Evolution of Quality of Service in IP Networks

Kathleen M. Nichols
knichols@ieee.org

Introduction

The Internet is an interconnection of networks, or clouds. Individual networks are administratively distinct and opaque (or should be) to outsiders. The emphasis is on rich connectivity that allows data packets to find new routes, even if some part of the old route fails during a connection. This is possible because each packet is routed separately. In contrast, the telephony network routes *calls*, using the same path for the duration of the call. Failure of some portion of the pre-arranged path thus results in the loss of the call.

Figure 1 shows how *you* and *me*, hosts on the Internet, might be connected through a number of network clouds, one of which has some interior connections shown. Notice that there are multiple possible paths, both through distinct clouds and inside the expanded cloud. IP QoS must ultimately work in this environment.

Defining Quality of Service

The objective of quality of service in packet networks is to quantify in some way the treatment a particular packet can expect as it transits a network. Adding differential QoS to a network can't and doesn't add bandwidth, thus if some packets get better treatment, others will get worse treatment. A workable QoS architecture is one that provides a framework to manage this unfairness according to policy.

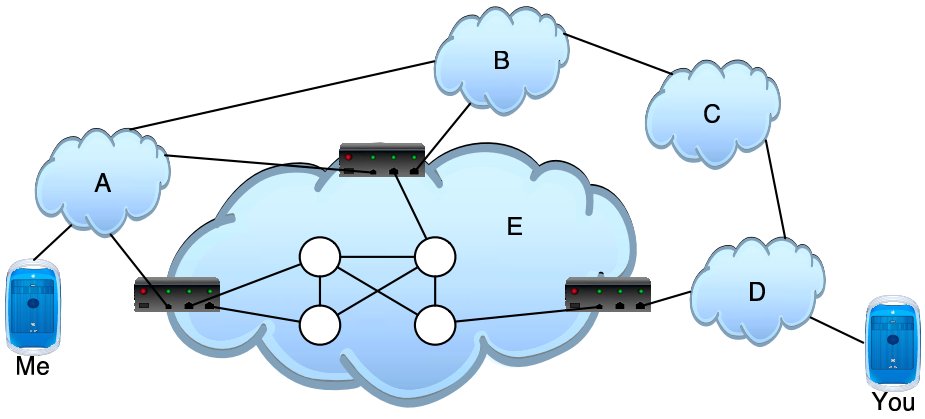
The Internet Engineering Task Force (IETF) makes standards for the Internet and publishes them as RFCs availble at http://www.ietf.org/rfc. The approach to QoS has evolved in the past 20 years.

Packet-oriented Approach

RFC 791, published in 1981, describes per-packet designators intended to indicate what kind of treatment a packet should get on a *per-cloud* basis. Each packet has a Type of Service (TOS) octet where the packet's importance (precedence field) and service requirements (TOS field) could be identified. From RFC 791:

> The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above a certain precedence at time of high load). The major

Figure 1: Connected through an Internet of Network Domains



> choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.
> *and*
>
> The Network Control precedence designation is intended to be used within a network only. The actual use and control of that designation is up to each network. The Internetwork Control designation is intended for use by gateway control originators only. If the actual use of these precedence designations is of concern to a particular network, *it is the responsibility of that network to control the access to, and use of, those precedence designations.* [author's emphasis]

Unfortunately, no architecture was developed to utilize these fields and, in most cases, no mechanisms existed in the network to give differential treatment to packets. An attempt to further define and redefine the 4-bit TOS field of the TOS octet was made with RFC 1349 (now obsolete). RFC 1349 gives guidelines for preferring routes with TOS matched to packet TOS, but the TOS definitions remain rather general, i.e., minimize delay (1000), maximize throughput (0100), maximize reliability (0010), maximize monetary cost (0001), and normal service (0000). There is no framework for attaching quantifiable measures to these qualitative phrases nor for requesting and being granted a specific TOS, either for a packet or a route. RFC 1349 followed a guideline it expresses as:

> The fundamental rule that guided this specification is that a host should never be penalized for using the TOS facility. If a host makes appropriate use of the TOS facility, its network service should be at least as good as (and hopefully better than) it would have been if the host had not used the facility. This goal was considered particularly important because it is unlikely that any specification which did not meet this goal, no matter how good it might be in other respects, would ever become widely deployed and used. A particular consequence of this goal is that if a network cannot provide the TOS requested in a packet, the network does not discard the packet but instead delivers it the same way it would have been delivered had none of the TOS bits been set.

Note that both of these early RFCs operated under the implicit assumption that TOS could be characterized on a linear scale from “better” to “worse,” not a quantifiable metric. More quantifiable and realistic approaches to specifying service may take on a less clearly hierarchical character. Specifically, it may be possible to provide low delay and delay variation for a small amount of throughput or an unspecified delay variation with higher throughput.

A Service-Oriented Approach

In the early 90’s, with the first Internet audio and video experiments, there was a new interest in IP QoS. This resulted in an approach termed Integrated Services (later known as IntServ) and outlined in RFC 1633 written in 1994. It stated the QoS problems as:

> Real-time QoS is not the only issue for a next generation of traffic management in the Internet. Network operators are requesting the ability to control the sharing of bandwidth on a particular link among different traffic classes. They want to be able to divide traffic into a few administrative classes and assign to each a minimum percentage of the link bandwidth under conditions of overload, while allowing "unused" bandwidth to be available at other times. These classes may represent different user groups or different protocol families, for example. Such a management facility is commonly called controlled link-sharing. We use the term integrated services (IS) for an Internet service model that includes best-effort service, real-time service, and controlled link sharing.

The identification of controlled link-sharing is clearly an aggregate treatment of packets, yet the document considers only flow-oriented QoS, not QoS for traffic aggregates. A flow is defined by RFC 1633 as:

> define the "flow" abstraction as a distinguishable stream of related datagrams that results from a single user activity and requires the same QoS. For example, a flow might consist of one transport connection or one video stream between a given host pair.

and the document takes the clear position that IntServ *requires* flow-level admission control and resource reservation(as seen in the following two passages):

> The first assumption is that resources (e.g., bandwidth) must be explicitly managed in order to meet application requirements. This implies that "resource reservation" and "admission control" are key building blocks of the service.
>
> We conclude that there is an inescapable requirement for routers to be able to reserve resources, in order to provide special QoS for specific user packet streams, or "flows". This in turn requires flow-specific state in the routers, which represents an important and fundamental change to the Internet model. The Internet architecture [h]as been founded on the concept that all flow-related state should be in the end systems [Clark88].

Thus IntServ’s model deliberately sets out to break that part of the Internet architecture. Admission control for the IntServ architecture is described as functioning as follows:

> Admission control is invoked at each node to make a local accept/reject decision, at the time a host requests a real-time service along some path through the Internet.

This is very similar to the telephony model but is not consistent with the Internet architecture as it is not only not scalable, but not practical administratively. Authentication and charging need to be carried out at the network level as they are for peering agreements today. The fact that the administration model does not match the Internet has been one factor in IntServ not being adopted.

A Packet- and Network Domain-Oriented Approach

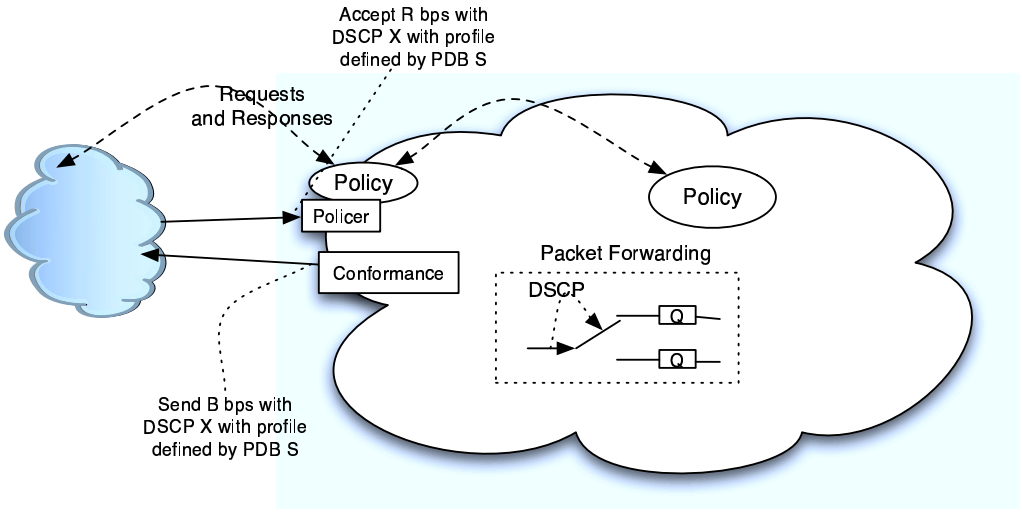
After some time of discussion in the Internet Research Task Force (IRTF) and an IETF Birds-of-a-Feather session covering requirements [FDIFS], the differentiated services (Diffserv) approach was officially started in the IETF in 1998. Differentiated services architecture is an approach to delivering QoS in a scalable, incrementally deployable way that keeps control of QoS local to the cloud, pushes work to the edges and boundaries, and requires minimal standardization, encouraging maximal innovation. This is accomplished by separating the packet forwarding path and control plane functions appropriately. The fields of the TOS octet were redefined to have a 6-bit Differentiated Services Code Point (DSCP) field, in bit positions 0-5. The DSCP is used to identify the behavior aggregate to which the packet belongs and thus can be used to index the forwarding path behavior it should receive. The forwarding behavior at each node is called a per-hop forwarding behavior (PHB).

This approach builds on both the earlier packet-oriented concepts and some of the concepts expressed in IntServ, but is also quite different. It differs from the original per-packet QoS by avoiding associating qualitative performance with the per-packet treatments and is similar in that packet marking meanings are inherently local. However, both the concept of agreeing on packet markings between two clouds and the primitives to enforce such agreements are called out in the architectural model and the standards [RFC2474, RFC2575]. The burden is on the network to control its boundaries. RFC 2474 describes using Diffserv in a network cloud:

> Services can be constructed by a combination of: setting bits in an IP header field at network boundaries (autonomous system boundaries, internal administrative boundaries, or hosts), - using those bits to determine how packets are forwarded by the nodes inside the network, and - conditioning the marked packets at network boundaries in accordance with the requirements or rules of each service.

Compared to IntServ, Diffserv rejects the model that admission control and resource reservation must be done on a per-router basis and moves these functions to a network domain level. The policy of the domain determines which packets are admitted at the network boundary after which treatment is simply by the DSCP mark in the packet header. The policies can reside in a central entity, be distributed at or close to boundaries, or some combination. Enforcement of the policies is distributed at the network boundaries. In one sense, Diffserv views network clouds similarly to how IntServ views routers. Along the packet forwarding path, there are similar primitive functions, i.e., classifiers and packet schedulers. On the other hand, Diffserv explicitly rejects the notion that services map to applications (or vice versa), but rather a service would be made available and admission and marking follows the money and administrative policy.

Figure 2: Control, Enforcement, and Packet Flow in a Diffserv Domain

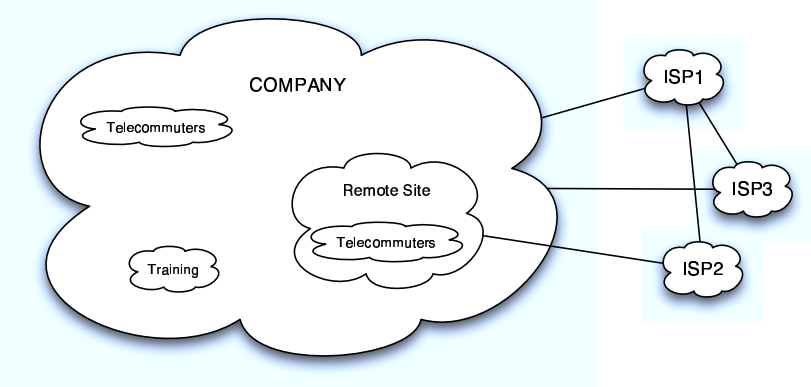


Diffserv defines network primitives that can be put together to deliver a particular behavior to packets as they transit a domain. Such a specific behavior is called a Per-Domain Behavior (PDB) and the network edge (using classifiers and policers) is set to enforce the PDB's requirments while the network interior (using packet scheduling) is configured to deliver the QoS levels specified for the PDB. A "flow" is admitted into a traffic aggregate associated with a PDB and the admission may be through static configuration, per-flow signalling, exchange of credentials ("cookies"), or any practicable means desired by that domain. Admission control changes may result in reconfiguring the network edge, but the interior configuration is regarded as a provisioning decision and only changed on longer time scales. Attributes of a PDB derive from how the edge is protected (what enters the behavior aggregate) and how the individual router hops treat each packet (per-hop behaviors).

Figure 2 shows that a domain may consult some policy entity located at its boundary which may in turn consult another policy entity, one which may be more "central" simply a "peer" entity at another boundary. The network domain's edge ensures the egress packet flow conforms to the agreement with the next domain and, in turn, policies that domain's traffic upon entry. Inside the domain, packet forwarding uses the value of the DSCP field to steer packets to different queues which are serviced by packet schedulers to deliver differential treatment.

The challenge in Diffserv is to carefully construct PDBs so that the per-hop behaviors are invariant under aggregation and a sensible QoS that gives a uniform expectation to all packets of the same aggregate results. This is much more challenging than many believe. It is easiest to do at the most restrictive end (limited bandwidth, strict traffic profiles, careful provisioning) and for the least restrictive classes ("best effort" type) than for other types of service.

Figure 3: Many Types of Clouds Multiply Interconnected



FITTING QOS TO THE INTERNET

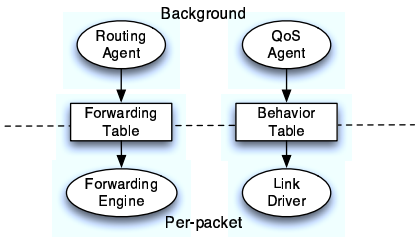
Today's Internet is made up of network domains, colloquially refered to as *clouds.* Clouds do not solely denote regions of different ownership, but are regions of relative homogeneity in terms of administrative control, technology, and/or bandwidth. They can be used to indicate regions whose resources differ or which are administered by different departments of one larger organization (Figure 3). As a roll-out strategy, Diffserv QoS can be deployed in only one cloud, doesn't need to be signaled per connection, and the state in most nodes can be reduced considerably as compared to connection-oriented approaches which tie up resources, require state for every connection and are not incrementally deployable or scalable.

When network clouds are operated by different organizations, the service expectations for packet traffic transiting a "foreign" cloud must be expressed in some manner and often represent a contractual agreement, containing an SLA (Service Level Agreement). This is true for the single class of traffic in the Internet today and adding QoS to IP traffic is expected to result in a catalog of service levels spelled out in one bilateral agreement where the method of binding packets to service level would be part of the agreement. This suggests looking at how expectations are expressed today to provide a framework where this approach can be extended.

ISP Service Level Agreements (SLAs) Today

Currently, most ISPs make service level guarantees as part of their contracts. For example, MCI's web site in February of 2004 targets monthly latency figures of 55 milliseconds or less for regional round trips within Europe or North America. MCI SLAs guarantee 99.5% or greater packet delivery for regional round trips within Europe or North America. The method of measurement, collection, and computation of the metrics is critical to understanding their meaning. According to its website, MCI, like most ISPs, takes these measurements in its core network by collecting pings which use the Internet Control Message Protocol (ICMP). This data is collected in 5 minute intervals and the statistics are derived from an average of all samples of the previous month. Thus the measures are not a strict upper bound, but an upper bound on the

Figure 4: Forwarding and Control Planes in a Router



average. For best-effort traffic, this is a reasonable approach. Diffserv PDBs are specified in RFC 3086 as a framework to extend these kinds of measures and methodologies.

Forwarding Path and Control Plane

Two very different functions must be carried out for packet delivery in the Internet. One is packet forwarding, a relatively simple task as it must be performed at line-rate on a per-packet basis. Packet forwarding uses the packet header to find an entry in the routing table that determines the packet's output interface. The other is routing, which sets and maintains the entries in that table and may need to reflect a range of transit and other policies as well as to keep track of route failures. Routing is more complex and continues to evolve. The separation of these data and control paths is an important part of the Internet architecture. Internet QoS follows this model by using a field of the packet header to find an entry in a behavior table that determines which output queue to put the packet into. This behavior table will be configured by a QoS agent and the development and evolution of this QoS agent may take place separately from the design of forwarding path features.

Similarly with basic packet forwarding, forwarding packets with differentiated QoS is a relatively simple task. There are only a small number of ways of differentiating behavior: packets can be dropped, sent on, or queued. Packet queues can be scheduled to provide different delay, throughput, and loss characteristics. Figure 5 shows the forwarding path primitives in the context of a cloud.

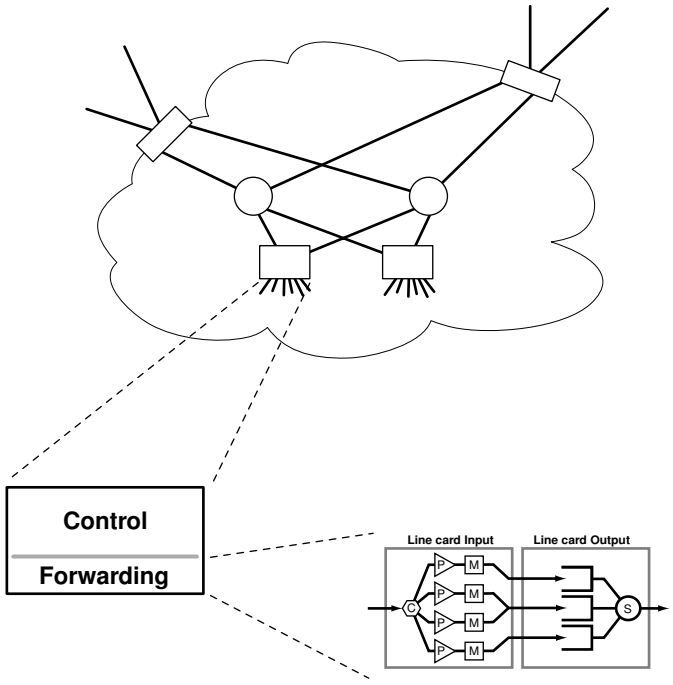
FORWARDING PATH BUILDING BLOCKS AND MECHANISMS

As noted, the IP QoS control plane can evolve over time, but the forwarding path must be capable of providing the required Diffserv primitives well from the first roll out of IP QoS.

Differentiated Services Forwarding Path Building Blocks

A differentiated services-compliant network node includes a classifier that selects packets based on the value of the DS field, along with buffer management and packet scheduling mechanisms capable of delivering the specific packet forwarding treatment indicated by the DSCP. Setting of the DSCP and conditioning of

Figure 5: Forwarding Plane Primitives in the Network Cloud



the temporal behavior of marked packets need only be performed at network boundaries and may vary in complexity.

Figure 4 shows the forwarding path building blocks as: classifier (C), Policer (P), Marker (M), Queue (Q), and Sharing/Shaping (S).

Classification takes apart (input) packet stream. Classifiers select packets based on the content of packet headers according to defined rules and may be one of two types: a multiple field (MF) classifier or a behavior aggregate (BA) classifier. MF classifiers filter on an arbitrary range of IP header fields. At *minimum* this should include the (possibly masked) six-tuple of source, destination, source port, destination port, protocol type, and TOS octet. Other packet fields may be included and it must eventually work for IPv6 as well as IPv4. BA classifiers filter on a packet's DSCP (bits 0-5 of the TOS octet) and all Diffserv-compliant nodes must be capable of classifying on (at least) the DSCP. A Diffserv-compliant node must be capable of running required classification at line rate.

Policing enforces the rules governing packet substreams. Packets are tested for conformance to a particular average rate and instantaneous rate. Packets which do not conform are forwarded to a particular policing action which might include dropping or re-marking (to become part of a different aggregate). Policers contain meters used to measure a traffic stream (which may have been *classified* from a more general traffic stream) against a traffic profile.

Marking propagates information about the aggregate downstream. Particular forwarding treatments are determined by the "mark" that appears in the packet's DSCP field. A Diffserv-compliant marker must be capable of writing a six-bit DSCP into the packet's TOS octet at its maximum forwarding rate.

Queues isolate traffic aggregates from each other. Service differentiation gives some traffic aggregates a level of service (or minimum level of service) that is independent of other traffic: in other words, isolating that class from other packets on the same wire. Nothing but separate queues can perform this function: drop preference schemes offer some protection for different aggregates but not isolation. For example, in the same queue, all streams will see the same (possibly large) delay. More queues at an output interface mean a customer can configure more isolated traffic aggregates. This might be useful at some boundaries, but when provisioning aggregates for an entire domain, it is unlikely that more than a small number will be practical.

Active queue management distributes packet drops and prevents congestion collapse. A queue that is configured for low latency, zero-loss traffic (such as the DB queue described in [RFC3248]) will not have any persistent queue, thus queue management will have no effect and is not required on such a queue. Queues may have active queue management as defined in RFC 2309 and may additionally meet the requirements of [RFC2597].

Sharing/Shaping constructs an (output) packet stream based on local policy and downstream agreements. Delivering a fixed bandwidth, independent of other traffic, requires time-based queue service (traffic shaping). Delivering relative link shares requires some variant of WRR queue service, of which there are many; some better than others at isolation. Packet scheduling for sharing the link may be done separately from a scheduler that additionally shapes packets in a possibly non-work-conserving way, though an example that perform both functions is Class-Based Queueing [CBQ]. There must be a link sharing packet scheduler for the queues. The shaping requirements depend on the role of the interface within the topology.

Edge and Interior Functionality

The DSCP is used to identify the forwarding treatment packets are to receive within a cloud, but at the network edge, it must be possible to identify packets by more complex criteria and mark them for the correct treatment. This puts the most complex "work" of the forwarding path at the edge or boundaries of the network clouds. It is the responsibility of each network cloud to monitor the traffic crossing its boundaries and only admit packets into its interior and into a particular behavior aggregate if it conforms to its administrative policies. Thus, the edge needs mechanisms that allow it to monitor and enforce the policies. The policies themselves may be static or may be set by some kind of admission control. When the network edge is a *trust boundary*, as occurs between clouds with differnent administration, then the control of the traffic entering the network must be the most strict. The monitoring needs to identify the packet, check its conformance to a particular traffic profile, and, if compliant, mark it with the appropriate DSCP that will be used in the interior. For packets exiting the network, it may be necessary to ensure that they conform to the traffic profile expected and/or contracted with the downstream network.

These monitoring primitives at the network edge are configured to match admission control and resource reservation policies. Such policies may be static or dynamic, and the time scale of applicability may cover a wide range. This process is called traffic conditioning. The resource that the edge controls is admssion to a particular behavior aggregate that can transit the interior of the cloud. The interior of the network is provisioned such that certain quantifiable characteristics accrue to different types of behavior aggregates. An admission control mechanism must be configured with the bounds on what can be admitted to each behavior aggregate. Thus the expectation is that each behavior aggregate's characteristics will be represented by its worst case values (e.g., of maximum delay), though a complex and sophisticated mechanism might work differently.

Interior nodes need only steer packets to the appropriate QoS behavior (most typically represented by a queue at the output interface) for the packet's DSCP *mark* and provide the expected per-hop behavior. Note that the output interface part of a boundary router may be considered to be in the interior of a cloud. The queues and sharing mechanism implement the Diffserv PHB. In the interior of a network, the number of required queues should be on the order of the number of distinct behavior aggregates corresponding to PDBs provisioned on that network, a small number. At the edge, queues may be used for additional customer isolation and thus devices intended for the edge may need more queues and more complex scheduling.

Clearly network nodes at the edge must have the most sophisticated classification. Frequently, but not always, the line rate of of packets crossing a boundary will be less than in the interior, hence it may be easier to implement these complex functions at boundaries. Administratively, it is generally easier to isolate one particular ingress/egress and use a smaller set of classifiers than at any interior link. Policers, shapers, and MF classifiers typically appear only at network boundaries.

FROM EDGE-TO-EDGE TO END-TO-END

The evolving model of IP QoS, based on the Diffserv architecture, recognizes both the reality and the strengths of the real Internet. The fundamental difference between this model and the IntServ model is that resources are controlled and allocated on a *per-cloud* basis, that is, admission control and resource allocation are seen as cloud or domain functions, not per-router functions. QoS characteristics are configured and guaranteed on a per-domain basis; traffic transiting a number of domains gets the concatenation of the guarantees of each domain. In a sense this is analogous to IntServ building QoS from transiting routers, but routers are not the right level of granularity for service guarantees and admission control decisions, both for scaling and administrative reasons. Not only is this model more implementable, it really is the only right choice for Internet QoS since it evolves out of existing practice.

Many advantages arise quite naturally from the cloud-based QoS. Since clouds can map to the independently administered regions of the Internet, the control lies within one administrative unit (e.g., company, governement agency, etc). There is the further advantage that this approach is architecturally agnostic in that within a cloud any technology might be used to deliver QoS. Similarly, it is agnostic to the signalling, resource reservation and admission within a cloud, thus these components can develop and evolve separately from the business of providing QoS mechanisms in the forwarding path. Focusing on clouds permits incremental deployability, bringing QoS to the Internet one cloud at a time. For QoS guarantees to be exchanged between clouds, there must be some bilateral agreement between the clouds but this can also be signalling agnostic in the early stages. It is anticipated that early choices in the bilateral agreements will be useful in the evolution of an Internet-wide standard when it is needed.

Putting together the Building Blocks within a Cloud

Edge-to-edge services are built by adding rules to govern behavior aggregates with regard to the initial packet marking, how particular aggregates are treated at boundaries, and temporal behavior of aggregates at boundaries. Different user-visible services can share the same aggregate. Services must be sensible and quantifiable under aggregation. The Diffserv Per-Domain Behavior (PDB) is the idealized edge-to-edge service. RFC 3086 defines the PDB as:

the expected treatment that an identifiable or target group of packets will receive from "edge-to-edge" of a DS domain. A particular PHB (or, if applicable, list of PHBs) and traffic conditioning requirements are associated with each PDB.

RFC 3086 discusses PDBs and their specification in detail and it should be consulted for those requiring a deeper understanding. The definition of a PDB is based on "ideal" conditions, that is no link errors or routing failures, so the service levels made visible to a customer might be somewhat more conservative or may be stated with a statistical probability based on the network operator's observed level of network up time.

Control Plane Functions

A repository of of policy is needed to keep track of priorities and limits on QoS allocations for individual users, projects, and/ or departments. An entity needs to receive requests for QoS, consult and update the database, and send configuration information to the routers, where indicated. RFC 2638 discussed these requirements and used the term Bandwidth Broker (BB) for a QoS Agent meeting them. A BB is part of the network infrastructure and must authenticate requests from users, though information can also be configured. Intradomain policy decisions and implementations remain up to each domain much as for intradomain routing.

The BBs functions can be accomplished by a single central entity, by cooperating entities, by a hierarchical or peer-level arrangement of entities as appropriate for a particular network. Here the term "BB" is used loosely to refer to the functionality regardless of how it is implemented in a particular network domain.

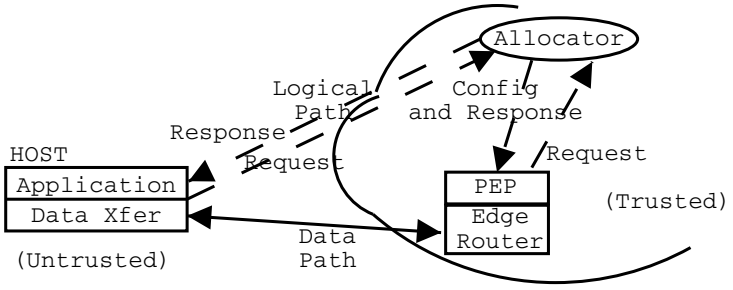
One of the BB functions is to allocate and control access to the "special" QoS levels across its domain. Figure 6 shows possible QoS control flows. A host application may send a request for special service directly to the network, using an RSVP-like signaling protocol. Requests can come from many other sources including network administrators. A request must include the requstor's identifying information as well as information identifying the flow, microflow, or aggregate for which the request is intended. At the Policy Enforcement Point (PEP), this message is recognized as a request and sent to the allocation agent. The allocator needs to check the BB policy database, the requestor's credentials, the time of day and any other relevant information, then returns a "yes" or "no" to the PEP (or directly to the host). If a "yes" is returned, the appropriate DSCP and policer information is sent to the PEP for enforcement (and to the host for conformance). If the reply goes directly to the host, then it must be cryptographically signed in some way so that the PEP will know it is valid.

The BB is a control plane entity used to implement a cloud's policy goals and to configure the forwarding path accordingly. The best way to do this is still an open question, yet using simple policies and static configuration, it is possible to deploy useful network QoS. Recalling the analogy to basic Internet packet delivery, we note that the Internet does not use a single routing protocol, thus we might expect a range of QoS control protocols in each cloud. Over time we expect a single method of gluing together cloud QoS will evolve.

Connecting Network Clouds

Each network cloud is free to use any method to provide QoS across its domain. In order to provide QoS across network clouds, clouds that exchange packet traffic must agree on how packets are to be marked and

Figure 6: Intracloud QoS Allocation Options



what level of QoS they will receive as they transit a domain. This can be a static agreement for specific traffic profiles of traffic marked with a particular DSCP that can be configured into the network's edge policers. If the communicating network clouds agree, some form of dynamic signalling can be used and a cloud's QoS agent can configure the edge policers and meters in response to a request, for the lifetime of the request.

Figure 7 shows an enterprise network (on the left) and its connection to an ISP. Assume that the enterprise has set up the *foo* PDB which utilizes the SP PHB to send voice-over-IP. The leaf routers have been set up to MF classify and police traffic, marking conformant traffic with the correct DSCP. Inside the enterprise network, the DSCP will be used to select the router treatment. If the any SP packets leave the enterprise network, the aggregate can be shaped to meet the traffic profile agreed upon with the ISP (and perhaps remarked to the ISP's specification). The shaping is in the enterprise network's best interests since this prevents the ISP from finding any packets out of profile and dropping them. At the ISP border, the packets may again be MF classified (to determine the originator) and may be remarked in the DSCP. If there is no signalling between domains, the shapers and policers are configured to reflect a contract agreement and the enterprise allocates its SP packets among hosts according to policy.

In 1997, in early diffserv discussions, an approach to interconnecting domains was proposed (documented in RFC 2638) where the BBs along the path communicate. Requests travel along the domains one hop at a time, as shown in Figure 8 (taken from RFC 2638). Each domain's BB may optionally query other adjacent domains if the first refuses the request. Further, each domain's BB may check resources and reserve them completely differently. All that must be agreed upon is the manner of communicating the request and the response.

In time, the protocols for connecting clouds might evolve to something structured and standardized, but this is not necessary in order for enterprise networks to deploy QoS within their boundaries or for an individual ISP (or small set of ISPs) to offer customers QoS to other sites connected to the ISP. The approach to providing QoS inside a cloud does not need to be exposed to the external protocol, only the traffic profiles acceptable and the bounds that can be expected on such traffic measures as delay. Individual clouds can use the signaling protocol and control plane QoS agent of their choice. This is analogous to a cloud's choice of an interior routing protocol vs the use of BGP for connecting Internet clouds.

A detailed treatment of Diffserv QoS, with examples, is contained in [DIFFINT]. Early work on a diffsev architecture [RFC2638] is partially obsolete, but the control plane architecture is still relevant.

Figure 7: Connecting an Enterprise Network to an ISP

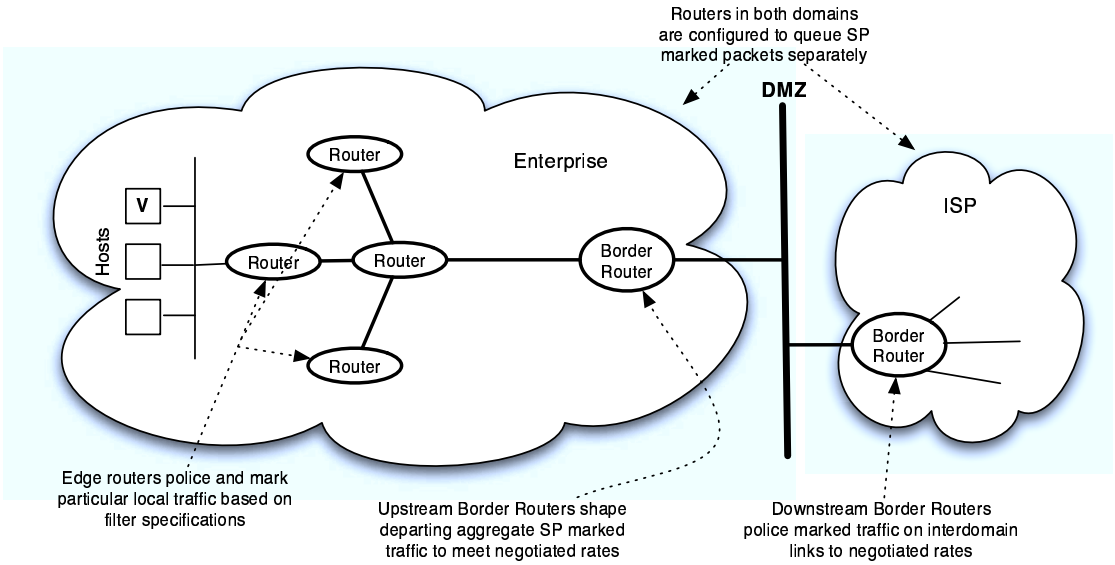
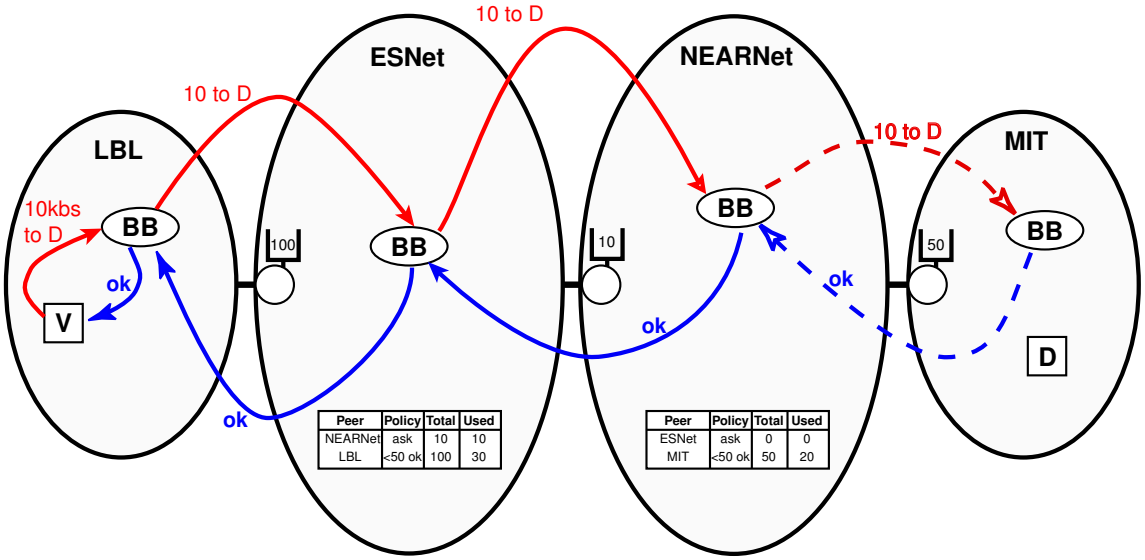


Figure 8: Dynamically Connecting QoS across Domains



STATUS AND FUTURE

Since the definition of PDBs, only three specifications have been proposed to the IETF. One was an Assured Rate PDB based on the Assured Forwarding PHB which is no longer active. In addition there was a Bulk Handling [BH] PDB proposed which has been superseded by a Lower Effort PDB [RFC 3662] incorporating the earlier work. Finally, there was a Virtual Wire PDB [VW] which is still under development by the author to provide delay bounded behavior[1].

Tthe major work of IP QoS is out of the standards arena. Good router implementations, enterprise network roll-outs and ISP service models are needed to advance the state of QoS. Once more work is done developing PDBs and services based on them, work can advance to control plane structures to concatenate PDBs.

REFERENCES

[RFC1633] RFC 1633, "Integrated Services in the Internet Architecture: an Overview", R. Braden, D. Clark, S. Shenkar, June 1994.

[RFC2474] RFC 2474, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", K.Nichols, S. Blake, F. Baker, D. Black, December 1998.

[RFC2475] RFC 2475, "An Architecture for Differentiated Service", S. Blake, F. Baker, D. Black, December 1998.

[RFC2597] RFC 2597, "Assured Forwarding PHB Group", F. Baker, J. Heinanen, W. Weiss, J.Wroclawski, June 1999.

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

[RFC3086] "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification," K. Nichols and B. Carpenter, April, 2001.

[RFC3248] RFC 3248, "A Delay Bound alternative revision of RFC2598", G. Armitage, A. Casati, J. Crowcroft, J. Halpern, B. Kumar, J. Schnizlein, March, 2002.

[Clark88] "The Design Philosophy of the DARPA Internet Protocols," D.D.Clark, Proc SIGCOMM 88, ACM CCR Vol 18, Number 4, August 1988, pages 106-114.

[FDIFS] Minutes on-line at http://www.ietf.org/proceedings/97apr/97apr-final/xrtft122.htm

[BH] "A Bulk Handling Per-Domain Behavior for Differentiated Services", B. Carpenter and K. Nichols, work in progress, 2001, available at http://www.packetdesign.com/publications.html

[VW] "Virtual Wire Per-Domain Behavior", V. Jacobson, K. Nichols, and K. Poduri, work in progress, 2000, available at http://www.packetdesign.com/publications.html.

[DIFFINT] B. Carpenter and K. Nichols, "Differentiated Services in the Internet", *to appear in* Proceedings of the IEEE.

[CBQ] S. Floyd and 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.

[1] A current version can be made available upon request.