NC STATE UNIVERSITY

A Step toward Realistic Performance Evaluation of High-Speed TCP Variants

Sangtae Ha, Yusung Kim, Long Le,
Injong Rhee, and Lisong Xu

PFLDnet 2006

February 2, 2006

Outline

- Introduction
- Experimental Methodology
- Results
- Conclusions

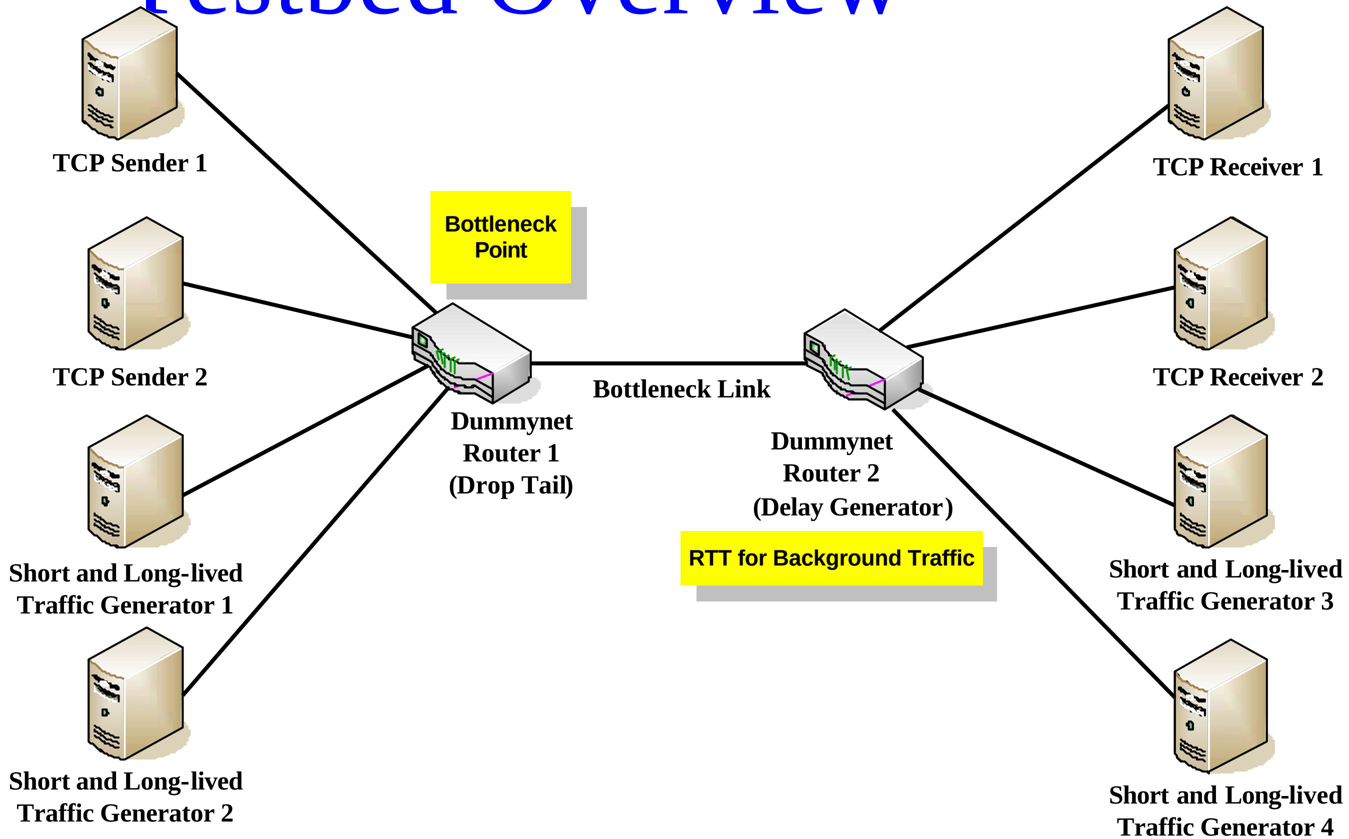
Introduction

- Two trends:
  - Growing link capacity
  - High-bandwidth demand of high-performance applications
- TCP performance is unsatisfactory on high-speed and long distance networks
- Many TCP variants promise to achieve better performance than TCP: BIC TCP, CUBIC, FAST, HSTCP, H-TCP, STCP, TCP-Westwood, LTCP, and TCP-Africa
- We need thorough evaluations of these protocols before we deploy them

Motivation

- Factors in constructing realistic testing environments: bottleneck bandwidth, RTT, network topology, router queues
- Background and reverse traffic:
  - Queue fluctuations
  - Reduce phase effects
  - Different loss patterns
  - Important factor in realistic evaluations but has not received sufficient attention

Experimental Methodology: Testbed Overview



- Use *dummynet* to provide per-flow propagation delays
  - Delays for high-speed TCP flows are set to certain values
  - Delays for background traffic are randomly sampled from a distribution obtained from an Internet measurement study

Background Traffic

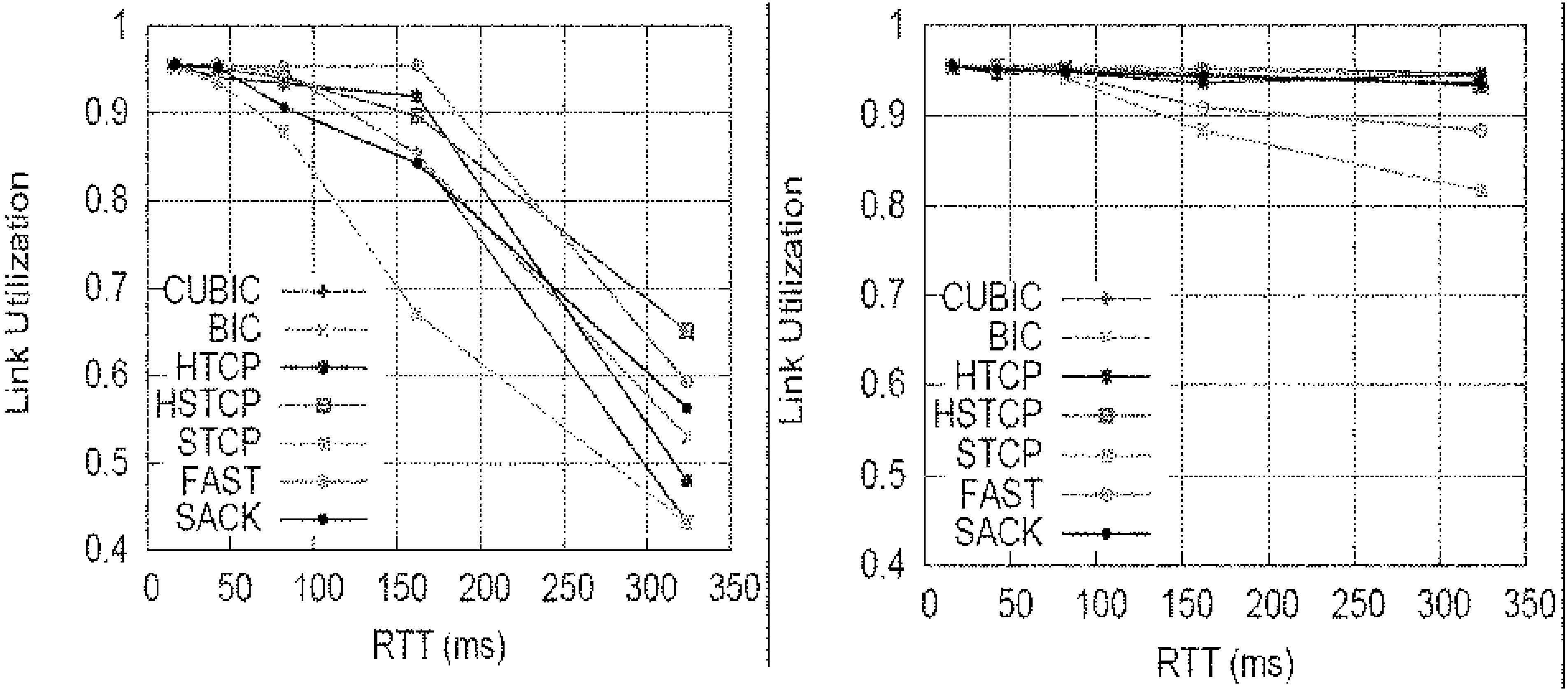
- Evaluate high-speed TCP variants with and without background traffic
- For background traffic, we use:
  - Long-lived flows are generated by *iperf* to emulate *ftp*
  - Short-lived flows emulate web sessions and are generated by using two parameter: file sizes (lognormal body with a heavy tail) and inter-arrival times (exponential)
- We also generate reverse traffic (short- and long-lived flows) to reduce phase effects

Experimental Methodology

Experimental Plan

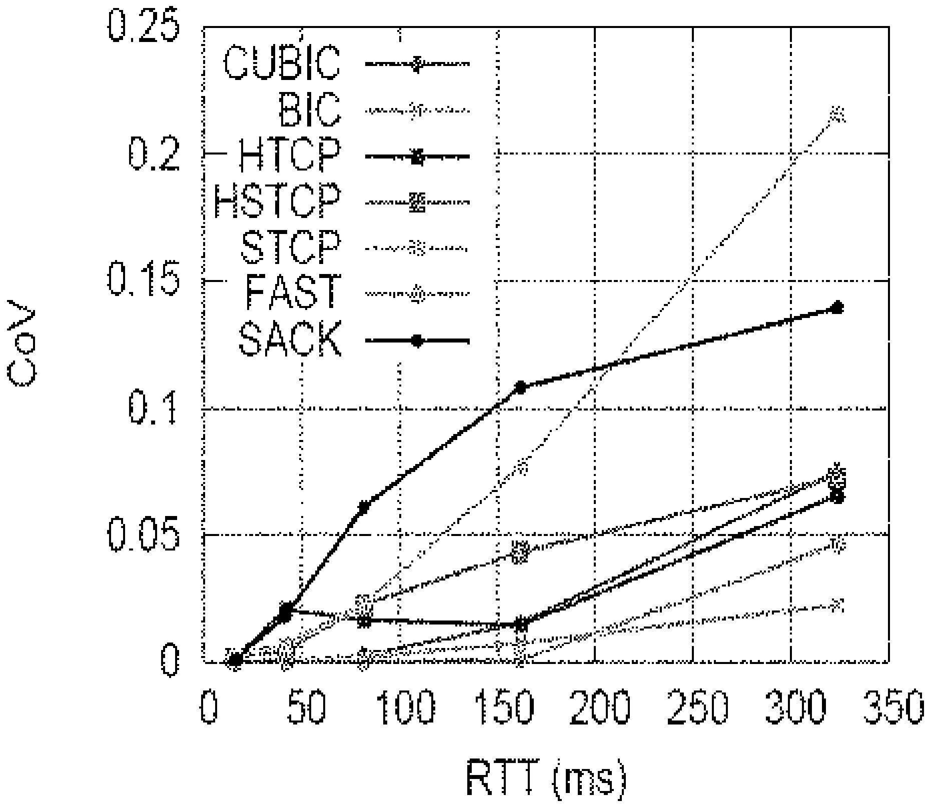
- We run *dummynet* on a PC router to configure the queue size
    - Also collect data to report link utilization and packet loss rate
- Experiments run for 1200 seconds
- Two high-speed flows start after 30 and 130 seconds
- Evaluate high-speed TCP variants using the following performance metrics: link utilization, stability, packet loss rate, RTT fairness, TCP friendliness, intra-protocol fairness, and convergence

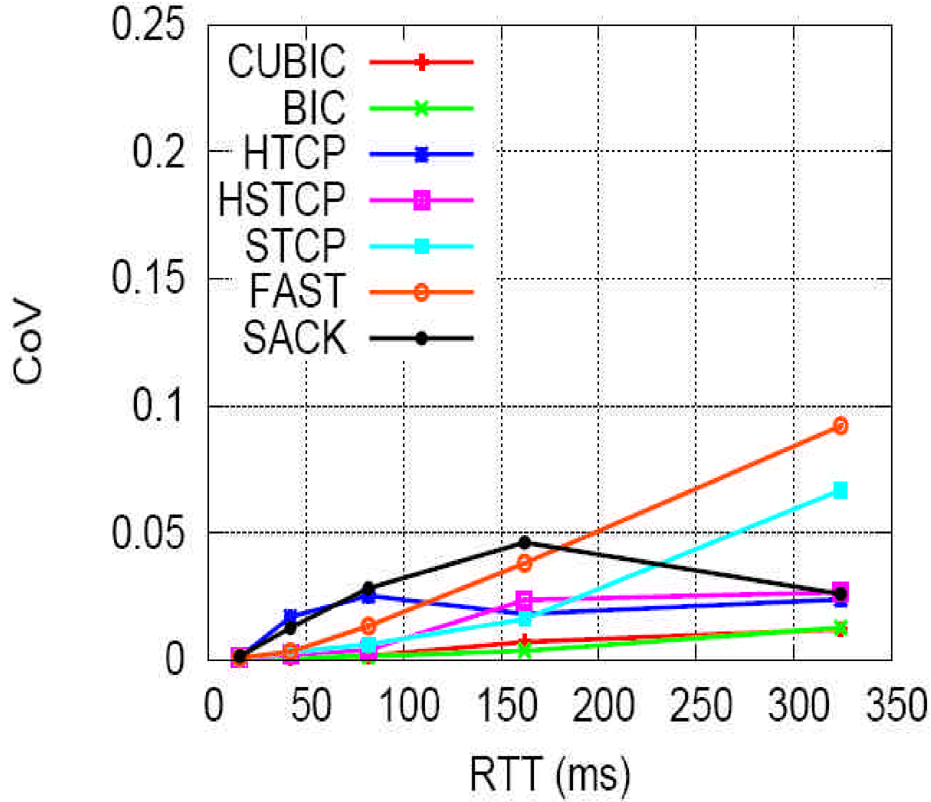
Link Utilization



- Link utilization is improved when background traffic is added
  - Most protocols obtain good utilization with background traffic

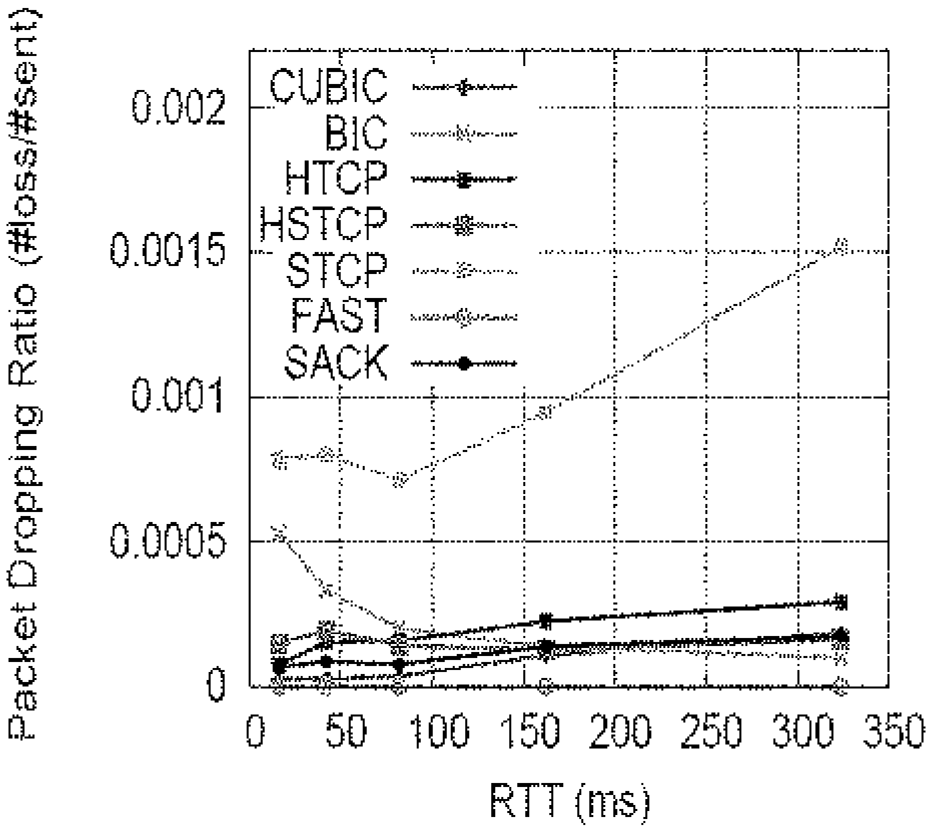
Stability

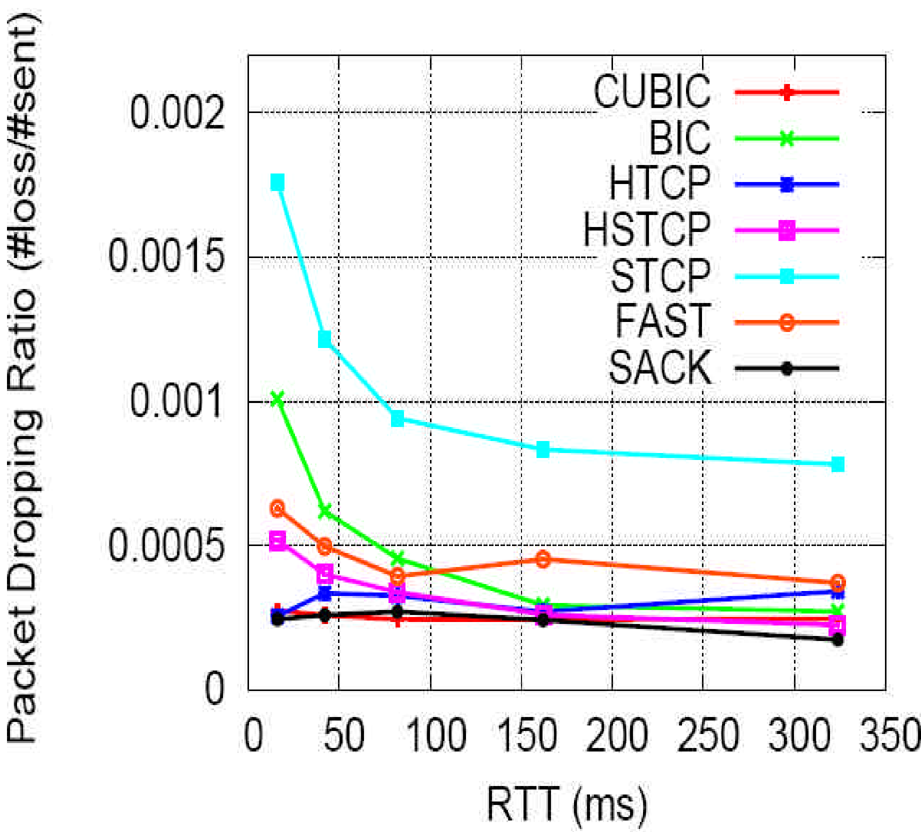




- We measure stability of a protocol as the CoV of the transmission rates
- Protocols show high instability without background traffic
- With background traffic, CoV is reduced and stability of all protocols is improved
  - But FAST increases CoV slightly with background traffic

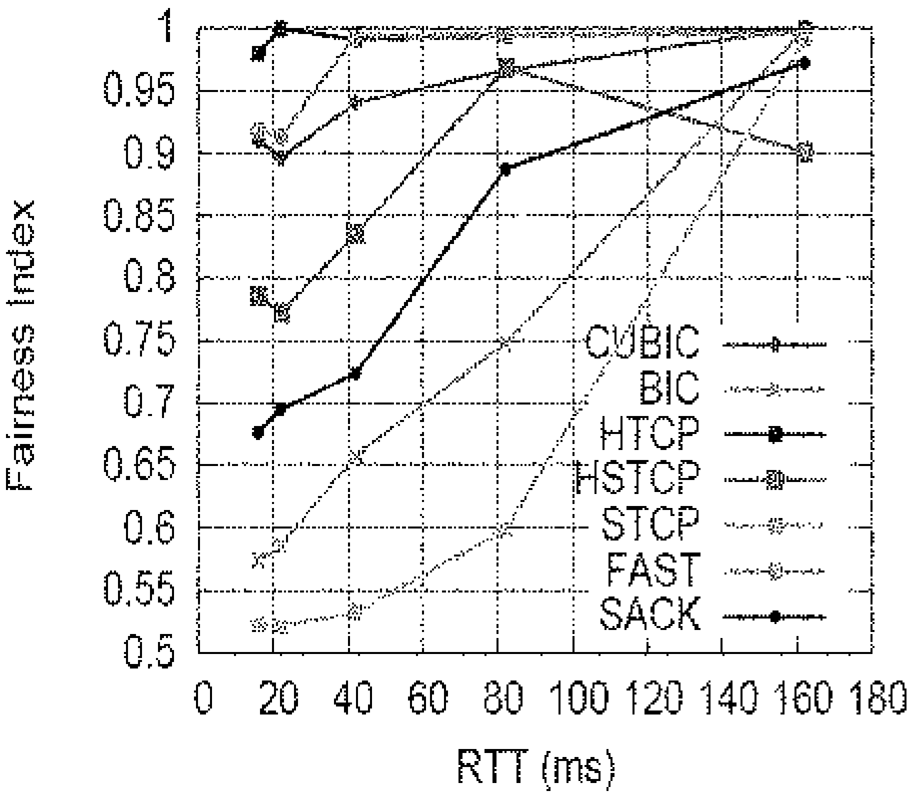
Packet loss rate

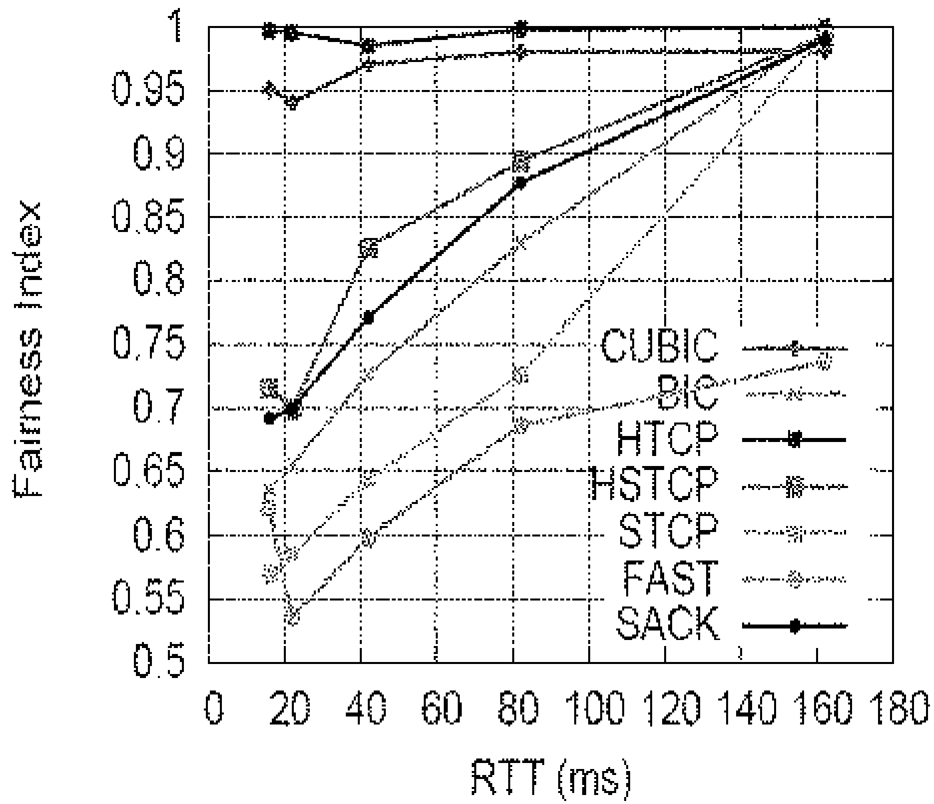




- Loss rate for all traffic measures the impact of high-speed flows on background traffic
- STCP has the highest packet loss rate with and without background traffic
- With background traffic, packet loss rate increases only slightly
  - High-speed flows are “nice” to background traffic

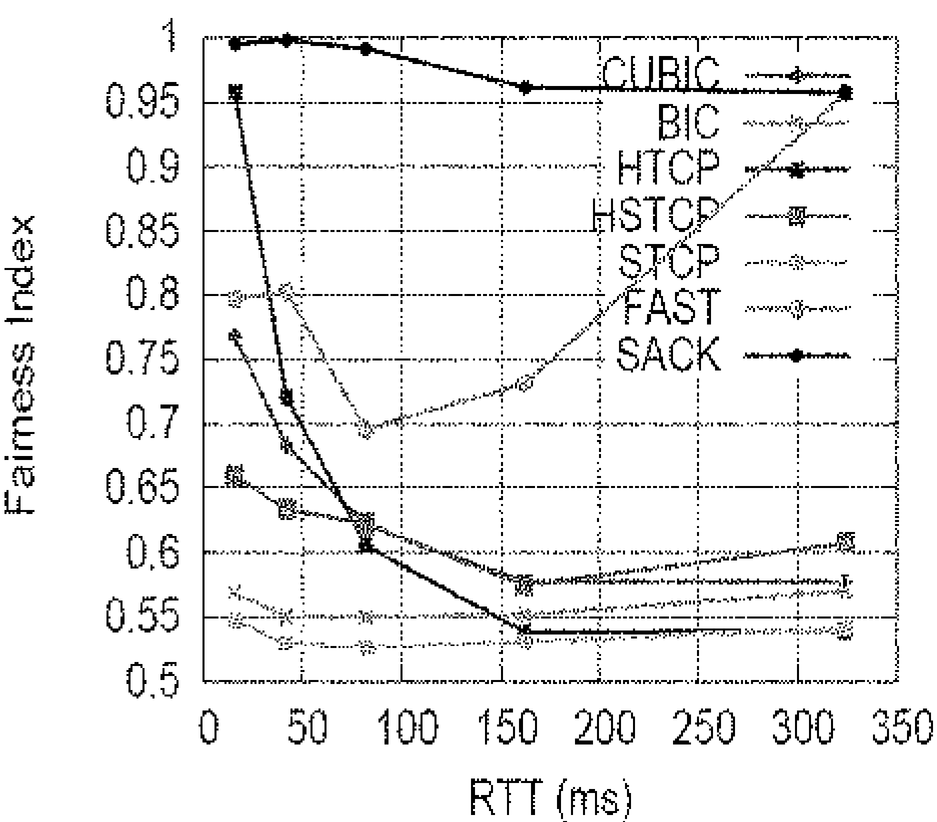
RTT Fairness

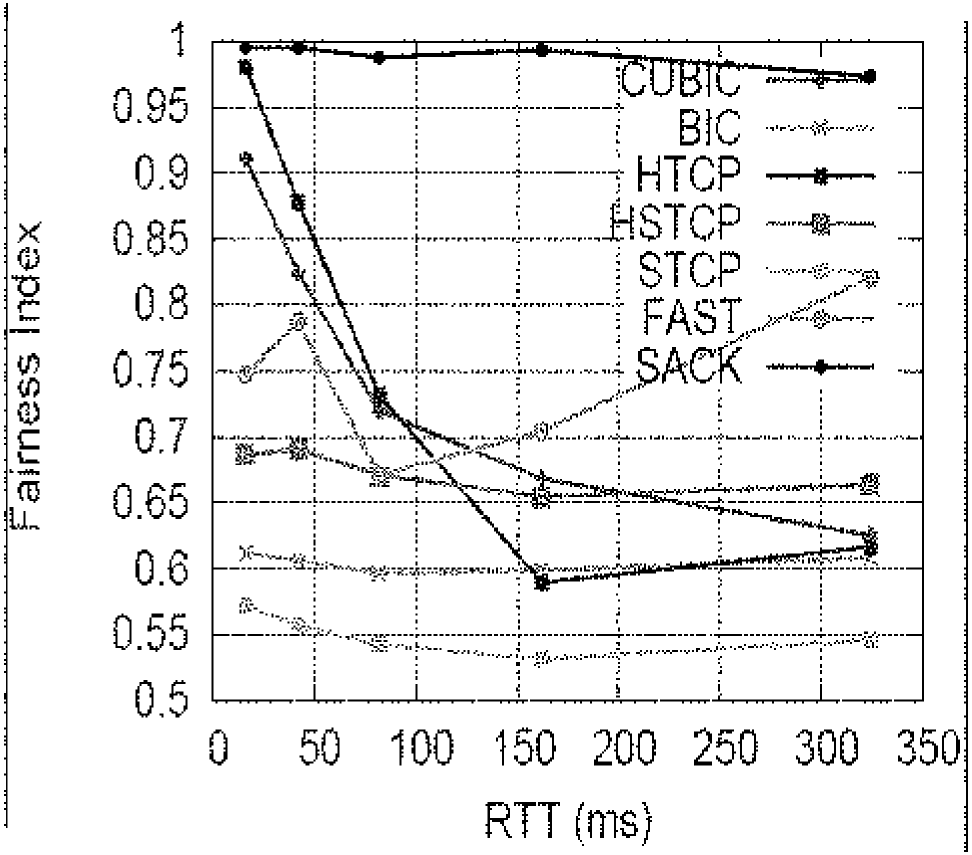




- Run two high-speed flows with different RTTs
  - One flow has a fixed RTT of 162 ms
  - RTT for the other flow varies between 16 and 162 ms
- Without background traffic, FAST shows almost perfect RTT fairness
- With background traffic, FAST has the lowest RTT fairness

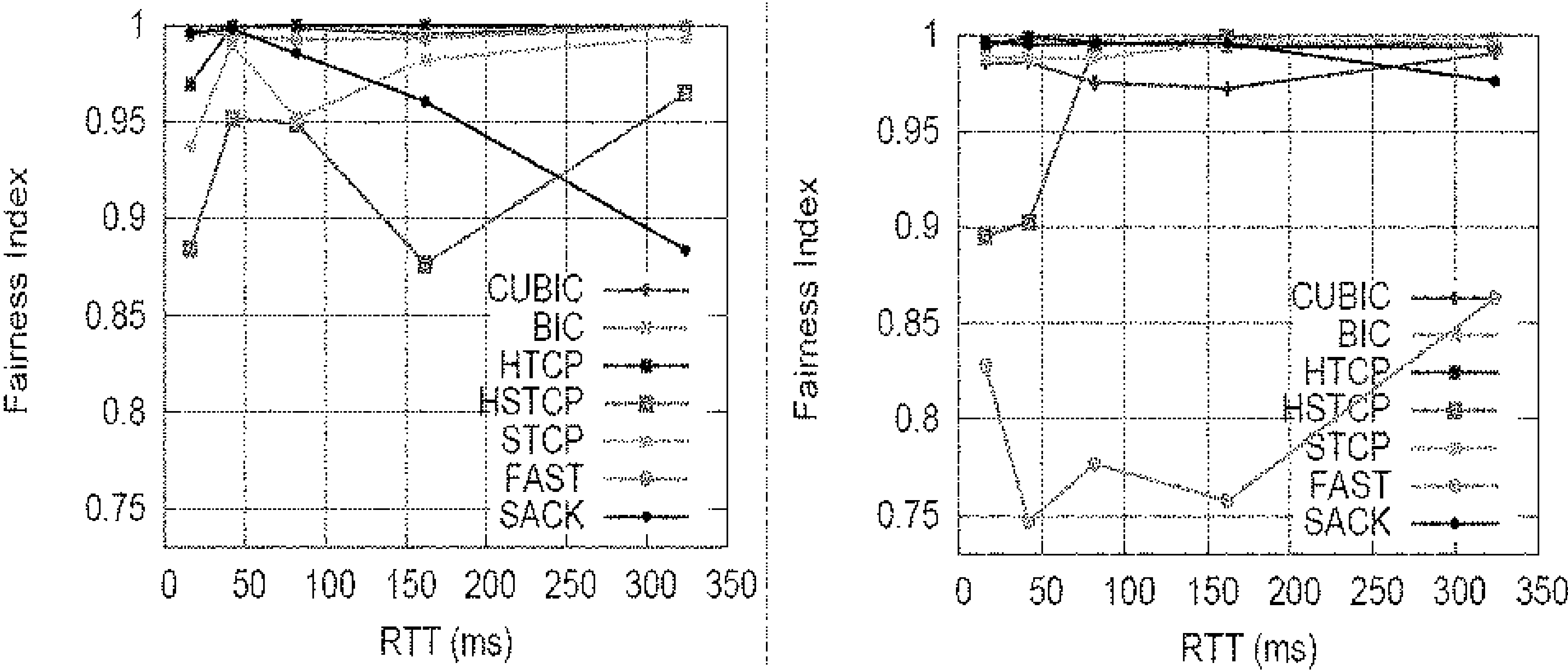
TCP Friendliness





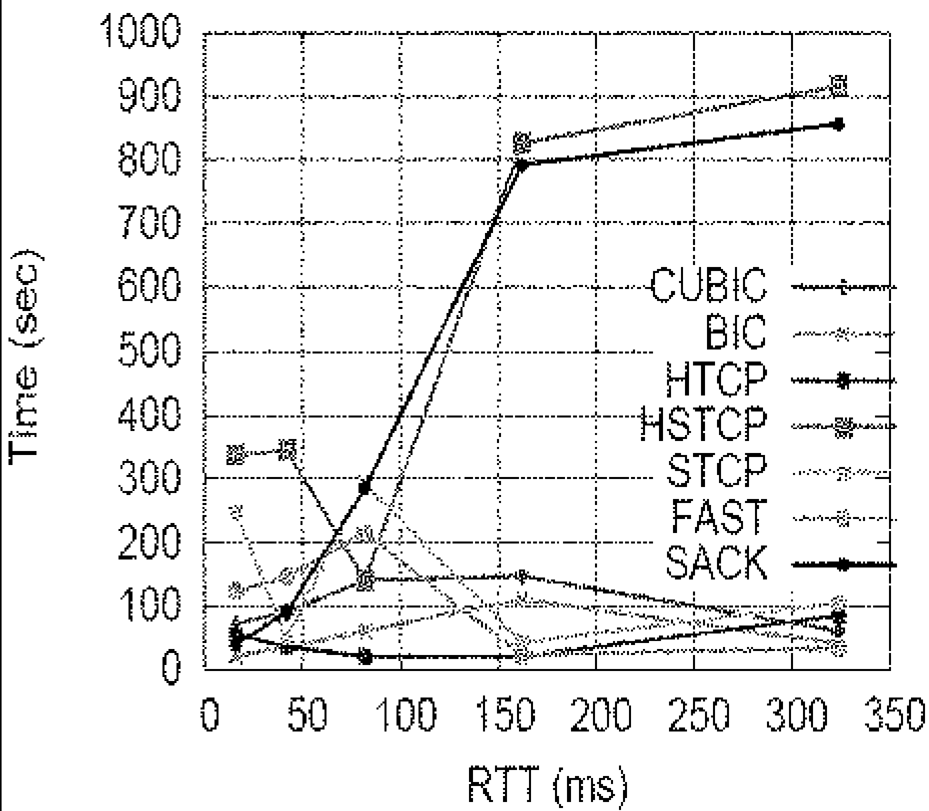
- TCP friendliness is measured as Jain's fairness index
- With background traffic, all protocols increase their TCP friendliness (except for STCP)
  - Background traffic increases randomness in packet loss patterns

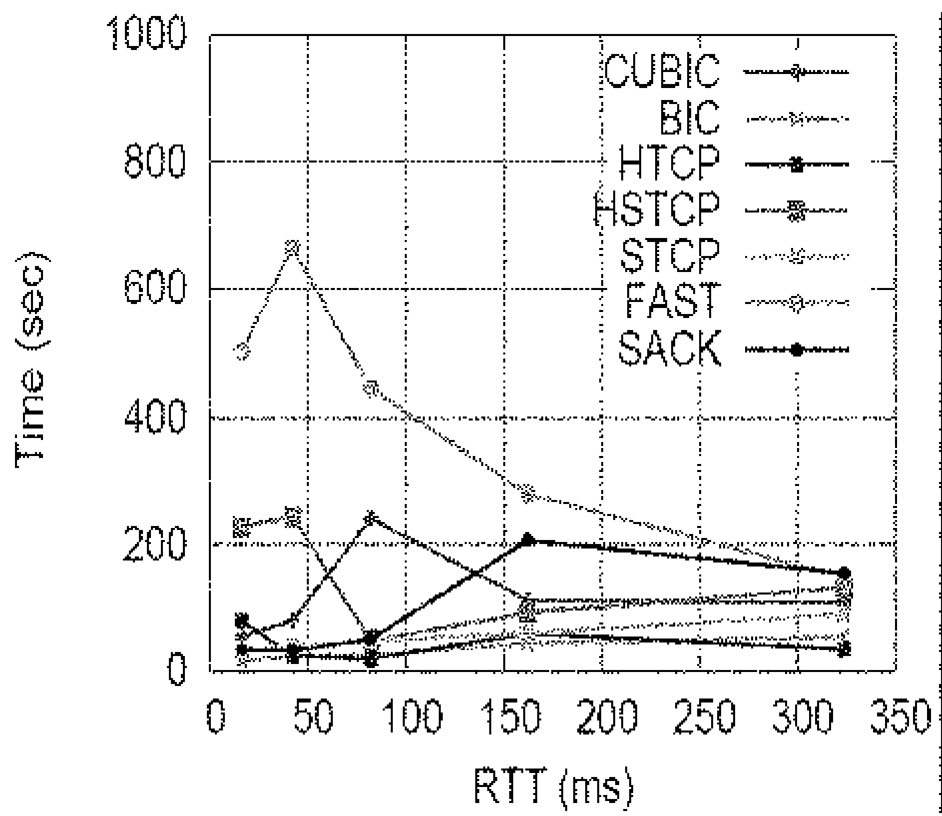
Intra-Protocol Fairness



- Intra-protocol fairness measures how two flows of the same protocol are fair to each other (using Jain's fairness index)
- HSTCP, TCP SACK and STCP show low fairness without background traffic
  - But fairness improves with background traffic
- Fairness for FAST decreases significantly with background traffic

Convergence





- Convergence time: Elapsed time when the second flow reaches 80% throughput of the first flow
- HSTCP shows converge very slowly without background traffic
  - With background traffic, HSTCP improves convergence time
- FAST's convergence time increases significantly with background traffic

Summary and Conclusions

- High-speed TCP variants exhibit rather complex protocol behaviors
- Different conclusions can be drawn from different evaluation scenarios
- Evaluating a new protocol without background traffic may not give the full picture
- Evaluation of a new protocol should use diverse scenarios that involve realistic models for traffic and propagation delays
- Future work will use more realistic traffic generators such as Harpoon and Tmix

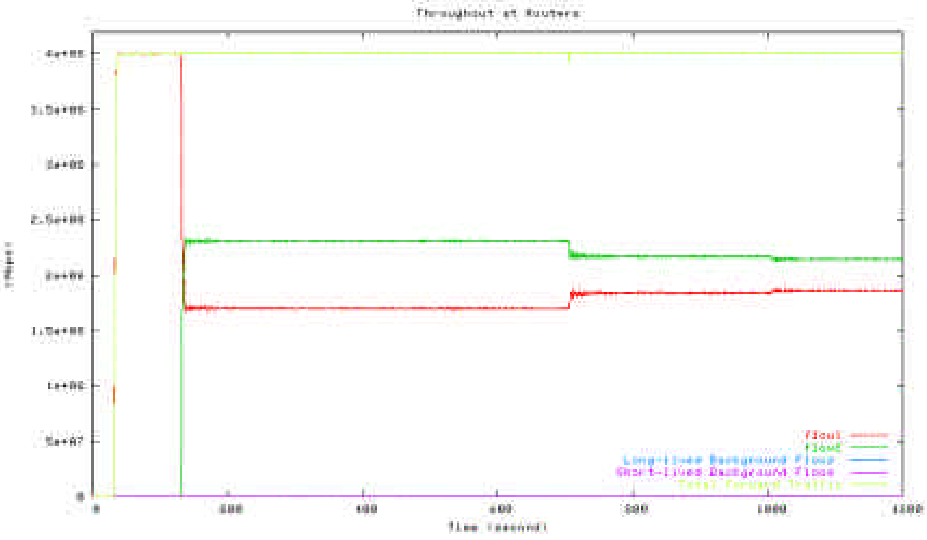
RTTs have exponential distribution

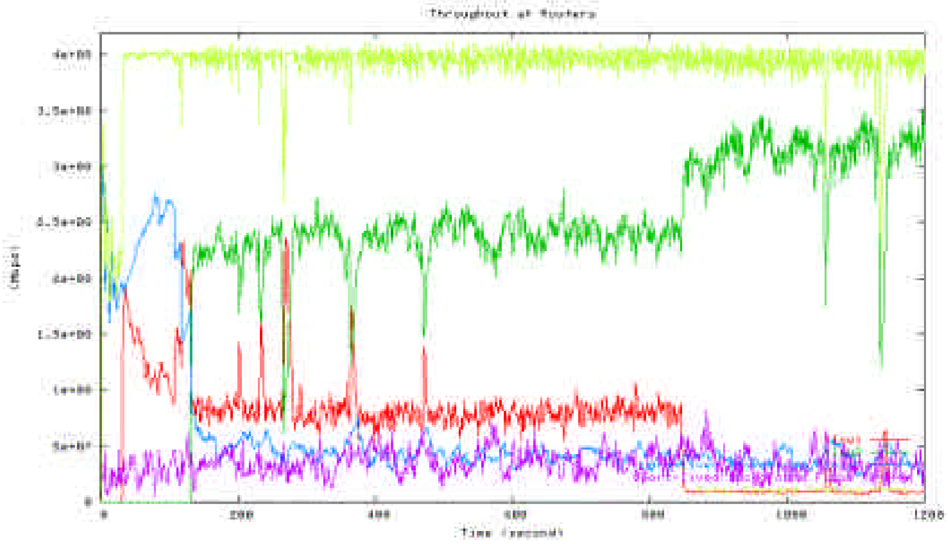
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Utilization

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

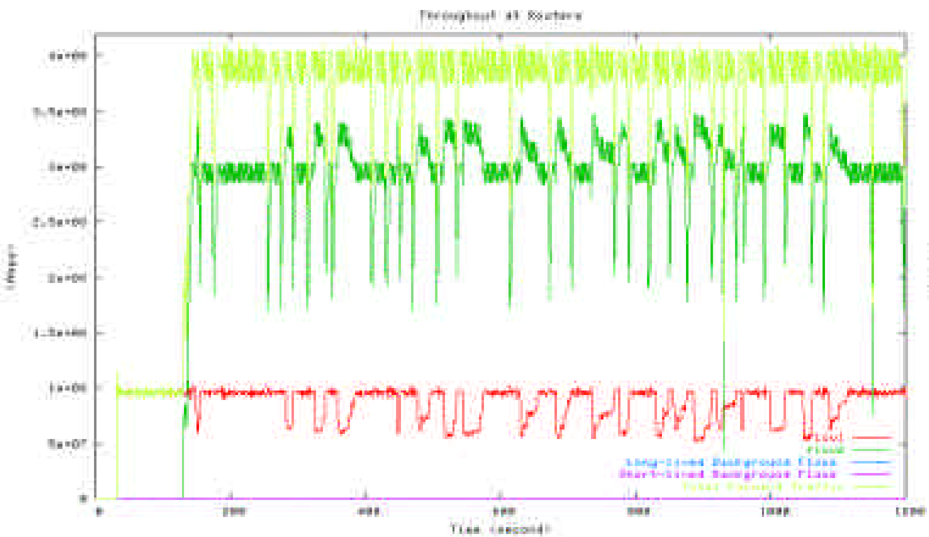
FAST

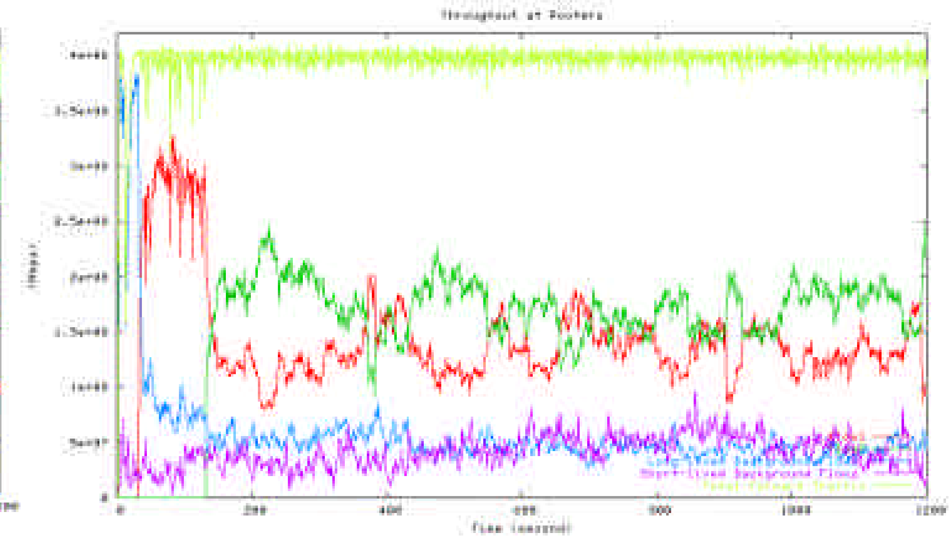




- Throughput of two FAST flows with and without background traffic, RTT=82ms
- Background traffic triggers FAST flows to adapt their congestion windows more frequently
  - But background traffic does not necessarily help FAST flows to converge

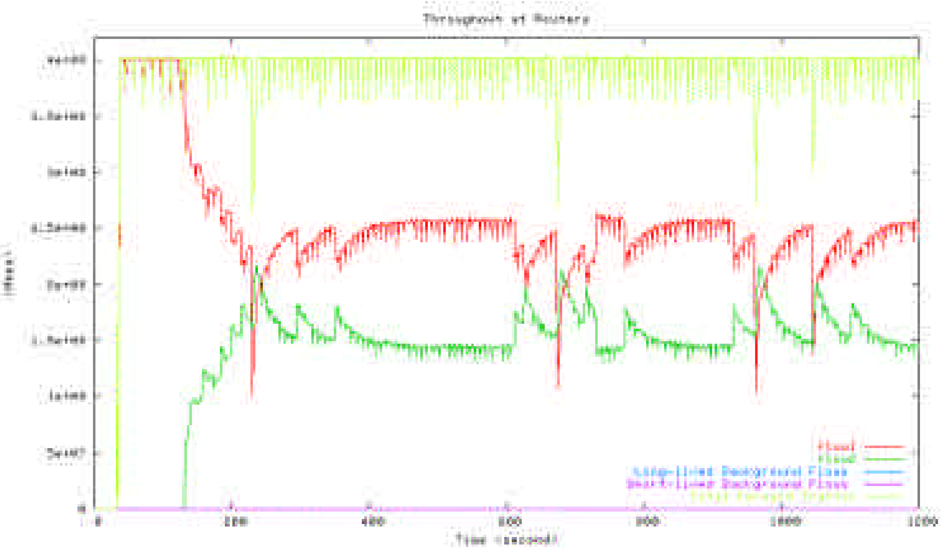
STCP

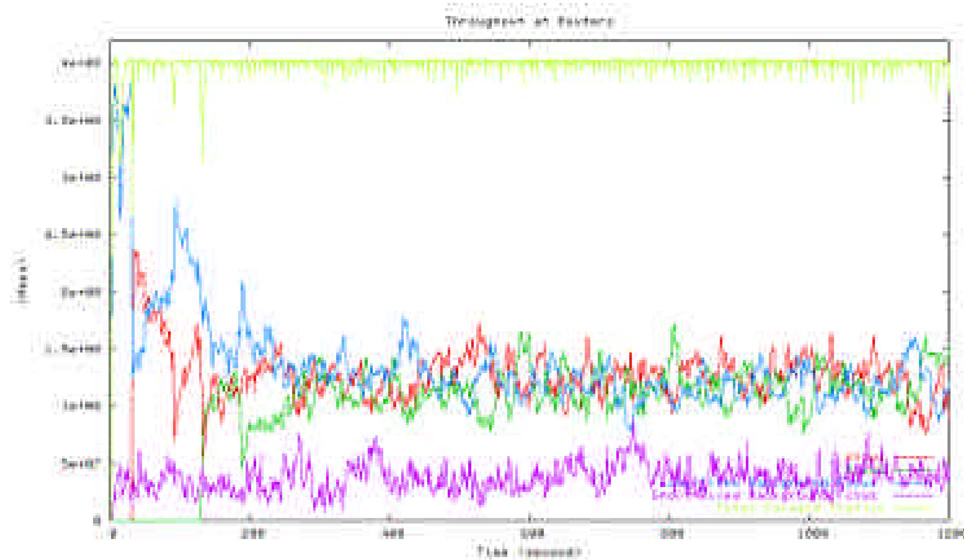




- Throughput of two STCP flows with and without background traffic, RTT=82ms
- Background traffic triggers STCP flows to adapt their congestion windows more frequently
  - Background traffic also helps STCP be fairer to each other

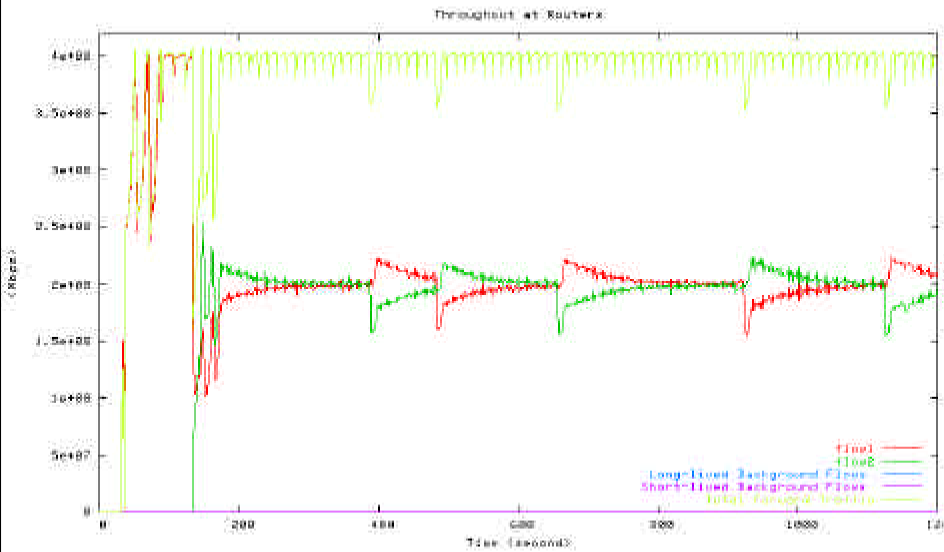
HSTCP

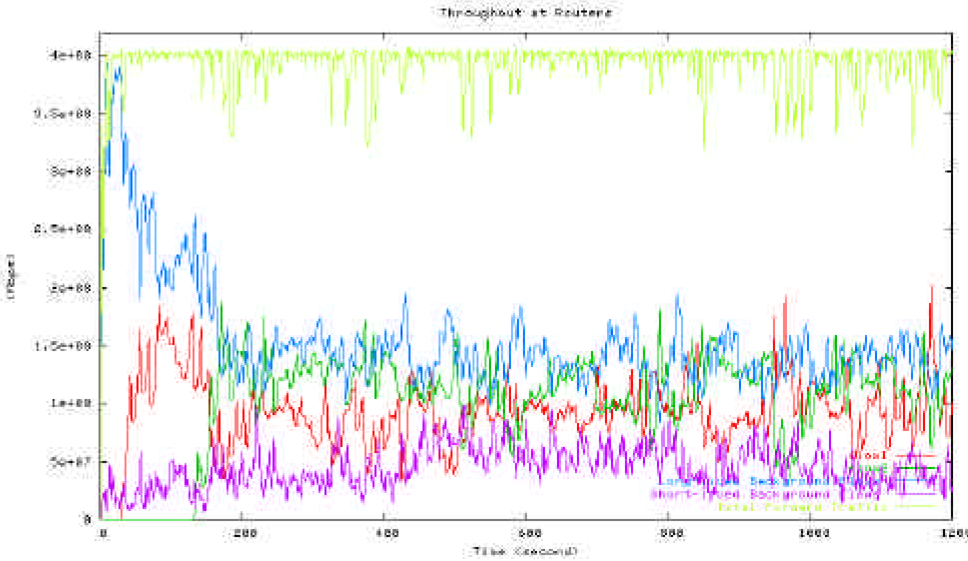




- Throughput of two HSTCP flows with and without background traffic, RTT=82ms
- Background traffic triggers HSTCP flows to adapt their congestion windows more frequently
  - Background traffic also helps HSTCP converge

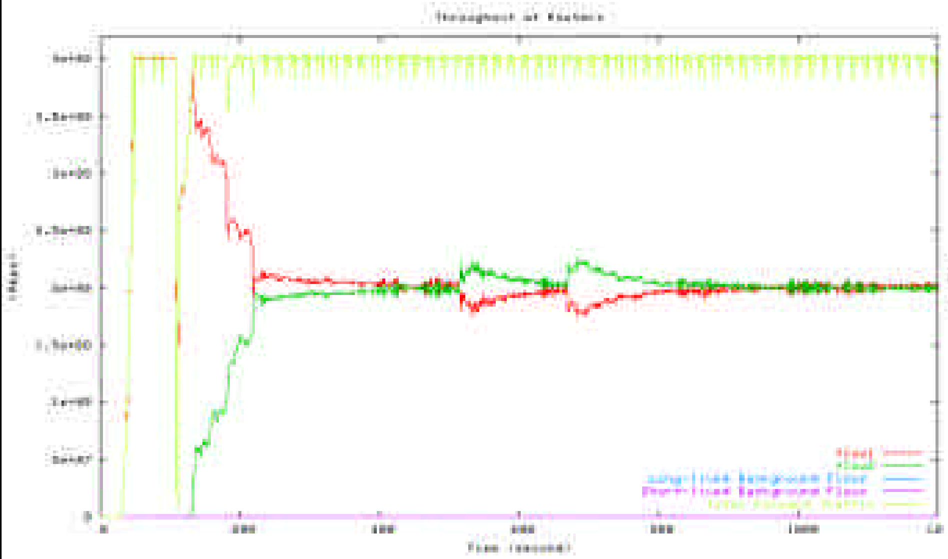
H-TCP

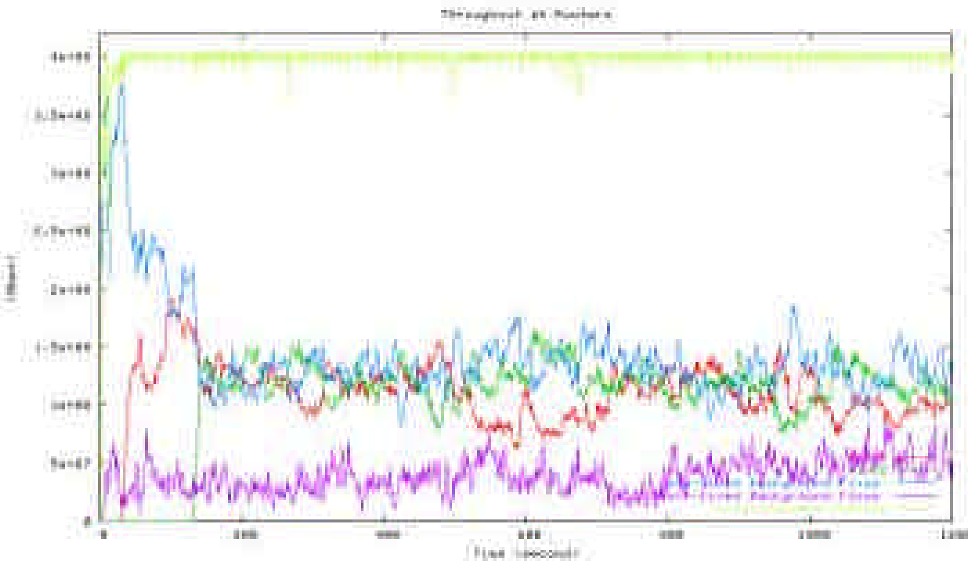




- Throughput of two H-TCP flows with and without background traffic, RTT=82ms
- Background traffic triggers H-TCP flows to adapt their congestion windows more frequently
  - Background traffic also helps H-TCP flows converge faster

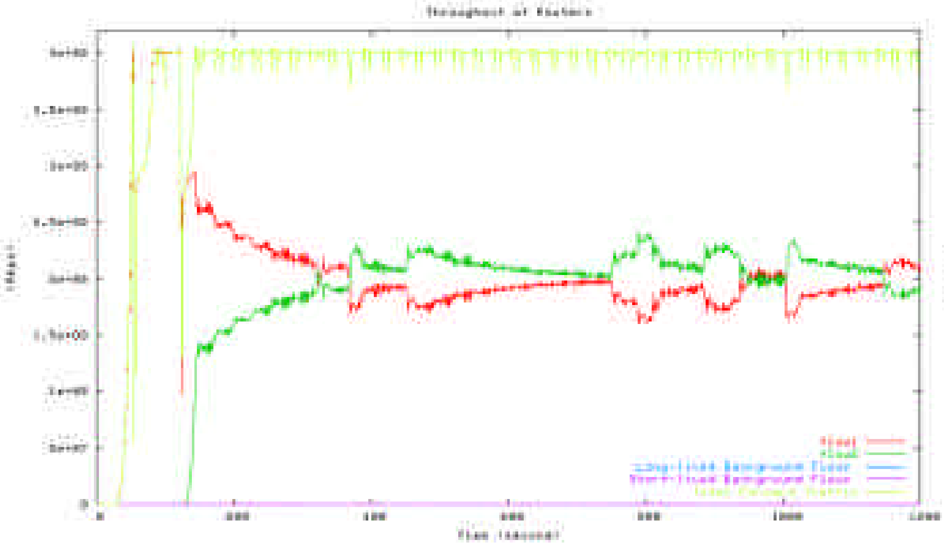
BIC

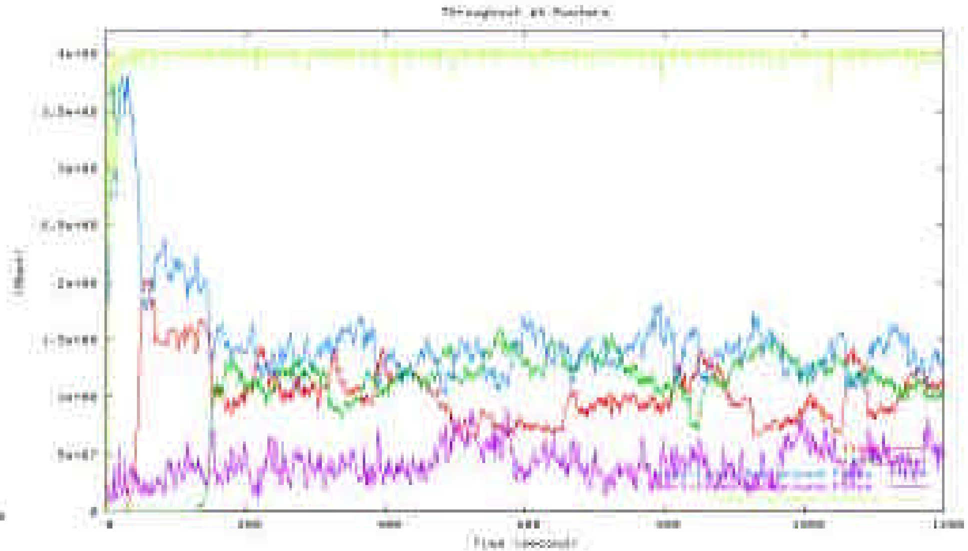




- Throughput of two BIC flows with and without background traffic, RTT=82ms
- Background traffic triggers BIC flows to adapt their congestion windows more frequently
  - Background traffic also helps BIC flows converge faster

CUBIC





- Throughput of two CUBIC flows with and without background traffic, RTT=82ms
- Background traffic triggers CUBIC flows to adapt their congestion windows more frequently
    - Background traffic also helps CUBIC converge faster