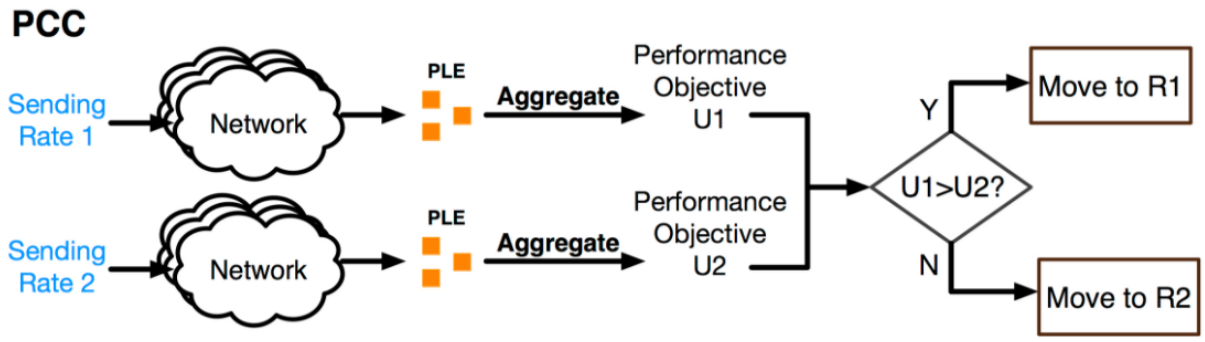
Transmission Control Protocol (TCP) Congestion Control new Architecture - Performance-oriented Congestion Control (PCC)

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).

PCC is designed and developed in the Academia. Its goal is to understand what rate will improve performance based on live experimental evidence. It tries out a high and low send rate, collects information on RTT, BW, and packet drops and then uses a fitness function to determine what to do.



PCC's control architecture is based on emprical observed performancee

Goals:

1. Review PCC - https://www.usenix.org/system/files/conference/nsdi15/nsdi15-paper-dong.pdf
2. Analyze the comparison to BBR - https://groups.google.com/forum/#!topic/bbr-dev/j7FITaY2V3M
3. Download the latest PCC code from https://github.com/modong/pcc
4. Raise Mininet topology with:
    a. Hosts instrumented with PCC
    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
5. Perform comparison between TCP (CUBIC) and PCC

Requirements:

Internet Networking Course