

---

# A Performance VS Trust Perspective in the Design of End- Point Congestion Control Protocols

---

K. Aleksandar & W.K Edward

Presented by: Chakchai So-In  
Discussion Leader : Amy Freestone  
*October 13<sup>th</sup> 2005*

---

## Outline

- Motivation
- Vulnerabilities
  - Receiver Misbehaviors
    - DOS and Resource Stealing
  - Analyze congestion control parameters
    - Long Time Scale: AIMD and RTO
    - Short Time Scale: Initial Window
  - Modeling receiver misbehaviors
- Network-based solutions
  - Core-Based and Edge-Based

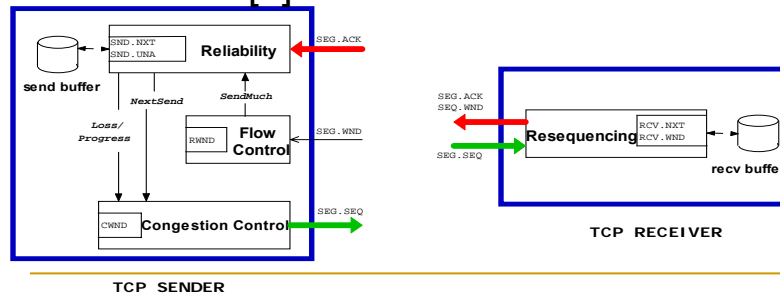
# Outline

- End-point solution
  - Long Time Scale Misbehavior
    - Sender-Side Verification
    - Evaluation
    - Advanced Congestion Control Mechanisms
  - Short Time Scale Misbehavior
- Conclusion
- Reference
- Question?



# Motivation

- Problem? (Optimized CC algorithm)
  - How to distinguish among..
    - Users with optimized protocol stacks
    - Cheater: no concern about fairness and network stability
    - Attackers: performs DOS
- Sender Centric [2]



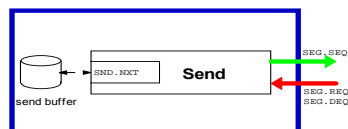
## Motivation

- Why Receiver-driven TCP? [H.-Y. Hsieh et al.'s]
  - Improved loss recovery
  - More robust congestion control
  - Bandwidth aggregation
  - Web response times
  - Improved power management for mobile devices
  - A solution to the handoff problem
  - Network-specific congestion control
  - Easy migration to a replicated server during handoffs

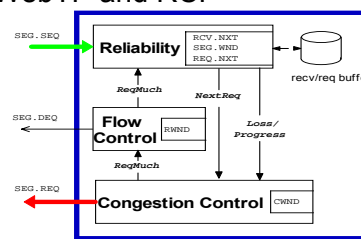


## Motivation (cont.)

- Receiver-centric [2]
  - Receiver controls *how much data can be sent, and which data should be sent* by the sender
  - Example:
    - Increase Receiver functionality: NETBLT, WTCP, TFRC, and TCP-Real
    - Full Receiver functionality: WebTP and RCP



RCP SENDER



RCP RECEIVER



## Vulnerabilities

### ■ Receiver Misbehaviors

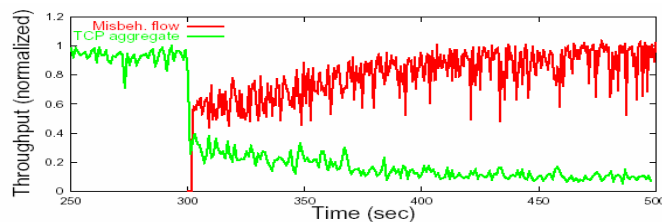
#### □ Denial-of Service attack (DOS)

- Flooding the sender with short request packet, which it replies and congests the network



#### □ Resource Stealing

- Re-tune parameters to steal BW from other flows while eluding detection (7TCPSack:1RCP) (e.g. not halving window)



## Vulnerabilities (cont.)

### ■ Analyze misbehavior parameters

#### □ Additive-increase Parameter ( $\alpha = 1/\text{RTT}$ )

- Achieve higher throughput ( $\alpha > 1$ )

#### □ Multiplicative-decrease parameter ( $\beta=0.5$ )

- Utilize more bandwidth ( $\beta > 0.5$ )

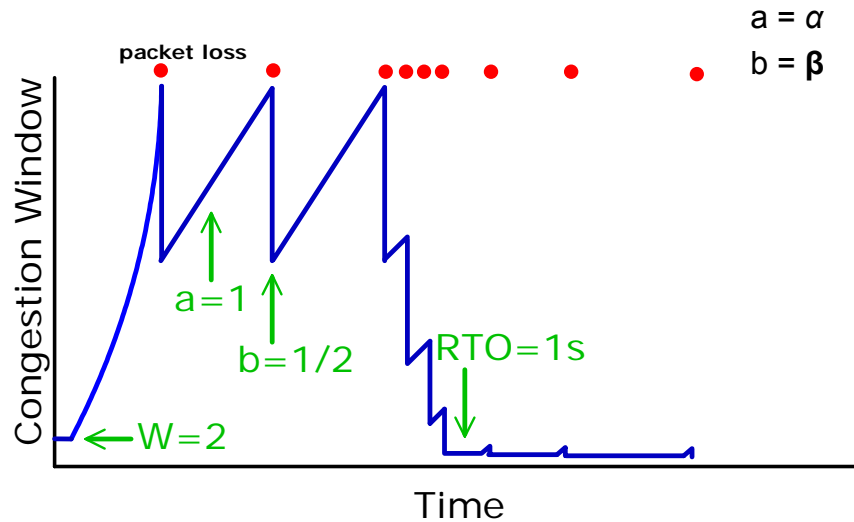
#### □ Retransmission timeout (RTO)

- RFC 2988 (Lower-upper bound: 1 and 60 sec)

#### □ The initial window size ( $W=2$ )

- RFC 2414 (2 to 4 segment: 4Kbytes)

## Vulnerabilities (cont.)



## Vulnerabilities (cont.)

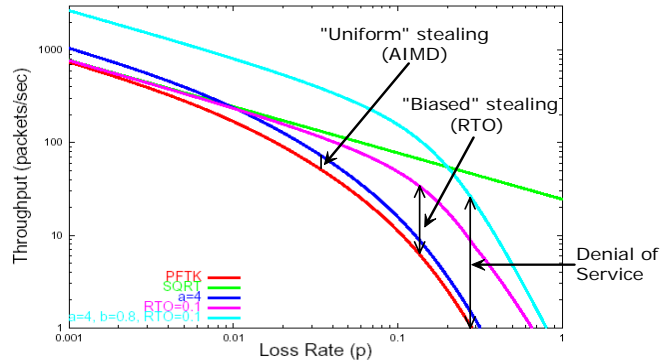
### ■ Modeling receiver misbehaviors [J. Padhye et al.]

- $d = 1/\beta$  ( $\beta = 0.5$  and  $\alpha = 1$ , (1) = (2))
  - $B$  = Average TCP Rate,  $RTT$  = Round-trip time
  - $p$  = Loss event rate
  - $RTO$  = TCP retransmission timeout
  - $b$  = The number of packets acknowledged by each ack ( $b = 1$ )

$$B \approx \frac{1}{RTT \sqrt{\frac{2bp}{3}} + RTO \min(1, 3\sqrt{\frac{3bp}{8}}) p(1 + 32p^2)} \quad (1)$$

$$\frac{1}{RTT \sqrt{\frac{2bp(d-1)}{\alpha(d+1)}} + RTO \min(1, 3\sqrt{\frac{bp(1+d)(d-1)}{2\alpha d^2}}) p(1 + 32p^2)} \quad (2)$$

## Vulnerabilities (cont.)

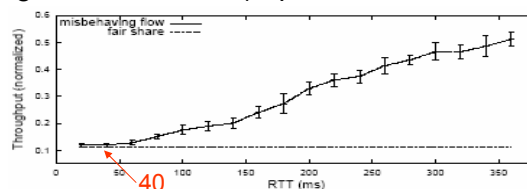


- PFTK (Equation (1), with  $b = 1$  and  $RTO = 1$ )
- SQRT is the "square-root" formula from [M. Mathis et al.] same as equation (1) without the RTO part
- The malicious receiver throughput from Equation 2 ( $\alpha$ ,  $\beta$  and  $RTO$ )



## Network-based solutions

- Core-Based
  - Fair Queuing (FRED, CHOKe, and SFB)
  - RED-PD [Mahajan et al.]
    - Use packet drop history to detect high BW flows and drop those.
    - Monitors flows at the router and compares their rates to the targeted bandwidth (Square-root TCP-friendly)



$\alpha = 25$

Steal 5 times

Figure 5. RED-PD is unable to detect a malicious flow

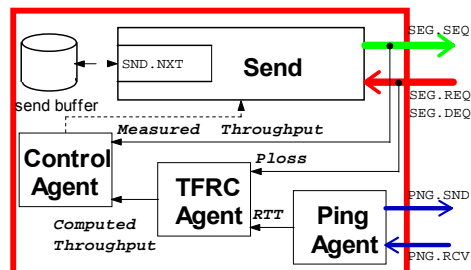


## Network-based solutions (cont.)

- Edge-Based (Monitor packets both direction)
  - D-WARD [Mirkovic et al]
    - Measure both outgoing (*data*) and incoming (*ack*) traffic and define the maximum allowable ratio of the two

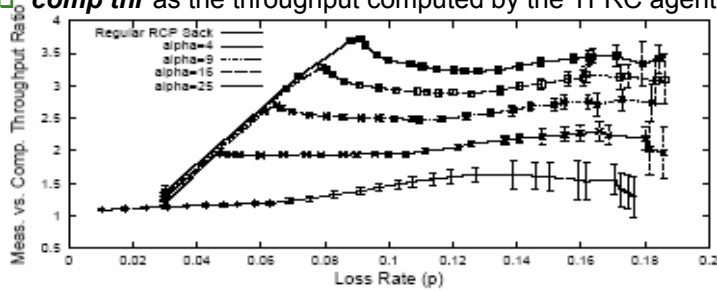
## End-point solution

- Sender-Side Verification
  - Add the minimum functionality to the sender : robustly detect receiver misbehavior over long time scales
    - Ping Agent: Measures RTT by pinging untrusted receiver
    - TFRC Agent: Computes “TCP-fair” rate in real time
    - Control Agent: Rate-limiting and dropping packets



## End-point solution (cont.)

- Evaluation (1 RCP vs 600 TCP-SACK)
  - Detecting Misbehaving Receiver (re-tune  $\alpha$ )
    - Clearly differ from the behaving flows' profile
    - Approximately proportional to  $\sqrt{\alpha}$
    - *meas thr* as the throughput measured by the RCP sender
    - *comp thr* as the throughput computed by the TFRC agent



## End-point solution (cont.)

- Evaluation (cont.)
  - Detection Threshold ( $k=1, 1.8, \text{ and } 3$ )
    - Behaving,  $P(k)$  is the false-alarm
    - Misbehaving,  $P(k)$  is the correct misbehavior-detection

$$P(k) = \text{Prob}\left(\frac{\text{meas-thr}}{\text{comp-thr}} > k\right).$$

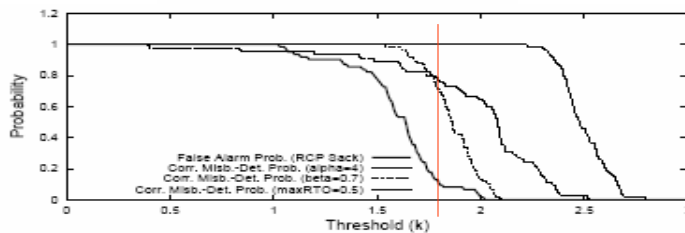


Figure 9. Detecting out-of-profile flows

## End-point solution (cont.)

- Advanced Congestion Control Mechanisms
  - TCP performance in wireless (Tcp-ELN & WestWood)
    - RCP-ELN flow is difficult to distinguish from a misbehaving flow ( $k=1,4,$  and  $7$ )

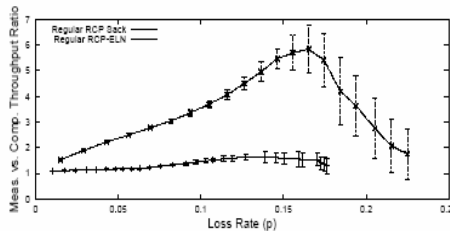


Figure 10. RCP-ELN significantly improves throughput

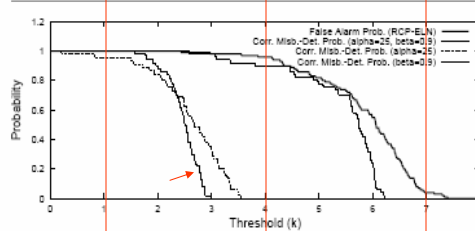


Figure 11. From the sender's perspective, RCP-ELN looks like a misbehaving flow



## Short Time Scale Misbehavior

- Initial Congestion Window ( $W$ ,  $RTO$ )
- Solution (Drawback for improve performance)
  - Rate-limit flows (10Mbps,  $W=100$ ,  $RTT=50ms$ , 200Kbps)
  - Smart RCP client

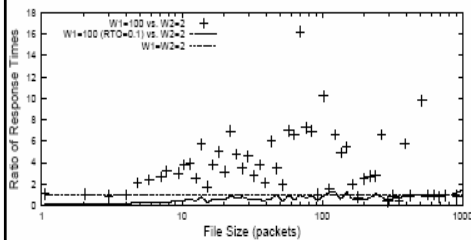


Figure 12. A greedy receiver can significantly degrade legitimate background web traffic

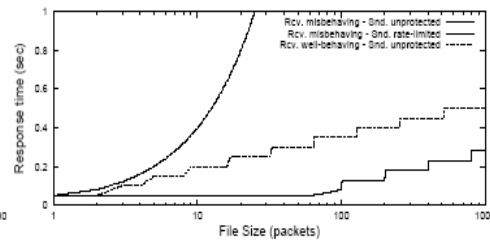


Figure 13. Protecting against short-time-scale misbehaviors



## Conclusion

- Receiver-based TCP
  - Highly vulnerable to receiver misbehaviors
- Network-based schemes
  - Limited the ability to detect and punish the endpoint misbehaviors
- End-point-based solution
  - Accurately detect long-time-scale receiver misbehaviors



## Reference

- K. Aleksandar and W.K. Edward, "A Performance vs. Trust Perspective in the Design of End-Point Congestion Control Protocols", ICNP 2004 12th IEEE International Conference on Network Protocols.
- Supranamaya Ranjan, Presentation Slide: "A Performance vs. Trust Perspective in the Design of End-Point Congestion Control Protocols", available at <http://www-e.rice.edu/~akuzma/Doc/akuzma/ICNP04.ppt>.
- CNET Staff, *How a "denial of service" attack works*, available at <http://news.com.com/2100-1017-236728.html?legacy=cnet>
- J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation", *IEEE/ACM ToN*, 8(2):133–145, 2000.
- S. Savage, N. Cardwell, D. Wetherall, and T. Anderson, "TCP congestion control with a misbehaving receiver", *ACM Computer Comm. Review*, 29(5):71–78, 1999.



Question?