Delay-based AIMD congestion Control

D. Leith[1], R. Shorten[1], G. McCullagh[1], J. Heffner[2], L. Dunn[3], F. Baker[3]

[1] *Hamilton Institute, NUI Maynooth, IE*

[2] *Pittsburgh Supercomputing Center, US*

[3] *Cisco Systems, US*

Hamilton Institute

Overview

1. Environments which challenge TCP
2. Delay-based congestion control
3. Delay-based AIMD algorithm
4. Experiment Results
5. Conclusions

Hamilton Institute

Environments which challenge TCP

Environments which challenge TCP

- High speed networks — large BDP
- Satellite links — very long delays
- Wireless links — non-congestive losses
- aDSL links — large latencies

Large BDPs, large queues, large transmission times, random losses.

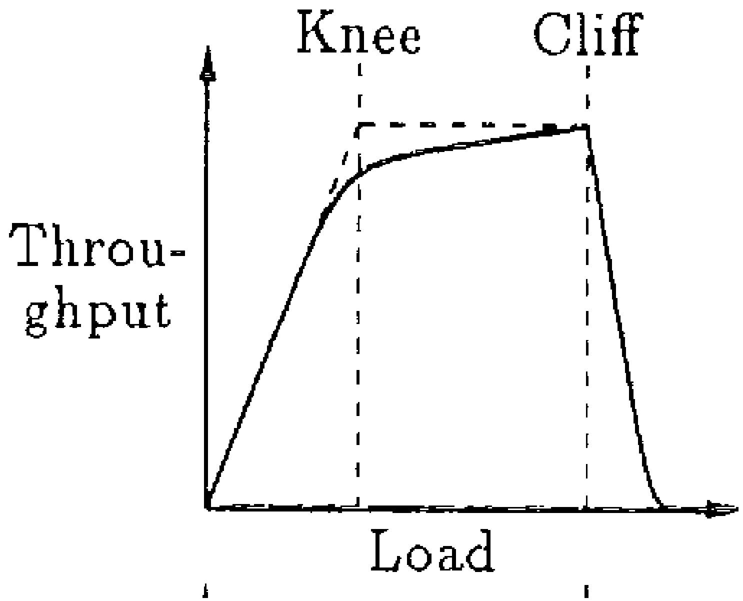
Proposed Solution

Combine ideas from high-speed and delay-based protocols

- Fast recovery to improve throughput on large BDP links
- Allow new flows to gain their share quickly, even with long transmission times
- Maintain low queueing delay

Hamilton Institute

Operation “At the knee of the curve”



Jain’s “Sweet spot” around which delay is low but throughput is high

Hamilton Institute

Delay-based Congestion Control

Hamilton Institute

Delay-based Congestion Control

- Suggested by Brakmo et al in 1994
- Proposed the "Vegas" protocol.
- Others include FAST, Compound-TCP.
- Vegas is one of the more widely explored delay-based cc algorithms

Delay-based Congestion Control

Possible benefits:

- No congestive losses
- Doesn't fill queues, lower delays
- Lower cost per congestion event

Problems:

- Coexistence with loss-based systems
- Difficulty in accurately measuring delay, sampling rates
- Limited correlation between delay and congestion
- Delay scales with number of flows

Hamilton Institute

Brief nalysis of Vegas

Hamilton Institute

Vegas

The expected and observed throughputs are:

$$r_{exp} = \frac{w}{T_{min}} \quad (1)$$

$$r_{obs} = \frac{w}{T_{min} + \tau} \quad (2)$$

$$\epsilon = r_{exp} - r_{obs} \quad (3)$$

$$= \frac{w}{T_{min}} - \frac{w}{T_{min} + \tau} \quad (4)$$

Hamilton Institute

Vegas

Once per round trip-time cong. window, w is adjusted:

$$w \leftarrow \begin{cases} w+1 & \epsilon < \alpha \\ w & \epsilon \in [\alpha, \beta] \\ w-1 & \epsilon > \beta \end{cases} \tag{5}$$

where α and β are design parameters.

Vegas Latency Scaling

So, at equilibrium Vegas maintains the congestion window such that

$$\alpha \leq \epsilon \leq \beta \quad (6)$$

$$\alpha \leq \frac{w}{T_{min}} - \frac{w}{T_{min}+\tau} \leq \beta \quad (7)$$

$$\alpha \leq \frac{w\tau}{T_{min}(T_{min}+\tau)} \leq \beta \quad (8)$$

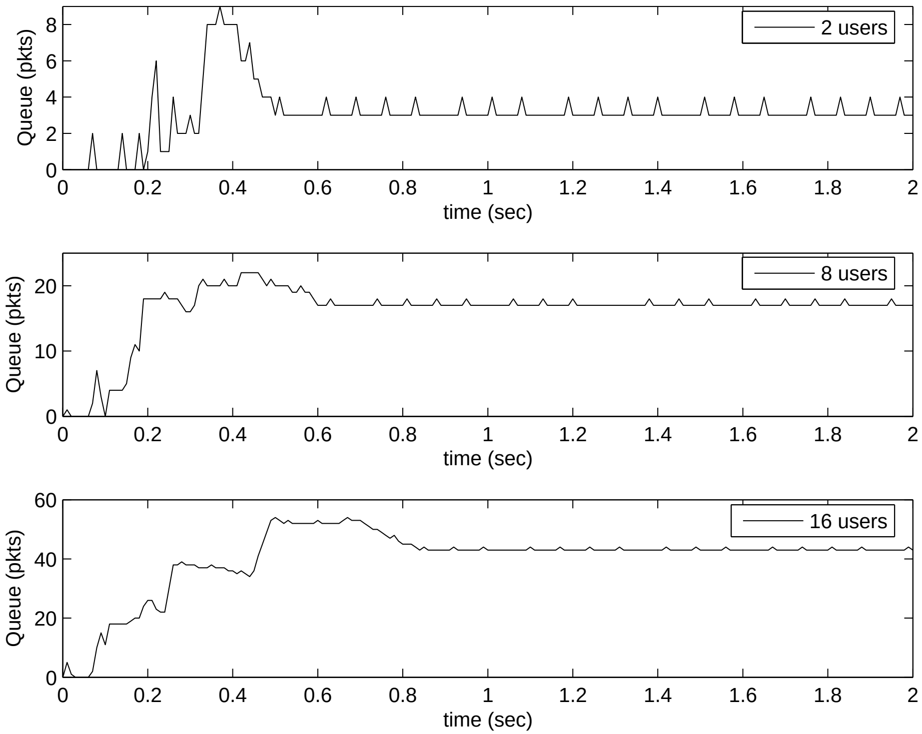
$$\alpha \leq \frac{\tau}{T_{min}} r_{obs} \leq \beta \quad (9)$$

For n flows:

$$n\alpha \leq \frac{\tau}{T_{min}} \sum r_{obs} \leq n\beta \quad (10)$$

Hamilton Institute

Vegas Latency Scaling



Simulated Queue Occupancy for 2, 8 and 16 vegas flows.

Bandwidth=5Mbps, delay=30ms, queue=100 packets

Hamilton Institute

Vegas

Vegas has a number of problems:

- Accurate *baseRTT* is critical to Vegas or it will underestimate the window size.
- Router queue occupancy scales with the number of flows, ie. Vegas doesn't maintain low delay, it only uses the delay as a signal.
- Vegas responds to any delay, whether or not it is the cause.

Even in ordinary network environments, with enough Vegas flows, persistent queueing occurs and queues can overflow.

Hamilton Institute

Delay-Based IMD

n lternative pproach

Hamilton Institute

Delay-based AIMD

Four main components:

1. Extra congestion event to react to queueing delay.
2. Modified β to drain queues
3. Modified α to improve congestion recovery
4. Experimental solutions for coexistence with loss-based flows.

Hamilton Institute

Extra congestion event

The congestion window is updated similarly to Reno, except to add an extra congestion event where $Cwnd > w_0$, a minimum window size:

$$Cwnd \leftarrow \begin{cases} Cwnd * \beta & loss \\ Cwnd * \beta & \tau > \tau_0 \\ Cwnd + \frac{\alpha}{Cwnd} & ACK \end{cases} \quad (11)$$

τ and τ_0 are the observed and threshold queueing delay.

Hamilton Institute

Draining the Queue

β, the backoff factor, is designed so as to empty the queue at every congestion event.

$$\beta = \delta \frac{RTT_{min}}{RTT(t)} \tag{12}$$

In practice, an extra factor $\delta \sim 0.9$ is added to ensure RTT_{min} is regularly seen.

Hamilton Institute

Recovering Quickly

The increase function, α is quadratic in time since the last back off, as in H-TCP and is balanced against β to maintain fairness.

$$q = min[1, 1 + 10(\delta - 1) + 0.5(\delta - 1)^2] \quad (13)$$

$$\alpha = 2(1 - \beta)q \quad (14)$$

Hamilton Institute

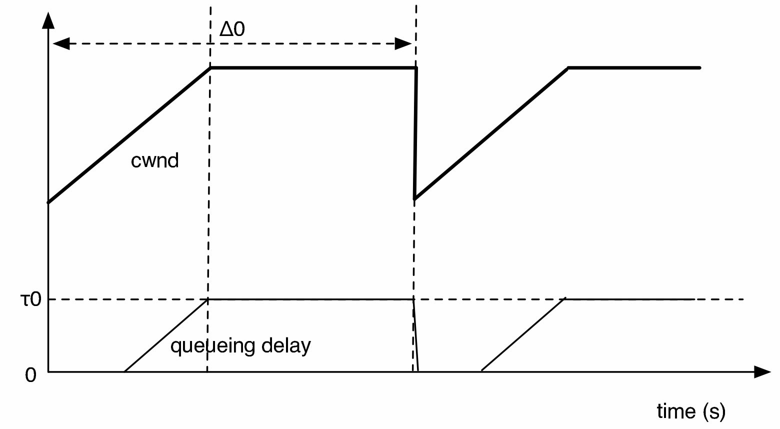
Coexistence with loss-based flows

Two experimental approaches thus far:

- Sliding delay threshold, τ_0
- Probabilistic losses at network endpoints

Hamilton Institute

Coexistence with loss-based flows



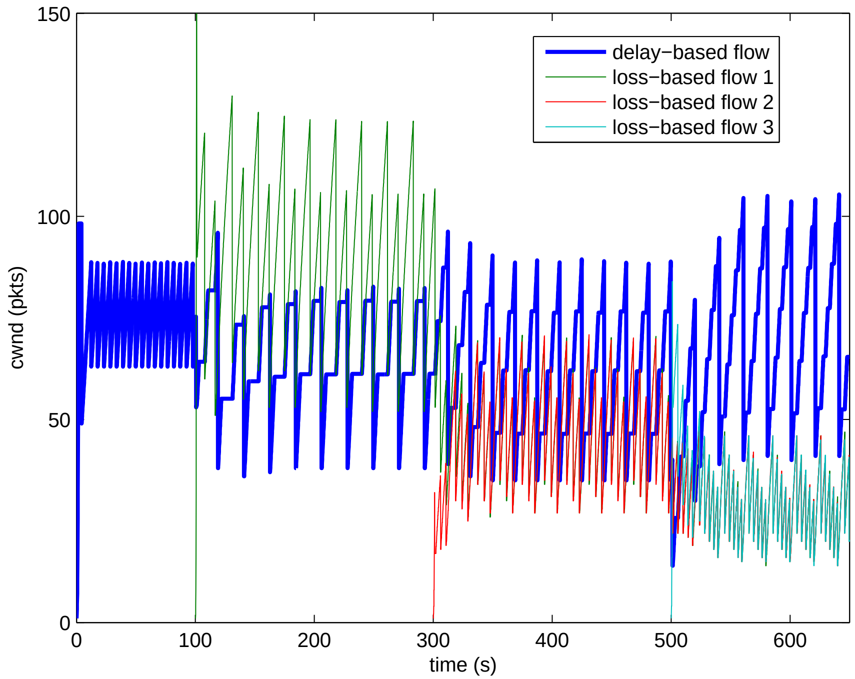
- Sliding delay threshold ($0 < \gamma < 1$)

$$\tau_0 = (1 - \gamma)\tau_0 + \gamma(RTT_{max} - RTT_{min}) \qquad (15)$$

- When $\tau > \tau_0$, stop increasing but delay back off until time since last back off $> \Delta_0$.

Hamilton Institute

Coexistence with loss-based flows

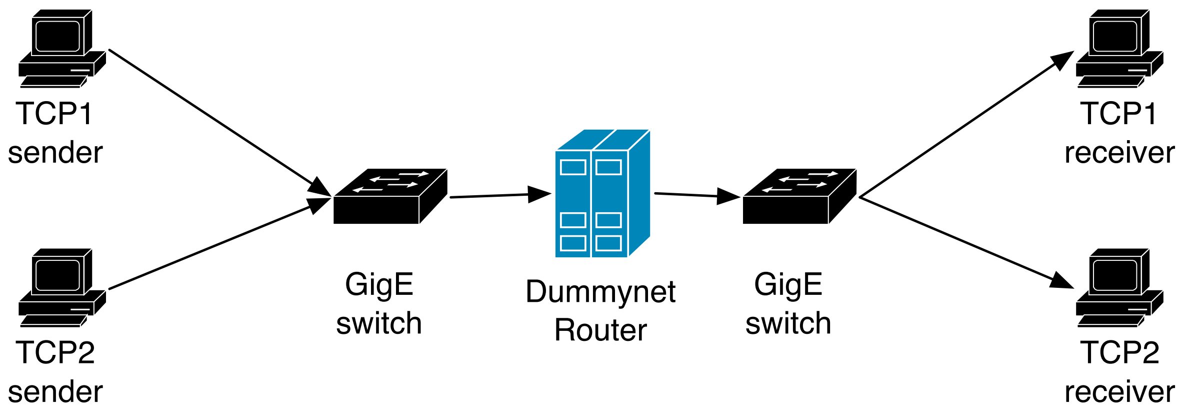


Coexistence of 1 delay and 3 loss-based flows (ns simulation, propagation delay 100ms, bandwidth 10Mbps, 100 packet queue).

Hamilton Institute

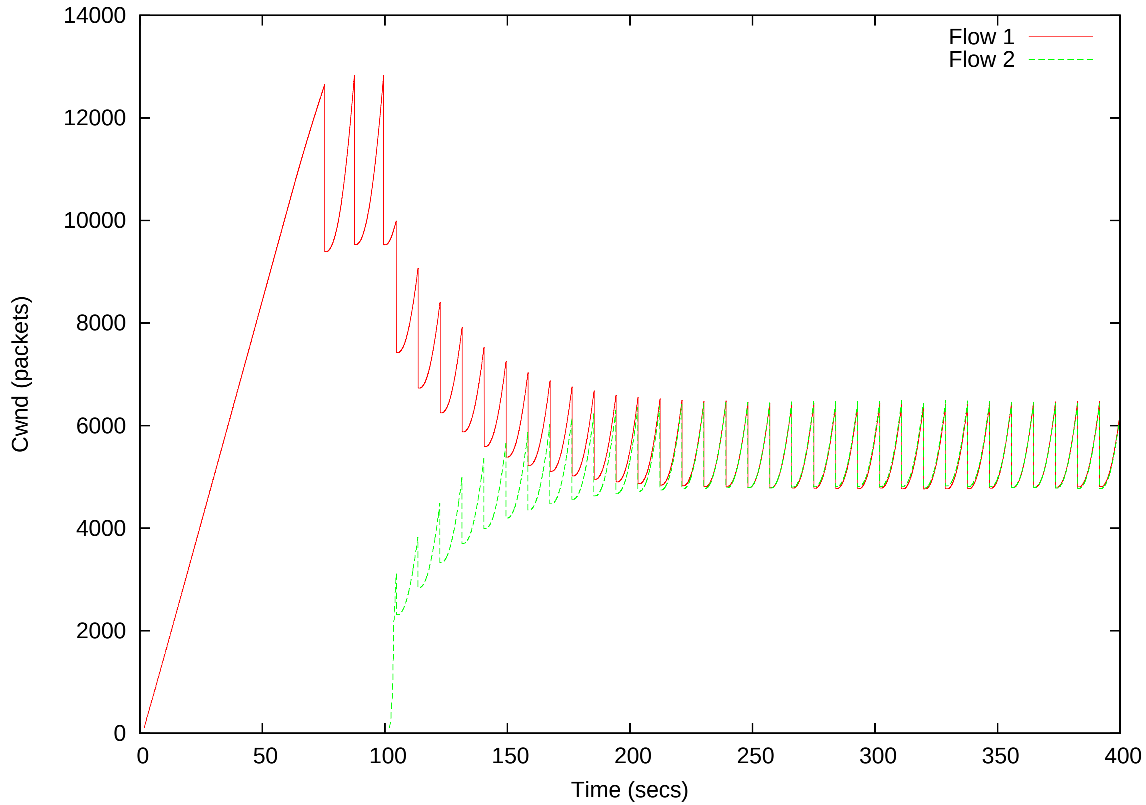
Results

- Delay-based in isolation
- Results from tests on dumbell topology testbed, linux 2.6.18 source and destination hosts, freebsd dummynet router.



Hamilton Institute

High Speed Link

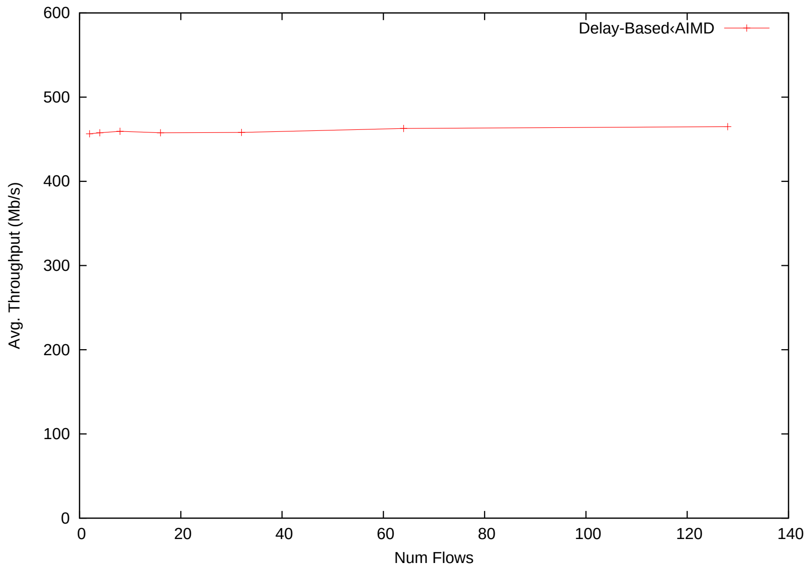


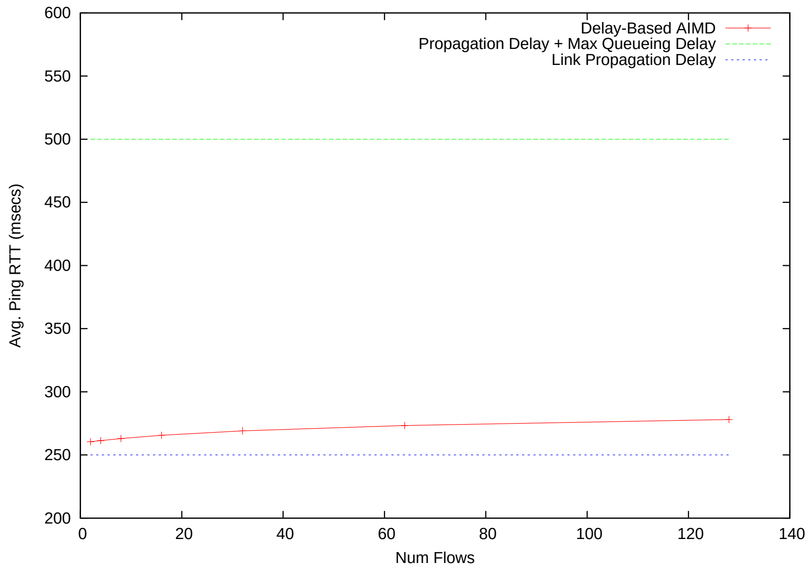
Convergence of DB-AIMD following startup of a second flow.

500Mbps link rate, 250ms RTT, 250ms of buffering.

Hamilton Institute

High Speed Link

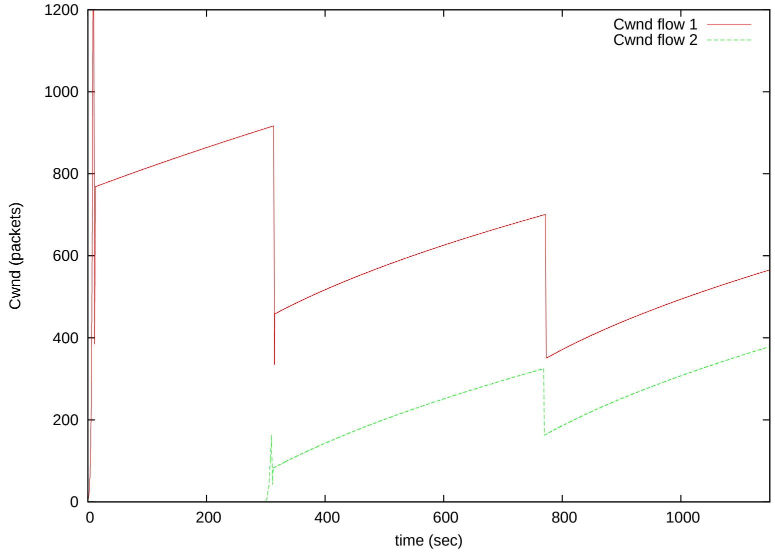


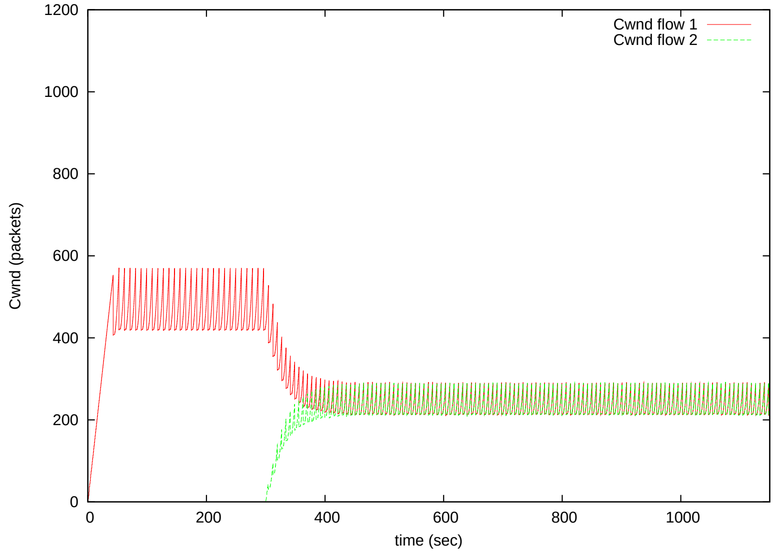


Link utilisation and Delay vs number of flows with DB-AIMD (using τ_0=50ms). 500Mbps link rate, 250ms RTT, bandwidth-delay product of buffering.

Hamilton Institute

GeoSat Link

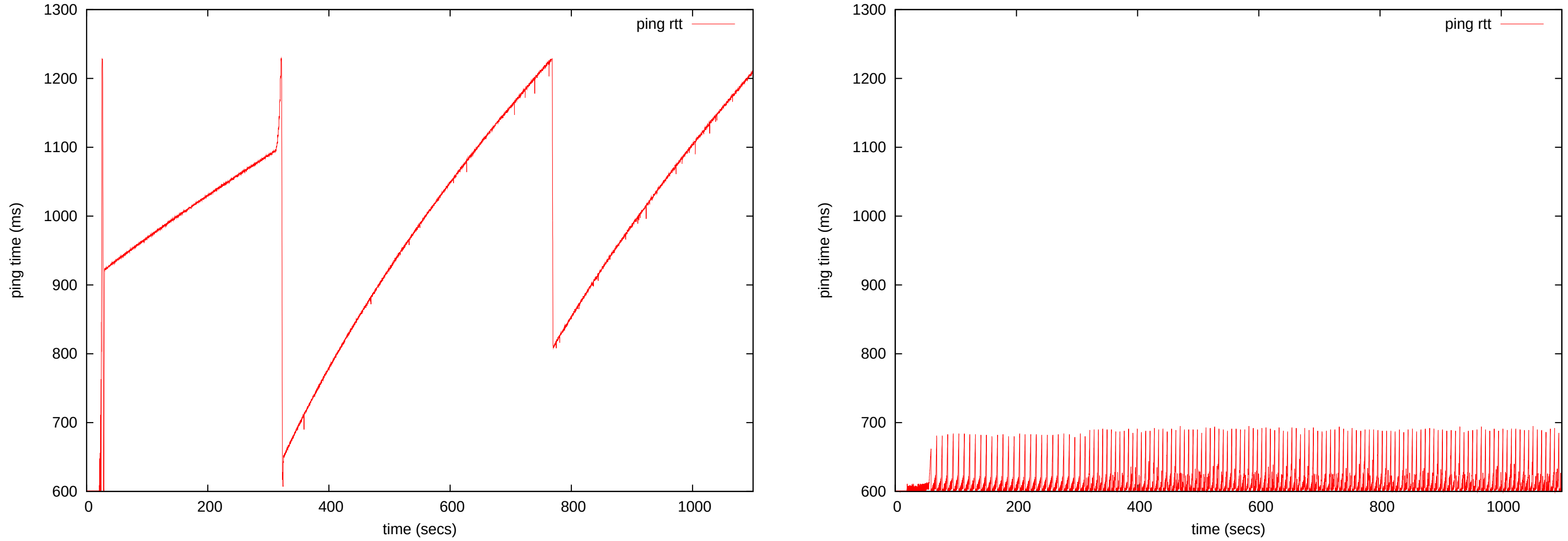




Congestion windows following startup of a second flow. Reno (left) and DB-AIMD (right). Note Reno's 15 min. congestion epoch duration. 10Mbps bandwidth, 600ms propogation delay, 600ms queue

Hamilton Institute

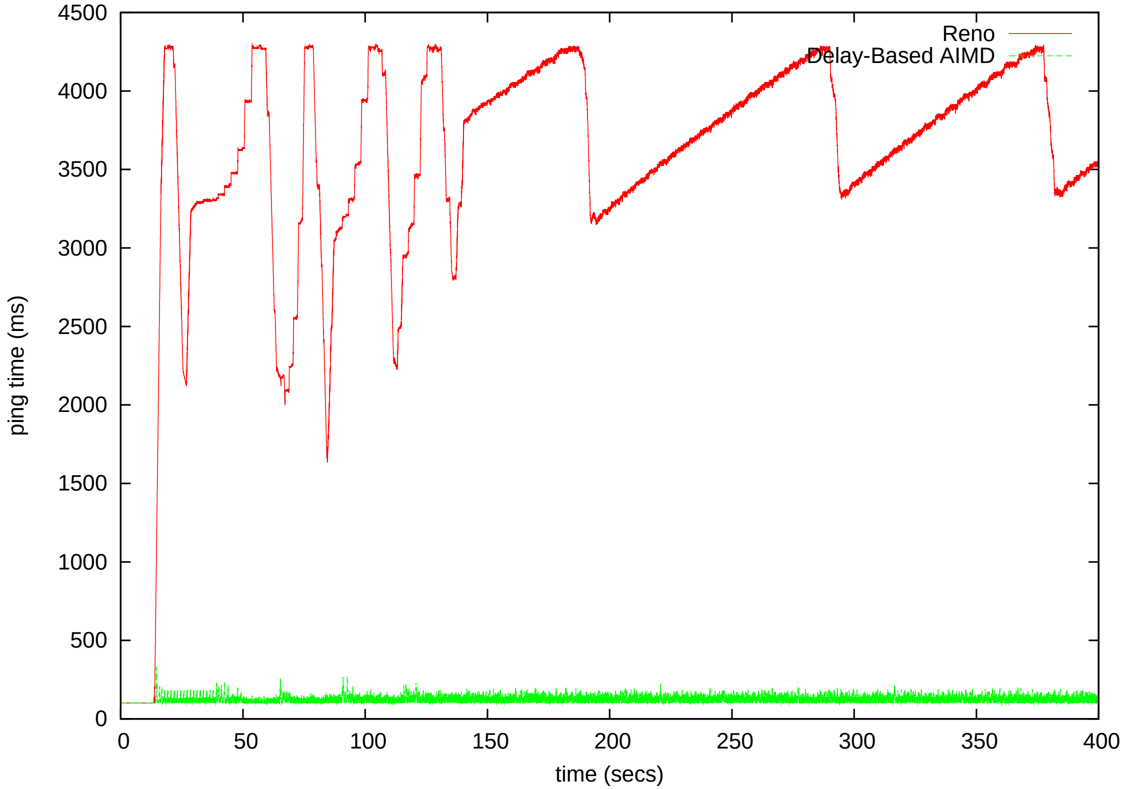
GeoSat Link



Ping times following startup of a second flow with Reno (left) and DB-AIMD (right). 10Mbps bandwidth, 600ms propagation delay, 600ms queue.

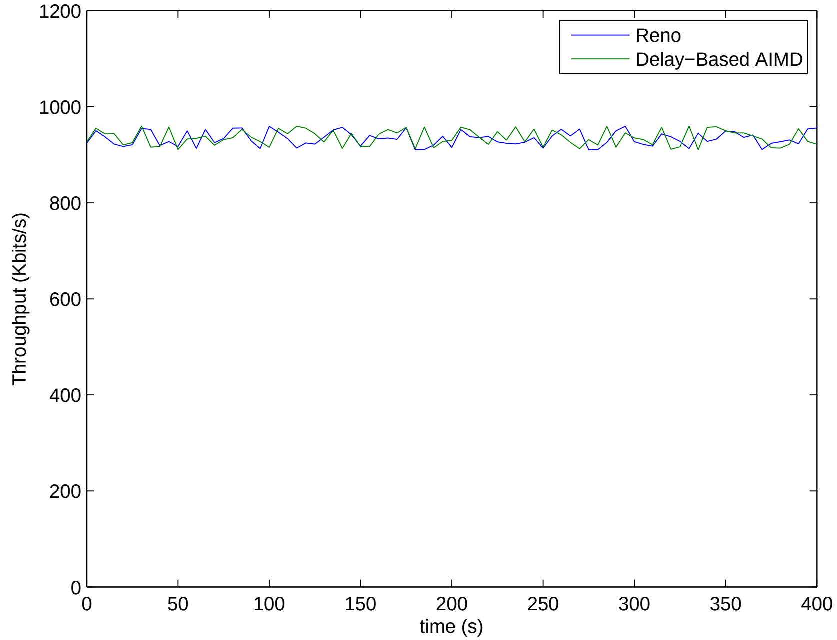
Hamilton Institute

DSL Link



Ping times for a link with 1Mbps bandwidth, 100ms propagation delay, 512KB queue. Staggered flows start every 30 secs

DSL Link



Total throughput of multiple staggered flows with Reno and DB-AIMD. 1Mbps bandwidth, 100ms propagation delay, 512KB queue.

Hamilton Institute

Review of Design Objectives

- At each congestion event, drain the queue completely
  . . . yes
- Maintain low queueing delay throughout operation
  . . . yes
- Quick startup of new flows
  . . . yes
- Full utilisation of large BDP links
  . . . almost
- Coexistence
  . . . feasible?

Hamilton Institute

Outstanding Issues

- Reverse-path queueing
- Several queues with competing traffic — multiple bottlenecks
- Importance of correlation between delay and congestion
- Spurious delay signals
- Route changes

Hamilton Institute

Conclusions

- Unlike previous delay-based schemes, DB-AIMD actively drains router queues, lowering delay
- Coexistence may be possible
- Vegas may have some issues but delay-based schemes should not be ignored
- Some overlap in work by: MC Weigle, K Jeffay, F Donelson Smith - Computer Communications, 2005, also Westwood, FAST

Hamilton Institute