Low Priority TCP: Receive-Window Control

Murari Sridharan

Microsoft

Collaborators: Peter Key, Richard Black MSR Cambridge

Background

- End-system based approach for content distribution, OS updates, prefetching etc
- Background Transfer Service (BATS)
  - Receiver window adaptation to create a low priority service
  - Simulation and experimental studies within user-mode process and in kernel mode by modifying the Windows TCP/IP stack.
- Tightly couple capacity interference and rate control
  - Rate controlled by adjusting receiver window
  - Rate obtained for a given receiver window is then used to infer whether rate is above or below available capacity, which can in turn trigger adaptation of the receiver window.
- Conforms to RFC guidance on TCP receive window operation

Design

- Key observations (paraphrasing 2 Theorems from the paper)
  - In both the delay and loss constrained cases, the goodput normalized by the receiver window is constant over a range [0,W^*_b] with slope 1/RTT, and decreasing over ($W^*_{b,}$ +infinity)
  - W^*_b is the target window to maximize background goodput while not interfering with foreground flows
- Operating point can change dynamically but the key idea is to drive receive window to the target

Theory Vs Simulation

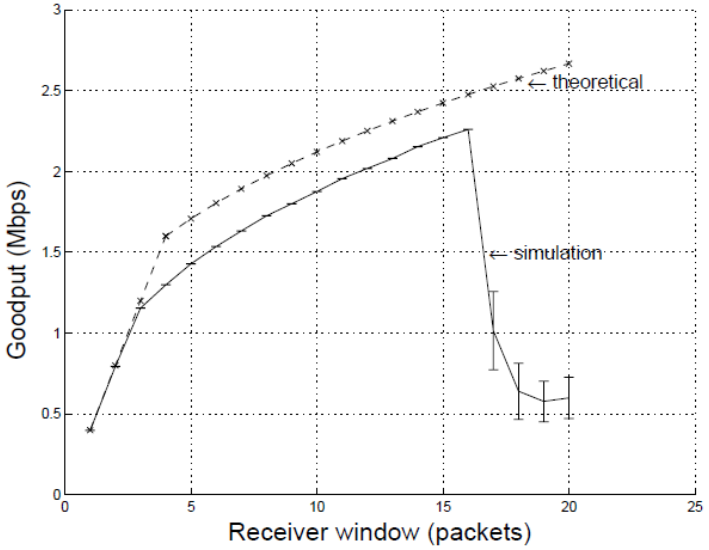

Figure 2: The goodput of the background flow versus the window size with 8 foreground flows, when $C = 2000$ packets per second, $B = 40$ packets.

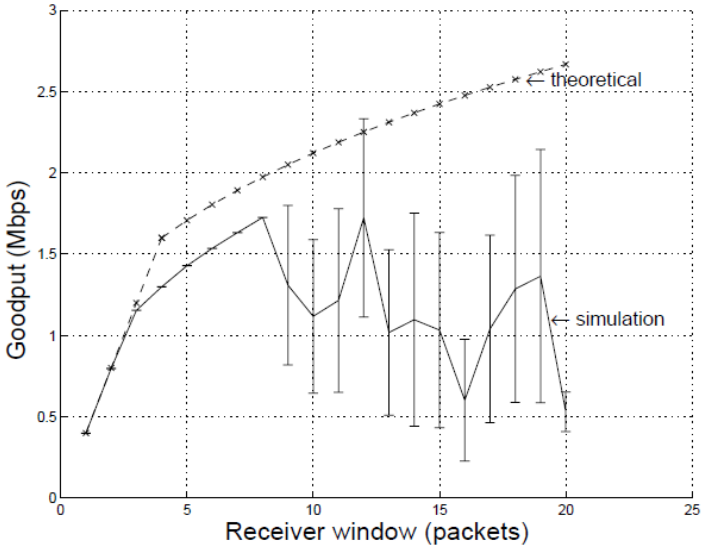

Figure 3: The goodput of the background flow versus the window size with 8 foreground flows, when $C = 2000$ packets per second, $B = 20$ packets.

Algorithm

- Rate limiting Mode
  - Used to get accurate RTT samples and/or to hibernate the connection
  - Allows window to be completely shut or opens to 2MSS
- Window Scaling Mode (not to be confused with TCP WS)
  - Primary mode of operation
  - Uses binary search to drive towards target window assuming the value lies between Wmin & Wmax
  - When no congestion is detected
    - If search space is large, Wmin = (Wmax + Wmin)/2
    - If search space is small, Wmax += 2MSS
  - When congestion is detected
    - If search space is large, Wmax = (Wmax + Wmin)/2
    - If search space is small, Wmin -= 2MSS
- Methods to detect congestion
  - Variances in RTT
  - CTCP style backlog estimation

Summary

- Maintains low delay, yields to TCP
- Consumes residual capacity effectively
- Requires no support from the network although additional information can be used to improve estimation of the target window
- Requires no changes in the sender
- Challenges
  - Getting a good basertt
  - When to dump the basertt? Route flaps, changing conditions
  - Eliminating noise in RTT estimation/detecting congestion
  - Yield to TCP over reasonable time scales
  - Ok to be conservative but flows should not starve

Further reading

http://research.microsoft.com/~peterkey/Papers/Allertonv3.pdf
http://research.microsoft.com/~peterkey/papers/kmw_sigmetrics2004.pdf