APPENDIX D

Specification of PEP and PDP Functionality in Support of Windows 2000 Quality of Service

As explained throughout this book, optimal QoS functionality is realized when hosts and network equipment cooperate to enable management of network resources. Windows 2000 hosts (and, to a lesser degree, Windows 98 hosts) support application-based, signaled QoS. This functionality enables significant value-add when compared with management systems that rely exclusively on push-provisioned, network-centric mechanisms (especially when considering higher-quality services).

Many of the benefits of Windows 2000 QoS mechanisms can be realized with only minimal support in the network. However, to fully realize these benefits, complementary functionality is required in network equipment in the form of signaling-aware policy enforcement points (PEPs) and policy decision points (PDPs). (PEPs and PDPs are discussed in depth in Chapter 9, "QoS Policy.") The purpose of this appendix is to encourage network equipment vendors to implement the functionality necessary to realize the additional value that results when hosts and network equipment cooperate to support signaled QoS. To this end, the appendix specifies various levels of PEP and PDP functionality that may be implemented by network equipment vendors in support of signaled QoS. All specified functionality is based on open standards and published protocols.

This appendix begins with an overview of PEP/PDP functionality. The overview is followed by specific descriptions of levels of incremental functionality. Each description includes a "Motivation" subsection that explains how this functionality benefits the QoS-enabled network.

The reader is expected to be familiar with the concepts developed in chapters 1–14 of this book.

Review of PEP/PDP Functionality in Support of Signaled QoS

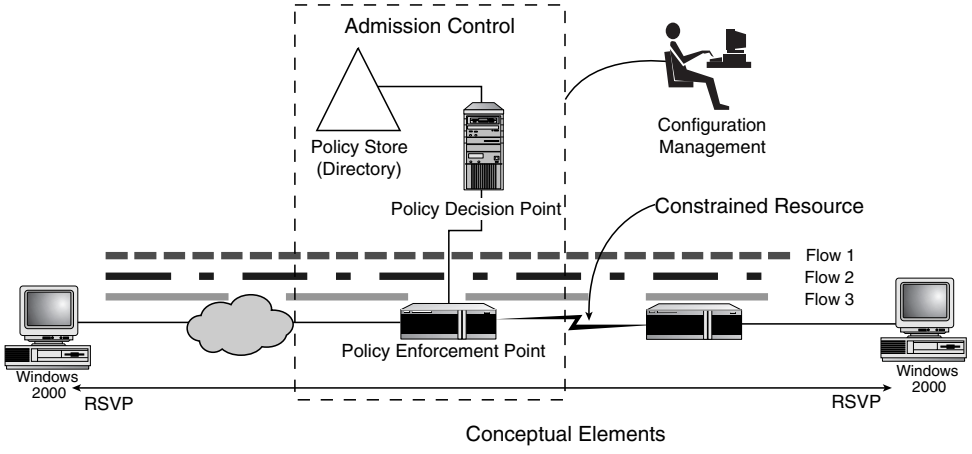
Note

The following review is very brief and is not intended to replace the detailed discussion of PEP/PDP functionality in Chapter 9.

Note that this section—and the entire appendix—focuses primarily on functionality related to signaled QoS.

Figure D.1 illustrates a simple network and serves as a reference diagram for the discussion in this appendix.

Figure D.1 Reference Configuration



Note

Note that Figure D.1 illustrates PEP and PDP functionality in separate devices. As explained in Chapter 9, "PEP" and "PDP" refer to logical functionality, which may be collocated in a single physical device or distributed across multiple physical devices.

The figure illustrates only a single PEP and a single PDP. In general, multiple PEPs and PDPs will be located at strategic points in the network topology (such as at ingress points to constrained network regions). A single PDP likely will support multiple PEPs. PDPs will require a form of policy data store, such as a directory.

The Role of the PEP/PDP in End-to-End Signaled QoS

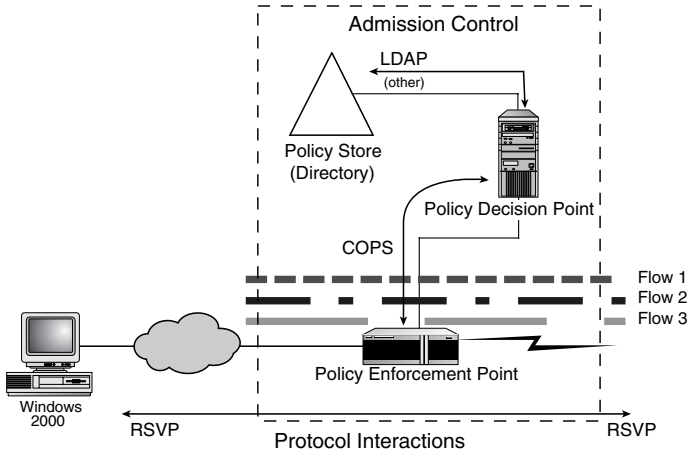
QoS-enabled networks employ a variety of mechanisms to provide different qualities of service guarantees. Any QoS-enabled network relies on fundamental traffic-handling mechanisms in the network devices through which data is carried. (See Chapter 3, "Queuing Mechanisms," for a discussion of traffic-handling mechanisms.) These devices are predominantly switches and routers. The traffic-handling mechanisms include mechanisms by which traffic can be classified as belonging to a specific *flow*, and queuing mechanisms by which traffic on a particular flow can be allotted more or less resources. Network devices that support traffic-handling mechanisms also act as PEPs because they are capable of applying QoS policies to the traffic passing through them.

In addition, network devices must support mechanisms by which their traffic-handling functionality can be provisioned or configured. Provisioning and configuration mechanisms include both *push* mechanisms and *signaled* mechanisms (RSVP). Typically, PEPs are associated with some form of *policy server*, which provides PDP functionality. In *push provisioning*, a PDP uses one or more well-known configuration protocols, such as Simple Network Management Protocol (SNMP), a command-line interface (CLI), or Common Open Policy Service (COPS) to "push" configuration information to associated PEPs.

In *signaled QoS*, RSVP messages arrive at PEPs (see Chapter 5, "RSVP," for a discussion of RSVP). These messages describe QoS traffic flows that are traversing the PEP and request network resources for these flows. The PEP extracts the relevant information from the RSVP messages and passes it to the associated PDP, typically using the COPS protocol. At a minimum, the PDP will glean classification information from these messages. To further clarify this point, RSVP messages indicate the sending (or receiving) user and the sending (or receiving) application, as well as the classification information (IP addresses and IP ports) by which to recognize the traffic associated with the specified user and application. This effectively provides a *binding* that can be used by the PDP to enhance the user interface presented to the network manager and that can be used in push provisioning of classification information.

In addition to gleaning classification information, the PDP may also decide whether a particular RSVP request is admissible. As such, a PEP and its associated PDP act as an *admission control agent* (see the section in Chapter 5 titled "Admission Control Agents"). Certain PEPs may include PDP functionality locally. Others may offload the PDP functionality to a separate policy server. PEP/PDPs work together to apply QoS policies. In doing so, they act on policy data that is typically stored in some form of database, commonly a distributed directory. Often the PDP communicates with this directory using the LDAP protocol. Figure D.2 illustrates the interaction among hosts, PEPs, PDPs, and the directory and the related protocols.

Figure D.2 Interactions Between Host, PEP, PDP and the Directory



By supporting RSVP signaling, network devices can offer improved manageability and can support networks with a higher quality/efficiency (QE) product. (See Chapter 2, "The Quality/Efficiency Product," for a discussion of the QE product.) Most of this appendix addresses varying levels of support for RSVP signaled QoS.

Traffic-Handling Mechanisms

Support for basic traffic-handling mechanisms is the most fundamental requirement of any network equipment purporting to support QoS. Specific implementations of traffic-handling mechanisms vary from vendor to vendor.

Motivation

Network devices support QoS by providing traffic-handling mechanisms that recognize the different traffic flows passing through them and handle them differently. This basic functionality is necessary to build a QoS network.

Incremental Functionality—Traffic-Handling Mechanisms

Well-known traffic-handling mechanisms include the following:

- DiffServ PHBs, including the class-selector (CS) PHB group, the expedited forwarding (EF) PHB, and the assured forwarding (AF) PHB group. (See [RFC 2474], [RFC 2475], [RFC 2597], and [RFC 2598]). Also see Chapter 6, "Differentiated Services."

- 802 `user_priority` traffic classes, per IEEE 802.1D-1998 (see [802.1D-1998]). Also see Chapter 7, "The Subnet Bandwidth Manager and 802 Networks."
- Traffic-handling mechanisms necessary to support the IntServ Guaranteed Service and Controlled Load Service. (See [RFC 2211] and [RFC 2212]). Also see Chapter 4, "Integrated Services."

Note

Note that the ISSLL working group of the IETF defines the *Null Service* [RFC 2997] in addition to the IntServ Guaranteed Service and Controlled Load Service. Support for this service does not dictate any particular traffic-handling mechanisms. Nonetheless, it is expected that PEPs will provide varying levels of QoS for applications requesting the Null Service by applying one or more of the various traffic-handling mechanisms enumerated.

Generally, Layer 3 devices (such as routers) should offer DiffServ PHBs or the Guaranteed and Controlled Load Services. Layer 2 devices (such as switches) should offer two or more traffic classes, based on 802 `user_priority`. However, certain high-end Layer 2 devices may offer traffic-handling mechanisms typically supported by Layer 3 devices (and vice versa). Microsoft intentionally does not specify the underlying queuing mechanisms by which traffic-handling mechanisms should be supported.

Application of Traffic-Handling Mechanisms

It is expected that the EF PHB will be used to construct a DiffServ service offering characteristics similar to a leased line. Work underway in the ISSLL working group of the IETF recommends that this PHB be used to support the IntServ Guaranteed Service across DiffServ networks. This work also recommends that one or more AF PHBs be used to support the IntServ Controlled Load Service across DiffServ networks.

It is expected that one of the AF PHBs (or an alternate PHB) will be available to provide *less-than-best-effort* (LBE) service. Traffic served by this PHB would be treated at a lower priority than all other traffic. This PHB would be used to facilitate early deployments of certain nonadaptive applications that tend to use network resources aggressively (and that cannot otherwise be safely deployed).

RSVP Signaling and Admission Control Agent Functionality

Microsoft encourages the development of network devices that support varying degrees of RSVP and admission control agent functionality. These include the following:

- *Subnet Bandwidth Manager* (SBM) client functionality
- RSVP *snooping* functionality
- *Resource-based* quantitative admission control agent functionality
- *Policy-based* admission control agent functionality (including support for Active Directory as policy data store)
- *Designated Subnet Bandwidth Manager* (DSBM) functionality

These features are listed in roughly ascending order of sophistication (with the exception of DSBM functionality). They are specified in detail in this section.

SBM Client Functionality

The ISSLL working group of the IETF has specified extensions to the RSVP protocol that optimize operation of the signaling protocol on shared-media networks.

Note

Note that throughout this book, the term *shared-media networks* is used to refer to Layer 2 networks composed of switches, bridges, hubs, and physically shared segments.

These extensions are specified in the form of the SBM (see [RFC 2814], [RFC 2815], [RFC 2816], and Chapter 7). Existing SBM implementations are available from several vendors of network equipment and may be used for interoperability testing.

To interoperate with the SBM, any device that transmits onto a shared-media network and that is RSVP-conversant must implement minor modifications to the RSVP protocol. These modifications comprise SBM client functionality.

Motivation

RSVP-conversant Layer 3 devices that transmit onto shared media networks and that are not SBM client-compliant will generally bypass PEP/PDPs (that rely on RSVP signaling) in the shared network. As a consequence, these will defeat signaling-based QoS management mechanisms used by the network manager. Even though shared-media networks are often considered overprovisioned, network managers may employ RSVP-capable PEP/PDPs in these networks for the purpose of enforcing enterprise QoS policies, conducting usage tracking, or providing high-quality guarantees to high-bandwidth applications. Microsoft's SBM-based Admission Control Service (ACS) is one such system (see Chapter 14, "The Microsoft Admission Control Service," for a discussion of the ACS). SBM client support requires only minimal incremental development work beyond basic RSVP protocol support.

Incremental Functionality—SBM Client

All RSVP-conversant Layer 3 devices transmitting onto a shared-media network *must* adhere to the SBM client functionality [RFC 2814]. These primarily include hosts and routers but also include any device that carries traffic between two or more different IP subnetworks, with one or more of these subnetworks being a shared-media subnetwork. SBM client functionality must be supported on each interface that transmits onto a shared-media subnetwork. SBM client functionality includes the capability to detect `I_AM_DSBM` announcements indicating the presence of a DSBM and to redirect RSVP signaling messages accordingly.

Note

SBM client functionality does not include the capability to act as a DSBM. DSBM functionality is required on Layer 2 devices (switches or other devices that carry traffic between interfaces on the same IP subnetwork) that provide RSVP-based admission control functionality. DSBM functionality will be discussed separately later in this appendix (in the section titled "DSBM Functionality").

RSVP Snooping Functionality

This is the minimal level of RSVP signaling support that benefits from Windows signaled QoS functionality. Network devices supporting policy-based QoS functionality *should* at least be capable of RSVP snooping. *Snooping* refers to the capability of a network device to glean robust classification information by *monitoring* Windows 2000 RSVP-generated messages that pass through it. (See the section titled "Snooping" in Chapter 5.) This information can then be used to enhance policy management systems. Snooping does not require the network device to actively participate in the RSVP signaling protocol.

Motivation

Common traffic-handling mechanisms in PEPs classify traffic based on common header fields, including IP addresses and ports. Policy management systems typically present a management interface that enables the network manager to manage QoS based on users and/or applications. The policy management system relies on a mapping from users and applications to IP ports and addresses. This mapping is typically generated by the policy management system based on some combination of the following mechanisms:

- Statically configured mappings that are manually created by the network manager
- Libraries included with the management system that identify well-known ports and the corresponding applications
- Dynamic Host Configuration Protocol (DHCP) servers (or other network services) that are integral parts of the management system and that correlate users with IP addresses
- Complicated classification hardware that peers deep into packets to "guess" the associated application based on well-known identifiers in arbitrarily deep packet fields (also known as *network-based application recognition*)

These mechanisms are often cumbersome or failure-prone. In addition, some may place a high burden on network classification hardware and may not scale well under heavy loads. PEP/PDPs can use RSVP snooping functionality to reduce the management burden associated with manually generated mappings and to improve the robustness and scalability of automated mechanisms. See the sections "Traffic Classification," in Chapter 6, and "Compiling and Installing Semi-Static Policies," in Chapter 9.

In IP Security (IPSec) environments, it is typically not possible for network elements to extract IP port numbers from encrypted packets. Thus, as more traffic is encrypted at hosts, network-based classification will become even less useful. In these environments, RSVP signaling messages carry the IPSec Security Parameter Index (SPI), which can be used by the PEP in place of IP ports to associate traffic with applications. For further discussion of the interoperation of RSVP and IPSec, see the section "RSVP with IP Security," in Chapter 5.

Incremental Functionality—RSVP Message Parsing Functionality

Snooping functionality begins with the capability to parse RSVP messages. In particular, network devices must be capable of parsing RSVP `POLICY_DATA` objects, RSVP `FILTERPSEC` objects, RSVP `SENDER_TEMPLATE` objects, and RSVP `SESSION` objects [RFC 2205] in RSVP messages. The `POLICY_DATA` objects identify the users and applications with which signaled traffic is associated. The `FILTERSPEC`, `SENDER_TEMPLATE`, and `SESSION` objects specify the IP

addresses and ports (or SPI) by which this traffic can be recognized. Thus, by extracting this information from an RSVP message, a policy management system can generate a reliable mapping between users and applications on one hand, and IP 5-tuple classification information on the other hand, for internal use.

The format of RSVP policy objects in general is defined in [RFC 2752]. The format of application identifier policy objects specifically is defined in [RFC 2872]. Microsoft-generated user identity `POLICY_DATA` objects contain Kerberos authenticated user IDs in the form of an X.500 distinguished name. (See the section "User Identification" in Chapter 12, "The GQoS API and the QoS Service Provider"). Processing of these objects requires the policy management system to participate in the authentication infrastructure as a service principle. Further details are documented in [RFC 1510] and [MS_KERB]. Also see the section "Security: Authenticating Identity Objects," in Chapter 9.

Incremental Functionality—Use of Policy Object Contents as Policy Locators

The policy management system must extract strings from the `POLICY_DATA` objects in parsed RSVP messages. These strings specify the following information:

- Kerberos-authenticated X.500 distinguished name. This is an NT domain user ID (generated automatically on behalf of all applications that use the Windows GQoS API).
- Application ID (generated by compliant QoS-aware applications).
- Subapplication ID (generated by compliant QoS-aware applications).

These strings should be offered to the network manager as *policy locators*. This means that the network manager should be able to use the policy management system to associate specific QoS policies or resources with combinations of these strings (just as they might otherwise associate specific policies or resources with specific IP addresses and ports).

In many organizations, enterprise-wide lists of user IDs and *organizational units* (OUs) are stored in Active Directory and are used for managing various user privileges. Policy management systems that use Active Directory and that participate in the Microsoft security infrastructure can offer the network manager the capability to provision network policies based on these lists.

Application identifiers and subidentifiers currently are not an integral part of the Microsoft Active Directory infrastructure (although they will be at some future date). A current list of application identifiers and subidentifiers signaled by well-known applications is available from Microsoft by request. This list can be offered to the network manager, by the policy management system, as a set of well-known application identifiers that can be associated with specific policies.

As more applications make use of the GQoS API, additional application identifiers and subidentifiers will appear on the network. The policy management system should provide a mechanism by which the additional identifiers can be added to the list of identifiers maintained by the system. This mechanism might automatically add new identifiers that appear on the network, and/or might allow the network manager to manually edit the list of available identifiers. In either case, the network manager should be offered the capability to associate policies with any of the identifiers listed by the system. See the section "Populating Policy Locators and Corresponding Policies," in Chapter 9.

Incremental Functionality—Use of SPI

For environments in which traffic is encrypted using IPSec, RSVP messages carry an SPI rather than IP ports and addresses. The format of the relevant RSVP objects is defined in [RFC 2207]. RSVP messages can still be used to learn mappings of applications and users to classification information. However, in this case, the classification information takes the form of an SPI rather than IP ports. Network management systems offering QoS management of IPSec-encrypted traffic must be capable of using the SPI that is signaled in RSVP messages to classify traffic.

Note

Note that the initial release of Windows 2000 does not carry an SPI in RSVP messages. This functionality is planned for a subsequent release.

Resource-Based Quantitative Admission Control Agents

Resource-based quantitative admission control is the minimal level of RSVP-based admission control functionality that should be offered by RSVP-conversant network devices. Devices that offer this functionality are *admission control agents*. A device that offers resource-based admission control must be capable of the following:

- Accepting configured admission control limits (in the form of token bucket parameters) for each service type that can be requested in an RSVP message
- Maintaining an account of the quantity of resources currently available for each service type (configured limits minus the total currently committed)
- Admitting or rejecting RSVP requests on this basis

Note

In resource-based admission control, the admit/reject decision is based solely on *resource* availability, with no regard for the user or the application associated with the request. Policy-based admission control considers user and application and is addressed later in this appendix. (Also see the sidebar in Chapter 14 titled "Resource- versus Policy-Based Admission Control").

Beyond the basic resource-based admission control, incremental functionality is described, which:

- Provides traffic-handling mechanisms at the device
- Enables the network manager to configure the device to append a DCLASS [RFC 2996] or TCLASS [RFC 2814] object to RSVP RESV messages transiting the device
- Polices in aggregate to quantitative traffic limits
- Instructs senders not to send additional traffic

Motivation

Enterprise network managers are expected to enable PEP/PDPs as admission control agents at key locations in the network, to leverage host-generated RSVP signaling. The application of resource-based admission control both enhances the quality experienced by certain applications and also makes optimal use of network resources (raising the QE product of the network). For an example, see the section "Supporting Higher Quality Services in the WAN" in Chapter 2.

Interaction Between Explicit Admission Control and Traffic Handling

The functionality described in this section as *admission control* is *explicit admission control* and is not necessarily linked to traffic handling or policing. (Policing is also known as *implicit admission control.* See the sidebar "Policing versus Admission Control," in Chapter 4). Explicit admission control can be used to coordinate resource allocation across PEP/PDPs and to control the behavior of cooperating senders.

Admission control alone can provide only very limited QoS. To more fully realize the advantages of QoS, it is necessary to also provide traffic-handling mechanisms in the network. Traffic-handling mechanisms may be provided in network devices that do not act as admission control agents, as well as in devices that do. When traffic-handling mechanisms are provided in devices that are also admission control agents, these mechanisms may or may not be linked to the admission control functionality of the device.

continues

In the conventional RSVP/IntServ model of a router, for example, the admission of an RSVP request results in the configuration of classification and traffic-handling mechanisms (in the same device) that allocate resources to the traffic described in the request and that provide the requested service level.

In a simpler model, the network device may implement aggregate traffic handling (based on DSCP or 802 `user_priority` values) without linking it to the admission control functionality of the device. In this case, the network manager may rely on a linkage of admission control and traffic handling that occurs outside of the device.

For example, assume that the network device supports aggregate traffic handling based on DSCP. Assume further that the only senders that mark DSCPs (for service other than best-effort service) cooperate, in the sense that they mark traffic only on flows that have been admitted through the process of explicit admission control. The specific DSCP marked is based on a mapping from the admitted IntServ service type to the DSCP. In this example, there is, in effect, a linkage between the explicit admission control enforced by the device and the traffic handling provided by the device. The linkage is implemented in the cooperating sender. This linkage enables the network manager to control the amount of traffic that will be handled at the device for each DSCP (and the corresponding PHB).

The examples of incremental functionality discussed in the following sections are based on various combinations of explicit admission control with aggregate traffic handling. These combinations are discussed throughout this book in various forms. See the following sections:

- Chapter 2: Figure 2.3 and the section "Combinations of Traffic Handling and Provisioning and Configuration Mechanisms"
- Chapter 5: the section "`DCLASS` and `TCLASS` Objects"
- Chapter 6: the section "DiffServ with RSVP"
- Chapter 7: the section "Aggregate Traffic Handling Based on 802 `user_priority`"
- Chapter 9: the section "Dynamic Policy Applied with Aggregate Traffic Handling"
- Chapter 10: Figure 10.15 and the section "Per-Conversation Admission Control to High-Quality Aggregate Traffic Classes on Private Leased Lines"
- Chapter 12: the section "Marking Behavior"
- Chapter 14: the section "Mapping Service Types to `user_priority` Marks"

Incremental Functionality—Resource-Based Admission Control

Resource-based admission control may be applied to RSVP PATH messages, RESV messages, or both. PEP/PDPs must offer the network manager (via a policy management interface) the capability to configure a table containing the information in Table D.1 for each interface.

Table D.1 Limits for Resource-Based Admission Control

IntServ Service Type	Total Send Resources Admissible	Total Receive Resources Admissible
Guaranteed	<token-bucket parameters>	<token-bucket parameters>
Controlled Load	<token-bucket parameters>	<token-bucket parameters>

It is not necessary to provide a table literally in this form. The most important aspect of the table illustrated is that it enables a network manager to specify admission control limits separately for each IntServ service type. It also enables the network manager to specify these limits separately for PATH messages and for RESV messages (PATH messages would be admitted based on the resource limits specified in the "Send Resources" column. RESV messages would be admitted based on the resource limits specified in the "Receive Resources" column.)

Note

In conventional RSVP processing, resource-based admission control is not applied to RSVP PATH messages, but rather to RESV messages. However, the capability to apply resource-based admission control to PATH messages may also be useful to network managers.

The limits in the table specify the *total* amount of resources that may be admitted on a particular interface for each service type across all currently admitted RSVP sessions. They do not specify per-session limits. Each RSVP request for a new session is admitted based on the total resources remaining at the time the request arrives. When the allowable resources have all been allotted, no additional requests will be admitted until resources are freed (by the expiration of an admitted request).

PEPs providing this functionality must be capable of parsing the appropriate RSVP messages and extracting the quantitative information (token bucket parameters) from the RSVP objects that describe a traffic flow. The PEP, in cooperation with the PDP, must then use the appropriate IntServ arithmetic [RFC 2210] to determine whether sufficient resources remain to be allocated to the traffic flow. If they do, the PEP/PDP should approve the resource request by allowing the message to pass unhindered (per standard

RSVP processing) and should reduce the amount of remaining resources accordingly. If remaining resources are insufficient to accommodate the resource request, the PEP/PDP should reject the resource request by sending an appropriate error message to the sender and/or the receiver.

Note

In the case of rejection of PATH messages, a PATH_ERROR error message should be sent to the sender. This message should include information regarding the reason for rejection in a POLICY_DATA object.

Incremental Functionality—Aggregate Traffic Handling

PEP/PDPs offering explicit admission control functionality can be enhanced by offering aggregate traffic-handling functionality in the form of DiffServ PHBs or support for two or more 802 user_priority traffic classes. In a minimal implementation, no linkage is required in the network device between the explicit admission control functionality and the aggregate traffic-handling mechanism.

In this case, network managers assume a mapping from the IntServ service type specified in an RSVP request to one of the aggregate traffic-handling classes supported by the PEP. The PEP need not be aware of this mapping. Senders are assumed to mark traffic on admitted flows with a DSCP or 802 user_priority value based on this mapping. Thus, the network manager can use a table of the form illustrated in Table D.1 to control the amount of traffic arriving at the device that is marked for each of the aggregate service levels supported by the PEP.

Note

For optimal flexibility in the case of DiffServ PEPs, the network manager should be able to define the mapping of DSCP to PHB within the PEP.

*Incremental Functionality—*DCLASS *or* TCLASS *Object Support*

Incremental functionality would extend Table D.1 to include an additional column, labeled "DCLASS" (specifies DSCP for Layer 3 devices) or "TCLASS" (specifies 802 user_priority for Layer 2 devices), as shown in Table D.2.

Table D.2 Limits for Resource-Based Admission Control with DCLASS/TCLASS Specification

IntServ Service Type	Total Send Resources Admissible	Total Receive Resources Admissible	DCLASS/TCLASS to Return with Admitted RESV Messages
Guaranteed	<token-bucket parameters>	<token-bucket parameters>	DCLASS 01 or TCLASS 01
Controlled Load	<token-bucket parameters>	<token-bucket parameters>	DCLASS 02 or TCLASS 02

The PEP would be required to append a DCLASS object (for DSCP) or a TCLASS object (for 802 user_priority) carrying the specified value to RESV messages corresponding to admitted flows. This mechanism enables network managers to drive the mapping from IntServ service to DSCP or 802 user_priority that is applied by upstream senders. See [RFC 2996] and [RFC 2814].

Incremental Functionality—Handling Rejected Traffic

In the usage described in the previous section, a DCLASS or TCLASS object is returned, with RESV messages, only for *admitted* traffic flows. Rejected traffic is thus relegated to best-effort service. In many cases, the network manager may wish to actually *demote* rejected traffic to an less-than-best-effort (LBE) service or to outright refuse its admission to the network. This functionality is particularly useful to facilitate the deployment of aggressive UDP applications in a manner that enables the network manager to protect the network resources.

Rejected traffic can be handled by the following methods:

- Relegating it to best-effort service (as described previously)
- Demoting it to LBE service by rejecting the RSVP request and returning a corresponding DCLASS/TCLASS to the sender with a PATH_ERR message
- Refusing admission of the traffic by instructing the sender not to send, using a DO_NOT_SEND policy error in a PATH_ERR message (see section titled "Withholding Transmission," in Chapter 12).

Note

The functionality described is documented in the IETF draft [POLICY_ERRS].

In order to support this functionality, Tables D.1 and D.2 should be enhanced as shown in Table D.3.

Table D.3 Table for Enhanced Admission Control Policies

IntServ Service Type	Admitted Traffic		Relegated to LBE		DENIED
	TB Limit	D/TCLASS	TB Limit	D/TCLASS	
Guaranteed	$TB1_{guar}$	D/TCLASS01	$TB2_{guar}$	D/TCLASS03	$TB3_{guar}$
Controlled Load	$TB1_{cntl\text{-}ld}$	D/TCLASS02	$TB2_{cntl\text{-}ld}$	D/TCLASS04	$TB3_{cntl\text{-}ld}$

Note that there are 6 token-bucket profiles specified in the table—three for Guaranteed Service and three for Controlled Load Service. These should be interpreted as follows:

- TB1—Requests for resources for the corresponding service type *up to* this threshold should be admitted. The specified DCLASS/TCLASS value should be returned to the sender with the RSVP RESV message.
- TB2—Requests for resources for the corresponding service *beyond* this threshold, but *below* TB3 should be rejected and the specified DCLASS/TCLASS value (corresponding to LBE) should be returned with an RSVP PATH_ERR message.
- TB3—Requests for resources for the corresponding service *beyond* this threshold should be rejected and the DO_NOT_SEND policy error should be returned with an RSVP PATH_ERR message.

It is always true that TB3 ≥ TB2 ≥ TB1. Note that if TB2 = TB1, then traffic is either marked for preferred service, for LBE service, or refused. However, if TB2 > TB1, then traffic exceeding TB1 but less than TB2 is marked neither for preferred service nor for LBE service. This traffic is unmarked and is relegated to best-effort service.

The functionality described in this section is likely to be most useful when applied on a per application or per-user basis as described in the subsequent section titled "Policy-Based Admission Control Agents."

Incremental Functionality—Aggregate Policing

The functionality described so far enables the network manager to control the amount of traffic marked for a certain service level only to the extent that senders cooperate. Senders are trusted to mark traffic for preferential treatment only on admitted flows, to mark certain traffic for LBE service, and possibly even to refrain from sending certain traffic.

Incremental functionality would further link explicit admission control and aggregate traffic handling by policing traffic to the limits specified in Tables D.1, D.2, or D.3. Traffic submitted in excess of the configured token bucket limits would be either discarded or demoted to a best-effort or less-than-best-effort aggregate service level.

Note that traffic would not be policed on a per-conversation basis. Instead, all traffic marked for a certain DSCP or 802 `user_priority` value would be policed in aggregate to the token bucket limits configured in the corresponding row from the configuration table (per the mapping of IntServ service type to DSCP or 802 `user_priority`).

This form of aggregate policing is particularly useful to service providers offering quantifiable resources at each of a number of service levels (such as might be offered by a DiffServ network in the form of a service-level agreement). It ensures that the cumulative resources used by upstream marking devices do not exceed the cumulative resources offered to these devices at each service level. It protects network resources from abuse by senders that do not mark in accordance with the rules. However, it does not protect well-behaved senders from rogue senders that send traffic through the same devices.

Policy-Based Admission Control Agents

PEP/PDP combinations supporting this functionality enable the network manager to admit or reject requests for resources based not only on the total quantity of available or admissible resources, but also on per-user or per-application admissible limits.

Motivation

Pure resource-based quantitative admission control (as described in the previous section) enables the network manager to control the use of network resources in PEPs on a per-service level and per-interface basis. However, this is a first-come, first-serve mechanism. Although it can be applied effectively to protect network resources and to enhance service to applications, it does not enable the network manager to control which users and applications are provided prioritized resources in the network. Because prioritized resources are generally costly, and because there are generally insufficient prioritized resources for all requesting applications or users, network managers require the capability to apply user- and application-based policies in determining which traffic is entitled to various resources and which is not.

Incremental Functionality—Policy-Based Admission Control

The functionality for policy-based admission control builds on the functionality required for resource-based admission control. In the case of pure resource-based admission control, the network manager uses a per-interface table to specify the quantifiable admissible

resource limits per service level (see Tables D.1, D.2, and D.3). In the case of policy-based admission control, the policy management system must present the network manager with additional configuration options that are based on the user and application associated with admitted traffic. Because of the added complexity of policy-based admission control, this functionality is usually implemented with the help of a policy server and a policy data store that are separate from the switch, router, or alternate PEP device.

Note

Where PDP and PEP functionality is separated into distinct hardware systems, these should each converse using the COPS protocol. This is necessary to promote interoperability between PEPs and PDPs provided by different vendors. The use of COPS for applying policy to RSVP requests is standardized in [RFC 2749].

To support policy-based admission control, the following must be true:

- The policy management system must provide a provisioning interface that can be used to associate certain policy locators (described in the section "RSVP Message Parsing Functionality," earlier in this appendix) or combinations thereof, with corresponding resource limits (and optionally with `DCLASS` and `TCLASS` objects or policy objects such as the `DO_NOT_SEND` object).
- When RSVP requests arrive at a PEP, the PEP should extract user and application identity objects from the messages (also described in the earlier section "RSVP Message Parsing Functionality"). These should be passed to the PDP. The PDP should use these as policy locators to locate the associated resource limits and optional `DCLASS`, `TCLASS`, or policy objects.

Population of Policies versus Application of Policies

It is worth elaborating on the two activities described. One activity is the *population* of the policy system with policy locator strings and associated QoS policies. The policy system may be populated beforehand with both policy locator strings and the associated policies (for example, resource limits or values for returned `DCLASS` and `TCLASS` objects). In addition, the policy system may "accumulate" or "learn" policy locator strings as applications signal them to the network. No policies can actually be applied to such accumulated policy locators until the network manager associates policies with them. (See the previous section "Incremental Functionality—Use of Policy Object Contents as Policy Locators.")

The second activity is the *application* of policies. As applications signal policy locators through the network, policy systems participate in the signaling process and apply the policies previously associated with the policy locators.

Upon receipt of an RSVP request, the PDP locates the applicable policies (as described previously). The resources requested in the RSVP message are then compared against the resources allowed per the located policies. If sufficient resources remain, the RSVP request should be admitted and the PDP should return an "admit" decision to the PEP. If there are insufficient resources, the PDP should return a "reject" decision to the PEP. The PDP should maintain the appropriate resource accounting, and the PEP should reflect the decision to the network using the appropriate RSVP error messages (as described previously and in [RFC 2750]). In addition to returning the admit/reject decision, if a DCLASS or TCLASS object has been associated with the policy locators (as illustrated in Tables D.2 and D.3), the PDP must supply this object to the PEP, to be appended to RSVP RESV messages or to RSVP PATH_ERR messages. In certain cases, policy may dictate that a certain traffic flow be refused admission altogether, in which case, the PDP should supply the DO_NOT_SEND policy object to the PEP, to be appended to the appropriate RSVP PATH_ERR messages.

Note

Note that in the case of policy-based admission control, resource limits and DCLASS/TCLASS objects are associated with specific users or applications. These limits do not supersede those limits applied for resource-based admission control. Instead, the net result of any policy decision reflects the application of both sets of limits.

This appendix does not specify the exact combinations of policy locators that can be associated with resource limits and DCLASS and TCLASS objects. Instead, it specifies the format of the policy locators, the associated resource-based quantitative parameters, and the DCLASS and TCLASS objects, all of which are carried in RSVP signaling messages. The format of the policy objects is specified in [RFC 2752] and [RFC 2872]. Resource limits are specified per the token bucket parameters tabulated previously.

The specification of the combinations of policy locators that are used to locate associated resources, the rules describing precedence among various policy locators, their format, and the format of the associated resource parameters are jointly referred to as the *policy schemas* of the policy management system. The ACS, which is discussed in Chapter 14, exemplifies a schema that uses Windows 2000 Kerberos authenticated user IDs (and the subnet of the parsed RSVP message) to locate associated per-service-level resource limits. The example schema does not use application identifiers and subapplication identifiers as policy locators. It also does not allow the network manager to associate DCLASS or TCLASS values with combinations of policy locators, nor to return the DO_NOT_SEND policy error. A fully featured policy management system would support the use of application identifiers and subidentifiers, as well as the capability to associate DCLASS or TCLASS values and the ability to return the DO_NOT_SEND policy error [POLICY_ERRS].

Incremental Functionality—Null Service Support

In the case of the Null Service (see the section "The Null Service," in Chapter 4), RSVP messages do not quantify required resources. In this case, the policy management system may still apply an admit/reject decision, but it is not based on a simple arithmetic calculation of requested resources against available resources. Instead, the admit/reject decision is typically based on a maximum number of admissible flows (corresponding to a specific user, group of users, or application) that can be specified at provisioning time by the network manager. Policy management systems that offer support for the Null Service must allow the network manager to specify a `TCLASS` or `DCLASS` object to be returned in response to requests for qualitative services. The effect of policy in this case is to admit or reject a flow and to determine the appropriate marking (DSCP or 802 `user_priority`) for admitted flows.

Incremental Functionality—Use of Existing Active Directory Schema

Microsoft's Active Directory currently supports quantitative QoS schemas, as described in Chapter 14. The Microsoft Local Policy Module (MS-LPM)—available directly from Microsoft—can be used to parse the Active Directory quantitative QoS schema. Policy systems based on Windows 2000 may choose to incorporate this LPM. See the section "Using the Microsoft LPM," in Chapter 14.

Incremental Functionality—Extending Active Directory

Third-party vendors of policy management systems are encouraged to enhance Microsoft's quantitative schema by adding support for application identifiers and subidentifiers, as well as the capability to associate `DCLASS` and `TCLASS` objects and the `DO_NOT_SEND` policy error with these policy locators. Documentation and examples for developing custom Active Directory schemas can be found in [SDK]. This includes information on the Active Directory Services Interface (ADSI) and the Microsoft Management Console (MMC).

Regarding the Use of Active Directory

Note that policy-based admission control, in general, does not necessarily require the use of Active Directory (or any other directory, for that matter). Policy-based admission control refers to the capability to make an admission control decision, at a PEP or PDP, subject to policy dictated by the network manager. Active Directory, however, is a well-suited repository for network policy information. In addition, enterprise network managers deploying Windows 2000-based hosts make broad use of Active Directory to manage users and applications. Thus, there are synergies to be realized by enabling PEPs and PDPs to use policy information from Active Directory. Microsoft has just begun to realize the

potential of Active Directory with respect to QoS policy information. Vendors are encouraged to expand on this usage.

DSBM Functionality

Layer 2 devices that purport to offer any form of RSVP-based functionality must be capable of intercepting and parsing RSVP messages. As such, they generally must become the DSBM on the subnetwork on which they reside. (As an exception, if every device in the shared subnetwork is RSVP-aware, then these may cooperatively provide RSVP functionality without any of the devices officially becoming the DSBM.) See Chapter 7 for a discussion of SBMs, DSBMs, and QoS on Layer 2 devices.

Layer 3 devices may also provide DSBM functionality. However, it is *sufficient* for these to provide only SBM *client* functionality to participate in RSVP signaling. In the absence of DSBM-capable switches, DSBM-capable routers take on the role of DSBM. Similarly, in the absence of DSBM-capable switches or routers, DSBM-capable hosts take on this functionality. DSBM functionality is ideally implemented in switches.

Motivation

DSBM-aware Layer 2 devices enable network managers to operate their Layer 2 networks at an improved quality/efficiency product. In addition, DSBM functionality is recommended if Layer 2 devices are to offer PEP/PDP functionality linked to RSVP signaling (such as required for participation in policy-based admission control).

Incremental Functionality—DSBM

DSBM functionality includes the capability to run for election as DSBM. Switches and other Layer 2 devices should run at the highest priority, with routers at second priority. Devices capable of becoming the DSBM must provide a mechanism by which their DSBM functionality can be disabled. Incremental functionality that may be provided by DSBMs includes the capability to append the `NonResvSendLimit` (see the section "`NonResvSendLimit`," in Chapter 7) to `I_AM_DSBM` messages. This functionality enables network managers to limit the amount of traffic sent by QoS-aware applications on shared segments. A user interface must be provided to enable the network manager to set the limits advertised for the `NonResvSendLimit`.