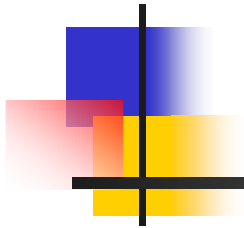


Quality-of-Service Support on the Internet



Dept. of Computer Science, University of Rochester



Quality of Service Support

Some Internet applications (i.e. multimedia) demand QoS guarantees in terms of delay, bandwidth, ...

Thus far: applying application-level end-to-end techniques to enhance the "best effort" service of the Internet.

- use UDP to avoid TCP congestion control (rate control) for rate-sensitive traffic
- compensate delay jitters: client-side buffering
- deal with network loss and delay loss
 - retransmissions if time permitting (for stored multimedia)
 - conceal errors: FEC, interleaving (for real-time interactive multimedia)



Improving QoS inside the Internet

- Future:** next generation Internet with QoS guarantees
- incorporating QoS techniques into Internet routers

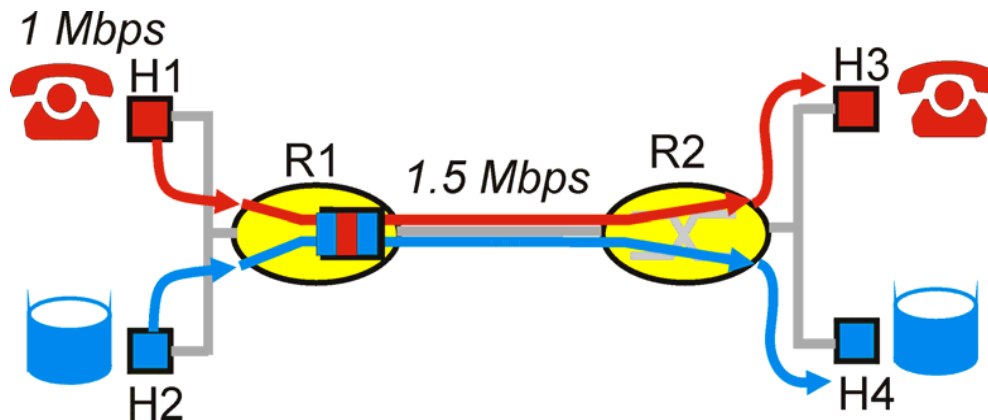
Principles

Mechanisms

Proposed architectures

Principle I for QoS Support: Packet Classification

- Example: 1Mbps IP phone, FTP share 1.5 Mbps link.
 - bursts of FTP can congest router, cause audio loss
 - may want to give priority to audio over FTP



Packet classification

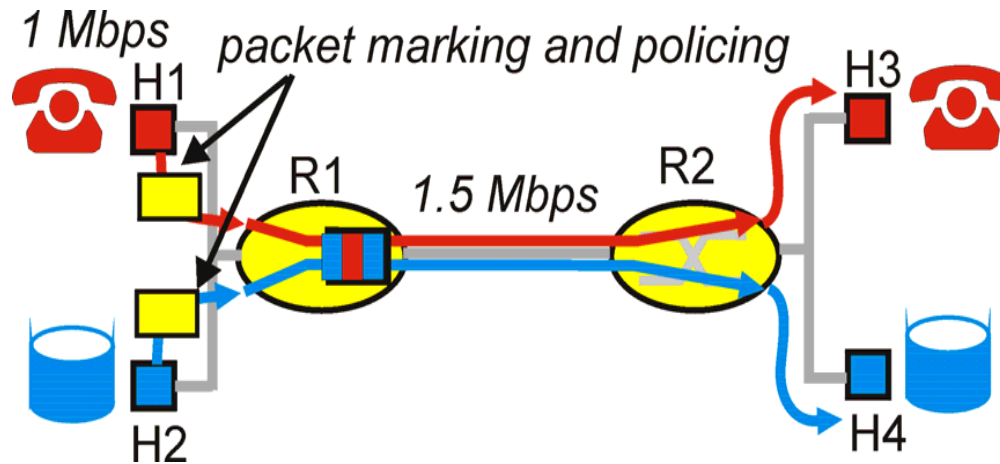
routers need to distinguish between different classes; and new router policy to treat packets accordingly: **packet marking (typically at the entrance)**

Principle II for QoS Support:

Isolation

What if applications misbehave (audio sends higher than declared rate)

- need to force source adherence to bandwidth allocations



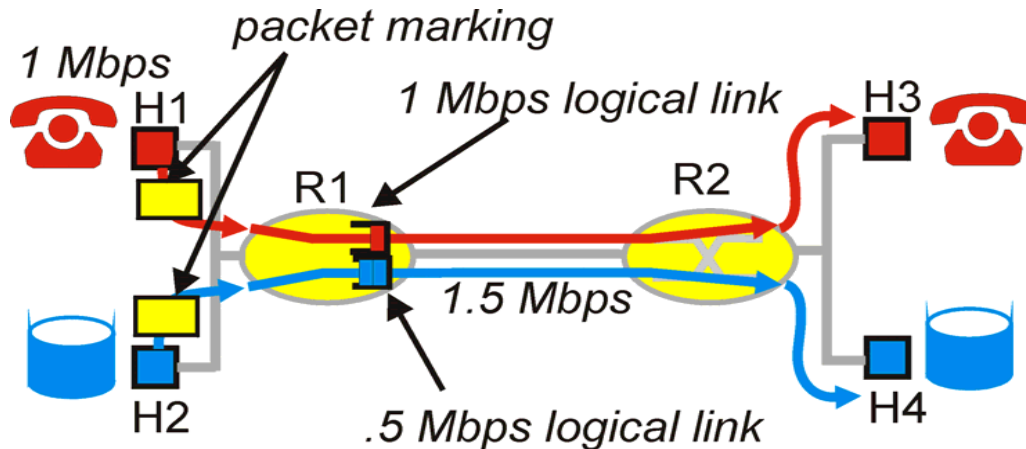
Isolation

provide protection for one class from abuses of others: **policing** (typically at the entrance)

Principle III for QoS Support: Efficiency

Efficient use of bandwidth when QoS is achieved:

- Not trivial: Allocating fixed bandwidth to flow - **inefficient** use of bandwidth if flow doesn't use its allocation

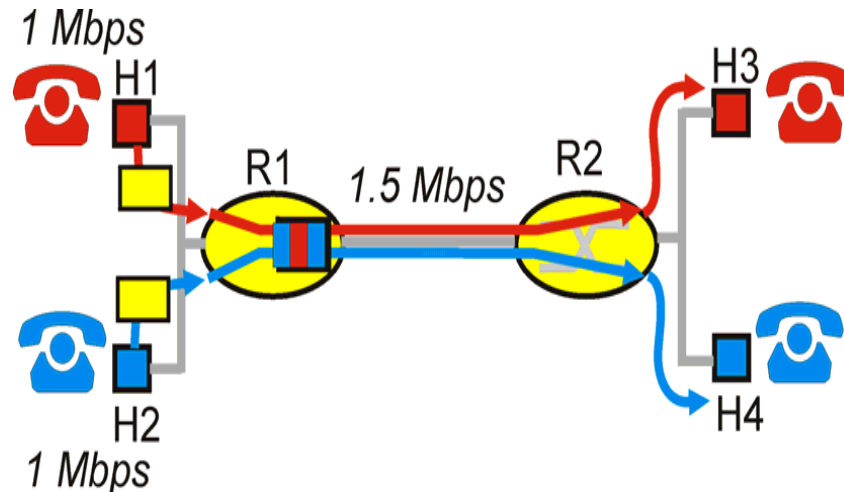


Efficiency

while providing isolation, it is desirable to use resources as efficiently as possible (work conserving): **scheduling (every router in the path)**

Principle IV for QoS Support: Go only where there is resource

Basic fact of life: can not support traffic demands beyond link capacity



Go only when there is enough resource

flow declares its needs, network may block flow (e.g., busy signal) if it cannot meet needs: reservation (what to reserve? where?)



Summary of QoS Principles

- Packet classification - packet marking
- Isolation - policing
- Efficiency (high resource utilization) - scheduling
- Go only when there is enough resource - reservation

Can be used together or individually, depending on the QoS objectives.

Let's next look at policing and scheduling mechanisms ...



Policing Mechanism

Goal: limit traffic to not exceed declared traffic pattern (at the entrance of the network).

Characteristics of traffic pattern (in terms of bandwidth usage):

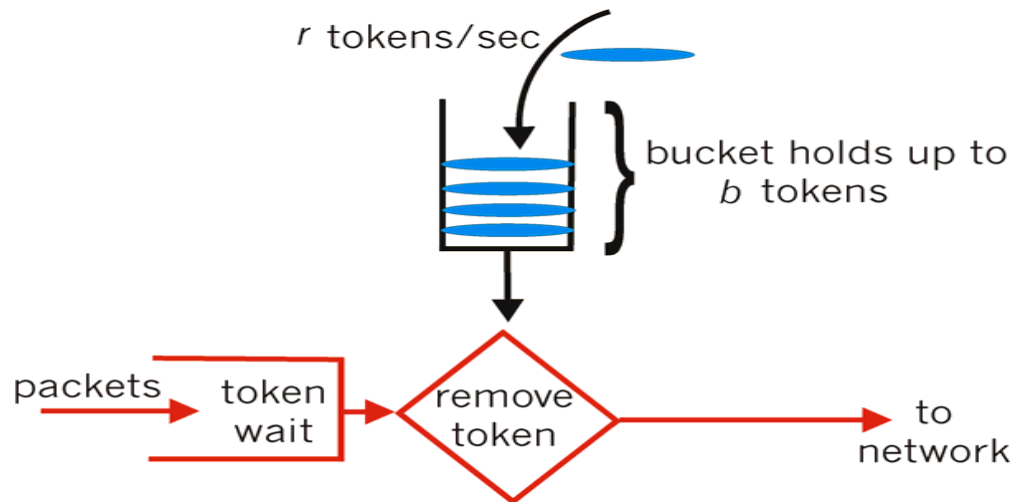
- **Average rate:** how many packets can be sent per unit of time
- **Peak rate:** the maximum or 95 percentile rate
 - what is the interval length: 100 packets per sec or 6000 packets per min?
- **Burst size:** maximum number of burst packets sent beyond the average rate over any time period
 - over any interval of length t : number of packets $\leq (r t + b)$

How to enforce a traffic pattern with an average rate and a burst size?

Policing Mechanism: Leaky Bucket

Leaky bucket

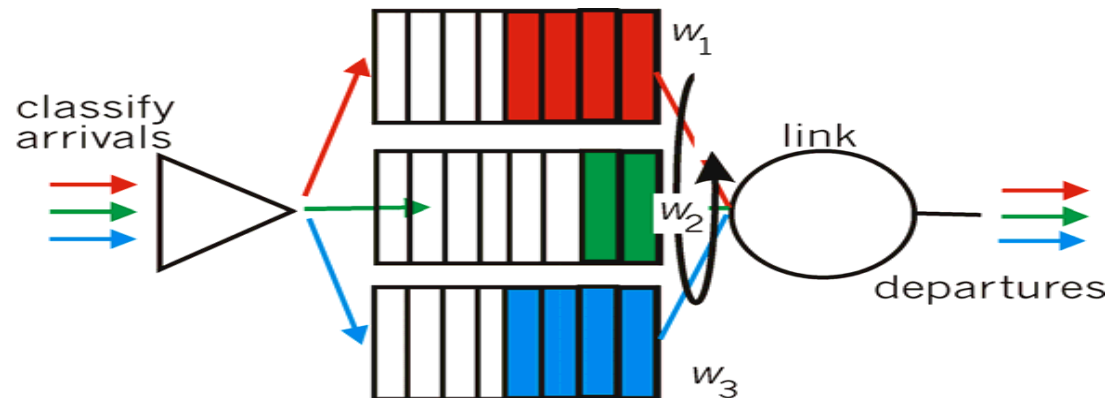
- bucket can hold b tokens
- tokens generated at rate r tokens/sec unless bucket full
- a token must accompany each admitted packet



- over any interval of length t : number of packets admitted less than or equal to $(r t + b)$
- limit input to specified average rate and burst size

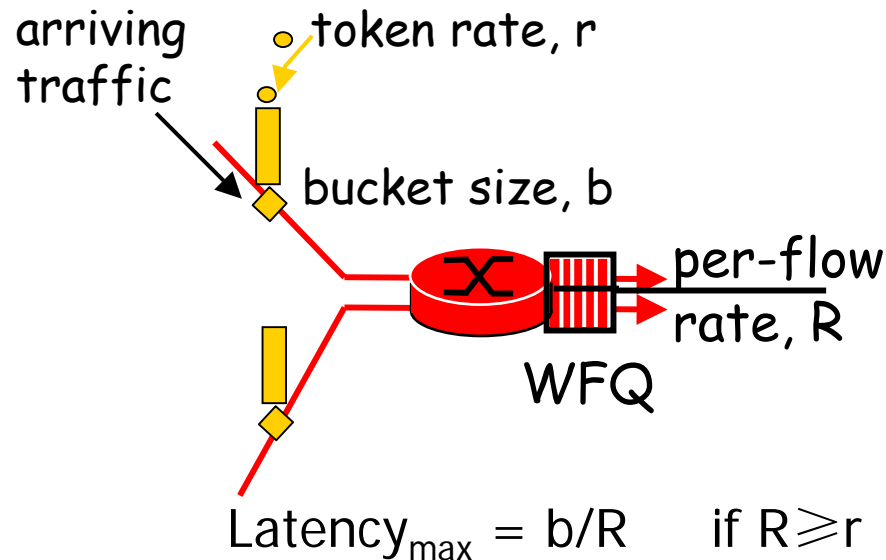
Packet Scheduling at Routers

- **FIFO (first in first out) scheduling:** send in order of arrival
- multiple classes
 - **Round robin scheduling:** cyclically scan class queues, serving one from each class (if available)
 - **Priority scheduling:** transmit packet in the highest priority queue
 - **Weighted fair queueing:** each class gets weighted amount of service in each round robin cycle



Combined Policing and Scheduling

Leaky bucket, WFQ combined to provide guaranteed upper bound on latency!



How should the Internet evolve to better support QoS for multimedia apps?

No major changes:

- apply only application-level end-to-end techniques

Integrated services (Intserv) philosophy:

- fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- requires new, complex software in hosts & routers

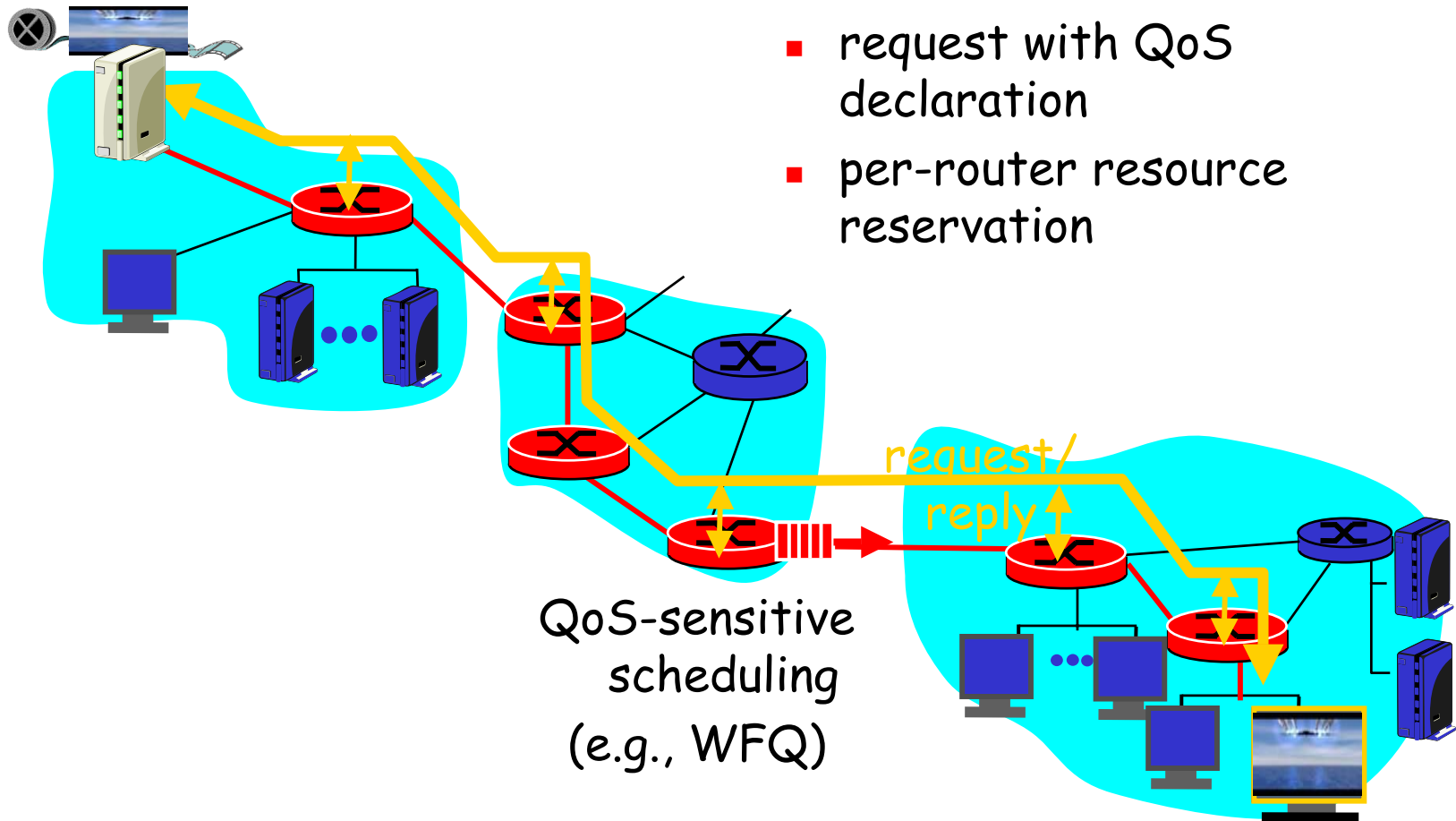
Differentiated services philosophy:

- Fewer changes to Internet infrastructure, yet provide differentiated services to different classes
- may not give firm guarantee on delay and bandwidth

Intserv: An Illustration

- **Resource reservation**

- request with QoS declaration
- per-router resource reservation





Differentiated Services

Concerns with Intserv:

- **Scalability & overhead:** signaling, maintaining per-flow router state difficult with large number of flows

Diffserv approach:

- simple functions in network core, relatively complex functions at edge routers (or hosts)
- provide differentiated services to different classes, may not give firm guarantee on delay and bandwidth

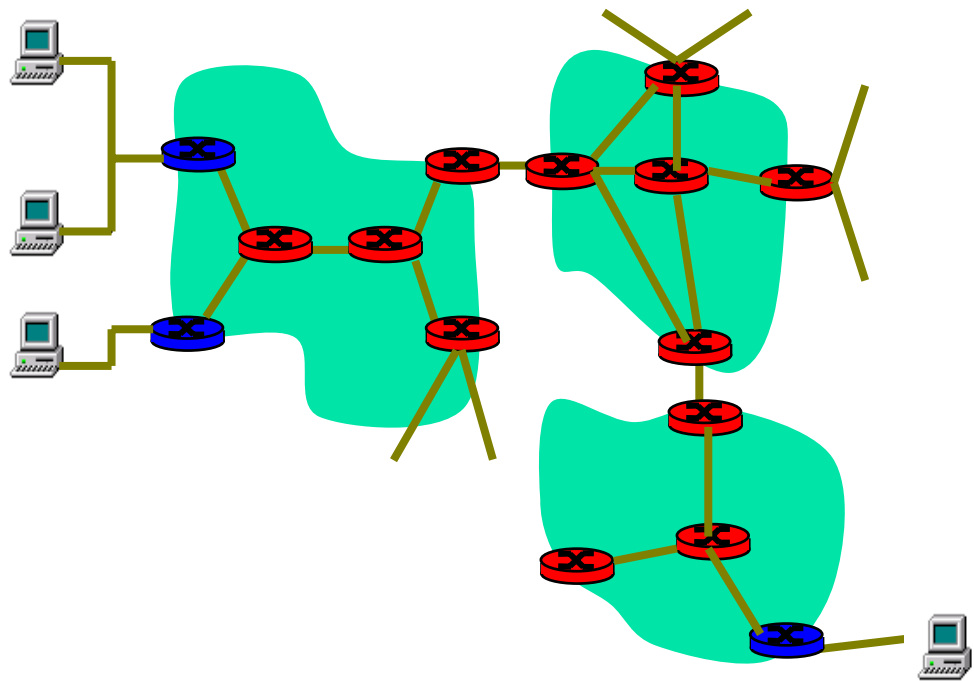
Diffserv Architecture

Edge router: 

- marks packets as **in-profile**
and **out-profile**

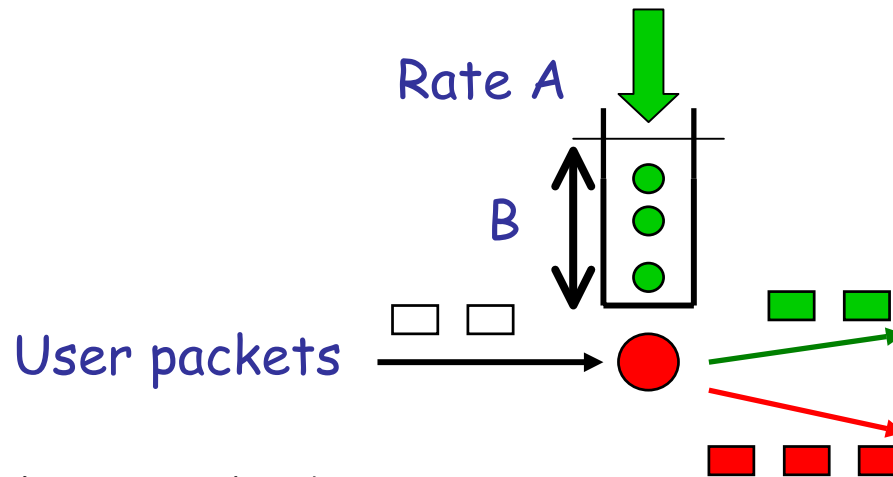
Core router: 

- buffering and scheduling
based on **marking** at edge
- preference given to **in-profile**
packets



Edge-router Packet Marking

- profile: pre-negotiated rate A , burst size B
- packet marking at edge based on traffic profile



Possible marking methods:

- conformance marking: conforming portion of flow marked differently than non-conforming one
- class-based marking: packets of different classes marked differently



Core-router Scheduling

provide differentiated services to different classes,
may not give firm guarantee on delay and bandwidth

- Class A packets get forwarded first before packets from class B
- Class A gets $x\%$ of outgoing link bandwidth over time intervals of a specified length



Multimedia Networking and Quality of Service Support: Summary

- multimedia applications and their QoS requirements
- applying application-level end-to-end techniques to enhance the "best effort" service of the Internet
- QoS support within the Internet: policing and scheduling, reservation mechanisms
- next generation Internet QoS architecture: Intserv, Diffserv



Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).