Compound TCP

Murari Sridharan
Windows TCP/IP Networking
Microsoft Corporation

(Collaborators: Kun Tan, Jingmin Song, MSRA & Qian Zhang, HKUST

The Compound TCP approach

- Synergy between loss and delay based approaches
  - Using delay to sense network congestion
  - Adaptively adjust aggressiveness based on network congestion level.
- One flow, two components
  - Loss based component: *cwnd* (standard TCP Reno)
  - Scalable delay-based component: *dwnd*
    - Vegas-like early congestion detector
  - TCP send window is controlled by

    $win = cwnd + dwnd$

CTCP congestion control

- *cwnd* is updated as TCP Reno
- *dwnd* control law
  - Binomial increase when no congestion
  - Multiplicative decrease when loss is detected
  - On detecting incipient congestion
    - Decrease *dwnd* and yield to competing flows
- The above control law kicks in only when the flow is in congestion avoidance and *cwnd* >= 40 packets. No changes to slow start phase.
- Gamma (B/flows) tuning by emulation

Summary

- CTCP achieves good efficiency, RTT fairness, TCP fairness and stability in a variety of environments.
    - Validated on test-beds, Microsoft IT high-speed links, Microsoft internal deployments, SLAC/Internet2/ESNet production links.
- Useful links
    - http://research.microsoft.com/wn/ctcp.aspx
    - http://research.microsoft.com/~padhye/tcpworkshop/
    - http://www.slac.stanford.edu/cgi-wrap/getdoc/slac-tn-06-005.pdf

High Speed Networking Deployment Challenges

- Application limitations
  - Bottleneck could be the applications. How much data to send? How to efficiently post receives to consume data?
- Interoperability challenges
  - Poor RFC compliance blocks deployment of high-speed extensions like Window Scaling and ECN
- Breakdown of end to end connectivity
  - Middle boxes
- Diagnostics
  - Performance limitations – TCP ESTATS MIB is a great start.
- Criteria for deploying high-speed congestion control algorithms