Computer Networking: Supervision 5

Daniel Chatfield

February 19, 2015

Topic 05 - Transport

1. *Transport flavours*
   (a) The User Datagram Protocol (UDP) is sometimes used instead of TCP. It provides very few features above those provided by IP.
      i. Give one feature provided by UDP but not by IP, and one provided by UDP but not by TCP.

      > Provides multiplexing/demultiplexing capabilities.

      ii. Explain the role of a port number in UDP. Why isn't it the Process ID? Explain the relation between TCP and UDP port numbers.

      > By using port numbers multiple flows can happen simulataneously from a single IP address. A single process might want multiple flows and thus using the process ID is not as flexible as assigning a port number.

      iii. Give three characteristics which might make an application protocol better suited to implementation over UDP than over TCP.

      > A real time stream (e.g. an online multiplayer game). If the packet is not delivered on time then retransmision is useless as the data is now stale.

2. *Error control, ARQ*
   (a) Why do error control protocols for packet switched networks use error detecting codes but not error correcting codes?

In a packet switched network the link is likely to either be fairly reliable or lose packets. In the first case error correcting codes add more overhead than simply retransmitting the corrupted packet. In the second case, since the entire packet is lost the codes are useless.

(b) A transport protocol for packet-switched networks uses a "sliding window" Automatic Repeat reQuest (ARQ) scheme for error control and flow control.

i. As well as error detecting codes, ARQ protocols use acknowledgments and timeouts to achieve error control. Briefly explain what these are, and how they are combined to achieve reliable transmission.

Acknowledgments Acknowledgments are packets sent by the reciever to confirm receipt of uncorrupted packets.

Timeouts In the context of an ARQ protocol, a timeout is used to trigger retransmission (i.e. when it can be assumed from the lack of an acknowledgment that the packet was lost).

In TCP acknowledgements are sent with the byte offset for the next packet it is expecting. If the sender recieves 3 ACKS with the same byte offset then it assumes that this is due to a lost packet (rather than duplicated ACKS or just out of order packets) and this triggers fast retransmission.

ii. What *two* error cases might cause a receiver to send a negative acknowledgment (NACK)? How are they detected? What happens if the NACK is lost?

NACKs can be sent for:

- Out of sequence packets
- If a received packet fails checksum verification

If a NACK is lost then the sender will retransmit when the timeout fires.

iii. In what circumstances will a receiver receive a packet with the same sequence number twice? What should it do in these circumstances?

This could be caused by:

- A lost or delayed ACK which resulted in unnecessary retransmission
- A duplicated packet

In these circumstances it should simply discard the second.

iv. Given that the protocol provides bidirectional communication, what optimisation can be made in the implementation of acknowledgments to reduce the total number of packets sent?

The acknowledgment packets can be piggy-backed onto other packets.

v. If two hosts are connected by a 100Mbps link with a round-trip time of 20ms, how big (in bytes) should the sliding window be to maximise link usage?

The bandwidth-delay product should be used: $100\,\text{Mbps} \times 20\,\text{ms} = 2Mb$ ($250\,\text{bytes}$)

vi. Give two reasons why, at a given time, the window size might be set to a smaller value.

- The receiver has limited the rate due to buffer limitations.
- There is contention for capacity (causing some packets to be lost).

(c) Consider a sliding window protocol with a window size of 5 using cumulative ACKs.

Retransmissions Retransmissions occur under two conditions:

- Reception of three duplicate ACKs (that is, three identical ACKs after the initial ACK)
- Timeout after 100 ms (timer starts at the beginning of the packet transmission)

Timing
- Data packets have a transmission time of 1 ms.
- ACK packets have zero transmission time.

- The link has a latency of 10 ms.
- The source A starts off by sending its first packet at time $t = 0$.

i. Assume all packets are successfully delivered except the following:

   - The first transmission of data packet #3
   - The ACK sent in response to the receipt of data packet #6

   When is data packet #3 first retransmitted (expressed in terms of msec after $t = 0$)?

   The last bit of data packet #1 reaches the destination at 11 ms (1 ms for transmission and 10 ms for latency).

   Similarly, the time for the other packets are 12 ms, *N/A*, 14 ms and 15 ms respectively.

   The acknowledgment packets for data packets #1, #2, #4 and #5 have sequence numbers 2, 3, 3 and 3 and arrive at times 21 ms, 22 ms, 24 ms and 25 ms respectively.

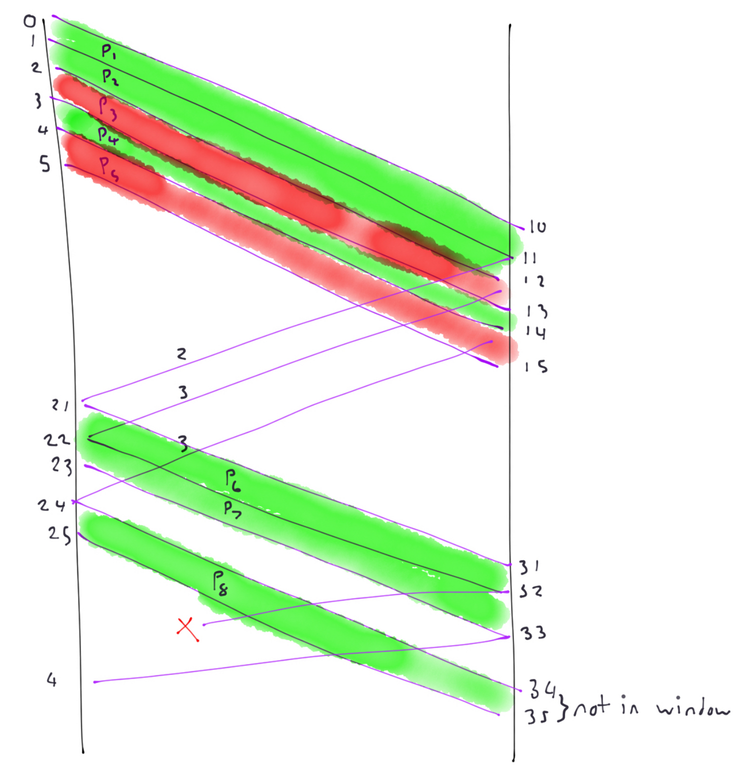
   When the acknowledgments for data packets #2 and #3 reach the sender they trigger packets 6 and 7 to be sent.

   Packet 7 reaches the reciever at time 33 ms and thus the acknowledgment reaches the sender at 43 ms. Since this is the third duplicate it triggers the retransmission.

ii. Consider the same scenario, but with everything successfully delivered except the following:

   - The first transmission of data packet #3
   - The first transmission of data packet #5
   - The ACK sent in response to the receipt of data packet #6

   When is data packet #3 first retransmitted (expressed in terms of msec after $t = 0$)?



No further transmission takes place until the timeout for packet fires at 102 ms

iii. Assume we can only observe the ACK packets arriving at the sender.

The same sliding window algorithm is used, with the same timings and retransmission policies apply.

Notation (read carefully): The notation A_x is used to mean that the ACK packet is acknowledging the receipt of all packets up to and including data packet x.

That is, A_5 is acknowledging the receipt of packet 5; to be clear, the notation does not mean that the receiver is expecting packet 5 as the next data packet.

Assume that the following ACK packets arrive (just the ordering is shown, no timing information is provided):

- A_1
- A_2
- A_3
- A_3
- A_4

- A_5
- A_6

Which of the following five scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network

(d) ACK packet A_3 was duplicated by the network

(e) ACK packet A_4 was delayed, arriving after A_5

C and D apply.

iv. With the same set up as in the previous problem, consider the following stream of ACK packets.

- A_1
- A_2
- A_3
- A_5
- A_4
- A_6

Which of the following five scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network

(d) ACK packet A_3 was duplicated by the network

(e) ACK packet A_4 was delayed, arriving after A_5

E is the only scenario that applies.

v. With the same set up as in the previous problem, consider the following stream of ACK packets.

- A_1
- A_2
- A_3
- A_3
- A_5
- A_6

Which of the following five scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network

(d) ACK packet A_3 was duplicated by the network

(e) ACK packet A_4 was delayed, arriving after A_5

B is the only scenario that applies.

3. *TCP specifics*

(a) Use the approximate equation for throughput as a function of drop rate:

$$throughput = \frac{\sqrt{1.5} \times MSS}{RTT\sqrt{p}}$$

Assume an RTT of 40 ms and an MSS of 1000 bytes. In the following questions ignore IP and TCP headers in your calculations.

i. What drop rate p would lead to a throughput of 1 Gbps?

Rearranging gives:

$$p = \left(\frac{\sqrt{1.5} \times MSS}{RTT \times throughput} \right)^2 = 6 \times 10^{-8}$$

ii. What drop rate p would lead to a throughput of 10 Gbps?

Rearranging gives:

$$p = \left(\frac{\sqrt{1.5} \times MSS}{RTT \times throughput} \right)^2 = 6 \times 10^{-10}$$

iii. If the connection is sending data at a rate of 10 Gbps, how long on average is the time interval between drops?

The number of packets per second is given by:

$$\frac{throughput}{MSS \times 8}$$

Multiplying this by the drop rate gives you the number of packets dropped a second and the reciprical of this is the average time interval between dropped packets.

$$\frac{MSS \times 8}{throughput \times p} = 1333\,\mathrm{s}$$

iv. What window size W (measured in terms of MSSes) would be required to maintain a sending rate of 10Gbps? (rounded down to the nearest integer)

The window size neccessary to fill the bandwidth-delay product is:

$$W = \frac{throughput \times RTT}{MSS \times 8} = 50,000$$

v. If a connection suffered a drop upon reaching 10Gbps, how long would it take for it to return to 10Gbps (after undergoing a fast retransmit)? (in seconds, rounded down to the nearest second)

☐

vi. Consider two TCP connections whose throughput obeys the TCP throughput equation listed above.

The first TCP connection has the following parameters: $MSS = 1000 bytes$, $RTT = 0.2ms$, $droprate = 0.5\%$

The second TCP connection has the following parameters: $MSS = 2000 bytes$, $RTT = 0.1ms$, $droprate = 8\%$

What is the ratio of throughputs (the throughput of the first TCP connection divided by the throughput of the second TCP connection)? Why?

> The ratio is 1. Not sure how to justify this - I just plugged the numbers in.

(b) Consider the plot of CWND versus time for a TCP connection.

i. At each of marked marked points along the timeline in the figure on the next page, indicate what event has happened, or what phase of congestion control TCP is in (as appropriate), from the following set: Slow-Start, Congestion-Avoidance, Fast-Retransmit, and Timeout.

> (a) Slow-Start
> (b) Fast-Retransmit
> (c) Congestion-Avoidance
> (d) Fast-Retransmit
> (e) Congestion-Avoidance
> (f) Timeout
> (g) Slow-Start

ii. Assume CWND is $10,000$ right before F, what is the value of SSTHRESH at G?

> $5,000$