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
Congestion Avoidance
Congestion Collapse
Throughput
Throughput
Goodput
Goodput

• 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)

Router Queue
(Who gets dropped can determine Fairness)

Typically, packets are only dropped when the queue overflows
“Drop-tail” queueing

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

Max threshold
Min threshold
Average queue length
Forced drop
Probabilistic early drop
No drop

Drop probability vs. time

Max threshold
Min threshold
Average queue length
Forced drop
Probabilistic early drop
No 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

Random Early Detection Algorithm

For each packet arrival:
  calculate the average queue size \( \bar{q} \)
  if \( \bar{q} \leq \min_{\text{th}} \) do nothing
  else if \( \min_{\text{th}} < \bar{q} < \max_{\text{th}} \) calculate drop probability \( p \)
    drop arriving packet with probability \( p \)
  else if \( \bar{q} \geq \max_{\text{th}} \) drop the arriving packet

The average queue length computation needs to be low pass filtered to smooth out transients due to bursts

\[ \bar{q} = (1 - w_q) \bar{q} + w_q q \]

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!

Vulnerability to Misbehaving Flows

- 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-queueing/link scheduling mechanisms
  - Identify and police non-responsive flows (not here)
  - Employ fair buffer allocation within a RED mechanism

Dealing With Non-Responsive Flows

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

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

TCP throughput (KBytes/sec)

0 10 20 30 40 50 60 70 80 90 100

0 200 400 600 800 1,000 1,200 1,400

UDP Bulk Transfer

FRED

RED

Time (sec.)

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.

CBT Implementation

- Implemented in Al-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.

CBT Evaluation

- RED Settings:
  - qsize = 60 pkts
  - max-th = 30 pkts
  - min-th = 15 pkts
  - qweight = 0.002
  - max-pro = 0.1
- CBT Settings:
  - mm-th = 10 pkts
  - udp-th = 2 pkts

Evaluation

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

TCP Throughput

Throughput and Latency

UDP Bulk Transfer

Elapsed Time (s)
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

- 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

Evaluation of Science?

- Category of Paper
- Science Evaluation (1-10)?
- Space devoted to Experiments?