Lightweight Active Router-Queue Management for Multimedia Networking

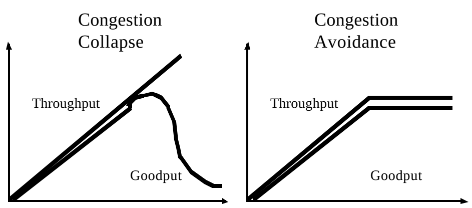
M. Parris, K. Jeffay, and F.D. Smith
Department of Computer Science
University of North Carolina

Multimedia Computing and Networking (MMCN)
January 1999

Outline

- Problem
  - Supporting multimedia on the Internet
- Context
  - Drop Tail
  - RED
  - FRED
- Approach
  - CBT
- Evaluation
- Conclusion

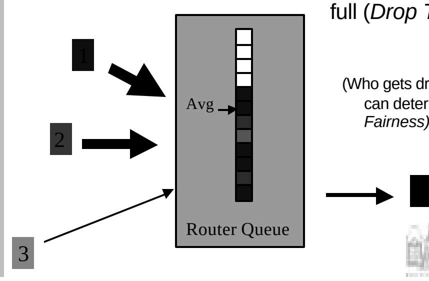
Congestion on the Internet



- Drops are the usual way congestion is indicated
- TCP uses congestion avoidance to reduce rate

Internet Routers

- Queue to hold incoming packets until can be sent
- Typically, drop when queue is full (*Drop Tail*)

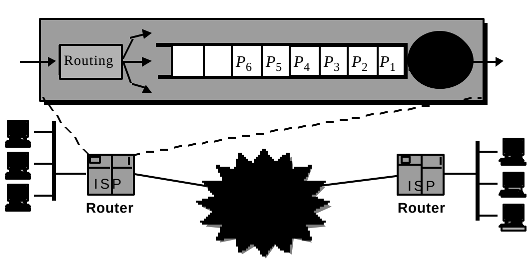


(Who gets dropped can determine *Fairness)*

Router-Based Congestion Control

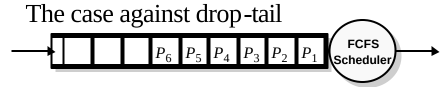
Solution 2: Closed-loop congestion control

- Normally, packets are only dropped when the queue overflows
  - "Drop-tail" queueing



Buffer Management & Congestion Avoidance

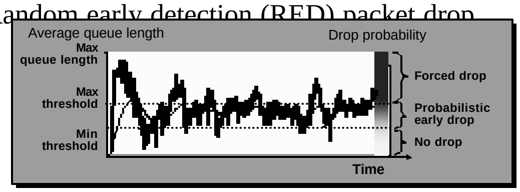
The case against drop-tail



- Large queues in routers are a bad thing
  - End-to-end latency is dominated by the length of queues at switches in the network
- Allowing queues to overflow is a bad thing
  - Connections that transmit at high rates can starve connections that transmit at low rates
  - Causes connections to synchronize their response to congestion and become unnecessarily bursty

Buffer Management & Congestion Avoidance

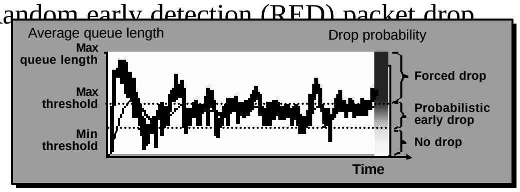
Random early detection (RED) packet drop



- Use an exponential average of the queue length to determine when to drop
  - Accommodates short-term bursts
- Tie the drop probability to the weighted average queue length
  - Avoids over-reaction to mild overload conditions

Buffer Management & Congestion Avoidance

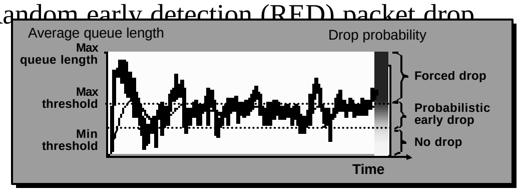
Random early detection (RED) packet drop



- Amount of packet loss is roughly proportional to a connection's bandwidth utilization
  - But there is no a priori bias against bursty sources
- Average connection latency is lower
- Average throughput ("goodput") is higher

Buffer Management & Congestion Avoidance

Random early detection (RED) packet drop



- RED is controlled by 5 parameters
  - *qlen* — The maximum length of the queue
  - w_q — Weighting factor for average queue length computation
  - min_{th} — Minimum queue length for triggering probabilistic drops
  - max_{th} — Queue length threshold for triggering forced drops
  - max_p — The maximum drop probability

Random Early Detection

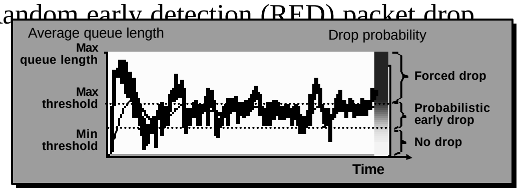
Algorithm

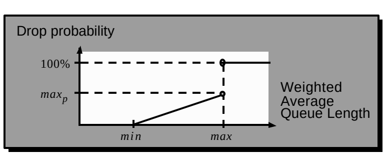
```
for each packet arrival:
  calculate the average queue size ave
  if ave    min_th
    do nothing
  else if min_th    ave    max_th
    calculate drop probability p
    drop arriving packet with probability p
  else if max_th    ave
    drop the arriving packet
```

- The average queue length computation needs to be low pass filtered to smooth out transients due to bursts
  - $ave = (1 - w_q)ave + w_q q$

Buffer Management & Congestion Avoidance

Random early detection (RED) packet drop





Random Early Detection

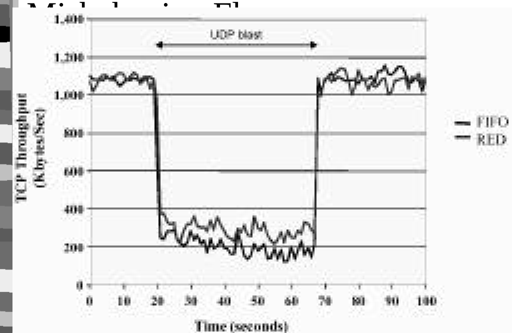
Performance

- Floyd/Jacobson simulation of two TCP (*ftp*) flows

Random Early Detection (RED) Summary

- Controls average queue size
- Drop early to signal impending congestion
- Drops proportional to bandwidth, but drop rate equal for all flows
- Unresponsive traffic will still not slow down!

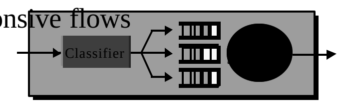
RED Vulnerability to



- TCP performance on a 10 Mbps link under RED in the face of a "UDP" blast

Router-Based Congestion Control

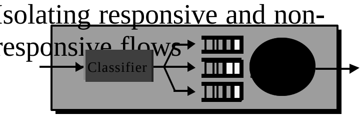
Dealing with heterogeneous/non-responsive flows



- TCP requires protection/isolation from non-responsive flows
- Solutions?
  - Employ fair-queuing/link scheduling mechanisms
  - Identify and police non-responsive flows (not here)
  - Employ fair buffer allocation within a RED mechanism

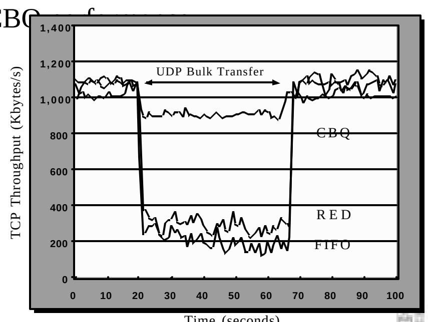
Flows

Isolating responsive and non-responsive flows

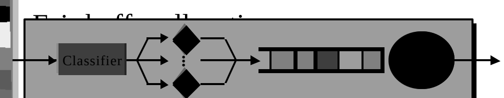


- Class-based Queuing (CBQ) (Floyd/Jacobson) provides fair allocation of bandwidth to traffic classes
  - Separate queues are provided for each traffic class and serviced in round robin order (or weighted round robin)
  - *n* classes each receive exactly 1/*n* of the capacity of the link
- Separate queues ensure perfect isolation between classes
- Drawback: 'reservation' of bandwidth and state information required

Dealing With Non-Responsive Flows



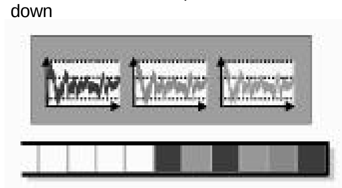
Dealing With Non-Responsive Flows



- Isolation can be achieved by reserving capacity for flows within a single FIFO queue
  - Rather than maintain separate queues, keep counts of packets in a single queue
- Lin/Morris: Modify RED to perform fair buffer allocation between active flows
  - Independent of protection issues, fair buffer allocation between TCP connections is also desirable

Flow Random Early Detect (FRED)

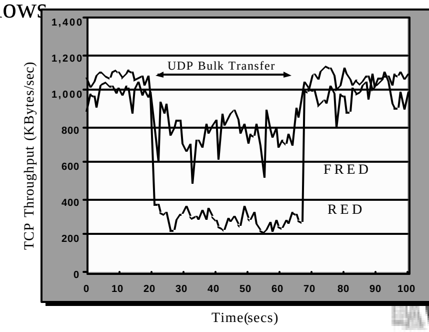
- In RED, 10 Mbps → 9 Mbps and 1Mbps → .9 Mbps
  - Unfair
- In FRED, leave 1 Mbps untouched until 10 Mbps is down



- Separate drop probabilities per flow
- "Light" flows have no drops, heavy flows have high drops

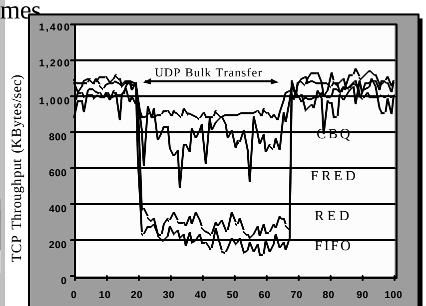
Flow Random Early Detection

Performance in the face of non-responsive flows



Congestion Avoidance *v.* Fair-Sharing

TCP throughput under different queue management schemes



- TCP performance as a function of the state required to ensure/approximate fairness

Queue Management Recommendations

- Recommend (Braden 1998, Floyd 1998)
  - Deploy RED
    - + Avoid full queues, reduce latency, reduce packet drops, avoid lock out
  - Continue research into ways to *punish* aggressive or misbehaving flows
- Multimedia
  - Does not use TCP
    - + Can tolerate some loss
    - + Price for latency is too high
  - Often low-bandwidth
  - Delay sensitive

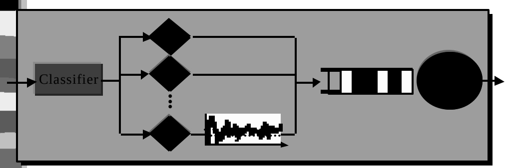
Outline

- Problem
  - Supporting multimedia on the Internet
- Context
  - Drop Tail
  - RED
  - FRED
- ***Approach***
  - CBT
- Evaluation
- Conclusion

Goals

- Isolation
  - Responsive (TCP) from unresponsive
  - Unresponsive: multimedia from aggressive
- Flexible fairness
  - Something more than equal shares for all
- Lightweight
  - Minimal state per flow
- Maintain benefits of RED
  - Feedback
  - Distribution of drops

Class-Based Threshold (CBT)

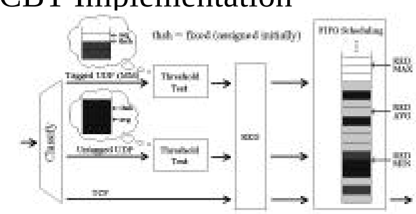


- Designate a set of traffic classes and allocate a fraction of a router's buffer capacity to each class
- Once a class is occupying its limit of queue elements, discard *all* arriving packets
- Within a traffic class, further active queue management may be performed

Class-Based Threshold (CBT)

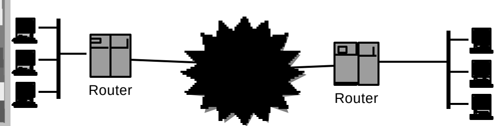
- Isolation
  – Packets are classified into 1 of 3 classes
  – Statistics are kept for each class
- Flexible fairness
  – Configurable thresholds determine the ratios between classes during periods of congestion
- Lightweight
  – State per class and not per flow
  – Still one outbound queue
- Maintain benefits of RED
  – Continue with RED policies for TCP

CBT Implementation



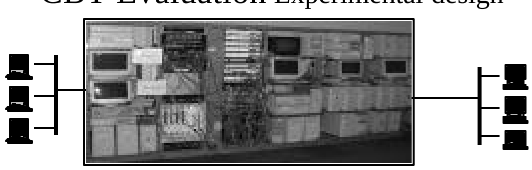
- Implemented in Alt-Q on FreeBSD
- Three traffic classes:
  –TCP
  –marked non-TCP ("well behaved UDP")
  –non- marked non-TCP (all others)
- Subject TCP flows get RED and non-TCP flows to a weighted average queue occupancy threshold test

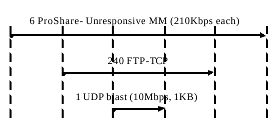
Class-Based Thresholds
Evaluation



- Compare:
  – FIFO queuing (Lower bound baseline)
  – RED (The Internet of tomorrow?)
  – FRED (RED + Fair allocation of buffers)
  – CBT
  – CBQ (Upper bound baseline)

CBT Evaluation Experimental design





Throughput and Latency

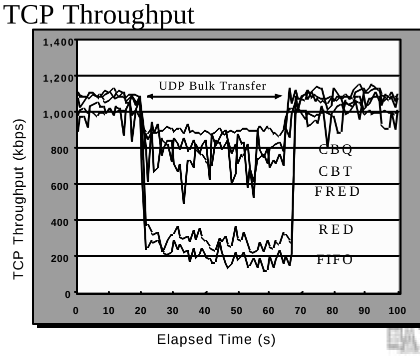
- RED Settings:

```
qsize   = 60 pkts
max-th  = 30 pkts
min-th  = 15 pkts
Wq      = 0.002
max-p   = 0.1
```

- CBT Settings:

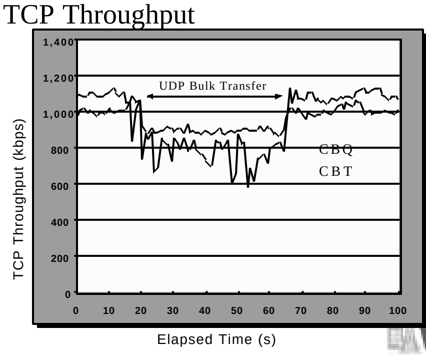
```
mm-th   = 10 pkts
udp-th  = 2  pkts
```

CBT Evaluation
TCP Throughput



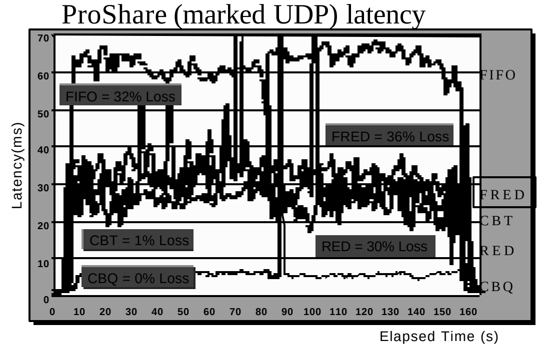
CBT Evaluation

TCP Throughput



CBT Evaluation

ProShare (marked UDP) latency



Conclusion

- RED/FIFO scheduling not sufficient
  - Aggressive unresponsive flows cause trouble
  - Low bandwidth unresponsive (VoIP) punished
- CBT provides
  - Benefits of RED for TCP only traffic
  - Isolation of TCP vs. Unresponsive
  - Isolation of Aggressive vs. Low Bandwidth
  - Lightweight overhead

Future Work?

Future Work

- How to pick thresholds?
  - Implies reservation
  - Dynamic adjustments of thresholds (D-CBT)
- Additional queue management for classes
  - Classes use “Drop Tail” now
- Extension to other classes
  - Voice
  - Video