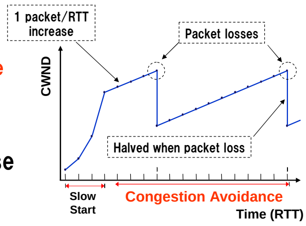
Protection Mechanisms for Well-behaved TCP Flows from Tampered-TCP at Edge Routers

Junichi Maruyama, Go Hasegawa, and Masayuki Murata
Osaka University, JAPAN

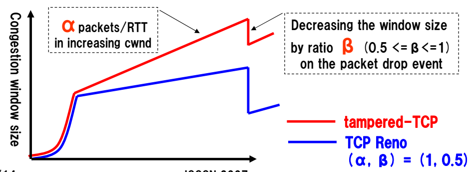
Congestion control in today's Internet

- Transmission Control Protocol (TCP)
  - Instrumental in preventing congestion collapse
  - Limit transmission rate at the source
  - Window-based rate control -- Congestion window (CWND)
- Four algorithms:
  - Slow-start
  - Congestion-avoidance
  - Fast-retransmit
  - Fast-recovery
- Additive Increase Multiplicative Decrease (AIMD) algorithm

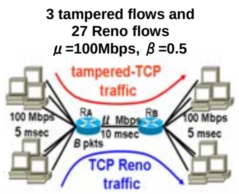


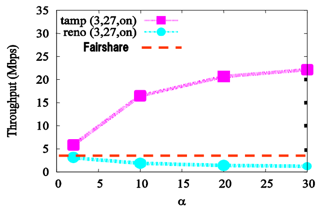
Definition of tampered-TCP

- TCP variants which modify their congestion control mechanisms for higher throughput
  - Larger increasing ratio (α) and smaller decreasing ratio (β)
    - Easy to be implemented to sender side TCP



Effect of tampered-TCP flows

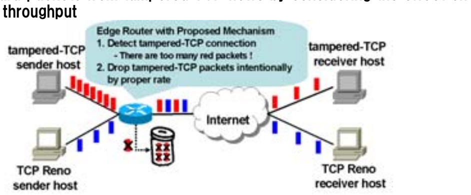




- Small number of tampered TCP flows significantly degrades throughput of co-existing TCP Reno flows
- Since tampered-TCP can be easily realized at end-hosts, routers must be equipped with additional mechanisms to protect well-behaved TCP Reno flows

Objectives of this work

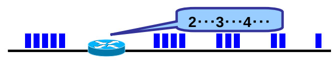
- Propose new mechanisms to detect and regulate tampered-TCP connections at edge routers
  - Check the tampering property of TCP flows passing through a edge router
  - Discard packets from tampered TCP flows by considering the effect on TCP throughput



- Two approaches
  - Cwnd-based mechanism and Throughput-based mechanism

Cwnd-based Mechanism (1/2)

- Monitor TCP flow's packets for estimating congestion window size
  - TCP sends packets in a window in bursty fashion
  - A long interval between successive two packets is considered as the boundary of two windows



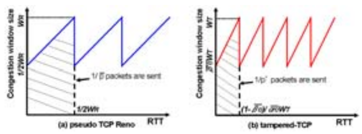
- Estimate increase and decrease ratio of the window size ($\overline{\alpha_e}, \overline{\beta_e}$) of each flow
  - From the changes in the estimated window size
- Estimate the packet loss ratio at the router
  - By using number of transmitted/discarded packets at the router, obtained from the router's MIB

Cwnd-based Mechanism (2/2)

- Assess the tampering property
  - The connection that doesn't satisfy the fairness against TCP Reno connections is considered as tampered-TCP

$$\frac{4\left(1-\overline{\beta}_e^{\,2}\right)}{3\,\overline{\alpha}_e} < \left(1-\gamma_w\right), \left(0<\gamma_w<1\right)$$

- Set the target packet discarding probability p'
  - p' is determined so as to equalize the throughput of the TCP Reno connection with that of the regulated tampered-TCP connection



p' is set so as to be $p' = p$

$$p' = \frac{\left(1+\overline{\beta}_e\right)}{3\left(1-\overline{\beta}_e\right)}\overline{\alpha}_e\, p$$

Throughput-based Mechanism

- Observe TCP flow's throughput T_o
  - Taken from traffic monitoring tools such as sFlow and NetFlow
- Calculate the estimated Reno's throughput T_e
  - TCP Reno's throughput under the same condition [18]

$$T_e = \frac{s}{\mathrm{RTT}\sqrt{\frac{2bp}{3}} + \mathrm{T}_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p\left(1+32\,p^2\right)}$$

- Assess the tampering property

$$\frac{T_o}{T_e} > \left(1+\gamma_t\right), \left(0<\gamma_t\right)$$

- Set the target packet discarding probability p'
  - The TCP throughput is proportional to the inverse of the square root of the packet loss rate

$$p'_{next} = \left(\frac{T_o}{T_e}\right)^2 p'_{prev}$$

[18] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and empirical validation," in Proceedings of ACM SIGCOMM '98, Sept. 1998.

Simulation setting

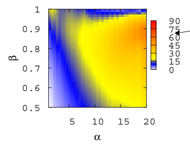
- 20 Reno flows and one tampered flow
- Bottleneck link: 50 Mbps, 10msec, 333 packets buffer
- Packet size: 1500 bytes
- Simulation time : 70 seconds
  - At the beginning, only TCP Reno flows transmit data
  - After 10 seconds, the tampered-TCP connection joins the network
- Metric
  - **Throughput ratio, detection time**, detection ratio, and false positive ratio

$$Throughput\ ratio = \frac{(Throughput\ of\ \text{tampered-TCP})}{(Throughput\ of\ \text{TCP Reno})}$$

Simulation results: Throughput ratio

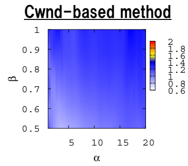
No control

$$Throughput\ ratio = \frac{(Throughput\ of\ \text{tampered-TCP})}{(Throughput\ of\ \text{TCP Reno})}$$

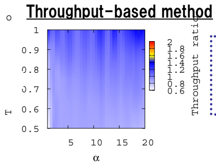


The throughput ratio is larger than 1 in almost all region, meaning that the tampered-TCP connection occupies the bandwidth

Cwnd-based method



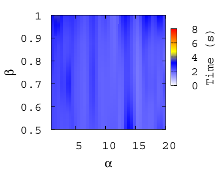
Throughput-based method



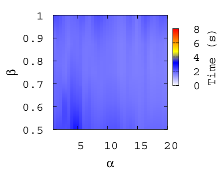
Proposed methods keep the throughput ratio to 1 in all parameter region

Simulation results: Detection time

Cwnd-based method



Throughput-based method



- Both methods detect tampered TCP flows in 2-3 seconds
  - Much faster than MIB-based detection mechanism, which normally requires 5 minutes

Conclusion and future works

Conclusion

- We proposed new mechanisms at edge routers to detect and control tampered-TCP connections
  - Cwnd-based method
  - Throughput-based method

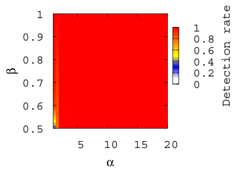
  ⇒The proposed methods can keep the throughput ratio around 1, by intentionally discards packets from tampered flows

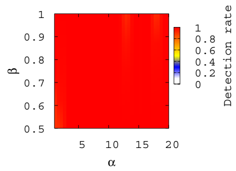
Future works

- Investigation of the performance of the proposed mechanisms in the actual Internet environment
- Simulations with TCP variants for high-speed and long-distance networks
  - High-speed TCP, compound TCP, CUBIC, …

Simulation results: Detection ratio

Cwnd-based method / **Throughput-based method**

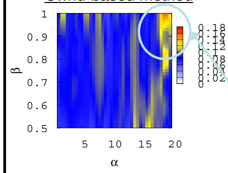




- Both of proposed methods can detect tampered TCP flows regardless of the degree of tampeing characteristics

Simulation results: False positive ratio

Cwnd-based method



The stronger the tampering property becomes, the larger false positive ratio becomes
↑
The throughput or window size of TCP Reno unstably changes

Throughput-based method

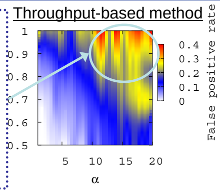


Table : Throughput and throughput ratio of misassessed TCP Reno and Successfully-assessed TCP Reno connections

Method	(α, β)	Misassessed Reno (Mbps)	Reno (Mbps)	Throughput ratio
Cwnd	(10, 0.7)	2.333	2.396	0.973
	(20, 0.9)	2.290	2.386	0.960
Through-put	(10, 0.7)	2.548	2.401	1.061
	(20, 0.9)	2.175	2.431	0.895