further see that each of these application classes has a different set of service requirements for the network. In Section 7.2 we examine streaming of stored audio/video in some detail. In Section 7.3, we'll investigate application-level techniques that can enhance the performance of multimedia applications in today's best-effort service Internet, and in Section 7.4, we'll cover several multimedia protocols in use in today's Internet. In Section 7.5, we'll investigate mechanisms within the network that can be used to distinguish one class of traffic (e.g., delay-tolerant applications such as multimedia) from another (e.g., elastic applications such as FTP), and provide differentiated service among several classes of traffic. Finally, in Section 7.6, we'll consider the case where the network must make performance *guarantees* to an application—e.g., that a packet-based IP telephone call will receive the same performance as if the call had been carried in a circuit-switched telephone network. We'll see that this will require the introduction of new network mechanisms and protocols.

7.1 Multimedia Networking Applications

In our discussion of application service requirements in Chapter 2, we identified a number of axes along which these requirements can be classified. Two of these axes—timing considerations and tolerance of data loss—are particularly important for networked multimedia applications. Timing considerations are important because many multimedia applications are highly **delay-sensitive**. We will see shortly that in many multimedia applications, packets that incur a sender-to-receiver delay of more than a few hundred milliseconds are essentially useless to the receiver. On the other hand, networked multimedia applications are for the most part **loss-tolerant**—occasional loss only causes occasional glitches in the audio/video playback, and these losses can often be partially or fully concealed. These delay-sensitive but loss-tolerant characteristics are clearly different from those of elastic applications such as the Web, e-mail, FTP, and Telnet. For elastic applications, long delays are annoying but not particularly harmful, and the completeness and integrity of the transferred data is of paramount importance.

7.1.1 Examples of Multimedia Applications

The Internet can support a large variety of exciting multimedia applications. In this subsection, we consider three broad classes of multimedia applications: streaming stored audio/video, streaming live audio/video, and real-time interactive audio/video.

In this chapter we do *not* cover download-and-then-play applications, such as fully downloading an MP3 over a P2P file-sharing application before playing back the MP3. Indeed, download-and-then-play applications are elastic, file-transfer applications without any special delay requirements. We examined file transfer (HTTP and FTP) and P2P file-sharing systems in Chapter 2.

CASE HISTORY

IPTV

Television content has traditionally been distributed over terrestrial microwave, hybrid-fiber coax (HFC), and geostationary satellite channels (see Section 1.2). But in today's Internet era, there is tremendous interest in IPTV— that is, distributing television content over the Internet.

One of the challenges of IPTV is dealing with the immense amount of bandwidth required, particularly at the server source. For example, consider distributing a major sporting event, such as a World Cup match, from a single server over the Internet to 100 million concurrent users. If the video rate is a modest 1 Mbps, then the server bandwidth required would be an outrageous 100 terabits/sec! Thus, classical client-server distribution is totally out of the question. If IP multicast were widely deployed throughout the Internet, it would be much easier to make IPTV a reality. Another alternative is to distribute the video over a multicast overlay network, such as those provided by content distribution networks (CDNs) (see Section 7.3).

Yet another alternative is to use peer-to-peer distribution, whereby each peer that receives a television channel also aids in redistributing the channel to other peers. Perhaps the greatest appeal of such an approach is the low distribution cost: if the individual peers collectively provide sufficient upstream bandwidth, little server bandwidth may be needed (perhaps only a few multiples of the video rate). At such low cost, anyone with a Web cam could distribute a live program to millions of users at negligible cost!

To date, a number of BitTorrent-like P2P IPTV systems have enjoyed successful deployment. The pioneer in the field, CoolStreaming, reported more than 4,000 simultaneous users in 2003 [CoolStreaming 2005]. More recently, a number of other systems, including PPLive and ppstream, have reported great success, with tens of thousands of simultaneous users watching channels at rates between 300 kbps and 1 Mbps. In these BitTorrent-like systems, peers form a dynamic overlay network, and exchange chunks of video with overlay neighbors. It will be interesting to follow how IPTV plays out over the next 5 to10 years. What underlying technology will be used: CDN or P2P, or some hybrid of the two? And will a significant fraction of World Cup fans watch the 2014 matches from the Internet?

Streaming Stored Audio and Video

In this class of applications, clients request on-demand compressed audio or video files that are stored on servers. Thousands of sites provide streaming of stored audio and video today, including CNN, Microsoft Video, and YouTube. This class of applications has three key distinguishing features.

- *Stored media.* The multimedia content, which is prerecorded, is stored at the server. Because the media is prerecorded, the user at the client may pause, rewind, fast-forward, or index through the multimedia content. The time from when the user makes such a request until the action manifests itself at the client should be on the order of one to ten seconds for acceptable responsiveness.
- *Streaming.* In a streaming stored audio/video application, a client typically begins playout of the audio/video a few seconds after it begins receiving the file from the server. This means that the client will be playing out audio/video from one location in the file while it is receiving later parts of the file from the server. This technique, known as **streaming**, avoids having to download the entire file (and incurring a potentially long delay) before beginning playout. There are many streaming multimedia clients, including RealPlayer from RealNetworks [RealNetworks 2009], Apple's QuickTime [QuickTime 2009], and Microsoft's Windows Media [Microsoft Media Player 2009].
- *Continuous playout*. Once playout of the multimedia content begins, it should proceed according to the original timing of the recording. Therefore, data must be received from the server in time for its playout at the client; otherwise, users experience frustrating buffering delays. Although stored media applications have continuous playout requirements, their end-to-end delay constraints are nevertheless less stringent than those for live, interactive applications such as Internet telephony and video conferencing (see below).

Streaming Live Audio and Video

This class of applications is similar to traditional broadcast radio and television, except that transmission takes place over the Internet. These applications allow a user to receive a *live* radio or television transmission emitted from any corner of the world. (For example, one of the authors of this book often listens to his favorite Philadelphia radio stations when traveling. The other author regularly listened to live broadcasts of his university's beloved basketball team while he was living in France for a year.) These applications are often referred to as Internet radio and IPTV. Today there are thousands of radio stations broadcasting over the Internet and a number of deployments of IPTV (see sidebar on IPTV).

Since streaming live audio/video is not stored, a client cannot fast-forward through the media. However, with local storage of received data, other interactive operations such as pausing and rewinding can be possible. Live, broadcast-like applications often have many clients who are receiving the same audio/video program. Distribution of live audio/video to many receivers can be efficiently accomplished using the IP multicasting techniques described in Section 4.7. However, today live audio/video distribution is more often accomplished through application-layer multicast (using P2P or CDN) or through multiple separate server-to-client unicast streams. As with streaming stored multimedia, continuous playout is required, although the timing constraints are less stringent than for real-time interactive applications. Delays of up to tens of seconds from when the user requests the delivery/playout of a live transmission to when playout begins can be tolerated.

Real-Time Interactive Audio and Video

This class of applications allows people to use audio/video to communicate with each other in real time. Real-time interactive audio over the Internet is often referred to as **Internet telephony**, since, from the user's perspective, it is similar to the traditional circuit-switched telephone service. Internet telephony can provide private branch exchange (PBX), local, and long-distance telephone service at very low cost. It can also facilitate the deployment of new services that are not easily supported by the traditional circuit-switched networks, such as presence detection, group communication, caller filtering, Web-phone integration, and more. There are numerous Internet telephone products currently available. For example, Skype users can make PC-to-phone and PC-to-PC voice calls. With real-time interactive video, also called video conferencing, individuals communicate visually as well as orally. There are also many real-time interactive video products currently available for the Internet, including Microsoft's NetMeeting, Skype video, and various Polycom products. Note that in a real-time interactive audio/video application, a user can speak or turn its head at any time. For a conversation with interaction among multiple speakers, the delay from when a user speaks or moves until the action is manifested at the receiving hosts should be less than a few hundred milliseconds. For voice, delays smaller than 150 milliseconds are not perceived by a human listener, delays between 150 and 400 milliseconds can be acceptable, and delays exceeding 400 milliseconds can result in frustrating, if not completely unintelligible, voice conversations.

7.1.2 Hurdles for Multimedia in Today's Internet

Recall that the IP protocol deployed in the Internet today provides a **best-effort service** to all the datagrams it carries. In other words, the Internet makes its best effort to move each datagram from sender to receiver as quickly as possible, but it does not make any promises whatsoever about the end-to-end delay for an individual packet. Nor does the service make any promises about the variation of packet delay within a packet stream. Because TCP and UDP run over IP, it follows that neither of these transport protocols makes any delay guarantees to invoking applications. Due to the lack of any special effort to deliver packets in a timely manner, it is an extremely challenging problem to develop successful multimedia networking applications for the Internet. Nonetheless, multimedia over the Internet has achieved considerable success to date. For example, streaming stored audio/video with user-interactivity delays of 5 to 10 seconds is now commonplace in the Internet. But during peak traffic periods, performance may be unsatisfactory, particularly when intervening links are congested (such as congested transoceanic links).

Internet phone and real-time interactive video have also found widespread use; for example, there are routinely more than seven million Skype users online at any given time. Real-time interactive voice and video impose rigid constraints on packet delay and packet jitter. **Packet jitter** is the variability of packet delays within the same packet stream. Real-time voice and video can work well when bandwidth is plentiful, and hence delay and jitter are minimal. But quality can deteriorate to unacceptable levels as soon as the real-time voice or video packet stream hits a moderately congested link.

The design of multimedia applications would certainly be more straightforward if there were some sort of first-class and second-class Internet services, whereby first-class packets were limited in number and received priority service in router queues. Such a first-class service could be satisfactory for delay-sensitive applications. But to date, the Internet has mostly taken an egalitarian approach to packet scheduling in router queues. All packets receive equal service; no packets, including delay-sensitive audio and video packets, receive special priority in the router queues. No matter how much money you have or how important you are, you must join the end of the line and wait your turn! In the latter half of this chapter, we'll examine proposed architectures that aim to remove this restriction.

So for the time being we have to live with best-effort service. But given this constraint, we can make several design decisions and employ a few tricks to improve the user-perceived quality of a multimedia networking application. For example, we can send the audio and video over UDP, and thereby circumvent TCP's low throughput when TCP enters its slow-start phase. We can delay playback at the receiver by 100 msecs or more in order to diminish the effects of network-induced jitter. We can timestamp packets at the sender so that the receiver knows when the packets should be played back. For stored audio/video, we can prefetch data during playback when client storage and extra bandwidth are available. We can even send redundant information in order to mitigate the effects of network-induced packet loss. We'll investigate many of these techniques in the rest of the first half of this chapter.

7.1.3 How Should the Internet Evolve to Support Multimedia Better?

Today there is a continuing debate about how the Internet should evolve in order to better accommodate multimedia traffic with its rigid timing constraints. At one extreme, some researchers argue that fundamental changes should be made to the Internet so that applications can explicitly reserve end-to-end bandwidth and thus receive a *guarantee* on its end-end performance. A **hard guarantee** means the application will receive its requested quality of service (QoS) with certainty. A **soft guarantee** means the application will receive its requested quality of service with high probability. These researchers believe that if a user wants to make, for example, an Internet phone call from Host A to Host B, then the user's Internet phone application should be able to reserve bandwidth explicitly in each link along a route between the two hosts. But permitting applications to make reservations and requiring the network to honor the reservations requires some big changes. First we need a protocol that, on the behalf of applications, reserves link bandwidth on the path from the senders to their receivers. Second, we must modify scheduling policies in the router queues so that bandwidth reservations can be honored. With these new scheduling policies, not all packets get equal treatment; instead, those that reserve (and pay) more get more. Third, in order to honor reservations, the applications must give the network a description of the traffic that they intend to send into the network. The network must then police each application's traffic to make sure that it abides by the description. Finally, the network must have a means of determining whether it has sufficient available bandwidth to support any new reservation request. These mechanisms, when combined, require new and complex software in the hosts and routers as well as new types of services. We'll cover these mechanisms in detail in Section 7.6.

At the other extreme, some researchers argue that it isn't necessary to make any fundamental changes to best-effort service and the underlying Internet protocols. Instead they advocate a laissez-faire approach:

- As demand increases, the ISPs (both top-tier and lower-tier ISPs) will scale their networks to meet the demand. Specifically, ISPs will provide enough bandwidth and switching capacity to provide satisfactory delay and packet-loss performance within their networks [Huang 2005]. The ISPs will thereby provide better service to their customers (users and customer ISPs), translating to higher revenues through more customers and higher service fees. To ensure that multimedia applications receive adequate service, even in the case of overload, an ISP may overprovision bandwidth and switching capacity. With proper traffic forecasting and bandwidth provisioning, soft QoS guarantees can be made.
- Content distribution networks (CDNs) replicate stored content and put the replicated content at the edges of the Internet. Given that a large fraction of the traffic flowing through the Internet is stored content (Web pages, MP3s, video), CDNs can significantly alleviate the traffic loads on the ISPs and the peering interfaces between ISPs. Furthermore, CDNs provide a differentiated service to content providers: content providers that pay for a CDN service can deliver content faster and more effectively. We'll study CDNs later in this chapter in Section 7.3.
- To deal with live streaming traffic (such as a sporting event) that is being sent to millions of users simultaneously, **multicast overlay networks** can be deployed. A multicast overlay network consists of user hosts and possibly dedicated servers scattered throughout the Internet. These hosts, servers, and the logical links between them collectively form an overlay network, which multicasts (see Section 4.7) traffic from the source to the millions of users. Unlike multicast IP, for which the multicast function is handled by routers at the IP layer, overlay networks multicast at the application layer. For example, the source host might send

the stream to three overlay servers; each of the overlay servers may forward the stream to other overlay servers and hosts; the process continues, creating a distribution tree on top of the underlying IP network. By multicasting popular live traffic through overlay networks, overall traffic loads in the Internet can be reduced over the case of unicast distribution.

Between the reservation camp and the laissez-faire camp there is a yet a third camp—the differentiated services (Diffserv) camp. This camp wants to make relatively small changes at the network and transport layers, and introduce simple pricing and policing schemes at the edge of the network (that is, at the interface between the user and the user's ISP). The idea is to introduce a small number of traffic classes (possibly just two classes), assign each datagram to one of the classes, give datagrams different levels of service according to their class in the router queues, and charge users according to the class of packets that they are sending into the network. We'll cover differentiated services in Section 7.5.

These three different approaches for handling multimedia traffic—making the best of best-effort service, differential QoS, and guaranteed QoS—are summarized in Table 7.1, and covered in Sections 7.3, 7.5, and 7.6, respectively.

7.1.4 Audio and Video Compression

Before audio and video can be transmitted over a computer network, it must be digitized and compressed. The need for digitization is obvious: computer networks transmit bits, so all transmitted information must be represented as a sequence of bits. Compression is important because uncompressed audio and video consume a tremendous amount of storage and bandwidth—removing the inherent redundancies with compression in digitized audio and video signals can reduce the amount of data that needs to be stored and transmitted by orders of magnitude. As an example, a

Approach	Unit of allocation	Guarantee	Deployment to date	Complexity	Mechanisms
Making the best of best-effort service	none	none, or soft	everywhere	minimal	application-layer support, CDN, over-provisioning
Differential QoS	classes of flows	none, or soft	some	medium	policing, scheduling
Guaranteed QoS	individual flows	soft or hard, once a flow is admitted	little	high	policing, scheduling, call admission and signaling

Table 7.1 ♦ Three approaches to supporting multimedia applications

single image consisting of 1024 pixels, with each pixel encoded into 24 bits (8 bits each for the colors red, green, and blue), requires 3 Mbytes of storage without compression. It would take seven minutes to send this image over a 64 kbps link. If the image is compressed at a modest 10:1 compression ratio, the storage requirement is reduced to 300 Kbytes and the transmission time also drops by a factor of 10.

The topics of audio and video compression are vast. They have been active areas of research for more than 50 years, and there are now literally hundreds of popular techniques and standards for both audio and video compression. Many universities offer entire courses on audio compression and on video compression. We therefore provide here only a brief and high-level introduction to the subject.

Audio Compression in the Internet

A continuously varying analog audio signal (which could emanate from speech or music) is normally converted to a digital signal as follows:

- The analog audio signal is first sampled at some fixed rate, for example, at 8,000 samples per second. The value of each sample is an arbitrary real number.
- Each of the samples is then rounded to one of a finite number of values. This operation is referred to as **quantization**. The number of finite values—called quantization values—is typically a power of two, for example, 256 quantization values.
- Each of the quantization values is represented by a fixed number of bits. For example, if there are 256 quantization values, then each value—and hence each sample—is represented by 1 byte. Each of the samples is converted to its bit representation. The bit representations of all the samples are concatenated together to form the digital representation of the signal.

As an example, if an analog audio signal is sampled at 8,000 samples per second and each sample is quantized and represented by 8 bits, then the resulting digital signal will have a rate of 64,000 bits per second. This digital signal can then be converted back—that is, decoded—to an analog signal for playback. However, the decoded analog signal is typically different from the original audio signal. By increasing the sampling rate and the number of quantization values, the decoded signal can approximate the original analog signal. Thus, there is a clear trade-off between the quality of the decoded signal and the storage and bandwidth requirements of the digital signal.

The basic encoding technique that we just described is called **pulse code modulation (PCM)**. Speech encoding often uses PCM, with a sampling rate of 8,000 samples per second and 8 bits per sample, giving a rate of 64 kbps. The audio compact disk (CD) also uses PCM, with a sampling rate of 44,100 samples per second with 16 bits per sample; this gives a rate of 705.6 kbps for mono and 1.411 Mbps for stereo.

A bit rate of 1.411 Mbps for stereo music exceeds most access rates, and even 64 kbps for speech exceeds the access rate for a dial-up modem user. For these reasons, PCM-encoded speech and music are rarely used in the Internet. Instead, compression techniques are used to reduce the bit rates of the stream. Popular compression techniques for speech include **GSM** (13 kbps), **G.729** (8 kbps), **G.723.3** (both 6.4 and 5.3 kbps), and a large number of proprietary techniques. A popular compression technique for near CD-quality stereo music is **MPEG 1 layer 3**, more commonly known as **MP3**. MP3 encoders typically compress to rates of 96 kbps, 128 kbps, and 160 kbps, and produce very little sound degradation. When an MP3 file is broken up into pieces, each piece is still playable. This headerless file format allows MP3 music files to be streamed across the Internet (assuming the playback bit rate and speed of the Internet connection are compatible). The MP3 compression standard is complex, using psychoacoustic masking, redundancy reduction, and bit reservoir buffering.

Video Compression in the Internet

A video is a sequence of images, typically being displayed at a constant rate—for example, at 24 or 30 images per second. An uncompressed, digitally encoded image consists of an array of pixels, with each pixel encoded into a number of bits to represent luminance and color. There are two types of redundancy in video, both of which can be exploited for compression. Spatial redundancy is the redundancy within a given image. For example, an image that consists of mostly white space can be efficiently compressed. Temporal redundancy reflects repetition from image to subsequent image. If, for example, an image and the subsequent image are exactly the same, there is no reason to re-encode the subsequent image; it is more efficient simply to indicate during encoding that the subsequent image is exactly the same.

The MPEG compression standards are among the most popular compression techniques. These include **MPEG 1** for CD-ROM-quality video (1.5 Mbps), **MPEG 2** for high-quality **DVD** video (3–6 Mbps), and **MPEG 4** for object-oriented video compression. The MPEG standard draws heavily on the JPEG standard for image compression by exploiting temporal redundancy across images in addition to the spatial redundancy exploited by JPEG. The **H.261** video compression standards are also very popular in the Internet. In addition there are numerous proprietary schemes, including Apple's QuickTime and Real Networks' encoders.

Readers interested in learning more about audio and video encoding are encouraged to see [Rao 1996] and [Solari 1997]. A good book on multimedia networking in general is [Crowcroft 1999].

CASE HISTORY

STREAMING STORED AUDIO AND VIDEO: FROM REALNETWORKS TO YOUTUBE

RealNetworks, a pioneer in audio and video streaming, was the first company to bring Internet audio to the mainstream. Its initial product—the RealAudio system released in 1995—included an audio encoder, an audio server, and an audio player. Allowing users to browse, select, and stream audio content from the Internet on demand, it quickly became a popular distribution system for providers of entertainment, educational, and news content.

Today audio and video streaming are among the most popular services in the Internet. Not only is there is a plethora of companies offering streamed content, but there is also a myriad of different server, player, and protocol technologies being employed. A few interesting examples (as of 2009) include:

- **Rhapsody from RealNetworks:** Provides streaming and downloading subscription services to users. Rhapsody uses its own proprietary client, which retrieves songs from its proprietary server over HTTP. As a song arrives over HTTP, it is played out through the Rhapsody client. Access to downloaded content is restricted through a Digital Rights Management (DRM) system.
- **MSN Video:** Users stream a variety of content, including international news and music video clips. Video is played through the popular Windows Media Player (WMP), which is available in almost all Windows hosts. Communication between WMP and the Microsoft servers is done with the proprietary MMS (Microsoft Media Server) protocol, which typically attempts to stream content over RTSP/RTP; if that fails because of firewalls, it attempts to retrieve content over HTTP.
- **Muze:** Provides an audio sample service to retailers, such as BestBuy and Yahoo. Music samples selected at these retailer sites actually come from Muze, and are streamed through WMP. Muze, Rhapsody, YouTube, and many other streaming content providers use content distribution networks (CDNs) to distribute their content, as discussed in Section 7.3.
- **YouTube:** The immensely popular video-sharing service uses a Flash-based client (embedded in the Web page). Communication between the client and the YouTube servers is done over HTTP.

What is in store for the future? Today most of the streaming video content is low-quality, encoded at rates of 500 kbps or less. Video quality will certainly improve as broadband and fiber-to-the-home Internet access become more pervasive. And very possibly our handheld music players will no longer store music—instead we'll get it all, on-demand, from wireless channels!