ECSE-4670: Computer Communications Netwoks Exam 1: SOLUTIONS

Time: **75 min (strictly enforced)**

Points: **50**

<u>YOUR NAME: XXXXXXX</u>

Be brief, but <u>DO NOT</u> omit necessary detail

{Note: Simply copying text directly from the slides or notes <u>will not</u> earn (partial) credit. Brief, clear and consistent explanation will.}

I. Below, you are given a true or false statement and asked a follow up question.

1. **[4 pts]** ***False statement:*** Connectivity is equivalent to a point-to-point physical link.

Qn: Explain how the notion of "connectivity" differs from an equivalent of a "physical link."

Connectivity refers to two hosts being able to communicate. Such communication could be established via a reliable transport protocol which can use underlying layers to reach remote hosts (those that are not directly connected by a physical link). Physical link has well-defined performance, but with connectivity performance is defined on a packet by packet basis.

Definition of connectivity and physical link -> 3pts

Any one alone -> 2pts

Difference is clearly articulated -> 4pts

2. **(6 pts) *False statement:*** Circuit-switching exploits statistical multiplexing and hence needs to tackle stability issues.

Qn: Explain:

a) Why circuit switching DOES NOT exploit statistical multiplexing, and

b) Why do statistically multiplexed systems (like packet-switched systems) need to tackle stability issues unlike circuit-switched systems.

a) Circuit switching involves reserving some fixed bandwidth when the circuit is setup. But statistical multiplexing means sharing a link among various connections depending on the demands posed by the connections. Since circuit switching does not care if individual connections offer any demand and reserves bandwidth, it is clear that it does not use statistical multiplexing. [3pts]

b) The statistically multiplexed systems provide gains by trading off delays and queuing of packets. If the system does not estimate the load of the constituent connections properly, there is a possibility that an excess traffic might be admitted. Such a situation can cause unbounded delay and queuing leading to system instability.

[3pts]

3. **(6 pts)** ***True statement:*** HTTP uses TCP. DNS uses UDP

Qn: a) Discuss why HTTP & DNS need TCP and UDP respectively.

b) What could happen in DNS if its UDP-based request or response packet is lost?

a) DNS is invoked prior to establishing TCP connections and hence response time is important. It cannot tolerate large latencies due to TCP connection setup. Also HTTP typically transfers larger amount of data, and needs a reliable mechanism. DNS requests on the other hand, are short packets which give the IP address for a name and probably some authoritative name server addresses. So the reliable protocol semantics of TCP would be costly in the case of DNS. [3pts] partial is 2pts

b) If DNS request packets are lost the resolver would intimate that there is a temporary failure in name resolution or it could stall without any reply. If a retransmit mechanism has been built into it, it could probably retry after a timeout.[3pts] partial is 2pts

4. (6 pts) ***False statement:*** The checksum (as UDP would compute) of the three numbers:

0111011001110110, 0010111010101110, 1000011100001011 is 0011010100110101

Qn: What is the correct checksum ? Show how you obtained your answer. Correct answer – 6pts. Correct procedure but wrong answer - 4pts. Reasonable attempt – 2pts

1101001111010000

5. **(6 pts)** ***False statement:*** Cumulative ACKs, Selective ACKs and NAKs are all equivalent.

Qn: Discuss 2 pros and 2 cons EACH of Cumulative ACKs, Selective acks and NAKs.

a) Cumulative ACKs, pros:
   a) Fewer ACKs can be sent for the same number of packets, so there can be a saving in reverse channel bandwidth.
   b) Sender window can move faster since It does not have to wait for all ACKs
b) Cumulative ACKs cons:
   a) Does not help in reducing retransmissions. Although the receiver might have later packets, it would ACK an earler one if an intermediate packet is lost.
   b) Can cause bursty transmission, since the window can increase suddenly due to a cumulative ACK.
c) Selective ACKs pros:
   a) Indicates the packets that need to be retransmitted explicitly and avoids unnecessary retransmission.
   b) Fewer ACKs sent
d) Selective ACKs cons:
   a) Sequence number problems – if sequence numbers wrap there is no way to find out if the ACK is for an older session.
   b) Higher buffer requirements since out-of-order packets are not discarded.
e) NAK pros:
   a) Indicates only the packets that were not received or corrupted and hence prevents unnecessary retransmissions.
   b) Fewer "ACK"s, no packets sent to confirm a proper receipt, hence bandwidth saved.
f) NAK cons:
   a) If a NAK for a packet is lost, that packet might never be recovered.
   b) Higher buffer requirements since out-of-order packets need to be processed.

½ pt for each point in "pro" or "con"

II. [10 pts] Consider a 1 Mbps WAN channel with 10 msec propagation delay. Data packets are 1000 bits long while ACK/NAK packets are negligible in length. Window size (N) = 5 packets, and there are 10 packets to be transmitted. Assume a Selective Repeat ARQ protocol with a new addition: NAKs are sent for packets detected lost at the receiver.

a) Assume no errors and no lost packets or ACKs/NAKs. How much time is required to complete the transfer of the 10 pkts and receive the final ack.

b) Now assume every 9^{th} packet which crosses the forward channel is lost. ACKs/NAKs are not lost or corrupted. How much time is required to complete the transfer of the 10 pkts and receive the final ack.

a) First packet of first window receives an ACK at 21ms. So first packet of 2^{nd} window receives an ACK at 42ms and the last packet receives an ACK at 42+4 = 46ms.

b) Retransmit caused – so 46+21 = 67ms are required.

c) 5pts each, partial 3pts

d) If they consider the 10^{th} packet as being lost, then they should get the result as given by the following formula:

Total = 46+ (Timeout)

The timeout should be greater than 21ms.

III. [7 pts] TCP RTT estimation and Timeout Setting

Assume a TCP flow has samples of RTT: (1, 3, 2) expressed in units of seconds. What is the Average RTT, Deviation and value of Timeout calculated by TCP when it receives all these samples? Assume that the initial value of Average RTT is 3s, initial value of Deviation is 0. EWMA parameter is 0.1.

Diff = RTT - SRTT

SRTT[n] = SRTT[n-1] + 1/10 * diff

Dev[n] = Dev[n-1] + 1/10 * (|diff| – dev[n-1])

All right -> 7pts, correct procedure -> 5pts, reasonable try -> 3pts

Iteration	RTT	Diff	SRTT	Dev
0	-	-	3	0
1	1	-2	2.8	0.2
2	3	0.2	2.82	0.2
3	2	-0.82	2.738	0.262

Timeout = 2.738 + 4*0.262 = 3.786s

IV. **[5 pts]** a) Why is UDP necessary when IP already provides a connectionless delivery service ?

b) Give an example using UDP sockets between the same pair of client and server machines. In your example three of the four fields (IP address1, port1, IP address2, port2) should be the same, but one field should be different. Explain how multiplexing & de-multiplexing is correctly done in this case.

a) IP provides protocol multiplexing, while UDP provides port (application) multiplexing. If the application were to receive all IP packets it would have to sort out packets which belong to it which can be both undesirable and time consuming.

b) Consider (client, port1, server, port2) and (client, port3, server, port2). There are 2 clients exchanging UDP packets with a server port. The fact that the client side port numbers are different is sufficient for demultiplexing the packets. 2 pts for a) and 3 pts for b).