UNIVERSITATEA "POLITEHNICA"
BUCUREŞTI

UNIVERSITÉ "JEAN MONNET"
SAINT-ETIENNE

Network Quality of Service

referat de doctorat

Răzvan Beuran

Coordonatori:

Prof. dr. ing. Vasile Buzuloiu

Prof. dr. Jean-Marie Becker

- 2002 -

Content

1 Introduction

Networks and especially Internet were used for many years by scientists, primarily for networking research and for exchanging information between each other [Pet-99]. Remote access, file transfer and e-mail were among the most popular applications. The model on which the network was based, datagrams, in which individual packets are forwarded independently to their destination, worked well for such applications. The World Wide Web changed fundamentally the Internet, being nowadays the world's largest public network. New applications, such as video conferencing, e-commerce, Internet telephony are being developed at an unprecedented speed. As we entered the twenty-first century, the Internet is destined to become the all-purpose global communication infrastructure. Network QoS may also an important issue for custom applications, such as the ATLAS data collection system being developed at CERN [Bar-01].

This phenomenal success of Internet has created new challenges. Many new applications have very different requirements from those that the Internet was originally designed for. One important issue is performance assurance. The datagram model, on which the Internet is based, has few resource management capabilities inside the network and therefore can not provide any resource guarantees to users. Accessing a web site or making an Internet phone call may be very difficult if some parts of the network are so busy that packets cannot get through. Real-time applications, such as video conferencing, also require some minimal level of resources in order to operate effectively. As the Internet becomes indispensable in our life and work, the lack of predictable performance is certainly a problem which needs addressing.

Another issue is service differentiation. Because the Internet treats all packets in the same way, it can only offer a single level of service. The applications, however, have diverse requirements. Interactive applications such as Internet telephony are sensitive to latency and packet losses. When the latency or the loss rate exceed certain levels these applications become literally unusable. In contrast, a file transfer can tolerate a fair amount of delay and losses without much degradation of perceived performance; nevertheless the efficiency of bandwidth usage decreases. Customer requirements also vary depending on what the Internet is used for. For example, organizations that use the Internet for bank transactions or for control of industrial equipment are probably willing to pay more to receive preferential treatment for their traffic. For many service providers, offering multiple levels of services to meet different customer requirements is vital for the success of their business. The capability to provide resource

assurance and service differentiation in a network is often referred to as quality of service (QoS).

Resource assurance is critical for many new Internet applications to flourish and prosper. The Internet will become a truly multiservice network only when service differentiation can be supported. Implementing these QoS capabilities in the Internet has been one of the toughest challenges in its evolution, touching on almost all aspects of Internet technologies and requiring changes to the basic architecture of the Internet. For more than a decade the Internet community has made continuous efforts to address the issue and developed a number of new technologies for enhancing the Internet with QoS capabilities. These efforts continue nowadays.

In what follows I shall make a general presentation of network QoS (Chapter 2), emphasizing the main requirements for a QoS enabled networking environment. Chapter 3 focuses on per-hop QoS solutions, that are the building blocks for any QoS architecture. This is followed by a more detailed discussion of the main solutions that have emerged in the last few years enabling QoS control in the Internet (Chapter 4). The architectures and mechanisms developed in these technologies address two key QoS issues in the Internet: resource allocation and performance optimization. In chapter 5 I give an application oriented point of view on QoS, with details about QoS parameters and application specific QoS requirements. Chapter 6 presents some experimental results obtained by testing from a QoS perspective a switch with a custom-built network tester. The report ends with a set of general conclusions, a glossary of QoS terms and references.

2 Network QoS

Regardless of the size and scope of an IP network, the observed end-to-end quality of service (QoS) is built from the concatenation of edge-to-edge QoS provided by each domain through which the traffic passes. Ultimately, the end-to-end QoS depends on the QoS characteristics of the individual hops along any given route. For example, the QoS experienced by an intra-LAN phone application depends solely on the LAN, whereas a wide area phone application experiences QoS that depends on the LANs at either end, the Internet service providers (ISPs) at either end, and the IP backbone in the middle.

Not surprisingly, much of the unpredictable and undifferentiated packet loss and jitter in today's IP services is due to the manner in which traditional best-effort routers cope with transient internal congestion. If a particular output port becomes the focal point for two or more inbound aggregate traffic streams, a best-effort router simply uses first in, first out (FIFO) queuing of packets destined for transmission on the associated outbound link. Queuing introduces latency (delay) and the potential for packet loss if a queue overflows. When traffic patterns are bursty, the queuing-induced latency varies unpredictably from packet to packet, manifesting itself as jitter in the affected traffic streams.

IP networks (enterprise, access and backbone) are being called upon to carry traffic belonging to a growing variety of customers with diverse requirements — for example, IP telephony, IP virtual private networks (VPNs), bulk data transfer, and mission-critical e-commerce. Each customer makes unique demands for some level of service predictability, even in the presence of transient congestion due to other traffic traversing the network. The demand for relative or absolute protection from other traffic on any particular network segment applies equally well to a high-speed LAN, a network based on T1 or E1 private links, a dial-up or ISDN access network, or a high-capacity backbone running at OC-48/STM-16 rates or higher etc. This demand leads directly to three technical requirements:

a) *Per-hop QoS* — The smallest controllable element in the network is the node (router or switch) joining two or more links. These nodes must be based on an architecture that allows sufficient differentiated queuing and scheduling to be applied at each hop and be able to appropriately utilize the QoS characteristics of inter-node links.

b) *Signaling and provisioning* — Controllable per-hop QoS and non-shortest path forwarding is of little use if it's not easily manageable. A practical solution requires some degree of automated distribution of QoS parameters and/or traffic engineering constraints to all the nodes (routers or switches) in the network. New information is distributed whenever a

customer imposes or changes specific end-to-end (or edge-to-edge) QoS requirements.

c) *Routing and traffic engineering* — Where multiple parallel paths exist through a network, distributing traffic across these paths can reduce the average load and burstiness along any given path. This practice improves the network's apparent service quality because each router is less likely to drop or jitter packets. Mechanisms for discovering and imposing non-shortest path forwarding are required.

These requirements are explored in more depth in the rest of this chapter.

2.1 Networking basics

Any network is built from a hierarchy of components. Any path from one point to another is usually formed from the concatenation of shorter paths (hops) at the same level. A path at some level N becomes one hop in a path at level N+1.

For example, the IP layer is made up of routers acting as switching points for IP packets and links that carry IP packets between routers. Each link is a single IP hop, yet the link itself might be made up of a number of its own hops and nodes. The link can be a single Ethernet, a segment of a bridged Ethernet network, an IP tunnel or an asynchronous transfer mode (ATM) virtual connection. In the case of a bridged Ethernet, one or more Ethernet switches may exist between the two routers. IP tunnels use one IP network to act as a link for another IP network (or sometimes the same IP network when certain types of traffic need to be hidden from sections of the network). An ATM virtual connection (VC) provides an end-to-end service between the ends of the VC, but in reality the VC may pass through many ATM switches along the way. The IP-level QoS between two points depends on both the routers along the path and the QoS characteristics of each link's technology. Clearly the inter-router packet transport builds on the QoS capabilities of each link. If the link technology has no controllable QoS, the routers can do little to compensate because they rely on each link to provide predictable inter-router connectivity. However, in the presence of QoS-enabled link technologies, the router's behavior makes or breaks the availability of IP-level QoS. All this are crucial elements in high speed networks [Cha-01].

Layering is fundamentally recursive. For example, the QoS characteristics of an ATM VC depend on the predictability of the inter-switch links as much as on the ATM switches themselves. An ATM VC may span

multiple ATM switches using SONET/SDH* circuits for inter-switch cell transport. The SONET/SDH circuit itself is made up of one or more hops through various rings and multiplexers. Finally, the SONET/SDH circuits may have been multiplexed onto a single fiber along with totally unrelated circuits using different optical wavelengths — this is done with wavelength division multiplexing (WDM), an optical fiber multiplication technology that allows lots of virtual fibers to be provisioned within a single physical segment of fiber. The Internet adds an extra wrinkle on the preceding model because many of the end-to-end paths used are not contained entirely within a single IP network. They are quite likely to span a number of independently administered IP networks, each with its own routing policies and QoS characteristics.

When only best-effort is required or expected, you don't really need to care about the intermediate networks along the path, as long as their routing policy allows them to forward traffic. However, to support end-to-end QoS, you need to know more about the network's dynamic behavior. You do not need to know how each network achieves its QoS goals. It is enough to simply characterize each network in terms of the latency, jitter and packet loss probabilities that may be imposed on the traffic, as well as its bandwidth. Because one person's network is another person's link, the notion of end-to-end QoS is based on that of edge-to-edge QoS. The QoS achieved from one end of a network to another is dependent on the concatenation of networks with their own edge-to-edge QoS capabilities, and each of these network's internal paths is built from links that may be networks in their own rights, again characterized by specific edge-to-edge QoS capabilities. The ability to control a network's edge-to-edge QoS behavior depends on the ability to control both the link and node behaviors at the network level.

2.2 Predictable per-hop behavior

The goal in a QoS-enabled environment is to make possible predictable service delivery to certain classes or types of traffic regardless of what other traffic is flowing through the network at any given time. An alternative expression of this goal is the process of aiming to create a multiservice IP network solution where traditional bursty traffic may share the same infrastructure (routers, switches and links) as traffic with more rigorous

* SONET and SDH are a set of standards for synchronous data transmission over fiber optic networks. SONET is short for Synchronous Optical NETwork and SDH is an acronym for Synchronous Digital Hierarchy; it is the United States version of the standard published by the American National Standards Institute (ANSI). SDH is the international version of the standard published by the International Telecommunications Union (ITU).

latency, jitter, bandwidth and/or packet loss requirements. Regardless of whether one focuses on enterprise, access or backbone networks (or some combination of them all), the end-to-end path followed by a single user's packets is merely a sequence of links and routers. So, our attention must initially be drawn to the dynamics of a router's forwarding behavior. Although a traditional router chiefly focuses on where to send packets (making forwarding decisions based on the destination address in each packet and locally held forwarding table information), routers for QoS-enabled IP networks must enable control of when to sends packets. We need to look more closely at those elements of a router that affect when packets are actually forwarded.

2.2.1 Basic QoS parameters

Each router is the smallest controllable convergence and divergence point for tens, hundreds and thousands of unrelated flows of packets. In most data networks, traffic arrives in fluctuating bursts. On regular occasions, the simultaneous arrival of packet bursts from multiple links, which are all destined for the same output link (itself having only finite capacity), leave a router with more packets than it can immediately deliver. For example, traffic converging from multiple 100Mbit per second Ethernet links might easily exceed the capacity of a 155Mbit per second OC-3/STM-1 wide area circuit, or traffic from a number of T3/E3 links (45Mbps/34Mbps) may simultaneously require forwarding out along a much smaller T1/E1 link (1.54Mbps/2Mbps). To cope with such occasions, all routers incorporate internal buffers (queues) within which they store excess packets until they can be sent onwards. Under these circumstances packets attempting to pass through the router experience additional delays. Such a router is said to be suffering from transient congestion. The end-to-end latency experienced by a packet is a combination of the transmission delays across each link and the processing delays experienced within each router. The delay contributed by link technologies such as SONET/SDH circuits, "leased line" circuits, or Constant Bit Rate (CBR) ATM virtual circuits is fairly predictable by design. However, the delay contributed by each router's congestion-induced buffering is not so predictable. It fluctuates with the changing congestion patterns, often varying from one moment to the next even for packets heading to the same destination. This randomly fluctuating component of the end-to-end latency is commonly referred to as jitter. Jitter can be assimilated to variance of delay, so we will sometimes refer only to delay.

Another issue is packet loss. Given that routers have only finite buffering (queuing) capacity, a sustained period of congestion may cause the buffer(s) to reach their capacity. When packets arrive to find buffer space

exhausted, packets must be discarded until buffer space becomes available. The traditional router has, effectively, only a single queue for each internal congestion point and no mechanism to isolate different classes or types of traffic from the effects of other traffic passing through it. The lack of predictiveness of the unrelated traffic passing through the shared queue at each internal congestion point is likely to have a heavy influence on each traffic stream's latency, jitter and packet loss. Some types of traffic (for example, TCP connections carrying e-mail) tolerate latency better than they tolerate packet loss, suggesting that long queues are ideal. However, other types of traffic — for example, User Datagram Protocol (UDP) carrying streaming video or audio — prefer that packets be discarded if held too long by the network, suggesting that shorter queues are better.

Any packet arriving and finding the queue partially full experiences additional latency because the packet must wait for the queue to drain the preceding packets. If a packet arrives when the queue is full, the packet has nowhere to go and is dropped. Jitter comes from the fact that the components of traffic are bursty and not correlated. In a typical IP environment packets are not of fixed length, adding further variability to the relationship between output link rate, the number of queued packets, and the latency experienced by these packets. Latency can also be a function of the subnet technology — for example, the back-off scheme of Ethernet. However, back-off on Ethernet simply reveals itself as temporal unpredictability of the link.

Another QoS parameter is throughput. In any network there is a relationship between loss rate, throughput and delay [Dav-99]. Unlimited buffer capacity would allow for the possibility of a zero loss rate; in that situation throughput alone would determine delay. However infinite buffers are only a theoretical notion (and would potentially cause an infinite delay). In any finite buffered system packet loss will occur occasionally. In general, allowing for more loss and delay in a traffic system will increase the amount of traffic that can be sent. For fixed loss rate, reducing delay implies that throughput must fall. For fixed throughput, reducing delay implies an increase in loss rate. For fixed delay, reducing loss rate will reduce available throughput. The loss rate/throughput/delay relationship also gives information for long-term network management, as it shows when loss and delay requirements cannot be met. Currently, network provision and measurement are often based only on bandwidth requirements. An important insight is given by the recognition of the fact that loss rates must be considered and to view the control of loss rates as an important part of the decision process.

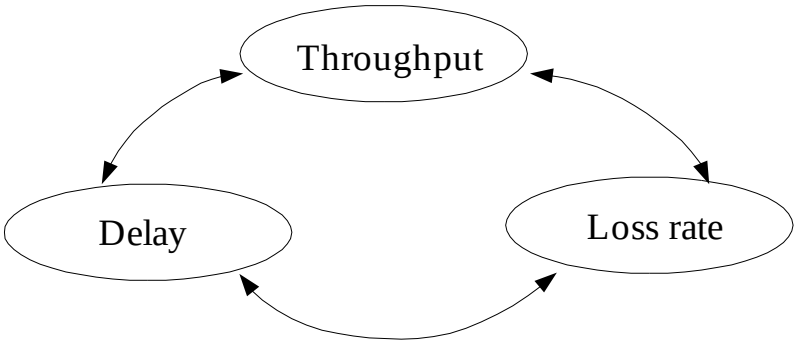


Figure 1 — The two degrees of freedom of the throughput, delay and loss rate relationship.

2.2.2 Classification, Queuing and Scheduling

So what must be improved? The delay, throughput and packet loss characteristics of any given IP network are ultimately determined by the QoS characteristics of links and the dynamics of queue utilization and queue management within each router. If network load exceeds service rate, a single queue at each internal congestion point is no longer sufficient. Instead, you need a queue for each identifiable class of traffic for which independent delay, throughput and packet loss characteristics are required. Each of these queues should have its own packet discard policies (for example, different thresholds beyond which packets are randomly or definitely discarded). Of course, the multiple queues per output interface are useless without a mechanism for assigning packets to the correct queues. A classification method is required over and above the router's traditional next-hop forwarding lookup. Finally, the queues must all share the finite capacity of the output link they feed into. This requirement implies the addition of a scheduling mechanism to interleave packets from each queue and, thus, mediate link access in a controllable and predictable manner. The preceding requirements can be captured as a statement that QoS-enabled networks require routers that can differentially Classify, Queue and Schedule (CQS) all types of traffic as needed [Arm-00]. Such routers will be said to have a CQS architecture.

2.2.3 Link-level QoS

Sometimes a router's scheduler must do more than simply interleave traffic at the IP packet level. The scheduler's capability to smoothly interleave traffic belonging to different queues depends on how quickly the outbound link can transmit each packet. For high-speed links (such as 155Mbit per second in OC-3/STM-1 circuits) a 1500-byte IP packet takes less than 80 μs to transmit. This allows the scheduler to divide the link's bandwidth into slots

up to 80 μs long — a very reasonable number, which drops to 20 μs on 622Mbit per second (OC-12/STM-4) circuits. However, at the edges of the Internet many links are operating at 1Mbit per second or slower, in the 56 to 128Kbit per second range for Integrated Services Digital Network (ISDN) in North America and Europe and down to 28.8Kbit per second in the case of many dial-up modem connections.

A 1500-byte IP packet takes around 94 ms to transmit over a 128Kbit per second link, blocking the link completely during this time. Regardless of whether jitter-sensitive traffic has been classified into a different queue, those packets experience a 94 ms jitter when the scheduler pulls the 1500-byte packet from another queue. Clearly, this poses some problems if QoS-sensitive applications are to be supported on the far side of typical low-speed access links. The basic solution is to perform additional segmentation of the IP packets at the link level in a manner transparent to the IP layer itself. The CQS architecture is then applied at the link level by queuing segments rather than whole packets, thus allowing the scheduler to interleave on segment boundaries. By choosing the smaller segment size appropriately, such an approach enables jitter-sensitive IP traffic to avoid being queued behind long IP packets. (However, nothing is gained for free — segmentation decreases overall transmission efficiency because each segment carries its own header to allow later re-combination of segments.)

Although ATM was originally designed for high-speed links, its design reflects a similar concern with minimizing the interval over which traffic on a given class could hold the link. The ATM cell is short by design, and each ATM switch is an example of a CQS architecture. Arriving cells are queued for transmission according to the contents of their Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) header fields. Taken together the VPI/VCI identify the VC to which the cell belongs, encoding both path information (where should the cell go next) and service-class information. Good ATM switches have queues for each traffic class on a per-port basis and have schedulers feeding cells out each port in accordance with the bandwidth guarantees given to each class.

CQS router architectures (see Figure 2) may be implemented in a number of permutations, each with its own specific consequences for the QoS characteristics of the IP network as a whole. The fundamental task of each router hop now becomes:

- To know where to send the packet (conventional forwarding);
- To know when to send it (the additional QoS requirement);
- To complete the preceding tasks independently of other traffic sharing the router.

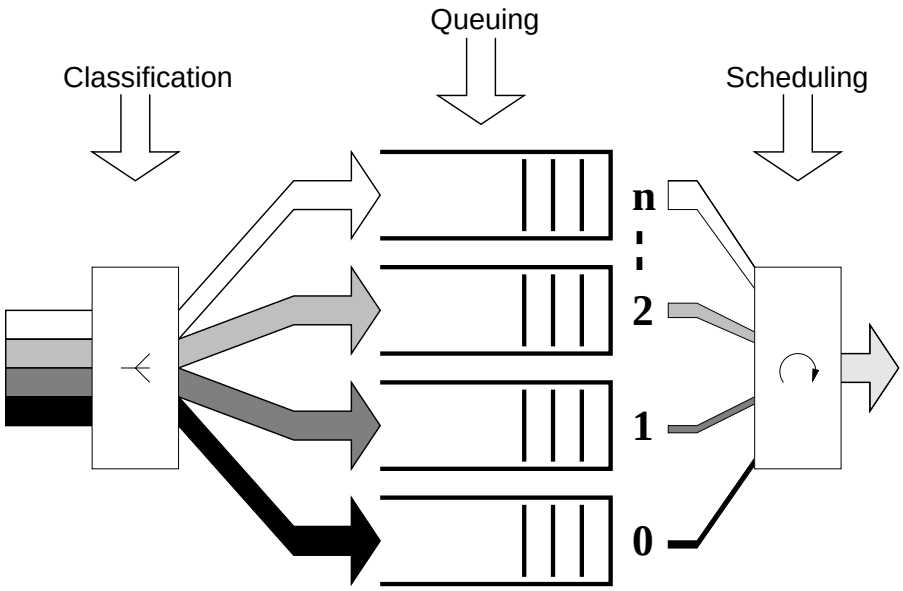


Figure 2 — The CQS architecture of a router.

2.3 Predictable edge-to-edge behavior

As noted earlier, any end-to-end service is constructed from both the concatenation and layering of edge-to-edge and per-hop behaviors. Network operators, focusing on the edge-to-edge capabilities of the networks under their control, have a range of possible per-hop behaviors to mix and match together. Over the years a number of solution spaces have emerged, each one reflecting a different set of assumptions and compromises with respect to the CQS and routing capabilities of routers within the network.

The first and most important observation is that network designers face a trade-off between the number of traffic classes carried by their networks and the number of traffic classes that their router's CQS architectures can handle. A number of solutions are based on distributed edge-and-core architectures, where the cores are fast routers with limited CQS capabilities and the edges are slower but with more advanced CQS capabilities. A second observation is that the Internet's existing shortest-path routing algorithms are not necessarily optimal for different classes of traffic across an arbitrary mesh of routers and links. A single metric may not be appropriate for all traffic traversing a particular section of the network. In addition, the destination-based forwarding paradigm itself makes it difficult to force subsets of available traffic into following alternative, non-shortest paths across any given network topology.

2.3.1 Edge-and-core models

Whether in hardware or software, the design of a good CQS architecture is generally nontrivial. In many software implementations, tight processing budgets make classification, queue management and scheduling difficult to introduce without affecting the overall peak performance of the box. Hardware implementations have only just started to become commonplace and until recently the development of a CQS implementation for IP locked into hardware was too commercially risky. The edge-and-core model allows core routers to leverage hardware implementations (for speed), while leaving complex (but slower) processing to software-based edge routers. The edge routers might be able to classify and independently queue hundreds or thousands of traffic classes, whereas the core routers are assumed to be limited to a handful of queues.

Limited numbers of queues in core routers leads to a new requirement that edge routers be able to smooth out the burstiness of traffic entering the network. In the preceding discussion of per-hop QoS control, individual traffic classes were permitted to be completely unpredictable on the assumption that you could accurately isolate and reschedule them at every potential congestion point. However, although a smart edge-dumb core model may have the requisite isolation granularity at the edges, it does not in the core. Multiple traffic classes will find themselves aggregated into shared queues within the core routers. The potential for unpredictable mutual interference is high unless the network imposes some level of predictability before the traffic reaches the core routers. The solution is for edge routers to manipulate the temporal characteristics of individual traffic classes (and, hence, the aggregate of those traffic classes) before they enter the core. The Internet Engineering Task Force (IETF) Differentiated Services model is one example and it will be discussed in Chapter 4.

Shaping and policing

The primary focus of a CQS architecture is the protection of traffic in each queue from the burstiness of traffic in another queue. On a per-hop basis, it is clear that, given appropriate isolation of all QoS sensitive traffic into distinct queues, a scheduler needs to guarantee only a certain worst-case servicing interval (or minimum bandwidth). If spare capacity is available, you might expect "good" scheduler behavior to allocate that capacity to any queues having packets waiting to be forwarded. However, this practice is not always desirable from a network-wide perspective.

Simply emptying a queue as fast as the line rate allows (in the absence of traffic in other queues) can increase the burstiness perceived by routers further downstream. As a result, a serious problem can develop if the downstream routers do not differentiate traffic with as much granularity as

the local router does. In addition, service providers may wish to limit the maximum rate that a customer can send through the network. If the customer frequently gets significantly better bandwidth than the guaranteed minimum (perhaps because the network is new and/or under-loaded), a perception issue surfaces: the customer begins to associate the typical performance with what he or she is paying for. If the spare capacity ever shrinks, the customer will receive edge-to-edge performance closer to the guaranteed minimum. However, the customer simply perceives the service to have degraded and is likely to complain. Managing customer expectations is an important part of running a business and in this case preemptive rate control is one of the technology-based tools that may be employed. Placing an upper bound on the maximum bandwidth (or minimum inter-packet interval) available to a traffic class is known as *traffic shaping*. A shaping scheduler is configured to provide both a minimum service interval (the time between pulling packets from the same queue) and a maximum service interval (to guarantee the latency bound or minimum bandwidth). Packets arriving with a shorter inter-packet interval than allowed by the scheduler are queued until transmission, thus smoothing out the original burstiness. A simple form of shaping scheduler is sometimes referred to as a "leaky bucket", because no matter how fast packets arrive they can only "leak out" at a fixed rate. Although the most obvious solution is to send equally spaced packets, similar to CBR in ATM, research in [Dav-99] shows that using a Poisson shaped traffic model can make scheduling a lot simpler for downstream routers.

Shaping is not a simple function to introduce into a best-effort router because this function presumes the existence of an appropriate CQS architecture. Although not quite so elegant, an alternative solution has been to introduce a packet-dropping behavior that is sensitive to excess burstiness of a traffic class. When too many packets arrive in too short an interval, packets are simply dropped. This process is known as policing. Policing can be implemented without queues or schedulers, although it typically needs some form of classification to differentiate between the policing rules imposed on different traffic classes. In its simplest form, each traffic class has an associated counter. The counter is incremented regularly every T seconds and decremented whenever a packet (belonging to the counter's class) is forwarded. If a packet arrives to be transmitted when the counter is zero, the packet is dropped instead. When no packets are being transmitted, the counter increments up to a fixed limit L. The net effect is that a packet stream arriving with an average inter-packet interval of T seconds (or greater) passes through untouched. However, if a burst of more than L packets arrive in less than T seconds, the counter reaches zero and extra packets are dropped. The value of L affects the burst tolerance of the policing function, and T sets the rate below which traffic is safe. This

practice is a severe, yet effective, way to modify the burstiness of traffic downstream from the policing router. The utility of policing is based on the assumption that most bursty traffic originates from applications using adaptive end-to-end transport protocols such as TCP. Packet loss is assumed to indicate transient congestion, and TCP reacts by slowing down the rate at which it injects packets into the network. Policing allows the network operator to fake the existence of transient congestion for a particular traffic class before it actually begins to occur further along the packet's path. Even if the traffic class is not using an adaptive end-to-end transport protocol, policing protects the rest of the network by continuing to drop packets that exceed the allowed parameters. Both shaping and policing are extremely useful tools for network designers who face a trade-off between the number of traffic classes carried by their networks and the number of traffic classes their network's CQS architectures can handle. The basic issue is that individual traffic classes can be permitted to be unpredictable only if you can accurately isolate them at every potential congestion point. If you lack that isolation capability, you must attempt to impose some level of predictability prior to the potential congestion point. In the smart edge/dumb core model, the solution is for each edge router to preemptively shape and/or police the individual traffic classes before they enter the core to impose some overall order, smoothness and predictability within each traffic class (and, hence, the aggregate of those traffic classes). Shaping may also be useful on the egress from a network in situations where the next network's aggressive policing would be otherwise detrimental.

Marking and reordering

Although shaping can be a sophisticated solution to smoothing out bursty traffic, simple policing is a blunt instrument. A number of variations have been introduced to soften the effect of edge-router policing. A policing node may choose to only mark packets (rather than discarding them immediately) if they exceed a burstiness threshold. Routers further along the path recognize these marked packets as having a lower priority than unmarked packets. If transient congestion begins to fill the queues in a downstream core router, its queue management algorithm can begin dropping marked packets before it begins dropping unmarked packets. Alternatively, the policing node might implement multiple levels of allowed average packet arrival rates — a lower rate below which packets are forwarded unmarked, an intermediate range of rates within which packets are marked and forwarded, and an upper threshold above which packets are dropped.

The impact on the core of the network is softer than would be achieved by a simple policing because many of the packets in the burst will

have been marked instead of dropped. The advantage of such a scheme is that, in the absence of other network congestion in the core, this particular traffic class can utilize more of the available bandwidth.

Many algorithms can be devised to provide multiple marking levels and threshold calculations. However, network designers who plan on using edge marking of traffic also need to carefully choose their core routers. The main point of concern is potential reordering of marked packets relative to unmarked packets within a traffic class. This situation can happen if the core router uses two separate queues to differentiate between marked and unmarked packets in the same traffic class. Because marked packets are of lower priority, an implementation might choose to effect this relative priority by assigning more scheduler bandwidth to the queue of unmarked packets than for the queue of marked packets.

As a consequence, a marked packet arriving before an unmarked packet in the same traffic class may find itself scheduled for transmission after the unmarked packet (or vice-versa). Assuming the marked packet makes it all the way to the other end, the receiving application perceives the traffic to contain out-of-order packets. Although the IP specifications do not preclude packets being reordered by the network, this practice should be avoided because most end-to-end protocols do not handle this case efficiently. In networks where marking is intended to increase a packet's drop probability, the solution is not too difficult. Let the core router initially ignore the policing marker when classifying packets into queues, ensuring all packets in a traffic class are placed in one queue regardless of drop priority. Then modify the packet drop threshold for that queue on the basis of whether the packet is marked or not. The core router's packet-dropping algorithm, thus, activates more aggressively for marked packets, achieving the desired edge-to-edge behavior.

2.3.2 Edge-to-edge routing

No particular restrictions dictate how routers and links are interconnected to form an IP network. The Internet's shortest-path routing mechanisms are based on the assumption that a network's topology is rarely static and must be tracked dynamically. In any realistic network, each router may have more than one output interface over which it could send a packet; the role of routing protocols is to establish a single interface over which a packet should be sent. To make the calculations tractable, the choice of the appropriate interface has largely been driven by algorithms using only a single metric to define the shortest path. However, two general concerns have been raised with this approach when it comes to supporting QoS. First is the argument that a single metric may not be appropriate for all traffic

traversing a particular section of the network. Second, the destination-based forwarding paradigm itself makes it difficult to force subsets of available traffic into following alternative, non-shortest paths across any given network topology.

QoS−based routing

QoS-based routing protocols attempt to take multiple metrics into account when building the network's forwarding tables. These protocols have been studied for years and often begin with an assumption that the network is built from conventional best-effort IP routers. Starting from this assumption, single-metric routing is seen to have a number of limitations when attempting to meet the mixed QoS demands of a multiservice environment. A metric can be considered a type of cost, with each link (hop) having a cost associated with it. The routing protocols attempt to find paths with minimal total cost summed over all the links to possible destinations. However, this cost cannot represent the interests and needs of all traffic types. Should it represent the link's latency, its available bandwidth, its packet loss probability, or perhaps the actual expense of sending packets over the link? For some traffic the choice is appropriate, whereas for other traffic the choice is wasteful of resources.

For example, consider a network where latency is the metric. Certainly the shortest path now suits applications with tight real-time requirements. But they are not alone. The network is most likely also being used by traditional, bursty data applications that care significantly less about latency. The traffic from these other applications also follows the minimum latency shortest paths, adding to the load on the best-effort routers along the path. An unfortunate side effect is that the bursty traffic consumes the same buffer space being used by the real-time traffic, increasing the jitter and average latencies experienced by all traffic through the routers. This approach also affects the accuracy of the latency costs that the routing protocols use to determine the shortest paths.

QoS-based routing creates multiple shortest-paths trees, covering the same actual topology of routers and links with each tree using different combinations of parameters as link metrics. The goal is to minimize unnecessary coexistence within routers of traffic with widely different QoS requirements. Packets with strict latency requirements are then forwarded by using the tree built with latency as a metric. Packets without real-time requirements might have a different tree built (for example, to minimize the financial cost of the path). Several practical issues exist with implementation of QoS based routing:

a) Each router needs to have multiple forwarding tables (or their functional equivalent) on which to perform each packet's destination-based

next-hop lookup, one for each type of shortest-path tree. Additional fields in the packet header are used to select one of the possible next hops associated with the packet's destination address. This situation complicates the design of the next-hop lookup engine.

b) An increase in routing protocol overhead occurs because the router's CPU must support an instance of each protocol for each unique shortest-path tree. This requirement causes an increase in the time it takes for a network of such routers to converge after a transient event in the network topology (for example, when links come or go or their costs somehow change). The convergence time increases further if the routing protocol is being asked to calculate trees based on multiple metrics simultaneously.

c) Metrics such as latency or available bandwidth are highly dependent on the actual traffic flowing across the network. A shortest-path tree built with statically configured latency values could become outdated when traffic begins to flow across the network. The alternative, of updating each link's cost with regular real-time measurements, poses a real control-theory problem: every cost update would result in a recalculation of the associated shortest-path tree, leading to continual processing load on all the routers.

Interestingly, the development of routers with CQS architectures somewhat reduces the need for QoS-based routing. For example, consider the example that uses latency as a cost metric. Now consider that every router has at least two queues per output interface, one for latency sensitive traffic and the other for all remaining traffic. All traffic is routed along lowest-latency paths. Assuming the routers appropriately classify traffic into the two queues, the service received by latency-sensitive traffic is independent of the burstiness of all other traffic types. Arguably then, any conventional, single-metric IP routing protocol, when coupled with routers based on a CQS architecture, can support multiple levels of service differentiation. The main not-necessarily-true assumption here is that sufficient capacity exists along the single tree to provide adequate service to all participants.

Explicit path control

The internal topologies of many networks are such that multiple paths can be found between most points. A major limitation of conventional IP forwarding is that single-metric, shortest-path trees use only one of the possible paths toward any given destination. Because lightly loaded alternative paths are not utilized, routers that exist on the shortest-path trees for many different network destinations can be subjected to high-average load — they become hot spots, potentially limiting the capability of the network to provide adequate service differentiation even if the router itself has a CQS architecture. As the average load on a hot-spot router rises,

the probability of random packet losses and jitter increases. Although this observation is most evident for networks containing regular best-effort routers, it also holds true (albeit to a lesser degree) when the network consists of multiple queue routers. To combat this problem, a network operator has two alternatives:

- Upgrade the routers and links to operate faster, which may require relatively large investments and is generally only a short term solution;
- Utilize additional packet-forwarding mechanisms that allow the traffic to be split across alternative paths (some of which may be just as short as the "official" shortest path and others that may be longer according to the prevailing metric).

When the network itself is built from cheap, low- to middle-bandwidth technology, the former approach may be entirely suitable. This description is most likely going to apply to enterprise environments where traffic growth has outpaced the deployed technology and a successor technology is easily deployable (for example, a 10Mbit per second Ethernet environment, where the upgrade to 100Mbit per second or 1Gbit per second Ethernet solutions are available). Simply upgrading equipment and/or links may not be an option when your network is already pushing the limits of available technology. High-performance IP backbones have this problem; their routers are usually pushed hard to support OC-12 and OC-48 rate interfaces, and to buy or provision such circuits across today's traditional carrier infrastructures presents a serious problem. In addition, although prices are dropping for OC-12 and OC-48 circuits, they remain an expensive resource. A preferable alternative is to build the equivalent aggregate capacity through parallelism — an IP topology rich in routers and lower-speed links across which the aggregate load can be distributed. Overriding shortest-path routing to more optimally utilize the underlying infrastructure of routers and links is often referred to as traffic engineering.

Traffic engineering through explicit path control is an important part of any solution to providing QoS, although the main impact is on the overall efficiency of the network itself, rather than directly impinging on the end-users. This approach also raises an interesting routing question: having discarded the information being provided by the existing IP routing protocols, network operators need to supply an external source of information to control the traffic-engineered routing within their networks.

Explicit path control can be achieved in a variety of ways, either avoiding or permuting every router's conventional destination-based forwarding decision. The methods available at the IP level include:

- Strict and loose source routing options;

- Forwarding tables with lookup on the destination address and other fields in the IP packet header;
- IP tunneling;
- Multiprotocol Label Switching (MPLS).

In theory, an IP packet can have optional header fields added that specify (either explicitly or approximately) the sequence of routers through which the packet must pass on its way to the destination. However, most routers do not efficiently process packets carrying such optional header fields (the peak performance "fast path" through a router is typically optimized for packets having no additional headers). Packets with optional headers are processed in a parallel "slow path", making this a poor choice if consistent QoS control is desired.

A slightly more feasible method is for the forwarding table to be constructed with regard not only for where the packet is going to but also for where it has come from. In this manner, it becomes possible to return different next-hop information for the same destination address just by taking the source address into account. However, this approach works only for a very constrained set of topologies and traffic engineering scenarios. It is also expensive in terms of memory space in the forwarding tables.

IP-IP tunneling forces the desired traffic patterns through the use of logical links. An IP packet is tunneled by placing it into the payload of another IP packet, which is then transmitted toward the desired tunnel endpoint. When the tunneling packet reaches its destination, the tunnel endpoint extracts the original IP packet and forwards it as though it had arrived over a regular interface.

Several problems exist with this solution:

a) Routers do not necessarily perform tunneling encapsulation and decapsulation in their "fast path"; this can be a major performance hit at the tunnel endpoints.

b) The tunneling encapsulation adds overhead to each packet, reducing the Maximum Transmission Unit (MTU) that can be supported by the virtual link represented by the tunnel if fragmentation within the tunnel is to be avoided.

c) The effective traffic engineering is very coarse — an IP-IP tunnel only allows control over the tunneled packet's final destination. Multiprotocol Label Switching (MPLS) is discussed in some detail later (Chapter 4), but it is worth noting here that the primary role of MPLS for service providers is traffic engineering. MPLS is a connection-oriented form of IP networking; packets have labels added and are forwarded along preconstructed label-switched paths (LSPs) by routers modified to switch MPLS frames (label-

switching routers, LSRs). There are several advantages of MPLS over encapsulation. First, the overhead per packet is reduced (an MPLS header is 4 bytes, compared to 20+ bytes for a complete encapsulating IP header). Second, the packet's actual hop-by-hop path within the backbone is under the control of the network operator when the LSP is established.

LSP and ATM VC are similar in many ways. Backbone operators who use ATM to transport their wide area IP traffic already utilize explicitly routed permanent virtual connections (PVCs) between the edges of their ATM networks and rely on the edge routers to map the correct traffic onto the appropriate PVCs. For many service providers, the move to MPLS is simply a generalization of ATM, with variable-length packets instead of fixed-length cells.

2.4 Signaling

Assuming that you can provide differentiated queuing and scheduling on a per-hop basis and have the appropriately controllable underlying link layers, the question becomes one of establishing and modifying the network's actual behavior. This matter requires coordination of the actual (rather than theoretical) behaviors along each path. A generic term for this process is signaling — the act of informing each hop along a path (or paths) how to recognize traffic for which a special processing behavior is required and the type of special processing required. Signaling can be achieved in a number of ways with varying degrees of timeliness, flexibility and human intervention (not all of which are conventionally considered signaling *per se*).

At one extreme sits dynamic edge-to-edge signaling, where the network is informed each and every time a new class of traffic requires specific support. The network itself responds on demand by internally establishing additional information (or modifying existing information) at each hop to achieve the requested edge-to-edge behavior. Examples of on-demand signaling include ATM's User Network Interface (UNI) and Network Node Interface (NNI) signaling protocols and the IETF's Resource Reservation Protocol (RSVP).

New network technologies frequently either do not have fully dynamic signaling protocols defined or have not matured to the point where reliable implementations of their signaling protocols exist. Under such circumstances the networks are usually provisioned for new services, often entailing human intervention to configure (or reconfigure) the controllers of the links and nodes along the affected paths. Provisioning is a form of signaling, even though the response time is usually orders of magnitude slower than dynamic signaling.

Because the number of links and nodes in a network can be quite large, many vendors are developing centralized controllers or servers where configuration or provisioning actually occurs. These controllers then automatically distribute the appropriate rules to the links and nodes in the network on behalf of the human operator. Designs that are more advanced allow these controllers to automatically react to changing network conditions in accordance with general policies that may be imposed by a human operator. Although such centralized schemes also constitute a dynamic mechanism, they differ from edge-to-edge signaling in that it is not user controlled.

The Internet has used a form of dynamic signaling for years: its routing protocols. Although it is generally thought of signaling and routing as distinct activities, protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) are the Internet's mechanisms for signaling topology changes. These mechanisms ensure the construction of up-to-date forwarding tables that reflect the best set of shortest paths across the network and adapt dynamically to changing topological conditions. Therefore, these mechanisms (that is, OSPF and BGP) qualify as signaling protocols. However, their focus is internal to the network itself and their actions are generally not explicitly triggered by some user's request. Furthermore, their actions are designed primarily to effect the construction of paths, not the allocation of resources or priority processing for specific traffic along those paths.

Typically, signaling in the IP context is thought of as the additional actions required to establish a particular edge-to-edge QoS over and above the default best-effort QoS. As previously noted this process can involve dynamic or provisioned behaviors (or some combination). In all cases, the process of establishing a desired edge-to-edge QoS requires careful balancing of existing per-hop resources and network-wide paths. When a signaling request states a particular QoS goal, there are a number of variables to consider. In theory, both the path and the resources along the path are open to modification. For a given path, the signaling protocol should determine whether resources (for example, queuing space or share of link bandwidth) are available along that path. If the first path checked does not support the desired QoS, an ideal signaling protocol would find another path and try again. As an example, ATM's Private Network Node Interface (PNNI) signaling tries different paths until it finds one that can support the requested edge-to-edge QoS. Trying alternative paths presumes that the network has the capability to force traffic along the path that is discovered to be capable of supporting the desired QoS.

In conventional connectionless IP, however, traffic must follow the shortest-path trees established by the routing protocols (using whatever

metric is specified by the network operator). As a consequence the IETF developed its RSVP signaling mechanism to simply follow (and adapt to) whatever routing exists in the network without attempting to discover alternative, non-shortest-path routes that might better support the requested QoS. Inherently a trade-off, RSVP avoids any reengineering of existing IP networks, doesn't reinvent or replicate the actions of the existing IP routing protocols, and can be introduced as a simple hardware or software upgrade to existing routers. However, if resources are exhausted along a particular shortest path, no simple way exists for RSVP to force traffic along a longer, but perhaps more lightly loaded path. Another issue with signaling is the amount of additional state information the routers must carry. State information is anything that the router needs to characterize the special traffic (for example, IP header information on which to classify the packets) and to process the special traffic (for example, associated queues, packet-drop parameters, and scheduler priorities or weights). The routing and forwarding tables already held by each router represent topological state information; adding signaling for QoS only increases the number of tables consuming valuable memory in routers.

Any realistic QoS solution for IP networking must cope with the often conflicting demands that it be easy to implement, not use a too large amount of state information, optimize the use of network resources, adapt dynamically to routing changes and work in a world where routing and signaling are decoupled.

Classes of service

Class of Service (CoS) is a way of managing traffic in a network by grouping similar types of traffic (for example, e-mail, streaming video, voice, large document file transfer) together and treating each type as a class with its own level of service priority. So its function can be also assimilated to signaling. Class of Service technologies do not necessarily guarantee a level of service in terms of bandwidth and delivery time; they offer a "best-effort." On the other hand, CoS technology is simpler to manage and more scalable as a network grows in structure and traffic volume.

There are three main CoS technologies:

- IEEE 802.1p/Q;
- IP Precedence (Type of Service);
- Differentiated Services (DiffServ).

IEEE 802.1p specification defines three bits within the IEEE 802.1Q field (VLAN tag) of the layer 2 packet header to be used for specifying priority. This QoS approach lacks however a global view on connection characteristics, since Ethernet is generally used only in LANs.

Layer 3 QoS mechanisms provide end-to-end control over QoS settings. A first attempt of IETF to use the Type of Service (ToS) in the IPv4 header was [RFC-1349]. The most recent version is called IP precedence and is defined in [RFC-1812]. Three of the ToS bits are used to create 8 priority levels and four are used to signal sensitivity to delay, throughput and packet loss.

Differentiated Services (DS or DiffServ), has been developed in draft form by an Internet Engineering Task Force (IETF) Working Group. DiffServ uses a different approach to managing packets than simple priority labeling. It uses an indication of how a given packet is to be forwarded, known as the Per Hop Behavior (PHB). The PHB describes a particular service level in terms of bandwidth, queuing theory, and dropping (discarding the packet) decisions. DiffServ will be discussed in more detail in Section 4.2.2.

2.5 Policies, authentication and billing

If you are offered a first-class seat on the plane for the price of an economy-class seat, you would probably take it. At worst you might wonder what the catch is, but in the end nobody refuses better service if it costs no more than worse service, whether the comfort of an airplane's seating, the speed of package delivery from a shipping service, or Internet access from a local ISP is being discussed. Of course, a practical problem immediately arises if you're not being charged a premium price for the premium service: everybody else wants it, too.

Any networking technology that offers differentiation of service levels must also address the need to differentiate each user's right to use particular service levels. If everyone has a right to use the best-service level at the same time, the resources would either run out, or the network would have to be engineered to cope. In general the network's resources are limited at various service levels, and so the task is one of allowing or disallowing particular users access to service levels based on their right to use. (If the network were engineered to handle everyone asking for premium service, without any differential impact on the cost of running the network, what would be the point of offering lesser service levels?) This right to use can be established in a number of ways — for example, payment of fees (financial cost) or administrative assignment (ranking of the user's importance). A commercial service provider would be inclined to utilize a fee basis: you get the service you pay for. A corporate enterprise network may determine service allocations based at least in part on the status of each user (or the user's department) within the company.

The whole issue of establishing and monitoring a user's right to use

certain service levels forms a set of problems that the Internet industry is only beginning to address. First are questions of policy (identifying the service classes that particular users are entitled to negotiate). Second is the problem of authentication (proving that the entity currently using the network is the claimed user, either during right-to-use negotiations or subsequent traffic transmission). Third is the question of billing (extracting the fee from the correct user) if fees are used to establish the right to use. Billing is even of interest to enterprise networks, where it may be used to provide additional granularity of usage control beyond the corporate status of a user or the user's department.

All three issues are also tightly coupled to the network's signaling because the network's signaling system must establish the requested edge-to-edge service levels and associate them with traffic coming from the user. If the users are utilizing dynamic, edge-to-edge signaling to negotiate their right to use, the signaling protocol itself must be tightly coupled with the policy, authentication and billing mechanisms. Human nature being what it is, the network must be capable of authenticating any user's request for, and use of, particular service levels. Users must not be billed for services they don't request and, of course, must be accurately billed for the service levels they do request. If the operator's fee structure is based in some part on the actual amount of usage, the consumption of services must also be tracked and authenticated. If dynamic, edge-to-edge user-signaling protocols (such as RSVP) are to be used in fee-for-use environments, these protocols clearly need to incorporate sufficiently strong user authentication fields. (An operator might attempt to deduce a user's identity from physical attachment points on the network, but in an age of dial-up IP access and mobile nodes this approach is rarely effective.) In the absence of such capabilities, the user and service provider are forced to rely on more traditional or manual channels to negotiate service levels (the fax, phone or postal service). Alternatively, the service provider can simply hope users don't go around impersonating each other when ordering service levels.

Enterprise environments are typically more structured and controlled, and in these environments authentication based solely on the node's topological position might be quite feasible. However, if the enterprise network includes mobile nodes or any likelihood that users will move around the network's topology, it will need to consider the same issues faced by a commercial service provider.

Two problems develop if the service provider decides to incorporate a usage-based component in the right-to-use fee. First, no clear industry consensus has emerged on what constitutes a realistic metric for use: is it simple packet counts, burstiness, peak or mean bandwidths, or some complex measurement of delivered latency and jitter?

Second, after you decide on a metric that you think the customers will understand, you face the problem of accurately measuring it in your network and reliably associating your measurements to particular users. Real-time measurement of traffic patterns is a major problem because it requires significant processing capabilities and needs to be undertaken for each and every instance of a distinct, user-defined traffic class.

Understanding the roles played by policy management, user authentication and billing models as important components of a global IP QoS solution and having the ability to assess whatever the industry offers are key issues for anyone who wishes to implement and provide QoS.

3 Per-hop QoS solutions

Providing end-to-end network QoS requires two elements deployed together:

- per-hop QoS solutions that enable each node to cope with the end-to-end QoS requirements and enforce QoS management;
- edge-to-edge QoS solutions, which are basically signaling techniques that ensure a coherent handling of streams from an end-to-end perspective, based on the local per-hop QoS solutions (see Chapter 4).

Per-hop QoS solutions are implemented at node level, that is in routers. As it was already mentioned in Chapter 2, the three main functions of a QoS enabled router are classification, queuing and scheduling.

Routers classify packets to determine which flow they belong to, and to decide what service they should receive. Classification may, in general, be based on an arbitrary number of fields in the packet header. Performing classification quickly on an arbitrary number of fields is known to be difficult, and has poor worst-case performance. Although classification is a normal function of any router, classification performance may have a more important impact on QoS-enabled routers due to the increased complexity of classification criteria.

Once the packets are classified they must be placed into queues. Mechanisms related to queue memory management must be employed in order to decide if packets should be dropped. This is the way to control packet loss, so it has a significant influence on global QoS characteristics. Active queue memory management mechanisms also aim at actively controlling congestion, an important cause of QoS problems.

Scheduling is the mechanism by which the decision of forwarding packets is taken. This is a very important component of a QoS-enabled router, since this it is a stage where delay and throughput can be controlled. Queue scheduling disciplines, should try to schedule packets in such a way as to provide a differentiated treatment of traffic;

For each function there exist a plethora of algorithms, each having certain advantages and disadvantages. In addition implementations in real devices may drift from standard algorithms, usually in order to decrease complexity, which means that the characteristics and guarantees change with respect to theoretical results.

3.1 Classification

The most well-known form of packet classification is used to route IP datagrams. From our point of view this is not very important, since any router must provide this functionality. However, there are other network services that require packet classification, such as access-control in firewalls, policy-based routing, provision of differentiated QoS and traffic billings.

In each case it is necessary to determine which flow an arriving packet belongs to so as to determine what kind of treatment it should get. The categorization function is performed by a *flow classifier* (also called a packet classifier) which maintains a set of rules according to which classification is performed. This is done based on the contents of the packet header(s).

Here we shall only provide a taxonomy of classification algorithms, see for example [Gup-01] for more details about each of them.

1) Basic data structure algorithms:
   - linear search;
   - hierarchical tries;
   - set-pruning tries;
2) Geometric algorithms:
   - grid-of-tries;
   - cross-producting;
   - 2D classification schemes;
   - area-based quadtree (AQT);
   - fat-inverted segment (FIS) tree;
3) Heuristic algorithms:
   - recursive flow classification;
   - hierarchical intelligent cuttings;
   - tuple space search;
4) Hardware-based algorithms:
   - ternary CAMs (Content Addressable Memories);
   - bitmap intersection.

Based on this type of algorithms packets are classified and they can be placed into queues. QoS-enabled routers should have multiple queues for each priority class.

3.2 Queuing

Queue memory management controls the number of packets in a queue by determining when and which packets are dropped when a queue experiences congestion [RFC-2309]. Therefore it allows the control of service class access to a limited router resource: packet buffer memory. The most used queue memory management techniques are:

a) *Tail drop* — this is in fact equivalent to an absence of queue memory management. A packet arriving at the tail of a queue whose resources are completely exhausted is discarded. This method is very easy to implement, but it has very poor characteristics regarding burst absorption.

b) *Random Early Detection (RED)* — an active queue management technique currently deployed in large IP networks. RED uses a packet *drop profile* to control the aggressiveness of its packet discard process. This profile defines a range of drop probabilities across a range of queue occupancy states, the drop probability increasing generally with queue occupancy once a certain threshold has been exceeded. Configuration in order to achieve predictable performance can be difficult, but if properly configured good delay characteristics and TCP performance are obtained.

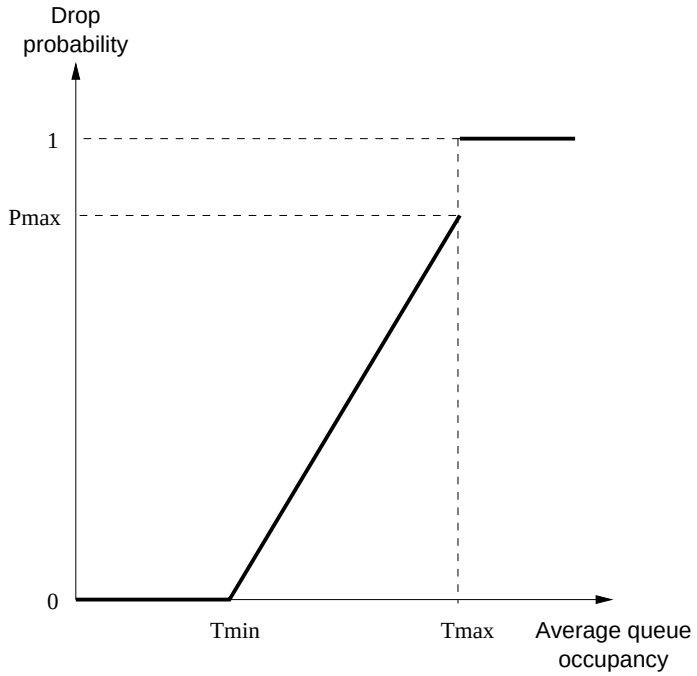


Figure 3 — Random Early Detection drop profile.

c) *Weighted Random Early Detection (WRED)* — an extension of RED that allows the assignment of different RED drop profiles for different types of traffic. In this way a finer granularity of control is provided.

d) *RED with* in *and* out *(RIO)* — a technique that uses two RED algorithms with different drop probabilities. One is used for packets that are *in* profile (for which traffic rate does not exceed the SLA bit rate) and the other for *out* profile (excess) packets. There are two thresholds for each queue in this case; as long as the queue size is below the first threshold no packets are dropped, but when the queue size is between the thresholds only *out* packets are randomly dropped. When the queue size exceeds the second threshold, indicating possible network congestion, then both *in* and *out* packets are randomly dropped, but *out* packets are dropped more aggressively. In DiffServ there exists a field which contains a bit to indicate the *in* and *out* packets.

e) *Explicit Congestion Notification (ECN)* — an experimental addition to the IP architecture designed to provide a different approach to active queue management. ECN responds to congestion by marking packets and relying on the destination host to send an explicit congestion notification to the source host. On reception of such a notification the source is expected to reduce its transmission rate.

3.3 Scheduling

Queue scheduling manages the amount of bandwidth allocated to each service class on an output port, thus controlling service class access to a limited network resource: link bandwidth. Here are the traditional queue scheduling disciplines that are most frequently encountered in real implementations:

a) *First-In First-Out queuing (FIFO)* — the most basic queue scheduling discipline, in which all packets are treated equally by placing them into a single queue and then servicing them in the same order that they were placed into the queue. It places an extremely low computational complexity load on the system, but it doesn't really allow for a differentiation between flows.

b) *Priority Queuing (PQ)* or Strict Priority (SP) — the basis for a class of scheduling algorithms that are designed to provide a relatively simple method of supporting differentiated service classes. Packets are first classified by the system then placed into different priority queues; they are scheduled from the head of a given queue (in FIFO order) only if all queues of higher priority are empty. The computational complexity is again relatively low, but an excessive volume of higher priority traffic may lead to complete resource starvation for lower priority traffic.

c) *Fair Queuing (FQ)* — the foundation of a class of queue scheduling disciplines designed to ensure that each flow has fair access to network

resources and to prevent bursty flows from consuming more than its fair share of output port bandwidth. Packets are first classified into flows, then assigned to a queue that is specifically dedicated to that flow. Queues are then serviced one packet at a time in round-robin order, empty queues being skipped. With FQ, bursty or misbehaving flows don't degrade the QoS delivered to other flows, but it cannot support flows with different bandwidth requirements.

d) *Weighted Round Robin (WRR) or Class-Based Queuing (CBQ)* — it was designed to address the limitations of FQ and PQ models. Packets are classified in various service classes then assigned to specifically dedicated queues which are serviced in round robin order. Allocation of different amounts of bandwidth is done either by allowing higher bandwidth queues to send more than a packet each time it is visited during a service round, or by visiting them multiples times in a single service round. WRR can be implemented in hardware, thus providing an efficient coarse control over the percentage of output port bandwidth allocated to each service class. The control is coarse due to the fact that the model is packet based and gives incorrect results for variable length packets.

e) *Weighted Fair Queuing (WFQ)* — a queuing discipline based on FQ that supports flows with different bandwidth requirements by giving each queue a weight that assigns it a different percentage of output port bandwidth. It supports variable length packets due to the bit-by-bit round robin approach. WFQ can provide strong upper-bound, end-to-end delay performance guarantees, at the expense of a relatively high computational complexity. This also sets a limit on the number of services classes that can be managed. Various enhancements have been proposed so that to reduce the computational complexity while keeping the accuracy as good as possible.

f) *Deficit Weighted Round Robin (DWRR)* — this is a class of disciplines that tries to overcome the limitations of the WRR and WFQ models. It accurately supports a weighted fair distribution of bandwidth when servicing queues that contain variable-length packets (unlike WRR). DWRR defines a scheduling discipline that has lower computational complexity than WFQ and can be implemented in hardware. It uses a *weight* to control bandwidth, a *deficit counter* to specify the maximum number of bytes that a queue is permitted to transmit each time it is visited by the scheduler. A *quantum* of service is defined (expressed in bytes), that is proportional to the weight of the queue and is used to increment the deficit counter.

4 Edge-to-edge QoS solutions

Internet Service Providers (ISP) are facing the challenge of offering improved QoS to their customers. Although access to virtually unlimited bandwidth via WDM (Wavelength Division Multiplexing) and Photonic Networks may potentially offer a solution to the QoS issue, access to such services on an universal basis is not a service class paradigm. Several organizations have proposed many service models and mechanisms to meet the demand for QoS. Further, a variety of classes have been proposed, which offer low delay and low jitter for applications such as Internet telephony and video teleconferencing, as well as a range of services for guaranteed (real-time), controlled-load (premium) and best-effort services [Hun-02].

4.1 Basic concepts

QoS support in IP networks can be traced back to the seminal Sigcomm92 paper on the Integrated Service Packet Network (ISPN) model by Clark, Shenker and Zhang [Cla-92]. Roughly, the ISPN model is built on four columns:

a) A QoS specification and requirements description, which could be seen as a description of a service level agreement that must be honored by a given QoS architecture.

b) Mechanisms for admission control or traffic conditioning when resources are finite and contention may arise.

c) Scheduling and other mechanisms to be in place at the network nodes to enforce preferential forwarding and processing of data packets. Network resources must be in place to assure the specified QoS.

d) Signaling and service interfaces to convey information on preferences and expectations (requirements) of data packet processing and forwarding from applications to relevant control elements of network resources and from there back to the applications.

Finally, a QoS architecture is needed that integrates all four columns into an end-to-end solution. At this point nothing is precluded in terms of the interpretation of such an end-to-end architecture. The requirement for an end-to-end QoS solution does not necessarily mean that a single resource reservation signaling protocol must be applied end-to-end. In fact, it is most likely that the end-to-end QoS management architecture will consist of many interoperable and concatenated QoS management architectures rather than one global end-to-end QoS infrastructure.

"Next Steps for the IP QoS Architecture" [RFC-2990] for example,

recognizes that, "both the Integrated Services architecture and the Differentiated Services architecture have some critical elements in terms of their current definition which appear to be acting as deterrents to widespread deployment [...] There appears to be no single comprehensive service environment that possesses both service accuracy and scaling properties". This statement sums up the reasons behind both the proposal of hybrid architectures composed of IntServ and DiffServ regions (with the associated problems related to mapping and interoperation procedures between different regions) and the necessity/opportunity of improving/ upgrading the IntServ and DiffServ paradigms.

Each column, however, may be realized differently. For example, resources may be claimed based on the granularity of flows or aggregates. Signaling would similarly reflect the right level of granularity and could be implicit or explicit. For example, RSVP [RFC-2205], one of most prominent signaling protocols has been adopted within the IntServ architecture for resource reservations. Admission control could be explicit by or implicit by overprovisioning and conditioning, etc.

In what follows a brief analysis is given of existing IP QoS solutions and the implied signaling issues. Where needed, we will also touch on related admission control or traffic conditioning issues. The main goal of this analysis is to understand whether the strict QoS requirements imposed on networks by future fixed and mobile applications are satisfied by the existing IP QoS solutions.

QoS solutions can be classified as being directed towards the control of resource allocation or performance optimization [Wan-01]. To each of this two directions we will dedicate a separate section.

Active resource allocation lacks from Internet. This concept is however important, since resources are finally those who determine the QoS characteristics of a link. The main existing edge-to-edge QoS solutions that are based on resource allocation are the following:

a) *Integrated Services* — uses an end-to-end per-flow resource reservation protocol (RSVP) that is applied in an end-to-end communication path.

b) *Differentiated Services* — uses a combination of edge policing, provisioning and traffic prioritization to make possible service differentiation.

c) *Integrated Services over Differentiated Services* — a framework that provides end-to-end QoS using the IntServ model over heterogeneous networks. Two main approaches are possible:

- Statically assigned trunk reservations based on Differentiated Services,

with several individual reservations being aggregated into a common reservation trunk that is statically configured.

- Dynamic trunk reservations with Aggregated RSVP, where several individual reservations are aggregated into a common reservation trunk. Additionally, these trunks are dynamically configured by using a signaling protocol that manages various mechanisms for dynamic creation of an aggregate reservation.

The solutions that lay the emphasis on performance optimization focus on changing the conventional IP routing architectures, by introducing additional capabilities in IP routing and new performance management tools. The main directions are multi-layer switching (e.g. multi-protocol label switching) and traffic engineering. This ideas will be further discussed in Section 4.3.

4.2 Resource allocation

Fundamentally, many problems we see in the Internet come down to the issue of resource allocation; packets get dropped or delayed because the resources in the network cannot meet all the traffic demands. A network, in its simplest form, consists of shared resources such as bandwidth and buffers, serving traffic from competing users. A network that supports QoS needs to take an active role in the resource allocation process and decides who should get the resources and how much.

The current Internet does not support globally any forms of active resource allocation. The network treats all individual packets exactly the same way and serves the packets on a first-come, first-served (FCFS) basis. There is no admission control either — users can inject packets into the network as fast as possible.

The Internet currently relies on the TCP protocol in the hosts to detect congestion in the network and reduce the transmission rates accordingly. TCP uses a window-based scheme for congestion control. The window corresponds to the amount of data in transit between the sender and the receiver. If a TCP source detects a lost packet, it slows the transmission rate reducing the window size by half and then increasing it gradually in case more bandwidth is available in the network.

TCP-based resource allocation requires all applications to use the same congestion control scheme. Although such cooperation is achievable within a small group, in a network as large as the Internet, it can be easily abused. For example, some people have tried to gain more than their fair share of the bandwidth by modifying the TCP stack or by opening multiple TCP

connections between the sender and receiver. Furthermore, many UDP-based applications do not support TCP-like congestion control and real-time applications typically cannot cope with large fluctuations in the transmission rate.

The service that the current Internet provides is often referred to as best-effort. Best-effort service represents the simplest type of service that a network can offer; it does not provide any form of resource assurance to traffic flows. When a link is congested, packets are simply dropped as the queue overflows.

Since the network treats all packets equally, any flows could get hit by the congestion. Although the best-effort service is adequate for some applications that can tolerate relatively large delay variation and packet losses, such as file transfer or e-mail, it clearly does not satisfy the needs of many new applications and their users. New architectures for resource allocation that support resource assurance and different levels of services are essential for the Internet to evolve into a multiservice network. Over the last decade the Internet community came up with Integrated Services and Differentiated Services, two new architectures for resource allocation in the Internet. The two architectures introduced a number of new concepts and primitives that are important to QoS support in the Internet:

- Frameworks for resource allocation that support resource assurance and service differentiation;
- New service models for the Internet in addition to the existing best-effort service;
- Languages for describing resource assurance and resource requirements;
- Mechanisms for enforcing resource allocation.

Integrated Services and Differentiated Services represent two different solutions. Integrated Services provide resource assurance through resource reservation for individual application flows, whereas Differentiated Services use a combination of edge policing, provisioning and traffic prioritization. Based on these two solutions some hybrid models have been created that address their limitations.

4.2.1 Integrated Services

Although the problems with the best-effort model have long been recognized, the real push for enhanced service architectures came in the early 1990s after some large-scale video conferencing experiments over the Internet. Real-time applications such as video conferencing are sensitive to the timeliness of data and so do not work well in the Internet, where the

latency is typically unpredictable. The stringent delay and jitter requirements of these applications require a new type of service that can provide some level of resource assurance to the applications.

In early 1990 the Internet Engineering Task Force (IETF) started the Integrated Services (IntServ) working group to standardize a new resource allocation architecture and new service models. At that time the World Wide Web, as we know it today, did not yet exist and multimedia conferencing was seen by many people as a potential killer application for the Internet. Thus the requirements of the real-time applications had major impacts on the architecture of Integrated Services. The Integrated Services architecture is based on per-flow resource reservation. To receive resource assurance, an application must make a reservation before it can transmit traffic onto the network.

Resource reservation involves several steps. First, the application must characterize its traffic source and the resource requirements. The network then uses a routing protocol to find a path based on the requested resources. Next, a reservation protocol is used to install the reservation state along that path. At each hop admission control checks whether sufficient resources are available to accept the new reservation. Once the reservation is established, the application can start to send traffic over the path for which it has exclusive use of the resources. Resource reservation is enforced by packet classification and scheduling mechanisms in the network elements, such as routers.

The Integrated Services working group proposed two new service models that a user can select: the guaranteed service and the controlled load service models. The guaranteed service model provides deterministic worst-case delay bound through strict admission control and fair queuing scheduling. This service was designed for applications that require absolute guarantees on delay. The other service model, the controlled load service, provides a less firm guarantee — a service that is close to a lightly loaded best-effort network. The Resource Reservation Protocol (RSVP) was also standardized for signaling an application's requirements to the network and for setting up resource reservation along the path. The Integrated Services model was the first attempt to enhance the Internet with QoS capabilities. The research and development efforts provided valuable insights into the complex issues of supporting QoS in the Internet. The resource allocation architecture, new service models and RSVP protocol were standardized in the late 1990s. But deployment of the Integrated Services architecture in the service provider's backbones has been rather slow for a number of reasons. For one, the Integrated Services architecture focused primarily on long-lasting and delay-sensitive applications. The World Wide Web, however, significantly changed the Internet landscape. Web-based applications now

dominate the Internet, and much of Web traffic is short-lived transactions. Although per-flow reservation makes sense for long-lasting sessions, such as video conferencing, it is not appropriate for Web traffic. The overheads for setting up a reservation for each session are simply too high. Concerns also arose about the scalability of the mechanisms for supporting Integrated Services. To support per-flow reservation, each node in a network has to implement per-flow classification and scheduling. These mechanisms may not be able to cope with a very large number of flows at high speeds. Resource reservation requires the support of accounting and settlement between different service providers. Since those who request reservation have to pay for the services, any reservations must be authorized, authenticated and accounted. At the present moment such supporting infrastructures simply do not exist in the Internet. When multiple service providers are involved in a reservation, they have to agree on the charges for carrying traffic from other service providers' customers and settle these charges among them. Most network service providers are currently connected through bilateral peering agreements. To extend these bilateral agreements to an Internet-wide settlement agreement is difficult given the large number of players. The Integrated Services architecture may become a viable framework for resource allocation in corporate networks. Corporate networks are typically limited in size and operated by a single administrative domain. Therefore many of the scaling and settlement issues we discussed above vanish. Integrated Services can support guaranteed bandwidth for IP telephony or video conferencing over corporate intranets. RSVP can also be used for resources allocation and admission control for traffic going out to wide-area networks.

The ideas, concepts, and mechanisms developed in Integrated Services also found their ways into later work on QoS. For example, controlled load service has influenced the development of Differentiated Services, and similar resource reservation capability has been incorporated into MPLS for bandwidth guarantees over traffic trunks in the backbones.

Resource reSerVation Protocol (RSVP)

An end-to-end per-flow resource reservation signaling protocol is applied in an end-to-end communication path, and it can be used by an application to make known and reserve its QoS requirements to all the network nodes in this path. This type of protocol is typically initiated by an application at the beginning of a communication session. A communication session is typically identified by the combination of the IP destination address, transport layer protocol type and the destination port number. The resources reserved by such a protocol for a certain communication session will be used for all packets belonging to that particular session. Therefore,

all resource reservation signaling packets will include details of the session to which they belong.

The end-to-end per-flow resource reservation signaling protocol most widely used today is based on the Integrated Service model and is called Resource reSerVation Protocol (RSVP) [RFC-2205] [RFC-2210]. RSVP was originally designed as a signaling protocol for applications to reserve network resources [RFC-2205]. It represents a fundamental change to existing Internet architecture where all flow-based state information exists in the end systems. The IntServ model comprises three classes of services [RFC-2211]:

- best-effort, for delay independent applications;
- guaranteed [RFC-2212], for applications requiring fixed delays;
- controlled-load (predictive), for applications requiring probabilistic delays.

Routers are required to reserve resources to provide QoS for specified flows each time a host needs to transmit data requiring a specific QoS level. The main RSVP messages are the PATH and RESV messages. The PATH message is sent by a source that initiates the communication session. It installs *path states* on the nodes along a data path and describes the capabilities of the source. The RESV message is issued by the receiver of the communication session, and it follows exactly the path that the PATH message traveled back to the communication session source. On its way back to the source, the RESV message carries reservation requests hop-by-hop and installs QoS states at each hop. These states are associated with the specific QoS resource requirements of the destination. The RSVP reservation states are temporary states (soft states) that have to be updated regularly. This means that PATH and RESV messages will have to be retransmitted periodically. If these states are not refreshed then they will be removed. The RSVP protocol uses additional messages either to provide information about the QoS state or explicitly to delete the QoS states along the communication session path. Following is a summary message types:

- PATH — message used by senders to establish a connection;
- RESV — message used by the receivers to reserve resources;
- RESV Confirm — message sent on request to a receiver in order to confirm a RESV message;
- PATH/RESV Error — messages sent to senders/receivers to signal an error that has occurred during for a PATH/RESV message;
- PATH/RESV Tear — messages sent by senders/receivers to delete PATH/RESV information in intermediate hops.

An overview of the RSVP functionality includes:

a) End-to-end reservation with aggregation of path characteristics such as fixed delay.

b) The same type of reservation functionality in all routers. Only policy handling separates the edge of the domain from other routers.

c) Multicast and unicast reservations with receiver initiated reservations. RSVP makes reservations for both unicast and many-to-many multicast applications, adapting dynamically to changing routes as well as to group membership.

d) Shared reservations for multiple flows.

e) Support for policy handling to handle multi-operator situations since more than one operator will be responsible for RSVP's operation.

f) Flexible object definitions. RSVP can transport and maintain traffic and policy control parameters that are opaque to RSVP. Each RSVP message may contain up to fourteen classes of attribute objects. Furthermore, each class of RSVP objects may contain multiple types to specify further the format of the encapsulated data. Moreover, the signaling load generated by RSVP on the routers is directly proportional to the flows processed simultaneously by these routers.

g) Support for unidirectional reservations, but not bidirectional. (In some wireless subnetworks, the initiation of the reservations is done on a bidirectional basis.)

h) Re-scheduling of signaling message in every router. The re-scheduling of session refresh messages (aggregated and non aggregated ones) depend on the router's own refresh period timer. This means, for example, that when a session refresh message arrives at a router at the beginning of a refresh period it might have to be re-scheduled for re-sending to the next hop at the end of the refresh period.

i) Signaling initiation of RSVP error indication messages: Any time that an erroneous situation occurs a router initiates an RSVP error message.

In case of wireless networks, when a mobile host moves or the connection moves from one base station to another, it could force the communication path to change its (source/destination) IP address. The change of IP address will require that RSVP establish a new RSVP session through the new path that interconnects the two end points involved in the RSVP session and release the RSVP session on the old path. During this time, the end-to-end data path connection is incomplete (i.e., QoS disruption), and it will negatively affect the user performance.

Moreover, processing of the individual flows in the networking infrastructure may impose a significant processing burden on the machines, thus hurting throughput. These issues make it reasonable to question the

scalability and performance in a large network that supports a huge number of users.

RSVP includes much more functionality and complexity than is required in some IP networks. The QoS problem in such networks may be significantly simpler to solve. The trade-off between performance and functionality is one of the key issues in such networks, and the majority of the functionality in RSVP is not required. This is true for five reasons:

1) Most of the QoS sensitive applications do not use the multicast capabilities of RSVP. Supporting only unicast and one-to-many multicast reservations is a reasonable trade-off, since they are considerably simpler than many-to-many multicast reservations. Note that even for the one-to-many multicast reservations capability, it should be ensured that this type of reservation will not outweigh the requirement for simplicity and scalability. Without the many-to-many reservation support, protocols do not necessarily have to be receiver-oriented. Such protocols perform one pass only during the setup instead of the two passes and therefore speed up the reservation initiation. Additionally, the initiation of bidirectional reservations in combination with many-to-many reservations is very complex.

2) Edge-to-edge communication with only one operator does not require policy handling in the interior routers.

3) Path characteristics, flexible traffic parameters and QoS definitions could be solved by network dimensioning and edge functionality.

4) The huge number of per-microflow states in intermediate routers might cause severe scalability problems.

5) Initiation or re-scheduling of signaling messages might load intermediate interior routers severely. Generally, it is sufficient that edge routers and/or signaling end-points initiate or re-schedule all the signaling messages. In this case, the intermediate interior routers only forward the messages and use a dedicated field of the message to signal to other routers. This approach lightens the load on the intermediate interior routers.

More recently RSVP has been extended to reserve resources for aggregate flows and to setup explicit routes with QoS perimeters for network signaling, as is the case for the Common Open Policy Service (COPS) protocol, which is a query response protocol used to exchange policy information between a network policy server and a set of clients [RFC-2748].

Bandwidth management

Although an SLA establishes an agreement with an ISP, it is still necessary for an intranet service customer to decide upon how such resources are shared. Individual hosts can make arbitrary decisions; however a more intelligent scheme establishes a bandwidth manager (BM) to make

decisions for all hosts based upon management policies [RFC-2998]. The BM would use protocols such as RSVP and/or LDAP (Lightweight Directory Access Protocol) to establish classification, shaping policies etc. at the boundary router, as well as with corresponding BMs in ISP networks and the destination intranet.

Although the signaling mechanism involved is similar to that described above for IntServ, there are four main differences:

- it is the sender that requests resources, not the receiver;
- a request can be rejected when the BM receives the PATH message from the sender; in IntServ a request may be rejected only when a router receives the RESV from the receiver;
- a BM can aggregate multiple requests and make a single request to the next BM;
- each domain behaves like a single node represented by the BM; ISP core routers are not involved in this process.

4.2.2 Differentiated Services

The Differentiated Services architecture was developed as an alternative resource allocation scheme for service providers' networks. By mid-1997 service providers felt that Integrated Services were not ready for large-scale deployment, and at the same time the need for an enhanced service model had become more urgent. The Internet community started to look for a simpler and more scalable approach to offer a better than best-effort service.

The IETF formed a new working group to develop a framework and standards for allocating different levels of services in the Internet. The new approach, called Differentiated Services (DiffServ), is significantly different from Integrated Services. Instead of making per-flow reservations, Differentiated Services architecture uses a combination of edge policing, provisioning, and traffic prioritization to make possible service differentiation.

In the Differentiated Services architecture, users' traffic is divided into a small number of forwarding classes. For each forwarding class, the amount of traffic that users can inject into the network is limited at the edge of the network. By changing the total amount of traffic allowed in the network, service providers can adjust the level of resource provisioning and hence control the degree of resource assurance to the users.

The edge of a DiffServ network is responsible for mapping packets to their appropriate forwarding classes. This packet classification is typically

done based on the service level agreement (SLA) between the user and its service provider. The nodes at the edge of the network also perform traffic policing to protect the network from misbehaving traffic sources. Nonconforming traffic may be dropped, delayed or marked with a different forwarding class. The forwarding class is directly encoded into the packet header. After packets are marked with their forwarding classes at the edge of the network, the interior nodes of the network can use this information to differentiate the treatment of the packets. The forwarding classes may indicate drop priority or resource priority. For example, when a link is congested, the network will drop packets with the highest drop priority first.

The DiffServ framework provides a methodology for offering a range of IntServ without the requirement for the substantial overhead needed for per-flow state information in every router as is the case with the IntServ model. Potentially DiffServ has been available in IPv4 by way of the Type Of Service (TOS) field, but to date it has been rarely used.

DiffServ defines a set of packet forwarding criteria constituting the per-hop behavior (PHB) [RFC2474]. The TOS field has been renamed as Differentiated Services, which contains a 6-bit field called DiffServ Code Point (DSCP). Packets are handled based upon the DiffServ field and therefore a variety of classes can be defined, thus creating a priority scheme (DSCP serves as a table index to look-up a PHB). However individual flows within a DiffServ class cannot be differentiated.

By the use of classification, policing, shaping and scheduling a variety of services can be provided. Nevertheless, DiffServ only defines the DiffServ and PHB fields. It is up to the implementer (i.e. ISP, manufacturer) to create/configure appropriate handling mechanisms. Services currently being defined include the two following forwarding PHB groups:

- assured service, for applications requiring better reliability than best-effort service;
- premium service, for applications requiring low delay and low jitter.

Assured service

Assured service [RFC2638] is intended for customers who require an improved QoS over best-effort, especially concerning packet loss. Assured service is based on Assured forwarding [RFC-2597] and it resembles ATM's ABR (Available Bit Rate) or VBR (Variable Bit Rate) services. Such a Service Level Agreement (SLA) will allocate bandwidth, but applications must share this bandwidth in accordance with their own policy. The ISP's ingress router performs classification and policing. If the traffic rate does not exceed the SLA bit rate then it is said to be *in* profile. Excess packets are *out* profile and

are handled by Random Early Detection (RED) or RIO (RED with *in* and *out*) queue management discipline (see Section 3.2).

This service could be used to implement, for example, the so-called Olympic service, which consists of three service classes: bronze, silver, and gold. Packets are assigned to these three classes so that packets in the gold class experience lighter load (and thus have greater probability for timely forwarding) than packets assigned to the silver class. Same kind of relationship exists between the silver class and the bronze class. If desired, packets within each class may be further separated by giving them either low, medium, or high drop precedence.

Drop precedence	***Class 1***	***Class 2***	***Class 3***	***Class 4***
Low	**001***01*0	**010***01*0	**011***01*0	**100***01*0
Medium	**001***10*0	**010***10*0	**011***10*0	**100***10*0
High	**001***11*0	**010***11*0	**011***11*0	**100***11*0

Table 1 — Recommended values for the DSCP in Assured Forwarding PHB.

Premium service

For applications which require a specific maximum or average bit rate then a premium service, based on Expedited forwarding [RFC-2598], is required. This is a low delay, low jitter service. Traffic rates in excess of the SLA will result in packet discard. This service is appropriate for IP telephony, video conferencing and certain Virtual Private Networks (VPN); it is similar to ATM's Constant Bit Rate (CBR).

For Premium Services it is necessary to support both static and dynamic SLAs in order for customers to request a different service level on the fly, without having first subscribed to them, although some admission control mechanism is needed. When Premium Service traffic arrives (bit P set) traffic may need to be reshaped before it leaves the customer's network to ensure that it conforms to the SLA profile.

Various schemes have been proposed to ensure a fair and even balance between Premium and Assured traffic flow with respect. example include:

- control Premium to Assured traffic flow to a specified ratio (e.g. 30%);
- packet rates in excess of the SLA can be discarded at the network ingress;
- implement a Weighted Fair Queueing (WFQ) scheme between Premium and Assured services.

The Premium Service queue should normally be empty or at least very short, thus ensuring low delay and jitter. However no guarantees are usually

provided regarding the bounds of this delays, or for loss rates. Uneven traffic flows can cause problems for Premium Services.

Differentiated Services do not require resource reservation setup. The allocation of forwarding classes is typically specified as part of the SLA between the customer and its service provider, and the forwarding classes apply to traffic aggregates rather than to individual flows. These features work well with transaction-orientated Web applications. The Differentiated Services architecture also eliminates many of the scalability concerns with Integrated Services. The functions that interior nodes have to perform to support Differentiated Services are relatively simple. The complex process of classification is needed only at the edge of the network, where traffic rates are typically much lower. Naturally this paradigm has the drawback that streams are no longer differentiated within one class, but it is generally assumed that their behavior is similar; in addition policing/shaping should act on them at network edges.

The Differentiated Services approach relies on provisioning to provide resource assurance. The quality of the assurance depends on how provisioning is carried out and how the resources are managed in the network. These issues are explored in the next section, where we discuss performance optimization in the networks. Because of the dynamic nature of traffic flows, precise provisioning is difficult. Thus it generally is more difficult, and certainly more expensive, to provide deterministic guarantees through provisioning rather than reservation.

4.2.3 Integrated Services over Differentiated Services

The IntServ over DiffServ architecture addresses the problem of providing end-to-end QoS using the IntServ model over heterogeneous networks. In this scenario, DiffServ is used in these networks to provide edge-to-edge QoS. The IntServ over DiffServ architecture allows at least two different possible deployment strategies.

The first is based on statically allocated resources in the DiffServ domain. In this strategy, the DiffServ domain is statically provisioned. Furthermore, with this strategy the devices in the DiffServ network region are not RSVP (or any other dynamic signaling) aware. However, it is considered that each edge node in the customer network consists of two parts. One part of a node is a standard IntServ that interfaces to the customer's network region and the other part of the same node interfaces to the DiffServ network region. All edge nodes in the customer network maintain a table that indicates the capacity provisioned per Service Level

Specification* (SLS) at each DiffServ service level. This table is used to make admission control decisions on IntServ flows that cross the DiffServ region.

A disadvantage of this approach is that the edge nodes in the customer network will not be aware of the traffic load in the nodes located within the DiffServ domain. Therefore, a congestion situation on a communication path within the DiffServ domain cannot be detected by any of these edge nodes. Congestion within a DiffServ domain may arise due to difficulties in static provisioning [RFC-2990]. Repeated steps of aggregation/ disaggregation of traffic or other stochastic disturbances may adversely affect the QoS. In contrast to the IntServ architecture, no mathematical proof of a reliable QoS delivery by DiffServ architectures has yet been provided.

An immediate conclusion is to take such possibilities into account from start. Accordingly, further improvements could be achieved by providing congestion signaling from within such a DiffServ domain to the border between the two administrative domains in question. As is the case with TCP control, it is anticipated that (some) "subscribers" to such a disturbed service would back off and thus improve the traffic load situation within the domain. Appropriate signaling mechanisms would be needed that reflect violation of a specified QoS level. If subscribers do back off the original QoS level would be resumed.

Feedback information and signaling is needed in the next generation of a DiffServ architecture that delivers its specified classes of service by a combination of resource provisioning and cooperation with the subscribers. This would be similar to native TCP/IP environments, but with integrated DiffServ characteristics. While resource provisioning is static and does cover the most common and regular case of QoS support, feedback signaling and adaptation or dynamic conditioning would deal with the (hopefully) rare event of insufficient provisioning. Note that the original service specification would explicitly entail the possibility of a reduction in the advertised DiffServ bandwidth and the expectation of subscribers to back-off according to the needs of reestablishing a DiffServ QoS class. More details on this concept are given in [Mee-01].

The second possible strategy is based on dynamically allocated resources in the DiffServ domain. According to [RFC-2998], this can be done using RSVP-aware DiffServ routers. However, this approach has most of the drawbacks described in Section 4.2.1, and per-microflow state information is kept in the intermediate routers. Furthermore, dynamic provisioning may be too slow to respond quickly enough to congestion events. Alternatively,

* Service Level Specification is a representation defined by IETF that allows dynamic exchange and negotiation of service level requirements (e.g. between bandwidth brokers) [God-02].

resources in the DiffServ domain can be dynamically allocated using Aggregated RSVP.

Static trunk reservations based on Differentiated Services

A significant problem in deploying an end-to-end per-flow resource reservation signaling scheme is its scalability. This can be solved by aggregating (trunking) several individual reservations into a common reservation trunk. The reservation trunks can either be statically or dynamically configured. When the reservation trunks are statically configured, no signaling protocol is required for performing the reservation of network resources but is likely to be a difficult management problem. However, due to the different mobility requirements (such as hand-over) and QoS requirements (such as bandwidth) that the multi-bitrate applications impose on a network that supports mobile users, it will be difficult to configure the trunked reservations statically and at the same time utilize the network efficiently.

In particular, and focusing on DiffServ [RFC-2475], an important open point is that such an architecture lacks a standardized admission control scheme, and does not intrinsically solve the problem of controlling congestion in the Internet. As previously explained, the edge nodes in the customer network will not be aware of the traffic load in the nodes located within the DiffServ domain.

Therefore, a congestion situation on a communication path within the DiffServ domain cannot be detected by any of these edge nodes. Congestion within a DiffServ domain may arise due to difficulties in static provisioning [RFC-2990]. Upon overload in a given service class, all flows in that class suffer a potentially harsh degradation of service. "A Framework for Integrated Services Operation over DiffServ Networks" [RFC-2998] recognizes this problem and points out that "further refinement of the QoS architecture is required to integrate DiffServ network services into an end-to-end service delivery model with the associated task of resource reservation". It is suggested next to define an "admission control function which can determine whether to admit a service differentiated flow along the nominated network path".

In the following we expand on this issue, in the framework of the typical hybrid reference network shown in Figure 4, which includes a DiffServ region in the middle of a larger network supporting IntServ end-to-end. Notice that some of the following considerations also apply to an all DiffServ network. The source host Tx, the destination host Rx, the Edge Routers and the Border Routers execute the functions listed in [RFC-2998]. In particular, we assume that both sending and receiving hosts use RSVP to communicate the quantitative QoS requirements of QoS-aware applications

running on the hosts. Obviously, admission control in the IntServ subnetworks is signaled using RSVP.

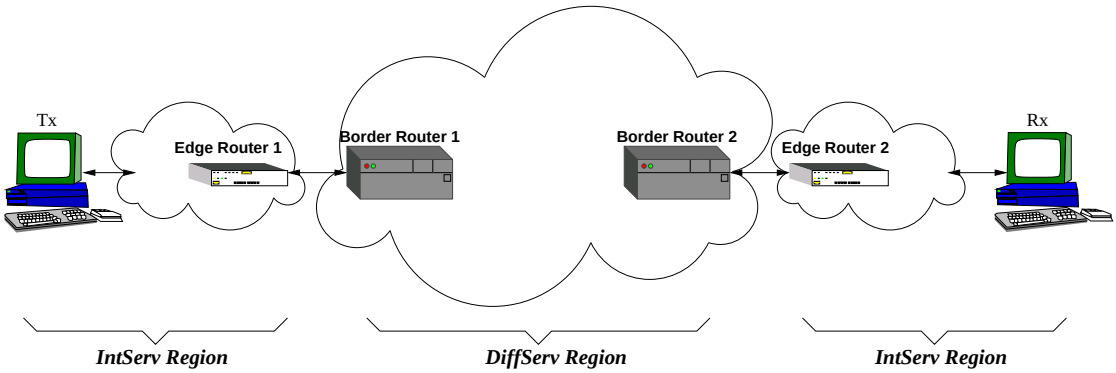


Figure 4 —Sample Network Configuration.

Requests for IntServ services must be mapped onto the underlying capabilities of the DiffServ network region. Aspects of such mapping include [RFC-2998]:

1) selecting an appropriate packet handling buffers, or a set of such buffers, for the requested service;
2) performing appropriate policing (perhaps including shaping or remarking) at the edges of the DiffServ region;
3) exporting IntServ parameters from the DiffServ region (e.g. for the updating of ADSPECs*);
4) performing admission control on the IntServ requests that takes into account resource availability in the DiffServ region.

In principle, the availability of DiffServ per-hop behaviors along with mechanisms to statically or dynamically limit the absolute level of traffic within a traffic class allows the DiffServ network cloud to act as a network element within the Integrated Services framework. In other words, an appropriately designed, configured and managed DiffServ network cloud can act as one component of an overall end-to-end QoS controlled data path using the Integrated Services framework, and therefore support the delivery of IntServ QoS services [Wro-01]. To this end, point 4 above, i.e. the admission control function is a decisive factor.

In fact, QoS aware services require that the amount of arriving traffic be limited by suitable admission control. Two issues are of interest [Wro-01]:

- the method used by the DiffServ cloud to determine whether sufficient resources are available;

* ADSPEC is a field in RSVP messages used by QOS-enabled network devices in the path between sender and receiver to advertise their service capabilities, resource availability, and transmission characteristics.

- the method used by the overall network to query the DiffServ cloud about this availability.

Within the cloud, the admission control mechanism is closely related to resource provisioning. If some form of static resource provisioning is used, the admission control function can be performed by any network component that is aware of this allocation, such as a properly configured boundary router. If resource allocation within the network cloud is dynamic (e.g. a dynamic bandwidth broker or signaling protocol) then this protocol can also perform the admission control function by refusing to admit new traffic when it determines that it cannot allocate appropriate new resources.

The key to providing absolute, quantitative QoS services within a DiffServ network is to ensure that at each hop in the network the resources allocated are sufficient to handle the arriving traffic. This can be done through a variety of mechanisms ranging from static provisioning to dynamic per-hop signaling within the cloud. Two situations are possible:

a) With per-cloud provisioning, sufficient resources are made available in the network so that traffic arriving at an ingress point can flow to "any" egress point without violating the resource allocation requirements. In this case, admission control and traffic management decisions need not be based on destination information.

b) With per-path provisioning, resources are made available in the network to ensure that the resource allocation requirements will not be violated if traffic arriving at an ingress point flows to one (in the unicast case) specific egress point. This requires that admission control and resource provisioning mechanisms take into account the egress point of traffic entering the network, but results in more efficient resource utilization.

The per-cloud versus per-path decision is independent of decisions about static versus dynamic provisioning. It is often assumed that dynamic provisioning is necessarily per-path, while static provisioning is more likely to be per-cloud. In reality, all combinations of options may be useful in different circumstances.

In any case, there need to be entities that are able to allow or refuse service requests, possibly on the basis of resource utilization. In other words, we also need an admission control function acting on the DiffServ cloud. We can proceed along two different routes:

a) Define an admission control function that is also able to operate *within* the DiffServ cloud. This could solve the problem even in the case of isolated or entirely DiffServ networks (that is, not part of an end-to-end RSVP loop).

b) Export suitable characteristics of the DiffServ cloud toward the

IntServ part so that admission control can be performed by the latter (that is, by RSVP).

Considering the second of these possibilities, in order to provide Guaranteed QoS, it would be necessary to export the error terms, referred to as C and D in the specification [RFC2212], which would allow the customer to calculate the bandwidth to request from the network in order to achieve a particular queuing delay target. The error term C is the rate-dependent error term. It represents the delay a datagram in a flow might experience due to the rate parameters of the flow. The error term D is the rate-independent, per-element error term and represents the worst-case non-rate-based transit time variation through the service element.

The difficulty in characterizing the parameters C and D is that, unlike the IntServ model, where the C and D terms are a local property of the router, in the case of DiffServ these terms depend not only on the topology of the cloud, but also on the internal traffic characteristics of potentially all traffic in the cloud handled with the buffers chosen to support the Guaranteed QoS. Hence, the existence of upper bounds on delay through the cloud implies centralized knowledge about the topology of the cloud and traffic characterization.

These considerations imply that determination of the bound on the delay through the DiffServ cloud should be performed off-line, perhaps as part of a traffic management algorithm, based on the knowledge of the topology, traffic patterns, shaping policies, and other relevant parameters of the cloud [Wro-01]. However, this turns out to be a rather difficult task and the amount of traffic requiring end-to-end guaranteed service across the DiffServ cloud should be rather small, potentially leading to severe inefficiencies.

Additionally, to provide a strict delay bound, the utilization factor of the bandwidth allocated to this traffic has to be deterministically bounded on all links in the network. This can be either ensured by signaled admission control (such as using dynamic resource reservation [Wes-01]) or by a static provisioning mechanism. It should be noted that if provisioning is used, then to ensure deterministic load/service rate ratio on all links, the network should be strongly overprovisioned to account for possible inaccuracy of traffic matrix estimates [Wro-01].

In conclusion, providing QoS aware service over a DiffServ cloud without admission control functions able to operate within the cloud itself potentially leads to severe inefficiencies. In fact, the worst case provisioning model targeting a particular utilization bound results in substantially more overprovisioning than the point-to-point provisioning using an estimated traffic matrix, which in turn is potentially more inefficient than explicit point-to-point bandwidth allocation using signaled admission control.

This brings us to option a) above, the definition of an admission control function within the DiffServ region. To this end, we cannot use explicit per flow signaling, since this would lead to a architecture with numerous states. Similarly, we do not want to modify the basic router operation by introducing packet marking schemes or forcing routers to parse and interpret higher layer information. What we would like to do is to implicitly convey the status of inner DiffServ routers to the edges of the cloud (or to the end points, when the DiffServ net is an isolated one), by means of scalable, DiffServ compliant procedures, so that suitable devices can make appropriate admission control decisions without violating the DiffServ paradigm. A possible way to do this has been proposed in [Wes-01] and [Bia-01].

Dynamic trunk reservations with Aggregated RSVP

The reservation trunks can be dynamically configured by using a signaling protocol that manages various mechanisms for dynamic creation of an aggregate reservation, classification of the traffic to which the aggregate reservation applies, determination of the bandwidth needed to achieve the requirement, and recovery of the bandwidth when the sub-reservations are no longer required.

The first router that handles the aggregated reservations could be called an Aggregator, while the last router in the transit domain that handles the reservations could be called a Deaggregator. The Aggregator and Deaggregator functionality is located in the edge nodes. In particular, an Aggregator is located in an ingress edge node, while a Deaggregator is located in an egress edge node, relative to the traffic source.

The aggregation region consists of a set of aggregation-capable network nodes. The Aggregator can use a policy that is based on local configuration and local QoS management architectures to identify and mark the packets passing into the aggregated region. For example, the Aggregator may be the base station that aggregates a set of incoming calls and creates an aggregate reservation across the edge-to-edge domain up to the Deaggregator. In this situation the call signaling is used to establish the end-to-end resource reservations. Based on policy, the Aggregator and Deaggregator will decide when the Aggregated states will be refreshed or updated.

One example of a protocol that can be used to accomplish QoS dynamic provisioning via trunk reservations is the RSVP Aggregation signaling protocol specified in [RFC-3175].

With regards to aggregated RSVP, even if the reservation is based on aggregated traffic, the number of re-negotiations of the allocated resources due to mobility (hand-over) does not decrease and each re-negotiation of

resources has the same performance requirements as the per-flow reservation procedure.

Note that the aggregated RSVP solution may use a policy to maintain the amount of bandwidth required on a given aggregate reservation by taking account of the sum of the underlying end-to-end reservations, while endeavoring to change it infrequently. However, such solutions (policies) are very useful assuming that the cost of the overprovisioned bandwidth is not significant. However, in networks where overprovisioning is not practical due to high costs of transmission links, a more dynamic QoS provisioning solution is needed. Furthermore, the aggregated RSVP scheme is receiver initiated and cannot support bidirectional reservations.

In the aggregated RSVP scheme the resource reservation states stored in all the RSVP-aware edge and interior nodes represent aggregated RSVP sessions or trunks of RSVP sessions. Therefore, the number of the resource reservation states in the aggregated RSVP scheme compared to the (per-flow) RSVP scheme is decreased. However, in a DiffServ-based domain the number of the aggregated RSVP sessions depends on:

a) The number of Aggregators/Deaggregators — this depends on the number of the edge nodes used. For example, in an IP-based wireless network, the number of the edge nodes can depend on the number of base stations and controlling gateways.

b) The network topology used — when the communication is performed in a meshed way (that is, all-to-all), it will imply that many communication paths will have to be maintained by the network simultaneously.

c) The number of DiffServ Code Points (DSCPs) used — more than one traffic class will be supported within a network.

Therefore, the number of the aggregated RSVP reservation states within such a network will be significant.

4.3 Performance optimization

Once the resource allocation architecture and service models are in place, the next issue is performance optimization; that is, how to organize the resources in a network in the most efficient way to maximize the probability of delivering commitments and minimize the cost of delivering these commitments. The connection between performance optimization and QoS support may seem less direct compared with resource allocation. Performance optimization is, however, an important building block in the deployment of QoS. Implementing QoS goes way beyond just adding mechanisms such as traffic policing, classification, and scheduling; fundamentally, it is about developing new services over the Internet. Service providers must make a good business case so that customers are willing to pay for the new services and the new services will increase the return of their investment in the networks. The cost-effectiveness of the new services made possible by QoS capabilities is a major factor in the dissemination of these services.

The Internet's datagram routing was not designed for optimizing the performance of the network. Scalability and maintaining connectivity in the face of failures were the primary design objectives. Routing protocols typically select the shortest path to a destination based on some simple metrics, such as hop count or delay. Such simple approaches are clearly not adequate for supporting resource allocation. For example, to make a reservation, we need to find a path with certain requested resources, such as bandwidth, but IP routing does not have the necessary information to make such decisions. Simply using the shortest-path algorithm for selecting paths is likely to cause high rejection rate and poor utilization. The shortest-path routing does not always use the diverse connections available in the network. In fact, traffic is often unevenly distributed across the network, which can create congestion hot spots at some points while some other parts of the network may be very lightly loaded.

Performance optimization requires additional capabilities in IP routing and performance management tools. To manage the performance of a network, it is necessary to have explicit control over the paths that traffic flows traverse so that traffic flows can be arranged to maximize resource commitments and utilization of the network. Multi-Protocol Label Switching (MPLS) has a mechanism called explicit routing that is ideal for this purpose. MPLS uses the label-switching approach to set up virtual circuits in IP-based networks. These virtual circuits can follow destination-based IP routing, but the explicit routing mechanism in MPLS also allows us to specify hop-by-hop the entire path of these virtual circuits. This provides a way to override the destination-based routing and set up traffic trunks based on traffic-engineering objectives.

The process of optimizing the performance of networks through efficient provisioning and better control of network flows is often referred to as traffic engineering. Traffic engineering uses advanced route selection algorithms to provision traffic trunks inside backbones and arrange traffic flows in a way that maximizes the overall efficiency of the network. The common approach is to calculate traffic trunks based on flow distribution and then set up the traffic trunks as explicit routes with the MPLS protocol. The combination of MPLS and traffic engineering provides IP-based networks with a set of advanced tools for service providers to manage the performance of their networks and provide more services at less cost.

4.3.1 Multi-layer switching

In order to achieve a well-engineered IP network that can provide the flow requirements for the DiffServ model, the conventional IP routing architecture has to change.

Multi-layer switching specifies an integration of layer 2 switching with layer 3 routing. Networks started to be constructed using an *overlay* model in which a logical IP router topology operates over and is independent of an underlying layer 2 switching technology such as Frame Relay or ATM. There were however complexities in operating this model. For example PVCs (Permanent Virtual Circuit) between routers had to be manually configured. Further, use of SVCs (Switched Virtual Circuit) mandated the resolution of IP to ATM addresses. Although this approach derives the benefits of both layer 2 and layer 3 architectures, difficulties arose in the complexity of mapping between two separate topologies, address spaces, routing protocols, signaling protocols and resource allocation systems.

Further evolution occurred for the *peer* model, in which integrated switches/routers maintained a single IP addressing space and ran a single IP routing protocol — just like a network of routers. Some work was required to map IP traffic to layer 2 switched path via IP switching control protocols. This work resulted in the evolution of multi-layer switching solutions. In particular MPLS represents an important effort designed to decrease the complexity of combining layer 2 switching and layer 3 routing into an integrated system.

Multi-layer switching solutions are characterized by two components: control and forwarding (based on label swapping).

Forwarding and control mechanisms

The control and forwarding components are common to all switching methodologies (including MPLS). The control component uses routing protocols such as OSPF, IS-IS* or BGP4 (Border Gateway Protocol version 4) to exchange control information and maintain forwarding tables with its neighbors. This forwarding table provides information necessary for a routing decision, thus forming a switched path between the input and output ports. The control components are separated from the forwarding component and thus each can be modified independently of the other.

The forwarding component of virtually all multi-layer switching solutions is based upon a label-swapping forwarding algorithm (for example this is the algorithm used to forward data in ATM and Frame Relay networks). Signaling and label distribution are fundamental to the operation of a label-swapping forwarding algorithm. A label is a short fixed-length value carried in the packet's header and is used to identify a Forwarding Equivalent Class (FEC). A label is similar to a connection identifier such as that used in ATM (Virtual Path Identifier/Virtual Circuit Identifier) or in Frame Relay (Data Link Connection Identifier), as it has only local significance and maps traffic to a specific FEC. FEC represents a class of packets that are forwarded over the same path even if their ultimate destinations are different.

Label-swapping forwarding algorithms require that a packet classification occurs at the network entry point and that an initial label be assigned to every packet. Within the network label switches ignore a packet's network layer header and forward the packet using the label switch; then the exit switch discards the label switch and forwards the packet using conventional longest-match IP forwarding. The created path is equivalent to a virtual circuit as it defines entry to exit points through the network and all packets follow this path.

Multi−Protocol Label Switching

MPLS [RFC-3031] was originally seen as an alternative approach for supporting IP over ATM. Although several approaches for running IP over ATM were standardized, most of the techniques are complex and have scaling problems. The need for more seamless IP/ATM integration led to the development of MPLS in 1997, a forwarding scheme which primarily evolved from Cisco's *Tag Switching* [RFC-2105]. The MPLS approach allows IP routing protocols to take direct control over ATM switches, and thus the IP control plane can be tightly integrated with the rest of the IP network.

---

* Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol defined by ISO 10589.

The technique that MPLS uses is known as label switching. A short, fixed-length label is encoded into the packet header and used for packet forwarding. When a label switch router (LSR) receives a labeled packet, it uses the incoming label in the packet header to find the next hop and the corresponding outgoing label. With label switching, the path that a packet traverses through, called the label switched path (LSP), has to be set up before it can be used for label switching.

In addition to improving IP/ATM integration, MPLS may also be used to simplify packet forwarding. Label lookup is much easier compared with prefix lookup in IP forwarding. With MPLS, packet forwarding can be done independent of the network protocols; thus forwarding paradigms beyond the current destination-based one can be easily supported. However, the driving force behind the wide deployment of MPLS has been the need for traffic engineering in Internet backbones. The explicit route mechanism in MPLS provides a critical capability that is currently lacking in the IP-based networks. MPLS also incorporates concepts and features from both Integrated Services and Differentiated Services. For example, MPLS allows bandwidth reservation to be specified over an LSP and packets can be marked to indicate their loss priority. All these features make MPLS an ideal mechanism for implementing traffic-engineering capabilities in the Internet.

The purpose of MPLS is not to replace IP routing but rather to enhance the services provided in IP-based networks by offering scope for traffic engineering, guaranteed QoS and virtual private networks (VPNs). MPLS works alongside the exiting routing technologies and provides IP networks with a mechanism for explicit control over routing paths. MPLS allows two fundamentally different data-networking approaches, datagram and virtual circuit, to be combined in IP-based networks. The datagram approach, on which the Internet is based, forwards packets hop-by-hop based on their destination addresses. The virtual circuit approach, used in ATM and frame relay, requires connections to be set up. With MPLS, the two approaches can be tightly integrated to offer the best combination of scalability and manageability.

MPLS defines new IP signaling and Label Distribution Protocols (LDP), as well as extensions to existing protocols in order to support multi-vendor interoperability. The control protocols are based on IP addressing and transport and therefore can be more easily integrated with other IP control protocols. This creates a unified IP-based architecture in which MPLS is used in the core for traffic engineering and IP routing for scalable domain routing. In several recent proposals the extension of the MPLS protocols to the optical transport networks has even been considered. MPLS does not implement any of the ATM Forum signaling or routing protocols so the complexity of coordinating two different protocol architectures is eliminated. In this way

MPLS brings significant benefits to IP-based networks. An LSR examines only the label in the forwarding packet. The network protocol can be IP or others which is why the technique is called multi-protocol label switching.

MPLS requires a protocol to distribute labels to setup Label Switched Paths (LSP) and this is defined as a LDP [RFC-3036]. An LSP is similar to an ATM VC and is unidirectional. MPLS LSRs use the protocol to negotiate the semantics of each label, i.e. how to handle a packet with a particular label from the peer. LSP setup can be control driven (triggered by control traffic such as routing updates) or data driven (triggered by the request of a flow or a traffic trunk). In MPLS a traffic trunk is an aggregation of flows with the same service class that can be sent over a LSP. The LSP between two routers can be the same as the layer 3 hop-by-hop route, or the sender LSR can specify an Explicit Route (ER) for the LSP. The ability to setup ERs is one of the most useful features of MPLS. A forwarding table indexed by labels is constructed as the result of label distribution. Each forwarding table entry specifies how to process packets carrying the indexing label.

Packets are classified and rerouted at the ingress LSRs of a MPLS-capable domain. MPLS headers are then inserted. When an LSR receives a labeled packet, it will use the label as the index to look up the forwarding table. This is faster than the process of parsing the routing table in search of the longest match carried out in IP routing. The packet is processed as specified by the forwarding table entry. The incoming label is replaced by the outgoing label and the packet is switched to the next LSR. This label-switching process is similar to ATM's VCI/VPI processing. Inside a MPLS domain packet forwarding, classification and QoS service are determined by the labels and Class of Service (CoS) fields. This makes core LSRs simple. Before a packet leaves a MPLS domain its MPLS label is removed.

MPLS LSPs can be used as tunnels. When a packet enters the start point of a tunnel its path is completely determined. With MPLS a packet's path is completely determined by the label assigned at the ingress LSR. There is no need to enumerate every intermediate router of the tunnel. MPLS is therefore more efficient in terms of header overhead than other tunneling mechanisms. Thus MPLS has the advantage of providing fast packet classification and forwarding as well as an efficient tunneling mechanism.

MPLS can be used together with DiffServ to provide QoS in IP-based networks [Fau-01]. In such an architecture it is likely that for each ingress-egress pair a separate LSP is created for each traffic class. In this case a total number of C*N*(N-1)/2 LSPs are needed, where C is the number of traffic classes and N is the number of boundary routers. To reduce the number of LSPs, all ingress routers to a single egress router can be merged into a sink tree. The total number of sink trees is then C*N. It is also possible to transmit packets of different traffic classes and use the CoS bits to

differentiate packet classes. In this case the number of sink trees is reduced to N. In this architecture as the number of transiting flows increases the number of flows in each LSP or sink tree also increases, although the number of LSPs or sink trees themselves need not increase, which makes the architecture more scalable. The operation of the routers is basically the same in this architecture as in the DiffServ field-based architecture described previously.

Whether a particular ISP's architecture is DiffServ field-based or MPLS-based is transparent to other ISPs. Therefore the DS field-based and the MPLS-based architectures can easily interoperate. Each customer domain still needs a bandwidth manager to allocate services and to request resources on behalf of the customer domain when the SLA is dynamic. Since LSPs are configured within the ISPs, resource requests can be easily hidden from the core routers by tunneling them from ingress to the egress routers. Therefore BMs may not be needed in MPLS-based ISP networks.

4.3.3 Traffic engineering

The basic problem addressed in traffic engineering is as follows: given a network and traffic demands, how can traffic flows in the network be organized so that an optimization objective is achieved? The objective may be to maximize the utilization of resources in the network or to minimize congestion in the network. Typically the optimal operating point is reached when traffic is evenly distributed across the network. With balanced traffic distribution, both queuing delay and loss rates are at their lowest points.

Obviously these objectives cannot be achieved through destination-based IP routing; there simply is not sufficient information available in IP routing to make possible such optimization. In traffic engineering, advanced route selection techniques, often referred to as constraint-based routing in order to distinguish them from destination routing, are used to calculate traffic trunks based on the optimization objectives. To perform such optimization, the traffic-engineering system often needs network-wide information on topology and traffic demands. Thus traffic engineering is typically confined to a single administrative domain.

The routes produced by constraint-based routing are most likely different from those in destination-based IP routing. For this reason these constraint-based routes cannot be implemented by destination-based forwarding. In the past, many service providers used ATM in the backbones to support constraint-based routing. ATM virtual circuits can be set up to match the traffic patterns; the IP-based network is then overlaid on top of these virtual circuits. MPLS offers a better alternative since it offers similar functions yet can be tightly integrated with IP-based networks.

The existing Internet backbones have used the so-called overlay model for traffic engineering. With the overlay model, service providers build a virtual network comprising a full mesh of logical connections between all edge nodes. Using the traffic demands between the edge nodes as input, constraint-based routing selects a set of routes for the logical connections to maximize the overall resource utilization in the network. Once the routes are computed, MPLS can be used to set up the logical connections as LSPs exactly as calculated by constraint-based routing.

The downside of the overlay model is that it may not be able to scale to large networks with a substantial number of edge nodes. To set up a full-mesh logical network with N edge nodes, each edge node has to connect to the other (N - 1) edge nodes, resulting in N*(N - 1) logical connections. This can add significant messaging overheads in a large network. Another problem is that the full-mesh logical topology increases the number of peers (neighbors that routers talk to) that a routing protocol has to handle; most current implementation of routing protocols cannot support a very large number of peers. In addition to the increased peering requirements, the logical topology also increases the processing load on routers during link failures. Because multiple logical connections go over the same physical link, the failure of a single physical link can cause the breakdown of multiple logical links from the perspective of IP routing.

Traffic engineering without full-mesh overlaying is still a challenge. One heuristic approach that some service providers have used is to adjust traffic distribution by changing the link weights in IP routing protocols. For example, when one link is congested, the link weight can be increased in order to move traffic away from this link. Theoretically one can achieve the same traffic distribution as in the overlay model by manipulating the link weights in the Open Shortest Path First (OSPF) routing protocol. This approach has the advantage that it can be readily implemented in existing networks without major changes to the network architecture.

Architectural issues

Switching techniques designed to offer traffic classification and speed — and as far as possible to minimize routing large volumes of traffic — is central to traffic engineering. However such techniques are very much what ATM was designed to do. The momentum behind MPLS and DiffServ in particular arises from the difficulties in interfacing with resource allocation systems in underlying protocols such as ATM. Further, it cannot be assumed that such a classification based underlying network even exists on an end-to-end basis. Trying to link together multiple classification-based frame level protocols to achieve seamless end-to-end connectivity creates many problems.

ATM and SDH are giving way to a simplified structure of IP over MPLS over optical transport (Figure 5). Although this simplification of the transport architectures makes a lot of sense, it is having an interesting side effect: layer 3 is enlarged at the expense of layers 1 and 2. Mechanisms to carry out control and management functions, such as multicasting, congestion management, transport configuration, protection switching, path management, security, VPN tunneling, caching, filtering etc. are still required. Simplifying the underlying layers still means that certain essential control and management functions must be carried out in other parts of the protocol stack.

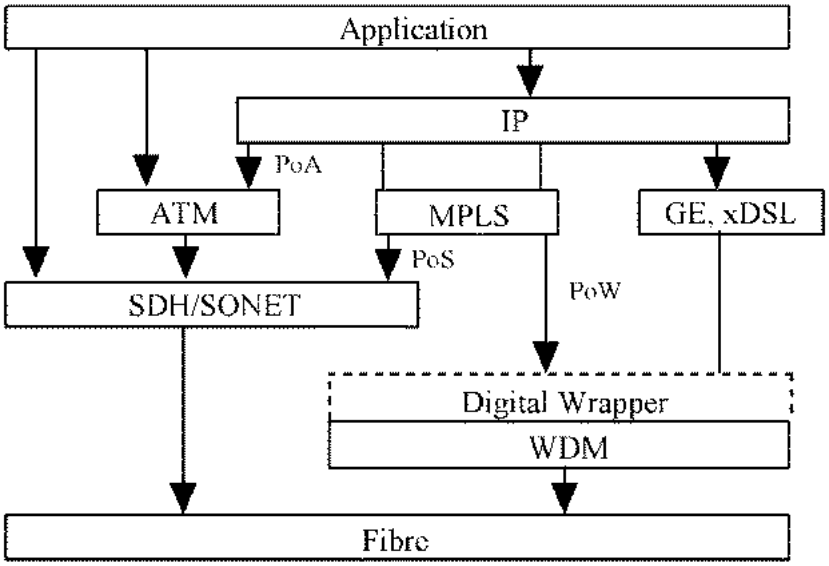


Figure 5 —Evolution in protocol layering [Hun-02].

Constraint−based routing

Network congestion can result from a shortage of resources or an uneven traffic distribution. Current dynamic routing protocols, such as RIP-2 (Routing Information Protocol version 2), IGRP (Interior Gateway Routing Protocol), OSPF etc. are based upon Bellman-Ford and Dijkstra's algorithms and use relatively simple metrics to determine the shortest path. More recent developments, such as the equal-cost multi-path options used in OSPF version 2 [RFC-2178] and IS-IS assist in distributing the load across multiple paths. QoS routing and Constraint-Based Routing (CBR) in particular is a more recent development which calculates routes where multiple constraints exist and offers a number of alternative paths that meet the QoS requirements [Cal-01]. CBR might be used to assign bandwidth or service class characteristics to an LSP or one may want to ensure that alternative routing via separate paths is available. Thus CBR takes into account the network topology, flow specifications, availability of links and other

specified policies. Metrics used by CBR include: hop count, bandwidth, transit delay, jitter, availability, monetary cost etc. This requires routers to distribute link state information and to determine optimum paths accordingly.

CBR can be based upon traffic classes, traffic trunks, flow-based and topology-based metrics, as well as source and destination addresses. The finer granularity of the parameters, the better the result, although this implies a need for greater bandwidth in order to distribute link-state data. CBR offers support to DiffServ in selecting routes to meet the QoS requirements. It offers support to RSVP in determining optimal paths for resource reservation by taking QoS requirements into consideration.

CBR also operates well with MPLS, as even though CBR determines the route based upon resources and topology information and MPLS uses its LDP to setup LSPs, the two benefit each other. The statistics resulting from the setting up of an MPLS's LSP can assist CBR in determining traffic flows between ingress/egress pairs. Thus CBR can calculate the routes for setting up LSPs. Physical paths can be determined by an off-line configuration program, but the benefits of using an online method of QoS routing such as CBR are significant. The forwarding states are installed across the network using RSVP signaling.

From an ISP's point of view the network administrator configures the LSP based upon individual constraints and then the network — using CBR — determines the optimal path for the collection of all LSPs given these constraints. Together MPLS and CBR are valuable tools for traffic engineering and provide the full specification of the constraints.

4.4 Conclusions

The need for QoS capabilities in the Internet stems from the fact that best-effort service and datagram routing do not meet the needs of many new applications, which require some degree of resource assurance in order to operate effectively. Diverse customer requirements also create a need for service providers to offer different levels of services in the Internet.

The Internet community has developed a number of new technologies to address these issues. Integrated Services and Differentiated Services provide new architectures for resource allocation in the Internet. Integrated Services use reservation to provide guarantee resources for individual flows. The Differentiated Services architecture takes a different approach. It combines edge policing, provisioning and traffic prioritization to provide different levels of services to customers.

MPLS and traffic engineering address the issues of bandwidth provisioning and performance optimization in Internet backbones. The explicit route mechanism in MPLS adds an important capability to the IP-based network. Combined with constraint-based routing in traffic engineering, MPLS and traffic engineering can help network providers make the best use of available resources and reduce costs.

We can conclude that even if the strict QoS requirements imposed by future fixed and mobile applications on networks are not completely satisfied by the existing IP QoS solutions and pending or open QoS signaling issues in the existing QoS solutions still exist, basic mechanisms are defined and they will be deployed more and more intensively as the QoS requirements of customers increase.

5 Application oriented QoS

The concept of QoS has emerged when it has been observed that some application requirements cannot be met by traditional best-effort Internet. Several solutions have been proposed that address this problem and they are slowly being deployed. It is very important in such circumstances to be able to determine the influence that QoS characteristics of a link have upon an application.

The first step is to define the QoS parameters that are to be measured. Since most applications operate on IP level or higher, we shall focus in what follows on IP QoS. There are two bodies that work on defining IP QoS parameters. One is the Internet Engineering Task Force (IETF) and the other is the International Telecommunication Union, and its Telecommunication Standardization Sector (ITU-T).

QoS requirements, actions and responses must all be made in terms of QoS parameters. ITU-T adopts statistical, probabilistic definitions for the QoS parameters. On the other hand IETF has a deterministic philosophy when defining IP Performance Metrics (IPPM). It is argued that definitions of metrics in terms of probability may involve hidden assumptions about the stochastic model of the behavior being measured, so IETF favors the definition of metrics in deterministic terms.

Secondly, specific application requirements regarding QoS must be formulated and the relationship between QoS parameters and the User Perceived Quality (UPQ) must be determined. Objective values have been provided by ITU-T for several application classes.

5.1 QoS parameters

Work in [ITU-350] identifies three distinct basic functions of communications, namely access, user information transfer and release, and three performance criteria: speed, accuracy and dependability. A 3 by 3 matrix method is used to systematically identify and organize the Parameters of Performance (PPs), both for QoS and Network Performance. The PPs may be primary, such as delay, or derived, such as availability of the system. The derived PPs are obtained via threshold interrupts on the primary ones. The 3 by 3 matrix method produces nine generic, primary PPs. Throughout this section we shall focus on the PPs related to the user information transfer which are summarized in Table 2, together with their corresponding primary QoS parameters. Out of these primary PPs, the availability of the service may be derived.

Performance Criteria	*QoS Parameters*
Speed	delay, throughput
Accuracy	probability of error
Dependability	probability of loss

Table 2 — ITU-T performance criteria and QoS parameters relative to user information transfer.

5.1.1 Statistical QoS parameters: I.380

In [ITU-380], the same 3 by 3 matrix method defined in [ITU-350] is used. Due to the connectionless nature of IP, as well as to the possibility of packet fragmentation, some differences are noted with respect to ATM, as detailed in the following IP packet transfer outcomes:

a) *Successfully transferred* — the IP packet reaches destination within T_{max}, with valid header(s) and no errors in the binary contents. This refers to all the fragments of an IP packet, if any.

b) *Errored* — the IP packet reaches destination within T_{max} with either wrong binary contents or one or more corrupted header(s) fields. Let us note that errored headers that cause discarding or misdirection account for IP packet loss; otherwise, an errored outcome is observed.

c) *Lost* — the IP packet is never delivered to destination or delivered beyond T_{max}. If any fragment of an IP packet is lost, the whole original packet is considered lost.

d) *Spurious* — the IP packet creates an egress event for which there was no corresponding ingress event.

The connectionless nature of IP implies that:

- an IP packet may leave a basic section to any permissible egress Measurement Point (MP);
- an IP packet may be fragmented and fragments may be routed differently;
- a packet or a fragment may be routed back to a basic section it left due to a change in routing table.

I.380 performance parameters

The IP packet transfer PPs are:

a) *IP Packet Transfer Delay (IPTD)* — it is defined for all successful and errored outcomes as $t_{receive}$ - t_{send}, where $t_{receive} > t_{send}$ and $t_{receive}$ - $t_{send} \leq T_{max}$. In case of fragmentation, $t_{receive}$ corresponds to the last fragment.

$$IPTD = t_{receive} - t_{send}; \quad t_{receive} > t_{send}, \; t_{receive} - t_{send} < T_{max}$$

b) *Mean IPTD* — the arithmetic average of IPTDs for a population of interest. Other statistical metrics can be used too, such as percentiles.

c) *IP Packet Delay Variation (IPDV)* — the 2-point IPDV for an IP packet is the difference between its IPTD and a reference IPTD. The reference IPTD can be, for example, the mean IPTD or the IPTD for the first packet.

$$IPDV = IPTD - IPTD_{reference}$$

d) *IP Packet Error Ratio (IPER)* — the ratio of total errored IP packet outcomes ($N_{errored}$) to the total successful IP packet outcomes ($N_{successful}$) plus errored packet outcomes.

$$IPER = \frac{N_{errored}}{N_{successful} + N_{errored}}$$

e) *IP Packet Loss Ratio (IPLR)* — the ratio of total lost IP packet outcomes (N_{lost}) to the total transmitted IP packets ($N_{transmitted}$). Related to this is service availability, which is defined to be one if IPLR exceeds a certain threshold (the suggested value is 0.75).

$$IPLR = \frac{N_{lost}}{N_{transmitted}}$$

f) *Spurious IP Packet Rate* — the total number of spurious IP packets observed during a time interval ΔT divided by ΔT.

g) *IP Packet Throughput (IPPT)* — the total number of successful IP packet transfer outcomes during a time interval ΔT divided by ΔT.

$$IPPT = \frac{N_{successful}}{\Delta T}$$

h) *Octet based IP Packet Throughput (IPOT)* — the total number of octets transmitted in successful IP packet transfer outcomes during a time interval ΔT divided by ΔT.

5.1.2 Deterministic QoS parameters: IPPM

The aim of IPPM [RFC-2330] is to develop a set of standard metrics that provide unbiased quantitative measures of quality, performance and reliability of operational Internet data delivery services. It shall provide end users, network operators and service providers of the Internet with a common understanding and accurate measurement of the performance and reliability of the Internet unidirectional end-to-end paths and IP clouds. For better insights, we rely on the Surveyor Project [Sur-**], an operational measurement infrastructure that measures end-to-end delay, loss and routing information throughout the Internet using IPPM.

Framework for IPPM

The following topics are to be addressed for every single QoS parameter to be defined [RFC-2330].

a) *Defining a Metric*

The metric is a specified quantity, related to the performance and reliability of the operational Internet and given in standard units of measurements (e.g. unit of information is bits). Metrics are to be defined in deterministic, rather than stochastic, probabilistic terms. Definitions in terms of probabilities may involve hidden assumptions about the stochastic model of the behavior being measured (e.g. correlated packet loss).

Unless explicitly stated, the definition of a metric applies to standard formed packets, with *B* octets length (as given in the header), which includes the header and the payload, and a valid header with a correct checksum. It should not be an IP fragment and does not contain IP options. It should also have a valid transport header with correct checksum and fields, if any.

The value of the metric depends on the type of the IP packet used to make the measurement, and is thus referred to generically as "a packet of type P", where the type should be mentioned. In Surveyor, the test packets used for loss and delay are 12-byte-long UDP packets with a Sequence Number (SN) and a timestamp.

b) *Types of Metrics*

Metrics fall into two categories:

- Analytical Metrics — owing to the existent, rich analytical framework (referred to as the A-Frame), it is important to generate network characterizations consistent with both the A-Frame and the practical settings. This enables non-empirical network studies correlate and understand real network behavior by abstracting the properties of the Internet components relevant to given analytic metrics (e.g. transmission speed in a router modeled as a FIFO queue).

- Empirical Metrics — when a relevant metric fails to fit into the A-frame because the analytical model lacks the detail and power of the real system, an empirical metric may be defined instead, along with a reference, effective measurement methodology (e.g. best flow capacity achievable along a path using RFC-2001 compliant TCP).

c) *Compositions of Metrics*

Metrics have two forms of composition:

- Spatial Composition — a metric can be applied to a complete path P and/or various sub-paths of P (e.g. total delay versus delay in some links).
- Temporal Composition — a metric can be applied at a time t_i and/or various other times t_j (e.g. flow capacity during five minutes versus flow capacity in another five minute duration).

Generally, compositions allow some form of extrapolation.

Instances of metrics

Metrics can be classified according to instances. Instances refer to the number of times a measurement is performed. Note that the quantity itself being measured may involve an arbitrary number of events.

a) *Singleton Metric*

This is a single (atomic) instance of measurement.

b) *Sample Metric*

It involves a number of distinct instances of singleton metrics. Samples give an indication on the variations and consistencies of the metric. Samples are collected in at least three ways:

- Periodic Sampling, i.e. at fixed time intervals. This method is simple but suffers from at least three shortcomings: i. the metric itself may exhibit periodicity too, ii. it is easily anticipated and iii. it may derive the network into a state of synchronization.
- Random Additive Sampling, i.e. independent, randomly generated intervals obeying to a distribution G(t). It avoids synchronization, but it may complicate frequency domain analysis (Fourier transform techniques assume that samples occur at fixed intervals) and it may be predictable unless G(t) is the exponential distribution.
- Poisson Sampling, i.e. G(t) is the exponential distribution, which is not predictable. The sampling is unbiased and asymptotically unbiased, even if the network state is affected by biasing. However, the exponential distribution is unbounded and may occasionally generate lengthy sampling intervals. In this case, uniform sampling can be used

instead. The value of the mean sampling rate is not constrained, except to note the extremes: if the rate is too large, then the measurement traffic will disturb the network and if the rate is too small, some interesting network behavior may not be captured.

Other sampling methods include Geometric Sampling.

c) *Statistical Metric*

This is a metric derived from a sample metric via computation of statistics of the values defined by the singleton metric on the sample (e.g. mean).

d) *Statistical Distributions*

Statistical distributions, such as percentiles, are yet another method for describing samples. This is achieved through an Empirical (Cumulative) Distribution Function (EDF)*, which is preferred to histograms which give no indication on time occurrences.

IPPM metrics

The main IPPM metrics are presented next.

a) *One-way delay* [RFC-2679] — is preferred to round-trip delay due to the asymmetry that may be found in paths, queuing (even if paths are symmetric), flows and resource provisioning and reservations. It is defined as the difference in time between the occurrence of the first bit of a packet at the transmitter and the last bit at the destination.

The methodology shall tell for what upper bound on delay the packet is deemed lost. In Surveyor, a value of 10 seconds has been retained. Note that if the packet is fragmented and if, for whatever reason, reassembly does not occur, then the packet will be deemed lost. One-way delay may be measured via timestamps. The possible statistics are percentile, median, minimum and inverse percentile.

$$delay = t_{receive} - t_{send}$$

b) *Round-trip delay* [RFC-2681] — the time difference between the occurrence of the first bit of a request packet at the transmitter and the reception of last bit of the response packet at the transmitter, considering that the destination receives the request and sends the response without any delay.

c) *Instantaneous Packet Delay Variation (ipdv)* [Dem-02] — for two consecutive packets is defined as the difference between the one-way delays

* Consider a number N of observations $X_1 ... X_N$. For a value x, an EDF F(x) is n(x)/N, where n(x) is the number of points X_i less than x. A percentile is the smallest value of x for which F(x) is less than or equal to a given percentage. Note that if N is odd, the 50% percentile equals the median.

for the second and the first packet. For jitter, which is considered to be a fairly vague notion, it is recommended to use the absolute value of *ipdv*.

$$ipdv = delay_m - delay_n$$
$$jitter = |ipdv|$$

d) *One-way packet-loss* [RFC-2680] — is defined to be 0 when a packet is received within a finite one-way delay. Again, a given methodology will have to include a way to distinguish between a packet loss and a very large (but finite) delay by selecting an appropriate "reasonable" threshold for the one-way delay. Note that corrupted packets, even if received, are counted lost. Also, if the packet is fragmented and if, for whatever reason, reassembly does not occur, then the packet will be deemed lost.

Let us note that, unlike this metric, loss rates at transport layers do not reflect unbiased samples. TCP transmission for instance occurs in bursts (and induces loss) and adapts its transmission according to loss. In Surveyor a Poisson stream with rate 2 is used to measure loss and delay.

Although packet loss average is mentioned in [RFC-2680], IETF prefers a more deterministic approach and [Koo-99] defines some patterns of loss. A bursty loss involves loss of consecutive packets of a stream. The loss distance is the difference in SN of two successively lost packets which may or may not be separated by successfully received packets. A loss period is a period that starts with the loss of a packet (given that the previous packet was successfully received) and ends at the last lost packet (given that the next packet is successfully received). The possible statistics are "noticeable loss rate" (a noticeable loss appears if the loss distance is smaller than a certain threshold), "loss period total", "loss period lengths" and "inter loss period lengths".

e) *Bulk Transfer Capacity (BTC)* [RFC-3148] — a measure of a network's ability to transfer significant quantities of data with a single congestion-aware transport connection (e.g. TCP). It is defined as the ratio between the amount of unique data bits transferred (i.e. not including header bits) and the elapsed time for the corresponding transfer.

$$BTC = \frac{amount\ of\ data}{transfer\ time}$$

f) Other metrics include: connectivity [RFC-2678] and network performance measurement for periodic streams [Rai-00].

As it can be seen, IETF IPPM and ITU-T I.380 are very consistent except for the fact that errored packets are distinguished from lost ones in I.380, which may be useful for applications that are able to distinguish between

them, such as IP telephony. In addition, ITU-T doesn't give so precise information regarding the sampling methods.

The ITU-T's on-going work on the definition of QoS in IP is a combination of its view to QoS in ATM and IntServ. The convergence of the two approaches is achieved by quantifying the bounds on the QoS parameters that are to be achieved through different packet delivery services. Each QoS class creates a specific combination of bounds on the performance values (see Section 5.2.5). Let us note that the performance objectives are given in statistical, probabilistic terms. For example the IP packet transfer delay objective is an upper bound on the mean IPTD; this means that some packets will in effect have transfer delays exceeding this value.

5.1.3 QoS measurements

IPPM work on measuring metrics is very thorough and concrete and is already in application and use by several projects, such as Surveyor. By comparison, ITU-T's work appears theoretical and vague in this issue.

IPPM and Surveyor

a) *Measurement Methodologies*

At least four measurement methodologies are identified:

- Direct measurement of a metric using injected test traffic.
- Projection of a metric from lower-level measurements. Let us note that the possible mis-mapping between the lower-level layers is not considered.
- Estimation of a constituent metric from more aggregated measurements. Again, no mention is made to the possible biases that may occur in this case.
- Estimation of metric at time t_1 from related metrics at time t_2.

Four important properties for any methodology are: i. repeatability, ii. continuity, iii. self-consistency and iv. goodness of fit. Continuity is defined as follows: a small variation in conditions results in a small variation in the resulting measurements. The metric itself is termed continuous in this case. Testing for goodness-of-fit of a distribution may be performed through the Anderson-Darling EDF test [RFC-2330], for instance.

Uncertainties and errors should be minimized, well documented and quantified.

b) *Issues Related to Time*

For a clock one may determine its offset, skew, drift and resolution.

Likewise, two clocks may be compared from the point of view of relative offset or relative skew.

Internet measurements are often performed by Internet hosts which introduce their own hardware effects on the measurements and yield the so-called host time. For a given packet P, a given host H and a given link L, we define the wire arrival time of P at H on L as the first time T at which a bit of P appeared on H's measurement point on L. Wire exit time of P at H on L is defined as the first time T at which all bits of P appeared on H's measurement point on L.

Some techniques have been suggested for differentiating between wire time as opposed to host time (e.g. 'scheduled' times may correct 'actual' times). In Surveyor measurements are made at the kernel level.

ITU–T

Measuring IP performance is vague in [ITU-380] as it is still for further study. So far, all it is said is that the following conditions should be documented:

- exact sections and portions of the network being measured;
- measurement time and the length of the samples;
- exact measurement traffic characteristics, including IP packet size and traffic pattern;
- type of measurement, whether in-service or out-of-service, active or passive;
- summary of measured data including means, worst-cases and empirical percentiles.

Other Projects

Among the other projects for measuring metrics, RIPE (Réseaux IP Européen) TTM (Test Traffic Measurements) [Rip-**] is the European equivalent of Surveyor. Other projects, such as the QBone [QBo-**] in the US and TF-TANT [TFT-**] in Europe target the measuring of DiffServ through the measurement of underlying PPs.

5.2 Application requirements

What drives the development of QoS techniques are applications. This is why it is very important to look at applications from the network side and correlate the network QoS parameters with User Perceived Quality (UPQ). Concrete information on how perceived quality* of an application changes under varying network QoS parameters is a very important issue. In this section I shall provide the information that is currently available and will direct our future research.

It seems natural to try to divide network applications into classes, allowing thus to deal with generic classes instead of individual applications. One can distinguish the following classes:

- unidirectional applications;
- unidirectional applications with time constraints;
- bidirectional applications;
- bidirectional applications with time constraints.

5.2.1 Unidirectional applications

In this class are included applications which are predominantly unidirectional. Typical examples are TCP based applications, such as those relying on FTP (file transfer) or HTTP (web access).

For FTP for example, which is a case of bulk transfer, delay parameters are not particularly important, although time-to-finish may be one of the parameters used to judge the perceptual quality. However packet loss influence is quite significant, first of all because of its effect on time-to-finish and, secondly, because of its influence on the efficiency of bandwidth utilization. Average goodput (total number of information bits received at the destination divided by transfer time) is thus an important metric.

Even though HTTP is in many regards similar to FTP, since the requests have usually small dimensions compared to responses, the delay seems to be more important from user's point of view given the interactivity of web applications. Although no real-time constraints are generally present, the response time of a web server can be considered as an important UPQ parameter.

5.2.2 Unidirectional applications with time constraints

In this category, the typical example is streaming. Such an application,

* Perceived quality measurement is application dependent and must be suitably defined.

be it pure audio or video (e.g. MPEG-2), has quite a strong dependence on packet loss. Delay is also important, but can always be compensated by using larger reception buffers. The one-way packet loss must be of the order of 10^{-5} in order to get VHS video quality. Video streaming has also relatively important bandwidth requirements (see Table 3).

Application	***Bandwidth requirements [Mbps]***	***MPEG Version***
Video conference	<0.384	MPEG-4
Video in a window	<1.5	MPEG-1
VHS quality (full screen)	1-2	MPEG-2
Broadcast NTSC	2-3	MPEG-2
Broadcast PAL	4-6	MPEG-2
Professional PAL	8-10	MPEG-2
Broadcast HDTV	12-20	MPEG-2
Professional HDTV	32-40	MPEG-2
Raw NTSC	168	*uncompressed*
Raw PAL	216	*uncompressed*
Raw HDTV	1000-1500	*uncompressed*

Table 3 —Video streaming bandwidth requirements.

5.2.3 Bidirectional applications

Bidirectional applications are those applications for which both communication directions are equally important. Some examples are: NFS (Network File System), DNS (Domain Name System), small HTTP transfers, remote access (telnet, VNC). Generally these are not real-time applications, but certain interactivity requirements are still made whenever a human user is directly involved.

Note that for VNC there exist also graphical versions, case in which the amount of data that travels in the two directions is asymmetric and the application becomes mainly unidirectional.

5.2.4 Bidirectional applications with time constraints

This type of applications is probably the most spectacular and the most demanding in terms of strain imposed on networks. These applications are running over RTP/UDP streams (for example H.323). Based on the amount of data transferred (i.e. bandwidth requirements) there are two main classes:

- applications with relatively low bandwidth requirements (e.g. IP telephony);
- applications with relatively high bandwidth requirements (video teleconferencing).

IP telephony or Voice over IP (VoIP) is one of the applications that begins to be used more and more, driving thus the efforts for providing QoS. Due to its interactivity users are quite sensible to small variations of network QoS parameters. Test results, based on subjective perceived quality, showed that a delay of 0-150 ms means good interactivity, 150-400 ms is tolerable and delays of 400 ms or higher give a bad interactivity. It also seems that packet loss is somewhat more important than jitter, which in its turn is more important than delay.

Determining the UPQ for VoIP is not a trivial matter also due to the coexistence of many codecs: G.711, G.723 etc. (see Table 4). Each of them has specific characteristics; in addition bandwidth requirements are slightly increased due to IP overhead. ITU-T has developed a framework for subjective and objective evaluations of speech quality, based on perceptual audio quality, that can also be used for evaluating VoIP user perceived quality [ITU-G.107], [ITU-P.800], [ITU-P.861], [ITU-P.862], such as the Mean Opinion Score (MOS).

Video teleconferencing (VTC) is a similar application which sends in addition video data. Several standards also exist in this field: H.323, H.332, H.261, H.263 etc. Demands for the audio quality are slightly lower, because of the presence of visual information, but a new problem appears: lip synchronization between video and audio which should be in the order of 1 to 2 video frames (around 50 ms). The bandwidth requirements are also somewhat bigger than for VoIP, but not so important as for video streaming. An ITU recommendation that could be used for an objective evaluation of UPQ in this case is [ITU-J.143].

Compression standard	*Bandwidth requirements [Kbps]*	*Compression delay [ms]*	*Mean Opinion Score [0-5]*
G.711	64	0.75	4.4
G.723	5.3	~30	3.5
G.723.1	6.3	30	3.98
G.726	16-40	1	4.2
G.728	16	2.5-5	4.2
G.729	7.9-8	10	4.2

Table 4 — VoIP compression overview.

5.2.5 ITU-T performance values

ITU has tried to establish bounds on network QoS parameters that are supposed to ensure good UPQ for several classes of applications [ITU-1541]. This IP performance values should be achieved for the IP performance parameters by use of appropriate network techniques (constrained routing etc.) and queue management (separate queues, drop priority etc.).

The six QoS classes are:

- 0 and 1 — real-time jitter sensitive (highly) interactive applications (e.g. VoIP, VTC);
- 2 and 3 — transaction data, (highly) interactive applications (e.g. signaling)
- 4 — low loss only applications (e.g. short transactions, bulk transfer, video streaming);
- 5 — traditional best-effort applications.

In the table below are provided, for several applications, the IP performance values for ITU performance parameters:

	Real-time applications	*VoIP*	*WWW, best-effort services*	*Streaming video (VHS quality)*
IPTD	150 ms	400 ms	Undefined	400 ms
IPDV	50 ms	50 ms	Undefined	17 ms
IPLR	10^{-3}	10^{-3}	Undefined	10^{-5}
IPER	10^{-4}	10^{-4}	Undefined	10^{-4}

Table 5 — ITU-T IP performance values for some applications.

The values indicated are generally upper bound on average values, except for IPDV where the upper bound is on the $1\text{-}10^{-3}$ quantile of IPTD minus the minimum IPTD.

6 Experimental results

We have performed several experiments aimed at determining the QoS characteristics of the basic element of computer networks — the switches. We have focused on the fundamental QoS mechanisms switches provide, namely queue scheduling algorithms.

Although QoS measurement techniques seem to be quite well defined when it comes to networks, switch testing implies the determination of many characteristics which influence the results. Most of these characteristics (for example head-of-line blocking*) depend directly on the switch architecture and alter significantly QoS parameters. Switch architectures may also interfere with and hinder proper functioning of basic scheduling mechanisms. The QoS parameters that were analyzed were latency and throughput/packet loss (by means of the number of transmitted and received packets).

We have run several tests on a switch with 12 Gigabit Ethernet ports. The frame size that was generally used is 1518 bytes, except for certain tests when the difference between the way in which scheduling algorithms handle packets of different sizes had to be emphasized.

This switch can differentiate packets in 4 priority queues: txq0, txq1, txq2 and txq3; txq3 corresponds to the highest priority. The available queue scheduling algorithms are:

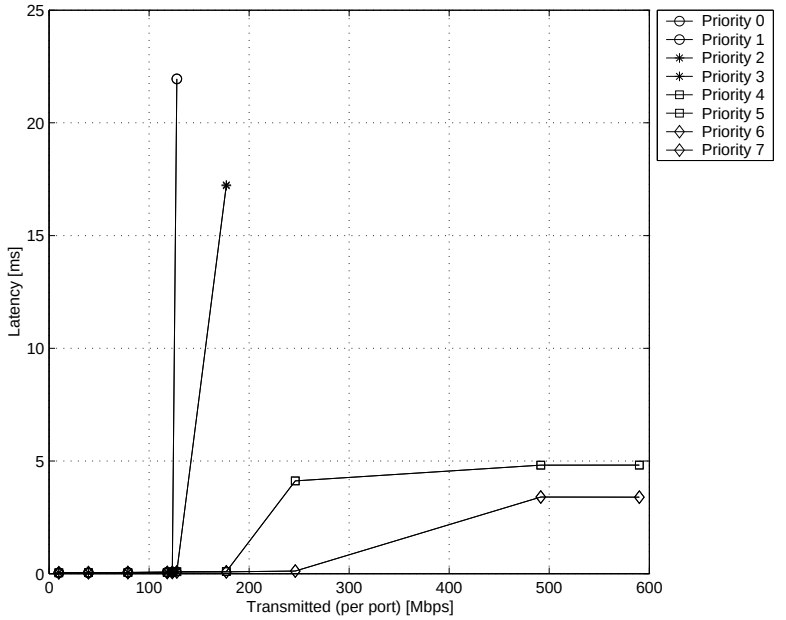
1) Strict Priority (SP) scheduling;
2) Weighted Round Robin (WRR) scheduling on a per packet or per 256-byte quanta basis;
3) Hybrid 1 scheduling (txq3 is serviced in SP manner and the other queues in WRR order);
4) Hybrid 2 scheduling (txq3 and txq2 are serviced in SP manner, the other queues in WRR order).

The priorities are extracted from VLAN tags, according to the following rules:

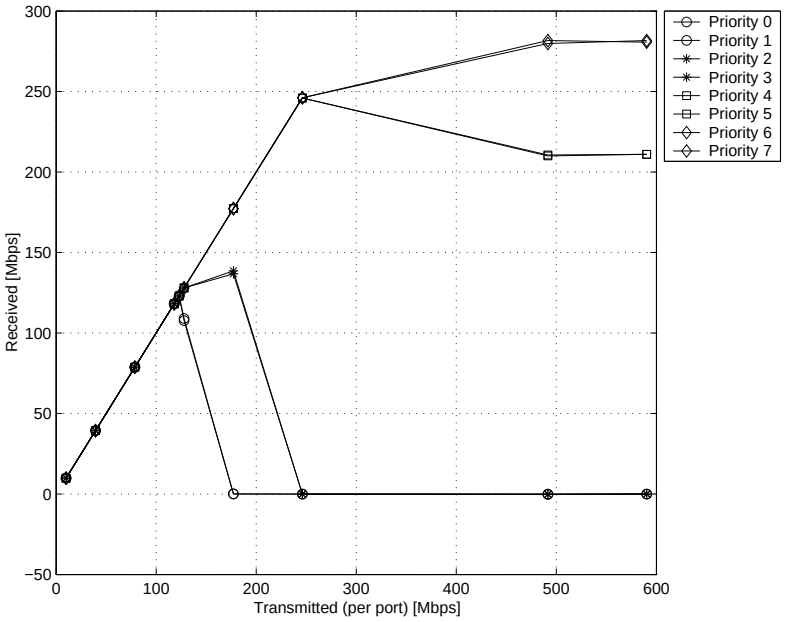
- VLAN priorities 0 and 1 ⇒ txq0;
- VLAN priorities 2 and 3 ⇒ txq1;
- VLAN priorities 4 and 5 ⇒ txq2;
- VLAN priorities 6 and 7 ⇒ txq3.

* Head-of-line blocking (HOL) causes some packets to remain in the input queues, because the destination is busy, thus preventing the next packets from being delivered, even if their destinations are free. It has been shown that throughput cannot exceed approximately 60% of backplane capacity if HOL blocking is present.

Strict priority queuing, if it's not used with other traffic shaping mechanisms, can lead to starvation of lower priority flows, as it is observable in Figure 6. However, behavior of the tested switch is different from theoretical SP, probably due to its architecture, which uses a backplane to connect switch ports in a fully-meshed way. For example, txq2 is not starved even if txq3 input reaches 1Gbps.



(a)



(b)

Figure 6 — Average latency (a) and throughput (b) for Strict Priority queue scheduling (frame size = 1518 bytes).

When using WRR scheduling at packet level, the fairness holds if all packets on all priorities have the same size (see Figure 7). The observed weights agree quite well with the expected ones (see Table 6). Let us note that the ratios of latencies are not proportional to the weights, as it happens for the throughput.

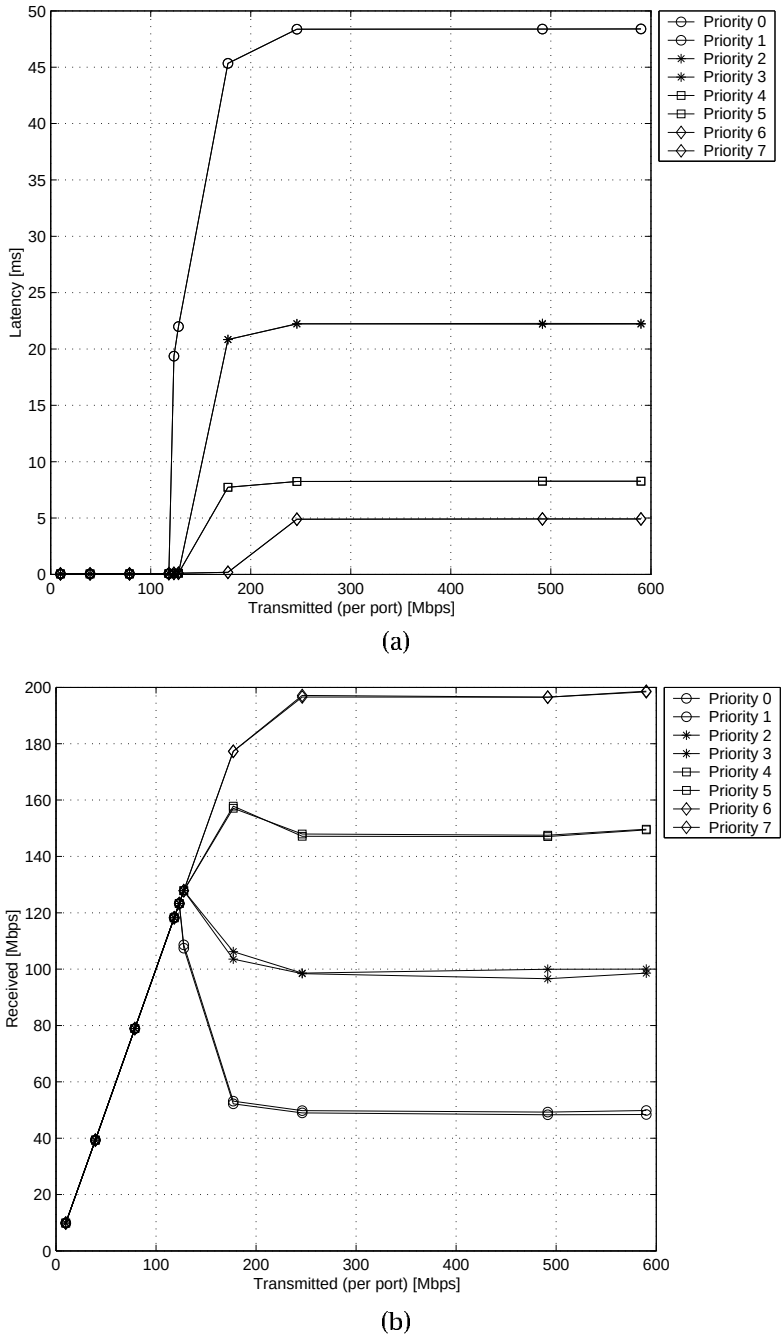
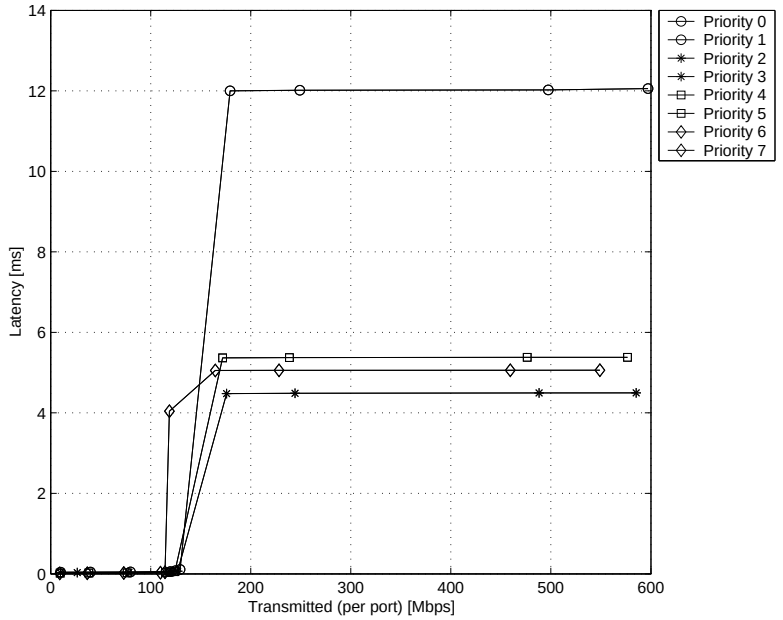
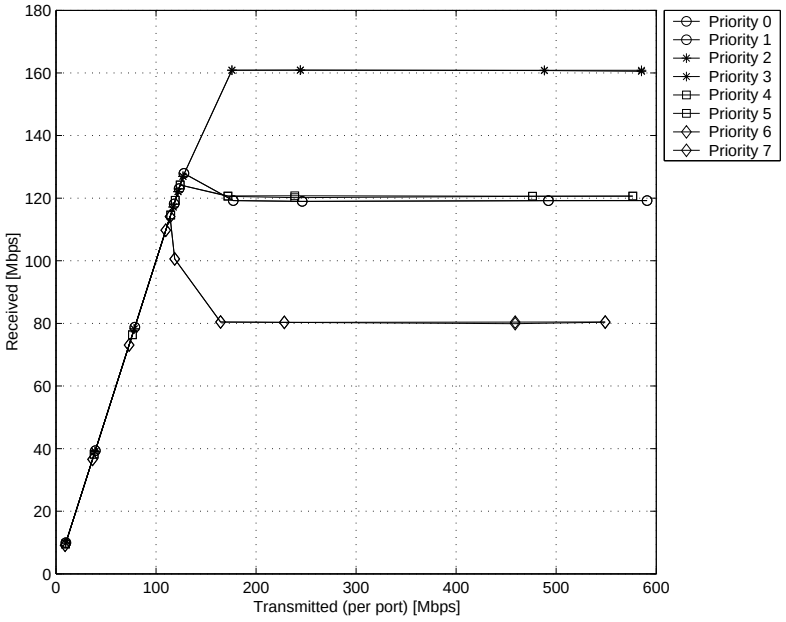


Figure 7 — Average latency (a) and throughput (b) for Weighted Round Robin queue scheduling on a per packet basis (frame size = 1518 bytes). The weights are 1, 2, 3 and 4 for txq0, txq1, txq2 and txq3, respectively.

On the contrary, when the packets have different sizes, WRR on a per packet basis is not fair anymore (see Figure 8) as far as throughput is concerned. Latencies change accordingly.



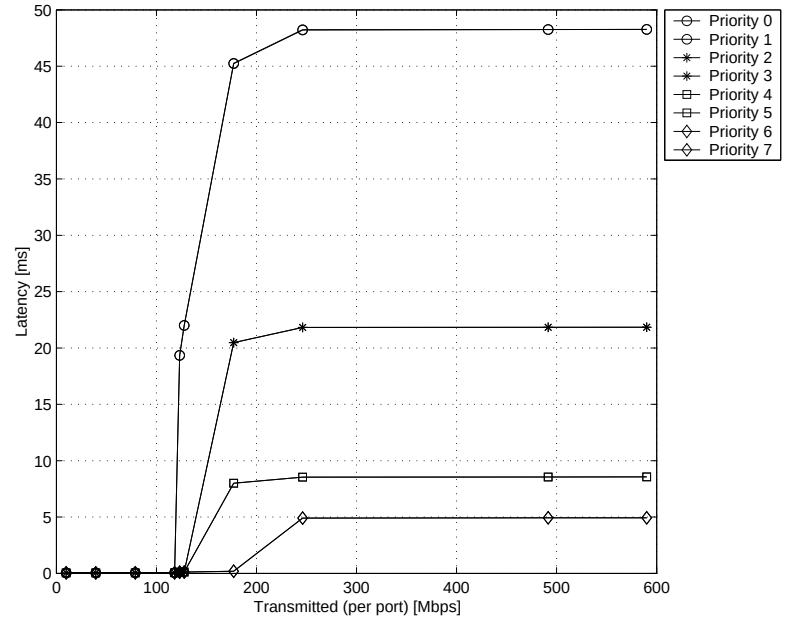
(a)



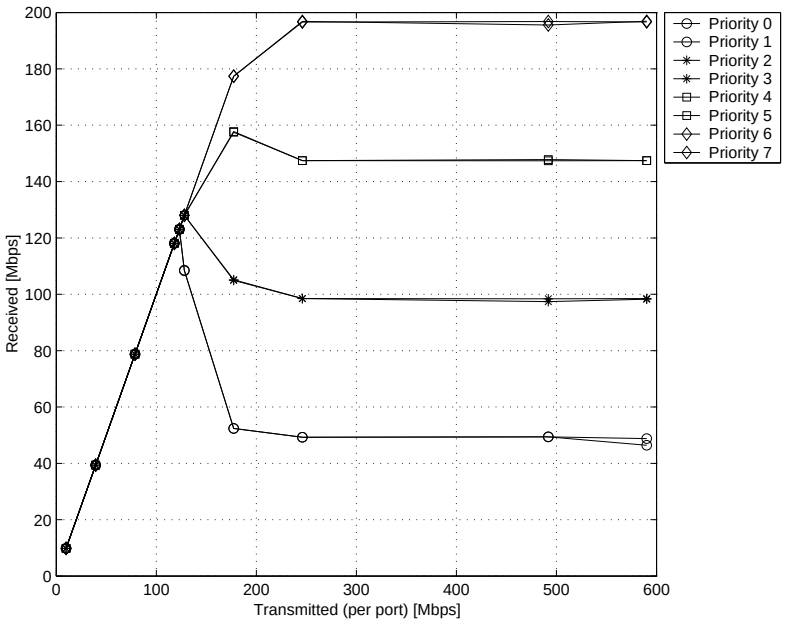
(b)

Figure 8 — Average latency (a) and throughput (b) for Weighted Round Robin queue scheduling on a per packet basis. The frame sizes are 1518, 1024, 512 and 256 bytes and the weights are 1, 2, 3 and 4 for txq0, txq1, txq2 and txq3, respectively.

In order to solve the problem of fairness one can use a version of WRR that uses quanta of 256 bytes when scheduling. In Figure 9 this scheduling mechanism is used for frames of equal length. The results are similar to those for WRR on a per packet basis.



(a)



(b)

Figure 9 — Average latency (a) and throughput (b) for Weighted Round Robin queue scheduling on a per 256-byte quanta basis (frame size = 1518 bytes). The weights are 10, 20, 30 and 40 for txq0, txq1, txq2 and txq3, respectively.

Figure 10 shows the results obtained when sending different size packets and using WRR on a per 256-byte quanta basis. Throughput ratios do not agree well with the expected weights (see Table 6). We consider this is due to packet processing overhead.

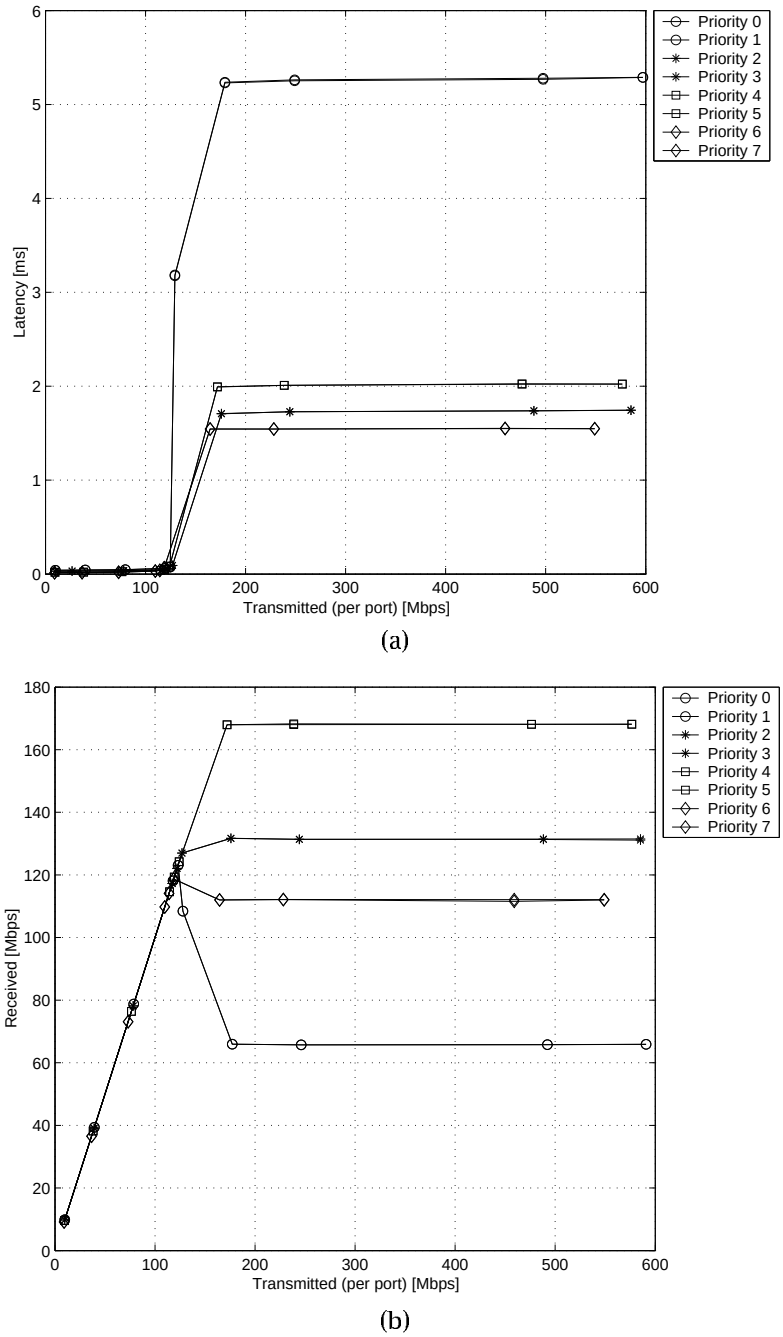
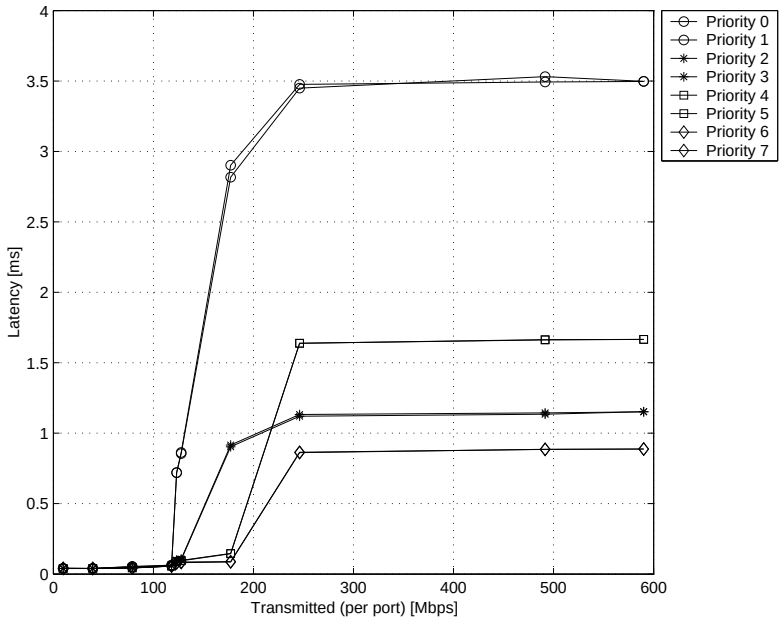
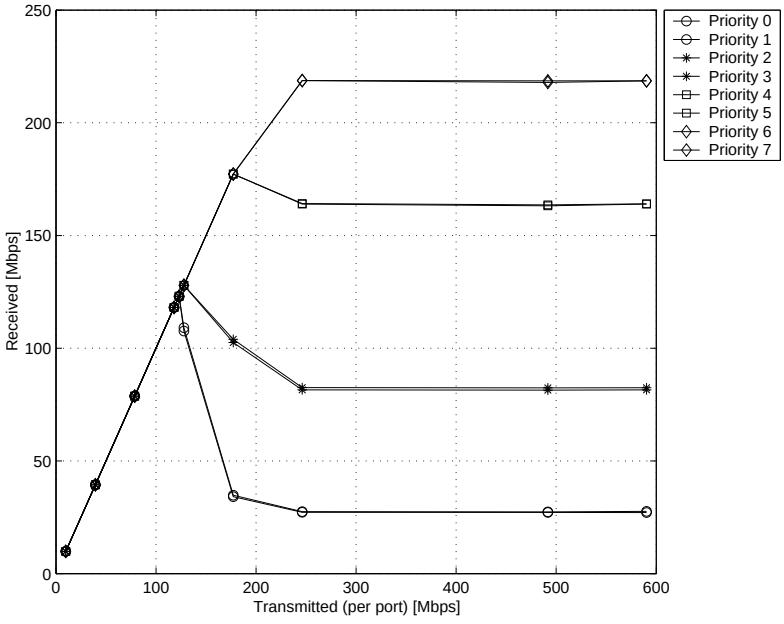


Figure 10 — Average latency (a) and throughput (b) for Weighted Round Robin queue scheduling on a per 256 byte quanta basis. The frame sizes are 1518, 1024, 512 and 256 bytes and the weights are 10, 20, 30 and 40 for txq0, txq1, txq2 and txq3, respectively.

The other scheduling techniques that are available are Hybrid 1 (see Figure 11) and Hybrid 2 (see Figure 12). Again the same problems with proper SP functioning are observed, causing txq3 not to receive exclusive service once it reaches 1Gbps.



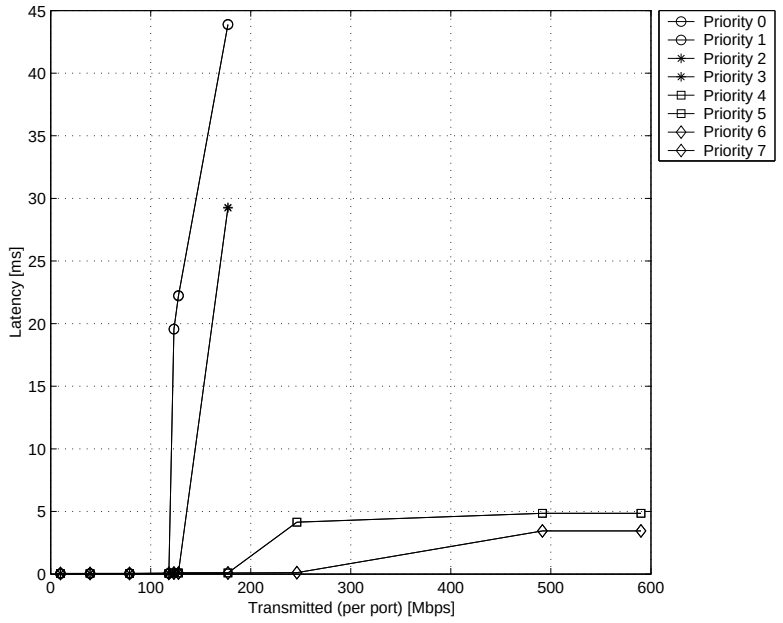
(a)



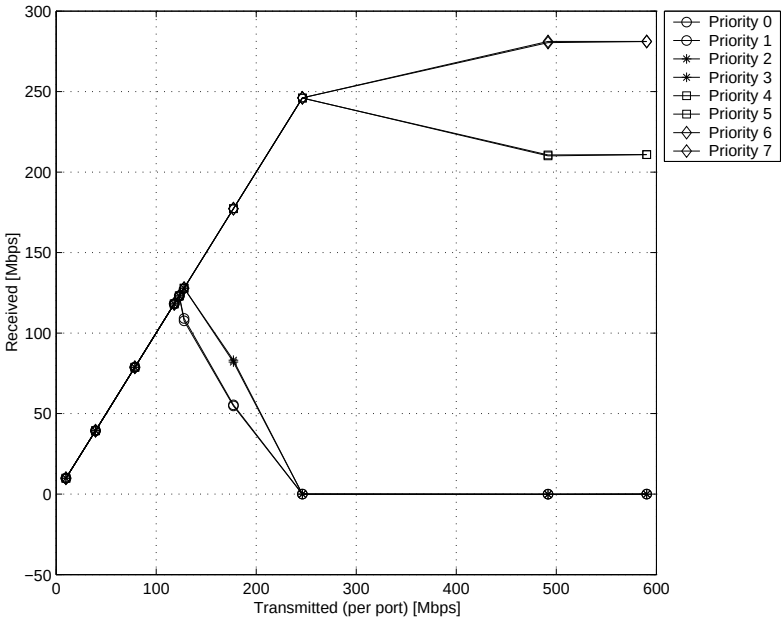
(b)

Figure 11 — Average latency (a) and throughput (b) for Hybrid 1 queue scheduling (frame size = 1518 bytes).

Due to the fact that in Hybrid 2 the two higher priorities are serviced in SP manner, starvation is again noticeable once the load exceeds a certain level, but service for high priorities is not correct, as indicated when discussing SP and Hybrid 1 results.



(a)



(b)

Figure 12 — Average latency (a) and throughput (b) for Hybrid 2 queue scheduling (frame size = 1518 bytes).

The table below presents the expected and observed throughput weights when using WRR scheduling, on a per packet and per 256 byte quanta basis. When the scheduling is done on a per-packet basis, then the weights refer to the number of packets; when 256-byte quanta are used, then the weights refer to the number of such quanta.

The observed and expected weights match very well when using only one packet size, for both versions of WRR. One can notice however a difference between expected and observed weights when frame sizes are different on distinct priorities. This was expected for per packet WRR, but not for per 256-byte quanta WRR, which should have provided better performance.

The weights indicated in the table below are computed by normalizing throughput to the sum of received frames at a load of 500 Mbps per port. If for WRR on quanta we compute the ratio to the number of received frames for txq0, then we obtain 20.3, 25.9 and 17.1, for txq1, txq2 and txq3, respectively. This is in relation with the decrease of packet sizes, which are 1518, 1024, 512 and 256 for txq0, txq1, txq2 and txq3, respectively. This seems to indicated that the overhead of having to process more packets causes the observed decrease in performance (in order to attain the same throughput, a number of 256 byte packets six more times bigger must be dealt with, than if using 1518 byte frames).

Scheduling mechanism	***Expected weights (txq0, txq1, txq2, txq3)***				***Observed weights (txq0, txq1, txq2, txq3)***			
WRR, per-packet (same packet size)	1.00	2.00	3.00	4.00	0.98	1.99	3.03	4.00
WRR, per-packet (different packet size)	1.00	2.00	3.00	4.00	2.45	3.35	2.52	1.68
WRR, 256 byte quanta (same packet size)	10.0	20.0	30.0	40.0	9.8	20.2	29.9	40.1
WRR, 256 byte quanta (different packet size)	10.0	20.0	30.0	40.0	13.6	27.7	35.3	23.4

Table 6 — Intended and observed weights for different variants of WRR scheduling.

7 Conclusions

The development of network based applications, especially those involving interactive or streaming media, is placing increasingly stringent demands on the delivered service of networks. The delivery of such requirements is, in general, easily met within a laboratory environment, but continuing to have a consistent delivered quality to applications once they are running over large scale networks remains a significant issue. One that can seriously jeopardize their large scale deployment, since users have certain minimum quality requirements.

Unpredictable performance has been the most significant problem in the deployment of IP-based networks. Demand for IP networks has exceed all predictions; yet the lack of quality of service (QoS) and methodology to traffic engineer these networks has been the single greatest impediment to their deployment. IP infrastructure has to go through a revolution in order to provide the network services and SLAs demanded by the industry. It is simply unacceptable to deploy best-effort networks for many of today's applications.

The issue is further complicated by the heterogeneous nature of networks involving a number of local area and access architectures (wired and wireless) and an equal number and variety of wide are architectures. The concatenation of these architectures must provide a framework for the delivered end-to-end IP service with mechanisms designed to meet customers' stringent service level requirements. Further, the nature of communications has expanded to include multicast networks, particularly in support of multimedia distribution services. Multicast QoS has yet to be realized on such networks.

IntServ, DiffServ and MPLS are all-important stepping stones in the evolution of a new IP infrastructure. Although ATM is the only networking service to offer classes of service, the realization that ATM would never be likely to be deployed end-to-end meant that new protocols and architectures had to be designed.

Important work is continually undertaken by the IETF and ITU working groups. In particular the Internet Traffic Engineering working group focuses on performance optimization of traffic, which involves the design, provisioning and tuning of IP networks. It also addresses issues such as constrain-based routing, resource allocation and the measurement of intra- and inter-domain traffic flows.

The DiffServ working group focuses on methodologies to provide classification of traffic flows to support various applications. This involves well-defined building blocks from which a variety of aggregate behaviors can be built including per-hop behavior and code-point specification.

The MPLS working group is responsible for standardizing a base technology for using label swapping over various link level technologies, such as packet-over-SONET/SDH, Frame Relay, ATM and IEEE 802 LAN architectures. Of significant importance is the provisioning of MPLS over WDM which has the potential to provide a very high speed service over a relatively simple architecture. These architectures must be scalable as well as supporting unicast and multicast traffic flows.

Important work still remains to be done in areas such as per-domain behavior, traffic conditioning, policy definition, infrastructure and enforcement, QoS routing and QoS multicasting. Finally, a security framework for many of these new architectures has yet to be designed. A consistent framework in which QoS is both defined and delivered, which permits the prediction of delivered QoS by networks and network equipment, not just during normal operation, but also during overload, is required.

Therefore we can conclude that QoS solutions are becoming mature, but their spreading is however slow, due to the lack of QoS control mechanisms in current routers. Nevertheless we believe that applications, whether run by corporate customers (IP telephony, video teleconferencing) or private users (web surfing, interactive games), will lead to the more and more extensive deployment of QoS in the near future.

Several open questions remain. First of all it's not at all clear at the moment how should QoS principles be mapped onto implementations. Edge-to-edge QoS solutions only give recommendations, leaving the final decision to manufacturers. Configuring devices QoS-enabled devices to provide the desired service is also an issue. Secondly, assuming certain solutions are implemented, one may wonder how well do they work. Our experience so far shows that switch architectures may cause a drift of the implementation behavior from theoretical principles.

There is also the question related to QoS measurements, especially at network level. How relevant such measurements can be, knowing that in an uncontrolled environment behavior at a moment may depend on everything that is happening with the network at that moment. A solution may be a kind of QoS voltmeter, that would permanently monitor the QoS level and check the SLAs, sending a warning whenever there is a disagreement between expected and observed QoS levels.

Another important topic is the study of the effects of QoS characteristics of a link upon applications, from a user perspective. The main goal of such an approach is to establish the relationship between QoS parameters and UPQ (User Perceived Quality) parameters for various types of applications. This would allow testing the QoS level of network connections with respect to specific applications.

QoS glossary

Access router

A router at a customer site, which connects to the network service provider WAN. Sometimes termed a Customer Premises Equipment (CPE) router.

Administrative Domain

A collection of network elements under the same administrative control and grouped together for administrative purposes. It is usually managed by a single corporate entity. For quality of service enforcement purposes, a network domain refers to any domain that shares a common QoS policy. It may or many not overlap other kinds of domains like IP or NT domains.

Admission Control

A policy decision applied initially to QoS requests for controlling the admission of network traffic from outside a given administrative domain. Admission Control is closely tied to accounting and relies on source authentication. Contrast with Policing, which occurs after a request is accepted and data is flowing.

Assured Forwarding (AF)

A specific DiffServ behavior, which divides IP packets into four separate per hop behavior (PHB) classes. Using these classes, a provider may offer different levels of service for IP packets received from a customer domain. Each Assured Forwarding class is allocated a specified amount of buffer space and bandwidth.

Asynchronous Transfer Mode (ATM)

A data framing and transmission architecture designed to carry voice, video and data, which has built-in QoS capabilities. ATM operates at Layer 2 of the OSI model and is a high-speed, connection-oriented, packet switching, multiplexing architecture. Bandwidth is divided into fixed-size cells of 53 bytes each, including headers, which are allocated to services on demand. Bandwidth can be dynamically allocated. ATM can offer bandwidth rates of up to multi-gigabit bandwidth. Although relatively few native ATM applications exist, TCP/IP traffic can be sent over an underlying ATM layer. In principle ATM could be used over the LAN, MAN or WAN; in practice, ATM is prevalent in WANs and Internet backbones. ATM services are likely to coexist with QoS-enabled IP networks for many years to come.

Autonomous System (AS)

A self-connected set of networks that are generally operated within the same administrative domain (see Administrative Domain).

Backbone network

A network linking together multiple domains, either enterprises or service provider. Each LAN is connected to the backbone via bridges or routers and the backbone facilitates LAN-to-LAN traffic. In addition to LANs, network segments, subnetworks, or individual devices can also be attached directly to the backbone.

Bandwidth

Transmission capacity of a computer channel or communications line or bus, usually stated in bits per second (bps). Bandwidth indicates the theoretical maximum capacity of a connection, but as the theoretical bandwidth is approached, negative factors such as transmission delay can cause deterioration in quality.

Bandwidth Manager (BM)

A traffic manager deployed at congestion points that limits access to network resources. It often requires locating a proprietary hardware device directly on the network and may be an additional point-of-failure. It cannot coordinate multiple traffic flows or resolve conflicting QoS requests made by multiple clients and is therefore not an end-to-end QoS solution.

Best-Effort Service

The default behavior of TCP/IP networks in the absence of QoS mechanisms. TCP/IP nodes will make their best effort to deliver a transmission but will drop packets indiscriminately in the event of congestion. The Internet today is a good example of best-effort service. Best effort is suitable for some network applications such as general file transfers or e-mail.

Border Gateway Protocol (BGP)

An Internet routing protocol used to pass routing information between different administrative routing domains or Autonomous Systems. BGP does not pass explicit topology information. BGP is often used between Internet service providers.

Border Router

Generally describes routers on the edge of an Autonomous System. Uses BGP to exchange routing information with another administrative routing domain. Can also describe any router that sits on the edge of a routing subarea, such as an OSPF area border router.

Bridging

In telecommunication networks, a bridge is a product that connects a local area network (LAN) to another local area network that uses the same protocol (for example, Ethernet or token ring). You can envision a bridge as being a device that decides whether a message from you to someone else is going to the local area network in your building or to someone on the local

area network in the building across the street. A bridge examines each message on a LAN, "passing" those known to be within the same LAN, and forwarding those known to be on the other interconnected LAN (or LANs).

In bridging networks, computer or node addresses have no specific relationship to location. For this reason, messages are sent out to every address on the network and accepted only by the intended destination node. Bridges learn which addresses are on which network and develop a learning table so that subsequent messages can be forwarded to the right network.

Bridging networks are generally always interconnected local area networks since broadcasting every message to all possible destinations would flood a larger network with unnecessary traffic. For this reason, router networks such as the Internet use a scheme that assigns addresses to nodes so that a message or packet can be forwarded only in one general direction rather than forwarded in all directions.

Class

An abstraction that can be determined by different policy criteria such as IP packet header content (e.g. source or destination addresses or port numbers or transport protocol), or time of day, ingress point, etc. The definition of a class can differ at different locations on the network.

Class-based Queuing (CBQ)

A public domain QoS methodology for classifying packets and queuing them according to criteria defined by an administrator to provide differential forwarding behavior for each traffic class. Packets are divided into a hierarchy of classes based on any combination of IP address, protocol, and application type. Each class is assigned a set of bandwidth priorities.

Class-based Fair Queuing (CFQ)

A per-class packet scheduling group of techniques that approaches link-sharing and real-time service requirements as simultaneous, and in some respect complementary, constraints at a gateway that can be implemented with a unified set of mechanisms [Flo-95].

Classifier

An entity which classifies packets based on the content of packet headers according to defined rules.

Class of Service (CoS)

A category based on type of user, type of application, or some other criteria that QoS systems can use to provide differentiated classes of service. The characteristics of the CoS may be appropriate for high throughput traffic, for traffic with a requirement for low latency or simply for best effort. The QoS experienced by a particular flow of traffic will be dependent on the number and type of other traffic flows admitted to its class.

Codepoint

Codepoint markings are made in a new implementation of the IP version 4 Type of Service (ToS) header called the DiffServ field and are used to select a per hop behavior (PHB). This marking takes place on the host or on a boundary or edge device.

Common Open Policy Service (COPS)

An IETF proposed standard defining a simple protocol for provisioning QoS by applying policy-based admission control over requests for network resources. COPS allows a policy server to control the devices on the network, such as routers and switches, so that a cohesive policy based on business priorities can be achieved. COPS is a companion protocol to RSVP. Future extensions will permit admission control information between policy servers and DiffServ clients.

Congestion

Situation in which there are too many packets present in the network, leading to performance degradation.

Congestion Avoidance

Congestion avoidance is the action a network takes to prevent congestion before it can occur, anticipating circumstances in which flows or aggregated flows might no longer receive designated service levels due to excessive traffic loads at points in the network. An example is the application of a drop policy such as RED to provide implicit feedback to host systems to reduce network traffic during congestion.

Congestion Control

Mechanisms that control traffic flow so that intermediate network devices and end stations are not overwhelmed.

Congestion Management

A mechanism at multiplexing points that imposes order when traffic exceeds network capacity for a flow or set of aggregated flows. It determines whether some packets must be discarded, and, if so, it preserves the more important packets. Queuing, scheduling and traffic shaping are among the most popular techniques.

Constant Bit Rate (CBR)

Multimedia streams audio and video are examples of CBR applications, since they send at a relatively steady data rate with constant bandwidth allocations. A class of service defined by ATM.

Controlled Load

Tightly approximates best-effort service under unloaded conditions — a high level but not guaranteed service. In the proposed IETF Integrated Service model, this level of service is designed for multimedia applications where time delay is not critical but quality of the delivery is important. This

service is appropriate for applications such as one-way voice or video, but not for real-time applications.

Convergence

In information technology, convergence is a term for the combining of personal computers, telecommunication, and television into a user experience that is accessible to everyone. In the U.S., an estimated 30% of homes have computers with modems. Virtually, 100% of homes have a TV set. Studies show a large populace of TV users who would embrace the Internet, video-on-demand and greater interaction with content, but who are diffident about buying and using a personal computer. For these reasons, both the computer and the television industries are embarked on bringing digital TV and the Internet to a larger market.

Converged Network

A network that combines varied traffic types such as data, voice and multimedia. Most analysts expect the converged network of the future to be based on Internet protocols. This trend is evident in corporate networks, which are starting to combine video teleconferencing on their traditional data networks, as well as in the merging of the telephone, cable television and Internet service industries.

Core

See Backbone Networks.

Core router

A router on the network service provider WAN that has no direct connections to any routers at customer sites.

Customer router

A router at a customer site that is not directly connected to the network service provider WAN.

Delay

See Latency.

Designated Subnetwork Bandwidth Manager (DSBM)

A device on a managed subnetwork that acts as the Subnetwork Bandwidth Manager (SBM) for the subnetwork to which it is attached. This is done through an election process specified in the IETF SBM protocol specification.

Differentiated Service (DiffServ)

A IETF standard for a small, well-defined set of per-packet building blocks from which a variety of services may be built, thereby providing a framework for delivering quality of service (QoS) in networks. This effort is largely focused on the use of the ToS field in IPv4 header as a QoS signaling mechanism, and it aims to provide definitions appropriate for aggregated

flows for any level of aggregation. At least two services have initially been defined under this effort: the "Assured Service" and the "Preferred Service", each with slightly different definitions of service that from a technical perspective might be called "engineered best effort". [RFC-2474] defines how to assign a class of service by setting the ToS field. DiffServ will be backward-compatible with current ToS field settings. DiffServ is expected to be used predominantly in IP backbone environments. With proper engineering, including edge policing, DiffServ can provide expedited handling appropriate for a wide class of applications, including lower delay for mission-critical applications and packet voice applications. DiffServ-capable routers need only track a small number of per-hop behaviors, and they service packets based on a single byte. Typically, DiffServ is associated with a coarse level of packet classification.

Differentiated Services Boundary

The edge of a DiffServ domain, where classifiers and traffic conditioners are likely to be deployed. A DiffServ boundary can be further sub-divided into ingress and egress nodes, where the ingress/egress nodes are the downstream/upstream nodes of a boundary link in a given traffic direction. A DiffServ boundary may be co-located with a host, subject to local policy.

Differentiated Services Domain

A contiguous portion of the Internet over which a consistent set of DiffServ policies are administered in a coordinated fashion. A DiffServ domain can represent different administrative domains or autonomous systems, different trust regions, different network technologies (e.g., cell/frame), hosts and routers, etc.

Differentiated Services Field

The IPv4 header TOS octet or the IPv6 Traffic Class octet when interpreted in conformance with the definition given in [RFC-2474].

Digital Subscriber Line (DSL)

DSL is a technology for bringing high-bandwidth information to homes and small businesses over ordinary copper telephone lines. xDSL refers to different variations of DSL, such as ADSL (Asymmetric DSL), HDSL (High data rate DSL), VDSL (Very high rate DSL) and RADSL (Rate Adaptive DSL). Assuming your home or small business is close enough to a telephone company central office that offers DSL service, you may be able to receive data at rates up to 6.1 Mbps (of a theoretical 8.448 Mbps), enabling continuous transmission of motion video, audio and even 3-D effects. More typically, individual connections will provide from 1.544 Mbps to 512 Kbps downstream and about 128 Kbps upstream. A DSL line can carry both data and voice signals and the data part of the line is continuously connected. DSL is expected to replace ISDN in many areas and to compete with the cable modem in bringing multimedia and 3-D to homes and small businesses.

Dense Wavelength Division Multiplexing (DWDM)

Dense Wavelength Division Multiplexing is an optical technology used to increase bandwidth over existing fiber optic backbones. DWDM works by combining and transmitting multiple signals simultaneously at different wavelengths on the same fiber. In effect, one fiber is transformed into multiple virtual fibers. So, if you were to multiplex eight OC-48 signals into one fiber, you would increase the carrying capacity of that fiber from 2.5 Gbps to 20 Gbps. Currently, because of DWDM, single fibers have been able to transmit data at speeds up to 400Gbps. And, as vendors add more channels to each fiber, terabit capacity is on its way.

A key advantage to DWDM is that it's protocol and bit-rate independent. DWDM-based networks can transmit data in IP, ATM, SONET /SDH and Ethernet, and handle bit-rates between 100 Mbps and 2.5 Gbps. Therefore, DWDM-based networks can carry different types of traffic at different speeds over an optical channel.

From a QoS stand point, DWDM-based networks create a lower cost way to quickly respond to customers' bandwidth demands and protocol changes.

Early Packet Discard

A congestion-avoidance mechanism generally found in ATM networks.

Explicit Congestion Notification (ECN)

An addition to IP, whereby routers set a Congestion Experienced (CE) bit in the packet header of packets from ECN-capable transport protocols in order to signal congestion to end nodes.

Edge Device

A device such as a router or a gateway that is deployed at the border of an administrative domain. Such devices control traffic through one point only. Traditionally used to describe an ATM-attached host or router that interfaces with an ATM network switch.

Edge-to-Edge QoS

Edge-to-edge QoS applies to QoS within a network that connects to other networks rather than hosts or end systems (the typical service provider network, for example), with some level of control over bandwidth, jitter, delay and loss, provided by the network.

End-to-End QoS (e2e QoS)

The ability of a network to deliver service needed by a specific network application from end-to-end, with the ability to provide both class of service and reserved bandwidth for different types of network traffic. End-to-end QoS coordinates and enforces predefined traffic management policies across multiple network devices.

Expedited Forwarding (EF)

A per hop behavior (PHB) in the DiffServ standard, used to create a virtual leased line service.

Fault-tolerant

Fault-tolerant describes a computer system or component designed so that, in the event that a component fails, a backup component or procedure can immediately take its place with no loss of service. Fault tolerance can be provided with software, or embedded in hardware, or provided by some combination. Leading vendors that specialize in fault-tolerant systems include Compaq Non-Stop (formerly Tandem), Marathon Technologies and Stratus Computer.

In the software implementation, the operating system (for example, Tandem Guardian) provides an interface that allows a programmer to "checkpoint" critical data at pre-determined points within a transaction. In the hardware implementation (for example, with Stratus and its VOS operating system), the programmer does not need to be aware of the fault-tolerant capabilities of the machine.

At a hardware level, fault tolerance is achieved by duplexing each hardware component. Disks are mirrored. Multiple processor are "lock-stepped" together and their outputs are compared for correctness. When an anomaly occurs, the faulty component is determined and taken out of service, but the machine continues to function as normal.

First In First Out (FIFO)

The most rudimentary form of scheduling where packets are delivered to the output port in the order in which they where received. It is used in best-effort IP service. FIFO offers no configuration options; thus it's sometimes accompanied by other schemes like RED (random early discard) and WFQ (weighted fair queuing).

Flow

A set of packets traversing a network element, all of which are covered by the same request for control of quality of service.

Gateway

A gateway is a network point that acts as an entrance to another network. On the Internet, a node or stopping point can be either a gateway node or a host (end-point) node. Both the computers of Internet users and the computers that serve pages to users are host nodes. The computers that control traffic within a company network or at a local Internet service provider (ISP) are gateway nodes.

In the network for an enterprise, a computer server acting as a gateway node is often also acting as a proxy server and a firewall server. A gateway is often associated with both a router, which knows where to direct

a given packet of data that arrives at the gateway, and a switch, which furnishes the actual path in and out of the gateway for a given packet.

Guaranteed Service (hard QoS, quantitative QoS, reserved bandwidth)

A service level that attempts to guarantee a minimal delay for traffic delivery. In the proposed IETF Integrated Service model, guaranteed service is intended for real-time applications, such as teleconferencing. Guaranteed service is an absolute reservation of network resources, typically bandwidth, which implies reservation of buffer space along with the appropriate queuing disciplines, and so on, to ensure that specific traffic gets a specific service level. Typically, guaranteed service is associated with a fine level of traffic classification, so that particular flows (or aggregates) have network resources reserved for them so that required guarantees can be met.

The primary services in use today are Controlled Load Service and Guaranteed Service, each having precise definitions in the context of this work. RSVP was developed as a QoS signaling mechanism to provide these types of flow-based services.

HTTP

The Hypertext Transfer Protocol (HTTP) is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. Relative to the TCP/IP suite of protocols (which are the basis for information exchange on the Internet), HTTP is an application protocol.

Essential concepts that are part of HTTP include (as its name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any Web server machine contains, in addition to the HTML and other files it can serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. A Web browser is an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol address indicated by the URL. The HTTP daemon in the destination server machine receives the request and, after any necessary processing, the requested file is returned.

H.323

A standard approved by the International Telecommunication Union (ITU) that defines how audio-visual conferencing data is transmitted across networks. Most video teleconferencing vendors have announced that their products will conform to H.323.

IEEE 802.1D Standard

802.1p was incorporated into and superseded by the IEEE 802.1D standard.

IEEE 802.1p Standard

An IEEE standard for improving support of time-critical and multicast intensive applications across bridged LANs. An elementary Layer 2 scheme that lets end-stations request priority and network devices enforce it using a tag in the packet header. Because it's a Layer 2 mechanism, 802.1p works on both IP and non-IP networks. 802.1p is a key enabler to QoS by enabling "Prioritized Ethernet" with up to 8 priorities in Ethernet and Token Ring networks. It enables Audio/Video traffic on switched Ethernet fabrics. The eight discrete priority levels range from the default of best effort, through excellent effort (a business-critical application, but tolerant of some delay), interactive multimedia (sensitive to delay or jitter), and reserved (highest priority). Because it must be implemented in the hardware of network devices, existing switches and routers need to be replaced with ones supporting this technology. Incorporated into and superseded by IEEE 802.1D.

IEEE 802.1Q Standard

An IEEE standard for providing a virtual LAN capability within a campus network. It establishes a standard format for frame tagging (Layer 2 VLAN markings), which will enable the creation of VLANs that use equipment from multiple vendors.

Internet Engineering Task Force (IETF)

An engineering and protocol standards body that develops and specifies protocols and Internet standards, generally in the network layer and above.

Integrated Services (IntServ)

The IETF Integrated Services Working Group is developing a set of standards that cover how application services define their QoS requirements, how this information is made available to routers on a hop-by-hop basis, and ways of testing and validating that the contracted QoS is maintained. With the IntServ approach, each network element is required to identify the coordinated set of QoS control capabilities it provides in terms of the functions it performs, the information it requires, and the information it exports. IntServ-capable routers must classify packets based on a number of fields and maintain state information for each individual flow. See RSVP.

International Telecommunications Union (ITU)

International organization under the auspices of the United Nations that develops radio (ITU-R) and telecommunications (ITU-T) standards. Prior to 1993, the ITU-T Standardization Sector was known as the CCITT (Consultative Committee for International Telegraphy and Telephony).

Internet Service Provider (ISP)

Communications service company that provides Internet access and services to its customers. ISPs range in size from small independents serving

a local calling area to large, established telecommunications companies. ISP services also range from the traditional e-mail and Web-page hosting to newer services, such as VoIP, VPNs, and e-commerce.

IPv4 (Internet Protocol version 4)

The most widely deployed version of the Internet Protocol, IPv4 provides some basic traffic classification mechanisms with its IP Precedence/CBQ and Type of Service header fields. However, network hardware and software traditionally have not been configured to use them.

IPv6 (Internet Protocol version 6)

An update to the Internet Protocol that is in the early phases of adoption. Most of the refinements concentrate on basics such as expanding the IP address numbering scheme to accommodate the growth of the Internet. However, IPv6 does include a Class header field that is explicitly intended to designate a Class of Service (an extension of IPv4's IP Precedence/CBQ field).

IP Precedence/CBQ

A 3-bit value in the IP packet header meant to designate the relative priority of a packet, applied on a host, access router or gateway, then used by core routers. Values range from 0 to 7, but typically 6 and 7 are not used by applications, since network control messages use these. For example, a brokerage firm might assign a higher IP Precedence/CBQ value to real-time stock trades than to e-mail to ensure that the trading gets expedited delivery. Same as Type of Service (TOS) bits. The DiffServ Codepoint has been designed to be backwards compatible with IP Precedence.

IP Telephony

Internet Protocol (IP) telephony is the method or protocol by which data is sent from one computer to another on the Internet rather than the traditional telephone company infrastructure to exchange spoken or other telephone information.

IPX

Short for Internetwork Packet Exchange, a networking protocol used by the Novell NetWare operating systems. Like UDP/IP, IPX is a datagram protocol used for connectionless communications. Higher-level protocols, such as SPX and NCP, are used for additional error recovery services. The successor to IPX is the NetWare Link Services Protocol (NLSP).

Jitter

The distortion of a signal as it is propagated through the network, where the signal varies from its original reference timing and packets do not arrive at its destination in consecutive order or on a timely basis, i.e. they vary in latency. In packet-switched networks, jitter is a distortion of the interpacket arrival times compared to the interpacket times of the original

transmission. Also referred to as delay variance. This distortion is particularly damaging to multimedia traffic.

Label Distribution Protocol (LDP)

A fundamental concept in Multi Protocol Label Switching (MPLS) is that two Label Switching Routers (LSRs) must agree on the meaning of the labels used to forward traffic between and through them. This common understanding is achieved by using the Label Distribution Protocol (LDP). LDP is the set of procedures and messages by which Label Switched Routers (LSRs) establish Label Switched Paths (LSPs) through a network by mapping network-layer routing information directly to data-link layer switched paths.

Local Area Network (LAN)

Data communications network connecting computers and related equipment, usually over an area not greater than 10 km.

Latency

Delay in a transmission path or in a device within a transmission path. In a router, latency is the amount of time between when a data packet is received and when it is retransmitted. Also referred to as propagation delay.

Layer 1

Layer 1 refers to the Physical layer of the OSI model. This layer conveys the bit stream through the network at the electrical/optical level. It provides the hardware means of sending and receiving data on a carrier.

Layer 2

Layer 2 refers to the Data Link layer of the OSI model. The Data Link layer is concerned with moving data across the physical links in the network. In a network, the switch is a device that redirects data messages at the layer 2 level, using the destination Media Access Control (MAC) address to determine where to direct the message.

Layer 3

Layer 3 refers to the Network layer of the OSI model. The Network layer is concerned with knowing the address of the neighboring nodes in the network, selecting routes and quality of service, and recognizing and forwarding to the transport layer incoming messages for local host domains.

A router is a layer 3 device, although some newer switches also perform layer 3 functions. The Internet Protocol (Internet Protocol) address is a layer 3 address.

Layer 4

Layer 4 refers to the Transport layer of the OSI model. This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures complete data transfer.

Layer 5

Layer 5 refers to the Session layer of the OSI model. This layer manages the set up, coordination and termination of conversations, exchanges and dialogs between the applications at each end.

Layer 6

Layer 6 refers to the Presentation layer of the OSI model. This layer is usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a pop-up window with the newly arrived text). Sometimes called the syntax layer.

Layer 7

Layer 7 refers to the Application layer of the OSI model. At this layer, communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is not the application itself, although some applications may perform application layer functions.)

Leaky Bucket

A traffic-shaping mechanism in which only a fixed amount of traffic is admitted to the network. The traffic is "leaked" into the network. Excess traffic is held in a queue until either it can be accommodated or must be discarded. See also Token Bucket.

Leased Line

A private, dedicated telecommunications line reserved for a single customer, often used to connect sites in a WAN. The bandwidth on a leased line depends on the service; for example, a T1 line provides 1.544 Mbps in North America, and an E1 line provides 2.048 Mbps in Europe and other countries.

Load Balancing

Distributing processing and communications activity evenly across a computer network so that no single device is overwhelmed. Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server. Busy Web sites typically employ two or more Web servers in a load balancing scheme. If one server starts to get swamped, requests are forwarded to another server with more capacity. Load balancing can also refer to the communications channels themselves.

Metropolitan Area Network (MAN)

Between a LAN and WAN in size and scope.

Multiprotocol Label Switching (MPLS)

An architecture on the IETF standards track for integrating a mechanism for label-swapping with layer 3 routing to accelerate packet forwarding. Forwarding decisions are based on fixed-length labels inserted

between the data-link and network layer headers to increase forwarding performance and path-selection flexibility. The outcome is to make an IP routed network more connection oriented in nature where traffic is routed along a labeled path in the topology. MPLS is expected to reduce the costs of providing VPNs.

Network Edge

The border between two network administrative domains.

Network Element

A networking device, such as a router, a switch, or a hub, where resource allocation decisions have to be made and the decisions have to be enforced and is therefore potentially capable of exercising QoS control over data flowing through it.

OC-n

Short for Optical Carrier, used to specify the speed of fiber optic networks conforming to the SONET standard. For example OC-3 has the speed level of 155.52 Mbps.

Open System Interconnection (OSI)

Open System Interconnection is the commonly referenced multi-layered communication model that includes Layers one through seven.

Over Provisioning

A way to address current limitations of best-effort networks by provisioning more bandwidth than expected network peak requirements. Over provisioning increases the probability, but does not guarantee the quality, of transmission of time-sensitive and bandwidth-intensive applications. Over provisioning is most costly for the WAN.

Packet Classification

A methodology for organizing packets into a group useful for QoS. Classification may be done over a range of granularities, from groups of aggregated flows to individual flows or even subflows. Typically, classification is done in a way similar to defining access lists, that is, based on some contents of the packet header. In this case, a packet may be classified by information in the L2, L3, or L4 headers (source/destination addresses, port numbers, subarea address, applications, user, as well as various Layer 2 attributes. Classification can also be done based on information within the packet payload. Classifications can broad for aggregated flows such as "traffic destined for a subnetwork X," or as narrow as a single flow or even subflow.

Permanent Virtual Circuit (PVC)

A permanent virtual circuit exhibits the characteristics of a leased line connection over a packet-switched, connection-oriented network and remains established over considerable periods of time.

Per Hop Behavior (PHB)

The forwarding treatment given to a specific class of traffic, based on criteria defined in the DiffServ field. Routers and switches use PHBs to determine priorities for servicing various traffic flows.

PHB group

A set of one or more PHBs that can only be meaningfully specified and implemented simultaneously, due to a common constraint applying to all PHBs in the set such as a queue servicing or queue management policy. A PHB group provides a service building block that allows a set of related forwarding behaviors to be specified together (e.g. four dropping priorities). A single PHB is a special case of a PHB group.

Policing

Packet-by-packet monitoring function at a network border (ingress point) that ensures a host (or peer or aggregate) does not violate its promised traffic characteristics. Policing means limiting the amount of traffic flowing into or out of a particular interface to achieve a specific policy goal. Policing typically refers to actions taken by the network to monitor and control traffic to protect network resources such as bandwidth against unintended or malicious behavior. Traffic shaping may be used to achieve policing goals or to do congestion management.

Policy

The combination of rules and services where rules define the criteria for resource access and usage to manage the bandwidth made available to specified traffic. A policy dictates a number of conditions that must be met before a specified action can be taken.

Policy Control

The application of rules to determine whether or not access to a particular resource should be granted.

Policy Decision Point (PDP)

The point where policy decisions are made, responsible for handling policy decisions on behalf of PEPs.

Policy Domain

The part of a network subject to policy. Policy applies to one domain, and policy domains don't overlap. Note that a policy domain is not the same as a DNS domain or Windows NT domain.

Policy Element

Subdivision of policy objects; contains single units of information necessary for the evaluation of policy rules. A single policy element carries an user or application identification whereas another policy element may carry user credentials or credit card information. Examples of policy elements include identity of the requesting user or application,

user/application credentials, etc. The policy elements themselves are expected to be independent of which QoS signaling protocol is used.

Policy Enforcement Point (PEP)

A port on a network device where the policy decisions are actually enforced.

Policy Management

Policy-based networking allows various kinds of traffic to get the priority and bandwidth needed to serve the network's users effectively. With the convergence of data, telephone, and video traffic in the same network, companies will be challenged to manage traffic so that one kind of service doesn't preempt another kind. Using policy statements, network administrators can specify which kinds of service to give priority at what times of day on what parts of their network.

Policy Object

Contains policy-related info such as policy elements and is carried in a request or response related to resource allocation decision.

Policy Resolution

When a policy server or other policy decision point is attempting apply a policy rule to address a particular situation it may find a number of possible policies for which all the matching criteria fit. In such cases, additional criteria should be introduced such as rule ordering or rule specificity in order to provide an unambiguous answer. Policy resolution may be user-assisted or automated depending on the content in which policy us being used.

Policy Server

A server that authorizes QoS requests received from routers or gateways and coordinates bandwidth usage on multiple network devices to ensure consistent end-to-end service throughout the data-path. A Policy Server ensures that packets receive the appropriate Quality of Service, based on a set of policies defined by the network administrator.

Premium Service

In DiffServ terms, Premium service is a peak-limited, extremely low-delay service, resembling a leased line. At the network edge, where a Premium class is first created, it must be either shaped or policed to a rate with no more than a two-packet burst. A policer for Premium service is set to drop packets that exceed the configured peak rate. For this service, the peak rate of the Premium class aggregate across any boundary must be specified and the rate must be smaller than the link capacity.

Profile

The bandwidth and burst requirements for a given class of service, either at the source site or between a source and destination site.

Provisioned Service

A service for which network resources are allocated ahead of time, in accordance with a service contract.

Quality of Service (QoS)

A collective measure of the level of service delivered to the customer. QoS can be characterized by several basic performance criteria, including availability (low downtime), error performance, response time and throughput, lost calls or transmissions due to network congestion, connection set-up time, and speed of fault detection and correction. Service providers may guarantee a particular level of QoS (defined by a service level agreement or SLA) to their subscribers.

QoS-enabled hardware and software solutions sort and classify IP packet requests into different traffic classes and allocates the proper resources to direct traffic based on various criteria including application type, user or application ID, source or destination IP address, time of day, and other user-specified variables.

QoS Policy

A set of actions a network takes to configure and signal for a particular QoS service to be provided to a particular traffic classification.

QoS Routing (QoSR)

A dynamic routing protocol that has expanded its path-selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilization, node resources consumption, delay and latency, and induced jitter.

QoS Signaling

QoS signaling is the means for transmitting QoS requests and parameters between devices or applications to deliver a QoS service requirement across the network. Either in-band signaling (for example, IP Precedence or 802.1p) or out-of-band signaling (RSVP) is used to indicate that a particular QoS service is desired for a particular traffic classification. IP Precedence and RSVP are the two most useful signaling mechanisms because they take advantage of the end-to-end nature of Layer 3 protocol and the growing ubiquity of IP.

QoS Strictness

There are three broad categories of QoS: best-effort, differentiated, and guaranteed. The "strictness" of the QoS service describes how tightly the service can be bound by specific bandwidth, delay, jitter, and loss characteristics. For example, the delay, loss and jitter characteristics can be kept within tight tolerances on a Constant Bit Rate (CBR) service; whereas they are much harder to bound on a typical Internet IP connection.

Queue

In programming, a queue is a data structure in which elements are removed in the same order they were entered. This is often referred to as FIFO (first in, first out). In contrast, a stack is a data structure in which elements are removed in the reverse order from which they were entered. This is referred to as LIFO (last in, first out).

Reservation

Part of a resource that has been dedicated for the use of a particular traffic type for a period of time through the application of policies.

Random Early Detection (RED)

A congestion-avoidance algorithm developed in the early 1990s built on the base-level TCP behavior of automatically slowing transmissions when packet loss is detected. RED tries to anticipate congestion by monitoring a queue on a router. When a specified threshold is reached, it randomly discards packets. This is an implicit signal that the originating applications should slow their transmissions before congestion becomes severe. Unlike CBQ, WFQ or SFQ, RED does not require flow-state in routers. Also a process of intelligent cell discard that occurs within an ATM switch when its buffer capacity is exceeded. RED discards cells in a round-robin fashion among affected connections.

Resource

Something of value in a network infrastructure to which rules or policy criteria are first applied before access is granted. Examples of resources include the buffers in a router and bandwidth on an interface.

Resource reSerVation Protocol (RSVP)

An IETF standard which allows an end device and a network to negotiate specific QoS characteristics. RSVP is a key component of the IETF Integrated Services Internet (IntServ) architecture, which is enhancing Internet protocols to support transmission of real-time data such as voice and video. Using RSVP, an application signals a request to reserve resources along a route from source to destination. RSVP-enabled routers then schedule and prioritize packets. A reservation for the required bandwidth is allowed or denied depending on the current network conditions. In a centrally managed QoS system, RSVP can be implemented according to policies that apply across the network. RSVP is expected to be utilized predominantly in the campus-level networks because of scalability problems have on the WAN.

Round Robin Queuing

An algorithm that services each queue in a predefined sequence. For example, it might empty 1500 bytes a piece from queue 1 (high priority), queue 2 (medium priority), and queue 3 (low priority), servicing each in turn.

Stochastic Fair Queuing (SFQ)

A hash function used to map flow to one of set of queues.

Service

A description of the overall treatment of (a subset of) a customer's traffic across a particular domain, path or end-to-end. In DiffServ, service descriptions are covered by administrative policy and services are constructed by applying traffic conditioning to create behavior aggregates which experience a known PHB at each node within the DiffServ domain. Multiple services can be supported by a single per-hop behavior used in concert with a range of traffic conditioners.

Service Level Agreement (SLA)

A contract between a service provider and customer defining provider responsibilities in terms of network levels (throughput, loss rate, delays and jitter) and times of availability, method of measurement, consequences if service levels aren't met or the defined traffic levels are exceeded by the customer, and all costs involved. The customer may be a user organization or another domain. A SLA may include traffic conditioning rules.

Signaling

Communications between devices to set up calls and tear them down, and to notify application requirements.

Soft QoS (Qualitative QoS)

A quality of service that does not consist of a 100% guarantee of one of the performance parameters (bandwidth, latency, jitter etc.) but delivers that performance with some percentage probability and in general gives better than best effort service.

Synchronous Optical Network (SONET)

North American transport standard for optical networks with speeds from approximately 52 Mbps (OC-1) to 10 Gbps (OC-192). Its European equivalent is SDH (Synchronous Digital Hierarchy).

Static policy

A static policy is one that is put in place at provisioning time, i.e. not in real time in response to the incidence of a new type of traffic at a PEP. This allows the policies to be resolved and for validation of the availability of resource to take place in advance.

Subnetwork Bandwidth Manager (SBM)

An IETF proposed standard for handling resource reservations on shared and switched IEEE 802-style local-area media.

Switched Virtual Circuit (SVC)

Virtual connection set up only for the duration of a single

communications session. In contrast, a permanent virtual circuit (PVC) remains available at all times.

Transport Control Protocol (TCP)

TCP manages the disassembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that assembles the packets into the original message. TCP uses a lower layer, the Internet Protocol (IP), to handle the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination.

TCP/IP uses the client-server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer. TCP/IP and the higher-level applications that use it are collectively said to be "stateless" because each client request is considered a new request unrelated to any previous one (unlike ordinary phone conversations that require a dedicated connection for the call duration). Being stateless frees network paths so that everyone can use them continuously. (Note that the TCP layer itself is not stateless as far as any one message is concerned. Its connection remains in place until all packets in a message have been received.)

TCP Rate Control

A technology implemented at network end points that attempts to regulate the introduction of traffic into the network.

Throughput

In data transmission, throughput is the amount of data moved from one place to another in a given time period.

Token Bucket

A traffic-shaping mechanism in which a predetermined amount of tokens in a bucket represent the capacity allowed to each class of traffic. Packets are forwarded until they exhaust their supply of tokens. When the token supply is exhausted, packets may be discarded or delayed until the bucket is replenished. This controls the transmit rate and accommodates bursty traffic. In some systems, a customer's token supply might correspond to a service fee. See also Leaky Bucket.

Topologies

The shape of a local-area network (LAN) or other communications system. There are three principal topologies used in LANs: bus topology, ring topology and star topology. These topologies can also be mixed. For

example, a bus-star network consists of a high-bandwidth bus, called the backbone, which connects a collections of slower-bandwidth star segments.

Traffic Conditioning

Control functions that can be applied to a behavior aggregate, application flow, or other operationally useful subset of traffic, e.g., routing updates. These may include metering, policing, shaping and packet marking. Traffic conditioning is used to enforce agreements between domains and to condition traffic to receive a differentiated service within a domain by marking packets and by monitoring and altering the temporal characteristics of the aggregate where necessary.

Traffic Conditioning Agreement (TCA)

An agreement specifying classifier rules and any corresponding traffic profiles and metering, marking, discarding and/or shaping rules which are to apply to the traffic streams selected by the classifier. A TCA encompasses all of the traffic conditioning rules explicitly specified within a SLA along with all of the rules implicit from the relevant service requirements.

Traffic Conditioner

An entity that performs traffic conditioning functions and which may contain meters, policers, shapers and markers. Traffic conditioners are typically deployed in DiffServ boundary nodes (i.e. not in interior nodes of a DiffServ domain).

Traffic Profile

A description of the temporal properties of a traffic stream such as rate and burst size.

Traffic Shaping

A group of techniques that attempt to regulate or meter the flow of packets through the network. See also Leaky Bucket, Token Bucket.

Traffic Stream

An administratively significant set of one or more microflows which traverse a path segment. A traffic stream may consist of the set of active microflows which are selected by a particular classifier.

Type of Service (ToS)

An 8-bit field within an IP header which can be used by the device originating the packet, or by an intermediate networking device, to signal a request for a specific QoS level. ToS uses three bits to tell a router how to prioritize a packet and one bit apiece to signal requirements for delay, throughput, and reliability. ToS is also known as IP precedence bit format and the IP precedence field. However, it has not been used much in practice.

User Datagram Protocol (UDP)

UDP is a communications method (protocol) that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). UDP is an alternative to the Transmission Control Protocol (TCP) and, together with IP, is sometimes referred to as UDP/IP. Like the Transmission Control Protocol, UDP uses the Internet Protocol to actually get a data unit (called a datagram) from one computer to another. Unlike TCP, however, UDP does not provide the service of dividing a message into packets (datagrams) and reassembling it at the other end. Specifically, UDP doesn't provide sequencing of the packets that the data arrives in. This means that the application program that uses UDP must be able to make sure that the entire message has arrived and is in the right order. Network applications that want to save processing time because they have very small data units to exchange (and therefore very little message reassembling to do) may prefer UDP to TCP. The Trivial File Transfer Protocol (TFTP) uses UDP instead of TCP.

UDP provides two services not provided by the IP layer. It provides a port number to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact. In the Open Systems Interconnection (OSI) communication model, UDP, like TCP, is in layer 4, the Transport Layer.

Virtual Circuit (VC)

A logical connection between two network nodes that acts as though it is a direct physical connection even though it may physically be packet based. The term is used most frequently to describe connections between two hosts in a packet-switching network.

Virtual Local Area Network (VLAN)

A networking architecture that allows end-systems on topologically disconnected subnetworks to appear to be connected on the same LAN. Predominately used in reference to ATM networking. Similar in functionality to bridging.

Virtual Private Network (VPN)

A virtual private network (VPN) is a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of a tunneling protocol and security procedures. For example, a company could contract with an ISP to set up a VPN to use the Internet to connect two geographically separated sites, rather than set up a dedicated WAN or use a Leased Line. A VPN can give the company the same capabilities at lower cost by sharing the public infrastructure.

Voice over IP (VoIP)

See IP Telephony.

Wavelength Division Multiplexing (WDM)

A mechanism to allow multiple signals to be encoded into multiple wavelengths.

Weighted Fair Queuing (WFQ)

Per-flow packet scheduling in network elements that automatically categorizes traffic flows into high and low priority, based on volume of packets seen by a router or switch. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission medium proportionally according to assigned weights. Like class-based queuing (CBQ), WFQ is designed to prevent any one traffic type from entirely eclipsing another. By default, WFQ favors lower-volume traffic flows over higher-volume ones (for example, a routine e-mail over a large FTP download).

Weighted Random Early Detection (WRED)

Combines IP precedence and Random Early Detection (RED) capabilities to provide differentiated performance characteristics for different classes of service. Packets with a higher IP precedence are less likely to be dropped than those with lower precedence

Wide Area Network (WAN)

A T1, T3, broadband, or other network covering an area generally larger than a city or metropolitan area network (MAN).

References

[Arm-00] G. Armitage, "Quality of Service in IP Networks", *New Riders*, 2000.

[Bar-01] F. R. M. Barnes, R. Beuran, R. W. Dobinson, M. J. LeVine, B. Martin, J. Lokier, C. Meiroşu, "Ethernet Networks for the ATLAS Data Collection System: Emulation and Testing", *Proc. of the 12th IEEE Real Time Congress on Nuclear and Plasma Sciences, Valencia*, June 2001, pp. 6-10.

[Bia-01] G. Bianchi, N. Blefari_Melazzi, "A Migration Path to provide End-to-End QoS over Stateless Networks by Means of a Probing-driven Admission Control", I*nternet Draft, work in progress*, 2001.

[Cal-01] R. Callon, "Constraint-Based LSP Setup using LDP", *Internet Draft, work in progress*, February 2001.

[Cha-01] H. J. Chao, X. Guo, "Quality of Service in High-speed Networks", *John Wiley & Sons Inc.*, 2001.

[Cla-92] D. D. Clark, S. Shenker, L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", *Proc. ACM SIGCOMM'92*, August 1992.

[Dav-99] N. Davies, J. Holyer, P. Thompson, "A Queueing Theory Model that Enables Control of Loss and Delay at a Network Switch" *Technical Report CSTR-99-011, Department of Computer Science, University of Bristol*, November 1999.

[Dem-02] C. Demichelis, P. Chimento, "Instantaneous Packet Delay Variation Metric for IPPM", *Internet Draft, work in progress*, April 2002.

[Fau-01] F. Le Faucher et al., "MPLS Support of Differentiated Services", *Internet Draft, work in progress*, February 2001.

[Flo-95] S. Floyd, V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", *IEEE/ACM Transactions on Networking*, Vol. 3 No. 4, pp. 365-386, August 1995.

[God-02] D. Goderis *et al.*, "Service Level Specification Semantics and Parameters", *Internet Draft, work in progress*, February 2002.

[Gup-01] P. Gupta, N. McKeown, "Algorithms for Packet Classification", *IEEE Network*, March/April 2001, pp. 24-32.

[Koo-99] R. Koodli, R. Ravikanth, "One-way Loss Pattern Sample Metrics", *Internet draft, work in progress*, March 1999.

[ITU-107] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning", *ITU*, May 2000.

[ITU-143] ITU-T Recommendation J.143, "User requirements for objective

perceptual video quality measurements in digital cable television", *ITU*, May 2000.

[ITU-350] ITU-T Recommendation I.350, "General Aspects of Quality of Service and Network Performance in Digital Networks, including ISDNs", *ITU*, March 1993.

[ITU-380] ITU-T Recommendation I.380, "Internet Protocol (IP) Data Communication Service - IP Packet Transfer and Availability Performance Parameters", *ITU*, February 1999.

[ITU-800] ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality", *ITU*, August 1996.

[ITU-861] ITU-T Recommendation P.861, "Objective quality measurement of telephone band (300-3400Hz) speech codecs", *ITU*, February 1998.

[ITU-862] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and codecs", *ITU*, February 2001.

[ITU-1541] ITU-T Recommendation Y.1541, "Network Performance Objectives for IP-Based Services", *ITU, draft*, October 2001.

[Hun-02] R. Hunt, "A review of quality of service mechanisms in IP-based networks", *Computer Communications*, no. 25, 2002, pp. 100-108.

[Mee-01] H. De Meer, P. O'Hanlon, "Segmented Adaptation of Traffic Aggregates", 9th *Int'l Workshop on Quality of Service, IWQoS'01*, Karlsruhe, 2001.

[Pet-99] L. Peterson, B. Davie, "Computer Networks: A System Approach", *San Francisco, Morgan Kaufmann*, 1999.

[QBo-**] Qbone, *The QoS project of the Internet2 consortium*, (http://qbone.internet2.edu).

[Rai-00] V. Raisanen, G. Grotefeld, "Network performance measurement for periodic streams", *Internet draft, work in progress*, March 2000.

[RFC-1349] P. Almquist, "Type of Service in the Internet Protocol Suite", *IETF RFC 1349*, July 1992.

[RFC-1812] F. Baker, "Requirements for IP Version 4 Routers", *IETF RFC 1812*, June 1995.

[RFC-2105] Y. Rehter, B. Davie, D. Katz, E. Rosen, G. Swallow, "Cisco Systems' Tag Switching Architecture Overview", *IETF RFC 2105*, February 1997.

[RFC-2178] J. Moy, "OSPF Version 2", *IETF RFC 2178*, July 1997.

[RFC-2205] R. Braden, L. Zhang, S. Berson, A. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", *IETF RFC 2205*, September 1997.

[RFC-2210] J. Wroclawski, "The use of RSVP with IETF integrated Services", *IETF RFC 2210*, September 1997.

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

[RFC-2212] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", *IETF RFC 2212*, September 1997.

[RFC-2309] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", *IETF RFC 2309*, April 1998.

[RFC-2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics", *IETF RFC 2330*, May 1998.

[RFC-2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", *IETF RFC 2474*, December 1998.

[RFC-2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", *IETF RFC 2475*, December 1998.

[RFC-2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", *IETF RFC 2597*, June 1999.

[RFC-2598] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", *IETF RFC 2598*, June 1999.

[RFC-2638] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", *IETF RFC 2638*, July 1999.

[RFC-2678] J. Mahdavi, V. Paxson, "IPPM Metrics for Measuring Connectivity", *IETF RFC 2678*, September 1999.

[RFC-2679] G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Delay Metric for IPPM", *IETF RFC 2679*, September 1999.

[RFC-2680] G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Packet Loss Metric for IPPM", *IETF RFC 2680*, September 1999.

[RFC-2681] G. Almes, S. Kalidindi, M. Zekauskas, "A Round-trip Delay Metric for IPPM", *IETF RFC 2681*, September 1999.

[RFC-2748] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", *IETF RFC 2748*, January 2000.

[RFC-2990] G. Huston, "Next Steps for the IP QoS Architecture", *IETF RFC 2990*, November 2000.

[RFC-2998] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine, "A Framework for Integrated Services Operation Over DiffServ Networks", *IETF RFC 2998*, November 2000.

[RFC-3031] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", *IETF RFC 3031*, January 2001.

[RFC-3036] L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas, "LDP Specification", *IETF RFC 3036*, January 2001.

[RFC-3148] M. Mathis, M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", *IETF RFC 3148*, July 2001.

[RFC-3175] F. Baker, C. Iturralde, F. Le Faucher, B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", *IETF RFC 3175*, September 2001.

[Rip-**] Réseaux IP Européen (RIPE) Test Traffic Measurements Service (TTM), (www.ripe.net/ttm).

[Sur-**] Surveyor Project, Advanced Network & Services (www.advanced.org/surveyor).

[Tan-97] A. S. Tanenbaum, "Computer Networks", *Prentice Hall*, 3rd edition, 1996.

[TFT-**] TF-TANT, *The Joint DANTE / TERENA Task Force*, (www.dante.net/tf-tant/).

[Wan-01] Z. Wang, "Internet Quality of Service: Architectures and Mechanisms", *Morgan Kaufmann*, 2001.

[Wes-01] L. Westberg, M. Jacobsson, D. Partain, G. Karagiannis, S. Oosthoek, V. Rexhepi, R. Szabo, P. Wallentin, "Resource Management in DiffServ Framework", *Internet draft, work in progress*, 2001.

[Wro-01] J. Wroclawski, A. Charny, "Integrated Service Mappings for Differentiated Services Networks", *Internet Draft, work in progress*, 2001.