CS 4457 - Computer Networks - 10/18/2021 Lecture Notes

Ethan Hanover (eph3rrp)

October 20, 2021

1 Principles of congestion control

- **What is congestion?** Too many sources that are sending too much data too quickly. (Different from flow control, which is one sender being too fast for its one receiver)
- **What happens during congestion?** Three examples to consider:
  1. Consider a situation where two devices on a network are losslessly sending at rate λ_{in} to a router capable of processing only R bps. As λ_{in} nears $R/2$, items will have to be buffered on the router, causing delay (which increases exponentially) and packet loss if the buffer fills up (buffer overflow).
  2. Consider the same situation as above, but now with packet loss. As λ_{in} approaches $R/2$, the throughput λ_{out} decreases below a linear relationship with λ_{in} because of the extra burden of retransmitting packets due to packet loss.
  3. Consider a situation where multiple clients send traffic at the same rate through one router. As their rates increase, the clients that are closer to the router fill up the router's buffer because they're closer, and the throughput of clients far upstream from the router approaches 0.

2 What to do about congestion

- **How do we know we're experiencing congestion?** Could have functionality at the router (IPv6/ECN), could check delay by sending test packets (BBR), could check packet loss (can use fast retransmit strategy: resend lost packets upon receiving a sequence of ACKs)
- **Approaches to combat congestion:** (with a window size of *cwnd*, the approximate rate of transmission is $cwnd/RTT$ - this is what can be modified according to the following strategies) (transmission rates measured by MSS - "maximum segment size")
  - AIMD (additive increase, multiplicative decrease) (similar to TCP Tahoe and Reno): Send at 1, 2, 3, ..., MSS until loss is detected at MSS X. Then drop down to $X/2$ MSS, then begin increasing to $(X/2)+1, (X/2)+2, ...$; repeat the process. If a packet times out, then go to 1 MSS. Yields a sawtooth shaped graph.
    - This isn't the best choice because it increases slowly, decreases quickly, and doesn't incorporate information about previous losses
  - TCP cubic (actual Linux implementation): increase MSS quickly at first, but slow down increase rate when close to the previous maximum, then drop to $X/2$ when loss is detected.
  - TCP Slow Start: (the starting phase of both strategies above) first, congestion window is set to 1 MSS. While packets there is no loss and and a maximum threshold value is not yet met, double the congestion window and retransmit and wait for ACK. Once congestion window reaches half of the value of the last timeout, switch to linear growth.