4

Capacity Admission Control

This chapter presents a taxonomy and review of the mechanisms available for capacity admission control in IP networks.

4.1 Introduction

Connection-oriented network technologies such as time division multiplexing (TDM) and asynchronous transfer mode (ATM) have an implicit admission control capability, which is used in establishing the path between sender and receiver, to ensure that there are sufficient resources for the connection. In contrast, IP is connectionless, and has no implicit admission control capability.

In Chapter 3, we described how to deploy the Differentiated Services architecture (Diffserv); Diffserv is by far the most widely deployed IP QOS architecture today, in both enterprise and SP networks. Diffserv effectively provides the capability to manage network capacity on a per-traffic type or class basis. In Chapter 1, we described that the SLAs for IP services are defined in terms of delay, jitter, packet loss rate, throughput, and availability; because the service level experienced by a particular class of traffic is dependent both upon how much capacity has been allocated to that class and upon the current offered load of that class, Diffserv enables differentiated delay, jitter, loss, throughput, and availability commitments to be supported for different classes of traffic.

Diffserv, however, supports no explicit mechanisms for admission control. In order to offer tightly bound service levels for real-time traffic and to assure consistent service within the SLA bounds, admission control mechanisms may be required to ensure that the

actual load for a class does not exceed acceptable levels. Without admission control mechanisms, if there is a chance that the available capacity for a real-time traffic class will be exceeded, then for applications using that class which do not degrade gracefully in the presence of congestion, such as VoIP and packet video, the service for all calls or streams in progress may be degraded. Hence, where admission control is not supported for traffic classes used for applications such as VoIP and packet video, the bandwidth for the class must be over-provisioned with respect to the peak load, in order to ensure that congestion does not occur. Such bandwidth over-provision obviously incurs a financial cost. Further, in practice, it may not always be viable to provision every segment of the network to cope for the peak load. In addition, if network capacity planning and provisioning is not accurate or is not reactive enough to new traffic demands, or in failure situations, there may be instances when congestion is unavoidable; in these cases, all calls or streams in progress will be degraded.

Admission control in general is the process of determining whether a new traffic flow, stream or logical connection may be accepted, taking into account resource and policy constraints. Resource admission control is the decision algorithm, which is used to determine whether a new flow can be granted its requested QOS without affecting those flows already granted admission, such that they continue to maintain their committed service. A resource admission control scheme could potentially consider a number of different resource constraints when processing an admission control decision, such as available CPU resources or memory at devices on the path of the requested reservation, or available class bandwidth on the links on the path. In practice, however, from a QOS perspective the main driver for admission control is to ensure that there is sufficient link or class capacity available at the required service level to accept a new request. It is this aspect of admission control which is the focus of this chapter, and we refer to it as "capacity admission control" as opposed to "call admission control" or "connection admission control." Call admission control has voice specific connotations and could relate to policy-based

admission control as well as resource-based admission control. Further, "connection admission control" is associated with providing policy or resource-based admission control for traffic "trunks," where a "trunk" is an aggregation of traffic from an ingress point to an egress point. The capability provided by "connection admission control" may not be one of real-time admission control, providing feedback to end-system applications on a per-call or session basis, but rather one of capacity management within the core of an IP or MPLS network, operating in the timescales closer to those of service provisioning, rather than call setup; MPLS traffic engineering is used in this context and is described in Chapter 6. Where the generic terms "admission control" or CAC are used in the remainder of this chapter they refer to capacity admission control.

There are a number of approaches to capacity admission control, none of which is universally deployed today. Different deployments, environments, services, and applications pose different requirements and it is not clear that there is a "one size fits all" solution to the problem of capacity admission control. Further, some technologies for admission control are still evolving. Hence, the rest of the chapter provides a taxonomy and review of the different approaches for capacity admission control, with discussion on the applicability and deployment considerations with each approach.

4.1.1 When is Admission Control Needed?

Considered generally, admission control is only practically useful if the following four conditions are met:

i. Without admission control, the offered load may exceed the available capacity

If there is always enough bandwidth for a flow or a class to support the offered load then you simply do not need CAC. Therefore, one approach to providing guaranteed support for services such as voice is to provision sufficient class bandwidth throughout the network to be able to ensure that the peak voice load can be serviced. However,

consideration needs to be given to the limitations of guaranteed bandwidth provisioning during different network failure conditions:

- *Network working case conditions*. If there were insufficient bandwidth to support the peak call load in normal working case conditions, then CAC would be required to cover both working and failure cases.
- *Single network element failure conditions*. If sufficient bandwidth provisioning to cope with the peak call load can be assured in network working case conditions only (i.e. in normal operation with no failures) then in all but the most trivial of topologies (i.e. those that are non-resilient) CAC may be required to cover network element (e.g. link or node) failure case conditions. In this case, during network failures, CAC provides the capability to reject new or rerouted service requests so that those already granted admission continue to maintain their committed service. Without CAC, during failure cases or downtime due to planned maintenance, for example, congestion may occur which can degrade all calls.

  Network planning and provisioning methods may be applied which consider single element failures, ensuring that sufficient bandwidth is provisioned when allowing for all single network element failure conditions. In cases such as this, admission control may not be required.
- *Multiple network element failure conditions*. Even where planning and provisioning take single element failures into account, in some topologies there can be unplanned failure cases (e.g. multiple simultaneous network element failures) where there is insufficient bandwidth to support the service load even though IP connectivity exists. In these cases, CAC may be required.

  If sufficient bandwidth can be provisioned to allow for multiple network element failures then admission control is not required; if connectivity verification shows that connectivity exists, then sufficient bandwidth must exist also. However, in meshed topologies, ensuring that sufficient bandwidth exists in multiple network element failure cases may not be a viable approach. Multiple network

element failure conditions may seem unlikely, however, in most networks elements will be shut down for planned maintenance; a failure during this time may constitute an instance of multiple network element failures.

Hence, network-by-network consideration is required to determine whether the prevalence, duration and impact of events, such as network element failures, which may lead to congestion resulting in service degradation, is sufficient to justify the cost and complexity of deploying admission control mechanisms.

ii. Service utility will degrade unacceptably as a consequence of exceeding available capacity for that flow or class

For some applications, as the bandwidth available to an application flow decreases, the utility of the application also reduces. When browsing the web for example, if the available bandwidth is reduced, the end-user experience may become less satisfactory, but may still be acceptable. Such applications are generally termed elastic applications, examples of which typically include TCP-based applications. An illustrative utility function plot for an elastic application is shown in Figure 4.1.

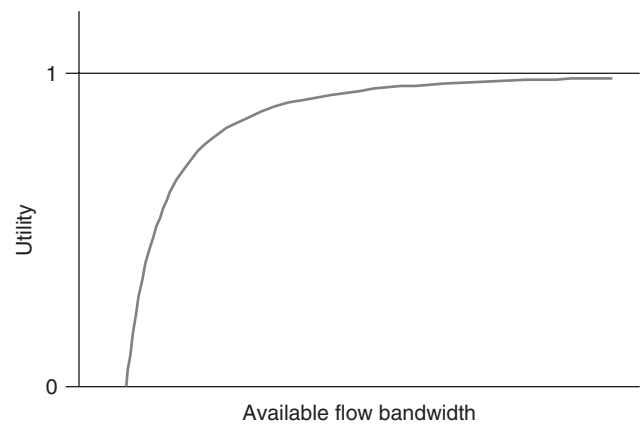

Figure 4.1 Elastic applications' utility function

It is noted that even for elastic applications, there will be some flow bandwidth threshold, below which the utility of the application will be zero, i.e. the application is unusable. Admission control is not generally required for elastic applications. There might be a requirement for admission control for elastic applications if there were the possibility of the application flow bandwidth being reduced to the level of zero utility for a critical elastic application; however, in practice such applications are not prevalent.

There are other applications for which utility is constant above a per flow bandwidth threshold, but when the bandwidth available to the flow falls below an acceptable level, the utility of the application drops to zero. Such applications are generally termed inelastic applications, which typically include VoIP and packet video-based applications. For example, consider a link, which has class capacity to support a maximum of twenty concurrent VoIP calls, within the bounds of the required SLA; if a twenty-first call is allowed to be set up, congestion will occur within that class and the service to all of the calls will be degraded. An illustrative utility function plot for an inelastic application is shown in Figure 4.2.

If there is the possibility of the application flow bandwidth being reduced to the level of zero utility for a critical inelastic application,

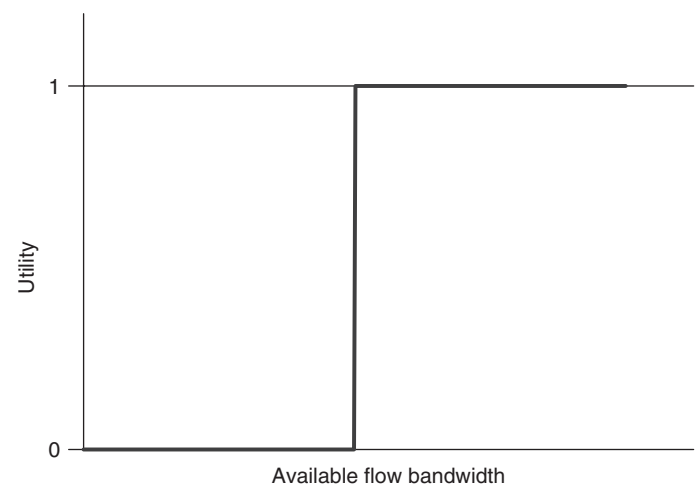

Figure 4.2 Inelastic applications' utility function

then admission control is required to deny a new call or stream if it would reduce the utility of the existing flows, which have already been successfully admitted. If admission control were used in the previous example, it would provide the capability to block the twenty-first call, thereby preventing the existing calls from being degraded and potentially allowing the blocked call to be re-routed where there is sufficient capacity. The mantra for applications which need admission control is that it is much better to refuse a new call than to degrade service for many calls in progress.

In practice, the main drivers for admission control are in support of VoIP and packet video services, both of which are generally inelastic applications. It is noted, however, that there are some VoIP and video which applications may attempt to adjust their rate of sending (and therefore their quality) dynamically based upon the performance they experience from the network (in terms of delay, jitter, and loss), and hence which may be considered in some way elastic. The existence of such applications, however, does not obviate the requirement for admission control. These applications still have minimum bandwidth requirements (i.e. related to minimum quality requirements) and the decision about whether or not admission control is required depends upon whether it is cost effective and practical to ensure that the bandwidth for the flow or class is over-provisioned with respect to the peak load. If it is not, then even for these "elastic" VoIP and video applications, admission control may be required.

iii. The source application knows how to respond to an admission control failure

Admission control is only useful if there is some way of communicating an unsuccessful admission control decision back to the end-system application such that it does not establish the requested flow or stream, and such that it can communicate the failure back to the end-user, e.g. for a VoIP call by returning a busy signal.

iv. It is acceptable from a service perspective to disallow a request

If, from a service perspective, it is not acceptable for admission control to disallow a requested call or session, then rather than CAC, more

bandwidth is needed, e.g. for a residential broadcast video service it would generally be unacceptable to have a CAC failure when simply changing channel.

4.1.2 A Taxonomy for Admission Control

As we discussed at the start of the chapter, there are a number of approaches to admission control in IP networks; there are also a number of criteria by which they could be classified. We classify the main approaches into three general classes, as follows:

- *Endpoint measurement-based CAC*. With endpoint measurement-based admission control approaches, admission control decisions are made by the application end points themselves. The end points measure characteristics of traffic to other destination end points to determine if new streams can be accepted to those end points. This approach to admission control is discussed in Section 4.6.
- *On-path network signaled CAC*. With "on-path" or "path-coupled" network signaled approaches to admission control, network nodes on the media (bearer) path between application end points are responsible for making the admission control decisions. This requires a network level signaling protocol to request and reserve resources along the same path that would be used by media traffic for the requested reservation. Such on-path approaches, which ensure that messages used for QOS signaling are routed only through the nodes on the media path, are implicitly topology-aware. Topology-aware approaches have a dynamic understanding of the network topology and are therefore able to adapt to changes in the available network capacity, due to network element (e.g. link and node) failures, for example.

  There are only two protocols, which are defined or being defined, for on-path signaling of such QOS requests in IP networks: RSVP and NSIS.
  - *RSVP*. The only practical implementations of topology-aware on-path admission control today use RSVP either as per flow

RSVP (as described in Sections 4.4.1–4.4.4) or as in RSVP-TE (as described in Section 4.4.6).
  - *NSIS*. An effort is currently underway within the IETF to standardize a new suite of extensible IP signaling protocols, which can be used for QOS signaling, and which are referred to generically as "NSIS." These are as described in more detail in Section 4.5.

- *Off-path CAC*. With "off-path" or "path-decoupled" admission control approaches, messages used for QOS signaling are routed through nodes that need not be on the data path for the media traffic. Off-path approaches can be either topology-aware or topology-unaware.
  - *Topology-unaware off-path CAC*. Topology-unaware off-path CAC typically consists of applying predefining limits of the available capacity between application endpoint pairs. Being topology-unaware, such approaches have no view of the actual network state, are not able to adapt dynamically to network changes, and hence make inefficient use of the available capacity. This approach to admission control is discussed in Section 4.2.
  - *Topology-aware off-path CAC*. There have been a number of recent developments in off-path topology-aware admission control systems, which are also known as "bandwidth managers" or "resource managers." Being topology-aware, this approach can adapt dynamically to the available network capacity and hence does not suffer the bandwidth inefficiency of topology-unaware approaches. This approach to admission control is discussed in Section 4.3.

This admission control taxonomy is illustrated in Figure 4.3.

In addition to the criteria used for the taxonomy, there are a number of other characteristics and capabilities, which can be used to differentiate different admission control approaches:

- *Layer 3 only?* Some admission control approaches only work in IP environments, e.g. they could not be used to make admission control decisions in layer 2 network environments, such as bridged or switched Ethernet networks, for example.

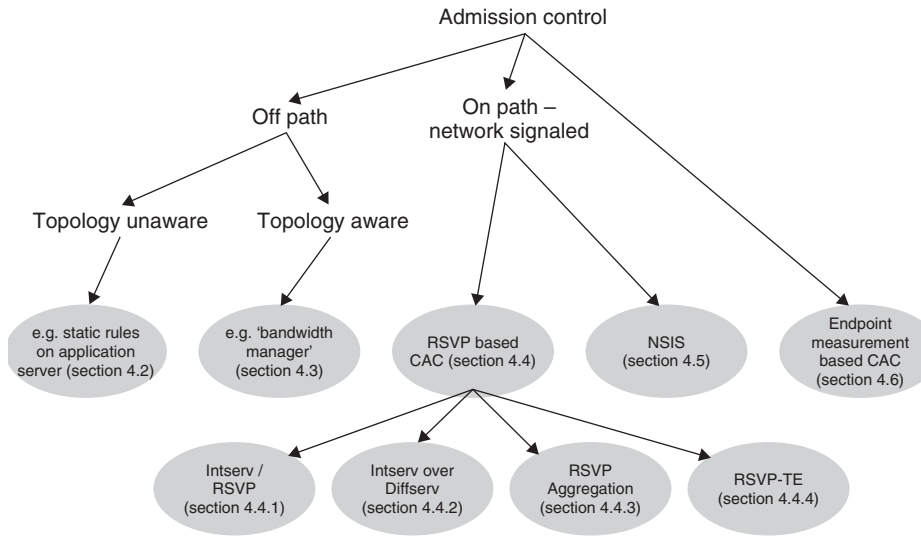

Figure 4.3 IP admission control taxonomy

- *Unicast and multicast?* IP unicast and multicast routing use fundamentally different forwarding paradigms. All admission control approaches described in this chapter support admission control for unicast applications. Although a number of the approaches described could potentially support multicast applications, in practice RSVP is the only approach that currently has this capability.
- *Unidirectional or bidirectional?* Reservations may be unidirectional or bidirectional. Some admission control approaches explicitly support the concept of bidirectional reservations, while other approaches require that bidirectional reservations be modeled as two unidirectional reservations.
- *Control plane or data plane resource reservation?* In Chapter 2, Section 2.1.5, we differentiate between control and data plane QOS functions. Resource admission control is associated with the reservation of resources; if a flow, stream, or connection is successfully admitted, then if that admitted flow's requirements are taken into account before accepting new reservations, by implication resources

were reserved for the flow. Depending upon the particular admission control approach that is used, resource reservation can be a control plane only function or both a control plane and data plane function, which for example configures the data plane QOS functions to guarantee resources for successful admissions. Further, depending upon whether the source is considered trusted or untrusted, resource reservation may also be combined with the configuration of data plane conditioning mechanisms to ensure the source is only sending what it is permitted to send.

A summary of the key characteristics and capabilities of the different admission control approaches is provided in Section 4.7.

4.1.3 What Information is Needed for Admission Control?

Whichever admission control approach is used, there is a common set of information needed in order to make an admission control decision:

- *From where to where?* For resources to be reserved, clearly there must be some information to define where the reservation is from and to. For IP-based approaches, this is normally defined by the source and destination IP addresses of the requested reservation.
- *Resources required?* The problem of admission control may often be stated as *"is there enough capacity to support the requested reservation?"* In practice, however, the admission control decision can use parameterized or measurement-based approaches to specify the resources required and determine if sufficient resources are available. The differences between parameterized and measurement-based algorithms are considered in Section 4.1.4.
- *At what service level?* A request for a reservation needs to define the service level at which the prospective reservation is requested. If reservations are being made in a Diffserv network for example, the reservation request could state whether the request is for expedited

forwarding (EF) or assured forwarding (AF) resources at each hop. The Integrated Services architecture (Intserv) defines the guaranteed and controlled load service types (see Section 4.4).

- *At what pre-emption priority level?* Some admission control schemes may optionally support the notion of pre-emption, with an associated pre-emption level with each reservation request. If there were insufficient capacity available to accept a new reservation request in addition to all of the existing reservations in progress, this would enable a request with a higher pre-emption priority to be able to pre-empt reservations of a lower priority. This could be used to allow emergency service calls, such as defined in [RFC 4542], to pre-empt standard calls, for example.

4.1.4 Parameterized or Measurements-based Algorithms

With many admission control approaches, there is a choice as to the possible admission control algorithms they can use and whether those algorithms are parameterized or measurement-based; hybrids, which rely on a combination of both approaches, are also possible.

4.1.4.1 Parameterized Algorithms

Parameterized approaches to admission control use resource accounting, in order to make an admission control decision. With such approaches, parameterized traffic descriptors are used to represent resource requirements for requests and a comparable descriptor is used to represent the available resources. The admission control system maintains state information detailing the requests that have been accepted and the remaining resources. When a new request is received, the traffic descriptor for that request is aggregated with those for requests previously admitted (and which are still in progress) and the result is compared against the descriptor of available resources to determine if the new request can be accepted. The performance of such an approach depends upon the accuracy of the parameterized descriptor used to represent traffic requirements and available resources.

The simplest parameterized resource descriptor uses a single variable to describe the traffic profile of the requested reservation, which could represent the peak rate of the reservation, for example. In this case, if the total available capacity (i.e. if no reservation existed) is defined by a (it is presumed that this limit defines the bounds at which the required QOS can be met), the reserved capacity is defined by r, and the capacity requested by a new reservation is n, the new reservation is accepted if the following condition is true:

$$r + n <= a$$

Using a peak rate traffic descriptor, however, makes no allowance for variation in the traffic profile over the duration of the reservation and hence allows no provision for statistical multiplexing gain. For constant bit rate applications this will not be an issue; however, for variable bit rate applications this may result in inefficient use of bandwidth. For variable bit rate applications the variation in traffic profile may be taken into account using a token bucket traffic descriptor, for example, with a specified average rate and a maximum burst characteristic to define the traffic profile of the requested reservation. It can be shown, however, that flows with different traffic profiles could share the same average rate and burst characteristics, and hence functions that aggregate these parameters across a number of flows may not produce a result that accurately reflects the profile of the traffic aggregate. The result of this effect is either that the aggregation functions either need to be conservative (i.e. pessimistic), and hence statistical multiplexing gain is reduced, or inaccuracies may lead to an incorrect decision to accept a new request when insufficient resources are available, and hence SLA guarantees may be violated. This is likely to be more of an issue with a relatively small number of flows; however, where the law of large numbers applies the probability of this issue occurring is low. Further, more complex traffic profile descriptors and aggregation functions are possible [KNIGHTLY], which aim to provide both a reasonable statistical multiplexing gain and statistical QOS guarantees.

4.1.4.2 Measurement-based Algorithms

Measurement-based admission control (MBAC) algorithms rely on using measurements of characteristics, such as the delay, jitter, loss, or utilization from traffic or elements on the path between two end-systems in order to determine whether to accept new reservation requests. MBAC algorithms can use measurements taken either from application end points, or from intermediate nodes on the data path between end-systems. Endpoint-based MBAC approaches can rely either on active monitoring or on passive monitoring of media traffic, and are considered further in Section 4.6.

MBAC approaches that use measurements from intermediate nodes on the data path between end-systems use passive measurement of statistics such as link or class utilizations, in order to estimate whether there is sufficient capacity available to accept a new request. With the simplest measurement-based approach, if the total available capacity (i.e. if no reservation existed) is defined by a (it is presumed that this limit defines the bounds at which the required QOS can be met), the measured load over the past measurement interval is defined by m, and the capacity requested by a new reservation is n, the new reservation is accepted if the following condition is true:

$$m + n <= a$$

More complicated MBAC algorithms are described in [BRESLAU1, JAMIN]. A benefit cited for measurement-based algorithms over parameterized algorithms is that they can achieve higher levels of network utilization (and hence greater statistical multiplexing gain) while meeting user quality of service requirements, as they do not require a traffic descriptor for each reservation and therefore do not suffer the potential issues associated with the aggregation of these traffic descriptors described in the preceding section.

The fundamental assumption, however, that MBAC algorithms are found upon is that measurements taken over the past measurement interval can be used to make accurate admission control decisions in the next measurement interval. This might be the case for high-speed links, such as core links, where a large number of flows are aggregated;

however, in cases where a small number of concurrent flows can cause congestion, this assumption is clearly incorrect. If, for example, multiple flows share a resource, but that resource only has sufficient capacity for a single flow at any point in time, then it is possible for multiple end points to determine that there is capacity available based upon measurements from their past measurement intervals. As a result, they may all start sending within their coincident current measurement intervals, with the result that congestion will occur and their QOS guarantees will be violated. Further, if the state of the network has changed since the last measurement interval, due to a network element failure, for example, then clearly end points may make potentially incorrect decisions to accept new reservations, based upon measurements that are no longer representative of the network's state. Therefore, measurement-based admission control approaches cannot deterministically ensure that congestion does not occur and hence that QOS guarantees will be assured; this is recognized in [BRESLAU1]:

> *. . . traffic measurements are not always good predictors of future behavior, and so the measurement-based approach to admission control can lead to occasional packet losses or delays that exceed desired levels.*

Further, from [JAMIN], which proposed "A measurement-based admission control algorithm for Integrated Service packet networks":

> *Measurement-based approaches to admission control can only be used in the context of service models that do not make guaranteed commitments, such as the [Integrated Services] controlled-load service model.*

Consequently, measurement-based admission control approaches are not generally used for real-time applications such as voice and video (and which would be supported by the Integrated Services guaranteed service model, as described in Section 4.4), which are the main applications demanding admission control today, hence measurement-based admission control is not widely deployed in practice.

4.2 Topology-unaware Off-path CAC

Topology-unaware off-path CAC typically consists of applying predefined limits of the available capacity between application end point pairs. Such approaches can be implemented in a distributed manner – on each VoIP gateway as a limit of the number of calls to the other gateways, for example – or in a centralized bandwidth management function, which could be an application server (e.g. video server, call server etc.) or policy server. However this approach is implemented, when a new request is received, it is compared against the currently available capacity between that pair of end points. If sufficient capacity exists, the request is admitted and the available capacity is updated accordingly. If a request would result in the capacity limit between that particular end point pair being exceeded, it would be denied.

Consider the example shown in Figure 4.4. In Figure 4.4, the call server maintains a table of available capacity, in terms of number of calls, from each voice gateway to every other voice gateway. If a user

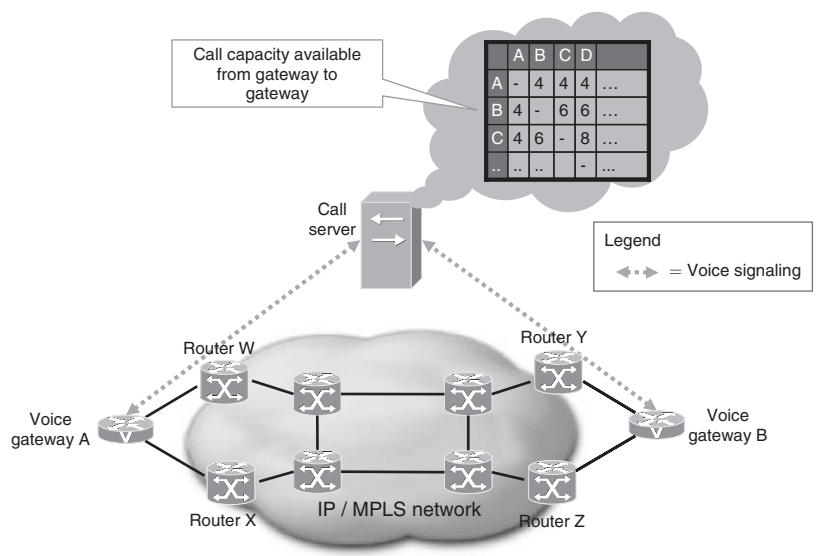

Figure 4.4 Topology-unaware off-path CAC: call server example

connected to voice gateway A attempts to place a call to a user connected to gateway B, in processing the call, the call server performs a lookup in the admission control table. Assuming that there is sufficient capacity available both from A to B, and from B to A as the call is bidirectional, the call will be allowed and the available call capacity in the admission control table would be reduced accordingly. If there were insufficient capacity available for the call to the particular destination, it could be blocked or could potentially be rerouted at the application level to another destination gateway where capacity was available. When the call ends, the admission control table is updated to reflect the consequent increase in available call capacity to the respective destination gateway. This example uses number of calls as the measure of capacity, which may be an acceptable approach where all calls are the same rate; however, where calls can be different rates, a better approach would be to maintain a matrix of the available bandwidth between gateways, and to compare that against the requested bandwidth for new calls.

Topology-unaware off-path CAC represents one of the simplest forms of admission control, but inevitably, it has a number of limitations. This approach may be effective in simple topologies; however, the key issue with all topology-unaware admission control approaches in general is that they do not consider the availability of resources along the specific network path that would be impacted by the request and cannot adapt in real time to changes in network capacity, caused by link or node failures for example. Therefore, in networks with resilient paths, the threshold values used in the tables of available admission control capacity need to defined taking network element failures into account, in order to ensure that such failures will not cause situations where a call is allowed to be placed but there is actually no network capacity available to support the call. In normal network conditions, when there are no failures, these low admission control thresholds do not reflect the state of the network capacity and hence result in inefficient use of available capacity. Consider the access links to gateway A in the example in Figure 4.4: the table shows 4 calls-worth of available capacity between gateway A and gateway B; for this value to be accurate in single element (link or node) failure cases then both of

the access links to gateway A would need to be able to support 4 calls independently, i.e. in normal working case conditions site A would be capable of supporting 8 calls in total. If this is the case, but both links are working, then only half of the available capacity can be used. Further inefficiencies may occur where topology-unaware off-path admission control is implemented in a distributed manner, but where the different admission control systems (e.g. different call servers) can share the same network resources and where each system has no visibility of the bandwidth currently reserved by the other systems.

To limit the capacity inefficiencies of topology-unaware off-path admission control approaches, as the network evolves, the tables of available capacity need to be updated accordingly. In large meshed topologies, the ongoing calculation and maintenance of these tables could become a significant operational overhead.

Approaches, which are able to adjust dynamically to changes in the network topology in real-time, overcome these issues. In practice, off-path topology-unaware admission control approaches are only generally used in simple deployments, for example to perform CAC for non-resilient access link connections.

4.3 Topology-aware Off-path CAC: "Bandwidth Manager"

Topology-aware off-path admission control systems, which are also known as "bandwidth managers" or "resource managers," act as an intermediary between the application control plane (e.g. call server, video on demand server etc.) and the network control plane, as shown in Figure 4.5. Such systems track the status of the network and provide the capability to process topology-aware admission control decisions on a per-call or per-stream basis. Being topology-aware, bandwidth manager-based approaches can adapt dynamically to the available network capacity and hence do not suffer the bandwidth inefficiency of topology-unaware approaches. Such off-path topology-aware admission control approaches could potentially provide a solution for most deployment environments: for both the access and for the core, for L2 and L3, for IP and MPLS.

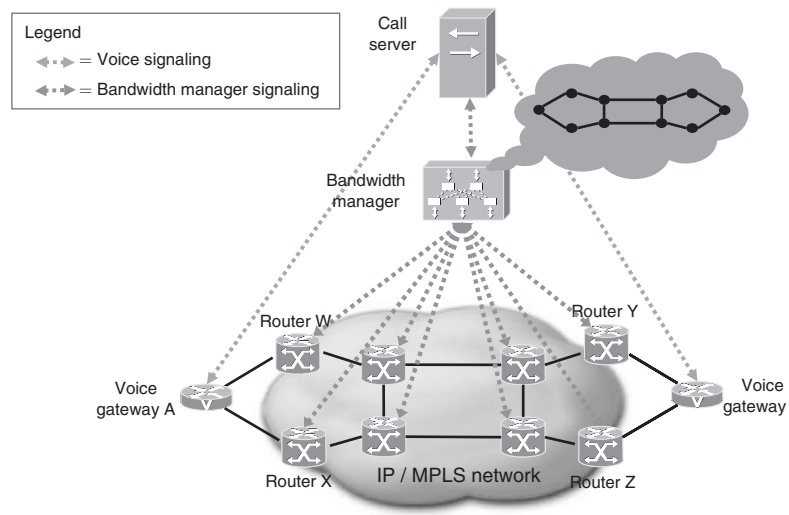

Figure 4.5 Bandwidth manager – topology-aware off-path CAC: call server example

The bandwidth manager maintains a dynamic topology map of the available network bandwidth resources, which in the context of a Diffserv deployment can be maintained on a per-class (service) basis. There are a number of ways that this topology map could be populated, including via an interface to another OSS system in order to extract the required information, or via a discovery process using Telnet/CLI, SNMP or other device protocols, or by participating in routing protocols such as OSPF, ISIS, or BGP. The bandwidth manager also maintains a mapping between these network bandwidth resources and IP addresses of application end points, which may be derived dynamically, from router's routing tables, for example. As requests for resources are received, the bandwidth manager uses the mapping to resolve the addresses of the end points that are passed in the requests to determine which underlying bandwidth resources are impacted by the request. The bandwidth manager verifies that sufficient bandwidth is currently available to satisfy the request; this could be based on bandwidth accounting or potentially use passive measurement statistics retrieved from devices on the data path for the requested reservation. The bandwidth manager then admits or

denies the request as appropriate, replies to the application request and updates the bandwidth resource map accordingly.

Effectively the bandwidth manager function is an area of network policy control, making dynamic policy decisions based upon the availability of network bandwidth resources. "Bandwidth managers" can also be considered a practical realization of a subset of the "bandwidth broker" functionality outlined in [RFC 2638]. In addition to performing admission control – which is the key functionality performed by a "bandwidth manager" – the "bandwidth broker" may apply data plane conditioning policies at the ingress points to the network for the requested reservation, and may also perform interdomain communication with bandwidth brokers of adjacent domains.

Off-path resource and bandwidth management functionality has been defined in a number of standards bodies, including the Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) working group within the European Telecommunications Standards Institute (ETSI) [RACS], which currently addresses admission control for the access network and the Multi Service Forum (MSF), which currently addresses admission control for the core network [MSF-TR-ARCH-005-FINAL].

The details of the bandwidth manager operation are most easily illustrated with an example. There are differences in the detail of the different standards in this area and some of the standards are still evolving; hence, the example in the following section is designed to illustrate some of the concepts and considerations of bandwidth manager operation, rather than rigidly representing a particular standard's implementation.

4.3.1 Example Bandwidth Manager Method of Operation: Next Generation Network Voice CAC

With the move to so-called "next generation networks" (NGNs), conventional public switched telephony network (PSTN) voice services are being migrated to IP/MPLS networks. The connection-oriented network technologies traditionally used to support PSTN services

implicitly have an admission control capability. One way to provide an equivalent capability with NGNs is by augmenting an IP/MPLS network with a bandwidth manager function, as described in this example.

It is noted that the bandwidth manager could be used in conjunction with a number of different network connectivity models. The choice of network connectivity model has an impact on the characteristics of the resultant admission control system, and affect how the bandwidth manager models and tracks the network topology.

- *IGP-based networks*. In this context, we refer to "IGP-based networks" as networks where forwarding decisions at each hop are determined by an interior gateway routing protocol (IGP), such as OSPF or ISIS, and hence where there is no implicit connection orientation, i.e. no end-to-end signaling function is used to set up the data paths.

  In IGP-based networks, it is possible for the bandwidth manager to participate passively in the IGP routing protocol process in order to model the network topology and to be able to predict the route within the network that a flow would take between two points. If such as system also has a view of the capacity on each of the network's links (either per-class or on aggregate) then it is possible that the system could perform network capacity admission control, receiving and processing requests for network bandwidth reservation and tracking the available capacity. It is noted, however, that the accuracy of such a system is dependent upon how accurately the bandwidth manager predicts the actual behavior of the network; inaccuracies will occur if there are multiple paths with the same IGP metric cost between two end-systems and the bandwidth manager does not model equal cost multipath (ECMP) algorithms (which determine how traffic is load-balanced over the equal cost paths) correctly, for example.

  Further, with conventional IGP routing, in the time interval immediately following network element failures, each router behaves autonomously and hence it is possible that the IGP will reroute traffic affected by the failure, before a new admission control decision can be made by the bandwidth manager, hence transient

congestion following network failures is possible with an IGP connection model. This issue is not specific to bandwidth manager deployments, but rather applies to any admission control approaches when they are used with a connectionless IGP-based IP or MPLS network.

- *MPLS traffic engineering-based networks*. MPLS TE (see Section 4.4.6) is implicitly connection-oriented and MPLS TE tunnels provide an explicit routing and signaled network level CAC capability. These capabilities can be used to overcome the issue of transient congestion following network element failures, which can happen with IGP-based network deployments, because re-routing of MPLS TE tunnels cannot happen before a network level admission control decision is made.

  Further, the use of TE tunnels abstracts the bandwidth manager from the task of modeling the detail of the physical network topology. When used in conjunction with a TE deployment the bandwidth manager just needs to track the status of logical TE tunnels and their available bandwidth.

A more detailed discussion on the characteristics of different network level connectivity models when they are used in conjunction with a bandwidth manager is provided in [MSF-TR-ARCH-008-FINAL].

The differences between these approaches highlight the importance of deciding whether CAC is required to cover working case or network failure case scenarios in a particular network deployment; this not only determines whether or not CAC is needed, but also determines which network connectivity models are required in conjunction with the bandwidth manager. If CAC is needed to cover normal working case conditions, and transient congestion following network failures is acceptable, then the combination of a plain IGP deployment in conjunction with a bandwidth manager may be acceptable. If, however, transient congestion following network failures is not acceptable, then MPLS TE is needed in conjunction with the bandwidth manager; MPLS TE also provides capabilities other than CAC, as discussed in Chapter 6, Section 6.2.3.

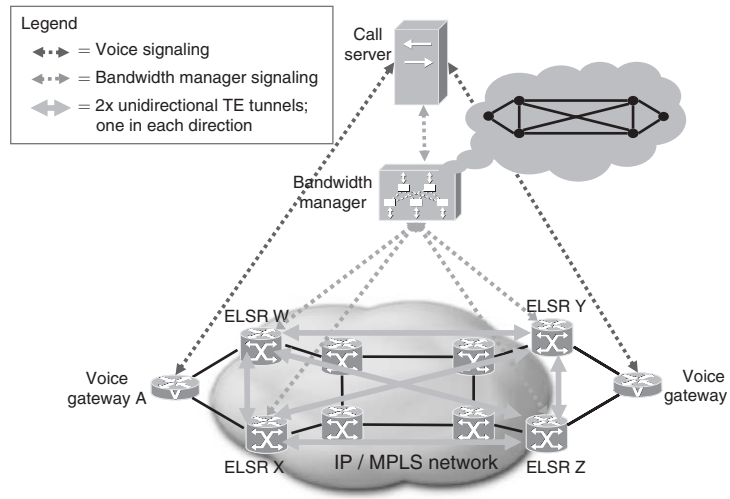

Figure 4.6 Example network topology

For this example, we assume an MPLS TE deployment, as shown in Figure 4.6; Gateway A and gateway B each resiliently connect to respective pairs of MPLS edge label switched routers (ELSRs). A full mesh of TE tunnels is configured to interconnect all of the E-LSRs (for simplicity two unidirectional tunnels are represented with a single bidirectional arrow in Figure 4.6).

The bandwidth manager maintains a map of the key network bandwidth resources needed to make valid admission control decisions, such as contended access connections, core TE tunnels etc. This approach abstracts the bandwidth manager from the detail of modeling the entire network state; TE enables this abstraction for the core. For example, an abstracted bandwidth manager representation of the network from Figure 4.6 is shown in Figure 4.7.

The representation in Figure 4.7 shows symmetrical bidirectional bandwidth resources; however, there is no requirement for bandwidth symmetry and this approach could work with asymmetrical bandwidth resources.

The interaction between the application server, e.g. call manager, and the bandwidth manager will be dependent upon the particular

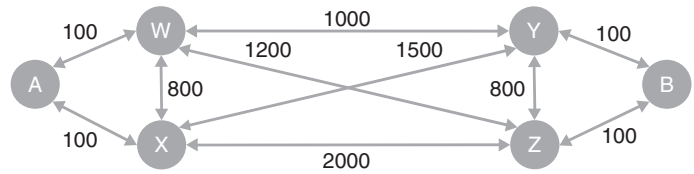

Figure 4.7 Possible bandwidth manager network bandwidth representation

application-signaling model used. In this example, we assume that the session initiation protocol (SIP) [RFC 3261] is used for call signaling, and that Diameter [RFC3588] is used between the call server (in this case SIP proxies) and the bandwidth manager. Diameter is specified as the protocol used for this function by both ETSI/TISPAN [Gq'] and the MSF [MSF2005.187]. Consider the simplified bandwidth manager call flow for a successful two party call setup and tear down shown in Figure 4.8 and the sequence of events that follows.

The call sequence of events is as follows:

- *Steps 1–4*. Conventional call signaling is used to set up a call from SIP End point_A (e.g. Voice Gateway A) to End point_B (e.g. Voice Gateway B).
- *Step 5*. The call server requests admission for a unidirectional call to be set up from Gateway A to Gateway B. The request will contain at least the following information:
  1. IP address of source gateway A
  2. IP address of destination gateway B
  3. Bandwidth requested for the call (or could be the CODEC used)
  4. Call identifier.

  Although this particular application-signaling model uses two separate unidirectional requests to the bandwidth manager per call, some application-signaling models may use a single bidirectional request per call. Unidirectional reservations may also be required for some applications other than voice, such as video-on-demand streams.

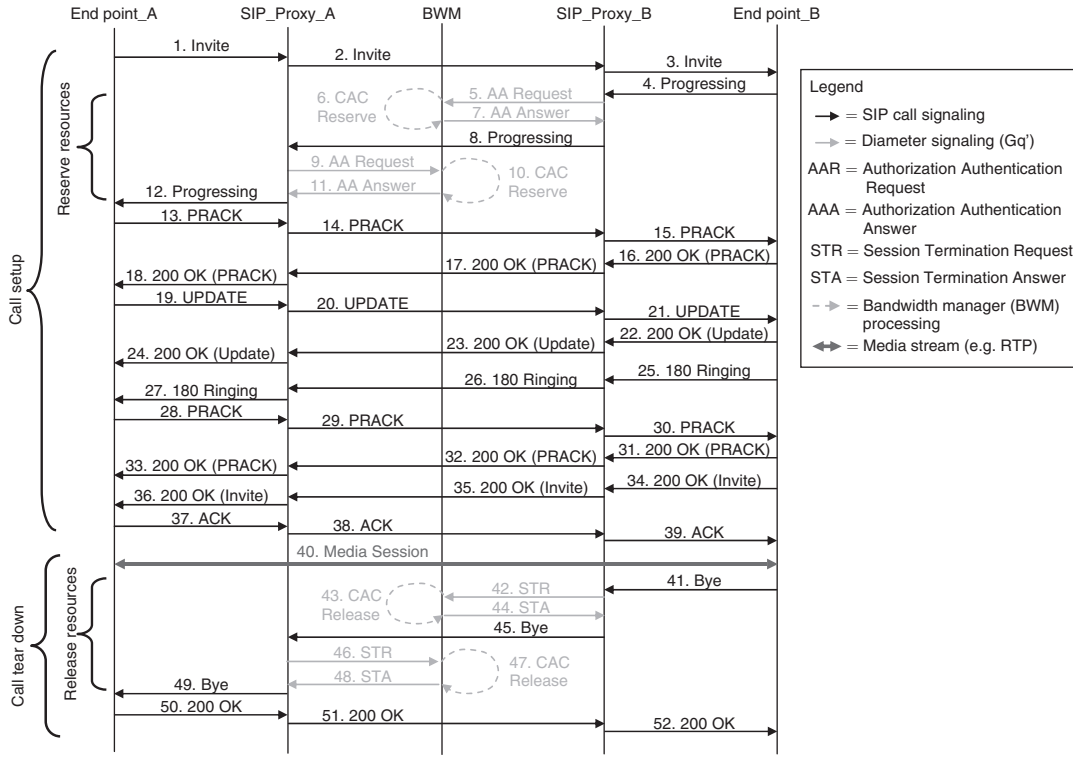

Figure 4.8 Bandwidth manager call flow: basic SIP call

- *Step 6*. Upon receiving the request, the bandwidth manager uses the source and destination gateway addresses to determine which underlying bandwidth resources will be impacted by the request. To do this the bandwidth manager uses an offline path computation function to determine which resources are impacted by the call; this path computation function may rely on IGP or BGP routing information to simulate the routing within the network. In this example, we assume that the path computation function determines that the following resources are impacted by the call:
  - the access link from Gateway A to Router_W
  - the TE tunnel from Router_W to Router_Y
  - the access link from Router_Y to Gateway B.

The bandwidth manager verifies that sufficient bandwidth is available on these resources to support the request. This could potentially use a parameterized or a measurement-based approach, as described in Section 4.1.4. At this stage, the bandwidth manager could also apply policy decisions, for example, that no more than 90% of resources will be used by normal services, allowing headroom for emergency services, e.g. such as defined in [RFC 4542].

- *Step 7*. Assuming that sufficient bandwidth is available to support the call the bandwidth manager replies positively to the call server.

  If there were insufficient bandwidth available on one or more of the resources affected by the call then a number of actions may be performed by the bandwidth manager:
  - In the simplest case, the bandwidth manager is aware of the resources in the network, tracks the state of those resources, and manages admission control decisions into available bandwidth accordingly. In this case, if there is insufficient bandwidth to support the request, the bandwidth manager would reply negatively to the call server, and the call would be cleared down as a result. Considering the TE tunnels specifically, either the TE tunnel head-end routers or an offline system, known as a tunnel server or path computation element (PCE) (as being defined by the PCE working group within the IETF [PCE]) would be responsible for tunnel path calculation (see Chapter 6). The head-end routers or PCE perform a constraint-based shortest path first (CSFP) computation to pick the least cost path that satisfies the configured tunnel constraints, including available bandwidth. The bandwidth manager would control admission to the TE tunnel bandwidth and need track only the status of TE tunnels. SNMP, for example, could be used by the bandwidth manager to discover what tunnels are configured on a head-end router and to track the status of tunnels and access network resources.
  - A more complicated bandwidth manager implementation could see the bandwidth manager attempt to trigger the resizing of some of the resources dynamically, if possible, such that they could

support the requested call. Clearly, an access link may represent a physical bandwidth constraint, which may not be able to be resized dynamically; the bandwidth manager may be able to trigger the resizing of a TE tunnel, however.

For example, when a request is received, the bandwidth manager resolves the request to the underlying resources impacted by the request. If there is sufficient bandwidth available in an impacted tunnel to support the call then the call is admitted; sufficient bandwidth in this context means that the currently allocated tunnel bandwidth is below a defined threshold. If there is not sufficient bandwidth to support the call, then the bandwidth manager may attempt to trigger the resizing (i.e. increasing) of the tunnel bandwidth; the call will be admitted or denied based upon the success or failure of the tunnel resizing. More complicated resizing regimes are possible, e.g. if the currently allocated tunnel bandwidth is greater than 80% utilized, but there is sufficient bandwidth to accept the call, then accept the call AND attempt to increase the tunnel bandwidth by 20%. Similar approaches could be applied to downsize tunnels when calls are cleared.

- *Step 8*. Call signaling continues.

- *Step 9*. The call server requests admission for a unidirectional call to be set up from Gateway B to Gateway A. The request will contain at least the following information:
  1. IP address of source gateway B
  2. IP address of destination gateway A
  3. Bandwidth requested for the call (or could be the CODEC used)
  4. Call identifier

- *Step 10*. Upon receiving the request, the bandwidth manager uses the source and destination gateway addresses to determine which underlying bandwidth resources will be impacted by the request and verifies that sufficient bandwidth is available on these resources to support the request. In this example, we assume that the path

computation function determines that the following resources are impacted by the call:
  - the access link from Gateway B to Router_Y
  - the TE tunnel from Router_Y to Router_W
  - the access link from Router_W to Gateway A.

- *Step 11*. Assuming that sufficient bandwidth is available to support the call, the bandwidth manager replies positively to the call server.

- *Steps 12–40*. Call signaling continues. It is noted that the destination phone rings only after the available bandwidth is confirmed in both directions. User B picks up the phone and the media session is successfully established between gateways A and B.

- *Step 41*. User connected to Gateway B hangs-up and call signaling starts to clear the call down.

- *Step 42*. The call server requests that the bandwidth manager clears the reservation from Gateway A to Gateway B for the call.

- *Step 43*. The bandwidth manager clears the reservation from Gateway A to Gateway B.

- *Step 44*. The bandwidth manager confirms that the reservation has been cleared to the call server.

- *Step 45*. Call signaling continues to clear the call down.

- *Step 46*. The call server requests that the bandwidth manager clears the reservation from Gateway B to Gateway A for the call.

- *Step 47*. The bandwidth manager clears the reservation from Gateway B to Gateway A.

- *Step 48*. The bandwidth manager confirms that the reservation has been cleared to the call server.

- *Steps 49–52*. Conventional call signaling continues and the call is cleared down.

4.4 The Integrated Services Architecture/RSVP

[RFC1633] laid out the philosophy of the Integrated Services or "Intserv" IP QOS architecture. Intserv defines an architecture that supports admission control and resource reservation/allocation in IP networks. It was designed to address the issues identified with IP precedence and type of service (see Chapter 2, Section 2.3.2, providing the capabilities needed to support applications with bounded SLA requirements, such as VoIP and video. Intserv tackles the problem of providing services level assurances to applications by explicitly managing bandwidth resources and schedulers on a per flow basis; resources are reserved and admission control is performed for each flow.

Intserv is defined by the following key facets:

- *Classification*. With Intserv, classification is performed on a per flow basis; at each Intserv capable router, complex classification is performed to identify a particular flow using the 5-tuple of source and destination IP addresses, source and destination UDP/TCP port numbers and IP protocol number. This requires per flow data plane state at each Intserv hop.
- *Scheduling*. Intserv requires that scheduling resources are also managed on a per flow basis, in order to ensure that the application requirements for that flow are met. This does not mean that scheduling resources (i.e. queues) have to be provisioned on a per flow basis although they can be. Alternatively, a number of flows may be mapped into a class, which is serviced from a single queue; all packets in the class will then get the same treatment from the scheduler.
- *Admission control*. In order to provide guarantees to each flow, admission control is performed at each hop to ensure that there are sufficient resources available to meet the requirements of the flow. If there are sufficient resources, the flow is admitted, else the flow is rejected. This requires per flow signaling and per flow control plane state at each Intserv hop.

In theory, a number of potential mechanisms could be used to perform admission control and to set up the per flow classifiers and queuing resources associated with an Intserv reservation. This could conceivably be via a centralized management system or via an end-to-end signaling protocol; signaling protocols exchange information between nodes to establish, maintain, and remove control plane state. In practice, however, the only way that Intserv has been implemented is using the resource ReSerVation Protocol (RSVP) as the end-to-end signaling protocol used to set up the Intserv reservation. RSVP was designed to support Intserv, hence Intserv has become synonymous with RSVP.

4.4.1 RSVP

RSVP [RFC2205] is defined by the following key characteristics:

- *IP Protocol*. RSVP does not use a transport layer protocol but rather is identified by IP protocol number 46.
- *Unidirectional reservations*. RSVP provides the capability to establish unidirectional reservations; if bidirectional reservations are required then two RSVP reservations are required, one in each direction.
- *Unicast and multicast*. RSVP supports reservations for both unicast and many-to-many multicast traffic. RSVP has capabilities for state merging and different classification filter styles to support reservations over a multicast distribution tree. Although RSVP supports multiple senders and receivers in support of many-to-many multicast applications, throughout the rest of this section we refer to sender and receiver (singular).
- *Receiver-initiated reservations*. RSVP reservations are initiated by the receiver; to instantiate a reservation, an RSVP receiver sends an RSVP reservation request (Resv) message upstream toward the sender. Each RSVP capable router receiving a Resv message creates reservation state and forwards the message to the next upstream RSVP router, until it reaches the sender. Receiver-initiated reservations are

an optimization associated with the merging of reservations on the multicast distribution tree as the reservations get closer to the sender.

Although RSVP reservations are set up by Resv messages which are transmitted from the flow receiver toward the sender, in the case of unicast reservations, a receiver implementation may choose to use the receipt of Path messages sent from the sender as the trigger for generating a Resv message, thereby effecting a source initiated reservation.

- *Routing*. RSVP is not a routing protocol, but rather relies on conventional unicast/multicast IP routing protocols for route determination. RSVP sets up and maintains reservations over an IP path or multicast distribution tree determined by the routing protocol; RSVP consults local routing tables to obtain routes. This approach has three consequences:
  - Routed paths established by interior gateway IP routing protocols (IGPs), such as OSPF and ISIS, may be asymmetrical; that is, the path through the network from a source to a destination may be different to the path from a destination to a source. Clearly then, if the reservation is receiver initiated, some mechanism is needed to ensure that the Resv signaling message from receiver to sender follows the reverse network path to that the media flow would follow from sender to receiver. The mechanism used in RSVP is to transmit an RSVP Path message from the flow sender toward the receiver; routers forward the Path message toward the receiver using conventional routing tables and therefore the Path message follows the same path as the media flow. The Path message sets up forwarding state (called "path state") on RSVP capable routers, which is subsequently used when forwarding the Resv message to ensure that it follows the reverse path to that the media flow will follow from sender to receiver. The sole purpose of path state is to ensure the correct forwarding of Resv messages along the reverse path; reservation state is thereby associated with corresponding path state.
  - If there are insufficient resources on the path chosen by the routing protocol, then a reservation may fail even though there

may be another path through the network with sufficient resources to support the reservation. This issue may be overcome with a constraint-based routing capability, which is provided by MPLS traffic engineering (see Section 4.4.6), which also uses RSVP as its signaling protocol.

  - Following network element failures, the IGP may reroute traffic affected by the failure. If there were flows from existing RSVP reservations, which were rerouted consequently, they may be rerouted before a new admission control decision can be made and before a reservation can be established on the new path. Hence, following network failures there may be a transient period where the service that a flow receives is impacted pending a new RSVP reservation being successfully established.

- *Soft state*. RSVP is a "soft state" protocol; this means that reservations time out if they are not refreshed; Path and Resv messages are sent periodically to refresh the state for each reservation. In the case of multicast flows, reservations can be one-to-many, and the members of the multicast distribution tree can change over the lifetime of the reservation. The soft state model allows RSVP to adapt resource reservations accordingly. The use of soft state also allows RSVP to adapt to changes in network topology, due to network element failures for example. Resending Path and Resv messages periodically also makes RSVP resilient to limited message loss.

- *L3 and L2*. With the "subnet bandwidth manager" extension to RSVP defined in [RFC2814], RSVP can provide admission control over IEEE 802-style LANs.

The use of RSVP within the context of the Integrated Services architecture is defined in [RFC2210]. RFC2210 specifies the structure and contents of the QOS parameters carried by RSVP, when setting up Intserv reservations, which determine how RSVP capable network elements will handle the flow's data. RFC2210 specifies three end-to-end reservation service types:

- *Guaranteed service*. The Intserv guaranteed service (GS) is defined in [RFC2212]; it is intended to support inelastic applications with

low-delay, low-jitter, low-loss, assured bandwidth requirements, such as VoIP and video. By comparison, such applications are typically supported with an EF PHB where Diffserv is deployed.

- *Controlled load service*. The Intserv controlled load (CL) service is defined in [RFC2211]; it is intended to support elastic applications with assured bandwidth requirements. By comparison, such applications are typically supported with an AF PHB where Diffserv is deployed. To quote RFC2211, the controlled load service provides: *"A QOS closely approximating the QOS that same flow would receive from an unloaded network element, but uses capacity (admission) control to assure that this service is received even when the network element is overloaded."*

- *Best-effort service*. Best-effort service is defined as the service, which flows receive that have neither had a successful GS or SL reservation established. The Intserv best-effort service is analogous to the service that would be supported with the default PHB where Diffserv is deployed.

The details of RSVP operation are most easily illustrated with an example.

4.4.2 RSVP Example Reservation Setup

The following example considers a unicast reservation, where the receipt of a Path message from the sender is used as the trigger for the receiver to originate a corresponding Resv message. Refer to Figure 4.9:

1. The sender application on the source host passes SENDER_TSPEC and ADSPEC objects to the RSVP stack via the RSVP Application Programmer Interface (API):
   - The SENDER_TSPEC defines the quantity of resources required at a particular service level (GS or GL). This is defined with a traffic specifier, which uses a token bucket definition. The ENDER_TSPEC

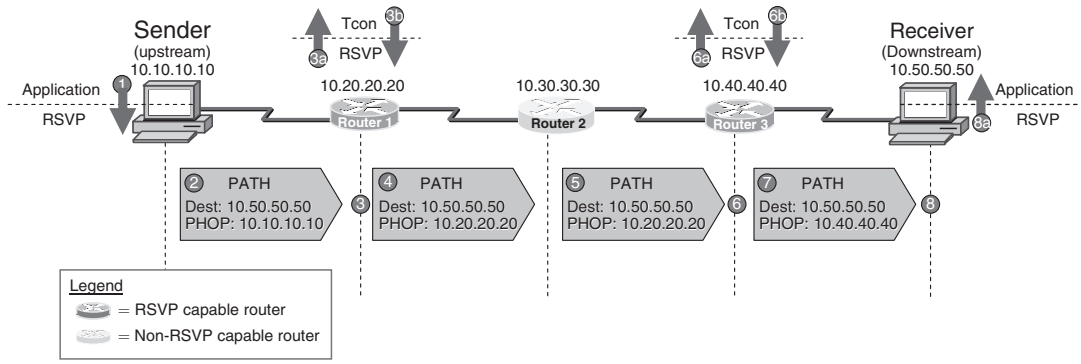

Figure 4.9 RSVP Path message processing example

is generated at the sender and is never modified by intermediate RSVP-capable routers transited en route to the receiver.

- The ADSPEC is generated at the sender and is modified by intermediate RSVP-capable routers transited en route to the receiver to advertise to both the receivers and sender the QOS characteristics of the end-to-end communication path. Receivers use the ADSPEC object to make reservation decisions.

2. The sender generates an RSVP Path message containing the SENDER_TSPEC and ADSPEC objects. In this example, the Path message has an IP source address of 10.10.10.10 and a destination address of 10.50.50.50. The previous hop (PHOP) object in the Path message is set to the sender's address (in this case 10.10.10.10); this is used by the next RSVP hop to set up path state used for Resv message reverse-path forwarding.

   The Path message is forwarded to the next hop on the path toward the IP destination address of the receiver; in this example the next hop is Router 1, which is the sender's default gateway.

3. The Path message is received by Router 1. The Path message has the IP Router Alert Option [RFC2113] set, which alerts the routers to look more closely and examine the contents of the packet, rather than simply forwarding the packet to the destination. As an RSVP capable router, Router 1 determines that the IP protocol

number of the packet is 46, which indicates that it is an RSVP message; the router then passes the Path message to its RSVP function for processing.

In processing the Path message, "path state" is installed which includes the unicast IP address of the previous hop upstream node; this is used for reverse-path forwarding of corresponding Resv messages. Further, the RSVP ADSPEC is passed to the RSVP traffic control function; this is responsible for QOS functions including classification, admission control, and scheduling. The traffic control function may optionally update the ADSPEC with information about the QOS control capabilities available at that point in the path, which might include delay and bandwidth availability information. The updated ADSPEC is then returned to RSVP for delivery to the next hop along the path.

4. Assuming no errors in RSVP processing, Router 1 forwards the Path message, containing the updated ADSPEC object, on toward the destination IP address (10.50.50.50), using its routing table to determine the next hop (hence outbound interface), still with source IP address 10.10.10.10 and destination IP address 10.50.50.50. The previous hop (PHOP) object in the Path message is set to Router 1's address (in this case 10.20.20.20); this is used by the next RSVP hop to set up path state used for Resv message reverse-path forwarding.

   If an error were to occur during the RSVP Path message processing – which could be caused if Router 1 has no route to the destination, for example – Router 1 will return a PathErr message to the sender; RSVP error messages are always hop-by-hop routed.

5. Router 2, on determining that the IP protocol number of the packet is 46, as a non-RSVP capable router simply forwards RSVP messages as though they were any other data packet; it uses its routing table to determine the next hop (hence outbound interface) toward the destination IP address (10.50.50.50) and forwards the packet on accordingly without changing it.

6. RSVP Path message processing is repeated at each RSVP capable router as per step 3.

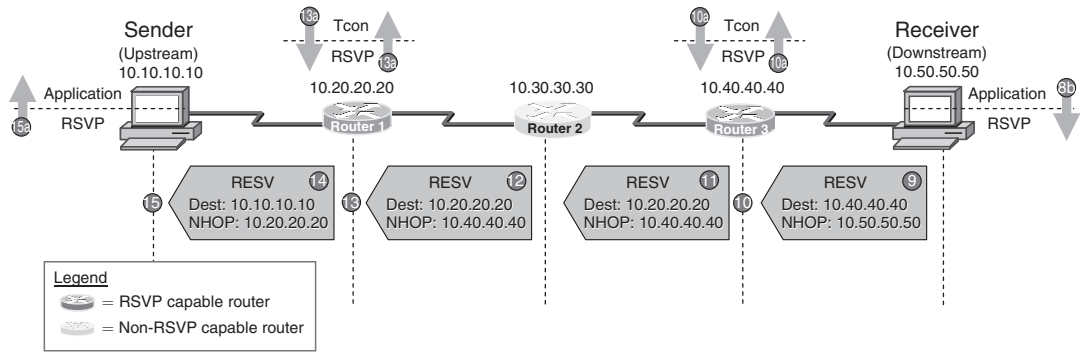

Figure 4.10 RSVP Resv message processing example

7. Assuming no errors in RSVP processing, Router 3 forwards the Path message, containing the updated ADSPEC object, on toward the destination IP address (10.50.50.50), using its routing table to determine the next hop (hence outbound interface), still with source IP address 10.10.10.10 and destination IP address 10.50.50.50. The previous hop (PHOP) object in the Path message is set to Router 3's address (in this case 10.40.40.40); this is used by the next RSVP hop to set up path state used for Resv message reverse-path forwarding.
8. The Path message arrives at the receiver:
   i. The SENDER_TSPEC and ADSPEC are given to the receiving application via the RSVP API.
   ii. Refer to Figure 4.10; the receiver application may use the ADSPEC object to make decisions about the reservation it is about to make. For example, if there were insufficient bandwidth available (as advertised by the ADSPEC) to support a high-definition video stream, the receiving application may decide to request a standard definition stream. The receiver application then supplies RSVP with reservation parameters via the RSVP API, this includes:
      - the requested service level for the reservation, i.e. guaranteed service (GS) or controlled load (CL) service
      - the RECEIVER_TSPEC, which describes the quantity of traffic for which resources should be reserved. This is defined with a traffic specifier, which uses a token bucket definition.

9. The receiver generates an RSVP Resv message which contains a "flow descriptor"; the flow descriptor comprises:
   - The FILTERSPEC, which specifies classification information by which the network can recognize the particular traffic flow that is to receive the QOS defined by the FLOWSPEC. The classifier is defined by the 5-tuple: source IP address, destination IP address, source port, destination port, and IP protocol.
   - The FLOWSPEC, which carries the information generated by the receiving application and which describes the Intserv service characteristics desired for the stream sent by the source in terms of the requested service level (GS or CL), the RECEIVER_TSPEC, and possibly other optional objects.

   The Resv message is forwarded upstream toward the sender. To ensure that the Resv message follows the same path as the Path message in reverse, the Resv message is hop-by-hop routed using Path state information setup during the processing of the Path message. Hence, in this example, the Resv message has an IP source address of 10.50.50.50 and a destination address of 10.40.40.40. In this example, the Resv message is forwarded to Router 3, which is the receiver's default gateway.

10. Router 3 as an RSVP capable router receives the Resv message addressed to it, identifies the RSVP message by IP protocol number 46, and hands the RSVP message over to its RSVP function for processing.
    i. The RSVP traffic control function performs the following functions:
       - Policy control may be performed to provide authorization for the QOS request.
       - Admission control is performed to determine if there are sufficient resources available to satisfy the request at the service level specified in the FLOWSPEC. In this example, the admission control decision would verify that sufficient resources are available on the interface on which the Resv message was received, i.e. the interface to the receiver, in the direction from Router 3 toward the receiver. Available RSVP implementations

generally used a parameterized approach, although a measurement-based approach could potentially be used also, as described in Section 4.1.4.

- Assuming the admission control decision is successful, per flow classifiers are instantiated based upon the FILTERSPEC, and per flow data plane scheduling resources are reserved to assure the quality of service specified for the flow by the FLOWSPEC. The reservation would be on the interface on which the Resv message was received, i.e. the interface to the receiver, in the direction from Router 3 toward the receiver. For a successful GS request, this may consist of assigning the flow to a strict priority queue; for a successful CL request, this may consist of assigning the flow to its own queue within a weighted fair queuinglike system, with a weighting defined to give the flow its requested resources.

ii. State merging, message forwarding, and error handling proceed according to the rules of the RSVP protocol.

11. Assuming no errors in RSVP processing, Router 3 forwards the Resv message upstream to the previous RSVP hop toward the sender. To ensure that the Resv message follows the same path as the Path message in reverse, the Resv message is hop-by-hop routed using path state information set up during the processing of the Path message. Hence, in this example, the Resv message has an IP source address of 10.40.40.40 and a destination address of 10.20.20.20 (i.e. Router 1) and Router 3 uses its routing table to determine how to forward the packet.

    If an error were to occur during the RSVP Resv message processing – which could be due to an admission control or policy control failure, for example – Router 3 will return a ResvErr message to the receiver; RSVP error messages are always hop-by-hop routed.

12. Assuming Router 2 – a non-RSVP capable router – receives the Resv message (which may not be guaranteed if there is more than one path between Router 1 and Router 3 as the IGP routing between Router 1 and Router 3 could be asymmetrical), as the router alert option is not set (it is only used in Path messages)

and the message is addressed to Router 1, Router 2 simply forwards the message as any other normal IP datagram. It uses its routing table to determine how to forward the packet on toward the destination IP address (10.20.20.20) and forwards the packet on accordingly without performing any RSVP processing.

Note that if the RSVP reservation for the flow is successfully established, the flow will receive reservationless best-effort service at non-RSVP capable routers.

13. RSVP Resv message processing is repeated at each RSVP capable router as per step 10.

14. Assuming no errors in RSVP processing, Router 1 forwards the Resv message toward the sender. To ensure that the Resv message follows the same path as the Path message in reverse, the Resv message is hop-by-hop routed using path state information set up during the processing of the Path message. Hence, in this example, the Resv message has an IP source address of 10.20.20.20 and a destination address of 10.10.10.10 and Router 1 uses its routing table to determine how to forward the packet.

15. The Resv message reaches the sender and is delivered to the application. The sending application receives the Resv message, knows that the reservation is successful and can start sending traffic belonging to the flow, knowing that the requested QOS is assured.

If the Resv message contains an optional confirmation request object, on receipt of the Resv message the sender will send a ResvConf message back to the receiver. As RSVP is a soft state protocol, path, and reservation state is refreshed by periodic Path and Resv messages. The sender or receiver can terminate the reservation at any time by sending a PathTear or ResvTear message to release path and/or reservation state.

Although it has been standardized and supported by most operating systems for a number of years, RSVP, as we have described it and as it was originally defined, has neither been widely used by applications

nor widely deployed. One of the main reasons cited for this is a lack of confidence in the scalability of Intserv, caused by the requirement to perform per flow processing and maintain per flow control plane state (i.e. path and reservation state) and data plane state (i.e. per flow classifiers and queuing resources) at each RSVP capable router. The amount of state that each RSVP capable router has to maintain scales in proportion to the number of concurrent reservations, which can potentially be large on high-speed links. A number of developments have aimed to overcome these control plane and data plane scaling concerns:

- Data plane scaling concerns have been addressed by developments that have defined how to support the Intserv architecture over Diffserv; this is described in Section 4.4.4.
- Control plane scaling concerns have been addressed by several efforts:
  - extensions aimed at reducing processing overhead requirements of refresh messages have been defined in [RFC2961]
  - methods for aggregating individual RSVP flow reservations over aggregate RSVP reservations. This is described in Section 4.4.5.

4.4.3 Application Signaling Interaction

The interaction between the application signaling and RSVP signaling will be dependent upon the particular application-signaling model used. In this example, we assume that SIP [RFC3261] is used for call signaling. Consider the call flow for a successful two party call setup and tear down shown in Figure 4.11 and the sequence of events that follows. The example considers a unicast reservation, where the receipt of a Path message from the sender is used as the trigger for the receiver to originate a corresponding Resv message.

The call sequence of events is as follows:

- *Step 1*. Conventional call signaling is used to set up a call from SIP End point_A (e.g. Voice Gateway A) to End point_B (e.g. Voice Gateway B).

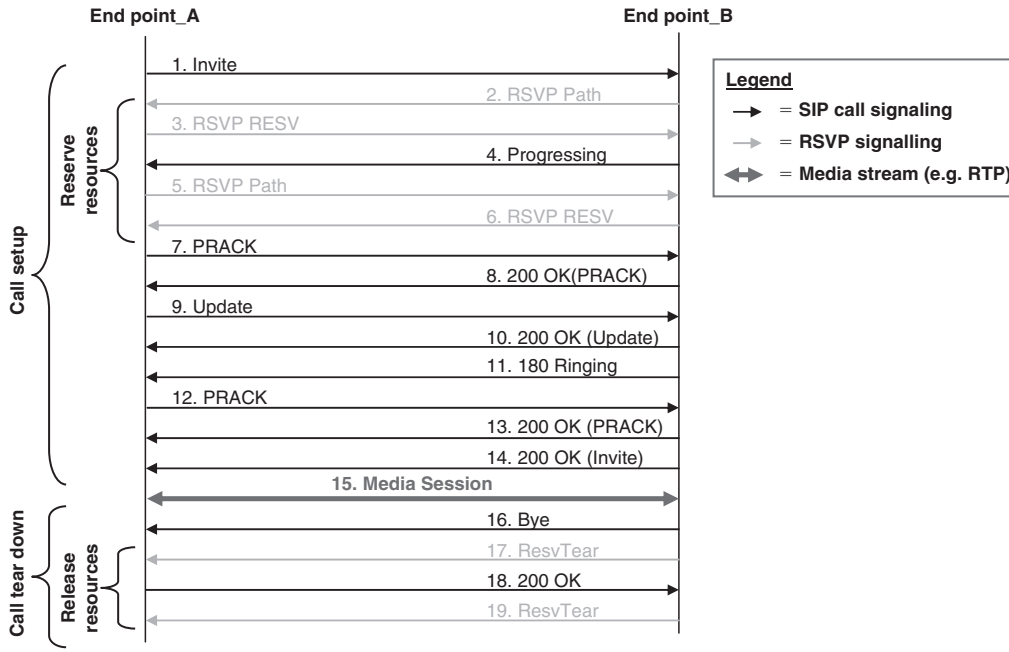

Figure 4.11 Bandwidth manager call flow: basic SIP call

- *Step 2*. SIP End point A originates an RSVP Path message to SIP End point_B.
- *Step 3*. SIP End point B responds with an RSVP Resv message back SIP End point_A to set up the reservation from A to B. Assuming that sufficient bandwidth is available to support the request, the RSVP reservation is successful and the call signaling continues.
- *Step 4*: Call signaling continues.
- *Step 5*. SIP End point B originates an RSVP Path message to SIP End point_A.
- *Step 6*. SIP End point A responds with an RSVP Resv message back SIP End point_B to set up the reservation from B to A. Assuming that sufficient bandwidth is available to support the request, the RSVP reservation is successful and the call signaling continues.

- *Steps 7–15*. It is noted that the destination phone rings only after the available bandwidth is confirmed in both directions. User B picks up the phone and the media session is successfully established between gateways A and B.
- *Step 16*. User connected to Gateway B hangs-up and call signaling starts to clear the call down.
- *Step 17*. SIP End point B originates an RSVP ResvTear message to SIP End point_A, to release the reservation from B to A.
- *Steps 18*. Conventional call signaling continues and the call is cleared down.
- *Step 19*. SIP End point A originates an RSVP ResvTear message to SIP End point_B, to release the reservation from A to B.

4.4.4 Intserv over Diffserv

[RFC 2998] defines "A Framework for Integrated Services Operation over Diffserv Networks" (a.k.a. "Intserv over Diffserv"); in this framework a Diffserv network is viewed as a network element in the end-to-end path of an Intserv reservation.

Consider Figure 4.12, which shows two Intserv regions interconnected with a Diffserv region. Within the Intserv region, RSVP signals per flow resource requirements to the network elements, which apply Intserv admission control to signaled requests. In addition, per flow traffic classifiers and traffic control mechanisms are configured on the network element to ensure that each admitted flow receives the service requested in strict isolation from other traffic. In contrast, within the Diffserv region, traffic is classified into one of a small number of aggregated flows or classes, based on the Diffserv codepoint (DSCP) in the packet's IP header. Intserv over Diffserv describes how end-to-end QOS could be provided in this context by marking the DSCP of RSVP identified flows such that they receive appropriate service within the Diffserv region.

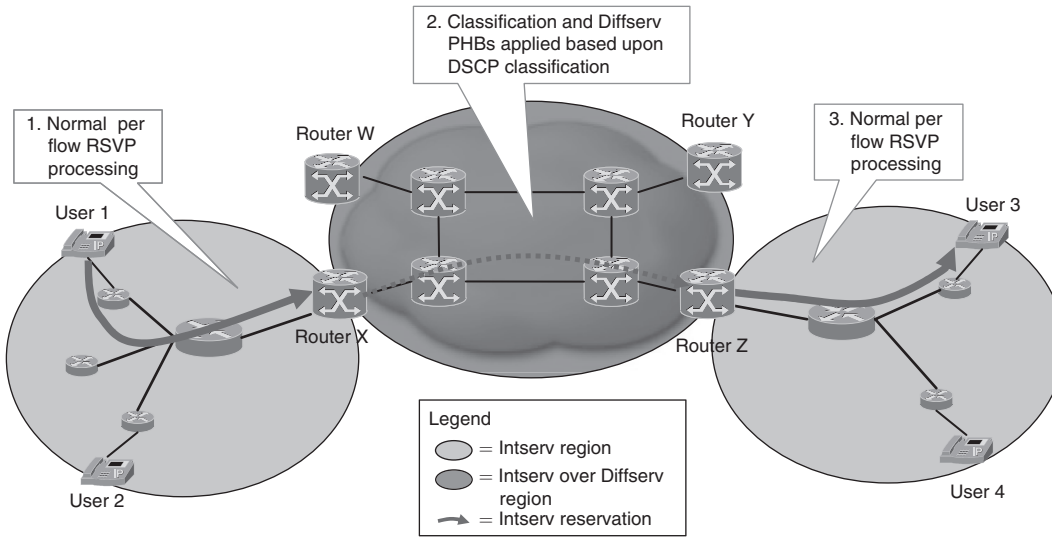

Figure 4.12 Intserv over Diffserv

Consider, for example, what would happen for a successful Intserv request for a flow from User 1 to User 2 in Figure 4.12:

1. Within User 1's local Intserv region, the routers are RSVP capable and normal Intserv/RSVP processing would occur.
2. Within the Diffserv region, the Diffserv-enabled routers would classify traffic based upon the DSCP in the header of each of the packets within the flow. Marking the DSCP of the packets within the flow appropriately would ensure that the packets would be serviced with a per-hop behavior (PHB) that will give them the required service. For example, packets within a flow of an Intserv guaranteed service reservation would be marked such that they were serviced with an EF PHB within the Diffserv domain. This marking could be done either at or close to the sender, or at the routers on the boundary of the Intserv and Diffserv regions, which in this example is at Router X and Router Y.
3. Within User 2's local Intserv region, the routers are RSVP capable and normal Intserv/RSVP processing would occur.

RSVP takes care of admission control in the Intserv regions; however, a number of potential models could be applied for how admission control decisions are made across the Diffserv region:

1. *No admission control over DS region*. In this model, the Diffserv region is statically provisioned and no devices within the Diffserv region are RSVP aware. The routers in the Diffserv region simply ignore RSVP messages. In order for the Intserv reservations to be assured end-to-end the Diffserv region must be capable of supporting the total amount of traffic that is admitted for each PHB.

2. *Admission control at regional boundary only*. In this model, the Diffserv region is statically provisioned and no devices within the Diffserv region are RSVP aware. The boundary routers on the border of the Intserv and Diffserv regions could be considered to have two halves; an Intserv half connecting to the Intserv region and a Diffserv half connecting to the Diffserv region. The border routers maintain a static table of the available resources within the Diffserv domain on a per-PHB basis. As Resv messages are received from the Intserv region destined for the Diffserv region, the border router maps the requested Intserv service level to a Diffserv PHB and performs admission control based upon its table of available resources for that PHB.

   With this approach, the admission control across the Diffserv region is not topology-aware, and therefore it suffers the same issues as all topology-unaware approaches. As described in Section 4.2, they do not consider the availability of resources along the specific path that would be impacted and cannot adapt in real time to changes in network capacity, caused by link or node failures for example, and therefore make inefficient use of the available bandwidth.

3. *Per flow admission control at every hop in DS region*. In this model, all routers within the Diffserv region are "RSVP aware" and are able to participate in some form of RSVP signaling and admission control. However, they classify and schedule traffic on aggregate, based on

DSCP, not based on the per flow classification criteria used by standard RSVP/Intserv routers. RSVP signaling is used for admission control only and per flow classification and scheduling are disabled; effectively the control plane of the routers in the Diffserv region is RSVP while their data plane is Diffserv. As Resv messages are received by a router within the Diffserv region it maps the requested Intserv service level to a Diffserv PHB and performs admission control based upon the currently available resources for that PHB.

This approach provides per flow topology-aware admission control across the Diffserv region. Further, it exploits the signaled admission control (i.e. control plane) benefits of RSVP signaling while maintaining the data plane scalability of Diffserv through aggregate classification, queuing and scheduling. This provides better scaling than "traditional" RSVP because there is no requirement to maintain per flow data plane state, i.e. for classification and scheduling, and hence data plane scaling is independent of number of flows. Hence, Intserv over Diffserv addresses the data plane scalability concerns of RSVP, but it does not address the control plane scalability concerns.

The use of aggregate Diffserv-based classification has its own consequences, however. Following network element failures, the IGP may reroute traffic affected by the failure. If there were flows from existing RSVP reservations that were rerouted as a consequence, they may be rerouted before a new admission control decision can be made and before a new reservation can be established on the new path. With Intserv over Diffserv, there is no isolation between different flows using the same PHB, hence the rerouted traffic may cause congestion within a class; this congestion would impact both rerouted flows and flows that were already successfully admitted onto this path. This service impacting congestion will last until some of the rerouted flows can be torn down. Hence, with Intserv over Diffserv, there may be transient service impacting congestion following network failures; the use of MPLS traffic engineering overcomes this issue (see Section 4.4.6).

It is noted that the example in Figure 4.12 shows distinct Intserv regions interconnected by an Intserv over Diffserv region; however,

it is possible that the Intserv over Diffserv region could extend from end-system to end-system with Intserv over Diffserv used end-to-end, which is a more likely deployment model in practice.

4. *Admission control at every hop in DS region via aggregated reservations*. This model aims to address the data plane scalability concerns of RSVP by aggregating individual flow reservations over aggregate RSVP reservations; this is described in more detail in the proceeding section.

4.4.5 RSVP Aggregation

[RFC3175] defines "RSVP aggregation," which allows a number of RSVP reservations to be aggregated into a single larger reservation. RFC3175 defines the concept of an aggregation region, across which a number of end-to-end reservations, which share a common ingress router to the aggregation region (the aggregator) and egress router from the aggregation region (the de-aggregator), can be aggregated into one larger reservation from ingress to egress. This is conceptually similar to the use of virtual paths (VPs) to aggregate virtual circuits (VCs) within ATM.

Consider, for example, what would happen for a successful Intserv request for a flow from User 1 to User 2 across the example network shown in Figure 4.13 and assume that path state has already been set up as required. The following example considers a unicast reservation, where the receipt of a Path message from the sender is used as the trigger for the receiver to originate a corresponding Resv message.

1. As the sender, User 1 originates the Path message toward User 3, the receiver. Within User 1's local Intserv region, the routers are RSVP capable and normal Intserv/RSVP processing occurs.
2. When the aggregation router on the ingress edge of the aggregation region (Router X) receives the Path message on an interface connected to the non-aggregated Intserv region, it performs normal Intserv/RSVP processing and installs path state accordingly.

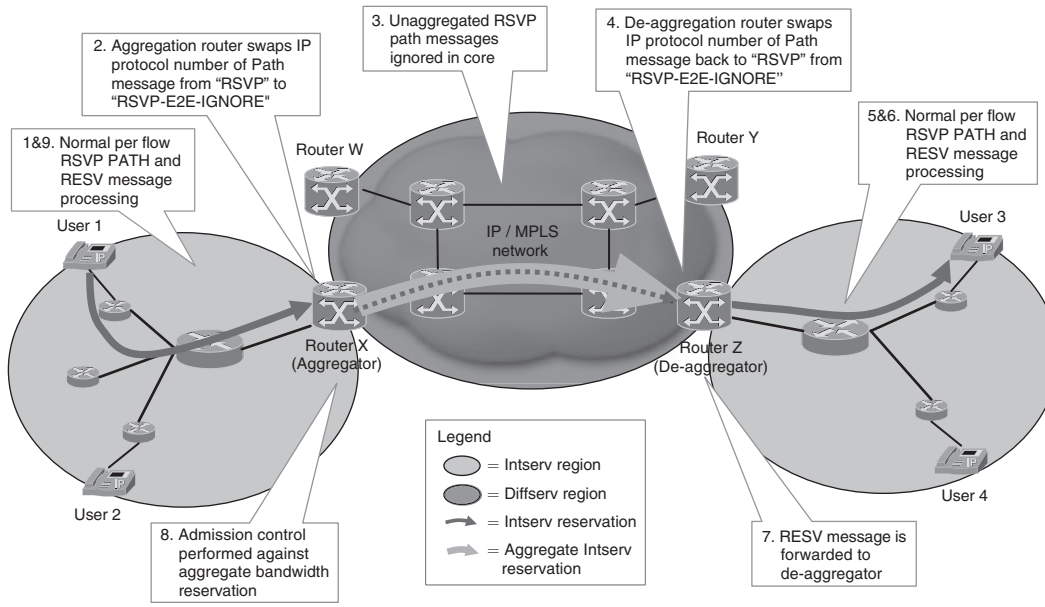

Figure 4.13 RSVP aggregation

Before sending the Path message onwards toward the receiver, however, it changes the IP protocol number of the message from 46 (for RSVP) to 134, which is designated for RSVP-E2E-IGNORE. It then forwards the Path message onwards toward the receiver, which is into the aggregation RSVP region.

3. Within the aggregation region, RSVP capable routers that receive Path messages with IP protocol RSVP-E2E-IGNORE (134), will ignore them rather than performing any RSVP processing. No path state will be installed and the Path messages will be forwarded as any other IP packet.

4. When the de-aggregation router on the egress side of the aggregation region (Router Z) receives the Path messages on an interface connected to the aggregated Intserv region, it installs path state, installing the aggregation router (Router X) as the previous hop upstream node.

Before sending the Path message onwards toward the receiver, however, it changes the IP protocol number of the message back from RSVP-E2E-IGNORE (134) to RSVP (46). It then forwards the Path message onwards toward the receiver, which is into the non-aggregated RSVP region.

In order to determine that a de-aggregator is one part of an aggregator/de-aggregator pair responsible for a particular aggregate reservation, on receipt of a Path message from the aggregation region a de-aggregator also sends a PathErr message back to aggregator, which enables end points for new aggregate reservations to be autodiscovered.

5. Within User 3's local Intserv region, the routers are RSVP capable and normal Intserv/RSVP processing would occur. User 3 receives the Path message and originates a Resv message toward the sender, User 1, in response.

6. When the de-aggregation router (Router Z) receives the Resv message, it performs normal Intserv/RSVP processing for its receiver-facing interface to the Intserv region, performing admission control and instantiating classifiers and scheduling resources accordingly.

   Assuming the Resv message processing is successful, i.e. sufficient resources were available to accept the request, the de-aggregation router then forwards the Resv message upstream to the previous RSVP hop toward the sender, which in this case is Router X.

   As the Resv message is hop-by-hop routed, routers in the aggregated RSVP region will forward the messages as any other IP packet, without performing any RSVP processing. For this reason the IP protocol of the Resv message does not need to be changed to RSVP-E2E-IGNORE.[1]

7. When the aggregation router (Router X) receives the Resv message, assuming that it has a preexisting aggregate RSVP reservation to the de-aggregator, it performs admission control for that request. At the aggregator, however, this admission control decision is not performed against the available resources on its receiverfacing

interface, but rather against the available resources on the aggregate RSVP reservation to the de-aggregator. If the aggregate reservation has not already been established, the receipt of the Resv message could be the trigger to set it up.

If the admission control decision is successful, i.e. if there are sufficient of the aggregate reservation resources available, the aggregation router then forwards the Resv message upstream to the previous RSVP hop toward the sender. If the admission control decision is unsuccessful, this could trigger a resizing on the aggregate reservation.

8. Within User 1's local Intserv region, the routers are RSVP capable and normal Intserv/RSVP processing would occur. Assuming User 1 receives the Resv message, it knows that the requested QOS is assured end-to-end and starts sending traffic associated to the request.

The aggregate RSVP reservation is set up within the aggregation region using an IP protocol of RSVP (46) rather than RSVP-E2E-IGNORE (134) and hence RSVP processing for the aggregate reservation is performed within the RSVP aggregate region. The main difference between the processing of an aggregate reservation and that of a normal reservation is that the data packets associated with the end-to-end reservations do not carry the same IP addresses as the aggregate path and Resv messages and hence cannot be classified using the 5-tuple used with conventional RSVP processing. There are several possible ways that traffic on an RSVP reservation may be classified:

- *Intserv over Diffserv*. If Intserv over Diffserv (see Section 4.4.4) is used in the aggregation region, per flow classification is not required but the DSCP of traffic on aggregate RSVP reservations would be marked such that they receive appropriate service within the Diffserv region.

  An issue with this approach is that there is no guarantee that the traffic from the end-to-end reservations using a particular aggregate reservation will follow the same path within the aggregation region

as the aggregate reservation itself. From the previous example, this would require that the forwarding path used through the aggregation region for traffic from User 1 to User 3 follows the same path as the Path messages for the aggregate reservation, from the aggregator to the de-aggregator. This may not be the case if there are multiple paths with the same IGP metric cost between Router X and Router W and equal cost multipath (ECMP) algorithms are used, which commonly rely on hashing functions using contexts such as source and destination addresses to determine how traffic is load-balanced over the equal cost paths, for example. Clearly if traffic from the end-to-end reservations follows a path other than that of their aggregate reservation, their QOS on that other path may not be assured. The routing design within a particular deployment may be able to be adjusted to ensure that there are no equal cost paths, and hence that this does not happen. Alternatively, tunneling can be used between the aggregator and de-aggregator.

- *Tunneling*. Traffic using an aggregate reservation may be tunneling from the aggregator to the de-aggregator, using IP-in-IP tunnels, GRE tunnels, or MPLS TE tunnels [draft-ietf-tsvwg-rsvp-dste-02.txt], for example. If tunneling is used, the traffic using an aggregate reservation may be classified by identifying the particular "tunnel" associated with that reservation or Intserv over Diffserv may be used.

  With tunneling approaches, implicitly traffic from the end-to-end reservations using a particular aggregate reservation follows the same path within the aggregation region as the aggregate reservation itself.

  The size of the aggregate reservation needs to be sufficient to support the guarantees of all of the end-to-end reservations that use that particular aggregate reservation. The size of aggregate reservations could be statically configured or dynamically determined using parameterized (i.e. sum the token buckets specified in the SENDER_TSPECs of the end-to-end reservations using that aggregate reservation) or measurement-based approaches, as described in Section 4.1.4. To reduce the frequency of resizing and churn, the aggregate reservation may be resized slowly in bandwidth

chunks, with hysteresis being applied to size increases and decreases. For example, if the currently allocated bandwidth from the aggregate reservation is greater than 80% utilized, but there is sufficient bandwidth to accept the call, then accept the call AND attempt to increase the tunnel bandwidth by 20%. Similar approaches could be applied to downsize aggregate reservations when end-to-end reservations are cleared down. Clearly, more complicated sizing schemes and heuristics are possible.

Where RSVP aggregation is used, the control plane (path and reservation) state and data plane (classification and scheduling) state required within the aggregation region is dependent upon the number of aggregate reservations and independent of the number of end-to-end reservations. If this is combined with Intserv over Diffserv, the data plane state it is also independent of the number of aggregate reservations. Further, RSVP aggregation supports the concept of recursive aggregation, allowing aggregate reservations themselves to be further aggregated. This could potentially reduce control plane state even further, at the cost of incurring the complexity of an additional level of aggregation.

4.4.6 RSVP Traffic Engineering

RSVP was originally designed to support an anticipated widespread demand for real-time applications over the Internet, such as teleconferencing. That anticipated demand has not materialized in practice and the widespread deployment of RSVP – at least RSVP as it has been described in the preceding sections, that is – has not resulted. However, the traffic engineering (TE) extension for RSVP [RFC 3209], referred to as RSVP-TE, has been widely deployed by a large number of network service providers.

RSVP-TE is used for traffic engineering within multiprotocol label switching (MPLS) networks. Used in this context, there are some significant differences from RSVP as it has been described in the previous sections. The most significant difference is that rather than using paths already established by the IGP, RSVP-TE is used to set up

the data path; RSVP-TE establishes MPLS label switched paths (LSPs), in addition to performing resource reservation and admission control.

In the context of admission control, there are several ways that RSVP-TE could be deployed:

- RSVP-TE could be used in conjunction with MPLS as the tunneling technology underlying RSVP aggregation as described in Section 4.4.5.
- RSVP-TE could be used to provide end-to-end reservations for MPLS attached end-systems, which support MPLS and RSVP-TE. In practice, this type of deployment is most likely in service provider network environments, between large-scale voice-over IP gateways for example.
- RSVP-TE is widely used for traffic engineering within service provider networks; in this context, it provides admission control for traffic "trunks" across the network, where a "trunk" is an aggregation of traffic from an ingress point to an egress point. In this context, the capability provided is not one of real-time admission control, providing feedback to end-system applications, but of capacity management within the core of an IP/MPLS network; traffic engineering in this context is described in Chapter 6, Section 6.2.3.

4.5 NSIS

An effort is currently underway within the Next Steps in Signaling (NSIS) [NSIS] Working Group within the IETF to standardize a new suite of extensible IP signaling protocols, which are referred to generically as "NSIS." QOS signaling is the first explicit use case that the NSIS protocols have addressed; however, they have been designed with the ability to support other use cases such as configuring firewall pinholes and network address translation (NAT) bindings. The NSIS framework is defined in [RFC4080].

NSIS consists of two protocol layers. The lower layer is a generic transport protocol layer referred to as the NSIS Transport Layer Protocol (NTLP); the General Internet Signaling Protocol (GIST) [GIST] is the protocol specified for the NTLP layer. GIST could be used by a number of NSIS Signaling Layer Protocols (NSLPs) at the signaling layer, although currently only two such protocols are defined: an NSLP for QOS Signaling [QNLSP] and a NAT/Firewall NSLP [NNLSP].

Similarly, to RSVP – when augmented with the various enhancements to the original specification, that is – NSIS relies on conventional routing protocols, uses a 5-tuple flow identifier, uses soft state and supports the aggregation of reservations. The NSIS QOS NLSP can also provide signaling capabilities for any QOS model or architecture, including Intserv and Diffserv.

However, NSIS has the following significant differences from RSVP:

- *Bidirectional reservations*. NSIS supports both unidirectional and bidirectional reservations, while RSVP supports unidirectional reservations only.
- *Unicast only*. RSVP supports both unicast and multicast traffic reservations. However, as multicast has not been widely deployed and support for multicast reservations added significantly to the complexity of RSVP, NSIS made the decision to support unicast reservations only.
- *Sender or receiver-initiated reservations*. Unlike RSVP which supports receiver-initiated reservations, NSIS supports both sender and receiver-initiated reservations.
- *L3 only*. NSIS provides no equivalent to the RSVP subnet bandwidth manager functionality, and can be used to make admission control decisions in IP networks only.

NSIS has also been designed to provide mobility and support standard IP security protocols.

NSIS has been designed from the outset with the benefit of knowledge of the issues experienced during the development and deployment

of RVSP; however, it is yet to be seen whether the functional capabilities that NSIS provides in addition to those already provided by RSVP will lead to its widespread deployment.

4.6 End-system Measurement-based Admission Control

IP endpoint measurement-based admission control (MBAC), which was first documented in [GIBBONS], relies on application end points to make admission control decisions themselves. End point use measurements of characteristics of traffic to other destination end points, in order to infer the state of the network and hence determine whether new streams can be established to those respective destinations with the required QOS. Endpoint MBAC can rely either on passive or active traffic monitoring (the use of passive and active approaches for network monitoring is discussed in Chapter 5); hybrid approaches are also possible:

- *Active monitoring*. Active network monitoring involves sending synthetic test streams of "probe" packets across the network to characterize the network performance. Endpoint MBAC using active monitoring relies on measuring characteristics of active monitoring probes – such as delay, jitter, loss or number of ECN marked (see Chapter 2, Section 2.3.4.4) probes [KELLY] – sent between end-systems, and using these measurements as basis for making an admission control decision. When an end point needs to set up a new flow, the previously measured characteristics are compared against defined thresholds to determine whether the flow will receive the required QOS and hence can be accepted.
- *Passive monitoring*. Endpoint MBAC using passive monitoring relies on measuring characteristics of pre-existing media streams between end-systems. Where the real-time protocol (RTP) [RFC 3550] is used, for example, the timestamp and sequence number information in the RTP header could be used to determine the delay, jitter, and loss of the received stream at the receiving end-system. As with the active measurement-based approach, these

measured characteristics can be used as the basis for making an admission control decision. ECN marking could also be used as an input to the admission control decision with a passive monitoring approach.

A purely passive monitoring approach presumes that there is already an active stream between the two end-systems, i.e. such that there is some current measurement data when a new flow needs to be established, and a new admission control decision is required. If this presumption is not correct then this approach could be augmented with the addition of an active monitoring stream, when there are no bearer streams active.

End-system measurement-based admission control approaches are implicitly topology-aware (assuming that active measurement probes follow the same path as the media traffic; with passive measurement this is implicit), hence can adapt to the available network capacity, and therefore do not suffer the bandwidth inefficiency of topology-unaware approaches. In addition, end-system-based MBAC approaches rely on end-to-end media or probe traffic at layer 4 or above; hence they can provide admission control capabilities transparently of the underlying layers.

Endpoint MBAC suffers the same potential issues as other measurement-based approaches (as discussed in Section 4.1.4.2), that measurements taken over the past measurement interval may not provide a good indication on which to base admission control decisions in the next measurement interval. Hence, despite significant research in endpoint-based admission control schemes [KEY, GANESH, BRESLAU2, BAIN], endpoint MBAC is not yet widely deployed, and it remains to be seen whether endpoint MBAC can provide the deterministic characteristics demanded by real-time applications such as voice and video.

4.7 Summary

At the start of this chapter, we highlighted that there are a number of approaches to capacity admission control, none of which is universally

Approach	Type of approach	Topology aware?	Multicast Support	L3 only?
e.g. Call server or video server based	Off path	No	No	No
Bandwidth manager	Off path	Yes	No	No
RSVP-based	On path network signaling based	Yes	Yes	No
NSIS	On path network signaling based	Yes	No	Yes
End-system MBAC	End-system MBAC	Yes	No	No

Figure 4.14 Summary of admission control approaches

deployed today. Due to the variety of potential deployment scenarios, applications, and services, there is currently no "one size fits all" solution to the problem of capacity admission control, and as a result, some technologies for admission control are still evolving. Hence, we summarize the key characteristics of the different admission control solutions, which are likely to affect their applicability to a particular deployment, in the table in Figure 4.14.

It is noted that admission control need not be implemented end-to-end through the network but rather is only required in those parts of the network where congestion may occur, and then only for those types of traffic that need it. For example, explicit admission control mechanisms may be deployed at the edges of the network, where bandwidth is scarce, and over-provisioning may be relied on in the core of the network. Further, the different approaches to admission control need not be mutually exclusive; it is possible for one approach to be used in the core of the network and another in the access.

References[2]

[BAIN] Alan Bain, Peter Key, Modeling the Performance of In-Call Probing for Multi-Level Adaptive Applications, Microsoft Research Technical Report, MSR-TR-2002-06, Jan 2002. Available online at: http://research.microsoft.com/research/pubs/view.aspx? msr_tr_id= MSR-TR-2002-06

[BRESLAU1] L. Breslau, S. Jamin, and S. Shenker, Comments on the performance of measurement-based admission control algorithms, *Proc. IEEE INFOCOM, 2000*, pp. 1233–1242, March 2000

[BRESLAU2] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, Endpoint admission control: Architectural issues and performance, in *Proc. of ACM SIGCOMM 2000*, pp. 57–69, August 2000

[draft-ietf-tsvwg-rsvp-dste] draft-ietf-tsvwg-rsvp-dste, Francois Le Faucheur et al., Aggregation of RSVP Reservations over MPLS TE/DS-TE Tunnels, IETF draft, September 2006

[draft-ietf-tsvwg-rsvp-ipsec] draft-ietf-tsvwg-rsvp-ipsec, Francois Le Faucheur et al., Generic Aggregate RSVP Reservations, IETF draft, February 2006

[GANESH] A. J. Ganesh, P. B. Key, D. Polis, and R. Srikant, Congestion notification and probing mechanisms for endpoint admission control, *Networking, IEEE/ACM Transactions*, Volume 14, Issue 3, June 2006, pp. 568–578

[GIBBONS] R. J. Gibbens and F. P. Kelly, Measurement-based connection admission control, in *15th International Teletraffic Congress*, volume 2b, pp. 879–888, Elsevier, 1997

[GIST] H. Schuzrinne and R. Hancock, GIST: General Internet Signaling Transport Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draft-ietf-nsis-ntlp-07.txt

[Gq'] ETSI TS 183 017 Gq' interface based on Diameter protocol

[JAMIN] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, A measurement-based admission control algorithm for Integrated Service packet networks, *IEEE/ACM Trans. Networking*, Vol. 5, pp. 56–70, February 1997

[KELLY] Frank P. Kelly, Peter B. Key, and Stan Zachary, Distributed Admission Control, *IEEE Journal on Selected Areas in Communications*, Vol. 18, no. 12, pp. 2617–2628, December 2000

[KEY] Peter Key and Laurent Massouli, Probing strategies for distributed admission control in large and small scale systems, *Proceedings of IEEE Infocom*, San Francisco 2003

[KNIGHTLY] E. W. Knightly and N. B. Shroff, Admission control for statistical QOS: Theory and practice, *IEEE Network*, Vol. 13, pp. 20–29, March/April 1999

[MSF2005.187] Olov Schelén, Implementation Agreement for Diameter interface to Bandwidth Manager, MSF Contribution MSF2005.187, May 2006

[MSF-TR-ARCH-005-FINAL] Chris Gallon, Olov Schelén, Bandwidth Management in Next Generation Packet Networks, MSF Technical Report, August 2005. Available at: http://www.msforum.org/ techinfo/reports/MSF-TR-ARCH-005-FINAL.pdf

[MSF-TR-ARCH-008-FINAL] John Evans, Network Engineering to Support the Bandwidth Manager Architecture, MSF White Paper, May 2006. Available at: http://www.msforum.org/techinfo/reports/ MSF-TR-ARCH-008-FINAL.pdf

[NNLSP] M. Stiemerling, H. Tschofenig, and C. Aoun, NAT/ Firewall NSIS Signaling Layer Protocol (NSLP), Internet draft, work in progress, July 2005; http://www.ietf.org/internet-drafts/draftietf-nsis-nslp-natfw-07.txt

[NSIS] http://www.ietf.org/html.charters/nsis-charter.html

[PCE] http://www.ietf.org/html.charters/pce-charter.html

[QNLSP] J. Manner et al., NSLP for Quality-of-Service signaling, Internet draft, work in progress, July 2005; http://www.ietf.org/ internet-drafts/draftietf- nsis-qos-nslp-07.txt

[RFC1633] R. Braden et al., RFC1633, Integrated Services in the Internet Architecture: an Overview, June 1994

[RFC2113] D. Katz, IP Router Alert Option, *RFC 2113*, February 1997

[RFC2205] R. Braden, Ed., Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, *RFC 2205*, September 1997

[RFC2210] J. Wroclawski, The Use of RSVP with IETF Integrated Services, *RFC 2210*, September 1997

[RFC2211] J. Wroclawski, Specification of the Controlled-Load Network Element Service, *RFC 2211*, September 1997

[RFC2212] S. Shenker et al., Specification of Guaranteed Quality of Service, *RFC 2212*, September 1997

[RFC2638] K. Nichols and V. Jacobson, A Two-bit Differentiated Services Architecture for the Internet, *RFC 2638*, July 1999

[RFC2814] R. Yavatkar et al., SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks, *RFC 2814*, May 2000

[RFC2961] L. Berger, RSVP Refresh Overhead Reduction Extensions, *RFC 2961*, April 2001

[RFC2998] Y. Bernet et al., *RFC* 2998, A Framework for Integrated Services Operation over Diffserv Networks, November 2000

[RFC3175] F. Baker et al., Aggregation of RSVP for IPv4 and IPv6 Reservations, *RFC 3175*, September 2001

[RFC3209] D. Awduche et al., RSVP-TE: Extensions to RSVP for LSP tunnels, *RFC 3209*, Dec. 2001; http://www.rfc-editor.org/rfc/rfc3209.txt

[RFC3261] J. Rosenberg et al., SIP: Session Initiation Protocol, *RFC 3261*, June 2002

[RFC3312] G. Camarillo, Ed. et al., Integration of Resource Management and Session Initiation Protocol (SIP), *RFC 3312*, October 2002

[RFC3550] H. Schulzrinne, RTP: A Transport Protocol for Real-Time Applications, *RFC 3550*, July 2003

[RFC3588] P. Calhoun et al., Diameter Base Protocol, *RFC 3588*, September 2003

[RFC4080] R. Hancock et al., Next Steps in Signaling: Framework, *RFC 4080*, June 2005; http://www.rfc-editor.org/rfc/rfc4080.txt

[RFC4542] F. Baker, J. Polk, Implementing an Emergency Telecommunications Service (ETS) for Real-Time Services in the Internet Protocol Suite, *RFC 4542*, May 2006

Notes

1. Only Path, PathTear and ResvConf messages use an IP protocol of RSVP-E2E-IGNORE (134).
2. The nature of the networking industry and community means that some of the sources referred to in this book exist only on the World Wide Web. All Universal Resource Locators (URLs) have been checked and were correct at the time of going to press, but their longevity cannot be guaranteed.