CS3611 Computer Networks (Spring 2023) Homework 3

Xiangyuan Xue (521030910387)

1. Yes. Both of these segments will be directed to the same socket at Host C. Although Host A and Host B use identical port numbers, their source IP addresses are different. Hence, the process at Host C can distinguish the two segments correctly.

2. (1) Source port number: 3366
   Destination port number: 23

   (2) Source port number: 4477
   Destination port number: 23

   (3) Source port number: 23
   Destination port number: 3366

   (4) Source port number: 23
   Destination port number: 4477

   (5) Yes. A and B can use identical port numbers since they are different hosts.

   (6) No. Using identical port numbers on the same host will lead to conflicts.

3. UDP offers a better control of what and when data is sent in a segment.
   - Since UDP does not maintain the state of a connection, it is easier for developers to implement applications.
   - UDP is free of connection establishment and retransmission, which promises a lower latency compared to TCP.
   - UDP has a smaller header to transmit and does not include flow and congestion control, which can seize larger bandwidth and transmit efficiently.

4. (1) The time required to use up the sequence number is

$$T_1 = \frac{2^{32} \times 8\text{bits}}{336\text{kbps}} \approx 28.4\text{hours}$$

   (2) The time required to use up the sequence number is

$$T_2 = \frac{2^{32} \times 8\text{bits}}{30\text{Mbps}} \approx 19.1\text{minutes}$$

   (3) The time required to use up the sequence number is

$$T_3 = \frac{2^{32} \times 8\text{bits}}{2\text{Gbps}} \approx 17.2\text{seconds}$$

(4) No. The sequence number will be used up in about 2.9 seconds. If some packet stays in the network for several seconds, duplicate sequence numbers will occur. To deal with this problem, TCP introduces a timestamp field in the header to uniquely identify each packet.

5. (1) TCP may split the data into multiple segments for buffer transmission, where the data have been differently encapsulated. While UDP put the data into a single segment, so the data transmitted are not fragmented. Hence, UDP applications have a better control of what data is sent in a segment.

   (2) Since UDP includes neither connection establishment and retransmission nor flow and congestion control, it has a much lower transmission latency. Hence, UDP applications have a better control of when the segment is sent.

6. (1) GBN: Host A sent 12 segments in total. They are $1, 2, 3, 4, 5, 6, 7$ and again $3, 4, 5, 6, 7$. Host B sent 11 ACKs in total. They are 1 and then 2 for 5 times and then $3, 4, 5, 6, 7$.

   SR: Host A sent 8 segments in total. They are $1, 2, 3, 4, 5, 6, 7$ and again 3. Host B sent 7 ACKs in total. They are $1, 2, 4, 5, 6, 7$ and then 3.

   TCP: Host A sent 8 segments in total. They are $1, 2, 3, 4, 5, 6$ and again 3 then 7. Host B sent 7 ACKs in total. They are 2 and 3 for 4 times and then $7, 8$.

   (2) TCP. Instead of waiting until time out, TCP retransmits the 3rd segment right after three duplicate ACKs, which is much faster than GBN and SR.

7. (1) TCP slow start is operating at the intervals $[1, 6]$ and $[23, 26]$.

   (2) A triple duplicate ACK. Because CWND drops by $\frac{1}{2}$.

   (3) A timeout. Because CWND drops to 1.

   (4) The value of ssthresh is initially 32, which can be observered at the 6th round.

   (5) The value of ssthresh is 21 at the 18th round, which is half of the CWND at the 16th round.

   (6) The value of ssthresh is 14 at the 24th round, which is half of the CWND at the 22th round.

   (7) Segment 1 is sent at the 1st round. Segments $2, 3$ are sent at the 2nd round. Segments $4, 5, 6, 7$ are sent at the 3rd round. Segments $8, 9, \ldots, 15$ are sent at the 4nd round. Segments $16, 17, \ldots, 31$ are sent at the 5nd round. Segments $32, 33, \ldots, 63$ are sent at the 6nd round. Segments $64, 65, \ldots, 96$ are sent at the 7nd round.

   Therefore, the 80th segment is sent at the 7th round.

   (8) The value of ssthresh will be 4, which is half of the CWND at the 26th round. Then CWND will be $4 + 3 = 7$.

   (9) The value of ssthresh is 21 at the 19th round, which is half of the CWND at the 16th round. Then CWND drops to 1 and do the slow start from the 17th round. Hence, CWND will be 16 at the 21st round.

8. Since the TCP receive buffer is large enough to hold the entire file, it is not the TCP flow control that prevent the process continuously passing data at a fast rate.

   Since there is never any packet loss and the timers never expire, it is also not the TCP congestion control that prevent the process continuously passing data at a fast rate.

   The bottleneck is the TCP send buffer. Since $S >> R$, the send buffer with limited capacity will be filled up quickly. Therefore, the rate that the process passes data to its TCP socket is limited to R bps, which is much slower than S bps.