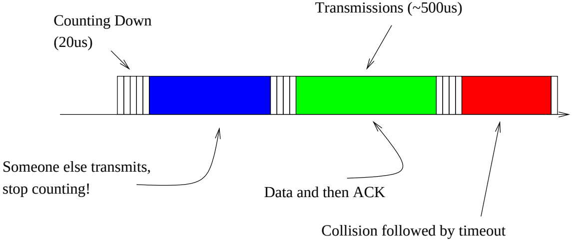
Inferring Queue State by Measuring Delay in a WiFi Network

David Malone, Douglas J Leith, Ian Dangerfield

11 May 2009

Wifi, Delay, Queue Length and Congestion Control

- RTT a proxy for queue length.
- Not too crazy in wired networks.
- For Wifi?

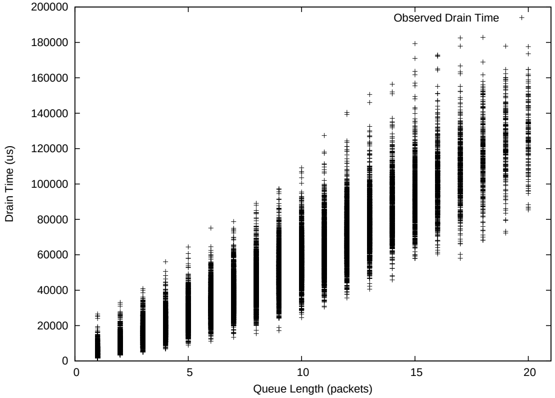


- With fixed traffic, what is impact of random service?
- What is impact of variable traffic (not even sharing buffer)?
- What will Vegas do in practice?
- Want to understand these for future design work.

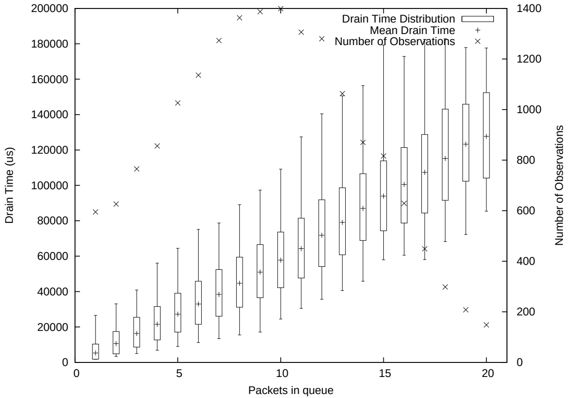
Sample Previous work

- V. Jacobson. pathchar — a tool to infer characteristics of internet paths. MSRI, April 1997.
- N Sundaram, WS Conner, and A Rangarajan. Estimation of bandwidth in bridged home networks. WiNMee, 2007.
- M Franceschinis, M Mellia, M Meo, and M Munafo. Measuring TCP over WiFi: A real case. WiNMee, April 2005.
- G. McCullagh. Exploring delay-based tcp congestion control. Masters Thesis, 2008.

Fixed Traffic: How bad is the problem?

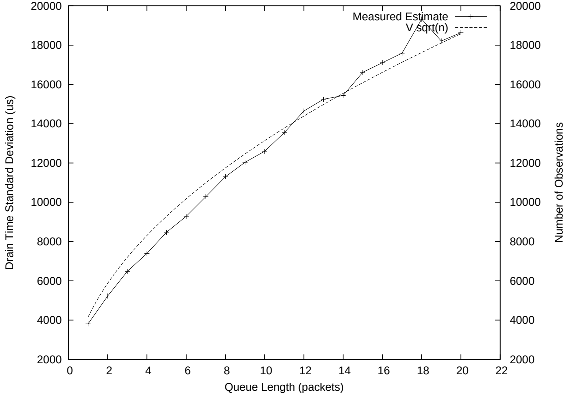


What do the stats look like?



Note: the variance is getting bigger.

How does it grow?

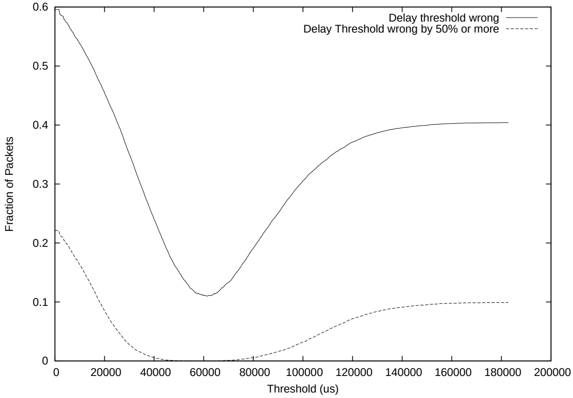


For fixed traffic service time looks uncorrelated.

Queue Length Prediction

- Suppose traffic is fixed.
- We have collect all statistics.
- Given an RTT, can we guess how full queue is?
- Easier: more or less than half full?

Results of thresholding



Using History

- Mistake 10% of time, not good for congestion control.
- Only using one sample, what happens if we use history.
- Obvious thing to do: filter.

Filters

7/8 Filter

$$\text{srtt} \leftarrow 7/8\,\text{srtt} + 1/8\,\text{rtt}.$$

Exponential Time

$$\text{srtt} \leftarrow e^{-\Delta T/T_c}\text{srtt} + (1 - e^{-\Delta T/T_c})\text{rtt}.$$

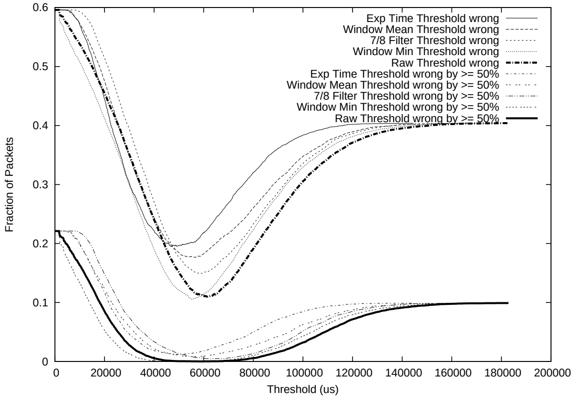
Windowed Mean

$$\text{srtt} \leftarrow \underset{\text{last RTT}}{mean}\,\text{rtt}.$$

Windowed Min

$$\text{srtt} \leftarrow \min_{\text{last RTT}}\,\text{rtt}.$$

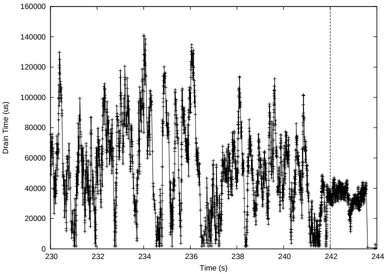
How much better do we do?

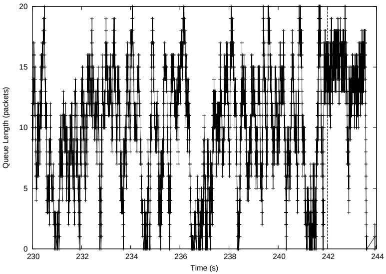


Variable Network Conditions

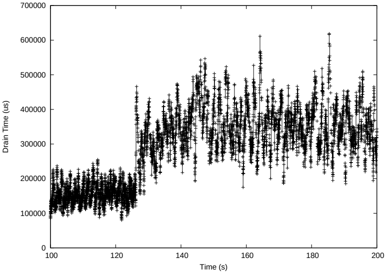
- Other traffic can change service rate.
- Need not even share same buffer.
- Nonlinear because of collisions.
- What happens when we add/remove competing stations?

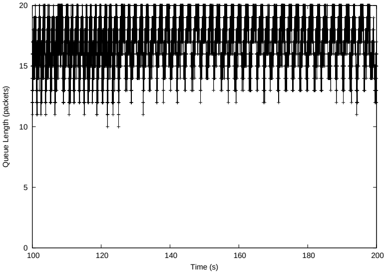
Removing stations (4 → 1)





Adding stations ($4 \rightarrow 8$, *ACK Prio*)





Even base RTT changes.

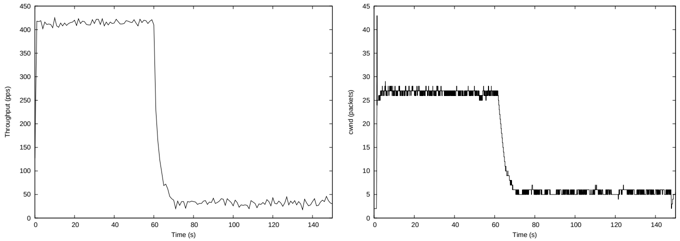
Vegas in Practice

$$\text{TargetCwnd} = \text{cwnd} \times \text{baseRTT}/\text{minRTT}$$

Make decision based on $\text{TargetCwnd} - \text{cwnd}$.

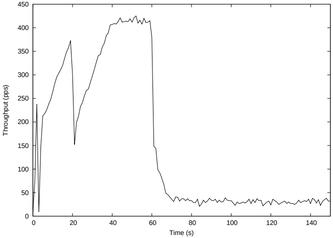
- Will Vegas make right decisions based on current RTT?
- Will Vegas get correct base RTT?
- Vary delay with dummynet.
- Vary BW by adding competing stations.

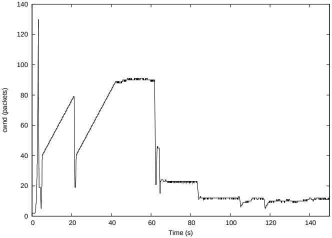
Vegas with 5ms RTT



Lower bound like $1 - \alpha/\text{cwnd}$.

Vegas with 200ms RTT





Sees loss and goes into Reno mode.

Conclusion

- With fixed traffic, delay is quite variable.
- Variability grows with buffer occupancy like $\sqrt{n}$.
- Obvious filters make things worse.
- Need to deal with change in traffic conditions.
- Linux Vegas does OK.
- Switch to Reno helps.
- Vegas insensitive at smaller buffer sizes.
- Variability at larger buffer sizes still a problem.