University of North Carolina at Chapel Hill

A Better-Than-Best Effort Forwarding Service For UDP

Lightweight Active Router Queue Management for Multimedia Networking

Kevin Jeffay
Mark Parris
Don Smith

http://www.cs.unc.edu/Reseach/dirt

Lightweight Active Queue Management for MM Networking

Summary

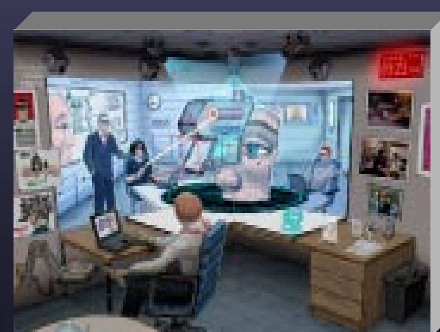
- The Internet is evolving to support quality-of-service
  - The mechanisms for realizing QoS are more about router queue management than about link scheduling
- There is a tension between providing QoS and supporting a multitude of transport protocols
- We are investigating a router queue management mechanism that attempts to balance these concerns

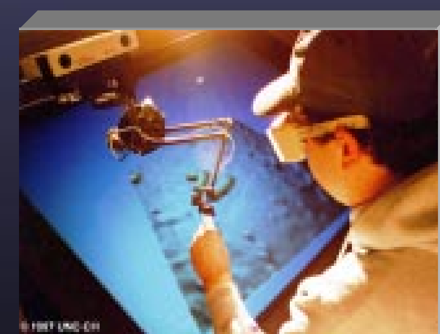
Lightweight Active Queue Management for MM Networking

Outline

- Performance of multimedia transmission on the Internet today
- Proposals for realizing quality of service
  - The integrated services architecture
  - The differentiated services architecture
- Active queue management
  - Random Early Detection (RED)
  - *Class-based thresholds* (CBT)
- Empirical evaluation

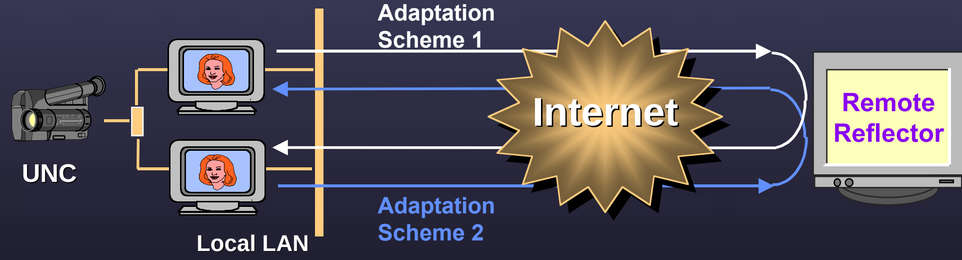
Research Context
Network support for immersive DVEs





- The Office of the Future
- The nanoManipulator system
- The principal characteristic:
  - Continuous media transmission
  - Low latency required for human-to-human communication, and the illusion of immersion

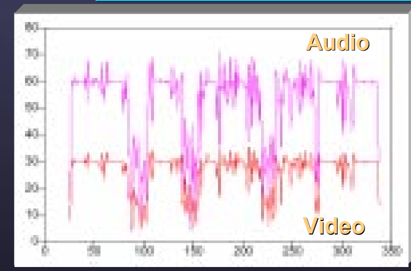
Performance of Multimedia Transmission on the Internet



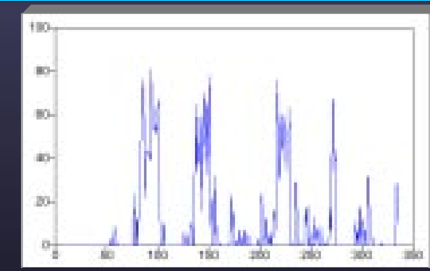
- What is multimedia performance like on the Internet today?
  - What are typical loss-rates?
  - What are typical latency (and jitter) values?
  - Would someone actually use an Internet videophone?

Performance of MM transmission

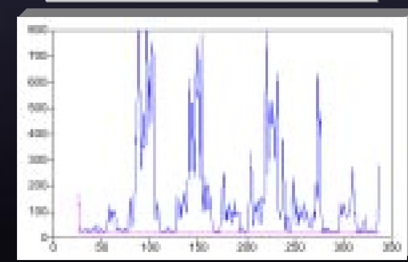
Performance of “raw” transmission



Throughput (frames/sec)



Packet Loss

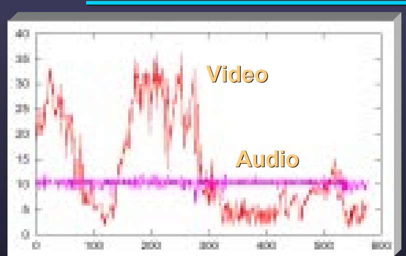


Audio Latency (ms)

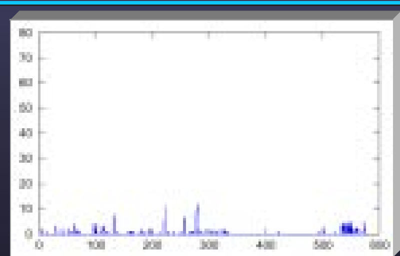
- “Out-of-the-box” ProShare performance
  - Frozen, motionless video
  - Clipped, broken audio

Performance of MM transmission

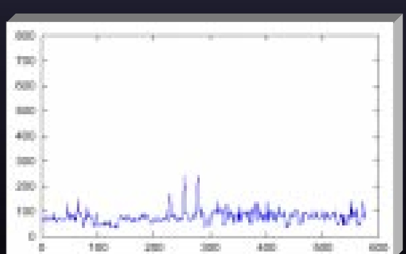
Performance of adaptive transmission



Throughput (frames/sec)



Packet Loss



Audio Latency (ms)

- End-system adaptation can ameliorate many of the effects of congestion
  - But can it do so reliably or predictably?

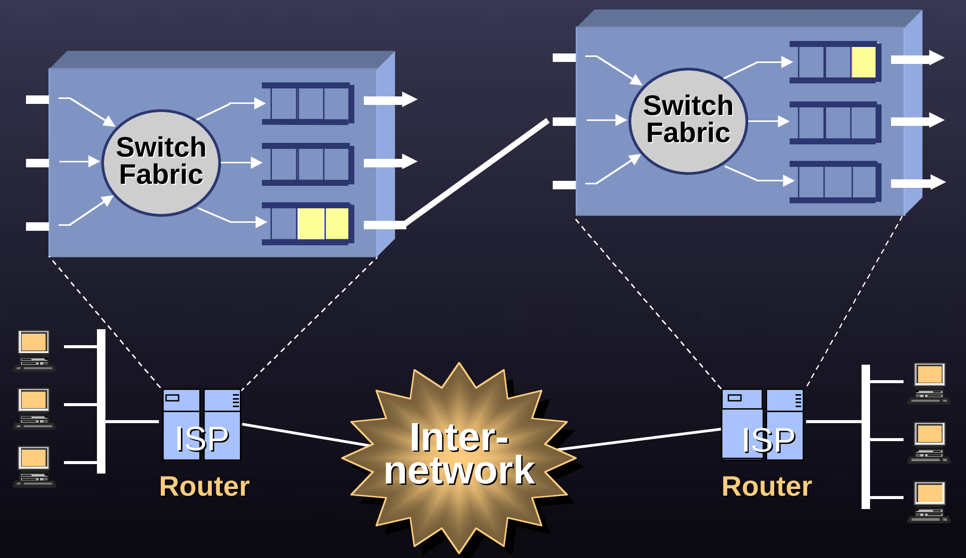
Performance of MM transmission

Performance summary

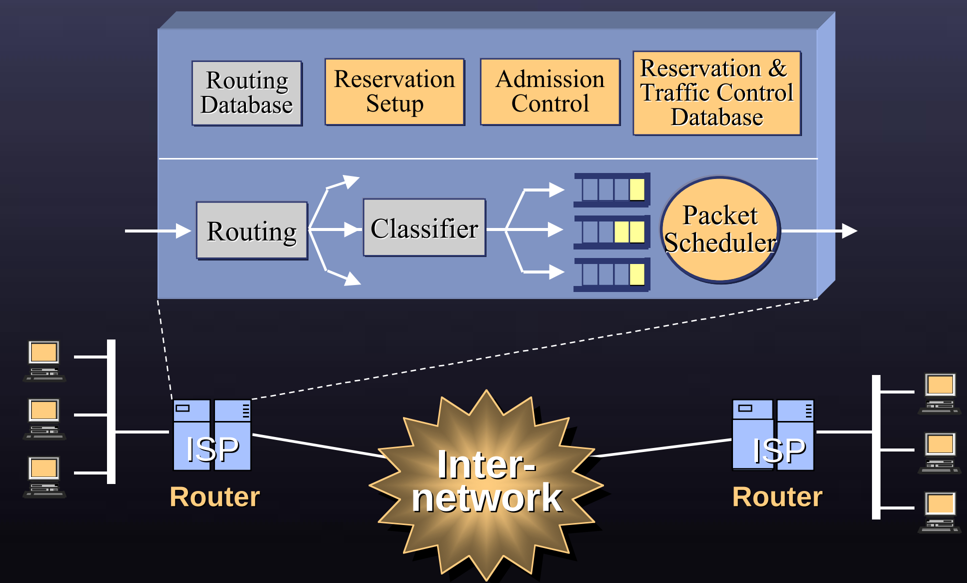
- Results of an Internet performance study from UNC to UVa
  - Repeated trials from 10 am to 7 PM weekdays
  - Scattered over three months

Time Slot	Sustainable	Not Sustainable
10:00-12:00	67%	33%
12:00-14:00	50%	50%
14:00-16:00	8%	92%
16:00-18:00	25%	75%
18:00-20:00	44%	56%
Percentage	39%	61%

The Nature of Congestion
Queueing delays in routers

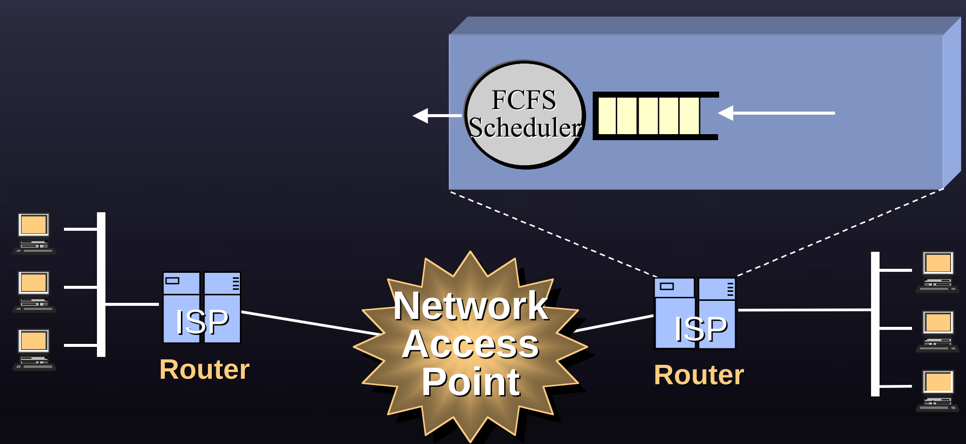


Towards QoS Networking
The Integrated Services Architecture



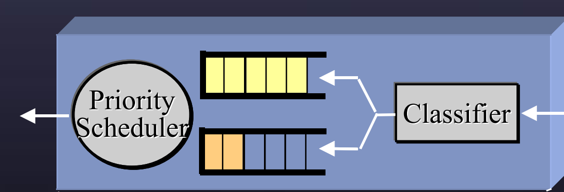
Towards QoS Networking
The *differentiated services* architecture

- ISPs allocate and sell capacity for a “premium” service



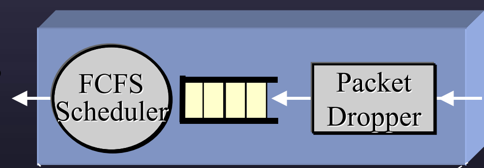
Towards QoS Networking
The *differentiated services* architecture

- ISPs allocate and sell capacity for a “premium” service
- Packets are marked according to “service profiles”

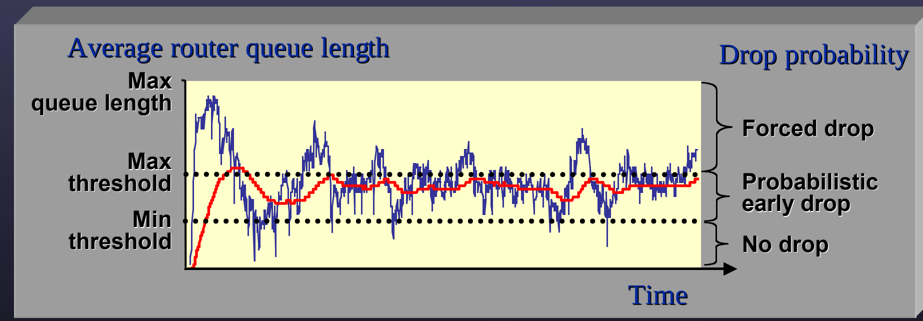


Realizing Differentiated Services
Active queue management

- This is significant utility in realizing differential services with a single router queue
  – In this model, a key technology for realizing differential services is a packet dropping policy



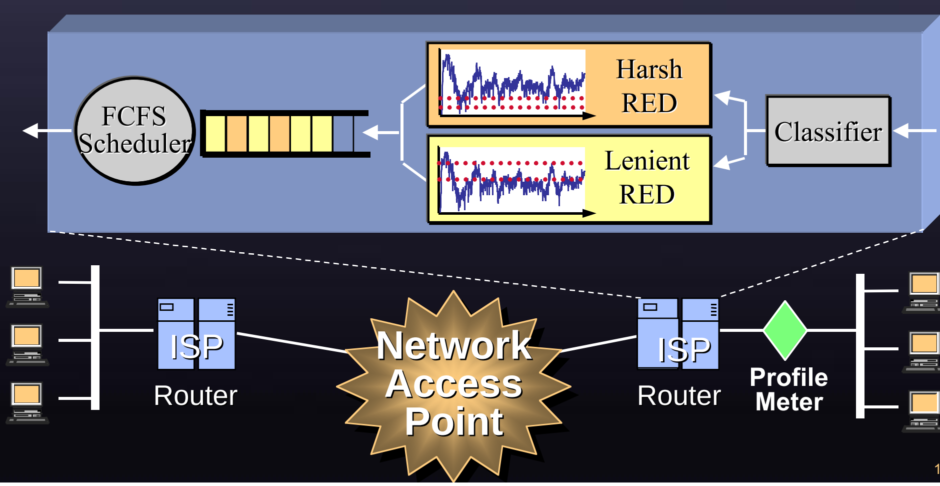
Realizing Differentiated Services
RED active queue management



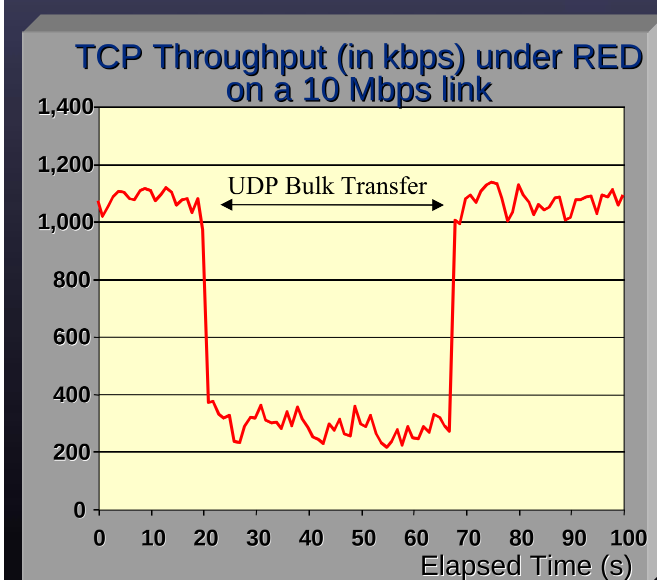
- Basic mechanism for realizing differentiated services is a RED (*random early detection*) congestion avoidance mechanism
  – Protects the network from congestive collapse
  – Increases effective network utilization
  – Decreases end-to-end latency

Realizing Differentiated Services
RED & diffserv

- Clark *et al*. RIO scheme
  – Apply "harsh RED" to out-of-profile packets and "lenient RED" to in-profile packets



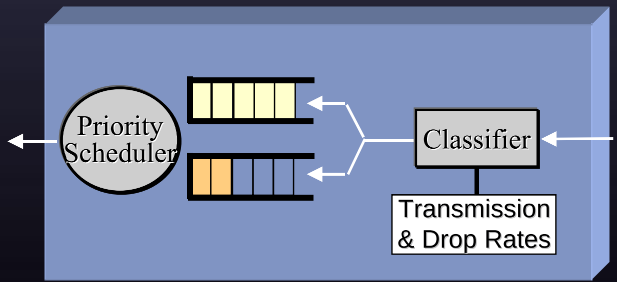
Active Queue Management
Responsive *v.* unresponsive flows



- A key assumption is that all flows respond to packet loss as a sign of congestion
- Unresponsive flows can starve responsive flows

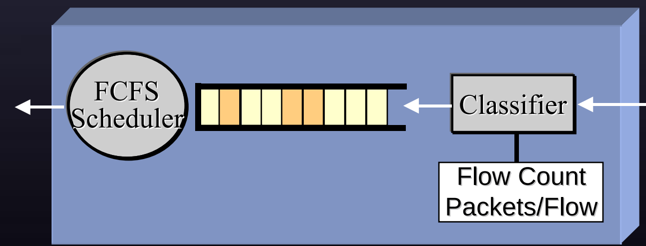
Active Queue Management
Responsive *v.* unresponsive flows

- What to do with unresponsive flows?
  - Floyd/Fall: Place them in a "penalty box"



Active Queue Management
Responsive *v.* unresponsive flows

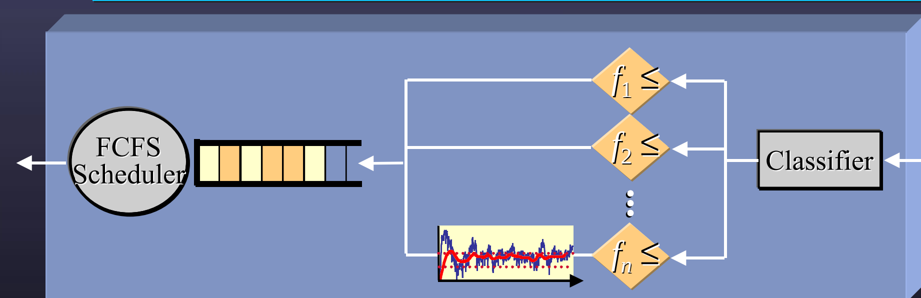
- What to do with unresponsive flows?
  - Floyd/Fall: Place them in a "penalty box"
  - Lin/Morris: Constrain them to consume no more then their "fair-share" of bandwidth



Active Queue Management
Responsive *v.* unresponsive flows

- What to do with unresponsive flows?
  - Floyd/Fall: Place them in a "penalty box"
  - Lin/Morris: Constrain them to consume no more then their "fair-share" of bandwidth
- Explicitly allocate capacity for them!
  - But rigorously police them

Managing Non-Responsive Flows
"Class-based thresholds"



- Designate a set of traffic classes and allocate a fraction of a router's buffer capacity to each class
- Once a class is occupying its limit of queue elements, discard *all* arriving packets
- Within a traffic class, further active queue management may be performed

Class-Based Thresholds
Analysis

- A CBT router is parameterized by:
  - n, the number of classes
  - $\{T_1, T_2, \ldots, T_n\}$ a set of class thresholds
- If class i is allocated capacity T_i then it will receive at least bandwidth

$$B_i = \frac{P_i T_i}{\sum_{j=1}^{n} P_j T_j} C$$

where C is the link capacity and P_i is the average class i packet size

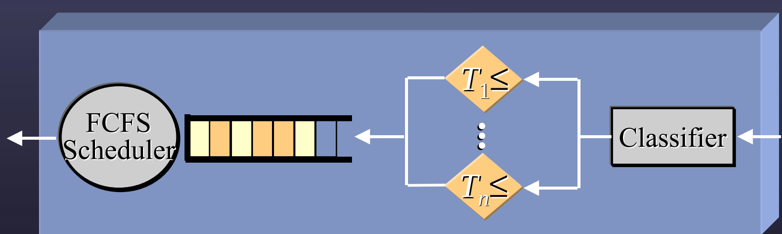
Class-Based Thresholds
Analysis

- The bandwidth actually received by a class is a function of that consumed by other classes
- Let $w_i = B_i / C$ be the "weight" of traffic class i
  - The expected link utilization of class i traffic
- If class j consumes ($load_j < B_j$) then class i receives at least bandwidth

$$B_i' = B_i + \frac{w_i}{\sum_{k=1, k \neq j}^{n} w_k} (B_j - load_j)$$

- CBT ensures weighted MAX-MIN fair allocation of bandwidth

Class-Based Thresholds
Analysis

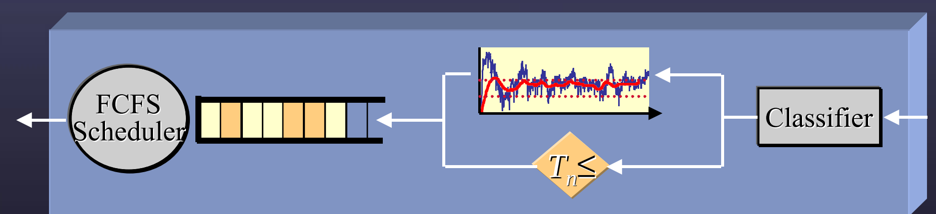


- All traffic classes experience the same worst case delay bound

$$D = \frac{1}{C} \sum_{j=1}^{n} P_j T_j$$

- Thus CBT trades link utilization for delay bounds

Class-Based Thresholds
Implementation & evaluation



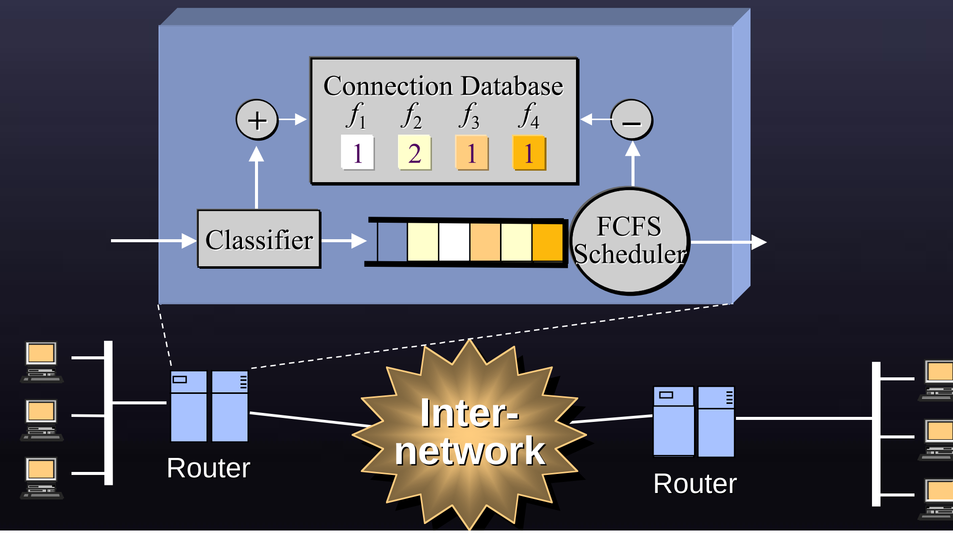
- CBT is implemented in Alt-Q on FreeBSD
- Three traffic classes currently supported:
  - TCP
  - marked non-TCP ("well behaved UDP")
  - non-marked non-TCP (all others)
- Subject TCP flows to RED and non-TCP flows to a simple queue occupancy threshold test

Class-Based Thresholds
Evaluation

- Compare:
  - FIFO queuing
  - RED
  - CBT
  - Fair buffer allocation

Evaluation
Fair buffer allocation (FRED)

- Flow Random Early Detection [Lin & Morris 97]

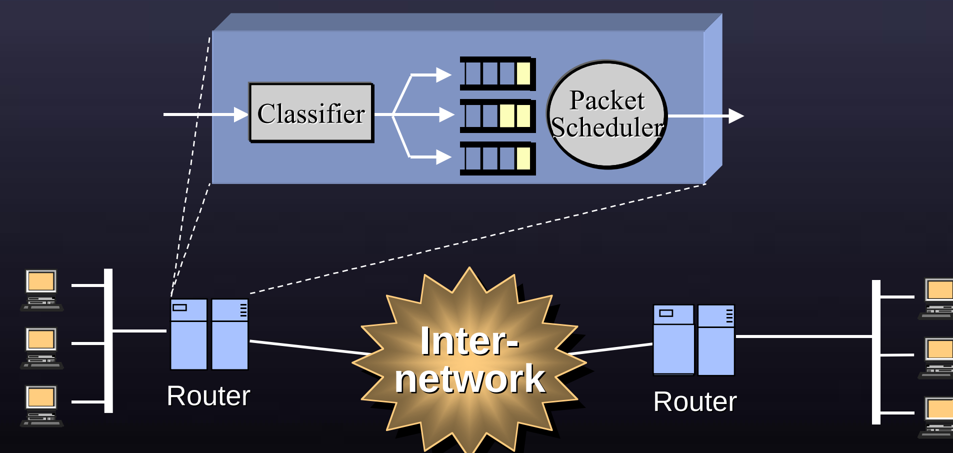


Class-Based Thresholds
Evaluation

- Compare:
  - FIFO queuing
  - RED
  - CBT
  - Fair allocation of buffers (FRED)
  - Packet scheduling

Evaluation
Packet scheduling

- Class-based queuing [Floyd & Jacobson 95]



Class-Based Thresholds
Evaluation

- Compare:
  - FIFO queuing (Negative baseline)
  - RED (The Internet of tomorrow)
  - FRED (RED + Fair allocation of buffers)
  - CBT
  - CBQ (Positive baseline)

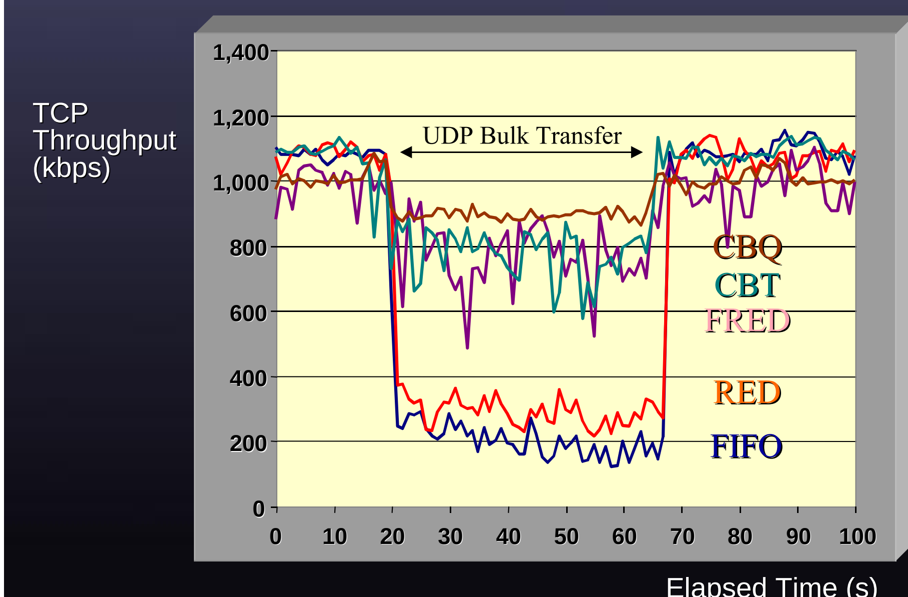
Evaluation
Experimental design

- Share a 10 Mbps link between:
  - 3,000 users browsing the web (8-9 Mbps of HTTP traffic)
  - 6-10 marked UDP ProShare flows
  - 1 unmarked UDP bulk transfer
- Performance metrics:
  - Aggregate TCP throughput
  - ProShare latency and loss
  - Algorithm complexity & state requirements

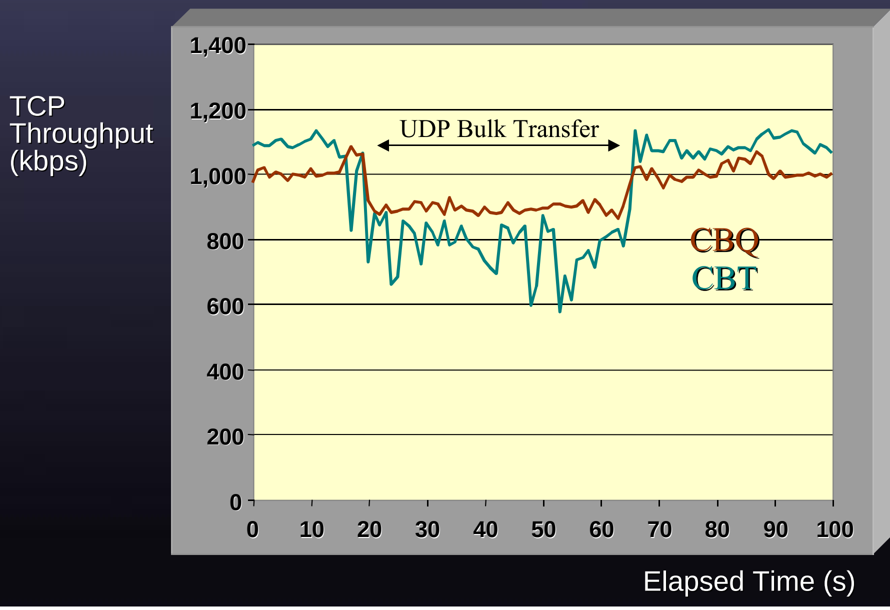
Evaluation
Experimental design

- Experimental facility

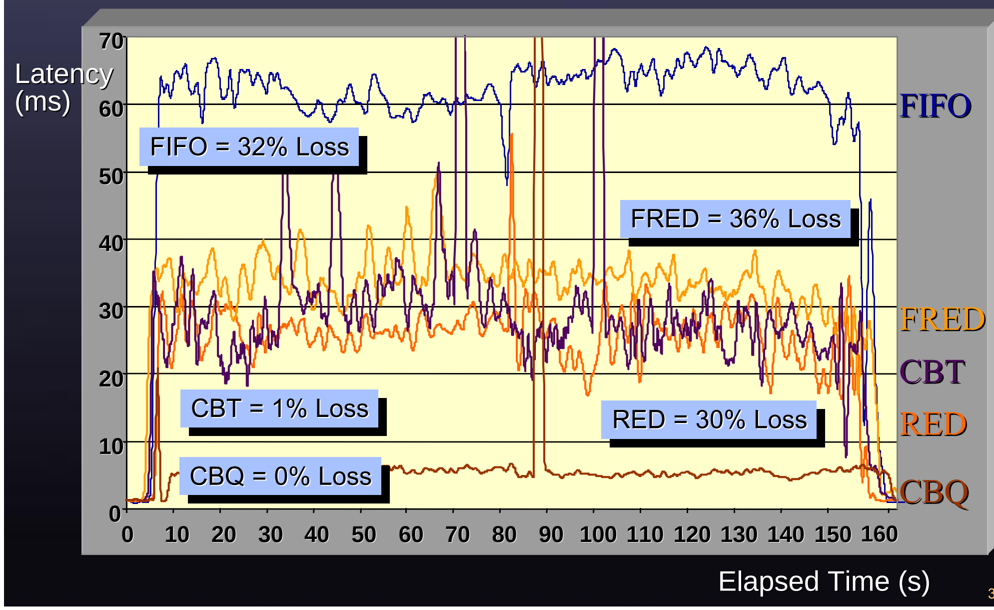
CBT Evaluation
TCP Throughput



CBT Evaluation
TCP Throughput



CBT Evaluation
ProShare (marked UDP) latency



Lightweight Active Queue Management
Summary

- Capacity allocation & protection are required for QoS
- Active queue management is at the heart of proposals for next generation QoS
- Current schemes are vulnerable to unresponsive flows
- Class-based thresholds is a compromise between RED queue management and packet scheduling

Lightweight Active Queue Management
Conclusions

- Capacity allocation & protection are required for QoS
  - Goals can be realized through simple queue management
- Class-based thresholds provides performance comparable to packet scheduling...
  - Better TCP throughput
  - Low latency and loss for multimedia applications
- … with lower state requirements and algorithmic complexity