# CSE 473S – Introduction to Computer Networks – Fall 2006

## Exam 3

## Problems, Solutions, and Grading Guidelines

Problem 1 (10 points). What does it mean that an HTTP connection is **persistent** with **pipelining**?

**Solution**:

**Persistent** *means that the server leaves the TCP connection open after sending a response.*

**Pipelining** *means that the client issues a request as soon as it encounters a reference without waiting for a response to a previous request.*

**Grading guidelines**:

5 points for a correct explanation of **persistent**

5 points for a correct explanation of **pipelining**

Problem 2 (15 points). Satellite $A$ sends station $B$ an infinite-size message over an asymmetric link. The link has capacity $R_1 = 10$ Mbps in direction $AB$ and capacity $R_2 = 100$ Kbps in direction $BA$. The propagation delay of the link is $d = 120$ ms in each direction. The communication uses the Go-Back-N protocol. In this protocol, the receiver responds to delivery of a data packet by sending an acknowledgment packet, and the sender does not transmit more than $N$ unacknowledged data packets, where $N$ is a finite integer. The overall size of a data packet is $L = 1,500$ bytes (remember that 1 byte = 8 bits whereas link capacities are measured in bps, or bits per second). The overall size of an acknowledgment packet is $l = 40$ bytes. There is no other traffic on the link. Nodal processing delays are negligible. Let $U$ be the utilization (in %) of the link from $A$ to $B$. What is the maximum possible value of $U$? What is the smallest $N$ that provides this value of $U$?

Show your derivations. Use mathematical expressions, not their verbal descriptions. Simplify your final analytical expression. Explain (briefly) your reasonings.

**Solution**: *Since* $\frac{l}{R_2} > \frac{L}{R_1}$*, transmission of acknowledgment packets from B to A becomes a bottleneck for utilizing the link from A to B efficiently after the initial window of N data packets is transmitted. When B transmits acknowledgment packets continuously, A responds to an acknowledgment packet of size l by sending a data packet of size L. Therefore, the maximum possible utilization of the link from A to B is* $U_{max} = \frac{L \cdot R_2}{l \cdot R_1} = 37.5\%$.

*The protocol provides this maximum utilization if B does not finish transmitting an acknowledgment packet for data packet N (i.e., the last packet from the initial window) before B receives data packet N + 1. Hence, N should be such that* $N\frac{l}{R_2} \geq \frac{l}{R_2} + \frac{L}{R_1} + 2d$*. This expression can be transformed to to* $N \geq 1 + \frac{R_2}{l}(\frac{L}{R_1} + 2d)$*. Then, the smallest value of N that provides the maximum utilization is* $N_{min} = 1 + \lceil \frac{R_2}{l}(\frac{L}{R_1} + 2d) \rceil = 1 + \lceil \frac{100\,Kbps}{40 \cdot 8\,bits}(\frac{1{,}500 \cdot 8\,bits}{10\,Mbps} + 2 \cdot 120\,ms) \rceil = 1 + \lceil 75.375 \rceil = 77$.

**Grading guidelines**:

Subtotal of 3 points for expressing the minimum round-trip time $T$ correctly with a formula such as $T = \frac{l}{R_2} + \frac{L}{R_1} + 2d$.

Subtotal of 3 points for realizing that transmission of acknowledgment packets from $B$ to $A$ is a bottleneck for utilizing the path from $A$ to $B$ efficiently.

Subtotal of 2 points for realizing the correct condition "the amount of time to transmit $N$ **acknowledgment** packets should be at least $T$".

Subtotal of 1 point for expressing the above condition correctly with an inequality such as $N\frac{l}{R_2} \geq \frac{l}{R_2} + \frac{L}{R_1} + 2d$. **Note that** the left portion $N\frac{l}{R_2}$ should contain $\frac{l}{R_2}$, not $\frac{L}{R_1}$.

Subtotal of 1 point for transforming the inequality into a correct expression for N, such as $N \geq 1 + \frac{R_2}{l}(\frac{L}{R_1} + 2d)$.

Subtotal of 1 point for expressing $N_{min}$ correctly with an expression such as $N_{min} = 1 + \lceil \frac{R_2}{l}(\frac{L}{R_1} + 2d) \rceil$.

Subtotal of 1 point for calculating $N_{min}$ correctly as 77.

Subtotal of 2 points for realizing that the maximum utilization of the path from $A$ to $B$ after the initial window of N data packets is $U_{max} = \frac{L \cdot R_2}{l \cdot R_1}$.

Subtotal of 1 point for calculating $U_{max}$ correctly as 37.5%.

Problem 3 (20 points). Host $A$ delivers a file to host $B$ after sending 32 data packets with full-size segments via a TCP Reno connection where the maximum segment size is equal to $L$, the threshold is initially set to $6L$, the receive window equals $7L$, the retransmission timeout is set to be 8 times larger than the maximum RTT (including queueing) on the path from $A$ to $B$, and the initial congestion window is $L$. There is no other traffic on the path. Nodal processing delays are negligible. Since the minimum RTT is larger than the time to transmit all 32 data packets over the bottleneck link, the delivery proceeds in rounds that all underutilize the path capacity. Host $B$ responds to each uncorrupted segment by sending an immediate acknowledgment packet. Acknowledgment packets as well as headers of data packets have size $0.1L$. Host $B$ discards out-of-order segments. Host $A$ uses the byte-counting implementation of congestion avoidance (where the congestion window increases exactly by $L$ per transmission round). A router on the path from $A$ to $B$ discards the 6-th, 7-th, 10-th, 11-th, and 13-th data packets. However, $B$ receives the other 27 data packets without corruption or loss. Complete the following table to show when each segment of the file is delivered. Explain your reasoning briefly and report the overall size of the file.

| Transmission round | Maximum congestion window | Transmitted packets (segments carried) | Packets discarded by the router (segments carried) | Segments received in order | Segments received out of order |
|---|---|---|---|---|---|
| 1 | $L$ | 1(1) | – | 1 | – |
| 2 | $2L$ | 2(2), 3(3) | – | 2, 3 | – |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |

**Solution**:

| *Transmission round* | *Maximum congestion window* | *Transmitted packets (segments carried)* | *Packets discarded by the router (segments carried)* | *Segments received in order* | *Segments received out of order* |
|---|---|---|---|---|---|
| *1* | $L$ | 1(1) | – | 1 | – |
| *2* | $2L$ | 2(2), 3(3) | – | 2, 3 | – |
| *3* | $4L$ | 4(4), . . . , 7(7) | 6(6), 7(7) | 4, 5 | – |
| *4* | $6L$ | 8(8), . . . , 11(11) | 10(10), 11(11) | – | 8, 9 |
| *5* | $L$ | 12(6) | – | 6 | |
| *6* | $2L$ | 13(7), 14(8) | 13(7) | – | 8 |
| *7* | $L$ | 15(7) | – | 7 | – |
| *8* | $2L$ | 16(8), 17(9) | – | 8, 9 | – |
| *9* | $3L$ | 18(10), 19(11), 20(12) | – | 10, 11, 12 | – |
| *10* | $4L$ | 21(13) , . . . , 24(16) | – | 13 , . . . , 16 | – |
| *11* | $5L$ | 25(17), . . . , 29(21) | – | 17 , . . . , 21 | – |
| *12* | $6L$ | 30(22), 31(23), 32(24) | – | 22, 23, 24 | – |
| *13* | | | | | |

Problem 3 (continued)

*The retransmission timeout is large enough to avoid unnecessary timeouts. Initially, the congestion window equals L, and the connection is in slow start. The congestion window grows to* $6L$ *during the* 4*-th transmission round upon receiving an acknowledgment for the* 5*-th segment. At that point, the TCP sender stalls because the* 6*-th,* 7*-th,* 10*-th, and* 11*-th data packets are discarded while delivery of the* 8*-th and* 9*-th segments trigger only two duplicate acknowledgments for the* 5*-th segment. Upon timeout, the congestion window is set to L, and the sender retransmits the* 6*-th segment in the* 12*-th data packet. Since the router discards the* 13*-th data packet during the* 6*-th round when the congestion window is* $2L$*, only one duplicate acknowledgment for the lost* 7*-th segment is generated. The second timeout reduces both the congestion window and threshold to* $L$*. Hence, the sender retransmits the* 7*-th segment in the* 15*-th data packet and switches to stay in congestion avoidance until the last* 32*-nd data packet brings the* 24*-th segment of the file. Thus, the file size is* $24L$*.*

**Grading guidelines***:*

*Subtotal of 4 points for showing that the congestion window grows to* $4L$ *during the* 3*-rd round allowing the sender to transmit packets* 4 *through* 7*.*

*Subtotal of 3 points for showing that congestion window grows to* $6L$ *during the* 4*-th round allowing the sender to transmit packets* 8 *through* 11*.*

*Subtotal of 3 point for showing that timeout reduces the congestion window to* $L$ *and causes retransmission of the* 6*-th segment in the* 12*-th data packet during the the* 5*-th round.*

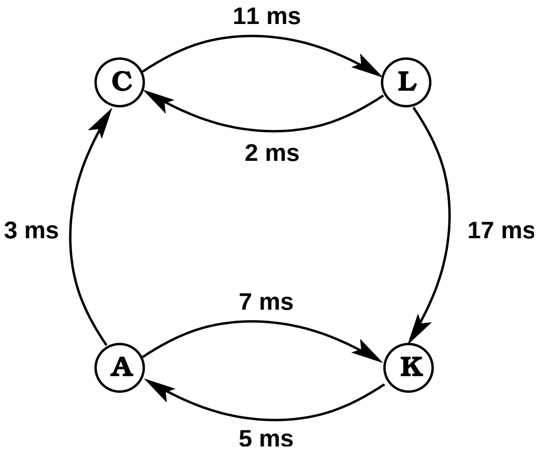
*Subtotal of 3 points for showing that the congestion window grows to* $2L$ *during the the* 6*-th round allowing the sender to transmit packets* 13 *and* 14*.*

*Subtotal of 3 points for showing that timeout reduces the congestion window to* $L$ *and causes retransmission of the* 7*-th segment in the* 15*-th data packet during the the* 7*-th round.*

*Subtotal of 3 points for realizing that the second timeout reduces the threshold to* $L$ *and causes the sender to operate in congestion avoidance during rounds* 8 *through* 12*.*

*Subtotal of 1 point for calculating the file size correctly as* $24L$*.*

Problem 4 (15 points). Consider the following network:



where the cost of a link is equal to the propagation delay of this link, as indicated. Each node runs the distance-vector algorithm. The links deliver distance vectors without corruption or loss. What will be the stabilized distance vector at node $K$? Explain your reasoning.

**Solution**: *The stabilized distance vector at node K will be*

| | **A** | **C** | **L** | **K** |
|---|---|---|---|---|
| **K** | **5 ms** | **8 ms** | $\infty$ | **0** |

*K knows about the direct link to A with cost 5 ms. Since A notifies K that A knows how to get to C at cost 3 ms, K learns about the path through A to C with cost 8 ms. Since C does not notify A that C knows how to get to L, K does not learn about the path through A and C to L, and K's estimate of its distance to L remains* $\infty$.

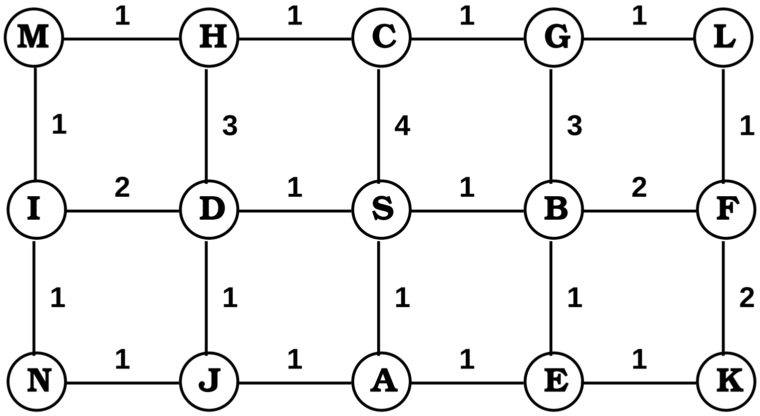
**Grading guidelines**:

5 points for distance to $A$ (4 points for answer 5 ms, and 1 point for a correct explanation)

5 points for distance to $C$ (4 points for answer 8 ms, and 1 point for a correct explanation)

5 points for distance to $L$ (4 points for answer $\infty$, and 1 point for a correct explanation)

Problem 5 (20 points). Consider the following network



that uses Dijkstra's algorithm for unicast routing and center-based trees for multicast routing. Node $L$ is the center (also known as core, rendezvous point, or root) and the only sender of a multicast group. Initially, the group has no receivers. Then, node $H$ joins the multicast group. Later, node $D$ joins the group. Show multicast trees for the group after: (a) node $H$ joins the group, and (b) node $D$ joins the group. Explain your reasoning.

**Solution**: *Dijkstra's algorithm discovers that the cheapest path from* $H$ *to* $L$ *is* $H-C-G-L$*, and the cheapest path from* $D$ *to* $L$ *is* $D-S-B-F-L$*. Nodes* $H$ *and* $D$ *graft these branches by unicasting join messages along the respective paths. Hence, (a) after node* $H$ *joins the group, the multicast tree consists of* $H-C-G-L$*, and (b) after node* $D$ *joins the group, the multicast tree consists of* $H-C-G-L-F-B-S-D$*.*

**Grading guidelines**:

6 points for indicating that each of the nodes joins the group by grafting a branch to $L$ along the cheapest path computed by Dijkstra's algorithm.

6 points for determining that the cheapest path from $H$ to $L$ is $H-C-G-L$.

6 points for determining that the cheapest path from $D$ to $L$ is $D-S-B-F-L$.

1 point for showing that after node $H$ joins the group, the multicast tree consists of $H-C-G-L$.

1 point for showing that after node $D$ joins the group, the multicast tree consists of $H-C-G-L-F-B-S-D$.

Problem 6 (20 points). A router has four input links ($I_0$, $I_1$, $I_2$, and $I_3$) and four output links ($O_0$, $O_1$, $O_2$, and $O_3$). Each of the input and output ports has a buffer for **two** packets and uses first-come first-served (FCFS) packet scheduling. A crossbar connects the input and output ports.

The router operates in rounds. The speed of each input or output link is **one** packet per round. The speed of each crossbar bus is equal to **two** packets per round (i.e., the speedup is 2). When a packet starts arriving from an input link, the corresponding input port checks its buffer occupancy: if the buffer is full, the input port discards the packet; otherwise, the input port stores the arriving packet in the buffer. At the beginning of every round, the router: (a) configures the crossbar to transfer packets that have arrived to the input ports during previous rounds, and (b) frees the input buffer space occupied by packets that were transferred over the crossbar during the previous round. The router configures the crossbar using **round-robin priorities**: during round $t$, input port $I_k$ has priority $(k+t)$ mod 4 where priority 0 is the lowest, and priority 3 is the highest. For example, during round 0, ports $I_0$, $I_1$, $I_2$, and $I_3$ have priorities 0, 1, 2, and 3 respectively; their priorities become respectively 1, 2, 3, and 0 during round 1. If two input ports schedule packets for transfer to the same output port, the crossbar transfers first the packet from the input port with a higher priority. When a packet starts arriving from the crossbar, the corresponding output port checks its buffer occupancy: if the buffer is full, the output port discards the packet; otherwise, the output port stores the arriving packet in the buffer. At the beginning of every round, each output port: (a) starts transmitting a packet to its output link (unless the output port buffer is empty), and (b) frees the buffer space occupied by a packet that was transmitted to the link during the previous round.

The router contains no packets originally. By the end of each round 0 through 3, every input link delivers one packet to the router (i.e., sixteen packets overall). Let $p_k^t$ be the packet that input link $I_k$ delivers to the router by the end of round $t$. Then, the destinations of the packets are as follows:

| Packets | Output Link |
|---|---|
| $p_1^0$, $p_2^0$, $p_0^1$, $p_1^1$, $p_3^1$, $p_1^2$, $p_2^2$, $p_3^2$, $p_1^3$, $p_3^3$ | $O_1$ |
| $p_0^0$, $p_3^0$, $p_2^1$, $p_0^2$, $p_0^3$, $p_2^3$ | $O_2$ |

When are the packets transmitted to their output links or discarded? Trace each of the packets by completing the table below (**it might be wise to start doing this with a pencil**):

| Round | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arrived to $I_0$ | $p_0^0$ | $p_0^1$ | $p_0^2$ | $p_0^3$ | | | | | | | | | | |
| Arrived to $I_1$ | $p_1^0$ | $p_1^1$ | $p_1^2$ | $p_1^3$ | | | | | | | | | | |
| Arrived to $I_2$ | $p_2^0$ | $p_2^1$ | $p_2^2$ | $p_2^3$ | | | | | | | | | | |
| Arrived to $I_3$ | $p_3^0$ | $p_3^1$ | $p_3^2$ | $p_3^3$ | | | | | | | | | | |
| $I_0$ buffer | | | | | | | | | | | | | | |
| $I_1$ buffer | | | | | | | | | | | | | | |
| $I_2$ buffer | | | | | | | | | | | | | | |
| $I_3$ buffer | | | | | | | | | | | | | | |
| Crossbar | | | | | | | | | | | | | | |
| $O_1$ buffer | | | | | | | | | | | | | | |
| $O_2$ buffer | | | | | | | | | | | | | | |
| Transmitted | | | | | | | | | | | | | | |
| Discarded | | | | | | | | | | | | | | |

Problem 6 (continued)

**Solution**:

| Round | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arrived to $I_0$ | $p_0^0$ | $p_0^1$ | $p_0^2$ | $p_0^3$ | | | | | | | | | | |
| Arrived to $I_1$ | $p_1^0$ | $p_1^1$ | $p_1^2$ | $p_1^3$ | | | | | | | | | | |
| Arrived to $I_2$ | $p_2^0$ | $p_2^1$ | $p_2^2$ | $p_2^3$ | | | | | | | | | | |
| Arrived to $I_3$ | $p_3^0$ | $p_3^1$ | $p_3^2$ | $p_3^3$ | | | | | | | | | | |
| $I_0$ buffer | $p_0^0$ | $p_0^1$ $p_0^0$ | $p_0^2$ $p_0^1$ | $p_0^3$ $p_0^2$ | $p_0^3$ | | | | | | | | | |
| $I_1$ buffer | $p_1^0$ | $p_1^1$ $p_1^0$ | $p_1^2$ $p_1^1$ | $p_1^3$ $p_1^2$ | $p_1^3$ $p_1^2$ | $p_1^3$ | | | | | | | | |
| $I_2$ buffer | $p_2^0$ | $p_2^1$ $p_2^0$ | $p_2^2$ $p_2^1$ | $p_2^3$ $p_2^2$ | $p_2^3$ $p_2^2$ | | | | | | | | | |
| $I_3$ buffer | $p_3^0$ | $p_3^1$ $p_3^0$ | $p_3^2$ $p_3^1$ | $p_3^2$ $p_3^1$ | | | | | | | | | | |
| Crossbar | | $p_3^0$ $p_1^0$ $p_0^0$ $p_2^0$ | $p_0^1$ $p_2^1$ $p_1^1$ | $p_3^2$ $p_3^1$ $p_0^2$ | $p_1^2$ $p_2^3$ $p_0^3$ $p_2^2$ | $p_1^3$ | | | | | | | | |
| $O_1$ buffer | | $p_1^0$ $p_2^0$ | $p_1^0$ $p_2^0$ | $p_3^1$ $p_1^0$ | $p_2^2$ $p_3^1$ | $p_1^3$ $p_2^2$ | $p_1^3$ | | | | | | | |
| $O_2$ buffer | | $p_3^0$ $p_0^0$ | $p_3^0$ $p_0^0$ | $p_0^2$ $p_3^0$ | $p_0^3$ $p_0^2$ | $p_0^3$ | | | | | | | | |
| Transmitted | | | $p_0^0$ $p_2^0$ | $p_3^0$ $p_1^0$ | $p_0^2$ $p_3^1$ | $p_0^3$ $p_2^2$ | $p_1^3$ | | | | | | | |
| Discarded | | | $p_0^1$ $p_2^1$ $p_1^1$ | $p_3^2$ $p_3^3$ | $p_1^2$ $p_2^3$ | | | | | | | | | |

**Grading guidelines**:

4 points for identifying correctly the priorities of input ports $I_0$, $I_1$, $I_2$, and $I_3$:

- 1, 2, 3, and 0 during round 1
- 2, 3, 0, and 1 during round 2
- 3, 0, 1, and 2 during round 3
- 0, 1, 2, and 3 during round 4

16 points (1 point for each packet) for identifying correctly what (discarded or transmitted) and when (during which round) happens to each of the 16 packets