QoS in Data Networks : Protocols and Standards

Arindam Paul, *apaul@cse.wustl.edu*

Abstract

This paper intends to provide a overview of past, current and evolving standards and protocols in the area of Quality of Service over computer networks.

See Also : QoS in Data Networks: Products| QoS/Policy/Constraint based routing| QoS over Data Networks (Prof. Jain's Lecture) | Quality of Service over IP: References| Books on Quality of Service over IP
Other reports on recent advances in networking
Back to Raj Jain's Home Page

Table of Contents

1. INTRODUCTION : Definition and Measurement

QoS is the capability of a network to provide better service to selected network traffic over various underlying technologies like Frame Relay, ATM, IP and routed networks etc [CISCO4]. In other words, it is that feature of the network by which it can

differentiate between different classes of traffic and treat them differently.

QoS in a entire network involves capabilities in the

1. End system software running on a computer, for example the operating system.
2. The networks that carry the data being sent back and forth from one endhost to another.

We shall be mostly discussing about point 2 here.The various formal metrics to measure QoS are

1. Service Availability : The reliability of users' connection to the internet device.
2. Delay : The time taken by a packet to travel through the network from one end to another.
3. Delay Jitter : The variation in the delay encountered by similar packets following the same route through the network.
4. Throughput : The rate at which packets go through the network.
5. Packet loss rate : The rate at which packets are dropped, get lost or become corrupted (some bits are changed in the packet) while going through the network.

Any network design should try to maximize 1 and 4, reduce 2, and try to eliminate 3 & 5.

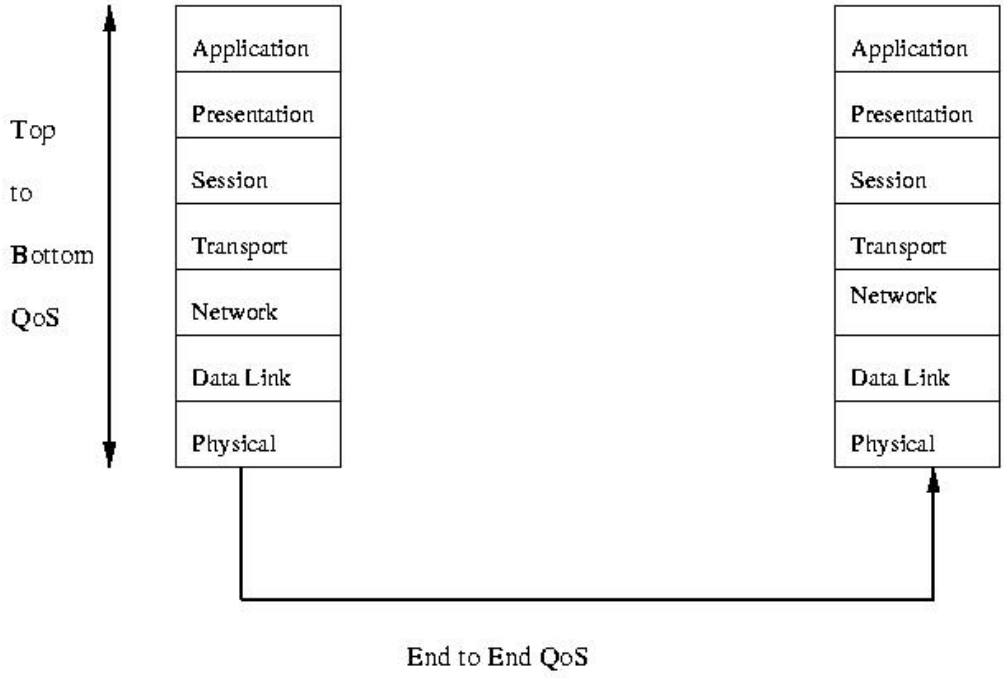


Figure 1 : 2 levels of QoS [stardust]

1.1 Why do we need QoS ?

Today's networks need to support multiple kinds of traffic over single network links. Different kinds of traffic demand different treatments from the network. Just like a first class passenger is ready to pay the premium for superior service, customers of today are ready to pay extra for preferential treatment of their traffic. And just as we cannot have a separate airplane for each first class passenger, similarly we cannot have separate network connections for each of our customers. Therefore much of the bulk of network traffic have to flow through lines where first class traffic and other classes of traffic have to share the bandwidth (just like economy class passengers share the airplane with first class passengers). We can only differentiate at places where the traffic flows through active network elements which have the capability to differentiate. Examples of such entities are routes, switches and gateways.

Therefore, the need to design networks which

1. can deliver multiple classes of service - that is they should be QoS conscious.
2. is Scalable - so that network traffic can increase without affecting network performance
3. can support emerging network intensive, mission critical applications which will be the key determinant of a companies success in the global world.

1.2 End to End QoS

The aim of building a QoS enabled network architecture is to bring together end hosts closer by increasing performance and reducing delay of the underlying network. In order to do this the network should implement service models so that services are specific to the traffic they service.

Three service models have been proposed and implemented till date.

1. Best Effort services
2. Integrated services
3. Differentiated services.

1.3 Best Effort Service

In this model, a application sends data whenever it feels like, as much as it feels like and without requiring anyone's permission. The network elements try their level best to deliver the packets to their destination without any bounds on delay, latency, jitter etc. But they give up if they cannot (for example if they don't receive an acknowledgment even after trying to deliver a packet even after transmitting a certain number of times), and without informing either the sender or the recipient. So the onus is really on the end systems to make sure that the packet goes through. An example of this service is delivered by the current day IP networks.

Back to Top

---

2. INTEGRATED SERVICES

This is a set of standards set down by IETF in which multiple classes of traffic can be assured of different QoS profiles by the network elements. Here the applications have to know the characteristics of their traffic before hand and signal the intermediate network elements to reserve certain resources to meet its traffic properties. According to the availability of resources, the network either reserves the resources and sends back a positive acknowledgment, or answers in the negative. This part of the standard is called "Admissions Control" - which decides between which traffic to grant protection and which not to. It is often decided by the policy decisions of the router/switch and is described in section 2.3. Please note that without any admissions control, it would mean granting all available resources to all classes of traffic - which is what we have in best effort networks. If the network says a "Yes", the application sends the data, which sticks to the traffic properties it had negotiated with the network. Note: if end application tried to send out-of-profile traffic, then the data is given best effort service, which may cause the packets being dropped alltogether.This signaling protocol is called **Resource reSerVation Protocol - RSVP**. Then the network fulfills its commitments by putting together various strategies classified under the following schemes :

1. Maintaining per-flow-state
2. Traffic shaping and policing
3. Congestion Avoidance
4. Congestion Management
5. Link Efficiency Mechanisms

IntServ differentiates between the following categories of application :

1. Elastic Applications: No problem for delivery as long as packets do reach their destination. Applications over TCP fall into this category since TCP does all the hard work of ensuring that packets are delivered. There is no demand on the delay bounds or bandwidth requirements. e.g. web browsing and email.
2. **Real Time Tolerant (RTT)**Applications : They demand weak bounds on the maximum delay over the network. Occasional packet loss is acceptable. e.g. video applications which use buffering, which hides the packet losses from the application.
3. **Real Time Intolerant (RTI)**Applications : This class demands minimal latency and jitter. e.g. 2 people in a videoconference. Delay is unacceptable and ends should be brought as close as possible. The whole application should simulate 2 persons talking face to face.

To service these classes, RSVP with the various mechanisms at the routers delivers the following classes of service :

1. **Guaranteed Service** [gqos] :This service is meant for RTI applications. This service guarantees

a. bandwidth for the application traffic
b. deterministic upper bound on delay.
It is important for interactive applications or real time applications. Applications can decrease delay by increasing demands for bandwidth.

2. **Controlled Load Service** [clqos]: This is meant to service the RTT traffic. The average delay is guaranteed, but the end-to-end delay experienced by some arbitrary packet cannot be determined deterministically. e.g. H.323 traffic.

2.1 Features of RSVP [rsvp2]

A data flow in RSVP is a sequence of messages that have the same source, destination and the same QoS. In RSVP resources are reserved for unidirectional data [ciscorsvp], or "simplex" flows [nortel2], going from sender to receiver. The sender is upstream and the receiver downstream.

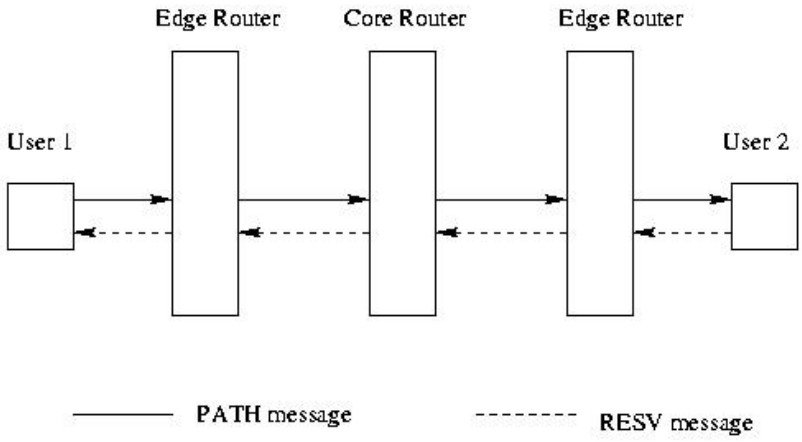


Figure 2 : IntServ Architecture

A host wanting to send some data requiring QoS sends a special data packet - called a PATH message - to the intended receiver. This packet has the characteristics of the traffic the sender is going to send within it. The router and intermediate forwarding devices install a path state with the help of this PATH message and become aware of their adjacent RSVP aware devices. Thus a path from the source to the destination is pinned down. If the path cannot be installed then a PATH Error message is sent upstream to the sender who generated the PATH message.

After the receiver gets the PATH message, it issues a RESV message. There can be 3 types of reservations - **shared reservations, wildcard filter type reservations and shared explicit type reservations**. They are described in section 2.2. By this time all devices along the path have established a path state and are aware of the traffic characteristics of the potential flow. RESV contains the actual QoS characteristics expected by the receiver. Different receivers may specify different Qos features for the same multicast flow. RESV exactly traces back the path taken by the PATH message. Thus each device along the path gets to know the actual QoS characteristics of the flow requested by the receiver & each decide independently how much of the demand it should satisfy or refuse altogether. If it refuses then a RESV Error message is issued downstream to the receiver who generated it in the first place.

Note : RSVP is NOT a routing protocol. Rather it uses the path established by standard routing protocols like OSPF and RIP to determine its next hops. RSVP is a transport layer protocol if one follows the OSI 7 layer model. Therefore with changes in its routes due to link or node failures, RSVP needs to update its path states at the various links. This is done by the sender issuing PATH messages and the receiver answering with RESV messages periodically. So the path states are "soft" - since they timeout after some time interval and become invalid. Once the PATH messages are received, the path states are again created. So to have persisting path states, PATH and RESV messages should be periodically issued.

After the RESV messages is received by the source, if there are no RESV Error messages, then the source sends a RESV Confirmation message to any node who wants it. It is a oneshot message. Immediately afterwards the sender starts transmitting its messages. The intermediate network's forwarding devices service the data granting it use of the reserved resources. At the end of transmission the sender issues a PATH Tear message to which the receiver answers with a RESV Tear message. They are routed exactly like PATH and RESV messages respectively. RESV & PATH Tear messages can be issued by either a end system to explicitly break a RSVP connection or by routers , due to a timeout on a state.

2.2 RSVP Reservation Types

Reservation types initiated by a receiver may be of the following types :

1. Distinct Reservation : The receiver requests to reserve a portion of the bandwidth for each sender. In a multicast flow with multiple senders each senders flow can thus be protected from other senders' flow. This style is also called as the Fixed Filter Style.

2. Shared Reservations : Here the receiver requests the network elements to reserve common resources for all the sources in the multicast tree to share among themselves. This style is important for applications like vide conferencing, where one sender transmits at a time, since it leads to optimum usage of resources at the routers. They are of 2 types

2a Wildcard Filter Type : The receiver requests resources to be reserved for all the sources in the multicast tree. Sources may come and go but they should share the same resources to send their traffic, so that the
sink can receive from all of them.
2b Shared Explicit Reservation : This is exactly like the wildcard filter type except that the receiver chooses a fixed set of senders out of all available senders in the multicast flow to share the resources.

Tunneling

In many areas of the internet, the network elements might not be RSVP or IntServ capable. In order for RSVP to operate through these non RSVP clods, RSVP supports tunneling through the cloud. RSVP PATH and RESV request messages are encapsulated in the IP packets and forwarded to the next RSVP capable router downstream and upstream respectively.

Now we will be looking into the various other strategies implemented at the network elements and forwarding devices such as router, switches and gateways which work in tandem with signaling protocols like RSVP to ensure end-to-end QoS.

2.3 Admissions Control

"A policy enforcement point (PEP) is a network device or a policy on a network device that is actually capable of enforcing policy" [rsvp]. It may be inside the source or destination or on any node in between. Local policy Module (LPM) is the module responsible for enforcing policy driven admission control on any policy aware node. The RSVP module requests the LPM on receipt of a RSVP message for a decision on it. The LPM interacts with the PEP, which in turn contacts the PDP with a request for a policy decision on the packet and then sends the packet to the RSVP module.

Policy decision point (PDP) is the logical entity which interprets the policies pertaining to the RSVP request & formulates a decision. It decides who gets what QoS,when, from where to where and so on. "PDP makes decisions based on administratively decided policies which reside on a remote database such as a directory service or a network file system"[rsvp].

PEP's and PDP's can reside on the same machine but that would lead to scalability and consistency problems. Therefore separate policy servers exist by which a single PDP can serve multiple PEP's. PDP's are useful centers for network monitoring since they are like the central headquarters where all QoS-requesting traffic have to get approval from.

2.4 COPS [rsvp]

The standard protocol that PDPs and PEPs use to communicate among themselves is called **COPS (Common Open Policy Service)**. It is a simple request-response protocol. PDPs are required to have states so that they can remember a PEP's request. When a PEP sends a policy decision request, called a COPS Request message, the PDP's (after getting the request) may reach a immediate decision by processing the packet and send the PEP its decision, called a COPS decision message. But after some deliberation on the history of similar packets (since it keeps track of history)it may revert its decision and send the PEP another COPS Decision message. Thus PDP's may send their decisions asynchronously. On receipt of a COPS Decision message the PEP has to change their enforcement strategies as well. After being done with a flow, PEP's may send explicit COPS Delete message to the PDP to remove the state associated with the request and stop any further decision making at the PDP on that request. PEP's can send Report messages to the PDP asynchronously reporting accounting and monitoring information relevant to a existing request state. COPS distinguishes between three request types :

a. Admission control requests : If a packet is just received by a PEP, it asks the PDP for a admission control decision on it.
b. Resource Allocation request : The PEP requests the PDP for a decision on whether, how and when to reserve local resources for the request.
c. Forwarding request : PEP asks PDP how to modify a request and forward it to the other network devices.

COPS relies on TCP for reliable delivery. It may use IPSec for security purposes. Since PDP's and PEP's are stateful with respect to path or reservation requests, so RSVP refresh messages need not be passed to them via COPS. If a path or reservation state timeouts

or a RSVP Tear message comes, then a COPS Delete message is issued to the PDP to remove the state. COPS also has provisions for changing the data headers so that it can communicate with other policy servers. COPS can accommodate RSVP in multicast flows since COPS distinguishes between 'forwarding requests' and 'admission control requests' and so differentiates between sender and receiver of RSVP flows and can control which messages are transmitted and where they are sent.

2.5 Congestion Management Techniques [cisco1]

These are methods implemented in routers to support the various signaling protocols and actually provide different classes of service. These are usually implemented at core routers. They involve

a. Creating different queues for different classes of traffic
b. A algorithm for classifying incoming packets and assigning them to different queues.
c. Scheduling packets out of the various queues and preparing them for transmission.

There are four types of queuing techniques commonly implemented

a. **First in first out (FIFO) queues**- Packets are transmitted in the order in which they arrive. There is just one queue for all the packets. Packets are stored in the queue when the network is congested and sent when there is no congestion. If the queue is full then packets are dropped.

b.Weighted Fair Queuing
Packets are classified into different "conversation messages" by inspection of the ToS value, destination and source port number, destination and source IP address etc. One queue is maintained for each "conversation". Each queue has some priority value or weight assigned to it (once again calculated from header data). Low volume traffic is given higher priority over high volume traffic. e.g. telnet traffic over ftp traffic. After accounting for high priority traffic the remaining bandwidth is divided fairly among multiple queues (if any) of low priority traffic. WFQ also divides packet trains into separate packets so that bandwidth is shared fairly among individual conversations. The actual scheduling during periods of congestion is illustrated through the following example [CISCO1]:

If there are 1 queue each of priority 7 to 0 respectively then the division of output bandwidth will be :

total = w0+w1+w2+w3+w4+w5+w6+w7 = Sw

priority 0 gets w0/Sw th of bandwidth, priority 1 gets w1/Sw th of bandwidth,
priority 2 gets w2/Sw th of bandwidth etc.

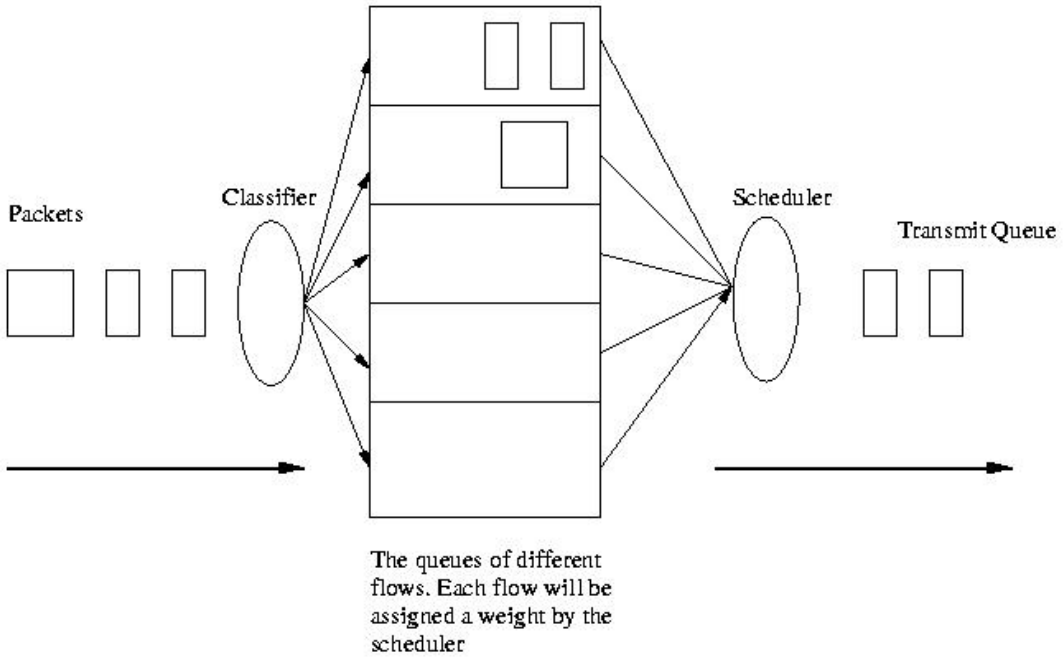


Figure 3 : Weighted Fair Queueing

The aim of WFQ is to ensure that low volume high priority traffic does get the service levels it expects. It also adapts itself whenever the network parameters change. WFQ cycles through the fair queues and picks up bytes proportional to the above calculation for

transmission from each queue. "WFQ acts as a preparator for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver guaranteed service. RSVP uses the mean data rate, largest amount of data the router will keep in the queue and the minimum QoS to determine bandwidth reservation." During congestion periods ordinary data packets are dropped but messages which have control message data still continue to get enqueued.

c. Custom Queuing

In this method separate queues are maintained for separate classes of traffic. The algorithm requires a byte count to be set per queue. That many bytes rounded of to the nearest packet is scheduled for delivery. This ensures that the minimum bandwidth requirement by the various classes of traffic is met. CQ round robins through the queues, picking the required number of packets from each. If a queue is of length 0 then the next queue is serviced. The byte counts are calculated as illustrated in the following example :

Suppose we want to allocate 20% for protocol A, 20% for protocol B, 20% for protocol C. Packet sizes for A is 1086 bytes, B is 291 bytes, C is 831 bytes.

Step1. Calculate % / size ratio : 20/1086 , 60 / 291 , 20 / 831
Step2. Normalize (by dividing by smallest number) : 1, .20619/.01842 , .02407 / .01842
Step3. Round upto nearest integer : 1, 12 ,2
Step4. Multiply each by corresponding byte size of packet : 1086 , 3492, 1662

Verify :

Step5. Add them : 1086 + 3492 + 1662 = 6240
Step6. 1086/6240 , 3492/6240 , 1662/6240 or 17.4 , 56 , 26.6 which are nearly equal to the ones at the top. CQ is a static strategy. It does not adapt to the network conditions. The system takes a longer while to switch packets since packets are classified by the processor card.

d. Priority Queuing

We can define 4 traffic priorities - high, medium, normal and low. Incoming traffic is classified and enqueued in either of the 4 queues. Classification criteria are protocol type, incoming interface, packet size, fragments and access lists. Unclassified packets are put in the normal queue. The queues are emptied in the order of - high, medium, normal and low. In each queue, packets are in the FIFO order. During congestion, when a queue gets larger than a predetermined queue limit, packets get dropped. The advantage of priority queues is the absolute preferential treatment to high priority traffic - so that mission critical traffic always get top priority treatment. The disadvantage is that it is a static scheme and does not adapt itself to network conditions and is not supported on any tunnels.

2.6 Link Efficiency Mechanisms [cisco3]

Applications like ftp create jumbograms - massive packets or trains of packets which move through the network like a single packet. These packets tend to congest the flow through the network and affect the higher priority traffic despite of themselves being low priority traffic. Link and Fragmentation Interleaving (**LFI**) strategies work with traffic queuing and shaping techniques to improve the efficiency of the QoS. CISCO implements LFI with MLP (Multilink point to point protocol, RFC 1717). MLP provides a method of splitting, recombining and sequencing datagrams across multiple logical channels. MLP uses LFI to break the jumbograms into smaller packets and interleave them with small sized packets of more high priority, interactive traffic. Incoming traffic is monitored for big packets, then they are broken down into smaller sized packets and enqueued using the various queuing policies. If we have WFQ operating at the output interface then these packets are interleaved and scheduled fairly, based on their assigned weights for transmission. To ensure correct transmission and reassembly, LFI adds multilink headers to the datagrams being transmitted.

Another strategy for improving link efficiency is **CPTR**- Compressed Real Time Protocol header - where the header of a RTP packet is compressed from 40 bytes to 2-5 bytes before transmission. The decompressor can easily reconstruct the headers since often they do not change and even if they do, the second order difference is constant.

2.7 Congestion Avoidance Mechanisms [cisco2]

Whereas congestion management deals with strategies to control congestion once it has sent in, congestion avoidance implements strategies to anticipate and avoid congestion in the first place. There are 2 popular strategies:

1. **Tail drop**: As usual at the output we have queues of packets waiting to be scheduled for delivery. Tail drop simply drops a incoming packet if the output queue for the packet is full. When congestion is eliminated queues have room and taildrop allows packets to be queued. The main disadvantage is the problem of TCP global synchronization where all the hosts send at the same time and stop at the same time. This can happen
because taildrop can drop packets from many hosts at the same time.

2. **Random Early Dropping**: **RED**strategies should only be employed on top of reliable transport protocols like TCP. Only then can they act as congestion avoiders. RED starts dropping packets randomly when the average queue size is more than a threshold value. The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold. After that a certain fraction - designated as mark probability denominator - of packets are dropped - once again randomly. The minimum threshold should be greater than some minimum value so that packets are not dropped unnecessarily. The difference between maximum and minimum threshold should be great enough to prevent
global synchronization.

3. **Weighted Random Early Dropping (WRED)** - is a RED strategy where in addition it drops low priority packets over high priority ones when the output interface starts getting congested. For IntServ environments WRED drops non-RSVP-flow packets and for Diff Serv environments WRED looks at IP precedence bits to decide priorities and hence which ones to selectively dump. WRED is usually configured at the core routers since IP precedence is set only at the core-edge routers. WRED drops more packets from heavy users than meager users - so that sources which generate more traffic will be slowed down in times of congestion. Non IP packets have precedence 0 - that is highest probability to be dropped. The average queue size formula is :

average = (old_average * 2^ (-n)) + (current_queue_size * 2^(-n))

where n is the exponential weight factor configured by the user. A high values of n means a slow change in the "average" which implies a slow reaction of WRED to changing network conditions - it will be slow to start and stop dropping packets. A very high n implies no WRED effect. Low n means WRED will be more in synch with current queue size and will react sharply to congestion and decongest ion. But very low n means that WRED will overreact to temporary fluctuations and may drop packets unnecessarily.

Back to Top

---

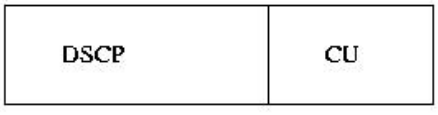
3. DIFFERENTIATED SERVICES (DiffServ) [diffserva]

According to this model network traffic is classified and conditioned at the entry to a network and assigned to different behavior aggregates[diffserva].Each such aggregate is assigned a single DS codepoint (i.e. one of the markups possible with the DS bits). In the core of the network packets are forwarded as per the per hop behaviors associated with the codepoints.

3.1 Key Features

DiffServ carves out the whole network into domains. A **DiffServ(DS) domain**is a continuous set of nodes which support a common resource provisioning and PHB policy. It has a well defined boundary and there are two types of nodes associated with a DS domain - **Boundary nodes**and **Interior nodes**. Boundary nodes connect the DS cloud to other domains. Interior nodes are connected to other interior nodes or boundary nodes - but they must be within the same DS domain. The boundary nodes are assigned the duty of classifying ingress traffic so that incoming packets are marked appropriately to choose one of the PHB groups supported inside the domain. They also enforce the **Traffic Conditioning Agreements (TCA)**between its own DS domain and the other domain it connects to .Interior nodes map the DS codepoints of each packet into the set of PHB's and impart appropriate forwarding behavior. Any non-DS compliant node inside a DS domain results in unpredictable performance and a loss of end to end QoS A DS domain is generally made up of a organization's intranet or an ISP - i.e. networks controlled by a single entity. DiffServ is extended across domains by **Service Level Agreement's (SLA)**between them. A SLA specifies rules such as for traffic remarking, actions to be taken for out-of-profile traffic etc. The TCA between domains are decided out of this SLA.

DS boundary nodes can be both **ingress**nodes and **egress**nodes depending on direction of traffic flow. Traffic enters the DS cloud through a ingress node and exits through a egress node. A ingress node is responsible for enforcing the TCA between the DS domain and the domain of the sender node. A egress node shapes the outgoing traffic to make it compliant with the TCA between its own DS domain the the domain of the receiver node.



DSCP (DiffServ Code Point) - 6 bits

CU (Currently unused) - 2 bits

DSCP = 101110 indicates EF PHB

Figure 4: The DS Byte

Unlike IntServ, DiffServ minimizes signaling by aggregation and per-hop behaviors. Flows are classified by predetermined rules so that they can fit into a limited set of class flows. This eases congestion from the backbone. The edge routers use the 8 bit ToS field, called the DS field in DiffServ terminology, to mark the packet for preferential treatment by the core transit routers. 6 bits of it are used and 2 are reserved for future use. Only the edge routers need to maintain per-flow states and perform the shaping and the policing. This is also desirable since the customer - service provider links are usually slow and so computational delay is not that much of a problem in these links. Therefore we can afford to do the computation intensive traffic shaping and policing strategies at the edge routers. But once inside the core of the service providers, packets need to be routed very fast and so we must incur minimum computational delay at any router/switch.

3.2 Per Hop Behaviors

"PHB is a description of the externally observable forwarding behavior of a DS node applied to a particular DS behavior aggregate" [diffserva]. The core routers in the DiffServ model need to only forward packets according to the specified **per-hop behaviors (PHB).**

If only 1 behavior aggregate occupies a link, the observable forwarding behavior will generally depend only on the congestion of the link. Distinct behavioral patterns are only observed when multiple behavioral aggregated compete for buffer and bandwidth resources on a node. A network node allocates resources to the behavior aggregates with the help of the PHBs. PHB's can be defined either in terms of their resources (buffer and bandwidth), or in terms of their priority relative to other PHB's or in terms of their relative traffic properties (e.g. delay and loss). Multiple PHB's are lumped together to form a PHB Group to ensure consistency. PHB's are implemented at nodes through some buffer management or packet scheduling mechanisms. A particular PHB Group can by implemented in a variety of ways because PHB's are defined in terms of behavior characteristics and are not implementation dependent.

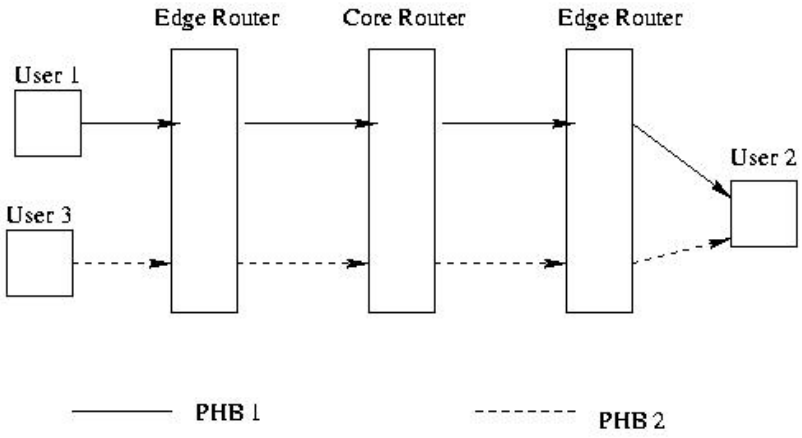


Figure 5 : The DiffServ Architecture

The standard for DiffServ describes PHB's as the building blocks for services. The focus is on enforcing service level agreements (SLA) between the user and the service provider. Customers can mark the DS byte of their traffic to indicate the desired service, or get them marked by the leaf router based on multifield classification (MF). Inside the core, traffic is shaped according to their Behavior Aggregates.. These rules are derived from the SLA. When a packet goes from one domain to another, the DS byte may be rewritten upon by the new networks edge routers. A PHB for a packet is selected at a node on the basis of its DS codepoint. The mapping from DS codepoint to PHB maybe 1 to 1 or N to 1. All codepoints must have some PHB associated with it. In absence of this condition, codepoints are mapped to a default PHB. Examples of the parameters of the forwarding behavior each traffic should receive are bandwidth partition and the drop priority. Examples of implementations of these are WFQ for bandwidth partition and RED for drop priority. 2 most popularly used PHB's are :

1. **Assured Forwarding**: It sets out-of-profile traffic to high drop priority. It has 2 levels of priority - four classes and 3 drop priorities inside each class. But the 4 classes are not implemented till date and only the 3 levels are implemented

2. **Expedited Forwarding**: It exercises strict admissions control and drops all excessive packets. Thus it prevents a queue from growing beyond a certain threshold. Forwarding is either based on priority of the packets or best effort. It guarantees a minimum service rate and has the highest data priority. So it is not affected by other PHB's.

3.2.1 Assured Forwarding [afphb]

"Assured Forwarding PHB Group provides forwarding of IP packets in N independent AF classes. Within each AF class, an IP packet is assigned one of M different levels of drop precedence"[rfc2597]. Current implementations support (N=)4 classes with (M=)3 levels of drop precedence in each class. At each DS node each of the 4 PHB classes is allocated a certain amount of resources (bandwidth, buffer). During periods of congestion, for packets in a particular PHB class, the higher the drop precedence of the packet, the higher its probability of being dropped. "Traffic in one PHB class is forwarded independently from packets in other PHB class - i.e. a DS node must not aggregate 2 or more traffic classes together ". Packets of the same microflow - packets belonging to the same AF class are never reordered at any DS node. At the edge of the DS domain, the traffic may be conditioned (shaped, discarded, increase or decrease of drop precedence values etc.).

The AF PHB is used to provide **Assured Services**to the customer, so that the customers will get reliable services even in times of network congestion. The customer gets a fixed bandwidth from the ISP - which is specified in their SLA. Then it is his responsibility to decide how his applications share the bandwidth.

3.2.2 Expedited Forwarding [efphb]

"EF PHB is defined as a forwarding treatment for a particular diffserv aggregate where departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate" [rfc2598]. Queues in the network are the reasons for loss, latency and jitter of the traffic. So to reduce loss, latency and jitter, one should reduce queues in the system. A queue-free service will guarantee bounded traffic rates. The EF traffic "should" receive this minimum rate of departure irrespective of how much other traffic at the node might be. The traffic should be conditioned (via policing and shaping the aggregate) so that the maximum arrival rate at any node is less than the minimum departure rate. Formally, the average departure rate measured over a time period equal to the time required to send a MTU at the configured rate should be greater than or equal to the configured rate. DS ingress routers must negotiate a rate less than this configured rate with adjacent upstream routers. To enforce this condition it must strictly police all incoming traffic. Packets violating the condition are dropped. The default EF configured rate is 0 - all packets marked with EF are dropped.

The EF PHB is used to provide **Premium Service**to the customer. It is a low-delay, low-jitter service providing near constant bit rate to the customer. The SLA specifies a peak bit rate which customer applications will receive and it is the customers responsibility not to exceed the rate, in violation of which packets are dropped.

EF PHB is implemented in a variety of ways. For example, if a PQ is used (as described in section 2.5) - then there must be a upper bound (configured by the network administrator) on the amount of EF traffic that should be allowed. EF traffic exceeding the bound is dropped.

3.3 Traffic Classification

"Traffic classification policy identifies the subset of network traffic which may receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates (by DS codepoint remarking) within the DS domain."[diffserva]. Packet classifiers use the information in a packet's header to select them. There are 2 types of classifiers :

1. **Behavioral Aggregate Classifiers** - These select packets on the basis of their DS codepoints.
2.**Multi Field Classifiers**- They select packets based on values of multiple header fields.

Classifiers send the packet to the conditioner module for further processing. Classifiers are configured by some management procedure on the basis of the relevant TCA. It is also the classifier's job to authenticate the basis on which it classifies packets.

3.4 Traffic Conditioning

The 4 functions performed under traffic conditioning are - metering, shaping, policing and/or remarking. The duty of traffic conditioning is to ensure that the traffic entering a DS domain complies with the TCA, between the sender's domain and the receiver's domain, and the domain's service provisioning policy. The conditioner of a DS boundary node marks a packet with its appropriate codepoint.

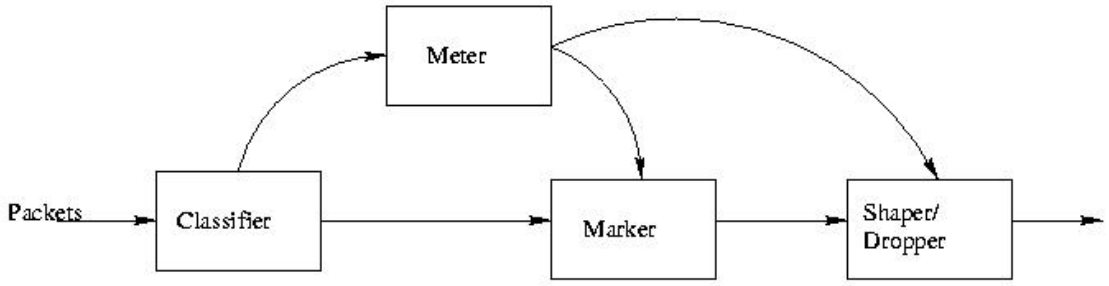


Figure 6: Various components of DiffServ

1**. Meters -** The conditioner receives packets from the classifier and uses a "meter" to measure the "temporal properties" of the stream against the appropriate traffic profile from the TCA. Further processing is done by the markers, shapers and policers based on whether the packet is in or out-of-profile. The meter passes this information to the other components along with the packet.
2. **Markers-** This marks a packet by setting the DS value to a correct codepoint (in its header). Thus a packet is categorized into a particular behavior aggregate. When a marker changes marks a packet which has already been marked, then it is said to "remark" the packet. The marker may be configured by various policies.
3. **Shapers -** They buffer the traffic stream and increase the delay of a stream to make it compliant with a particular traffic profile. Packets might be discarded if there is crunch of buffer space.
4. **Droppers**- As their name suggests, they drop packets of a stream to make the stream profile compliant. Droppers can be implemented as special case of a shaper with buffer size = 0.

Back to Top

---

4. MULTIPROTOCOL LABEL SWITCHING (MPLS) [mplsa]

MPLS combines the best of both worlds - ATM's circuit switching and IP's packet routing. It is a hybrid technology which enables very fast forwarding in the core and conventional routing at the edges. Packets are assigned a label at the entry to a MPLS domain, which is often the core backbone of a provider, and are switched inside the domain by a simple label lookup. The labels determine the quality of service the packet gets. The packets are stripped of the labels at the egress router and might be routed in the conventional fashion thereafter before it reaches its final destination..

In conventional IP routing, the next hop for a packet is chosen by a router on the basis of the packet's header information and the result of running a network layer routing algorithm. Choosing the next hop at the routers is thus a composition of 2 functions :

1. Partitioning the whole set of of possible packets into Forwarding Equivalence Classes (FEC).
2. Mapping each FEC to a next hop

The mapping of packets to FEC's is done at every router where largest prefix match algorithms is used to classify packets into FECs. In MPLS the assignment is done only once - at the entry of the packet in the MPLS domain (at the ingress router). The packets are assigned a fixed length value - a "label" - depending upon the FEC category to which it belongs & this value is sent alongwith the packet. At later hops, routers and switches inside the MPLS domain **donot have to use complex search algorithms, but** simply use this **label** to index into their routing tables which gives the address of the next hop for the packet, and a new value for the label.The packets label is replaced by this new label and the packet is forwarded to the next hop. This is exactly like **switching**. It is **multiprotocol**since its techniques can be used with any network layer protocol.

4.1 Labels

"A label is a short fixed length locally significant identifier which is used to identify a forwarding equivalence class. The label which is put on a particular packet represents the forwarding equivalence class to which that packet is assigned" [mplsa]. The label is "locally significant" in the sense that 2 routers agree upon using a particular label to signify a particular FEC, among themselves. The same label can be used to distinguish a different FEC by another pair of routers. In the same spirit, the same FEC can be assigned a different label by other routers. A label L is a arbitrary value whose binding to FEC F is local to routers R1 and R2 (say). The label itself is a function of the packet's Class of Service and the FEC. A router which uses MPLS is called a **Label Switching Router (LSR)**. The path taken by a packet after being switched by LSRs through a MPLS domain is called a **Label Switched Path (LSP).**

If routers R1 and R2 agree to bind a FEC F to label L for traffic going from R1 to R2, then R1 is called the "upstream" label switched router (LSR) and R2 is the "downstream" LSR. If R1 & R2 are not adjacent routers, then R1 may receive packets labeled L from R2 & R3. Now a potential error condition appears if L is bound to FEC F12 between R1 and R2, and if L is bound to FEC F13 between

R1 and R3, and F12 is not equal to F13. To prevent such confusion, routers like R1 should agree to a one-to-one mapping between labels and FEC's.

In MPLS, the final decision for the binding of a label L to a FEC F is made by the downstream LSR with respect to that binding. (i.e. who will be receiving the traffic). The downstream LSR then announces the binding to the upstream LSR. Thus labels are distributed in a bottom-up fashion. This distribution is accomplished with the **label distribution protocols** [nortel3]**.** LSRs using a label distribution protocol among themselves to exchange label-to-FEC-binding information are known as "label distribution peers". The label distribution protocol also comprises of the procedures used by LSR's to learn about each others MPLS capabilities [mplsa].

4.2 The Label Stack

Instead of just a single label, a labeled packet usually carries multiple labels organized as a Last In First Out stack. This is called as the label stack. The labels are either in the form of an encapsulation or some form of a markup inside the packet header. At a router, forwarding decisions are always based on the topmost label on the stack irrespective of what other labels it might be carrying or might have carried. The labels are numbered inside out - from the bottom of the stack to the top. The label stack is useful to implement tunneling and hierarchy.

The important components of the label stack are

1. The **Next Hop label Forwarding Entry**(**NHLFE**) : This is used to forward a labeled packet. It consists of

    - The next hop for the packet
    - The actions to change the label stack - this can be one of the following
        - replacing the topmost label with a new label
        - popping off the top label from the stack
        - replacing the top label by a new label and pushing some additional labels into the stack

It may also consist of

    - The encapsulation to be used while transmitting the packet
    - The encoding procedure for the label stack
    - Any additional information needed to handle the packet properly

If a packet's next hop is the current LSR then the LSR pops the topmost label and takes appropriate action based on the label below.

2. The **Incoming Label Map (ILM) :** This maps each incoming packet's label to a set of NHLFE's. If the cardinality of the resulting NHLFE set is more than 1 then exactly one NHLFE is chosen from the set.

3. The **FEC to NHLFE Map (FTN)**: This is used to label and forward unlabelled packets. Each FEC is mapped to a set of NHLFE's by the FTN. If the cardinality of the resulting NHLFE set is more than one then exactly one NHLFE is chosen from the set.

4. **Label Swapping**: This is the combination of the above mentioned procedures (1,2 & 3) to forward packets. Depending on the type of the packet, forwarding can be of the following 2 categories :

- Labeled packet - A LSR gets the packet, uses the ILM to map the label to a NHLFE , uses the information in the NHLFE to perform operations on the label stack and finally forward the packet to the next hop as specified in the NHLFE.
- Unlabelled packet - A LSR gets the packet, uses the header information to map the packet to a FEC, then uses the FTN to get a NHLFE. Then the information in the NHLFE is used to forward the packet to its next hop and perform operations on the label stack (like encoding the new label stack).

4.3 Label Switched Path (LSP)

"A level m LSP for a packet P is the sequence of routers" [mplsa] :

1. which begins with a LSP ingress LSR that pushes the level m label on the stack
2. "all of whose intermediate LSR's make their forwarding decision by label switching on a level m label".
3. which ends at a LSP egress LSR, where the forwarding decision is taken on a level m-k (k>0) label, or some non-MPLS forwarding procedure.

A sequence of LSR's is called the "LSP for a particular FEC F" if it is a LSP of level m for a particular packet P when P's level m label is a label corresponding to FEC F.

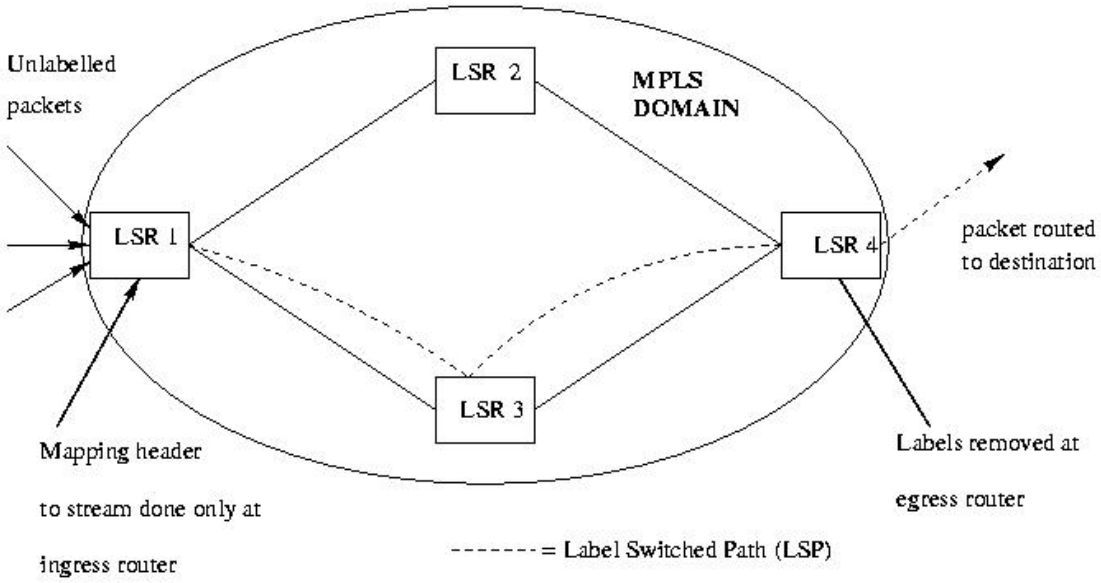


Figure 7: The MPLS Architecture

The level m label of the level m LSP < R1,R2,...Rn > can be popped at either the LSR Rn or the LSR R(n-1). If it is popped at LSR R(n-1) then the egress LSR Rn will have to do only 1 label lookup. Otherwise Rn will have to to a label lookup followed by either another label lookup(when Rn is a intermediate router for level m-1 LSP) or an address lookup (in case m=1). A penultimate node must pop the label if it is explicitly requested by the egress node or if the next node does not support MPLS.

4.4 Route Selection

Route selection refers to the problem of mapping a FEC to a LSP. A LSP can be chosen for a particular FEC in 2 ways :

1. **Hop by hop routing**- the route is chosen at each LSR in the same manner as in conventional IP forwarding.
2. **Explicit routing** - Here the boundary ingress LSR specifies the particular LSP that a packet will take through the MPLS cloud. This is done by explicitly declaring all the LSP's along the path. This is almost like TCP/IP source routing although it has several advantages over TCP/IP source routing. If the whole LSP is declared then it is "strictly explicitly routed". Else for a partial LSP stated, it is called "loosely explicitly routed".

4.5 Advantages over IP routing

The advantages of MPLS over IP routing are
1. MPLS forwarding is done by switches in ASIC's capable of doing simple table lookup and replacement but which cannot do computation intensive jobs like the prefix search algorithm in good time.
2. The ingress router can distinguish between packets coming at different ports or on any other criteria which cannot be learnt from the header data by assigning them to different FECs. In IP routing, only the header information forms the basis of forwarding decisions.
3. Packets entering the network from different ingress routers can be differentiated. This cannot be done in IP since routers address is not carried as part of header data.
4. Algorithms for mapping FECs to labels may become more and more complex without affecting the network performance too much since it is a one time affair at the ingress routers.
5. Sometimes certain packets are desired to be routed along certain specific paths which is decided before hand or when the packet enters the network. In IP, this requires source routing. But in MPLS this is easily done with the help of a label without the packet becoming too large by carrying all its hops addresses.

MPLS Tunneling

Sometimes a router R1 (say) wishes to deliver a packet straightway to another router R2 (say) which is not one of its next hop routers for that packet and neither is R2 its final destination. So to implement the tunnel, R1 pushes a additional label at the top of the label stack. Of course this label has to be agreed upon beforehand by all the routers from R1 to R2 who will be handling this packet. In its turn, R2 pops off the label once it receives the packet. This method of tunneling is much faster than in IP where the data is encapsulated in a IP network layer packet. The tunnel in this case would be a LSP (Label Switched Path) <R1,R11,R12, ... , R2>.

Tunnels can be hop by hop routed LSP tunnel or explicitly routed lsp tunnel [mplsa].The label stack mechanism allows LSP tunneling at any depth.

Back to Top

---

5. CONSTRAINT BASED ROUTING [xiao99]

This is a extension of **QoS based routing** which takes into account the viability of a route with respect to satisfying certain QoS requirements and also other constraints of the network such as policy. The goals are twofold :
1. Select routes which can meet the QoS requirements
2. Increase utilization of network and load distribution- a longer and lowly congested path may be better for QoS demanding traffic than the shortest and highly congested path.

Routers have to exchange various link state information and compute routes based on the information on the fly. The most popularly used link state distribution protocol is to extend link state information contained in the advertisements of OSPF. But this congests the links even further due to frequent link state advertisements. The way to reduce this congestion is to advertise only when there has been some substantial change in the network parameters - like a sharp fall in bandwidth etc. The algorithm to calculate routing tables is based on the twin parameters of **hop count and bandwidth**. The order is O(N*E) where N is the hop count and E is number of links in the network. The reason for choosing these 2 particular factors is as follows. Hop count is important since the more hops a traffic traverses, the more resources it consumes. So it is a important metric to consider while making routing decisions. A certain amount of bandwidth is also desired by almost all QoS sensitive traffic. The other QoS factors like delay and jitter can be mapped to hop count and bandwidth. In constraint based routing routing tables have to be computed much more frequently than with dynamic routing since routing table computations can be triggered by a myriad of factors like bandwidth changes, congestion etc. Therefore the load on routers is very high. To **reduce load**the following are implemented :

1. A large timer value to reduce frequency of the computations.
2. Choose bandwidth and hop count as constraints.
3. Preprocessing : Prune links beforehand which are obviously out of contention to be a potential route for certain kinds of flows. E.g. A 10 Mbps traffic is not likely going to be routed on a 1 Mbps link.

The **advantages**of using constraint based routing is :
1. Meeting Qos requirements of the flows better
2. Better network utilization

The **disadvantages**are :
1. High computation overhead
2. Big routing table size
3. A long path may consume more resources that the shortest path.
4. Unstability in the routes : Since routing tables are being updated all too often, the routes remain in the transient state much of the time and while routes are being updated, the protocol might not be sensitive to further network changes. This may lead to race conditions.

There is a certain tradeoff involved in constraint based routing between resource conservation and load balancing. Better load balancing may lead to traffic being diverted to less congested and longer links due to which the traffic travels over more hops and consumes more resources. The compromise is to use the shortest path when there is heavy network loads and use the widest path when the load is minimum.

Back to Top

---

6. SUBNET BANDWIDTH MANAGER (SBM) [sbm]

The SBM technology provides a signaling mechanism to reserve resources in erstwhile best-effort LAN technologies to support RSVP flows. Thus it provides a extension of Layer 3 reservation protocols - RSVP (which is independent of subnetwork technologies) to Layer 2 and thus attempts to provide better end-to-end QoS.

"Some Layer 2 technologies have always been QoS-enabled, such as Asynchronous Transfer Mode (ATM). However, other more common LAN technologies such as Ethernet were not originally designed to be QoS-capable. As a shared broadcast medium or even in its switched form, Ethernet provides a service analogous to standard "best effort" IP Service, in which variable delays can affect real-time applications. However, the [IEEE] has "retro-fitted" Ethernet and other Layer 2 technologies to allow for QoS support by providing protocol mechanisms for traffic differentiation.

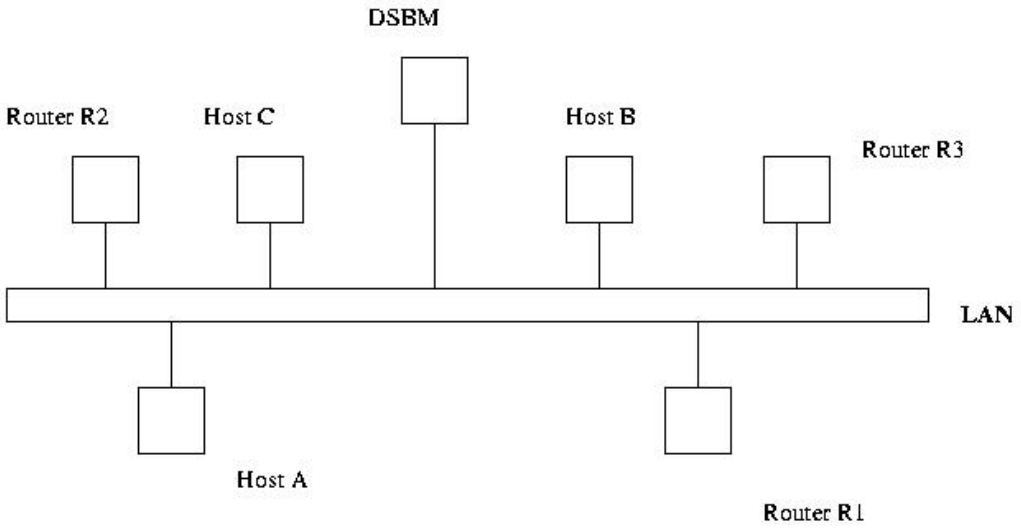


Figure 8: A SBM managed LAN

The IEEE 802.1p, 802.1Q and 802.1D standards define how Ethernet switches can classify frames in order to expedite delivery of time-critical traffic. The Internet Engineering Task Force [IETF] Integrated Services over Specific Link Layers [ISSLL] Working Group is chartered to define the mapping between upper-layer QoS protocols and services with those of Layer 2 technologies, like Ethernet. Among other things, this has resulted in the development of the "Subnet Bandwidth Manager" (SBM) for shared or switched 802 LANs such as Ethernet (also FDDI, Token Ring, etc.). SBM is a signaling protocol [SBM] that allows communication and coordination between network nodes and switches in the" SBM framework "and enables mapping to higher-layer QoS protocols" [stardust].

6.1 Basic Algorithm

"A **Designated SBM (DSBM)**is a protocol entity that resides in a L2 or L3 device and manages resource on a L2 segment. At most one DSBM exists for each L2 segment. A managed segment is a segment with a DSBM present and responsible for exercising admission control over requests for resource reservation. A **managed segment**includes those interconnected parts of a shared LAN that are not separated by DSBMs" [sbm]. The DSBM is responsible for admission control over the rescue reservation requests originating from the DSBM clients in its managed segment. More than one SBM might reside on a single segment. One of the SBM's is elected to be a DSBM. One DSBM can preside over multiple L2 segments provided they are joined by some SBM transparent device.

The steps in the algorithm are

1. DSBM Initialization : The DSBM gathers information regarding resource constraints, such as the amount of bandwidth that can be reserved, from each of its managed segment. Usually this information is configured in the SBM's and is static.
2. DSBM Client Initialization : Each client in the managed domain searches for the existence of a DSBM on each of its interfaces. If there are no DSBMs the client itself might participate in a election to become a DSBM for that segment.
3. Admission Control : DSBM clients do the following :
3a. Whenever they receive a RSVP PATH message, they forward it to their DSBM instead of the destination address. The DSBM modifies the message, builds or adds to a PATH state, adds its own L2/L3 address to the PHOP object, and then forwards it to its destination address.
3b. When a client wishes to issue a RESV message, it looks up the DSBM's address from the PHOP object of the PATH message and sends the RESV message to the DSBM's address.
3c. The DSBM processes the RESV message - if resources are not available then a RESVError message is issued to the RESV-requester, else if resources are abundant and the reservation is made then the DSBM forwards the packet to the PHOP based on its PATH state.
3d. If the domain encompasses more than one managed segment, then PATH messages propagate through the DSBM of each segment and PATH states are maintained at each DSBM. The RESV message succeeds in reaching its destination only if admission control associated with it succeeds at each DSBM.

6.2 The Election Algorithm

Each SBM is assigned a "SBM priority" which it uses to contest in a DSBM election round. While starting up, a SBM listens for DSBM advertisements to find out if there are any DSBM's in its L2 domain. If no DSBM exists then the SBM initiates a election

process. It sends out a DSBM_WILLING message containing its "SBM priority" and its IP address. Each SBM compares the priorities of incoming DSBM_WILLING messages. If they find their own priority to be more than some incoming priority then they also send out DSBM_WILLING messages. After a entire round of message passing the SBM finding its own priority to be the highest, send out a I_AM_DSBM message. All SBMs receiving this message then enter a Idle state.

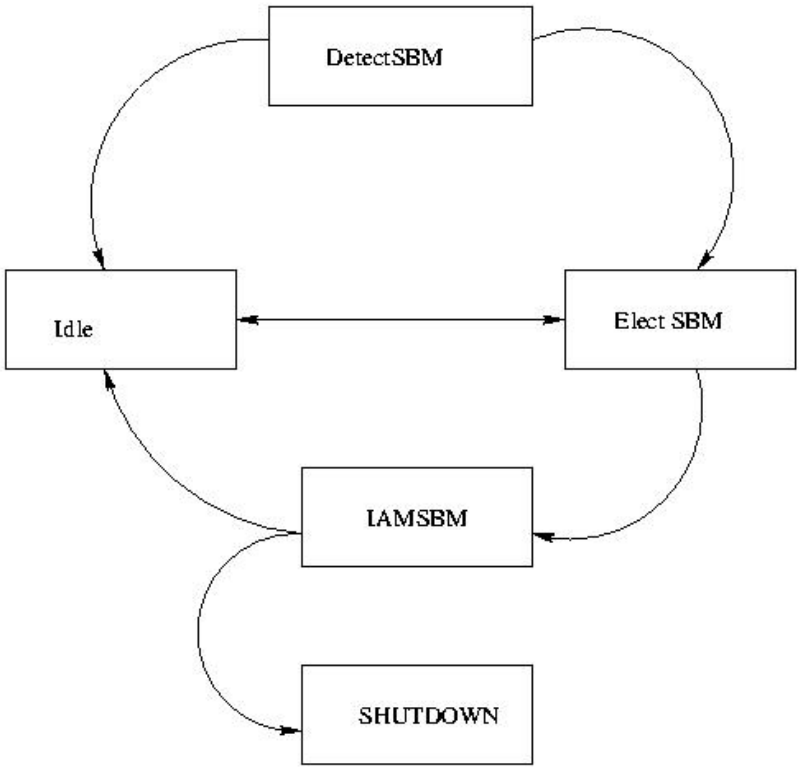


Figure 9 : State Diagram of a SBM

While starting up, if a SBM finds that a DSBM already exists in its domain then it keeps quite till such time as election of a DSBM becomes necessary.

6.3 non DSBM SBM's

The SBMs which donot get elected to be the DSBM serve 2 useful purposes

1. They keep on listening to the periodic announcements from the DSBM In case the DSBM server crashes or becomes non functional for some reason, they initiate a fresh round of election and elect a new DSBM. Thus they enhance the fault tolerance of the domain.
2. The load of the election algorithm is distributed among the various SBMs. Each of the SBM's send out its own priority only if it finds its priority higher than the incoming priorities.
3. If a SBM connects two otherwise disjoint L2 domains, then it makes it feasible for each one to have a separate DSBM, otherwise both the domains might have selected a single DSBM. The two DSBMs have a simpler task to perform as each of them would be presiding over a simpler network.

Back to Top

---

7. DISCUSSIONS : Advantages and disadvantages of various protocols

Best effort delivery has no guarantee of service. There is no definite bound on parameters such as delay jitter, bandwidth, delay etc. Therefore applications that require QoS cannot run over purely best effort networks. IntServ was proposed to reserve resources in a best effort network in advance so that selected flows can enjoy the priviledge of being treated with more resources. But they had the following disadvantages : [rj99]

IntServ Disadvantages :

1. Intserv makes routers very complicated. Intermediate routers have to have modules to support RSVP reservations and also treat flows according to the reservations. In addition they have to support RSVP messages and coordinate with policy servers.
2. It is not scalable with the number of flows. As the number of flows increases, routing becomes incredibly difficult. The backbone core routers become slow when they try to accomodate an increasing number of RSVP flows.
3. RSVP is purely receiver based. The reservations are initiated by willing receivers. But in many cases, it is the sender who has the onus of initiating a QoS based flow. Thus RSVP fails to accomodate for such flows.
4. RSVP imposes maintenance of soft states at the intermediate routers. This implies that routers have to constantly monitor and update states on a perflow basis. In addition the periodic messages sent add to the congestion in the network
5. There is no negotiation and backtracking.

To solve the problem of scalability, DiffServ was proposed. It introduced the concept of "aggregating flows" so that the number of flows in the backbone of a provider's network remain managably low. But it also has several disadvantages :

DiffServ Disadvantages:

1. Providing quality of service to traffic flows on a perhop basis often cannot guarantee end-to-end QoS. Therefore only Premium service wil work in a purely DiffServ setting.
2. DiffServ cannot account for dynamic SLA's between the customer and the provider. It assumes a static SLA configuration. But in the real world network topologies change very fast.
3. DiffServ is sender-oriented. Once again, in many flows, the receiver's requests have to be accounted for.
4. Some long flows like high bandwidth videoconferencing requires per-flow guarantees. But DiffServ only provides guarantees for the aggregates.

MPLS has been proposed to be a combination of the better properties of ATM and IP. It proposes switching at the core based on labels on IP packets.

Constraint based routing and other standards

Constraint based routing is used to complement various properties of IntServ, DiffServ and MPLS. It is used to compute paths so that Qos requirements are met for DiffServ flow. MPLS uses this information to lay down its LSP's. On the other hand, MPLS's perflow statistics help constraint based routing to find out better paths. Thus they share a mutually symbiotic relationship and provide a rich set of traffic engineering capabilities. The paths found by constraint based routing can be used to route RSVP PATH messages, so that subsequent RSVP flows traverse the best possible paths. IntServ, DiffServ and RSVP are essentially transport layer protocols, constraint based routing is a network layer protocol and MPLS is a network cum link layer protocol. [xiao99]

Back to Top

---

8. CONCLUSION: Network Architectures of the future

In the emerging QoS framework, packets are switched very fast inside the core. The core backbone infrastructure is made out of high speed fiber. Dense Wavelenth Division Multiplexing (DWDM) would operate in the fibers. Core routers use MPLS to switch packets. ATM switches can also be used inside the core to complete the switching process. At the edge routers, all policy related processing such as classification, metering, marking etc. would take place. Resource provisioning at the edge cloud would be done with RSVP and IntServ. At the core DiffServ might be used to keep the number of traffic aggregates at a manageable level. The basic philosophy of the architecture is to remove as much of computation intensive functions as possible from the backbone routers, and push these functions towards the edges - the edge routers. That way, the core routers would be free to do high speed forwarding of the packets and they would remain simple to manage.

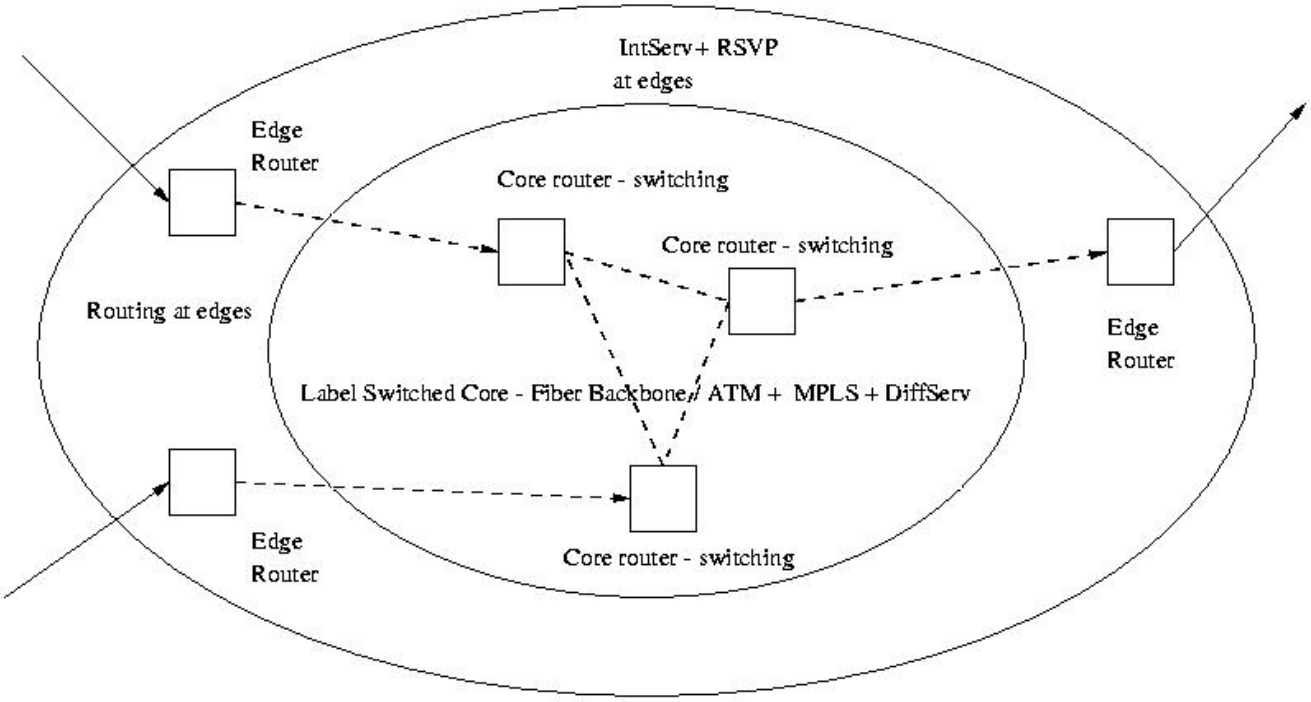


Figure 10 : Network Architecture

Back to Top

Summary

Delivering Quality of Service to network traffic is of pivotal importance for mission critical applications in this wired world of today and tomorrow. Integrated services tried to solve the problem by resource reservation. But it failed due to scalability issues. Diffserv took to prioritization and aggregation to limit the number of traffic classes in the backbone. But DiffServ also failed because end-to-end QoS was difficult to guarantee by aggregating flows. MPLS is a halfway house between IntServ and DiffServ and supports routing at the edges (IP philosophy) where IntServ can be used, and switching at the core (ATM philosophy), using DiffServ techniques. Subnet Bandwidth Management is a technique to support RSVP flows at the L2 layer and to seamlessly integrate L2 and L3 resource reservation techniques and provide better endtoend QoS. ATM had inbuilt QoS mechanisms. But making end systems ATM compliant was too expensive. So IP stayed at the edges. In the new century, we can expect to find IntServ in the edges. In the core, packets would be forwarded using fast switching technologies. DiffServ would be operating in the core to keep the number of flows managable.

Back to Top

References

1. **[nortel1]**IP QoS - A Bold New Network Nortel / Bay Networks White Paper, Sept 1999, 24 pages
http://www.nortelnetworks.com/prd/isppp/collateral/ip_qos.pdf

A tutorial on Nortel's view of future networks and the prevalent network architecture.

2. **[gqos]**Specification of Guaranteed Quality of Service (RFC 2212), 19 pages
http://www.rfc-editor.org/in-notes/rfc2212.txt

Describes the guaranteed quality of service class in intserv.

3. **[mplsa]**Multiprotocol Label Switching Architecture draft-ietf-mpls-arch-06.txt, August 1999, 60 pages

Describes the entire mpls concept and the basic architecture.

4. **[nortel3]**IP Traffic Engineering using MPLS Explicit Routing in Carrier Networks, Nortel NetworksWhite Paper April 1999, 8 pages
http://www.nortelnetworks.com/products/library/collateral/55046.25-10-99.pdf

Describes various MPLS features and label distribution protocols.

5. **[clqos]**Specification of the Controlled-Load Network Element Service (RFC 2211), 16 pages
http://www.rfc-editor.org/in-notes/rfc2211.txt

The formal specification of controlled load service of intserv.

6. **[xiao99]**Internet QoS : the Big Picture Xipeng Xiao & Lionel M. Ni, IEEE Network, January 1999,25 pages .

Describes the major QoS issues and protocols.

7. **[rsvp2]** Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification (RFC 2205), 110 pages
http://www.rfc-editor.org/in-notes/rfc2205.txt

This document gives the formal specification for RSVP.

8. **[ciscorsvp]**Resource Reservation Protocol (RSVP) CISCO White Papers, Jun 1999 , 15 pages

A very concise and to the point summary of RSVP.

9. **[nortel2]** Preside Quality of Service Nortel Networks Position Paper, 11 pages

Describes the future network architecture and support of QoS.

11 **[CISCO1]** Congestion Management Overview, CISCO White Papers, 1999, 14 pages http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart2/qcconman.htm

Describes various queueing techniques as implemented in CISCO IOS.

12 **[CISCO2]** Congestion Avoidance Overview, CISCO White Papers, 1999, 16 pages
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart3/qcconavd.htm

Describes various RED techniques as implemented in CISCO IOS.

13. **[CISCO3]** Link Efficiency Mechanisms, CISCO white papers, 1999, 8 pages
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart6/qclemech.htm

LFI implementation as in CISO IOS product.

14. **[CISCO4]** QoS overview, CISCO white papers 1999. 24 pages
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcintro.htm

A excellent overview of QoS capabilities in CISCO's IOS product + definitions.

15. **[diffserva]** An Architecture for Differentiated Services RFC 2475, 36 pages
http://www.rfc-editor.org/in-notes/rfc2475.txt

This described the Diff Serv architecture and model in detail

16. **[afphb]** Assured Forwarding PHB Group RFC 2597, 10 pages
http://www.rfc-editor.org/in-notes/rfc2597.txt

The AF PHB is described here

17. **[efphb]** Expedited Forwarding PHB Group RFC 2598, 10 pages
http://www.rfc-editor.org/in-notes/rfc2598.txt

The EF PHB is described here

18. **[sbm]** A Protocol for RSVP-based Admission Control over IEEE 802-style networks draft-ietf-issll-is802-sbm-09.txt, 67 pages

The Subnet Bandwidth Manager is proposed here

19. **[stardust]** QoS Protocols and Architectures, Stardust White Paper, 17 pages
http://www.stardust.com/qos/whitepapers/protocols.htm

A brief tutorial on the common QoS protocols is mentioned here.

20. **[rsvp]**D. Durham, R. Yavatkar, Inside the Internet's Resource Reservation Protocol, John Wiley and Sons, 1999, 351 pages

Provides excellent coverage of all aspects of RSVP and some topics of IntServ. Excellent figures.

21. **[rj99]**QoS over Data Networks, Raj Jain, CIS 788 handouts, The Ohio State University, Fall 99, 4 pages of slides (6 slides per page) http://www1.cse .wustl.edu/~jain/cis788-99/h_6qos.htm

Provides excellent overview of recent advances in the area of Quality of Service issues.

Back to Top

List of Acronyms

AF	Assured Forwarding
COPS	Common Open Policy Service
DiffServ	Differentiated Services
DSBM	Designated Subnet Bandwidth Manager
EF	Expedited Forwarding
IntServ	Integrated Services - IETF Standard
LDP	Label Distribution Protocol
LFI	Link Fragmentation and Interleaving
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multi Protocol Label Switching
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHP	Per Hop Behaviour
QoS	Quality of Service
RED	Random Early Dropping
RSVP	Resource Reservation Protocol
SBM	Subnet Bandwidth Manager
SLA	Service Level Agreement
TCA	Traffic Conditioning Agreement
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Dropping

Back to Top

Last Modified : 19th November 1999.
Note : This paper is available online at http://www.cse.wustl.edu/~jain/cis788-99/qos_protocols/index.html