son processes [WTSW 95]. The implications of these ideas are still the subject of much debate [GB 96, HL 96, RE 96]. Some researchers argue that self-similarity is interesting, but of marginal consequence. Others doubt the validity of the measurements or their representativeness. However, a growing number of researchers believe that future network designers must factor in self-similar traffic into their designs [GB 96, PF 94]. We have yet to see the practical implications of this "revolution," but it is likely to play an increasingly important role in designing traffic-management algorithms of the future.

14.4 Traffic classes

We saw in Section 14.2 that networks that provide heterogeneous qualities of service (or *integrated service networks*) are likely to cost less than networks that provide a single quality of service. In this section, we study the expected service requirements, corresponding to utility functions, of applications in integrated services networks. Instead of studying the service requirements for individual applications, we find it convenient to place applications in *traffic classes* that represent the shared requirements of a set of widely used applications [Garrett 94]. Traffic classes also represent the types of service provided by the network. Because these correspond to application requirements, in the rest of the chapter, we will not distinguish between application classes and service classes, calling both "traffic classes."

14.4.1 Guaranteed-service and best-effort applications

We partition applications into two fundamental classes, depending on whether they require a performance guarantee from the network. The difference between *guaranteed service* and *best-effort service* is like flying with a reservation and flying standby. With a reservation, an airline guarantees that with high probability, you will get a seat on the airplane. The airline thus guarantees you bandwidth (one seat), a nominal delay (which may be affected by weather conditions), and a loss probability (that the flight is overbooked, and you are "bumped"). On the other hand, if you travel standby, you may fly or you may not. The trade-off is that flying standby is cheaper, and, if you have time to spare, you may not care about the uncertainty.

An integrated service network guarantees traffic from a *guaranteed-service (GS)* application a particular quality of service, if its traffic obeys a given traffic descriptor (see Section 13.3.1 for a discussion of traffic descriptors). Typical guaranteed-service applications include audioconferencing, telephony, videoconferencing, remote sensing, video-on-demand, interactive multiplayer games, distance learning, and collaborative environments. With these applications, users derive utility from the network only if the network bounds the delay and provides a minimum amount of bandwidth. The utility function for a guaranteed-service application penalizes traffic that does not meet its ser-

vice requirement, which is typically described by parameters for three orthogonal quantities: *bandwidth*, *delay*, and *loss*.

- Typical bandwidth parameters are the minimum bandwidth and the sustained bandwidth.
- Typical delay parameters are the mean delay, the worst-case delay, the 99-percentile delay, and the delay jitter.
- Typical loss parameters are the loss probability and a bound on the maximum number of packets lost over every consecutive set of packets transmitted.

We describe these performance parameters in more detail in Section 9.2.3.

An integrated-services network does not guarantee *best-effort (BE)* applications a service quality. The network undertakes to deliver BE packets only when bandwidth is available, and packets may be dropped at any time. Typical best-effort applications are the ones found on the Internet today, such as file transfer, name service, email, network news, and the World Wide Web. These applications are willing to adapt to whatever quality of service is available. The utility function for a best-effort application does not degrade significantly with a drop in service quality. More precisely, unlike a guaranteed-service application, a best-effort application derives utility from the network even if its packets suffer large delays, or it receives only a small bandwidth allocation from the network.

Guaranteed-service and best-effort applications also differ in the *degree of synchrony* between the endpoints. We define this to be the time scale at which the peer applications or their users interact with each other. With *synchronous* or *interactive* applications, the endpoints interact with each other on the time scale of a round-trip-time propagation delay. For example, with a telephone call, a user's actions (such as replying to a question) depend on the actions of his or her peer (such as asking a question) about a round-trip-time ago. Synchronous applications require the network to guarantee them a bandwidth and delay bound. Thus, they naturally are guaranteed-service applications. With *bounded-asynchronous* or *noninteractive* applications, the sender and receiver interact with each other on the time scale of a few minutes to a few days. Traffic from bounded-asynchronous applications can be delayed in the network if necessary, or even dropped, because the delay from retransmitting lost information does not affect user utility. Thus, these applications fall naturally into the best-effort class. In other words, the separation between guaranteed-service and best-effort applications also partitions applications based on their inherent degree of synchrony and interactivity.

A third way to look at the GS–BE dichotomy is based on an application's sensitivity to time and delay. GS applications are also called *real-time* applications, because their utility depends on the real (as opposed to virtual) time. BE applications are also called *elastic* applications, because they can elastically adapt to changes in network quality of service. Many applications on the Internet (as of 1996), such as FTP, telnet, and interactive chat, are elastic applications. They perform well if sufficient resources are avail-

able, but automatically scale back when resources become scarce. Although WWW is currently supported on a best-effort network, it would probably do a lot better if at least some Web sites could be accessed using guaranteed-service connections. In this sense, the Web is an application of the future trapped in a network of the past!

14.4.2 Traffic subclasses

We can further subdivide guaranteed-service and best-effort applications based on their relative sensitivity to bandwidth and delay. The two standardization bodies involved in setting standards for integrated services networks are the *ATM Forum*, which is a commercial association of ATM equipment manufacturers and researchers, and the *Internet Engineering Task Force* or *IETF*, which is the Internet's standardization body. Both bodies have proposed a tentative classification of guaranteed-service and best-effort applications into subclasses. The ATM Forum classifies applications based on their *bandwidth* sensitivity, and the IETF based on sensitivity to *delay* [Garrett 94, CWSA 95] (Table 14.1). Keep in mind, though, that what follows is a snapshot of the standardization efforts as of late 1995. It is possible, and even likely, that as we learn more about engineering integrated services networks, these classifications will change.

14.4.3 ATM Forum subclasses

The ATM Forum subdivides the guaranteed-service class into *constant bit-rate (CBR)* and *variable bit-rate (VBR)* subclasses. A CBR application generates a constant cell-smooth traffic stream (that is, with the same spacing among all transmitted cells) and expects that the receiver will receive the stream with a small delay jitter, also called *cell delay variation*. CBR applications model applications on the current circuit-switched telephone network and are implicitly assumed to require a minimal-delay, low-delay-jitter service. CBR service thus models telephone service, with the bonus that a stream can reserve bandwidth at an arbitrary rate, not just a multiple of 64 Kbps.

	Guaranteed service *(synchronous, interactive, real-time)*		Best effort *(bounded-asynchronous, noninteractive, elastic)*		
Bandwidth sensitivity *(ATM Forum)*	Constant bit-rate (CBR)	Variable bit-rate (VBR)	Available bit-rate (ABR)	Unspecified bit-rate (UBR)	
Delay sensitivity *(IETF)*	Intolerant (guaranteed service)	Tolerant (controlled load service)	Interactive burst	Interactive bulk	Asynchronous bulk

Table 14.1: Traffic subclasses according to the IETF and the ATM Forum.

Variable bit-rate

The VBR subclass models applications that generate traffic in bursts, rather than in a smooth stream. A VBR source is modeled as having an intrinsic long-term bit rate (called its *sustained cell rate*), with occasional bursts (of limited length) at a rate as high as a specified peak rate. A VBR application expects that the network will carry its bursty stream with minimal delay. However, since VBR sources are likely to be multiplexed together to obtain statistical multiplexing gain (see Section 4.4), we implicitly assume that VBR applications can tolerate higher delays and higher delay variations than can CBR sources. VBR applications can specify a worst-case end-to-end delay, a worst-case delay jitter, and a worst-case loss fraction. Typical applications in the VBR class are those sending compressed video streams (such as video-on-demand) where the bit rate varies with the degree of achievable compression.

Constant bandwidth versus constant quality

Variable bit-rate service was principally designed for carrying compressed video traffic [VPV 88]. Video streams need to be compressed because uncompressed streams take up too much bandwidth (up to 270 Mbps per stream, depending on the picture size and resolution). Moreover, they can be compressed by a factor of 30 to 100 with a fairly small loss of resolution.

We can compress video streams in one of two ways. With constant-bit-rate compression, the output of the compression engine (also called a *video coder*) is constant bit-rate, but variable quality. Specifically, scenes with a lot of motion or flashing colors take up much more bandwidth than other scenes. To preserve a constant bandwidth, the coder must degrade the spatial or temporal resolution of such scenes. We can avoid this by coding at a constant quality, but a variable bit-rate. With such coders, scenes with more visual information take up more bandwidth than others. Compressed video with a constant bit-rate is suitable for CBR service, and compressed video with a variable bit-rate is suitable for VBR service.

Most video coders available in 1996 are constant-bit-rate coders, because variable-bit-rate service is not widely available. These coders generate traffic at 64 Kbps (to fit in a single telephone circuit) or at multiples of 64 Kbps, such as 384 Kbps. The next generation of coders, based on the *Moving Pictures Expert Group* or *MPEG* video coding standard, are likely to be variable-bit-rate coders. These coders will probably use variable-bit-rate service from future integrated services networks.

Unspecified bit-rate and available bit-rate

The ATM Forum divides best-effort services into two subclasses. The *unspecified bit-rate* or *UBR* class comes closest in spirit to the current service on the Internet. A UBR source neither specifies nor receives a bandwidth, delay, or loss guarantee. It is assumed

to be able to deal with fluctuations in these parameters by using techniques such as forward error-correction or application-level flow control. In contrast, *available bit-rate (ABR)* service guarantees a zero-loss rate if sources obey the dynamically varying traffic management signals from the network. The network uses *resource management cells* (Section 13.4.9) to inform an ABR source about the currently available bandwidth at the bottleneck in its path. If the source obeys these signals, it is guaranteed zero loss. However, the network does not need to guarantee a delay or mean bandwidth bound (sometimes, the network may guarantee a minimum bandwidth). The difference between an ABR application and a UBR application is that an ABR application is willing to listen to resource management signals to obtain a zero loss bound. An ABR application differs from a VBR application in that a VBR application need not alter its behavior in response to network signals: sufficient resources are reserved for the source that its performance bounds are met. In contrast, an ABR application is guaranteed only a zero *loss* bound, and it must follow the network's orders to obtain this bound. A network provider will presumably compensate for this inconvenience by charging a lower price for ABR service than for VBR service. Common Internet applications such as FTP, WWW, and telnet are likely to be the initial applications using UBR/ABR service.

14.4.4 IETF subclasses

Guaranteed-service applications

The IETF divides the guaranteed-service class into *tolerant* and *intolerant* subclasses based on an application's sensitivity to delay [SCZ 93, BCS 94]. A tolerant application requires a nominal mean delay, but is tolerant of "occasional" lapses from this mean. In other words, its utility function does not degrade much if some of its packets get a large delay. An example of a tolerant application is an interactive-voice application that is willing to drop packets delayed significantly beyond the mean. The IETF does not quantify how often the delay bound can be violated or by how much, since this requires extensive source characterization, which is difficult to achieve in real life.

In contrast, *intolerant* applications require a worst-case delay bound and cannot tolerate a delay larger than the bound (in other words, its utility function degrades significantly if packets are delayed beyond the bound). An example of an intolerant application is a multiparticipant interactive game, where user satisfaction derives from a small response time. Both tolerant and intolerant applications implicitly require either a constant or variable bit-rate bandwidth guarantee. Both implicitly require a guarantee of low loss, though the IETF proposals do not address this issue.

The network serves tolerant applications with *controlled-load* service. With this service, during call establishment, the user informs the network of its expected traffic pattern. The network (using an unspecified admission control algorithm) denies a call if it will appreciably degrade the service quality of existing calls. The precise specifications for controlled-load service are still being defined. Intolerant applications are served with *guaranteed service*, which is nearly identical to VBR service in ATM networks.

Best-effort applications

The IETF divides best-effort applications into three subclasses, based on their delay sensitivity. The *interactive burst* subclass models applications such as paging and messaging. These applications require bounded asynchronous service, where the bound is fairly tight (at least at human time scales). The second class of service is the *interactive bulk* class, such as file transfer, where a human being may be waiting for the transfer to complete. The network should give such applications a lower delay than traffic from the *asynchronous bulk* class, which are truly asynchronous. An example of an application in the asynchronous bulk subclass is Usenet (Internet news).

14.4.5 Some notable points

Three points about providing heterogeneous qualities of service are worth noting.

- Current networks provide only a limited service menu. The Internet gives all applications a single best-effort quality of service. The telephone network essentially provides only voice-quality calls, though users can also purchase multiples of a single voice call (for example, as a DS1 or DS3 circuit). Recently, specialized packet-switched networks called *frame-relay* networks have been built mainly for interconnecting geographically separated LANs. Frame-relay service does allow LAN administrators to ask for a sustained rate and a peak rate, similar to the ATM Forum VBR specification. However, all applications on the wide-area link share this specification. Thus, it is a step in the right direction, but still far removed from the capabilities we expect from a full-scale integrated service network.

- We have focused here on *application* requirements. Besides these, both the IETF and the ATM Forum recognize the need for meeting *organizational* quality-of-service requirements. For example, if several organizations share a wide-area link, they may want to partition the link into prespecified bandwidth shares when they overload the link. If organizations A and B each pay for half the cost of a link, they may want to ensure that when both have traffic to send, the link sends equal amounts from A and B, and when one is inactive, the other gets the excess capacity. This link-sharing agreement constrains the set of applications that can be admitted on the link from each organization. It also constrains the service order among packets from the two organizations. We will study these interactions in Section 14.6.

- We have concentrated on numerically expressed quality-of-service parameters for three performance metrics: delay, bandwidth, and loss. Users may be sensitive to other qualities of service that we have not considered here. Examples are media synchronization (such as voice synchronization with video), security, availability, freedom from billing errors, and privacy of billing records. Although we recognize the need for providing quality of service along these

dimensions, they are orthogonal to the performance metrics described above. Moreover, many of these have been extensively studied and implemented in the current telephone network and the Internet. Therefore, we will not study them in this book.

14.5 Time scales of traffic management

Let us return to our busy executive participating in a videoconference and examine some steps the network must take to meet the service requirements of her voice and video traffic. Video (depending on how we code it) may require nearly a hundred times more bandwidth than voice, but voice is more sensitive to delay. Therefore, the network must be made aware of the bandwidth requirements of both streams and told that if a queue builds up, link schedulers should give voice packets priority over video packets. The videoconferencing application communicates with the network using *signaling*, which is done just once, when the call is set up. Traffic-management mechanisms operating at this time scale, by choosing an appropriate route through the network, and by reserving sufficient resources at each multiplexing point, ensure that the voice and video transferred during the call meet their bandwidth and delay requirements. After call setup, while the conference is going on, the network must make scheduling decisions at a much faster time scale. Moreover, if the video application is required to obey a traffic descriptor, it may need to regulate traffic at this time scale. Thus, to provide a videoconferencing service, the network must provide traffic management not only at the time scale of a session, but also at a very fast time scale corresponding to packet scheduling. Traffic-management mechanisms operating at multiple time scales cooperate to serve traffic efficiently and maximize the utility delivered by the network.

Table 14.2 outlines the five time scales at which we must control a network, the mechanisms at each time scale, and whether these mechanisms operate in the network, at the endpoints, or both [Keshav 91]. We have already studied some of these mechanisms (shaded in the table) in earlier chapters. In the rest of the chapter we will study the remaining mechanisms in more detail. Bear in mind that at each time scale, the network operator seeks to carry out the same optimization: to maximize overall user satisfaction at least cost.

14.5.1 Less than one round-trip time

One round-trip time (RTT) is the smallest time between sending a message and getting a response from the network or destination. Depending on the network diameter, this can be anywhere from hundreds of microseconds to hundreds of milliseconds. The main idea is that at this time scale, because the source cannot adapt to network conditions, all control is open-loop. The mechanisms that operate at this time scale are *scheduling, traffic regulation and policing,* and *forward error correction.* This time scale is also called the *cell-level* time scale.

Time scale	Mechanism	Network	Endpoint	Reference
Less than one round-trip time (cell level)	Scheduling and buffer management	X	X	Section 14.6
	Regulation and policing	X	X	Section 13.3
	Routing (datagram networks)	X	X	Chapter 11
	Error detection and correction	X	X	Chapter 7
One or more round-trip times (burst level)	Feedback flow control	X	X	Section 13.4
	Retransmission		X	Section 12.4.7
	Renegotiation	X	X	Section 14.7
Session (call level)	Signaling	X	X	Section 14.8
	Admission control	X		Section 14.9
	Service pricing	X		Section 14.2.2
	Routing (connection-oriented networks)	X		Chapter 11
Day	Peak-load pricing	X		Section 14.10
Weeks or longer	Capacity planning	X		Section 14.11

Table 14.2: Time scales of control, and the mechanisms at these time scales.

14.5.2 One or more round-trip times

At the multiple RTTs time scale, endpoints can *react* to changes in the network, thus allowing them to scale their load in response to network state dynamically. The mechanisms operating at this time scale for best-effort sources are *feedback flow control* and *retransmission*. Because it takes at least one RTT for a source to inform every scheduler along the path about its traffic descriptor and service requirement, the multiple RTT time scale is the shortest time-scale at which applications in the guaranteed-service class can *renegotiate* their traffic and service descriptors. This is also called the *burst-level* time scale.

14.5.3 Session

The session or call-level time scale is the time over which applications establish, use, and tear down an end-to-end connection. Guaranteed-service applications declare their traffic descriptors and resource requirements using the *signaling* mechanism at this time scale. To meet these requirements, the network must do *admission control*, allowing some calls and denying others. In connection-oriented networks, *routing* is done at this time

scale. *Service pricing*, which limits call volumes, also operates at this time scale. Connectionless networks do not make a distinction between the multiple-RTT and session time scales.

14.5.4 Day

A strong diurnal cycle, based on the working day, dominates network usage. Usage peaks during working hours, with a dip during lunchtime, and slacks off as night progresses. A network provider can use *peak-load pricing* to shift part of the peak load to off-peak hours, thus decreasing the peak load.

14.5.5 Weeks or longer

Over a longer time scale, the network provider can dynamically adapt network topology to match traffic demands. Since link and switch provisioning takes time and can be expensive, these changes are carried out in the weeks-to-months time scale.

14.6 Scheduling

What scheduling discipline must we use to simultaneously satisfy the performance requirements of guaranteed-service and best-effort applications? We have seen in Chapter 9 that scheduling disciplines such as weighted fair queuing and rate-controlled static priority scheduling allow individual connections to obtain guarantees on bandwidth, delay, and delay jitter. Thus, packets from guaranteed-service sources should be scheduled according to one of these disciplines. These sources should reserve enough resources (such as a service weight for WFQ) to meet their performance requirements. In contrast, packets from best-effort sources should receive fair service, as described in Chapter 9.

We can meet the performance bounds of both GS and BE connections by implementing a scheduler with multiple priority levels, where the highest priority level is devoted to packets from GS connections, and the lower levels to packets from BE connections. Because packets from GS connections have higher priority than packets from BE connections, BE traffic interferes minimally with GS traffic. Similarly, we can give delay-sensitive BE applications lower delays by scheduling them at a higher priority level than delay-insensitive BE applications. Although a connection assigned to a higher-priority BE level is not guaranteed an absolute delay bound, it gets a smaller delay than lower-priority BE connections.

14.6.1 Hierarchical link sharing

We mentioned earlier that scheduling should meet not only individual, but also organizational performance requirements. Although we could use the disciplines in Chapter 9, aggregating connections from the same organization, this does not completely capture