Microsoft

Operating System

Quality of Service Technical White Paper

White Paper

Abstract

Within the past few years, there has been a rapid growth in network traffic. New applications, particularly multimedia applications, have placed increasing demands on networks, straining their ability to provide customers with a satisfactory experience. In answer to this situation, numerous mechanisms have surfaced for providing quality of service (QoS) networks. The ultimate goal of these mechanisms is to provide improved network service to the applications at the edges of the network. This white paper reviews emerging QoS mechanisms and how they are integrated to optimize the utilization of network resources. It then specifically discusses Microsoft's QoS mechanisms.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

0899

Table of Contents

1 Introduction

During the past several years, numerous mechanisms have surfaced for providing quality of service (QoS) networks. The ultimate goal of these mechanisms is to provide improved network 'service' to the applications at the edges of the network. This whitepaper reviews emerging QoS mechanisms and how they are integrated to optimize the utilization of network resources. It then specifically discusses Microsoft's QoS mechanisms.

1.1 Organization of this Document

- Chapters 2 and 3 define network QoS and introduce the basic QoS technologies available.
- Chapter 4 discusses the varying levels of guarantees that can be expected from a QoS-enabled network and the tradeoffs that can be expected in providing them.
- Chapters 5 and 6 introduce a sample network incorporating the QoS mechanisms discussed and describe how the various mechanisms can be integrated to provide end-to-end QoS functionality.
- Chapter 7 describes the application of policies in the QoS enabled network.
- Chapter 8 describes Microsoft's QoS components in detail.
- Chapter 9 describes the current level of support for various QoS mechanisms in generally third party network equipment.
- Chapter 10 includes references to Internet Engineering Task Force (IETF) documents describing the QoS mechanisms discussed in this whitepaper.

2 What is Network QoS?

Let's assume the following simplistic view of the host/network system: Applications run on hosts and exchange information with their peers. Applications send data by submitting it to the operating system, to be carried across the network. Once data is submitted to the operating system, it becomes network *traffic*. Network QoS refers to the ability of the network[1] to handle this traffic such that it meets the service needs of certain applications. This requires fundamental traffic handling mechanisms in the network, the ability to identify traffic that is entitled to these mechanisms and the ability to control these mechanisms.

QoS functionality can be perceived to satisfy two customers - network applications and network administrators. It appears that these are often at odds, since in many cases the network administrator limits the resources used by a particular application while the application attempts to seize resources from the network. These apparently conflicting goals can be reconciled by realizing that the network administrator is chartered with maximizing the utility of the network *across the full range* of applications and users.

2.1 QoS Parameters

Different applications have different requirements regarding the handling of their traffic in the network. Applications generate traffic at varying rates and generally require that the network be able to carry traffic at the rate at which they generate it. In addition, applications are more or less tolerant of traffic delays in the network and of variation in traffic delay. Certain applications can tolerate some degree of traffic loss while others cannot. These requirements are expressed using the following QoS-related parameters:

- Bandwidth - the rate at which an application's traffic must be carried by the network
- Latency - the delay that an application can tolerate in delivering a packet of data
- Jitter - the variation in latency
- Loss - the percentage of lost data

[1] We consider the *network* to include host network related software and hardware as well as any network equipment that resides between communicating hosts.

If infinite network resources were available, then all application traffic could be carried at the required bandwidth, with zero latency, zero jitter and zero loss. However, network resources are not infinite. As a result, there are parts of the network in which resources are unable to meet demand. QoS mechanisms work by controlling the allocation of network resources to application traffic in a manner that meets the application's service requirements.[2]

2.2 Fundamental QoS Resources and Traffic Handling Mechanisms

Networks interconnect hosts using a variety of network devices, including host network adapters, routers, switches, and hubs. Each of these contains network *interfaces.* The interfaces interconnect the various devices via cables and fibers. Network devices generally use a combination of hardware and software to *forward* traffic from one interface to another.[3] Each interface can send and receive traffic at a finite rate. If the rate at which traffic is directed to an interface exceeds the rate at which the interface can forward the traffic onward, then *congestion* occurs. Network devices may handle this condition by *queuing* traffic in the device's memory until the congestion subsides. In other cases, network equipment may discard traffic to alleviate congestion. As a result, applications experience varying latency (as traffic backs up in queues, on interfaces) or traffic loss.

The capacity of interfaces to forward traffic and the memory available to store traffic in network devices (until it can be forwarded) are the fundamental resources that are required to provide QoS to application traffic flows. Mechanisms internal to network devices determine which traffic gets preferential access to these resources. These are the fundamental traffic handling mechanisms that comprise the QoS enabled network.

2.3 Allocating QoS Resources in Network Devices

Devices that provide QoS support do so by intelligently allocating resources to submitted traffic. For example, under congestion, a network device might choose to queue the traffic of applications that are more latency-tolerant instead of the traffic of applications that are less latency-tolerant. As a result, the traffic of applications that are less latency-tolerant can be forwarded immediately to the next network device. In this example, interface capacity is a resource that is granted to the latency-intolerant traffic. Device memory is a resource that has been granted to the latency-tolerant traffic.

In order to allot resources preferentially to certain traffic, it is necessary to identify different traffic and to associate it with certain resources. This is typically achieved as follows: Traffic arriving at network devices is identified in each device and is separated into distinct *flows*[4] via the process of *packet classification*. Traffic from each flow is directed to a corresponding *queue*. The queues are then *serviced* according to some *queue-servicing algorithm.* The queue-servicing algorithm determines the rate at which traffic from each queue is submitted to the network, thereby determining the resources that are allotted to each queue and to the corresponding flows. Thus, in order to provide network QoS, it is necessary to provision the following in network devices:

- Classification information by which devices separate traffic into flows.
- Queues and queue-servicing algorithms that handle traffic from the separate flows.

We will refer to these jointly as *traffic handling mechanisms*. These traffic-handling mechanisms must be provisioned or configured in a manner that provides useful end-to-end services across a network. As such,

[2] Certain applications adapt (within limits) to network conditions. These applications can be said to implement a form of *application QoS*. In this discussion, we focus on network QoS mechanisms rather than application QoS.

[3] Hosts typically include only a single network interface that is used to forward traffic from applications to the network or from the network to applications.

[4] For the purpose of this discussion, a flow is a subset of all packets passing through a network device, which has uniform QoS requirements.

the various QoS technologies that we will discuss will fall into the category of a traffic handling mechanism or a provisioning or configuration mechanism.

3 QoS Technologies

In the following sections, we describe QoS traffic handling mechanisms and the associated provisioning and configuration mechanisms.

3.1 Traffic Handling Mechanisms

In this section we discuss the more significant traffic handling mechanisms. Note that underlying any traffic handling mechanism is a set of queues and the algorithms for servicing these queues. (In Appendix A of this whitepaper, we discuss some general approaches to queuing and queue-servicing). Traffic handling mechanisms include:

- 802.1p
- Differentiated service per-hop-behaviors (diffserv)
- Integrated services (intserv)
- ATM, ISSLOW and others

Each of these traffic-handling mechanisms is appropriate for specific media or circumstances and is described in detail below.

3.1.1 802.1p

Most local area networks (LANs) are based on *IEEE 802* technology. These include Ethernet, token-ring, FDDI and other variations of shared media networks. *802.1p* is a traffic-handling mechanism for supporting QoS in these networks[5]. QoS in LAN networks is of interest because these networks comprise a large percentage of the networks in use in university campuses, corporate campuses and office complexes.

802.1p[6] defines a field in the layer-2 header of 802 packets that can carry one of eight priority values. Typically, hosts or routers sending traffic into a LAN will mark each transmitted packet with the appropriate priority value. LAN devices, such as switches, bridges and hubs, are expected to treat the packets accordingly (by making use of underlying queuing mechanisms). The scope of the 802.1p priority mark is limited to the LAN. Once packets are carried off the LAN, through a layer-3 device, the 802.1p priority is removed.

3.1.2 Differentiated Services (Diffserv)

Diffserv[7] is a layer-3 QoS mechanism that has been in limited use for many years, although there has been little effort to standardize it until very recently. Diffserv defines a field in the layer-3 header of IP packets, called the diffserv codepoint (DSCP)[8]. Typically, hosts or routers sending traffic into a diffserv network will mark each transmitted packet with the appropriate DSCP. Routers within the diffserv network use the DSCP to classify packets and apply specific queuing or scheduling behavior (known as a *per-hop behavior* or *PHB*) based on the results of the classification.

[5] Since LAN resources tend to be less costly than WAN resources, 802.1p QoS mechanisms are often considered less important than their WAN related counterparts. However, with the increasing usage of multimedia applications on LANs, delays through LAN switches do become problematic. 802.1p tackles these delays.

[6] 802.1p is often defined together with 802.1q. The two define various VLAN (virtual LAN) fields, as well as a priority field. For the purpose of this discussion, we are interested only in the priority field.

[7] a.k.a. *Class of Service*

[8] The DSCP is a six-bit field, spanning the fields formerly known as the type-of-service (TOS) fields and the IP precedence fields.

An example of a PHB is the *expedited-forwarding* (or EF) PHB. This behavior is defined to assure that packets are transmitted from ingress to egress (at some limited rate) with very low latency. Other behaviors may specify that packets are to be given a certain priority relative to other packets, in terms of average throughput or in terms of drop preference, but with no particular emphasis on latency. PHBs are implemented using underlying queuing mechanisms.

PHBs are individual behaviors applied at each router. PHBs alone make no guarantees of end-to-end QoS. However, by concatenating routers with the same PHBs (and limiting the rate at which packets are submitted for any PHB), it is possible to use PHBs to construct an end-to-end QoS service. For example, a concatenation of EF PHBs, along a pre-specified route, with careful admission control, can yield a service similar to leased-line service, which is suitable for interactive voice. Other concatenations of PHBs may yield a service suitable for video playback, and so forth.

3.1.3 Integrated Services (Intserv)

Intserv is a service framework. At this time, there are two services defined within this framework. These are the *guaranteed* service and the *controlled load* service. The guaranteed service promises to carry a certain traffic volume with a quantifiable, bounded latency. The controlled load service agrees to carry a certain traffic volume with the 'appearance of a lightly loaded network'. These are *quantifiable* services in the sense that they are defined to provide quantifiable QoS to a specific quantity of traffic. (As we will discuss in depth later, certain diffserv services by comparison, may not be quantifiable).

Intserv services are typically (but not necessarily) associated with the RSVP signaling protocol, which will be discussed in detail later in this whitepaper. Each of the intserv services define *admission control* algorithms which determine how much traffic can be admitted to an intserv service class at a particular network device, without compromising the quality of the service. Intserv services do not define the underlying queuing algorithms to be used in providing the service.

3.1.4 ATM, ISSLOW and Others

ATM is a link layer technology that offers high quality traffic handling. ATM fragments packets into link layer *cells*, which are then queued and serviced using queue-servicing algorithms appropriate for the particular ATM service. ATM traffic is carried on *virtual circuits* (VC*)* which support one of the numerous ATM services. These include constant-bit-rate (CBR), variable-bit-rate (VBR), unknown-bit-rate (UBR) and others. ATM actually goes beyond a strict traffic handling mechanism in the sense that it includes a low level signaling protocol that can be used to set up and tear down ATM VCs.

Because ATM fragments packets into relatively small cells, it can offer very low latency service. If it is necessary to transmit a packet urgently, the ATM interface can always be cleared for transmission in the time it takes to transmit one cell. By comparison, consider sending normal TCP/IP data traffic on slow modem links without the benefit of the ATM link layer. A typical 1500-byte packet, once submitted for transmission on a 28.8 Kbps modem link, will occupy the link for about 400 msec until it is completely transmitted (preventing the transmission of any other packets on the same link). *Integrated Services Over Slow Link Layers* (ISSLOW) addresses this problem. ISSLOW is a technique for fragmenting IP packets at the link layer for transmission over slow links such that the fragments never occupy the link for longer than some threshold.

Other traffic handling mechanisms have been defined for various media, including cable modems, hybrid fiber coax (HFC) plants, P1394, and so on. These may use low level, link-layer specific signaling mechanisms (such as UNI signaling for ATM).

3.1.5 Per-Conversation vs. Aggregate Traffic Handling Mechanisms

An important general categorization of traffic handling mechanisms is that of *per-conversation* mechanisms vs. *aggregate* mechanisms. This categorization refers largely to the classification associated with the mechanism and can have a significant effect on the QoS experienced by traffic subjected to the mechanism.

Per-conversation traffic handling mechanisms are mechanisms that handle each *conversation* as a separate flow. In this context, a conversation includes all traffic between a specific instance of a specific application on one host and a specific instance of the peer application on a peer host. In the case of IP traffic, the source/destination IP address, port, and protocol (also known as a 5-tuple) uniquely identify a conversation. Traditionally, intserv mechanisms are provided on a per-conversation basis.

In aggregate traffic handling mechanisms, some set of traffic, from multiple conversations, is classified to the same flow and is handled in aggregate. Aggregate classifiers generally look at some aggregate identifier in packet headers. Diffserv and 802.1p are examples of aggregate traffic handling mechanisms at layer-3 and at layer-2, respectively. In both these mechanisms, packets corresponding to multiple conversations are marked with the same DSCP or 802.1p mark.

When traffic is handled on a per-conversation basis, resources are allotted on a per-conversation basis. From the application perspective, this means that the application's traffic is granted resources completely independent of the effects of traffic from other conversations in the network. While this tends to enhance the quality of the service experienced by the application, it also imposes a burden on the network equipment. Network equipment is required to maintain independent state for each conversation and to apply independent processing for each conversation. In the core of large networks, where it is possible to support millions of conversations simultaneously, per-conversation traffic handling may not be practical.

When traffic is handled in aggregate, the state maintenance and processing burden on devices in the core of a large network is reduced significantly. On the other hand, the quality of service perceived by an application's conversation is no longer independent of the effects of traffic from other conversations that have been aggregated into the same flow. As a result, in aggregate traffic handling, the quality of service perceived by the application tends to be somewhat compromised. Allocating excess resources to the aggregate traffic class can offset this effect. However, this approach tends to reduce the efficiency with which network resources are used.

3.2 Provisioning and Configuration Mechanisms

In order to be effective in providing network QoS, it is necessary to effect the provisioning and configuration of the traffic handling mechanisms described consistently, across multiple network devices. Provisioning and configuration mechanisms include:

- Resource Reservation Protocol (RSVP) signaling and the Subnet Bandwidth Manager (SBM)
- Policy mechanisms and protocols
- Management tools and protocols

These are described in detail in the paragraphs below.

3.2.1 Provisioning vs. Configuration

In this whitepaper, we use the term *provisioning* to refer to more static and longer term management tasks. These may include selection of network equipment, replacement of network equipment, interface additions or deletions, link speed modifications, topology changes, capacity planning, and so forth. We use the term *configuration* to refer to more dynamic and shorter term management tasks. These include such management tasks as modifications to traffic handling parameters in diffserv networks. The distinction between provisioning and configuration is not clearly delineated and is used as a general guideline rather than a strict categorization. The terms are often used interchangeably unless otherwise specified.

3.2.2 Top-Down vs. Signaled Mechanisms

It is important to note the distinction between *top-down* QoS configuration mechanisms and *signaled* QoS configuration mechanisms. Top-down mechanisms typically 'push' configuration information from a management console down to network devices. Signaled mechanisms typically carry QoS requests (and

implicit configuration requests) from one end of the network to the other, along the same path traversed by the data that requires QoS resources. Top-down configuration is typically initiated on behalf of one or more applications by a network management program. Signaled configuration is typically initiated by an application's changes in resource demands.

3.2.3 RSVP and the SBM

RSVP is a signaled QoS configuration mechanism. It is a protocol by which applications can request end-to-end, per-conversation, QoS from the network, and can indicate QoS requirements and capabilities to peer applications. RSVP is a layer-3 protocol, suited primarily for use with IP traffic. As currently defined, RSVP uses intserv semantics to convey per-conversation QoS requests to the network. However, RSVP per-se is neither limited to per-conversation usage, nor to intserv semantics. In fact, currently proposed extensions to RSVP enable it to be used to signal information regarding traffic aggregates. Other extensions enable it to be used to signal requirements for services beyond the traditional guaranteed and controlled load intserv services. In this section we discuss RSVP in its traditional per-conversation, intserv form. Later in this whitepaper we will discuss its applicability to aggregated services and to services which are not traditionally intserv.

Since RSVP is a layer-3 protocol, it is largely independent of the various underlying network media over which it operates. Therefore, RSVP can be considered an abstraction layer between applications (or host operating systems) and media-specific QoS mechanisms.

There are two significant RSVP messages, PATH and RESV. Transmitting applications send PATH messages towards receivers. These messages describe the data that will be transmitted and follow the path that the data will take. Receivers send RESV messages. These follow the path seeded by the PATH messages, back towards the senders, indicating the profile of traffic that particular receivers are interested in. In the case of multicast traffic flows, RESV messages from multiple receivers are 'merged', making RSVP suitable for QoS with multicast traffic.

As defined today, RSVP messages carry the following information:

- How the network can identify traffic on a conversation (classification information)
- Quantitative parameters describing the traffic on the conversation (data rate, etc.)
- The service type required from the network for the conversation's traffic
- Policy information (identifying the user requesting resources for the traffic and the application to which it corresponds)

Classification information is conveyed using IP source and destination addresses and ports. In the conventional intserv use of RSVP, an Intserv service type is specified and quantitative traffic parameters are expressed using a *token-bucket model*. Policy information is typically a secure means for identifying the user and/or the application requesting resources. Network administrators use policy information to decide whether or not to allocate resources to a conversation.

3.2.3.1 How RSVP Works

PATH messages wind their way through all network devices en-route from sender to receivers. RSVP aware devices in the data path note the messages and establish state for the flow described by the message. (Other devices pass the messages through transparently).

When a PATH message arrives at a receiver, the receiver responds with a RESV message (if the receiving application is interested in the traffic flow offered by the sender). The RESV message winds its way back towards the sender, following the path established by the incident PATH messages. As the RESV message progresses toward the sender, RSVP-aware devices verify that they have the resources necessary to meet the QoS requirements requested. If a device can accommodate the resource request, it installs classification state corresponding to the conversation and allocates resources for the conversation. The device then allows

the RESV message to progress on up toward the sender. If a device cannot accommodate the resource request, the RESV message is rejected and a rejection is sent back to the receiver.

In addition, RSVP aware devices in the data path may extract policy information from PATH messages and/or RESV messages, for verification against network policies. Devices may reject resource requests based on the results of these policy checks by preventing the message from continuing on its path, and sending a rejection message.

When requests are not rejected for either resource availability or policy reasons, the incident PATH message is carried from sender to receiver, and a RESV message is carried in return. In this case, a *reservation* is said to be installed. An installed reservation indicates that RSVP-aware devices in the traffic path have committed the requested resources to the appropriate flow and are prepared to allocate these resources to traffic belonging to the flow. This process of approving or rejecting RSVP messages is known as *admission-control* and is a key QoS concept.

3.2.3.2 The SBM

The SBM is based on an enhancement to the RSVP protocol, which extends its utility to shared networks. In shared sub-networks or LANs (which may include a number of hosts and/or routers interconnected by a switch or hub), standard RSVP falls short. The problem arises because RSVP messages may pass through layer-2 (RSVP-unaware) devices in the shared network, implicitly admitting flows that require shared network resources. RSVP-aware hosts and routers admit or reject flows based on availability of their private resources, but not based on availability of shared resources. As a result, RSVP requests destined for hosts on the shared subnet may result in the over-commitment of resources in the shared subnet.

The SBM solves this problem by enabling intelligent devices that reside on the shared network to volunteer their services as a 'broker' for the shared network's resources. Eligible devices are (in increasing order of suitability):

- Attached SBM-capable hosts
- Attached SBM-capable routers
- SBM-capable switches which comprise the shared network

These devices automatically run an election protocol that results in the most suitable device(s) being appointed *designated* SBMs (DSBM). When eligible switches participate in the election, they subdivide the shared network between themselves based on the layer-2 network topology. Hosts and routers that send into the shared network discover the closest DSBM and route RSVP messages through the device. Thus, the DSBM sees all messages that will affect resources in the shared subnet and provides admission control on behalf of the subnet.

3.2.4 Policy Mechanisms and Protocols

Network administrators configure QoS mechanisms subject to certain policies. Policies determine which applications and users are entitled to varying amounts of resources in different parts of the network.

Policy components include:

- A data-store, which contains the policy data itself, such as user names, applications, and the network resources to which these are entitled.
- Policy decision points (PDPs) - these translate network-wide higher layer policies into specific configuration information for individual network devices. PDPs also inspect resource requests carried in RSVP messages and accept or reject them based on a comparison against policy data.
- Policy enforcement points (PEPs) act on the decisions made by PDPs. These are typically network devices that either do or do not grant resources to arriving traffic.
- Protocols between the data-store, PDPs and PEPs

3.2.4.1 Policy Data Store - Directory Services

Policy mechanisms rely on a set of data describing how resources in various parts of the network can be allocated to traffic that is associated with specific users and/or applications. Policy *schemas* define the format of this information. Two general types of schemas are required. One type describes the resources that should be allocated in a top-down provisioned manner. The other describes resources that can be configured via end-to-end signaling. This information tends to be relatively static and (at least in part) needs to be distributed across the network. Consequently, directories tend to be suitable data stores.

3.2.4.2 Policy Decision Points and Policy Enforcement Points

Policy decision points (PDPs) interpret data stored in the schemas and control policy enforcement points (PEPs) accordingly. Policy enforcement points are the switches and routers through which traffic passes. These devices have the ultimate control over which traffic is allocated resources and which is not. In the case of top-down provisioned QoS, the PDP 'pushes' policy information to PEPs in the form of classification information (IP addresses and ports) and the resources to which classified packets are entitled.

In the case of signaled QoS, RSVP messages transit through the network along the data path. When an RSVP message arrives at a PEP, the device extracts a *policy element* from the message, as well as a description of the service type required and the traffic profile. The policy element generally contains authenticated user and/or application identification. The router then passes the relevant information from the RSVP message to the PDP for comparison of the resources requested against those allowable for the user and/or application (per policy in the data-store). The PDP makes a decision regarding the admissibility of the resource request and returns an approval or denial to the PEP.

In certain cases, the PEP and the PDP can be co-located in the network device. In other cases, the PDP may be separated from the PEP in the form of a *policy server*. A single policy server may reside between the directory and multiple PEPs. Although many policy decisions can be made trivially by co-locating the PDP and the PEP, there are certain advantages that can be realized by the use of a policy server.

3.2.4.3 Use of Policy Protocols

When RSVP messages transit RSVP-aware network devices, they cause the configuration of traffic handling mechanisms in PEPs, including classifiers and queuing mechanisms, that provide intserv services. However, in many cases, RSVP cannot be used to configure these mechanisms. Instead, more traditional, *top-down* mechanisms must be used.

These protocols include Simple Network Management Protocol (SNMP), command line interface (CLI), Common Open Protocol Services (COPS) and others. SNMP has been in use for many years, primarily for the purpose of monitoring network device functionality from a central console. It can also be used to *set* or configure device functionality. CLI is a protocol used initially to configure and monitor Cisco network equipment. Due to its popularity, a number of other network vendors provide CLI-like configuration interfaces to their equipment. COPS is a protocol that has been developed in recent years in the context of QoS. It was initially targeted as an RSVP-related policy protocol but has recently been pressed into service as a general diffserv configuration protocol. All these protocols are considered top-down because, traditionally, a higher level management console uses them to *push* configuration information down to a set of network devices.

In the case of signaled QoS (as opposed to top-down QoS), detailed configuration information is generally carried to the PEP in the form of RSVP signaling messages. However, the PEP must outsource the decision whether or not to honor the configuration request to the PDP. COPS was initially developed to pass the relevant information contained in the RSVP message from the PEP to the PDP, and to pass a policy decision in response. Obviously, when PEP and PDP are co-located no such protocol is required.

A protocol is also required for communication between the PDP and the policy data-store. Since the data-store tends to take the form of a distributed directory, LDAP is commonly used for this purpose.

4 Tradeoffs in the QoS Enabled Network

In previous sections we reviewed a number of QoS mechanisms. In following sections, we'll see how these mechanisms can be combined to build a QoS-enabled network. In this section we'll discuss the requirements of the QoS enabled network and the pragmatic tradeoffs which must be considered in its design.

Earlier in this whitepaper we effectively stated that network QoS provides the ability to handle application traffic such that it meets the service needs of certain applications. We also stated that, if network resources were infinite, the service needs of all applications would be trivially met. It follows that QoS is interesting to us because it enables us to meet the service needs of certain applications when resources are finite. In other words:
A QoS enabled network: *should provide service guarantees appropriate for various application types while making efficient use of network resources.*

4.1 Varying Quality of Service Guarantees

Different qualities of service guarantees are appropriate for different applications. The *quality* of a guarantee refers to the level of commitment provided by the guarantee. This is not necessarily related either to the actual amount of resources committed, nor to the cost of the resources. For example, a guarantee that commits to carry 100 Kbps with a per-packet latency not to exceed 10 msec is a high quality guarantee. A guarantee that commits to carry 1 Mbps with the appearance of a lightly loaded network is a lesser quality guarantee. A guarantee that offers no commitment regarding latency bound or drop probability is a low quality guarantee.

The first two levels of guarantee described correspond to the guaranteed and controlled-load intserv services. The third corresponds to the standard best-effort service ubiquitously available today. There are other levels of guarantee that may be useful. For example, one could imagine varying degrees of better-than-best-effort (BBE) which offer to carry traffic with lower latency or at higher rates than it would be carried if it were best-effort, but make no specific quantifiable commitments. Often the terms 'quantitative QoS' and 'qualitative QoS' are used to refer to services such as guaranteed and controlled load on the one hand versus BBE on the other hand.

An appraisal of the quality of the guarantee is not a judgement regarding its value to the end-user, but rather a statement of its suitability to different applications. For example, a BBE level of guarantee may be entirely satisfactory to a web surfing application while a guaranteed service level of guarantee is required to handle interactive voice traffic. While the quality of the guaranteed service is higher, it would be excessive for a web surfing application. From a cost/performance perspective, the end user of a web surfing application would likely be more satisfied with the lower quality guarantee. Cost is a pragmatic consideration related to the efficiency with which network resources are used. If cost were not a concern, it would be desirable to support the highest quality guarantees possible.

4.1.1 Providing Low Quality of Service Guarantees

Low quality guarantees are relatively easy to provide in an efficient manner by using simple QoS mechanisms. For example, existing best-effort corporate networks generally provide a very low level of guarantee with very few QoS mechanisms. Users may be able to web-surf fairly painlessly (assuming that the targeted web servers are not a bottleneck). The extent of QoS mechanism present in these networks is that the network administrator keeps an eye on the network usage level and, from time to time, (as the number of users on the network grows), adds capacity to (re-provisions) the network. It may take one second for a typical web query to complete, or it may take five, depending on the time of day and the activity level of other users on the network. However, the service level perceived by the network users, remains relatively satisfactory.

If web surfing were deemed critical to the jobs of the corporate network users, it might make sense for the network administrator to use simple top-down QoS configuration mechanisms to improve the service perceived by web surfing users. For example, the network administrator might identify those devices in the

corporate network that tend to congest, and configure them with classifiers to recognize web surfing traffic and to direct it to high priority queues in the devices. This is essentially a top-down, diffserv approach. It would tend to improve the service level perceived by reducing the average time it takes for web queries to complete.

This is quite an efficient approach, as no resources have been added to the network or committed to web surfers. However, while it does provide a quality of service guarantee that is better than best-effort, it is still a relatively low quality of service guarantee. There are no bounds on the latency perceived by the users. Further, the latency might degrade significantly in the event that an unusually high number of users decided to web-surf simultaneously (thereby overwhelming the higher priority queues in the network devices). This condition would be especially severe if all simultaneous users resided on the same subnet and/or connected to web-servers on the same subnet. In this case, unusually high demands would be placed on a smaller set of network devices. Thus, the quality of the service guarantee would depend on the number of simultaneous web surfing users and their location in the network topology.

The network administrator might attempt to limit such degradations in quality of service by adding capacity to those network devices that tend to congest. However, much of the time, there would not be an unusually high number of users web surfing simultaneously and those that were would tend to be distributed across the network (rather than co-located on a single subnet). Thus, much of the time, the added capacity would be unused. As a result, network resources would be used inefficiently.

A simple analogy to non-network traffic engineering is helpful in illustrating the quandary faced by the network administrator. Consider the urban developer faced with the task of building a street system. The developer should probably design roads with the capacity to carry average expected traffic loads. Remote areas of the city will generally require smaller roads. Central, highly trafficked areas of the city will generally require larger roads. This approach is efficient. On occasion, a large number of drivers might flock to a remote area of the city for a specific event. As a result, the smaller road serving this part of the city will become congested. The developer could reduce the odds of such congestion by building large roads even to remote parts of the city. However, this would be inefficient since, most of the time, these roads would be relatively underutilized.

4.1.2 Providing High Quality of Service Guarantees

Providing high quality of service guarantees is more challenging than providing low quality of service guarantees. In the previous example, the network administrator has the option of provisioning the network for average expected load. Under extreme conditions, congestion might cause web surfing response times to increase, but the application would still be useable.

Consider instead, an IP telephony application. IP telephony users each require from the network a guarantee to carry 64 Kbps, with a maximum end-to-end latency no higher than 100 msec. A higher latency renders the service useless. In this example, the network administrator resorting to top-down QoS configuration mechanisms has no choice but to over-provision the network. (In the subsequent section, we will see how the use of signaling QoS configuration addresses this problem.) For example, assume that out of 1000 *potential* users of IP telephony, there are *on the average* 10 simultaneous users. Efficiency considerations would suggest that a device in the center of the network should be provisioned to accommodate 10 simultaneous users at a latency of 100 msec.

Assume that telephony sessions between 10 users are currently in progress (the network is at capacity). Let's see what happens when two additional users attempt to place an IP telephony call. The incremental traffic would overload the low latency service queue in the network device, thereby raising latencies above 100 msec and compromising service to all 12 IP telephony users. At this point, all resources allotted to IP telephony would be wasted since none of the 12 users would perceive satisfactory performance.

In this example, provisioning for average load dramatically compromises the quality of service guarantee that can be given to IP telephony users. The chance of compromise is directly proportional to the chance

that the network is required to carry *even one* IP telephony session beyond that number for which it is provisioned. Generally, to provide high quality guarantees in a top-down provisioned QoS network requires significant over-provisioning.

4.1.3 End-to-End Requirement

Although the QoS mechanisms to provide a particular guarantee may vary from point to point in the network, the guarantee must be valid *end-to-end*. The network provider offers guarantees because the network administrator can charge for guarantees. The network administrator can charge for guarantees because the network user is willing to pay for guarantees. The network user is willing to pay for guarantees only because the experience of the network user is improved as a result of the guarantee. The experience of the network user is improved only if the quality of the connection between the user's endpoints is improved. Hence the end-to-end requirement. Certain large providers may claim that they are able to charge their peer network providers for guarantees, without concern for the end customer. However, this is not a sustainable model. Ultimately, the provider's peer or the provider's peer's peer is collecting money from the end user to pay its provider.

4.2 Efficiency vs. Quality of Guarantees

There is no clear dividing line between the network provisioning requirements to support low quality guarantees and those to support high quality guarantees. The higher the quality of guarantees desired, the more it is necessary to over-provision the network for the same level of user satisfaction. Thus, the lower the efficiency with which network resources will be used. In providing a QoS-enabled network, there exists a continuum of provisioning options in which the quality of guarantees available is traded off against efficiency of network resource usage.

4.3 Signaling

In the previous examples we considered only top-down provisioning of the network. In the following discussion, we see that by using a signaling approach to QoS configuration, it is possible to shift the quality of guarantee versus efficiency tradeoff in the network administrator's favor.

Consider again the IP telephony example. Let's assume that users of the IP telephony application signal an RSVP request for resources to the network before actually obtaining the resources. The device in the center of the network is aware of the capacity in its low latency queue and is able to listen to and respond to RSVP signaled requests for resources. In this case, the network device installs classifiers in response to signaling requests from the first ten IP telephony users. These classifiers are used to identify traffic entitled to the low latency queue in the device. The device would reject the RSVP request from the eleventh and twelfth user. No classifiers would be installed for these users and their traffic would not impact the quality of guarantees already made to the first ten users.

In this example, the network is able to offer very high quality guarantees to some limited number of simultaneous users. It refuses guarantees beyond this number in order to preserve the quality of the guarantees that are offered to sessions already in progress. This is achieved without any over-provisioning. In this sense, the network in this example is optimal. However, it is also somewhat unrealistic. It assumes a single device in the center of the network through which all traffic passes. In reality, network topologies are far more complex. Providing optimal efficiency while maintaining high quality guarantees would require that every network device participate in signaling, that these devices be able to strictly enforce the allocation of resources to one conversation versus another, that applications be able to precisely quantify their resource requirements and so on. In general, this is not the case. And so, while the support of signaling in the network can shift the quality of guarantee versus efficiency tradeoff in the network administrator's favor, it cannot, in a real network, simultaneously offer high quality of guarantees and optimal efficiency.

4.3.1 The Costs and Benefits of Signaling

We have shown that signaling can improve the tradeoff between quality of guarantee and efficiency of network resource usage. However, this comes at a cost. Signaling itself requires network resources. Any form of signaling generates additional network traffic. RSVP signaling, due to its soft state, does so continually (albeit at low volumes). In addition, in order for the signaling to be useful, it is necessary for network devices to intercept signaling messages and to process them. This consumes processing resources in the network devices. When analyzing the benefits of signaling it is necessary to consider these effects.

There are ways to exploit the benefits of signaling while reducing its inherent impact on network resources. These include aggregation of signaling messages and reduction in the density of signaling nodes.

4.3.1.1 Aggregation of Signaling Messages

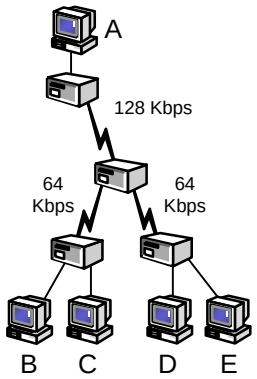
In the case of standard RSVP signaling, messages are generated for each conversation in progress. In those parts of the network through which there is frequently a large number of conversations, it is possible to aggregate signaling messages regarding aggregate resources. For example - in the case of a transit network interconnecting two corporate subnetworks, per-conversation RSVP requests between the subnetworks might be aggregated at the boundaries between the subnetworks and the transit network. The per-conversation signaling messages would still be carried end-to-end, but would not be processed within the transit network. Instead, aggregate signaling messages would be exchanged between edges of the transit network and would reserve resources in the transit network to support the number of simultaneous end-to-end conversations. The aggregate reservation would be adjusted from time to time in response to demand.

4.3.1.2 Signaling Density

In theory, optimal efficiency is attained when every device in the network participates in signaling and admission control. However, this is costly in terms of signaling processing overhead, signaling latency, and so forth. As an alternative, the network administrator may configure only certain key devices to participate in signaling and admission control. A relatively sparse configuration of signaling and admission control devices reduces the costs associated with signaling overhead but also compromises the benefits of signaling in terms of the quality of guarantees which can be offered or the efficiency with which network resources can be used. To see why this is the case, it is necessary to understand the awareness of traffic patterns that is implicit in RSVP signaling and is key to admission control.

4.3.1.3 Signaling and Awareness of Traffic Patterns

Consider the network illustrated in the following diagram:



For the example, assume the following:

- All routers participate in RSVP signaling.
- One QoS session requiring 64 Kbps is initiated between host A and host B.

- Another session requiring 64 Kbps is initiated between host A and host D.

In this case, one RSVP request for 64 Kbps would reach the three routers in the data path between host A and host B. Another RSVP request for 64 Kbps would reach the three routers between host A and host D. The routers would admit these resource requests because they would not over-commit any of the links[9]. If instead, hosts B and C each attempted to simultaneously initiate a 64 Kbps QoS session to host A, the router serving these hosts would prevent one or the other of these sessions from being established.

RSVP signaling enables an awareness of traffic patterns. Because resource requests arrive at each device that would be impacted by admission of the request, it is possible to refuse requests that would result in the over-commitment of resources. Two simultaneous requests for 64 Kbps could be admitted if one were along the right branch of the network and the other along the left branch of the network. However, if both were along the same branch of the network, one of the requests would not be admitted.

Now assume that the network administrator reduces the density of signaling-enabled network devices by disabling the processing of QoS signaling messages in the lower three routers (serving hosts B, C, D and E). Only the topmost router participates in signaling, becoming in effect, the admission control agent for itself as well as the remaining routers in the network. In this case, requests for resources up to 128 Kbps would be admitted regardless of the location of the participating hosts. Service guarantees would be low quality guarantees, as it would be possible for traffic from one host to compromise service for a session granted to the other.

The quality of guarantees could be maintained if the topmost router were configured to limit admission of resource requests to 64 Kbps. However, this would result in inefficient use of network resources as only one conversation could be supported at a time, when in fact two could be supported if their traffic were distributed appropriately. Alternatively, all 64 Kbps links in the network could be increased to 128 Kbps links to avoid over-commitment of resource requests, but the increased capacity would be used only in the event that hosts B and C (or D and E) required resources simultaneously. If this were not the case, such over-provisioning would also be inefficient.

We see that, in general, by reducing the density of signaling enabled devices, we reduce the value of signaling in terms of the tradeoff between quality of guarantees and efficiency of network resource usage. This is because the network administrator has imperfect knowledge of network traffic patterns. If the network administrator knew with certainty, in the above example, that hosts B and C (or hosts D and E) never required low latency resources simultaneously, they could be offered high quality guarantees without signaling and without incurring the inefficiencies of over-provisioning. In smaller networks, it is very difficult for the network administrator to predict traffic patterns. In larger networks, it tends to be easier to do so. Thus, reductions in the density of signaling aware devices tends to compromise efficiency less in large networks than in small networks.

4.3.1.4 Other Benefits of Signaling

There are other benefits of signaling which are unrelated to the tradeoff between quality of guarantees and efficiency of network resource usage. These include the end-to-end integration of QoS on disparate network media as well as the provision of classification and policy information to network devices. These benefits will be discussed later in the paper.

4.4 Sharing Network Resources - Multiple Resource Pools

The QoS-enabled network must provide both low and high quality guarantees. High quality guarantees are typically made practical via the use of signaling, admission control, and strict policing along specific routes.

[9] In practice, routers would not be configured to allow all resources available to be reserved for a particular conversation. However, for simplicity's sake, we assume in this case that the entire link resources can be reserved.

In order to maintain the quality of these guarantees, it is important to prevent traffic that makes use of lower quality guarantees from stealing resources committed to higher quality guarantees. However, traffic using lower quality guarantees is not policed as strictly as traffic using higher quality guarantees. Specifically, it tends not to be policed based on its route through the network. As a result, it may appear at various locations in the network in volumes above those anticipated. To prevent such unexpected traffic from compromising higher quality guarantees, it is necessary to assign this traffic lower priority in its use of network resources at specific devices. This does not mean that applications requiring lower quality guarantees are deemed to be lower priority by the network administrator. In fact, typically, the percentage of available resources at any node that is allocated to high quality guarantees is only a very small fraction of the total resources available, with the majority remaining available for lower quality guarantees. It does mean, however, that under congestion conditions, traffic requiring lower quality guarantees will be deferred in favor of traffic requiring higher quality guarantees up to some limit.

In effect, there are several *resource pools* in the diffserv network. These are used by traffic requiring different quality guarantees. Traffic is separated by:

- Aggregating it according to the service level to which it is entitled.
- Policing traffic requiring higher quality guarantees such that it does not starve traffic using lower quality guarantees.

We can identify four general resource pools by the traffic for which they are used:

Quantifiable traffic requiring high quality guarantees - This type of traffic requires a specifically quantifiable amount of resources. These resources are typically allocated as a result of RSVP signaling, which quantifies the amount of resources required by the traffic flow. The highest priority queues are reserved for this traffic. This traffic is subjected to strict admission control and route-dependent policing. Examples of this type of traffic include IP telephony traffic and other interactive multimedia traffic.

Non-quantifiable persistent traffic requiring high quality guarantees - This type of traffic requires resources that cannot be specifically quantified. However, it tends to be *persistent* in the sense that it consumes resources along a known route for some reasonable duration. Resources are allocated to this class of traffic as a result of RSVP signaling that does not specifically quantify the resources required by the traffic flow. This signaling informs the network of the application sourcing the traffic as well as the route taken through the network. The information facilitates prediction of traffic patterns, enabling reasonable quality guarantees. However, since resource requirements are not strictly specified, resource consumption cannot be strictly policed and the traffic is forced to use queues that are of lower priority than those available for quantifiable traffic. Examples of this type of traffic include traffic of client-server, session oriented, mission critical applications such as SAP and PeopleSoft.

Non-quantifiable, non-persistent traffic requiring low or medium quality guarantees - This type of traffic is relatively unpredictable. Its resource requirements cannot be quantified, and its route through the network is fleeting and subject to frequent changes. The overhead of signaling cannot be justified, as it would provide little information to assist the network administrator in managing the resources allocated to this traffic. Because the impact of this traffic is so unpredictable, it is forced to use queues that are of lower priority than those used by signaled traffic. As a result, only low quality guarantees can be offered to such traffic. An example of this type of traffic is web surfing.

Best-effort traffic - this is all the remaining traffic, which is not quantifiable, not persistent, and does not need any quality of service guarantees. The network administrator must assure that there are resources available in the network for such traffic but need provide no specific quality of service for it. This traffic uses default FIFO queues and receives those resources that are 'left-over' after the requirements of higher priority traffic have been satisfied.

The QoS network administrator is faced with the task of provisioning admission control limits for each of these classes of traffic. By doing so, the administrator is effectively dividing the network resources into the resource pools mentioned at the start of this section.

4.5 Quality/Efficiency Product and Overhead

We can summarize this section by recognizing the tradeoffs inherent in designing a QoS enabled network. Recall that the goal of QoS enabling a network is to provide the various qualities of guarantee required by the customer's applications, while maintaining efficient use of network resources. We can measure the quality of a QoS network by the product of the quality of guarantees it offers and the efficiency of resource usage. We will refer to this metric as the *quality/efficiency product* of the network.

A third factor to consider in the design of a QoS network, is the *overhead*. Overhead refers to the processing and storage overhead in network elements that is directly attributable to the QoS mechanisms themselves (whether for traffic handling or for signaling processing)[10]. All QoS mechanisms impose an overhead on the network, increasing its cost. The cost of any QoS mechanism in terms of its overhead must be weighed against the potential improvement in the quality/efficiency product. In general, the greater the overhead that the network administrator is willing to tolerate, the higher the quality/efficiency product which can be attained.

Note that this tradeoff, between overhead and quality/efficiency product is a local decision, which may vary from one part of a network to another. For example, it may be quite acceptable to over-provision certain LAN segments, accepting that the only way to obtain quality guarantees through these parts of the network is to use them inefficiently (low quality/efficiency product). This approach requires no QoS overhead in these LAN segments. On the other hand, it may be prohibitively expensive to over-provision certain WAN segments. QoS mechanisms would be employed in these parts of the network with the goal of attaining a higher quality/efficiency product. Thus, any debate as to the value of one or another QoS mechanism, should be considered in these terms.

The following table illustrates variations of the general QoS mechanisms we have discussed so far and their impact in terms of overhead vs. quality/efficiency product:

Mechanism	Overhead	Quality/Efficiency
FIFO traffic handling	None	Low
Aggregate traffic handling	Low	Medium
Per-flow traffic handling	High	High
Top-down provisioning	Low	Low
Aggregate signaling	Medium	Medium
Per-flow signaling	High	High
Sparse signaling	Medium	Medium
Dense signaling	High	High

4.5.1 Simultaneous Support for Multiple Traffic Types

Note that in general, a single part of the network may be designed with a variety of tradeoff points to accommodate differing traffic types. For example, while the WAN part of the network may use per-flow signaling and traffic handling to provide a high quality/efficiency product for IP telephony traffic, it may handle traffic from less demanding applications on a FIFO basis with no signaling. Thus, the network administrator divides the WAN subnet into multiple resource pools (as described earlier in this section) appropriate for the types of traffic it will carry.

[10] At first glance it might appear that *overhead* is captured in the *efficiency* metric. However, overhead is defined to be the cost of resources dedicated to the QoS mechanisms themselves, while efficiency relates to the raw network resources that are bandwidth and buffer space.

4.5.2 Management Burden

Note that we use the term overhead in reference to the work required from the network to provide QoS. Such overhead is not to be confused with what is commonly called *management overhead*. We will refer to the latter as *management burden* here, in order to avoid confusion with *overhead*. These are different concepts. For example, extensive use of signaling may significantly reduce management burden (as compared with top-down provisioning). However, it does result in higher overhead. A classic example of incurring additional overhead in the interest of reducing management burden is the use of address resolution protocols (such as ARP) versus statically configured (MAC address) tables.

5 The Sample QoS Network

In this section we'll present a sample network which we will use as a basis for subsequent discussion. The sample network is intended to reflect a realistic network incorporating multiple subnets of varying types. It is illustrated below:

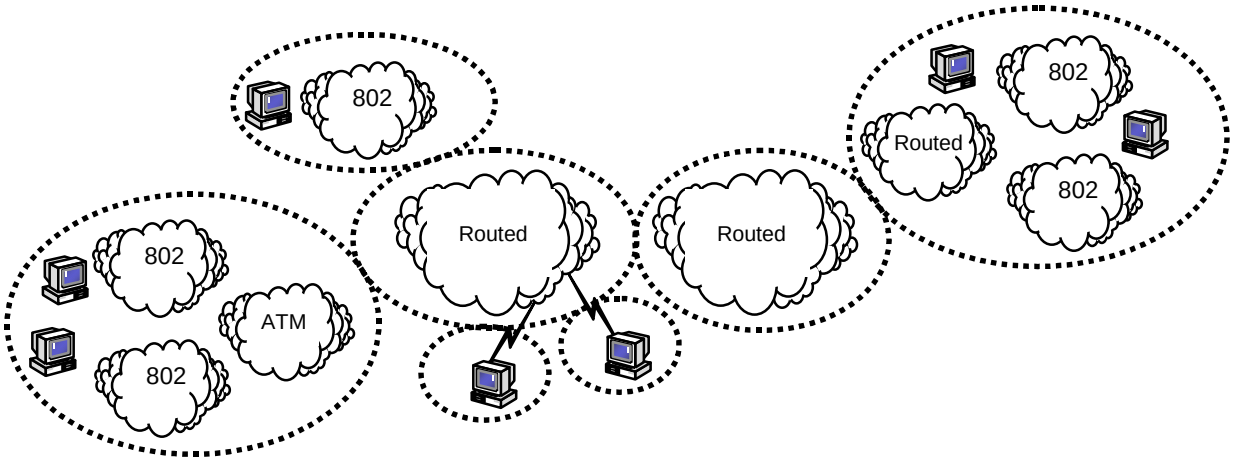


Figure 1 - Sample Network

The two large routed networks in the center of the diagram represent large network providers. The peripheral networks represent customer networks. There are three corporate or campus customer networks illustrated and two individual home customer networks. The network providers can be considered *transit* networks, as they contain no hosts[11] or end-stations. The various customer networks contain hosts. The bold dashed ovals separate the larger network into sub-networks. For the sake of simplicity, we assume that these also correspond to administrative domains (ADs)[12].

All the corporate or campus networks are illustrated as a combination of smaller routed networks, ATM networks and 802 LANs. Private customer networks are illustrated as single hosts connected via dial-up lines. Interconnections between networks are not clearly identified (other than the dial-in connections to the private customer networks). Interconnections could range from SONET rings to high speed leased lines, to xDSL connections, cable connections, low-speed modems, and so on. Interconnections may be represented as networks in their own right. Generally, some pair of interconnection devices is implied.

5.1 Assumptions Regarding the Sample Network

We assume that the network:

[11] In general, large provider networks may offer services, in which case they would also contain hosts.
[12] All devices within a single AD are managed by a single administrator with consistent economic objectives. The notion of ADs is generally recursive, in the sense that there may be multiple ADs within a larger AD, just as there may be local governments subject to a federal government.

1. Includes an arbitrary number of concatenated subnetworks of arbitrary media.
2. Is required to provide a combination of high quality and low quality guarantees on an end-to-end basis.
3. Must meet certain objectives in terms of efficiency of resource usage.
4. Must meet certain objectives in terms of overhead of QoS mechanisms.
5. Must be manageable.

5.2 Subnet Local QoS Mechanisms

Each subnetwork provides local QoS mechanisms. These include:

- Various traffic handling mechanisms in devices, as appropriate for the scale and media of the subnet.
- Policy servers (PDPs) and policy data-stores which provide QoS top-down provisioning capabilities, as well as interaction with end-to-end QoS signaling (as described previously).
- Agents in various network devices that are able to participate in end-to-end QoS related signaling.

5.3 Global QoS Mechanisms

Other QoS mechanisms are global in the sense that they span multiple sub-networks. These include:

- Per-conversation, end-to-end RSVP signaling, which is generated by certain hosts for certain application traffic.
- Inter-domain or intra-domain signaling in the form of aggregated RSVP, MultiProtocol Label Switching (MPLS) signaling, bandwidth broker interactions, and so forth. [13]
- High level cross-network provisioning and configuration applications.

5.3.1 The Role of RSVP in Providing High Quality End-to-End QoS

As discussed previously, guarantees must be valid *end-to-end*, across multiple subnets. Lower quality guarantees can be provided without requiring tight coupling between the QoS mechanisms in different subnets. However, high quality guarantees require tight coupling between these mechanisms.

As an example, it is possible to independently configure devices in each subnet (in a top-down manner) to prioritize some set of traffic (as identified by IP port) above best-effort traffic (BBE service). This will indeed improve the quality of service perceived by the prioritized application, in all parts of the network. However, this is a low quality guarantee, as it makes no specific commitments regarding available bandwidth or latency.

On the other hand, consider the quality of guarantee required to support a videoconference. A videoconferencing application requires that all subnets between the videoconferencing peers be able to provide a significant amount of bandwidth at a low latency. To do so efficiently requires that all devices along the data path commit the required amount of low latency bandwidth, for the duration of the videoconference. As we have seen, high quality guarantees such as these generally require signaling across network devices in order to make efficient use of network resources. In our sample network, multiple subnets, based on multiple media (and varying traffic handling mechanisms) must be coordinated via this signaling. RSVP with intserv is particularly suitable for this purpose because it expresses QoS requirements in high-level, abstract terms. Agents in each subnet are able to translate the media independent, abstract requests into parameters that are meaningful to the specific subnet media. The ISSLL (*Integrated Services Over Specific Link Layers*) working group of the IETF has focused on the definitions of mappings from integrated services (intserv) to numerous media, including 802 networks, ATM, slow links (e.g. traditional modems) and, recently, diffserv.

In our model, hosts generate RSVP signaling when it is necessary to obtain high quality guarantees. The network listens to this signaling at strategic points in the network. We will refer to devices that participate in

[13] The terms *MPLS* and *Bandwidth Broker* are defined later in this document.

RSVP signaling as *RSVP agents* or alternatively as *signaling* or *admission control agents.* As we have shown, appointing such agents at varying densities can provide varying quality/efficiency products. At a minimum we assume one or more admission control agents in each subnet. Each agent uses the mappings defined in ISSLL to translate high level end-to-end RSVP requests into parameters that are meaningful to the media for which the agent is responsible. The admission control agent then determines, based on resource availability and/or policy decisions, (with the cooperation of PDPs) whether an RSVP request is admissible or not. Any admission control agent along the route from sender to receiver may veto an RSVP request for resources. Requests that are not vetoed by any device are considered admitted and result in the return of an RSVP RESV message to the requesting transmitting host.

5.3.1.1 Service Mappings

An important component of the end-to-end service model described above is the mapping from intserv services to the corresponding traffic handling mechanisms in each of the subnets on the end-to-end path. As mentioned previously, the definition of such mappings is the responsibility of the ISSLL working group of the IETF. A mapping includes definition of the underlying media-specific service suitable to provide the intserv service. It also includes admission control guidelines. These are used to determine the marginal impact that will result from admission of additional traffic to an underlying traffic handling mechanism. Based on this impact, additional traffic may be admitted or may be refused admission.

5.4 Host QoS Mechanisms

Hosts play an important role in end-to-end QoS. Host QoS mechanisms include:

- Generation of RSVP signaling for conversations requiring high quality guarantees, including identification of both the user and the application requesting resources
- DSCP marking
- 802.1p marking
- Traffic scheduling

Hosts generate RSVP signaling for conversations requiring high quality guarantees. These include conversations generating both quantifiable and non-quantifiable traffic, so long as they are persistent. Hosts then proceed to mark and schedule traffic based on the results of the signaling requests. If a signaling request for resources at a specific intserv service level is admitted, the host will mark traffic on the corresponding conversation with the appropriate DSCP and 802.1p marks based on the ISSLL mapping from intserv to diffserv and 802, respectively. (Note that the network may override default mappings). If a signaling request specifies quantifiable parameters, the host schedules traffic in accordance with the requested parameters.

Although the role of the host is most pronounced in the context of signaled QoS, it may also participate in supporting top-down provisioned QoS. It does so by enabling policy agents to provision classification, scheduling and marking information in transmitting hosts, to control traffic that is non-persistent (for which signaling messages are not generated).

6 Unifying the Subnets of the Sample Network

In this section, we will discuss QoS mechanisms in each of the subnetworks comprising the sample network and how they are integrated with the global QoS mechanisms of the end-to-end network.

6.1 Focus on Signaled QoS

As discussed previously, providing end-to-end guarantees requires coordination of resource allocation across all subnets on the end-to-end path. Top-down provisioning is adequate for providing low quality guarantees. To the extent that top-down provisioning management systems are able to integrate information regarding network topology, current resource usage in various parts of the network and fine-grain classification information, the quality of the guarantees provided can be improved. However, for any

persistent conversation, host-based signaling provides information to the network. This information can be used to improve the quality of guarantees provided even further. For this reason (and due to the general end-to-end focus of this whitepaper), the following discussion will tend to focus on signaled QoS mechanisms. These mechanisms can be superimposed on the background of a top-down provisioned approach, so long as the network administrator enforces the separation of resource pools as described earlier.

6.2 Large Routed Networks - Diffserv

We'll start with the large routed networks shown at the center of the sample network. These represent large provider networks such as those of Internet Service Providers (ISPs). These networks are generally constructed with many large routers that are interconnected by high speed, wide area links. These routers typically carry traffic from thousands (if not millions) of simultaneous conversations. The overhead of providing per-conversation traffic handling or of listening to per-conversation signaling in these networks is prohibitive. However, from previous discussion we also know that using signaling and per-conversation QoS mechanisms can provide high quality guarantees most efficiently. Given that it is necessary to support high quality as well as low quality guarantees in this network, we are faced with a choice between incurring signaling and per-conversation overhead or accepting that the network will be operated inefficiently.

Due to the large amount of traffic aggregated in these networks, traffic patterns are relatively predictable and variance in load over time at any device is relatively small. In this case, minor over-provisioning (slight inefficiency) can yield a major improvement in the quality of guarantees that can be offered. It follows that, in general, in large subnetworks, it is preferable to incur minor inefficiencies rather than to incur the overhead of dense signaling and per-conversation QoS mechanisms. Diffserv is ideally suited to this tradeoff as it does not inherently rely on signaling and it handles traffic in aggregate. However, in practical terms, in order to support high quality guarantees through a diffserv network, some minimal signaling overhead must be incurred. The strategy we describe for supporting QoS in large networks is, therefore, based on diffserv style aggregate traffic handling, coupled with sparse processing of signaling messages when high quality guarantees are required.

6.2.1 Diffserv Aggregate Traffic Handling

Diffserv is implemented by supporting aggregate traffic handling mechanisms known as per-hop-behaviors (PHBs) in network devices. Packets entering the diffserv network are marked with diffserv codepoints (DSCPs) which invoke particular PHBs in the network devices. Currently defined PHBs include *expedited forwarding* (EF), and *assured forwarding* (AF). The EF PHB offers low latency, and is intended to provide *virtual leased line* (VLL) service. VLL service offers high quality guarantees and emulates conventional leased line services. The AF PHB offers a range of service qualities, generally lower than EF supported services but higher than traditional best-effort services. The AF PHB uses a group of twelve DSCPs specifying one of four relative priorities and one of three drop-precedence levels within each priority.

6.2.2 Service Level Agreements

In diffserv terms, the quality guarantees offered by the diffserv network are reflected at the edge of the network in the form of *service level agreements* (SLA). SLAs specify the parameters of a service that can be invoked by particular DSCPs and the amount or rate of traffic that the provider agrees to carry at the specified service level. Traffic submitted in excess of the negotiated rate is subjected to some alternative treatment, also specified in the SLA. SLAs may offer one or more service levels.

6.2.3 Functionality at the Edge of the Diffserv Network

Minimal diffserv functionality requires that the customer mark traffic submitted to the provider's network with the appropriate DSCP and that the provider *polices* submitted traffic on a per-customer, per-DSCP basis. The provider must police to verify conformance to the SLA, thereby limiting the resources consumed by the customer's traffic in the provider's network. Excess traffic is typically delayed, discarded, or re-marked to a less valuable DSCP. In order to avoid excess traffic from being arbitrarily penalized in the diffserv network, the customer may *shape* submitted traffic to assure that it conforms to the SLA.

In certain cases, the provider may offer *value-added* services such as marking or shaping traffic on behalf of the customer. Traffic may be marked or shaped on an aggregate level or at finer granularities in order to provide a level of traffic isolation that suits the customer's requirements. These services are referred to as *provider marking* or *provider shaping*.

Many interesting issues arise regarding the implementation of policing, marking and shaping functionality. These are beyond the scope of this document.

6.2.4 Provisioning the Diffserv Network

Provisioning of the diffserv network includes (in order of increasingly dynamic tasks):

- Selection of network equipment
- Selection of interfaces and interface capacity
- Topology determination
- Selection of enabled PHBs
- Determination of DSCP to PHB mappings
- Determination of queuing parameters associated with each PHB

These provisioning tasks determine the aggregate capacity of the provider's network, across all customers. As a result of such provisioning, the network provider effectively divides network resources into the various *resource pools* (described earlier) serving different qualities of guarantees.

6.2.5 Configuration of the Diffserv Network

We use the term configuration to refer to more dynamic tasks that affect per-customer resource allocation. This configuration includes (in order of increasingly dynamic tasks):

- Configuring per customer, per-service level policing parameters at the network ingress.
- Configuring value-added services such as provider marking or provider shaping at the network ingress.

The first of these tasks is quite different from the second. In the first, the provider configures the minimal information necessary to protect the provider's resources per the terms of the SLA. This includes classification criteria sufficient to recognize the originating customer and DSCP of each submitted packet and the corresponding per-customer, per-DSCP aggregate resource limits. The second task pertains to the configuration of information that determines which *subset* of the customer's traffic gains access to the aggregate resources available to the customer at each service level. The provider has no direct interest in how aggregate resources are divvied up among customer flows (so long as aggregate resource consumption is not being exceeded). This is actually a matter of internal customer policy. Any enforcement of internal customer policy should, from the provider’s perspective, be considered a value-added service.

Note that the first configuration task is relatively static as it changes only with the SLA (on the order of once per-month, per-customer). The second may be far more dynamic.

6.2.5.1 Configuration of Value-Added Services

Customers purchase aggregate capacities from providers at different service levels. It is in the customer's interest to assure that these resources are being used in an effective manner. When the customer relies on a provider's value-added services to mark and possibly shape customer traffic flows, the customer is also relying on the provider to determine the allocation of negotiated resources among individual customer traffic flows. In this case, it is important that the customer is able to effectively communicate to the provider the appropriate value-added configuration information.

Such information tends to be more dynamic and more voluminous than the simpler per-customer, per-service level configuration information (summarized in the basic SLA). As a result, the typical mechanisms by which SLA configuration information is communicated (e.g. monthly phone calls between a representative of the customer and a representative of the provider) tends to be unsuitable for communication of value-added configuration information. The difficulties in communicating value-added configuration information to the provider suggest that it is preferable for the customer to mark and shape traffic directly, eliminating the need for the provider to configure value-added parameters.

6.2.6 Using RSVP for Admission to the Diffserv Network

The customer should mark and shape traffic such that the volume of traffic marked for any particular service level is consistent with the resources available per the SLA and the customer's expectation regarding quality of guarantee. For example, consider the IP telephony example described earlier. If the SLA provides sufficient capacity to carry 10 IP telephony calls, the customer should avoid marking traffic from more than 10 simultaneous telephony sessions for the low latency service level[14]. In addition, the customer should assure that high value resources are used subject to some policy that defines the relative importance of different users and/or applications. RSVP signaling between hosts in the customer's network and admission control agents at the edges of the provider's network can be used to achieve both these goals.

Let's look at how admission control can be applied at an ingress point to a provider's network. Either the provider's ingress router or the customer's egress router (or both) can be configured to act as the admission control agent. The router acting as admission control agent should be configured to listen to per-conversation RSVP signaling. (Routers within the diffserv network are not required to listen to RSVP signaling. Instead, they pass RSVP signaling messages transparently.) In addition, it should be configured with the per-service level capacities available to the customer, per the SLA. It is also necessary for the router to understand the mapping from the intserv service level requested in RSVP requests to the corresponding diffserv service level (as described in section 5.3.1.1). Now, when an RSVP request is issued for data that will traverse the provider's network, it will arrive at the router serving as the admission control agent. The router has sufficient information to inspect the resources requested and to map the requested service level to the corresponding service level in the SLA. If the resources requested are available per the SLA, then the router admits the reservation request by allowing the RSVP request to pass unhindered. If resources are not available, the router rejects the request by blocking the RSVP request and returning an error.

In this mode, the router that is the admission control agent listens to per-conversation RSVP requests for the sake of tracking the customer's resource usage against the SLA. The router does not necessarily apply any per-conversation traffic handling. In the case that the admission control agent is the diffserv provider's ingress router, it uses diffserv aggregate traffic handling. Further, the router does not enforce any per-conversation admission control. Instead, it is the responsibility of the customer to make use of the admission control information provided by the edge device and to apply the appropriate marking and policing internally. Typically, well-behaved transmitters will respond by marking packets sent on admitted flows, with the DSCP that maps to the service level requested. Upstream senders should also refrain from marking traffic corresponding to rejected conversations. Alternatively, the sender may:

- Mark for a lesser DSCP.
- Refrain from sending traffic on the conversation altogether.
- Reduce its rate to a rate deemed admissible by the edge device.

Note that admission control agents may return a *DCLASS* object upstream in response to RSVP signaling requests. This object informs upstream senders of the appropriate DSCP to be marked in packets transmitted on the corresponding flow (thereby overriding the default mapping). In a subsequent section we will discuss in further detail how end systems and/or upstream devices mark DSCPs based on the results of RSVP signaling.

[14] When lower quality guarantees are expected, then the constraints can be relaxed accordingly.

RSVP signaling can also be used to enforce customer policies that determine which users and/or applications are entitled to use resources in the provider's network. This can be accomplished by configuring the customer's egress router to listen to RSVP signaling and to forward the policy objects contained in these messages (which identify the sending user and application) to a policy decision point.

Later in this whitepaper, we will discuss how Microsoft components can be used to provide the admission control functionality described in this section.

6.2.7 Dynamic SLAs and RSVP Signaling

In the previous section, we described the use of RSVP signaling to provide admission control to a diffserv network that provides static SLAs. In the near term, diffserv network providers are expected to be able to provide only static SLAs. This is because the existing QoS provisioning tools themselves are top-down and relatively static.

In the future, we can expect to see increasing demand for dynamic SLAs. Dynamic SLAs are preferable as they enable the provider to respond to changing resource demands from customers, thereby improving the quality/efficiency product of the diffserv network. This is particularly important when high quality guarantees are to be offered. However, dynamic SLAs require that the provider be able to re-provision the network core dynamically. Such re-provisioning is more complex than static provisioning. It also carries associated overhead and potential security problems. Nonetheless, these are not insurmountable problems and the potential reward in terms of improved quality/efficiency product is significant.

There are a number of mechanisms by which dynamic SLAs may be provided. Each of these requires a relatively dynamic QoS signaling protocol between the customer network and the provider network[15]. The protocol must provide a means by which the customer can request changes in the SLA and must result in any necessary re-provisioning of the provider's network (or refusal of the request). An obvious choice for this protocol is RSVP.

Recall that hosts will typically generate per-conversation RSVP signaling when high quality guarantees are required. We've already seen how this signaling can be used to provide admission control against static SLAs. We can leverage RSVP signaling further to assist in actual re-provisioning of the diffserv network itself. We discuss methods for doing so in the following paragraphs. These methods enable providers to optimize their networks for specific tradeoffs between the quality/efficiency product of the networks and the overhead they are willing to incur.

6.2.7.1 Triggering Re-Provisioning Based on Per-Conversation Signaling

As in the case of static SLAs, the network administrator configures the ingress router at the edge of the diffserv network to listen to per-conversation RSVP signaling and configures the devices in the core of the network to ignore the per-conversation messages flowing through them. The ingress router tracks the cumulative resources requested from customers at each intserv service level. As these reach high or low water marks, the ingress router triggers re-provisioning in the diffserv core, as appropriate.

6.2.7.2 Re-Provisioning the Core

Dynamic internal re-provisioning may be effected by various mechanisms. One such mechanism is via use of a *bandwidth broker*. The bandwidth broker is a hypothetical device, which has knowledge of the provider's network topology and current resource usage and is able to effect re-provisioning of the network

[15] In a sense, even static SLAs make use of a signaling protocol between customer and provider. In this case, the protocol consists of periodic change order requests (typically in the form of a phone call) from customer to provider to modify parameters of the SLA. The management burden associated with these requests may be significant, especially if services such as provider marking are involved. These requests may be followed by a lengthy period of negotiation and internal re-provisioning before the modified SLA terms are actually available to the customer.

to accommodate changes in resource requirements (or to refuse such changes). A more practical re-provisioning mechanism uses RSVP signaling internal to the diffserv network. The network administrator may configure strategic devices within the diffserv network to process either per-conversation or aggregate RSVP signaling. These devices in effect comprise a distributed bandwidth broker.

Note that, regardless of the use of per-flow or aggregate RSVP signaling for admission control and re-provisioning of the diffserv network, the actual traffic handling in a diffserv network is always aggregate, by definition.

6.2.7.3 Processing RSVP Signaling Messages in the Core

In processing RSVP signaling messages in the core, the network administrator is again faced with a variety of options. The lowest overhead option is to use edge devices that generate aggregate RSVP messages to re-provision major paths in the diffserv network, in response to changing demands from the periphery (signaled in the form of per-conversation or aggregate RSVP signaling messages). Devices at strategic locations within the diffserv network would process these messages. The network administrator can improve the quality/efficiency product of the diffserv network by enabling these devices more densely, or alternatively, can reduce the QoS overhead in the diffserv network by enabling these devices more sparsely.

If the network administrator is willing to incur the associated overhead, the administrator may chose to simply process per-conversation RSVP signaling in the core of the network (as opposed to aggregating them into aggregate signaling at the edges). Again, the administrator is faced with the choice of how densely or sparsely to enable these devices to select the appropriate tradeoff in quality/efficiency product versus overhead.

6.2.8 Provisioning for High Quality Guarantees

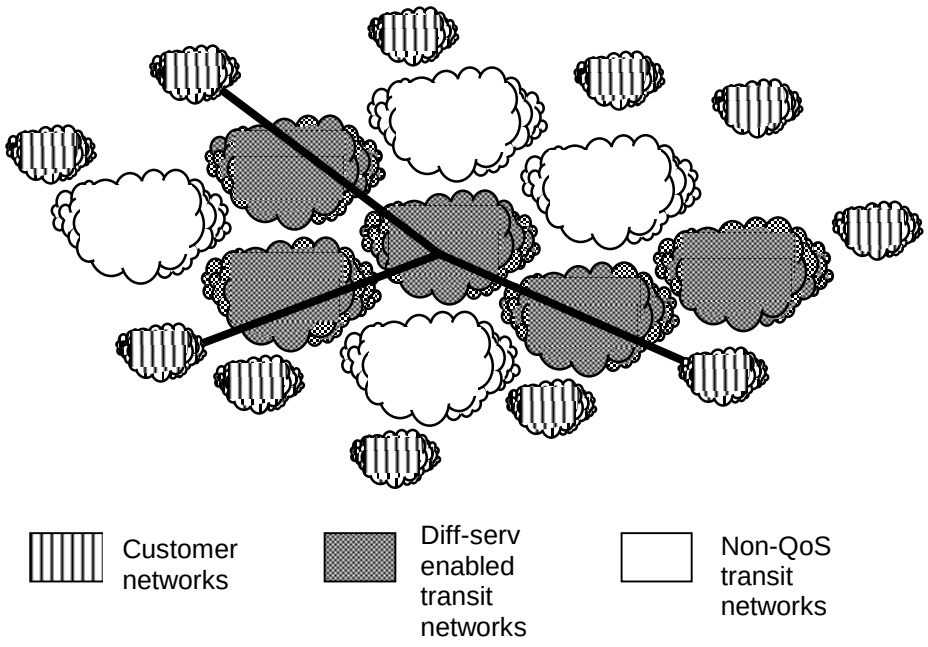
As we have shown, to provide high quality guarantees in an efficient manner requires good knowledge of traffic patterns in a network and an awareness of the volume of traffic that will be arriving at each network device for each service level. Since diffserv networks tend to be large, and variance in traffic patterns can be relatively low, it is feasible to offer some medium-quality guarantees while incurring only low losses in efficiency (section 6.2). However, in order to offer high quality guarantees, it is necessary to strictly control the amount of traffic, arriving at various locations in the network, claiming high quality treatment.

One mechanism for doing this along specific routes in the network, is to statically provision the capacities of high priority queues in various devices to accommodate high quality guarantees for a limited amount of traffic. In order to prevent rogue high priority marked traffic from claiming excessive resources along these routes (or other routes), it is necessary to strictly police the volume of traffic marked for high priority queues, throughout the network. Using this approach, it is possible to offer high quality guarantees at the network edges, for a limited volume of traffic, traversing a known route through the network. These guarantees are typically reflected in an SLA by specifying the egress point(s) of the traffic that can be accommodated at high service levels. (The customer should also expect to be policed based on these egress points.) This approach assumes that routes in the diffserv network can be reasonably well determined based on the traffic's ingress and egress points.

The mechanism discussed in the previous paragraph is consistent with the provisioning of static SLAs. A more dynamic mechanism for offering high quality guarantees is to respond to a customer's signaling requesting high quality guarantees. In this approach, the total capacity available in various devices for high quality guarantees is still statically provisioned, but is available to be shared among all customers in response to changing demand. By listening to (and responding to) per-conversation RSVP requests from customers (at least at strategic branch points), the provider can offer topology-aware admission control and high quality guarantees without predetermining the routes available to specific customers.

6.2.9 Emerging Diffserv Networks

For the near future however, we are unlikely to see extensive participation in per-conversation signaling by devices in diffserv networks. As a result, we are likely to see diffserv services offered as illustrated below:



In this diagram, we see a number of end customer networks, interconnected by transit networks. The customer networks can all communicate with each other using the basic best-effort service which exists today. Those that are interconnected by diffserv-enabled transit networks benefit from the low and medium quality QoS guarantees offered by these networks. Overlaid on top of the QoS enabled transit networks, we also see several provisioned QoS 'trunks' that offer high quality guarantees between a statically provisioned, limited set of endpoints (indicated by the heavy line). These form a sort of QoS VPN (virtual private network). Low and medium quality QoS guarantees will dominate the transit networks, with high quality QoS guarantees offered on specific routes, on a limited basis.

6.3 Switched Local Area Networks - 802

In this section, we'll discuss switched 802 networks. These are representative of many corporate or campus networks in which some number of hosts, ranging from the members of a small workgroup to an entire building or campus, are served by a number of interconnected switches. In larger campuses, switches may be grouped into subnetworks that are interconnected by layer 3 routers. We will focus initially on QoS mechanisms within the scope of a single switched subnetwork. Later, we will discuss QoS issues related to the interconnection of these subnetworks.

The discussions regarding the application of diffserv in large routed networks can be readily applied to many instances of switched networks. We observed that in large routed networks, small inefficiencies could result in significant quality gains due to the low variance of traffic patterns in the network. In switched networks, we can also accept some degree of inefficiency since local area switched resources tend to be quite inexpensive. It may also be true that switched networks support a large number of simultaneous users and that therefore the variance in traffic patterns is small. However, while this may be true near the core of

certain very large switched networks, it is not true near the edges of these networks, where some relatively small number of hosts are attached to each switch. Nonetheless, for existing applications, the bandwidth available near the edges of switched networks tends to be significantly higher than the bandwidth demanded by the hosts, rendering efficiency of resource usage unimportant.

Given that efficiency is of secondary concern in these switched networks, we find that these networks can provide relatively high quality guarantees using relatively low-overhead QoS mechanisms. In particular, we find that aggregate traffic handling mechanisms tend to provide reasonable QoS on switched networks. To the extent that we wish to extract higher quality/efficiency products from these networks, we may combine the aggregate traffic handling mechanisms with some degree of signaling processing.

In its use of QoS mechanisms, the switched network is analogous to the large routed network. Whereas the large routed network uses diffserv as an aggregate form of traffic handling, the switched network uses 802.1p as its aggregate form of traffic handling. While the large routed network appoints some number of routers near its edge as a minimal set of admission control agents, the switched network typically uses some number of SBM-capable switches as its admission control agents. Since the 802 network is analogous to the diffserv network, many of the considerations and issues discussed in the context of the diffserv network apply to the 802 network. In the following sections we revisit some of those considerations and issues and note differences between the two network types.

6.3.1 8021.p Aggregate Traffic Handling

Modern LAN switches provide multiple forwarding queues on each interface. These effectively provide different per-hop behaviors[16]. A particular forwarding queue is selected in each device by the *802.1p tag* included in the MAC header of packets submitted to the switch. The 802.1p tag carries one of eight priority values, corresponding to one of eight possible service levels in the network. The scope of these tags is the 802 subnet in which they are generated. 802.1p tags are not carried across layer 3 devices such as routers, but instead are dropped at the edge of the 802 network. As such, they are not carried across the routed networks illustrated at the center of the sample network illustrated previously.

6.3.2 Marking 802.1p Tags

As is the case with DSCPs, 802.1p tags can be generated either by the host transmitting a packet or by routers or switches in the network through which packets are carried. In either case, the device generating the tag may select a tag based on top-down provisioned criteria or, alternatively, may do so based on participation in RSVP signaling (or both - see section 6.5.1.3 for related discussion). In the top-down provisioning model, some device near the edge of the 802 cloud (host, switch or router) would be configured with classification criteria (by which packets would be identified as belonging to a certain flow) and the corresponding tag. This mechanism inherits the common problems associated with top-down provisioning, namely, that the quality/efficiency product of the network is limited. In the alternate model, hosts generate RSVP signaling describing the traffic they will be sending and its requirements from the network. Hosts or network devices then use the results of this signaling to determine how to tag packets on particular flows. This mechanism supports a greater quality/efficiency product.

Certain applications will not generate signaling. As a result, it is likely that some combination of top-down provisioned and signaling-based mechanisms will be used to effect packet marking. As has been discussed previously, this requires the network administrator to consider the 802 network resources to be divided into pools. The set of tags allowed by top-down provisioning should not claim resources from the same pool as those tags that are allowed as a result of signaling.

[16] Per-hop behaviours is a term borrowed from diffserv and should be used carefully when applied to switches. Commonly, the different queues in 802.1p switches are related based on strict priority. However other behaviours may be implemented.

6.3.3 Using RSVP Signaling for Admission to the 802 Network

RSVP signaling may be used in various forms for admission to 802 networks. In the simplest case, RSVP signaling is not actually processed by any device within the layer 2 subnetwork. Rather, devices sending into the network apply admission control by admitting or rejecting RSVP requests up to a provisioned limit. This is analogous to the example presented in the first paragraph of section 6.2.6, in which routers at the edges of a diffserv network are provisioned with a static SLA and admit or reject RSVP requests up to the limits specified in the SLA.

From a practical viewpoint, this approach is not really suitable for 802 networks. The primary reason is that 802 networks tend to be less formally provisioned than diffserv clouds (in part because bandwidth tends to be cheaper in the local area than in the wide area). The diffserv model presented assumes that the static SLA provisioned at ingress points to the diffserv network is reasonably reliable. The diffserv provider has incentive to carefully provision the network and to provide reliable SLAs because money changes hands based on the reliability of these SLAs. In addition, ingress and egress points to the diffserv network tend to be limited in number and carefully controlled. In layer 2 networks, the addition of ingress points is trivial and tends to happen more frequently than in a routed network. These concerns are particularly applicable in the common case of 802 networks that support large numbers of directly attached hosts. In this case, an SLA would be implicit for each host capable of transmitting into the 802 cloud.

6.3.4 The Role of the SBM in Providing Admission Control to 802 Networks

The SBM is a device capable of participating in an extended form of RSVP signaling that is suitable for shared networks. The SBM protocol can be enabled on devices *in* the 802 network at various densities, considering the same tradeoffs that result from enabling RSVP admission control agents in a diffserv network at various densities. At the lowest density, the network administrator may choose to enable a single switch in the core of the layer 2 network to act as the admission control agent for the entire layer 2 network. In this case, this device is the *designated SBM* (DSBM). At the other extreme, the network administrator may choose to enable every switch in the 802 network to act as admission control agents. In this case, the DSBM election protocol will result in the division of the 802 network into a number of *managed segments*, each managed by a DSBM. The denser the distribution of DSBMs, the higher the overhead associated with processing signaling messages, and the higher the quality/efficiency product which can be expected from the 802 subnetwork. The sparser the distribution, the lower the overhead and the lower the quality/efficiency product.

6.3.5 Mapping Intserv Requests to 802 Aggregate Service Levels

Admission control to the 802 network in response to signaling requests relies on a mapping of requested intserv service levels to the appropriate 802.1p tag. As is the case with mapping intserv to diffserv, a simple default mapping is assumed. DSBMs are able to override this default by appending a *TCLASS* object to RSVP RESV messages flowing through the DSBM en-route upstream. The TCLASS object is analogous to the DCLASS object described in section 6.2.6 and informs upstream devices of the 802.1p tag which should be used to mark packets sent on the admitted flow.

6.3.6 Beyond Aggregate Admission Control

Because SBMs are able to insert themselves in the RSVP control path it is possible for layer 2 devices to provide QoS functionality beyond the aggregate traffic handling and admission control described. SBMs can actually install aggregate or per-flow policers and finer-grain traffic handling, in response to RSVP signaling, thereby offering increased quality/efficiency product from the 802 subnetwork. However, because the incentive to achieve optimal efficiency in these networks is not high, it is unlikely that network administrators will choose to incur the associated overhead.

6.3.7 Behavior Expected When Sending onto 802 Shared Subnets

When an 802 subnet is managed by one or more DSBMs, the existence of the DSBM is advertised by the periodic transmission of *I_AM_DSBM* messages. Senders on shared subnets are expected to detect the

presence of a DSBM by listening for these messages. When an SBM is detected, senders are expected to divert RSVP signaling messages to the DSBM, rather than to the next layer 3 hop to which the message would otherwise be directed. This is required in order for the DSBM to be able to manage resources on the shared subnet. This functionality is referred to as *SBM Client* functionality. In addition, senders are expected not to tag packets for 802.1p prioritization unless such tagging has been approved in response to signaling (see section 8.3).

The restrictions described so far prevent hosts from marking traffic without policy approval, but impose no restrictions on the transmission of unmarked (best-effort) traffic. So long as devices in the network are capable of traffic isolation (by the use of dedicated switch ports and separation by tag or mark), there is no need to prevent senders from sending best-effort traffic. However, under certain conditions, network administrators may wish to limit any traffic sent by the host without network approval. To this end, DSBMs may be configured to advertise a *NonResvSendLimit* on the managed subnet. This value specifies the maximum rate at which hosts may send in the absence of an approved reservation. See section 8.1.3.2.

In order to maintain control of network resources, it is required that all senders sending onto a shared subnet implement full SBM client functionality. Senders not implementing this functionality should be isolated on separate subnets.

6.4 ATM Networks

ATM technology can be considered in the context of several types of subnetworks. For example, many providers offer large ATM based networks. In addition, ATM may be used as a campus backbone technology. The first example corresponds to the large routed networks illustrated at the center of the sample network. The second corresponds to the smaller ATM network illustrated in the customer domain at the lower-left corner of the sample network. When considered in the context of large provider networks, it is unlikely that ATM will be exposed directly to the customer as the QoS interface to the provider's network. It is more likely that ATM will be used to provision the provider's network such that it is able to provide a more abstract QoS interface, such as diffserv. One of the reasons for this is that the same scalability issues that apply to supporting per-conversation traffic handling in the form of per-conversation RSVP apply equally to ATM. Large provider's will not want to track per-conversation ATM VCs on behalf of customers. Instead, they are likely to provide VCs or VPs on a per-customer, per-aggregate service level basis.

6.4.1 ATM Per-Conversation or Aggregate Traffic Handling

In large provider networks, ATM VCs or VPs will likely be used as an aggregate traffic handling mechanism. Greater flexibility is possible when considering the use of ATM to provide QoS in smaller campus backbone type environments, where scalability is less of a concern. In these environments, the network administrator may map per-conversation intserv service requests to individual VCs. This approach is the current best practice recommended by the ISSLL working group of the IETF. It applies to switched VC environments, including LANE (ATM LAN emulation). Alternatively, the network administrator may choose to provision VCs or virtual paths (VP) to carry multiple conversations requiring the same service level, in so providing aggregate traffic handling.

6.4.2 ATM Edge Devices

ATM edge devices may provide varying degrees of QoS support. Regardless of the specific mechanism used, the edge device must address the fundamental problem of determining which traffic should be directed to which VC/VP. Several options are described below.

6.4.2.1 Dedicated Per-Conversation VCs

This mode of operation offers the highest quality/efficiency product from the ATM network but carries a cost in overhead. In this mode, it is necessary for an ATM edge device to initiate user network interface (UNI) signaling to establish a VC with the appropriate QoS parameters, for each conversation. Although

this could be done implicitly, based on the arrival of packets corresponding to new conversations and a marked DSCP and/or 802.1p tag, there would be little point in doing so[17]. If per-conversation VCs are to be established then the edge device should do so in response to explicit RSVP signaling. In this case, the edge device would have to appear as a layer 3 RSVP-aware hop or alternatively, as a DSBM. In the case that the edge device separates one IP subnet from another IP subnet it should behave as a layer 3 RSVP-aware routing hop. In the case of a mixed layer 2 subnet (in which there exist both ATM and non-ATM segments in the same IP subnet), the edge device would intercept RSVP messages in its capacity as DSBM.

In either case, VCs are established in response to RSVP signaling. A mapping from intserv service type and intserv quantifiable parameters to ATM service types and quantifiable parameters is defined by the ISSLL working group of the IETF. In this example, admission control at the RSVP level simply reflects the results of lower level UNI signaling.

6.4.2.2 Aggregate Per-Service Level VCs

This mode of operation offers a lower quality/efficiency product but at significantly reduced overhead. Aggregate traffic handling in an ATM subnetwork is similar but not equivalent to aggregate traffic handling in a diffserv or an 802.1p subnetwork. Diffserv and 802.1p subnetworks offer aggregate traffic handling in the form of disjoint PHBs (or priority queues) that are invoked by the arrival of a packet with the appropriate mark or tag. On the other hand, ATM subnetworks offer aggregate traffic handling by establishing a VC of the appropriate ATM service type. In an ATM subnetwork, it is necessary to determine when to establish VCs, between which endpoints to establish them, and for how much capacity. This is similar to the diffserv network-provisioning problem discussed in section 6.2.4, but somewhat more complicated. It is more complicated because VCs must be established between specific pairs of endpoints whereas diffserv PHBs are provisioned at individual nodes.

One approach is to establish a mesh of PVCs at network provisioning time. The permanent virtual circuit (PVC) mesh can then be used to provide the equivalent of SLAs at the edges of the ATM network. Edge devices admit RSVP requests subject to these SLAs. Another alternative is to allow aggregate VCs to be established and torn down based on demand. Either approach can be applied to signaled flows as well as to non-signaled flows. In the case of signaled flows, this mode is similar to the mode of operation described in section 6.2.7.1. In the case of non-signaled flows, VCs would be established on demand (as interpreted by the number of packets submitted for a specific service level to a specific destination). In the first case, packets are routed to a VC based on the intserv service type requested in the signaling messages for the associated flow. In the second case, packets are routed to a VC based on a mapping from DSCP, 802.1p or pre-provisioned classification criteria[18].

6.5 Small Routed Networks

Our sample network illustrates a small routed network in the top-right customer network. Small routed networks can be operated as diffserv provider networks (in which case, many of the considerations discussed in the context of large diffserv provider networks apply). However, these networks may also be operated as per-conversation RSVP/intserv networks. Since these networks are smaller than the large provider networks discussed in the context of diffserv, the tradeoffs are somewhat different. Specifically,

[17] Presumably, the goal of per-conversation VCs (as opposed to aggregate VCs) is good traffic isolation based on the resource requirements of each flow. However, in this case, the requirements of the flow are only indicated via an aggregate service level (in the form of DSCP or 802.1p tag). Therefore, the edge device would not know the appropriate parameters to use in establishing a dedicated VC.

[18] Note that mappings from DSCP (or 802.1p) to ATM service type are implied by the existence of mappings from intserv service types to each of these. In other words, we assume that intserv is a unifying abstraction for service types. Thus, any layer two medium-specific set of services should have a corresponding mapping from intserv services. This mapping can then be used to deduce mappings from one layer two medium to another. Thus, if there exist N interesting media and associated sets of services, only N mappings are required, rather than N squared mappings.

the number of conversations tends to be smaller, reducing the concerns regarding QoS overhead. In addition, the gain of over-provisioning may not be as high as it is in the large provider networks, due to the increased variance in resource usage. Therefore, efficiency might be more of a concern in these networks, arguing for support of a signaled QoS approach.

6.5.1 Hybrid of Signaled Per-Conversation and Aggregate QoS

A signaling-only approach precludes QoS for traffic generated by non-signaling applications. Therefore, such routed networks are likely to be operated using both signaled and provisioned QoS, just as the larger provider networks are operated. In the smaller networks, we are likely to see devices enabled to process RSVP signaling in greater densities than the provider networks. In addition, these devices will be configured to provide both per conversation traffic handling (based on signaled 5-tuple), in addition to aggregate traffic handling, (based on DSCP). Routers that are not enabled to process RSVP signaling will behave just as the routers in the core of the diffserv network, handling traffic based on DSCP exclusively. Thus, just as resource pools are separated in the large networks, between signaled and non-signaled traffic (by separation of DSCPs), they will be separated in smaller routed networks. Such hybrid functionality poses some interesting administration challenges and router functional requirements.

6.5.1.1 Required Router Functionality

In these hybrid networks, routers that are signaling-enabled are required to identify traffic that should be treated on a per-conversation basis as well as traffic that should be treated on an aggregate basis. These routers will classify arriving packets in a hierarchical manner. First, packets that match a signaled 5-tuple will be directed to the corresponding per-conversation traffic handling mechanism. Traffic that does not match a signaled 5-tuple will either be treated according to the DSCP marked in the submitted packet, will be re-marked based on some configured classification criteria, or will be treated as best-effort. How this traffic is treated at different routers in the small routed network, depends largely on the location of the device relative to trust boundaries and on the capabilities of hosts in the network.

6.5.1.2 Trust Boundaries

In smaller networks operated on behalf of a single administrative domain, trust boundaries tend to be vaguer than in the larger provider networks. In the larger networks, all customers submit traffic at well-defined ingress points, subject to SLAs. This is where money changes hands. Ingress devices to provider networks are either configured to remark all traffic, based on provisioned classification information, or to trust marked traffic but to police to per-service level aggregate limits negotiated in the SLA. In the smaller networks, under a single administrative domain, real money does not change hands within the network and policies tend to be more trusting. For example, routers in the engineering department may trust DSCPs marked in all submitted packets. Routers in the marketing department may do the same. Only routers at which traffic from multiple departments is merged would enforce a version of an internal SLA. Enforcement of the SLA would apply to traffic handled in aggregate. Traffic handled based on per-conversation reservations would be policed based on signaled per-conversation parameters.

6.5.1.3 Host Capabilities

Routers in the small routed network can be used to separate hosts of varying capabilities. (Note that similar considerations apply to smart switches in 802 LANs and 802.1p). As QoS functionality is rolled out, we can expect to see networks supporting hosts that:

1. Provide no QoS functionality
2. Mark DSCPs without signaling
3. Signal and mark DSCPs based on the results of signaling

If hosts with the varying levels of capabilities are all supported by the same router, then this router must use fairly complex classification policies to recognize traffic sourced by the different types of hosts and to apply the appropriate marking and policing. Specifically, traffic for which signaling requests were generated

should be policed based on 5-tuple (unless the router is configured for aggregate traffic handling, in which case, traffic should be policed based on DSCP). Traffic from hosts trusted to mark their own DSCP should be verified. Traffic from these hosts must be separated from traffic originating from hosts that are not trusted or not capable of marking their own DSCP.

Router marking and policing requirements can be simplified by separating different sets of hosts behind different routers (or switches with similar capabilities). In such a scenario, hosts providing no QoS functionality would be isolated behind routers that are configured to mark DSCPs on their behalf. QoS capable hosts would be placed behind routers that trust but verify marked DSCPs or respond to signaling requests.

6.6 Small Office and Home Networks

In the sample network diagram, we showed a couple of subnetworks as hosts, connected to the large provider network via a slow dial-up link. These can be considered to be small office or home PCs or networks (*SOHO* networks) connected to their ISP via a 56 Kbps modem link. From the perspective of the large provider network, the SOHO network is just another customer network, albeit a very small one. As such, much of the previous discussion regarding boundary functionality between providers and customers applies here. Beyond this however, the interface between the provider and the customer may be unique in that it may be a slow interface.

6.6.1 Aggregate Traffic Handling

For the foreseeable future, providers are unlikely to support signaling from SOHO network customers. Instead, they are likely to provide QoS by negotiating static SLAs with these customers, which will allow them to submit traffic marked for two or more aggregate service levels. Hosts in the customer networks may still generate RSVP signaling and may mark packets based on the results of this signaling. However, the provider will be unlikely to participate in the signaling process.

6.6.2 ISSLOW

Slow links present problems when they are required to carry both interactive audio traffic and data traffic. For example, 1500 byte data packets submitted to a slow link will occupy the link for almost half a second. Any audio packets that need to be sent after a data packet has been submitted to the link are subjected to severe latencies. ISSLOW functionality on a transmitting interface fragments the larger data packets, allowing audio packets to be interspersed, thereby largely eliminating the latency problem. This is particularly useful for e-commerce applications in which a customer may be, for example, perusing catalog images over the web, while speaking with a sales representative. It is also useful in peer-to-peer video-conferencing scenarios. In order to be useful, ISSLOW must be supported at least on the provider's sending interface and ideally on the customer's as well. ISSLOW can be invoked in response to RSVP signaling from the customer, or based on heuristics. An example of such heuristics would be the detection of a conversation which carries audio-size packets (28 - 128 bytes) at typical audio rates (6 Kbps - 64 Kbps). Detection of such a conversation would cause other traffic to be fragmented.

ISSLOW fragmentation is based on the relatively common PPP multilink protocol. Because it fragments at the link layer, it imposes relatively low overhead.

7 Applying Policies in the Sample Network

Policy is a much-overused term. There are policies in selecting network equipment, policies in selecting vendors, policies in selling services, policies in provisioning networks, policies in granting resources, and so on. In this section we discuss policies specifically related to the granting of network resources after the network has been built and all long term provisioning has been applied. The policies in which we are interested determine specifically how much resource of each type is granted to which users and applications.

7.1 Granting Resources Based on Policy vs. Availability

To the first order, resources are granted based on availability. For example, a provider's SLA, from the *customer's* perspective, specifies resources available at each service level, without regard for the particular customer's user or application that may claim these resources[19]. A customer policy might specify which users and/or applications in the customer's network are allowed to make use of these resources. Thus, rather than use available resources on a first-come-first-serve basis, the customer applies policy that restricts resource usage to certain consumers. Similarly, an RSVP-enabled router might be configured to admit requests for up to 100 Kbps traffic for the guaranteed service level. Policy would tell it which users or applications are entitled to use the 100 Kbps capacity.

7.2 Provisioned Policies

It is possible to provision certain policies in a top-down manner. For example, a network provider might provision devices in the provider's network to provide a specific customer a specific capacity at a certain service level. This is a fairly *coarse grain* policy. It can be simply provisioned, so long as there is an easy way to identify traffic originating from the customer. Assuming that all traffic from the customer originates from source addresses on, for example, subnet 2.3.4.0, then the network provider can provision devices within the network to recognize this source address and to police traffic sent from this address to the appropriate limits. This constitutes a *provisioned* policy of the provider regarding the specific customer.

We will now look at the resources available in the provider's network, from the customer's perspective. The customer would like to apply *finer-grain* policies. For example, the customer may want to restrict the usage of capacity in the expensive service level to a group of privileged users running important applications. Fine-grain policies such as these can be applied using a relatively static provisioning approach, however, the finer-grain the policies, the more cumbersome this approach becomes.

In order to apply fine-grain policies, it is first necessary to define these policies in terms of classification criteria and the resources to which classified packets are entitled. Let's say, for example, that all users from the marketing department are entitled to certain privileges distinct from those to which users from the engineering department are entitled. In this case, classification criteria would have to include the set of IP source addresses for all marketers and the set of IP source addresses for all engineers. If these IP groups of users were separated by subnet, this classification criteria could be expressed in a relatively compact form, however, in the general case, management of the required classification criteria would be extremely cumbersome.

An additional complication of such statically provisioned policy information is that it is hard to reconcile it with resource availability. For example, assume that it is necessary to install a policy to the effect that only executives using the IP telephony application are entitled to make use of the low latency services in the network. Assume that it is possible to define classification criteria that recognize traffic from executives using IP telephony, and also assume that the classification criteria can be used to direct this traffic to the low latency queue in each device. Recall that this queue has limited capacity and it may be possible to accommodate only 10 simultaneous users at any given node, out of the set of all executives.

The desired effect is a combination of resource availability and policy criteria, in the form: *allow up to 10 simultaneous executives using IP telephony to access these resources*. This is very difficult to implement using a static provisioning approach to policy. It would be necessary to provision classification criteria for only ten executives at a time. The problem is in determining which executives to allow at any point in time. The subset of executives that should be allowed at any time changes dynamically. Of course, this applies primarily to policies regarding applications that require high quality guarantees. For applications that do not require high quality guarantees, considerations regarding resource availability are not as strict and it is

[19] From the *provider's* perspective, these resources represent the fraction of available resources at the provider's ingress (and further into the network) that are granted to the specific customer and, as such, represent a policy regarding the specific customer.

therefore possible to simply allow all executives, based on statistical assumptions regarding executive resource usage.

7.3 Dynamic Enforcement of Policies

From the example in the previous section, we see that it is difficult to enforce fine-grain policies in a useful manner by using a provisioning approach. Policies, by their nature, are relatively static. However, efficient *enforcement* of these policies requires a more dynamic approach than provisioning. In this section we discuss the application of policy based on dynamic signaling. This approach is particularly applicable to applications that signal and that require relatively high quality guarantees.

We have previously discussed the use of signaling to effect dynamic admission control based on the availability of resources. This approach relies on the appointment of admission control agents in the signaling path. These agents consider the availability of requested resources along a path before admitting a resource request. If the resources are available, devices along the path install classification criteria corresponding to the traffic for which resources were requested. We can enforce policies by requiring the admission control agents to consider not only resource availability but also policies regarding who is entitled to these resources.

There are a number of advantages to applying fine-grain policies in this manner. First of all, this approach separates classification criteria from policy information. It allows the network administrator applying policy to think in terms of users, groups of users, and applications. At various points in the network at which the administrator wishes to enforce policy, the administrator constructs a database of users and applications and the resources to which they are entitled. Admission control agents at these points can then make policy decisions by comparing the requesting user and application (as indicated by the policy objects included in signaled resource requests), against the policies constructed by the network administrator.

In addition, this approach installs classification criteria in devices dynamically, in response to the results of a signaled request. The results of the signaled request are based on merging of both resource availability and policy information. The effect of this approach (as applied to the example of policies regarding executives using IP telephony) is that at any time, the classification criteria installed in a device will allow resources only to a subset of executives that does not exceed the capacity available.

Dynamic enforcement of policies in response to signaling is implemented by requiring certain admission control agents to apply a policy check in the process of admitting or rejecting a signaled resource request. This is typically implemented by enabling a device in the network through which data (and resource requests) flow, to outsource policy requests. Outsourcing policy requests consists of stripping policy objects describing the requesting user and application from a signaled resource request, and forwarding the policy objects, with a specification of the requested resources, to a policy server. The policy server then consults a database of users, applications, and privileges to which they are entitled and returns an admit/reject decision to the network device. The network device acts as a PEP and the policy server is the PDP.

7.4 Scope of Policies

The scope of a specific set of policies generally does not extend beyond a single administrative domain. For example, the policies of the large network provider determine the allocation of resources among customers of the provider. The provider does not care which of each customer's users are making use of the resources allotted to the customer. That is a matter of each customer's internal policies.

The policy objects inserted by the hosts that originate resource requests are very fine-grain. They describe individual users and the application used by the user. The scope of these objects is the customer network. These objects can be acted upon by PEPs and PDPs in the customer network, but are not useful to the provider's network. In the short run, the provider's networks will tend to be relatively statically provisioned, supporting static SLAs only. As a result, providers have little need to dynamically enforce policies. Their policies can be enforced via static provisioning, as described in section 7.2.

However, in the long run, providers will offer dynamic SLAs to their customers. These will allow customers to grow or shrink their resource usage, subject to policies. This flexibility will be reflected in the form of dynamic SLAs and will be supported by signaling-based admission control within the provider's network. In this environment, the provider will need to dynamically enforce policies regarding varying resource usage by different customers. The provider can be expected to apply the same approach described for dynamic enforcement of policies in the customer's network. However, the provider's admission control agents will apply policies based on policy objects describing customers, not individuals *within* a customer's network. If the provider processes per-conversation signaling requests from customers, it will be necessary to insert policy objects describing the customer in the signaling request. These may either replace (or be appended to) the original user/application policy objects. In general, as resource requests traverse administrative domain boundaries, it will be necessary to insert policy objects that are meaningful to each domain interested in dynamic enforcement of policies.

Note that, if aggregate signaling is used within the provider's network, as described in section 6.2.7.3, then policy objects pertinent to the provider are easily separated from those pertinent to the customer. In fact, the customer's policy objects remain invisible to the provider.

7.4.1 Multicast and Policy Objects

RSVP signaling is designed to merge reservations for multicast resources as appropriate. For example, requests from two receivers for resources for the same multicast session will automatically be merged along paths that are common to both receivers. This is appropriate from a resource perspective as both receivers can be satisfied using only one set of resources. However, such merging of requests complicates policy decisions. If policies dictate that one receiver is entitled to resources and another is not, what is the appropriate policy decision? Is it to admit the request, thereby enabling a *free-rider*, or is it to reject the request, thereby penalizing an entitled receiver? Furthermore, how are the policy objects conveyed upstream? Should merged requests include all policy objects from each pre-merge request? This approach could lead to unmanageably large sets of policy objects. Many of the multicast issues affecting policy have not yet been resolved.

8 The Microsoft QoS Components

In this section, we'll describe the QoS components provided in the Microsoft® Windows® family of operating systems and how they are used to implement the mechanisms described above. Windows 98 contains only user-level components, including:

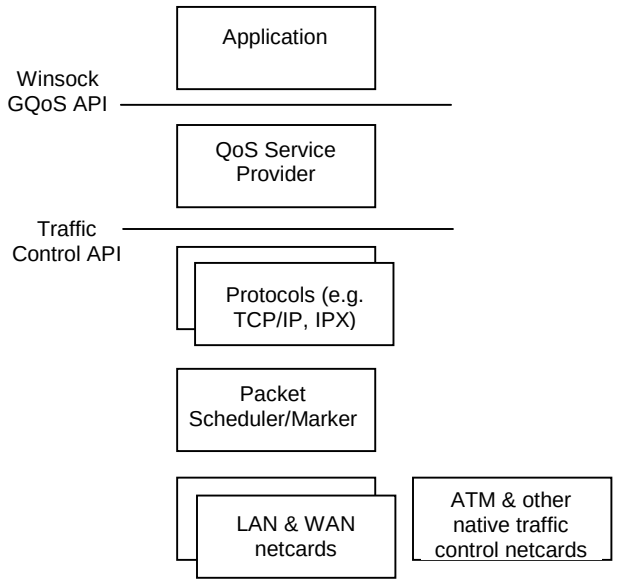
- The application component described in Section 8.1.1.
- The Winsock2 and GQoS APIs described in Section 8.1.2.
- The QoS service provider described in Section 8.1.3.

The Windows 2000 operating system contains all of the above as well as all the other components described in this section.

There are two primary groups of QoS components - those that reside in the host protocol stack and those that comprise the SBM and Admission Control Service (ACS).

8.1 The Host Protocol Stack

The following diagram illustrates the host protocol stack:



In the following paragraphs, we'll describe each QoS related component.

8.1.1 Application

Applications reside at the top of the stack. These may or may not be QoS-aware and may require varying qualities of guarantees. We recommend that applications that are session oriented and that can benefit from QoS, use the Generic QoS (GQoS) API. This is especially important for applications requiring high quality guarantees. Applications that are QoS aware invoke the services of the underlying QoS service provider (QoS SP) via the GQoS API. It is strongly recommend that ISVs implement the minor changes required to add GQoS support to Winsock2 applications. It is also expected that network administrators will require multimedia applications to conform to the GQoS API specification in order for them to be broadly deployable without abusing network resources. Mission critical, non-multimedia applications, such as client/server database applications, will have to conform to the GQoS API in order to enable network administrators to prioritize these applications on corporate networks.

Certain management utilities may be used to invoke QoS on behalf of applications that are not QoS-aware. These work via the traffic control API (TC API). Applications that are not QoS-aware will not be able to receive the quality of guarantees that would otherwise be achievable unless the underlying network is accordingly over-provisioned.

The following applications are currently enabled to use the GQoS API:

- NetMeeting® conferencing software (Windows 98 and Windows 2000)
- TAPI 3.0 (Windows 2000)

Following the release of Windows 2000:

- Windows Media™ Technologies
- A major enterprise resource planning (ERP) application

- Other multimedia and non multimedia applications, to be announced

8.1.2 Winsock2 & GQoS API

The Winsock2 API is a common API for use by network applications. Several Winsock commands carry QoS parameters and can be used to invoke QoS services from the operating system. These commands comprise a subset of the Winsock2 API, known as the GQoS (generic QoS) API. The purpose of this API is to enable applications to invoke the QoS they need with little understanding of the QoS mechanisms available or the specific underlying network medium. The API is very abstract and requires only very simple directives from the application. For applications that are QoS savvy and that do want additional control over the underlying mechanisms, extensions to the API provide additional control.

In the spirit of simplifying the interface presented to the application programmer, the GQoS API does not expose RSVP, diffserv, 802.1p or any other protocol or media-specific QoS mechanism to the application programmer. Instead, the sending application programmer specifies one of the following services:

- Guaranteed (generally specified for low and bounded latency applications, such as interactive voice)
- Controlled Load (generally specified for applications that are somewhat jitter tolerant but require the appearance of a lightly loaded network with a specific capacity, for example, streaming video)
- Qualitative (specified for applications that require better than best-effort service but are unable to quantify their requirements)

In addition to specifying a service, the sending application is expected to provide an indication of its average sending rate. It is recommended that applications also include an application ID and sub application ID (corresponding to the specific application sub-flow, such as *print flow* vs. *time-critical database transaction*). The application IDs are especially important for applications invoking the Qualitative service, as these provide no quantitative criteria by which to evaluate the application's impact on the network. Receiving applications must, at a minimum, indicate to the GQoS API that they are interested in network QoS. Certain qualitative applications may be allotted network QoS in response to the sender's use of GQoS, with no requirement for the receiver to invoke the GQoS API.

The underlying QoS service provider coordinates the various QoS mechanisms in the network in response to the application's request. These mechanisms include RSVP signaling and traffic scheduling, as well as DSCP marking, 802.1p tagging[20] that is based on the results of signaling.

8.1.3 The QoS Service Provider

The QoS service provider (QoS SP) is the entity that responds to the GQoS API. It provides the following services:

- RSVP signaling
- QoS policy support
- Invocation of traffic control

8.1.3.1 RSVP Signaling

RSVP signaling is generated by default on behalf of applications using the GQoS API. The QoS SP initiates and terminates all RSVP signaling on behalf of the applications. It provides status regarding reservation state to applications that are interested, but does not require the application to understand RSVP signaling.

8.1.3.2 SBM Client Functionality

The QoS SP provides full SBM client functionality. This means that it detects the presence of a DSBM on a shared subnet and routes signaling requests via the DSBM (as opposed to the next layer-3 hop). In addition,

[20] We will use the term *marking* to refer to both DSCP marking and 802.1p tagging.

the QoS SP presents the results of the DSBM's advertised *NonResvSendLimit* (see section 6.3.7) to applications via the GQoS API. This enables GQoS compliant applications to avoid or restrict their sending in response to administrator policies.

8.1.3.3 QoS Policy Support

In support of QoS policy, the QoS SP inserts a Kerberos encrypted Windows NT user ID into RSVP signaling messages, both on sender and receiver. In addition, the QoS SP inserts any application identification provided by the application via the GQoS API. The inserted objects identify the Microsoft® Windows NT® operating system user and application such that application and/or user-specific policy can be applied in the network.

8.1.3.4 Invocation of Traffic Control

The QoS SP actually enforces policy by invoking traffic control in accordance with the network's response to signaling messages. In general, the QoS SP identifies two types of traffic control: greedy traffic control and non-greedy traffic control. Non-greedy traffic control is invoked in immediate response to an application's request for QoS. Greedy traffic control is enabled only if (and to the degree) approved by the network, in response to the RSVP signaling.

The TC API is quite complex and provides a high degree of control. The QoS SP abstracts the complexity of the TC API via the GQoS API such that applications can remain relatively simple.

8.1.4 The Traffic Control API

The TC API provides the QoS SP and third party traffic management applications with a high degree of control over traffic control (TC) functionality in the kernel. The fundamental APIs that comprise the TC API are *CreateFlow* and *CreateFilter*. CreateFlow causes a flow to be created in the kernel network stack. The flow has certain actions and characteristics associated with it. These include marking behavior (DSCP, 802.1p, and other media-specific marks or tags), packet scheduling behavior and other media-specific behavior, as appropriate. CreateFilter is called to attach a *filter* to a flow. A filter specifies classification criteria, which determine the set of packets that will be directed to the associated flow. Multiple filters may be attached to a single flow. Filters may be fully specific (no wildcards) or may include wildcards. The generic packet classifier (GPC) is used for the purpose of packet classification. Scheduling parameters are expressed using the common *token-bucket* model. Filters are expressed in the form of an IP 5-tuple and a mask.

Note that, at present, the TC API and the corresponding functionality are applicable to transmitted traffic only. In future versions of Windows operating systems, TC functionality will be available to control the treatment of received traffic as well as transmitted traffic. Traffic control functionality is available in Windows 2000, but not in Windows 98 (with the exception of limited DSCP marking).

The TC API separates traffic control consumers from traffic control providers. In the illustration above, the QoS service provider is a traffic control consumer, while the packet scheduler and ATM network card are traffic control providers.

8.1.4.1 Traffic Control Providers

Traffic control providers include all modules that implement any traffic control functionality in response to the traffic control API. Traffic control functionality available in Windows 2000 includes:

- Packet scheduling
- 802.1p marking
- DSCP marking
- ISSLOW link layer fragmentation (per PPP multilink) for latency reduction on slow links
- ATM VC control and cell scheduling

The packet scheduler component is implemented as an intermediate driver. It provides traffic control functionality over standard LAN adapters, as well as over NDISWAN and WAN drivers. Since ATM LANE presents an Ethernet interface to the network stack, the packet scheduler also provides traffic control over LANE. On the other hand, classical IP over ATM (CLIP) provides traffic control functionality directly to the traffic control API without requiring the packet scheduler. Additional traffic control providers planned for the future include cable modem drivers, P1394 drivers, and other media- specific drivers.

8.1.5 Packet Scheduler

The packet scheduler is used to provide traffic control over drivers and network cards that have no inherent packet scheduling capability. It schedules packets on separate QoS queues as created via the TC API. It also is responsible for effecting the marking of DSCPs and media-specific priority tags (such as 802.1p) on transmitted packets.

8.1.5.1 Scheduling

The scheduling components of the packet scheduler include:

- A conformance analyzer, which checks packets for conformance to a traffic descriptor
- A shaper, which delays packets until they can be legitimately transmitted per the traffic descriptor (non-work-conserving queuing)
- A sequencer, which determines the sequence in which packets from different flows may access the link when it is congested

Flows may be individually configured in the packet scheduler for variations of the following modes:

- Borrow mode - allows traffic on the flow to borrow resources from higher priority flows that are temporarily idle (at the expense of being marked non-conforming and demoted in priority)
- Shape mode - delays packets submitted for transmission until they conform to a specified traffic descriptor (non-work conserving)
- Discard mode - discards packets that do not conform to a specified traffic control descriptor.

By default, the packet scheduler implements a mapping from requested service type to one of these modes, and to an internal priority level, as follows:

Service Type	**Mode**	**Priority**
Network Control[21]	Borrow mode	Highest priority
Guaranteed Service	Shape mode	High priority
Controlled Load	Borrow mode	Medium priority
Qualitative	Borrow mode	Low priority
All other traffic	Borrow mode	Lowest priority[22]

These defaults may be overridden as appropriate.

The packet scheduler provides the flexibility to invoke a broad range of traffic control functionality, including both work -conserving and non-work-conserving schemes, the ability to proportionately share link resources (such as in weighted fair queuing) and so forth. It is possible to simultaneously configure different flows for different modes.

[21] This service type may be requested via the traffic control API, but is not available via the GQoS API. It is reserved for use by critical traffic management applications.

[22] Note that packets deemed non-conforming to the traffic descriptor are demoted in priority to a level lower than that of best-effort traffic. This demotion may be reflected in internal sequencing as well as marking and tagging.

8.1.5.2 Marking

In addition to scheduling, the packet scheduler effects the marking of transmitted packets. The reason that this functionality is mediated via the packet scheduler is to enable it to demote non-conforming packets. By default, packets are marked based on a mapping from the service type associated with a flow, according to the following mapping:

Service Type	DSCP	802.1p
Network Control	30 (6)	7
Guaranteed Service	28 (5)	5
Controlled Load	18 (3)	3
Qualitative	0 (0)	0
All other traffic	0 (0)	0

Note: The actual DSCP is a six-bit field carrying the value indicated. Three of the six bits comprise a subset of the DSCP field, formerly referred to as the *IP Precedence field.* The equivalent IP precedence values are shown in parentheses.

There are several cases in which the default mapping may be overridden. These are described below. (Note that this describes marking behavior in response to the TC API, which *bypasses the policy mechanisms of the QoS SP and the network*. Consequently, this behavior does not fully describe marking in response to the GQoS API and network policy, which is mediated by the QoS SP. For information regarding marking in response to the GQoS API, see section 8.3

- Non conformance - packets that are deemed by the packet scheduler to be non-conforming to the traffic descriptor provided, may be marked with a mark other than the default mapped from the service type. Typically, the mark will indicate a lower priority than that which would be applied to conforming packets.
- Registry override - it is possible to define new static mappings in the registry. These can be defined on a per-interface basis. Mappings can be defined both for conforming and non-conforming packets.
- TCLASS and DCLASS - these objects can be supplied with the *CreateFlow* API (or the related *ModifyFlow* API) at any time, to dynamically override the 802.1p or DSCP marking, respectively, for the flow. These objects are not directly accessible to applications using the GQoS API. Rather, the network is expected to signal these to the QoS SP, which in turn provides them to traffic control via the TC API.

Note that the packet scheduler marks neither DSCP nor 802.1p directly. Rather, it *effects* this marking. In the case of the DSCP, the marking is still performed by the core operating system. However, in the case of 802.1p, the marking is actually performed by the network card driver (or hardware) which generates the packets. The packet scheduler provides the network card driver a suggested 802.1p value with each packet. Ethernet drivers may use the suggested value directly. Other media drivers interpret the suggested value and map it to their media-specific link layer tagging or marking mechanism, as appropriate.

8.2 The Subnet Bandwidth Manager and Admission Control Service

The SBM and the ACS are Microsoft's QoS policy components.

8.2.1 The Subnet Bandwidth Manager

The SBM protocol defined by the IETF extends RSVP to be useful in a shared media subnetwork. In shared media subnets, there is no single agent accountable for the shared resources. The SBM protocol defines how agents in the subnetwork elect a *Designated SBM* (or DSBM). The DSBM then advertises its existence on the shared subnet and is accountable for the shared resources of the subnet. Devices sending RSVP PATH messages onto a shared subnet are required to detect the presence of the DSBM and to route their messages *through* the DSBM, instead of directly to the next layer 3 hop. The DSBM is then able to apply admission

control based on the resources (and policies) of the shared subnet, before relaying the RSVP message to the next layer 3 hop.

Microsoft's ACS is a service that combines the resource-based admission control functionality of an SBM with policy based admission control using the Active Directory. The ACS leverages the fact that the SBM (by advertisement of its presence on a shared subnet) is able to insert itself into the RSVP reservation path and can, therefore, effect admission control. To use the ACS to apply policy-based admission control, it is necessary to enable the ACS on a Windows 2000 server. The SBM component of the ACS will then run for election with other DSBM capable devices on the same shared subnet[23]. If the ACS is to be used for admission control on the subnet, it may be necessary to disable DSBM functionality on other devices (switches and routers) on the subnet.

8.2.1.1 The Local Policy Module and Extensibility

When PATH or RESV messages are intercepted by the SBM, they are handed off for policy processing by a *Local Policy Module* (LPM). Microsoft's LPM simply extracts the policy-related objects from the RSVP message, applies the appropriate Kerberos processing to the user ID, and compares the requesting user ID, and the resources requested, against privileges configured in the Active Directory. Based on the results of the comparison, the RSVP request is either admitted or rejected by the ACS. The interface between the SBM and the LPM is an open interface -— the *LPM API*. Third party ISVs may use this interface to install alternate policy modules in the ACS. These policy modules may use intermediate third party policy servers rather than accessing the active directory directly. They may also be used to provide special resource-based admission control such as might be required in the case of cable modem head-ends. Multiple policy modules can be cascaded in series in a single ACS server.

The extensibility described in the previous paragraph enables third parties to use the ACS to apply policies against their policy servers. In this model, the ACS is acting as a policy enforcement point (PEP)[24]. As discussed below, it is usually preferable to use a router as a PEP. Standard routers, acting as PEPs, use the COPS protocol to outsource policy decisions to a policy server, which in turn uses the Active Directory as the policy data store. In this environment, additional extensibility is provided by allowing Microsoft's LPM to be run on third party policy servers. This mode of operation enables the policy server to readily parse Microsoft's Active Directory resident QoS schema.

8.2.2 Applicability of the ACS

The functionality of the ACS is nothing more than standard PEP/PDP functionality. In the near future, routers and switches will provide this functionality, since these are the actual policy enforcement points and

[23] Note that the DSBM election protocol defines a prioritization by device type. Highest priority is given to switches, with routers next and hosts last. This order favors devices that optimize the quality/efficiency product of the shared subnet. For optimal quality/efficiency product, a layer 2 subnet should be constructed entirely of DSBM capable switches with dedicated ports (no dumb hubs or yellow wires). In this case, the DSBM election protocol will divide the shared subnet into a set of managed segments, each controlled by a DSBM. The layer 2 network, from an RSVP perspective, will appear to have a routed topology. By comparison, if a host or router at the edge of the layer 2 subnet is the DSBM, it makes admission control decisions without detailed knowledge regarding the internal topology of the subnet. Therefore, a host or router DSBM reduces the quality/efficiency product of the subnet. On shared subnets, which are usually over-provisioned, the increased quality/efficiency product rarely justifies the increased overhead that results from every switch acting as a DSBM.

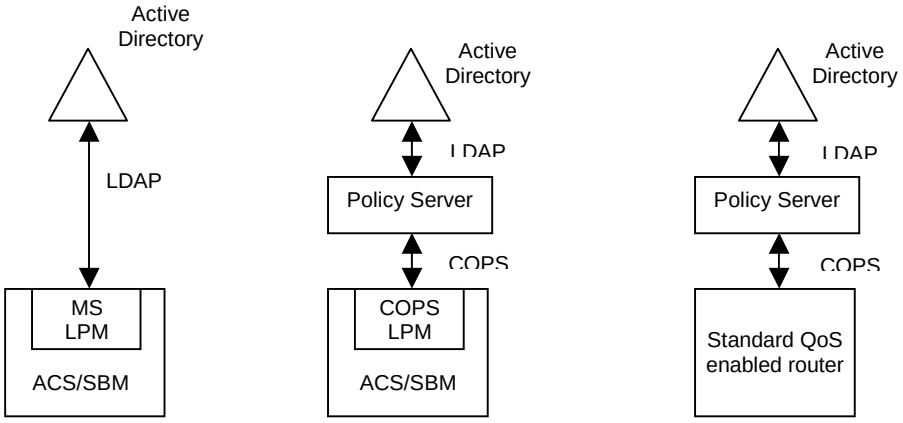
[24] The ACS is a policy enforcement point in the sense that it is able to veto signaled admission control requests. Unlike router-based PEPs, it is not strictly speaking, the final enforcer. Ultimate enforcement is the ability to forward packets or to not forward packets, which is reserved for devices that are actually in the data path. Nonetheless, by blocking admission control, the ACS is able to prevent the allotment of high priority resources to traffic on signaled flows.

these are the devices through which reservation messages naturally flow. In the interim, Microsoft's host-based implementation of the ACS enables early adopters of QoS technology to effect policy-based admission control.

It is a common misconception that it is necessary to install an ACS on every subnet in order to benefit from QoS policy. This is not the case. An ACS enables significant control over network resources when installed even on a small number of carefully selected subnetworks. It is true that the SBM-based ACS functionality imposes awkward topological constraints in certain conditions. In particular, when it is necessary to apply policies that are specific to a point to point link (such as a WAN link), the SBM-based ACS cannot be readily used. In these circumstances the routing and remote access service (RRAS)-based ACS should be used. The RRAS-based ACS provides point to point routing with ACS policy control and can be used, for example to drive WAN links.

8.2.3 Variations of the ACS

The following diagram illustrates variations of the SBM/ACS described previously.



The leftmost example illustrates a Windows 2000-based host, on which the ACS service is enabled. The ACS uses the standard Microsoft LPM that uses LDAP to directly access the active directory.

The center example shows the same SBM platform, however, the Microsoft LPM has been replaced with a third party COPS LPM. The COPS LPM accesses an intermediate policy server using COPS. The policy server retrieves policy data from the active directory, using LDAP. This configuration allows policy decisions to be offloaded from the ACS/SBM platform. The benefits of doing this are twofold: first, the network device that intercepts the QoS control messages can be a very lightweight device (since the policy decision work is done elsewhere). Second, a distributed set of policy servers can make distributed policy decisions.

Finally, the rightmost example illustrates an industry standard router. The router uses COPS to offload the policy decision to a third party policy server. The third party policy server may use Microsoft's LPM to parse Active Directory QoS schema.

8.3 How Hosts Mark and Shape Traffic Based on Network Policy

In the previous section, we discussed the use of the TC API to mark and shape traffic. Since marked packets may obtain resources in the network that would otherwise be available to other packets, we consider marking to be *greedy* behavior. As such, marking should be subjected to policy controls. Shaping, by comparison, can only *reduce* the rate of transmitted traffic (no amount of shaping can make a 10 Mbps Ethernet interface transmit faster than 10 Mbps). Therefore, shaping is considered *non-greedy* behavior and need not be subjected to policy controls.

In order to assure that packet marking is subjected to policy control, the TC API is made available only to administrative authorities (it can be invoked only by applications having administrator privileges for the operating system). These include the QoS SP and, possibly, additional network management applications. Non-administrative applications are unable to directly effect packet marking. Instead, these ask the QoS SP for a particular service level (and, in the case of quantitative applications, for a specific quantity of resources at this service level). The behavior of the QoS SP in response to application requests is described in the following paragraphs.

In general, the QoS SP applies non-greedy traffic control (requested shaping behavior) on behalf of the application as soon as the application requests QoS. At the same time, the QoS SP begins RSVP signaling to the network. Network devices along the data path review these signaling requests both as the PATH message flows downstream and as the RESV message flows back upstream. PEPs are able to assess the impact of the resource request on their available resources. They use PDPs to subject the request to verification against installed policies. When verifying admissibility, PEPs that use aggregate traffic handling assume default mapping from the requested intserv service level to an aggregate service provided by the device. Alternatively, PEPs and PDPs may work together to dictate an alternate mapping by returning to the host a DCLASS or TCLASS object (to effect marking of the DSCP or 802.1p tag for packets transmitted on the corresponding flow). Any PEP along the path may veto the reservation request due to insufficient resources or restrictive policies. A veto has the effect of refusing admission control to the requesting hosts and preventing the transmitting host from marking packets.

In the case that the RESV arrives at the transmitting host, the resource request has successfully transited all admission control agents in the network and may be considered admitted. Admission of a request permits the QoS SP on the transmitting host to invoke greedy traffic control, marking packets based on a default mapping, or according to a returned DCLASS or TCLASS object. As a result, packets are marked for priority only while the network approves the transmitting host's resource request. Until the request is admitted (or at any time that the request is rejected or revoked), the QoS SP will not mark packets for better than best-effort behavior. The default mappings used by the host are as indicated in the tables in section 8.1.5.2. Note that for qualitative service, the default marks are equivalent to best effort. In order to cause traffic on qualitative flows to be marked for anything other than best-effort, it is necessary for a PEP to return a DCLASS or TCLASS object to the transmitting host.

8.3.1 Coordination of Greedy Behavior not Subjected to Policy

The QoS SP does not signal to the network for applications that do not generate persistent traffic. If it is necessary to mark traffic generated by these applications, this must be done either by network management applications making direct use of the traffic control API, or by the network itself. Persistent applications (that mark in response to signaling and policy) share the same resources as non-persistent applications or other applications that do not signal. Therefore, network management applications that effect the marking of traffic on behalf of non-signaling applications must be sure to reconcile the resources used by these applications against the resources used by signaling applications. The network administrator must enforce static limits on the type and quantity of resources available through signaled policy and those claimed by marking without signaling and policy, or must dynamically manage admission control to both pools of resources simultaneously. This requirement is described in section 4.4.

9 Current QoS Functionality Available in Network Equipment

In this section, we discuss the current state of implementation of QoS functionality in different types of network equipment

9.1 Hosts

As described, Windows hosts provide a broad range of QoS functionality, including signaling, policy, marking and traffic shaping. Host functionality integrates marking and shaping behavior with signaling and policy and presents a unified mechanism-independent API to applications. In addition, traffic control is

directly accessible to network management applications. In general, RSVP signaling is available on Windows 2000 and Windows 98. Traffic control (including marking and scheduling) is available only on Windows 2000.

Various Unix and Linux implementations provide a range of QoS functionality including RSVP signaling, diffserv marking and sophisticated scheduling algorithms. However, these are generally not abstracted into a unified API and are not integrated with network policy in a manner that can provide a full range of quality of guarantees.

9.2 Routers

9.2.1 RSVP Signaling

All major router vendors support per-conversation RSVP signaling in varying degrees on some subset of their products[25]. In general, RSVP admission control may be configured separately from the traffic handling mechanisms on these routers. This enables network administrators to mix and match per-conversation admission control with either aggregate or per-conversation traffic handling. SBM client functionality is available from several router vendors.

Router vendors are in the process of implementing functionality to translate intserv requests to diffserv service levels. A major router vendor is demonstrating DCLASS functionality based on network policies.

9.2.2 Traffic Handling

Those routers providing RSVP support also provide the corresponding per-conversation traffic handling mechanisms. In addition, most router vendors provide a simple form of diffserv today, by their ability to group traffic for different treatment based on values in the IP precedence field or TOS field of packet headers.

9.2.3 Policy Functionality

Most router vendors provide SNMP monitoring (and in certain cases configuration) of QoS functionality. Several vendors provide CLI or CLI-like interfaces to this functionality. A small number of router vendors provide COPS interfaces to corresponding policy servers, which may be used to apply both signaled and provisioned QoS. Provisioned QoS interfaces are more common today than signaled QoS interfaces, however increasing numbers of routers are adding support for management of signaled QoS via COPS.

9.3 Switches

9.3.1 Signaling and SBM Functionality

Several switch vendors support varying degrees of SBM functionality and are able to act as DSBMs on shared subnets. Some switches return a TCLASS object in response to host signaling.

9.3.2 Traffic Handling

High-end and midrange switches support 802.1p today. This is a relatively new standard and so legacy switches generally will not support 802.1p. It is unlikely that low-end LAN devices, (such as the dumb hubs common in many offices) will support 802.1p. Many newer routers will mark 802.1p headers in packets they submit to a LAN. Windows 2000 hosts will do so if the network card driver is 802.1p capable and enabled for 802.1p marking. Note that the IEEE has yet to standardize a mechanism for automatically negotiating 802.1p functionality. As a consequence, it is possible to incorrectly configure senders and receivers so that they are unable to communicate. It is generally recommended that network administrators deploy 802.1p on a subnet wide basis.

[25] At least one major router vendor is also in the process of implementing aggregate RSVP signaling.

9.4 Policy Servers

QoS policy today focuses on top-down provisioned QoS. Network administrators may use existing policy servers to configure QoS parameters in network devices to prioritize aggregated traffic based on addresses or ports. For example, traffic from the engineering department may be given priority over traffic from the marketing department, based on different source IP subnet addresses. This type of policy is very broad, or *coarse-grain*, as opposed to, for example, per-application or per-user policy. Such policy is usually configured independently in each network device with little effort to integrate policies across devices.

High-end policy management vendors are developing more integrated policy-based management, using a central data-store to push consistent configuration information to multiple devices. In addition, RSVP-capable routers are being extended to recognize RSVP policy elements and to communicate directly with directory based policy data, using LDAP, or indirectly, via COPS and emerging policy servers.

Standard policy schemas for policy data are currently under definition in the IETF. These address both configured and signaled QoS. While the majority of commercially available policy management systems today provide schemas for provisioned QoS, Microsoft's Active Directory and ACS provide a schema for signaled QoS.

10 IETF References

Many of the concepts discussed in the previous pages are described in IETF drafts and RFCs that are in various stages of standardization. These are listed below. Note that many of these are works in progress and as such, should not be considered official standards. Nonetheless, much of the described functionality is already being offered by equipment vendors, thus leading to the establishment of de facto standards.

10.1 RSVP

See documents listed under http://www.ietf.org/html.charters/rsvp-charter.html, including:

RFC 2205 - RSVP Functional Specification
RFC 2207 - RSVP Extensions for IPSec Data Flows

- RSVP aggregation - http://search.ietf.org/internet-drafts/draft-baker-rsvp-aggregation-01.txt

10.2 Intserv

See documents listed under http://www.ietf.org/html.charters/intserv-charter.html, including:

RFC 2210 - Use of RSVP with Integrated Services
RFC 2211 - Specification of the Controlled Load Quality of Service
RFC 2212 - Specification of the Guaranteed Quality of Service
RFC 2215 - General Characterization Parameters for Integrated Services Network Elements

10.3 Differentiated Services

See documents listed under http://www.ietf.org/html.charters/diffserv-charter.html, including:

RFC 2475 - Architecture for Differentiated Services
RFC 2474 - Definition of the Differentiated Service Field
RFC 2597 - Assured Forwarding PHB Group
RFC 2598 - Expedited Forwarding PHB

- Framework for differentiated services - http://www.ietf.org/internet-drafts/draft-ietf-diffserv-framework-02.txt
- Conceptual model for diffserv routers - http://www.ietf.org/internet-drafts/draft-ietf-diffserv-model-00.txt

10.4 Integrates Services Over Specific Link Layers

See documents listed under http://www.ietf.org/html.charters/issll-charter.html, including:

RFC 2382 - A Framework for Integrates Services and RSVP Over ATM
RFC 2379 - RSVP Over ATM Implementation Guidelines

- Integrated services over slow links (ISSLOW) - http://www.ietf.org/internet-drafts/draft-ietf-issll-isslow-06.txt
- The Subnet Bandwidth Manager (SBM) - http://www.ietf.org/internet-drafts/draft-ietf-issll-is802-sbm-08.txt
- Framework for integrated services over 802 networks - http://www.ietf.org/internet-drafts/draft-ietf-issll-is802-framework-07.txt
- Mapping integrated services to 802.1p - http://www.ietf.org/internet-drafts/draft-ietf-issll-is802-svc-mapping-04.txt
- Framework for the interoperation of intserv and diffserv - http://www.ietf.org/internet-drafts/draft-ietf-issll-diffserv-rsvp-02.txt
- Usage of the DCLASS object - http://www.ietf.org/internet-drafts/draft-ietf-issll-dclass-00.txt
- The qualitative service type - http://search.ietf.org/internet-drafts/draft-moore-qualservice-00.txt

10.5 QoS Policy

See documents listed under http://www.ietf.org/html.charters/rap-charter.html, including:

- Framework for policy based admission control - http://www.ietf.org/internet-drafts/draft-ietf-rap-framework-03.txt
- The COPS protocol - http://www.ietf.org/internet-drafts/draft-ietf-rap-cops-07.txt
- RSVP extensions for policy control - http://www.ietf.org/internet-drafts/draft-ietf-rap-rsvp-ext-06.txt
- COPS usage for RSVP - http://www.ietf.org/internet-drafts/draft-ietf-rap-cops-rsvp-05.txt
- Identity representation for RSVP - http://www.ietf.org/internet-drafts/draft-ietf-rap-rsvp-identity-04.txt
- COPS for provisioned QoS - http://www.ietf.org/internet-drafts/draft-ietf-rap-pr-00.txt
- Format for application IDs - http://www.microsoft.com/windows2000/library/howitworks/communications/trafficmgmt/appident.asp
-

11 Appendix A - Queuing and Scheduling Hardware/Software

Queuing and scheduling are the building blocks of the QoS traffic handling mechanisms. These are available both in standalone network devices as well as in host network components. The simplest network devices forward traffic from the source (ingress) interface to the destination (egress) interface in first-in-first-out (FIFO) order. More sophisticated devices are able to provide QoS by using intelligent queuing and scheduling schemes. We present an overview of these schemes in this section. Each of the queuing mechanisms described may be used to handle traffic on a per-conversation basis or on an aggregate basis.

11.1.1 Work-Conserving Queue Servicing

Traffic passing through a network device is classified to different queues within the device. A variety of queue-servicing schemes can then be used to remove traffic from the queues and forward it to egress interfaces. Most queue servicing schemes currently in use, are *work-conserving*. That is - they do not allow interface resources to go unused. So long as there is capacity to send traffic and there is traffic to be sent, work-conserving schemes will forward packets to the egress interface. If the interface is not congested then these schemes amount to first-in-first-out (FIFO) queuing. However, if the interface is congested, then packets will accumulate in queues in device memory, awaiting capacity on the interface. When capacity becomes available, the device must decide which of the queued packets should be sent next. In general, packets from certain queues will be given priority over packets from other queues. Thus, under congestion traffic is not serviced in a FIFO manner, but rather according to some alternate queue-servicing scheme.

Many work-conserving queue-servicing algorithms have been defined. Examples are *weighted fair queuing* (WFQ), *deficit round robin* (DRR), *stochastic fair queuing* (SFQ), *round robin* (RR), *strict priority*, etc. These all try to allocate some minimum share of the interface's capacity to each queue (during congestion), while allowing additional capacity to be allocated when there is no traffic queued on higher priority flows. These servicing schemes also try to minimize the latency experienced by packets on some or all flows.

11.1.2 Non-Work-Conserving Queue Servicing

A different type of queuing scheme is *non-work-conserving*. This type of scheme may allow interface capacity to go unused. These schemes are often referred to as *packet-shaping* schemes. A packet-shaping scheme limits the rate at which traffic on a certain flow can be forwarded through the outgoing interface. Packet-shaping is often used for multimedia traffic flows. In this case, there is no advantage to sending traffic sooner than necessary (voice data will generally not be played any faster than it was recorded) and downstream resources may be spared by limiting data transmission to the rate at which it can be consumed. Non-work-conserving schemes require a real-time clock to pace the transmission of traffic on the shaped queue.

It is possible to combine both work-conserving and non-work-conserving schemes on the same interface. In this case, work-conserving schemes may make use of capacity that is not used by non-work-conserving schemes.

11.1.3 ISSLOW

A special queuing mechanism is optimized for slow network interfaces. This mechanism is referred to as *ISSLOW* (integrated services over slow links). The purpose of this scheme is to dramatically reduce the latency that would be experienced by certain packets (typically, small audio packets) when the capacity of the interface is very low. It is specifically targeted at interfaces that forward onto relatively slow modem links. A typical 1500-byte data packet, once forwarded onto a typical modem link, may occupy the link for almost half a second. Other packets that have the misfortune of being queued behind the data packet will experience a significant latency. This is unacceptable for latency-intolerant (such as telephony) traffic. To avert this problem, ISSLOW scheduling mechanisms break the data packet into smaller packets (link-layer

fragmentation), such that they do not occupy the link for as long a period of time. Higher priority, latency-intolerant packets can then be interspersed between these smaller packets.

11.1.4 ATM

ATM interfaces fragment packets into very small cells. These cells are typically queued and scheduled for transmission by hardware on the ATM interface. Due to the small cell size, it is possible to schedule traffic very precisely and with low latency. ATM interfaces implement both work-conserving and non-work-conserving schemes. These do not require ISSLOW since the small cell size and typically high media rates do not present the latency problems observed on slow links.