Quality of Service and Multicasting in Computer Networks
Sonia Fahmy
Department of Computer Sciences
Purdue University
fahmy@cs.purdue.edu
http://www.cs.purdue.edu/homes/fahmy/
(slides 4-6 and 10-20 used with permission of Raj Jain)

Why QoS?
Traffic Management on the Information Superhighway

What is QoS?
- Predictable Quality: Throughput, Delay, Loss, Delay variation (jitter), Error rate
- Opposite of best effort (random quality)
- Mechanisms:
  - Capacity Planning: links, routers, switches
  - QoS based path determination, Route pinning
  - Admission and policy control, Signaling, Shaping, Policing (leaky and token buckets)
  - Classification, Queuing, Scheduling, Buffer management (drop thresholds and probabilities), Feedback and response to feedback

QoS Triangle
Low Capacity
High Traffic
- Senders want to send traffic any time with high load, high burstiness
- Receivers expect low delay and high throughput
- Since links are expensive, providers want to minimize the infrastructure
- If one of the three gives in ⇒ no problem

Integrated Services
- Best Effort Service
- Controlled-Load Service: Performance as good as in an unloaded datagram network. No quantitative assurances
- Guaranteed Service:
  - Firm bound on data throughput and delay.
  - Every element along the path must provide delay bound.
  - Is not always implementable. e.g., Shared Ethernet.
**Integrated Services**

- Routing messages
- RSVP messages
- Admission Control
- Policy Control
- Forwarding Table
- Per-flow QoS Table
- Classifier
- Scheduler

**RSVP**

- Resource ReSerVation Protocol
- Internet signaling protocol
- Carries resource reservation requests through the network including traffic specs, QoS specs, network resource availability
- Sets up reservations at each hop
- Per-flow \( \Rightarrow \) scalability in backbones?

![RSVP diagram](image)

**Differentiated Services**

<table>
<thead>
<tr>
<th>Ver</th>
<th>Hdr Len</th>
<th>Precedence</th>
<th>ToS</th>
<th>Unused</th>
<th>Tot Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>4b</td>
<td>4b</td>
<td>3b</td>
<td>4b</td>
<td>1b</td>
<td>16b</td>
</tr>
</tbody>
</table>

- Externally Observable Forwarding Behavior
- Example:
  - \( x\% \) of link bandwidth
  - Minimum \( x\% \) and fair share of excess bandwidth
  - Priority relative to other PHBs
- PHB Groups: Related PHBs, PHBs in the group share common constraints, e.g., loss priority, relative delay

**Per-Hop Behaviors**

- In \( \rightarrow \) PHB \( \rightarrow \) Out

**Expedited Forwarding**

- Also known as “Premium Service”
- Virtual leased line
- Guaranteed minimum service rate
- Policed (leaky bucket): Arrival rate < Minimum Service Rate
- Not affected by other data PHBs
  \( \Rightarrow \) Highest data priority (if priority queueing)
- Code point: 101 110

**Assured Forwarding**

- PHB Group
- Four Classes: No particular ordering
- Three drop preference per class
- Marking and selective buffer management (drop)
- 3 drop preferences for TCP and UDP scenarios
**Multiprotocol Label Switching**
- Label = Circuit number = VC Id
- Ingress router/host puts a label.
- Exit router strips it off.
- Switches switch packets based on labels.
  Do not need to look inside ⇒ Fast.

![MPLS Diagram](image1)

**Traffic Engineering Objectives**
- User’s Performance Optimization
  ⇒ Maximum throughput, Min delay, min loss, min delay variation
- Efficient resource allocation for the provider
  ⇒ Efficient Utilization of all links
  ⇒ Load Balancing on parallel paths
  ⇒ Minimize buffer utilization
- Current routing protocols (e.g., RIP and OSPF)
  find the shortest path (may be over-utilized).
- QoS Guarantee: Selecting paths that can meet QoS
- Enforce Service Level agreements
- Enforce policies: Constraint based routing ⇒ QoSR

![Traffic Engineering Diagram](image2)

**MPLS Mechanisms for TE**
- Signaling, Admission Control, Routing
- Explicit routing of LSPs
- Constrained based routing of LSPs
- Allows both Traffic constraints and Resource Constraints (Resource Attributes)
- Hierarchical division of the problem (Label Stacks)
- Traffic trunks allow aggregation and disaggregation
  (Shortest path routing allows only aggregation)

![MPLS Mechanisms Diagram](image3)

**Traffic Trunks**
- Trunk: Aggregation of flows of same class on same LSP
- Trunks are routable
  ⇒ LSP through which trunk passes can be changed
- Class ⇒ Queue, LSP ⇒ Next hop
  Class can be coded in Exp or Label field. Assume Exp.

![Traffic Trunks Diagram](image4)

**Flows, Trunks, LSPs, and Links**
- Label Switched Path (LSP):
  All packets with the same label
- Trunk: Same Label+Exp
- Flow: Same MPLS+IP+TCP headers

![Flows, Trunks, LSPs, and Links Diagram](image5)
Bandwidth Broker
- Repository of policy database. Includes authentication
- Users request bandwidth from BB
- BB sends authorizations to leaf/border routers
- Tells what to mark
- Ideally, need to account for bandwidth usage along the path
- BB allocates only boundary or bottleneck

End-to-end View
- ATM/PPP backbone, Switched LANs/PPP in Stub
- IntServ/RSVP, 802.1D, MPLS in Stub networks
- DiffServ, ATM, MPLS in the core

Why Multicast?
- Multipoint communication
  = exchange of information among multiple senders and multiple receivers (multicast group)
- Popular applications requiring multipoint support include:
  - conferencing, distance learning, software distribution, searching, server and database synchronization
- Complicated by variation in group size and dynamics and bandwidth requirements
- Using multicasting ⇒ traffic and management overhead not α number of participants

Multicast Requirements
- Address formats (e.g., class D) and allocation
- Group membership management (e.g., IGMP)
- Routing (e.g., DVMRP)
- Forwarding and replication
- Flow control
- Congestion control

Reliable Multicast Transport
- Consolidation necessary to prevent implosion. Solutions: probabilistic suppression, scoping, NAKs or bitmaps, router assist, scheduling, hierarchy
- Optimization of repair traffic. Solutions: local retransmitters, subcasting, FEC

Multicast Challenges
- Goals:
  - Fairness: TCP friendliness
  - High utilization of resources
  - Low loss and delay
  - Stability and fast transient response
  - Simplicity and generality (assume no layering)
- Window-based TCP-like multicast congestion control can under-estimate bandwidth share:
  - RTTmax wait before window update
  - Window set to minimum bandwidth-delay product
  ⇒ Rate-based preferred
  - How to detect the congested sub-tree and its parameters?
  - Drop-to-zero problem: how to filter loss indications?
Multicast Challenges (cont)

- TCP friendliness:
  - TCP throughput is a function of loss rate, RTT and timeout values
  - Where to estimate loss rate, RTT and timeout?
    - Receivers can compute rates and send them back to sender
    - Sender can estimate parameters more easily without extra traffic
  - Should work with ACKs, NAKs, bitmaps, ECN, etc.

Approaches

- Sender-based parameter estimation for RTT and loss. Additive increase when no NAKs received during an RTT-dependent interval. Multiplicative decrease when NAK or ECN signaling a new congestion epoch received.
- Select an ACKer (pgmcc)
- Equation-based (Floyd and Handley)

Layered Multicast and Coordination

- RLM: receivers join appropriate layers
- RLC: sync. points
- Sender can use feedback to determine number and rates of layers (UCI/Rutgers)
- Router uses differential drop (AT&T)
- Coordinator determines TCP/UDP for each sender (Balakrishnan’s CM (ecn IETF group), RACOON)

Approach (cont’d)

- Automatic split of heterogeneous groups depending on minimum and maximum rate thresholds
- Proactive FEC: increase FEC during high congestion
- Use-it-or-lose it for inactive senders and rate thresholds

Thank You!

Questions?