# IP QOS: Traveling in First Class on the Internet

**Chris Metz** • *Cisco Systems* • *chmetz@cisco.com*

The global reach and ubiquity of the Internet has created a transport and delivery vehicle for all sorts of applications. Some new ones, like voice over IP (VoIP) and packetized video, can present multimedia data in real time. Others that are low-bandwidth and text-based may include a "high-priority" label because they process mission-critical business information.

In both cases, the network must handle the application packets in a special way so that the data is delivered to the end user ahead of other traffic. But the Internet and, more generally, IP networks offer no easy way to either identify such packets or subsequently give them special handling.

This situation is beginning to change. Indeed, the concept of Quality of Service—that is, the network capability to provide a nondefault service to a subset of the aggregate traffic—has now entered the IP lexicon.[1] IP QoS will no doubt have a significant economic impact as the Internet evolves from a best-effort connection engine into a universal transport and service-delivery medium for large volumes of voice, real-time, and corporate business data.

## The Problem with IP: One Class Only

Until recently, IP networks supported one service class: best effort. The network would make its best attempt to deliver packets to their destinations but with no guarantees and no special resources allocated for any of the packets. The reasons IP has never had any notion of QoS are various.

First, the original TCP/IP protocol suite was built on the idea of fair and equitable access to all and no special treatment for anyone. With the exception of the communicating endpoints, no connection state was to be maintained anywhere in the network. If a packet did not arrive safely at the destination, it was up to the source to retransmit the original packet.

Second, the internal workings of early routers (and many in current operation) used a first-in, first-out (FIFO) queuing strategy. If more packets arrived than the router could handle and the queue filled up, newly arriving packets (the tail of the queue) were dropped.

Third, the impending collapse of the Internet brought on by increased traffic and FIFO-based routers was headed off at that pass by changes to TCP that allowed it to adapt its sending rate to available network capacity.[2] Specifically, TCP started clocking its sending rate to the arrival and frequency of acknowledgments sent by the receiver. If the network was congested, the sender reduced its information transfer rate; if the network had available capacity, it increased its sending rate.

Finally, there was no driving need to re-architect the TCP/IP protocol suite to support QoS since there were no applications that really needed it. The dominant applications have been and continue to be HTTP, FTP, and e-mail, all of which use TCP and can therefore adapt their sending rates to whatever capacity the network offers.

## Early Mechanisms for Differentiation

Support for QoS over IP-based networks is not an issue that has lacked attention or interest. In their seminal 1992 paper, Clark, Shenker, and Zhang outlined an architecture to support real-time traffic flows over a packet data network.[3] In addition to describing different service classes the network should support (guaranteed, predictive, and best-effort), the paper describes two important mechanisms that are still used today.

**Token bucket filter.** The first was a token bucket filter that characterizes the application traffic load receiving a particular service. As shown in Figure 1, it can be conceptualized as a bucket of depth *B* that is replenished with tokens, or credits, at a rate of *R* tokens per second. When a packet arrives at the router, some number of tokens (based on the packet size) are subtracted from the bucket. A packet cannot be sent unless there are sufficient tokens in the bucket. A token bucket filter allows a source to transmit a burst of packets equal to the total number of tokens in the bucket, which is less than or equal to *B*.

A traffic source is said to conform to the parameters of the token bucket filter if it sends packets at a rate less than or equal to *R*. Therefore the network can easily understand and enforce traffic characterized by a token bucket filter because conforming traf-

fic will never exceed $R(t) + B$ for any increment of time equal to $t$. Each implementation must decide what to do with nonconforming traffic. The token bucket filter is used in many router implementations to quantify and enforce the treatment a particular flow will receive from the network.

**Weighted fair queuing**. The second important mechanism described by Clark, Shenker, and Zhang was a weighted-fair-queuing (WFQ) algorithm used to schedule packets for outbound transmission from the routers or switches. WFQ is variant of the fair-queuing algorithm in which individual packets of a flow are time-stamped based on their arrival rate at the router, their scheduled departure time from the router, and their length.[4] The departure queue of the WFQ scheduler is reordered every time a new packet arrives so the packets with the smallest time stamps are transmitted first. The original FQ algorithm provides a fair share of the available bandwidth for each flow ($1/N$) for $N$ number of flows while the weighted variant allows a particular flow to receive more than its fair share of bandwidth.

Two properties make the WFQ scheduler an ideal choice for supporting real-time traffic flows. First, a particular flow is guaranteed its allocated share of the bandwidth irrespective of the behavior of other flows traveling through the same router. Second, a WFQ scheduler is work-conserving, which means that the router will always transmit available packets and therefore the link is never idle. Moreover, Parekh and Gallager proved that a network of routers could quantifiably bound delay for traffic conforming to a token bucket filter and scheduled using WFQ.[5] Sophisticated scheduling algorithms like WFQ—essential for QoS support—are currently implemented in many advanced routers and switches.

## Reservations Required

As the number of real-time applications grew, so did the realization that best-effort service was inadequate to support them.

**IntServ.** In 1994, the Internet community began work to define an Integrated Services Architecture (IntServ) that would extend the existing IP architectural model to support both real-time and best-effort traffic flows.[6]

The IntServ architecture defines a flow as a stream of packets with common source addresses, destination addresses, and port numbers. IntServ suggested that for a flow to receive a desired level of service in terms of quantifiable bandwidth or delay, it is necessary to install and maintain flow-specific state in the network. Of course a router only has a finite amount of buffers and CPU, and is attached to links with a maximum bandwidth. Thus each router in the network would have to exercise a degree of discretionary control over what flows would be allocated what resources based on available capacity. This idea that the network might deny service because of insufficient resources ran contrary to the notion of the connectionless, best effort, "send-packets-whenever" kind of service offered by traditional IP.

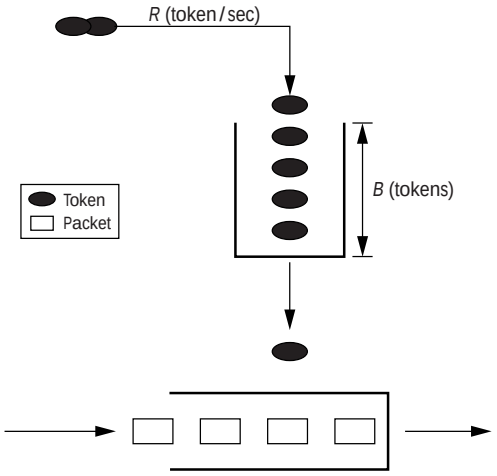


**Figure 1. Token bucket filter. The flow of packets through a router can be enforced by establishing a "bucket "of depth *B* that is replenished with tokens or credits at a rate of *R* tokens per second, and the quantification can be used to enforce the treatment of a particular flow.**

The basic components of the IntServ architecture are traffic control, traffic classes, and the setup protocol.

Traffic control includes *admission control*, which checks to see if the resources in the host or router can support a particular service; the *packet classifier*, which examines the source address, destination address, and port fields in each packet to determine what class the packet belongs to; and the *packet scheduler*, which schedules the packet for transmission on the outbound link.

IntServ supports two traffic classes in addition to best-effort service: *guaranteed service* supports real-time traffic flows that require a quantifiable bound on delay; *controlled load* approximates a best-effort service over an uncongested network.

The setup protocol enables a host or application to request a specific amount of resources from the network. It delivers the reservation request generated by the application to each router's traffic control component.

**RSVP.** The de facto setup protocol in the IntServ architecture is the Resource Reservation Protocol, otherwise known as RSVP. With RSVP, the application source (the sender) transmits a Path message along the routed path to the unicast or multicast destination (the receiver). The purpose of the Path message is twofold: to mark the routed path between the sender and receiver and to collect information about the QoS viability of each router along that path.

Upon receiving the Path message, the destination host or hosts can gauge what services the network can support (for example, guaranteed service or controlled load) and then generate an RSVP reservation (Resv) message. The Resv message contains traffic and QoS objects that are processed by the traffic control component of each router as it follows the reverse path upstream toward the sender. If the router has sufficient capacity, then resources along the path back toward the receiver are reserved for that flow. If resources are not available, RSVP error messages are generated and returned to the receiver.

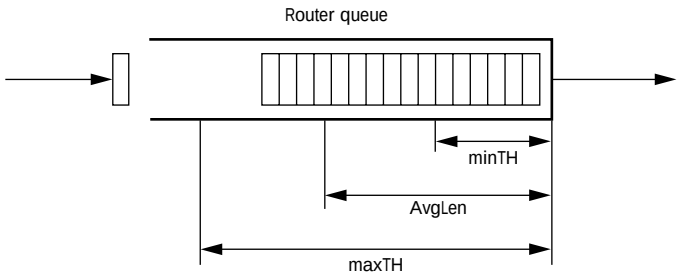


**Figure 2. Random Early Detection, or RED, queue.**

The per-flow reservation state maintained in the routers will be deleted unless RSVP Path and Resv messages are periodically sent by the sender and receivers, respectively.

The work of the IETF's IntServ and RSVP working groups culminated in RFCs 2205 through 2216, which document the IntServ architecture, its components, and the RSVP protocol. In addition, a number of vendors—including industry stalwarts Microsoft, Cisco, and Intel—are currently or will soon ship RSVP-supported products.[7]

**Open issues.** However, while RSVP was touted as the solution to IP's QoS shortcomings, its applicability and scalability over large networks—in particular the Internet—are limited.[8] For example, consider a core router in a large ISP supporting 10,000 VoIP flows set up using RSVP. Since RSVP is unidirectional, the router would have to maintain state information on 10,000 flows in each direction while processing frequent RSVP refresh messages.

In addition, the current version of RSVP lacks both adequate security mechanisms to prevent unauthorized parties from instigating theft-of-service attacks, and policy control—that is, techniques to authenticate and authorize applications or endusers wishing to reserve resources.

Other efforts to deliver better service over IP networks include placing IP traffic over ATM virtual connections that support QoS. This lets an IP flow receive a quantifiable level of service commensurate with the QoS specified for the ATM VC. Unfortunately, this support is limited to the ATM portion of the end-to-end path, which is mostly confined to backbone networks. The interaction of the IP QoS semantics defined in the IntServ model with those supported by various datalink layers (for example, ATM) is being addressed in the Integrated Services over Special Link-Layers (ISSLL) working group of the IETF.

## Random Discard

Another popular approach to QoS is to minimize the depth of router queues by intelligently discarding packets.

Recall that routers using FIFO queues drop all packets at the tail of the queue when the queue fills. TCP sources use packet drops as an implicit signal of network congestion and reduce their information transfer rates accordingly. Although this allows overflowing router queues to drain, it also reduces the rate at which TCP sources send data, leading to low network utilization and reduced throughput. TCP sources then detect available network capacity and ramp up their sending rates. This results in full queues, tail drops, and another round of reduced sending rates.

This cycle of inefficiency is called *global synchronization*. The remedy is a packet-discarding strategy called Random Early Detection. The idea behind RED is quite simple: Packets are randomly discarded with increased probability as the size of the queue grows.

RED defines a minimum queue depth (minTH), a maximum queue depth (maxTH), and a time-based average queue length (AvgLen), as shown in Figure 2. If AvgLen<minTH, then no packets are dropped; if AvgLen>maxTH, then all new packets are dropped; if minTH<AvgLen <maxTH, then packets are randomly dropped with increasing probability as the AvgLen increases.

This probabilistic queue management scheme gracefully instructs some TCP sources to reduce their sending rates so that the router queue does not overflow. This allows the router to support new TCP connections, handle periodic bursts of data, and maintain high network utilization.

RED is implemented on many routers and has been collectively endorsed by the Internet community as a sound queue management strategy for improving and maintaining high network utilization during periods of congestion.[9]

## Differentiated Services

Differentiated Services (DiffServ) is the current approach for supporting IP QoS. A number of factors have driven its design.

First, the solution has to scale. To achieve this, individual host-to-host microflows are aggregated into a single larger aggregate flow and then that single aggregate flow receives special treatment. This type of aggregated behavior is not unlike the process of packet forwarding performed inside of an IP router in which all packets sharing a common destination prefix are forwarded to a single next-hop router.

Second, the solution should be applicable to all applications and should not require a special control protocol or new application programming interfaces as is the case with RSVP.

Third, router and switch technologies are advancing rapidly, with OC-48 (2.4-Gbit) line rates supported today and OC-192 (10-Gbit) coming soon. Core routers and switches operating at these speeds do not need to be burdened with the instantiation of per-flow or per-customer state. A more efficient and scalable option is to provision per-class or per-service state.

Finally, ISPs are desperate to offer a portfolio of services their customers will pay for. QoS is one of them.

The approach taken by DiffServ (DS) is to classify individual microflows at the edge of the network into one of several unique service classes (such as gold, silver, and bronze) and then apply a per-class service in the middle of the network. The classification is performed at the network

ingress based on an analysis of one or more fields in the packet. The packet is then marked (turning on some code points, or bits, in the packet header) as belonging to a particular service class and then injected into the network. The core routers that forward the packet examine the code points in the packet header to determine how the packet should be treated (for example, what transmission queue the packet should be placed in). To accomplish this, the DiffServ architecture defines several components.[10]

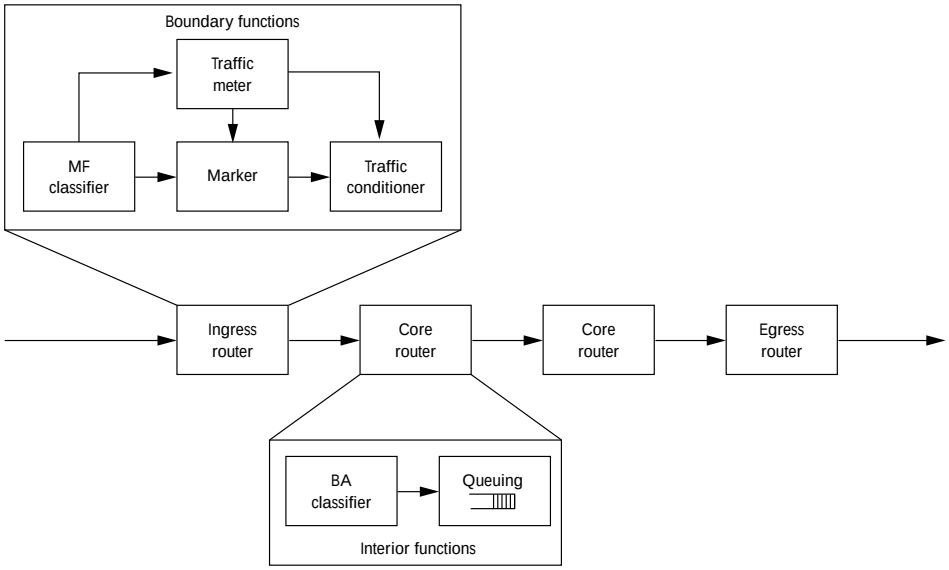


**Figure 3. The DiffServ boundary and interior elements.**

- The *DS-field* is a bit pattern contained in the header of each packet that denotes the service (termed per-hop behavior or PHB) the packet should receive at each hop as it is forwarded through the network.[11] The type-of-service (TOS) field in IPv4 and the traffic class field in IPv6 have been redefined respectively as the DS-fields. The 8-bit DS-field contains 6 bits for DS code points (DSCP) and 2 bits that are currently undefined.
- The *per-hop behavior* (PHB) defines the service the packet receives at each hop as it is forwarded through the network. A PHB may be expressed in relative (compared to other PHBs) or absolute (such as bandwidth or delay) terms.
- *A behavior aggregate* (BA) is a group of packets with the same DSCP. A PHB is applied to each BA inside the network.
- The *boundary router* is positioned at the edge of a DiffServ-capable network. This device is responsible for packet classification, metering, packet marking, and possibly traffic conditioning (such as policing or shaping). Network administrators are responsible for configuring the classifier, which defines the fields to be examined in each packet, and any other actions (for example, dropping packets that do not conform to a token bucket filter) deemed necessary to deliver the service to the external customer. The functions defined for the boundary router can be performed on a router, firewall, or even a host.
- *Interior nodes* can be core switches or routers that provide the PHB based on the DSCP bits contained in the DS-field. These devices typically employ a queue management and scheduling discipline to provide the PHB. RED and WFQ are examples of mechanisms used by routers and switches to support a PHB.

Figure 3 illustrates the DiffServ boundary and interior functions. Packets enter the network through an ingress boundary router. Each packet passes through a multifield (MF) classifier, which works with a traffic meter to determine the next action to be performed. The role of the traffic meter is to measure the packet's conformance with a traffic profile agreed upon by the network provider and the customer. In-profile packets or those that fall inside the parameters of the profile will be treated differently from out-of-profile packets. The DSCP bits in the DS-field may then be marked and the packet conditioned (for example, shaped or dropped) before entering the network.

The core routers in Figure 3 contains a simple BA classifier that determines the PHB to be applied to the packet. All packets belonging to a BA are handled the same way. Again, the PHB is an externally observable behavior performed on each node that is realized through internal queue management and scheduling techniques. In addition, observe that the complex "high-touch" per-packet processing is only performed at the edge of the network by the boundary or ingress device. The net result of the DiffServ machinery is that a particular aggregate flow is provided with a special service as it traverses the network.

The IETF DiffServ Working Group is finishing work on two PHBs: expedited forwarding (EF) and assured forwarding (AF).

The EF PHB was designed to support low loss, low delay, and low jitter connections. It appears as a point-to-point virtual leased line (VLL) service between endpoints with a peak bandwidth. To minimize jitter and delay, packets must spend little or no time in router queues. Therefore the EF PHB requires that the traffic be conditioned to conform to the peak rate at the boundary, and the network of routers be provisioned such that this peak rate is less than the minimum packet departure rate at each router in the network. The EF PHB uses a single DSCP bit to

## IP QoS Resources on the Web


**IETF IntServ Working Group**
http://www.ietf.org/html.charters/intserv-charter.html
**IETF ISSLL Working Group**
http://www.ietf.org/html.charters/issll-charter.html
**IETF DiffServ Working Group**
http://www.ietf.org/html.charters/diffserv-charter.html
**Random Early Detection (RED) Queue Management**
http://www-nrg.ee.lbl.gov/floyd/red.html
**Internet2 Qbone** http://www.internet2.edu/qos/qbone/
**RSVP** http://www.isi.edu/rsvp/
**QoS Forum** http://www.stardust.com/qosform/

indicate that the packet should be placed in a high-priority queue on the outbound link of each router hop.

The AF PHB defines four relative classes of service with each service supporting three levels of drop precedence. Twelve distinct DSCP bit combinations define the AF classes and the drop precedence within each class. When congestion is encountered at a router, packets with a higher drop precedence will be discarded ahead of those with a lower drop precedence. The four AF classes define no specific bandwidth or delay constraints other than that AF class 1 is distinct from AF class 2, and so on.

As a means of providing a scalable and coarse level of service suitable for the ISP-size and enterprise networks, DiffServ holds much promise. It is certainly more scalable than the fine-grained, per-flow approach of RSVP/IntServ and does not require new applications or extensive router upgrades. Moreover, DiffServ gives network providers some degree of latitude in deploying and operating different network infrastructures (for example, ATM or routers) that can support different PHBs.

### What's Next?

Still, some issues remain. For example, how will network policies (such as filters, rules, and parameters) be managed and installed on a potentially large number of boundary components? Possibilities include manual configuration, Simple Network Management Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), and Common Open Policy Server (COPS). Work has now begun in the Policy Framework Working Group of the IETF to define a framework and schemata for managing network QoS policies.

Another issue concerns extending DiffServ across network or ISP boundaries. Obviously this requires some form of bilateral agreement and coordinated network configuration between two consenting providers to support each other's BA PHBs. This could be implemented by having a bandwidth broker in each domain manage DiffServ policies and then communicate that information across ISP boundaries.[12]

What's next on the IP QoS front? The ISSLL working group is studying ways in which RSVP/IntServ and DiffServ can interwork. Efforts are under way to evolve RSVP so that it can set up and maintain state in the network that can benefit aggregate traffic flows transported over large ISP backbones. One application involves the use of RSVP as a means of setting up MPLS explicit paths between ingress and egress routers for traffic engineering purposes. Another application is to employ RSVP extensions to reserve network resources for an aggregate number of microflows such as a bundle of VoIP calls. The Internet2 community has undertaken creation of the Qbone to understand how it can employ IP QoS mechanisms over next-generation high-speed networks. In the long run it will most likely be a combination of the aforementioned solutions along with more bandwidth that will enable the Internet to offer QoS. ■

### REFERENCES


1. P. Ferguson and G. Huston, *Quality of Service—Delivering QoS on the Internet and in Corporate Networks*, Wiley Computer Publishing, 1998.
2. V. Jacobson, "Congestion Avoidance and Control," *Computer Comm. Rev.*, Vol. 18, No. 4, Aug. 1988, pp. 314-329; also available at ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.
3. D. Clark, S. Shenker, and L. Zhang, "Supporting Realtime Applications in an Integrated Services Packet Network: Architecture and Mechanisms," *ACM Sigcomm Proc.*, 1992.
4. A. Demers, S. Shenker, and S. Keshav, "Analysis and Simulation of a Fair Queuing Algorithm," *ACM Sigcomm Proc.*, 1989.
5. A. Parekh and B. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks—the Multiple Node Case," *IEEE/ACM Trans. on Networking*, Apr. 1994, pp. 137-150.
6. B. Braden, S. Shenker, and D. Clark, " Integrated Services in the Internet Architecture: an Overview," RFC 1633, IETF IntServ Working Group, available at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1633.txt.
7. G. Gaines and M. Festa, "A Survey of RSVP/QoS Implementations," update 2, RSVP Working Group, 1 July 1998, available at http://www.iit.nrc.ca/IETF/RSVP_survey/ietf_rsvp-qos_survey_02.txt.
8. A. Mankin, ed., "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement: Some Guidelines on Deployment," RFC 2208, IETF RSVP Working Group, available at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2208.txt.
9. B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309, IETF, available at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2309.txt.
10. S. Blake et al., "An Architecture for Differentiated Services," RFC 2475, IETF DiffServ Working Group, available at ftp://ftp.isi.edu/in-notes/rfc2475.txt.
11. K. Nichols, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, IETF DiffServ Working Group, available at ftp://ftp.isi.edu/in-notes/rfc2474.txt.
12. K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," Internet Draft, Nov. 1997, available at http://www-nrg.ee.lbl.gov/papers/2bitarch.pdf.