) begins, UE2 receives the data. Session 2 of service 1 ends at some point followed by termination of service 1. UE2 leaves the service at that point.

21.1.6 MBMS Channel Structure

UMTS Terrestrial Radio Access (UTRA) frequency division duplexing (FDD) radio interface has logical channels, which are mapped to transport channels, which are again mapped to physical channels. Logical to transport channel conversion happens in the medium access control (MAC) layer, which is a lower sublayer in the data link layer (Layer 2).

Let's introduce some of the logical channels and transport channels that are relevant in the context of the MBMS service. UL and DL refer to uplink and downlink respectively.

Logical Channels:

- Dedicated control channel (DCCH) (UL/DL).
- Dedicated traffic channel (DTCH) (UL/DL).
- MBMS p-t-m control channel (MCCH) (DL).
- MBMS p-t-m traffic channel (MTCH) (DL).
- MBMS p-t-m scheduling channel (MSCH) (DL).

Transport Channels:

1. Data flow for DCCH, DTCH (UL/DL) can be mapped to a dedicated transport channel (DCH).
2. Data flow for DCCH, DTCH, MCCH, MTCH, MSCH is mapped to a forward access channel (FACH).
3. Control flow for DCCH, DTCH, MCCH, MTCH, MSCH (DL) is mapped to a downlink shared channel (DSCH).
4. Control flow for DCCH, DTCH, MCCH, MTCH, MSCH (UL) is mapped to a random access channel (RACH).

Physical Channels:

1. The data flow for DCH is mapped to a dedicated physical data channel (DPDCH).
2. The control flow for DCH is mapped to a dedicated physical control channel (DPCCH).
3. FACH is mapped to a secondary common control physical channel (SCCPCH).
4. DSCH is mapped to physical downlink shared channel (PDSCH).
5. RACH is mapped to a physical random access channel (PRACH).

21.1.7 Usage of MBMS Channel Structure

There are two transmission modes in the air interface to provide MBMS service:

- Point-to-point transmission (p-t-p).
- Point-to-multipoint transmission (p-t-m).

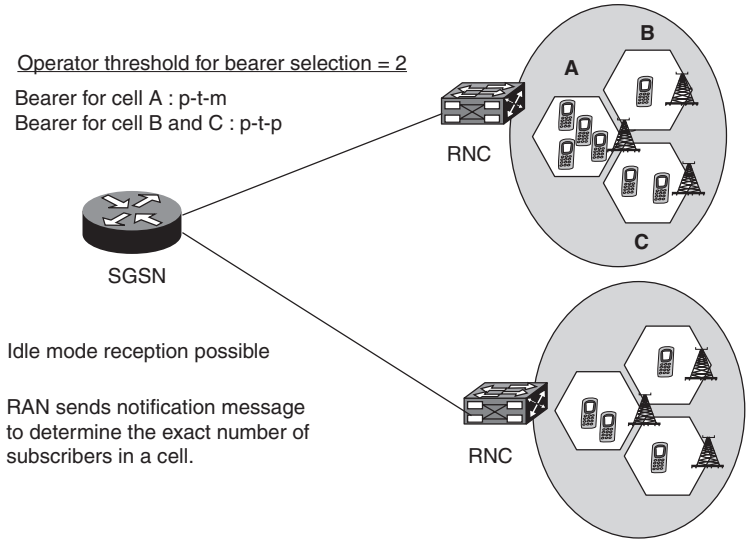


Figure 21.10 Dynamic choice of point-to-point vs. point-to-multipoint transmission mode in MBMS.

3GPP Release 6 provides the option to the UTRAN to dynamically choose between p-t-p and p-t-m transmission modes based on a threshold number of mobiles within a cell. Figure 21.10 depicts the concept.

Figure 21.11 shows an example scenario where one cell (cell on the left) uses p-t-m while another cell (cell on the right) uses p-t-p state as there is a single UE that has joined the corresponding multicast group. From the MBMS operation point of view, procedures are

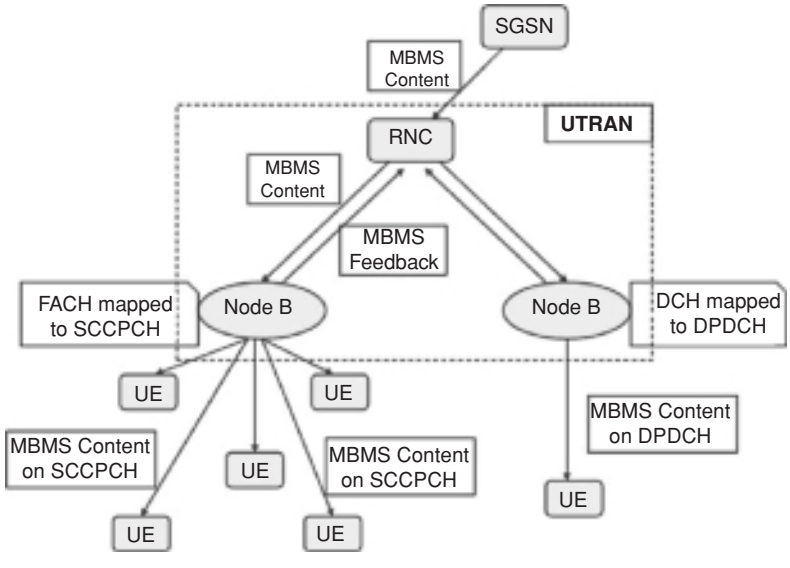


Figure 21.11 Usage of point-to-point and point-to-multipoint transmission in MBMS.

obviously simpler if the content is always provided in a point-to-multipoint manner without shifting users back and forth between different states.

21.1.7.1 Point-to-Point Transmission

Point-to-point transmission is used to transfer MBMS specific control/user plane information as well as dedicated control/user plane information between the network and the UE when there is only one UE in a multicast group.

Two logical channels are used in p-t-p transmission mode:

- Dedicated control channel (DCCH).
- Dedicated traffic channel (DTCH).

The DCCH and DTCH are mapped into transport channel DCH, which in turn is mapped into physical channels, dedicated physical data channel (DPDCH) and dedicated physical control channel (DPCCH), for data and control flow respectively.

21.1.7.2 Point-to-Multipoint Transmission

Point-to-multipoint transmission is used to transfer MBMS specific control/user plane information between the network and several UEs.

Three logical channels are used in p-t-m transmission mode:

- MBMS p-t-m control channel (MCCH).
- MBMS p-t-m traffic channel (MTCH).
- MBMS p-t-m scheduling channel (MSCH).

The MBMS point-to-multipoint Control Channel (MCCH) is used for a p-t-m downlink transmission of control plane information between network and UEs. The control plane information on MCCH is MBMS specific and is sent to UEs in a cell with an activated (joined) MBMS service.

The MCCH in the downlink (DL) is mapped into a transport channel called the forward access channel (FACH), which in turn is mapped into a physical channel called the secondary common control physical channel (SCCPCH).

The MBMS point-to-multipoint traffic channel (MTCH) is used for a p-t-m downlink transmission of user (data) plane information between network and UEs. The data plane information on MCCH is MBMS specific and is sent to UEs in a cell with an activated (joined) MBMS service.

The MTCH is always mapped to one specific forward access channel (FACH) (as indicated on the MCCH), which in turn is mapped into physical channel called secondary common control physical channel (SCCPCH).

The MBMS p-t-m scheduling channel (MSCH) is used for a p-t-m downlink transmission of MBMS service transmission schedule between network and UEs. The control plane information on MCCH is MBMS and S-CCPCH specific and is sent to UEs in a cell with an activated (joined) MBMS service.

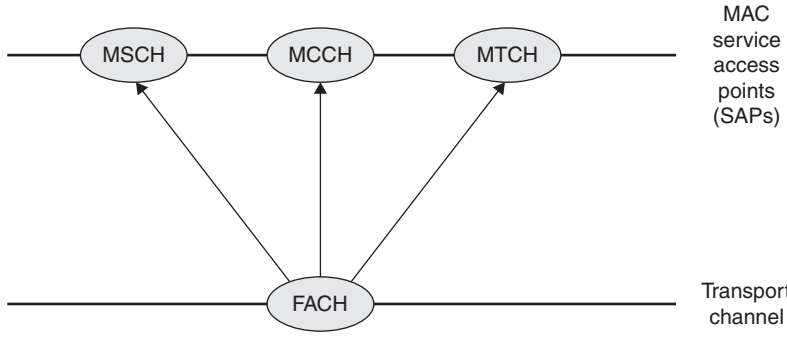


Figure 21.12 Logical channels mapped onto transport channel, seen from the UE side.

The MSCH is always mapped to one specific FACH (as indicated on the MCCH), which in turn is mapped into physical channel called the secondary common control physical channel (SCCPCH). Due to different error requirements the MSCH is mapped to a different FACH than MTCH.

The mappings as seen from the UE and UTRAN sides are shown in Figures 21.12 and 21.13 respectively.

21.1.8 MBMS Security

MBMS introduces the concept of a point-to-multipoint service into a 3GPP system. The broad security requirement is the delivery of MBMS data in a secured manner to a set of authorized users [26–31]. MBMS security specifications are given in TS 33.246. The main functions of MBMS security are:

- Authenticating and authorizing the user.
- Key management and distribution.
- Protection of the transmitted traffic.

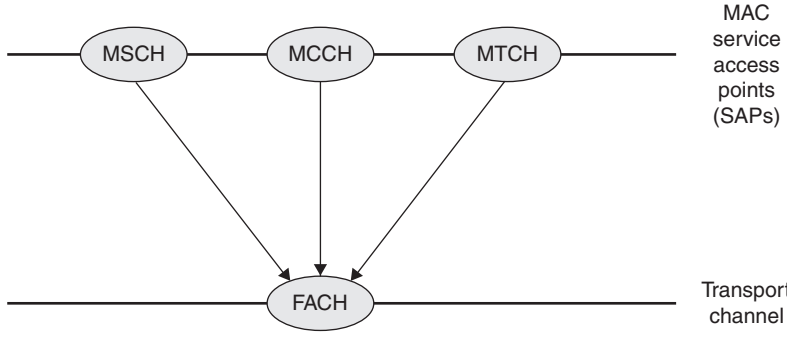


Figure 21.13 Logical channels mapped onto transport channel, seen from the UTRAN side.

21.1.8.1 MBMS Security Requirements

The main requirements of MBMS security are stated below:

- Reliable and secured distribution of keys should happen to registered users only.
- Group members should be able to authenticate that keys are being distributed by authorized entities.
- Rekeying should happen each time a new user joins a group, leaves a group, or key is compromised.
- Rekeying should happen periodically for added protection.
- Security issues for streaming and download should be handled separately.
- Digital rights management (DRM) issue with download should be handled properly.
- MBMS and DRM security mechanisms must be integrated into one system.

21.1.8.2 MBMS Security Highlights

The 3GPP MBMS security functionality is split between the BM-SC and the UE. BM-SC is responsible for user authorization and all key management functionalities-generation/distribution of keys for multicast security. The UE is responsible for receiving or fetching security keys from the BM-SC, securely storing keys, and for decrypting MBMS data for the multicast service. The BM-SC and UE use symmetric keys for security operations. Generic Bootstrapping Architecture (GBA) is used to establish a shared secret between UE and BM-SC. From the security perspective, the UE is split into two components: (i) ME (mobile equipment) and (ii) a universal integrated circuit card (UICC) that contains key management functions needed to implement GBA_U while ME is expected to support both GBA_U and GBA_ME while also utilizing key management functions on the UICC. Here are the main keys involved in MBMS security:

1. The MRK (MBMS request key) is:
   a. Used to authenticate the UE to the BM-SC when for example performing key requests.
   b. Generated simultaneously with MBMS User Key (MUK) using GBA (details in a later section).
   c. Stored in ME.
2. The MUK (MBMS user key) is generated along with MRK using GBA. It is:
   a. Used to protect point-to-point transfer of MSKs to the UE.
   b. Stored in ME or UICC.
3. The MSK (MBMS service key) is a shared secret between BM-SC and UE for accessing a particular service and is:
   a. Used to protect the delivery of MTKs.
   b. Sent to UE in Multimedia Internet Key (MIKEY) message (RFC 4738).
   c. Stored in ME or UICC.
4. MTK (MBMS traffic key) derived from MSK and is:
   a. Used to decrypt the received MBMS data on the ME.
   b. Sent to UE in MIKEY message.
   c. Stored in ME.

The function to derive MTK can be realized either in ME or UICC.

21.1.8.3 MBMS Authentication and Authorization

UE authentication and authorization in MBMS happens when:

1. UE wants to join a user service.
2. UE sends a request to establish a bearer to receive MBMS traffic.
3. UE requests and receives keys for the MBMS service.

Authentication during joining a user service is done using HTTP digest in which UE uses MRK as password.

Authentication during establishment of the MBMS bearer(s) to receive an MBMS user service is done using the authenticated point-to-point connection between UE and BM-SC.

Authorization for the MBMS bearer establishment happens by the network making an authorization request to the BM-SC to ensure that the UE is allowed to establish the MBMS bearer(s) corresponding to an MBMS User Service Authentication during requesting of MSK(s) as well as during post-delivery procedures is done using HTTP digest.

21.1.8.4 MBMS Key Generation

Here are the steps in generation of MBMS keys (Figure 21.14):

a. A UE that desires to receive a secure MBMS service must first use the generic bootstrapping architecture (GBA) [13] to establish a shared secret with the UMTS network for application specific purposes; this secret is then passed by the network to the BM-SC and is stored in UE.
b. Then the shared secret is used by both the UE and the BM-SC to derive two UE specific keys, the MBMS request key (MRK) and the MBMS user key (MUK).
c. The UE then initiates the user service registration procedure with the BM-SC, during which the two parties authenticate each other by using the MRK. If the UE has subscribed to the service, it is registered by the BM-SC as a key recipient.
d. The UE then performs the MSK request procedure, asking the BM-SC for the key of a specific MBMS service. The BM-SC sends this MBMS service key (MSK) to UE with the MSK delivery procedure, using the MUK to protect it. BM-SC may periodically send a new MSK to the UE so as to invalidate older keys.
e. UE decrypts the MBMS Service Key (MSK) using MUK.
f. The actual content is protected by a service specific MBMS traffic key (MTK). This key is distributed as a part of the content itself, using the MSK to protect it. MTK is transmitted over either a p-t-p or a p-t-m bearer to all UEs, unlike previous keys that are sent over p-t-p bearers to individual UEs. The MTK may also be periodically refreshed so as to invalidate older keys. BM-SC sends the MBMS data encrypted with MTK to ME.
g. In addition to sending the encrypted data to UE, BM-SC also sends a random number RAND and Key_id to UE. Key_id is needed to identify the MSK and other relevant information needed to compute MTK.
h. The UE generates MTK using RAND, MSK and other information.
i. The UE can then decrypt MBMS-data using MTK.

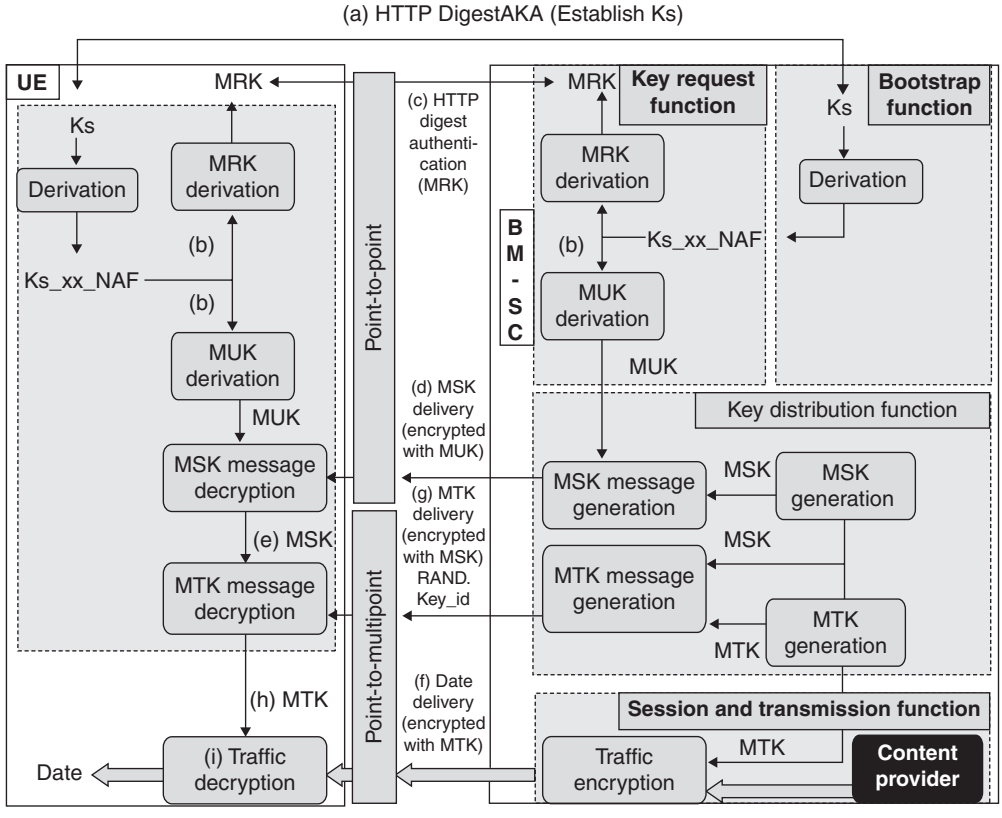


Figure 21.14 Generating keys for secure MBBS services.

Note that while MRK and MUK are "UE" specific, MSK and MTK are "service" specific and common to all UEs receiving a service. MSK is separately received at each UE protected by a MUK and it is used to recover the MTK received by all UEs. Each UE then uses the common MTK to decrypt the protected content.

To summarize, nearly all the security functionality for MBMS, except for the normal network bearer security, resides in either the BM-SC or the UE. The BSF (bootstrapping server function) is a part of GBA (generic bootstrapping architecture). The UE and the BM-SC use GBA to establish shared keys that are used to protect the p-t-p communication between the UE and the BM-SC.

The BM-SC is responsible for establishing shared secrets with the UE using GBA, authenticating the UE with the HTTP digest authentication mechanism, registering and deregistering UEs for MBMS user services, generating and distributing the keys necessary for MBMS security to the UEs with MIKEY protocol and for applying the appropriate protection to data that is transmitted as part of a MBMS user service. BM-SC also provides the MBMS bearer authorization for UEs attempting to establish MBMS bearer.

With MBMS, the same data may be delivered to many UEs over p-t-m bearers, therefore confidentiality and integrity have to be achieved on a one-to-many basis. This means that the

data source must share the same secret with many UEs. To prevent a UE from using this secret after leaving a service, the shared secret must be periodically updated.

21.2 Digital Video Broadcast – Handhelds (DVB-H)

Bandwidth in a cellular network is precious and it is currently used to offer voice and data services. Since the bandwidth requirement for providing video services is significantly higher, video services can very easily consume the bandwidth of the cellular network. In that case, the revenue generated from video services would have to exceed the revenue generated from the existing voice and data services to make economic sense for the operator to decide to do that. However, to satisfy the above-mentioned economic requirement, the price point for video services becomes unsustainable. In other words, the consumers will not be willing to pay for video at that price point. Thus, there is a dilemma. On one hand, the operators would like to provide video services to their customers and on the other hand, the price point for offering video services using their cellular network turns out to be infeasible. One way out of this dilemma is to deliver mobile video services to customers using an alternative delivery mechanism that uses a network other than the cellular network. Digital video broadcast – handhelds (DVB-H) is such an alternative mechanism to deliver video at high quality using higher bandwidth to user's handheld devices.

It is an extension of DVB-T (digital video broadcasting – terrestrial) with additional features customized for meeting the requirements of handheld, battery-powered devices [12]. DVB-H is aimed at providing mobile TV services with acceptable quality with a large geographic coverage even when moving at high speed. DVB-H is compatible with DVB-T providing a seamless integration and handover from terrestrial to mobile environment and vice versa. DVB-H uses time slicing technology to reduce power consumption. In fact, bursts of data are transmitted in small time slots and each burst may contain as much as 2MB including forward error correction (FEC) codes for robustness in one-way transmission channel. The hand-held device switches on only when the data for the selected service is available, stores the data in a buffer during the selected time slot, and plays it out from the buffer either after completely downloading the content or as it gets streamed as in the case of live content. Since the device is on only when receiving the desired content, it can save energy in proportion to the time it is off. Given that the receiver is inactive most of the time, it can save up to 90% power. DVB-H uses multi-protocol encapsulation (MPE)-forward error correction (FEC). An optional, multiplexer-level, forward error correction scheme means that DVB-H transmissions can be even more robust. This is advantageous when considering the hostile environments and poor antenna designs typical of handheld receivers.

A DVB-H works in three different bands as listed below, and can provide an aggregate data rate of 15 mbps that can support six video channels in addition to standard data channels:

- VHF-III (170–230 MHz).
- UHF-IV (470–862 MHz).
- L (1.452–1.492 GHz).

The data rates for audio and video in DVB-H system are shown in Table 21.2.

Table 21.2 Data rates in audio and video in DVB-H system.

Type of DVB receiver	h.264/avc level	Video resolution	Maximum bitrate	Typical application
A	1	QCIF (180*144)	128 kbits/sec	UMTS Telephone
B	1.2	CIF (360*288)	384 kbits/sec	UMTS Telephone, PDA
C	2	CIF (360*288)	2 Mbits/sec	Pocket Receiver
D	3	SDTV (720*576)	10 Mbits/sec	TV set
E	4	HDTV (1920*1080)	20 Mbits/sec	TV set

A typical architecture with both DVB-H and UMTS is shown in Figure 21.15. Note that the mobile terminal on the right has two receivers, one for DVB-H and the other for UMTS. DVB-H receiver will receive TV content from a broadcast network operator while the UMTS receiver will receive voice and data content from a wireless/Internet service provider. In fact, the UMTS network can be used as a reverse channel whereby user can interact with the TV content broadcast to it by the DVB-H network.

21.3 Forward Link Only (FLO)

Forward link only (FLO) technology is the equivalent of DVB-H in North America [25]. FLO technology was designed specifically for the efficient and economical distribution of the same multimedia content, such as a television channel, to millions of wireless subscribers simultaneously. It actually reduces the cost of delivering such content and enhances the user experience, allowing consumers to "surf" channels of content on the same mobile handsets they use for traditional cellular voice and data services. In designing FLO technology, Qualcomm

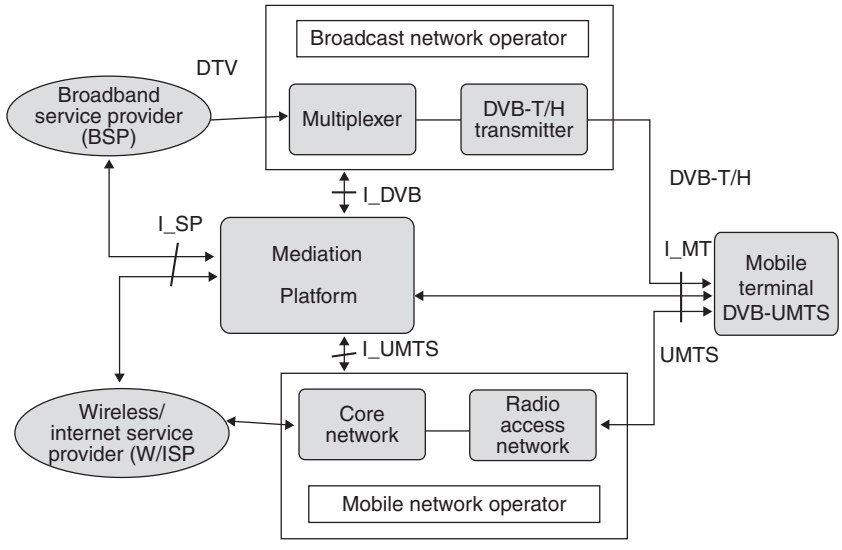


Figure 21.15 Combined DVB-H and UMTS network architecture.

has effectively addressed key challenges, namely lower cost and better quality, involved in the wireless delivery of multimedia content to mass consumers. Forward link only is usually combined with CDMA2000 EV-DO technology to enable an interactive multimedia streaming experience for the consumers. It is a proprietary solution of Qualcomm based on the same principal as DVB-H, meaning that it provides an out-of-band mechanism for distributing video to mobile handsets.

Forward link only can be deployed in a number of frequency bands utilizing various bandwidths and transmit power levels. The relative performance of a given modulation mode is defined by the choice of modulation, turbo and RS code rates. The frequency bands suitable for multicast distribution (including FLO technology) are similar to those used for unicast wireless IP and voice. These range from 450 MHz to 3 GHz. As video reception requires the device to be held in the hand rather than against the head it enables transmission at higher power level. This improves the performance in the PCS bands (1900 MHz) by 1–2 dB and in the cellular bands (800 MHz) by 3–4 dB. To maximize coverage area per cell and minimize the cost-per-bit delivered to the user, the design of a network supporting multimedia services benefits from higher power levels than those typically licensed for voice applications. In the FCC assigned licenses for 698–746 MHz in 6 MHz blocks for a variety of broadcasting, mobile and fixed services, with a maximum transmit power of 50 kW effective radiated power (ERP).

Forward link only has a longer range and higher available bandwidth compared to EV-DO and hence is the appropriate technology for large-scale video content distribution. In fact, FLO air interface is designed to support frequency bandwidths of 5, 6, 7 and 8 MHz. A highly desirable service offering can be achieved with a single radio frequency channel with 5 MHz allocation using TimeDivision Duplex (TDD). The air interface of FLO supports a broad range of data rates, ranging from 0.47 to 1.87 bits per second per hertz. In a 6 MHz channel, a FLO physical layer can achieve up to 11.2 mbps. The different data rates available enable tradeoffs between coverage and throughput.

Forward link only-based programming provides 30 frames-per-second (fps) a quarter video graphics array (QVGA) or 240 × 320 pixels with stereo audio. Currently, FLO programming includes 14 real-time streaming video channels of wide-area content (such as national content) and five real-time streaming video channels of local market-specific content. In addition, FLO programming provides 50 nationwide non-real-time channels (consisting of prerecorded content) and 15 local non-real-time channels, with each channel providing up to 20 minutes of content per day that can be delivered in the background seamlessly and made available for viewing on demand. Conceptually, a MediaFLO-enabled handset receives video "multicast"

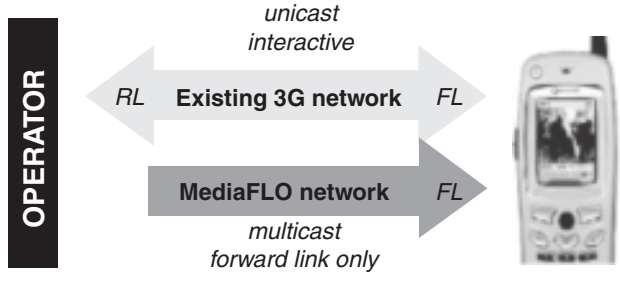


Figure 21.16 Co-existence of EV-DO (unicast) and FLO (multicast) network.

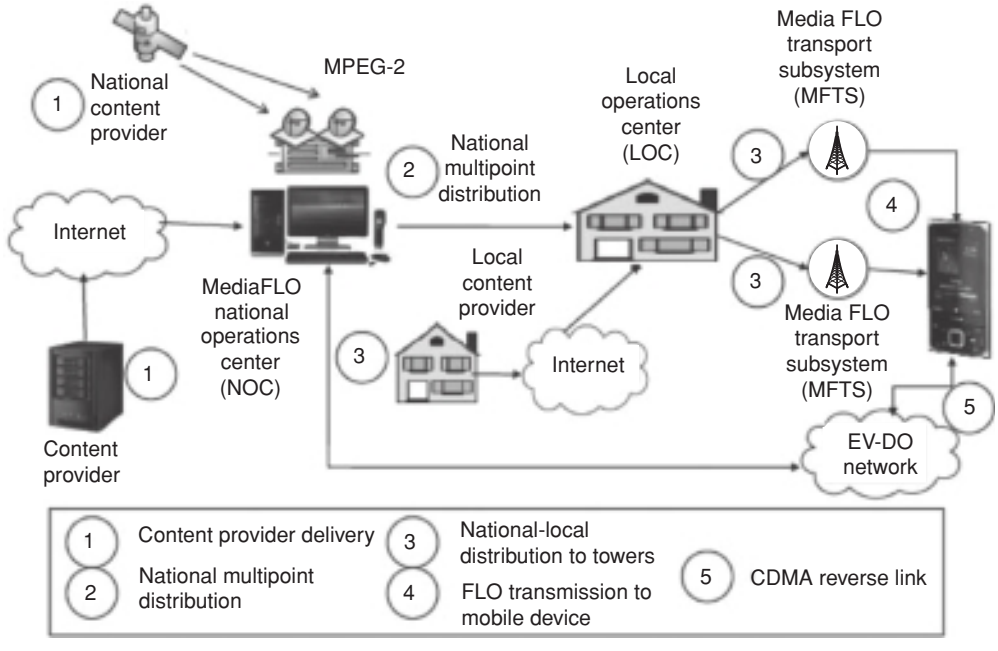


Figure 21.17 Typical FLO content delivery network.

from the MediaFLO network and "unicast" content from existing 3G network. It also uses the 3G network as the reverse channel to carry user interactions as shown in Figure 21.16.

In integrated EV-DO FLO network consists of five subsystems:

- National operation center (NOC).
- Local operations center (LOC).
- MediaFLO transport subsystem (MFTS).
- 3G (EV-DO) network.
- FLO-enabled devices (also known as MediaFLO Handsets).

Figure 21.17 shows a typical FLO content delivery network.

As shown in Figure 21.17, the NOC serves as an access point for national content providers, while LOC does the same for local content providers. They distribute wide area content and program guide information to mobile devices. The NOC also handles billing, content management infrastructure and distribution for the network. It also manages user-service subscriptions, the delivery of access and encryption keys, and provides billing information to cellular operators. The NOC typically includes one or more LOCs to serve as an access point from which local content providers can distribute local content to mobile devices in the associated market area.

The MediaFLO transport subsystem (MFTS) consists of FLO transmitters that transmit FLO waveforms to deliver content to mobile devices.

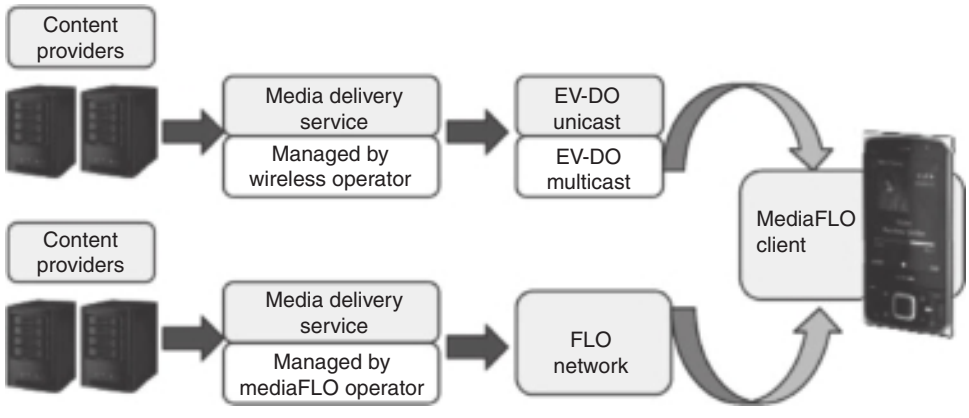


Figure 21.18 Integrated service model: FLO and EV-DO.

The 3G (EV-DO) Network belongs to the wireless operator(s) and supports interactive services to allow mobile devices to communicate with the NOC in order to facilitate service subscriptions and access key distribution.

Forward link only-enabled devices (as shown in the figure) receive FLO waveforms containing subscribed content services and program-guide information while also act as regular cellphones to serve as telephones, Internet access devices and as gaming consoles. The handsets are primarily used for making and receiving phone calls, so it is of paramount importance to preserve the battery power despite accessing TV and video content via the FLO network. In fact, FLO has been designed specifically to optimize power consumption through intelligent integration on the device and optimized delivery over the network.

In summary, video delivery to mobile handsets requires an integrated service model combining FLO and EV-DO (3G) network as shown in Figure 21.18.

21.4 Digital Rights Management (DRM) for Mobile Video Content

In this section, we describe the digital rights management (DRM) system as prescribed by the Open Mobile Alliance (OMA). The OMA DRM system enables content providers to associate permission with media objects that define how they should be consumed. The media objects controlled by the DRM system can be games, photos, music clips, video clips, streaming media, and so on. The content is distributed in an encrypted form and as a result, it is not usable without the associated rights object on a device. Essentially, the DRM system provides a mechanism for the users to purchase permissions embodied in rights objects associated with each media object. Naturally, the rights objects need to be handled in a secure and uncompromising manner. The DRM system is independent of the media object formats and the given operating system or run-time environment. The OMS DRMv2 architecture is shown in Figure 21.19.

The OMA DRMv2 architecture supports four distinct but related functions:

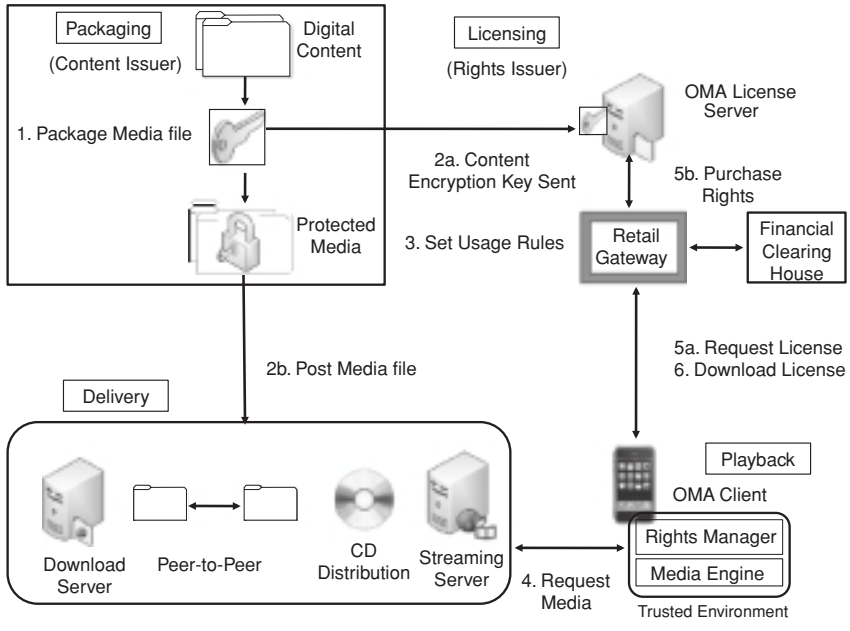


Figure 21.19 OMA DRMv2 architecture.

- **Packaging**: this applies to the content owner who needs to encrypt its digital content using an encryption key (step 1 in Figure 21.19).
- **Licensing**: this applies to the rights issuer who uses an OMA License Server to store the content encryption key received from the content owner (step 2a). There is also a retail gateway that contains the usage rules for each media object (step 3) and also interfaces with a financial clearing house for the actual transaction when the end-user buys rights to a media object.
- **Delivery**: this applies to the content distributor who stores the encrypted content in a download server, streaming server, peer-to-peer content delivery servers or in CDs (step 2b).
- **Playback**: this applies to the consumer who wants to consume content in a legal and rightful manner. When the OMA client residing on the consumer's mobile handset requests a media object (step 4) from the content distributor, the encrypted content is downloaded on to the handset and the client is simultaneously redirected to the retail gateway for requesting the license corresponding to the content (step 5a). After the necessary payment associated with the acquisition of the license (step 5b), the license is downloaded on to the mobile handset (step 6). Once the license is available, the rights manager signals the media engine to decrypt the content and play it.

21.5 Summary

In this chapter, we looked at how video content, including real-time television content and non-real-time on-demand video content can be distributed in an efficient and cost-effective

manner to mobile handsets. Video can be distributed in cellular networks in one of three ways: (i) full download, (ii) progressive download and (iii) streaming. Full download has the disadvantage of the initial wait time as downloading a video in its entirety takes time. However, it has the advantage of high-quality playback as the video is stored in the local disk and is played from there and is not subject to network impairments. Full download is suitable for non real-time video. Streaming, on the other hand, works by filling out a playout buffer at the client (usually a few seconds) and playing back from the buffer as soon as the buffer fills up. The advantage of streaming is almost instant playback as only the playout buffer needs to be filled. However, since the content is streamed from the server across a network to the client, network impairments, such as packet loss and jitter have significant impact on the quality of experience. Specifically, if the rate of transmission of the video across the network is lower than the playback rate, the layout buffer runs out resulting in freezing of the video until the streaming rate picks up and fills up the playout buffer again. Streaming is useful for live and/or real-time video delivery. A good compromise between the two extremes (full download and play) and streaming is progressive download in which a portion of the video is first downloaded and stored and played back while the next segment of video is downloaded. Progressive download is suited for real-time video delivery, and can be potentially used for live video delivery. Segments are downloaded in their entirety, so the initial wait time is limited by the time needed to download that segment. Moreover, as the video is played from the local disk, playback quality is excellent.

When multiple people simultaneously request the same video to be played, p-t-p delivery of video is not economical in terms of bandwidth consumption. A better approach is to use multicast using p-t-m delivery, whereby the source transmits the video once regardless of the number of recipients.

On one hand, 3GPP has defined a standard called MBMS for setting up multicast trees in cellular 3G networks and on the other hand, overlay networks, such as, DVB-H and FLO have gained popularity for distribution of video content in an out-of-band manner to the same handsets. DVB-H and FLO technology help reduce the cost of distribution of video content by using more powerful transmitters covering large areas and by not using up the network capacity of the cellular network. The state-of-the art broadcast/multicast systems in mobile networks combines both mechanisms such that DVB-H/FLO is used for distributing mobile video content and the 3G network is used for providing the reverse channel to facilitate interaction with the content. The MBMS standard also defines a security architecture for protecting multicast video content from non-subscribers if the content is not free. OMA defines a DRM system for protecting the rights of the content owners when copyrighted content is distributed to consumers.

References

[1] 3GPP (2004) Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1. TS 22.246; V6.2.0.
[2] 3GPP (2005) Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs. TS 26.346; V6.2.0.
[3] 3GPP (2004) Multimedia Broadcast/Multicast Service (MBMS); Stage 1. TS 22.146; V6.6.0.
[4] Koodli, R. and Puuskari, M. (2001) Supporting packet-data QoS in next-generation cellular networks. *IEEE Communications*, **39**(2), 180–188.
[5] 3GPP (2005) Multimedia Broadcast/Multicast Service (MBMS) user services; Architecture and functional description. TS 23.246; V6.8.0.

[6] Xylomenos, G. (2005) Group Management for the Multimedia Broadcast/Multicast Service. Proceedings of the 14th IST Mobile Summit, 2005.
[7] 3GPP (2005) Introduction of the Multimedia Broadcast Multicast Service (MBMS) in the Radio Access Network (RAN); Stage 2. TS 25.346;V6.6.0.
[8] 3GPP (2005) Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture. TS 33.220; V6.6.0.
[9] 3GPP (2005) Security of Multimedia Broadcast/Multicast Service. TS 33.246; V6.4.0.
[10] Zhang, B. and Mouftah, H.T. (2003) Forwarding state scalability for multicast provisioning in IP networks. *IEEE Communications*, **41**(6), 46–51.
[11] Diot, C., Levine, B.N., Lyles, B. *et al.* (2000) Deployment issues for the IP multicast service and architecture. *IEEE Network*, **14**(1), 78–88.
[12] Faria, G., Henriksson, J.A., Stare, E. and Talmola, P. (2006) DVB-H: Digital broadcast services to handheld devices. *Proceedings of the IEEE*, **94**(1), 194–209.
[13] Parkvall, S., Englund, E., Lundevall, M. and Torsner, J. (2006) Evolving 3G mobile systems: broadband and broadcast services in WCDMA. *IEEE Communications*, **44**(2), 68–74.
[14] Xylomenos, G., Vogkas, V. and Thanos, G. (2008) The multimedia broadcast/multicast service. *Wireless Communications and Mobile Computing*, **8**(2), 255–265
[15] 3GPP TS 22.146. (2004) Multicast Broadcast Multimedia Service (MBMS)-Stage 1.
[16] 3GPP TS 22.246. (2004) MBMS User Services.
[17] 3GPP TS 23.246. (2004) Multimedia Broadcast/Multicast Service (MBMS); Architecture and Functional Description (Release 6).
[18] 3GPP TS 25.346. (2004) Introduction of Multimedia Broadcast Multicast Service (MBMS) in RAN.
[19] 3GPP TS 26.346. (2004) MBMS teleservice codecs and protocols.
[20] 3GPP TS 33.246. (2004) Security of Multimedia Broadcast/Multicast Service.
[21] 3GPP TR 29.846. (2004) Multimedia Broadcast Multicast Service; CN1 Procedure Description (Rel-6).
[22] 3GPP TR 25.803. (2004) S-CCPCH performance for MBMS; (Release 6).
[23] 3GPP TR 25.992. (2004) Multimedia Broadcast Multicast Service (MBMS); UTRAN/GERAN Requirements.
[24] 3GPP TR 23.846. (2004) Multimedia Broadcast/Multicast Service; Architecture and Functional Description.
[25] FLO Technology Overview. http://www.qualcomm.com/common/documents/brochures/tech_overview.pdf. (accessed June 13, 2010).
[26] OMA Digital Rights Management V2.0 http://www.openmobilealliance.org/technical/release_program/drm_v2_0.aspx (accessed June 13, 2010).
[27] Hawkes, P. and Subramanian, R. (2003) Explanation of BAK-based key management, Tech. Rep. 3GPP S3-030040, QUACOMM, France.
[28] 3rd Generation Partnership Project (2002) Network architecture; release 5, Tech. Rep. 3GPP TS 23.002 V5.9.0, Dec.
[29] Wallner, D., Harder, E. and Agee, R. (1999) Key management for multicast: Issues and architectures, RFC 2627, June.
[30] Wong, C.K., Gouda, M. and Lam, S.S. (2000) Secure group communications using key graphs. *IEEE/ACM Transactions on Networking*, **8**(1), 16–31.
[31] Sun, Y., Trappe, W. and Liu, K.J.R. (2002) An Efficient Key Management Scheme for Secure Wireless Multicast. IEEE International Conference on Communications, 2002, pp. 1236–1240.

22

IP Multimedia Subsystem (IMS) and IPTV

The traditional Communication Service Providers (CSPs) network was based on circuit-switched technology, which was architected for providing voice services. Value-added voice services, such as toll-free (800 number) calling, required the introduction of intelligent network (IN) or advanced intelligent network (AIN) [14, 15] components in the architecture. However, due to the economic benefit of the packet-switched network and for the ease of interoperability with the Internet, CSPs switched from circuit-switched networks to packet-switched (Internet protocol (IP)-based) networks. With the evolution of the Internet from a mere data network to a multimedia network, a tremendous amount of activities started happening in the Internet engineering task force (IETF) community, which is a standards committee for the Internet and novel protocols, such as, session initiation protocol (SIP) [16] and a plethora of surrounding protocols started evolving in the Internet community enabling rich multimedia services on the Internet. The implications of such development in the Internet community were severe for CSPs – it threatened their existence by reducing them to "connectivity" providers with the entire intelligence residing outside the CSPs network for providing value-added services. Application service providers (ASPs) can now provide value-added multimedia services to the end-users (the so-called customers of the CSPs) without any help from CSPs other than "connectivity".

The competitive threat from ASPs, with dire business consequences for CSPs, made them think of alternatives to their existing circuit-switched network with IN that simply did not have the richness and flexibility of IP-based service offerings. IP Multimedia System (IMS) is the genesis of that thought process and is indeed the major strategic shift from CSP perspective [1–9]. The IMS architecture went on to leverage the IETF protocols, such as SIP as the core signaling protocol on top of their IP-based transport network while re-architecting the other components in a traditional CSP network including the core switching system and the repository of subscribers (home location register (HLR)), so that the newly defined components can deliver at least the same set of services, if not more, as compared with those offered by the ASPs while retaining the full control of the subscribers. IMS actually helps CSPs counter the threat of ASPs.

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

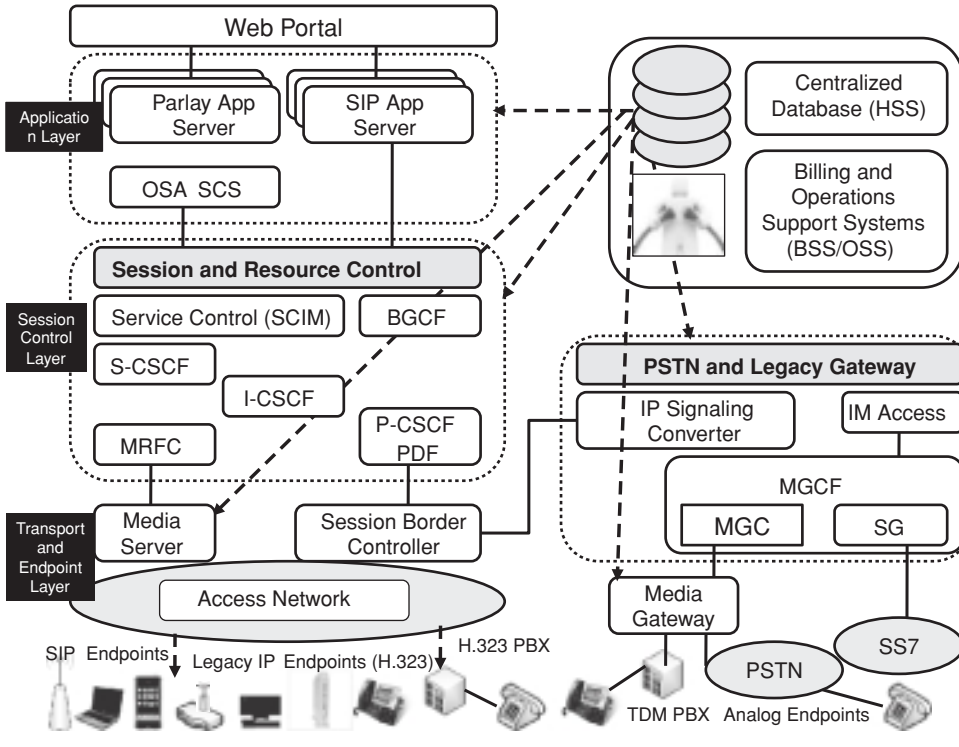


Figure 22.1 Functional architecture of IP multimedia subsystem (IMS).

22.1 IMS Architecture

Figure 22.1 schematically represents the functional architecture of IMS.

The core of IMS architecture is the session and resource control block, which constitutes the main switching fabric of the CSP's network and provides support for maintaining sessions (such as voice conversations, video telephony and multimedia conferencing) within the network [1–9]. It leverages the support of the transport and endpoint layer for delivering media and content of a session among the endpoints. The application layer consists of application servers (Parlay and SIP servers) that provide value-added services to the end users while interfacing with the session control layer via the session and resource control block. The entire system is dependent on a support system of centralized database and a billing/operations support system (BSS/OSS) consisting of billing mediation, operations, administration, maintenance and provisioning (OAM&P).

22.1.1 Layering on IMS Architecture

Before the advent of IMS architecture, applications for CSPs, such as, instant messaging (IM), voice over IP (VoIP), video-on-demand (VoD) and so on had to be developed independently even though they shared a lot of commonality (Figure 22.2). For example, all of the above

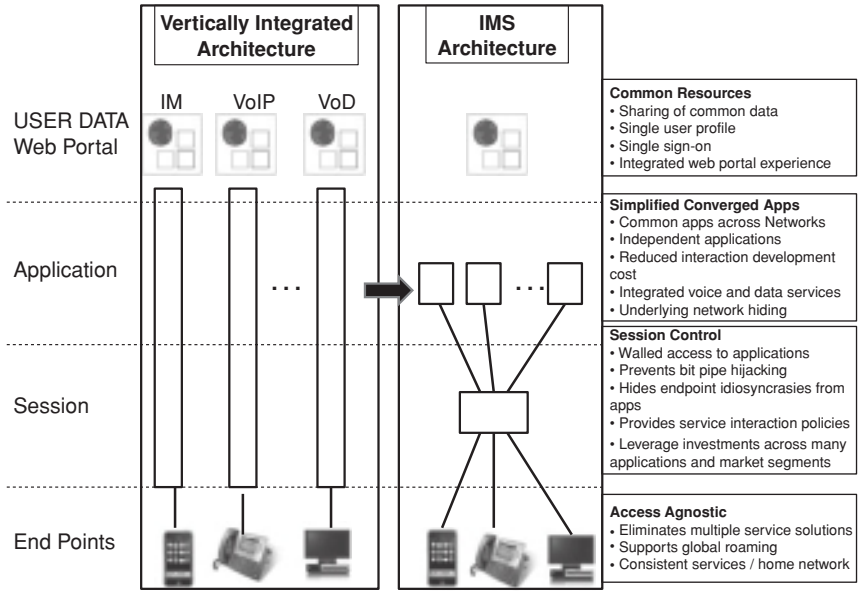


Figure 22.2 Layering of IMS architecture.

applications would most likely share a common user data base and go through the steps of session set up, maintenance and teardown. The IMS extracts the commonality between applications so that the common aspects can be shared among them. As shown in Figure 22.2, IMS architecture neatly encapsulates the functionalities into four categories: (i) common resources, (ii) converged applications, (iii) session control and (iv) access agnostic functions. Common resources, such as the user data base and portal together with single sign on can be shared among the applications. Session control and management can be shared across applications. Access agnostic support such as global roaming and consistent service across networks is also independent of the application. Only application-specific things are left for the applications to implement, thereby simplifying development of converged applications. This concept is referred to as the layering of IMS architecture.

22.1.2 Overview of Components in IMS Architecture

22.1.2.1 Session and Resource Control Entities

Session control entities shown in Figure 22.3:

- Provide walled access to applications.
- Provide application interaction control.
- Provide centralized network routing.
- Simplify application development.
- Hide endpoint idiosyncrasies from apps.

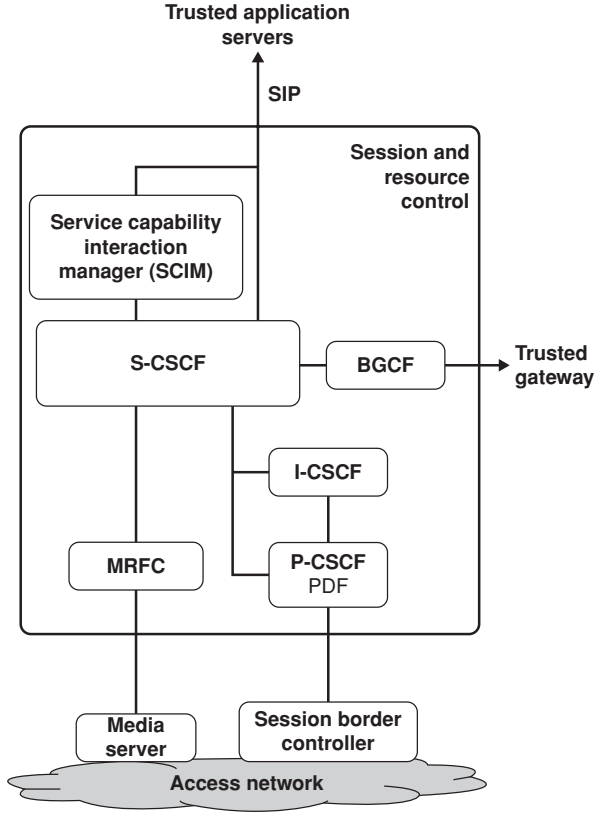


Figure 22.3 Session and resource control entities.

Here are the main session control entities:

1. Service capability interaction manager (SCIM) orchestrates service delivery among application servers within the IMS architecture.
2. Call session control function (CSCF) has three variations:
   a. S-CSCF – serving: session control entity for endpoint devices.
   b. I-CSCF – interrogating: entry point to IMS from other networks.
   c. P-CSCF – proxy: entry point to IMS for devices.

Functionally CSCFs follows the Internet paradigms:

   a. P-CSCF → I-CSCF → S-CSCF.
   b. Stateless entities at network edge, stateful entities in core.
   c. Simple processing at edge, complex processing in core.
   d. Security and authentication requirements increase towards core.

3. BGCF – Breakout gateway control function:
   a. Selects network to use for PSTN/PLMN interworking.
4. PDF – Policy decision function:
   a. Authorizes QoS requests.

Resource control entities:

- Provide centralized media resource control.
- Provide efficient scalable infrastructure.
- Help leverage investments across many applications and market segments.

The main resource control entity is:

- Multimedia resource function controller MRFC.

This controls the media server (multimedia resource function processor or MRFP).

22.1.2.2 Support Systems

Support systems in IMS consist of two subsystems:

- Home subscriber server (HSS) or "hss collective". Consists of AAA and databases.
- Billing and operations support system BSS/OSS. Consists of billing mediation, operations, administration and provisioning (OAM&P).

The HSS is the master database in the IMS architecture, which may be viewed logically as a single entity although physically it can be made up of several physical databases. It consists of:

1. The AAA server, which provides IP based authentication, authorization, and accounting functions.
2. Databases that contain the following information:
   a. Subscriber profiles. Subscriber specific information that is used for service and feature authorization.
   b. Dynamic subscriber information. Current session registration data (S-CSCF address, access network). Network policy rules. Subscription resource usage, QoS, valid times and routes, geographical service area definitions, policy rules for the applications serving a user, and so on.
   c. Equipment Identity Register (EIR). Information such as records of stolen equipment.

22.1.2.3 Application Layer Entities

Application layer entities are needed for supporting IMS-based applications. They mainly consist of application servers, which provide services and applications. The main components are:

- Session initiation protocol (SIP) application servers.
- Parlay application servers.
- Open service access (OSA).
- Service capability server (SCS) and OSA AS.

Advanced intelligent networking (AIN). Interworking servers also reside in this layer.

22.1.2.4 Transport and End Point Layer Entities

Figure 22.4 shows the relationship of various multimedia resources and controllers vis-à-vis the session control system:

1. PSTN and legacy gateway:
   a. Media gateway (MGW) is a component in the bearer path that provides interworking between RTP/IP and PCM bearers.
   b. Media gateway control function MGCF is a component in the signaling path.
2. SIP and other IP endpoints:
   a. Multimedia resource function processor (MRFP) (also known as media server) is a component in the bearer path that:
      i. Mixes incoming media streams (for example, for multiple parties).
      ii. Sources media streams (for multimedia announcements).

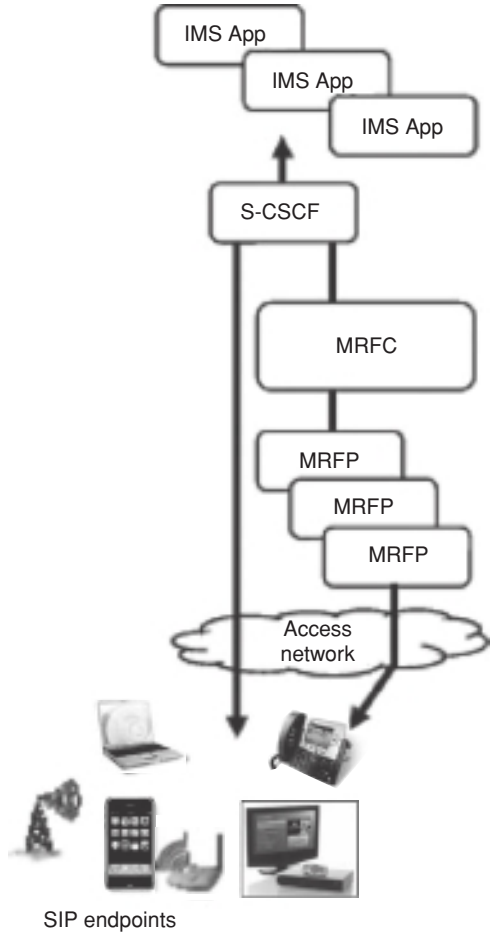


Figure 22.4 Multimedia resources in IMS.

iii. Processes media streams (for example, audio transcoding, media analysis).
iv. Provides tones and announcements.
v. Supports DTMF within the bearer path.

b. Session border controller is a component in the signaling path that
i. Provides support for setting up, conducting and tearing down interactive media sessions.
ii. Maintains full session state.
iii. Provides security to the network transport and end devices by protecting them from attacks, such as denial of service.
iv. Provides QoS via a policy-based framework for prioritization of flows.

22.1.3 Some Important Components in IMS Architecture

22.1.3.1 Proxy CSCF (P-CSCF)

Proxy CSCF (P-CSCF) acts as the security element at edge of IMS network providing initial entry point for user equipment (Figure 22.5).

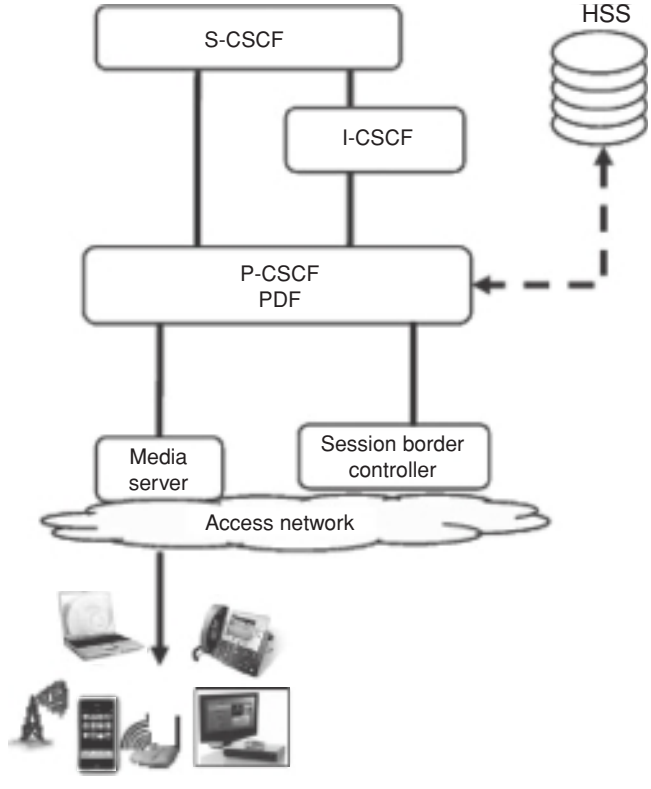


Figure 22.5 Proxy CSCF (P-CSCF).

Proxy CSCF (P-CSCF) is the first contact point within the IMS for the subscriber. It has well known address(es) within the network and can either be statically configured or discovered via DHCP. It provides the following functionalities:

1. Authentication and authorization:
   a. It routes incoming requests based on registration status:
      i. Sends the SIP REGISTER request received from the UE to an I-CSCF determined using the home domain name, as provided by the UE.
      ii. Sends SIP messages received from the UE to the SIP server S-CSCF, whose name the P-CSCF has received as a result of the registration procedure.
      iii. Rejects unauthorized requests.
   b. Authorizes the bearer resources for the appropriate QoS level via Policy Decision Function (PDF).
2. SIP compression and decompression.
3. It acts as a stateful SIP proxy:
   a. It generates CDR events.
   b. It can act as user agent and terminate calls in abnormal situations.
   c. Detects and handles emergency session establishments.

22.1.3.2 Interrogating CSCF (I-CSCF)

The interrogating CSCF (I-CSCF) acts as an IMS network routing proxy and provides scalability support for S-CSCF (Figure 22.6).

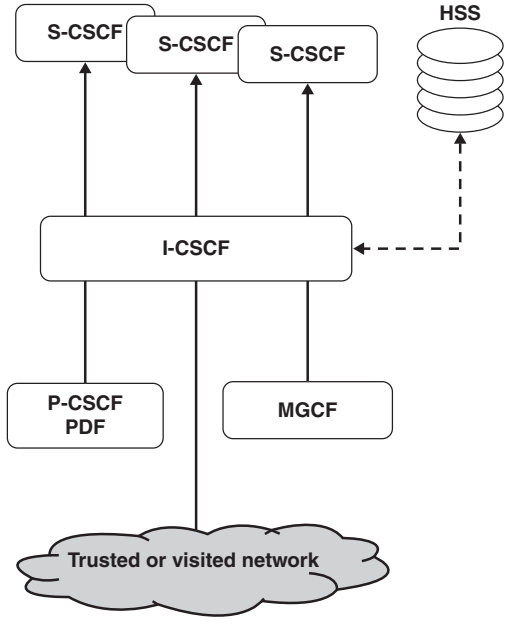


Figure 22.6 Interrogating CSCF (I-CSCF).

The I-CSCF is the initial contact point for incoming network connections. It has well known address(es) within the network and can either be statically configured or discovered via DHCP. I-CSCF provides the following functionalities:

1. Selects S-CSCF for a user performing SIP registration:
   a. Provides S-CSCF fan out to support scalability.
   b. Selection can be static or dynamic based on current conditions and user location.
2. Routes request to proper S-CSCF or external network element:
   a. Query HSS for the address of S-CSCF to handle call.
   b. If no S-CSCF is currently assigned, (e.g. unregistered termination), then assign S-CSCF to handle the SIP request.
3. Acts as a stateless SIP proxy. Generates CDR events.
4. Provides topology hiding inter-network gateway (THIG). Hides configuration, capacity and topology of network from outside.

22.1.3.3 Serving CSCF (S-CSCF)

The serving CSCF (S-CSCF) coordinates application server interactions and performs network routing (Figure 22.7).

The S-CSCF helps interface application servers with IMS and provides the routing intelligence within IMS network. S-CSCF provides the following functionalities:

1. Acts as registrar and notification server:
   a. IETF RFC 3261 compliant registrar.
   b. IETF RFC 3265 compliant event notifications.
   c. Generally there is a 1-1 binding between registered endpoint and S-CSCF.

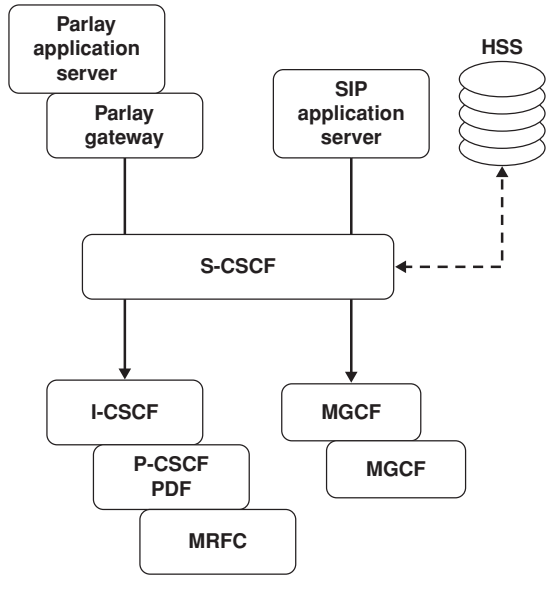


Figure 22.7 Serving CSCF (S-CSCF).

2. Locally stores subscriber data:
   a. Retrieves the subscriber data from the HSS.
   b. Includes filter criteria information, which indicates the application servers to contact for specified events.
3. Provides session control and routing:
   a. Provides session control for the registered endpoint's sessions.
   b. Behaves as both SIP proxy and user agent.
   c. Generates session level CDRs.
4. Provides bearer authorization. Ensures that media types and quantities indicated by SDP for a session are within boundaries of subscriber's profile.
   a. Provides application interaction. Interacts with Application Services platforms for the support of services.

22.2 IMS Service Model

The S-CSCF and application servers interact for providing novel services in a CSP's IMS-based network. Here are some of the important details about the interaction (Figure 22.8):

1. The S-CSCF can transfer control to an application server (AS) at specific points, referred to as service point triggers (SPT), during SIP signaling between two SIP end-points. Service point triggers include:
   a. Any initial SIP Method, such as, INVITE, REGISTER, SUBSCRIBE, MESSAGE.
   b. Registration type.
   c. Presence or absence of any header.
   d. Content of any header.
   e. Direction of the request with respect to the served user (originating or terminating).
   f. Session description information (i.e. SDP).

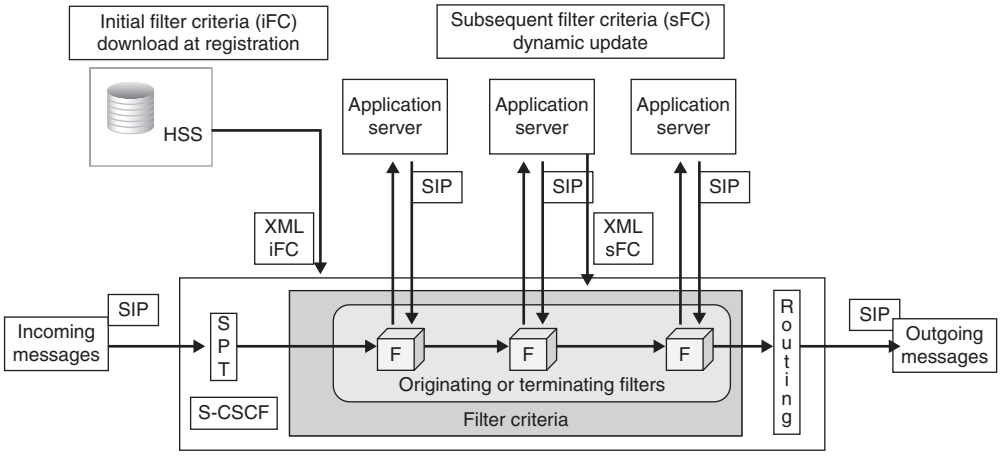


Figure 22.8 IMS service model: Interaction between S-CSCF and application server(s).

2. Filter criteria can be set in the SPTs (blocks marked "F" in Figure 22.8 indicate filter criteria). A filter criteria triggers one or more SPTs in order to send the related request to one specific AS. The set of filter criteria that is stored for a service profile of a specific user is called "application server subscription information".

   One type of filter criterion is referred to as initial filter criterion (iFC) – it is usually downloaded at the registration time and is used to define a desired application interaction and includes the following information in XML format:
   a. The SPT(s), where multiple SPTs may be linked with logical expressions (e.g. AND, OR, NOT).
   b. Address of AS to be contacted.
   c. Priority of initial filter criteria.
   d. Default handling when AS cannot be reached.
   e. Optional Service Information for the message body.
3. If a trigger point match is found:
   a. The S-CSCF forwards the request to the application server (AS) indicated in the current filter criteria.
   b. The S-CSCF may forward messages to a number of application servers in the order indicated by the filter criteria.
4. The application server, on receiving the message from S-CSCF:
   a. Performs the service logic.
   b. May modify the request and send the request back to S-CSCF.
   c. May continue or disengage in subsequent messaging.
5. After the last AS is contacted, the message is routed towards the intended destination.

22.3 IMS Signaling

22.3.1 SIP Registration/Deregistration

Signaling involved in Registration/Deregistration is shown in Figure 22.9.

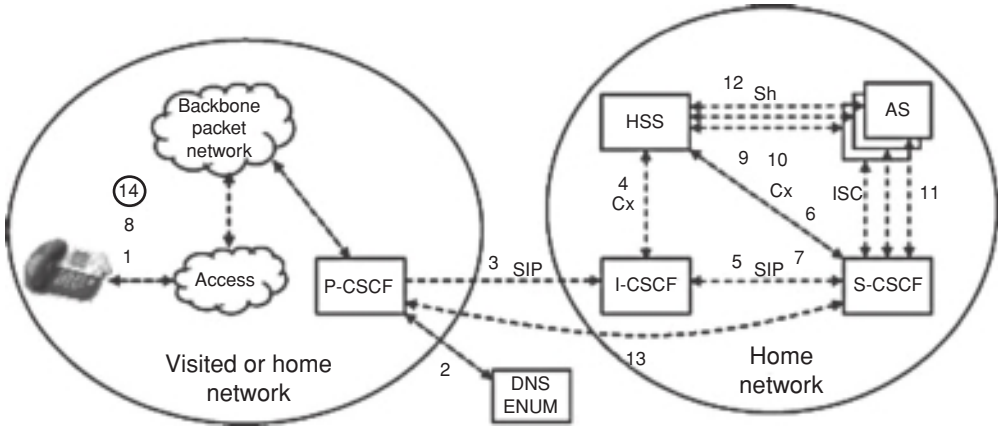


Figure 22.9 Signaling involved in SIP registration/de-registration.

The steps are:

1. Initiate SIP registration.
2. Query DNS to obtain routing information for I-CSCF.
3. Forward SIP REGISTER to home network.
4. Retrieve information needed for S-CSCF selection.
5. Forward SIP REGISTER to S-CSCF.
6. Retrieve and select authentication vector.
7. Reject with authentication data.
8. Re-initiate SIP registration (steps 1–5).
9. Store S-CSCF name.
10. Retrieve subscriber profile and filter criteria.
11. Register with AS(s) based on filter criteria.
12. AS(s) retrieve subscriber profile (if needed).
13. P-CSCF SUBSCRIBE, for deregistration.
14. UE SUBSCRIBE, for deregistration.

22.3.2 IMS Subscriber to IMS Subscriber

Signaling involved in IMS subscriber to IMS subscriber is shown in Figure 22.10.

The steps are:

1. Initiate SIP invitation.
2. Retrieve subscriber profile (if needed).
3. Apply service logic.
4. Retrieve address of called (CLD) Party Home Network and Forward INVITE.
5. Identify registrar of CLD party and forward INVITE.
6. Retrieve subscriber profile (if needed).

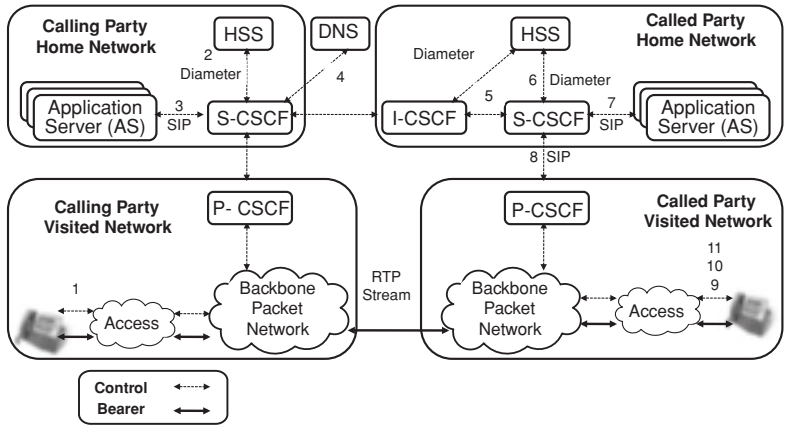


Figure 22.10 IMS subscriber to IMS subscriber signaling.

7. Apply service logic.
8. Forward INVITE to CLD party.
9. SDP negotiation/resource reservation control.
10. Ringing/alerting.
11. Answer/connect.

22.4 Integration of IPTV in IMS Architecture

22.4.1 Functional Architecture and Interfaces

Figure 22.11 shows a simplified functional architecture of IPTV in IMS architecture [10–13]. The main functional units incorporated in IMS to support IPTV are:

- IPTV supporting functions.
- IPTV applications functions.
- IPTV media control functions.
- IPTV media delivery functions.

User equipment (UE) interfaces with IPTV supporting functions and IPTV applications functions using the Xt interface, whereas the HSS interfaces with them using the Sh interface.

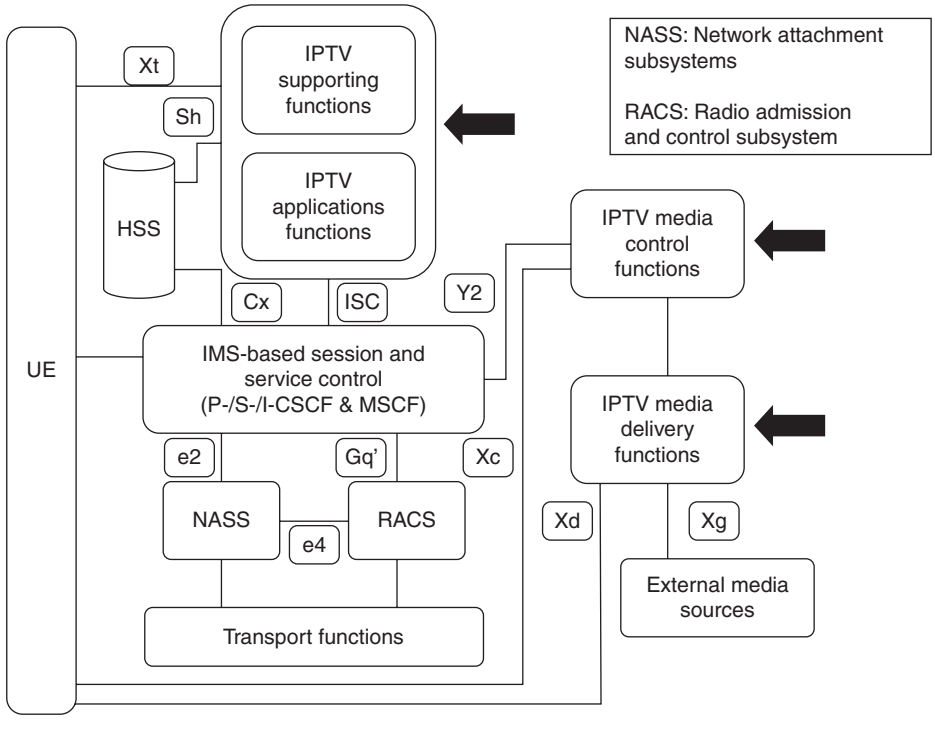


Figure 22.11 Simplified IMS/IPTV functional architecture.

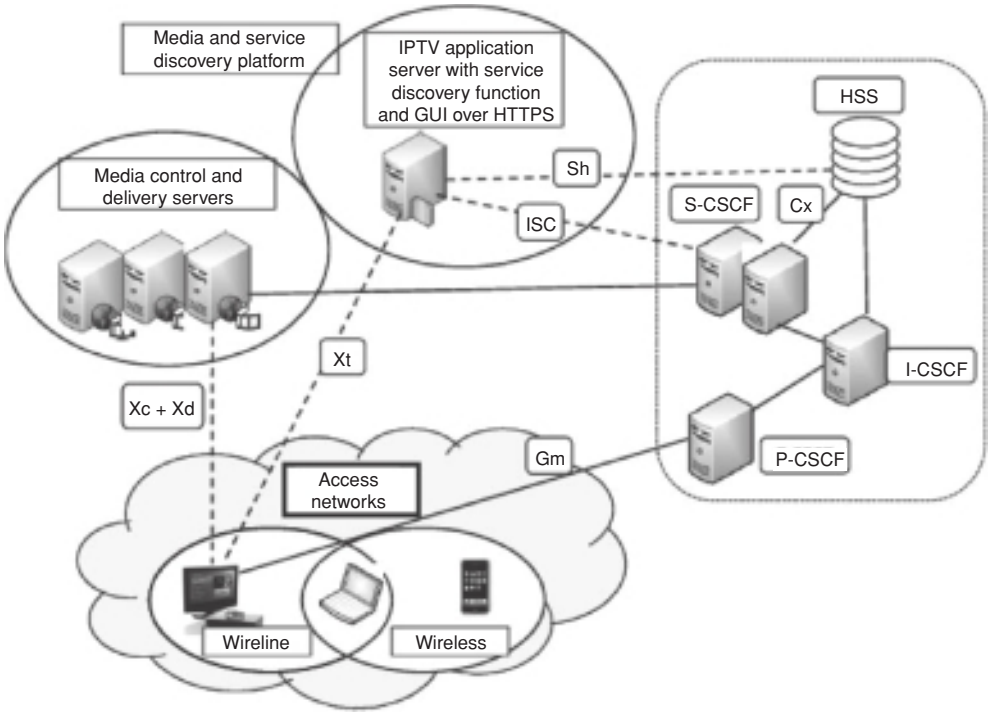


Figure 22.12 IMS architecture enhanced with IPTV support.

P-/S-/I-CSCF interface with IPTV control functions using the Y2 interface. The UE uses the Xc and Xd interfaces to interface with IPTV control functions and IPTV delivery functions respectively. These interfaces are shown along with the relevant IPTV servers and switches in Figure 22.12.

22.4.2 Integrated IMS-IPTV Architecture

The IPTV subsystem, in the context of IMS, has two components:

- **Media and Service Discovery**: this consists of the IPTV application server with service discovery capability and is invoked in the beginning to discover the available services.
- **Media Control and Delivery**: this consists of the media server that streams the video content.

22.4.3 Discovery and Selection of IPTV Service and Establishment of an IPTV Session

The IMS user first logs on to IPTV using a GUI. The IPTV application server, upon receiving the log-in credentials from the user over the Xt interface, authenticates her against HSS and retrieves her profile information over the Gm interface as shown in Figure 22.13. Upon

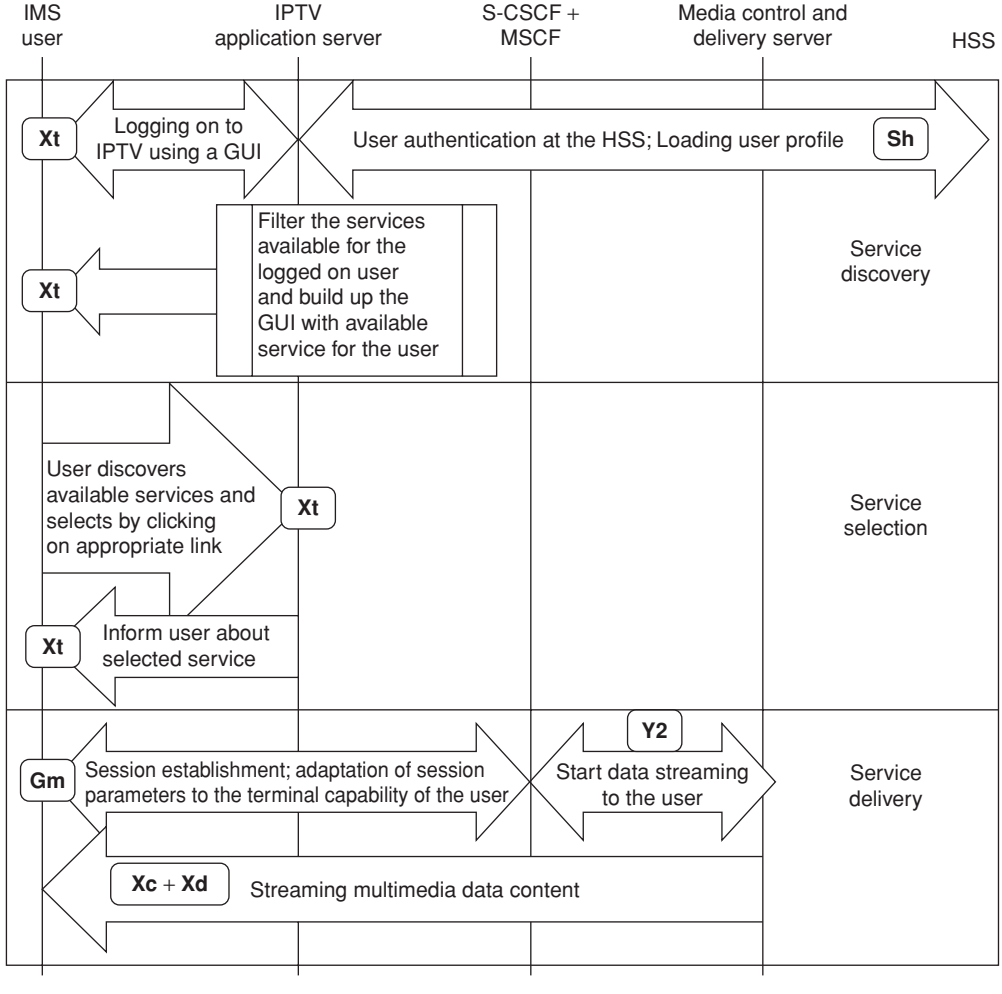


Figure 22.13 Discovery and selection of IPTV service and establishment of an IPTV session.

receiving the user's profile information from the HSS, IPTV Application server filters the available services using the retrieved profile information and passes on to the IMS user (over the Xt interface) only those services that are applicable to her. The IMS user then selects the desired service from the list by clicking on the appropriate link. The IPTV application server informs the user about the selected service, again using the Xt interface. These phases are referred to as discovery and selection of IPTV services.

Session establishment happens right after that, via P-CSCF, using the Gm interface. During session establishment, information about the terminal capabilities of the IMS user is taken into account. S-CSCF informs the media control and delivery functions (Figure 22.13), using the Xc interface, to start streaming to the user. Streaming multimedia data is then streamed to the end user by the media delivery server over the Xd interface.

While IPTV services can be delivered from outside the IMS infrastructure of the CSP, it is in the interest of CSPs to use IMS for enabling novel multimedia services while not losing control of their customers.

22.5 Summary

The competitive threat from ASPs made CSPs think of alternatives to their existing circuit-switched network with INs that simply did not have the richness and flexibility of IP-based service offerings. IMS is the genesis of that thought process and is indeed a major strategic shift from CSP perspective. The IMS architecture leverages the IETF protocols, such as SIP, as the core signaling protocol on top of their IP-based transport network while rearchitecting the other components in a traditional CSP network including the core switching system and the repository of subscribers (HLR), so that the newly defined components can deliver the same set of services, if not more, as compared to those offered by the ASPs while retaining the full control of the subscribers.

Before the advent of IMS architecture, applications for CSPs, such as, IM, VoIP, and VoD had to be developed independently even though they shared a lot of commonality. The IMS extracts the commonality between applications and neatly encapsulates the common functionalities into four categories: (i) common resources, (ii) converged applications, (iii) session control and (iv) access agnostic functions. Common resources, such as, user data base and portal together with single sign-on can be shared among the applications. Session control and management can be shared across applications. Access agnostic support such as global roaming and consistent service across networks is also independent of the application. Only application-specific things are left for the applications to implement, thereby simplifying development of converged applications. This concept is referred to as the layering of IMS architecture.

The components of IMS were briefly described followed by the IMS service model, which includes transferring of control via SPTs. Signaling was also illustrated for couple of cases, such as SIP registration/deregistration and IMS subscriber to IMS subscriber signaling. After the basic concepts of IMS architecture were explained, the details of interfacing IPTV with IMS architecture were discussed.

References

[1] IP Multimedia Subsystems: http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem (accessed June 13, 2010).
[2] IMS Architecture. http://www.metaswitch.com/sbc-session-border-controller/ims-architecture.aspx (accessed June 13, 2010).
[3] The Importance of Standard IMS Architecture. Rakesh Khandelwal. http://www.iec.org/newsletter/may07_2/analyst_corner.pdf (accessed June 13, 2010).
[4] 3GPP TS 24.229 IP Multimedia Call Control Protocol Based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP).
[5] 3GPP TS 23.228 Technical Specification Group Services and System Aspects IP Multimedia Subsystem (IMS).
[6] 3GPP TS 23.981 Interworking Aspects and Migration Scenarios for IPv4–Based IMS Implementations.
[7] Camarillo, G. and García-Martín, M.-A. (2006) *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, John Wiley & Sons, Ltd. ISBN 0-470-01818-6.
[8] Poikselka, M., Niemi, A., Khartabil, H. and Mayer, G. (2006) *The IMS: IP Multimedia Concepts and Services*, John Wiley & Sons, Ltd. ISBN 0-470-01906-9.

[9] Iiyas, M., Ahson, S.A. (eds) (2008) *IP Multimedia Subsystem (IMS) Handbook*, CRC Press. ISBN 1420064592.
[10] IMS-based IPTV. Dr. Khalid Ahmad. http://www.atis.org/supercomm/Presentations/IPTV%20Track/IMS-Based%20IPTV.pdf (accessed June 13, 2010).
[11] Levent-Levi, Tsahi. The Architecture of IPTV in an IMS Deployment. http://blog.radvision.com/voipsurvivor/2008/09/18/the-architecture-of-iptv-in-an-ims-deployment/ (accessed June 13, 2010).
[12] Riede, C., Al-Hezmi, A. and Magedanz, T. (2008) Session and Media Signaling for IPTV via IMS. Proceedings of the 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications. Article No.: 20, 2008, ISBN: 978-1-59593-984-5.
[13] IPTV IMS Multimedia Lab, http://www.i2ml.com/ (accessed June 13, 2010).
[14] Intelligent Networks. http://www.itu.int/TELECOM/wt95/pressdocs/ft-03-e.html (accessed June 13, 2010).
[15] Intelligent Network. http://en.wikipedia.org/wiki/Intelligent_network (accessed June 13, 2010).
[16] Rosenberg, J., Schulzrinne, H., Camarillo, G. *et al.* Session Initiation Protocol. RFC 3261.

23

Summary of Part Three

Chapter 14 discussed the evolution of network architecture for the communication service providers (CSPs). Specifically, from their isolated network silos, their networks have evolved into a "converged" network spanning IPTV, mobile, telephone and broadband Internet – enabling them to provide "quadruple" play services in a seamless manner. Furthermore, this "converged" network architecture enables the CSPs to blend services across networks very quickly in creative ways so as to extract some value from the end-to-end value chain beyond just connectivity revenue. Convergence in network is possible because of Internet protocol (IP) that is gluing together presumably disparate networks. Furthermore, the IP multimedia subsystem (IMS) provides a common infrastructure for "policy" management, "session" management and services. With the converged network and service infrastructure in place, CSPs can provide video on demand, broadcast television and interactive services across multiple networks over multiple devices.

Chapter 15 focused on IPTV services, namely broadcast audio/video/data services, the video-on-demand (VoD) service and interactive service. Internet protocol TV standards with respect to display quality and transport mechanisms spanning data encapsulation, transmission protocols, quality of service, service delivery platform, content distribution and forward error correction (FEC) were discussed in addition to the, mechanisms that enable advanced services, such as, search capabilities in television.

For point-to-point communication, such as for VoD, IP is used in unicast mode while for point-to-multipoint (p-t-m) communication, such as for live TV, IP is used in multicast mode. Furthermore, video could be transported either using TCP or UDP. Typically, progressive download or the download-and-play mode of IPTV uses HTTP/TCP as the transport layer protocol. An alternative reliable delivery mechanism for video over multicast networks is to use file delivery over unidirectional transport or FLUTE protocol. Most commonly video is transported in IPTV in "streaming" mode in which audio/video data is encapsulated in MPEG-2 transport stream (TS), which could either directly go over UDP/IP or leverage RTP/UDP/IP. Internet protocol multicast is used for p-t-m streaming and that requires the receiving end-points of IPTV transmission to join relevant IP multicast groups using Internet group management protocol (IGMP). The receivers of the IPTV stream provide feedback

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul

about the quality of reception to the server using RTP control protocol (RTCP). In the case of VoD, signaling is done using real-time streaming protocol (RTSP).

There are two distinctly different approaches to delivering live TV or broadcast content in an IPTV network. The first approach is based on the idea of the best effort delivery paradigm of the Internet whereas the second approach is based on the idea of delivering live TV content over traffic-engineered network paths. In the best effort paradigm, the live TV content is distributed using IP multicast over a shared network infrastructure where it is statistically multiplexed with other types of traffic. Lost packets are recovered at the end hosts using forward error correction (FEC). In the alternative approach, p-t-m, label switched paths (LSPs) are used to set up traffic-engineered paths from the source to the destinations whereby live TV traffic is delivered with a guaranteed quality of service over the IPTV network.

Chapter 16 introduced the challenges in delivering video over multiple delivery networks, such as broadband, mobile (cellular, WiMax etc.) and television networks (IPTV, digital cable, digital satellite etc.). Specifically, it was stressed that the heterogeneity in terms of screen size, viewing distance, aspect ratio, resolution, user interface, codecs, media formats and bandwidth availability for TV, Mobile and Web/PC makes it challenging to transform an application written for one screen to be made available on other screens. Potential approaches, such as rule-based media transformation, which enables the transformation of media from one format/codec to another based on the capabilities of the display device; static versus dynamic transformation, which transcodes the more "popular" content ahead of time and uses "on-the-fly" transcoding for less frequently accessed content; and usage of "standardized" templates for different categories of applications – are approaches to address the challenges of multichannel content/application rendering. A commercial transcoder called Rhozet was presented in detail to provide a complete understanding of how transcoding systems work in practice. Finally, a case study of a virtual personal multimedia library (VPML) was presented to show how exactly video and other multimedia content can be delivered over multiple channels with an optimal user experience. Specifically, it was shown that VPML transcodes content into various formats and bitrates and delivers the appropriate version of the content based on the capabilities of the requesting device.

Chapter 17 focused on quality of service (QoS) requirements for IPTV networks at multiple layers: application layer, transport layer and network layer. While application-layer requirements focus on the end-to-end application level bit rates, transport-layer requirements focus on latency, jitter and packet loss rate, and network-layer requirements focus on packet loss rate as a function of bit rate and interval between two consecutive uncorrected packet loss events. From the application-layer point of view, both standard definition TV (SDTV) and high definition TV (HDTV) requirements were described for broadcast TV as well as for VoD. Quality of service requirements at the transport layer are for latency, jitter and packet loss where packet loss is characterized by loss distance, loss duration and loss rate. Quality of service considerations for the transport layer requires the underlying network to meet particular standards. The network layer QoS is characterized by packet loss ratio. The main QoS requirement at the network layer is about maintaining packet loss ratio below a threshold in order to meet the transport layer requirements. If the network is not able to meet the specified performance objectives with respect to packet loss, forward error correction (FEC) or interleaving techniques can be used at the network layer and error concealment, application-layer FEC and/or automatic repeat request (ARQ) can be used at the application layer to achieve the required performance level. In addition to bearer path QoS requirements, QoS requirements

for the control functions including limiting the channel zapping time under a threshold and reducing video on demand trick-mode latency were also described.

Chapter 18 was dedicated to describing techniques followed by an IPTV service provider to meet the QoS requirements. First, it is absolutely essential for a service provider to predict traffic growth as accurately as possible and engineer the network with enough capacity to enable congestion-free transport in order to ensure minimal packet loss, packet reordering and packet jitter. Further, the network should have enough redundancy to ensure fast failover of circuits to minimize outage in service delivery. Second, IPTV service providers would have to leverage IP multicast for replication, distributed caching for serving content locally, flexible content insertion and storage at most economical points in order to avoid high cost of transmission, and do video admission control in order to prevent injection of new traffic in the network when it is congested. Third, in order to prevent overloading of network with video traffic especially when there is sudden surge in requests, it is important to (dis)allow certain requests from being honored. However, the goal would be to minimize "blocking" and hence clever techniques such as predictive analytics to forecast capacity requirements and proactively reserving resources in the network and/or dynamically adding resources in the network to handle additional traffic would play a critical role in ensuring best possible user experience for customers. Finally, despite every attempt in engineering the network, there is a finite probability of congestion and contention for resources in the network leading to degraded quality of experience. The only way this can be eliminated would be to constantly monitor the video and audio quality at multiple points of the network, starting from the video headend equipment, such as MPEG encoders and streaming servers to the residential gateway and IP set top boxes, with multiple intermediate points in the delivery network, and take action when the quality degrades beyond a threshold. Network diagnostics and reporting; network performance and fault management; MPEG 2/4 analysers and video monitors are some of the key challenges from an end-to-end service management perspective.

A representative architecture for end-to-end QoE assurance was presented to highlight the components of such a system. Then IPTV monitoring was discussed with its various domains and monitoring points leading to an end-to-end monitoring architecture. Quality of experience monitoring for IPTV can be done using "active" monitoring in which a test device injects test packets into the network, and some other device measures test packets at some other points within the network. Passive monitoring can also be used, in which a test device just observes characteristics of packets on a network link. The observed characteristics can be used for flow analysis. These monitoring techniques can be combined into a "hybrid" monitoring system to get the best of both worlds.

Both the video and audio qualities can be measured with subjective and objective methods. There are several products in the market that focus on IPTV QoE monitoring. However, we focused on a specific tool from a vendor in a quest to understand what kind of monitoring support is available in practice.

Chapter 19 focused on security issues of digital video content spanning unauthorized access and piracy in all stages of its lifecycle. In a converged network, wired or wireless, that carries data, voice and video, the magnitude and scope of content and its handlers and consumers are enormous. Such a vast content network and the content traversing across the network cannot be protected using traditional security solutions. A security framework called UCOMAP was proposed as a solution for protecting digital video content throughout its complete lifecycle in a converged network. The proposed framework is based on trusted and proven PKI technology,

which is not known to have been breached. Each device, agency or individual that creates, handles, or consumes content needs to carry a digital certificate and private keys, and all content, in whatever form, raw or processed, needs to be stored and transmitted encrypted and signed. Only certified devices associated with consumers that are capable of encrypting, decrypting, signing and verifying content are able to receive, process, store and transmit copyrighted digital content in the UCOMAP framework. Copyright protection of the decrypted content is provided by secure digital watermarking and fingerprinting.

Chapter 20 discussed how VoDe can be offered in a cost-effective manner. Specifically, a service provider needs to be cognizant of several resources including the I/O bandwidth at the video server, network bandwidth, I/O bandwidth at the client, storage space at the client while providing an optimal user experience in terms of quick start-up time and uninterrupted high-quality of video. Both closed-loop and open-loop systems were described. Closed-loop systems enable video servers to schedule video streams based on client requests while in open-loop systems, video servers schedule video streams without taking into account the client requests. There are two broad categories of closed-loop systems: (i) client-initiated and (ii) client-initiated with prefetching.

In a client-initiated multicast scheme, a client makes a request for a video and waits until that video is eventually multicast. When a channel becomes available, the server selects a batch of pending requests for a video according to some scheduling policy. In the client-initiated with prefetching (CIWP) scheme, two clients that request the same video at different times can share a multicast stream; the earlier arrival client is not delayed. For two requests spaced a few minutes apart, the complete video is multicast by the server in response to the first request. The second request is satisfied by transmitting first few minutes of the video while requiring the client to continuously prefetch and buffer remainder of the video from the first multicast group. The CIWP approach takes advantage of the resources (such as disk storage space and network bandwidth) on the client side to save server network – I/O bandwidth by multicasting segments of video.

Open-loop systems can also be categorized into broad classes: (i) server-initiated and (ii) server-initiated with prefetching. In server-initiated multicast scheme, the server multicasts video objects periodically via dedicated server network-I/O resources. Clients join an appropriate multicast group to receive the desired video stream. This scheme guarantees a maximum service latency of inter-session interval independent of the arrival time of the requests. In a server-initiated with prefetching (SIWP) scheme, a video object is divided into segments, each of which is multicast periodically via a dedicated multicast group. The client prefetches data from one or more multicast groups for later playback. A SIWP scheme takes advantage of resources (e.g., disk storage space and network bandwidth) at the client end, and therefore significantly reduces the server network-I/O resources required. This scheme also guarantees a maximum service latency independent of the arrival time of the requests and performs better for hot videos than for cold videos.

Usually, the closed-loop systems perform well for low client request rates while the open-loop systems perform better for high client request rates. We also looked at a hybrid scheme that combines the best closed-loop scheme with the best open-loop scheme and applies the open-loop scheme for the “hot” videos and the closed-loop scheme for “not-so-hot” videos for an optimal overall system performance.

Chapter 21 was dedicated to video content distribution, including real-time television content and non-real-time on-demand video content, to mobile handsets. Video can be distributed in

cellular networks in one of three ways: (i) full download, (ii) progressive download and (iii) streaming. Each of these approaches has its advantages and disadvantages and they are hence suited for different types of content. When several people simultaneously request the same video to be played, multicast using p-t-m delivery is the most efficient way of delivering content. On one hand, 3GPP has defined a standard called MBMS for setting up multicast trees in cellular 3G networks and, on the other hand, overlay networks, such as DVB-H and FLO have gained popularity for distribution of video content in an out-of-band manner to the same handsets. Such technology helps reduce the cost of distribution of video content by using more powerful transmitters covering large areas and by not using up the network capacity of the cellular network. The state-of-the art broadcast/multicast systems in mobile networks combine both mechanisms such that DVB-H/FLO is used for distributing mobile video content and the 3G network is used for providing the reverse channel to facilitate interaction with the content. The MBMS standard also defines security architecture for protecting multicast video content from nonsubscribers if the content is not free. The Open Mobile Alliance (OMA) defines a DRM system for protecting the rights of the content owners when copyrighted content is distributed to consumers.

Chapter 22 focuses on the integration of IPTV with the IP multimedia system (IMS) infrastructure of CSPs. The IMS architecture leverages the IETF protocols, such as, SIP as the core signaling protocol on top of their IP-based transport network, while rearchitecting the other components in a traditional CSP network including the core switching system and the repository of subscribers, the home location register (HLR), so that the newly defined components can deliver at least the same set of services, if not more, as compared to those offered by the over the top (OTT) players while retaining the full control of the subscribers.

The IMS consists of four subsystems: (i) common resources, (ii) converged applications, (iii) session control and (iv) access agnostic functions. Common resources, such as a user data base and portal together with single sign-on can be shared among the applications. Session control and management can be shared across applications. Access-agnostic support, such as global roaming and consistent service across networks is also independent of the application. Only application-specific things are left for the applications to implement, thereby simplifying development of converged applications. This concept is referred to as the layering of IMS architecture. IMS components were briefly described followed by the IMS service model which includes transferring of control via service point triggers (SPTs). The IMS signaling was also illustrated for couple of cases, such as SIP registration/deregistration and IMS subscriber to IMS subscriber signaling. After the basic concepts of IMS architecture were explained, the details of interfacing IPTV with IMS architecture were discussed.

Index

Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions
Sanjoy Paul