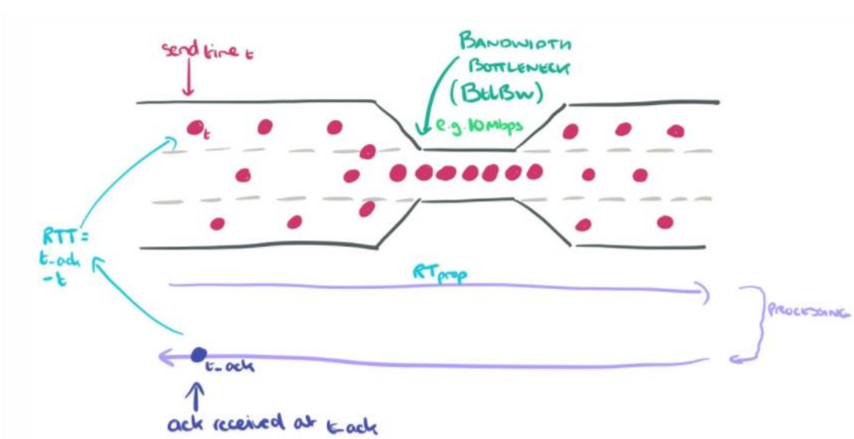
Transmission Control Protocol (TCP) Congestion Control new Architecture - Bottleneck Bandwidth and Round-trip propagation time (BBR)

Abstract:

By all accounts, today's Internet is not moving data as well as it should. The main root cause of it is the design choice made when TCP congestion control was created in the 1980s—interpreting packet loss as "congestion". Since its deployment about 27 years ago, TCP's congestion control architecture has been known for its poor performance and un-fairness on noisy links and the rapidly changing networks. Over the years many protocol "patches" have addressed problems in specific network conditions such as in satellite or wireless links, but these were only point solutions. These changes still suffer from the fundamental architectural deficiency of TCP: certain predefined packet-level events are hardwired or mapped to certain predefined control responses. TCP reacts to events that can be as simple as "one packet loss" or can involve multiple signals like "one packet loss and Round Trip Time (RTT) increased by x%". Similarly, the control response might be "halve the rate" or a more complex action like "reduce the window size w by a function of RTT". Today's modern networks have an immense diversity of conditions such as: random loss and zero loss, shallow queues and high latency, high range of links from Kbps to Gbps, last-mile networks with seconds of latency, load balancers and more. These factors add complexity far beyond what can be summarized by the relatively simplistic assumptions embedded in a hardwired mapping.

Lately, new congestion control architecture was proposed to improve the TCP performance over modern networks: Performance-oriented Congestion Control (PCC) and Bottleneck Bandwidth and Round-trip propagation time (BBR).

BBR is developed by Google. Its main concept is to continuously track the slowest link (bottleneck) bandwidth in each direction of the TCP connection as well as the RTT of the connection.



Goals:

1. Review Google BBR – https://blog.acolyer.org/2017/03/31/bbr-congestion-based-congestion-control http://queue.acm.org/detail.cfm?id=3022184, https://github.com/google/bbr
2. Raise Mininet topology with:
   a. Hosts instrumented with BBR
   b. RYU controller
   c. OVS switches configured with QoS to generate bottleneck – refer to: https://osrg.github.io/ryu-book/en/html/rest_qos.html
   d. Links with latency
3. Perform comparison between TCP (CUBIC) and BBR

Requirements:

Internet Networking Course