A Testbed Study on IW10 vs IW3

draft-ietf-tcpm-initcwnd-00.txt

H.K. Jerry Chu - hkchu@google.com
Yaogong Wang - ywang15@ncsu.edu

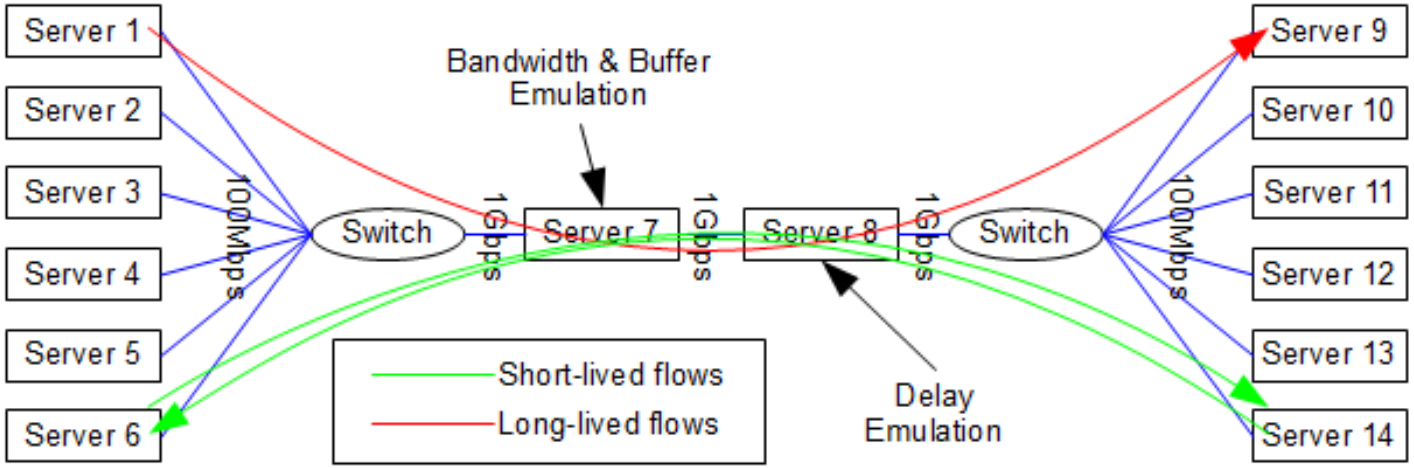
Questions/Concerns

- How does IW10 affect competing sessions, or cross traffic with long lived connections?
- How does IW10 perform
  - on slow, e.g., 56Kbps or 64Kbps links?
  - if browsers opening 4-6 simultaneous connections? (previous testbed used single conn Poisson arrival)
  - when SACK is either not available, or not adequately implemented?

Testbed Setup

- Two independent testbeds
  - one at Google
  - the other at Prof. Injong Rhee's lab in NCSU
- Dumbbell topology with Ethernet switches and Linux boxes as endhosts and routers
- 2.6.26 & 2.6.35 kernels (2.6.26 patched to increase initial advertised window)
- Some differences in test results to be investigated

Testbed at NCSU



- Netem on the router to inject equal delay on both directions
- HTB qdisc to limit bottleneck bandwidth and max qlen
- Details at http://research.csc.ncsu.edu/netsrv/?q=content/iw10

Linux Kernel Bugs (or Features?)

- Packet with FIN bit set is sent regardless of cwnd
  - IW10 is really IW11
- Many factors affecting send side buffer mgmt (TSO, tcp_wmem, socket write size, skb splitting…) may limit the initial burst size < IW
  - E.g., IW10, TSO off, default tcp_wmem only emits 9 pkts in the first burst
- Certain RPC sizes always trigger RST at the end

Tools

- To configure IW (iproute2):
  - ip route change default via <gw> dev eth? initcwnd <IW>
- ‘initrwnd’ has been added to the lastest iproute2
- Netperf: emulates web/HTTP transactions
- Iperf: emulates long-lived bulk transfer traffic

Test Parameters

- Requests arrival rate (Poisson)
- Single vs simultaneous opens (batch arrival)
- Bottleneck link bandwidth: 64Kbps or 20Mbps
- Max router qlen (packets): 40 for 64Kbps link, 500 for 20Mbps link
- RTT: 300ms
- IW3 vs IW10 vs IW3+IW10 (50/50 mixed)

Test Parameters (cont')

- Req size: 200B
- Resp size: 15KB
  - the largest burst size allowed by IW10, chosen to get a worst case measurement
- SACK: on or off (sysctl_tcp_sack)

Performance Metrics

- UCT – user completion time (measured from the client side) per netperf transaction
- Throughput – background, long-lived flows
- PLR – packet loss rate measured at the router
- Link utilization - % of time when qlen > 0 (sample every sec.)
- Others – queue occupancy graphs, cwnd/ssthresh graphs

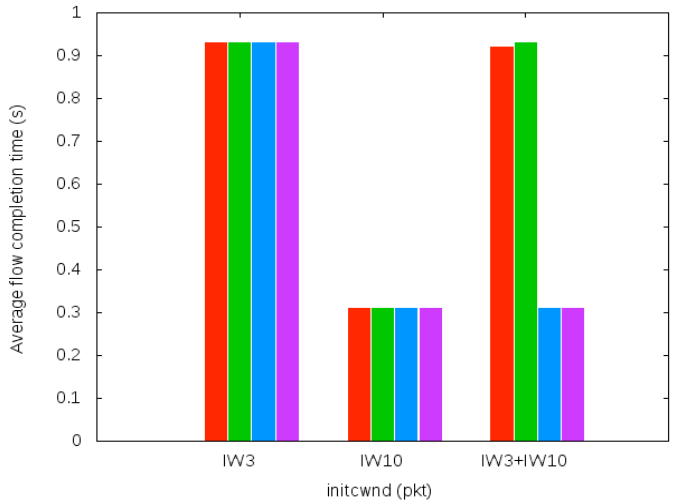
Key Findings

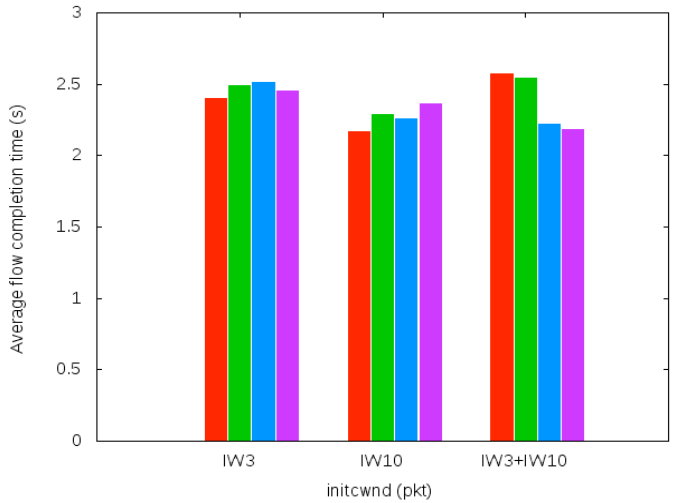
- IW10 tends to cause higher PLR than IW3, and in case of extreme load, a lot higher PLR
- Regardless of PLR, IW10 always seemed to improve, or at least not hurt UCT
- No serious fairness problem was detected
- Long lived flows seemed to perform equally or better under IW10 than IW3
- SACK is not required for IW10 to perform

Slow link tests (64Kbps, 40 buffers)

- Very long RTT (upto many secs)
  - long serialization delay (~2secs for 15KB)
  - queuing delay upto 7.5secs (40 packets)
- Five-simultaneous-open tests the worst case
  - Five IW10 bursts ALWAYS overflow router buffer
  - Five IW3 bursts never overflow router buffer (unless colliding with two other Poisson arriving bursts)
- Numbers unstable

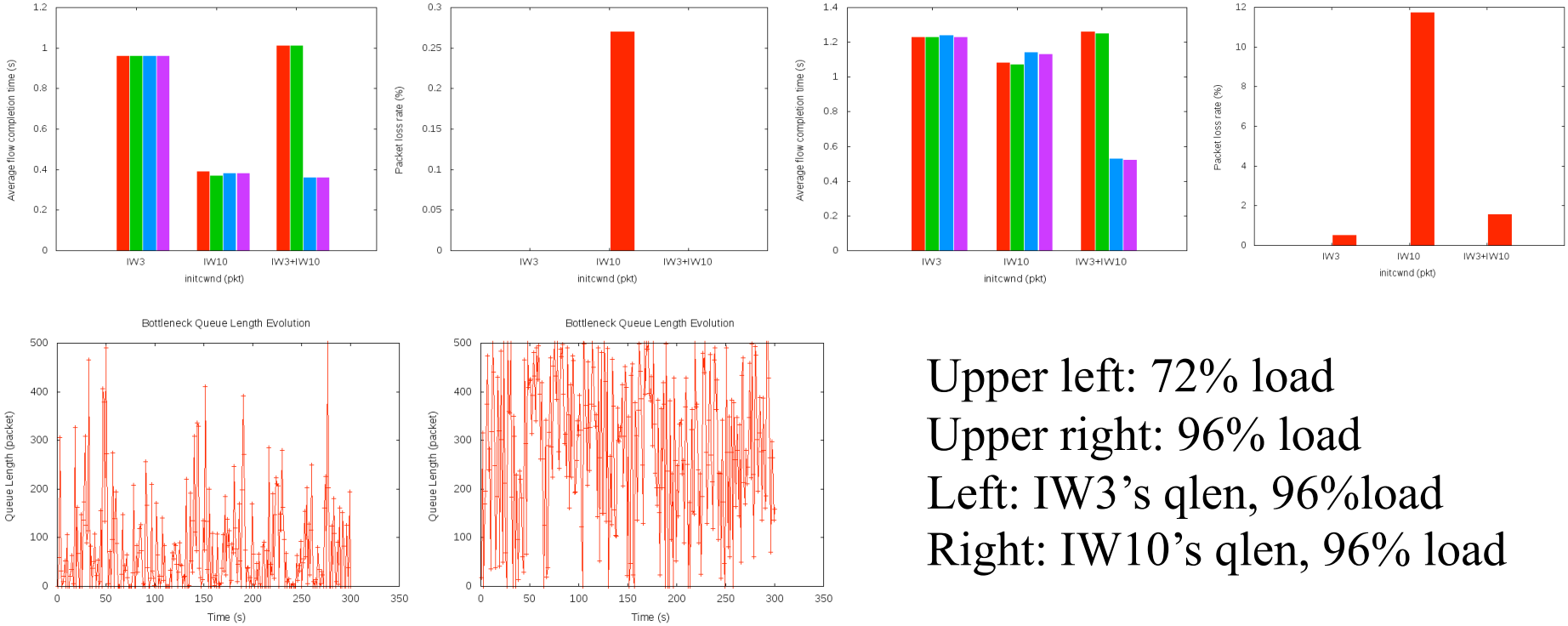
Average User Completion Time





- Different bars represent four client-server pairs in the testbed
  - see http://research.csc.ncsu.edu/netsrv/?q=content/iw10
- Faster links (left): UCT is dominated by RTTs
  - IW3/3RTTs vs IW10/1RTT, RTT=300ms
- Lightly loaded slow links (right): UCT is dominated by serialization delay
  - IW10's UCT is only slightly lower than IW3's

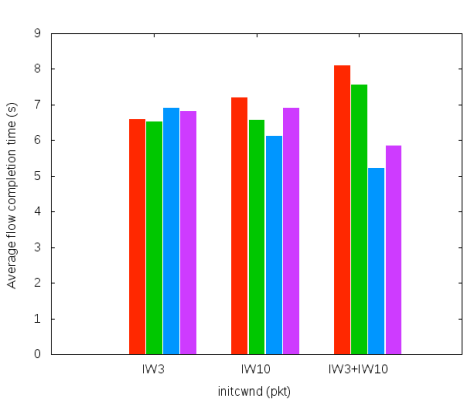
UCT Comparison (20Mbps, 5-open)

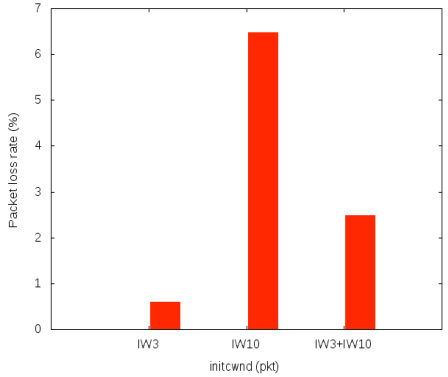


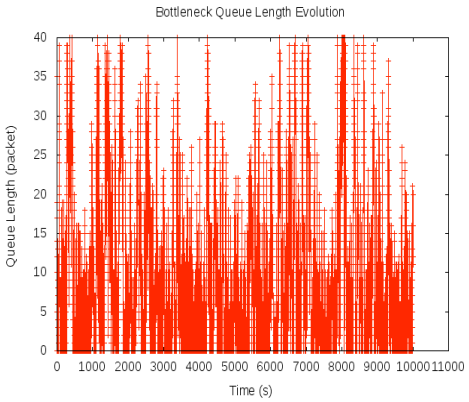
Upper left: 72% load
Upper right: 96% load
Left: IW3's qlen, 96%load
Right: IW10's qlen, 96% load

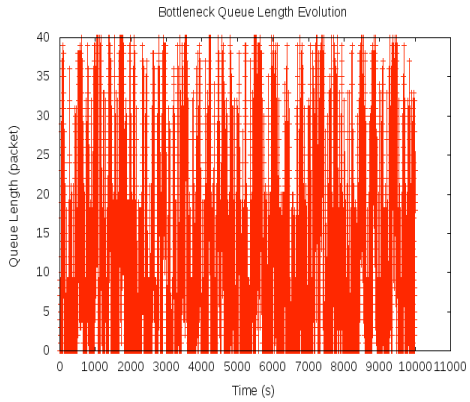
- Under heavy load IW10 lost UCT advantage due to high PLR
- IW10 UCT exhibits long tails

UCT for Higher Load Slow Links



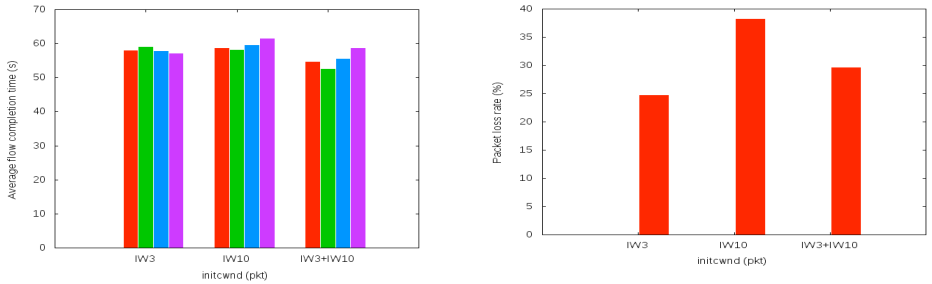




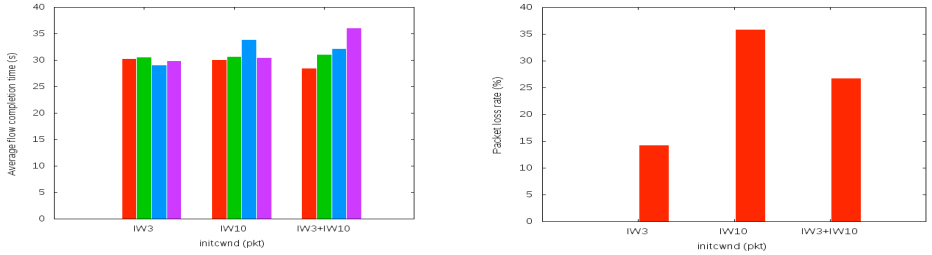


- Dominated by queuing delay at higher load (75% single open)
- IW10's longer average qlen offsets its round trip saving
  - IW3: middle right
  - IW10: right most
- IW10 performed much better when mixed with IW3 because they were subject to the same (smaller) queue

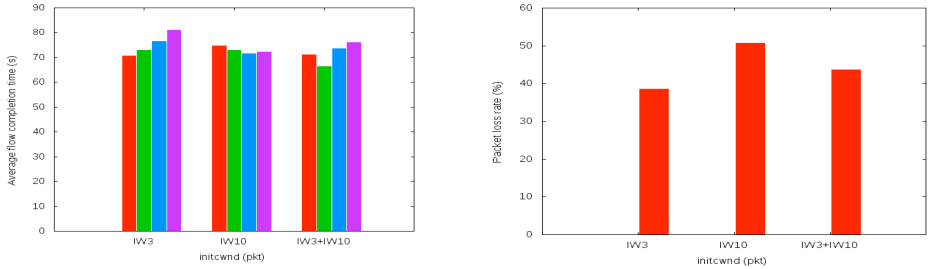
PLR comparison (64Kbps)



Single-open, 97.5% load



5 simultaneous opens, 75% load

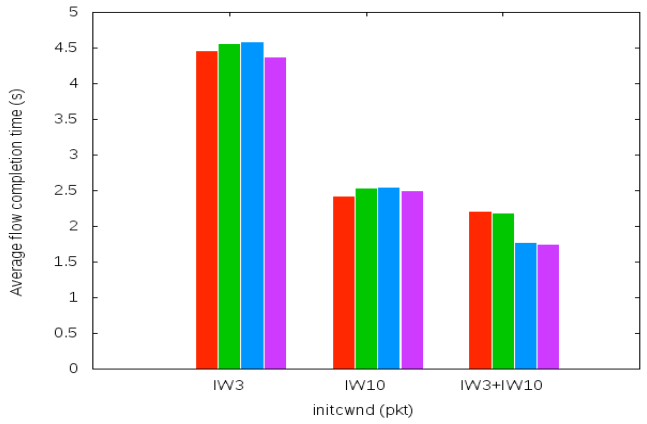


5 simultaneous opens, 97.5% load

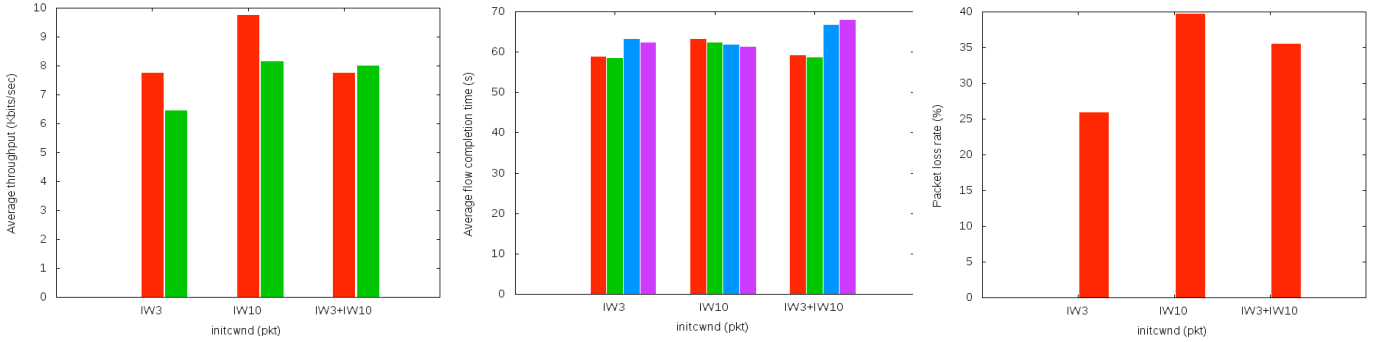
- IW10 > IW3+IW10 > IW3, simultaneous opens >> single open
- UCT > 70secs caused by multiple rounds of queuing delay

Faireness – RPC flows

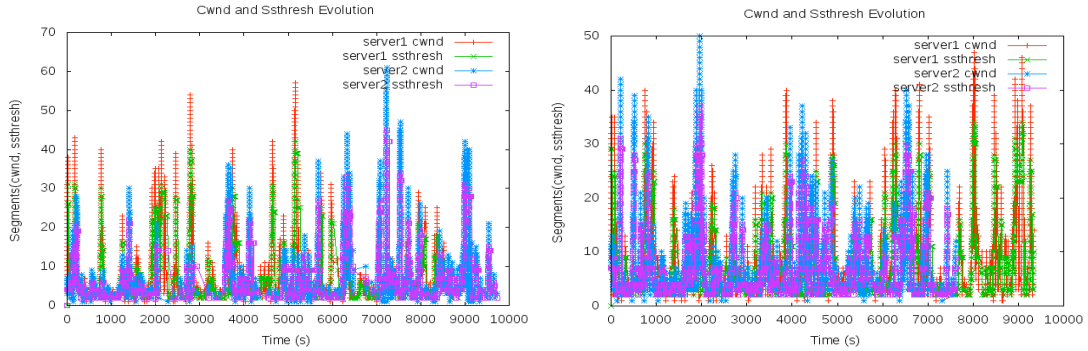
- IW3 only suffered limited performance (UCT) loss when competing against IW10 in light to median loads, but not at high load (why?)
- IW3's UCT often improved when mixed with IW10 under heavy load



IW10's Impact to Cross (long-lived) Traffic



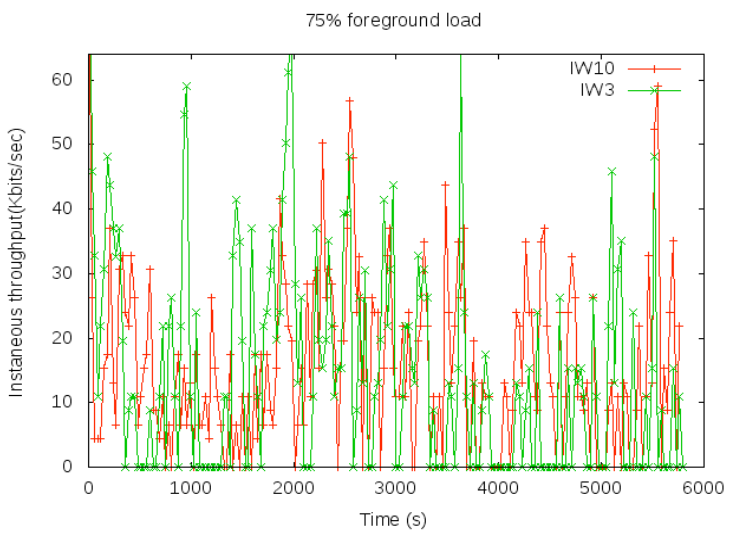
64Kbps, 75% load, simultaneous opens

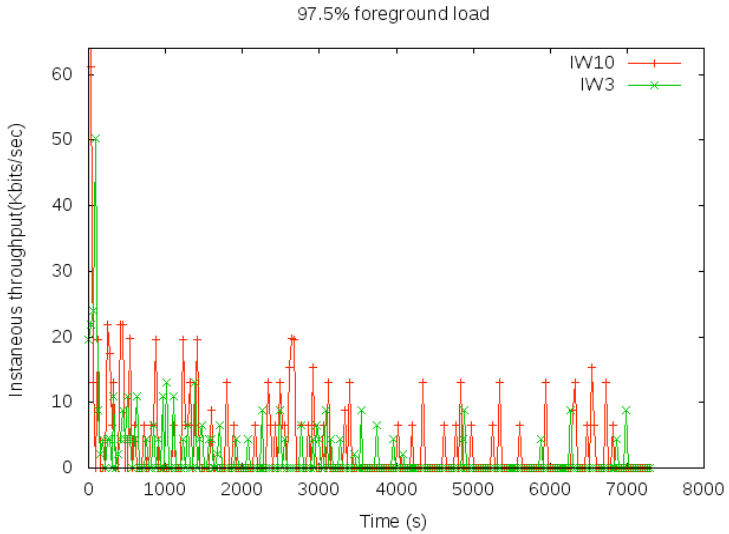


cwnd/ssthresh graphs of the two long-lived flows
left: under IW3
right: under IW10

- IW10 doesn't seem to cause any more damage to other long lived flows than IW3
- Sometimes it's the other around as shown above and next slide

Comparison of Impacts to Cross Traffic





- 64Kkpbs, 5 simultaneous RPC flows, 2 long-lived background flows
- Background flows perfomed better under IW10 than IW3 (more obvious under high load (right graph))

Is SACK required for IW10 to Perform?

- From the testbed at Google
  - SACK does help reducing UCT but only by a small percentage for both IW10 and IW3
- 24 hours, 3-way parallel experiment at Google's frontend servers
  - IW10+NewReno still beats IW3+SACK

Photos download	Avg response time	Retransmission rate
IW10+SACK	2.6secs	4.1%
IW10+NewReno	2.8secs	4.1%
IW3+SACK	3.0secs	3.3%