Implementation of QoS mechanisms in MPLS networks

Avneesh Sachdev
(avneesh@procket.com)
Agenda

• QoS background
• MPLS value addition to QoS
• Interaction of MPLS QoS with routing
• Procket QoS architecture and implementation
• Summary
QoS background

- Service provided by traditional network varies under congestion or during convergence
  - Variable delay, jitter, loss, available bandwidth

- Traffic has diverse service needs
  - Voice requires low delay, jitter, loss

- One approach: Avoid congestion by over provisioning
  - Must guarantee no congestion at all times
  - May be wasteful to run network at low utilization

- Alternative: Make network aware of service needs of traffic
  - Most effective when there is congestion or you create congestion
  - Some packets have to lose for others to win
QoS background: Granularity

- **Fine grained**
  - *QoS provided to individual applications or flows (Intserv approach)*

- **Coarse grained**
  - *Network aware of large classes of data or aggregated traffic (Diffserv approach)*

- **Coarse grained approach scales better in the core**
MPLS Value Addition to QoS
MPLS and QoS: Connection oriented

• Resource allocation
  • Route flows (LSPs) through network with admission control
  • Police and/or monitor traffic at LSP ingress
  • Bandwidth constraints may be per link (TE) or different for different classes of traffic on a link (DS-TE)
  • For best results, apply admission control to all traffic that uses resources in a pool
    • Engineer all traffic
    • Or, use distinct PHB Scheduling Classes (queues) for engineered and other traffic
  • Controlled load => less probability of congestion => "better" service
MPLS and QoS: Connection oriented

- Easy identification of flow along LSP path (P2P LSPs)
  - Enables flexible definition of flow (by ingress)
  - Allows for more classes -- a flow can be mapped to a traffic class at signaling time (as in L-LSPs)
  - Leaves open possibility of QoS guarantees on a finer grained level, as opposed to an aggregate basis

- Interoperation with ATM / Frame Relay
MPLS and QoS: Source routing

• Fast Reroute
  • Minimizes drops in the network due to slow IP convergence
  • Enables high availability of service
  • For traffic that requires low loss

• Planned downtime
  • Traffic can be seamlessly rerouted around an LSR box before bringing it out of operation
  • "Costing out" a link in IGP may not be good enough
MPLS and QoS: Low overhead tunneling

- Diffserv transparency
  - Can use Pipe tunneling model to make transit Diffserv domain transparent to traffic

- Routing table size on transit routers
  - Allows routing decisions to be made at LSP ingress
  - QoS-aware routing may require larger routing tables
Interaction of MPLS QoS with routing
Issue: QoS-aware routing

• Different paths in the network may be optimal for different services

• Control plane decision on path is easy when a predetermined service is required for some traffic
  • Say, an L2 circuit, or all IP traffic matching a given prefix

• Maintaining a routing table per class is expensive for control and forwarding plane
• Tunnel 1 is set up by R1 for carrying real time traffic
• Control plane on R1 treats traffic destined to PBX as real time
  10.1.1/24 -> Tunnel 1
• QoS applied by R1 based on matched prefix, not packet marking
Intra-domain IP routing

• Possible extension to IGP shortcuts
  • Use single topology IGP as today with TE / DS-TE extensions
  • Run single SPF, modify IGP shortcuts to compute distinct tunnels as next hop(s) for desired classes
  • Don't need routing table per service or modification to longest match lookup
  • Next hop information more complex, requires sharing across prefixes for efficiency
  • Doesn't require upgrading boxes other than to run TE / DS-TE
**IGP shortcuts**

• R1 signals Tunnel 1 for class C1 and Tunnel 2 for class C2.
• Paths to 10/8 on R1 are as follows

<table>
<thead>
<tr>
<th>Prefix</th>
<th>C1 next hop</th>
<th>C2 next hop</th>
<th>Default next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/8</td>
<td>Tunnel 1</td>
<td>Tunnel 2</td>
<td>R2</td>
</tr>
</tbody>
</table>
Inter-domain routing considerations

- Routing per service in BGP would be expensive
- Existing BGP infrastructure should be okay if desired services are supported by selected paths
- Desirable for SPs to agree on standard service definitions for easy translation at border routers
- Domains that do not support QoS will increasingly not be used for transit
- Inter-AS TE could be used on a tactical basis to source route traffic
• AS 300 has QoS enabled, AS 200 does not
• BGP policies avoid paths across AS 200
Procket QoS architecture and implementation
Procket Logical Architecture

- IM: Interface Multiplexer
- iAPP: Input Adaptive Packet Processor
- oAPP: Output Adaptive Packet Processor
- TSE: Terabit Switch Engine
- MSU: Memory Switch Units
- SC: System Controller

Procket QoS
Procket Architecture

- Distributed processing on line cards
  - 40 Gbps processors
  - Input Adaptive Packet Processor (iAPP): Fully programmable forwarding, accounting
  - Output Adaptive Packet Processor (oAPP): Fully programmable encapsulation, queuing discipline, accounting

- Centralized shared memory switch fabric
  - 200 ms buffer
  - Up to 32 queues per physical interface. At least 4 queues per DS-3
  - 36K queues per chassis
  - Packets read in and out of memory once. No separate input/output buffers

- 480 Gbps full duplex line rate processing with 40 byte packets
- Fully non-blocking, any DS3 to any DS3
Procket QoS

Procket Architecture/Packet Flow: QoS

- Packet Classification
- Modify DSCP/EXP/Precedence
- Policing
- RED profile selection
- Queue selection

- RED, WRED
- Report queue fullness to TSE (RED)
- Queue lengths

- Programmable queuing discipline
- PQ, DWRR
- PQ(DWRR)
- DWRR(PQ)
- Rate shaping

Media Adapters

Line Cards

Switch Cards

Line Cards

Media Adapters
**Procket QoS Implementation**

- Provides QoS in hardware without impact to system throughput
- UI designed so you can focus on delivering services and not on turning knobs.
  - *System intelligently maps service requirements to hardware*
- Classification on EXP, DSCP, Precedence
- Programmable RED, WRED
- Input policing, metering & marking
- Output marking
- Output rate shaping per queue
Procket QoS Implementation

- Queuing algorithms implemented in microcode
  - PQ and DWRR currently, set up in two levels of hierarchy
  - PQ for low latency, DWRR to distribute bandwidth

- E-LSPs with uniform tunneling model

- Extensive statistics per interface
  - Packet drops per class
  - Current and average queue depth
  - In-spec and out-of-spec packets per class
Example: Service requirements

- **Real time**
  - Small packets, delay and jitter sensitive. Low loss.
  - Interactive voice and video
  - Possibly Call signaling

- **Premium**
  - Low loss

- **Streaming media**
  - Less strict delay, jitter requirements

- **Best effort**
  - Everything else
Example: Configuration

```yaml
qos
  class premium
    exp internetwork-control
  class realtime
    exp critical
dscp CS5
  class streaming-media
    exp flash-override
dscp CS4
output-behavior low-latency
  meter rate 3000 burst 100
  depth 50
service-profile backbone
  class default
  class premium
  class realtime
    output-behavior low-latency
class streaming-media
queueing-discipline priority (realtime, dwrr (premium [45], default [20],
  streaming-media [35]))
```
Procket QoS Futures

- Classification enhancements
  - Based on policies
  - 802.1p
- More queuing disciplines
  - WFQ
  - MDRR?
- Interface rate shaping
- Remapping of traffic between services
- More MPLS support
  - DS-TE
  - L-LSPs
Summary

• QoS allows you to control treatment of packets by network
  • *Most useful when there is congestion*

• MPLS is a useful technology for QoS
  • *Enables better resource allocation, high availability, easier management, Diffserv transparency, smaller table sizes...*

• Incremental approach to intra-domain QoS-aware routing using TE and IGP shortcuts

• Common service definitions desirable for easier inter-domain operation

• Procket architecture and QoS implementation deliver easily manageable QoS without comprise on performance