CS3251: Computer Networking I, Spring 2017

Homework-3

March 30, 2017

DUE DATE: Friday April 7, 5pm
Maximum number of points: 120

Note-1: Your homework solutions should be electronically formatted as a single PDF document that you will upload on T-Square. If you have to include some handwritten parts, please make sure that they are very clearly written and that you include them as high resolution images.
Note-2: Please show your work in some detail. We cannot give you full credit if your answer is not sufficiently explained.
Note-3: After a homework is graded and the solutions are released, you have at most one week to ask follow-up questions to the TAs.

1 Problem-1 (20 points)

In this exercise you will use Wireshark and "traceroute". First, please learn how the traceroute utility works by reading the corresponding Wikipedia article or another resource. Run traceroute a few times from a Unix host to become familiar with its output.

Then, choose three Internet destinations that you can traceroute. Make sure that you receive a response from every router along the path (in other words, the traceroute output should be complete, giving you the entire route to that destination). Perform at least 10 different traceroutes to each of the three destinations, separated by about 8-12 hours (note that you will need to collect these measurements over 3-5 days – do not start late). Then, analyze the collected traceroutes to answer the following questions:

1. Do you see any route changes in the three paths? If so, identify the route segments that change once or more during your measurements.

2. Select one route to each of the three destinations. Can you identify the ISPs (or more generally, the Autonomous Systems) that each route traverses?

3. Can you identify the geographical location of each router along the path? This is sometimes possible based on the name of the router interfaces. You can also use the free IP geolocation service www.maxmind.com.

2 Problem-2 (20 points)

Consider a network with three routers R1, R2, and R3. Each router is connected with the two other routers. Router R1 is also connected to a subnet A, R2 is also connected to a subnet B, and R3 is also connected to a subnet C.

1. How many subnets exist in this network? (Hint: the correct answer is NOT three)
2. Assign network addresses to each subnet. All addresses must be allocated from the 214.97.254/23 address block. Subnet A should have enough addreses to support 250 interfaces. Subnet B should have enough addresses to support 120 interfaces. And Subnet C should have enough addresses to support 120 addresses. For each subnet, you should show the assigned addresses as a prefix (e.g., a.b.c.d/x). Keep in mind though that a certain subnet may use a prefix that is contained in another subnet's prefix.
3. Based on your previous answer, show the forwarding tables for each of the three routers.

3 Problem-3 (20 points)

Consider a network with four routers: w, x, y, z. The network includes five links with the following costs: c(x,y)=4, c(x,z)=50, c(y,w)=1, c(z,w)=1, and c(y,z)=3. Suppose that the network routers use distance-vector routing. Further, the poisoned reverse technique is used to avoid the "count-to-infinity" problem.

1. Suppose that the routing algorithm has converged. What is the distance vector that each router advertises to every other router?
2. Suppose that the link cost between x and y suddenly increases to 60. Will there be a "slow convergence" (or "count-to-infinity") problem? How many iterations will be required for the routing protocol to converge?

4 Problem-4 (20 points)

Consider the network of Fig.5.13 in the 7th edition of the textbook (Fig.4.42 in the 6th edition).

1. Consider the BGP path information that reaches the stub network W. What is W's view of the network topology? In other words, which BGP path would W use to reach X or Y? Repeat the same task for stub X.
2. Suppose there is one more stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, while A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V to come from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

5 Problem-5 (20 points)

Suppose that you want to design a router with 8 full-duplex ports. Each port should have a capacity of 10Gbps. You can assume that all packets are 64 bytes long.

You have two design options:

1. To use a shared bus with 128 parallel bit lines. The clock speed of the bus is 1GHz. This means that you can transfer 128 bits per nanosecond from one input to one output. After a packet is transferred to the right output, there is a 6 nanosecond delay due to bus arbitration overhead before you can transfer the next packet.

2. To use a 32×32 crossbar switch that can transfer in parallel 64 bits from every input as long as each input is connected to a different output. The clock speed of the switch is also 1GHz. The delay to reconfigure the switch (i.e., to connect its inputs to a different set of outputs) is 3 nanoseconds.

Please answer the following questions, explaining your answers:

1. In the shared bus option, what is the maximum aggregate throughput that your router can achieve?

2. In the crossbar option, how would you connect the eight input interfaces to the 32 switch inputs (and similarly for the outputs)? What is the maximum aggregate throughput that your router can achieve?

3. Which design option would you prefer if you want to ensure that your router can support the full capacity (i.e., 10Gbps) of each of its 8 ports?

6 Problem-6 (20 points)

Deficit Round Robin (DRR) is a packet scheduling algorithm that operates as follows. The DRR scheduler scans all N buffers in a specified round-robin sequence. A "deficit counter" $c(i)$ is maintained for each buffer i ($i = 1 \dots N$). When a non-empty buffer i is scanned, its deficit counter $c(i)$ is incremented by a "quantum value" δ (i.e., $c(i) = c(i) + \delta$). The value of the deficit counter is the maximum amount of bytes that can be sent from that buffer at that point in time: if the deficit counter is greater or equal than the size $L(i)$ of the packet at the head of buffer i, this packet can be transmitted and the value of the counter is decremented by the packet size (i.e., $c(i) = c(i) - L(i)$). When that transmission is complete, the size of the next packet at the head of buffer i is compared to the counter value $c(i)$, and so on. Once the buffer i is empty, or the value of the deficit counter is insufficient (i.e., $c(i) < L(i)$), the scheduler is ready to scan the next buffer. If a buffer is empty, its deficit counter value is reset to 0.

1. Suppose that a router interface has four buffers and that each buffer contains one packet. The length of the packets are 16, 10, 12, and 8 bytes, respectively. The quantum δ is 4 bytes. Assume that all counters are equal to 0 initially. The link transmits 1 byte per nanosecond. Show the deficit counter at each buffer as a function of time and indicate when each of the four packets will be transmitted. You can assume that there are no other packet arrivals.

2. Suppose now that $N = 2$ and that each buffer is never empty. Suppose that the packets of the first buffer are four times larger than the packets in the second buffer. How does the DRR scheduler share the transmission capacity among the two buffers? You can assume that the smallest packet size is an integer multiple of δ.

3. Suppose that N=4 and that the transmission capacity is 10Mbps. You want to provide at least 4Mbps to buffer-1, 2Mbps to buffer-2, 1Mbps to buffer-3 and 1Mbps buffer-4. How would you modify the previous DRR algorithm to make it "weighted" so that you can share the transmission capacity in this unequal manner?