Tutorial 1 Solutions (Week 5)

Note: Some questions are from past exams

Section I - Multiple Choice, Fill-in Questions

Instructions: Circle the letter beside the choice that is the best answer for each question. For multiple choice, choose ***only ONE answer unless specifically asked to do otherwise***. For Fill-in and Short Answer questions, provide ***ONLY the number of answers requested*** in the spaces indicated.

1. List the four different types of delays encountered in packet switched networks:
   a. processing delay
   b. queuing delay
   c. transmission delay
   d. propagation delay

2. Which of the following services does the Internet network layer provide for the Internet transport layer? (possible multiple answers)
   a. in-order delivery of data segments between processes
   b. best effort delivery of data segments between communicating hosts
   c. multiplexing and demultiplexing of transport layer segments
   d. congestion control

3. Consider the operation of downloading a Web page consisting of an index page that references 3 JPEG objects. Ignoring latency involved in transferring the objects themselves, fill in the blanks below with the correct values:
   a. Utilizing HTTP/1.0 with no parallel connection capability, the number of RTTs required to download the page is 8RTT.
   b. Utilizing the default mode of HTTP/1.1, 3 RTTs are required to download the page.

4. Consider a TCP connection between sender A and receiver B. Sender A sends a 900 byte TCP segment with sequence number 3100 and header length 20 bytes. What acknowledgement number will receiver B reply with to inform sender A that it has received this segment correctly and in order? (Ignore the possibility of a cumulative ack for this question.)

   Answer:

3980

5. DNS responses have a TTL field. Why is this necessary?
   a. The TTL field is decremented at each DNS server that the response passes through on its way to the client, and servers drop responses with a TTL of 0, so the TTL field prevents responses from looping indefinitely.

b. The TTL field allows DNS servers to prevent cache poisoning.

c. The TTL field is necessary for scalability: if DNS servers could never time out entries, over time they would accumulate infinite state.

d. The TTL field causes DNS servers to delete entries after some time, so that if the host moves and the underlying address changes, the server will eventually get the correct address.

Section II - Numerical Problems

Instructions: Calculate the values requested and provide a *numeric answer* for each question. You may use a calculator if desired, but problems have been developed in such a way that calculators should not be required. *Show your work* for each problem. Select the numeric result of your calculations from the choices provide, or fill in the blanks where requested.

1. Calculate the ***end-to-end delay***, $d_{end\text{-}end}$, between the source host and the destination host in a network with 4 routers between source and destination? Assume that the network is NOT congested (i.e. d_{queue} is insignificant), and that:
    i. all packets are 10,000 bits in length,
    ii. each link is 5 kilometers long,
    iii. the processing time is 10msec at the source host and at each router,
    iv. the transmission rate of each link is 1Mbps,
    v. the propagation speed of each link is 2.5 x 10^8 meters/second.

    CALCULATIONS:

The delay along one hop = dproc + dprop + dtrans = 10 msec + 5 x 10^3/2.5 x 10^8 + 10,000/1 x 10^6 = 10 msec + 0.02msec + 10msec = 20.02 msec

The packet wil be transmitted along 5 hops (first hop is from the source, and once at each router).

Thus the end-to-end delay = 5 x 20.02 = 100.1 msec

ANSWER:

a. 88 milliseconds

b. 100.1 milliseconds

c. 110 milliseconds

d. 1.21 seconds

2. Given that the previously calculated values for Estimated RTT and RTT Deviation are as shown below, and that the new sample RTT shown has just been measured, what Timeout interval will TCP use for the ***next*** transmitted segment?
    i. EstimatedRTT (k) = 4 msec
    ii. DevRTT (k) = 2 msec
    iii. new SampleRTT = 8 msec

iv. α = .125
v. β = .25

CALCULATIONS:

Estimated RTT = (1-0.125)(4) + 0.125(8) = 4.5
DevRTT = (1-0.25)(2) + 0.25(8-4) = 2.5
Timeout = 4.5 + 4(2.5) = 14.5

ANSWER:

a. 8.5 milliseconds
b. 9.0 milliseconds
c. 12.75 milliseconds
d. 14.5 milliseconds

Section III – Short Questions

Instructions: Answer the questions. *Show your work* for each problem.

1. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010101, 01110000, 01001100. What is the 1's complement of the sum of these 8-bit bytes? (Note although TCP and UDP use 16-bit words in computing the checksum, for this problem we will only consider 8-bit summands). Show all work. Is it possible that a 1-bit error will go undetected by the checksum? How about a two-bit error?

Answer:

```
   0 1 0 1 0 1 0 1
 + 0 1 1 1 0 0 0 0
 -----------------
   1 1 0 0 0 1 0 1
```

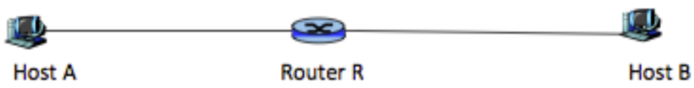
```
  11000101
 +01001100
 ---------
  00010001
```

The last addition has a overflow so we add it to the above sum resulting in a final rsult of 00010010. One's complement of this is 11101101 which is the final checksum.

All one-bit errors will be detected by checksum, but two-bit errors can be undetected (e.g. if the last digit of the first word is converted to a zero and the last digit of the second word is converted to a 1).

2. Consider two hosts A and B communicating with each other using the Go-Back-N data transfer protocol. The hosts have two bi-directional links and one store-and-forward router, R between them as depicted in Figure below. The propagation delay across each link is D in each direction. The links have a transmission rate of R bps in each direction. Assume that each data packet is of size P bits, and that acknowledgements are of negligible size. There is no other traffic on the network. In

the absence of errors or packet losses, and if the hosts always have data available to transmit, what is the minimum value of window size so that the hosts never have to be idle without transmitting? Assume that processing and queueing delays are negligible. Justify your answer.



Answer:

Let us assume that N is the size of the sender window.

The total delay for a packet to travel from host A to the router = transmission delay for the packet + propagation delay = P/R + D.
The delay for the packet to reach host B = transmission delay for the packet + propagation delay = P/R + D.
Thus, the total delay for a packet to travel from host A to B = 2D + 2P/R.

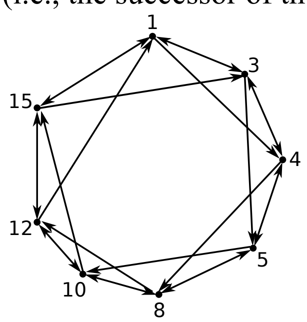
The total delay in transmitting an acknowledgement packet from B to A is equal to 2D, since the time transmit the acknowledgment is negligible.

Hence, the total RTT between A and B = 4D + 2P/R

In order for the sender to transmit continuously, the size of the sender window N should be such that there are sufficient packets waiting to be transmitted until the subsequent ACK packets start arriving from the destination. In other words, the sender window should be sufficiently large so that host A is still transmitting after the RTT (which is when the subsequent ACK arrives from the destination), i.e. N (P/R) >= 4D + 2P/R

Thus, N >= (4D + 2P/R) / (P/R) i.e. N >= 4DR/P + 2.

3. Consider the circular DHT with shortcuts in Figure below, where each node in the DHT also keeps track of (i) its immediate predecessor, (ii) its immediate successor, and (iii) its second successor (i.e., the successor of the node's immediate successor).



a. Suppose that peer 1 wants to learn where file with content ID 9 is stored. Write down the sequence of DHT protocol messages that the nodes exchange until peer 1 discovers the location of the file.

Answer: The file is located at Peer 10. So the sequence of queries would be as follows: Peer1->Peer4->Peer8->Peer10

b. Suppose that peer 3 learns that peer 5 has left. How does peer 3 update its successor state information?

Answer: Peer 3 would ask Peer 4 (its immediate successor) for its two successors. Peer 4 would reply indicating Peer 5 and 8. Peer 3 would then update its second successor to be Peer 8.

c. Now consider that the DHT nodes do not keep track of their second successor (the figure should look like Figure 2.27(a) from the book). Suppose that a new peer 6 wants to join the DHT and peer 6 initially only knows the IP address of peer 15. What steps are taken?

Answer:
Peer 6 will contact Peer 15 with a join request.
Peer 15, whose successor is peer 1, knows that Peer 6 should not be its successor. Peer 15 will forward the join request from Peer 6 to Peer 1.
Peer 1, whose successor is peer 3, knows that Peer 6 should not be its successor. Peer 1 will forward the join request from Peer 6 to Peer 3. The actions of peers 3 and 4 are identical to those of peers 15 and 1.
The join request will finally arrive at peer 5. Peer 5 knows that its current successor is peer 8, therefore peer 6 should become its new successor. Peer 5 will let peer 6 knows that its successor is peer 8. At the same time, peer 5 updates its successor to be peer 6.

4. Consider the Go-Back-N protocol with a sender window size of w and (a sufficiently large) sequence number range of size N. Suppose that at time t the next in-order packet that the receiver is expecting has a sequence number of k. Assume that packets cannot be re-ordered in the network.

a. What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer.

Answer: Here we have a window size of w. Since the receiver is expecting packet k, it has received packet k-1, and has ACKed that and all other preceding packets. If the sender has received all of these ACK's, then sender's window is [k, k+w-1].

Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains k-1 and w packets up to and including k-1. The sender's window is thus [k-w,k-1]. By these arguments, the senders window is of size w and begins somewhere in the range [k-w,k].

b. What are the possible values of the ACK field in all the acknowledgement packets currently propagating back to the sender at time t? Justify your answer.

Answer: If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the w-1 packets before that. If none of those w ACKs have been yet received by the sender, then ACK messages with values of [k-w,k-1] may still be propagating back.

Because the sender has sent packets [k-w, k-1], it must be the case that the sender has already received an ACK for k-w-1. Once the receiver has sent an ACK for k-w-1 it will never send an ACK that is less that k-w-1. Thus the range of in-flight ACK values can range from k-w-1 to k-1.