Lecture 15: TCP over wireless networks

Mythili Vutukuru

CS 653 Spring 2014

March 13, Thursday

TCP - recap

- Transport layer – TCP is the dominant protocol
- TCP provides in-order reliable byte stream abstraction between end hosts
- TCP tries to fully exploit the available bandwidth in the network, without causing congestion
- Basic idea – keep a window of packets (called congestion window, or cwnd) outstanding to fill the “pipe” between sender and receiver
- Pick cwnd in a way that does not over- or under-utilize the network
- Different TCP algorithms have different ways of adjusting cwnd in response to what is observed
- TCP also has other mechanisms like in-order reassembly and retransmissions for reliability.

TCP cwnd and bandwidth delay product

- Ideally, cwnd must be set to the bandwith delay product (BDP), which is the product of
  - The bottleneck bandwidth of the link
  - The minimum RTT between sender and receiver (that includes just transmission and propagation delays, without queueing delays)
- Justification: once the sender sends BDP worth of packets on the link, the ack for the first packet comes back.
- If cwnd is lower than BDP, the bottleneck may be underutilized. If cwnd higher than BDP, leads to congestion
- Typically, the buffer size at the bottleneck link must be at least BDP to ensure that a burst of up to BDP can be accommodated without underflowing the bottleneck link

TCP cwnd and bandwidth delay product (2)

- If cwnd is too high, the bottleneck buffer will overflow and drop packets. Lost packets lead to duplicate acknowledgments (dupacks) or no acks (timeouts), depending on how many packets are lost
- TCP dynamically adjusts cwnd to approximate BDP (since bottleneck bandwidth is a varying quantity)
  - If all going well, increase cwnd to probe for a higher bandwidth
  - If some issue arises (packet loss, high RTT etc), decrease cwnd
- Most TCP variants differ in their adjustment of cwnd in response to observed network conditions (like losses, RTT)

TCP over wireless

- In the early days of wireless LANs (90s), the wireless access link had high loss rates.
- Ideally, TCP should lower cwnd only when buffer drops occur (indicating congestion), not when packets are lost due to poor signal or interference
- So early fixes to TCP over wireless include several solutions to mask the wireless losses from the TCP sender, so that cwnd is not reduced in response to wireless losses
  - End to end fixes like "explicit loss notification" to indicate wireless losses explicitly
  - Split TCP connection over wired and wireless parts
  - Locally retransmit lost packets at link layer by snooping on TCP duplicate acknowledgements
  - See the reference 'A comparison of mechanisms for Improving TCP performance over wireless links' on the class website for more details. You can focus on the first couple of sections to get a basic idea.

TCP over wireless (2)

- Most wireless links have reasonably low loss rates these days, masked mostly by effective retransmissions schemes. As a result, wireless loss is not as serious a problem now.
- Current problem with wireless links is highly variable bandwidth and delay. This leads to problems estimating the bandwidth delay product, which in turn leads to difficulties in correctly setting buffer sizes and cwnd values.
- Reasons for variable bandwidth and delay
  - Retransmissions to cover losses
  - Contention on channel and backoff
  - Wide range of available bit rates to mask channel quality changess
  - Sharing channel with a low bit rate node

Buffer bloat

- Buffer size less than BDP leads to under utilization.
- Buffer size >> BDP leads to excessive queueing delay
- To deal with variable bandwidth, the accepted notion today is to conservatively overestimate buffer space, and provision large buffers in the last hop wireless routers (base stations, access points)
- These large buffers ensure throughput is high and link is utilized, however they cause excessive delays. This phenomenon is called buffer bloat.
- Buffer bloat leads to excessive delay, especially for interactive applications (like web browsing) sharing the wireless link with a large TCP transfer (video download)
- See the reference “Bufferbloat: Dark buffers in the Internet” on the class website for more details

Buffer sizing

- Here is a brief explanation of why bottleneck buffer should be approximately equal to the bandwidth delay product (BDP). For more detail, see section 2 in the reference 'Sizing Router Buffers'
- Suppose the sender has a cwnd W, bottleneck link has buffer size B, and bandwidth delay product is BDP.
- Suppose the buffer fills up and drops packets when W = Wmax. Then, the number of packets outstanding at this stage is those in flight (BDP) plus those in buffer. So we have Wmax = BDP + B
- Most TCP variants reduce W to half when packet drop happens. So, sender window goes to Wmax/2.
- Now the sender cannot send any packets until he gets Wmax/2 acks. In this time when the sender is silent, a correctly sized buffer will have just enough packets to send such that Wmax/2 acks are generated. So B >= Wmax/2.
- Putting these together, we get 2B = BDP + B, or B = BDP.
- So, when bottleneck buffer is equal to BDP, the buffer will just drain before sender starts sending next packet.