Equation-based Congestion Control

for Unicast and Multicast Applications

Jörg Widmer

Praktische Informatik IV, University of Mannheim /
AT&T Center for Internet Research at ICSI (ACIRI)

Feb 05, 2001

Why Use Congestion Control?

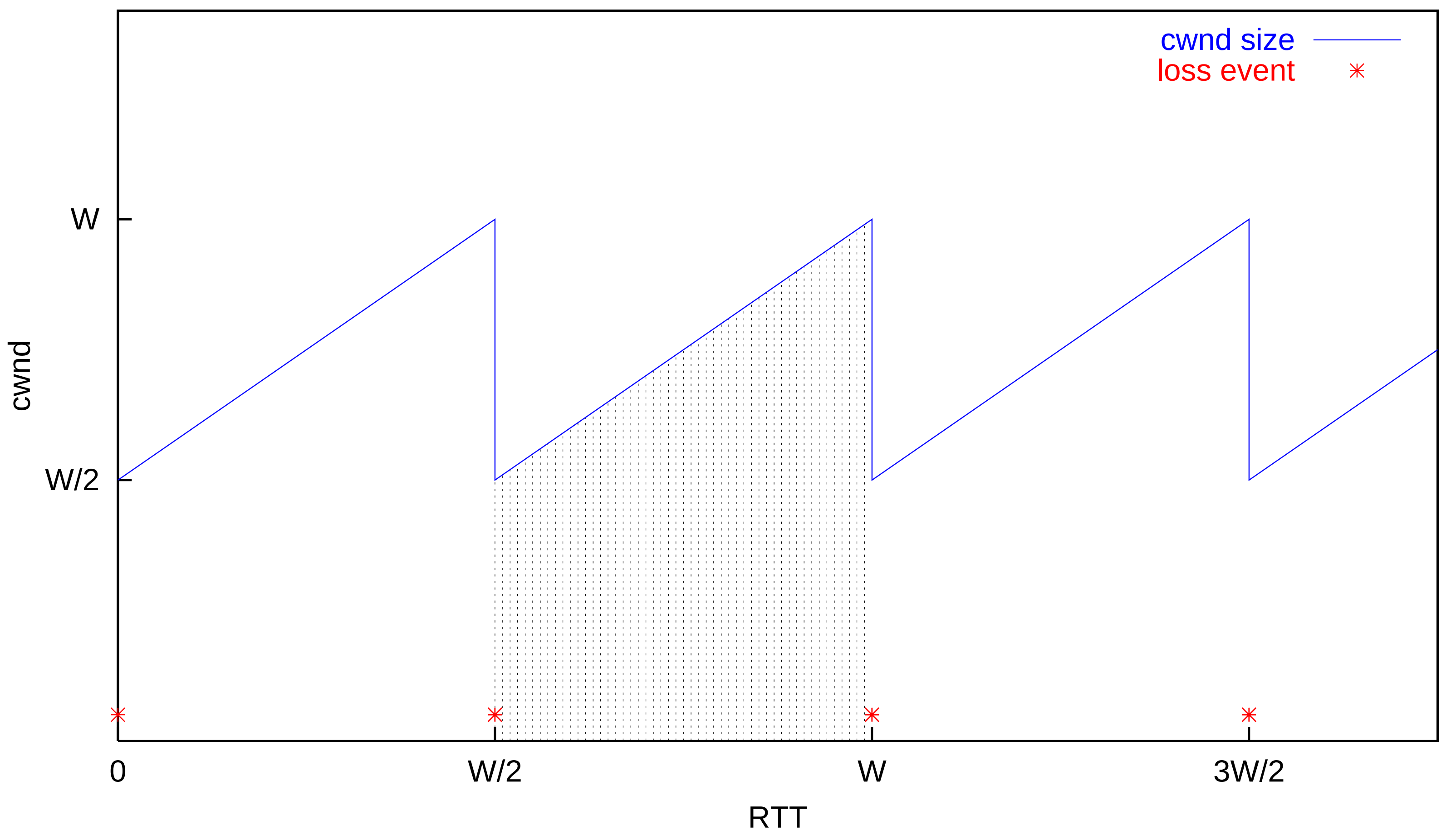
- Increasing volume of non-TCP traffic
- Multicast Transport Protocols
- High speed Internet connections for end-users
- Low delay in the network

Requirements for a Congestion Control Protocol

- Stability
- Wide adaptive range
- Fairness (TCP-Friendliness)
- Responsiveness
- High performance

Modeling TCP Throughput

- Window-based congestion control
- Additive Increase Multiplicative Decrease (AIMD)
  - increase by $\sim$1 segment per RTT
  - decrease by half in case of packet loss
  - decrease to one in case of timeout

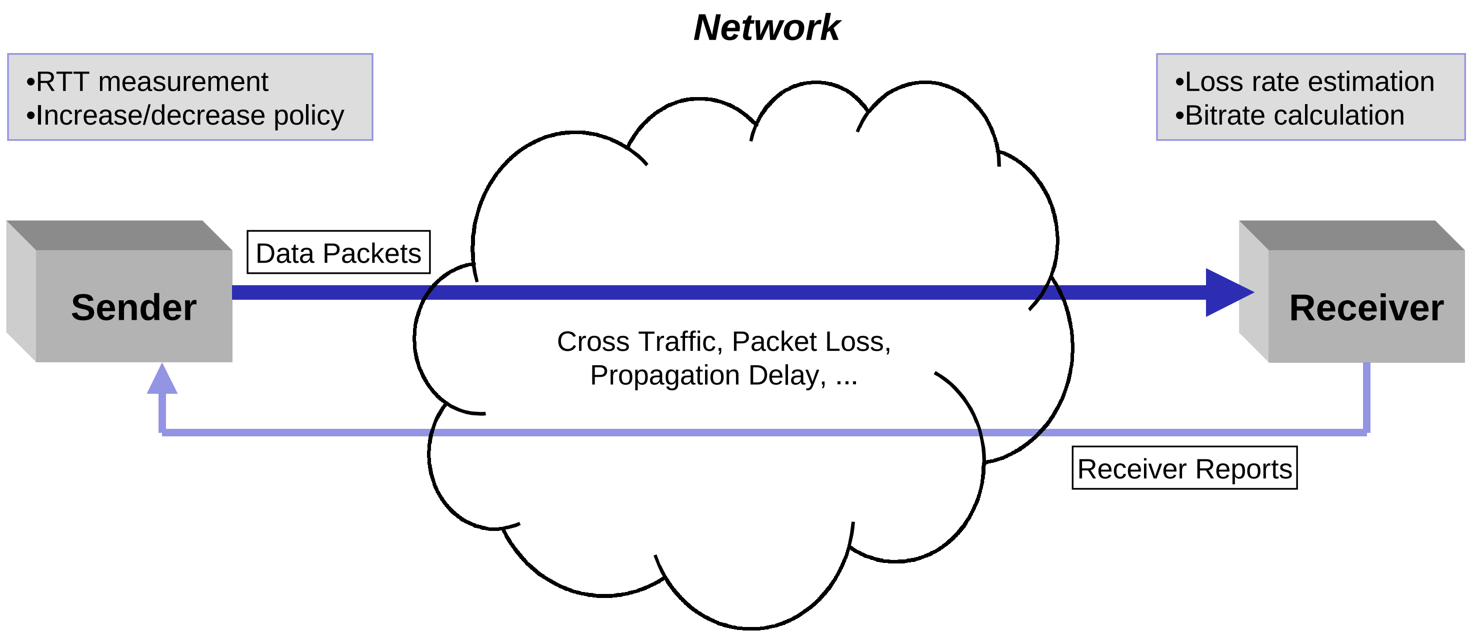

cwnd size
loss event
W
W/2
cwnd
0
W/2
W
3W/2
RTT

Formula for TCP Throughput

$$T(l) = \frac{s}{t_{RTT}\left(\sqrt{\frac{2p}{3}} + 12\sqrt{\frac{3p}{8}}\,p\,(1 + 32\,p^2)\right)}$$

- p = packet loss rate
- s = packet size
- t_{RTT} = round-trip time

TFRC Overview



- Sending rate as a function of the loss rate and RTT

TFRC

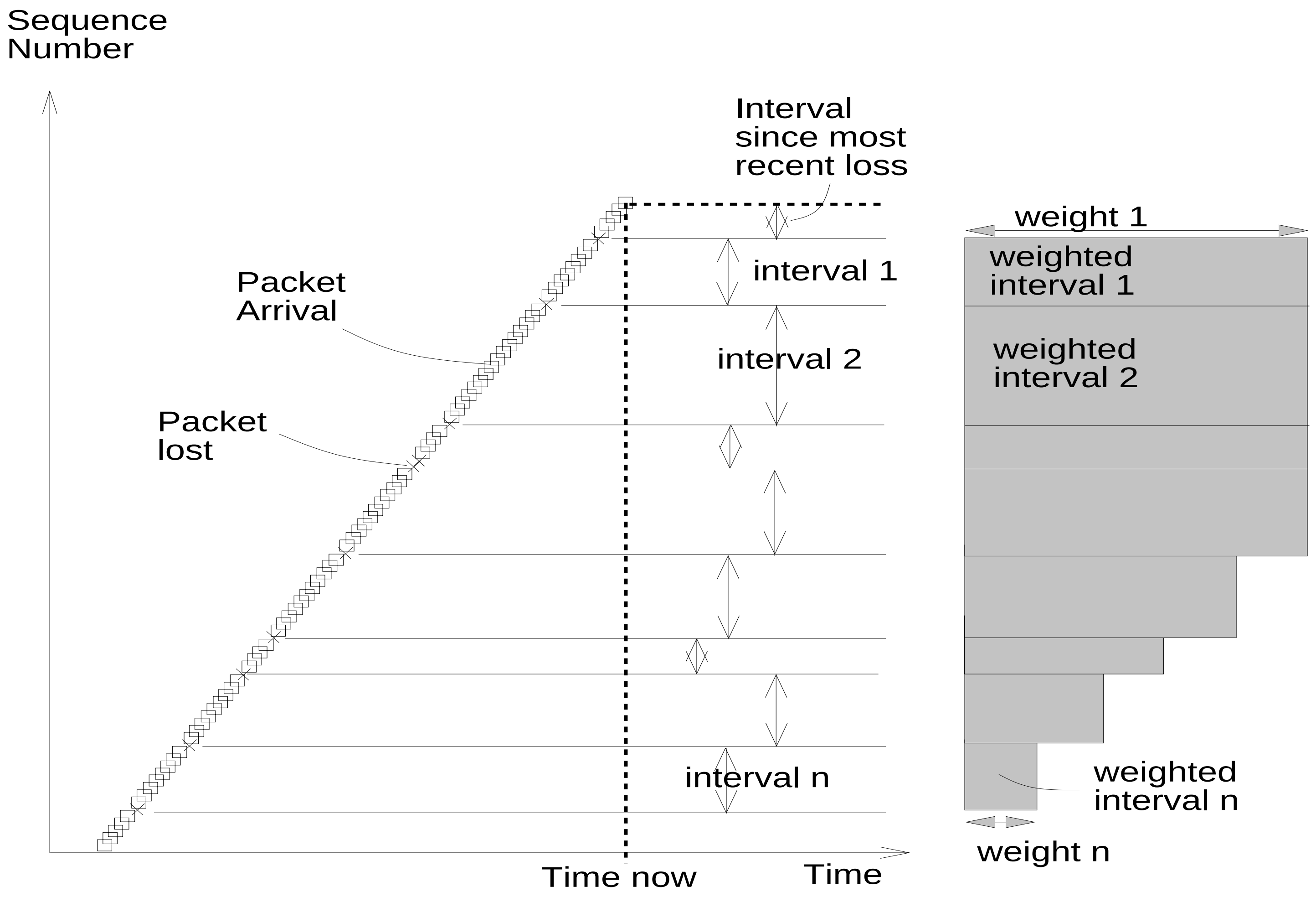
Basically TFRC functions as follows:

1. The receiver measures the packet loss rate and feeds this information back to the sender.
2. The sender uses the feedback messages to measure the round-trip time to the receiver.
3. The sender uses the control equation to derive an acceptable transmission rate from the measured loss rate and RTT.
4. The sender's transmission rate is then adjusted directly to match the calculated transmission rate.

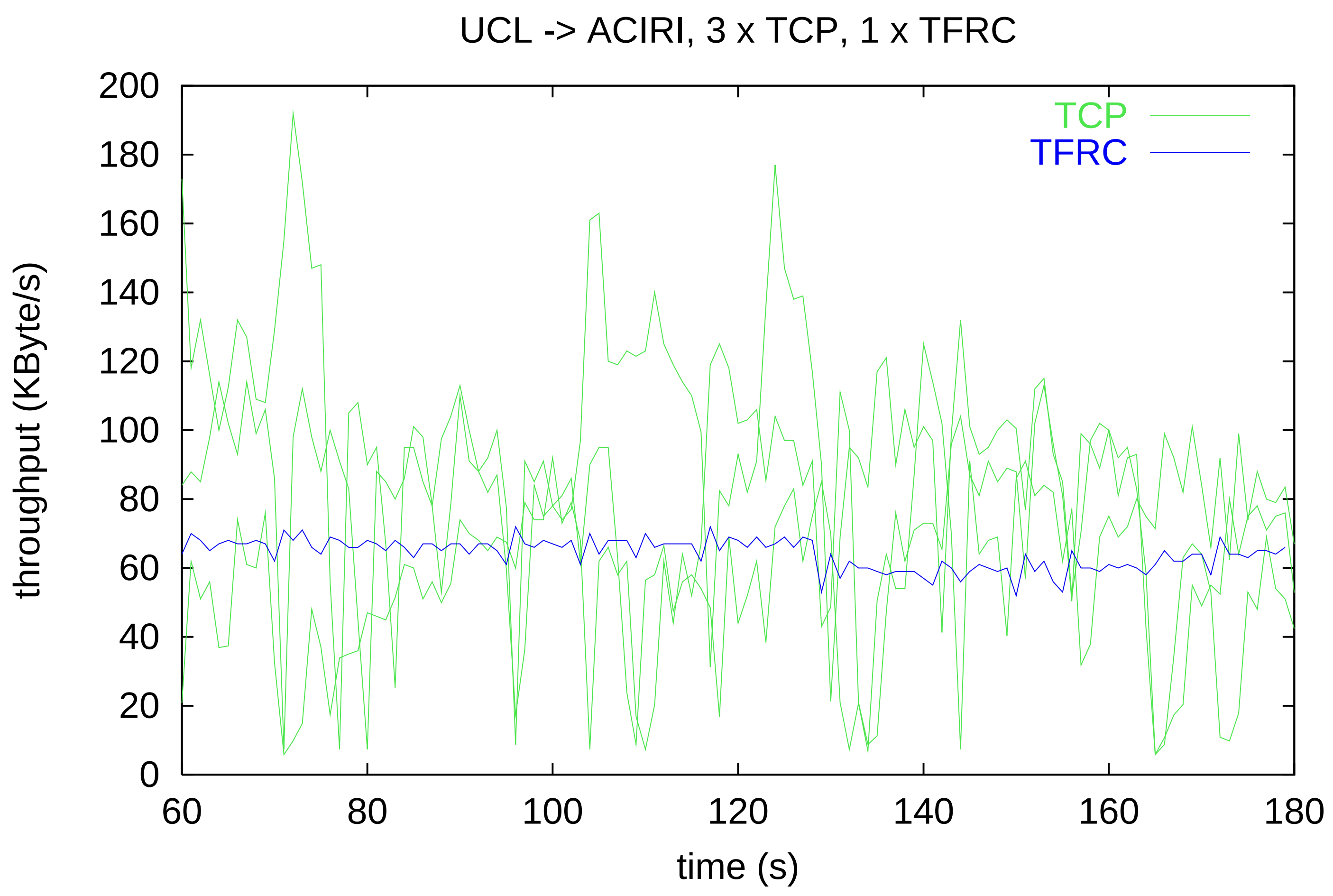
Round-trip Time Measurements

- Sender timestamps data packets
- Receiver echoes the timestamp in the next report
- Sender calculates instantaneous RTT as difference of current time and timestamp value
- Smoothe RTT samples using an exponentially weighted moving average

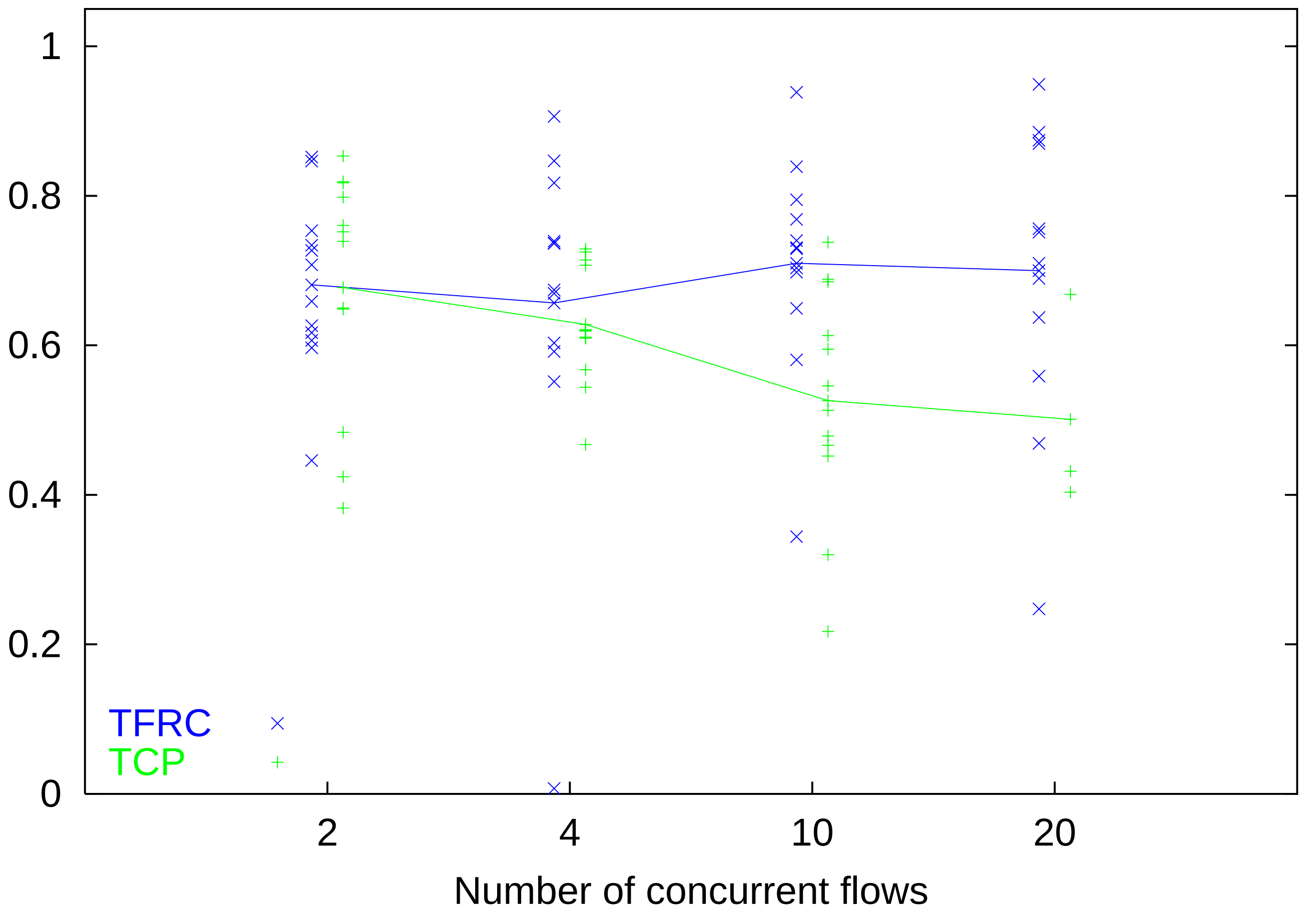
Measuring the Loss Event Rate



Throughput Comparison



Intra-Protocol Fairness



Extending the Framework to Multicast

Two main challenges:

- Scalable RTT measurements to a large receiver set
- Timely feedback from the “right” receivers without provoking a feedback implosion

...and some minor difficulties

- Slowstart
- Responsiveness
- Joining and leaving receivers
- Receiver heterogeneity

Design Issues

- Moving functionality from sender to receiver to improve scalability
  - Rate calculation
  - RTT measurements
- Adjust sending rate to the “worst” receiver
- Too large heterogeneities among the receivers have to be resolved at the application level

Current Limiting Receiver (CLR)

- Receivers give feedback only when their calculated rate is less than the current sending rate
- But then how do we increase the transmission rate?
  (We cannot afford to increase the transmission rate in the absence of feedback, as the feedback path from the slowest receiver may be congested or lossy.)

- Concept of the *current limiting receiver* (CLR)
- CLR is permitted to send immediate feedback without any form of suppression
- CLR will change if another receiver sends feedback indicating that a lower transmission rate is required

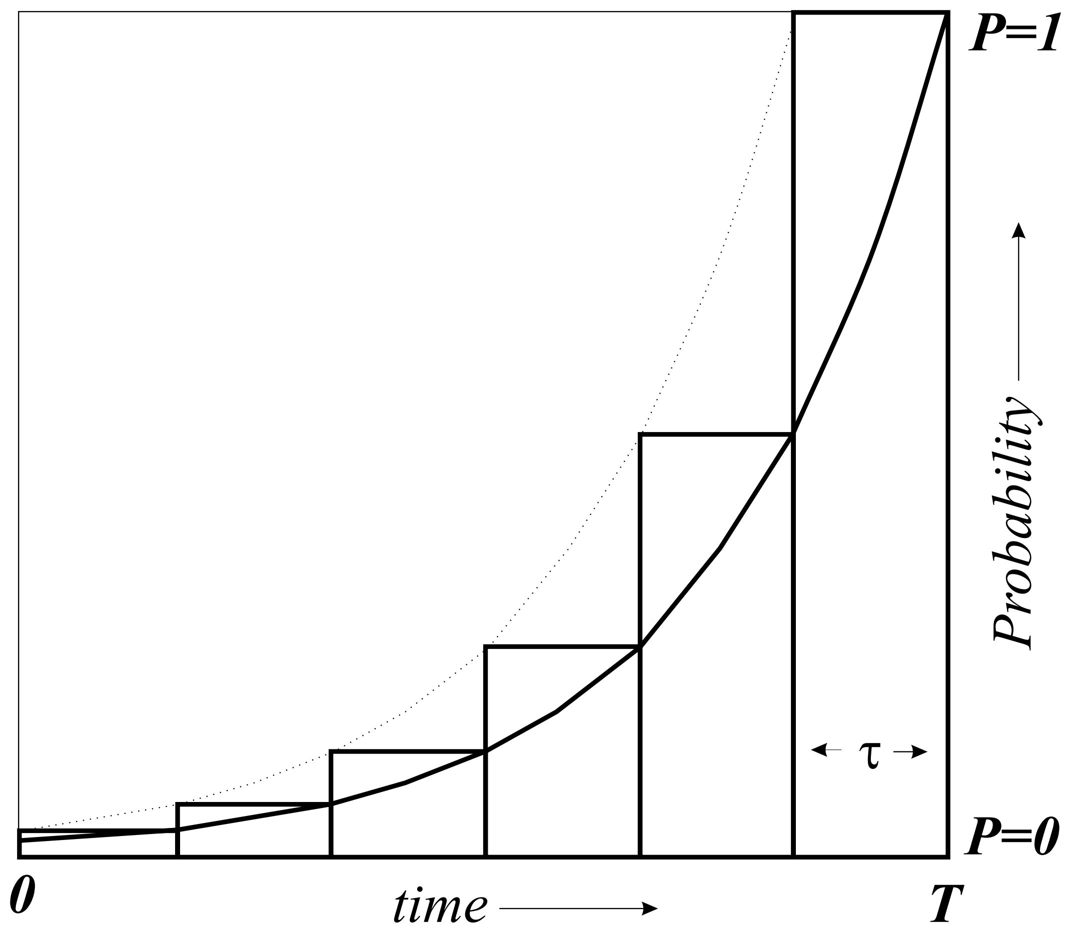
Measuring the Round-Trip Time

- RTT measurements at receivers
- One-way delay jitter used to adjust the RTT between measurements
- Initial RTT: use conservative estimate (e.g. 500ms)

- RTT measurement and rate calculation at the sender when a receiver reports a low rate but does not have a valid RTT to reduce oscillations

Feedback Suppression

- Sender sends feedback request
- Receivers set feedback timer and wait
- When the timer expires and no other feedback was received in the meantime they send a response
- Feedback timer: $t = T \cdot (1 + \log_N(x))$
  where $T = 4 \cdot RTT_{max}, N = 10000, x \in [0, 1]$


P=1
P=0
Probability
τ
0
time
T

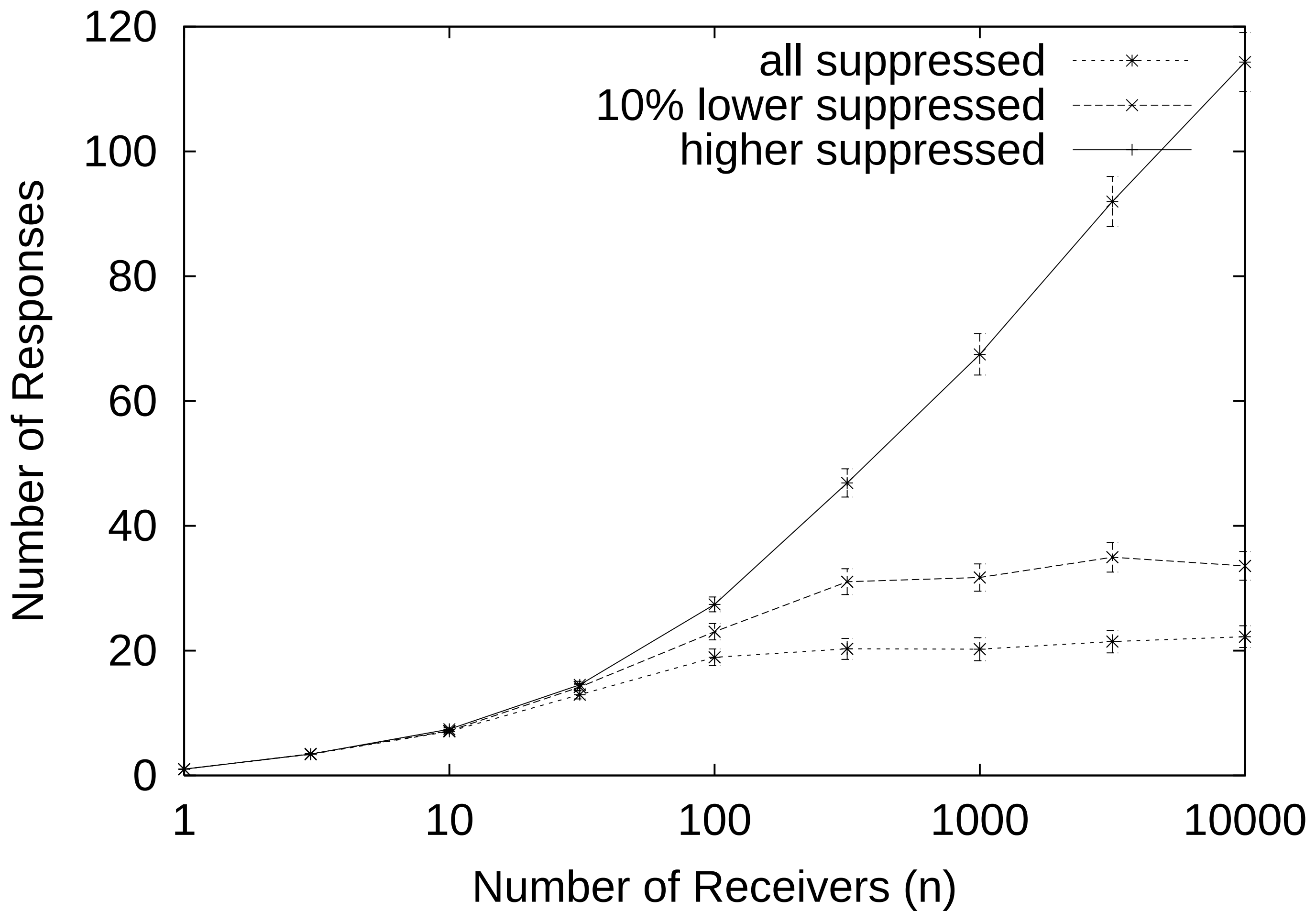
Biased Feedback Timer

- Low-rate receivers should give feedback earlier than others
- "Importance" of a receiver's feedback: $r = \frac{\text{calculated rate}}{\text{sending rate}}$
- Biased feedback timer: $t' = r\gamma T + (1-\gamma)T \cdot (1 + \log_N x)$
  - $(1-\gamma)T \cdot (1 + \log_N x)$ to prevent a feedback implosion
  - $r\gamma T$ to spread out responses with respect to their importance

Canceling the Feedback Timer

Different alternatives to cancel the feedback timer:

- after first feedback is received
- after lower-rate feedback is received
- after feedback reporting a rate that is a certain percentage lower is received (compromise)


all suppressed
10% lower suppressed
higher suppressed
Number of Responses
Number of Receivers (n)
0
20
40
60
80
100
120
1
10
100
1000
10000

Conclusions

- Implementation as transport protocol
- No substitute for TCP but alternative for flows that would otherwise not use any congestion control at all
- So far only multicast simulations, we really need a real-world implementation (e.g. rdist)
- Still several open issues that have to be solved (particularly for Multicast Congestion Control)
- ... but so far it looks quite promising