Performance Analysis and Improvement of HighSpeed TCP with TailDrop/ARED Routers

Zongsheng ZHANG
Go HASEGAWA
Masayuki MURATA

{zhang,hasegawa,murata}@ist.osaka-u.ac.jp
Osaka University

Background

- Success of the Internet.
- Continuous and explosive growth of the Internet
  - Especially High-speed Long-delay network.
- New application – Data Grid, SANs, · · ·
  - Hosts have gigabit-level network interface.
  - Perform backups, synchronize databases
- TCP (Transmission Control Protocol)
  - Traditional transport-layer protocol – TCP Reno used in the above applications
  - Can't achieve sufficient throughput.

Problems with TCP Reno[1]

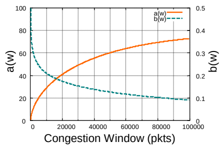
- AIMD (Additive Increase Multiplicative Decrease) algorithm of TCP Reno in congestion avoidance phase:
  - No packet loss (AI): increase congestion window by one packet/RTT → too slowly
  - Packet loss (MD): decrease congestion window by half → too dramatically
- Example: For fully utilize a link of 10 Gbps with
  - Packet size: 1,500 bytes
  - RTT (Round Trip Time): 100 ms

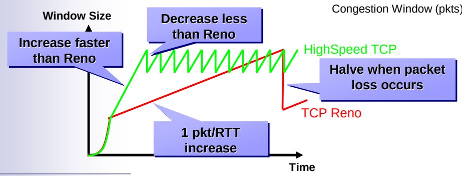
This requires:
  - Congestion window = 83,333 packets → Packet drop rate = 2×10^{-10}
    It is impossible with present optical fiber and router technology.

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

HighSpeed TCP – HSTCP[1]

- AIMD of HSTCP
  - AI: 1 → a(w)
  - MD: 0.5 → b(w)





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

Objectives

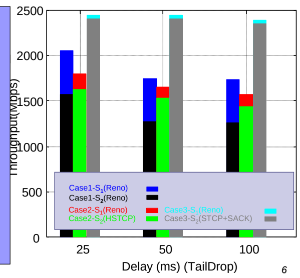
- Consider the mixture situation of TCP Reno and HSTCP
- Evaluate the performance of HighSpeed TCP
  - Simulation.
  - Metrics: Throughput and fairness against TCP Reno connections
- Propose 'Gentle' HighSpeed TCP
  - Higher throughput and better fairness than the original HighSpeed TCP

Evaluati

Case3: TCP Reno is used for S_1 and HSTCP+SACK is used for S_2.

- TailDrop,
- 3 Cases: Case1, Case2, Case3.

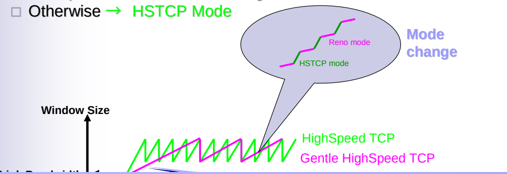
- Problems with HSTCP
  - Under-utilization (Case2)
  - Unfairness (Case3)
- Reasons
  - Buffer overflow leads that many packets are lost in bursty fashion
  - HSTCP still opens its CWND quickly even when packets are going to be stored in the router buffer



Gentle HighSpeed TCP – gHSTCP

- 2 modes in congestion avoidance phase

  Observes the packet transmission time and its RTT:
  - If the positive correlation is recognized → Reno mode
  - Otherwise → HSTCP Mode

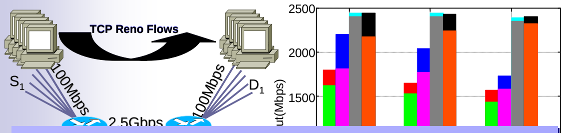


- gHSTCP:
  - Prevent bursty packet drop.
  - Provide suitable fairness to the competing TCP Reno flows.

Evaluati[on]

Case5: TCP Reno is used for S_1 and gHSTCP+SACK is used for S_2.

- TailDrop, different access link bandwidth.
- 2 Cases: Case4, Case5.



- gHSTCP outperforms HSTCP in terms of:
  - Throughput.
  - Fairness.
- But the performance does not satisfactory, especially as the delay increases.

Why introduce ARED

- Problems with TailDrop
  - burst packet drop.
  - global synchronization.
  - Unfairness.
- RED (Random Early Detection) is recommended.
  - But control parameters in RED is highly sensitive to the network condition.
- ARED (Adaptive RED)
  - an improved version of RED.

ARED – Adaptive RED

- Difference between RED and ARED
  - Key point of ARED:

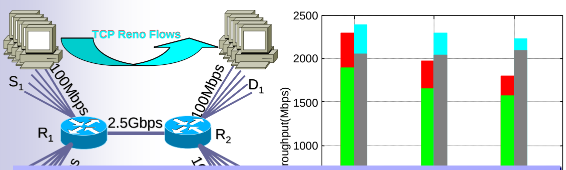
    max_p is dynamically adapted to keep the average queue size within the target queue boundaries according to network conditions.
- Target queue range:

  $[min_{th}+0.4(max_{th}-min_{th}),\ min_{th}+0.6(max_{th}-min_{th})]$

Evaluati[on]

Case6: TCP Reno is used for S_1 and HSTCP is used for S_2.

- ARED, different access link bandwidth.
- 2 Cases: Case6, Case7

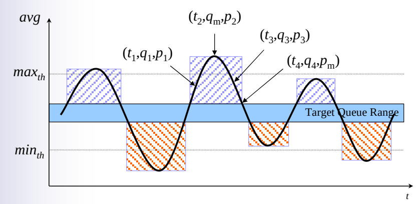


- Even with ARED
  - Can't fully utilize link bandwidth
  - Fairness is not good.

Algorithm used by ARED

- Sketch of ARED

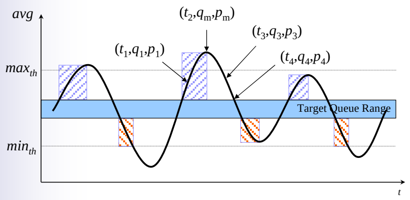
```
if (avg > target  and max_p < top)
    increase max_p:
if (avg < target  and max_p > bottom)
    decrease max_p:
```



Gentle Adaptive RED -- gARED

- Sketch o

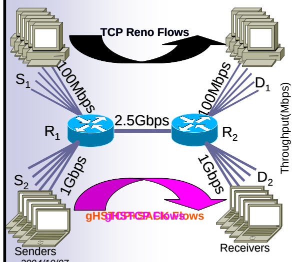
if (avg > target and avg > old_avg and max_p < top)
  increase max_p:
if (avg < target and avg < old_avg and min_th < avg)
  decrease max_p:

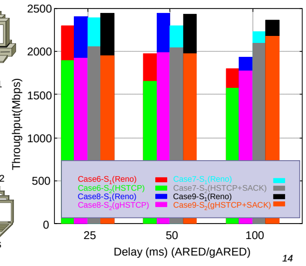


Evaluati

Case9: TCP Reno is used for S_1 and gHSTCP+SACK is used for S_2.

- gARED, d...
- 2 Cases: Case8, Case9.





Conclusion

- gHSTCP+gARED can provide better performance in terms of fairness and throughput.
  - gHSTCP: two modes in congestion avoidance phase based on the changing trend of RTT
  - gARED: adapt max_p according to average queue length and its trend in variation
- Future works: further investigation, e.g.
  - Recover effectively from simultaneous packet losses
  - The impact on short-live traffic

Thanks