18: Enhanced Quality of Service

Mark Handley

Traditional best-effort queuing behaviour in routers

- Data transfer:
  - datagrams: individual packets
  - no recognition of **flows**
  - connectionless: no signalling
- Forwarding:
  - based on per-datagram, forwarding table look-ups
  - no examination of “type” of traffic – no **priority** traffic

Isn't Best Effort Service Sufficient?

In theory, yes.

- If there's sufficient capacity for all the real-time flows (as in the phone network) then best effort is sufficient.
  - Queues do not build
  - No packet loss occurs
- If there's insufficient capacity, calls will either block if we have reservations or give degraded service if we don't.
  - Neither of these is acceptable.
  - Thus there must be sufficient capacity.

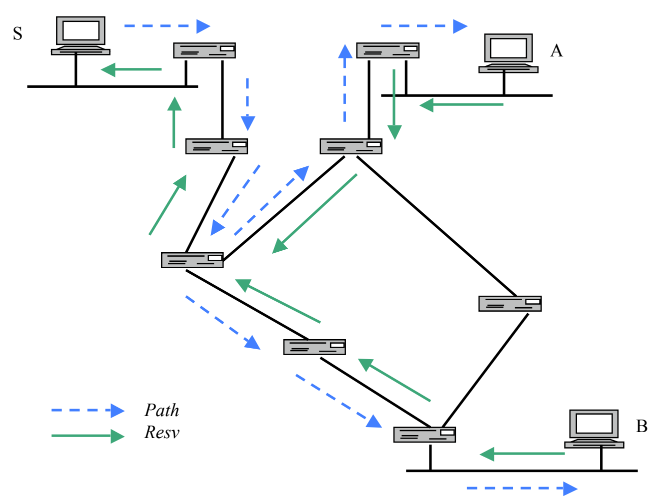
Isn't Best Effort Service Sufficient?

In practice, probably not:

- When demand grows exponentially, ISPs trail the demand curve at least some of the time.
- TCP traffic expands to fill available bandwidth and produces loss in doing so.
- Simple prioritization of real-time traffic leads to falsely described traffic.
- Getting from here to there is difficult - someone has to pay for the infrastructure.

RSVP and Intserv

RSVP: Details



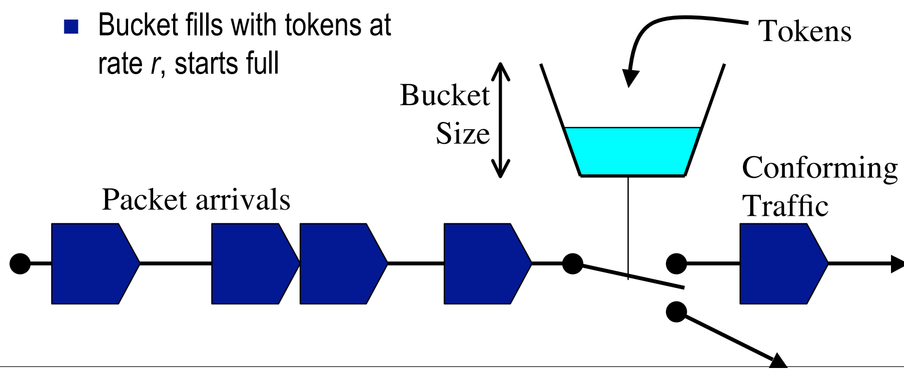
RSVP: Reservation Styles

Several styles of reservation are supported:

- Fixed Filter
  - separate reservations for each listed sender.
  - E.g.: several video streams.
- Shared Explicit
  - one reservation shared between several listed senders.
  - E.g.: video with floor control
- Wildcard
  - one reservation for any senders.
  - E.g.: audio with silence detection in a large group

Token bucket

- Three parameters:
  - *b*: bucket size [B]
  - *r*: bucket rate [B/s]
  - *p*: peak rate [B/s]
- Bucket fills with tokens at rate *r*, starts full
- Tokens allow transmission
- Burst allowed at rate *p:*
  - data sent < *rt* + *b*



Intserv: Integrated Services

Two Intserv service models were standardized:

- Controlled Load Service
  - This is the one you want.
  - If you want Intserv at all.
- Guaranteed Service
  - Practically no-one needs this.

Controlled Load Service

- The goal is to make it look like the network is unloaded.
  - It does not guarantee jitter bounds or no loss
  - both are very low though.
- Traffic is policed at the network edges and split/merge points.
  - If it exceeds the reservation, it is treated as best effort.
  - A token-bucket is used for policing and specified in reservation requests.
  - Admission control ensures that reservations do not exceed the available bandwidth.
- Controlled Load packets get priority over Best Effort
  - Best Effort packets are not pre-empted, so some jitter is seen.
  - Cumulative jitter can lead to small, temporary queues.

Why isn't everyone doing it?

The protocols and mechanisms work OK.
It solves the problem people *thought* they wanted solved.

Some minor issues:

- Extra traffic due to soft-state refreshes
- Route changes & router failure:
  - QoS degrades to best-effort, need to re-negotiate QoS

Two Serious Problems:

- Charging/authentication
- Router State

RSVP/Intserv Charging

- A reservation goes hop-by-hop across many ISPs.
  - Why should I reserve bandwidth for some receiver I've never heard of?
- Need negative feedback to discourage reservations or everything gets reserved.
  - Essentially this means charging.
  - Vanilla RSVP needs n^2 billing arrangements between n ISPs.

Router State

- Backbone routers currently handle O(1,000,000) simultaneous connections.
- We don't want a significant proportion of these to have reservation state:
  - Fast router memory is very expensive.
  - CPU Cycles to check the flow spec are in very short supply.

Router State: Solutions?

- Only police/install state at the edges.
  - Most of the congestion is at the edges.
  - Do something different (or nothing at all) in the backbone.

Diffserv

Differentiated Services

There are two ways to get different service for your packets:

1. Install filter state in routers.
2. Use the filter to recognize compliant packets.
3. Give them different service.

1. Set bits in the packets.
2. Use the bits to recognize compliant packets.
3. Give them different service.

Intserv does the former, Diffserv does the latter.

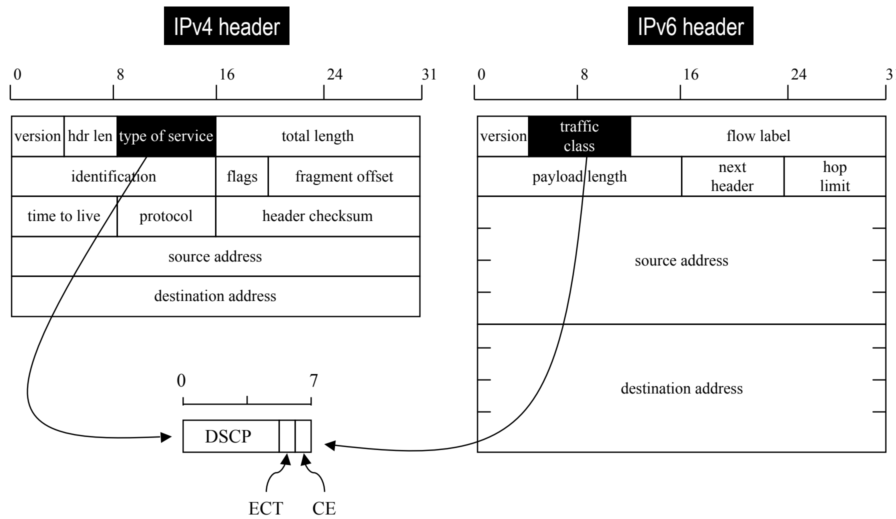
Traffic Limitations

- Can't give all traffic better service!
  - Must limit the amount of traffic that gets better service
- **Intserv:** On demand request from end-system, travels hop-by-hop.
  - Can be refused if insufficient capacity available.
  - Difficult to bill.
- **Diffserv:** Service Level Agreements (SLA)
  - much coarser grain.
  - source agrees to limit amount of traffic in given class.
  - network agrees to give that traffic "better" service.
  - network bills more than they'd charge for best-effort connectivity.

Diffserv Bits

- There are not many bits in an packet we can use.
  - 8 TOS bits, but 2 of those allocated to ECN
- If this is to go fast, the bits must specify the behaviour that the router should apply to the packet.
  - Thus there are not many behaviours we can specify.
  - Actually there aren't that many we want to specify either.
  - Allocating the bits as codepoints makes better use.

Diffserv codepoints



Services vs Hop-by-hop behaviours.

An end-to-end service is comprised of three parts:

- Admission control
- Policers that set or clear diffserv codepoints.
- Routers that use these diffserv codepoints to give different service.

A small number of diffserv codepoints (per-hop behaviours) can provide a large number of end-to-end services depending on the admission control and policing.

- In practice only two defined.

“Expedited Forwarding” - RFC2598

- *Virtual leased line* service
  - Marked packets get minimal delay and very low loss
  - e.g., put EF packets in high priority queue
  - Data rate specified in SLS.
  - Traffic exceeding the SLS is dropped.

- To make this a true “absolute” service, all SLAs must sum to less than the link speed.
  - More likely, a way to assure relatively low delay

“Assured Forwarding” - RFC2597

- Some packets are marked as low-drop probability and others as high-drop probability.
  - Packets are all serviced in order - this makes TCP implementations perform well.
  - Traffic exceeding the SLS is re-marked (i.e., it loses its assurance)

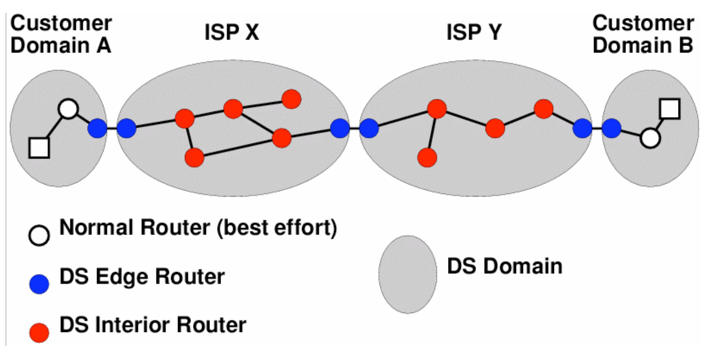
- Can be implemented using variations of RED
  - different drop probabilities for different classes

Assured Forwarding Example

- Suppose we have a congested link with 10% premium traffic and 90% best-effort. traffic.
  - The overall drop rate is 5%
  - We can give the premium traffic *no loss* if we increase the loss rate for the best-effort traffic to 5.56% (or 5.06% if it's TCP)

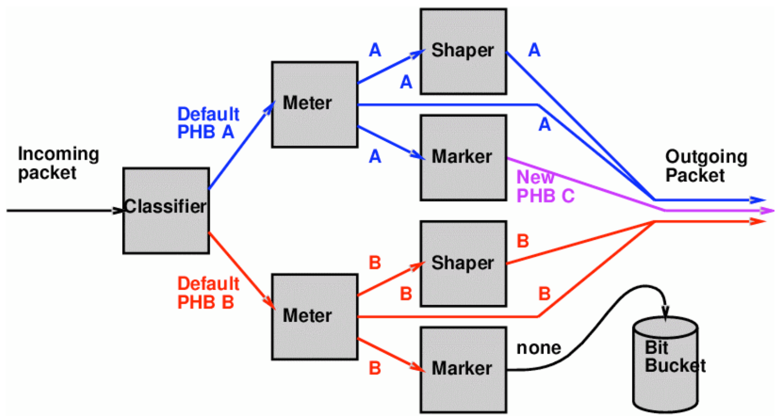
- Can get a large improvement in service for the small class of traffic without imposing much of a penalty on the other traffic.
  - This depends on the SLAs to control the premium traffic, as this is no longer getting a congestion control signal.

SLAs and TCAs



- *Service Level Agreements* exist between DS domains
  - These specify *Traffic Conditioning Agreements* - how the edge routers should condition the traffic
- Interior routers forward purely based on the *per-hop-behaviours* specified by diffserv codepoints in the TOS bits.

Diffserv Edge Routers



Diffserv Summary

Advantages:

- Very simple to implement
  - Minimal router state.
- Can be applied to different granularities
  - flows
  - institutions
  - traffic types
- Realistic economic model
  - Bilateral SLAs

Disadvantages:

- Expedited Forwarding has low efficiency
  - Must be small fraction of traffic.
- Assured Forwarding is just better best effort
  - Not low delay.
  - No guarantees
- Bandwidth broker for dynamic SLAs is still fictional

Comparison

	Intserv	Diffserv
Signalling	from application	network management, application
Granularity	flow	flow, source, site (aggregate flows)
Classification	destination address, protocol & port number	packet class (other mechanisms possible)
Scope	end-to-end	between networks, end-to-end

Note: They are not necessarily mutually exclusive -
eg Intserv reservation within a Diffserv flow

Summary

- Probably do need QoS mechanisms for IP, though not universally.
- Per flow:
  - INTSERV/RSVP
  - does not scale well, hard to provision, hard to bill
- Customer/provider services:
  - DIFFSERV
  - still maturing
  - sane economics, but few customers.
- Reality: not much QoS deployed.