National University of Singapore
School of Computing
CS5229: Advanced Computer Networks
Semester I, 2021/2022

Lecture 2 Training
Window-Based End-to-End Congestion Control

Release date: 20th August 2021
Due: 26th August 2021, 23:59

In Lecture 2, we discussed how TCP does a slow start in the beginning of the transmission. In this lecture training, we will investigate how a sender's congestion window (cwnd) evolves during the slow start (on a per-RTT basis) and how it impacts network congestion. We will also examine the effect of the initial congestion window size on the slow start and how it impacts user experience, especially during web browsing. Consider the following network scenario:

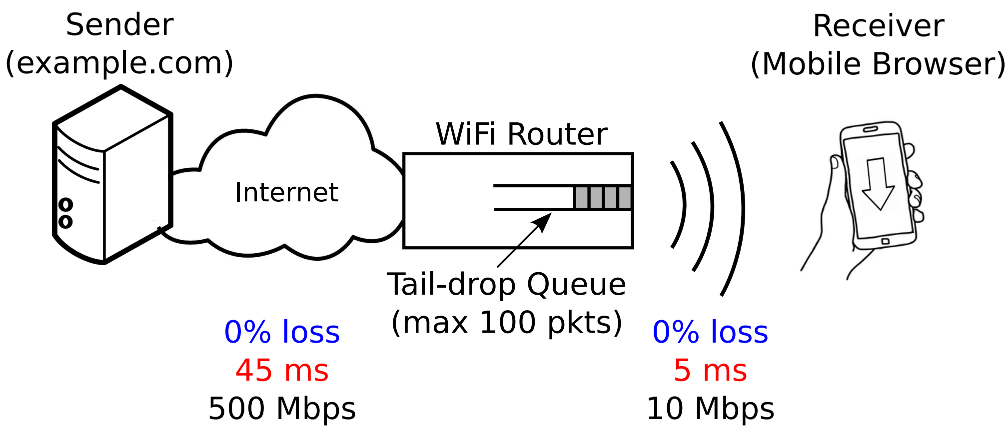

Figure 1: A simple network scenario

The mobile browser (TCP receiver) in Figure 1 wants to download a webpage from a server hosting `example.com` (TCP sender).

Unlike Lecture 1 training, assume that there are no transmission losses in the network for simplicity. Packet loss can only happen when the queue on the WiFi router overflows (> 100 packets). The link speeds are in Mega bits per second, where 1 Mbps = 1,000,000 bits per second. Also, assume that the WiFi router adds no hop delay or queueing delay (no matter the queue size). We consider the size of each data packet to be 1,500 bytes. Furthermore, assume[1] that each packet can carry 1,500 bytes of TCP payload. For example, the sender would need 2 packets to transmit a webpage of 3 KB (3,000 bytes) to the receiver.

The sender uses the following slow start mechanism which proceeds in time intervals of the RTT: (i) In the first RTT interval, the sender's initial cwnd is equal to 2 packets. The sender bursts these 2 packets and receives corresponding 2 ACKs from the receiver. (ii)

[1]In practice, a typical MTU-sized packet is 1,518 bytes long "on the wire" and can carry only 1,460 bytes of TCP payload: Ethernet header & checksum (14 + 4 = 18 bytes) + IP header (20 bytes) + TCP header (20 bytes) + TCP payload (1,460 bytes)

For each subsequent RTT interval, the sender increases the `cwnd` by 1 packet for every ACK it receives during the previous RTT interval. (iii) The receiver always sends 1 ACK for each packet received from the sender. This means that, the sender would effectively double its `cwnd` in each subsequent RTT interval. (iv) The sender would exit the slow start and switch to congestion avoidance if/when it detects the first packet loss.

Given this information, answer the following questions on Coursemology:

Q1 In which RTT[2] will the queue in the WiFi router start building up (queue size greater than 0)? Note that the RTT numbering starts with 1 i.e. in RTT #1, the sender sends 2 packets and receives 2 ACKs.

Q2 In which RTT will the first packet drop occur at the WiFi router?

Q3 Suppose the mobile browser needs to download a webpage of 90 KB (90,000 bytes) from `example.com`. How many RTTs would it take to complete the webpage download? A webpage download is considered to be "complete" when the sender receives the ACK for the last packet sent.

Next, suppose we upgrade our WiFi router and now our wireless link has a link speed of 100 Mbps.

Q4 In this new scenario, in which RTT will the queue in the WiFi router start building up?

Q5 In which RTT will the first packet drop occur at the WiFi router?

Q6 Suppose we want to download a webpage of 90 KB (90,000 bytes) from `example.com` again. How many RTTs would it take to complete the webpage download?

Instead of upgrading the WiFi router, suppose we increased the sender's initial `cwnd` to 10 packets and kept the wireless link's speed at 10 Mbps.

Q7 In this case, how many RTTs will it take to download the same 90 KB webpage as above?

Solutions

Q1 The queue will start building up only when queue's enqueue rate is greater than its dequeue rate. Now, the queue's enqueue rate is equal to the sender's sending rate while its dequeue rate is equal to the wireless link's speed i.e. 10 Mbps.

Let's compute the sender's sending rate first. The TCP sender sends `cwnd` worth of packets over a time period of 1 RTT. Therefore, the sending rate is cwnd/RTT. However, since the `cwnd` changes during each RTT, the effective sending rate in each RTT depends on the `cwnd` in that RTT. The sender starts with an initial `cwnd` of 2 packets in the first RTT and doubles its `cwnd` in each subsequent RTT. Therefore, the sender's `cwnd` in the nth RTT is 2^n. The enqueue rate in the nth RTT would then be:

$$enqueue\ rate = sending\ rate = 2^n\, pkts/RTT_{100ms} \tag{1}$$

The unit here is to be read as "packets per RTT". We use the subscript 100 ms to denote that the RTT is 100 ms in this case. To compare this enqueue rate with the

[2] we refer to a RTT time interval simply by "RTT". Note that the time is divided into intervals of the RTT.

dequeue rate, we need to first convert the dequeue rate from Mbps to "packets per $\text{RTT}_{100\text{ms}}$".

$$\begin{aligned} dequeue\ rate = 10\,Mbps &= \frac{10 \times 10^6}{8}\,bytes/s = \frac{125 \times 10^4}{1500}\,pkts/s \\ &= \frac{833.33 \times 100}{1000}\,pkts/RTT_{100ms} = 83.33\,pkts/RTT_{100ms} \end{aligned} \tag{2}$$

To answer the question, we need to find the first RTT in which the enqueue rate would be greater than the dequeue rate:

$$\begin{aligned} &enqueue\ rate > dequeue\ rate \\ &2^n > 83.33 \end{aligned} \tag{3}$$

For $n = 7$, the enqueue rate is 128 pkts/$\text{RTT}_{100\text{ms}}$ which is greater than the dequeue rate. Therefore, the queue will start building up in the **7$^{\text{th}}$ RTT**.

Q2 To find the RTT in which the first packet drop will occur, we need to track how the queue will build over each RTT. As determined in Q1 above, for 10 Mbps wireless link speed, the queue will not start building until the 7$^{\text{th}}$ RTT. For example, in the 6$^{\text{th}}$ RTT, the sender will enqueue 64 packets while wireless link will dequeue 83.33 packets. The queue size at the end of an RTT is the queue size at the end of the previous RTT plus the extra packets added to the queue in the current RTT. Therefore, the queue size at the end of n$^{\text{th}}$ RTT can be written as:

$$queueSize_n = queueSize_{n-1} + \ extraPackets_n \tag{4}$$

Here, $extraPackets_n$ are the number of extra packets added to the queue in the n$^{\text{th}}$ RTT. The number of extra packets is the difference between the packets enqueued be the sender and the packets dequeued by the wireless link. Therefore,

$$extraPackets_n = 2^n - 83.33 \tag{5}$$

$$\therefore\ queueSize_n = queueSize_{n-1} + 2^n - 83.33 \tag{6}$$

Now, let's find the queue size at the end of the 7$^{\text{th}}$ RTT using equation 6. Note that the queue size at the end of the 6$^{\text{th}}$ RTT is zero:

$$queueSize_7 = queueSize_6 + 2^7 - 83.33 \tag{7}$$

$$= 0 + 128 - 83.33 = 44.67\,pkts \tag{8}$$

Let's continue this analysis to the further RTTs.

$$queueSize_8 = queueSize_7 + 2^8 - 83.33 \tag{9}$$

$$= 44.67 + 256 - 83.33 = 217.34\,pkts \tag{10}$$

Since the queue can only hold 100 packets, the first packet drop would occur in the **8$^{\text{th}}$ RTT**.

Q3 To transmit a 90 KB webpage, the sender would need to send 90,000/1500 = 60 packets. We know that in n$^{\text{th}}$ RTT, the sender sends 2$^{\text{n}}$ packets. Therefore, we need to sum up the total packets sent by the sender in each RTT starting from the beginning:

$$2^1 + 2^2 + 2^3 + 2^4 + 2^{\mathbf{5}} = 62\,pkts > 60\,pkts \tag{11}$$

Therefore, it would take **5 RTTs** to complete the webpage download.

Q4 In the new scenario, due to the increase in the wireless link speed, the dequeue rate has increased 10 times compared to Q1 i.e. it has now become 833.33 pkts/$\text{RTT}_{100\text{ms}}$. Therefore, on the same lines as equation 3, we need to find the first n for which the following equation becomes true:

$$2^n > 833.33 \tag{12}$$

For $n = 10$, the enqueue rate is 1024 pkts/$\text{RTT}_{100\text{ms}}$ which is greater than the dequeue rate. Therefore, in this case, the queue will start building up in the **10$^{\text{th}}$ RTT**.

Q5 For this question, we can do the same analysis as in Q2. The only thing different would be that the wireless link would dequeue 833.33 packets in every RTT. Also, from Q4, we know that the queue will start building up from the 10$^{\text{th}}$. Therefore, using equation 6 (modified: 83.33 $\rightarrow$ 833.33) for the 10$^{\text{th}}$ RTT, we get a queue size of:

$$\begin{aligned} queueSize_{10} &= queueSize_9 + 2^{10} - 833.33 \quad (13) \\ &= 0 + 1024 - 833.33 = 190.67\, pkts \quad (14) \end{aligned}$$

Since the queue can only hold 100 packets, the first packet drop would occur in the **10$^{\text{th}}$ RTT**.

Q6 Same as Q3 i.e. **5 RTTs**. Upgrading the WiFi router for 100 Mbps wireless link speed does not change how the `cwnd` would evolve during the slow start. To affect the evolution of `cwnd` during the slow start, a packet loss needs to occur. There is no packet loss happening in the 10 Mbps case which could have been averted by increasing the wireless link speed to 100 Mbps and delaying the packet loss to a further RTT. In fact, at 10 Mbps, 60 packets (the 90 KB webpage) are transmitted in 5 RTTs, which is even before the queue would start building in the 7$^{\text{th}}$ RTT. Therefore, with 100 Mbps wireless link, it would still take 5 RTTs to complete the webpage download since the `cwnd` will evolve exactly in the same way as it would with the 10 Mbps wireless link.

Q7 With 10 packets as the initial `cwnd`, the packets sent in successive RTTs would be 10, 20, 40 and so on. Since 10 + 20 + 40 > 60 pkts, it is clear that in this case it would take **3 RTTs** to complete the webpage download. This is a 40% improvement compared using an initial `cwnd` of 2 packets.