DiffServ and IntServ

6.829 Fall 2001
Notes by Matt Lepinski & Peter Portante

December 10, 2001

There are two approaches to QoS architectures referred to hereafter as Integrated Services (Int Serv) and Differentiated Services (Diff Serv). Under Int Serv, the QoS is explicitly requested, using some signaling mechanism (e.g. RSVP), and is characterized as an "end-to-end" solution. Under Diff Serv, the QoS is implicitly requested and provided on a per-hop basis, usually through Service Level Agreements between ISPs and their customers.

The [CF98] paper covers the Diff Serv topic, while the [CSZ92] paper covers the Int Serv topic. Note that the methodology for implementing these two service models in the internet today is a bit different from the descriptions in the paper. The papers provide an overview of some of the issues.

Both schemes exist to provide service beyond Best Effort. Scalability is a goal of both architectures.

1 Differentiated Services (Diff Serv, [CF98])

The goal of Diff Serv is to allow ISPs to make more money by offering service classes which are better than best effort. The idea behind Diff Serv is similar to airline ticket pricing, where airlines charge more to first and business class passengers who in return receive preferential treatment over those who pay the standard fare. Unlike Int Serv, which attempts to provide applications with end-to-end service guarantees, with Diff Serv, ISPs provide customers only with certain *per hop behaviors* and make no claims about end-to-end performance.

However, it is reasonable for customers to think that preferential treatment by their local ISP will result in better overall performance because internet bottlenecks frequently occur at the edges. Additionally, Diff Serv has the following advantages over Int Serv:

- Diff Serv is easier to implement than Int Serv because per hop behaviors can be provided without cooperation from upstream or downstream ISPs.

- Since Diff Serv guarantees are relative (i.e. Service Class 1 receives better service than Service Class 2) Diff Serv can provide this differentiation for any number of flows. This means that no admission control is needed.

- Diff Serv Service Level Agreements (SLA) fit nicely into the current ISP business model.

1.1 Diff Serv Implementation

Diff Serv policy is implemented within a Differentiated Services Region. A DS Region is a group of routers which all use the same Diff Serv policy (typically all the routers belonging to a single ISP). Each DS Region has a set of traffic classes (6 is common in practice) and a policy for associating a traffic aggregate with a traffic class. A Diff Serv Router contains the following pieces:

1. A Classifier which looks at each incoming packet and determines which traffic class it belongs in.

2. Multiple Queues, one for each traffic class.

3. A scheduler (using some scheduling algorithm like weighted fair queueing) that decides when each queue gets to send packets.

4. A Meter which determines the the rate of a particular traffic aggregate. This is important because most SLAs specify that a traffic aggregate gets a certain class of service provided that the traffic aggregate doesn't exceed some load. If the rate for a particular traffic aggregate is too high (it is out of profile) then the excess packets either need to be dropped or the traffic needs to be shaped. (Shaping traffic involves holding excess packets in a buffer and letting them trickle out at the rate specified in the SLA).

5. A Dropper/Shaper to deal with traffic which the meter determines to be out of profile

1.2 Scalability

Unfortunately, correctly classifying and metering each incoming packet takes valuable cycles and so it is generally not feasible to do this work at every router within an ISP. Typically, edge routers have extra cycles available for these kind of tasks but high-speed core routers do not. The solution to this problem is to use dynamic packet state (similar to the mechanism used in the Core Stateless Fair Queueing paper). Edge routers do classification and metering and then have a Marker which sticks the result of the classification into the IP header (the portion of the IP header which is used is referred to as the Diff Serv Code Point (DSCP)). The entire process of classifying, metering, marking and shaping/dropping packets in the edge router is known as traffic conditioning.

1.3 Dealing with Excess Traffic

As noted earlier, the Meter in the edge router measures the rate of a traffic aggregate and determines if traffic is out of profile. One solution is to simply drop every packet that is out of profile at the edge router. However, traffic is very bursty and since not all sources are continually sending at their maximum rate, it is possible that the system is able to handle the excess packets. Therefore, dropping at the edge router is often undesirable. The solution then is to let all of the packets through at the edge, but to mark those packets which are out of profile so that they can be dropped later if the network does not have the capacity to handle the extra traffic.

We then want some mechanism for a bottleneck core router to give preferential treatment to those packets which are marked as being in profile. The easiest solution would be to maintain separate queues for in profile and out of profile packets. This is a bad idea because often a particular flow contains both in profile and out of profile packets and so maintaining separate queues would lead to a large amount of reordering. The solution is to run RED on each queue in the router but to have RED treat each queue as two virtual queues. This algorithm is known as RIO (RED IN/OUT).

RIO maintains a variable q_{ave-in} which is the average number of in profile packets in the queue. RIO then uses this variable along with the RED parameters MIN_{in} and MAX_{in} to determine when to drop in profile packets. Additionally, RIO maintains a variable q_{tot} which is the average size of the queue (including both in profile and out of profile packets). RIO uses this variable along with the RED parameters MIN_{out} and MAX_{out} ($MIN_{out} < MIN_{in}$ and $MAX_{out} < MAX_{in}$) to determine when to drop out of profile packets.

1.4 In Practice

In practice, Diff Serv implementations use six traffic classes. These classes include one Expedited Forwarding (EF) class, Four Assured Forwarding Classes and one Best Effort class. Standard implementations have the following properties:

- No reordering of packets within a traffic class.
- No standard for bandwidth allocation between classes.
- For each class, there are three drop precedences.
- Expedited Forwarding has the property that whenever a packet is in this queue it gets sent out immediately. Note: This requires that edge routers drop all out of profile EF packets (since core routers never drop EF packets, you can't just mark EF packets out of profile and send them on).

1.5 Paper contributions

The authors of [CF98] provide two notable contributions. The first being the movement of packet classification to the edge routers using DSCP fields in the IP header. The second being a set of policies on how to deal with excess traffic.

2 Integrated Services (Int Serv, [CSZ92])

In contrast to Diff Serv, Int Serv takes an end-to-end approach to the task of providing a QoS beyond Best Effort to application flows through absolute or statistical guarantees. Note that most of the complexity of Int Serv results from handling multicast. Additional difficulties arise attempting to perform effective admission control in the face of "controlled load".

2.1 End-to-End QoS classes

There are three end-to-end QoS classes:

- Best Effort Service

  As the default class of service provided by the internet, this service makes no guarantees for delivery rate, capacity or even successful delivery of data transmitted through the network.

- Guaranteed Service

  ([RFC 2212] provides the in-depth description of this service)

  This class of service provides firm (mathematically provable) bounds on end-to-end datagram queueing delays. This class of service makes it possible to provide a service that guarantees both delay and bandwidth. [RFC 2212]

- Controlled Load Service

  ([RFC 2211] provides the in-depth description of this service)

  This class of service provides the client data flow with a quality of service closely approximating the QoS that same flow would receive from an unloaded network element, but uses capacity (admission) control to assure that this service is received even when the network element is overloaded. [RFC 2211]

2.2 End-to-End QoS Mechanisms

We have covered aspects of the internet's existing end-to-end Best Effort service in previous lectures. Here we describe how Guaranteed and Controlled Load services achieve their provided service levels.

These last two services share three aspects of their design. They contain a signaling protocol used to request service, an admission control scheme to help ensure proper service, and scheduling algorithms in routers which provide the basis for the service.

- Signaling protocol

  A signaling protocol is used to request one of the last two classes of service. The signaling mechanism provides the traffic characteristics (or specification) and a reservation for the resulting traffic specification provided.

The Resource ReSerVation Protocol (RSVP, [RFC 2205]) is a signalling protocol which takes a traffic specification (TSpec) as input, and returns a PATH message confirming the resource reservation as output. Receivers initiate the reservation request which allows the protocol to accommodate both unicast and multicast traffic flows.

A TSpec consists of the following information:

- Maximum Transmission Rate
- Average Transmission Rate
- Maximum Burst
- Minimum Packet Size
- Maximum Packet Size

As a TSpec flows through the network, each router provisions resources based on a flow's TSpec, adjusting the TSpec based on what it decides it can handle before forwarding it on to the next router.

Once the TSpec arrives at the sender, a PATH message is returned to the receiver on the reverse path confirming the reservation. For Guaranteed Service, the PATH message also contains an RSpec, which indicates the level of service reserved for this flow.

Since each router along the path implicitly keeps a record of a flow's reservation, to avoid keeping track of too much state, routers throw out this information after a certain period of time. This means the state of reservation has to be periodically refreshed.

There are two aspects to signaling which complicate the design, support for multicast and route changes (since neither service appears to pin routes down).

Below is a summary of the characteristics of the RSVP signaling protocol taken directly from [RFC 2205]:

- Makes resource reservations for both unicast and many-to- many multicast applications, adapting dynamically to changing group membership as well as to changing routes.
- Is simplex, i.e., it makes reservations for unidirectional data flows.
- Is receiver-oriented, i.e., the receiver of a data flow initiates and maintains the resource reservation used for that flow.
- Maintains "soft" state in routers and hosts, providing graceful support for dynamic membership changes and automatic adaptation to routing changes.
- Is not a routing protocol but depends upon present and future routing protocols.

- Transports and maintains traffic control and policy control parameters that are opaque to RSVP.
- Provides several reservation models or "styles" (defined below) to fit a variety of applications.
- Provides transparent operation through routers that do not support it.
- Supports both IPv4 and IPv6.

The use of RSVP under Guaranteed and Controlled Load services is specified in [RFC 2210, 2211 and 2212].

- Admission control

  In order to provide a given service level it is necessary to only admit flows up to the point where the service could no longer be provided, e.g. the telephone system. Guaranteed and Controlled Load services use admission control in slightly different ways.

- Scheduling algorithms

  Guaranteed and Controlled Load services require scheduling algorithms in network routers in order to implement their QoS. Variants on Weighted Fair Queueing are used with Guaranteed Service and FIFO variants are used under Controlled Load.

2.3 Guaranteed Service

Guaranteed Service provides an assured level of bandwidth to a conforming flow, with a bounded delay and no queueing losses. This is accomplished by combining parameters from individual network elements (routers, subnets, etc.) to produce the guarantee of service.

A conforming flow adheres to a token bucket behavior. Given this description of a flow, a network element computes various parameters describing how the element will handle the flow's data. By combining these individual parameters, it is possible to compute the maximum delay a piece of data will experience on the flow's path (assuming no network element failures or routing changes during the life of the flow). [RFC 2212]

A flow can be described as a token bucket flow with parameters (r, b), where r is the average rate and b is the maximum burst size, if for any time interval, T, the number of bytes sent during T is $\leq (r * T + b)$.

2.3.1 Signaling

The RSVP protocol is modified to add an RSpec along with the TSpec when a receiver initiates its request. A RSpec consists of simply a rate term, R, and a slack term, S, where the slack term is the difference between the requested rate and the reserved rate.

The returned PATH message then contains a modified RSpec describing the exact level of service provided to the requesting flow.

2.3.2 Admission Control

Admission control is straightforward for this service. When a new flow requests service, it is only added if the flow does not cause the combined set of flows in service to exceed the network's capacity.

$$\sum_i alloc + req \leq \text{available G.S. bandwidth}$$

It should be noted that under bursty traffic, the network might end up underutilized. This is a potential drawback to this service.

2.3.3 Scheduling

The routers participating in a guaranteed service offering use a variant on Weighted Fair Queueing for scheduling flows.

Since the flow's traffic conforms to a token bucket behavior, then if WQ: $R_i >= r_i$, there is a maximum bound on end-to-end delay:

$$\text{Max queue } delay < \frac{b}{r_i}$$

2.3.4 Summary

In summary, there are two costs to guaranteed service:

- under utilization of the network resulting from accommodation of bursty traffic
- high jitter resulting from WFQ spreading out bursts.

2.4 Controlled Load

This service class tries to remove the costs associated with Guaranteed Service. It relaxes the admission control policy to address network under-utilization issues, and uses FIFO queueing in the routers to address high jitter.

The end-to-end behavior provided to an application by a series of network elements providing controlled-load service tightly approximates the behavior visible to applications receiving best-effort service *under unloaded conditions* from the same series of network elements. [RFC 2211]

2.4.1 Signaling

Uses the receiver initiated RSVP protocol, but does not rely on RSpec like data as in the Guaranteed Service. Clients provide a TSpec estimating the data traffic they generate, just like they would for Guaranteed Service. A PATH message is returned, but if the clients traffic should exceed its previously specified TSpec, the client's flow may encounter delayed or lost packets.

2.4.2 Admission control

Unlike Guaranteed Service, where admission control is based on whether or not a flow, considering its TSpec, can be strictly accommodated by the network, here network elements are allowed to admit flows with some level of overcommitment. A network element could admit a flow even though the sum of all existing flows' requested TSpecs is greater than the available network capacity, if it notices that flows in general are falling short of their TSpec.

So a network element use the following algorithm to determine if a flow is admitted:

$$\sum_i TSpec_{avg} < \text{avail C.L. bandwidth}$$

The network element sums all known flows' average TSpec, which is based on individual observation, instead of taking it from the RSVP setup message. This would allow network elements to accommodate fluctuations in flow traffic rates while maximizing the number of allowed flows.

2.4.3 Scheduling

In order to alleviate the effects on bursty traffic by WFQ scheduling, routers use FIFO scheduling variants. A FIFO will receive a burst and transmit a burst without spreading out the packets as WFQ thus reducing jitter.