Implementation Experiments on HighSpeed and Parallel TCP

Zongsheng Zhang

Go Hasegawa

Masayuki Murata

Osaka University

Outline

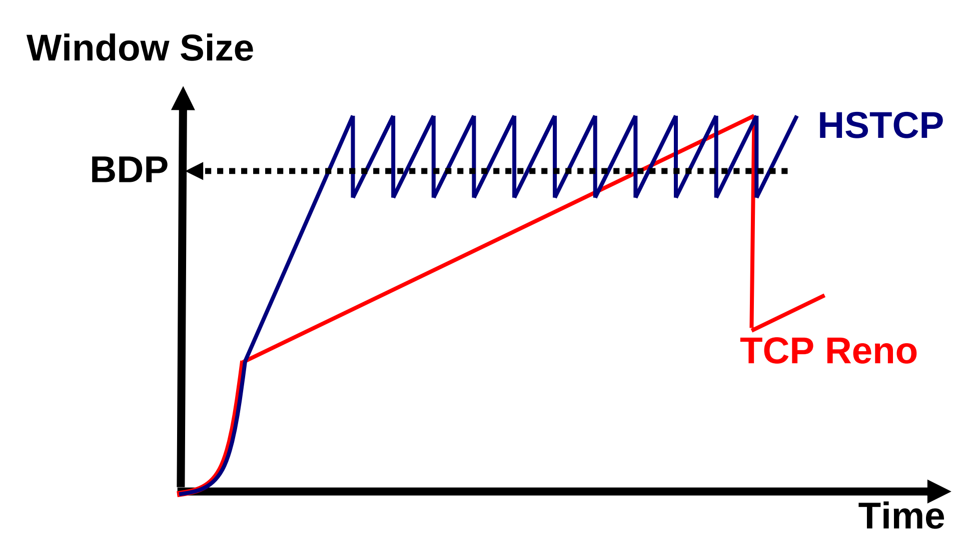
- Introduction
- Background of HSTCP and gHSTCP
- Why to evaluate in a test-bed network
- A refined algorithm of gHSTCP
- Results
- Conclusion & future work

Introduction

- What's wrong with TCP?
  - TCP was designed when T1 was a fast network.
  - It doesn't perform well under high bandwidth and high delay networks (LFNs) because of congestion window (CWND) algorithms.
- Solutions:
  - One traditional method: parallel TCP mechanism
  - New method: New algorithms for updating CWND, such as HighSpeed TCP, Scalable TCP, FAST TCP.

HighSpeed TCP [3] (HSTCP)

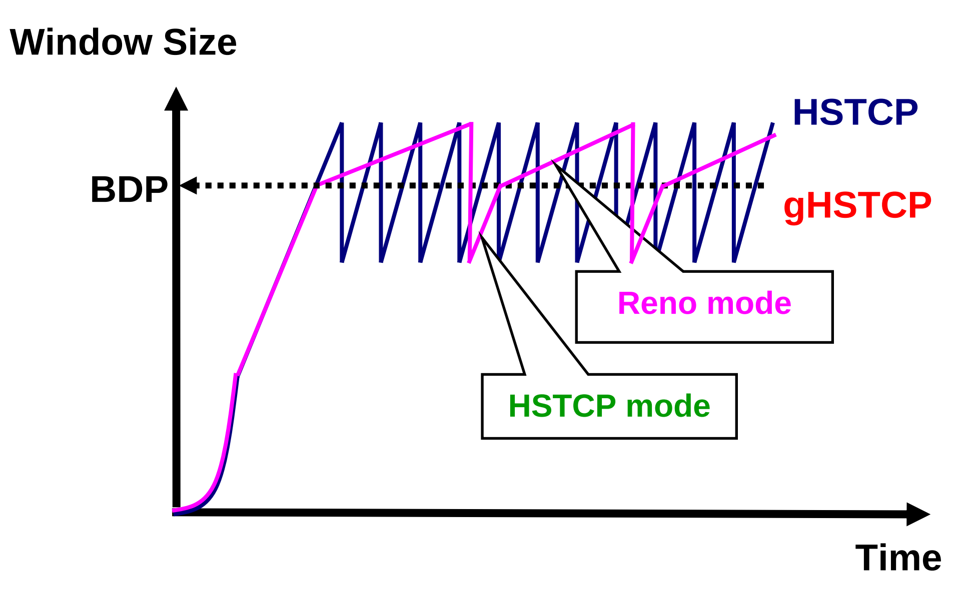
- HSTCP: a simple and representative example.
- It uses the Additive Increase and Multiplicative Decrease (AIMD) principle of TCP Reno.
- It is easily deployed in the current Internet.
- Currently, HSTCP is the only protocol recommended by *IETF* for LFNs.
- However, unfairness is a drawback.



[3] S. Floyd, "HighSpeed TCP for large congestion windows," *RFC 3649*, December 2003.

gentle HighSpeed TCP [1] (gHSTCP)

- gHSTCP is based on HSTCP.
- Based on observation of the packet transmission time and its RTT, 2 modes in congestion avoidance phase:
  - the positive correlation → Reno mode
  - otherwise → HSTCP Mode



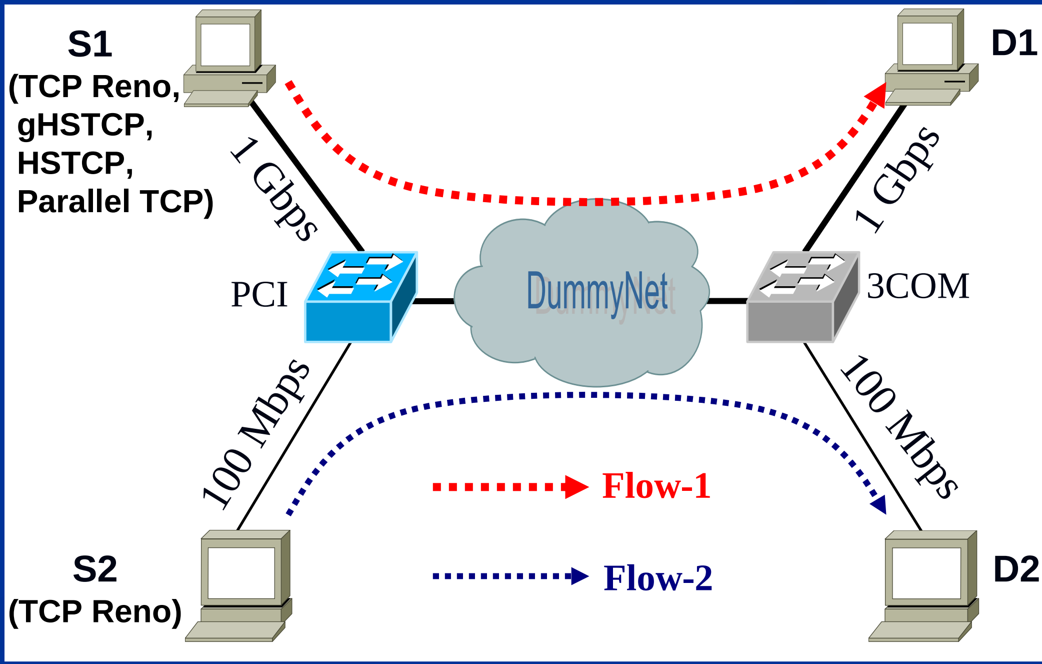
[1] Z. Zhang, G. Hasegawa, and M. Murata, "Performance analysis and improvement of HighSpeed TCP with TailDrop/RED routers," *Proc. of MASCOTS*, October 2004.

Simulation vs. Emulation (real network)

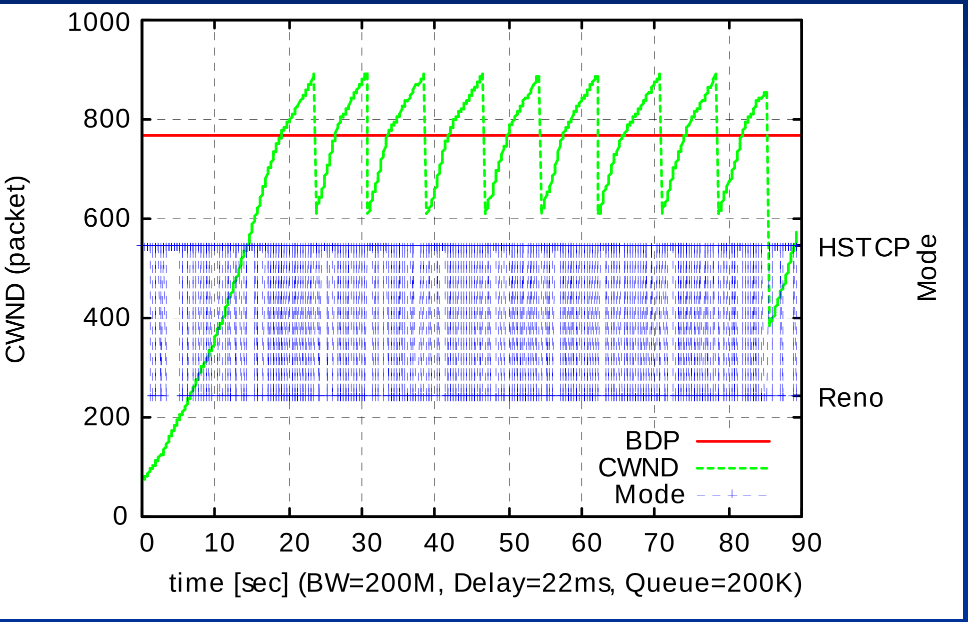
- The simulation condition is relatively ideal compared to the real network.
- gHSTCP is evaluated only by simulation [1].
- Is it suitable for the real network?
  - The heterogeneity of the real network, such as individual links, network equipments, protocols and applications.
  - The emulation network is more similar to a real network.
  - For applying in the real network, it is necessary to evaluate gHSTCP in test-bed network.

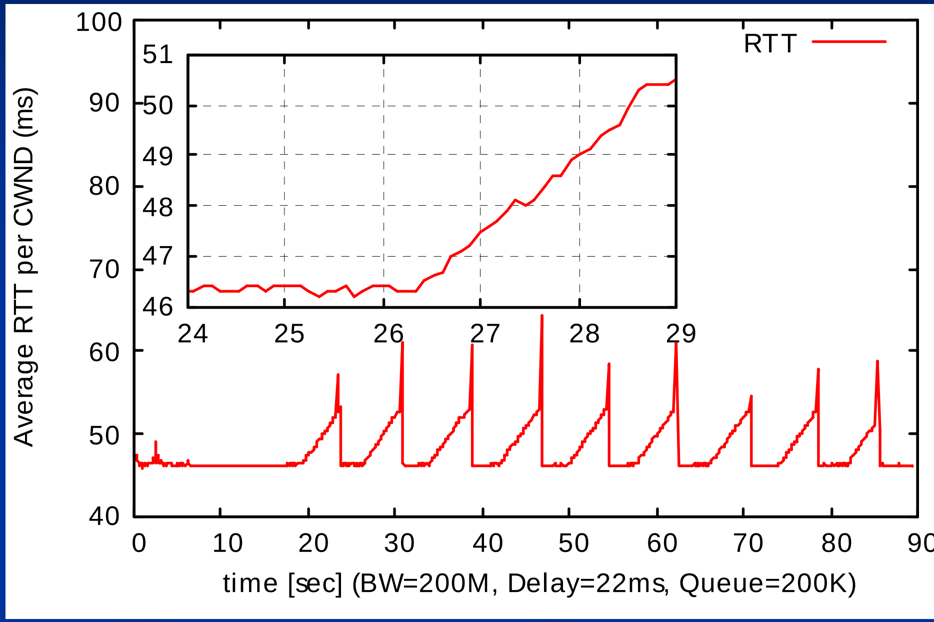
Setting of the test-bed network

- Dummynet is used as the infrastructure.
- It can emulate:
  - bottleneck link bandwidth
  - bottleneck link delay
  - buffer size of router
- TCP stack of S1 is different in each experiment.
- S2 uses TCP Reno.



Check the algorithm of gHSTCP





- Only Flow-1 exists, S1 uses gHSTCP.
- Problems: RTT's oscillations lead to unnecessary mode switching behavior.
- Lower ability for catching bottleneck link bandwidth and unfairness against competing TCP Reno traffic.

A refined algorithm of gHSTCP

Idea: the RTT is larger than the propagation delay when the link bandwidth is fully utilized.

Notation: RTT_min is minimum of average RTT in 1-cycle.
RTT_std is standard deviation of RTT.

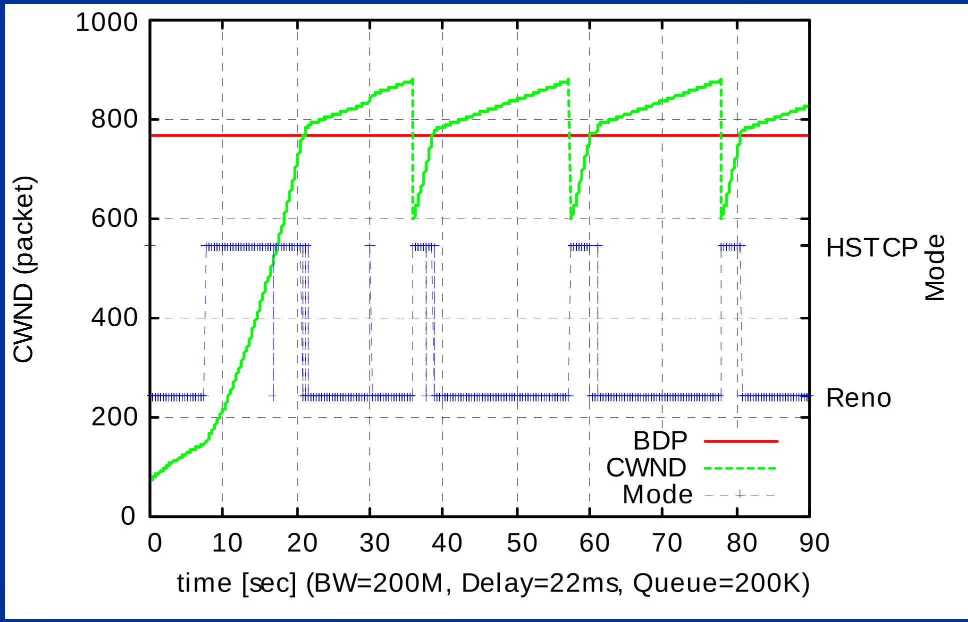
If RTT < RTT_min + 2*RTT_std
HSTCP mode is used.

If RTT >= RTT_min + 2*RTT_std and
RTT < RTT_min + 4*RTT_std
(using the original algorithm of gHSTCP)
the mode is decided by the RTT trend.

If RTT >= RTT_min + 4*RTT_std
Reno mode is used.

Results of the refined algorithm

- If CWND < BDP
  - gHSTCP can catch the link bandwidth as quickly as the original HSTCP.
- If CWND > BDP
  - gHSTCP can provide better fairness with respect to competing TCP Reno flows.



Test gHSTCP in test-bed netowork

- The metrics of evaluation:
  - Throughput
  - Utilization
  - Fairness (Jain's fairness index)
- Two scenarios
  - Scenario-1: BW=100 Mbps, Delay=23 ms, buffer of router = 200 Kbytes.
  - Scenario-2: BW=200 Mbps, Delay=23 ms, buffer of router = 500 Kbytes.
- Flow-1 uses TCP Reno/gHSTCP/HSTCP/parallel TCP
- There are 2 TCP Reno connections in Flow-2. The socket buffer size is set to 64 KB or 512 KB in different experiments, respectively.

Results (Scenario-1)

- Bottleneck link bandwidth = 100 Mbps, Buffer of router = 200 Kbytes.

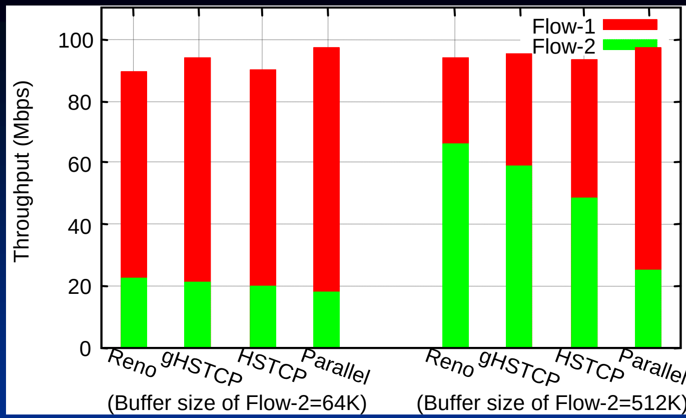
When the buffer size of S2 is set to 512 Kbytes:

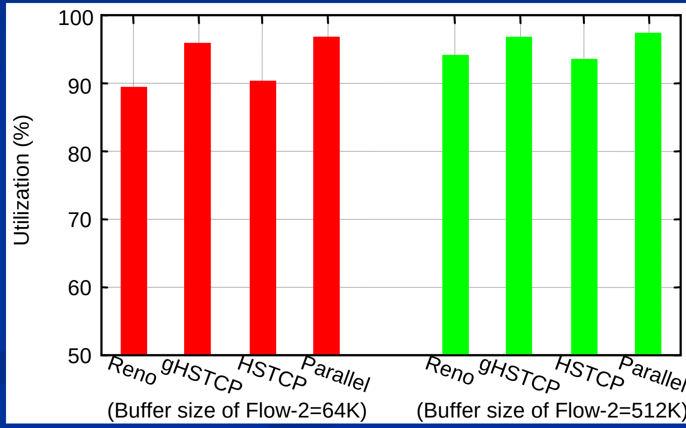
•All of utilization is larger than 90%.

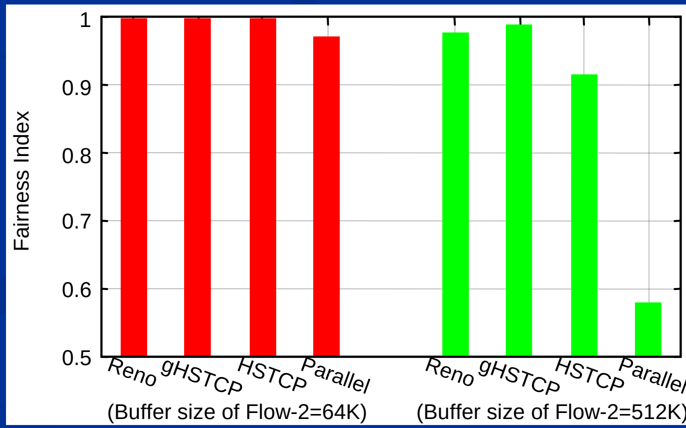
•The fairness is determined by the algorithms of TCP and the competing flows.

•The fairness is very poor when parallel TCP is used.

•gHSTCP outperforms HSTCP in terms of utilization and fairness.







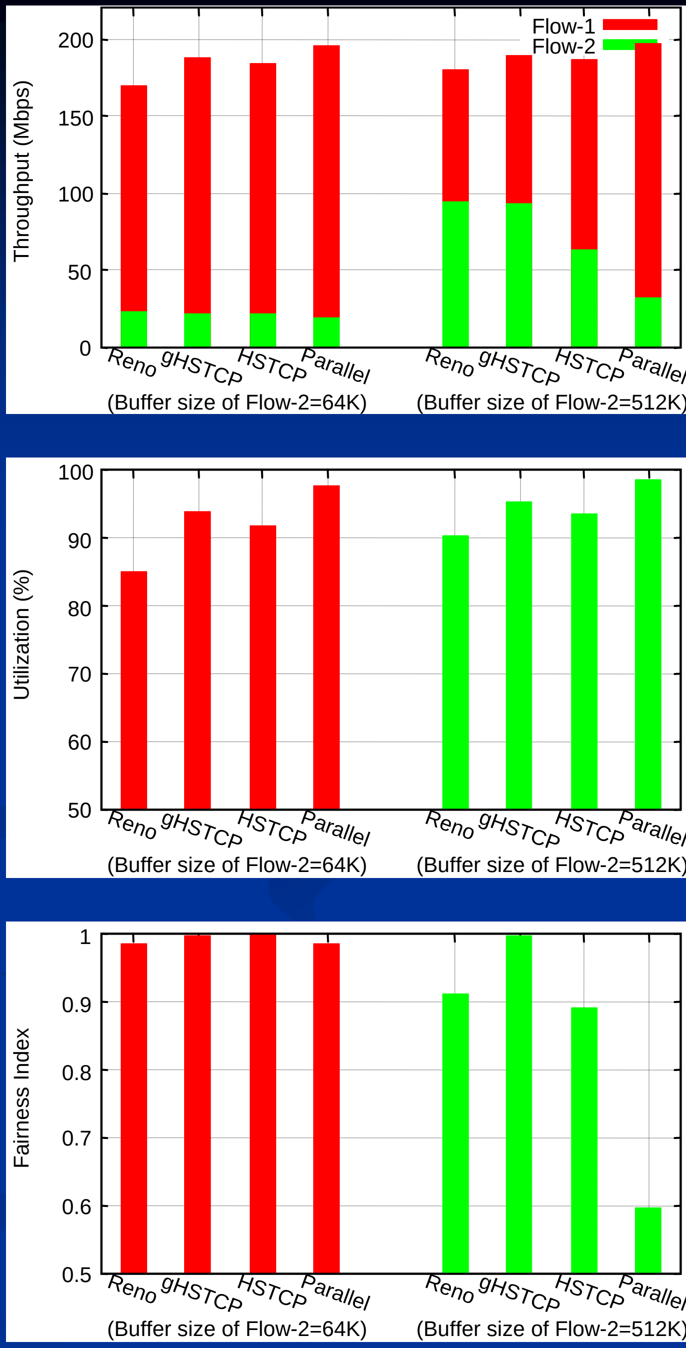
Results (Scenario-2)

- Bottleneck link bandwidth = 200 Mbps, Buffer of router = 500 Kbytes.

The difference from Scenario-1: The bottleneck link is larger than the access link bandwidth of Flow-2.

•Even when the buffer size of S2 is set to 512 Kbytes, there still exists redundant link bandwidth on the bottleneck link.

•gHSTCP can utilize this redundant link bandwidth very well. Whereas, HSTCP or parallel TCP pillages vast resources from Flow-2.



Conclusions & Future works

- A refined gHSTCP algorithm is proposed.
- The performances of TCP Reno, HSTCP and gHSTCP are evaluated experimentally.
- The parallel TCP mechanism is evaluated as a candidate for LFNs.
- gHSTCP offers the best tradeoff in terms of utilization and fairness.
- Future works
  - Test using Active Queue Management (AQM).
  - Test in a higher speed network and the Internet.
  - Evaluate parallel TCP by analysis.