Exponential TCP (EXP-TCP)

Decoupling End-to-End Efficiency and Fairness Control in High BDP Networks

Shudong Jin
Case Western Reserve University

with Dan Liu

PFLDnet workshop, Marina Del Rey, CA

February 8, 2007

Motivation

- Existing TCP variants for high BDP networks
  - HSTCP, STCP, BIC TCP, H-TCP, etc
  - FAST TCP
  - XCP, VCP
- Decoupling efficiency and fairness control
  - Proposals such as HSTCP and STCP have failed in.
  - Why? Good efficiency requires high aggressiveness in high BDP networks, but “rich-gets-richer” among competing flows (resulting in poor fairness)
- Observations
  - To achieve good efficiency, we need more aggressive increase (what we care is the **absolute** increase, e.g., exponential)
  - To achieve good fairness, we need to set the **relative** increase rate of competing flows appropriately

Design

- Multiplicative-decrease

$$cwnd \leftarrow (1-\beta) \times cwnd,$$

  - Set β to a small value 1/8, resulting in a moderate decrease and often high network utilization.

- Exponential-increase (not multiplicative-increase), on each ACK

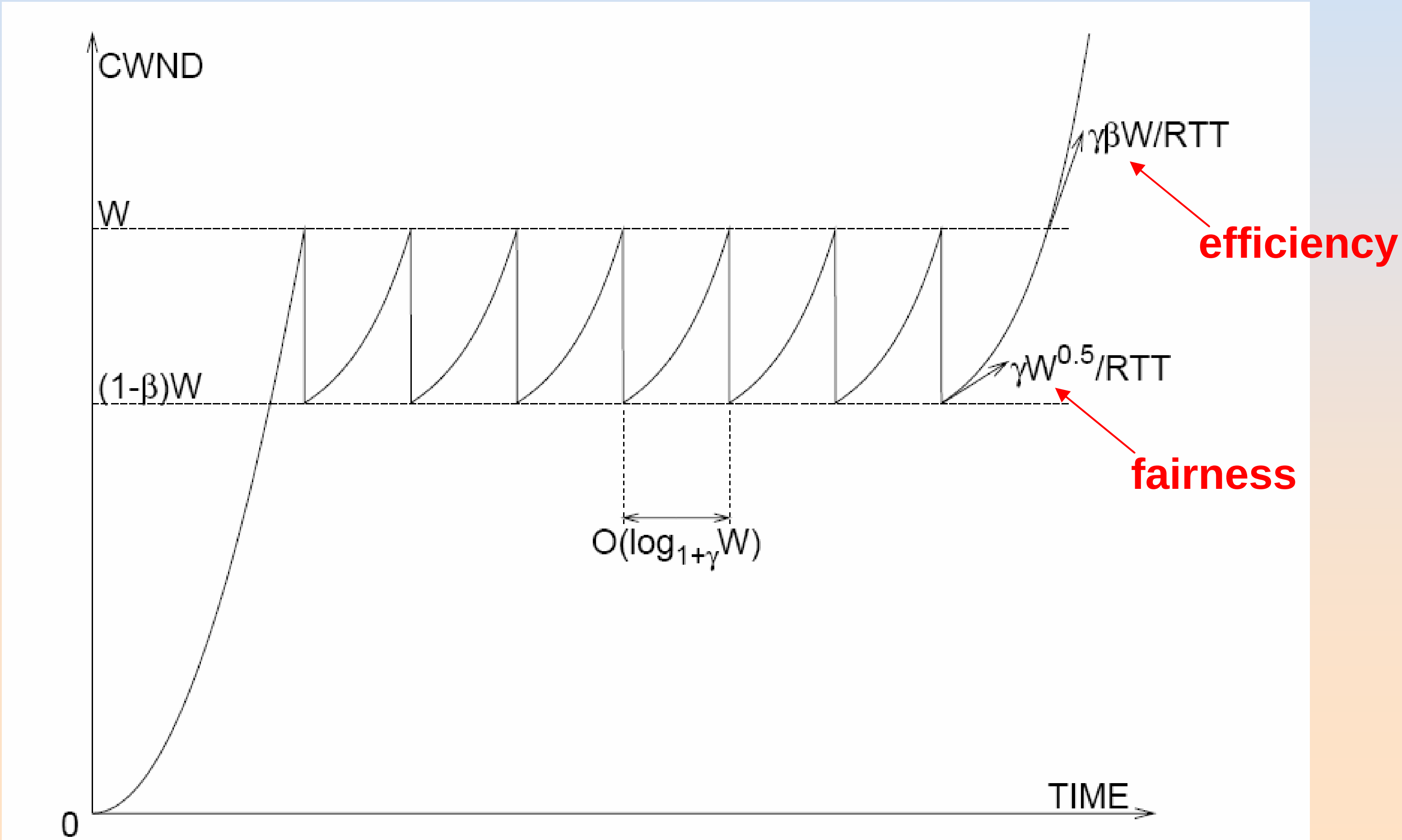
$$cwnd \leftarrow cwnd + \gamma \left(1 - \frac{cwnd_0}{cwnd} + \frac{\sqrt{cwnd_0}}{cwnd}\right),$$

  - Set γ, which controls the rate of exponential increase, to a small value. The value of $cwnd_0$ is the congestion window size just after the last decrease.

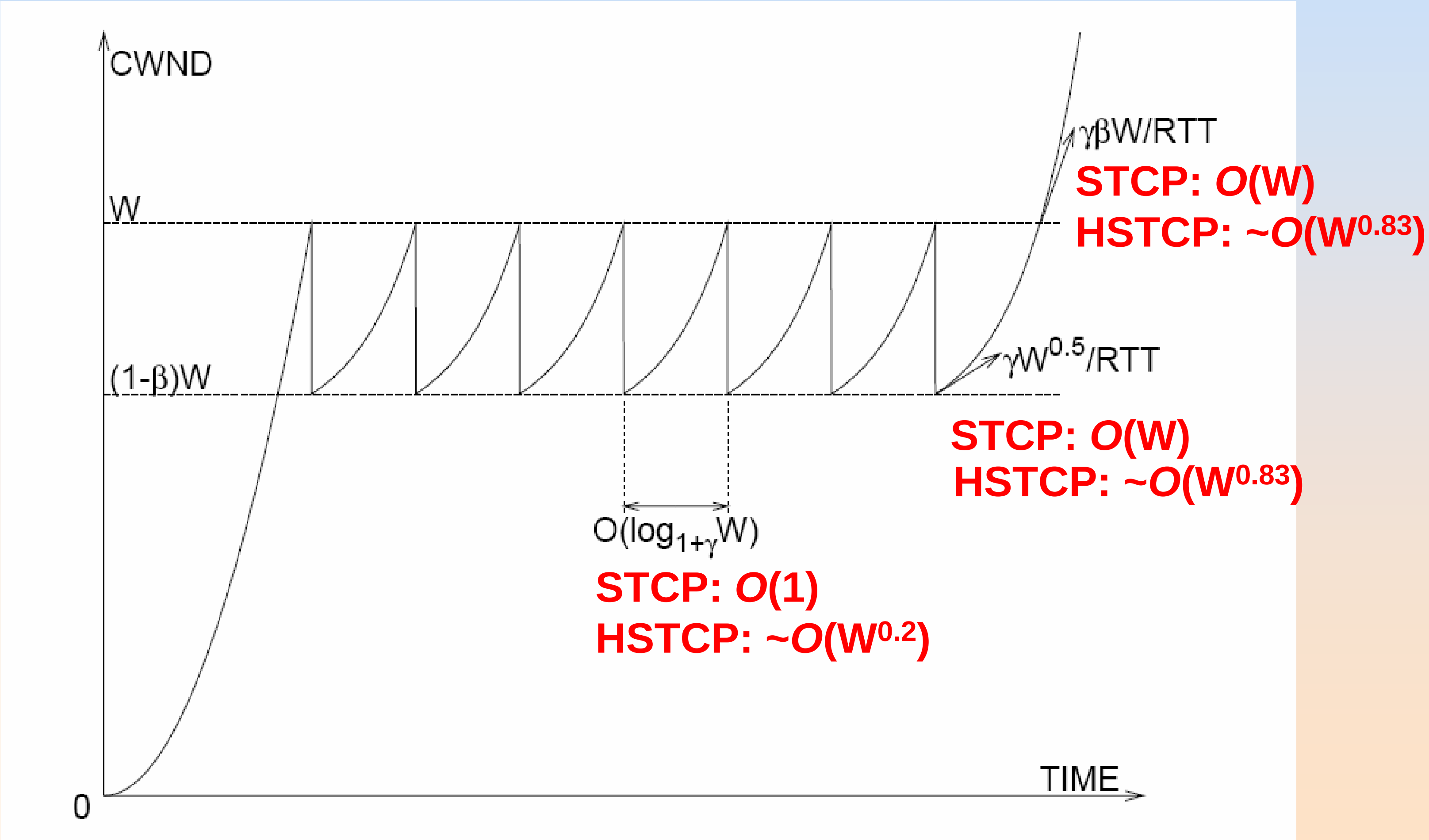
- Simplicity (as simple as AIMD)
  - Only two parameters: β and γ.
  - Quick to increment $cwnd$: $cwnd_0$ is a constant until next decrease

Explanation

- To achieve efficiency: absolute increase is exponential
- To achieve fairness: relative increase is ~ \sqrt{cwnd}



Comparison


CWND
W
(1-β)W
0
TIME
γβW/RTT
STCP: O(W)
HSTCP: ~O(W0.83)
γW0.5/RTT
STCP: O(W)
HSTCP: ~O(W0.83)
O(log1+γW)
STCP: O(1)
HSTCP: ~O(W0.2)

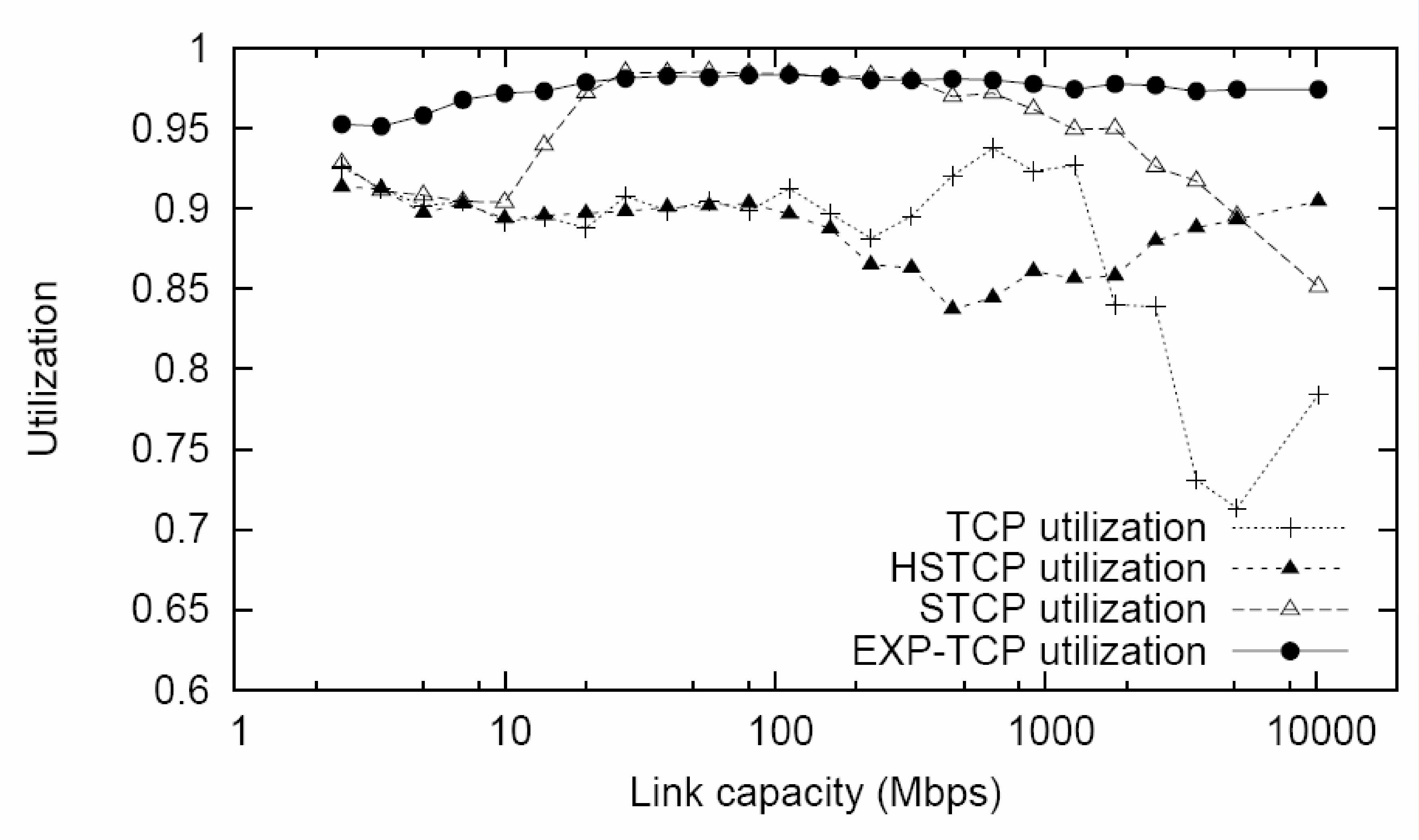
Simulation

- Network and traffic
  - ns-2 simulation, with a simple dumbbell network
  - two-way traffic, and saturated reverse path (the pressure of ACK compression).
  - Capacity, number of flows, mixed long flows and Web traffic, etc.
  - different propagation delays to eliminate artificial synchronization.

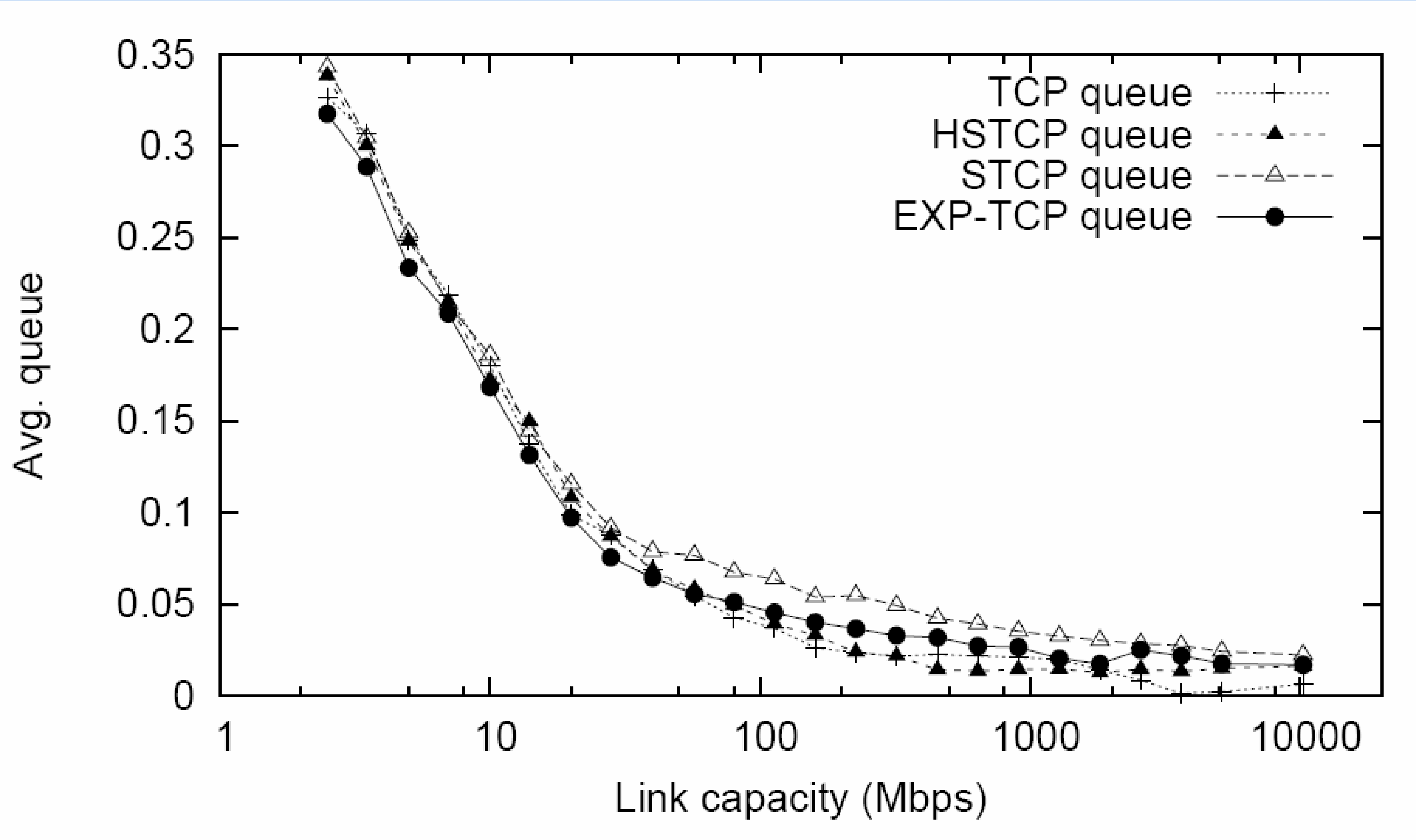
- Queue settings
  - Bottleneck queue size is always set to BDP.
  - RED queues on the bottleneck in most simulations. Standard parameters: min_thresh =0.1*BDP, max_thresh =0.3*BDP, q_weight =0.002, max_p=0.1, gentle =ON.
  - ECN bits, although the performance metrics except loss rate do not change much.

- Compared protocols
  - Standard TCP, HSTCP, STCP, and EXP-TCP (γ=0.05 and β=0.125)
  - SACK1 variant

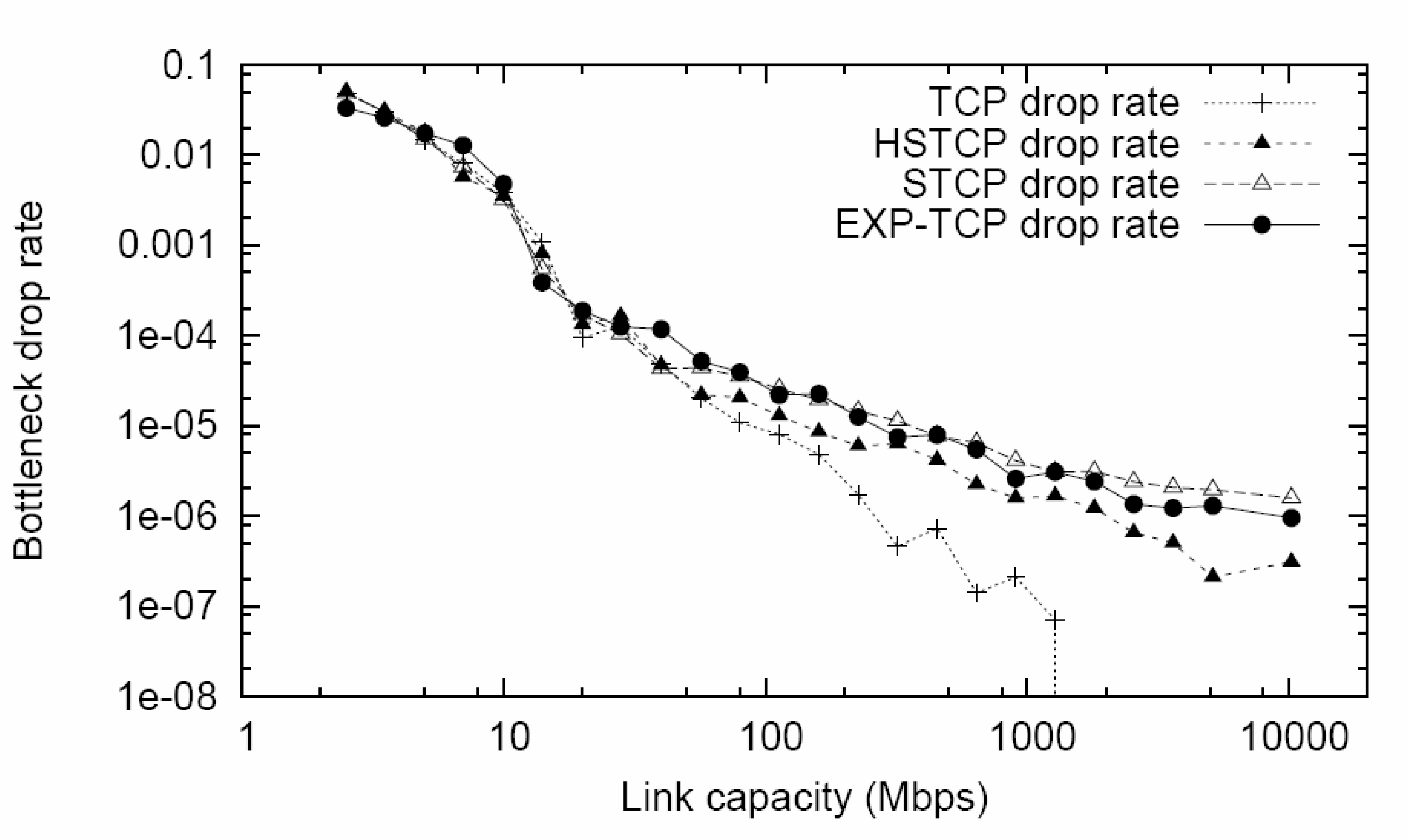
- RED queues at bottleneck (2.5Mbps to 10Gbps)
- 16 flows in each direction, variable propagation delays (60-100ms)
- Each simulation run lasts 120 seconds
- Average utilization over the last 100 seconds



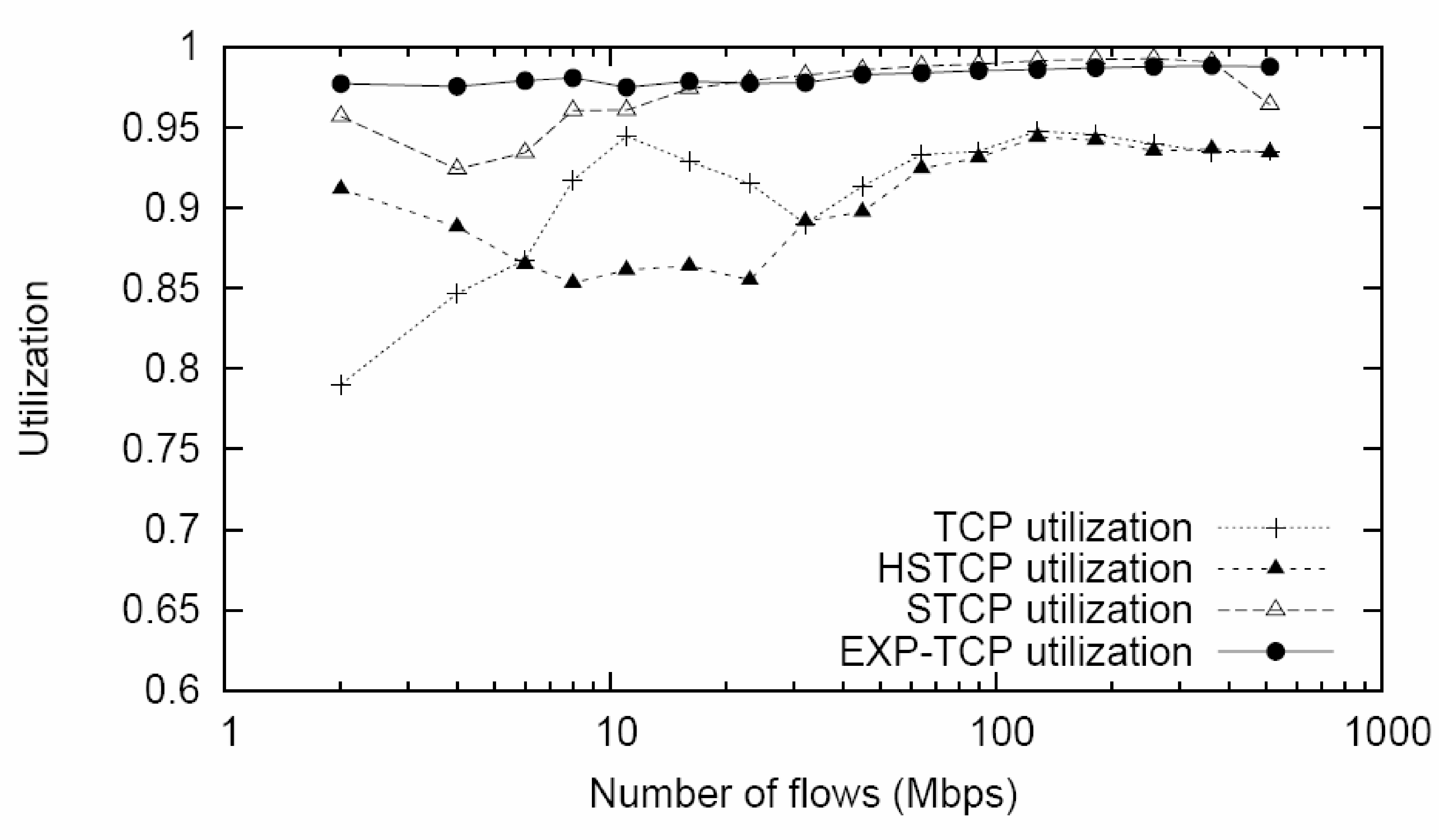
- RED queues at bottleneck (2.5Mbps to 10Gbps)
- 16 flows in each direction, variable propagation delays (60-100ms)
- Each simulation run lasts 120 seconds
- Average queue length over the last 100 seconds



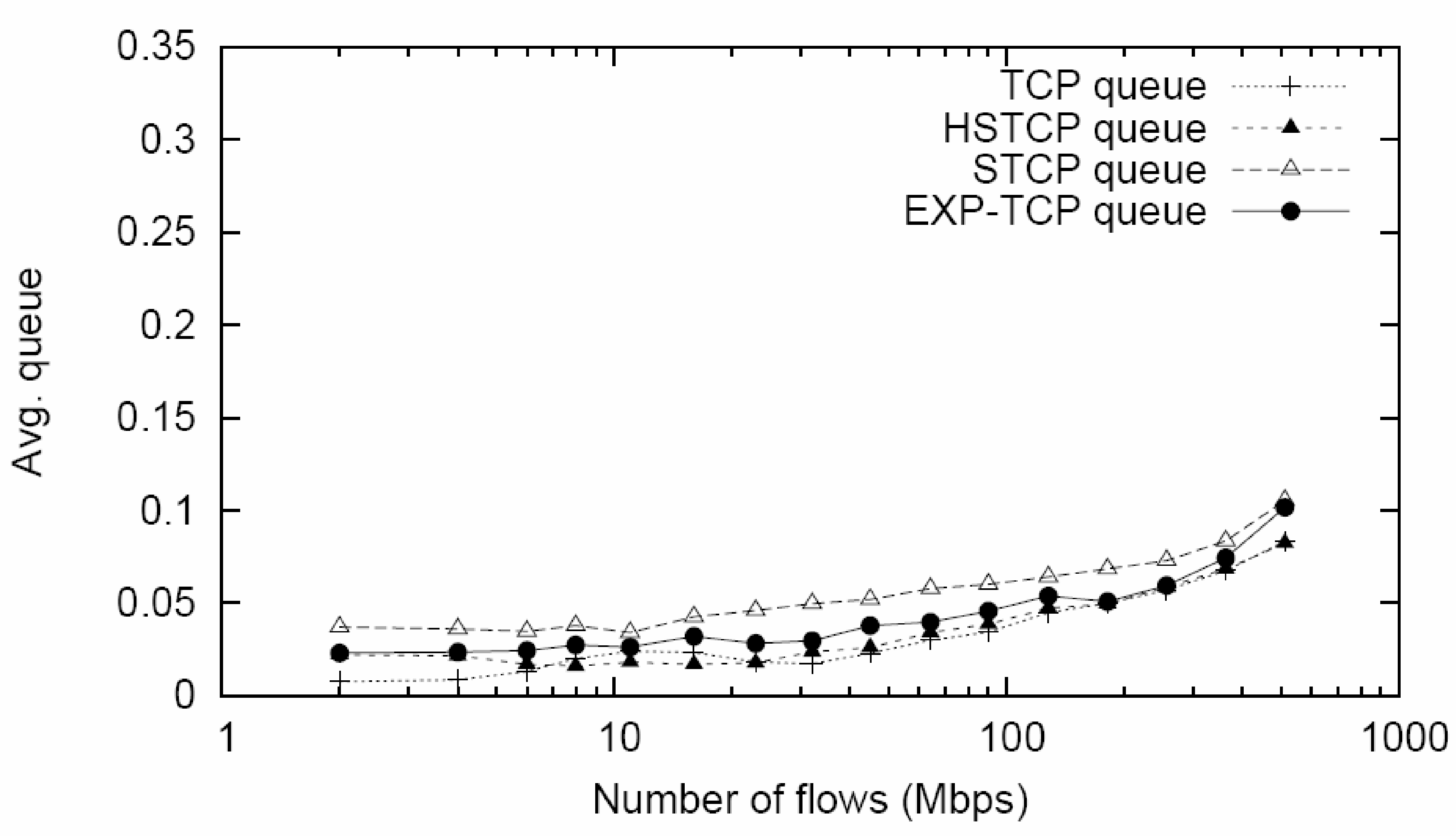
- RED queues at bottleneck (2.5Mbps to 10Gbps)
- 16 flows in each direction, variable propagation delays (60-100ms)
- Each simulation run lasts 120 seconds
- Average drop rate over the last 100 seconds



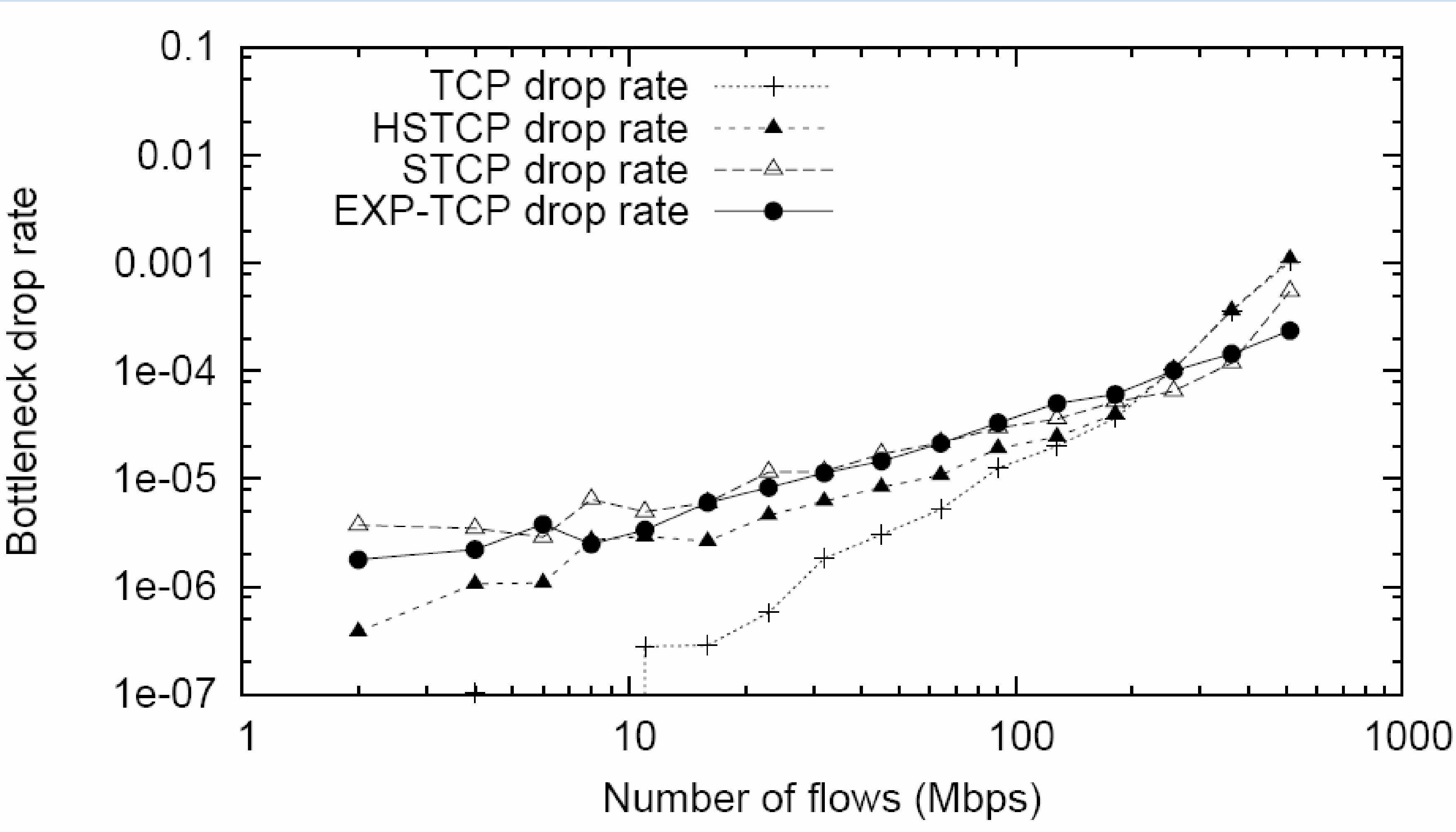
- RED queues at bottleneck (500Mbps)
- up to 512 flows in each direction, variable propagation delays
- Each simulation run lasts 120 seconds
- Average utilization over the last 100 seconds



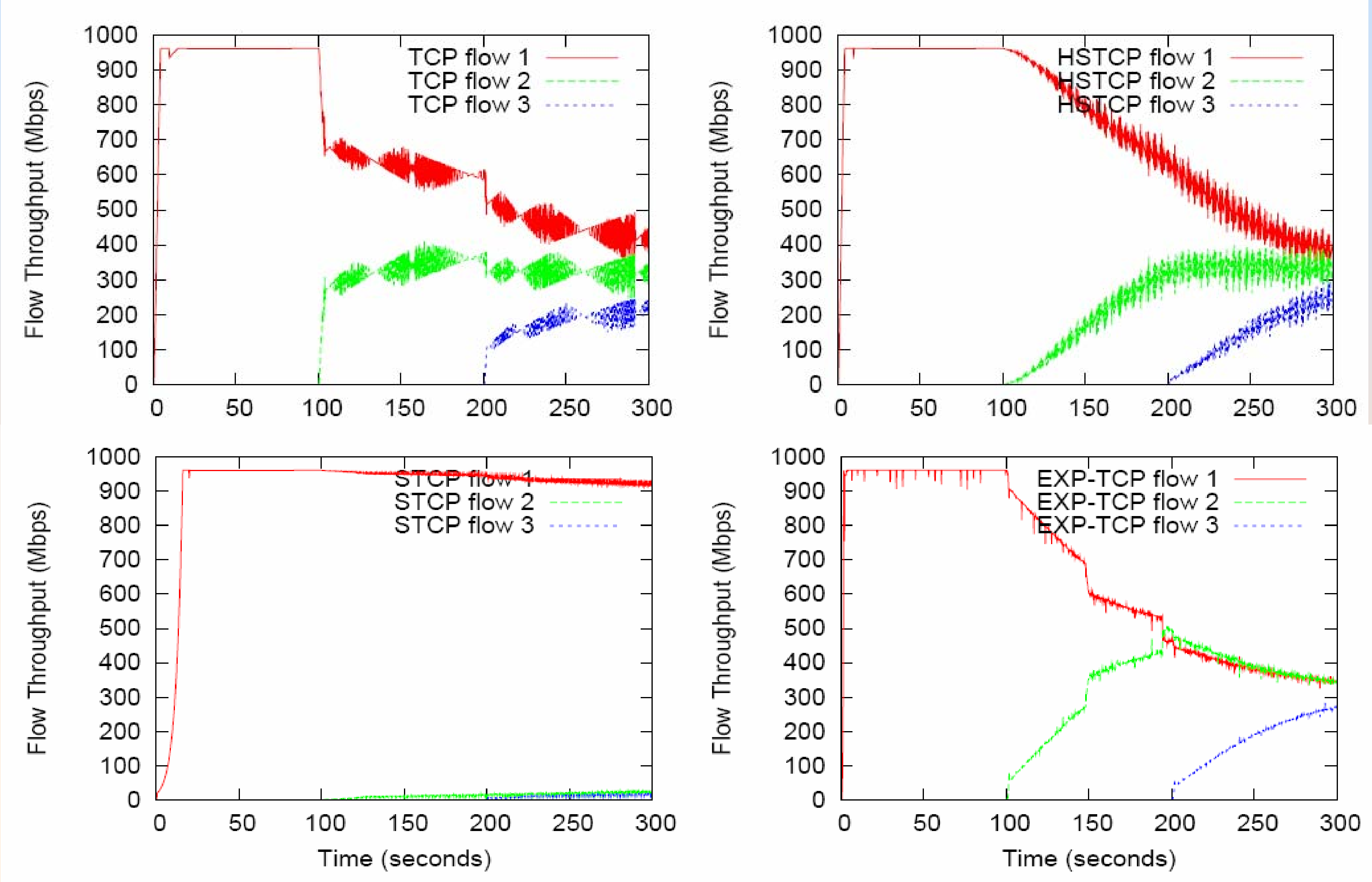
- RED queues at bottleneck (500Mbps)
- up to 512 flows in each direction, variable propagation delays
- Each simulation run lasts 120 seconds
- Average queue length over the last 100 seconds



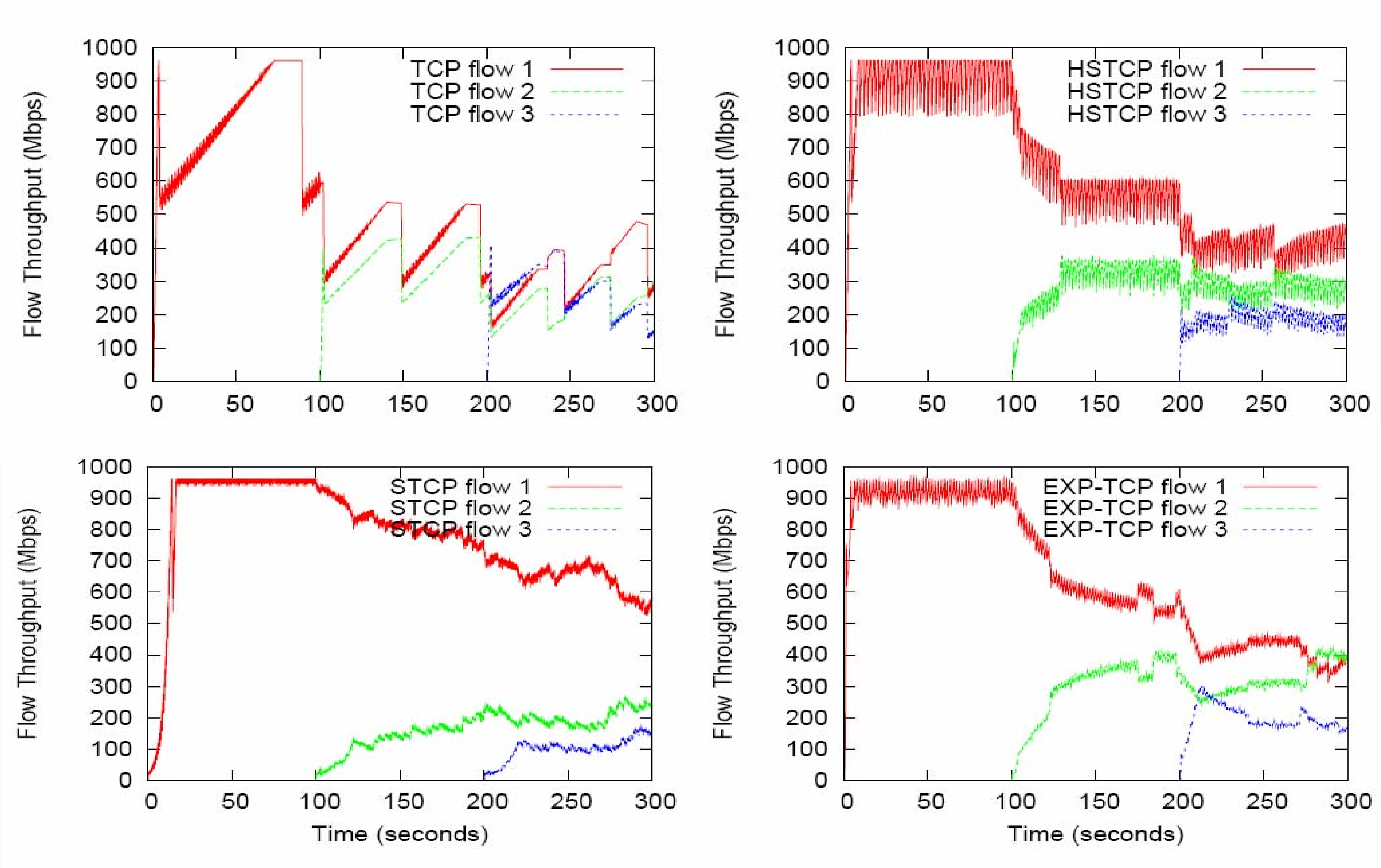
- RED queues at bottleneck (500Mbps)
- up to 512 flows in each direction, variable propagation delays
- Each simulation run lasts 120 seconds
- Average drop rate over the last 100 seconds



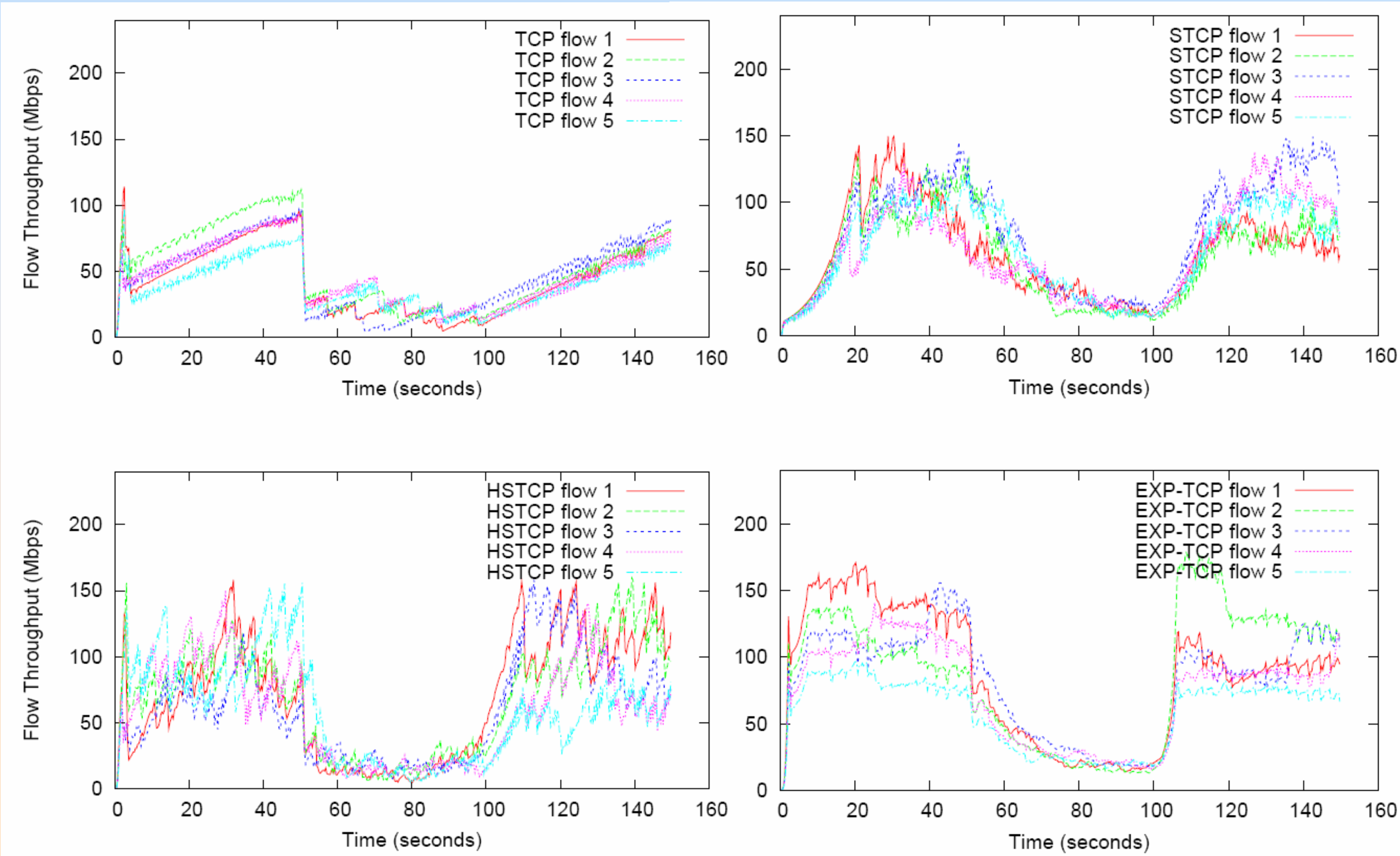
- DropTail queues at bottleneck (1Gbps)
- 3 homogenous flows in one direction, 40ms propagation delays
- The flows start at time 0s, 100s, and 200s, respectively



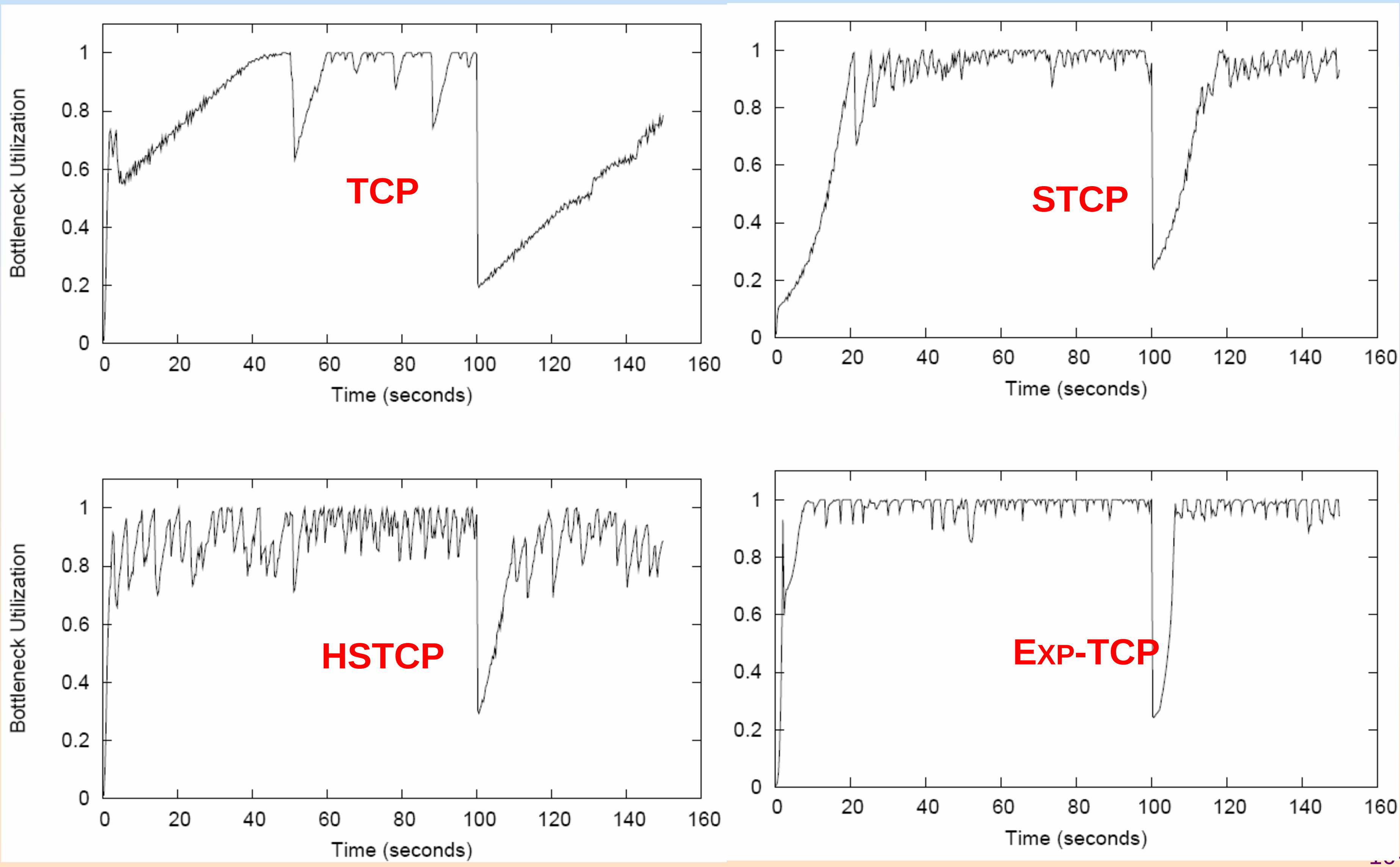
- RED queues at bottleneck (1Gbps)
- 3 flows in one direction, slightly variable propagation delays
- The flows start at time 0s, 100s, and 200s, respectively



- RED queues at bottleneck (1Gbps)
- 10 flows in each direction, joined by 40 more in time [50,100]



- RED queues at bottleneck (1Gbps)
- 10 flows in each direction, joined by 40 more in time [50,100]



Conclusion

- It is possible to decouple efficiency and fairness control in end-to-end congestion control algorithms
- Future work
  - More complex network configurations, multiple bottleneck, RTTs
  - RTT fairness
  - Comprehensive comparisons with other end-to-end algorithms, including BIC TCP, H-TCP, FAST TCP, etc
- Simulation code (ns-2 modification, Tcl)