RAP: An End-to-End Rate-Based Congestion Control Mechanism for Realtime Streams in the Internet

Reza Rejai, Mark Handley, Deborah Estrin
U of Southern California
Marina Del Rey, CA 90292, USA

IEEE Infocom, 1999

Outline

- Introduction
- Approach
- Evaluation
- Conclusions

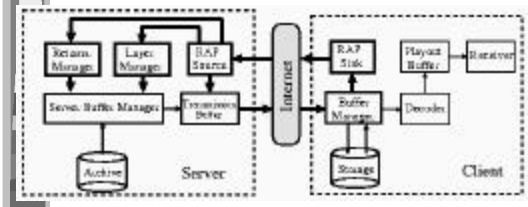
Introduction

- MM is delay sensitive, semi-reliable, rate-based
  - Not in Internet
- Still MM apps have grown
  - Those typically allow larger delay (ex: VoD)
  - So can do buffering to remove variance
- Must react to congestion in TCP-Friendly fashion

Research Approach

- Separate congestion control from error and quality control
- RAP module
  - Congestion control via AIMD (converges to fair)
  - Congestion detection vial loss (ECN possible)
  - Order of RTT
- Layer manager
  - Quality control
  - Layer manager (maximum layers to fit bwidth)
  - Less than RTT by buffering
- (RAP module primary in this paper)

RAP Architecture



This paper concentrates on RAP module

Related Work

- MM could use special services or resource reservation
  - *DiffServ* or *IntServ*
  - But would still be need for low-cost mm traffic
- TCP without retransmission
  - Still window-based, fluctuating
- Some work adaptively encodes based on feedback from network
  - CPU intensive for all
  - Not shown to work for large range of environs
- Netshow and RealPlayer claim to be adaptive
  - No scientific evaluation

The RAP Protocol

- RAP "machinery" mainly at source
  - Receiver acks each packet
  - Sender calculates stuff (loss and RTT)
- Sender must have:
  - *Decision function*
  - *Increase/Decrease algorithm*
  - *Decision frequency*

Decision Function

- No congestion then periodically increase rate
- Congestion then immediately decrease rate
  - Gaps and timeouts
- Acks have more than just last sequence
  - Can send back 'holes' to avoid responding to single loss events

Increase/Decrease Algorithm

- Uses AIMD
- Change inter-packet gap (IPG)
  - Smaller gap then higher rate
  - Larger gap then lower rate
  - Step-wise decrease on no congestion
  - Double on congestion

Decision Frequency

- Change rate no more than 1 per RTT
  - Step "length" is RTT
- Then, if "height" is 1 packet
  - Emulate TCP during congestion avoidance
- (Hint at startup phase, but since 'long-lived', startup is not crucial)
  - (Me, but what about sudden changes in bandwidth? Can be 'like' startup phase!)
- Special cases:
  - Clustered losses
  - Fine-grain rate adaptation
  - RED gateways

Clustered Losses

- Packets lost together are part of same congestion signal
- $Seq_{firstLoss}$ is first packet lost
- $Seq_{lastSent}$ is last one sent
- Ignore
  - $Seq_{lastSent} > seq > Seq_{firstLoss}$
- Similar to TCP SACK

Fine-Grain Rate Adaptation

- Keep long term RTT and short-term RTT
- IPG is based on short-term RTT
- IPG' is based on ratio of long-term to short-term
  - Use it to adjust rate

Random Early Detection Gateways

- TCP has trouble with multiple losses in a row in one window
  - Drop Tail queues
  - TCP times-out to window size of 1
  - RAP goes down exponentially
- RED distributes losses
  - Also smaller queue sizes (hopefully)
  - Wants 1 loss for each flow for each RTT
- With RED, RAP should be totally fair to TCP
- Configuring RED still difficult
  - Major topic of CC meetings

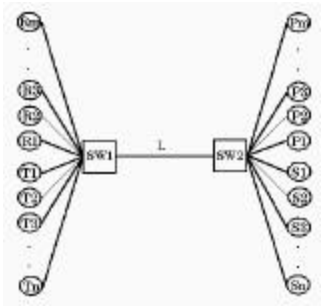
Outline

- Introduction
- Approach
- *Evaluation*
- Conclusions

Evaluation by Simulation (NS)

- TCP-friendliness
- Ability to cope with background traffic
- Interaction with RED gateways
- Fine-grain rate adaptation

Simulation Topology



- Standard "Dumbell"
- TCP with RAP
  - Infinite amount to send

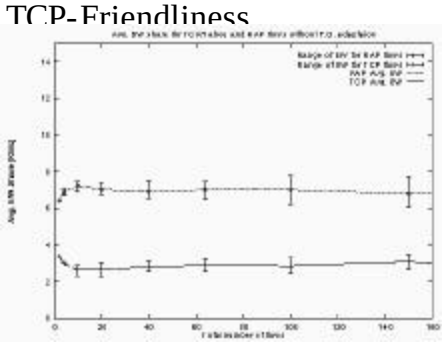
Simulation Parameters

Packet Size	100 Byte
ACK Size	40 Byte
Bottleneck Delay	20 ms
B/W per Flow	5 KByte/s
B/W of Side Links	1.25 MByte/s
Tot. Delay of Side-Links	6 ms
Simulation Length	120 sec
TCP Maximum Window	1000
TCP Timeout Granularity	100 ms

- Data and ack packet sizes same
- RTT same for all
- Router has 4x bwidth x delay
- Simulations at steady state

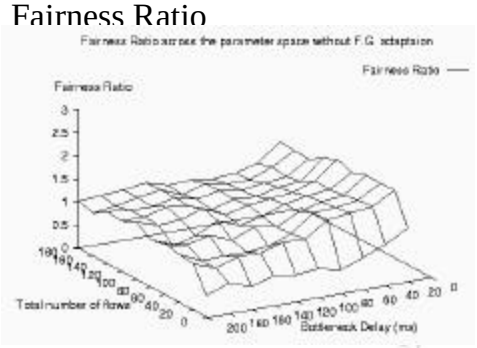
(Me, small packets!)

TCP-Friendliness



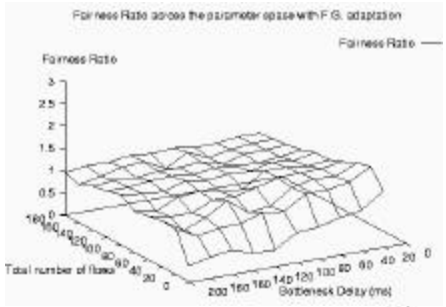
- Bwidth scaled up with number of flows
- Tahoe cannot handle multiple losses in 1 window
- Rest of tests are with SACK (more AIMD)

Fairness Ratio



- Half RAP, Half TCP
- Many flows fair
- Low delay, not fair

Fine-Grain Fairness



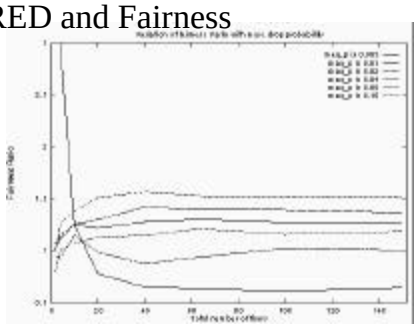
Better for low-delay than before

RED Routers

Min. Threshold	5 Packets
Max. Threshold	0.5 * Buffer
Bottleneck B/W	5 KByte/s * No. of Flows
Bottleneck Delay	20 ms
Buffer Size	12 * RTT * Bottleneck B/W
q_weight	0.002

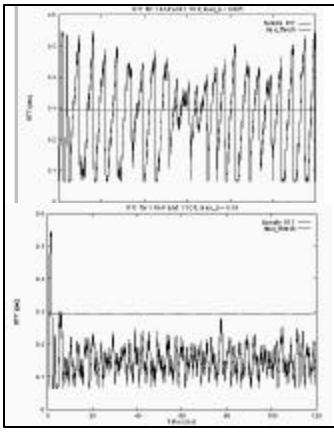
- No buffer overflow
  - Typically, recommend 2-4 bwidth x delay
- Other work shows max_p needs to be tuned to number of flows

RED and Fairness



- Small simulations vary lots w/num flows
- Large simulations steady with num flows
- Unfair when avg q is at maxth

Sample RTT



Low RTT means fewer drops
But too low has small window and then timeouts

Conclusion

- Rate-based
- Reasonably Fair to TCP
  - Unfair when TCP not AIMD (timeout)
- More Fair when RED gateway
- Future
  - Build it and run on Internet
  - Applications on top (Layered adaptation)

Winner?

- MM-Flow
- TFRC
- RAP
- Debate?

Evaluation of Science?

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