Controlling Congestion

Q13.1 How does TCP accomplish a linear increase?

The way we accomplish this linear increase is by having the window `cwnd` increase by 1/`cwnd` for each acknowledgement that comes in. To see that this is equivalent, note that if `cwnd` is 3, then we have three outstanding packets. For each one that comes back, it will increase by 1/3: from 9/3 to 10/3, 11/3, and finally 12/3, or 4. We can't send a fraction of a packet, so we don't see the effect of the increase until it reaches 4. Then, we are allowed 4 outstanding packets, and for each that comes back, `cwnd` will increase by 1/4: from 16/4 to 17/4, 18/4, 19/4, and finally 20/4, or 5. You can see this process in the space-time diagram in Illustration 36, where space is the network between the two end-hosts.

Q13.2 What extra steps could cause the evolution of the window size up to 10 RTTs in the TCP Reno example?

- At Time 6, the sender has received five ACKs, so it sends six packets.
- At Time 7, the sender detects a lost packet. It halves the window size to three and sends three packets.
- At Time 8, the sender has received three ACKs, so it sends four packets.
- At Time 9, the sender has received four ACKs, so it sends five packets.

Q13.3 What extra steps could cause the evolution of the window size up to 13 steps in the TCP Vegas example?

- At Times 7 and 8, we get the seventh and eighth packets back. Witnessing that we are again above the threshold, we continue to decrease the window, first to 7, and then to 6. We send no new packets during this time.

- At Time 9, we get an acknowledgement for the ninth packet, with an RTT of 51.3 ms. So, 6 packets/51.3 ms $\approx$ 117 packets/sec is the current throughput, while the no-congestion throughput is 6/50 ms = 120 packets/sec. The difference is exactly the threshold, so the window size stays constant at 6, and the end host send one more packet for a total of six outstanding.
- At Times 10 and 11, the 10th and 11th packets come back, and the difference is lower than the threshold. The window is then increased to 7 and 8, respectively.
- At Times 12 and 13, the difference is exactly the threshold, so the window size is kept the same.

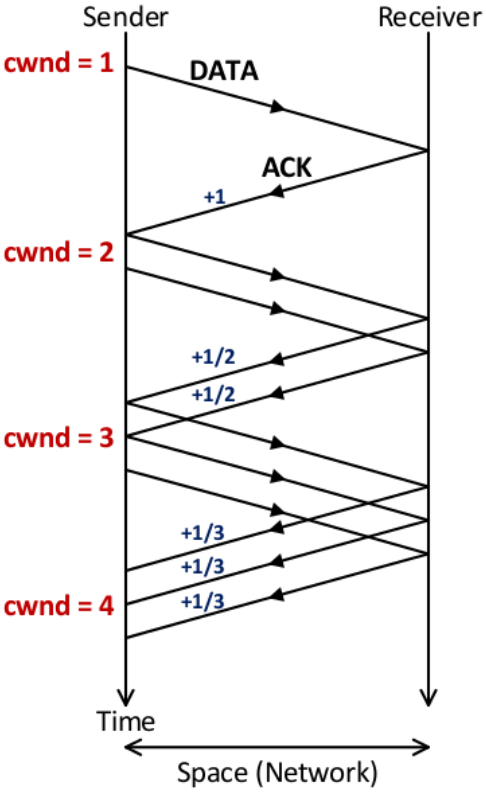

Illustration 36: A space-time diagram of TCP packets being sent and acknowledged. Space is the network in between the two end-hosts. The window starts at 1, and when this packet is acknowledged, the sender increases it to 2. When these two packets are acknowledged, it goes up to 3, and so on. This illustrates the additive-increase feature of TCP windowing when there's no congestion. We also show how much each acknowledgement adds to the window size, *i.e.*, 1/`win`.