Towards an Integrated Services Internet

Scott Shenker
Palo Alto Research Center
Xerox Corporation

Proposals discussed in talk:

- Joint work with Braden, Clark, Deering, Estrin, Herzog, Jamin, Mitzel, Partridge, Wroclawski, Zappala, Zhang, ...

- Currently under consideration by two IETF Working Groups (INT-SERV and RSVP).

The Current Internet

Designed for *traditional* data applications:

- FTP, Telnet, Email, etc.
- Elastic (scalable) delay requirements

Service Model:

- *Best-Effort* service
- Single class of service

Implementing Mechanisms:

- FIFO packet scheduling
- *Stateless* (connectionless) architecture
- No admission control
- Congestion control algorithms

The Next Generation

New audio and video applications:

- Wide variety of service requirements
- Stricter delay requirements
- No congestion control

Cannot be supported by current architecture!

- Interference between “realtime” and “data” applications
- When everybody gets the same best-effort service, then best-effort isn’t good enough for everybody....
- Can’t efficiently meet strict delay requirements

Integrated Services Internet

Need *integrated services* architecture

- Offer different qualities of service (QoS)
- Allow applications to choose the appropriate QoS

Radical change in architecture

- Much more efficient use of bandwidth
- Only the delay-sensitive applications will get low delay service

Service Model

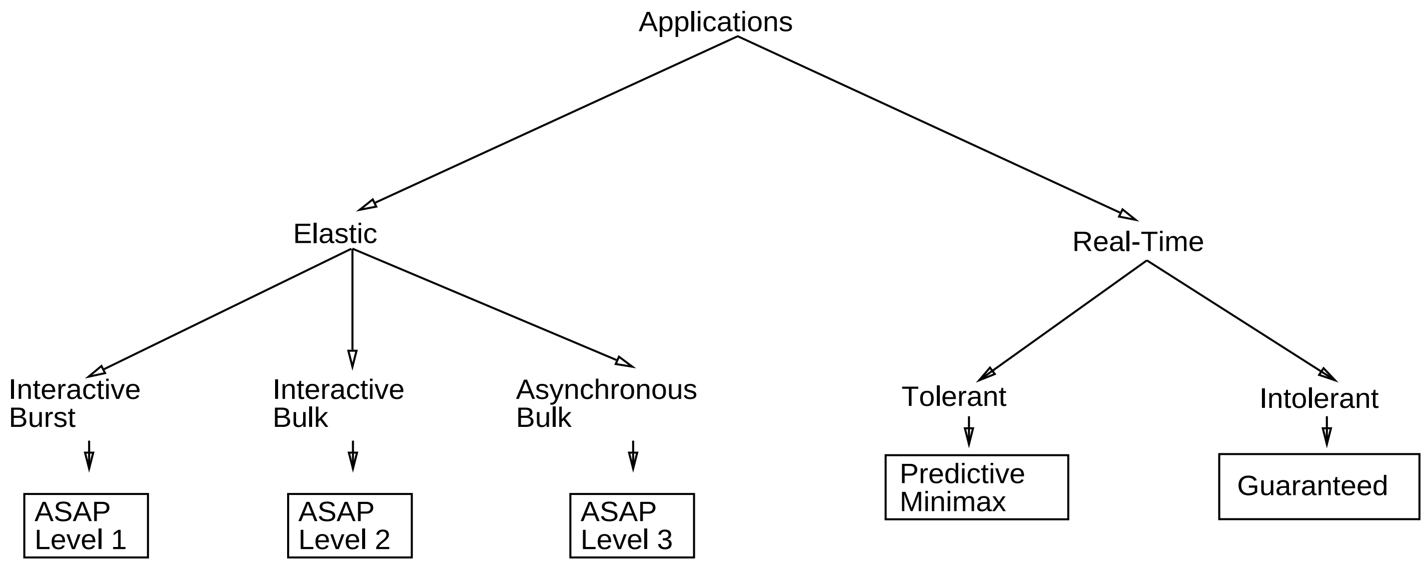
Fundamental Question:

- What qualities of service should the Internet offer?

This *Service Model* is the centerpiece of the Internet architecture:

- Will be embedded in applications
- Must be stable (extensible)
- Independent of implementing mechanisms
- Designed to meet application needs

Proposed Service Model



Elastic Applications:

- ASAP: *As soon as possible*

Realtime Applications:

- Intolerant: Guaranteed Service
  - *Worst-case* delay bound
- Tolerant: Predictive Service
  - Delay bound based on predictions
  - Minimax (qualitative)

Implementing Mechanisms

Packet Scheduling:

- Providing different classes of service requires non-FIFO packet scheduling
- Packet schedulers are the key enabler of integrated services

Resource Reservation Protocol: RSVP

- RSVP forwards service requests to switches
- Admission Control accepts or rejects request
- No longer a *stateless* architecture

Packet Scheduler

Guaranteed Service: Worst-case bounds

- Need to provide *isolation*
- Weighted Fair Queueing, ...
- Stop-and-Go Queueing

Predictive Service: Looser bounds

- Want to maximize *sharing*
- FIFO (with priority)
- Maximize statistical multiplexing

Challenge:

- Translating end-to-end services into per-switch requirements in heterogeneous networks

Resource Reservation Protocol: RSVP

Design Philosophy:

- Receivers initiate requests
- Multicast compatibility crucial
- Soft-state in switches
- Separation of reservation from filter
- Provide different *reservation styles*

Mechanistic Outline:

- Sources send periodic PATH messages to multicast group
- Receivers send periodic reservation requests up reverse path
- Aggregation of requests controlled by specified reservation style
  - Fixed Filter
  - Wildcard Filter
  - Dynamic Filter

Open Problems

Lots and lots and lots of details.....

- Various interfaces
- Switch scheduling requirements
- Integration of link-sharing naming hierarchy into architecture

ASAP service

- Fair Queueing or FIFO
- Form of congestion notification

Dropping:

- Preemptable packets
- Expendable packets
- Preemptable flows

Authentication

- Needed with nontrivial QoS

Guiding Principles

Maximize statistical multiplexing

- Predictive service

Multicast is fundamental

- RSVP uses receiver-initiated reservations

Soft-State not Hard-State

- RSVP uses periodic refreshes of state information

Separating service model from mechanism

- Don't specify scheduling algorithms
- Standardize service model