Lecture 9 (Transport Layer pt 4)

1 Principles of Congestion Control

Network unable to handle the number of packets that are present, it is different from flow control (end node faces flow issues)

1.1 Causes and Cost

1. Limited link capacity (will cause an issue even with infinite buffer size) - delay exponentially rises with input throughput
2. Limited buffer capacity
   i. when sender has complete knowledge of buffer, $\lambda'_{in} = \lambda_{out}$ (no loss)
   ii. when sender has partial knowledge, packer loss happens and $\lambda'_{in} - \lambda_{out}$ is kind of exponential
   iii. when sender has no knowledge, duplicate transmission occurs and difference is even larger
   iv. when sender has high transmission rate, other clients' flows are obstructed

2 TCP Congestion Control

1. Determines congestion based on packet loss and delay (end to end)
2. Follows AIMD (Additive Increase Multiplicative Decrease) policy - increase sending rate until packet loss and then decrease sending rate
3. $TCP\ rate = \dfrac{cwnd}{RTT}$
4. Slow start phase is exponential increase, `cwnd` doubles until drop or slow start threshold
5. On drop, `cwnd` is halved and then AIMD followed (TCP Tahoe [old version] resets `cwnd` to 1 though, current version is TCP Reno)
6. `ssthresh` is also halved on packet loss

2.1 TCP Cubic

1. Idea is to probe for usable bandwidth and increase faster than linear towards W_{max} initially and slow down closer to it

2. K is the time when `cwnd` will reach W_{max}, increase W as function of cube of distance between current time and K
3. Default in Linux and very popular around

2.2 Delay-Based TCP Congestion Control

1. Idea is to ensure RTT doesn't change since congestion increases RTT largely
2. Keep the pipe "just full enough, but no fuller"
3. Try to change `cwnd` to make measured throughput converge towards uncongested throughput (using RTT_{min})
4. Google uses a similar idea called TCP BBR (internal backbone network)

3 Explicit Congestion Notification (ECN)

Network layer conveys this information to TCP using an ECN bit and values in the header