Transport Benchmarking

Panel Discussion

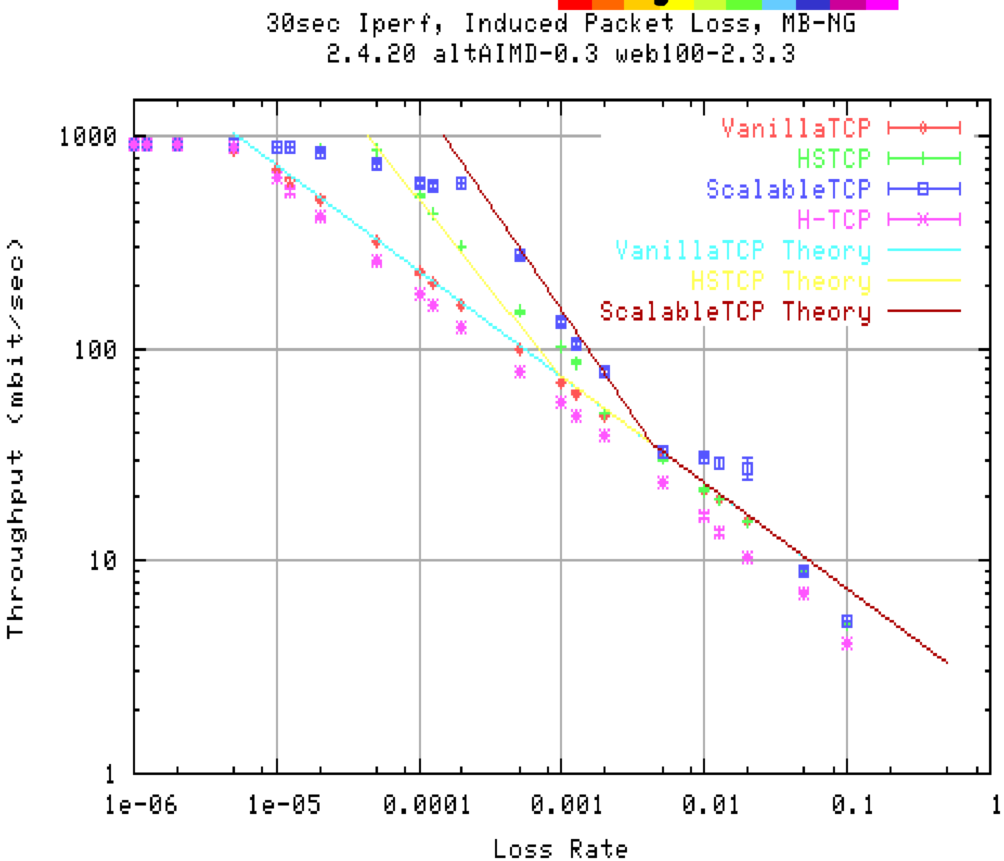
Richard Hughes-Jones
The University of Manchester

www.hep.man.ac.uk/~rich/ then "Talks"

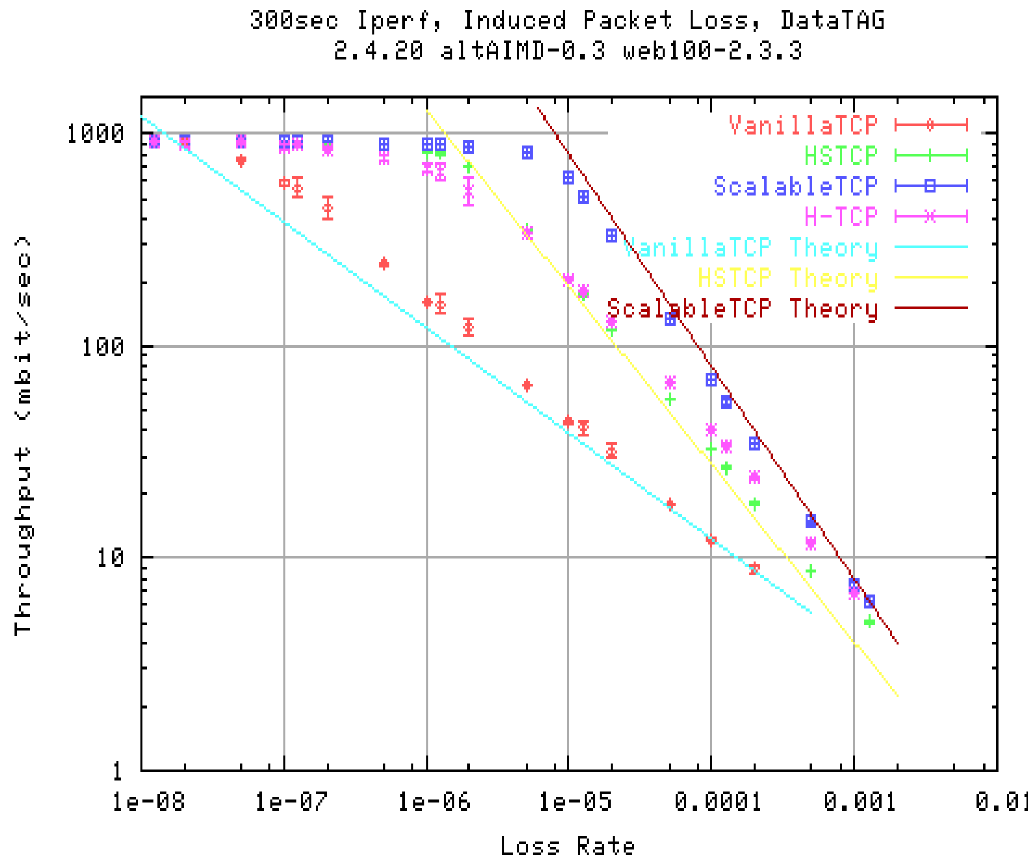
Packet Loss and new TCP Stacks

- TCP Response Function
  - Throughput vs Loss Rate – further to right: faster recovery
  - Drop packets in kernel

MB-NG rtt 6ms

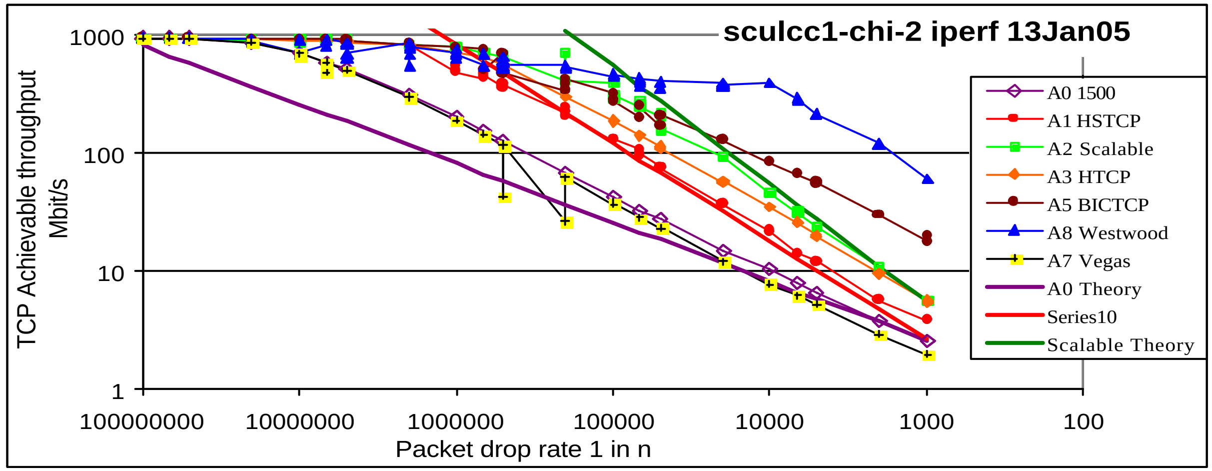


DataTAG rtt 120 ms

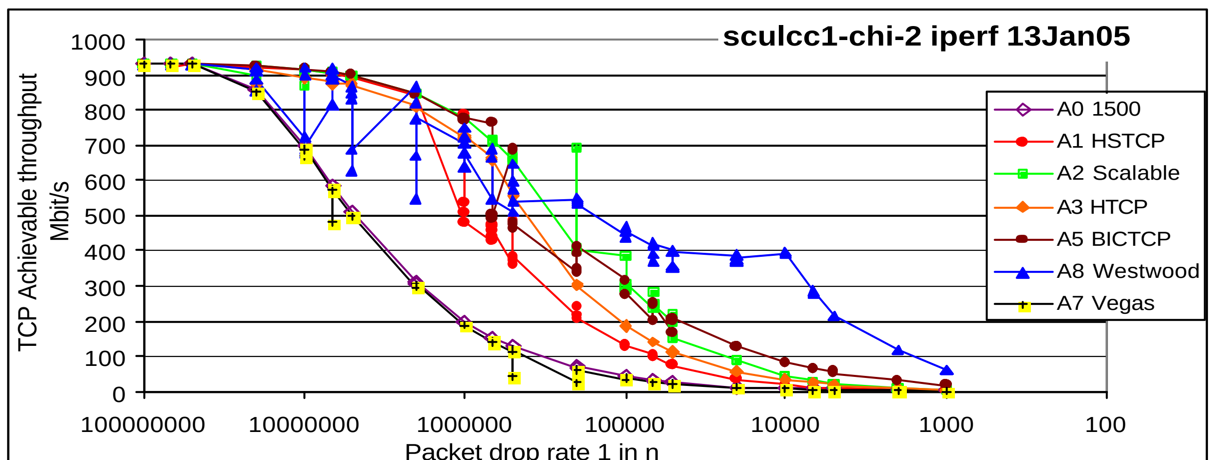


Packet Loss and new TCP Stacks

- TCP Response Function
  - UKLight London-Chicago-London rtt 177 ms
  - 2.6.6 Kernel
  - Agreement with theory good



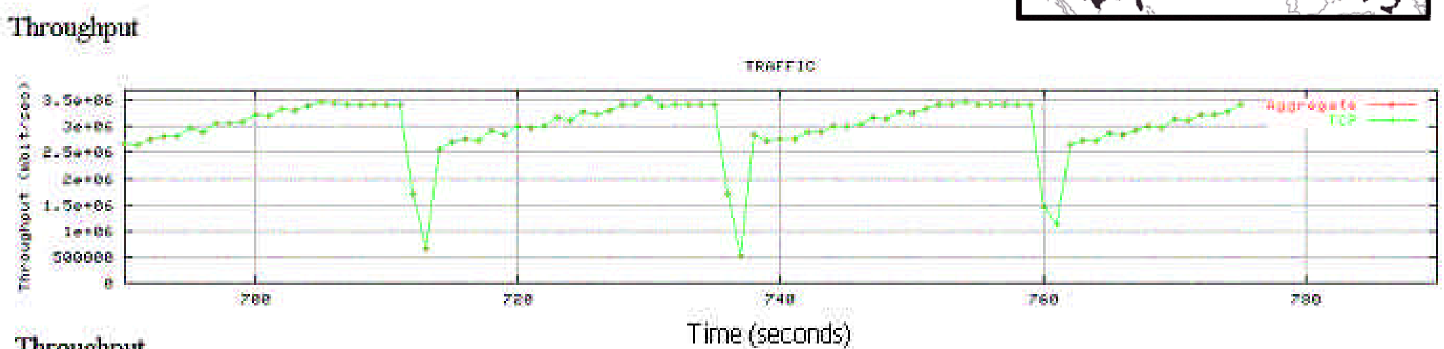
  - Some new stacks good at high loss rates



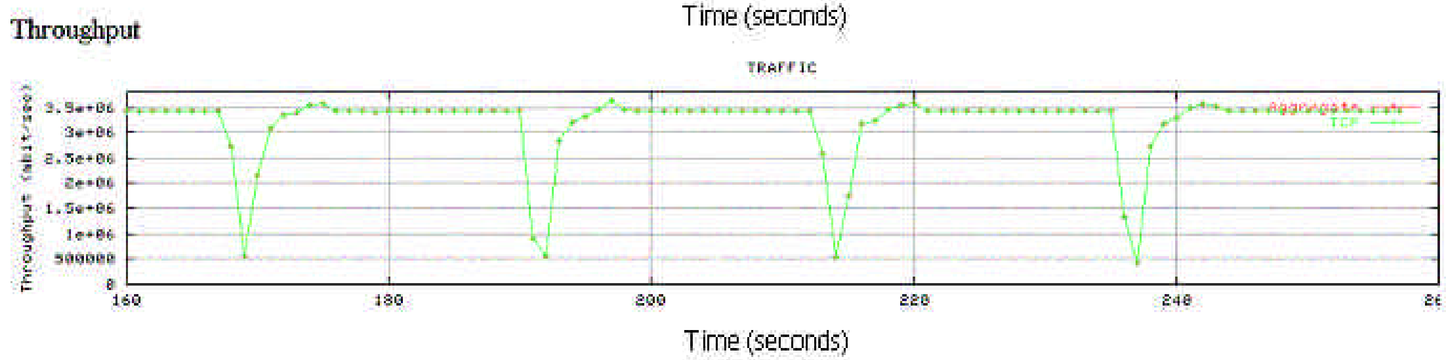
TCP Throughput – DataTAG

- Different TCP stacks tested on the DataTAG Network
- rtt 128 ms
- Drop 1 in 10^6

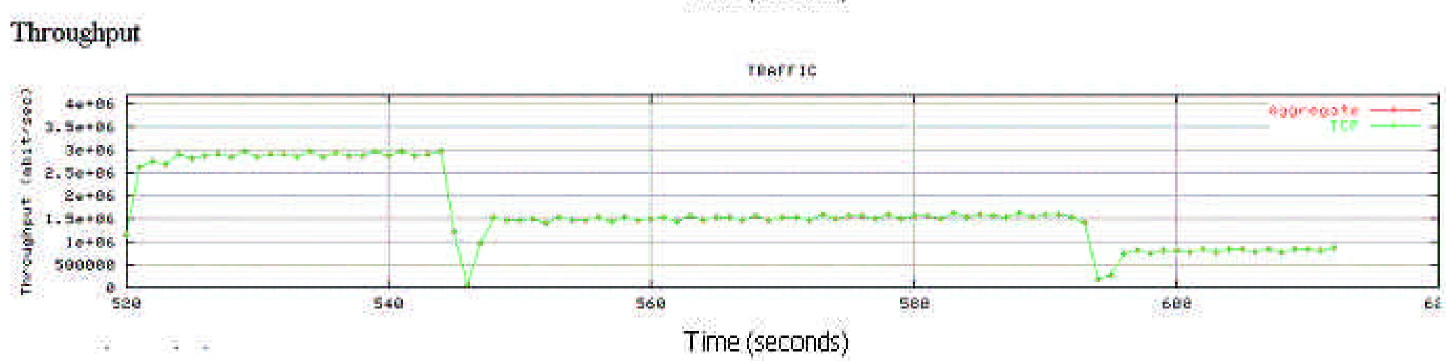
- High-Speed
  - Rapid recovery



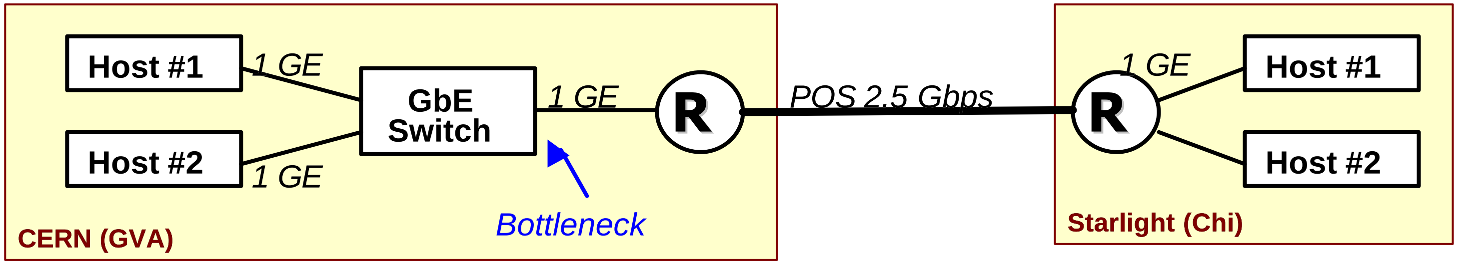
- Scalable
  - Very fast recovery



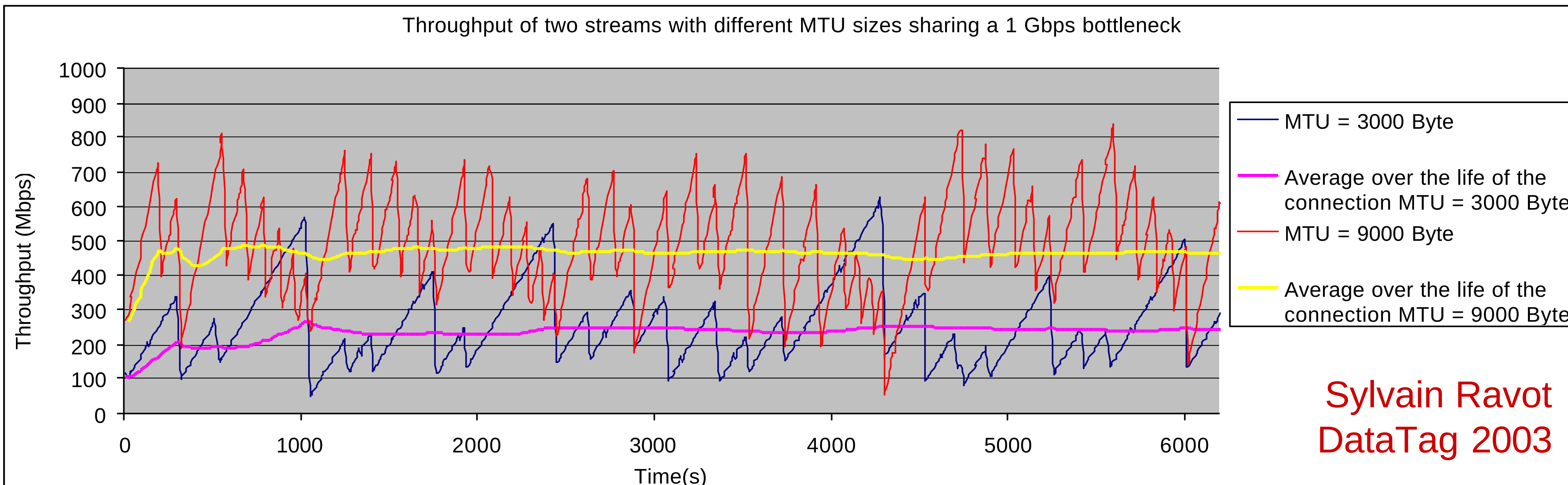
- Standard
  - Recovery would take ~ 20 mins



MTU and Fairness

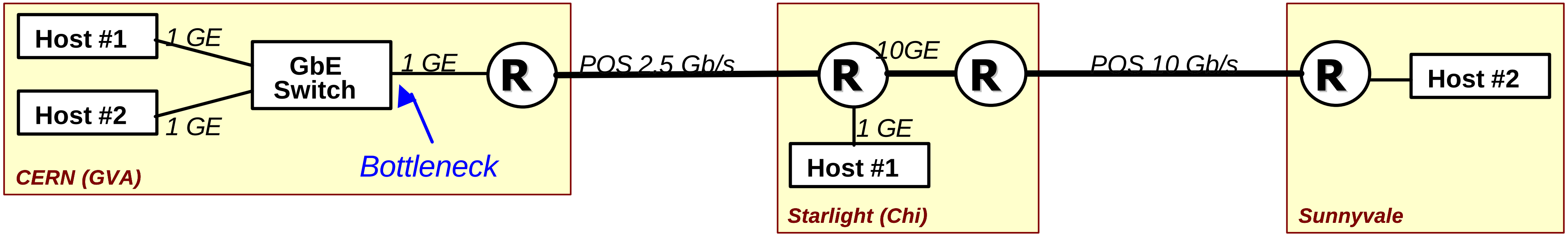


- Two TCP streams share a 1 Gb/s bottleneck
- RTT=117 ms
- MTU = 3000 Bytes ; Avg. throughput over a period of 7000s = 243 Mb/s
- MTU = 9000 Bytes; Avg. throughput over a period of 7000s = 464 Mb/s
- Link utilization : 70,7 %

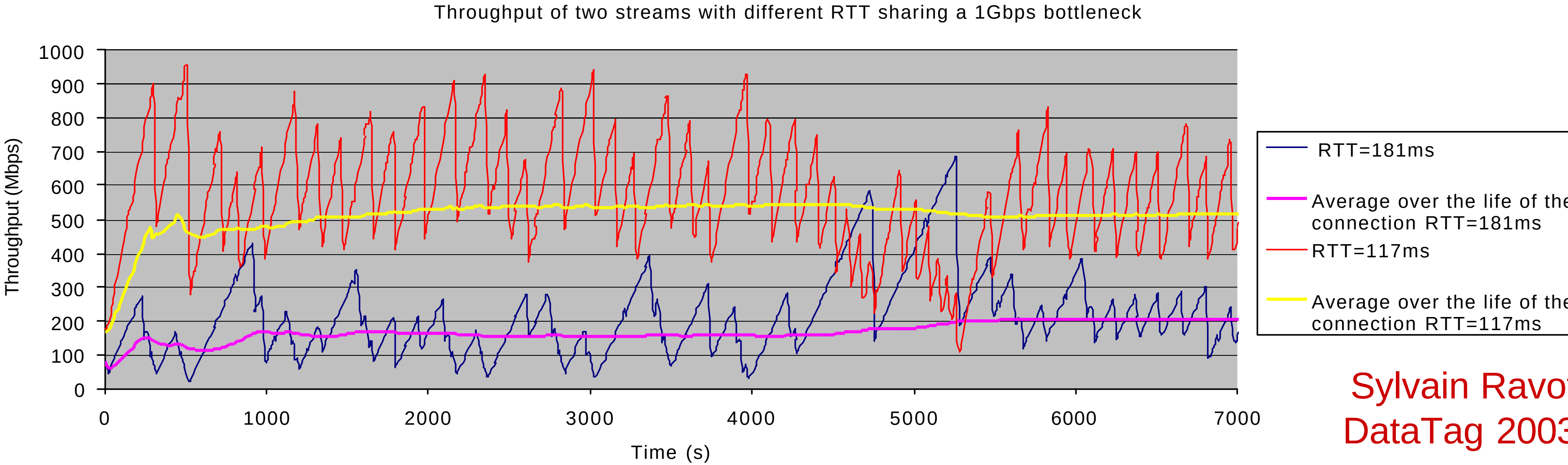


Sylvain Ravot
DataTag 2003

RTT and Fairness



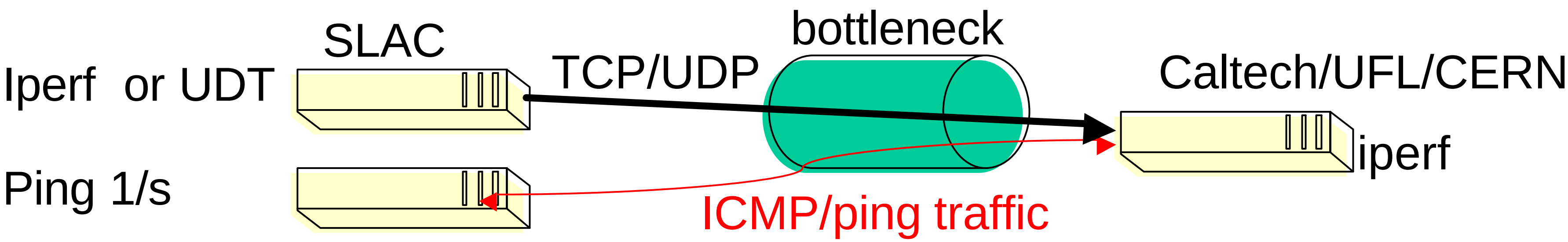
- Two TCP streams share a 1 Gb/s bottleneck
- CERN <-> Sunnyvale RTT=181ms ; Avg. throughput over a period of 7000s = 202Mb/s
- CERN <-> Starlight RTT=117ms; Avg. throughput over a period of 7000s = 514Mb/s
- MTU = 9000 bytes
- Link utilization = 71,6 %



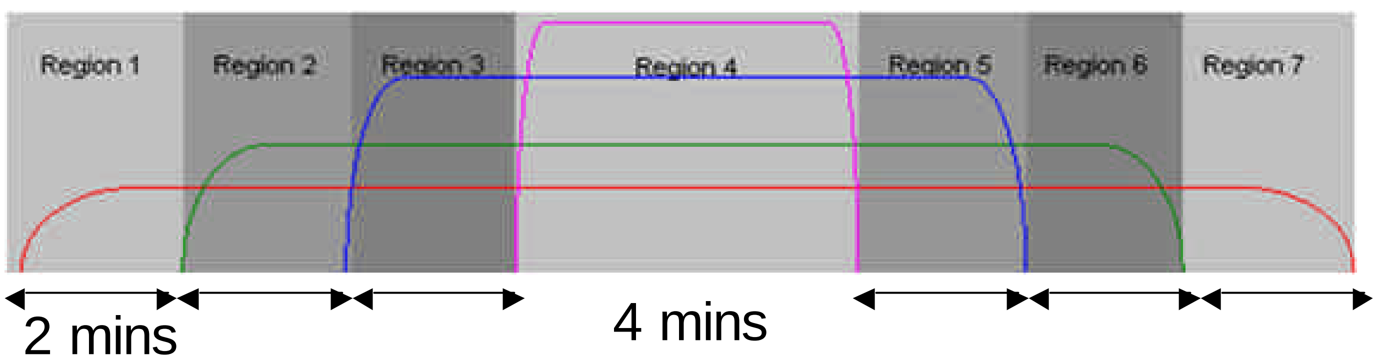
Sylvain Ravot
DataTag 2003

TCP Sharing & Recovery: Methodology (1Gbit/s)

- Chose 3 paths from SLAC (California)
  - Caltech (10ms), Univ Florida (80ms), CERN (180ms)
- Used iperf/TCP and UDT/UDP to generate traffic



- Each run was 16 minutes, in 7 regions



Les Cottrell
PFLDnet 2005

TCP Reno single stream

Les Cottrell
PFLDnet 2005

- Low performance on fast long distance paths
  - AIMD (add *a=1* pkt to *cwnd* / RTT, decrease *cwnd* by factor *b=0.5* in congestion)
  - Net effect: recovers slowly, does not effectively use available bandwidth, so poor throughput
  - Unequal sharing

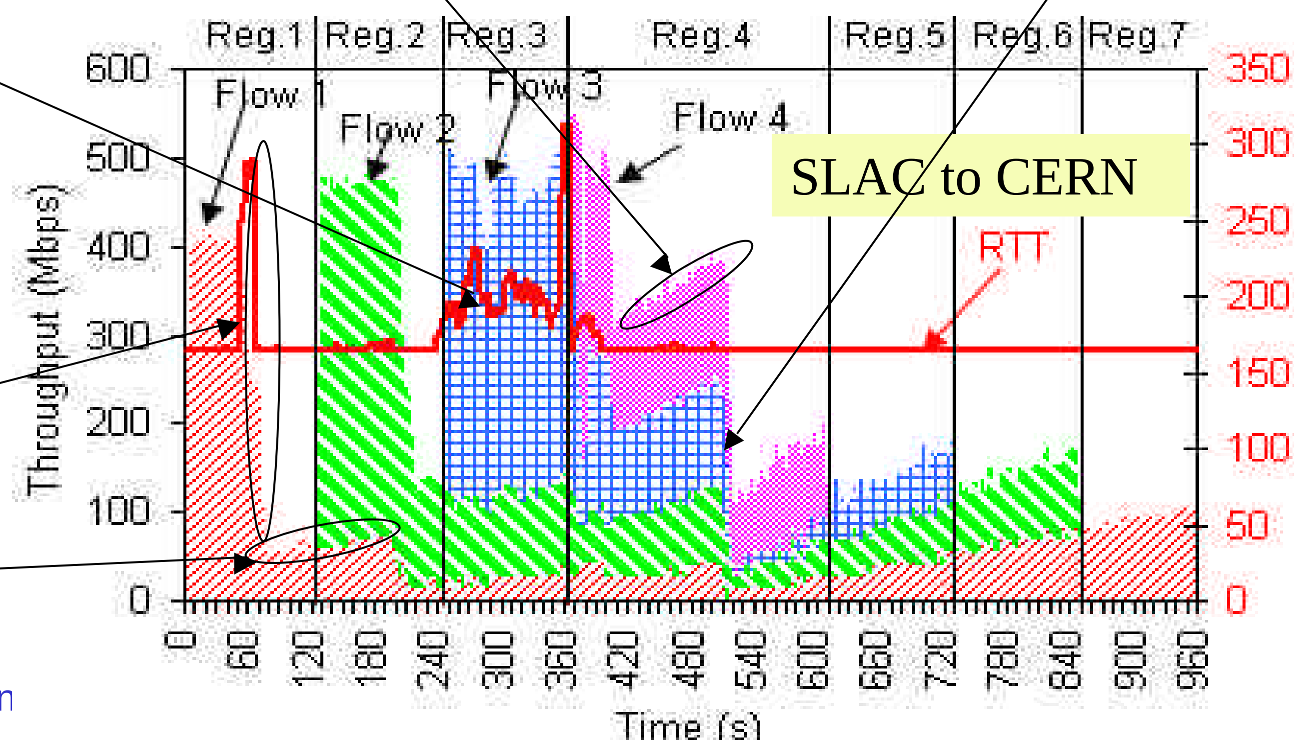
Remaining flows do not take up slack when flow removed

Increase recovery rate

RTT increases when achieves best throughput

Congestion has a dramatic effect

Recovery is slow



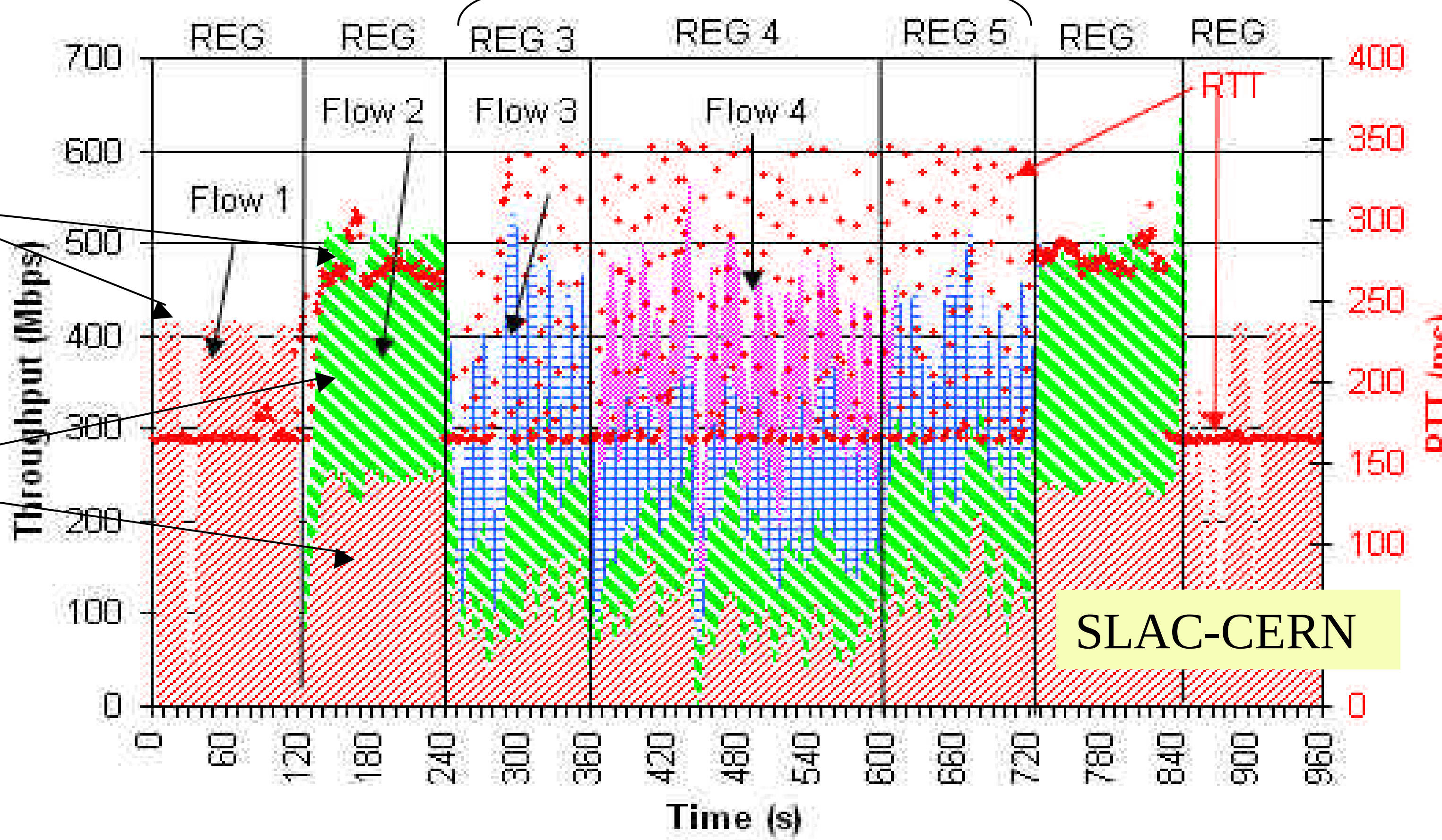
Hamilton TCP

- One of the best performers
  - Throughput is high
  - Big effects on RTT when achieves best throughput
  - Flows share equally

> 2 flows appears less stable

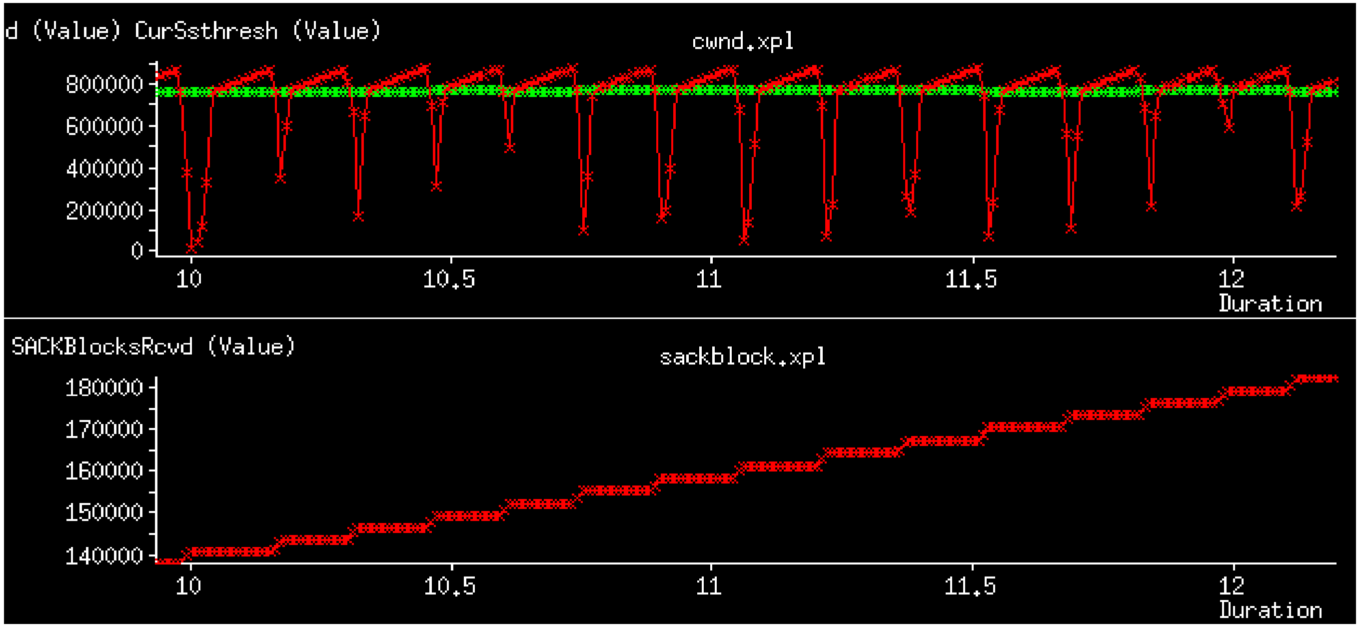
Appears to need >1 flow to achieve best throughput

Two flows share equally



Implementation problems: SACKs …

- Look into what's happening at the algorithmic level e.g. with web100:



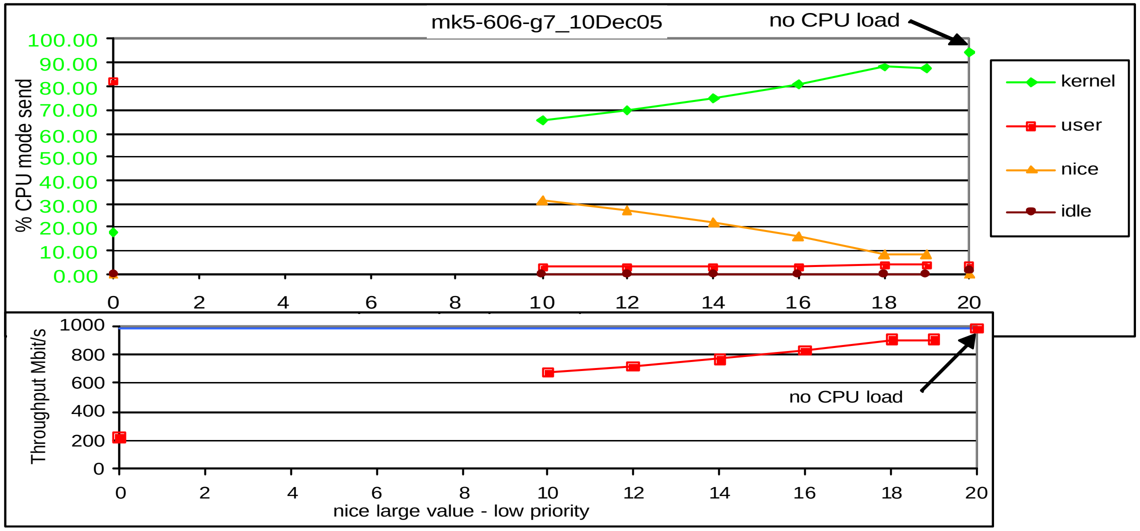
Scalable TCP on MB-NG with 200mbit/sec CBR Background

Yee-Ting Li

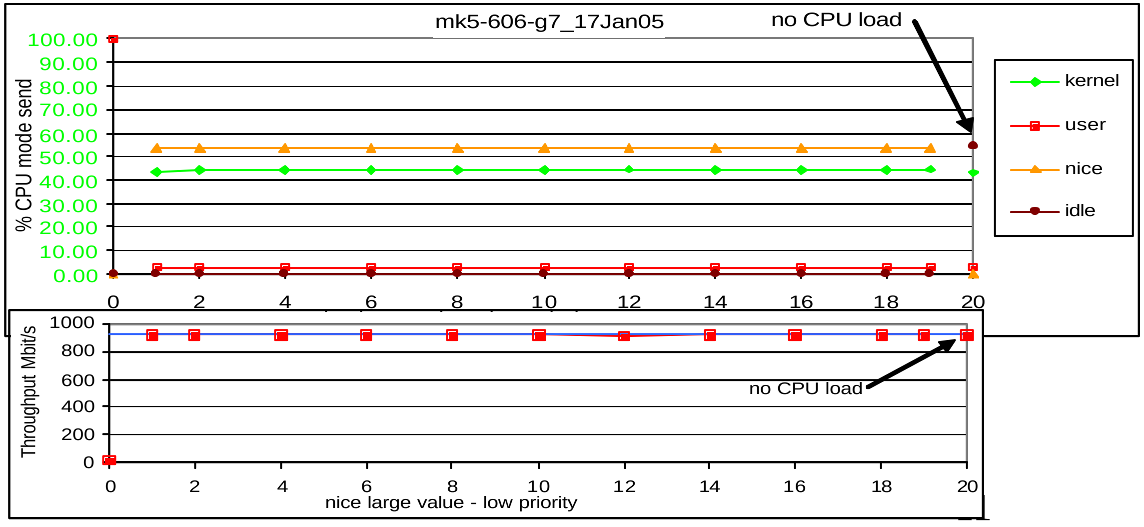
- Strange hiccups in cwnd → only correlation is SACK arrivals
- The SACK Processing is inefficient for large bandwidth delay products
  - Sender write queue (linked list) walked for:
    - Each SACK block
    - To mark lost packets
    - To re-transmit
  - Processing so long input Q becomes full
  - Get Timeouts

TCP Stacks & CPU Load

- Real User problem!
- End host TCP flow at 960 Mbit/s with rtt 1 ms falls to 770 Mbit/s when rtt 15 ms
- 1.2GHz PIII   rtt 1 ms
  - TCP iperf 980 Mbit/s
    - Kernel mode 95% Idle 1.3 %
  - CPULoad with nice priority
    - Throughput falls as priority increases
  - No Loss No Timeouts
- Not enough CPU power



- 2.8 GHz Xeon   rtt 1 ms
  - TCP iperf 916 Mbit/s
    - Kernel mode 43% Idle 55%
  - CPULoad with nice priority
    - Throughput constant as priority increases
  - No Loss No Timeouts
- Kernel mode includes TCP stack and Ethernet driver



Check out the end host: bbftp

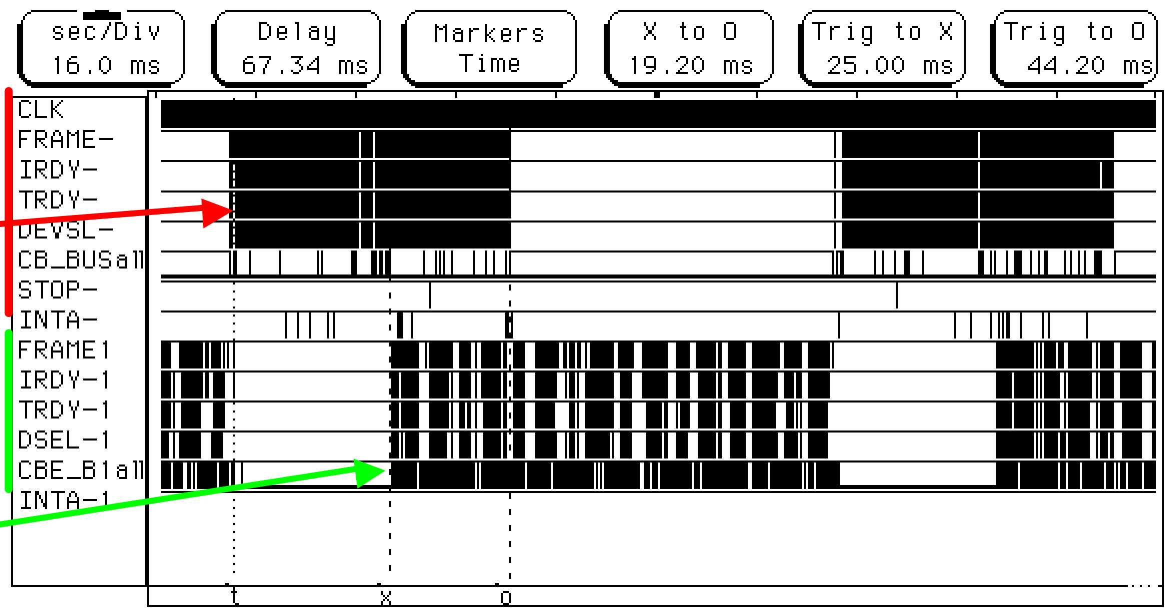
- **What is the end-host doing with your protocol?**
- **Look at the PCI-X buses**
- **3Ware 9000 controller RAID0**
- **1 Gbit Ethernet link**
- **2.4 GHz dual Xeon**
- **~660 Mbit/s**

PCI-X bus with RAID Controller

Read from disk
for 44 ms every 100ms

PCI-X bus with Ethernet NIC

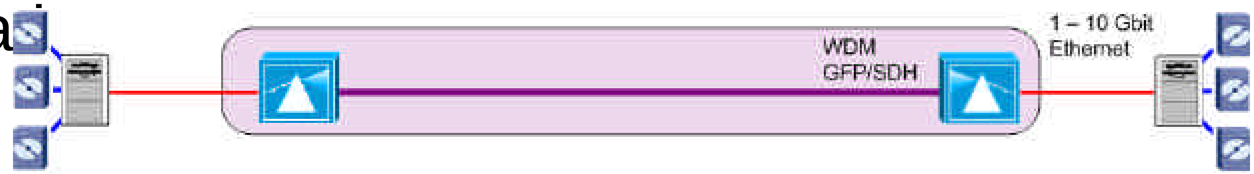
Write to Network
for 72 ms

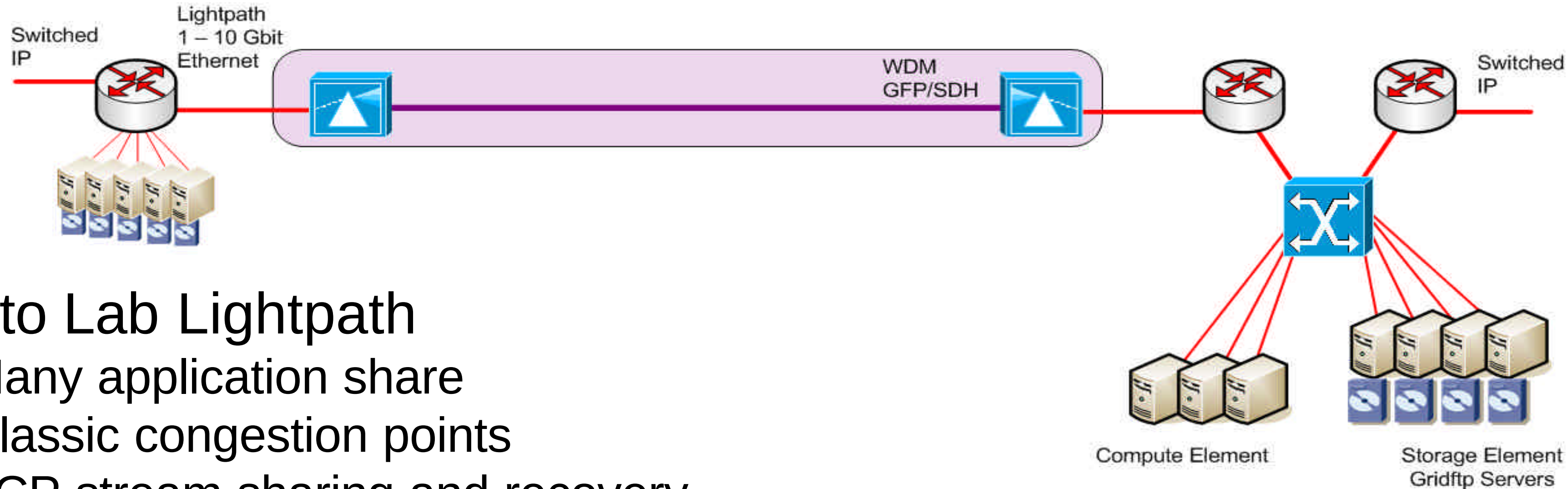


Transports for LightPaths

- For a Lightpath with a BER 10^{-16} i.e. a packet loss rate 10^{-12} or 1 loss in about 160 days, what do we use?

- Host to host Lightpath
  - One Application
  - No congestion
  - Lightweight framing

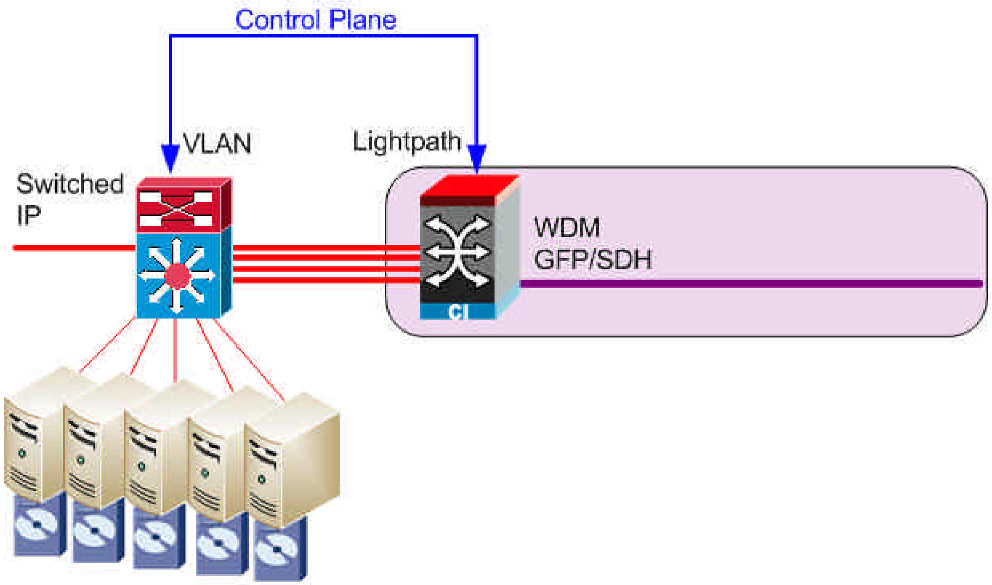




- Lab to Lab Lightpath
  - Many application share
  - Classic congestion points
  - TCP stream sharing and recovery
  - Advanced TCP stacks

Transports for LightPaths

- User Controlled Lightpaths
  - Grid Scheduling of CPUs & Network
  - Many Application flows
  - No congestion
  - Lightweight framing



- Some applications suffer when using TCP may prefer to use UDP DCCP XCP …
- E.g. With e-VLBI the data wave-front gets distorted and correlation fails
- Consider & include other transport layer protocols when defining tests.

A Few Items for Discussion

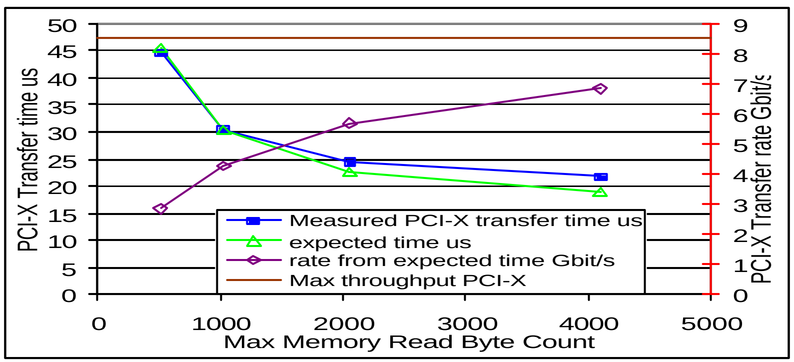
- Achievable Throughput
- Sharing link Capacity (OK what is sharing?)
- Convergence time
- Responsiveness
- rtt fairness (OK what is fairness?)
- mtu fairness
- TCP friendliness
- Link utilisation (by this flow or all flows)
- Stability of Achievable Throughput
- Burst behaviour
- Packet loss behaviour
- Packet re-ordering behaviour
- Topology – maybe some “simple” setups
- Background or cross traffic - how realistic is needed? – what protocol mix?
- Reverse traffic
- Impact on the end host – CPU load, bus utilisation, Offload
- Methodology – simulation, emulation and Real links ALL help

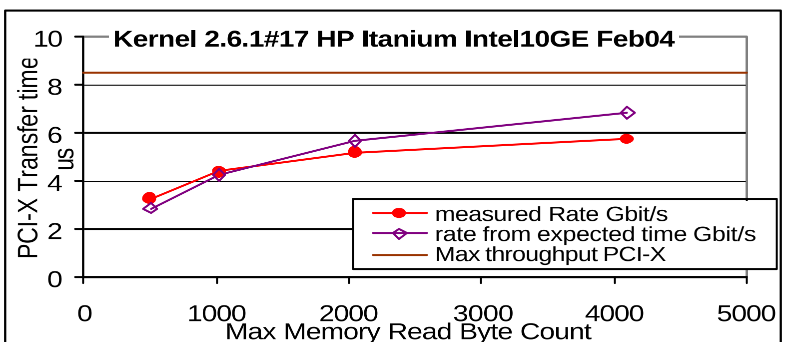
Any Questions?

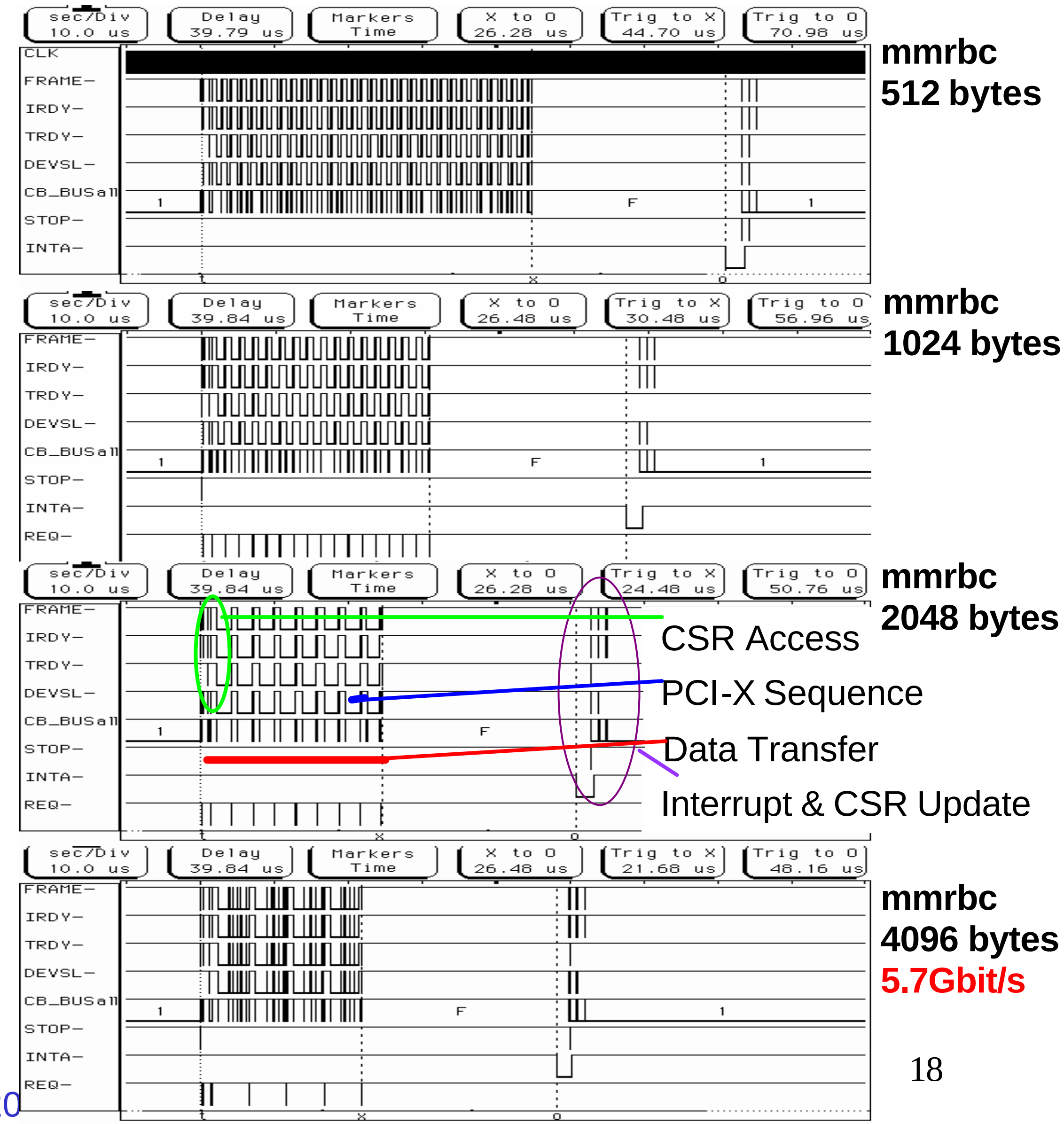
Backup Slides

10 Gigabit Ethernet: Tuning PCI-X

- 16080 byte packets every 200 µs
- Intel PRO/10GbE LR Adapter
- PCI-X bus occupancy vs mmrbc
  - Measured times
  - Times based on PCI-X times from the logic analyser
  - Expected throughput ~7 Gbit/s
  - Measured 5.7 Gbit/s







More Information Some URLs 1

- UKLight web site: http://www.uklight.ac.uk
- MB-NG project web site: http://www.mb-ng.net/
- DataTAG project web site: http://www.datatag.org/
- UDPmon / TCPmon kit + writeup:
  http://www.hep.man.ac.uk/~rich/net
- Motherboard and NIC Tests:

  http://www.hep.man.ac.uk/~rich/net/nic/GigEth_tests_Boston.ppt
  & http://datatag.web.cern.ch/datatag/pfldnet2003/

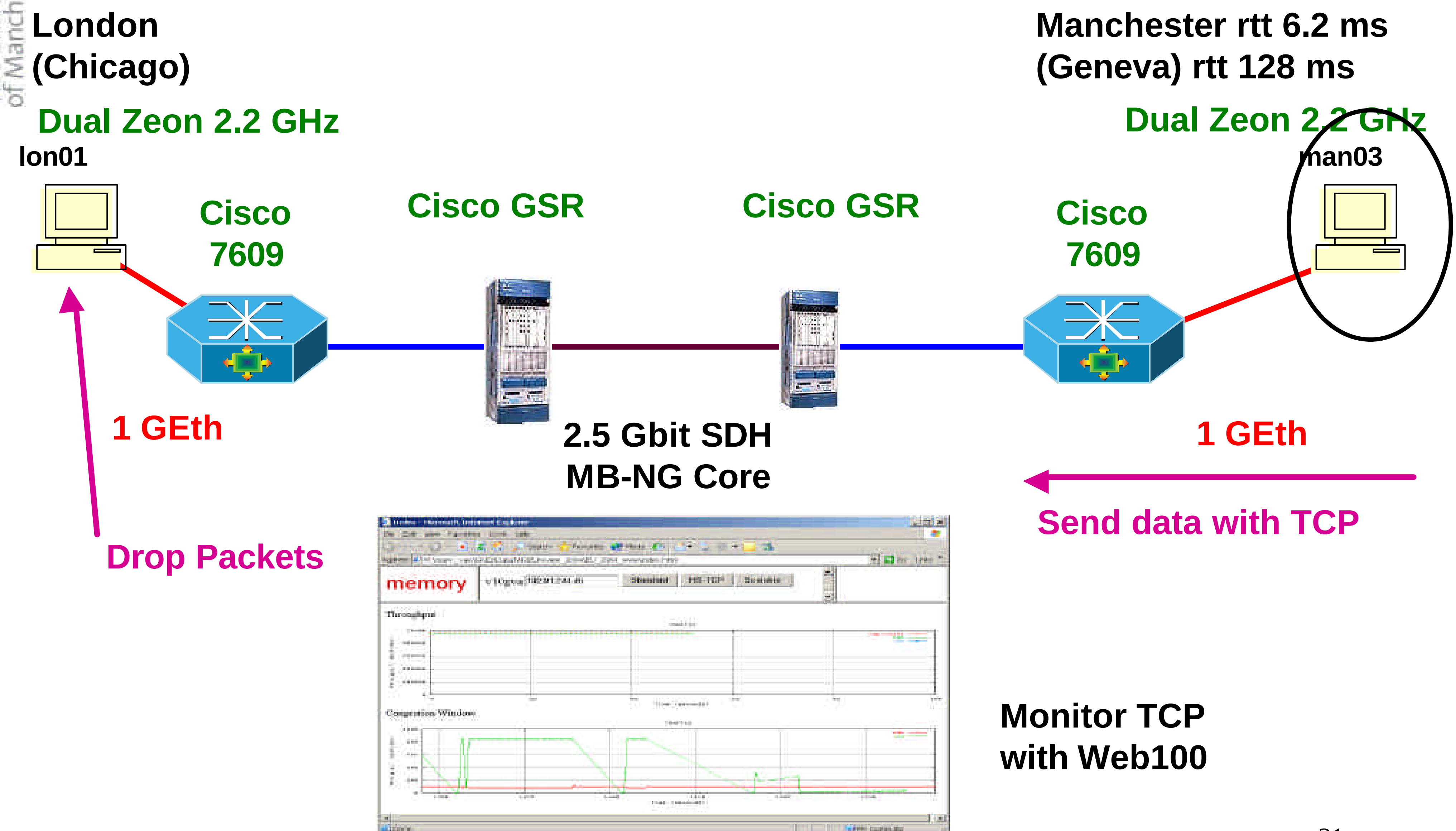
  "Performance of 1 and 10 Gigabit Ethernet Cards with Server Quality Motherboards" FGCS Special issue 2004

  http:// www.hep.man.ac.uk/~rich/
- TCP tuning information may be found at:
  http://www.ncne.nlanr.net/documentation/faq/performance.html
  & http://www.psc.edu/networking/perf_tune.html
- TCP stack comparisons:
  "Evaluation of Advanced TCP Stacks on Fast Long-Distance Production Networks" Journal of Grid Computing 2004
- PFLDnet http://www.ens-lyon.fr/LIP/RESO/pfldnet2005/
- Dante PERT http://www.geant2.net/server/show/nav.00d00h002

More Information Some URLs 2

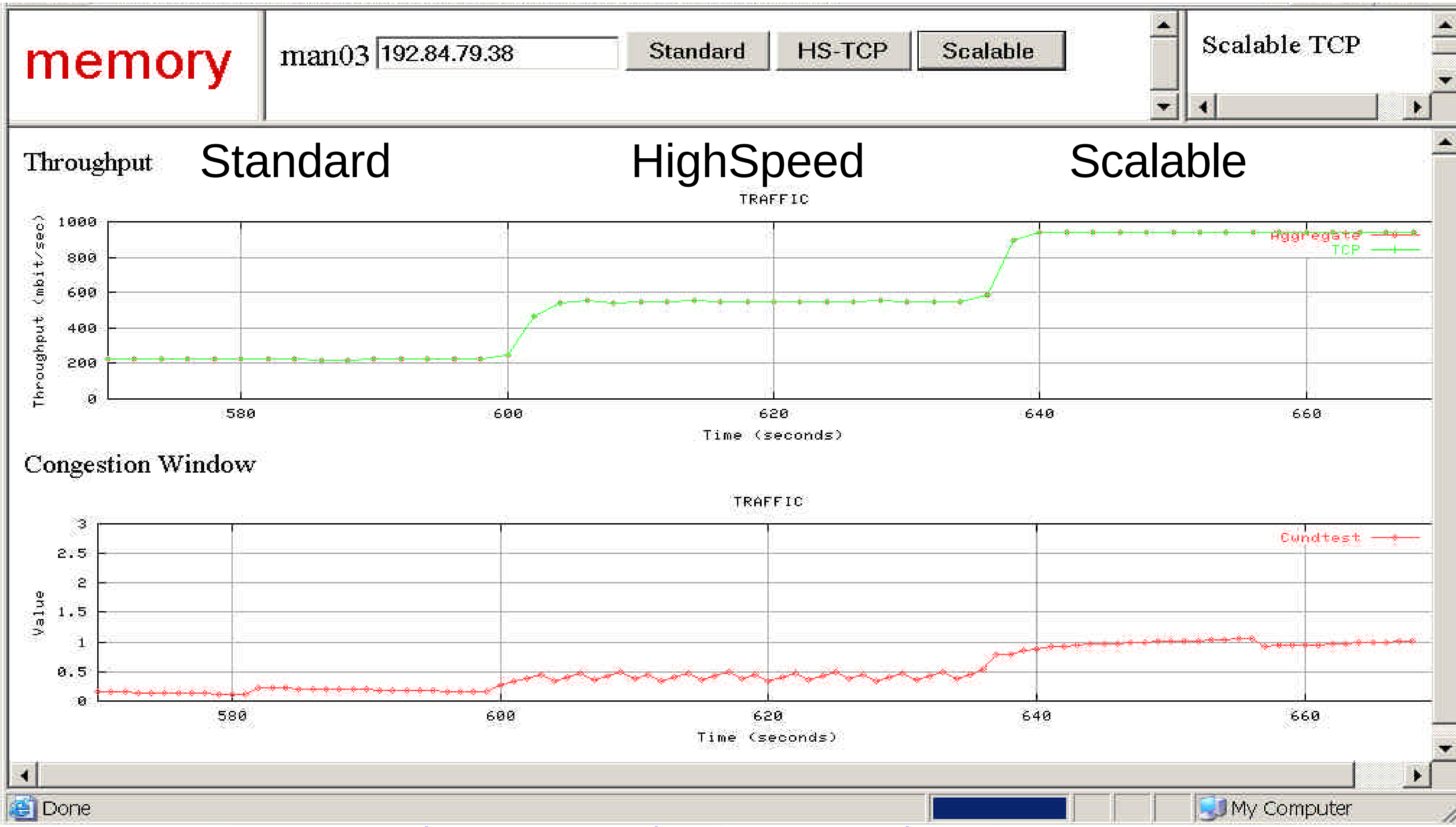
- Lectures, tutorials etc. on TCP/IP:
  - www.nv.cc.va.us/home/joney/tcp_ip.htm
  - www.cs.pdx.edu/~jrb/tcpip.lectures.html
  - www.raleigh.ibm.com/cgi-bin/bookmgr/BOOKS/EZ306200/CCONTENTS
  - www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ap1.htm
  - www.cis.ohio-state.edu/htbin/rfc/rfc1180.html
  - www.jbmelectronics.com/tcp.htm
- Encylopaedia
  - http://www.freesoft.org/CIE/index.htm
- TCP/IP Resources
  - www.private.org.il/tcpip_rl.html
- Understanding IP addresses
  - http://www.3com.com/solutions/en_US/ncs/501302.html
- Configuring TCP (RFC 1122)
  - ftp://nic.merit.edu/internet/documents/rfc/rfc1122.txt
- Assigned protocols, ports etc (RFC 1010)
  - http://www.es.net/pub/rfcs/rfc1010.txt & /etc/protocols

High Throughput Demonstrations

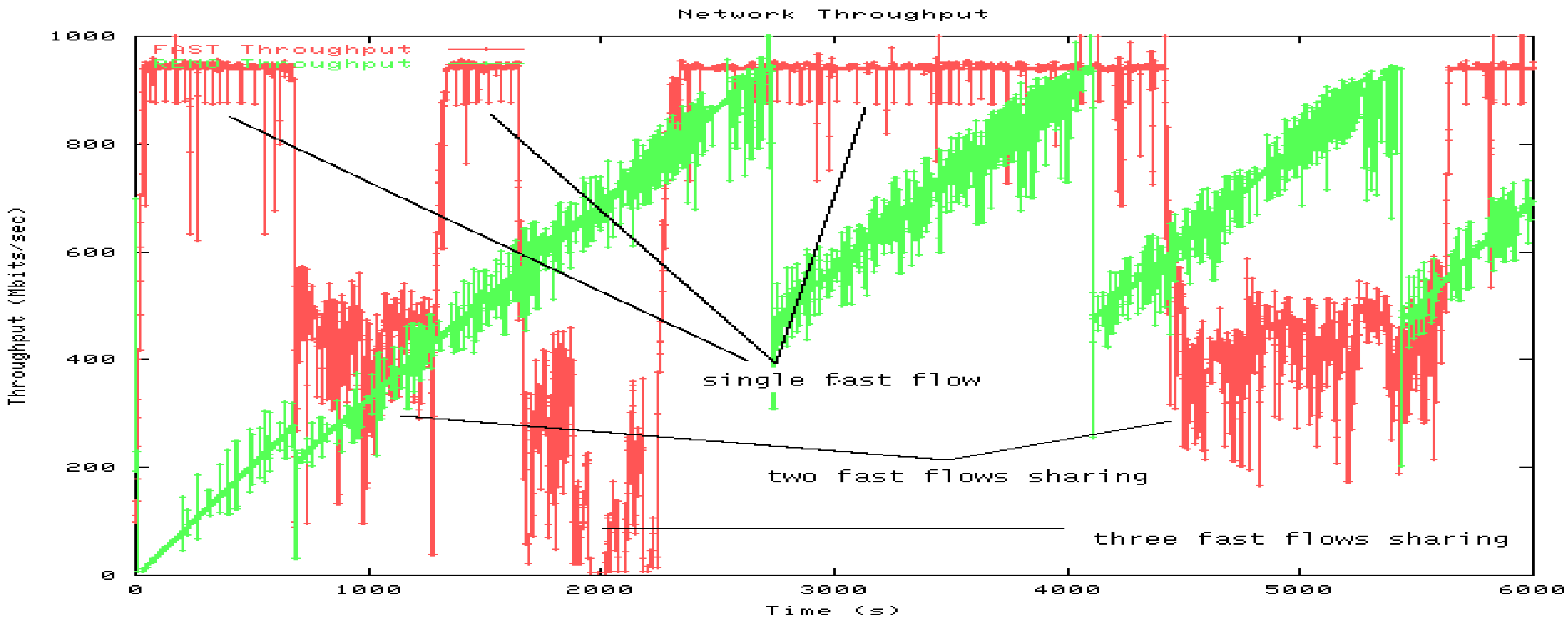


High Performance TCP – MB-NG

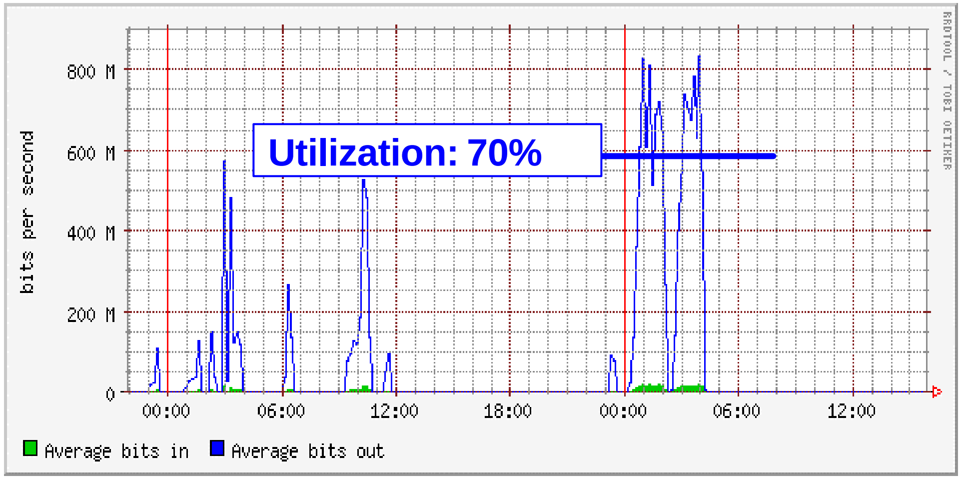
- Drop 1 in 25,000
- rtt 6.2 ms
- Recover in 1.6 s



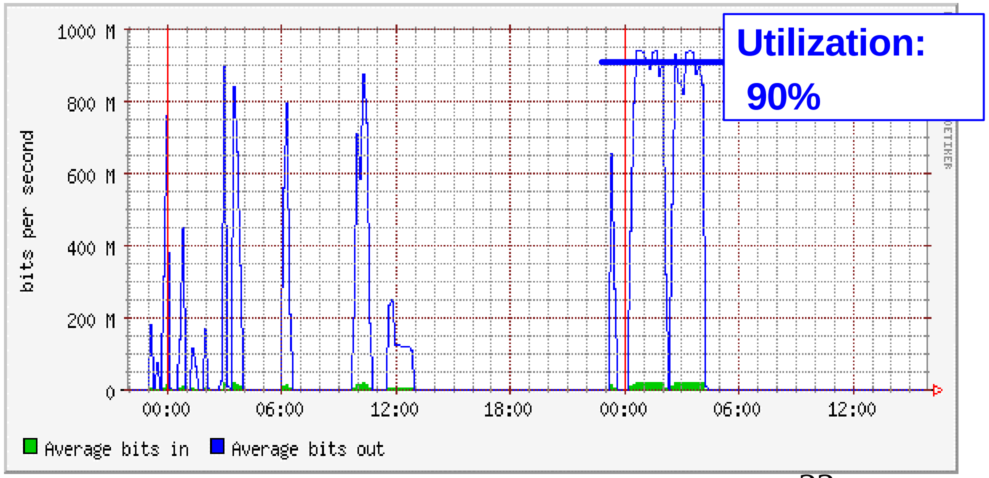
FAST TCP vs newReno



➔ **Traffic flow Channel #1 : newReno**



➔ **Traffic flow Channel #2: FAST**



Fast

- As well as packet loss, FAST uses RTT to detect congestion
  - RTT is very stable: ***s(RTT)*** ~ 9ms vs 37±0.14ms for the others

2nd flow never gets equal share of bandwidth

Big drops in throughput which take several seconds to recover from