FAST TCP in Linux

Cheng Jin David Wei Steven Low

California Institute of Technology

Outline

- Overview of FAST TCP.
- Design details.
- SC2002 experiments.
- On-going experiments.

Motivation

- Clear and present need in the HENP community.
- High capacity network infrastructure available today.
- Existing TCP protocol does not perform well.

FAST vs Linux TCP

	Flows	Bmps Peta	Throughput (Mbps)	Transfer (GB)
Linux TCP txqlen=100	1	1.86	185	78
Linux TCP txqlen=10000	1	2.67	**266**	111
FAST 19.11.2002	1	9.28	**925**	387
Linux TCP txqlen=100	2	3.18	317	133
Linux TCP txqlen=10000	2	9.35	**931**	390
FAST 19.11.2002	2	18.03	**1,797**	753

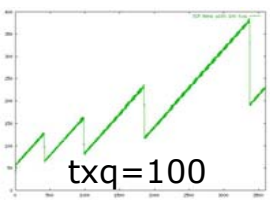
Distance = 10,037 km; Delay = 180 ms; MTU = 1500 B; Duration: 3600 s
Linux TCP Experiments: Jan 28-29, 2003

Aggregate Throughput

FAST
- Standard MTU of 1500 bytes
- Utilization averaged over one hour

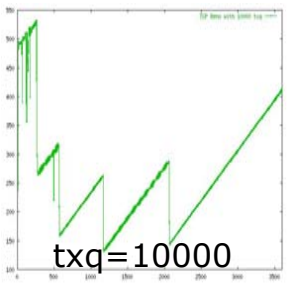
Average utilization

19%



Linux TCP

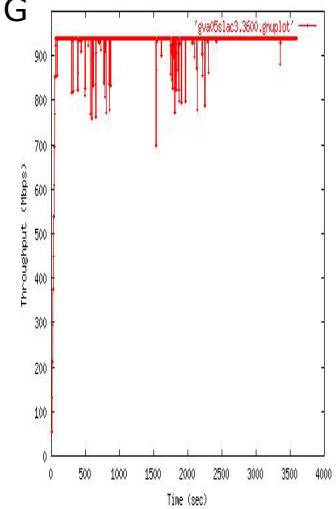
27%



Linux TCP

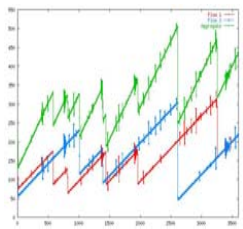
1G

95%



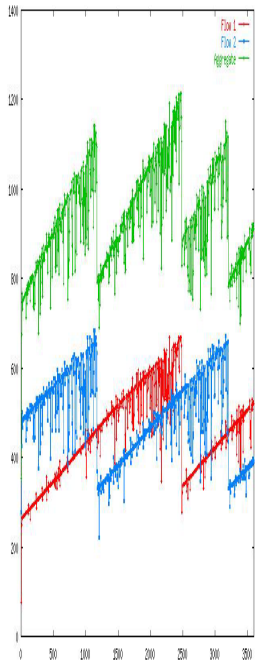
FAST

16%



Linux TCP

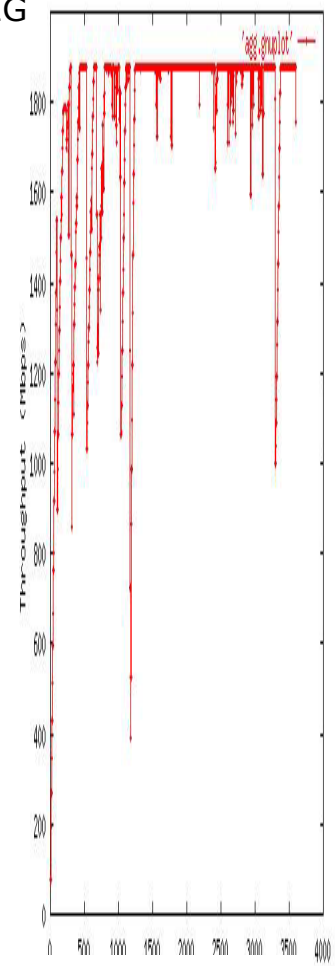
48%



Linux TCP

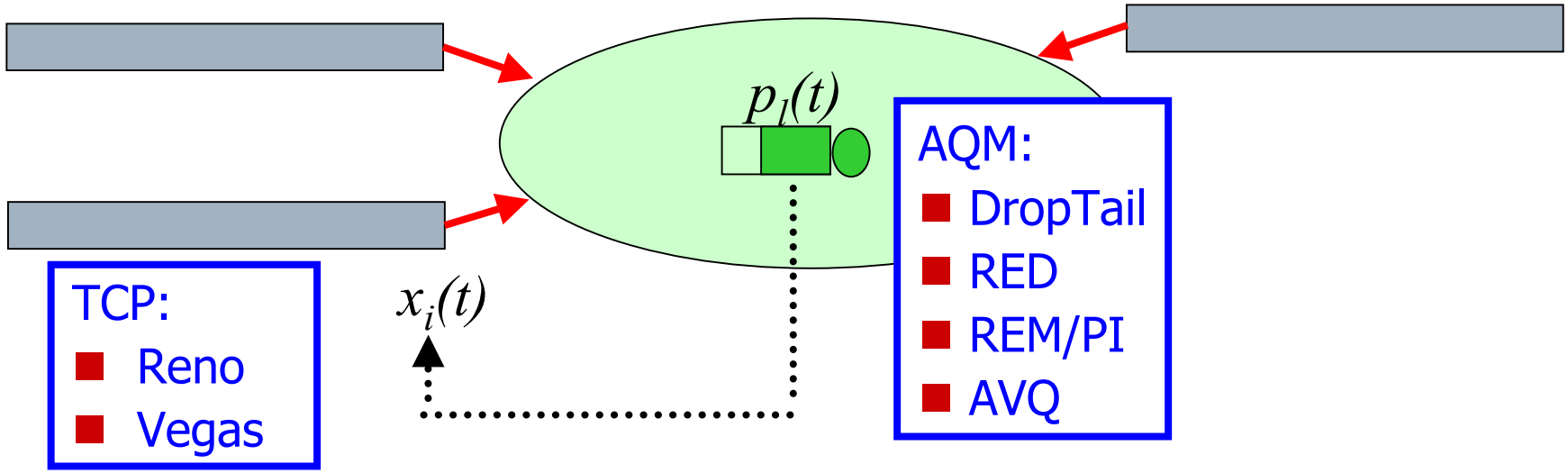
2G

92%



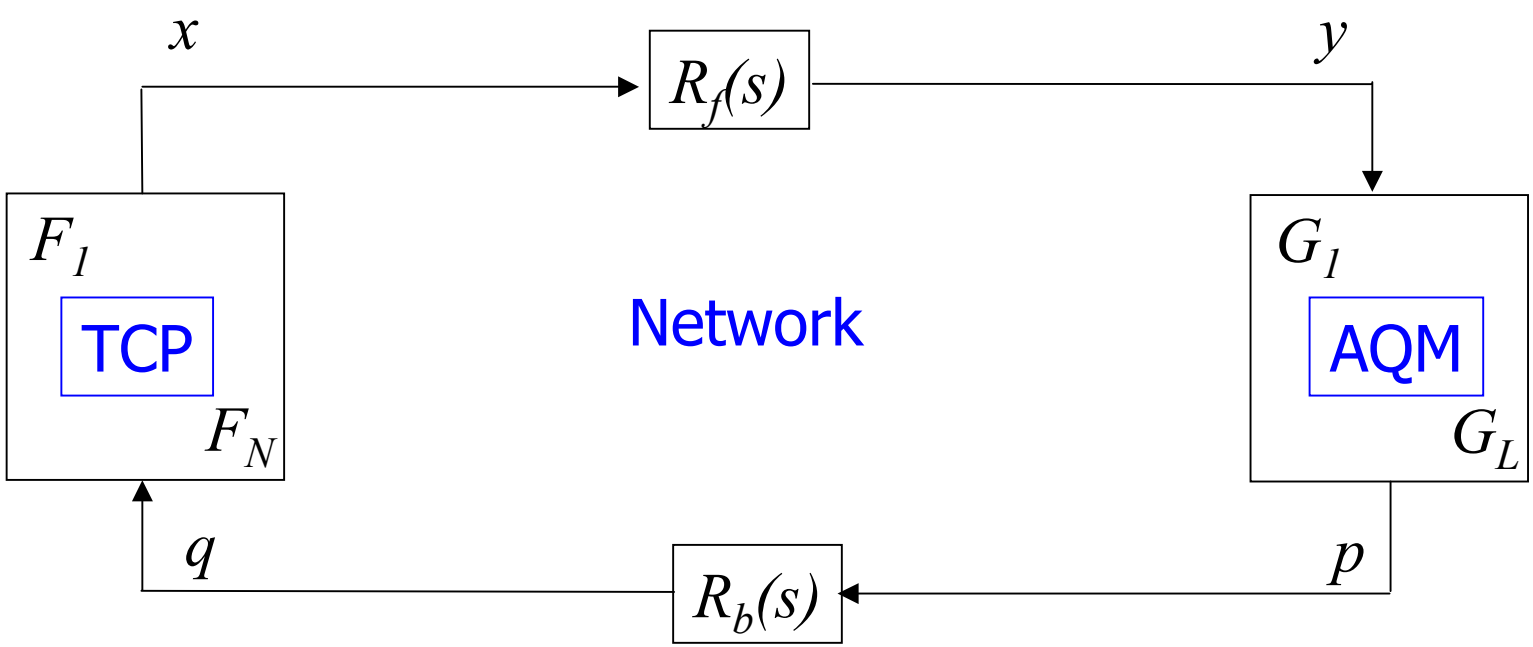
FAST

TCP/AQM



- Network congestion control has two components
  - TCP: adapts sending rate (window) to congestion.
  - AQM: adjusts & feeds back congestion information.
- They form a distributed feedback control system
  - Equilibrium & stability depends on both TCP and AQM.
  - Delay, capacity, routing, number of connections.

Network Model



$$[R_f]_{li} = e^{-s\overrightarrow{\tau_{li}}} \quad \text{if source } i \text{ uses link } l$$

$$[R_b]_{li} = e^{-s\overleftarrow{\tau_{li}}} \quad \text{if source } i \text{ uses link } l$$

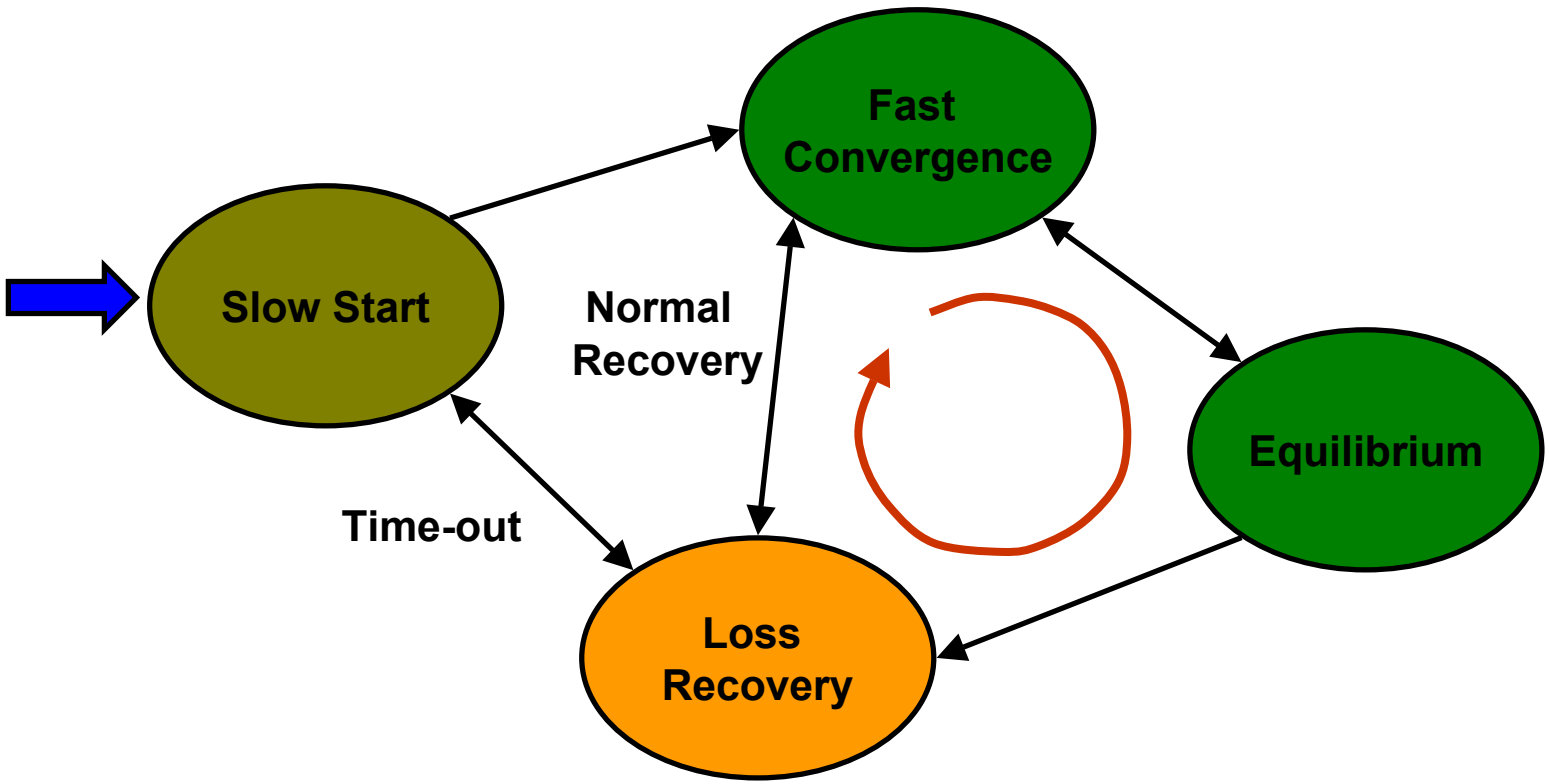
Fast AQM Scalable TCP

- Equilibrium Properties
  - Uses both delay and loss.
  - Achieves fairness defined by its utility function.
  - Achieves high utilization.
- Stability Properties
  - Stable for arbitrary delay, capacity, routing, and load
  - Good performance and robustness.

Summary of Changes

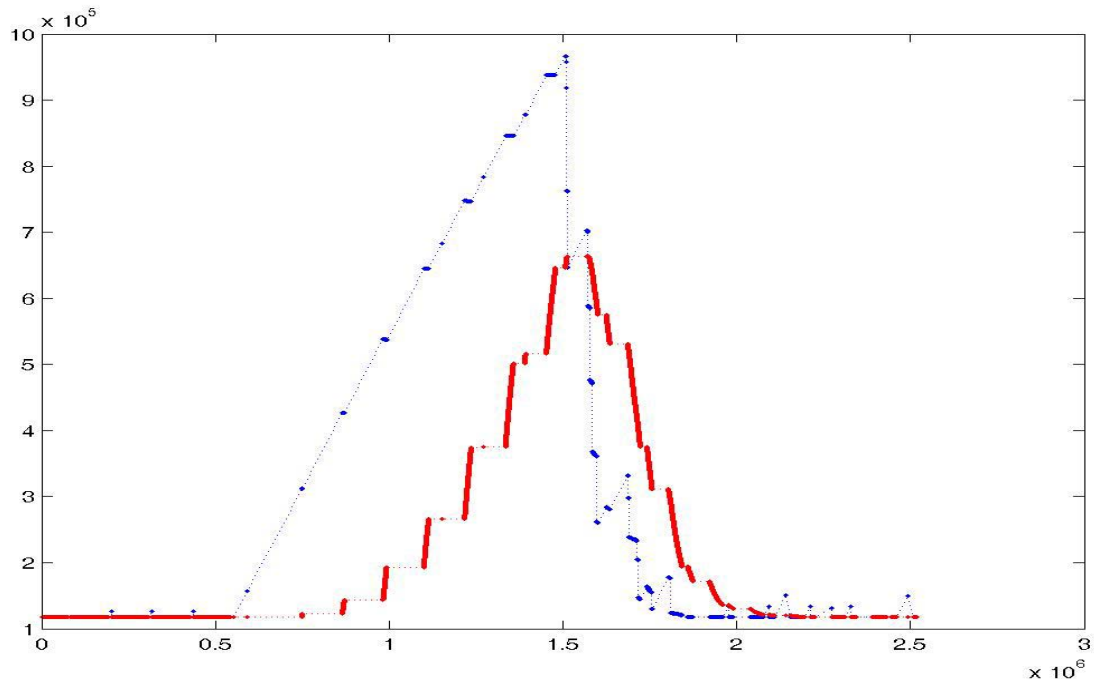
- Reacting to both delay and loss.
- Maintaining an equilibrium defined by a target queueing delay.
- Rapid converging when equilibrium changes.
- Pacing to smooth out traffic.

FAST TCP Flow Chart



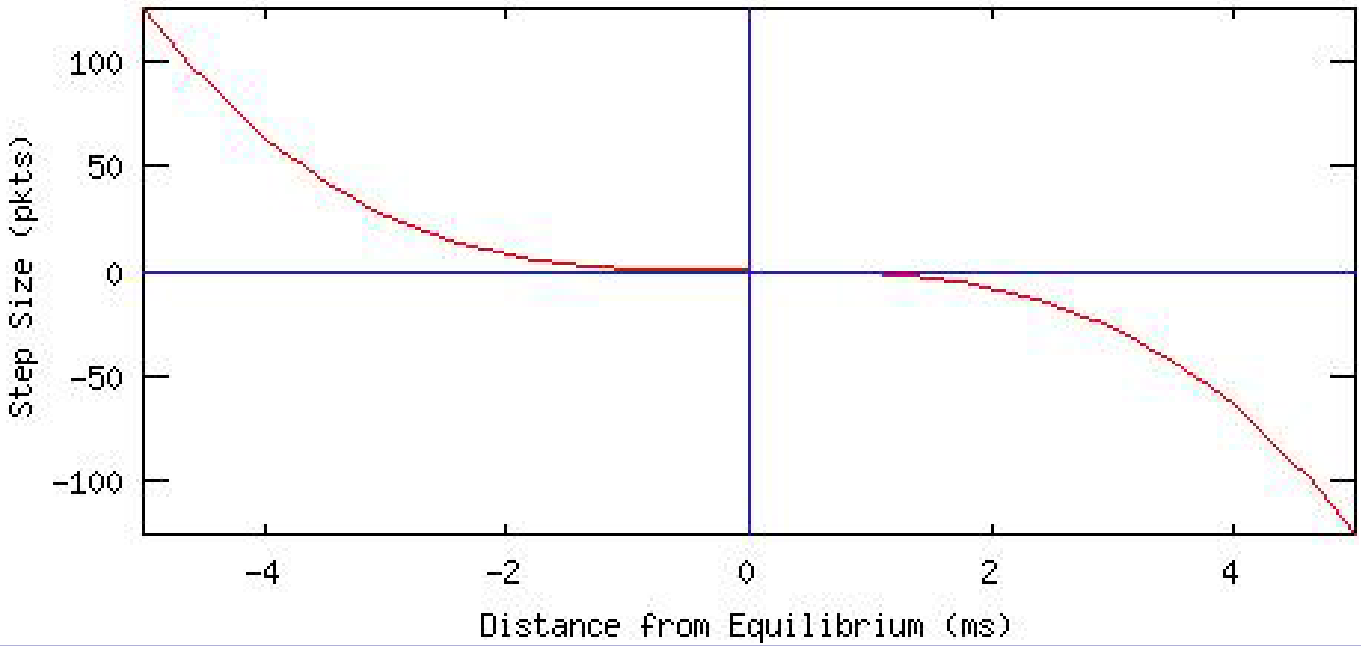
Queueing Delay Estimation

- Measure avg. RTT and minimum RTT.
- High resolution kernel timestamp.
- Exponential averaging of RTT samples.



Fast Convergence

- Monitor the per-ack queueing delay to avoid overshoot.
- Window increment scales with the "distance" from equilibrium.



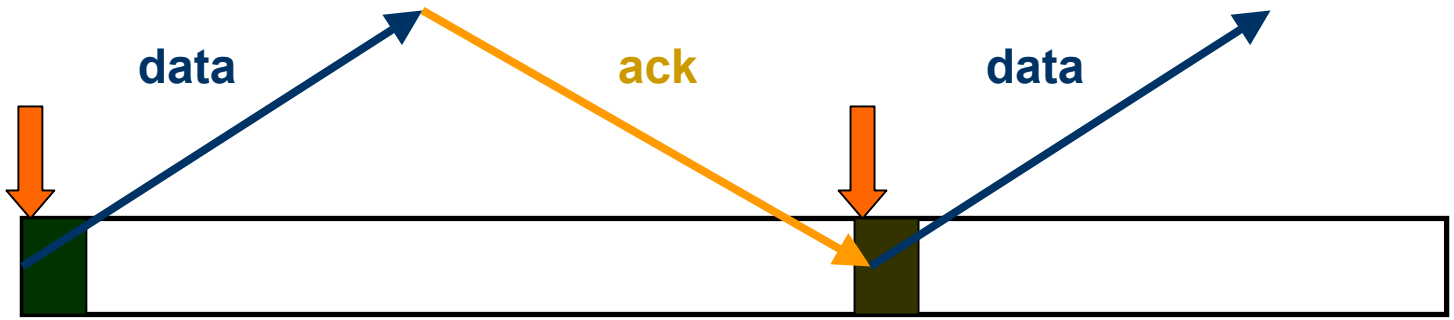
Equilibrium

- Define an equilibrium neighborhood around the target queueing delay.
- Small *cwnd* adjustment in large time-scale -- one per RTT.
- Per-ack delay monitoring to enable timely detection of changes in equilibrium.

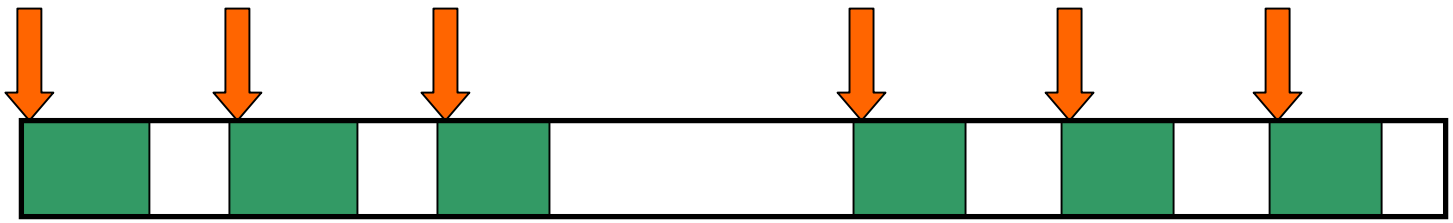
Pacing

- Why is it important?
  - Avoid queueing delay caused by bursty traffic.
- What do we pace?
  - Increment to congestion window.
- Software pacing
  - Must balance performance with overhead.

Pacing Example

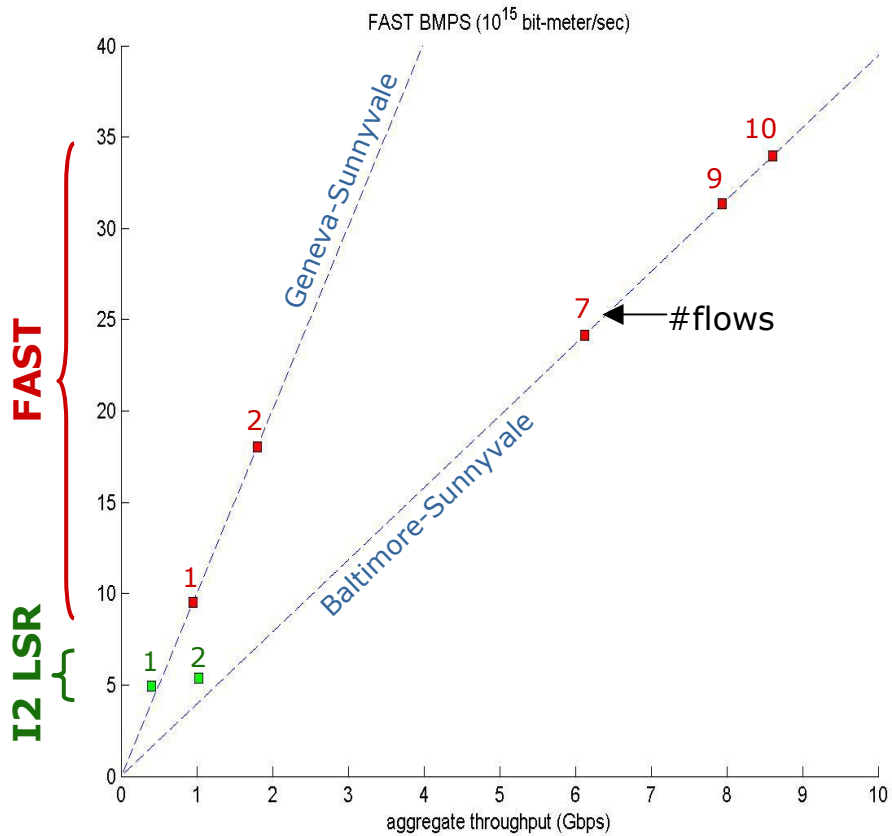


***cwnd* increments are scheduled at fixed intervals.**



SCinet
Caltech-SLAC experiments

SC2002
Baltimore, Nov 2002



Highlights

FAST TCP

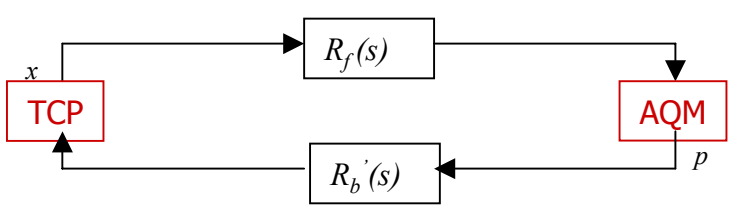
- Standard MTU
- Peak window = 14,255 pkts
- Throughput averaged over > 1hr
- 925 Mbps single flow/GE card
  - 9.28 petabit-meter/sec
  - 1.89 times LSR
- 8.6 Gbps with 10 flows
  - 34.0 petabit-meter/sec
  - 6.32 times LSR
- 21TB in 6 hours with 10 flows

Implementation

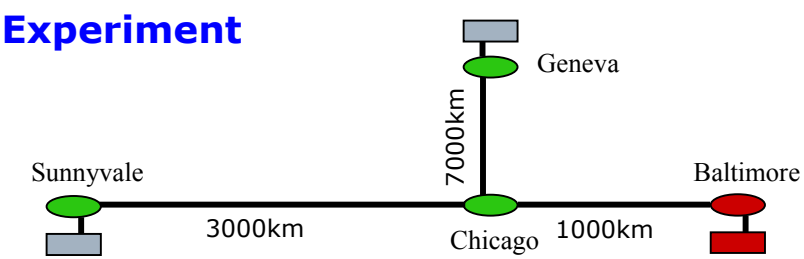
- Sender-side modification
- Delay based

Theory

Internet: distributed feedback system



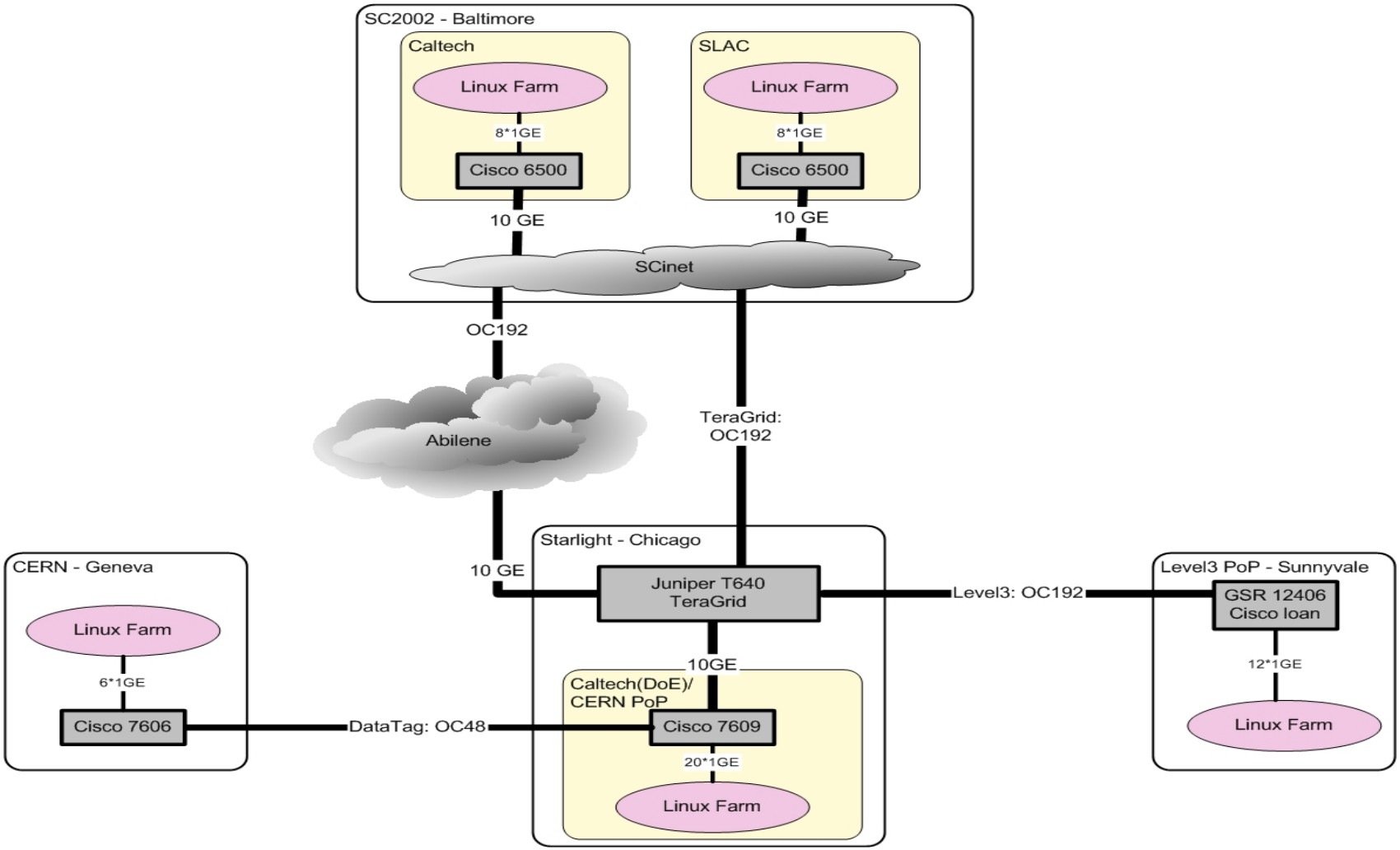
Experiment



http://netlab.caltech.edu/FAST

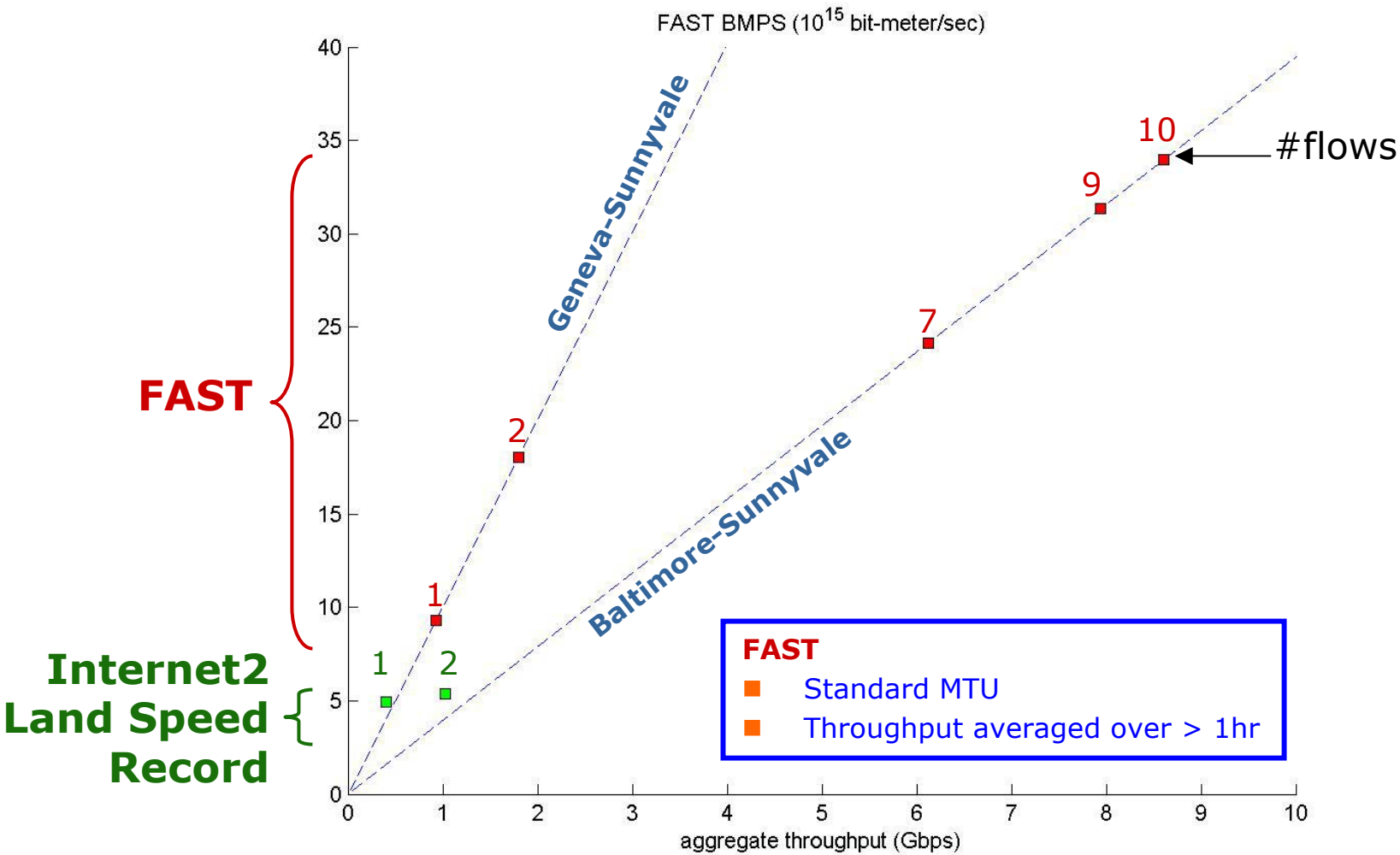
C. Jin, D. Wei, S. Low
FAST Team and Partners

SC2002 Network



(Sylvain Ravot, Caltech/CERN)

FAST BMPS

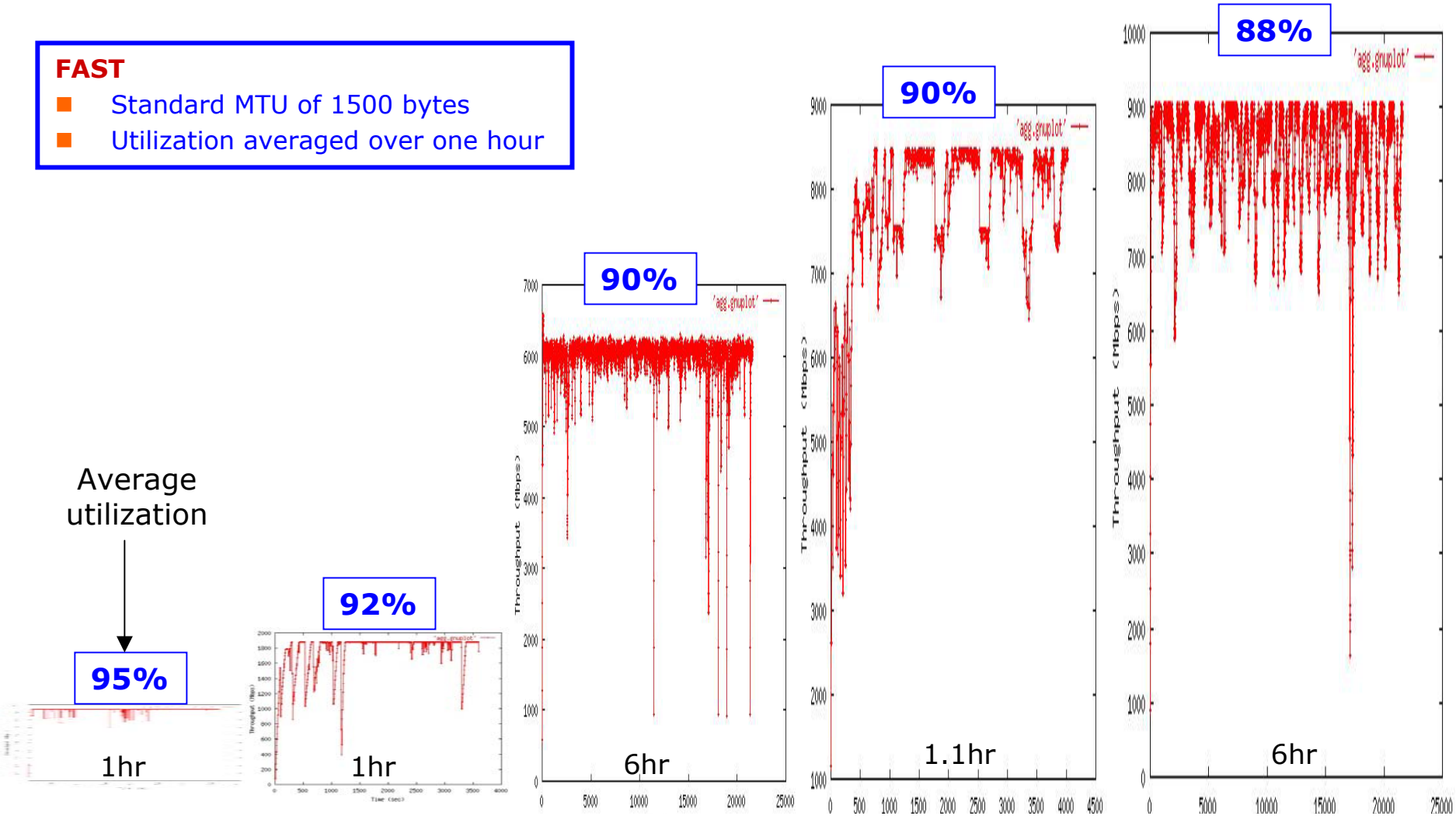

FAST BMPS (10^{15} bit-meter/sec)
aggregate throughput (Gbps)
Geneva-Sunnyvale
Baltimore-Sunnyvale
#flows
FAST
Internet2 Land Speed Record
FAST
Standard MTU
Throughput averaged over > 1hr

Aggregate Throughput

FAST

- Standard MTU of 1500 bytes
- Utilization averaged over one hour

Average utilization



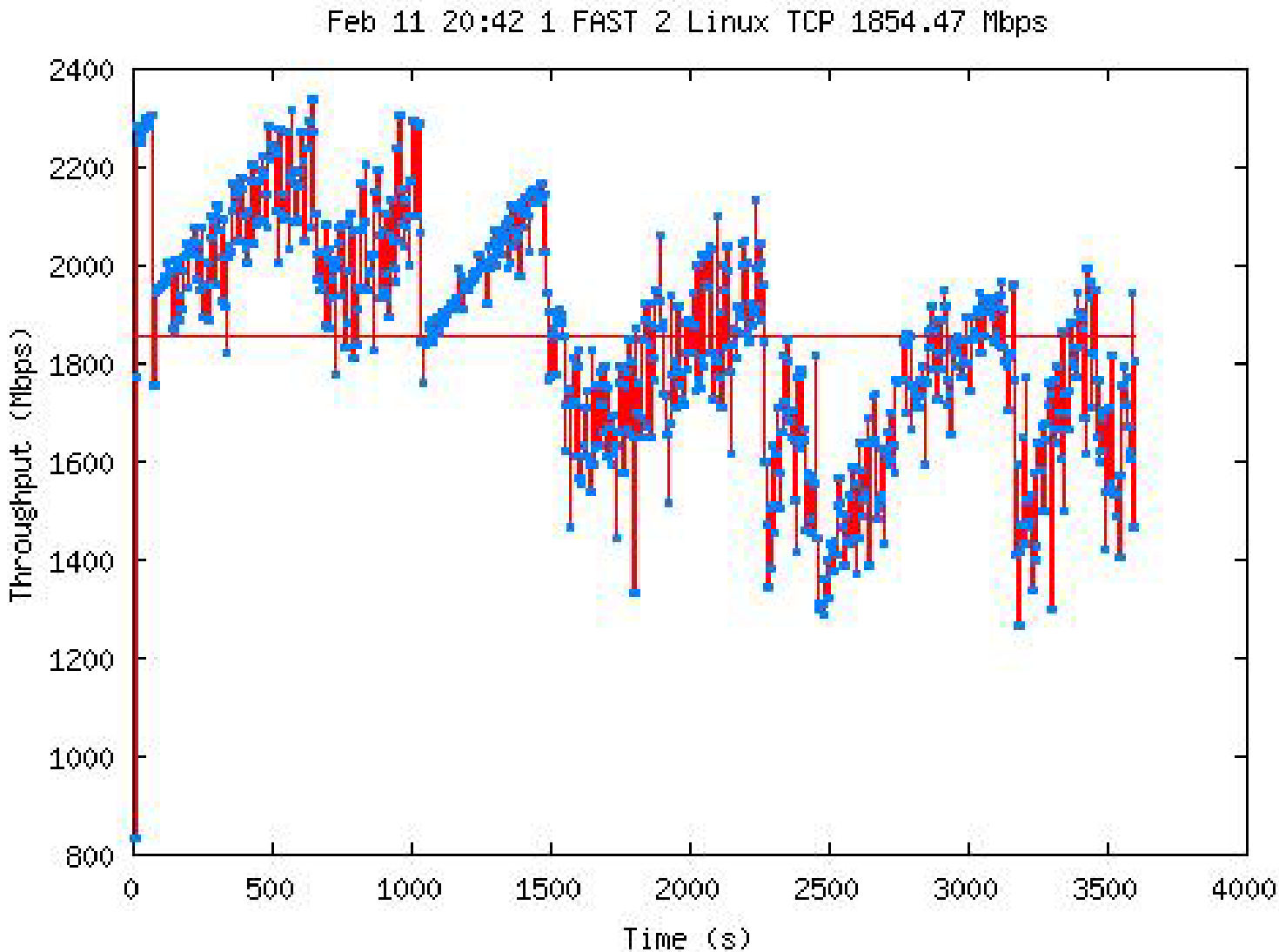
1 flow | 2 flows | 7 flows | 9 flows | 10 flows

Acknowledgments

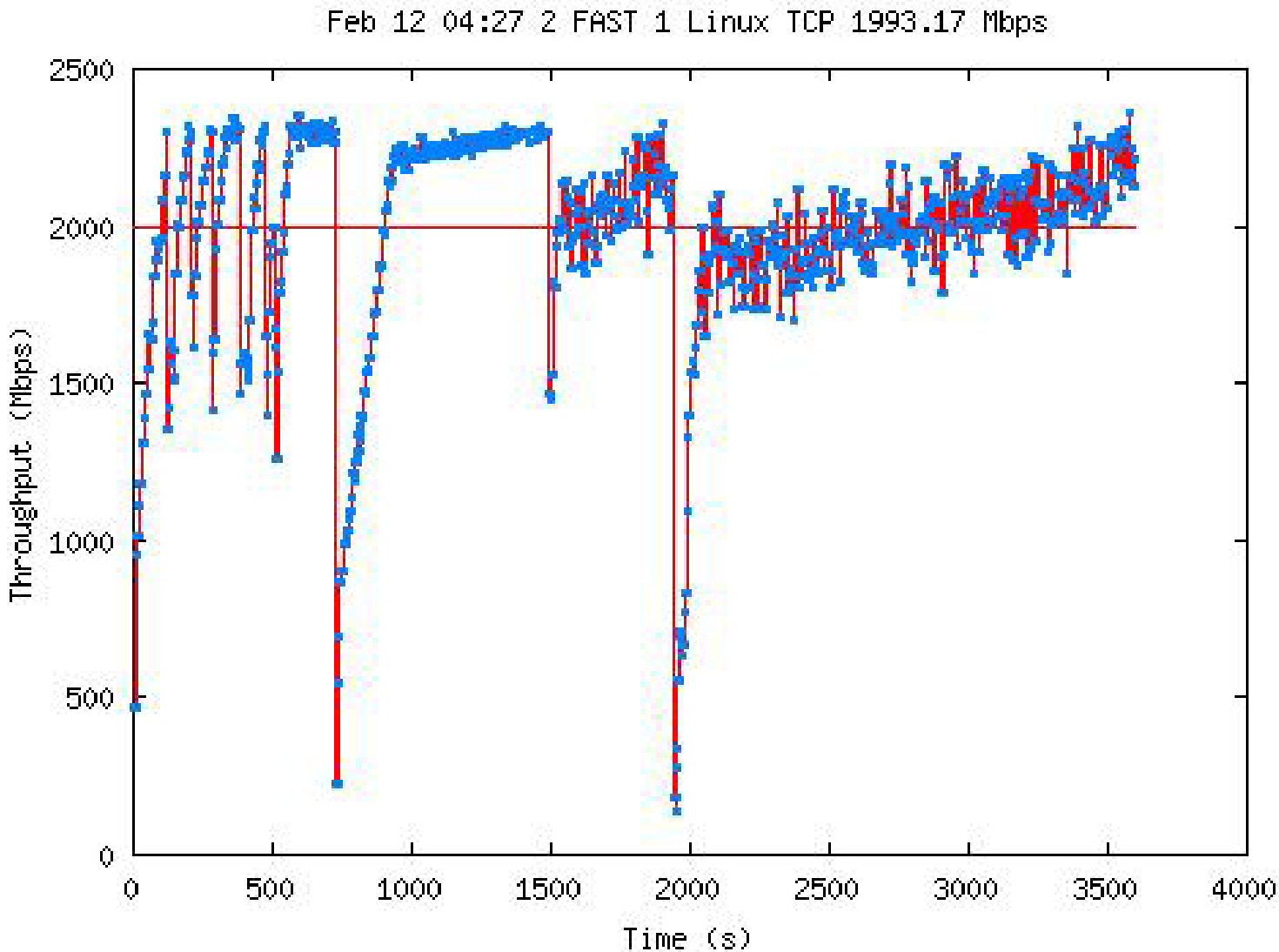
http://netlab.caltech.edu/FAST

- **Prototype**
  C. Jin, D. Wei
- **Theory**
  D. Choe (Postech/Caltech), J. Doyle, S. Low, F. Paganini (UCLA), J. Wang, Z. Wang (UCLA)
- **Experiment/facilities**
  - Caltech: J. Bunn, C. Chapman, C. Hu (Williams/Caltech), H. Newman, J. Pool, S. Ravot (Caltech/CERN), S. Singh
  - CERN: O. Martin, P. Moroni
  - Cisco: B. Aiken, V. Doraiswami, R. Sepulveda, M. Turzanski, D. Walsten, S. Yip
  - DataTAG: E. Martelli, J. P. Martin-Flatin
  - Internet2: G. Almes, S. Corbato
  - Level(3): P. Fernes, R. Struble
  - SCinet: G. Goddard, J. Patton
  - SLAC: G. Buhrmaster, R. Les Cottrell, C. Logg, I. Mei, W. Matthews, R. Mount, J. Navratil, J. Williams
  - StarLight: T. deFanti, L. Winkler
  - TeraGrid: L. Winkler
- **Major sponsors**
  ARO, CACR, Cisco, DataTAG, DoE, Lee Center, NSF

Fairness Experiment I

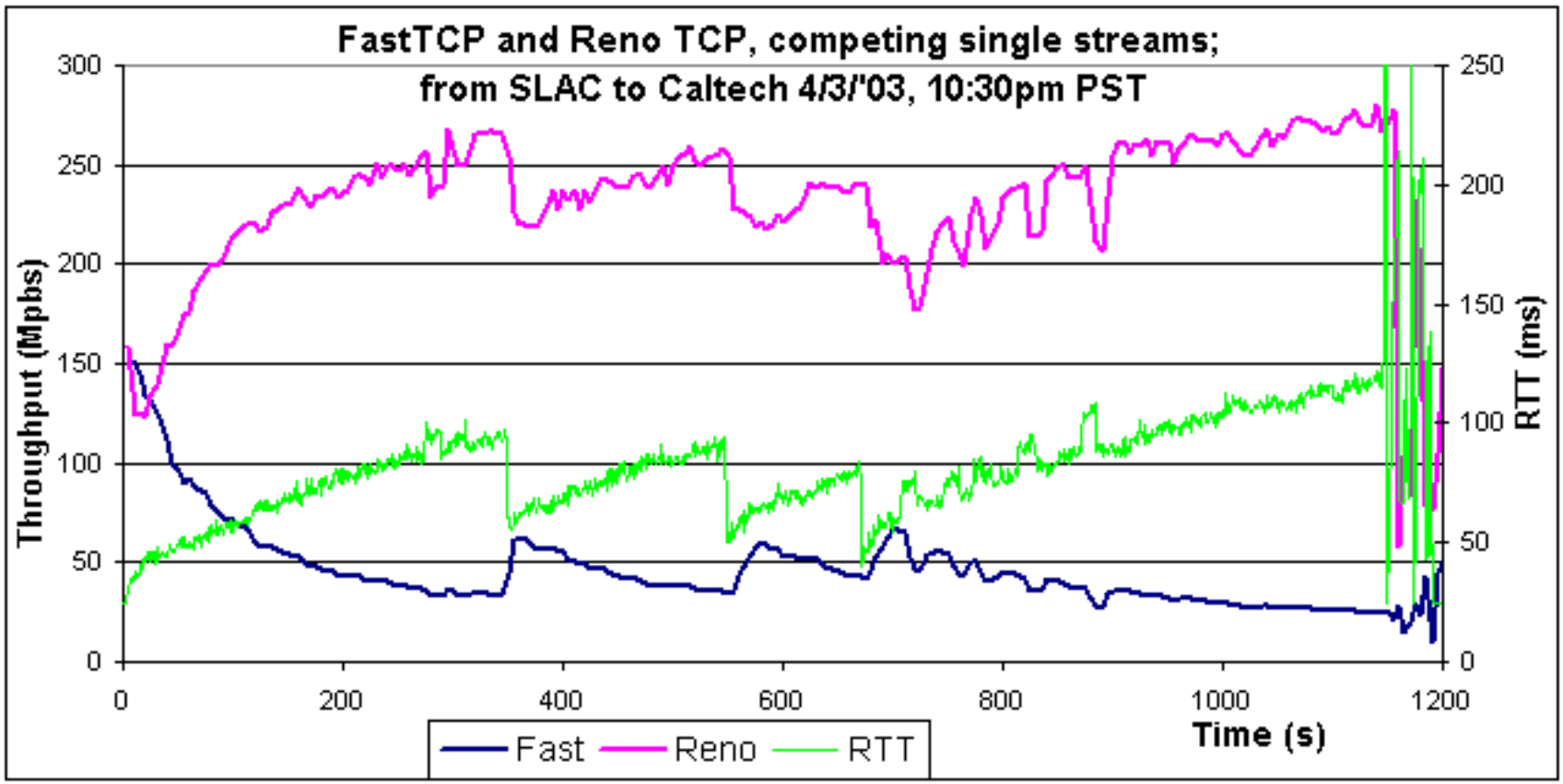


Fairness Experiment II



FAST Reacting to Delay

Should FAST react to "queueing delay" in this case?



* Done by Les Cottrell and Fabrizio Coccetti at SLAC

10GbE Experiment

- ~2.4 Gbps using jumbo frame from Sunnyvale to Geneva (all stacks).
- Frequent packet loss with 1500-byte MTU even at small window sizes (all stacks).
- Bottleneck router had enough buffer space.

Future Work

- Work on an improved beta version.
- More stability and fairness experiments.
- More 10 GbE WAN tests.
- Extensive testing on shared links.