Experimental Evaluation of Gentle HighSpeed TCP

Zongsheng Zhang
Go Hasegawa
Masayuki Murata
Osaka University
Outline

• Introduction
• Backgrounds of HSTCP & gHSTCP
• Why to evaluate in emulation networks
• Refined algorithm of gHSTCP
• Results
• Conclusions & future works
Introduction

• What’s wrong with TCP?
  – TCP was designed when T1 was a fast network.
  – It doesn’t perform well in fast long-distance networks (FLDNs) because of congestion window (CWND) algorithms.

• Solutions:
  – Traditional method: parallel TCP mechanism
  – New methods: new algorithms for updating CWND, e.g., HSTCP, Scalable TCP, FAST TCP.
HighSpeed TCP \(^3\) (HSTCP)

- HSTCP: a representative of high speed protocols.
- It uses the Additive Increase and Multiplicative Decrease (AIMD) principle.
- It may be easily deployed in the Internet.
- Currently, HSTCP is the only protocol recommended by IETF for FLDNs.
- However, unfairness is a drawback.

---

Gentle HighSpeed TCP \([1]\) (gHSTCP)

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

---

Simulation vs. Emulation (real network)

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

Settings of the emulation 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.
Validation of the gHSTCP algorithm

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

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

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

If $RTT < RTT_{\text{min}} + 2*RTT_{\text{std}}$
    HSTCP mode is used.
If $RTT \geq RTT_{\text{min}} + 2*RTT_{\text{std}}$ and
    $RTT < RTT_{\text{min}} + 4*RTT_{\text{std}}$
    (using the original algorithm of gHSTCP)
    the mode is decided by the RTT trend.
If $RTT \geq RTT_{\text{min}} + 4*RTT_{\text{std}}$
    Reno mode is used.
Result of the refined algorithm

- If CWND < BDP
  - gHSTCP can catch 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 emulation network

• 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 each experiment, respectively.
• BW of bottleneck = 100 Mbps, Buffer of router = 200 Kbytes.
  – Exp-1: S1 uses TCP Reno
  – Exp-2: S1 uses gHSTCP
  – Exp-3: S1 uses HSTCP
  – Exp-4: S1 uses parallel TCP

When the buffer size of S2 is set to 64 Kbytes:
• The main limit on S2 is its socket buffer size.
• Fairness is better in all cases.
• Parallel TCP achieves the best utilization, but the worst fairness.
• The throughput of gHSTCP is slightly less than that of parallel TCP, it is better than others.

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)

- BW of bottleneck = 200 Mbps,
- Buffer of router = 500 Kbytes.
  - Exp-5: S1 uses TCP Reno
  - Exp-6: S1 uses gHSTCP
  - Exp-7: S1 uses HSTCP
  - Exp-8: S1 uses parallel TCP (16 connections)

On the whole, the utilization and fairness trends are the same as those demonstrated in Scenario-1.

- Parallel TCP achieves the best utilization, but the worst fairness.
- gHSTCP offers higher utilization and better fairness than the other protocols.
- That is, gHSTCP is the best tradeoff in terms of link utilization and fairness.
Conclusions & Future works

- The 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 FLDNs.
- gHSTCP offers the best tradeoff in terms of utilization and fairness.
- Future works
  - Test with Active Queue Management (AQM).
  - Test in a higher speed network and the Internet.
  - Evaluate parallel TCP by analysis.