

a given type of traffic can have monetary implications. For example, it is common to pay for a long-distance phone call (on a per-call basis). One cannot be expected to pay for a phone call if the call does not receive the necessary network resources to ensure high quality. Other services might not have such monetary implications and might be offered on a best-effort basis. Such a situation creates an incentive for certain applications (or individuals) to steal better service without paying for it. Therefore, we need authentication functions to identify users and ensure that a given user is who he or she claims to be. Furthermore, a given user may be entitled to a given level of QoS under certain circumstances but not under other circumstances.

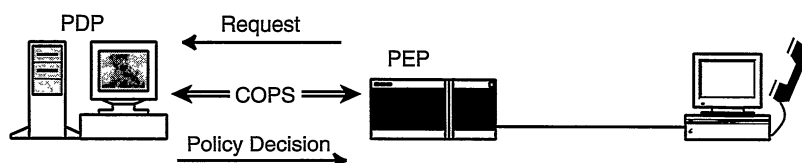
Finally, the Internet has already seen *denial of service* (DoS) attacks, in which certain systems (or persons) flood the network with traffic, causing congestion and service failure for everyone else. Therefore, certain rules must be established to specify which circumstances lead to which actions, and there must be methods for ensuring that those rules are applied. This can be considered as a type of policing function.

Appropriately, the IETF has developed a protocol known as the *Common Open Policy Service protocol* (COPS), which RFC 2748 specifies. COPS is a client-server protocol, as depicted in Figure 8-2. A *Policy Enforcement Point* (PEP), such as a router that needs to enforce certain rules, queries a *Policy Decision Point* (PDP) that makes the actual policy decisions. The PEP can be considered the policeman and the PDP can be considered the judge. Given these analogies, the acronym COPS is particularly fitting.

The Resource Reservation Protocol (RSVP)

The previous sections of this chapter gave a brief overview of QoS issues and solutions. Next we delve deeper into the technical solutions and start with RSVP.

Figure 8-2
COPS
request/response



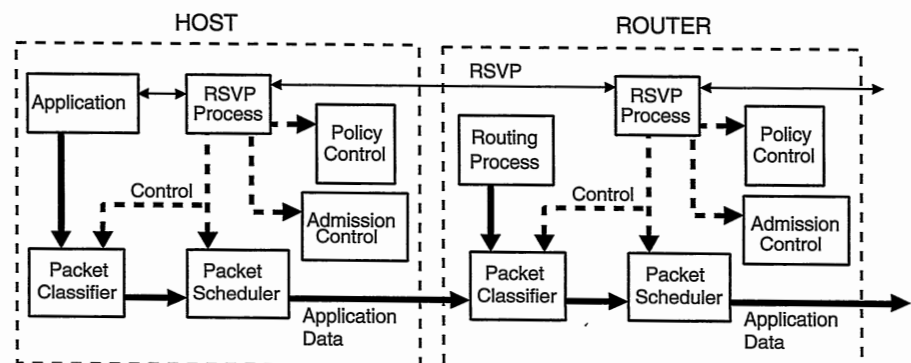
In order to use RSVP, routers and hosts must implement certain functions. These functions are shown in Figure 8-3. Prior to data transfer when a QoS reservation is being established, the admission control function determines whether sufficient resources exist to satisfy the requested QoS, and the policy control function determines whether the user has the authority to make the requested reservation. The policy control function could, for example, use COPS to communicate with a PDP to make the policy determination. During data transfer, the packet classifier determines the QoS to be applied to a given packet, and on the outgoing interface, the packet scheduler or other link-layer function determines when a given packet is to be forwarded. Together, the packet classifier, packet scheduler, and admission control are known as traffic control.

RSVP Syntax

Before describing the messages and functions of RSVP in greater detail, we must first mention a few points about the syntax of RSVP messages. The syntax of the information contained in RSVP messages is specified in RFC 2215, "General Characterization Parameters for Integrated Service Network Elements." Basically, RSVP is used to implement a number of services and these services are identified by particular service numbers. For example, the guaranteed service has the service number 2 and the controlled-load service has the service number 5. A number of parameters are associated with each service, and each parameter has a parameter number.

Both services and parameters are specified in messages with the format *Type-Length-Value* (TLV). Thus, a type field identifies each service or parameter, followed by a length field, followed by the data itself.

Figure 8-3
RSVP within hosts
and routers

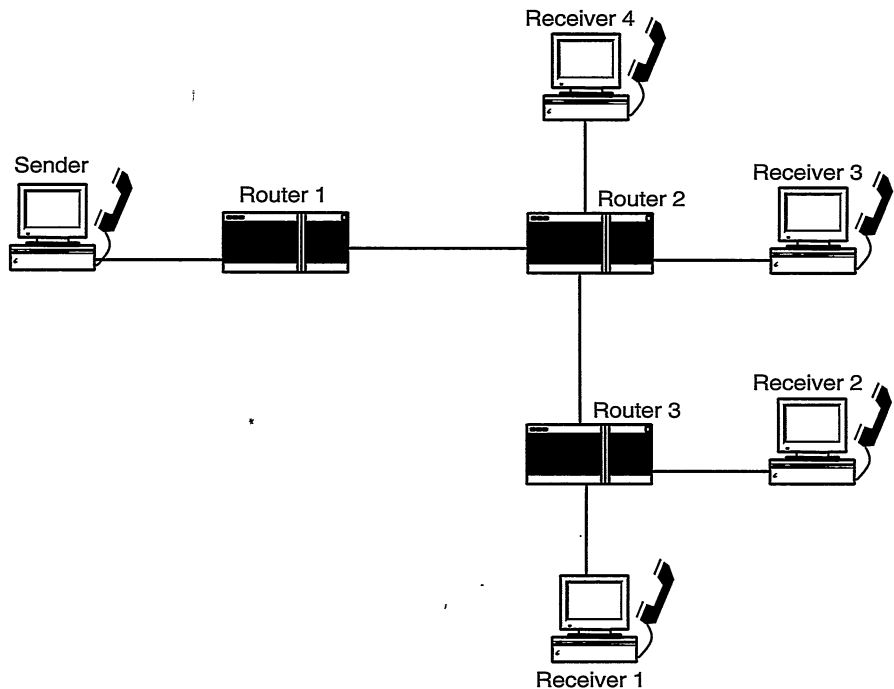


In some cases, data needs to be sent that is not associated with any particular service or that is common to a number of different services. For this purpose, service number 1 is defined. If a message is to be sent and contains both general and service-specific information, then the message will specify service number 1 containing the general information parameters, followed by service information (such as for service number 2) containing service-specific information.

Establishing Reservations

As previously described, RSVP reserves resources along a path from the receiver of user data back to the sender of that data. The reason is because RSVP is designed with multicast in mind. With multicast, there may be a limited number of senders and multiple receivers. Figure 8-4 depicts a scenario where only one sender and many receivers are used. From sender to receiver, the data path branches. In the reverse direction, the data path merges. Let's assume that Receivers 1 and 2 have different QoS objectives.

Figure 8-4
RSVP used with
multicast



In that case, the resources needed between Routers 2 and 3 do not need to be the sum of the requirements of Receivers 1 and 2, just the maximum. Equally, the resources required between Routers 1 and 2 do not need to be the sum of the requirements of all four receivers, just the maximum.

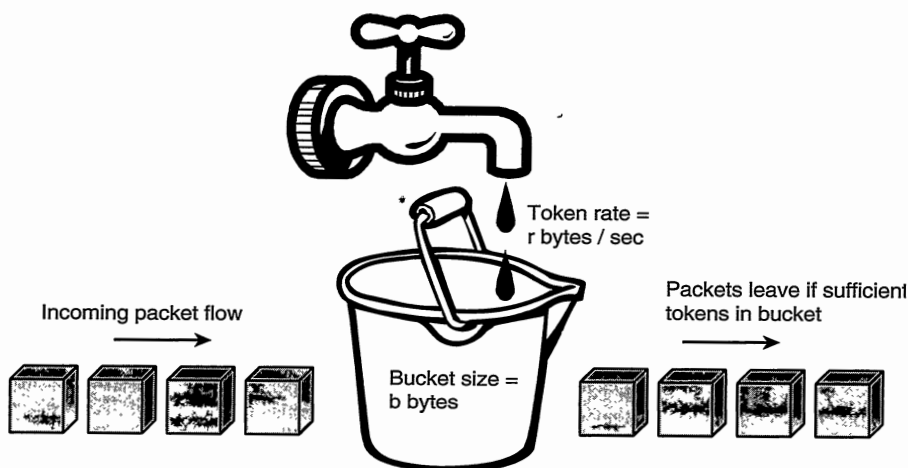
RSVP deals with unidirectional data transfer only. In a typical two-party VoIP conversation, resource reservation must be carried out in both directions, assuming that both parties intend to speak.

The sender of data sends a PATH message to the intended receiver of the data and includes within that message a TSpec, which describes the characteristics of the data to be sent. The receiver responds with an RESV message, which contains a flowspec indicating the type of resource reservation required. In the case of multiple receivers, the flowspec might be modified as it is passed back towards the sender as a result of different receivers having different QoS requirements. This is called *merging flowspecs*.

TSpec A TSpec is sent from a sender in a PATH message and is also included in the information sent from a receiver to a sender in an RESV message. It takes the form of a token bucket specification plus a *peak rate* (p), a *maximum packet size* (M), and a *minimum policed unit* (m). The token bucket specification defines a *bucket size* (b) and a *token rate* (r), as depicted in Figure 8-5.

This conceptual bucket has a capacity of b bytes and is filled with tokens at a rate of r bytes per second. Upon arrival of a packet, the packet will be

Figure 8-5
Token bucket



transmitted onwards only if the number of tokens in the bucket is at least as large as the packet. Otherwise, the packet might be dropped or queued until the bucket accumulates enough tokens. If the incoming flow is less than the taken bucket rate r , then the bucket will eventually fill up, at which point the generation of tokens is suspended until such time as a packet arrives to consume some tokens. Provided that the average rate flowing toward the bucket is less than or equal to r , the token bucket should not constrain the flow. Furthermore, the rate of flow toward the bucket may exceed r for a period of time, but it cannot do so indefinitely without running out of tokens and being subject to delay or packet loss as a result. Note that a token bucket is not a physical container of packets, but a numerical counter of tokens to schedule the releasing time of packets.

A token bucket approach is more flexible than the classic leaky bucket scheduling mechanism. A leaky bucket technique maintains a constant outward flow regardless of the incoming flow rate. Although a token bucket technique can constrain flow, it does enable variations in the incoming flow rate and permits the outbound flow rate to vary in accordance with the incoming flow rate.

The p , which should be greater than r , is the maximum data rate that the sender will use. The sender should not use that rate for any extended period of time, since to do so would be incompatible with the token bucket specification. The M is the maximum size of IP data packet that the sender will send. If any packets exceed that size, then they will be considered to be outside the limits of the TSPEC and will not be treated with the same QoS as packets that are within the specification.

The m is measured in bytes. All packets less than m bytes in length are considered to be m bytes long for two reasons. The first is to enable nodes in the network to estimate the amount of resources that will be needed to process each packet. The second is to bound the bandwidth overhead consumed by link-level headers. The maximum bandwidth overhead of link-layer headers will be the ratio of link-layer header size to m . Putting all these components together, Figure 8-6 shows the format of a sender TSPEC. The Token Bucket TSPEC is defined in RFC 2215 as having parameter number 127.

Flowspec The flowspec, sent in an RESV message from a receiver, contains a TSPEC of its own, plus an indication of the QoS control service being requested. Currently, two services can be requested: the guaranteed service and controlled-load service. The flowspec for controlled-load service is depicted in Figure 8-7 and the flowspec for guaranteed service is depicted in Figure 8-8.

Figure 8-6

Sender TSpec

3	3	2	2	2	2	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V = 0		Reserved										Overall length = 7 (excluding header)																	
Service header = 1.		0	Reserved										Length of service 1 data = 6																
127		Flags (0)										Parameter length = 5 (excluding header)																	
Token Bucket Rate (r) (32-bit IEEE floating point number)																													
Token Bucket Size (b) (32-bit IEEE floating point number)																													
Peak Data Rate (p) (32-bit IEEE floating point number)																													
Minimum Policed Unit (m) (32-bit integer)																													
Maximum Packet Size (M) (32-bit integer)																													

V = message format version

Figure 8-7

The flowspec for controlled-load service

3	1	3	0	2	9	8	2	7	6	5	4	3	2	2	1	2	0	9	8	1	7	6	5	4	3	2	1	1	0	9	8	7	6	5	4	3	2	1	0	
V = 0				Reserved												Overall length = 7 (excluding header)																								
Service header = 5				0	Reserved												Length of service 5 data = 6																							
127				Flags (0)												Parameter length = 5 (excluding header)																								
Token Bucket Rate (r) (32-bit IEEE floating point number)																																								
Token Bucket Size (b) (32-bit IEEE floating point number)																																								
Peak Data Rate (p) (32-bit IEEE floating point number)																																								
Minimum Policed Unit (m) (32-bit integer)																																								
Maximum Packet Size (M) (32-bit integer)																																								

V = message format version

The flowspec for controlled-load service is simply a TSpec. The flowspec for guaranteed service is a TSpec plus a *Rate* (R) term and a *Slack* (S) term. The R term indicates the bandwidth required and should be equal to or greater than r . If it is greater, then extra bandwidth is being requested, which will reduce queuing delays. The S term is a number of microseconds. It indicates the difference between the desired delay and the delay that would be achieved if the rate R is used. A network element may use the Slack to reduce the amount of resources reserved for this flow if necessary.

The flowspec for guaranteed service

3	1	0	9	8	7	6	5	4	3	2	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
V = 0				Reserved												Overall length = 10 (excluding header)																										
Service header = 2						0	Reserved												Length of service 2 data = 9																							
127						Flags (0)												Parameter length = 5 (excluding header)																								
Token Bucket Rate (r) (32-bit IEEE floating point number)																																										
Token Bucket Size (b) (32-bit IEEE floating point number)																																										
Peak Data Rate (p) (32-bit IEEE floating point number)																																										
Minimum Policed Unit (m) (32-bit integer)																																										
Maximum Packet Size (M) (32-bit integer)																																										
130						Flags (0)												Parameter length = 2 (excluding header)																								
Rate (R) (32-bit IEEE floating point number)																																										
Slack Term (S) (32-bit integer)																																										

V = message format version

Filter Spec An RSVP session is identified only by a destination IP address and a protocol ID. The session identification can optionally contain a destination port number. Thus, an RSVP session identification does not include any information about the sender of data. Given that the destination port is optional, and the session is not identified by any sender-related information, a problem might occur in determining the specific data flow for which a given reservation is requested. For example, a given receiver might be receiving two streams of data from two different senders (as could be the case in a three-party VoIP call). A need exists to indicate what QoS is being requested for the flow from each sender, but the RSVP session identification itself does not include sender information. This is where the filter spec comes into play.

A filter spec defines the set of data packets (that is, the flow) to which a particular QoS is to be applied, and it is an input to the packet classifier function. The format of the filter spec depends on whether IPv4 or IPv6 is being used. At a minimum, the filter spec contains a sender IP address and optionally a sender port number. The IP address is used to identify the sender of the data to which the requested QoS is to be applied. The source port number is used for situations such as a video conference where both a video and an audio stream are set up from a given sender to a given receiver with a different QoS requirement for each stream.

Given that the filter spec can contain a port number, each router in the path must be able to examine the details of the header to determine the port number, which raises a number of considerations. First, IP datagrams in the flow must not be fragmented, which in turn means that we must use path *maximum transmission unit* (MTU) discovery. Second, IP security procedures might encrypt the header, which means that RSVP must include security functions to decrypt the header. Third, the IPv6 header is of variable length, which might cause a significantly greater processing effort at routers.

The filter spec is included in PATH messages from the sender to the receiver and in RESV messages from the receiver to the sender. When sent in a PATH message, it is known as a sender template.

ADSpec As previously described, a sender first sends a PATH message downstream towards one or more receivers. The TSpec in the PATH message indicates the type of data to be sent and is passed from the sender to the receiver without modification. The receivers respond with an RESV message to indicate the QoS requested and the RESV is propagated upstream towards the sender.

Having a receiver request a QoS that intervening routers cannot meet is pointless, however. For example, the RESV message might specify a bandwidth requirement that exceeds the bandwidth of one of the links in the path from sender to receiver. Therefore, a need exists for a receiver to be informed about the network between the sender and receiver, so that the receiver does not request something that the network cannot provide. To meet this need, RSVP includes the capability for the sender and routers to indicate their QoS capabilities within a PATH message from the sender to receiver. This is done through the use of an ADSpec, where the AD indicates advertising.

A sender constructs an initial ADSpec, indicating what it can support, and it includes the ADSpec in the PATH message. At each router along the path from the sender to receiver, the ADSpec is updated and passed on. By the time the PATH message gets to the receiver, the ADSpec provides a pretty good indication of what the receiver can reasonably request in terms of QoS from the routers and from the sender.

One interesting point about the ADSpec is that it can also indicate that one or more routers in the path from the sender to receiver does not support RSVP. Such a situation is perfectly possible. In the event that an RSVP-capable router sends a PATH message on a link to a router that does not support RSVP, or if an RSVP-capable router receives a PATH message on a link from a router that does not support RSVP, then the RSVP-capable

router sets a bit in the ADSpec to indicate that RSVP is not supported end to end.

The general format of the ADSpec is depicted in Figure 8-9. The format includes general information (service number 1), optionally followed by either or both guaranteed service information and controlled-load service information.

The general information specifies whether a *nonintegrated-service-capable* (non-IS) hop is involved in the path (bit marked X in Figure 8-9). For the purposes of RSVP, non-IS means lacking RSVP support. If this bit is set, then the information in the rest of the ADSpec is no longer relevant, since QoS has to be end to end and an obvious break has taken place in the QoS chain from the sender to receiver.

The ADSpec also indicates the number of hops between IS-capable nodes, the path bandwidth, the path MTU, and the minimum path latency. The minimum path latency is the delay that would be experienced along the path if no additional queuing delays took place.

Following the general information in the ADSpec, a guaranteed service data fragment and/or a controlled-load data fragment may contain information related to those two services. These data fragments are inserted if the application might use either or both services for the session in question.

Figure 8-9

The format of the ADSpec

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	
V = 0		Reserved														Overall length (excluding header word)																
Service number = 1		X	Reserved														Length of general parameters = 8															
Param ID = 4 (# of IS hops)		Flags														Parameter 4 length = 1																
IS Hop count (32-bit unsigned integer)																																
Param ID = 6 (path bandwidth)		Flags														Parameter 6 length = 1																
Path Bandwidth Estimate (32-bit IEEE floating point number)																																
Param ID = 8 (min path latency)		Flags														Parameter 8 length = 1																
Minimum Path Latency (32-bit integer)																																
Param ID = 10 (composed MTU)		Flags														Parameter 10 length = 1																
Composed MTU (32-bit unsigned integer)																																
Guaranteed Service Fragment (service 2) (optional)																																
Controlled-Load Service Fragment (service 5) (optional)																																

V = message format version

RSVP Messages We have already discussed a number of RSVP messages, such as PATH and RESV, and have also looked at some of the information carried in these messages. We now look at the format of these message and others in a little more detail.

Every RSVP message contains a common header, as shown in Figure 8-10. The various messages are Path (1), Resv (2), PathErr (3), ResvErr (4), PathTear (5), ResvTear (6), and ResvConf (7). The checksum is used to ensure that the message has been received without error. The Send_TTL is the IP TTL (time to live) value of the message. With normal IP forwarding, RSVP can use this field to determine if a non-RSVP hop has been involved by comparing the value in the RSVP header with the IP TTL value, because a non-RSVP node will update the IP TTL value but not the RSVP Send_TTL.

Following the common header, the message contains a number of objects, such as a sender TSpec, an ADSpec, and so on. Each object has a header and contents, as shown in Figure 8-11. The Class-num and C-type identify the object. The Class-num identifies the object and the C-type enables a different version of the object to exist. For example, the format of a filter spec varies depending on whether IPv4 or IPv6 is being used. In that case, the Class-num indicates that the object is a filter spec and the C-type indicates whether it is the IPv4 or IPv6 variant.

The following are examples of the object types defined in RSVP. The list is not exhaustive, and the reader should consult RFC 2205 for a complete listing and associated definitions:

- **SESSION Class** This has a Class-num value of 1. It has a C-type value of 1 for IPv4 and a C-type value of 2 for IPv6. This object

Figure 8-10
The RSVP common header

Byte 0		Byte 1	Byte 2	Byte 3
Vers = 1	Flags (0)	Message type	RSVP checksum	
Send_TTL		Reserved	RSVP length	

Figure 8-11
The RSVP object format

Byte 0	Byte 1	Byte 2	Byte 3
Length (bytes)		Class-num	C-type
Object contents			

includes the IP destination address, the protocol ID for the type of data being sent, and optionally the destination port.

- **FLOWSPEC Class** This has a Class-num of 9 and a C-type value of 2 (the C-type value of 1 is obsolete). The flowspec is described earlier in this chapter.
- **SENDER_TEMPLATE Class** This has a Class-num value of 11. For IPv4, the C-type value is 1 and for IPv6, the C-type value is 2. The sender template is sent in a Path message and is, in fact, a filter spec sent by the sender.
- **RSVP_HOP Class** This object contains the IP address of the interface through which the last RSVP-capable node passed this message. The object is used in the Path message and is saved at each node along the route. The purpose is to ensure that an RESV message from the receiver to the sender follows the same path back through the network as was taken by the PATH message. This object has a Class-num value of 3, a C-type value of 1 for IPv4, and a C-type value of 2 for IPv6.
- **TIME_VALUES Class** This is a timeout period in milliseconds and indicates how long the message is to be considered valid. This object has a Class-num value of 5 and a C-type value of 1.
- **ERROR_SPEC Class** This is an object that is included in RSVP error messages. The object includes the IP address of the node at which the error was detected, plus an error code identifying the type of error, plus some additional error information. This class has a Class-num value of 6. The C-type value is 1 for IPv4 and 2 for IPv6.
- **STYLE Class** We can select different reservation styles in the case of multiple receivers and/or multiple senders. A fixed-filter style means that a receiver uniquely identifies the individual sender and the corresponding data stream from that sender to which a given reservation is to apply. A wildcard-filter style indicates that the reservation request applies to all data streams from all senders in a given session. The shared-explicit style lists specific senders, indicating that a given request applies to data streams from those senders. The applicable Class-num value for the STYLE class is 8 and the C-type value is 1.

Example Reservations Figures 8-12 and 8-13 show two examples of successful reservations. Figure 8-12 shows a simple case of one sender and one receiver. In this case, a PATH message is propagated through the

Figure 8-12

Successful reservation
for a single sender
and a single receiver

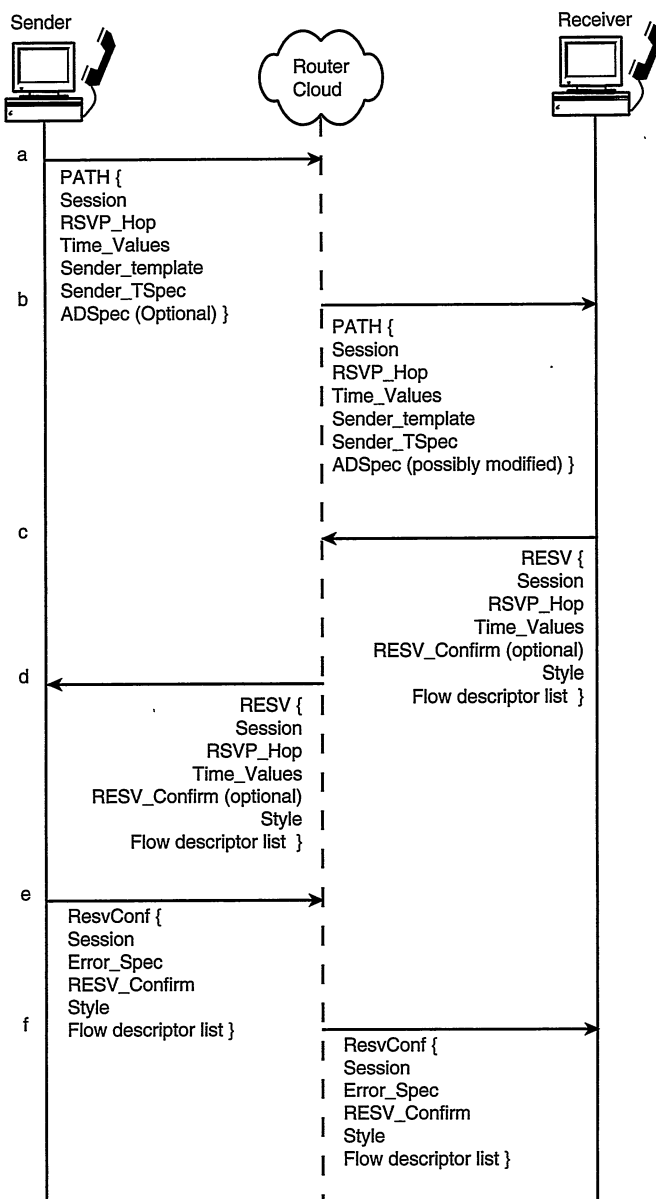
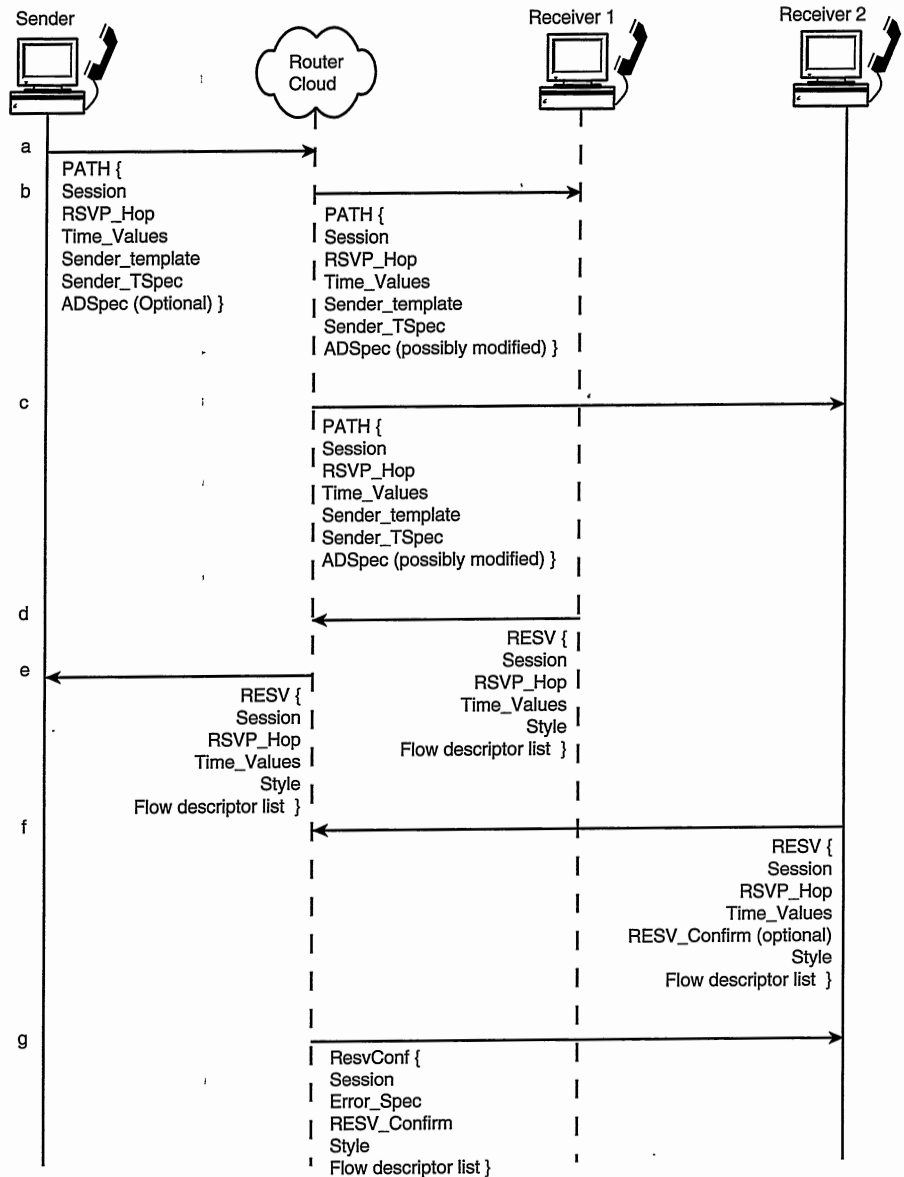


Figure 8-13
Successful reservation
for a single sender
and two receivers



network from the sender to receiver. The network nodes do not modify the sender template and TSpec, but they can modify the ADSpec.

The receiver responds with an RESV message, which can optionally contain a request that the reservation be confirmed. The RESV message indicates the style of reservation to be applied and a list of flow descriptors. Each such descriptor is a flowspec and/or filter spec, depending on the reservation style in question.

In Figure 8-12, the receiver has requested confirmation of the success of the reservation. Consequently, a ResvConf message is returned. The Error_Spec contains an error code of 0 in the case of a successful reservation. The Resv_Confirm object is copied from the RESV message. The style and flow descriptor indicate the particular reservations that are being confirmed. The information in these elements might not be identical to the corresponding information on the RESV message, because in the case of multiple senders, the reservation might not be successful for data streams from all the senders.

In the multicast example of Figure 8-13, Receiver 1 sends an RESV message just before Receiver 2. In this example, the QoS requirement of Receiver 1 is stronger than that of Receiver 2. In that case, there is no need to modify the reservation already sent back towards the receiver, because the reservation from Receiver 1 is good enough to support the requirements of both receivers. In the example, Receiver 2 requested a confirmation, but Receiver 1 did not. Note that the confirmation is sent from the router at which the reservations merge.

This procedure is used when a router receives an RESV message with a request for confirmation and a reservation is already in place that meets or exceeds the requirements of the reservation to be confirmed. In certain cases, this situation can lead to a false confirmation, particularly if the existing reservation request fails at some point closer to the sender. Therefore, a reservation confirmation should only be considered an indication of high probability that a reservation has succeeded.

Reservation Errors

Of course, a given resource reservation could fail, in which case an error message is returned. Two error message exist: PathErr and ResvErr. PathErr messages are simply passed back to the sender whose PATH message led to the error.

ResvErr messages are sent to a receiver when a given reservation request fails. With merged flowspecs, it is important that a reservation

request that fails for a given reason does not deny service for a different request that would have succeeded on its own. For example, imagine two receivers who both send reservation requests and those requests meet at a single router that resides in the path from the sender to receivers. If one receiver's request fails at the router due to an excessive bandwidth request, the other receiver's request should not fail if it has requested a smaller bandwidth that could be supported.

Guaranteed Service

RFC 2212 defines guaranteed service. This service involves two parts: first, ensuring that there will not be any packet loss, and second, ensuring minimal delay. Ensuring that there will not be any loss is a function of the token bucket depth (b) and token rate (r) specified in the TSPEC. At a given router, provided that buffer space of value b is allocated to a given flow and that a bandwidth of r or greater is assigned, then no loss should be experienced.

Delay is a function of two components. The first is a fixed delay due to processing within the individual nodes and is only a function of the path taken. The second component of delay is the queuing delay within the various nodes. Let us assume that a given node supports a token bucket model (b, r) and offers a given stream a bandwidth of R for a particular flow. If R is less than r , then the token bucket algorithm will not introduce any queuing delay, because the rate of tokens supplied to the token bucket will exceed the rate of the incoming data flow. If R is greater than or equal to r , then the token bucket can introduce a queuing delay. The delay in such a case is bounded by b/R .

In reality, other queuing delay components will also play a part: C and D . C is a rate-dependent error term and represents the delay that a flow might experience as a function of the rate of the flow. For example, C/R could represent the delay involved in breaking up a datagram into the payload of *Asynchronous Transfer Mode* (ATM) cells. D is a fixed delay term within the node itself, not dependent on the rate. C is measured in bytes and D is measured in microseconds. A node that provides guaranteed service must ensure that the total delay is less than $b/R \leq C/R \leq D$.

Recall that the ADSPEC can contain a guaranteed service object. This object has the form shown in Figure 8-14. One can see that the object includes cumulative values for $C(\text{tot})$ and $D(\text{tot})$. In addition, values of C and D since the last traffic-resaping point in the path are also included. Traffic reshaping involves buffering at a node in the network to correct deviations from the original TSPEC in the received data. This buffering is set

enced by any successfully delivered packet. RFC 2211 describes controlled-load service.

The service specification does not identify specific characteristics of network elements that need to be minimized. Rather, a node that implements controlled-load service should ensure that it offers a given flow the necessary bandwidth and buffer space to support the TSpec indicated in a given request. This is the reason why the flowspec for controlled-load service is just a TSpec with no additional information. Equally, the controlled-load data fragment of an ADSpec offers no additional information beyond that contained in the general part of the ADSpec.

One important point about the controlled-load service is the QoS that would be experienced by packets that fall outside of the TSpec. There is no guarantee that such packets would get a QoS anything near the QoS of packets that fall within the TSpec. In fact, such packets are likely to experience significant delay or loss.

Removing Reservations and the Use of Soft State

Reservations can be removed in two ways: explicitly by a sender or receiver, or as a result of a timeout.

Two messages exist for the explicit teardown of the reservation state within network nodes: PathTear and ResvTear. PathTear is used by a sender and travels towards all receivers, deleting path state and reservation state in all nodes along the path. ResvTear is used by a receiver, and it effectively does the same thing in reverse. A given receiver does not have the authority to delete the reservation state for other receivers, however. Therefore, in the case of merged reservations, a ResvTear message will propagate upstream in the network only to the extent that deletion of a reservation state applies only to the receiver in question.

RSVP uses a soft state approach to managing reservations. In other words, reservations need to be refreshed on a regular basis or they will time out. Both PATH and RESV messages contain a refresh period (R) within the TIME_VALUES object. To keep a reservation active, the receiver should send a new RESV message after R seconds. Within each node is a timer, L, that is some function of R. If the node does not receive a refreshing message within L seconds, then the reservation state is deleted. L should be set to a value that is significantly larger than R so that one or more RESV messages could be lost without deleting the reservation state.

This setting is important because RSVP does not include reliability functions for ensuring that RSVP messages have guaranteed delivery and it is possible that refreshing messages could be lost. In fact, the timeout of reservations exists for the same reason. A PathTear or ResvTear message at the end of a session might be lost. If the reservation does not time out, then we could have a situation where network resources are being reserved when they are no longer needed, while new reservations fail due to a lack of available resources.



DiffServ

RSVP is a comprehensive QoS mechanism. Of the QoS solutions available, it comes the closest to circuit emulation, but it does so at a significant cost. RSVP requires that RSVP-enabled routers examine IP headers in some detail, and it requires that routers maintain state for all existing reservations. If RSVP is used on a session-by-session basis (such as on a per-VoIP-call basis), then there might be millions of concurrent reservations. Such a situation can mean significant overhead. Consequently, RSVP has difficulty scaling to very large systems.² In many cases, we can provide a good QoS to numerous applications without the complexity and overhead of RSVP. DiffServ is one means for achieving this goal.

If we think of QoS purely in terms of bandwidth, the only way to increase the QoS offered to a given data stream is to increase the bandwidth. Increasing bandwidth can be achieved by physically adding bandwidth or by offering one application extra bandwidth at the expense of another application. RSVP works by reserving a set bandwidth for a particular data stream. Once all the bandwidth is consumed, further reservation attempts fail. The approach with DiffServ is that it is acceptable to offer one application greater QoS at the expense of another application if the first application really needs it for acceptable performance and the second application will not really notice a big difference. For example, we can offer a greater QoS to interactive traffic (such as voice) at the expense of non-real-time traffic (such as e-mail), because it matters greatly if voice is subject to delay.

²This statement is true when RSVP is used to reserve resources for each individual session. As we shall see later in this chapter, however, traffic-engineering extensions for RSVP (RSVP-TE) mean that RSVP can be used for traffic-engineering functions. Since RSVP-TE is not used on a per-session basis, it does not have the same scaling limitations as basic RSVP.

It does not greatly matter if an e-mail takes minutes instead of seconds to be delivered. DiffServ simply offers the capability to prioritize certain types of traffic and it offers certain types of handling in the network depending on the priority of a given traffic stream.

The DiffServ Architecture

Recall that the IPv4 header has a ToS field, and the IPv6 header has a corresponding Traffic Class field. DiffServ renames this field the DS field. As shown in Figure 8-15, DiffServ uses six bits of the field to encode the *DS codepoint* (DSCP) and does not currently use the remaining bits of the field. In fact, those two bits are used for explicit congestion notification, as specified in RFC 3168. Within the network, the values of the DSCP in a given IP packet are used to handle the packet according to a certain PHB. Therefore, all that is required within the core of the network is the capability to examine the DSCP and act accordingly.

The DiffServ architecture is defined in RFC 2475, "An Architecture for Differentiated Services," with clarifications added by RFC 3260. In the DiffServ architecture, the packets of a given stream must be marked with the appropriate DSCP value so that routers in the network can provide the correct PHB. To avoid situations where every packet from every stream is marked to get the highest priority, functions at the edge of the network must ensure that only qualified packets are marked with a particular DSCP value. These functions include metering to measure the packet rate, policing to ensure that traffic meets a particular agreed profile, traffic shaping and dropping if necessary, and the actual marking itself. Collectively, these functions are known as traffic conditioning. Of course, it is also necessary to classify packets to identify the stream to which they belong or other characteristics that lead to the selection of a particular DSCP value. These functions work in a combined manner, as depicted in Figure 8-16.

One advantage of the DiffServ architecture is that the classification, policing, and traffic-shaping functions are pushed to the edge of the network, while the requirements on routers within the network are kept to a minimum. This means that the bulk of the effort and investment involved

Figure 8-15

DiffServ usage of IP
ToS/Traffic Class octet

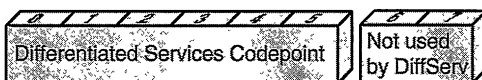
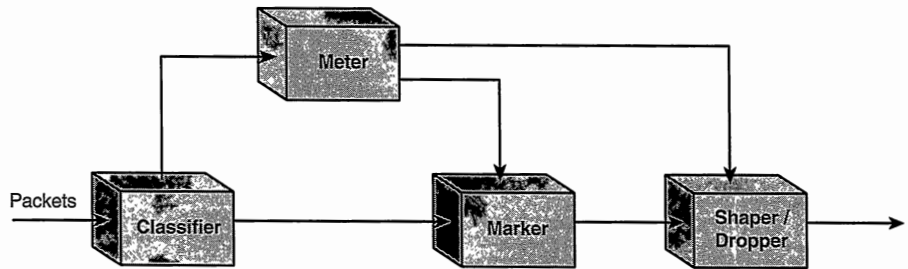


Figure 8-16

Classifying and conditioning traffic



in using DiffServ is placed at or near the host that runs an application that wants to use DiffServ. The amount of change required in the core of the network is minimal and does not change with the number of user applications that want to use DiffServ. Consequently, DiffServ scales extremely well.

A disadvantage of DiffServ is that it works on a per-hop basis. In other words, each router in the network provides priority to a certain packet according to that packet's DSCP value, but DiffServ does not offer a mechanism that reserves resources from end to end. Thus, although DiffServ can enable better QoS for certain applications compared to others, it does not provide a QoS guarantee for any application, particularly in the case of network congestion. Therefore, we must strive to ensure that the network does not become congested in the first place.

The Need for SLAs

The most critical component of successful DiffServ operation is at the edge of the network. It is vital that application packets be correctly classified and marked, and applications must be prevented from abusing the system. Therefore, a given network domain and packet source must agree on the rules related to packet classification and traffic conditioning, and these functions must be implemented at or near the source of the packets. The functions can be implemented at the source itself (such as within a customer's host[s]), or they could be implemented in an edge router provided by the network operator (such as an ISP or an *Internet telephony service provider* [ITSP]).

In either case, an SLA must be in place between the customer and network operator. Such an agreement should include a definition of the traffic profile to be sent from the source to the network, the classification and marking rules, and the behaviors to be applied within the network for specific DSCP values. The traffic profile might include a token bucket specifi-

cation, as described previously. The classification rules could be based on combinations of source address, destination address, source port, destination port, protocol ID, time of day, or other criteria.

The customer and provider should agree on the classification criteria and, just as important, the SLA should specify what is to happen if traffic is received that is outside the traffic profile. In such a situation, actions could include the dropping of packets, marking nonconformant packets with a different DSCP value than that used for conforming packets, traffic shaping, or additional charges to the customer.

SLAs should not only be used between network operators and their customers, but they should also be placed between different network operators. In the context of this discussion, a network operator can be considered to be a particular DiffServ-capable domain with a common set of policies and PHB definitions. If traffic needs to be passed from one operator to another, the two operators should agree on the rules related to the service that particular streams receive and in particular the meaning to be applied to particular DSCP values. Only in this way can we achieve end-to-end QoS.

Per-Hop Behavior (PHB)

PHB is the treatment that a DiffServ router applies to a packet with a given DSCP value. Note that a router deals with multiple flows from many sources to many destinations. Many of the flows may have packets marked with a DSCP value that indicates a certain PHB. That set of flows from one node to the next that share the same DSCP codepoint is known as an *aggregate*. From a DiffServ perspective, a router operates on packets that belong to specific aggregates. When a router is configured to support a given PHB (such as dedicating bandwidth), the configuration is with respect to aggregates rather than specific flows from a specific source to a specific destination.

Currently, two PHBs are defined: EF and AF. These are described in RFC 3246 and RFC 2597, respectively.

Expedited Forwarding (EF) The objective with EF PHB is to provide a service that is low loss, low delay, and low jitter, such that the service approximates a virtual leased line. The basic approach is to minimize the loss and delay experienced in the network by minimizing queuing delays. This job can be done by ensuring that, at each node, the rate of departure of packets from the node is a well-defined minimum and that the arrival rate at the node is always less than the defined departure rate. To meet this

objective, traffic-conditioning functions at the edge of the network are necessary to ensure that the incoming rate is always below the configured outgoing rate. The recommended DSCP value for the EF PHB is 101110.

The EF PHB can be implemented in network nodes in a number of different ways. For example, the EF PHB could be implemented by a priority queue. Such a mechanism could enable the unlimited preemption of other traffic such that EF traffic always gets first access to outgoing bandwidth. This approach could, however, lead to unacceptably low performance for non-EF traffic. Therefore, it is important that the implementation of EF in a given node is designed such that EF traffic does not inflict enormous damage to other traffic. One way to do this would be to police the rate of incoming EF traffic through a token bucket limiter. Another way to implement the EF PHB would be through the use of a weighted round-robin output scheduler, where the share of the output bandwidth allocated to EF traffic is equal to some configured rate. That output rate should be greater than or equal to the rate expected for incoming EF traffic.

Assured Forwarding (AF) AF PHB is defined in RFC 2597. Its objective is to provide a service that ensures that high-priority packets are forwarded with a greater degree of reliability than lower-priority packets.

Normally, the traffic into a DiffServ network from a particular source should conform to a particular traffic profile. In particular, the rate of traffic should not exceed some pre-agreed maximum. In the event that it does, the source of the traffic runs the risk that the data will be lumped in with normal best-effort IP traffic and will be subject to the same delay and loss possibilities. In fact, in a DiffServ network, certain resources will be allocated to certain behavior aggregates, which means that a smaller share is allocated to standard best-effort traffic. Thus, receiving best-effort service in a DiffServ network could be worse than receiving best-effort service in a non-DiffServ network. A given subscriber to a DiffServ network might want the latitude to occasionally exceed the requirements of a given traffic profile without being too harshly punished. AF PHB offers this possibility.

AF PHB is a means for a provider to offer different levels of forwarding assurances for packets received from a customer. AF PHB enables packets to be marked with different AF classes and, within each class, to be marked with different drop precedence values. Within a router, resources are allocated according to the different AF classes. If the resources allocated to a given class become congested, then packets must be dropped. The packets to be dropped are those with higher drop precedence values. From a customer's point of view, this offers the possibility of saying something like, "Please don't drop my packets, but if you have

to, then drop these particular ones before any others, as these are less important.”

AF has four classes and three drop precedence levels in each class. Therefore, a given packet has a class (i) and a drop precedence (j), and the associated DSCP is AFij. The recommended values of the codepoints are shown in Table 8-1. We can see that the three most significant bits indicate the class and the three least significant bits indicate the drop precedence within the class.

Within a DiffServ network, the AF implementation must detect and respond to long-term congestion by dropping packets and respond to short-term congestion by queuing. These requirements imply a function that monitors short-term congestion and, based on short-term congestion levels, derives a smoothed long-term congestion level. When the smoothed congestion level is below a particular threshold, no packets should be dropped. If the smoothed congestion level is between a first and second threshold level, then packets with the high precedence level should be dropped. As the congestion level rises, more and more of the high-drop-precedence packets should be dropped until such time as a second congestion threshold is reached, at which point 100 percent of the high-drop-precedence packets are dropped. If the congestion continues to rise, then packets of the medium precedence level should also start being dropped.

The implementation must treat all packets within a given class and precedence level equally. Thus, if 50 percent of the packets with a given class and precedence value are to be dropped, then that 50 percent should be spread evenly across all packets for that class and precedence. Put another way, a given flow should experience the same drop rate as every other flow with the same class and precedence level. Note that different AF classes are treated independently and are given independent resources. Thus, when packets are to be dropped, they are dropped for a given class and drop precedence level. Packets of one class and precedence level might experience a 50 percent drop rate, while the packets of a different class with the same precedence level are not dropped at all. Regardless of the amount

Table 8-1

Recommended AF
codepoint values

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

of packets that need to be dropped, a DiffServ node must not reorder AF packets within a given AF class regardless of their precedence level.



Multiprotocol Label Switching (MPLS)

We can see that RSVP and DiffServ take very different approaches to solving the QoS problem. RSVP, in its basic form, tackles the issue by reserving resources in advance of a given session. DiffServ, on the other hand, takes the approach of sharing network resources according to priority. Since RSVP was initially designed to operate on a session-by-session basis (and millions of concurrent sessions might take place), it has difficulty scaling to large systems. DiffServ can scale well, but it operates on a hop-by-hop basis and does not ensure that end-to-end resources are available for a given session. An ideal solution is one that offers the best of both worlds: the capability to ensure end-to-end resource availability for a large number of sessions. MPLS is one such solution.

MPLS is not primarily a QoS solution, although it may be used to support QoS requirements. Rather, MPLS is best viewed as a new switching architecture. Standard IP switching requires that every router must analyze the IP header and make a determination of the next hop, based on the content of that header. The primary driver in determining the next hop is the destination address in the IP header. The destination address is compared with entries in a routing table, and the longest match between the IP header and the addresses in the routing table determines the next hop. The approach with MPLS is to attach a label to the packet. The content of the label is specified according to an FEC, determined at the point of ingress to the network. The packet and label are passed to the next node, where the label is examined and the FEC is determined. The FEC is then used as a simple lookup in a table that specifies the next hop and a new label to use. The new label is attached and the packet is forwarded.

So what is the big difference between this and standard routing based on the IP header? To begin with, the FEC is determined at the point of ingress to the network, where information might be available that cannot be indicated in the IP header. For example, the FEC could be chosen according to a combination of the destination address, QoS requirements, the ingress router (or interface), or any of a multitude of criteria. The FEC can impli-

citly indicate such information, and routing decisions in the network can automatically take that information into account. In particular, a given FEC can force a packet to take a particular route through the network without having to cram a list of specific routers into the IP header. This fact can be important for ensuring QoS, where the bandwidth available on a given path and the path length both have a direct impact to the quality that is achieved. Imagine a network that supports different types of traffic. Between any two points in the network, we might want one type of traffic to take a particular path while another type of traffic takes a different path. For example, VoIP traffic from point A to point B should take a very direct path (to minimize latency), while other traffic is forced along a less direct route.

In some ways, MPLS is similar to ATM. MPLS enables a specific type of traffic to take a specific path from ingress to egress, and that path can be established in advance. The path in question can be engineered with appropriate resources, and the point of ingress can include policing functions to ensure that the traffic demand does not exceed the allocated resources. This approach is similar to the connection admission functions invoked at the establishment of an ATM *Virtual Channel Connection* (VCC).

In fact, MPLS and ATM are competing technologies. Although most people in telecommunications recognize that MPLS is likely to be the long-term choice for supporting multiple services on a single network, many question its maturity. ATM, on the other hand, is stable, mature, and widely deployed. Despite these advantages, however, ATM has a number of disadvantages, including its complexity and the famous (or infamous) ATM cell tax. As MPLS sees more deployment, we can expect MPLS to supplant ATM in the long term.

Meanwhile, however, a middle ground is apparent. The fact that MPLS places a label in front of an IP packet does not necessarily imply a new layer between layers 2 and 3. Obviously, the fact that a new label is attached to the packet can mean that a new layer exists, if only for the purpose of interpreting the contents of the label. The label itself can be carried in an existing layer 2 protocol, however, provided an appropriate field exists in which to carry it. For example, the label could be carried in a Frame Relay *Data Link Connection Identifier* (DLCI) field. Similarly, the label could be carried in ATM *Virtual Path Identifier* (VPI) or *Virtual Channel Identifier* (VCI) fields. If one has an existing ATM infrastructure and wants to migrate to MPLS, the mapping of MPLS labels to ATM VPI/VCI fields is one way to begin the migration without risking a quantum leap from one technology to another.

The MPLS Architecture

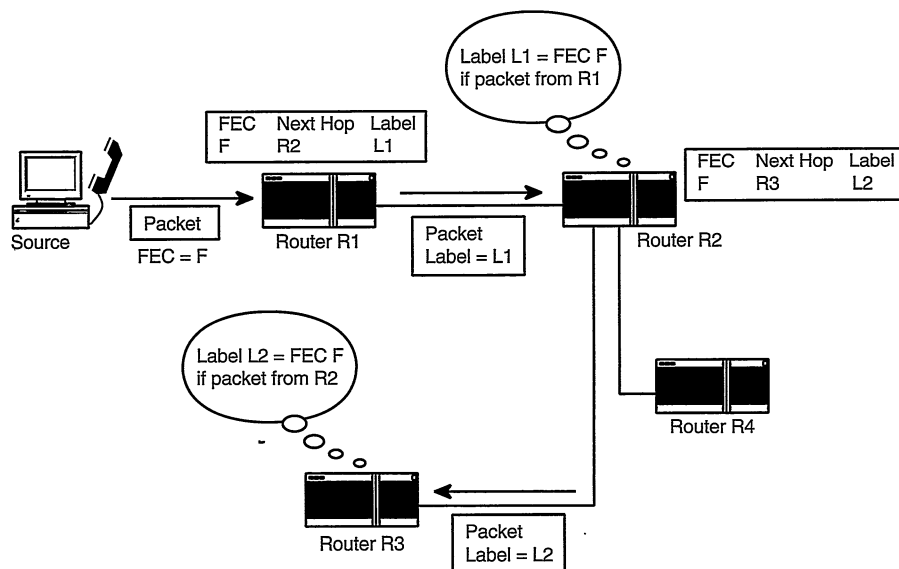
The MPLS architecture is specified in RFC 3031, “Multiprotocol Label Switching Architecture.” As already described, MPLS involves the determination at the point of ingress to the network of an FEC value to apply to a packet. That FEC value is then mapped to a particular label value and the packet is forwarded with the label. At the next router, the label is evaluated and the corresponding FEC is determined. A lookup is then performed to determine the next hop and the new label to apply. The new label is attached and the packet is forwarded to the next node.

This process suggests that the value of the label can change as the packet moves through the network, which it can. What does not change is the FEC.

Basically, the process works as shown in Figure 8-17. An incoming packet is assigned a particular FEC based on some criteria. In the figure, the packet has an FEC of F. Note that the FEC is not a value carried in the packet. Rather, the FEC can be considered as a combination of factors: the port number, originating address, destination address, ToS bits, and so on.

At the ingress router (R1), the FEC of F means that the packet must be sent to router R2 and that it should have a label value of L1. When the packet and label arrive at R2, R2 knows that the label value L1 means an FEC of F for packets received from R1. R2 then proceeds to look up the next

Figure 8-17
Label value and FEC
relationships



hop and label value. It determines that the packet should be forwarded to R3 with a label value of L2. At R3, the packet and label are received. R3 knows in advance that the label value L2 means an FEC of F for packets from R2 and it uses this information for its routing decision, and so on.

An important point to note is the fact that the ingress router (R1 in our example) has the job of assigning the first label. That initial assignment determines subsequent handling within the network. In other words, the first router determines the FEC. The assigned FEC can be based on information that only the first router knows, such as the ingress interface. The ingress router is known as a *label edge router* (LER).

We can see that the relationship between FEC and the label value is a local affair between two adjacent *label-switching routers* (LSRs). If a given router is upstream from the point of view of the data flow, then it must have an understanding with the next router downstream as to the binding between a particular label value and FEC.

For example, in Figure 8-17, the label value L1 is the outgoing label at R1 for packets with FEC F going to R2. At R2, the incoming label value of L1 is bound to FEC F for packets incoming from R1. Also at R2, the label value L2 is the outgoing label for packets with FEC F going to R3, and so on. Thus, the binding between the label value and FEC is limited to the two ends of an interface between adjacent LSRs for a particular direction of data flow. Also, if R2 were to receive packets with the label value L1 on a link from a different LSR, such as R4, then a different FEC might apply. The fact that the meaning of a given label depends on the incoming interface means a greater routing flexibility than with standard routing tables (where the outgoing path from a router depends on the destination IP address but not on the incoming interface).

Label-Switched Paths (LSPs) and Label Distribution The foregoing discussion shows that the path through a network is determined solely by the FEC that applies at the point of ingress. Such a path is known as a *label-switched path* (LSP), and it may be established in a number of ways. First, we can easily imagine a scenario where a network engineer determines that a certain FEC shall apply to a certain type of traffic from a certain source to a certain destination. That engineer could then determine the most appropriate path through the network and configure each of the applicable router interfaces with the necessary FEC/label bindings. Alternatively, we can use a protocol that enables the routers to automatically share FEC/label bindings. Such a protocol is known as a *label distribution protocol*.

The primary function of label distribution protocols is the establishment and maintenance of LSPs, which includes the establishment of FEC/label

bindings at each LSR in the LSP. In the MPLS architecture, the downstream LSR decides on the particular binding. The downstream LSR uses a label distribution protocol to communicate the binding to the upstream LSR. A downstream LSR can directly distribute a label/FEC binding to an upstream LSR, which is known as *unsolicited downstream*. Additionally, we have *downstream on demand*, whereby an upstream LSR requests a binding from a downstream LSR.

Several label distribution protocols are available. In some cases, existing protocols such as RSVP have been extended to provide support for label distribution. In addition, brand new protocols have been developed, such as the *Label Distribution Protocol* (LDP) and *Constraint-Based LDP* (CR-LDP), which is an extension of LDP. If a given label distribution protocol is used between two LSRs to exchange label/FEC binding information, then the two LSRs are known as *label distribution peers*. Although the MPLS architecture does not mandate that any particular label distribution protocol be used, we discuss three such protocols later in this chapter: LDP, CR-LDP, and RSVP-TE.

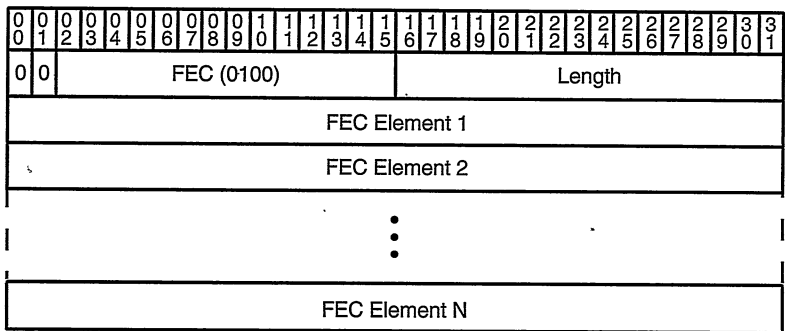
FEC and Label Formats

In order for a downstream LSR to communicate a label/FEC binding to an upstream LSR, the label distribution protocol messages must include fields that specify FEC and label values. For example, a downstream LSR can use the LDP Label Mapping message to communicate a particular label/FEC binding to an upstream LSR. The message includes specific FEC values and label values. Clearly, FECs and labels must have specific formats. Those formats depend on the label distribution protocol being used.

In LDP, an FEC is comprised of a list of FEC elements, as shown in Figure 8-18, with each FEC element either a host address or address prefix. The format of a label to be used between any two adjacent LSRs, however, depends on the layer 2 protocol used on the link between them. For protocols (such as the *Point-to-Point Protocol* [PPP]) that do not already support a field that can carry a label value, the label takes the generic format of a four-octet field placed in front of the IP packet. The label contains four fields: a 20-bit label field, which carries the actual label value; a 3-bit *experimental* (Exp) field, which is for experimental use; an 8-bit TTL field; and a 1-bit stack (S) field, which indicates whether this label is the bottom of the label stack. We will discuss the concept of a label stack shortly.

When such a label format is communicated using a label distribution protocol (during LSP establishment), the generic format of Figure 8-19 is

Figure 8-18
The FEC format
in LDP



used for layer 2 protocols such as PPP. For example, in the Label Mapping message of LDP, a parameter of the form shown in Figure 8-19 would be passed to indicate that a generic-format, four-octet label of a particular value is to be associated with a particular FEC. The first four-octet word indicates the format of the label and the second word includes the label itself. Once the LSP is established, packets from the upstream LSR to the downstream LSR that correspond to the FEC in question would be prepended by the corresponding four-octet label (that is, the second word of Figure 8-19).

Both ATM and Frame Relay contain fields that could be used to carry an MPLS label. For example, the VPI and VCI fields of the ATM header could carry an MPLS label. Equally, the Frame Relay DLCI field could carry an MPLS label. In the case of ATM, the label is also four octets long, but it takes the form shown in Figure 8-20. In a Label Mapping message, the format and value of the label would be communicated with the label format of Figure 8-20. The first word of Figure 8-20 indicates the format of the label, and the second word conveys the label value itself. The V-bits indicate whether both the VPI and VCI are significant or just one of them. 00 indicates that both are significant, 01 indicates that only the VPI is significant, and 10 indicates that only the VCI is significant.

When Frame Relay is used at layer 2, the label is mapped to the DLCI, as shown in Figure 8-21. The Len field indicates the number of bits in the DLCI. 0 indicates that the DLCI is 10 bits long, and 2 indicates that the DLCI is 23 bits long. Values 1 and 3 are reserved.

The Label Stack Much of the discussion so far has suggested that a given packet can have only one label. Although such a situation is valid, a given packet can have more than one label. In fact, MPLS describes a label stack, where there may be several labels, one after the other, in advance of

Figure 8-19

The generic label format

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
0	0	Generic Label (0200)														Length											
Label																		Exp	S	TTL							

Figure 8-20

The ATM label format

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
0	0	ATM Label (0201)														Length											
Res	V	VPI														VCI											

Figure 8-21

The Frame Relay label format

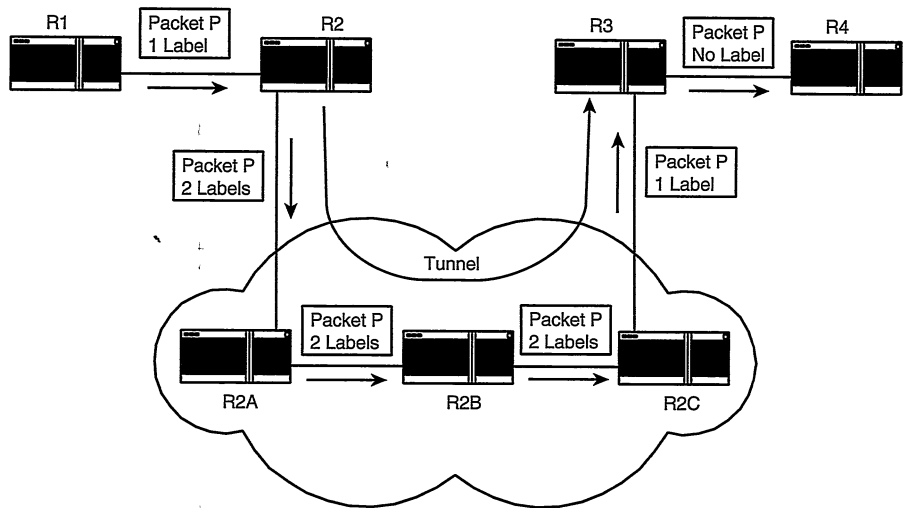
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
0	0	ATM Label (0202)														Length											
Reserved				Len				DLCI																			

the actual IP packet. In such a situation, an LSR that receives the labeled packet will base its actions on the first (top) label. The obvious question is, "Why might we need more than one label?" The answer is tunneling.

Imagine a VoIP network operator that provides international service between the United States and the United Kingdom. That provider will have gateways and routers on both sides of the Atlantic, but might not own the network in between. Rather, the network operator could purchase service from another party, such as a large IP network operator. That IP network operator might offer service to many different customers, and the traffic from each of those customers would be carried over the same network, but would need to be kept logically separate within that common network.

Suppose that a given packet with FEC F needs to follow the LSP R1, R2, R3, and R4. Let's assume that R2 and R3 are not directly connected, but form the two ends of the tunnel R2, R2A, R2B, R2C, and R3. Therefore, the actual path taken is R1, R2, R2A, R2B, R2C, R3, and R4, as depicted in Figure 8-22. When a packet travels from R1 to R2, it has a single label. R2 determines that the packet P must enter the tunnel. Therefore, R2 attaches another label on top of the first, so that the label stack has a depth of 2. In fact, it may replace the first label with a label that is meaningful to R3 and then place a new label on top. The packet P has two labels as it passes through the tunnel. When it reaches R2C, the LSR recognizes that it is the penultimate LSR in the tunnel. (Both R2 and R3 are effectively part of the

Figure 8-22
An LSP tunnel



tunnel.) R2C pops the stack and forwards the packet to R3 with only one label. R3 uses the label to route to R4. In fact, R3, being the penultimate LSR in the overall LSP, may also pop the stack and pass the packet to R4 unlabeled.

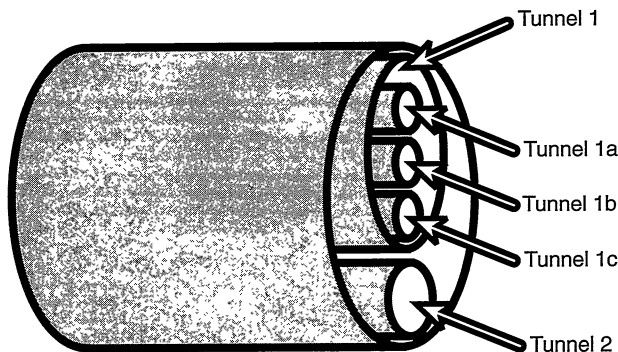
The scenario depicted in Figure 8-22 is effectively an LSP nested within another LSP. The MPLS architecture enables multiple levels of nesting, which can be useful in many situations. Consider again our transatlantic VoIP provider. That provider might want to send more than just VoIP traffic between the two countries. In addition to the actual revenue-generating VoIP traffic, we can easily imagine a situation where the *Information Technologies* (IT) department must send billing information, and the Network Operations department needs to send network alarm and provisioning data. Each department could have its own IP networks on either side of the Atlantic. The company as a whole could purchase service from an international IP network operator, with its total traffic carried in a tunnel through the international IP operator's network. Within that tunnel, each department's traffic would be carried within its own tunnel, as depicted in Figure 8-23.

Actions at LSRs

An LSR's actions depend on the value of the label it receives from an upstream LSR. In fact, the action taken by the LSR is specified by the *Next*

**Figure 8-23**

MPLS tunnels within tunnels



Hop Level Forwarding Entry (NHLFE), which indicates the next hop, the operation to perform on the label stack, and the encoding to be used for the stack on the outgoing link. The operation to perform on the stack may mean that the LSR should replace the label at the top of the stack with a new label. The operation might require the LSR to pop the label stack or replace the top label with a new label, and then add one or more additional labels on top of the first label.

The next hop for a given labeled packet might be the same LSR. In such a case, the LSR pops the top-level label of the stack and forwards the packet to itself. At this point, the packet might still have a label to be examined, or it might be a native IP packet without a label, in which case the packet is forwarded according to standard IP routing.

A given label could possibly map to more than one NHLFE. This situation might occur where load-sharing takes place across multiple paths. In such a case, the LSR chooses one of the NHLFEs to use according to some internal procedures.

In general, if a router happens to know that it is the penultimate LSR in a given path, then it should remove any labels and pass the packet to the final LSR without a label. This minimizes the amount of processing that the ultimate LSR needs to undertake. If the penultimate LSR passes a labeled packet to the final LSR, then the final LSR must examine the label, determine that the next hop is itself, pop the stack, and forward the packet to itself. The LSR must then reexamine the packet to determine what to do with it. If, on the other hand, the packet arrives without a label, the final LSR has one less step to execute. How a particular LSR determines that it is the penultimate LSR for a given path is a function of label distribution and the distribution protocol used.

MPLS Traffic Engineering

One of the most important applications of MPLS is in the area of traffic engineering, which can be summarized as the modeling, characterization, and control of traffic to meet specified performance objectives. As described in RFC 2702, the performance objectives in question may be traffic oriented or resource oriented. The former deals with QoS and includes aspects such as minimizing delay, jitter, and packet loss. The latter deals with optimum usage of network resources, particularly network bandwidth.

These two objectives are not necessarily mutually exclusive. For example, congestion avoidance is a major goal related to both network resource objectives and QoS objectives. From a resource point of view, we want to avoid situations where one part of the network is congested while another part of the network is underutilized, particularly if the underutilized resources could be used to carry some of the traffic that is experiencing congestion. Equally, from a QoS point of view, we want to allocate traffic streams to available resources to ensure that those streams do not experience congestion and the consequent packet loss and delay.

Congestion is primarily caused in two ways. The first is simply due to a lack of sufficient resources in the network to accommodate the offered load. The second is the steering of traffic towards resources that are already loaded, while other resources remain underutilized. The expansion of capacity or the application of congestion control techniques such as flow control can correct the first situation. The second situation can be addressed by good traffic engineering, that is, ensuring that traffic is directed through the network in a manner that is consistent with the needs of that traffic. For example, we might want to direct VoIP traffic along the shortest path in order to minimize latency. Traffic that is less delay sensitive could be sent along a less-direct path.

The current situation with IP routing and resource allocation is that the routing protocols are not well equipped to deal with traffic-engineering issues. For example, a protocol such as *Open Shortest Path First* (OSPF) can actually promote congestion because it tends to force traffic down the shortest route, even though other acceptable routes might be less loaded. In many network deployments, the solution to this problem has been to use traffic-engineering functions at layer 2 in the network. ATM, for example, enables virtual circuits to be easily rerouted in cases of congestion. MPLS also enables traffic-engineering functions.

Traffic Trunks RFC 2702 introduces the concept of a traffic trunk, which can be considered to be a set of flows that share specific attributes.

In practice, these attributes include the ingress and egress LSRs, the FEC, and possibly other characteristics such as the average rate, peak rate, priority, and policing attributes. For example, all VoIP packets from a given source to a given destination might be considered a single traffic trunk. Meanwhile, other traffic (e-mail, for example) between the same source and destination might be considered a different traffic trunk.

A traffic trunk is something that can be (and is) routed over a given LSP. In fact, it is possible to explicitly specify the LSP that a traffic trunk should use. This has the immediate advantage of being able to steer certain traffic away from the shortest path, which is likely to get congested before other paths. If we understand the available network resources and the expected demand for each type of traffic, then we can route each type of traffic through the network in a manner that best meets QoS requirements. In other words, we can ensure that those applications with strong QoS needs are routed along paths that have the resources to meet those QoS needs. Moreover, we can change the LSP that a given traffic trunk will use. This capability enables the network to adapt to changing load conditions whether through administrative intervention or through automated processes within the network. Thus, we can choose to send a certain type of traffic over a certain path, and we can change to a second path if the first path fails.

As described in RFC 2702, traffic engineering in an MPLS network has three main aspects:

- Mapping packets to FECs
- Mapping FECs to traffic trunks
- Mapping traffic trunks onto the physical network topology through LSPs

How individual packets are assigned to a given FEC and how FECs are assigned to traffic trunks are decisions that are made at the ingress to the network. These decisions can be made according to any number of criteria (such as the physical ingress interface, the IP ToS value, or the source IP address). We simply need to ensure that the criteria are fully understood by both the MPLS network provider and the source of packets (either a customer or another network provider). This requirement again emphasizes the need for SLAs between providers and between providers and customers.

Of the three mappings that need to take place, the third is at the heart of ensuring that the network provides the quality that is needed for a given type of traffic. It involves constraint-based routing, whereby traffic is matched with network resources according to the characteristics of the traf-

fic and the characteristics of available resources. For example, one characteristic of traffic is a bandwidth requirement and one characteristic of a path is the maximum bandwidth that it offers. Obviously, path selection should be limited to those paths that can offer at least the bandwidth needed by the traffic trunk.

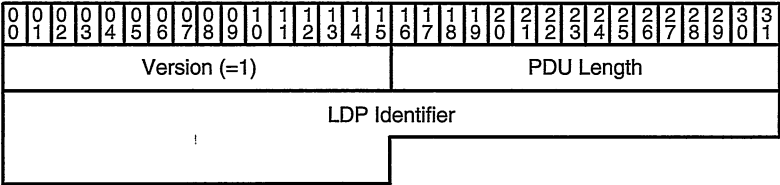
Label Distribution Protocols and
Constraint-Based Routing

We have already described the concept of label distribution protocols, which is the means by which LSRs share label/FEC bindings so that traffic belonging to a particular FEC can follow a particular LSP.

The Label Distribution Protocol (LDP) One of the most common such protocols is LDP. LDP includes a number of messages that are exchanged between LSRs for the purpose of establishing and maintaining signaling relationships between the LSRs, requesting and providing label/FEC bindings, and maintaining the overall health of an LSP. LDP conveys information through *LDP protocol data units* (PDUs), which comprise a common header, followed by an LDP message.

Figure 8-24 shows the format of the LDP common header. The LDP identifier indicates the label space of the LSR that is sending the message in question. The first four octets of the LDP identifier include a globally unique identifier for the LSR. The remaining two octets indicate the label space of the LSR. A given LSR might reuse the same label values across different interfaces, in which case the value of the label space identifier indicates the interface to which a label value applies. If an LSR uses just a single label space for the whole platform, then the label space value of 0 is used.

Figure 8-24
The LDP PDU
common header



LDP Messages LDP offers four categories of messages:

- Discovery messages
- Session Management messages
- Advertisement messages
- Notification messages

Discovery messages are used to announce and maintain the presence of an LSR on the network. An LSR periodically sends Hello messages to other LSRs to indicate its presence on the network. The LSR can use basic discovery or extended discovery. Basic discovery means that LDP Hello messages are sent to the well-known multicast address for “all routers on this subnet,” which means that the Hello messages are sent only to other routers that are directly connected at the link layer (layer 2). Extended discovery enables an LSR to send Hello messages to routers that are not directly connected at layer 2. Such Hello messages are known as targeted Hello messages and are sent to specific IP addresses that the LSR is aware of (perhaps through configuration data loaded in the LSR).

Upon receipt of a Hello message, an LSR can reply with a Hello message of its own. At this point, both LSRs are aware of the other’s presence and know each other’s IP addresses.

Session Management messages are used to establish, maintain, and terminate sessions between LDP peers. Once two LSRs have exchanged Hello messages, they proceed to exchange Initialization messages. An Initialization message includes a number of mandatory and optional parameters. Among the mandatory parameters is the receiver’s LDP identifier. Since the LDP common header contains the sender’s LDP identifier, the Initialization message effectively contains both LDP identifiers: the sender’s and the receiver’s. This combination uniquely identifies a relationship between LSRs that began with the exchange of Hello messages.

Among the optional parameters of the Initialization message are ATM and Frame Relay address ranges. These values indicate the ATM VPI/VCI or Frame Relay DLCI ranges that can be used if the link between the LSRs uses ATM or Frame Relay.

Once two LSRs have exchanged Initialization messages, an LDP session is established between them. Thereafter, the session is maintained through the exchange of LDP PDUs. If an LSR does not receive an LDP PDU after a “keep-alive” time period, then it will conclude that the connection between the LSRs has failed. In the event that no LDP PDUs need to be sent between LSRs, they can periodically send LDP keep-alive messages to ensure that the LDP session does not time out.

Advertisement messages are used to create, change, and delete label mappings for FECs. LDP includes two main messages related to label mapping: the Label Request message and the Label Mapping message. The Label Request message is used by an LSR that operates in downstream-on-demand mode and indicates that the upstream LSR wants to obtain a label/FEC binding. The Label Mapping message contains the actual label/FEC binding. In downstream-on-demand mode, the Label Mapping message is sent as a response to a Label Request message. In unsolicited downstream mode, the message can be sent without first receiving a Label Request message.

LSRs can operate according to either independent label distribution control or ordered label distribution control. With independent label distribution control, an LSR can advertise a label mapping to an LDP peer any time it desires. Thus, an LSR can send a Label Mapping message to an upstream LSR without first having received a Label Mapping message from a downstream LSR. With ordered label distribution control, an LSR can advertise a label mapping to an upstream LSR only if the LSR has an FEC/label binding for the next hop or if the LSR itself is the MPLS egress point.

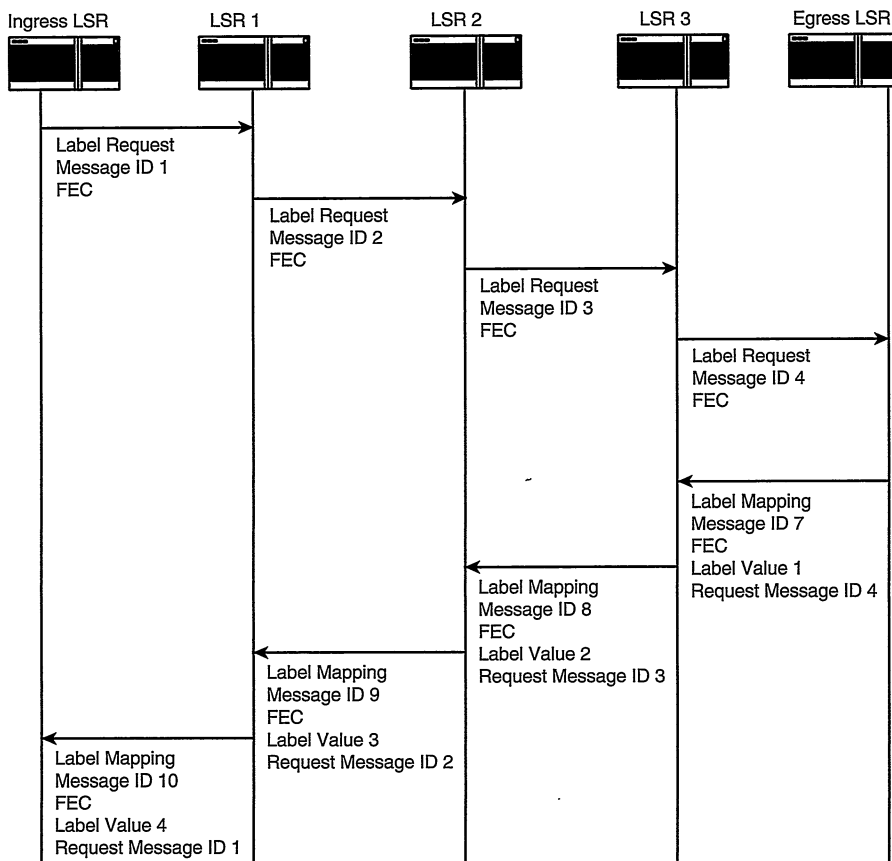
If we combine downstream-on-demand with ordered label distribution control, then label distribution would work as follows. The ingress LSR would send a Label Request message to a downstream LSR and that message would propagate through the network to the egress LSR. The egress LSR would respond to its next upstream LSR with a Label Mapping message. That upstream LSR would store the FEC/label binding for its egress interface, create a new FEC/label binding for its ingress interface, and communicate that binding to the next upstream LSR with a Label Mapping message of its own. The process continues until the ingress LSR receives the Label Mapping message in response to its original request. Figure 8-25 depicts the signaling sequence for such a scenario.

Figure 8-26 shows the format of the LDP Label Request message and Figure 8-27 shows the format of the LDP Label Mapping message. The term TLV refers to the structure Type-Length-Value, whereby a given field includes an identifier to indicate the type of field in question (T), a length subfield (L) and the actual value of the field itself (V).

The Label Request message has two optional parameters: the Path Vector TLV and the Hop-count TLV. Both of these parameters help to detect loops. The Path Vector TLV is simply a list of LSRs. Each LSR that propagates a Label Request message adds its own LSR ID (the first four octets of the LDP identifier described previously) to the list of LSR IDs. Similarly, each LSR that forwards a Label Request message increments the

Figure 8-25

LSP establishment
with downstream-on-
demand and ordered
label distribution
control

**Figure 8-26**

The LDP Label
Request message

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	
0	Mess. Type = Label Request (0401)																Message Length										
Message ID																											
FEC TLV																											
Optional Parameters																											

Hop-count TLV by 1 (except when the Hop-count TLV was previously set to 0, which means unknown).

The Label Mapping message contains the same optional parameters as the Label Request message for the same reason—loop detection. In addition, the Label Mapping message can optionally carry a Label Request

Figure 8-27

The LDP Label Mapping message

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	
0	Mess. Type = Label Mapping (0400)																Message Length											
Message ID																												
FEC TLV																												
Label TLV																												
Optional Parameters																												

message ID TLV. This parameter is simply the message ID of the Label Request message that prompted this Label Mapping message, which enables an upstream LSR to correlate a given Label Mapping message with the request that instigated the message. The example signal flow of Figure 8-25 shows the use of the Label Request message ID TLV.

In addition to the Label Request and Label Mapping messages, we also have the Label Abort Request message, the Label Withdraw message, the Label Release message, the Address message, and the Address Withdraw message.

An LSR sends a Label Abort Request message when it wants to cancel a Label Request and has not yet received a Label Mapping message in response to the original request. The Label Abort Request message includes the message ID of the original request so that the downstream LSR knows which request is being cancelled.

A downstream LSR sends a Label Withdraw message to an upstream LSR to indicate that the upstream LSR can no longer use a particular FEC/label binding. An upstream LSR sends a Label Release message to a downstream LSR to indicate that it no longer requires the label/FEC mapping that was previously provided by the downstream LSR. An upstream LSR must send a Label Release message in response to a Label Withdraw message.

An LSR uses an Address message to advertise its interface address(es) to LDP peers. The LSR performs such advertisements prior to the exchange of any Label Request or Label Mapping messages. Moreover, the LSR sends an Address message whenever a new interface is added to the LSR. An LSR uses the Address Withdraw message to withdraw previously advertised interface addresses. An LSR would send also the message if, for example, an interface is taken out of service. Both the Address and Address Withdraw messages contain an Address List TLV. This TLV comprises an address family field indicating the type of addresses in the list (IPv4 or IPv6), followed

by a list of addresses corresponding to the interfaces being advertised or withdrawn.

Notification messages are used to provide advisory information and signal error information. The Notification message includes a Status TLV, which can carry status or error information. If, for example, a loop were detected during LSP establishment, then a Notification message would be returned with Status Code 0000000B (loop detected). A Notification message can optionally contain an LDP PDU (or part of one) or an LDP message (or part of one) if a particular PDU or message generated the error. For example, if a given LDP message caused an error at the receiving LSR, then that LSR would return a Notification message with an appropriate status code plus as much of the received message as necessary to indicate the problem in the message.

Constraint-Based LDP (CR-LDP) The foregoing description of LDP shows how an LSP can be established end to end. The LSP is effectively a network path that is associated with a given FEC. As we have also described, the significance of a given FEC is established at the network edge, where information such as the bandwidth requirements, IP ToS setting, the ingress router, and the incoming interface can be known. Unfortunately, LDP does not enable that additional information to be passed in LDP messages as part of LSP establishment. LDP only passes the destination address or address prefix, which does not enable downstream LSRs to make LSP routing decisions that can take other information into account. For example, an LER might want to establish an LSP that has certain bandwidth requirements. We need a mechanism whereby such requirements can be conveyed during label advertisement and mapping so that the LSP establishment can include the allocation of the appropriate resources. CR-LDP is one way to do this.

CR-LDP is specified in RFC 3212 and is an extension of LDP, whereby an LSP can be established subject to certain constraints, such as traffic parameters, resource requirements, and other characteristics. In fact, CR-LDP can enable the establishment of LSPs according to explicit routing requirements such that the LSP establishment messages can specifically indicate which LSRs should be included in the LSP.

Explicit Routes (ERs) A *Constraint-Based LSP* (CR-LSP) is an LSP that is established subject to a number of criteria. It is established based on information available at the edge of the network. An *explicit route* (ER) is one type of CR-LSP where some or all the nodes to be used are specified. In a strict ER, all nodes in the path are specified and the path is not allowed to

include nodes that are not specified. In a loose ER, a number of nodes in the path are specified, but other nodes can also be allowed. CR-LDP enables ER information to be included in the LDP Label Request message. The overall idea is to define very specific paths for traffic that has specific characteristics.

Traffic Characteristics CR-LDP enables a number of traffic characteristics to be specified through the use of a traffic parameters TLV, as shown in Figure 8-28. The peak rate indicates the maximum rate at which traffic should be sent to the CR-LSP. It includes two components: the *peak data rate* (PDR), measured in bytes/second, and the *peak burst size* (PBS), measured in bytes. Each of these could be components of a token bucket specification where the bucket is filled at a rate equal to PDR and the bucket size equals PBS.

The committed rate indicates the rate that the network commits to be available to the CR-LSP. It has the components *committed data rate* (CDR) and *committed burst size* (CBS). These may also be used in a token bucket specification. The value of PDR should be greater than or equal to the value of CDR.

The *excess burst size* (EBS) can be used at the edge of the MPLS domain for the purposes of traffic conditioning and can be used to measure the extent by which traffic exceeds the committed rate. The EBS could also form part of a token bucket specification, where the token bucket size is EBS and the token rate is CDR.

The frequency indicates how often the CDR should be made available. Frequent (1) means that the available bandwidth should average at least

Figure 8-28
CR-LDP traffic
parameter

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3																		
0	0	Traffic Parameter TLV (0801)														Length																													
Flags				Frequency				Reserved				Weight																																	
Peak Data Rate (PDR)																																													
Peak Burst Size (PBS)																																													
Committed Data Rate (CDR)																																													
Committed Burst Size (CBS)																																													
Excess Burst Size (EBS)																																													

the CDR over any short interval. Very frequent (2) means that the available rate should average at least the CDR over any packet interval at the CDR.

The weight determines the CR-LSPs share of any possible excess bandwidth above the committed rate and can be used within a given MPLS domain to determine which traffic streams get access to any available additional bandwidth.

The Flags field of the Traffic Parameters TLV is shown in Figure 8-29. The field includes a number of indicators related to each of the other fields in the Traffic Parameters TLV. Specifically, the flags indicate whether each of the traffic parameters requirements is negotiable or not as follows:

- F1 relates to PDR: 0 = non-negotiable, 1 = negotiable.
- F2 relates to PBS: 0 = non-negotiable, 1 = negotiable.
- F3 relates to CDR: 0 = non-negotiable, 1 = negotiable.
- F4 relates to CBS: 0 = non-negotiable, 1 = negotiable.
- F5 relates to EBS: 0 = non-negotiable, 1 = negotiable.
- F6 relates to weight: 0 = non-negotiable, 1 = negotiable.

If a given component of the Traffic Parameters TLV is negotiable, then an LSR might offer a lower value than is requested. For example, an LER might request a PDR of 10,000 bytes/second. If, however, the LER indicates that the requested PDR is negotiable, then the resultant LSP might support a lower value (say, 8,000 bytes/second).

Given that a request to establish an LSP might include an ER requirement as well as a Traffic Parameters TLV, one might find an inconsistency between the different components of the request. For example, an LER might request that a CR-LSP support a specific traffic profile and follow a specific path. We can easily imagine a situation where the path in question does not have the available resources to support all the requested traffic parameters. In such a case, the request to establish a CR-LSP could be rejected. If, however, certain traffic parameters are negotiable, the network might still be able to establish the CR-LSP along the requested path but with lesser resource commitments than originally requested.

Figure 8-29

CR-LDP traffic
parameter flags

Reserved	F6	F5	F4	F3	F2	F1
----------	----	----	----	----	----	----

Given that many of the traffic parameters of CR-LDP can form part of a token bucket specification, we can make sure that an LSP is established to meet criteria that will be policed at the network edge. In other words, the ingress router could implement a token-bucket-based traffic-policing technique for a given stream and could establish an LSP through the network that meets the requirements of the token bucket specification. Provided that the source of traffic complies with the token bucket specification, network resources will support the traffic.

Resource Classes CR-LDP enables the use of resource classes to specify which links may be used in a given CR-LSP. The purpose is to help limit the set of possible links to those that meet certain resource requirements. For example, a resource class could indicate OC-48, meaning that all links of OC-48 capacity belong to that resource class. Similarly, *Gigabit Ethernet* (Gig-E) could be a resource class. Resource classes are sometimes known as colors, where resources of the same color belong to the same class. A given resource can belong to more than one resource class. Note that CR-LDP does not specify the specifics of resource classes; it simply provides a means for indicating a resource class in LDP messages. CR-LDP does so by allowing for a resource class TLV. It is only necessary that all LSRs within a given domain have the same understanding of resource classes and the contents of the resource class TLV.

Preemption CR-LDP enables a CR-LSP to be allocated to a given traffic trunk, so that a CR-LSP is established based on certain resource requirements. If a CR-LSP cannot be established due to a lack of available resources, then CR-LDP enables one to reroute other traffic in order to make room. This is called *preemption*.

Preemption involves the assignment of two priority levels to a given CR-LSP. The first, called *setupPriority*, indicates the authority assigned to a given CR-LSP to preempt (bump) another. The second, called *holdingPriority*, indicates how much authority is required by another CR-LSP to bump the CR-LSP that currently has the resources. B can only bump A, if B's *setupPriority* is higher than A's *holdingPriority*. Actually, priority values are assigned in descending order. The value 0 is most important and the value 7 is least important. For a given CR-LSP, its *holdingPriority* should be at least as good as its *setupPriority*. Otherwise, having bumped a less important CR-LSP, it can be immediately bumped by a CR-LSP with the same priority as itself.

Modified LDP Messages for CR-LDP Using the new information defined in CR-LDP, the Label Request and Label Mapping messages are modified as shown in Figures 8-30 and 8-31.

The LSPID is a unique identifier for the CR-LSP in an MPLS network. The LSPID is composed of the ingress LSR router ID (or any of its own IPv4 addresses) and a locally unique CR-LSPID to that LSR. The LSPID also includes an action indicator flag, which indicates the action that should be taken if the LSP already exists on the LSR in question. For example, we might want to modify the traffic parameters associated with a given LSR (such as add more bandwidth), in which case the action indicator flag would indicate that the LSP is to be modified. If set to 0, the action indicator flag indicates that this is an initial LSP establishment. The ability to indicate

Figure 8-30

CR-LDP Label Request message

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	
0	Mess. Type = Label Request (0401)																Message Length											
Message ID																												
FEC TLV																												
LSPID TLV																(CR-LDP, mandatory)												
ER-TLV																(CR-LDP, optional)												
Traffic TLV																(CR-LDP, optional)												
Pinning TLV																(CR-LDP, optional)												
Resource Class TLV																(CR-LDP, optional)												
Preemption TLV																(CR-LDP, optional)												

Figure 8-31

CR-LDP Label Mapping message

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	
0	Mess. Type = Label Mapping (0400)																Message Length											
Message ID																												
FEC TLV																												
Label TLV																												
Label Request Message ID																												
LSPID TLV																(CR-LDP, optional)												
Traffic TLV																(CR-LDP, optional)												

the required action means that the parameters associated with an LSP can be changed without having to release and reestablish the LSP.

The ER TLV specifies the path to be taken through the network. The TLV is composed of a number of ER-hop TLVs. Each ER-hop TLV indicates the type of hop (T), the length of the hop address (L), and the value of the address associated with the hop (V). Four types of ER hops are defined: an IPv4 address, an IPv6 address, an autonomous system number, and an LSPID. For each ER-hop TLV, a single bit specifies whether this is a loose hop or a strict hop. If this is a strict hop, then the address in the ER-hop TLV must match the LSR that has received this Label Request message. Otherwise, we have an error situation. If this is a loose hop, then the LSR receiving the message must pass the request on towards the LSP indicated in the address part of the ER-hop TLV.

The ability to indicate an existing LSPID as a hop to be traversed is quite useful and enables us to easily establish MPLS tunnels within tunnels. Recall our earlier example of a transatlantic VoIP operator that purchased international IP transport from an international IP service provider. Let's assume that the international IP provider establishes an MPLS tunnel for the VoIP operator to use. That tunnel could have a specific LSPID. If the VoIP operator wants to establish multiple LSPs through that tunnel (one LSP for bearer traffic, one for signaling traffic, one for admin traffic, and so on), the establishment of each of the LSPs could specify the LSPID of the overall tunnel as one of the ER-hop TLVs. This action would effectively establish each new LSP as a tunnel within the overall tunnel allocated to the VoIP operator.

The Pinning TLV is used with loose ERs. It includes a means of indicating whether or not a path may be changed at a given LSR if a better next hop becomes available later. Of course, such changes cannot apply to a strict ER.

Enabling Traffic Engineering and QoS with CR-LDP The foregoing description of CR-LDP indicates how we can apply traffic requirements and explicit routing requirements to the establishment of an LSP. By ensuring that an LSP is established according to certain criteria, we can ensure that the traffic carried on that LSP always gets the resources it needs. For example, we can set up an LSP that meets certain resource requirements (such as bandwidth) and reserve those resources so that they are dedicated to that FEC's traffic. Moreover, we can specifically indicate the route that a given LSP needs to take through the network.

Imagine a traffic-engineering tool used to engineer an MPLS network. The tool has a *graphical user interface* (GUI) that shows the topology of the

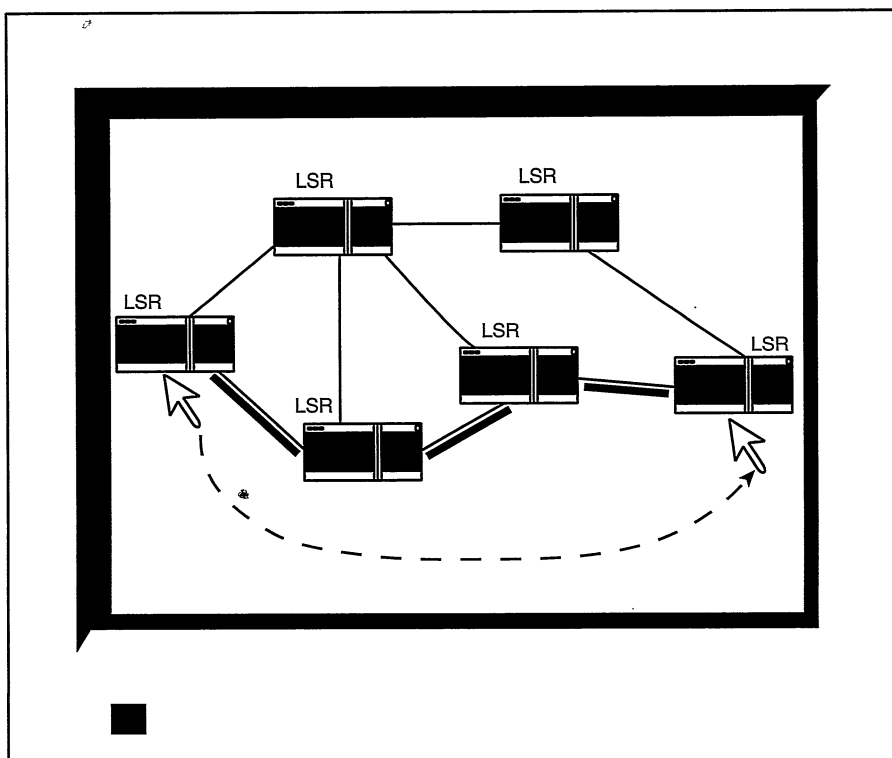
IP network, as depicted in Figure 8-32. The tool is provisioned with data regarding the network resources (such as router configurations and link bandwidths) and has a command interface to the various LSRs in the network.

To establish an LSP, the network architect would simply “drag” the LSP from one side of the network to the other on the tool’s GUI interface. The tool would then prompt the user for LSP constraints such as bandwidth requirements, explicit routing requirements, and so on. Once the user has entered the required data, the tool would send the appropriate command to the first LSR, which would trigger the LSR to send a CR-LDP Label Request message through the network with the various parameters. The various routers would reserve the necessary resources and send the necessary CR-LDP Label Mapping messages to set up the LSP. Not only would the network have established the LSP, the tool would also store information about the LSP.

If the network architect wants to set up another LSP, then he or she would repeat the process. This time, however, the tool would be aware of

Figure 8-32

Example GUI for an
MPLS engineering
tool



committed resources on the network and command the edge router to send a CR-LDP Label Request message that meets the requirements for the required LSP while also maintaining the integrity of the first LSP.

RSVP Traffic Engineering (RSVP-TE)

Our discussion of RSVP earlier in this chapter focused on the usage of RSVP for the reservation of resources on a session-by-session basis. When used in such a manner, RSVP has difficulty scaling to support large systems. With the appropriate extensions, however, RSVP can be easily adapted to resource reservation for traffic trunks rather than individual sessions. In other words, we can reserve resources for traffic of a given class from one edge of the network to the other rather than for each individual session.

RSVP-TE is defined in RFC 3209, "Extensions to RSVP for LSP Tunnels." This specification describes the additions made to RSVP messages (PATH and RESV) to enable RSVP to be used as the label distribution and traffic-engineering protocol for MPLS networks.

In order to support LSP establishment using RSVP, a number of new objects are added to RSVP messages: the Label_Request object, the Label object, the Explicit_Route object, the Record_Route object, and the Session_Attribute object.

The Label_Request object can have three formats. One has no specific label range, one includes the maximum and minimum VPI and VCI values and would be used with ATM interfaces, and one includes the maximum and minimum DLCI values and would be used with Frame Relay interfaces.

The Label object is a four-octet parameter with the same structure as shown in the label portion of Figure 8-19 earlier in this chapter.

The Explicit_Route object is a list of addresses that a particular LSP must follow and is similar to the ER TLV of CR-LDP, which is composed of a number of ER-hop elements. Like the ER-hop elements of CR-LDP, the Explicit_Route addresses of RSVP-TE have a loose/strict indicator bit.

The Record_Route object is a list of IPv4 or IPv6 addresses for the nodes that a given RSVP message has passed through. The object can be used to detect loops in RSVP signaling and it can also be used as an input to the Explicit_Route object. In some ways, this object is similar to the Path Vector TLV of LDP.

The Session_Attribute object is similar to the Preemption TLV of CR-LDP, and it includes a setupPriority and holdingPriority for an LSP.

RSVP establishes LSPs by using these new objects with the existing RSVP PATH and RESV messages. The PATH message is enhanced to include the Label_Request object and optionally the Explicit_Route and Session_Attribute objects. The PATH message already includes traffic data (sender TSpec, ADSpec, and so on), but the traffic data in this case would apply to a traffic trunk rather than an individual session between individual users. With the addition of the new objects, the PATH message now includes much of the same information that is included in a CR-LDP Label Request message and can be used for the same purpose.

The RESV message is enhanced to include the Label object and optionally the Record_Route object. These new objects are attached to a filter spec in an RESV message. Recall that the filter spec in RSVP indicates a specific traffic flow. If a Filter spec is applied to a traffic trunk between two points, then the filter spec is equivalent to an identification of an FEC. If we use RSVP to assign a label to that filter spec, then we have effectively assigned a label to the FEC.

Based on the foregoing, we can see that LSP establishment using RSVP-TE would work as follows. The ingress LSR would send an RSVP PATH message with traffic parameters for the traffic trunk in question with a Label_Request object, and potentially one or more of the Explicit_Route, Record_Route, and Session_Attribute objects. That message would propagate through the network to the point of egress. At that point, an RSVP RESV message would be returned, including a Label object and optionally a Record_Route object. The net effect of this process is the same as that of CR-LDP. An LSP is established through the network according to specific traffic requirements and potentially along a specific, predefined path.

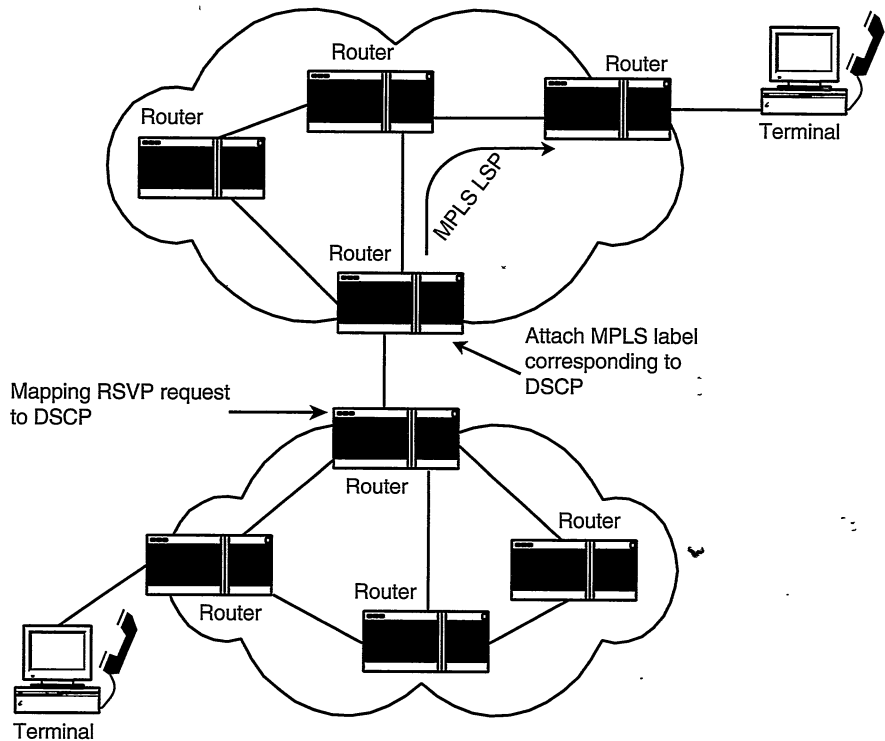


Combining QoS Solutions

The QoS solutions described in this chapter take different approaches and each has its advantages and disadvantages. Although RSVP at the session level is very powerful, it has the drawback that it requires a path state to be maintained in each router, something that can be difficult and costly in a large network. DiffServ is simpler, but it is more of a prioritization technique than a resource-guarantee mechanism. MPLS offers great promise as an overall solution, but requires significant changes to all routers that want to use it. Therefore, each of the solutions has its drawbacks and no single solution is likely to be a silver bullet.

Figure 8-33

Combining QoS techniques for an end-to-end solution



Combining the different solutions in smart ways, however, can allow them to be used in different parts of the network where they can be put to best use. For example, if RSVP is used in one domain and DiffServ is used in another, then it is possible to map an RSVP service request (such as guaranteed service) to an appropriate DiffServ PHB (such as EF). Similarly, it might be possible to map a particular DiffServ behavior aggregate to an MPLS FEC at an MPLS edge router. Figure 8-33 gives one example of how different solutions might be combined.

CHAPTER

9

Designing a Voice over IP Network

In general, the design of any network involves striking a balance between three requirements: meeting or exceeding the capacity needed to handle the projected demand, minimizing the capital and operational cost of the network, and ensuring high network reliability and availability. In short, we can refer to these three issues as cost, capacity, and quality. Meeting one or more of the requirements often means making sacrifices elsewhere, such that it is impossible to divorce one network design consideration from any of the others. For example, a lower cost might well mean a lower network capacity or a lower network quality. Thus, we will never get a network that is remarkably cheap to implement and operate while still offering high capacity and high quality. Instead, we must aim to establish some happy medium where we satisfy at least the most important criteria to an acceptable degree. Of course, the question is, "What is an acceptable degree?"

If we want to design a carrier-grade network, one design criteria to include is an overall availability of at least 99.999 percent (five-nines), which corresponds to a down time of no more than about 5 minutes per year. Other design criteria will be based upon projected usage of the network and business considerations. For example, network capacity will be driven by expected traffic demand. Network cost will usually be the result of a negotiation between the carrier's engineering organization and the financial/commercial department. At the end of the day, the cost of the network must be sufficiently low to ensure that it costs less to build and operate than the revenue it will generate.

In general, designing a network has no "one-size-fits-all" solution. A number of general rules and an overall approach, however, can be applied in most network design efforts. The overall approach includes the following aspects:

- Understanding the expected traffic demand, such as where traffic will come from and go to, and what typical per-subscriber usage is expected
- Establishing network design criteria, such as build-ahead, voice-coding schemes, network technology (such as softswitch versus H.323), whether to use silence suppression, whether to include network-level redundancy, and so on
- Vendor and product selection, including the development of a *Request for Information* (RFI) or *Request for Proposal* (RFP) and the comparison of vendor responses
- Network topology, connectivity, and bandwidth requirements
- Physical connectivity

Often the network design process is iterative. For example, one might apply certain design criteria to develop a draft design that meets capacity

requirements only to find that such a design would be too expensive. In such a case, one might need to revise the design assumptions in order to design a more cost-effective network.

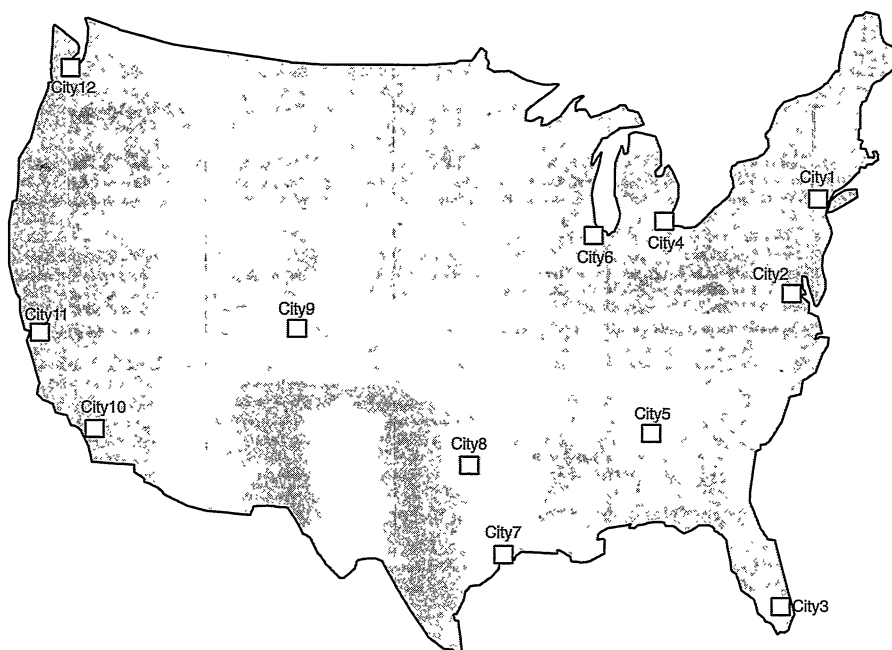
In order to illustrate the various steps in the process, we focus much of this chapter on a fictitious *Voice over IP* (VoIP) network. We will design this network from scratch, with an explanation of each step along the way. For some of the steps, we will perform all the detailed calculations. The fictitious network will be required to provide long-distance service in North America. Our network will provide service to subscribers in 12 cities, City 1 through City 12, as shown in Figure 9-1.

Design Criteria

In order to develop a network design, we must establish certain design criteria. Among other things, we need to specify whether we will have a build-ahead or some capacity buffer. For example, it is pointless to design a network that will only support the immediate demand. Taking such an

Figure 9-1

The cities for the VoIP example network design



approach, however, would mean that we would constantly be redesigning the network as traffic increases.

Build-Ahead or Capacity Buffer

Rather than design for the immediate expected load, we could include a certain build-ahead in the design process. For example, we could include a one-year build-ahead, which means that the initial network implementation will be designed for the traffic load expected one year after launch. (We establish example traffic projections later in this chapter.)

A build-ahead serves two purposes. First, it avoids the necessity for constant redesigning as traffic demand increases. Second, it provides a buffer in case traffic demand increases faster than expected. Given that our network design will be based on a number of input assumptions, and given that assumptions can be incorrect, such a buffer is wise.

Although a build-ahead is a wise approach for a completely new network, a simple design buffer might be better if we do not expect sudden, drastic traffic growth. Imagine, for example, a situation where we have a traditional circuit-switched network and we want to replace that network with a new VoIP network. In such a case, we should have a good understanding of the traffic demand and it is possible that forecasted growth might not be very great. Nonetheless, it is wise to ensure that the network has some extra capacity to cater for sudden demand changes (as might occur if the marketing department decides to suddenly offer some new pricing deal). One way or another, a network should be sized such that the expected traffic demand is less than some percentage of network capacity (say 70 or 80 percent). For example purposes, in our example design, we will assume a 12-month build ahead.

Fundamental Technology Assumptions

As described in earlier chapters, VoIP has several technology solutions. For example, we could choose to deploy an H.323 architecture or a softswitch-based architecture. Given that the softswitch architecture is becoming the most accepted VoIP approach, let's apply that architecture to our network design.

The technology choice does not end there, however. We must also decide on the type of signaling solution to deploy. Should we use the *Media Gateway Control Protocol* (MGCP) or MEGACO? MEGACO is probably the best

choice as it is the media gateway control protocol of the future. Should we use external *signaling gateways* (SGs) with Sigtran between *media gateway controllers* (MGCs) and SFs or should we deploy MGCs that support *Signaling System 7* (SS7) directly? In many cases, we will find that we need to make a trade-off between what we would ideally like and what vendors can provide at a good price.

For example, we might want MEGACO, but the overall best-value vendor solution might support only MGCP in the near-term. Similarly, we might want to deploy SGs, but the overall best solution on the market might include SS7 links directly on the MGC. In such cases, the agreement with the vendor should include a commitment by the vendor to implement the desired functionality within a specific time period.

Later in this chapter we delve further into the issues surrounding product and vendor selection. In the meantime, let's assume that our solution will use MEGACO between MGCs and *media gateways* (MGs) and the *Session Initiation Protocol* (SIP) between MGCs. Let's also assume that we use separate signaling gateways, with *SS7 MTP3-User Adaptation Layer* (M3UA) over *Stream Control Transmission Protocol* (SCTP) between the MGCs and the SGs.

Network-Level Redundancy

If we are to design a carrier-grade network, then all the products in the network must include significant levels of redundancy. From a network design perspective, however, we must decide whether to include redundancy at the network level.

We saw in Chapter 6, "Media Gateway Control and the Softswitch Architecture," that an MG can be handed off to a secondary MGC if a primary MGC fails. If the secondary MGC normally carries traffic of its own, we should make sure that the secondary MGC has the capacity to handle some (or all) of the load of the new MG in addition to existing traffic. In other words, we must design the network so that excess capacity is available in the secondary MGC. We must also decide how much excess capacity we will include. For example, we might decide that the secondary MGC should have the capacity to handle its normal expected load (including build-ahead), plus only 50 percent of the load that would be carried by the failed MGC.

Implementing network-level redundancy can be done in several ways. If, for example, we have 3 MGCs (MGC-A, MGC-B, and MGC-C), then each could back up another. Thus, MGC-B could be a backup for some or all of MGC-A's traffic, MGC-C could be a backup for some or all of MGC-B's

traffic, and MGC-A could be a backup for some or all of MGC-C's traffic. Alternatively, we might choose to have a single redundant MGC for the whole network. That MGC would begin handling traffic in the event of failure of any MGC on the network.

Again, the choice of redundancy technique depends on the capabilities of the vendor. For example, although a single MGC acting as a backup for all others might be the most cost-effective solution, it might require that the backup MGC be loaded with all the port/termination configuration data related to all the MGs on the network. The on-board availability of such data might be required in order for the backup MGC to quickly assume control in the event of a failure.

The number of supported ports might be a capacity-limiting factor on the MGC, however, which might mean that a single backup MGC for the whole network would not be possible. You might think that we should not have such a problem. After all, MEGACO enables an MGC to query an MG as to the capabilities and current status of its terminations. Thus, the back-up MGC could automatically load data for the MG(s) that it must support without having data in advance for all terminations on all MGs. MEGACO does support such capabilities, but whether a given MGC supports that aspect of MEGACO is another matter and could be vendor dependent. In our example, we will assume that we can implement a single backup MGC to support the failure of any active MGC.

Voice Coder/Decoder (Codec) Selection Issues

We must make a number of decisions regarding voice coding. First, we must choose the actual *coder/decoder* (codec) to use. Second, we must select the packetization interval. Third, we must decide whether or not to use silence suppression. These decisions can have a profound impact on the *quality of service* (QoS) offered to our subscribers. As with many of our decisions, however, a trade-off takes place between the quality we offer, the cost of the network, and the capabilities of our vendors.

Clearly, G.711 is the simplest (and highest-quality) codec we can deploy and should be the first choice if we want to ensure superior voice quality. G.711 uses higher bandwidth than other codecs, however. Of course, we can choose any number of coding schemes (G.723.1, G.729, G.726, and so on), many of which provide good quality and that enable lower-bandwidth consumption than G.711. The network architecture should at least perform a

laboratory test of each of the codec choices under ideal and degraded circumstances before making any firm decision. Those tests should include tests with silence suppression turned on and turned off.

The packetization interval is also important. A longer packetization interval means a net lower overhead for the *Real-Time Transport Protocol* (RTP), *User Datagram Protocol* (UDP), and *Internet Protocol* (IP) headers, which means lower bandwidth consumption overall. On the other hand, a longer packetization interval means a longer delay, which impacts voice quality. Typically, we will find that a packetization interval of about 20 to 40 milliseconds provides a good balance between delay and bandwidth consumption. The exact packetization delay will depend on the chosen codec. For example, G.729 operates on 10-millisecond speech samples, so we could not choose a 25-millisecond packetization interval with G.729. Similarly, G.723.1 operates on 30-millisecond speech samples, which means that we could not choose a 20-millisecond packetization delay with that codec. For our example network design, we will assume that we use G.729B (G.729 with silence suppression) and with a packetization interval of 20 milliseconds.

Voice Activity Factor Since we have decided that we will use silence suppression, our VoIP MGs will transmit little or no IP voice traffic during periods of silence. Consequently, we can reduce the amount of bandwidth that we will require on the network. Therefore, we must decide how much of a typical conversation is actual voice rather than silence. Clearly, if a typical voice conversation is 80 percent voice and 20 percent silence, greater bandwidth will be required compared to 60 percent voice and 40 percent silence.

In fact, we consider voice activity on a one-way basis only. This is because transmission facilities are generally symmetrical. For example, if we deploy a DS3 (approximately 45 Mbps), then we have about 45 Mbps of bandwidth in each direction. If we typically have 40 percent voice activity per person in a call, then there is voice activity for about 80 percent of the time and silence for 20 percent of the time. In each direction, however, we are consuming approximately 40 percent of the bandwidth that would be consumed in a call where each party is speaking 100 percent of the time. In other words, we are consuming 40 percent in the direction from A to B and 40 percent in the direction from B to A, which amounts to 40 percent overall.

Statistics show that in a typical conversation each party speaks for approximately 30 to 40 percent of the time, with the remainder of the time

consumed by silence. For the purposes of our example network design, we will assume a voice activity factor of 40 percent.

Blocking Probability

The Erlang is the standard measure of traffic on a circuit-switched network. Since our VoIP network will have interfaces to external circuit-switched networks (such as the *Public Switched Telephone Network* [PSTN]), we will need to use Erlangs to dimension the interfaces to those networks.

One Erlang corresponds to a channel being occupied for one hour. Alternatively, we can think of an Erlang as representing one hour of conversation. This one hour could be a single conversation that lasts 1 hour, 2 conversations that last 30 minutes each, 5 conversations that last 12 minutes each, and so on. At first glance, one would think that a single channel can carry one Erlang. That is not the case. Imagine 5 callers that each make a 12-minute call. In order for those five calls to all be carried on the same channel in a given one-hour period, then one call would need to begin exactly when another call ends. Different users will not be synchronized in such a manner. Therefore, we will likely find that one of our five callers will attempt to call when a different caller is using the channel. That call attempt will fail.

Depending on the number of available channels and the amount of offered traffic, there is a statistical probability that a channel will be available when a user wants to make a call. If we want to ensure that there will always be a channel available to carry a call, we would need an infinite number of channels, which is obviously not feasible. Instead, we want a high probability that a channel will be available. Alternatively, we want a low probability that a call will be blocked due to a lack of available channels. This probability is known as *blocking probability*.

For interfaces towards the PSTN, we must decide which blocking probability we want. The lower the blocking probability, the greater the service quality and the greater the number of required channels (and hence, greater cost).

Appendix A, "Table of Erlang B," provides a table of Erlang values for combinations of channel counts and blocking probabilities. For our network design, we will assume a blocking probability of 0.1 percent on interfaces between the VoIP network and external circuit-switched networks.

QoS Protocol Considerations and Layer 2 Protocol Choices

As described in Chapter 8, “Quality of Service,” we can choose a number of protocols to help ensure high QoS on the network. The simplest approach is to use *Differentiated Service* (DiffServ), which is a form of traffic prioritization. The *Resource Reservation Protocol* (RSVP) comes closest to circuit emulation, but can have difficulty scaling if resources are reserved on a per-session basis. MPLS can provide the best of both worlds by enabling us to reserve resources for traffic trunks. For the purposes of our example network, we will assume the use of MPLS to help ensure high QoS.

This assumption does not necessarily mean that we will totally disregard DiffServ or RSVP. For example, we might choose to employ RSVP-traffic engineering (RSVP-TE) for Multiprotocol Label Switching (MPLS) label distribution and traffic engineering purposes. Moreover, we can choose to mark voice packets from an MG with a specific *DiffServ Code Point* (DSCP) value, which a *label edge router* (LER) will map to a particular *forwarding equivalence class* (FEC). Alternatively, we might want the MG to label the packet directly.

The ultimate choice will depend on the capabilities of the MG we choose and the routers we choose. It will likely be more effective for the MG to mark packets with a particular DSCP and let a separate LER take care of label assignment. Since we will probably find that one or more other nodes (such as an MGC, SG, or *element management system* [EMS]) will be unable to apply labels directly, we will need a separate LER to perform that function. Given that such an LER exists, we can allow it to function at the edge of the MPLS network for all nodes in a location, which means that we do not need to restrict an MG selection to those products that natively support MPLS.

Clearly, we will need to design a wide area IP network to support traffic to and from the various cities. For that *wide area network* (WAN), we must select a Layer 2 protocol (the layer below IP). We have several choices, such as Frame Relay, *Asynchronous Transfer Mode* (ATM), or the *Point-to-Point Protocol* (PPP). Considering our choice of MPLS for QoS purposes, we do not need ATM for QoS purposes. Therefore, let's assume that we use PPP at Layer 2 in our WAN.

Product and Vendor Selection

Having established our overall network design criteria, and understanding our traffic demand, we must choose vendors and products that provide the functionality, capacity, and quality we desire at a cost-effective price. Traffic forecasts are described later in this chapter. For now, let's assume that we have already determined our estimated traffic demand and that we will use the expected demand as an input to our product and vendor selection process.

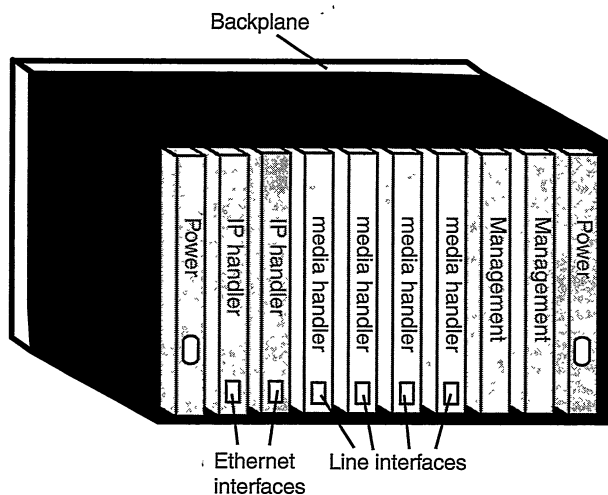
Generally, the selection process will involve the preparation of a *Request for Information* (RFI) or a *Request for Proposal* (RFP). Such documents will address all the functionality and capacity required of the network elements, management, and billing systems. In addition, the RFI/RFP should solicit general information regarding the vendor (such as the company size, revenues, revenues from VoIP solutions, and the existing installed base of the vendor's VoIP solution). The last thing we want to do is take a risk on a vendor that is not financially sound or not sufficiently experienced. Finally, the RFI/RFP should request information regarding interworking that has been performed between the vendor's equipment and that of other suppliers. In general, we want a solution that is standards compliant and that can interwork with products from other vendors.

Generic VoIP Product Requirements

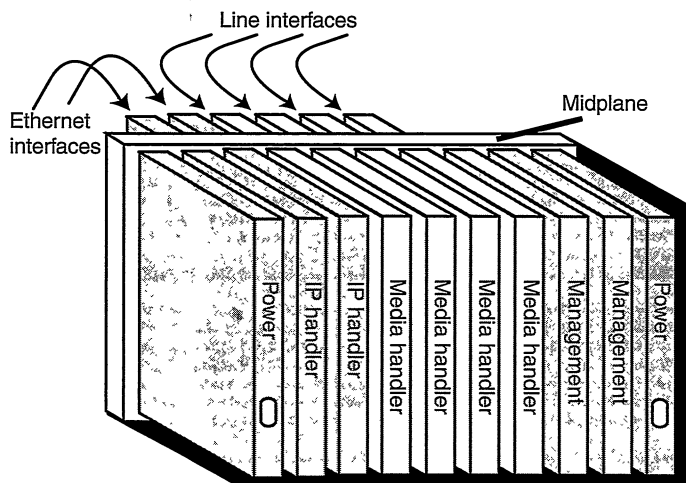
In addition to the requirements that are specific to the network in question (such as support for a specific SS7 or *multifrequency* [MF] signaling variant or for specific codecs), a number of requirements are critical to the performance of any carrier-grade network.

Node-Level Redundancy Every node in the network should include redundancy within itself. Typically, we will require $N + 1$ redundancy for all internal components, management components, media-handling components, power supplies, fans, and disk drives. For example, if an MG requires one active management card to maintain the health of the platform, then we should have at least two such cards, with one as a standby in case the primary fails. If a gateway can be equipped with, say, up to eight media-handling cards, then seven should normally be active, with one spare in case an active card fails.

Typically, an MG will be based on a backplane or midplane design, as shown in Figure 9-2. In a backplane-based system, it is common for media

Figure 9-2Backplane- and
midplane-based MGs

Backplane-based Gateway Architecture



Midplane-based Gateway Architecture

cards to handle both the media manipulation (such as voice coding and echo cancelling) and the physical interface to the outside world. Alternatively, we may find that certain cards are dedicated to line interface functions, while other cards are dedicated to media-handling functions. Moreover, one might

need different cards depending on the coding scheme. Such a situation is generally undesirable. If different cards are needed for different coding schemes, and if we want to implement a variety of coding schemes, then we will need to populate multiple different cards, thereby using up cards that could be used for interfacing with the outside network. The result can be an overall reduction in the call-handling capacity of the network.

A midplane design is often preferable to a backplane design. With a midplane design, the handling of information that is sent on an interface is separated from the handling of the interface itself. In a midplane architecture, the physical interfaces to the outside world typically enter at the back of the chassis, while the cards that process the data on those interfaces are at the front of the chassis. In a midplane architecture, we can designate a spare media-handling card at the front of the chassis. Thus, if a media-handling card fails, the spare can take over and be linked to the interface card that the failed media-handling card was supporting. This arrangement allows for $N + 1$ redundancy of media-handling cards, something that is often impossible with a backplane design.

Of course, redundancy does not apply only to an MG; it also applies to other nodes such as MGCs, SGs, and so on. For each of these nodes, we require redundant interfaces to other nodes (such as redundant Ethernet interfaces). In the case of an SG, we will have SS7 links to separate *signal transfer points* (STPs).

Node Availability Each node on the network should provide at least 99.999 percent availability. One way to gauge the availability of a node is for the vendor to provide *Mean Time Between Failure* (MTBF) values for each component of a given node as well as for the node as a whole. Often, we will find that MTBF values are in tens of years, which means that we could expect decades to pass before a given component fails. Although this sounds like an impressive duration, do not be overly impressed. The MTBF statistics for a complete node will be much lower than the lowest MTBF value for a single component.

Assume, for example, that a gateway has five cards, with MTBF values of 5 years, 10 years, 10 years, 20 years, and 20 years. Then in a given 100-year period, we can expect a total of 50 failures ($20 + 10 + 10 + 5 + 5$). Thus, our overall MTBF is two years.

Although hardware failures can certainly occur, we will most likely find that software failures are more common and in some cases can be catastrophic. A software bug that occurs only under some strange call scenario might not be a major problem, particularly if it means that only the call in question fails. There can be, however, problems that affect all nodes at the

same time. For example, a software problem that occurs only during the changeover to daylight savings time would strike all nodes simultaneously and could disable a complete network. For these reasons, one should require that a vendor provide a list of known software bugs plus copies of tests (and test results) that have been run on the software load being considered.

Alarms and Statistics As a general rule, a given node should produce an alarm whenever a fault situation arises or whenever operations and maintenance staff need to be informed of a specific condition. Typically, alarms will fall into one of three categories: minor, major, and critical. There might also be a fourth: observational, which means the occurrence of a condition that is not a fault but that needs to be communicated to operations personnel.

Not only should nodes provide alarms when faults occur, but alarms should also be generated whenever the load on a network element reaches some percentage of available capacity. For example, an MGC might issue an alarm when total call handling reaches 75 percent capacity, with a further alarm when the load reaches 90 percent capacity. The alarm threshold levels should be settable by the network operator.

Every node should be able to produce a great deal of statistics regarding carried traffic, node performance, and so on. For example, an MGC should be able to provide statistics and reports regarding traffic patterns (including when the busy hour occurs, traffic distribution, and *mean holding time* [MHT]). Equally, an MG should provide reports and statistics regarding IP network performance (such as packets lost, average latency, and jitter) and circuit-switched network performance (such as trunk hold times). A large range of statistics and reports is vital for the network operator to fully understand the performance of the network.

Element Management

How we will manage the network elements is a major consideration in vendor selection. Often, for example, element management considerations might force us to choose a single vendor for all network elements rather than mix and match MGCs, SGs, and so on. Although we might be tempted to pick the best in class for each type of network element, such an approach might lead to difficulty, as each type of network might need to be managed separately.

Imagine, for example, that we need to provision a new trunk group on the network. We need to provision data on an MG, an MGC, and potentially

on an SG (for example, if we have an M3UA application server associated with the new trunk group). It would be nice to have a single EMS that would be used to provision all the correct data for the various nodes, ensuring the consistency of data between the nodes. If we have to use separate provisioning systems to provision each type of node, we run the risk of making errors and introducing data inconsistencies across network elements.

Typically, the interface between the network elements and the EMS will use the *Simple Network Management Protocol* (SNMP) version 2 or higher. Above the EMS, we might require or already have a *Network Management System* (NMS). This situation is often the case where a VoIP network is being introduced as part of a larger network that includes circuit-switched equipment, signaling equipment, transmission equipment, and so on. The EMS will need to have a northbound interface towards the EMS. That interface should be standardized based on, for example, SNMP or the *Common Object Request Broker Architecture* (CORBA).



Traffic Forecasts

Obviously, we need to design a network that will support the projected traffic demand. Consequently, projecting subscriber usage is a critical first step in the network design process. This projection often involves a certain amount of up-front guesswork, particularly if this is the first network of a given type in a given market. If one is building a network in competition with another provider, then one can forecast subscriber growth based on the competitor's subscriber numbers, which are often publicly available. For example, one might assume that the new network will attract a certain percentage of subscribers of a given type (for example, 10 percent of the residential users and 10 percent of the business users in the first year, greater percentages in the second year, and so on). Typically, a marketing or forecasting group will be responsible for determining the subscriber forecast and providing that information to the network designers.

Let's assume the subscriber counts shown in Table 9-1.

Voice Usage Forecast

Let's assume that we intend to provide service only to residential users (which is done by many of the 1010-XXX long-distance service providers). Let's also assume (for simplicity) that the average user in each city makes

Table 9-1

Subscriber
projections

	Launch + 3 months	Launch + 6 months	Launch + 9 months	Launch + 1 year	Launch + 18 months	Launch + 2 years	Launch + 3 years
City1	15,000	48,000	98,800	159,900	301,200	427,600	627,200
City2	8,300	26,400	54,400	88,000	165,700	235,200	345,000
City3	9,000	28,800	59,300	96,000	180,800	256,600	376,400
City4	7,800	25,000	51,400	83,200	156,700	222,400	326,200
City5	9,800	31,200	64,300	104,000	195,800	278,000	407,700
City6	12,600	40,400	83,000	134,400	253,100	359,200	526,900
City7	11,700	37,500	77,100	124,800	235,000	333,600	489,300
City8	11,300	36,000	74,100	120,000	225,900	320,700	470,400
City9	7,500	24,000	49,400	80,000	150,600	213,800	313,600
City10	13,800	44,200	90,900	147,200	277,200	393,400	577,100
City11	12,000	38,400	79,100	128,000	241,000	342,100	501,800
City12	10,500	33,600	69,200	112,000	210,900	299,400	439,100

about the same number of calls. Thus, we assume that any differences in overall traffic demand will be based on differences in subscriber numbers between cities rather than differences in average per-user demand. Since we typically bill per minute and on a monthly basis, we will commonly find that traffic forecasts are often provided in terms of *minutes of use* (MoU) per subscriber per month.

Let's assume our minutes of use per subscriber are as shown in Table 9-2. These figures apply to originating calls only.

The first job of the network designer is to convert such a forecast into measurements that are more meaningful in network design. In particular, the network designer must establish the traffic demand during the busy hour and the MHT per call.

Example: Average User with 120 MoUs per Month of Domestic Long-distance Traffic Assume, for example, that 60 percent of the traffic occurs during weekdays (that is, 40 percent on weekends) and that there are 21 work days per month. Assume that on a given day, 20 percent of the

Table 9-2

Per-subscriber
minutes of use
projections

	Launch + 3 months	Launch + 6 months	Launch + 9 months	Launch + 1 year	Launch + 18 months	Launch + 2 years	Launch + 3 years
City1	110	115	120	122	125	130	135
City2	110	115	120	122	125	130	135
City3	110	115	120	122	125	130	135
City4	110	115	120	122	125	130	135
City5	110	115	120	122	125	130	135
City6	110	115	120	122	125	130	135
City7	110	115	120	122	125	130	135
City8	110	115	120	122	125	130	135
City9	110	115	120	122	125	130	135
City10	110	115	120	122	125	130	135
City11	110	115	120	122	125	130	135
City12	110	115	120	122	125	130	135

voice traffic occurs during the busy hour. For a domestic long-distance service, much of the traffic is likely to occur in the evening, as opposed to business-generated long distance, which occurs during work hours. Then the average busy-hour usage (in minutes of use) per subscriber is given by

$$(\text{MoUs per month}) \times (\text{fraction during work days}) \times (\text{percentage in busy hour}) / (\text{work days per month})$$

Thus, in our example, we get $120 \times 0.6 \times 0.20/21 = 0.686$ MoU/sub/busy hour.

We convert this number to Erlangs, which is the standard measurement of voice network traffic. An Erlang is equivalent to one hour of usage. Therefore, one Erlang is equivalent to 60 MoUs. Thus, busy-hour Erlangs/sub = $0.686/60 = 0.0114$ Erlangs/sub/busy hour.

Next, we need to understand the average call length, known as the MHT. The combination of Erlangs and MHT will enable us to determine the number of calls in the busy hour as well as the total traffic. The quantity we seek

is the number of *busy-hour call attempts* (BHCA). Although busy-hour Erlangs is often a driving factor in the design of a network, so is the BHCA. Typically, we will find that Erlangs are the driving factor for those network elements that reside in the bearer path, while BHCA tends to be the critical factor for call-control entities such as MGCs. Let's assume that the average call length is 5 minutes (300 seconds).

BHCA/subscriber is given by

$$\text{BHCA} = \text{Erlangs} \times 3600/\text{MHT}$$

Thus, we have $\text{BHCA}/\text{sub} = 0.0114 \times 3600/300 = 0.137$.

In other words, we assume that a subscriber with 120 MoUs per month will make 0.137 calls each busy hour.

If we apply the above busy hour and MHT assumptions (60 percent in the work week, 20 percent in the busy hour, 21 work days in a month, and a 300-second MHT), we can use simple spreadsheet formulas to convert the assumptions of Tables 9-1 and 9-2 into total busy-hour Erlangs and BHCA requirements for each city. The busy-hour Erlangs per city are shown in Table 9-3 and the BHCA values per city are shown in Table 9-4. As was the case for Table 9-2, the figures in Tables 9-3 and 9-4 apply only to calls originated by our subscribers, not calls made to our subscribers.

Note that the input assumptions of Tables 9-1 and 9-2 should be revisited on a regular basis to make sure they are still valid.

Traffic Distribution Forecast

Knowing how much traffic will originate in each city is one thing. Knowing where that traffic will go is something else. Every call that enters the network must also leave. If a given call enters the network at City 1 and leaves at City 2, then resources will be needed in each city, not just at the point of ingress. Therefore, we need to understand the traffic distribution. In other words, for each city, we need to know where the calls will go.

Table 9-5 provides an example of traffic distribution for each city in our network. We can see that the table includes data for traffic that is destined for a city not served by the network. The fact is that our example network is rather simple, as it involves only 12 cities. If our network were to be deployed, the network operator would need to partner with some other operator to terminate calls outside network footprint. In fact, due to the limited scope of our example, most of the calls handed to the network in a given city would likely be destined for a location not served by the network and

Table 9-3

Busy-hour erlangs
(originating) per
city

	Launch + 3 months	Launch + 6 months	Launch + 9 months	Launch + 1 year	Launch + 18 months	Launch + 2 years	Launch + 3 years
City1	157.1	525.7	1129.1	1857.9	3585.7	5294.1	8064.0
City2	87.0	289.1	621.7	1022.5	1972.6	2912.0	4435.7
City3	94.3	315.4	677.7	1115.4	2152.4	3177.0	4839.4
City4	81.7	273.8	587.4	966.7	1865.5	2753.5	4194.0
City5	102.7	341.7	734.9	1208.4	2331.0	3441.9	5241.9
City6	132.0	442.5	948.6	1561.6	3013.1	4447.2	6774.4
City7	122.6	410.7	881.1	1450.1	2797.6	4130.3	6291.0
City8	118.4	394.3	846.9	1394.3	2689.3	3970.6	6048.0
City9	78.6	262.9	564.6	929.5	1792.9	2647.0	4032.0
City10	144.6	484.1	1038.9	1710.3	3300.0	4870.7	7419.9
City11	125.7	420.6	904.0	1487.2	2869.0	4235.5	6451.7
City12	110.0	368.0	790.9	1301.3	2510.7	3706.9	5645.6

Table 9-4

Originating BHCA
per city

	Launch + 3 months	Launch + 6 months	Launch + 9 months	Launch + 1 year	Launch + 18 months	Launch + 2 years	Launch + 3 years
City1	1,886	6,309	13,550	22,295	43,029	63,529	96,768
City2	1,043	3,470	7,461	12,270	23,671	34,944	53,229
City3	1,131	3,785	8,133	13,385	25,829	38,123	58,073
City4	981	3,286	7,049	11,600	22,386	33,042	50,328
City5	1,232	4,101	8,818	14,501	27,971	41,303	62,902
City6	1,584	5,310	11,383	18,739	36,157	53,367	81,293
City7	1,471	4,929	10,574	17,401	33,571	49,563	75,492
City8	1,421	4,731	10,162	16,731	32,271	47,647	72,576
City9	943	3,154	6,775	11,154	21,514	31,765	48,384
City10	1,735	5,809	12,466	20,524	39,600	58,448	89,038
City11	1,509	5,047	10,848	17,847	34,429	50,826	77,421
City12	1,320	4,416	9,490	15,616	30,129	44,482	67,747

would need to be handed off to a partner network immediately. Therefore, the distribution in Table 9-5 is not realistic, but it will serve the purpose of illustrating the type of information needed.

Node Locations and Bandwidth Requirements

When we have established our expected traffic demand and traffic distribution, developed our design criteria, and selected our vendors and products, the next step is to establish the initial network topology. The topology of the network will specify how many network elements of a given type will be in each location and the bandwidth requirements between those network elements and the outside world. Initially, we will deal with logical connectivity. Later, once we are satisfied with the logical connectivity (such as the bandwidth required between nodes), then we can develop greater detail regarding the actual connections. That greater detail will involve the specifications of physical connections to routers, such as point-to-point versus ring transport, and so on.

MG Locations and PSTN Trunk Dimensioning

Clearly, we will need to place at least 1 MG in each of the 12 cities where we will provide service. Given that our MGs will connect to the PSTN, our first task is to determine the size of the trunk groups to the PSTN. To do so, we use the Erlang values of Table 9-3 and the traffic distribution statistics of Table 9-5. The Erlang values of Table 9-3 indicate the amount of traffic that we will receive from the PSTN in each city. From the traffic distribution percentages of Table 9-5, we can calculate how much traffic will be sent to the PSTN in each city. Once we have the total Erlang counts, we can use standard Erlang tables to determine the number of DS0s needed. First, however, we must understand how many individual trunk groups we will have to the PSTN (and to our partner *long distance* [LD] carrier) in each city.

Erlang calculations are not linear. As the number of channels in a trunk group increases, the overall traffic per channel increases. For example, a trunk group with 200 DS0s can carry more than twice the traffic of a trunk group with 100 DS0s for the same blocking probability. Because of this fact, we cannot simply take the total traffic in and out of a gateway, look up an

Table 9-5

Projected traffic distribution

To→		City1	City2	City3	City4	City5	City6	City7	City8	City9	City10	City11	City12	Other
From →	City1	0	10%	8%	7%	9%	12%	8%	7%	5%	11%	8%	7%	8%
	City2	12%	0	8%	5%	8%	11%	9%	7%	6%	12%	9%	5%	8%
	City3	12%	8%	0	5%	10%	10%	7%	6%	5%	11%	9%	7%	10%
	City4	11%	5%	5%	0	9%	15%	9%	8%	6%	10%	7%	6%	9%
	City5	11%	8%	9%	6%	0	9%	7%	7%	5%	10%	9%	8%	11%
	City6	11%	9%	6%	10%	8%	0	7%	7%	6%	9%	8%	7%	12%
	City7	11%	8%	7%	6%	8%	9%	0	13%	8%	9%	7%	5%	9%
	City8	10%	8%	7%	6%	8%	10%	13%	0	8%	9%	8%	5%	8%
	City9	10%	6%	5%	5%	7%	8%	10%	10%	0	9%	8%	8%	14%
	City10	11%	8%	6%	5%	6%	10%	8%	8%	7%	0	13%	9%	9%
	City11	10%	7%	6%	5%	6%	10%	8%	7%	7%	14%	0	10%	10%
	City12	11%	7%	5%	5%	6%	10%	8%	8%	8%	11%	10%	0	11%
	Other	0	0	0	0	0	0	0	0	0	0	0	0	0

Erlang table, and determine the number of channels required. Instead, we must allocate the total traffic across the various trunks groups and perform an Erlang table lookup (or Erlang calculation) for each trunk group individually.

In a real network, we would need to consider the PSTN networks in each city and determine what type of connectivity we need to various PSTN switches. For example, in one city we might choose to simply interface to one or two *Local Exchange Carrier* (LEC) tandem switches. In other cities, we might choose to include some direct end-office trunks, which can result in cheaper interconnection under certain circumstances. For simplicity, however, let's just assume that we connect to three separate LEC tandems in each city and also assume that the traffic is evenly spread across those tandems.

Therefore, in each city we will have three both-way trunk groups to and from separate LEC tandem switches. In addition, we will have a one-way trunk group from our MG(s) to a partner LD carrier. Recall that some of the incoming traffic from the LEC will be destined for a city that our network does not serve. We will need to send that traffic directly to a partner LD carrier (the Other column in Table 9-5). Consequently, we will need a trunk group to that partner network. Since we do not expect the partner network to send traffic to our network, we can assume that the trunk groups to the partner network will be one-way.

To determine the size of the trunks to the LEC and our partner network in each city, we combine the information of Tables 9-3 and 9-5. Table 9-5 tells us how much traffic we will receive from the LEC in a given city. Table 9-5 also tells us how much traffic we will send to the LEC in each city and our partner LD carrier in each city. Table 9-6 provides the result of combining Tables 9-3 and 9-5 and tells how much traffic will be carried on each of the trunk groups to and from each gateway location.

The quantities in Table 9-6 are calculated from Tables 9-3 and 9-5. For each city, Table 9-3 provides a projected traffic volume towards the network. Since we are assuming a 12-month build-ahead, we assume the traffic load specified in the Launch + 1 year column of Table 9-3. Consider City 1. Table 9-3 tells us that we expect to receive 1857.9 Erlangs of traffic from the PSTN. Clearly, the MG(s) in City 1 must handle that traffic. Moreover, we have already assumed that the LEC will have three trunk groups and that the traffic will be spread evenly across those trunks groups. Thus, we receive 619.3 Erlangs on each trunk group from the LEC.

From Table 9-5, we can see that 8 percent of the 1857.9 Erlangs received from the LEC must be immediately passed on a trunk group to our LD partner (the LD carrier that carries traffic to destinations that our network does

Table 9-6

Erlang demand per city and trunk group

	LEC Trunk Group 1			LEC Trunk Group 2			LEC Trunk Group 3			LD Partner Trunk Group		
	From	To	Total	From	To	Total	From	To	Total	From	To	Total
City1	619.3	513.2	1132.5	619.3	513.2	1132.5	619.3	513.2	1132.5	0	148.6	148.6
City2	340.8	392.0	732.8	340.8	392.0	732.8	340.8	392.0	732.8	0	81.8	81.8
City3	371.8	327.9	699.7	371.8	327.9	699.7	371.8	327.9	699.7	0	111.5	111.5
City4	322.2	302.6	624.8	322.2	302.6	624.8	322.2	302.6	624.8	0	87.0	87.0
City5	402.8	378.3	781.1	402.8	378.3	781.1	402.8	378.3	781.1	0	132.9	132.9
City6	520.5	498.3	1018.8	520.5	498.3	1018.8	520.5	498.3	1018.8	0	187.4	187.4
City7	483.4	411.3	894.6	483.4	411.3	894.6	483.4	411.3	894.6	0	130.5	130.5
City8	464.8	388.8	853.5	464.8	388.8	853.5	464.8	388.8	853.5	0	111.5	111.5
City9	309.8	325.9	635.7	309.8	325.9	635.7	309.8	325.9	635.7	0	130.1	130.1
City10	570.1	499.6	1069.7	570.1	499.6	1069.7	570.1	499.6	1069.7	0	153.9	153.9
City11	495.7	427.4	923.2	495.7	427.4	923.2	495.7	427.4	923.2	0	148.7	148.7
City12	433.8	347.5	781.3	433.8	347.5	781.3	433.8	347.5	781.3	0	143.1	143.1

not serve). Thus, we must send 148.6 Erlangs from City 1 to our LD partner. Also from Table 9-5, we can see that 12 percent of the traffic that arrives from the LEC in City 2 is destined for City 1. This corresponds to 122.7 Erlangs, which must be handled by the MG(s) in City 1 and passed to the LEC in City 1. Similarly, Table 9-5 tells us that a certain percentage of traffic from the various other cities will be destined for City 1 and must be handled by the MG(s) in City 1. We simply sum these amounts to determine the total traffic that must be passed from our MG(s) in City 1 to the LEC in City 1. We have assumed that such traffic will be evenly spread across the three LEC trunk groups.

The same calculations are performed for each city served by our network. As for many of the calculations so far, the calculations can easily be performed with a spreadsheet program.

Based on the values calculated in Table 9-6, we use the Erlang table of Appendix A to determine the number of DS0 channels needed for each trunk group, assuming a blocking probability of 0.1 percent. Alternatively, we can use the Visual Basic® code of Appendix B as a module in a spreadsheet so that the Erlang calculations can be performed automatically. From the number of DS0s, we can determine the number of DS1 interfaces required (if our MGs use DS1 interfaces) or the number of DS3 interfaces required (if our MGs use DS3 interfaces). The results are shown in Table 9-7.

A single DS3 corresponds to 28 DS1s. Table 9-7 assumes that a given DS3 card in an MG can handle DS1s belonging to different trunk groups. This is a normal situation. Not only does Table 9-7 specify how many DS1s are required on each interface, but it also specifies the total number of DS3 cards, including the allocation of an extra DS3 card to provide $N + 1$ redundancy. Based on the information in Table 9-7, the interfaces to our external networks will be as shown in Figure 9-3.

MGC, SG, and EMS Dimensioning and Placement

As shown in Figure 9-3, we will place an MG in each of the 12 cities in our example network. We must also determine the placement of other network elements such as MGCs, SGs, and our EMS. Let's begin with the MGCs.

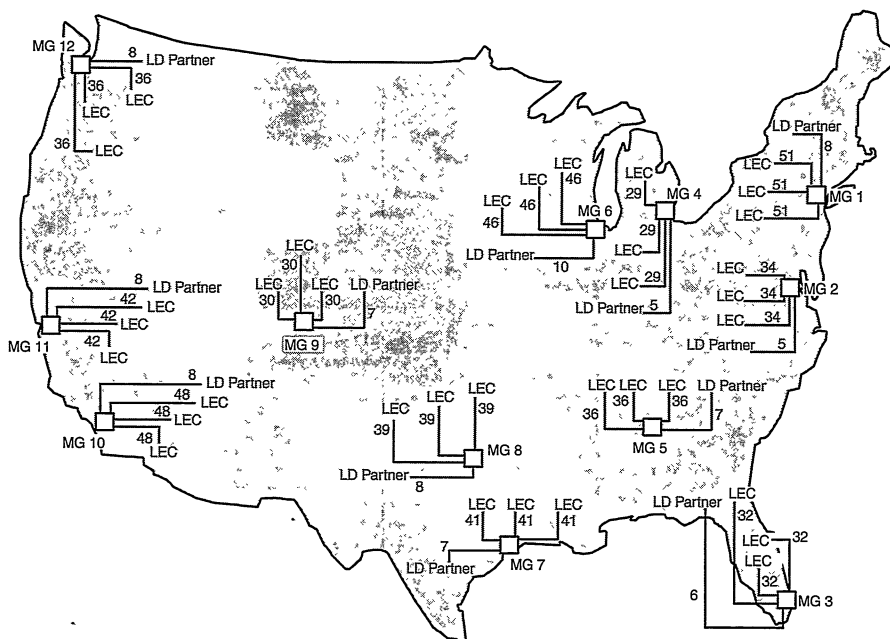
Table 9-7

DS0, DS1, and DS3 interfaces and cards required per MG location

	LEC trunk group 1		LEC trunk group 2		LEC trunk group 3		LD partner trunk group		DS1 Total	DS3 Total	DS3 cards (N + 1)
	DS0	DS1	DS0	DS1	DS0	DS1	DS0	DS1			
City1	1208	51	1208	51	1208	51	183	8	161	6	7
City2	798	34	798	34	798	34	109	5	107	4	5
City3	763	32	763	32	763	32	142	6	102	4	5
City4	686	29	686	29	686	29	114	5	92	4	5
City5	847	36	847	36	847	36	166	7	115	5	6
City6	1093	46	1093	46	1093	46	224	10	148	6	7
City7	965	41	965	41	965	41	163	7	130	5	6
City8	922	39	922	39	922	39	142	6	123	5	6
City9	697	30	697	30	697	30	162	7	97	4	5
City10	1145	48	1145	48	1145	48	189	8	152	6	7
City11	993	42	993	42	993	42	183	8	134	5	6
City12	847	36	847	36	847	36	176	8	116	5	6

Figure 9-3

PSTN DS1
connections for the
example VoIP
network



MGC Quantities and Placement The capacity of an MGC is usually limited by the total BHCA supported. In some cases, we might also find that the total number of MG ports (such as MEGACO terminations) supported is a limiting factor. For our network, let's assume that the BHCA is the limiting factor of the MGC and that the BHCA capacity of an MGC is 150,000 BHCA.

Table 9-4 provides the BHCA projections for our example network. Since we are planning the network with a 12-month build-ahead, we need to consider the BHCA values in the Launch + 1 year column of Table 9-4. At first glance, we might believe that we simply need to divide the BHCA total by the MGC BHCA capacity to determine the number of MGCs required. Unfortunately, the solution might not be that simple. If a call passes between two MGs controlled by the same MGC, then the call consumes capacity on just one MGC. If, however, a call passes between two MGs that are controlled by different MGCs, then that single call consumes capacity on two MGCs. Consequently, the number of MGCs we require depends on which MGs a given MGC is to control and the distribution of traffic to and from those MGs. Consequently, determining the number and location of MGCs can be an iterative process.

We could start by dividing the total BHCA projection by the BHCA capacity of an MGC to develop an initial estimate of the number of MGCs. We then allocate MGs to MGCs. Next we determine the total BHCA to be supported by each MGC, including the impact of calls that cross MGC boundaries, and see if our initial MGC allocation still fits within the MGC BHCA limit. If it does, then we can proceed. If it does not, then we may need to change the MG-MGC allocation, potentially adding MGC capacity.

When establishing the initial MG-MGC allocation, we should arrange the network such that a given MGC will control MGs that are relatively close by. If we look again at Figure 9-1, we can see that it would be preferable to have a single MGC control MGs in Cities 8, 9, 10, 11, and 12, rather than Cities 1, 3, 8, 11, and 12. The former arrangement is simply easier to visualize, while the latter is less intuitive, which could lead to confusion among operations staff.

Given our assumption of 150,000 BHCA capacity per MGC and the BHCA demand of Table 9-4, we make an initial assumption that two MGCs should be able to handle the total demand. Let's therefore assume that MGC1 supports MGs in Cities 1 through 6 and that MGC2 supports MGs in Cities 7 through 12. In terms of placement, let's assume that MGC1 is located in City 2, while MGC 2 is located in City 9.

In order to make sure that this allocation will work correctly, we need to revisit the BHCA demand of Table 9-3 and the traffic distribution of Table 9-5 to make sure that the total BHCA per MGC is less than the MGC BHCA capacity limit. An analysis of Tables 9-4 and 9-5 provides the following information:

- BHCA for calls within footprint of MGC1 = 50,369.7 (a)
- BHCA for calls within footprint of MGC2 = 53,810.5 (b)
- BHCA for calls from MGC1 to MGC2 = 42,420.0 (c)
- BHCA for calls from MGC2 to MGC1 = 45,462.6 (d)
- BHCA load for MGC1 is (a) + (c) + (d) = 138,252.4 BHCA
- BHCA load for MGC2 is (b) + (c) + (d) = 141,693.2 BHCA

Since both of these values are below the BHCA capacity limit of a single MGC, our initial MGC allocation is feasible. We have also assumed that we will have a backup MGC. Let's call it MGC3 and place it in City 8.

SG Quantities, Dimensioning, and Placement Like MGCs, SGs will typically be limited by BHCA capacity. In addition, we might find that the number of SS7 links supported also limits the capacity of an SG. Let's

assume that our SGs will be BHCA limited, with a BHCA capacity of 500,000 BHCA (not an unreasonable number for a carrier-grade machine).

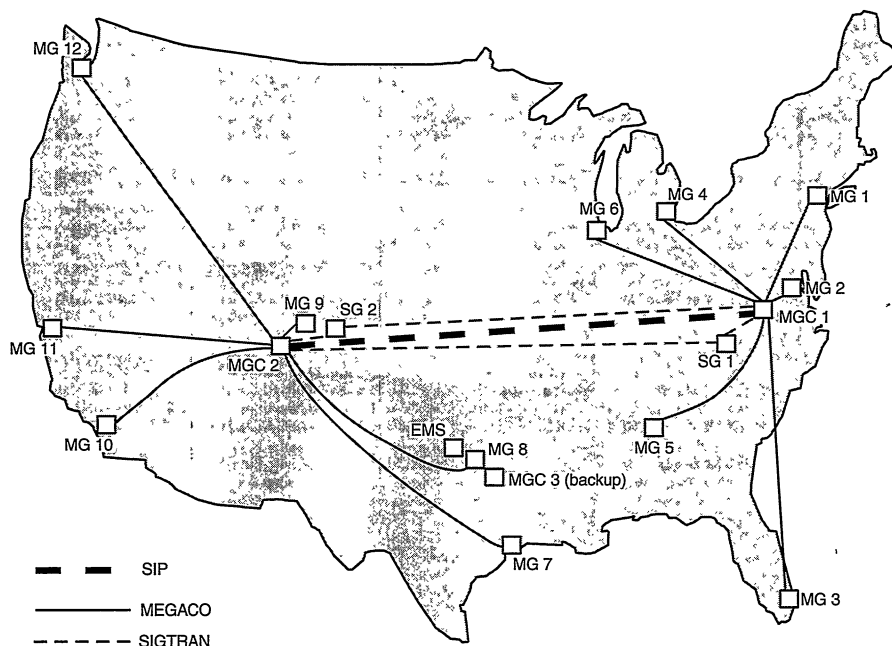
From Table 9-4, we can see that a single SG would have the capacity to support the whole network. We would never allow a single point of failure on our network, however. Therefore, we will deploy two SGs, each of which will be connected to both MGCs. Given that we have placed our two primary MGCs at cities 2 and 9, it is reasonable to place our SGs at the same locations.

From the foregoing discussions, our MG-MGC allocation and SG-MGC allocation are as shown in Figure 9-4. Figure 9-4 also assumes that we have chosen to place our EMS in City 8.

Calculating VoIP Bandwidth Requirements

So far, we have determined the connectivity between our MGs and external networks. We have also determined the types and numbers of interface cards required in each of our MGs. In this section we will calculate the bandwidth required between MGs for VoIP traffic. Later, we will calculate the

Figure 9-4
MG-MGC homing
and signaling
connections



additional network bandwidth needed for IP signaling (such as MEGACO) and *Operation, Administration & Maintenance* (OA&M) purposes.

The bandwidth required for a single call depends on the following factors:

- Voice-coding scheme
- Packetization interval
- The use of silence suppression
- Probability of excessive packet collision if silence suppression is used

Impact of Voice-Coding Schemes Different voice-coding schemes generate different numbers of bits for a certain duration of speech. For example, G.711 operates at 64 Kbps, while G.729 operates at 8 Kbps. For a 20-millisecond sample, G.711 generates 1,280 bits (160 octets), while G.729 generates 160 bits (20 octets).

IP, UDP, and RTP Overhead For every voice packet that we send, we have a good deal of overhead that is sent in addition to the coded voice itself. Recall that voice is carried using RTP over UDP over IP. Each of these protocols has a header that needs to be transported in addition to the voice payload. The IPv4 header is at least 20 octets long (assuming that no optional fields are included). The UDP header is 8 octets long, and the RTP header is at least 12 octets long. Together, IP, UDP, and RTP add 40 octets. Therefore, for a 20-millisecond sample of G.729-coded voice, we have a total of 60 octets. If we use a 10-millisecond sample, then we have a total of 50 octets. Clearly, the smaller the sample size, the greater the net overhead.

Silence Suppression Silence suppression is a valuable tool for reducing the total bandwidth demand. If we assume a voice activity factor of 40 percent, then we can save approximately 60 percent bandwidth compared to a network that does not use silence suppression. As described in Chapter 3, “Speech-Coding Techniques,” however, periods of silence do not mean a complete cessation in the transmission of voice-related packets. During periods of silence, the voice coder will periodically generate periodic *silence descriptor* (SID) frames. A SID frame is quite small compared to a speech frame, however. Moreover, significant time gaps typically take place between SID frames during periods of silence. Consequently, the bandwidth occupied by SID frames is small and can be disregarded.

Peaks in the Number of Simultaneous Speakers Unfortunately, a voice activity factor of 40 percent does not mean that we can simply calcu-

late the bandwidth needed without silence suppression and then multiply by 40 percent to determine the actual required bandwidth. Although a voice activity factor of 40 percent does imply that an average user is speaking for only 40 percent of the time, there might be brief periods where many more people are speaking. If we design the network based only on an average of 40 percent voice activity, then we risk packet delay or loss if brief periods occur when more than 40 percent of users are speaking at the same time. Therefore, we need to include extra bandwidth to accommodate such situations. The question is how much extra bandwidth is needed. The answer is given by a binomial distribution function.

Consider n speakers. If voice activity is 40 percent, then the probability of an individual user speaking at a given instant is 40 percent. If we want to determine the probability that exactly x subscribers are speaking at a given time, then the probability is given by

$$Pa(x) = \binom{n}{x} p^x (1 - p)^{n-x}, \text{ where } p = 0.4 \text{ in our example.}$$

The probability that there are no more than x speakers at a time is given by

$$Pb(x) = \sum_{y=0}^x Pa(y)$$

We need to determine the value x , such that the probability is very high that no more than x speakers are active at a given time. Let's assume that we want a probability of 0.1 percent or less that packets will be lost or delayed as a result of too many speakers talking at one time. This is equivalent to saying that we need to find the value of x such that $Pb(x) = 0.999$ or greater. From an algorithmic perspective, we start at $x = 0$ and determine $Pa(0) + Pa(1) + Pa(2) + \dots$, until the running total exceeds 0.999. The value of x at that point is the value we seek. Fortunately, most spreadsheet programs already have built-in formulas that perform these calculations.

Let's assume a voice activity factor of 40 percent. If, for example, we have up to 1,000 simultaneous calls and we want to make sure that no more than 0.1 percent of the packets are dropped due to too many simultaneous speakers, then the value of x is 448. In other words, we need to dimension the network for 448 simultaneous speakers in a given direction.

Number of Simultaneous Calls for Our Example Network From Table 9-7, we know the number of DS1s that we will deploy to external networks in each city. We can use this information to determine the maximum

number of simultaneous calls in and out of the IP network at each MG. Recall, however, that the DS1s to our LD partner network represent traffic that arrives at an MG from the LEC and is routed directly to the LD partner. Therefore, when considering the bandwidth to the IP network at an MG, we should take the total number of DS1s to and from the LEC and subtract the total number of DS1s to our partner network. The resulting number of DS1 enables us to determine the maximum number of simultaneous calls to and from the IP network at each gateway.

As described previously, we cannot simply multiply the maximum number of simultaneous calls by the voice activity factor (40 percent) and use the result as a basis for calculating our VoIP bandwidth demand. We should use a binomial probability function to accommodate for a situation where more than 40 percent of the users are speaking at a given instant. The results of these calculations are shown in Table 9-8. The column “Max. calls for VoIP bandwidth planning (x)” is calculated based on a voice activity factor of 40 percent and a probability of 99.9 percent that no more than x per-

Table 9-8

DS1 demand
to/from IP network

	Total DS1s to/from LEC	DS1s to LD partner	Net DS1s to/from IP network	Max. simul- taneous calls to/from IP network	Max. calls for VoIP bandwidth planning (x)
City1	153	8	145	3,480	1,482
City2	102	5	97	2,328	1,005
City3	96	6	90	2,160	935
City4	87	5	82	1,968	855
City5	108	7	101	2,424	1,045
City6	138	10	128	3,072	1,313
City7	123	7	116	2,784	1,194
City8	117	6	111	2,664	1,144
City9	90	7	83	1,992	865
City10	144	8	136	3,264	1,393
City11	126	8	118	2,832	1,214
City12	108	8	100	2,400	1,035

sons will be speaking in a given direction at a given instant. When we speak about direction, we mean either the direction from the MG towards the IP network or from the IP network towards the MG. We can see that the values of x are greater than the voice activity factor times the maximum number of simultaneous calls.

As mentioned previously, popular spreadsheet programs include binomial probability functions for the types of calculations required to generate the information in Table 9-8. One drawback of those functions, however, is the fact that binomial calculations include a good deal of factorials, which can lead to extremely high or extremely low interim results prior to the final result. If the input data values are large, the spreadsheet calculations can generate a floating point overflow as the algorithm is being executed. If you detect such a situation, you should be aware that a normal distribution closely approximates a binomial distribution for large values. Therefore, if you find that a spreadsheet binomial function fails due to a large input value, you can use a normal distribution function instead and still achieve a very accurate result.

VoIP Bandwidth for Our Example Network The per-call bandwidth for a VoIP network is calculated as follows:

$$\text{Voice packet size} + 40 \text{ octets (for IP, UDP, and RTP)} + \\ \text{WAN layer 2 overhead} + \text{MPLS overhead (if applicable)}$$

In our case, the voice packet size is based on a 20-millisecond packet duration with G.729 coding and corresponds to 20 octets. For our WAN network, we will use PPP, which has a two-octet header size. In addition, we have stated that we intend to use MPLS within our WAN for QoS purposes. Since we will use PPP at layer 2, we will need to use a generic MPLS label structure, with four octets per label. Assuming just a single MPLS label is in the stack, the use of MPLS and PPP will add a total of six octets to our packet size. Therefore, our packet size is 66 octets. Since each packet corresponds to 20 milliseconds of speech, our total bandwidth per call is 3,300 KBps or 26.4 Kbps. We can clearly see that this is far greater than the native 8 Kbps of G.729, with the extra bandwidth caused by the various overheads we need to accommodate.

Recall from Chapter 2, "Transporting Voice by Using IP," that we use *RTP Control Protocol* (RTCP) in conjunction with RTP for quality feedback purposes. Recall also that RTCP bandwidth should be limited to about 5 percent of the actual VoIP bandwidth. Therefore, we need to add 5 percent to the VoIP bandwidth just calculated. Consequently, the IP network

bandwidth per call is $(26.4 \text{ Kbps}) \times 1.05 = 27.72 \text{ Kbps}$. We multiply this rate by the value x in Table 9-8 to determine the bandwidth required between gateways in the IP network. We then use the traffic distribution of Table 9-5 to determine the amount of bandwidth required between pairs of MGs. As for most of the calculations described in this chapter, simple spreadsheet formulas can be used to make the task relatively easy. The resulting bandwidth requirement between MGs is shown in Table 9-9 in *megabits per second* (Mbps).

Signaling and OA&M Bandwidth Not only will traffic take place between MGs, but traffic will exist between MGs and their controlling MGCs. As a rule of thumb, we allocate the same bandwidth for MG-MGC signaling per call as we allocate for RTCP traffic. Unlike RTCP traffic, however, we need to allocate our MG-MGC bandwidth for all calls, not just those that happen to have active speech at a given instant.

Next, we allocate bandwidth for MGC-SG signaling. As another rule of thumb, we will allocate the same bandwidth for MGC-SG signaling as we allocate for MG-MGC signaling.

In addition to calculating bandwidth for signaling between SGs and MGCs, we must also calculate the number of SS7 links required between our SGs and the STPs of the networks to which we connect. First, we must remember that our SGs will appear to an external network as an STP pair. For connection to an external network, each of our two SGs will connect to two STPs of the external network. Such an approach will create the standard SS7 quad arrangement described in Chapter 7, "VoIP and SS7" (see Figure 7-4). Second, we will design our SS7 links to run at 40 percent load under normal conditions. By applying such a design rule, we create an environment where we can lose an SS7 link, and a different link can handle the traffic while running at no more than an 80 percent load. Moreover, with a quad arrangement, we can lose a complete STP and still send traffic through the alternative STP of the destination network without overload.

An SS7 link that carries *ISDN User Part* (ISUP) traffic can handle about 30 calls per second. This equates to over 100,000 BHCA. Knowing this number enables us to determine how many SS7 links we will need from our SG pair to each pair of STPs that we connect to.

Note that we do not necessarily need to connect to separate STPs or even have separate SS7 links for each trunk group on our network. We need to have separate SS7 links and connect to separate STPs only for traffic to different networks. For example, the LEC in City 1 and the LEC in City 2 might happen to be the same network. In such a case, we might connect to a single STP pair with SS7 links that carry traffic related to both City 1 and

Table 9-9

VoIP bandwidth required between MGs (Mbps)

	City1	City2	City3	City4	City5	City6	City7	City8	City9	City10	City11
City1	—	—	—	—	—	—	—	—	—	—	—
City2	3.96	—	—	—	—	—	—	—	—	—	—
City3	3.62	2.24	—	—	—	—	—	—	—	—	—
City4	3.05	1.31	1.37	—	—	—	—	—	—	—	—
City5	3.86	2.34	2.88	2.10	—	—	—	—	—	—	—
City6	5.00	3.26	2.64	3.90	3.01	—	—	—	—	—	—
City7	3.95	2.71	2.33	2.28	2.61	3.08	—	—	—	—	—
City8	3.45	2.39	2.14	2.11	2.55	3.19	4.79	—	—	—	—
City9	2.43	1.56	1.35	1.40	1.67	2.19	2.76	2.70	—	—	—
City10	4.96	3.34	2.90	2.36	2.88	3.94	3.42	3.35	2.64	—	—
City11	3.79	2.55	2.45	1.85	2.57	3.49	2.84	2.78	2.34	5.48	—
City12	3.51	1.86	1.87	1.62	2.28	3.08	2.30	2.26	2.36	3.81	3.61

City 2. In a real deployment, the exact SS7 connectivity will be agreed between the network operators as part of an interconnect agreement.

We must also allocate bandwidth for MGC-MGC signaling. We again apply the same rule of thumb. For each call that will pass between MGs controlled by different MGCs, we will assume that the MGC-MGC signaling bandwidth is equal to the per-call bandwidth between a single MGC and an MG.

Finally, we need to allocate some bandwidth for OA&M traffic between each network element and the EMS. The amount of bandwidth allocated to such traffic will depend on many factors, including the number and frequency of reports and statistics, the number of and frequency of call data records (for billing purposes), the number of commands issued to network elements, and the number of alarms generated. No general rule exists for determining the bandwidth needed for such functions. For our network, we will assume that each network element will require approximately 128 Kbps of bandwidth for such functions.



Physical Connectivity

The previous sections of this chapter have shown how we calculate the bandwidth requirements between the various network elements and the outside world. We have performed the detailed calculations required to determine the MG-MG bandwidth and we have explained how to calculate other bandwidth requirements. Knowing how much bandwidth will be required between different nodes does not, however, specify how the network will actually be laid out. After all, we are unlikely to build a fully meshed network, because to do so would be wasteful. Therefore, the next step is to determine how we will connect the different cities together to provide the bandwidth we need between various points. Moreover, we should design the connectivity so that each city has an alternative path to every other city. Such a design will ensure that our network does not fail, even if we lose some transport facilities.

If we consider Figure 9-4, we see that City 2 (MGC1 location) forms a natural hubbing location for the eastern part of our network. Similarly, City 9 (MGC2 location) forms a natural hubbing point for the western part of our network. As a first step, we could assume a network layout where all traffic to and from Cities 1 through 6 passes through City 2 and all traffic to and from Cities 7 through 12 passes through City 9. With a connection

between City 2 and City 9, we would have a network layout as shown in Figure 9-5.

Unfortunately, such a network design is not sufficiently reliable. For example, a break in the connection between City 2 and City 9 would disable all connections between the east and west sides of the network. We need to have two paths from each location to each other location. Such an approach will mean that the failure of any given transport facility or the complete loss of a location (such as City 2) will not jeopardize a great portion of the network. Figure 9-6 is one way we could provide the network redundancy and diversity we need.

Finally, we must consider the connectivity within a given location. After all, there is no point in having redundancy and diversity across the network if we have single points of failure within a given location. In general, our VoIP network elements will have *Fast Ethernet* (FE) or *Gigabit Ethernet* (Gig-E) interfaces. In order to provide sufficient diversity, the FE or Gig-E interfaces from our network element should be connected to a pair of Ethernet switches. These Ethernet switches should then be connected to a pair of routers, as shown in Figure 9-7.

Figure 9-5
Minimal network
connectivity

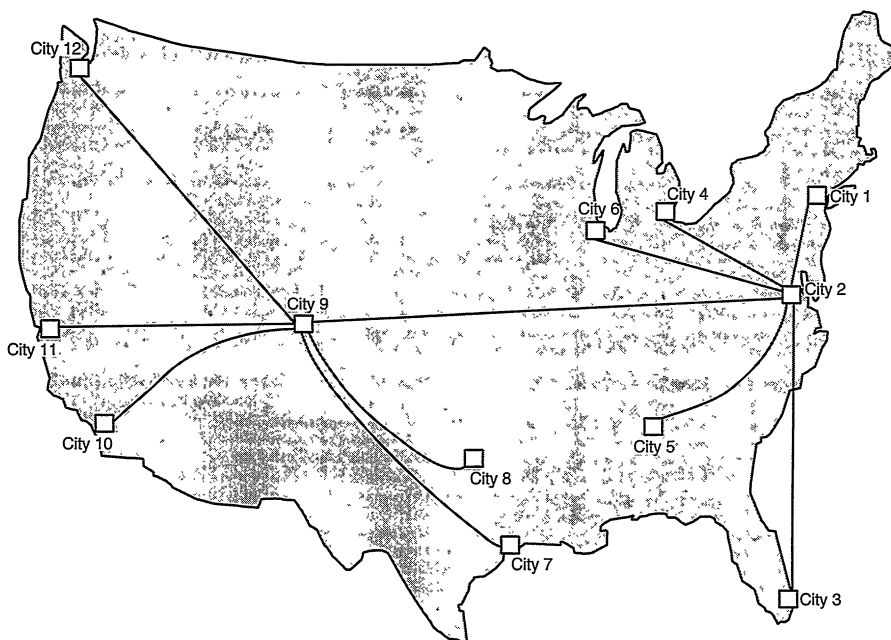


Figure 9-6
Robust network
connectivity

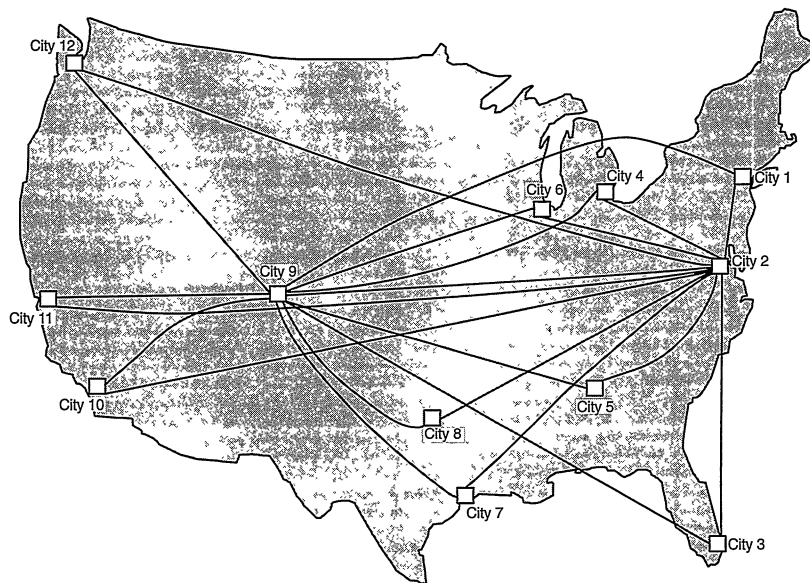


Figure 9-7
Fault-tolerant local
connections

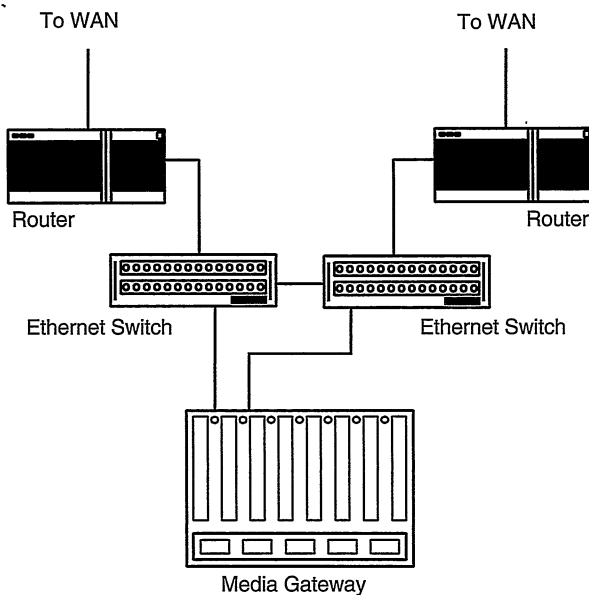


Figure 9-7 shows a fault-tolerant arrangement for connecting an MG to the IP network. A similar arrangement should apply to other network elements such as an MGC, SG or EMS.

APPENDIX A

Table of Erlang B

The following table specifies the number of channels required for a given blocking probability and traffic demand.

In order to determine the required number of channels for a given blocking probability, select the column corresponding to the blocking probability in question. Within that column, find the Erlang value that provides the closest match to the required Erlang value. The leftmost column of the same row provides the required number of channels.

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
2	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3	0.4
3	0.1	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.5	0.6	0.7	0.8	0.9
4	0.2	0.4	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.8	0.8	0.9	1.1	1.3	1.4	1.5
5	0.5	0.6	0.8	0.9	1.0	1.1	1.1	1.2	1.2	1.3	1.3	1.4	1.7	1.9	2.1	2.2
6	0.7	1.0	1.1	1.3	1.4	1.5	1.6	1.7	1.8	1.8	1.9	1.9	2.3	2.5	2.8	3.0
7	1.1	1.4	1.6	1.8	1.9	2.1	2.2	2.2	2.3	2.4	2.4	2.5	2.9	3.2	3.5	3.7
8	1.4	1.8	2.1	2.3	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.1	3.6	4.0	4.3	4.5
9	1.8	2.3	2.6	2.9	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	4.3	4.7	5.1	5.4
10	2.3	2.8	3.1	3.4	3.6	3.8	4.0	4.1	4.2	4.3	4.4	4.5	5.1	5.5	5.9	6.2
11	2.7	3.3	3.7	4.0	4.3	4.5	4.6	4.7	4.9	5.0	5.1	5.2	5.8	6.3	6.7	7.1
12	3.2	3.9	4.2	4.6	4.9	5.1	5.3	5.4	5.6	5.7	5.8	5.9	6.6	7.1	7.6	7.9
13	3.7	4.4	4.8	5.3	5.6	5.8	6.0	6.1	6.3	6.4	6.5	6.6	7.4	8.0	8.4	8.8
14	4.2	5.0	5.4	5.9	6.2	6.5	6.7	6.8	7.0	7.1	7.2	7.4	8.2	8.8	9.3	9.7
15	4.8	5.6	6.1	6.6	6.9	7.2	7.4	7.6	7.7	7.9	8.0	8.1	9.0	9.6	10.2	10.6
16	5.3	6.2	6.7	7.3	7.6	7.9	8.1	8.3	8.5	8.6	8.7	8.9	9.8	10.5	11.1	11.5
17	5.9	6.9	7.4	7.9	8.3	8.6	8.8	9.0	9.2	9.4	9.5	9.7	10.7	11.4	12.0	12.5
18	6.5	7.5	8.0	8.6	9.0	9.3	9.6	9.8	10.0	10.1	10.3	10.4	11.5	12.2	12.9	13.4
19	7.1	8.2	8.7	9.4	9.8	10.1	10.3	10.6	10.7	10.9	11.1	11.2	12.3	13.1	13.8	14.3
20	7.7	8.8	9.4	10.1	10.5	10.8	11.1	11.3	11.5	11.7	11.9	12.0	13.2	14.0	14.7	15.2
21	8.3	9.5	10.1	10.8	11.2	11.6	11.9	12.1	12.3	12.5	12.7	12.8	14.0	14.9	15.6	16.2
22	8.9	10.2	10.8	11.5	12.0	12.3	12.6	12.9	13.1	13.3	13.5	13.7	14.9	15.8	16.5	17.1
23	9.6	10.9	11.5	12.3	12.7	13.1	13.4	13.7	13.9	14.1	14.3	14.5	15.8	16.7	17.4	18.1
24	10.2	11.6	12.2	13.0	13.5	13.9	14.2	14.5	14.7	14.9	15.1	15.3	16.6	17.6	18.4	19.0
25	10.9	12.3	13.0	13.8	14.3	14.7	15.0	15.3	15.5	15.7	15.9	16.1	17.5	18.5	19.3	20.0
26	11.5	13.0	13.7	14.5	15.1	15.5	15.8	16.1	16.3	16.6	16.8	17.0	18.4	19.4	20.2	20.9
27	12.2	13.7	14.4	15.3	15.8	16.3	16.6	16.9	17.2	17.4	17.6	17.8	19.3	20.3	21.2	21.9
28	12.9	14.4	15.2	16.1	16.6	17.1	17.4	17.7	18.0	18.2	18.4	18.6	20.1	21.2	22.1	22.9
29	13.6	15.1	15.9	16.8	17.4	17.9	18.2	18.5	18.8	19.1	19.3	19.5	21.0	22.1	23.0	23.8
30	14.2	15.9	16.7	17.6	18.2	18.7	19.0	19.4	19.6	19.9	20.1	20.3	21.9	23.1	24.0	24.8

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
31	14.9	16.6	17.4	18.4	19.0	19.5	19.9	20.2	20.5	20.7	21.0	21.2	22.8	24.0	24.9	25.8
32	15.6	17.3	18.2	19.2	19.8	20.3	20.7	21.0	21.3	21.6	21.8	22.0	23.7	24.9	25.9	26.7
33	16.3	18.1	19.0	20.0	20.6	21.1	21.5	21.9	22.2	22.4	22.7	22.9	24.6	25.8	26.8	27.7
34	17.0	18.8	19.7	20.8	21.4	21.9	22.3	22.7	23.0	23.3	23.5	23.8	25.5	26.8	27.8	28.7
35	17.3	19.6	20.5	21.6	22.2	22.7	23.2	23.5	23.8	24.1	24.4	24.6	26.4	27.7	28.8	29.7
36	18.5	20.3	21.3	22.4	23.0	23.6	24.0	24.4	24.7	25.0	25.3	25.5	27.3	28.6	29.7	30.7
37	19.2	21.1	22.1	23.2	23.9	24.4	24.8	25.2	25.6	25.9	26.1	26.4	28.3	29.6	30.7	31.6
38	19.9	21.9	22.9	24.0	24.7	25.2	25.7	26.1	26.4	26.7	27.0	27.3	29.2	30.5	31.6	32.6
39	20.6	22.6	23.7	24.8	25.5	26.1	26.5	26.9	27.3	27.6	27.9	28.1	30.1	31.5	32.6	33.6
40	21.4	23.4	24.4	25.6	26.3	26.9	27.4	27.8	28.1	28.5	28.7	29.0	31.0	32.4	33.6	34.6
41	22.1	24.2	25.2	26.4	27.2	27.8	28.2	28.6	29.0	29.3	29.6	29.9	31.9	33.4	34.5	35.6
42	22.8	25.0	26.0	27.2	28.0	28.6	29.1	29.5	29.9	30.2	30.5	30.8	32.8	34.3	35.5	36.6
43	23.6	25.7	26.8	28.1	28.8	29.4	29.9	30.4	30.7	31.1	31.4	31.7	33.8	35.3	36.5	37.6
44	24.3	26.5	27.6	28.9	29.7	30.3	30.8	31.2	31.6	31.9	32.3	32.5	34.7	36.2	37.5	38.6
45	25.1	27.3	28.4	29.7	30.5	31.1	31.7	32.1	32.5	32.8	33.1	33.4	35.6	37.2	38.4	39.6
46	25.8	28.1	29.3	30.5	31.4	32.0	32.5	33.0	33.4	33.7	34.0	34.3	36.5	38.1	39.4	40.5
47	26.6	28.9	30.1	31.4	32.2	32.9	33.4	33.8	34.2	34.6	34.9	35.2	37.5	39.1	40.4	41.5
48	27.3	29.7	30.9	32.2	33.1	33.7	34.2	34.7	35.1	35.5	35.8	36.1	38.4	40.0	41.4	42.5
49	28.1	30.5	31.7	33.0	33.9	34.6	35.1	35.6	36.0	36.4	36.7	37.0	39.3	41.0	42.3	43.5
50	28.9	31.3	32.5	33.9	34.8	35.4	36.0	36.5	36.9	37.2	37.6	37.9	40.3	41.9	43.3	44.5
51	29.6	32.1	33.3	34.7	35.6	36.3	36.9	37.3	37.8	38.1	38.5	38.8	41.2	42.9	44.3	45.5
52	30.4	32.9	34.2	35.6	36.5	37.2	37.7	38.2	38.6	39.0	39.4	39.7	42.1	43.9	45.3	46.5
53	31.2	33.7	35.0	36.4	37.3	38.0	38.6	39.1	39.5	39.9	40.3	40.6	43.1	44.8	46.3	47.5
54	31.9	34.5	35.8	37.2	38.2	38.9	39.5	40.0	40.4	40.8	41.2	41.5	44.0	45.8	47.2	48.5
55	32.7	35.3	36.6	38.1	39.0	39.8	40.4	40.9	41.3	41.7	42.1	42.4	44.9	46.7	48.2	49.5
56	33.5	36.1	37.5	38.9	39.9	40.6	41.2	41.7	42.2	42.6	43.0	43.3	45.9	47.7	49.2	50.5
57	34.3	36.9	38.3	39.8	40.8	41.5	42.1	42.6	43.1	43.5	43.9	44.2	46.8	48.7	50.2	51.5
58	35.1	37.8	39.1	40.6	41.6	42.4	43.0	43.5	44.0	44.4	44.8	45.1	47.8	49.6	51.2	52.6
59	35.8	38.6	40.0	41.5	42.5	43.3	43.9	44.4	44.9	45.3	45.7	46.0	48.7	50.6	52.2	53.6
60	36.6	39.4	40.8	42.4	43.4	44.1	44.8	45.3	45.8	46.2	46.6	46.9	49.6	51.6	53.2	54.6

channels	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
61	37.4	40.2	41.6	43.2	44.2	45.0	45.6	46.2	46.7	47.1	47.5	47.9	50.6	52.5	54.2	55.6
62	38.2	41.0	42.5	44.1	45.1	45.9	46.5	47.1	47.6	48.0	48.4	48.8	51.5	53.5	55.1	56.6
63	39.0	41.9	43.3	44.9	46.0	46.8	47.4	48.0	48.5	48.9	49.3	49.7	52.5	54.5	56.1	57.6
64	39.8	42.7	44.2	45.8	46.8	47.6	48.3	48.9	49.4	49.8	50.2	50.6	53.4	55.4	57.1	58.6
65	40.6	43.5	45.0	46.6	47.7	48.5	49.2	49.8	50.3	50.7	51.1	51.5	54.4	56.4	58.1	59.6
66	41.4	44.4	45.8	47.5	48.6	49.4	50.1	50.7	51.2	51.6	52.0	52.4	55.3	57.4	59.1	60.6
67	42.2	45.2	46.7	48.4	49.5	50.3	51.0	51.6	52.1	52.5	53.0	53.4	56.3	58.4	60.1	61.6
68	43.0	46.0	47.5	49.2	50.3	51.2	51.9	52.5	53.0	53.4	53.9	54.3	57.2	59.3	61.1	62.6
69	43.8	46.8	48.4	50.1	51.2	52.1	52.8	53.4	53.9	54.4	54.8	55.2	58.2	60.3	62.1	63.7
70	44.6	47.7	49.2	51.0	52.1	53.0	53.7	54.3	54.8	55.3	55.7	56.1	59.1	61.3	63.1	64.7
71	45.4	48.5	50.1	51.8	53.0	53.8	54.6	55.2	55.7	56.2	56.6	57.0	60.1	62.3	64.1	65.7
72	46.2	49.4	50.9	52.7	53.9	54.7	55.5	56.1	56.6	57.1	57.5	58.0	61.0	63.2	65.1	66.7
73	47.0	50.2	51.8	53.6	54.7	55.6	56.4	57.0	57.5	58.0	58.5	58.9	62.0	64.2	66.1	67.7
74	47.8	51.0	52.7	54.5	55.6	56.5	57.3	57.9	58.4	58.9	59.4	59.8	62.9	65.2	67.1	68.7
75	48.6	51.9	53.5	55.3	56.5	57.4	58.2	58.8	59.3	59.8	60.3	60.7	63.9	66.2	68.1	69.7
76	49.4	52.7	54.4	56.2	57.4	58.3	59.1	59.7	60.3	60.8	61.2	61.7	64.9	67.2	69.1	70.8
77	50.2	53.6	55.2	57.1	58.3	59.2	60.0	60.6	61.2	61.7	62.1	62.6	65.8	68.1	70.1	71.8
78	51.1	54.4	56.1	58.0	59.2	60.1	60.9	61.5	62.1	62.6	63.1	63.5	66.8	69.1	71.1	72.8
79	51.9	55.3	56.9	58.8	60.1	61.0	61.8	62.4	63.0	63.5	64.0	64.4	67.7	70.1	72.1	73.8
80	52.7	56.1	57.8	59.7	61.0	61.9	62.7	63.3	63.9	64.4	64.9	65.4	68.7	71.1	73.1	74.8
81	53.5	56.9	58.7	60.6	61.8	62.8	63.6	64.2	64.8	65.4	65.8	66.3	69.6	72.1	74.1	75.8
82	54.3	57.8	59.5	61.5	62.7	63.7	64.5	65.2	65.7	66.3	66.8	67.2	70.6	73.0	75.1	76.9
83	55.1	58.6	60.4	62.4	63.6	64.6	65.4	66.1	66.7	67.2	67.7	68.2	71.6	74.0	76.1	77.9
84	56.0	59.5	61.3	63.2	64.5	65.5	66.3	67.0	67.6	68.1	68.6	69.1	72.5	75.0	77.1	78.9
85	56.8	60.4	62.1	64.1	65.4	66.4	67.2	67.9	68.5	69.1	69.6	70.0	73.5	76.0	78.1	79.9
86	57.6	61.2	63.0	65.0	66.3	67.3	68.1	68.8	69.4	70.0	70.5	70.9	74.5	77.0	79.1	80.9
87	58.4	62.1	63.9	65.9	67.2	68.2	69.0	69.7	70.3	70.9	71.4	71.9	75.4	78.0	80.1	82.0
88	59.3	62.9	64.7	66.8	68.1	69.1	69.9	70.6	71.3	71.8	72.3	72.8	76.4	78.9	81.1	83.0
89	60.1	63.8	65.6	67.7	69.0	70.0	70.8	71.6	72.2	72.8	73.3	73.7	77.3	79.9	82.1	84.0
90	60.9	64.6	66.5	68.6	69.9	70.9	71.8	72.5	73.1	73.7	74.2	74.7	78.3	80.9	83.1	85.0

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
91	61.8	65.5	67.4	69.4	70.8	71.8	72.7	73.4	74.0	74.6	75.1	75.6	79.3	81.9	84.1	86.0
92	62.6	66.3	68.2	70.3	71.7	72.7	73.6	74.3	75.0	75.5	76.1	76.6	80.2	82.9	85.1	87.1
93	63.4	67.2	69.1	71.2	72.6	73.6	74.5	75.2	75.9	76.5	77.0	77.5	81.2	83.9	86.1	88.1
94	64.2	68.1	70.0	72.1	73.5	74.5	75.4	76.2	76.8	77.4	77.9	78.4	82.2	84.9	87.1	89.1
95	65.1	68.9	70.9	73.0	74.4	75.5	76.3	77.1	77.7	78.3	78.9	79.4	83.1	85.8	88.1	90.1
96	65.9	69.8	71.7	73.9	75.3	76.4	77.2	78.0	78.7	79.3	79.8	80.3	84.1	86.8	89.1	91.1
97	66.8	70.7	72.6	74.8	76.2	77.3	78.2	78.9	79.6	80.2	80.7	81.2	85.1	87.8	90.1	92.2
98	67.6	71.5	73.5	75.7	77.1	78.2	79.1	79.8	80.5	81.1	81.7	82.2	86.0	88.8	91.1	93.2
99	68.4	72.4	74.4	76.6	78.0	79.1	80.0	80.8	81.4	82.0	82.6	83.1	87.0	89.8	92.1	94.2
100	69.3	73.2	75.2	77.5	78.9	80.0	80.9	81.7	82.4	83.0	83.5	84.1	88.0	90.8	93.1	95.2
102	70.9	75.0	77.0	79.3	80.7	81.8	82.7	83.5	84.2	84.8	85.4	85.9	89.9	92.8	95.2	97.3
104	72.6	76.7	78.8	81.1	82.5	83.7	84.6	85.4	86.1	86.7	87.3	87.8	91.8	94.8	97.2	99.3
106	74.3	78.5	80.5	82.8	84.3	85.5	86.4	87.2	87.9	88.6	89.2	89.7	93.8	96.7	99.2	101.4
108	76.0	80.2	82.3	84.6	86.2	87.3	88.3	89.1	89.8	90.5	91.1	91.6	95.7	98.7	101.2	103.4
110	77.7	81.9	84.1	86.4	88.0	89.2	90.1	90.9	91.7	92.3	92.9	93.5	97.7	100.7	103.2	105.5
112	79.4	83.7	85.8	88.3	89.8	91.0	92.0	92.8	93.5	94.2	94.8	95.4	99.6	102.7	105.3	107.5
114	81.1	85.4	87.6	90.1	91.6	92.8	93.8	94.7	95.4	96.1	96.7	97.3	101.6	104.7	107.3	109.6
116	82.8	87.2	89.4	91.9	93.5	94.7	95.7	96.5	97.3	98.0	98.6	99.2	103.5	106.7	109.3	111.7
118	84.5	89.0	91.2	93.7	95.3	96.5	97.5	98.4	99.2	99.9	100.5	101.1	105.5	108.7	111.3	113.7
120	86.2	90.7	93.0	95.5	97.1	98.4	99.4	100.3	101.0	101.7	102.4	103.0	107.4	110.6	113.4	115.8
122	87.9	92.5	94.7	97.3	98.9	100.2	101.2	102.1	102.9	103.6	104.3	104.9	109.4	112.6	115.4	117.8
124	89.6	94.2	96.5	99.1	100.8	102.1	103.1	104.0	104.8	105.5	106.2	106.8	111.3	114.6	117.4	119.9
126	91.3	96.0	98.3	100.9	102.6	103.9	105.0	105.9	106.7	107.4	108.1	108.7	113.3	116.6	119.4	121.9
128	93.1	97.8	100.1	102.7	104.5	105.8	106.8	107.7	108.5	109.3	109.9	110.6	115.2	118.6	121.5	124.0
130	94.8	99.5	101.9	104.6	106.3	107.6	108.7	109.6	110.4	111.2	111.8	112.5	117.2	120.6	123.5	126.1
132	96.5	101.3	103.7	106.4	108.1	109.5	110.5	111.5	112.3	113.1	113.7	114.4	119.1	123.6	125.5	128.1
134	98.2	103.1	105.5	108.2	110.0	111.3	112.4	113.4	114.2	115.0	115.6	116.3	121.1	124.6	127.6	130.2
136	100.0	104.9	107.3	110.0	111.8	113.2	114.3	115.2	116.1	116.8	117.5	118.2	123.1	126.6	129.6	132.3
138	101.7	106.6	109.1	111.9	113.7	115.0	116.2	117.1	118.0	118.7	119.4	120.1	125.0	128.6	131.6	134.3
140	103.4	108.4	110.9	113.7	115.5	116.9	118.0	119.0	119.9	120.6	121.4	122.0	127.0	130.6	133.7	136.4

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
142	105.1	110.2	112.7	115.5	117.4	118.7	119.9	120.9	121.8	122.5	123.3	123.9	128.9	132.6	135.7	138.4
144	106.9	112.0	114.5	117.4	119.2	120.6	121.8	122.8	123.6	124.4	125.2	125.8	130.9	134.6	137.7	140.5
146	108.6	113.8	116.3	119.2	121.1	122.5	123.6	124.6	125.5	126.3	127.1	127.7	132.9	136.6	139.8	142.6
148	110.4	115.5	118.1	121.0	122.9	124.3	125.5	126.5	127.4	128.2	129.0	129.7	134.8	138.6	141.8	144.6
150	112.1	117.3	119.9	122.9	124.8	126.2	127.4	128.4	129.3	130.1	130.9	131.6	136.8	140.6	143.8	146.7
152	113.8	119.1	121.8	124.7	126.6	128.1	129.3	130.3	131.2	132.0	132.8	133.5	138.8	142.6	145.9	148.8
154	115.6	120.9	123.6	126.5	128.5	129.9	131.2	132.2	133.1	133.9	134.7	135.4	140.7	144.6	147.9	150.8
156	117.3	122.7	125.4	128.4	130.3	131.8	133.0	134.1	135.0	135.9	136.6	137.3	142.7	146.6	149.9	152.9
158	119.1	124.5	127.2	130.2	132.2	133.7	134.9	136.0	136.9	137.8	138.5	139.2	144.7	148.6	152.0	155.0
160	120.8	126.3	129.0	132.1	134.0	135.6	136.8	137.9	138.8	139.7	140.4	141.2	146.6	150.6	154.0	157.0
162	122.6	128.1	130.8	133.9	135.9	137.4	138.7	139.8	140.7	141.6	142.4	143.1	148.6	152.7	156.1	159.1
164	124.3	129.9	132.7	135.8	137.8	139.3	140.6	141.7	142.6	143.5	144.3	145.0	150.6	154.7	158.1	161.2
166	126.1	131.7	134.5	137.6	139.6	141.2	142.5	143.5	144.5	145.4	146.2	146.9	152.6	156.7	160.1	163.3
168	127.9	133.5	136.3	139.4	141.5	143.1	144.3	145.4	146.4	147.3	148.1	148.9	154.5	158.7	162.2	165.3
170	129.6	135.3	138.1	141.3	143.4	144.9	146.2	147.3	148.3	149.2	150.0	150.8	156.5	160.7	164.2	167.4
172	131.4	137.1	139.9	143.1	145.2	146.8	148.1	149.2	150.2	151.1	151.9	152.7	158.5	162.7	166.3	169.5
174	133.1	138.9	141.8	145.0	147.1	148.7	150.0	151.1	152.1	153.0	153.9	154.6	160.4	164.7	168.3	171.5
176	134.9	140.7	143.6	146.9	149.0	150.6	151.9	153.0	154.0	155.0	155.8	156.6	162.4	166.7	170.3	173.6
178	136.7	142.5	145.4	148.7	150.8	152.4	153.8	154.9	156.0	156.9	157.7	158.5	164.4	168.7	172.4	175.7
180	138.4	144.3	147.3	150.6	152.7	154.3	155.7	156.8	157.9	158.8	159.6	160.4	166.4	170.7	174.4	177.8
182	140.2	146.1	149.1	152.4	154.6	156.2	157.6	158.7	159.8	160.7	161.6	162.3	168.3	172.8	176.5	179.8
184	142.0	147.9	150.9	154.3	156.4	158.1	159.5	160.6	161.7	162.6	163.5	164.3	170.3	174.8	178.5	181.9
186	143.7	149.8	152.8	156.1	158.3	160.0	161.4	162.5	163.6	164.5	165.4	166.2	172.3	176.8	180.6	184.0
188	145.5	151.6	154.6	158.0	160.2	161.9	163.3	164.4	165.5	166.5	167.3	168.1	174.3	178.8	182.6	186.1
190	147.3	153.4	156.4	159.8	162.1	163.8	165.2	166.4	167.4	168.4	169.3	170.1	176.3	180.8	184.7	188.1
192	149.1	155.2	158.3	161.7	163.9	165.6	167.0	168.3	169.3	170.3	171.2	172.0	178.2	182.8	186.7	190.2
194	150.8	157.0	160.1	163.6	165.8	167.5	168.9	170.2	171.2	172.2	173.1	173.9	180.2	184.8	188.8	192.3
196	152.6	158.8	161.9	165.4	167.7	169.4	170.8	172.1	173.2	174.1	175.0	175.9	182.2	186.9	190.8	194.4
198	154.4	160.7	163.8	167.3	169.6	171.3	172.7	174.0	175.1	176.1	177.0	177.8	184.2	188.9	192.8	196.4
200	156.2	162.5	165.6	169.2	171.4	173.2	174.6	175.9	177.0	178.0	178.9	179.7	186.2	190.9	194.9	198.5

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
205	160.6	167.0	170.2	173.8	176.1	177.9	179.4	180.7	181.8	182.8	183.7	184.6	191.1	195.9	200.0	203.7
210	165.1	171.6	174.8	178.5	180.9	182.7	184.2	185.4	186.6	187.6	188.6	189.4	196.1	201.0	205.1	208.9
215	169.6	176.2	179.5	183.2	185.6	187.4	188.9	190.2	191.4	192.4	193.4	194.3	201.0	206.0	210.3	214.1
220	174.0	180.7	184.1	187.8	190.3	192.1	193.7	195.0	196.2	197.2	198.2	199.1	206.0	211.1	215.4	219.3
225	178.5	185.3	188.7	192.5	195.0	196.9	198.5	199.8	201.0	202.1	203.1	204.0	211.0	216.1	220.5	224.5
230	183.0	189.9	193.3	197.2	199.7	201.6	203.2	204.6	205.8	206.9	207.9	208.8	215.9	221.2	225.6	229.7
235	187.5	194.5	198.0	201.9	204.4	206.4	208.0	209.4	210.6	211.7	212.8	213.7	220.9	226.2	230.8	234.9
240	192.0	199.1	202.6	206.6	209.2	211.2	212.8	214.2	215.4	216.6	217.6	218.6	225.9	231.3	235.9	240.1
245	196.5	203.7	207.3	211.3	213.9	215.9	217.6	219.0	220.3	221.4	222.5	223.4	230.8	236.4	241.0	245.3
250	201.0	208.3	211.9	216.0	218.7	220.7	222.4	223.8	225.1	226.2	227.3	228.3	235.8	241.4	246.2	250.5
255	205.6	212.9	216.6	220.7	223.4	225.5	227.2	228.6	229.9	231.1	232.2	233.2	240.8	246.5	251.3	255.7
260	210.1	217.5	221.2	225.4	228.1	230.2	232.0	233.4	234.7	235.9	237.0	238.0	245.8	251.5	256.5	260.9
265	214.6	222.1	225.9	230.1	232.9	235.0	236.7	238.3	239.6	240.8	241.9	242.9	250.8	256.6	261.6	266.1
270	219.1	226.8	230.6	234.9	237.6	239.8	241.5	243.1	244.4	245.6	246.8	247.8	255.8	261.7	266.7	271.3
275	223.7	231.4	235.2	239.6	242.4	244.6	246.4	247.9	249.3	250.5	251.6	252.7	260.7	266.7	271.9	276.5
280	228.2	236.0	239.9	244.3	247.2	249.3	251.2	252.7	254.1	255.4	256.5	257.6	265.7	271.8	277.0	281.8
285	232.8	240.6	244.6	249.0	251.9	254.1	256.0	257.5	258.9	260.2	261.4	262.5	270.7	276.9	282.2	287.0
290	237.3	245.3	249.3	253.8	256.7	258.9	260.8	262.4	263.8	265.1	266.2	267.3	275.7	282.0	287.3	292.2
295	241.9	249.9	254.0	258.5	261.5	263.7	265.6	267.2	268.6	269.9	271.1	272.2	280.7	287.0	292.5	297.4
300	246.4	254.6	258.6	263.2	266.2	268.5	270.4	272.0	273.5	274.8	276.0	277.1	285.7	292.1	297.6	302.6
305	251.0	259.2	263.3	268.0	271.0	273.3	275.2	276.9	278.3	279.7	280.9	282.0	290.7	297.2	302.8	307.8
310	255.6	263.9	268.0	272.7	275.8	278.1	280.0	281.7	283.2	284.5	285.8	286.9	295.7	302.3	307.9	313.1
315	260.1	268.5	272.7	277.5	280.5	282.9	284.9	286.6	288.1	289.4	290.7	291.8	300.7	307.4	313.1	318.3
320	264.7	273.2	277.4	282.2	285.3	287.7	289.7	291.4	292.9	294.3	295.5	296.7	305.7	312.4	318.2	323.5
325	269.3	277.8	282.1	287.0	290.1	292.5	294.5	296.2	297.8	299.2	300.4	301.6	310.7	317.5	323.4	328.7
330	273.9	282.5	286.8	291.7	294.9	297.3	299.4	301.1	302.6	304.0	305.3	306.5	315.7	322.6	328.5	333.9
335	278.5	287.2	291.6	296.5	299.7	302.1	304.2	305.9	307.5	308.9	310.2	311.4	320.7	327.7	333.7	339.2
340	283.1	291.8	296.3	301.2	304.5	307.0	309.0	310.8	312.4	313.8	315.1	316.3	325.7	332.8	338.8	344.4
345	287.7	296.5	301.0	306.0	309.3	311.8	313.9	315.6	317.2	318.7	320.0	321.2	330.7	337.9	344.0	349.6
350	292.3	301.2	305.7	310.8	314.1	316.6	318.7	320.5	322.1	323.6	324.9	326.2	335.7	342.9	349.1	354.8

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
355	296.9	305.9	310.4	315.5	318.9	321.4	323.5	325.4	327.0	328.5	329.8	331.1	340.7	348.0	354.3	360.0
360	301.5	310.6	315.1	320.3	323.7	326.2	328.4	330.2	331.9	333.3	334.7	336.0	345.8	353.1	359.5	365.3
365	306.1	315.2	319.9	325.1	328.4	331.1	333.2	335.1	336.7	338.2	339.6	340.9	350.8	358.2	364.6	370.5
370	310.7	319.9	324.6	329.8	333.3	335.9	338.1	339.9	341.6	343.1	344.5	345.8	355.8	363.3	369.8	375.7
375	315.3	324.6	329.3	334.6	338.1	340.7	342.9	344.8	346.5	348.0	349.4	350.7	360.8	368.4	374.9	380.9
380	319.9	329.3	334.0	339.4	342.9	345.5	347.8	349.7	351.4	352.9	354.3	355.6	365.8	373.5	380.1	386.2
385	324.5	334.0	338.8	344.2	347.7	350.4	352.6	354.5	356.2	357.8	359.2	360.6	370.8	378.6	385.3	391.4
390	329.1	338.7	343.5	348.9	352.5	355.2	357.5	359.4	361.1	362.7	364.1	365.5	375.8	383.7	390.4	396.6
395	333.8	343.4	348.3	353.7	357.3	360.0	362.3	364.3	366.0	367.6	369.1	370.4	380.9	388.8	395.6	401.9
400	338.4	348.1	353.0	358.5	362.1	364.9	367.2	369.1	370.9	372.5	374.0	375.3	385.9	393.9	400.8	407.1
410	347.7	357.5	362.5	368.1	371.7	374.5	376.9	378.9	380.7	382.3	383.8	385.2	395.9	404.1	411.1	417.5
420	356.9	366.9	372.0	377.7	381.4	384.2	386.6	388.6	390.5	392.1	393.6	395.0	406.0	414.2	421.4	428.0
430	366.2	376.4	381.5	387.2	391.0	393.9	396.3	398.4	400.2	401.9	403.5	404.9	416.0	424.4	431.8	438.5
440	375.5	385.8	391.0	396.8	400.7	403.6	406.0	408.2	410.0	411.7	413.3	414.8	426.1	434.6	442.1	448.9
450	384.8	395.2	400.5	406.4	410.3	413.3	415.8	417.9	419.8	421.6	423.2	424.6	436.1	444.8	452.4	459.4
460	394.1	404.7	410.0	416.0	420.0	423.0	425.5	427.7	429.6	431.4	433.0	434.5	446.2	455.1	462.8	469.9
470	403.4	414.1	419.5	425.6	429.6	432.7	435.3	437.5	439.4	441.2	442.9	444.4	456.2	465.3	473.1	480.3
480	412.7	423.6	429.1	435.3	439.3	442.4	445.0	447.2	449.2	451.0	452.7	454.3	466.3	475.5	483.5	490.8
490	422.1	433.1	438.6	444.9	449.0	452.1	454.8	457.0	459.1	460.9	462.6	464.1	476.4	485.7	493.8	501.3
500	431.4	442.5	448.2	454.5	458.7	461.9	464.5	466.8	468.9	470.7	472.4	474.0	486.4	495.9	504.1	511.8
510	440.8	452.0	457.7	464.1	468.3	471.6	474.3	476.6	478.7	480.6	482.3	483.9	496.5	506.1	514.5	522.2
520	450.1	461.5	467.3	473.8	478.0	481.3	484.0	486.4	488.5	490.4	492.2	493.8	506.6	516.3	524.8	532.7
530	459.5	471.0	476.8	483.4	487.7	491.0	493.8	496.2	498.3	500.3	502.1	503.7	516.7	526.6	535.2	543.2
540	468.8	480.5	486.4	493.1	497.4	500.8	503.6	506.0	508.2	510.1	511.9	513.6	526.7	536.8	545.5	553.7
550	478.2	490.0	496.0	502.7	507.1	510.5	513.4	515.8	518.0	520.0	521.8	523.5	536.8	547.0	555.9	564.1
560	487.6	499.5	505.5	512.4	516.8	520.3	523.1	525.6	527.8	529.8	531.7	533.4	546.9	557.2	566.3	574.6
570	497.0	509.0	515.1	522.0	526.5	530.0	532.9	535.4	537.7	539.7	541.6	543.3	557.0	567.4	576.6	585.1
580	506.4	518.6	524.7	531.7	536.2	539.8	542.7	545.3	547.5	549.6	551.5	553.2	567.1	577.7	587.0	595.6
590	515.8	528.1	534.3	541.3	546.0	549.5	552.5	555.1	557.4	559.4	561.4	563.2	577.2	587.9	597.3	606.1
600	525.2	537.6	543.9	551.0	555.7	559.3	562.3	564.9	567.2	569.3	571.3	573.1	587.2	598.1	607.7	616.5

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%

610	534.6	547.1	553.5	560.7	565.4	569.1	572.1	574.7	577.1	579.2	581.2	583.0	597.3	608.4	618.0	627.0
620	544.0	556.7	563.1	570.4	575.1	578.8	581.9	584.5	586.9	589.1	591.1	592.9	607.4	618.6	628.4	637.5
630	553.4	566.2	572.7	580.0	584.9	588.6	591.7	594.4	596.8	599.0	601.0	602.8	617.5	628.8	638.8	648.0
640	562.8	575.8	582.3	589.7	594.6	598.4	601.5	604.2	606.6	608.8	610.9	612.8	627.6	639.1	649.1	658.5
650	572.3	585.3	591.9	599.4	604.3	608.1	611.3	614.0	616.5	618.7	620.8	622.7	637.7	649.3	659.5	669.0
660	581.7	594.9	601.6	609.1	614.1	617.9	621.1	623.9	626.4	628.6	630.7	632.6	647.8	659.5	669.9	679.5
670	591.2	604.5	611.2	618.8	623.8	627.7	630.9	633.7	636.2	638.5	640.6	642.6	657.9	669.8	680.2	690.0
680	600.6	614.0	620.8	628.5	633.6	637.5	640.7	643.6	646.1	648.4	650.5	652.5	668.0	680.0	690.6	700.4
690	610.1	623.6	630.4	638.2	643.3	647.3	650.6	653.4	656.0	658.3	660.4	662.4	678.1	690.3	701.0	710.9
700	619.5	633.2	640.1	647.9	653.1	657.1	660.4	663.3	665.8	668.2	670.3	672.4	688.2	700.5	711.3	721.4
710	629.0	642.7	649.7	657.6	662.8	666.8	670.2	673.1	675.7	678.1	680.3	682.3	698.3	710.7	721.7	731.9
720	638.4	652.3	659.3	667.3	672.6	676.6	680.0	683.0	685.6	688.0	690.2	692.2	708.4	721.0	732.1	742.4
730	647.9	661.9	669.0	677.0	682.3	686.4	689.8	692.8	695.5	697.9	700.1	702.2	718.5	731.2	742.4	752.9
740	657.4	671.5	678.6	686.7	692.1	696.2	699.7	702.7	705.3	707.8	710.0	712.1	728.6	741.5	752.8	763.4
750	666.8	681.1	688.3	696.5	701.9	706.0	709.5	712.5	715.2	717.7	720.0	722.1	738.8	751.7	763.2	773.9
760	676.3	690.7	697.9	706.2	711.6	715.8	719.3	722.4	725.1	727.6	729.9	732.0	748.9	762.0	773.6	784.4
770	685.8	700.3	707.6	715.9	721.4	725.6	729.2	732.3	735.0	737.5	739.8	742.0	759.0	772.2	783.9	794.9
780	695.3	709.9	717.2	725.6	731.2	735.4	739.0	742.1	744.9	747.4	749.7	751.9	769.1	782.5	794.3	805.4
790	704.8	719.5	726.9	735.4	740.9	745.2	748.8	752.0	754.8	757.3	759.7	761.9	779.2	792.7	804.7	815.9
800	714.3	729.1	736.6	745.1	750.7	755.1	758.7	761.8	764.7	767.2	769.6	771.8	789.3	803.0	815.1	826.4
810	723.8	738.7	746.2	754.8	760.5	764.9	768.5	771.7	774.6	777.2	779.6	781.8	799.4	813.2	825.4	836.9
820	733.3	748.3	755.9	764.6	770.3	774.7	778.4	781.6	784.5	787.1	789.5	791.7	809.6	823.5	835.8	847.4
830	742.8	757.9	765.6	774.3	780.0	784.5	788.2	791.5	794.4	797.0	799.4	801.7	819.7	833.7	846.2	857.9
840	752.3	767.5	775.2	784.0	789.8	794.3	798.1	801.3	804.3	806.9	809.4	811.7	829.8	844.0	856.6	868.4
850	761.8	777.1	784.9	793.8	799.6	804.1	807.9	811.2	814.2	816.8	819.3	821.6	839.9	854.2	867.0	878.9
860	771.3	786.8	794.6	803.5	809.4	814.0	817.8	821.1	824.1	826.8	829.3	831.6	850.1	864.5	877.3	889.4
870	780.8	796.4	804.3	813.3	819.2	823.8	827.6	831.0	834.0	836.7	839.2	841.6	860.2	874.8	887.7	899.9
880	790.4	806.0	814.0	823.0	829.0	833.6	837.5	840.9	843.9	846.6	849.1	851.5	870.3	885.0	898.1	910.4
890	799.9	815.6	823.6	832.8	838.8	843.4	847.3	850.7	853.8	856.5	859.1	861.5	880.4	895.3	908.5	920.9
900	809.4	825.3	833.3	842.5	848.6	853.3	857.2	860.6	863.7	866.5	869.0	871.5	890.6	905.5	918.9	931.4

channels	Blocking probability														
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	5.0%
910	818.9	834.9	843.0	852.3	858.4	863.1	867.1	870.5	873.6	876.4	879.0	881.4	900.7	915.8	941.9
920	828.5	844.5	852.7	862.0	868.2	872.9	876.9	880.4	883.5	886.3	888.9	891.4	910.8	926.0	952.4
930	838.0	854.2	862.4	871.8	878.0	882.8	886.8	890.3	893.4	896.3	898.9	901.4	920.9	936.3	962.9
940	847.5	863.8	872.1	881.5	887.8	892.6	896.6	900.2	903.3	906.2	908.9	911.3	931.1	946.6	973.4
950	857.1	873.5	881.8	891.3	897.6	902.4	906.5	910.1	913.2	916.1	918.8	921.3	941.2	956.8	983.9
960	866.6	883.1	891.5	901.1	907.4	912.3	916.4	920.0	923.2	926.1	928.8	931.3	951.3	967.1	994.4
970	876.2	892.8	901.2	910.8	917.2	922.1	926.3	929.8	933.1	936.0	938.7	941.3	961.5	977.3	1004.9
980	885.7	902.4	910.9	920.6	927.0	932.0	936.1	939.7	943.0	945.9	948.7	951.2	971.6	987.6	1015.4
990	895.3	912.1	920.6	930.4	936.8	941.8	946.0	949.6	952.9	955.9	958.6	961.2	981.7	997.9	1025.9
1000	904.8	921.7	930.3	940.1	946.6	951.7	955.9	959.5	962.8	965.8	968.6	971.2	991.9	1008.1	1036.4
1010	914.4	931.4	940.0	949.9	956.4	961.5	965.7	969.4	972.7	975.8	978.6	981.2	1002.0	1018.4	1046.9
1020	923.9	941.0	949.7	959.7	966.2	971.3	975.6	979.3	982.7	985.7	988.5	991.2	1012.1	1028.7	1057.4
1030	933.5	950.7	959.5	969.4	976.1	981.2	985.5	989.2	992.6	995.7	998.5	1001.1	1022.3	1038.9	1067.9
1040	943.1	960.4	969.2	979.2	985.9	991.0	995.4	999.1	1002.5	1005.6	1008.5	1011.1	1032.4	1049.2	1078.4
1050	952.6	970.0	978.9	989.0	995.7	1000.9	1005.2	1009.0	1012.4	1015.5	1018.4	1021.1	1042.5	1059.5	1088.9
1060	962.2	979.7	988.6	998.8	1005.5	1010.7	1015.1	1018.9	1022.4	1025.5	1028.4	1031.1	1052.7	1069.7	1099.4
1070	971.8	989.4	998.3	1008.6	1015.3	1020.6	1025.0	1028.9	1032.3	1035.4	1038.4	1041.1	1062.8	1080.0	1109.9
1080	981.4	999.0	1008.1	1018.3	1025.2	1030.5	1034.9	1038.8	1042.2	1045.4	1048.3	1051.1	1072.9	1090.3	1120.4
1090	990.9	1008.7	1017.8	1028.1	1035.0	1040.3	1044.8	1048.7	1052.2	1055.3	1058.3	1061.1	1083.1	1100.5	1130.9
1100	1000.5	1018.4	1027.5	1037.9	1044.8	1050.2	1054.7	1058.6	1062.1	1065.3	1068.3	1071.1	1093.2	1110.8	1141.4
1110	1010.1	1028.1	1037.2	1047.7	1054.6	1060.0	1064.5	1068.5	1072.0	1075.2	1078.2	1081.0	1103.4	1121.1	1152.0
1120	1019.7	1037.7	1047.0	1057.5	1064.5	1069.9	1074.4	1078.4	1082.0	1085.2	1088.2	1091.0	1113.5	1131.3	1162.5
1130	1029.3	1047.4	1056.7	1067.3	1074.3	1079.8	1084.3	1088.3	1091.9	1095.2	1098.2	1101.0	1123.6	1141.6	1173.0
1140	1038.8	1057.1	1066.4	1077.1	1084.1	1089.6	1094.2	1098.2	1101.8	1105.1	1108.2	1111.0	1133.8	1151.9	1183.5
1150	1048.4	1066.8	1076.2	1086.9	1094.0	1099.5	1104.1	1108.1	1111.8	1115.1	1118.1	1121.0	1143.9	1162.2	1194.0
1160	1058.0	1076.5	1085.9	1096.6	1103.8	1109.3	1114.0	1118.1	1121.7	1125.0	1128.1	1131.0	1154.1	1172.4	1204.5
1170	1067.6	1086.2	1095.6	1106.4	1113.6	1119.2	1123.9	1128.0	1131.6	1135.0	1138.1	1141.0	1164.2	1182.7	1215.0
1180	1077.2	1095.9	1105.4	1116.2	1123.5	1129.1	1133.8	1137.9	1141.6	1144.9	1148.1	1151.0	1174.4	1193.0	1225.5
1190	1086.8	1105.5	1115.1	1126.0	1133.3	1138.9	1143.7	1147.8	1151.5	1154.9	1158.0	1161.0	1184.5	1203.2	1236.0
1200	1096.4	1115.2	1124.8	1135.8	1143.1	1148.8	1153.6	1157.7	1161.5	1164.9	1168.0	1171.0	1194.7	1213.5	1246.5

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
1210	1106.0	1124.9	1134.6	1145.6	1153.0	1158.7	1163.5	1167.6	1171.4	1174.8	1178.0	1181.0	1204.8	1223.8	1240.9	1257.0
1220	1115.6	1134.6	1144.3	1155.4	1162.8	1168.5	1173.4	1177.6	1181.3	1184.8	1188.0	1191.0	1214.9	1234.1	1251.3	1267.5
1230	1125.2	1144.3	1154.1	1165.2	1172.6	1178.4	1183.3	1187.5	1191.3	1194.7	1198.0	1201.0	1225.1	1244.3	1261.6	1278.1
1240	1134.8	1154.0	1163.8	1175.0	1182.5	1188.3	1193.2	1197.4	1201.2	1204.7	1207.9	1211.0	1235.2	1254.6	1272.0	1288.6
1250	1144.4	1163.7	1173.6	1184.8	1192.3	1198.2	1203.1	1207.3	1211.2	1214.7	1217.9	1221.0	1245.4	1264.9	1282.4	1299.1
1260	1154.0	1173.4	1183.3	1194.6	1202.2	1208.0	1213.0	1217.3	1221.1	1224.6	1227.9	1231.0	1255.5	1275.1	1292.8	1309.6
1270	1163.6	1183.1	1193.1	1204.5	1212.0	1217.9	1222.9	1227.2	1231.1	1234.6	1237.9	1241.0	1265.7	1285.4	1303.2	1320.1
1280	1173.3	1192.8	1202.8	1214.3	1221.9	1227.8	1232.8	1237.1	1241.0	1244.6	1247.9	1251.0	1275.8	1295.7	1313.6	1330.6
1290	1182.9	1202.5	1212.6	1224.1	1231.7	1237.7	1242.7	1247.0	1251.0	1254.5	1257.9	1261.0	1286.0	1306.0	1324.0	1341.1
1300	1192.5	1212.2	1222.3	1233.9	1241.6	1247.5	1252.6	1257.0	1260.9	1264.5	1267.9	1271.0	1296.1	1316.2	1334.4	1351.6
1310	1202.1	1221.9	1232.1	1243.7	1251.4	1257.4	1262.5	1266.9	1270.8	1274.5	1277.8	1281.0	1306.3	1326.5	1344.8	1362.1
1320	1211.7	1231.7	1241.8	1253.5	1261.3	1267.3	1272.4	1276.8	1280.8	1284.4	1287.8	1291.0	1316.4	1336.8	1355.2	1372.7
1330	1221.3	1241.4	1251.6	1263.3	1271.1	1277.2	1282.3	1286.7	1290.7	1294.4	1297.8	1301.0	1326.6	1347.1	1365.6	1383.2
1340	1231.0	1251.1	1261.4	1273.1	1281.0	1287.1	1292.2	1296.7	1300.7	1304.4	1307.8	1311.0	1336.7	1357.4	1376.0	1393.7
1350	1240.6	1260.8	1271.1	1282.9	1290.8	1297.0	1302.1	1306.6	1310.7	1314.4	1317.8	1321.0	1346.9	1367.6	1386.4	1404.2
1360	1250.2	1270.5	1280.9	1292.8	1300.7	1306.8	1312.0	1316.5	1320.6	1324.3	1327.8	1331.0	1357.0	1377.9	1396.8	1414.7
1370	1259.8	1280.2	1290.6	1302.6	1310.5	1316.7	1321.9	1326.5	1330.6	1334.3	1337.8	1341.0	1367.2	1388.2	1407.2	1425.2
1380	1269.5	1289.9	1300.4	1312.4	1320.4	1326.6	1331.8	1336.4	1340.5	1344.3	1347.8	1351.0	1377.4	1398.5	1417.6	1435.7
1390	1279.1	1299.7	1310.2	1322.2	1330.2	1336.5	1341.7	1346.3	1350.5	1354.2	1357.8	1361.0	1387.5	1408.7	1428.0	1446.2
1400	1288.7	1309.4	1319.9	1332.0	1340.1	1346.4	1351.7	1356.3	1360.4	1364.2	1367.7	1371.1	1397.7	1419.0	1438.4	1456.7
1410	1298.4	1319.1	1329.7	1341.9	1349.9	1356.3	1361.6	1366.2	1370.4	1374.2	1377.7	1381.1	1407.8	1429.3	1448.8	1467.3
1420	1308.0	1328.8	1339.5	1351.7	1359.8	1366.2	1371.5	1376.1	1380.3	1384.2	1387.7	1391.1	1418.0	1439.6	1459.1	1477.8
1430	1317.6	1338.5	1349.2	1361.5	1369.7	1376.0	1381.4	1386.1	1390.3	1394.2	1397.7	1401.1	1428.1	1449.9	1469.5	1488.3
1440	1327.3	1348.3	1359.0	1371.3	1379.5	1385.9	1391.3	1396.0	1400.3	1404.1	1407.7	1411.1	1438.3	1460.1	1479.9	1498.8
1450	1336.9	1358.0	1368.8	1381.1	1389.4	1395.8	1401.2	1406.0	1410.2	1414.1	1417.7	1421.1	1448.4	1470.4	1490.3	1509.3
1460	1346.6	1367.7	1378.6	1391.0	1399.3	1405.7	1411.1	1415.9	1420.2	1424.1	1427.7	1431.1	1458.6	1480.7	1500.7	1519.8
1470	1356.2	1377.4	1388.3	1400.8	1409.1	1415.6	1421.1	1425.8	1430.1	1434.1	1437.7	1441.1	1468.7	1491.0	1511.1	1530.3
1480	1365.8	1387.2	1398.1	1410.6	1419.0	1425.5	1431.0	1435.8	1440.1	1444.0	1447.7	1451.2	1478.9	1501.3	1521.5	1540.8
1490	1375.5	1396.9	1407.9	1420.5	1428.8	1435.4	1440.9	1445.7	1450.1	1454.0	1457.7	1461.2	1489.1	1511.5	1531.9	1551.4
1500	1385.1	1406.6	1417.7	1430.3	1438.7	1445.3	1450.8	1455.7	1460.0	1464.0	1467.7	1471.2	1499.2	1521.8	1542.3	1561.9

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
1510	1394.8	1416.4	1427.4	1440.1	1448.6	1455.2	1460.7	1465.6	1470.0	1474.0	1477.7	1481.2	1509.4	1532.1	1552.7	1572.4
1520	1404.4	1426.1	1437.2	1449.9	1458.4	1465.1	1470.7	1475.5	1479.9	1484.0	1487.7	1491.2	1519.5	1542.4	1563.1	1582.9
1530	1414.1	1435.8	1447.0	1459.8	1468.3	1475.0	1480.6	1485.5	1489.9	1493.9	1497.7	1501.2	1529.7	1552.7	1573.5	1593.4
1540	1423.7	1445.6	1456.8	1469.6	1478.2	1484.9	1490.5	1495.4	1499.9	1503.9	1507.7	1511.3	1539.9	1562.9	1583.9	1603.9
1550	1433.4	1455.3	1466.5	1479.4	1488.0	1494.8	1500.4	1505.4	1509.8	1513.9	1517.7	1521.3	1550.0	1573.2	1594.3	1614.4
1560	1443.0	1465.0	1476.3	1489.3	1497.9	1504.7	1510.3	1515.3	1519.8	1523.9	1527.7	1531.3	1560.2	1583.5	1604.7	1625.0
1570	1452.7	1474.8	1486.1	1499.1	1507.8	1514.6	1520.3	1525.3	1529.8	1533.9	1537.7	1541.3	1570.3	1593.8	1615.1	1635.5
1580	1462.4	1484.5	1495.9	1508.9	1517.7	1524.5	1530.2	1535.2	1539.7	1543.9	1547.7	1551.3	1580.5	1604.1	1625.5	1646.0
1590	1472.0	1494.3	1505.7	1518.8	1527.5	1534.4	1540.1	1545.2	1549.7	1553.8	1557.7	1561.3	1590.7	1614.4	1635.9	1656.5
1600	1481.7	1504.0	1515.5	1528.6	1537.4	1544.3	1550.0	1555.1	1559.7	1563.8	1567.7	1571.4	1600.8	1624.6	1646.3	1667.0
1610	1491.3	1513.8	1525.3	1538.5	1547.3	1554.2	1560.0	1565.1	1569.6	1573.8	1577.7	1581.4	1611.0	1634.9	1656.7	1677.5
1620	1501.0	1523.5	1535.0	1548.3	1557.2	1564.1	1569.9	1575.0	1579.6	1583.8	1587.7	1591.4	1621.1	1645.2	1667.1	1688.0
1630	1510.7	1533.2	1544.8	1558.1	1567.0	1574.0	1579.8	1585.0	1589.6	1593.8	1597.7	1601.4	1631.3	1655.5	1677.5	1698.6
1640	1520.3	1543.0	1554.6	1568.0	1576.9	1583.9	1589.8	1594.9	1599.5	1603.8	1607.7	1611.4	1641.5	1665.8	1687.9	1709.1
1650	1530.0	1552.7	1564.4	1577.8	1586.8	1593.8	1599.7	1604.9	1609.5	1613.8	1617.7	1621.5	1651.6	1676.1	1698.3	1719.6
1660	1539.7	1562.5	1574.2	1587.7	1596.7	1603.7	1609.6	1614.8	1619.5	1623.8	1627.7	1631.5	1661.8	1686.3	1708.7	1730.1
1670	1549.3	1572.2	1584.0	1597.5	1606.5	1613.6	1619.5	1624.8	1629.4	1633.7	1637.8	1641.5	1671.9	1696.6	1719.1	1740.6
1680	1559.0	1582.0	1593.8	1607.3	1616.4	1623.5	1629.5	1634.7	1639.4	1643.7	1647.8	1651.5	1682.1	1706.9	1729.5	1751.1
1690	1568.7	1591.7	1603.6	1617.2	1626.3	1633.4	1639.4	1644.7	1649.4	1653.7	1657.8	1661.6	1692.3	1717.2	1739.9	1761.7
1700	1578.3	1601.5	1613.4	1627.0	1636.2	1643.3	1649.3	1654.6	1659.4	1663.7	1667.8	1671.6	1702.4	1727.5	1750.3	1772.2
1710	1588.0	1611.2	1623.2	1636.9	1646.0	1653.2	1659.3	1664.6	1669.3	1673.7	1677.8	1681.6	1712.6	1737.8	1760.7	1782.7
1720	1597.7	1621.0	1633.0	1646.7	1655.9	1663.1	1669.2	1674.5	1679.3	1683.7	1687.8	1691.6	1722.8	1748.0	1771.1	1793.2
1730	1607.4	1630.7	1642.8	1656.6	1665.8	1673.0	1679.1	1684.5	1689.3	1693.7	1697.8	1701.7	1732.9	1758.3	1781.5	1803.7
1740	1617.0	1640.5	1652.6	1666.4	1675.7	1683.0	1689.1	1694.4	1699.3	1703.7	1707.8	1711.7	1743.1	1768.6	1791.9	1814.2
1750	1626.7	1650.3	1662.4	1676.3	1685.6	1692.9	1699.0	1704.4	1709.2	1713.7	1717.8	1721.7	1753.3	1778.9	1802.3	1824.8
1760	1636.4	1660.0	1672.1	1686.1	1695.5	1702.8	1708.9	1714.3	1719.2	1723.7	1727.8	1731.7	1763.4	1789.2	1812.7	1835.3
1770	1646.1	1669.8	1681.9	1696.0	1705.3	1712.7	1718.9	1724.3	1729.2	1733.7	1737.8	1741.8	1773.6	1799.5	1823.1	1845.8
1780	1655.7	1679.5	1691.7	1705.8	1715.2	1722.6	1728.8	1734.3	1739.2	1743.7	1747.8	1751.8	1783.7	1809.8	1833.5	1856.3
1790	1665.4	1689.3	1701.6	1715.7	1725.1	1732.5	1738.7	1744.2	1749.1	1753.7	1757.9	1761.8	1793.9	1820.0	1843.9	1866.8
1800	1675.1	1699.0	1711.4	1725.5	1735.0	1742.4	1748.7	1754.2	1759.1	1763.6	1767.9	1771.8	1804.1	1830.3	1854.3	1877.3

channels	Blocking probability															
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	4.0%	5.0%
1810	1684.8	1708.8	1721.2	1735.4	1744.9	1752.3	1758.6	1764.1	1769.1	1773.6	1777.9	1781.9	1814.2	1840.6	1864.7	1887.9
1820	1694.5	1718.6	1731.0	1745.2	1754.8	1762.3	1768.6	1774.1	1779.1	1783.6	1787.9	1791.9	1824.4	1850.9	1875.1	1898.4
1830	1704.2	1728.3	1740.8	1755.1	1764.7	1772.2	1778.5	1784.0	1789.0	1793.6	1797.9	1801.9	1834.6	1861.2	1885.5	1908.9
1840	1713.8	1738.1	1750.6	1764.9	1774.5	1782.1	1788.4	1794.0	1799.0	1803.6	1807.9	1812.0	1844.7	1871.5	1895.9	1919.4
1850	1723.5	1747.9	1760.4	1774.8	1784.4	1792.0	1798.4	1804.0	1809.0	1813.6	1817.9	1822.0	1854.9	1881.8	1906.3	1929.9
1860	1733.2	1757.6	1770.2	1784.6	1794.3	1801.9	1808.3	1813.9	1819.0	1823.6	1827.9	1832.0	1865.1	1892.1	1916.8	1940.4
1870	1742.9	1767.4	1780.0	1794.5	1804.2	1811.8	1818.2	1823.9	1829.0	1833.6	1838.0	1842.0	1875.2	1902.3	1927.2	1951.0
1880	1752.6	1777.2	1789.8	1804.4	1814.1	1821.7	1828.2	1833.8	1838.9	1843.6	1848.0	1852.1	1885.4	1912.6	1937.6	1961.5
1890	1762.3	1786.9	1799.6	1814.2	1824.0	1831.7	1838.1	1843.8	1848.9	1853.6	1858.0	1862.1	1895.6	1922.9	1948.0	1972.0
1900	1772.0	1796.7	1809.4	1824.1	1833.9	1841.6	1848.1	1853.8	1858.9	1863.6	1868.0	1872.1	1905.7	1933.2	1958.4	1982.5
1910	1781.7	1806.5	1819.2	1833.9	1843.8	1851.5	1858.0	1863.7	1868.9	1873.6	1878.0	1882.2	1915.9	1943.5	1968.8	1993.0
1920	1791.4	1816.2	1829.0	1843.8	1853.7	1861.4	1868.0	1873.7	1878.9	1883.6	1888.0	1892.2	1926.1	1953.8	1979.2	2003.5
1930	1801.0	1826.0	1838.8	1853.6	1863.6	1871.3	1877.9	1883.7	1888.8	1893.6	1898.1	1902.2	1936.3	1964.1	1989.6	2014.1
1940	1810.7	1835.8	1848.7	1863.5	1873.5	1881.3	1887.8	1893.6	1898.8	1903.6	1908.1	1912.3	1946.4	1974.4	2000.0	2024.6
1950	1820.4	1845.5	1858.5	1873.4	1883.3	1891.2	1897.8	1903.6	1908.8	1913.6	1918.1	1922.3	1956.6	1984.6	2010.4	2035.1
1960	1830.1	1855.3	1868.3	1883.2	1893.2	1901.1	1907.7	1913.5	1918.8	1923.6	1928.1	1932.3	1966.8	1994.9	2020.8	2045.6
1970	1839.8	1865.1	1878.1	1893.1	1903.1	1911.0	1917.7	1923.5	1928.8	1933.6	1938.1	1942.4	1976.9	2005.2	2031.2	2056.1
1980	1849.5	1874.9	1887.9	1902.9	1913.0	1920.9	1927.6	1933.5	1938.8	1943.6	1948.1	1952.4	1987.1	2015.5	2041.6	2066.6
1990	1859.2	1884.6	1897.7	1912.8	1922.9	1930.9	1937.6	1943.4	1948.7	1953.6	1958.2	1962.4	1997.3	2025.8	2052.0	2077.2
2000	1868.9	1894.4	1907.5	1922.7	1932.8	1940.8	1947.5	1953.4	1958.7	1963.6	1968.2	1972.5	2007.4	2036.1	2062.4	2087.7
2010	1878.6	1904.2	1917.4	1932.5	1942.7	1950.7	1957.4	1963.4	1968.7	1973.6	1978.2	1982.5	2017.6	2046.4	2072.8	2098.2
2020	1888.3	1914.0	1927.2	1942.4	1952.6	1960.6	1967.4	1973.3	1978.7	1983.6	1988.2	1992.5	2027.8	2056.7	2083.2	2108.7
2030	1898.0	1923.7	1937.0	1952.3	1962.5	1970.6	1977.3	1983.3	1988.7	1993.6	1998.2	2002.6	2037.9	2067.0	2093.6	2119.2
2040	1907.7	1933.5	1946.8	1962.1	1972.4	1980.5	1987.3	1993.3	1998.7	2003.6	2008.3	2012.6	2048.1	2077.2	2104.0	2129.8
2050	1917.4	1943.3	1956.6	1972.0	1982.3	1990.4	1997.2	2003.2	2008.7	2013.6	2018.3	2022.6	2058.3	2087.5	2114.4	2140.3
2060	1927.1	1953.1	1966.4	1981.9	1992.2	2000.3	2007.2	2013.2	2018.6	2023.6	2028.3	2032.7	2068.5	2097.8	2124.8	2150.8
2070	1936.9	1962.9	1976.3	1991.7	2002.1	2010.3	2017.1	2023.2	2028.6	2033.6	2038.3	2042.7	2078.6	2108.1	2135.2	2161.3
2080	1946.6	1972.6	1986.1	2001.6	2012.0	2020.2	2027.1	2033.1	2038.6	2043.6	2048.3	2052.8	2088.8	2118.4	2145.6	2171.8
2090	1956.3	1982.4	1995.9	2011.5	2021.9	2030.1	2037.0	2043.1	2048.6	2053.7	2058.4	2062.8	2099.0	2128.7	2156.0	2182.3
2100	1966.0	1992.2	2005.7	2021.3	2031.8	2040.0	2047.0	2053.1	2058.6	2063.7	2068.4	2072.8	2109.1	2139.0	2166.4	2192.9

channels	Blocking probability														
	0.01%	0.05%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	2.0%	3.0%	5.0%
2110	1975.7	2002.0	2015.5	2031.2	2041.7	2050.0	2056.9	2063.1	2068.6	2073.7	2078.4	2082.9	2119.3	2149.3	2176.9
2120	1985.4	2011.8	2025.4	2041.1	2051.6	2059.9	2066.9	2073.0	2078.6	2083.7	2088.4	2092.9	2129.5	2159.6	2187.3
2130	1995.1	2021.6	2035.2	2050.9	2061.5	2069.8	2076.8	2083.0	2088.6	2093.7	2098.4	2102.9	2139.7	2169.9	2197.7
2140	2004.8	2031.3	2045.0	2060.8	2071.4	2079.7	2086.8	2093.0	2098.5	2103.7	2108.5	2113.0	2149.8	2180.2	2208.1
2150	2014.5	2041.1	2054.8	2070.7	2081.3	2089.7	2096.7	2102.9	2108.5	2113.7	2118.5	2123.0	2160.0	2190.4	2218.5
2160	2024.2	2050.9	2064.7	2080.6	2091.2	2099.6	2106.7	2112.9	2118.5	2123.7	2128.5	2133.1	2170.2	2200.7	2228.9
2170	2034.0	2060.7	2074.5	2090.4	2101.1	2109.5	2116.6	2122.9	2128.5	2133.7	2138.5	2143.1	2180.4	2211.0	2239.3
2180	2043.7	2070.5	2084.3	2100.3	2111.0	2119.5	2126.6	2132.9	2138.5	2143.7	2148.6	2153.1	2190.5	2221.3	2249.7
2190	2053.4	2080.3	2094.1	2110.2	2120.9	2129.4	2136.5	2142.8	2148.5	2153.7	2158.6	2163.2	2200.7	2231.6	2260.1
2200	2063.1	2090.1	2104.0	2120.0	2130.8	2139.3	2146.5	2152.8	2158.5	2163.7	2168.6	2173.2	2210.9	2241.9	2270.5
2210	2072.8	2099.8	2113.8	2129.9	2140.7	2149.3	2156.4	2162.8	2168.5	2173.7	2178.6	2183.3	2221.0	2252.2	2280.9
2220	2082.5	2109.6	2123.6	2139.8	2150.6	2159.2	2166.4	2172.7	2178.5	2183.7	2188.7	2193.3	2231.2	2262.5	2291.3
2230	2092.3	2119.4	2133.5	2149.7	2160.6	2169.1	2176.4	2182.7	2188.5	2193.8	2198.7	2203.3	2241.4	2272.8	2301.7
2240	2102.0	2129.2	2143.3	2159.5	2170.5	2179.0	2186.3	2192.7	2198.5	2203.8	2208.7	2213.4	2251.6	2283.1	2312.1
2250	2111.7	2139.0	2153.1	2169.4	2180.4	2189.0	2196.3	2202.7	2208.5	2213.8	2218.7	2223.4	2261.7	2293.4	2322.5
2260	2121.4	2148.8	2162.9	2179.3	2190.3	2198.9	2206.2	2212.6	2218.4	2223.8	2228.8	2233.5	2271.9	2303.7	2332.9
2270	2131.1	2158.6	2172.8	2189.2	2200.2	2208.8	2216.2	2222.6	2228.4	2233.8	2238.8	2243.5	2282.1	2313.9	2343.3
2280	2140.9	2168.4	2182.6	2199.0	2210.1	2218.8	2226.1	2232.6	2238.4	2243.8	2248.8	2253.5	2292.3	2324.2	2353.7
2290	2150.6	2178.2	2192.4	2208.9	2220.0	2228.7	2236.1	2242.6	2248.4	2253.8	2258.8	2263.6	2302.4	2334.5	2364.2
2300	2160.3	2188.0	2202.3	2218.8	2229.9	2238.6	2246.0	2252.5	2258.4	2263.8	2268.9	2273.6	2312.6	2344.8	2374.6
2310	2170.0	2197.8	2212.1	2228.7	2239.8	2248.6	2256.0	2262.5	2268.4	2273.8	2278.9	2283.7	2322.8	2355.1	2385.0
2320	2179.8	2207.6	2221.9	2238.5	2249.7	2258.5	2266.0	2272.5	2278.4	2283.8	2288.9	2293.7	2333.0	2365.4	2395.4
2330	2189.5	2217.4	2231.8	2248.4	2259.6	2268.5	2275.9	2282.5	2288.4	2293.9	2299.0	2303.8	2343.1	2375.7	2405.8
2340	2199.2	2227.2	2241.6	2258.3	2269.5	2278.4	2285.9	2292.4	2298.4	2303.9	2309.0	2313.8	2353.3	2386.0	2416.2
2350	2208.9	2237.0	2251.4	2268.2	2279.5	2288.3	2295.8	2302.4	2308.4	2313.9	2319.0	2323.8	2363.5	2396.3	2426.6
2360	2218.7	2246.8	2261.3	2278.1	2289.4	2298.3	2305.8	2312.4	2318.4	2323.9	2329.0	2333.9	2373.7	2406.6	2437.0
2370	2228.4	2256.6	2271.1	2287.9	2299.3	2308.2	2315.7	2322.4	2328.4	2333.9	2339.1	2343.9	2383.8	2416.9	2447.4
2380	2238.1	2266.3	2280.9	2297.8	2309.2	2318.1	2325.7	2332.4	2338.4	2343.9	2349.1	2354.0	2394.0	2427.2	2457.8
2390	2247.8	2276.1	2290.8	2307.7	2319.1	2328.1	2335.7	2342.3	2348.4	2353.9	2359.1	2364.0	2404.2	2437.5	2468.2
2400	2257.6	2285.9	2300.6	2317.6	2329.0	2338.0	2345.6	2352.3	2358.4	2364.0	2369.2	2374.1	2414.4	2447.7	2478.6

APPENDIX B

Visual Basic® Code for Erlang Calculations

The following code can be used in a spreadsheet macro to enable the automatic calculation of Erlang values. The function GetBlocking generates a blocking value based on two inputs: Erlangs and circuits. The function GetCircuits generates a number of circuits based on two inputs: Erlangs and blocking. The function GetErlangs calculates the number of Erlangs based on two inputs: circuits and blocking.

```
Attribute VB_Name = "Erlangs"
Option Explicit
Function Ceil(Value As Double) As Integer
    Dim i As Integer
    i = Value
    If i < Value Then
        i = i + 1
    End If
    Ceil = i
End Function

Function GetBlocking(Erlangs As Double, Circuits As Integer) As Double
    Dim Blocking As Double
    Dim i As Integer
    If (Erlangs = 0) Or (Circuits = 0) Then
        GetBlocking = 0
        Exit Function
    End If
    Blocking = 0
    For i = 1 To Circuits
        Blocking = (1 + Blocking) \ (i / Erlangs)
    Next
    GetBlocking = 1 / (1 + Blocking)
End Function

Function GetCircuits(Blocking As Double, Erlangs As Double) As Integer
    Dim fR As Double, fMid As Double
    Dim l As Integer, r As Integer, mid As Integer
    If (Blocking < 0.0001) Or (Blocking > 0.9999) Or (Erlangs = 0)
    Then
        GetCircuits = 0
        Exit Function
    End If
    l = 0
    r = Ceil(Erlangs)
    fR = GetBlocking(Erlangs, r)
    While (fR > Blocking)
        l = r
        r = r + 32
        fR = GetBlocking(Erlangs, r)
    End While
```

```

Wend
While ((r - 1) > 1)
    mid = Ceil((1 + r) / 2)
    fMid = GetBlocking(Erlangs, mid)
    If (fMid > Blocking) Then
        l = mid
    Else
        r = mid
    End If
Wend
GetCircuits = r
End Function

Function GetErlangs(Blocking As Double, Circuits As Integer) As
Double
    Dim l As Double, r As Double, i As Double, Delta As Double

    If (Blocking < 0.0001) Or (Blocking > 0.9999) Or (Circuits = 0)
Then
        GetErlangs = 0
        Exit Function
    End If

    i = 0
    Do
        i = i + 1
        Loop While (Blocking > GetBlocking(Circuits * i, Circuits))

    r = i * Circuits
    l = (i - 1) * Circuits
    Delta = r - l

    While (Delta > 0.001)
        Delta = (r - l) / 2
        If (Blocking > GetBlocking(l + Delta, Circuits)) Then
            l = l + Delta
        Else
            r = l + Delta
        End If
    Wend

    GetErlangs = l
End Function

```

GLOSSARY OF ACRONYMS

ABNF	Augmented Backus-Naur Form
ACELP	Algebraic Code-Excited Linear Prediction
ACF	AdmissionConfirm message
ACM	Address Complete message
ADPCM	Adaptive Differential Pulse Code Modulation
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
AMI	Alternate Mark Inversion
AMR	Adaptive Multi-Rate
ANM	Answer message
ANSI	American National Standards Institute
API	Application Programming Interface
ARJ	AdmissionReject message
ARQ	AdmissionRequest message
AS	Application server
ASN.1	Abstract Syntax Notation 1
ASP	Application Server Process
ATM	Asynchronous Transfer Mode
AUCX	AuditConnection (MGCP command)
AUEP	AuditEndpoint (MGCP command)
BCF	BandwidthConfirm message
BCP	Best Current Practice
BGP	Border Gateway Protocol
BHCA	Busy-hour call attempts
BRI	Basic Rate Interface
BRJ	BandwidthReject message
BRQ	BandwidthRequest message
CAP	Competitive Access Provider
CAS	Channel-Associated Signaling
CBR	Constant bit rate
CBS	Committed burst size
CCBS	Call Completion to Busy Subscriber
CCS	Common Channel Signaling
CDMA	Code Division Multiple Access
CDR	Committed data rate
CELP	Code-Excited Linear Prediction
CIC	Circuit Identification Code
CLEC	Competitive Local Exchange Carrier
CNG	Comfort noise generation

COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CPE	Customer Premises Equipment
CPG	Call Progress
CRCX	CreateConnection (MGCP command)
CRLF	Carriage return and line feed
CR-LSP	Constraint-Based LSP
CS-1	Capability Set 1
CSCF	Call State Control Function
CSeq	Command sequence
CSRC	Contributing Source
DCF	DisengageConfirm message
DiffServ	Differentiated Services
DLCX	DeleteConnection (MGCP Command)
DNS	Domain Name System
DPC	Destination Point Code
DPCM	Differential Pulse Code Modulation
DRJ	DisengageReject message
DRQ	DisengageRequest message
DS	DiffServ
DS0	Digital Signal 0 (equivalent to 64 Kbps)
DS1	Digital Signal 1 (equivalent to 1.544 Mbps)
DS3	Digital Signal 3 (equivalent to 44.736 Mbps)
DSCP	DiffServ codepoint
DSL	Digital Subscriber Line
DSP	Digital signal processor
DTMF	Dual-tone multifrequency
DTX	Discontinuous transmission
DWDM	Dense Wave Division Multiplexing
EBS	Excess burst size
EC	Echo cancellation
EF	Expedited Forwarding
EFR	Enhanced Full Rate
EMS	Element Management System
EPCF	EndpointConfiguration (MGCP command)
ER	Explicit route
ESP	Enhanced service provider
ETSI	European Telecommunications Standards Institute
FCS	Frame Check Sequence
FDM	Frequency Division Multiplexing

FE	Fast Ethernet
FEC	Forwarding Equivalence Class
FISU	Fill-In Signal Unit
FoIP	Fax over IP
FR	Full Rate
GCF	GatekeeperConfirm message
Gig-E	Gigabit Ethernet
GMSK	Gaussian Minimum Shift Key
GRJ	GatekeeperReject message
GRQ	GatekeeperRequest message
GSM	Global System for Mobile communications
GSTN	General Switched Telephone Network
GT	Global Title
HDLC	High-level Data Link Control
HDSL	High-bit-rate Digital Subscriber Line
HTTP	Hypertext Transfer Protocol
IAB	Internet Architecture Board
IACK	InfoRequestAck message
IAM	Initial Address message
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGMP	Internet Group Message Protocol
IGRP	Internet Gateway Routing Protocol
IN	Intelligent Network
INAK	InfoRequestNak message
INAP	Intelligent Network Application Part
IP	Internet Protocol
IRQ	InformationRequest message
IRR	InformationRequestResponse message
ISDN	Integrated Services Digital Network
ISDN-UP or ISUP	ISDN User Part
ISP	Internet service provider
ITSP	IP telephony service provider
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union- Telecommunications Standardization Sector (formerly CCITT)
IUA	ISDN User Adaptation Layer

IVR	Interactive Voice Response
JPEG	Joint Photographic Experts Group
LAN	Local area network
LCF	LocationConfirm message
LD	Long distance
LD-CELP	Low-Delay Code-Excited Linear Prediction
LDP	Label Distribution Protocol
LEC	Local Exchange Carrier
LER	Label edge router
LMDS	Local Multipoint Distribution Service
LPC	Linear Prediction Coding
LRJ	LocationReject message
LRQ	LocationRequest message
LSP	Label-switched path
LSR	Label-switching router
LSSU	Link Status Signal Unit
M2UA	MTP2 User Adaptation Layer
M2PA	MTP2 Peer-to-Peer Adaptation Layer
M3UA	MTP3 User Adaptation Layer
MAP	Mobile Application Part
MC	Multipoint controller
MCF	MessageConfirm message
MCU	Multipoint controller unit
MDCX	ModifyConnection (MGCP command)
MF	Multifrequency
MFC	Multifrequency Compelled
MG	Media gateway
MGC	Media gateway controller
MGCP	Media Gateway Control Protocol
MHT	Mean holding time
MIME	Multipurpose Internet Mail Extension
MIPS	Million instructions per second
MMUSIC	Multiparty Multimedia Session Control
MOS	Mean Opinion Score
MoU	Minutes of use
MP	Multipoint processor
MPE	Multi-Pulse Excited
MPLS	Multiprotocol Label Switching
MP-MLQ	Multi-Pulse Maximum Likelihood Quantization
MRF	Multimedia Resource Function
MSU	Message signal unit

MTA	Media terminal adapter
MTBF	Mean Time Between Failures
MTP	Message Transfer Part
MTU	Maximum transmission unit
NGW	Network gateway
NHLFE	Next Hop Level Forwarding Entry
NIF	Nodal Interworking Function
NIST	National Institute of Standards and Technology
NMS	Network Management System
NSF	Non-Standard Facilities
NSM	Non-Standard Message
NTFY	Notify (MGCP command)
NTP	Network Time Protocol
OA&M	Operations, Administration, and Maintenance
OPC	Origination point code
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PBS	Peak burst size
PBX	Private branch exchange
PC	Personal computer
PC	Point code
PCM	Pulse Code Modulation
PDD	Post-Dial Delay
PDP	Policy Decision Point
PDR	Peak data rate
PEP	Policy Enforcement Point
PHB	Per-hop behavior
PINT	PSTN and Internet Interworking
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PQ	Priority queue
PRACK	ProvisionalResponseAcknowledgment message
PSTN	Public Switched Telephone Network
PT	Payload Type
PVC	Permanent virtual circuit
QCIF	Quarter Common Intermediate Format
QoS	Quality of service
RAC	ResourceAvailableConfirm message
RAck	ResponseAcknowledgment Header
RAI	ResourceAvailableIndicate message
RAS	Registration, Admission, and Status

RCF	RegistrationConfirm message
REL	Release message
RFC	Request for Comments
RFI	Request for Information
RFP	Request for Proposal
RIP	Routing Information Protocol
RIP	Request in Progress
RLC	Release Complete message
RPE-LTP	Regular Pulse-Excited Long term Prediction
RQNT	NotificationRequest (MGCP command)
RRJ	RegistrationReject message
RRQ	RegistrationRequest message
RSeq	Response sequence
RSIP	RestartInProgress (MGCP command)
RSVP	Resource Reservation Protocol
RSVP-TE	RSVP Traffic Engineering
RTCP	Real-Time Control Protocol
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SAP	Session Announcement Protocol
SCCP	Signaling Connection Control Part
SCP	Service control point
SCTP	Stream Control Transmission Protocol
SDES	Source Description
SDP	Session Description Protocol
SG	Signaling gateway
SID	Silence Insertion Description
SIF	Signaling Information Field
SIMPLE	SIP for Instant Message and Presence Leveraging Extensions
SIO	Service Information Octet
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLS	Signaling Link Selection
SNMP	Simple Network Management Protocol
SOHO	Small Office/Home Office
SONET	Synchronous Optical Network
SP	Signaling point
SPC	Signaling point code
SS7	Signaling System Number 7
SSN	Subsystem Number

SSP	Service switching point
SSRC	Synchronization Source
STP	Signal transfer point
SUA	SCCP User Adaptation Layer
TA	Terminal adapter
TCAP	Transaction Capabilities Application Part
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TFA	Transfer Allowed
TFP	Transfer Prohibited
TLV	Type-Length-Value
ToS	Type of Service
TSAP	Transport Service Access Point
TTL	Time to live
UCF	UnregistrationConfirm message
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URJ	UnregistrationReject message
URL	Uniform Resource Locator
URQ	UnregistrationRequest message
UTF	Unicode Standard Transformation Format
V5UA	V5.2 User Adaptation Layer
VAD	Voice activity detection
VBR	Variable bit rate
VoATM	Voice over ATM
VoIP	Voice over IP
WAN	Wide area network
WRR	Weighted Round-Robin
XRS	UnknownMessageResponse message

REFERENCES

Chapter 2

- Braden, R. "Requirements for Internet Hosts—Communication Layers." RFC 1122, October 1989.
- Braden, R. "Requirements for Internet Hosts—Application and Support." RFC 1123, October 1989.
- Bradner, S. "The Internet Standards Process—Revision 3." RFC 2026, October 1996.
- Deering, S. "Host Extensions for IP Multicasting." RFC 1112, August 1989.
- Deering, S. and R. Hinden. "IP Version 6 Addressing Architecture." RFC 2373, July 1998.
- Fenner, W. "Internet Group Management Protocol, Version 2." RFC 2236, November 1997.
- Mills, D. "Network Time Protocol (Version 3) Specification, Implementation, and Analysis." RFC 1305, March 1992.
- Mogul, J. "Broadcasting Internet Datagrams." RFC 919, October 1984.
- Mogul, J. and J. Postel. "Internet Standard Subnetting Procedure." RFC 950, August 1985.
- Moy, J. "Multicast Extensions to OSPF." RFC 1584, March 1994.
- Perkins, C. et al. "RTP Payload for Redundant Audio Data." RFC 2198, September 1997.
- Postel, J. "Internet Protocol." RFC 791, September 1981.
- Postel, J. "Transmission Control Protocol." RFC 793, September 1981.
- Postel, J. and J. Reynolds. "Domain Requirements." RFC 920, October 1984.
- Rekhter, Y. and T. Li. "A Border Gateway Control Protocol 4 (BGP-4)." RFC 1771, March 1995.
- Schulzrinne, H. "RTP Profile for Audio and Video Conferences with Minimal Control." RFC 1890, January 1996.
- Schulzrinne, H. "RTP: A Transport Protocol for Real-Time Applications." Internet-draft, work in progress, November 2001.

- Schulzrinne, H. et al. "RTP: A Transport Protocol for Real-Time Applications." RFC 1889, January 1996.
- Schulzrinne, H. and S. Casner. "RTP Profile for Audio and Video Conferences with Minimal Control." Internet-draft, work in progress, November 2001.

Chapter 3

- ITU-T. "Methods for subjective determination of transmission quality." Recommendation P.800, August 1996.
- ITU-T. "Objective quality measurement of telephone-band (300–3,400 Hz) speech codecs." Recommendation P.861, February 1998.
- ITU-T. "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs." Recommendation P.862, February 2001.
- ITU-T. "Artificial voices." Recommendation P.50, September 1999.
- ITU-T. "Pulse code modulation (PCM) of voice frequencies." Recommendation G.711, November 1988.
- ITU-T. "32kb/s adaptive differential pulse code modulation (ADPCM)." Recommendation G.721, October 1988.
- ITU-T. "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)." Recommendation G.726, December 1990.
- ITU-T. "Coding of speech at 16 kbit/s using low-delay code excited linear prediction." Recommendation G.728, September 1992.
- ITU-T. "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s." Recommendation G.723.1, March 1996.
- ITU-T. "Silence suppression scheme." Recommendation G.723.1, Annex A, November 1996.
- ITU-T. "Coding of speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)." Recommendation G.729, March 1996.

- ITU-T. "Reduced complexity 8 kbit/s CS-ACELP speech codec." Recommendation G.729, Annex A, November 1996.
- ITU-T. "A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70." Recommendation G.729, Annex B, October 1996.
- ITU-T. "6.4 kbit/s CS-ACELP speech coding algorithm." Recommendation G.729, Annex D, September 1998.
- ITU-T. "11.8 kbit/s CS-ACELP speech coding algorithm." Recommendation G.729, Annex E, September 1998.
- ETSI. "Enhanced Full Rate (EFR) speech transcoding." EN 300 726 (GSM 06.60), December 1999.
- ETSI. "Adaptive Multi-Rate (AMR) speech transcoding." EN 301 704 (GSM 06.90), April 2000.
- Schulzrinne, H. and S. Petrack. "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals." RFC 2833, May 2000.
- TIA/EIA. "High Rate Speech Service Option 17 for Wideband Spread Spectrum Communications Systems." IS-733, March 1998.

The complete volume of the ITU material, from which the texts reproduced are extracted, can be obtained from

International Telecommunication Union

Sales and Marketing Service

Place des Nations

CH-1211 Geneva 20 (Switzerland)

Telephone: +41 22 730 6141 (English) / +41 22 730 61 42 (French) /
+41 22 730 61 43 (Spanish)

Telex: 421 000 uit ch / Fax: +41 22 730 51 94

X.400: S = Sales; P = itu; A = 400net; C = ch

E-mail: sales@itu.int

Web address: www.itu.int/publications

Chapter 4

- ITU-T. "Packet-based multimedia communications systems."
Recommendation H.323 version 4, November 2000.
- ITU-T. "Call signalling protocols and media stream packetization for packet-based multimedia communication systems."
Recommendation H.225.0 version 4, November 2000.
- ITU-T. "Control protocol for multimedia communication."
Recommendation H.245 version 8, July 2001.
- ITU-T. "ISDN user-network interface layer 3 specification for basic call control." Recommendation Q.931, May 1998.
- ITU-T. "Video codec for audiovisual services at $p \times 64$ kbit/s."
Recommendation H.261, March 1993.
- ITU-T. "Generic procedures for the control of ISDN supplementary services." Recommendation Q.932, May 1998.

Chapter 5

- Agrawal, H. et al. "SIP-H.323 Interworking Requirements." work in progress.
- Camarillo, G. et al. "Integration of Resource Management and SIP." work in progress.
- Campbell, B. et al. "Session Initiation Protocol Extension for Instant Messaging." work in progress.
- Donovan, S. "The SIP INFO Method." RFC 2976, October 2000.
- Handley, M. et al. "SIP: Session Initiation Protocol." RFC 2543, March 1999.
- Handley, M. et al. "Session Announcement Protocol." RFC 2974, October 2000.
- Handley, M. and V. Jacobson. "SDP: Session Description Protocol." RFC 2327, April 1998.
- Roach, A.B. "Session Initiation Protocol (SIP)-Specific Event Notification." RFC 3265, June 2002.
- Rosenberg, J. "The Session Initiation Protocol UPDATE Method." work in progress.

- Rosenberg, J. and H. Schulzrinne. "Reliability of Provisional Responses in SIP." RFC 3262, June 2002.
- Rosenberg, J. and H. Schulzrinne. "An Offer/Answer Model with the Session Description Protocol (SDP)." RFC 3264, June 2002.
- Rosenberg, J. et al. "Session Initiation Protocol (SIP) Extensions for Presence." work in progress.
- Rosenberg, J. et al. "SIP: Session Initiation Protocol." RFC 3261, June 2002.
- Schulzrinne, H. et al. "Real Time Streaming Protocol (RTSP)." RFC 2326, April 1998.
- Sparks, R. "The SIP REFER Method." work in progress.
- Sparks, R. "Internet Media Type message/sipfrag." work in progress.
- Zimmerer, E. et al. "MIME media types for ISUP and QSIG Objects." RFC 3204, December 2001.

Chapter 6

- Arango, M. et al. "Media Gateway Control Protocol (MGCP) Version 1.0." RFC 2705, October 1999.
- Cuervo, F. et al. "MEGACO Protocol version 1.0." RFC 3015, November 2000.
- Greene, N. et al. "Media Gateway Control Protocol Architecture and Requirements." RFC 2805, April 2000.
- Pantaleo, M. "The Megaco/H.248 Gateway Control Protocol, version 2." work in progress.

Chapter 7

- ANSI. "Telecommunications-Signalling System Number 7 (SS7)-Message Transfer Part (MTP)." T1.111, 1996.
- ANSI. "Telecommunications-Signalling System Number 7 (SS7)-Integrated Services Digital Network (ISDN) User Part." T1.113, 1995.
- ANSI. "Telecommunications-Signalling System Number 7 (SS7)-Signaling Connection Control Part (SCCP)." T1.112, 1996.

- ANSI. "Telecommunications-Signalling System Number 7 (SS7)-Transaction Capability Application Part (TCAP)." T1.114, 1996.
- Freed, N. and N. Borenstein. "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types." RFC 2046, November 1996.
- George, T. et al. "SS7 MTP2-User Peer-to-Peer Adaptation Layer." work in progress.
- ITU-T. "Functional Description of the Message Transfer Part (MTP) of Signalling System No. 7." Recommendation Q.701, March 1993.
- ITU-T. "Signalling System No. 7—ISDN User Part Functional Description." Recommendation Q.761, December 1999.
- ITU-T. "Functional description of the Signalling Connection Control Part." Recommendation Q.711, March 2001.
- ITU-T. "Functional description of Transaction Capabilities." Recommendation Q.771, June 1997.
- Loughney, J. "Signalling Connection Control Part User Adaptation Layer (SUA)." work in progress.
- Morneault, K. et al. "ISDN Q.921-User Adaptation Layer (IUA)." RFC 3057, February 2001.
- Morneault, K. et al. "Signaling System 7 (SS7) Message Transfer Part (MTP) 2 User Adaptation Layer." work in progress.
- Ong, L. et al. "Framework Architecture for Signaling Transport." RFC 2719, October 1999.
- Sidebottom, G. et al. "SS7 MTP3-User Adaptation Layer (M3UA)." work in progress.
- Stewart, R. et al. "Stream Control Transmission Protocol." RFC 2960, October 2000.
- Telcordia Technologies. "Specification of Signaling System Number 7." GR-246-CORE, December 1999.
- Weilandt, E. et al. "V5.2-User Adaptation Layer (V5UA)." work in progress.

Chapter 8

- Andersson, L. et al. "LDP Specification." RFC 3036, January 2001.
- Awduche, D. "Requirements for Traffic Engineering over MPLS." RFC 2702, September 1999.
- Awduche, D. et al. "RSVP-TE: Extensions to RSVP for LSP Tunnels." RFC 3209, December 2001.
- Blake, S. et al. "An Architecture for Differentiated Services." RFC 2475, December 1998.
- Braden, R. et al. "Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification." RFC 2205, September 1997.
- Davie, B. et al. "An Expedited Forwarding Per-Hop Behavior (PHB)." RFC 3246, March 2002.
- Durham, D. et al. "The COPS (Common Open Policy Service) Protocol." RFC 2748, January 2000.
- Grossman, D. "New Terminology and Clarifications for DiffServ." RFC 3260, April 2002.
- Heinanen, J. et al. "Assured Forwarding PHB." RFC 2597, June 1999.
- Jamoussi, B. et al. "Constraint-Based LSP Setup Using LDP." RFC 3212, January 2002.
- Ramakrishnan, K. et al. "The Addition of Explicit Congestion Notification (ECN) to IP." RFC 3168, September 2001.
- Rosen, E. et al. "Multiprotocol Label Switching Architecture." RFC 3031, January 2001.
- Shenker, S. and J. Wroclawski. "General Characterization Parameters for Integrated Service Network Elements." RFC 2215, September 1997.
- Shenker, S., C. Partridge, and R. Guerin. "Specification of Guaranteed Quality of Service." RFC 2212, September 1997.
- Wroclawski, J. "Specification of Controlled-Load Network Element Service." RFC 2211, September 1997.

INDEX

A

- A parameter, MGCP, 260
- A-law, G.711, 90
- AbS codecs (Analysis by Synthesis), 91–92
- access management, VoIP, 19
- ACELP codec, 94–97
- ACF messages, H.323, 122
- ACK messages, SIP, 167
- ACK method, SIP requests, 171
- ACM messages, ISUP, 324–325
- ad hoc conference calls, H.245, 156
- adaptation modules, SCTP, 331–336
- add command, MEGACO/H.248,
286, 302–303
- addresses
 - H.323
 - aliases, 114
 - TSAP identifiers, 113
 - SIP, 175–176
 - SS7, 323–324
- Adler-32 checksum, SCTP, 339
- admission, H.323, 122–124
- ADPCM codec, 90–91
- ADSpecs, RSVP, 394–395
- advanced services, 9
- advertisement messages, LDP, 423
- AF PHB, DiffServ, 408
- alarms, network design, 449
- alerting messages, H.323, 132
- aliases, H.323 addresses, 114
- analog lines, MGCP endpoints, 256
- ANM messages, ISUP, 326
- Annex A, G.729, 96
- Annex B, G.729, 96
- Annex D, G.729, 98
- Annex E, G.729, 98
- announcement server access points,
MGCP endpoints, 256
- applicability statements, Internet RFCs, 33
- Application layer, OSI model, 29
- architecture
 - DiffServ, 405
 - H.323, 107
 - MEGACO/H.248, 281
 - MPLS, 412–413
 - Sigtran, 330–331
 - SIP, 163
 - softswitch, 250–251
 - SS7, 318
- areas, OSPF, 37
- ARJ messages, H.323, 124
- ARQ messages, H.323, 122
- AS (autonomous system), OSPF, 36
- ASP state management messages,
M3UA, 359–360
- ASP traffic management messages,
M3UA, 360–361
- associations, SCTP, 338, 346–347
- ATM (Asynchronous Transfer Mode), 35
- ATM trunk-side interface endpoints,
MGCP, 256
- Attributes field, SDP, 205
- AUCX command, MGCP, 259
- audit descriptor, MEGACO/H.248, 292
- AuditCapabilities command,
MEGACO/H.248, 287, 305
- AuditValue command, MEGACO/H.248,
287, 305
- AUEP command, MGCP, 259

B

- B parameter, MGCP, 260
- backbone areas, OSPF, 37
- backplane systems, MGs, 446
- bake-offs, SIP development, 162

- bandwidth
 - allocation, QoS, 382–383
 - H.323, 125
 - network design
 - calculations, 463
 - requirements, 455
- BCF messages, H.323, 125
- BCP (best current practices),
 - Internet RFCs, 33
- BGP (Border Gateway Protocol), 38
- BHCA (busy-hour call attempts), 453
- bidirectional logical channels, H.245, 148
- blocking probability, 444
- body, SIP messages, 170
- BRJ messages, H.323, 125
- BRQ messages, H.323, 125
- build-ahead, network design, 440
- busy hour Erlangs, network design, 451
- BYE messages, SIP, 167
- BYE method, SIP requests, 171

C

- C parameter, MGCP, 260
- call agents
 - MGCP, call setup, 276–277
 - softswitch architecture, 250
- call control
 - media separation, 248–249
 - MGCP, 254
- call establishment, SIP, 166
- call forwarding, SIP, 233
- call scenarios, H.323, 135–136, 139–142
- call setup
 - MEGACO/H.248, 307–311
 - MGCP, 274–277
- call setup messages, H.323, 131
- call signaling
 - H.225.0, 111
 - H.245 control signaling interaction,
 - 134–135
 - H.323, 129

- call termination, SIP, 188
- call-completion service, SIP, 168
- Call-Proceeding messages, H.323, 132
- called-endpoint signaling, H.323, 141
- calling features, SIP, 233
- CANCEL method, SIP requests, 171
- cancelling registration, H.323, 120
- Capabilities Exchange, H.245, 146
- capacity buffer, network design, 440
- carrier-grade telephony, 2–3
- CAS (Channel Associated Signaling), 314
- cascaded codecs, 100
- CCS (Common Channel Signaling), 314
- centralized conferences, H.323, 109
- chunks, SCTP, 338, 341
- CIC (Circuit Identification Code), ISUP, 326
- circuit switching, 6
- click-to-talk applications, 9
- clients, SIP, 164
- closing logical channels, H.245, 149
- codecs
 - cascaded, 100
 - H.323, 114
 - network design, 442
 - selection, 98–99
 - source, 89
 - waveform, 88
- coding speech. *See* speech coding.
- combining QoS solutions, 434
- command failure error descriptor,
 - MEGACO/H.248, 302
- commands
 - MEGACO/H.248, 286, 300–301
 - MGCP, 258–259
 - command encapsulation, 270–273
 - CRLF, 258
 - encapsulation, 260
 - SP character, 259
- conference bridge access points, MGCP
 - endpoints, 256
- conference calls, H.245, 155
- Connect messages, H.323, 133

- Connection Data field, SDP, 204
- connectionless signaling, SCCP, 318
- connections, MGCP, 255–257
- consultation hold, SIP, 234, 236
- contexts, MEGACO/H.248, 282–283
- control chunks, SCTP, 342–344
- control protocol requirements, media
 - gateway control, 252
- control signaling, H.245, 142
- controlled-load service, QoS, 402–403
- CORBA (Common Object Request Broker Architecture), 450
- CR-LDP, 426
 - message modification, 430–431
 - preemption, 429
 - resource classes, 429
 - traffic characteristics, 427–428
 - traffic engineering, 431–432
- CRCX command, MGCP, 259
- criteria for network design, 439
- CRLF, MGCP commands, 258

D

- D parameter, MGCP, 261
- Data Link layer, OSI model, 28
- data transfer, SCTP, 348
- DCF messages, H.323, 128
- decentralized conferences, H.323, 109
- decomposing H.323 gateways, 158
- delay, 17, 401
- descriptors, MEGACO/H.248, 288
- designing VoIP networks, 438–439
 - alarms, 449
 - bandwidth, 455, 463
 - BHCA, 453
 - blocking probability, 444
 - build-ahead, 440
 - busy hour Erlangs, 451
 - capacity buffer, 440
 - codecs, 442

- DS0 channels, 459
- element management, 449
- fundamental technology assumptions, 440
- LEC interfaces, 457
- maximum simultaneous calls, 465–467
- MGC placement, 461–462
- MGs, 455, 459
- MHT, 452
- nodes
 - availability, 448
 - locations, 455
 - redundancy, 446–448
- overhead, 464
- peaks and binomial distribution
 - function, 465
- per-call bandwidth, 467
- physical connections, 470–471
- product requirements, 446
- product selection, 446
- PSTN trunk dimensions, 455–457
- QoS protocols, 445
- redundancy, 441
- SG placement, 462
- signaling and OA&M bandwidth, 468–470
- silence suppression, 464
- statistics, 449
- traffic distribution forecasting, 453
- traffic forecasts, 450
- vendor selection, 446
- voice activity factor, 443
- voice usage forecasts, 450–451
- voice-coding schemes, 464
- DiffServ (Differentiated Service), 404, 445
 - AF PHB, 408
 - architecture, 405
 - EF PHB, 407–408
 - PHB, 407
 - QoS, 384
 - SLAs, 406
- DigitMap descriptor, MEGACO/H.248, 293
- Discovery messages, LDP, 422
- disengage session, H.323, 127

- distributed gateway systems, 252
- DLCX command, MGCP, 259
- DPCM (differential PCM), 90
- DRJ messages, H.323, 128
- DRQ messages, H.323, 127
- DSCPs (DiffServ Code Points), 445
- DS0 (digital signal 0), 90
 - MGCP endpoints, 255
 - network design, 459
- DTMF digits (dual-tone multifrequency), 101

E

- E parameter, MGCP, 262
- e-commerce, 3
- EF PHB, DiffServ, 407–408
- element management, network design, 449
- EMS (element management system), 445
- encapsulating MGCP commands,
 - 260, 270–273
- encoding tones, 101
- end-to-end QoS, 380
- endpoints
 - H.323
 - admission, 122–124
 - location, H.323, 121
 - registration, H.323, 120
 - status, H.323, 126
 - MGCP, 255–257
 - SCTP, 338
- Entity headers, SIP, 181
- EPCF command, MGCP, 259
- Erlangs, 444
- error codes, MEGACO, 296–297
- Error descriptor, MEGACO/H.248, 296
- ES parameter, MGCP, 261
- event notification, SIP, 214–215
- events, MGCP, 278
- Events descriptor, MEGACO/H.248, 291
- extensions, SIP, 212

F

- F parameter, MGCP, 262
- Facility messages, H.323, 134
- Fast Connect procedure, H.245, 151–153
- FECs (forwarding equivalence classes),
 - MPLS, 414, 445
- filter specs, RSVP, 393
- five nines reliability, telephony, 2
- fixed delay, 401
- flowspecs, RSVP, 391
- forking proxies, SIP, 195–196

G

- G.711
 - handling tones, 101
 - waveform codec, 90
- G.723.1, 94
- G.728, 92
- G.729 codec, 96
- gatekeepers, H.323, 108, 118–119
- gateways
 - H.323, 107, 158
 - media/signaling conversion, 249
- GCF messages, H.323, 118
- general headers, SIP, 181
- global title addressing, SCCP, 318
- GRQ messages, H.323, 118
- guaranteed service, QoS, 401

H

- H.225.0, 111
- H.245, 112
 - ad hoc conference calls, 156
 - bidirectional logical channels, 148
 - Capabilities Exchange, 146
 - closing logical channels, 149
 - conference calls, 155

- control signaling, 134–135, 142
- Fast Connect procedure, 151–153
- logical channels, 144
- master-slave determination, 147
- messages, 144
 - encapsulation, 154–155
- pre-arranged conference calls, 156
- procedures, 145
- unidirectional logical channels, 148
- H.248, 250, 281
 - Add command, 286, 302–303
 - architecture, 281
 - Audit descriptor, 292
 - AuditCapabilities command, 287, 305
 - AuditValue command, 287, 305
 - call setup, 307–311
 - command failure error descriptor, 302
 - commands, 286, 300–301
 - contexts, 282–283
 - DigitMap descriptor, 293
 - Error descriptor, 296
 - Events descriptor, 291
 - LocalControl descriptor, 290
 - Local descriptor, 291
 - Media descriptor, 289
 - messages, 285
 - Modem descriptor, 288
 - Modify command, 286, 302–303
 - Move command, 287, 302–303
 - Multiplex descriptor, 289
 - Notify command, 287, 306
 - Observed Events descriptor, 295
 - packages, 299
 - packages descriptor, 299
 - Remote descriptor, 291
 - response syntax, 301
 - ServiceChange command, 287, 306
 - ServiceChange descriptor, 292
 - Signals descriptor, 291
 - Statistics descriptor, 295
 - Stream descriptor, 290
 - Subtract command, 286, 304
 - Termination State descriptor, 289
 - terminations, 282
 - Topology descriptor, 298
 - transactions, 284
- H.323, 106
 - addresses, 113–114
 - Alerting messages, 132
 - architecture, 107
 - bandwidth change, 125
 - call scenarios, 135–136, 139–142
 - call setup messages, 131
 - call signaling, 129
 - Call-Proceeding messages, 132
 - cancelling registration, 120
 - centralized conferences, 109
 - codecs, 114
 - Connect messages, 133
 - decentralized conferences, 109
 - disengage session, 127
 - endpoints
 - admission, 122–123
 - location, 121
 - registration, 120
 - status, 126
 - Facility messages, 134
 - gatekeepers, 108, 118–119
 - gateways, 107, 158
 - interworking SS7, 374
 - MCs, 108
 - MCUs, 107
 - messages, 112
 - MPs, 108
 - pregranted admission, 124
 - Progress messages, 133
 - protocols, 111–112
 - RAS signaling, 115–116
 - rejecting registration, 120
 - Release Complete messages, 133
 - Request in Progress messages, 129
 - resource availability messages, 128
 - service control messages, 128
 - signaling, 109–110

H.323 (*continued*)
 SIP interworking, 240, 243
 terminals, 107
 zones, 108
 headers
 IP, 34
 SIP messages, 170, 176–178
 TCP, 39
 hybrid codecs, 89

I

I parameter, MGCP, 260
 I2 parameter, MGCP, 263
 IAB (Internet Architecture Board), 31
 IACK messages, H.323, 127
 IANA (Internet Assigned Numbers Authority), 32
 identifiers, MGCP endpoints, 257
 IESG (Internet Engineering Steering Group), 32
 IETF (Internet Engineering Task Force), 31
 IIM messages, M2UA, 366
 IN (intelligent network), 9
 INAK messages, H.323, 127
 INFO method, SIP, 173, 213–214
 instant messaging, SIP, 216–217
 International Softswitch Consortium, 250
 Internet, 26
 RFCs, 32
 standards, 30
 Internet Society, 30
 interworking
 MGCP and SIP, 279
 softswitch and SS7, 369
 Invitation messages, SIP, 185–187
 INVITE method, SIP requests, 171
 IP (Internet Protocol), 3, 27, 33
 headers, 34
 multicasting, 10

packets, 4
 protocol suite, 27–29
 route masks, 36
 routing, 35
 routing tables, 36
 voice transport, 7–8
 IP-based voice services, 10
 IRQ messages, H.323, 127
 IRR messages, H.323, 127
 ISUP (ISDN User Part), 317, 324
 ACM messages, 324–325
 ANM messages, 326
 CIC, 326
 encapsulation in SIP, 370–373
 messages, 317
 REL messages, 326
 RLC messages, 326
 ITU-T Recommendation G.711, 13
 ITU-T Recommendation G.723.1, 94
 ITU-T Recommendation G.728, 92
 ITU-T Recommendation P.800, 81
 IVR access points, MGCP endpoints, 256

J–L

jitter, VoIP, 18
 K parameter, MGCP, 263
 L parameter, MGCP, 261
 labels, MPLS
 distribution, 413, 445
 formats, 414
 stack, 415–417
 label switching, QoS, 385
 layers, OSI model, 28
 LC parameter, MGCP, 261
 LCF messages, H.323, 122
 LD-CELP codec, 92–93
 LDP (Label Distribution Protocol),
 414, 421–425
 LEC interfaces, network design, 457
 LERs (label edge routers), 445

- link alignment, M2PA, 368–369
- LocalControl descriptor,
 - MEGACO/H.248, 290
- LocalDescriptor, MEGACO/H.248, 291
- logical channels, H.245, 144
- lower bandwidth requirements, VoIP, 12
- LRJ messages, H.323, 122
- LRQ messages, H.323, 121
- LSPs, MPLS, 413
- LSRs, MPLS, 417–418

M

- M parameter, MGCP, 261
- M-law, G.711, 90
- M2PA, 367–369
- M2UA
 - IIM message, 366
 - MAUP message, 364–365
 - SCTP, 363
- M3UA, SCTP, 351, 441
 - messages, 354–355, 358–363
 - services, 353–354
- magic cookie, 193
- management messages, M3UA, 357
- managing network elements, 449
- mandatory fields, SDP, 200
- master-slave determination, H.245, 147
- MAUP messages, M2UA, 364–365
- maximum simultaneous calls, network
 - design, 465–467
- MCs (multipoint controllers), H.323, 108
- MCUs (multipoint controller units),
 - H.323, 107
- MD parameter, MGCP, 262
- MDCX command, MGCP, 259
- measuring service level, QoS, 378
- media conversion, gateways, 249
- media descriptor, MEGACO/H.248, 289
- media gateway control
 - control protocol requirements, 252
 - MEGACO, 251
 - MGCP, 251, 254
 - protocols, 253
- Media Information field, SDP, 204
- media negotiation, SIP, 209–210
- media separation, call control, 248–249
- MEGACO, 23, 250, 281, 440
 - Add command, 286, 302–303
 - architecture, 281
 - Audit descriptor, 292
 - AuditCapabilities command, 287, 305
 - AuditValue command, 287, 305
 - call setup, 307–311
 - command failure error descriptor, 302
 - command syntax, 300–301
 - contexts, 282–283
 - descriptors, 288
 - DigitMap descriptor, 293
 - error codes, 296–297
 - Error descriptor, 296
 - Events descriptor, 291
 - LocalControl descriptor, 290
 - Local Descriptor, 291
 - Media descriptor, 289
 - messages, 285
 - Modem descriptor, 288
 - Modify command, 286, 302–303
 - Move command, 287, 302–303
 - Multiplex descriptor, 289
 - Notify command, 287, 306
 - Observed Events descriptor, 295
 - packages, 299
 - packages descriptor, 299
 - Remote Descriptor, 291
 - response syntax, 301
 - ServiceChange command, 287, 306
 - ServiceChange descriptor, 292
 - SignalsDescriptor, 291
 - SIP interworking, 312
 - Statistics descriptor, 295
 - Stream descriptor, 290
 - Subtract command, 286, 304
 - Termination State descriptor, 289

- MEGACO (*continued*)
 - terminations, 282
 - Topology descriptor, 298
 - transactions, 284
- messages
 - H.245, 144
 - encapsulation, 154–155
 - H.323, 112
 - ISUP, 317
 - LDP, 422–425
 - MEGACO/H.248, 285
 - modification, CR-LDP, 430–431
 - RSVP, 388, 396
 - SIP, 169–170
 - headers, 176–178
- methods, SIP requests, 170–171
- MFC signaling (Multifrequency Compelled), 314
- MGCP (Media Gateway Control Protocol), 23, 250, 440
 - A parameter, 260
 - AUCX command, 259
 - AUEP command, 259
 - call control, 254
 - call setup, 274–277
 - calls, 257
 - command encapsulation, 260, 270–273
 - connections, 255–257
 - CRCX command, 259
 - D parameter, 261
 - DLCX command, 259
 - E parameter, 262
 - endpoints, 255–257
 - EPCF command, 259
 - ES parameter, 261
 - F parameter, 262
 - I parameter, 260
 - I2 parameter, 263
 - K parameter, 263
 - L parameter, 261
 - LC parameter, 261
 - M parameter, 261
 - MD parameter, 262
 - MDCX command, 259
 - N parameter, 262
 - NTFY command, 259
 - O parameter, 262
 - off-hook events, 278
 - P parameter, 261
 - packages, 278–279
 - PL parameter, 262
 - Q parameter, 262
 - R parameter, 262
 - RC parameter, 262
 - RD parameter, 263
 - responses, 264
 - return codes, 264, 267–273
 - RM parameter, 263
 - RQNT command, 259
 - RSIP command, 259
 - S parameter, 263
 - SDP session descriptions, 258
 - SIP interworking, 279
 - T parameter, 261
 - X parameter, 263
 - Z parameter, 263
 - Z2 parameter, 263
- MGCs (media gateway controllers), 441
 - placement, network design, 461–462
 - softswitch architecture, 250
- MGs (media gateways), 441
 - backplane systems, 446
 - locating, network design, 455, 459
 - midplane systems, 448
 - softswitch architecture, 250
- MHT, network design, 452
- midplane systems, MGs, 448
- MIME content messages, SIP, 232
- modem descriptor, MEGACO/H.248, 288
- modify command, MEGACO/H.248, 286, 302–303
- MOS (Mean Opinion Score), 16
 - speech coding techniques, 20
 - speech quality, 81–82

move command, MEGACO/H.248,
287, 302–303
MPLS (Multiprotocol Label Switching),
410–413
FECs, 414
label distribution, 413, 445
label formats, 414
label stack, 415–417
LSPs, 413
LSRs, 417–418
QoS, 385
traffic engineering, 419
traffic trunks, 419–420
MPs (multipoint processors), H.323, 108
MSUs (message signal units), SS7, 322
MTP (Message Transfer Part), SS7, 316
MTP2, SS7, 316
MTP3, SS7, 317
multiplex descriptor, MEGACO/H.248, 289

N

N parameter, MGCP, 262
network architecture, SS7, 318–319
network design. *See* designing VoIP
networks.
network entities, SIP, 164
network gateways, signaling/media
conversion, 249
Network layer, OSI model, 28
NMS (Network Management System), 450
node availability, network design, 448
node locations, network design, 455
node-level redundancy, network design,
446–448
nonuniform quantization, speech coding, 88
notification messages, LDP, 426
notify command, MEGACO/H.248, 287, 306
NTFY command, MGCP, 259
Nyquist Theorem, 12, 86

O

O parameter, MGCP, 262
Observed Events descriptor,
MEGACO/H.248, 295
off-hook events, MGCP, 278
OK responses, SIP, 167
optional fields, SDP, 200–201
OPTIONS method, SIP requests, 171, 212
ordering of fields, SDP, 202
Origin field, SDP, 203
OSI model (Open System Interconnection)
IP protocol suite correspondence, 29
layers, 28
OSPF (Open Shortest Path First), 36
areas, 37
AS (autonomous system), 36
backbone areas, 37
overhead, network design, 464
overprovisioning, QoS, 381
overview of IP, 27

P

P parameter, MGCP, 261
packages
MEGACO/H.248, 299
MGCP, 278–279
Packages descriptor, MEGACO/H.248, 299
packet loss, VoIP, 18
packet relay endpoints, MGCP, 256
packets
IP, 4
SCTP, 338
parameters, MGCP, 260–263
payload data chunks, SCTP, 344–345
PCM (pulse code modulation), 90
peaks and binomial distribution function,
network design, 465
per-call bandwidth, network design, 467
performance requirements, SS7, 327

PHB (Per-Hop Behavior), DiffServ, 407
 physical connections, network design, 470–471
 Physical layer, OSI model, 28
 PL parameter, MGCP, 262
 plosive sounds, 84
 policies, QoS, 386
 populating IP routing tables, 36
 port numbers, TCP, 39
 pre-arranged conference calls, H.245, 156
 preemption, CR-LDP, 429
 pregranted admission, H.323, 124
 Presentation layer, OSI model, 29
 prioritizing traffic, VoIP, 19
 procedures, H.245, 145
 product requirements, network design, 446
 product selection, network design, 446
 progress messages, H.323, 133
 protocols
 BGP, 38
 H.323, 111–112
 IP, 3–4, 33
 LDP, 414
 media gateway control, 253
 MGCP, 23, 250, 440
 RSVP, 383
 RTP, 22, 254
 RTSP, 162
 RTP, 384
 SAP, 162
 SCTP, 330
 SDP, 162, 170, 196, 258
 SIP, 23, 162, 248
 SMTP, 40
 TCP, 4, 38
 UDP, 29
 provisional response reliability, SIP, 220–222
 proxies, SIP, 164
 proxy servers, SIP, 164, 191–193
 proxy states, SIP, 194–195
 PSQM (Perceptual Speech Quality Measurement), 82

PSTN trunk dimensions, network design, 455–457
 PSTN-SIP interworking, 236, 239

Q

Q parameter, MGCP, 262
 Q.931 signaling, H.225.0, 111
 QoS (quality of service), 2, 23, 378
 combining solutions, 434
 controlled-load service, 402–403
 CR-LDP, 426
 message modification, 430–431
 preemption, 429
 resource classes, 429
 traffic characteristics, 427–428
 traffic engineering, 431–432
 DiffServ, 404
 AF PHB, 408
 architecture, 405
 EF PHB, 407–408
 PHB, 407
 SLAs, 406
 end-to-end, 380
 guaranteed service, 401
 LDP, 421–425
 measuring service level, 378
 MPLS, 410–411
 architecture, 412–413
 FECs, 414
 label distribution, 413
 label formats, 414
 label stack, 415, 417
 LSPs, 413
 LSRs, 417–418
 traffic engineering, 419
 traffic trunks, 419–420
 policies, 386
 protocols, network design, 445
 RSVP-TE, 433–434
 SLAs, 380

- solutions
 - bandwidth allocation, 382–383
 - DiffServ, 384
 - label switching, 385
 - MPLS, 385
 - overprovisioning, 381
 - resource reservation, 381–383
 - RSVP, 383
 - traffic prioritization, 381
- TCP, 379
- quantization, 87–88
- queuing delay, 401

R

- R parameter, MGCP, 262
- RAC messages, H.323, 128
- RAI messages, H.323, 128
- RAS (Registration, Admission, and Status), 111, 115–116
- RC parameter, MGCP, 262
- RCF messages, H.323, 120
- RD parameter, MGCP, 263
- redirect servers, SIP, 165, 189
- redundancy
 - network design, 441
 - node-level, 446
- REFER method, SIP, 217, 220
- REGISTER method, SIP requests, 171–172
- registrar servers, SIP, 166
- registration messages, SIP, 183–185
- rejecting registration, H.323, 120
- REL messages, ISUP, 326
- Release Complete messages, H.323, 133
- reliability, VoIP, 21
- RemoteDescriptor, MEGACO/H.248, 291
- removing reservations, RSVP, 403
- request headers, SIP, 181
- request in progress messages, H.323, 129
- request-line, SIP messages, 170
- Request-URIs, SIP requests, 170

- requests, SIP, 164
 - ACK method, 171
 - BYE method, 171
 - CANCEL method, 171
 - INFO method, 173
 - INVITE method, 171
 - method tokens, 170
 - OPTIONS method, 171
 - redirect servers, 165
 - REGISTER method, 171–172
 - Request-URIs, 170
- reservations, RSVP
 - errors, 400
 - establishing, 389–390
 - examples, 397, 400
 - removing, 403
- resource availability messages, H.323, 128
- resource classes, CR-LDP, 429
- resource management, SIP, 224–228
- resource reservation, QoS, 381–383
- response headers, SIP, 181
- responses
 - MEGACO/H.248, 301
 - MGCP, 264, 270–273
 - SIP, 165, 173–174
- retry-after headers, SIP messages, 170
- return codes, MGCP, 264, 267–273
- RFCs (Requests for Comments), 32
 - RFC 2044, 198
 - RFC 2198, 102
 - RFC 2327, 198
 - RFC 2543, 162
 - RFC 2705, 253
 - RFC 2833, 102
 - RFC 2885, 253
 - RFC 2886, 253
 - RFC 2960, 337
 - RFC 3015, 253
 - RFC 3261, 162
- RFIs (Requests for Information), 446
- RFPs (Requests for Proposal), 446
- RIP messages, H.323, 129
- RLC messages, ISUP, 326

- RM parameter, MGCP, 263
- route masks, IP, 36
- routing, IP, 35
- routing key management messages,
 - M3UA, 361–362
- routing tables, IP, 36
- RQNT command, MGCP, 259
- RRJ messages, H.323, 120
- RRQ messages, H.323, 120
- RSIP command, MGCP, 259
- RSVP (Resource Reservation Protocol),
 - 383, 445
 - ADSpecs, 394–395
 - filter specs, 393
 - flowspecs, 391
 - messages, 396
 - QoS, 383
 - reservations
 - errors, 400
 - establishment, 389–390
 - examples, 397, 400
 - removing, 403
 - soft state, 403–404
 - traffic control, 388
 - TSpecs, 390–391
- RSVP-TE (traffic engineering),
 - 433–434, 445
- RTSP (Real-Time Streaming Protocol), 162
- RTP (Real-Time Transport Protocol), 22,
 - 102, 254, 384

S

- S parameter, MGCP, 263
- sampling
 - Nyquist Theorem, 12
 - voice, 86
- SAP (Session Announcement Protocol), 162
- scalability of VoIP, 21
- SCCP (Signaling Connection Control Part),
 - 317–318
- SCI messages, H.323, 128
- SCPs (service control points), SS7, 321
- SCR messages, H.323, 128
- SCTP (Stream Control Transmission Protocol), 330, 337, 441
 - adaptation modules, 331–336
 - Adler-32 checksum, 339
 - associations, 338, 346–347
 - chunks, 338, 341
 - control chunks, 342–344
 - data transfer, 348
 - endpoints, 338
 - M2UA, 363
 - M3UA, 351
 - messages, 354–355, 358–363
 - services, 353–354
 - packets, 338
 - payload data chunks, 344–345
 - signaling network architecture, 352
 - streams, 340
- SDP (Session Description Protocol),
 - 162, 170, 196
 - Attributes field, 205
 - Connection Data field, 204
 - descriptions, 228–229
 - mandatory fields, 200
 - Media Information field, 204
 - optional fields, 200–201
 - ordering of fields, 202
 - Origin field, 203
 - session descriptions, 198, 258
 - session-level fields, 199
 - SIP interworking, 206–207
 - subfields, 203
- selecting codecs, 98–99
- sequence numbers, TCP, 40
- servers, SIP, 164–166
- service control messages, H.323, 128
- service level measurement, QoS, 378
- ServiceChange command, MEGACO/H.248,
 - 287, 306
- ServiceChange descriptor,
 - MEGACO/H.248, 292
- session disengage, H.323, 127
- session info description, SDP, 198
- Session layer, OSI model, 29

- session management messages, LDP, 422
- session-level fields, SDP, 199
- setup messages, H.323, 131
- SGs (signaling gateways), SS7,
324, 441, 462
- signaling
 - conversion, gateways, 249
 - H.323, 109–110, 115–116
 - links, SS7, 316
 - network architecture, SCTP, 352
 - OA&M bandwidth, 468–470
 - SIP resource management, 224–228
 - transport in IP networks,
Sigtran, 328–330
- signaling network management
 - messages, M3UA, 357–359
- SignalsDescriptor, MEGACO/H.248, 291
- Sigtran
 - architecture, 330–331
 - signaling transport in IP networks,
328–330
- silence suppression, network design, 464
- SIMPLE (SIP for Instant Messaging and
Presence Leveraging Extensions),
216–217
- SIP (Session Initiation Protocol),
23, 162, 248, 441
 - ACK messages, 167
 - addresses, 175–176
 - architecture, 163
 - BYE messages, 167
 - calls
 - establishment, 166
 - forwarding, 233
 - termination, 188
 - call-completion service, 168
 - calling features, 233
 - clients, 164
 - consultation hold, 234, 236
 - enhancements to specification, 162
 - entity headers, 181
 - event notification, 214–215
 - extensions, 212
 - flexibility, 167
 - forking proxies, 195–196
 - general headers, 181
 - H.323 interworking, 240, 243
 - INFO method, 213–214
 - instant messaging, 216–217
 - invitation messages, 185–187
 - media negotiation, 209–210
 - message headers, 176–178
 - messages, 169–170
 - MGCP interworking, 279
 - MIME content messages, 232
 - network entities, 164
 - OK responses, 167
 - optional user-specified information
fields, 168
 - OPTIONS method, 212
 - provisional response reliability, 220–222
 - proxies, 164
 - proxy servers, 164, 191–193
 - proxy states, 194–195
 - PSTN interworking, 236, 239
 - redirect servers, 165, 189
 - REFER method, 217, 220
 - registrar servers, 166
 - registration messages, 183–185
 - request headers, 181
 - requests, 164
 - ACK method, 171
 - BYE method, 171
 - CANCEL method, 171
 - INFO method, 173
 - INVITE method, 171
 - method tokens, 170
 - OPTIONS method, 171
 - REGISTER method, 171–172
 - Request-URIs, 170
 - responses, 165
 - headers, 181
 - status codes, 173–174
 - SDP interworking, 206–207
 - servers, 164
 - signaling and resource management,
224–228

SIP (*continued*)

- SIP INVITE messages, 166
- softswitch signaling, 251
- UPDATE method, 222–224
- user-agent servers, 165
- SIP-T (SIP for Telephony), 239
- SIP INFO method, SIP requests, 173
- SIP INVITE messages, 166
- SIPits (SIP Interoperability Tests), 162
- SLAs (service-level agreements), 4
 - DiffServ, 406
 - QoS, 380
- SMTP (Simple Mail Transfer Protocol), 40
- SNMP (Simple Network Management Protocol), 450
- socket addresses, TCP, 40
- soft state, RSVP, 403–404
- softswitch architecture, 250–251
- softswitch-SS7 interworking, 369
- sound classifications, speech coding, 82–84
- source codecs, 89
- SP character, MGCP commands, 259
- speech coding, 13, 80
 - AbS codecs, 91–92
 - ACELP codecs, 94–97
 - ADPCM, 90–91
 - cascaded codecs, 100
 - G.711, 90
 - G.729 codecs, 96
 - hybrid codecs, 89
 - LD-CELP codecs, 92–93
 - MOS, 20
 - nonuniform quantization, 88
 - Nyquist theorem, 86
 - quantization, 87
 - selecting codecs, 98–99
 - sound classifications, 82–84
 - source codecs, 89
 - vocoders, 89
 - voice quality, 81–82
 - voice sampling, 86

- VoIP, 13, 20
 - waveform codecs, 88
- speech quality
 - MOS, 81–82
 - VoIP, 17
- SPs (signaling points), SS7, 320
- SS7 (Signaling System 7), 314–315, 441
 - addresses, 323–324
 - interworking
 - H.323, 374
 - softswitch, 369
 - M2PA, 367
 - MSUs, 322
 - MTP, 316
 - MTP2, 316
 - MTP3, 317
 - network architecture, 318–319
 - performance requirements, 327
 - SCPs, 321
 - SGs, 324
 - signaling links, 316
 - SPs, 320
 - SSPs, 321
 - STPs, 320
- SSPs (service switching points), SS7, 321
- standards, Internet, 30–32
- start-line, SIP messages, 169
- statistics, network design, 449
- Statistics descriptor, MEGACO/H.248, 295
- status codes, SIP responses, 173–174
- status of endpoints, H.323, 126
- STD numbers, Internet standards, 33
- STPs (signal transfer points), SS7, 320, 448
- Stream descriptor, MEGACO/H.248, 290
- streams, SCTP, 340
- subfields, SDP, 203
- subject headers, SIP messages, 170
- subtract command, MEGACO/H.248, 286, 304

T

- T parameter, MGCP, 261
- TCAP (Transaction Capabilities Application Part), 318
- TCP (Transmission Control Protocol),
 - 4, 29, 38
 - headers, 39
 - port numbers, 39
 - sequence numbers, 40
 - socket addresses, 40
- TCP/IP, 5
- telephony
 - carrier-grade, 2–3
 - five nines reliability, 2
- terminals, H.323, 107
- Termination State descriptor,
 - MEGACO/H.248, 289
- terminations, MEGACO/H.248, 282
- tone encoding, 101
- tone generation, 101
- topology descriptor, MEGACO/H.248, 298
- traffic characteristics, CR-LDP, 427–428
- traffic control, RSVP, 388
- traffic distribution forecasting, network design, 453
- traffic engineering
 - CR-LDP, 431–432
 - MPLS, 419
- traffic forecasts, network design, 450
- traffic prioritization
 - QoS, 381
 - VoIP, 19
- traffic trunks, MPLS, 419–420
- transactions, MEGACO/H.248, 284
- transmission efficiency, VoIP, 14
- Transport layer, OSI model, 28
- transporting voice via IP. *See* voice transport.
- TSAP identifiers, H.323 addresses, 113
- TSspecs, RSVP, 390–391
- TTL, IP headers, 35

U

- UDP (User Datagram Protocol), 29
- unidirectional logical channels, H.245, 148
- unvoiced sounds, 83
- UPDATE method, SIP, 222–224
- URJ messages, H.323, 120
- URQ messages, H.323, 120
- user-agent clients, SIP, 164
- user-agent servers, SIP, 165
- user-specified information fields, SIP, 168

V

- vendor selection, network design, 446
- video conferencing, 10
- VoATM (Voice over Asynchronous Transfer Mode), 14
- vocoders, 89
- VoFR (Voice over Frame Relay), 14
- voice activity factor, network design, 443
- voice quality, speech coding, 81–82
- voice sampling, speech coding, 86
- voice transport via IP
 - flexibility of IP, 8
 - lower equipment cost, 7
- voice usage forecasts, network design, 450–451
- voice-coding schemes, network design, 464
- voice/data integration, advanced services, 9
- voiced sounds, 83
- VoIP (Voice over IP), 2–4
 - access management, 19
 - carrying voice, 5
 - challenges to meet, 16
 - delay, 17
 - IP as transport mechanism, 6
 - jitter, 18
 - lower bandwidth requirements, 12
 - markets, 15

VoIP (*continued*)

- network design, 438–439
 - alarms, 449
 - bandwidth, 455, 463
 - BHCA, 453
 - blocking probability, 444
 - build-ahead, 440
 - busy hour Erlangs, 451
 - capacity buffer, 440
 - codecs, 442
 - DS0 channels, 459
 - element management, 449
 - LEC interfaces, 457
 - maximum simultaneous calls, 465–467
 - MGC placement, 461–462
 - MGs, 455, 459
 - MHT, 452
 - nodes, 446–448, 455
 - overhead, 464
 - peaks and binomial distribution
 - function, 465
 - per-call bandwidth, 467
 - physical connections, 470–471
 - product requirements, 446
 - product selection, 446
 - PSTN trunk dimensions, 455–457
 - QoS protocols, 445
 - redundancy, 441
 - SG placement, 462
 - signaling and OA&M bandwidth, 468–470

- silence suppression, 464
- statistics, 449
- traffic distribution forecasting, 453
- traffic forecasts, 450
- vendor selection, 446
- voice activity factor, 443
- voice usage forecasts, 450–451
- voice-coding schemes, 464
- packet loss, 18
- reliability, 21
- scalability, 21
- speech coding, 13, 20
- speech quality, 17
- SS7, 315
- traffic prioritization, 19
- transmission efficiency, 14

W

- waveform codecs, 88
- Web collaboration, 10
- wildcards, MGCP endpoint identifiers, 257
- wiretap access points, MGCP, 256

X–Z

- X parameter, MGCP, 263
- Z parameter, MGCP, 263
- Z2 parameter, MGCP, 263
- zones, H.323, 108

ABOUT THE AUTHOR

Daniel Collins has worked in the telecommunications industry for 15 years. He spent approximately nine years with Ericsson in various countries, including Ireland, Australia, the United Kingdom and the United States. During that time he worked extensively with both wireline and wireless network technologies. He helped to develop and deploy 2G wireless systems in Europe; he played a major role in the adaptation of GSM standards for use in the United States; and he was a major contributor to the launch of some of the earliest PCS networks in North America.

Since leaving Ericsson, Daniel has worked for a new telecommunications carrier and, more recently as a consultant. In a consultancy capacity, he has provided wireless and VoIP engineering expertise to numerous network operators, consultancy companies and infrastructure vendors. Daniel's clients include PrimeCo Personal Communications (now part of Verizon Wireless), Synacom Technology, AT&T Wireless, Alcatel USA and several other companies.

Collins is co-author of *3G Wireless Networks*, published by McGraw-Hill. He holds a degree in Electrical and Electronic engineering from the National University of Ireland.

NEW EDITION OF—

CUTTING-EDGE SOLUTIONS CUT-TO-THE-CHASE STYLE

UNANIMOUS 5-STAR ONLINE READER REVIEWS!

"For techs and execs alike ... eloquent and well organized description of VoIP for anyone trying to understand and/or exploit this emerging communications opportunity."

"If you want to work on VoIP this is the ONLY book you need."

"Excellent ... We believe this book to be an essential read, and recommend it highly."

"If you need to understand SIP, H.323, Megaco, or Softswitches, this book is a must."

"Great explanation of the latest VoIP technology ... He uses a lot of call-flow diagrams, which I find particularly useful."

—on the First Edition

In the race to put carrier-quality voice over IP (VoIP) — those using this book run faster and falter less. International expert Daniel Collins' greatly enhanced *Carrier Grade Voice over IP* brings you leading-edge signaling schemes, protocol apps, and quality of service (QoS) techniques — and more important, the answers to making them work together.

Skipping needless history, chitchat, and math, Collins gets right down to solutions with solid information on protocols' purposes and uses. He gives you the newest information with hands-on details, showing you how to:

- Use ITU and IETF protocols to build next-generation networks
- Enable advanced features and services, including expanded SIP coverage
- Provide wireline-quality service with resource reservation schemes and QoS techniques, including the latest information on MPLS advances
- Overcome limitations inherent in IP
- Seamlessly interwork VoIP systems with traditional telephony networks
- Design real-world VoIP networks (new chapter)
- Pull the pieces together into a system that works — and does not break

Complete with numerous specific examples of how the protocols are used and integrated, this just-in-time guide supplies the solutions you need to roll out competitive-quality VoIP.

Jumpstart carrier-quality VoIP with help from international expert Daniel Collins

ISBN 0-07-140634-4



5 5 9 9 5



Cover Design and Illustration: Kevin McGuinness

Visit McGraw-Hill at

www.accessMHtelecom.com

McGraw-Hill

A Division of The McGraw-Hill Companies

