College of Computing
Georgia Institute of Technology

# CS 7260: Internetworking Protocols and Architectures: Spring 2007

# Quiz I

There are 16 questions and 12 pages in this quiz booklet (including this page). Answer each question according to the instructions given. You have **85 minutes** to answer the questions.

If you find a question ambiguous, write down any assumptions you make. **Be neat and legible.** If I can't understand your answer, I can't give you credit! There are three pretty challenging questions (clearly marked); you may want to look through the whole quiz and save those for last.

Use the empty sides of this booklet if you need scratch space. You may also use them for answers, although you shouldn't need to. *If you do use the blank sides for answers, make sure to clearly say so!*

**Note well: Write your name in the space below AND your initials at the bottom of each page of this booklet.**

**THIS IS AN "OPEN NOTES, OPEN PAPERS" QUIZ.**
**NO ENCRYPTED WIRELESS TRAFFIC.**
**MAKE SURE YOU'VE READ ALL THE INSTRUCTIONS ABOVE!**

Initial here to indicate that (1) you've read the instructions and (2) you agree to abide by the Georgia Tech Honor Code:

The last page has easy bonus questions, which you can answer outside of the allotted time. Rip the last page off of your quiz for five bonus points. Turn it in anonymously if you like. You won't get the five points if you don't tear off the page (this is to make certain you've read this far ;).

*Do not write in the boxes below*

| 1-6 (xx/20) | 5-10 (xx/13) | 11-13 (xx/18) | 14-17 (xx/14) | Bonus (xx/5) | Total (xx/70) |
|---|---|---|---|---|---|
| | | | | | |

**Name:**

# I Warmup

**1. [4 points]:** From Lecture 6, we learned that multihoming is a widely practiced technique for increasing resilience to faults. Which of the following are true about a stub AS that uses IP-based multihoming?

**(Circle ALL that apply)**

**A.** If the stub AS connects to a single upstream AS, it needs a public AS number.

**B.** If the stub AS connects to multiple upstream ASes, it needs to run BGP.

**C.** If the stub AS connects to multiple upstream ASes, it needs its own address space that is not a subset of either provider's address space.

**D.** Due to BGP routing policies of its upstream ISPs and other ASes, the stub AS cannot control which of its upstream links are used for incoming traffic at all.

**Answer 1** The answer is: (B). As discussed in Lecture 6, a stub AS can connect to a single AS and use BGP with a private AS number, and anytime it has multiple upstream ASes, it must run an interdomain routing protocol (the one in use today is BGP). A stub AS can connect to multiple upstream ASes using a subset of address space from one of its providers. Stub ASes can control inbound traffic to some degree by splitting prefixes and with AS path prepending. ■

**2. [4 points]:** Which of the following network elements violate the principle of "fate sharing"?

**(Circle ALL that apply)**

**A.** Stateful firewalls

**B.** DNS servers

**C.** "NAT" (network address translation) boxes

**D.** A load balancer that selects paths based on the (source IP, destination IP) tuple

**E.** None of the above

**Answer 2** The answer is (A) and (C); when either of these network elements fail, the state associated with the connections that pass through these elements is destroyed, which will cause the end-to-end sessions to fail. This concept was covered in Lecture 1. ■

**3. [4 points]:** Suppose that a link is being monitored with both packet flow monitoring and a full packet trace.

Assume that routes do not change, the interface and packet filters do not drop any packets, that the packet trace contains full payloads, and that the flow records are based on *all* packets that traverse the link (*i.e.*, that there is no sampling).

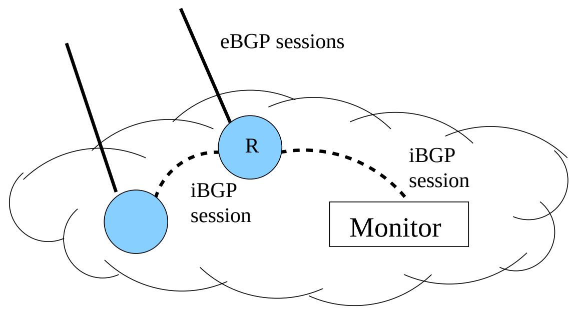
Which of the follow statements are true about each of the traces?

**(Circle ALL that apply)**

- **A.** The packet trace can be used to calculate the exact duration of every flow that crosses the link.
- **B.** The trace of flow records can always be used to calculate the exact duration of every flow that crosses the link.
- **C.** Both the packet trace and the trace of flow records can be used to determine the number of bytes in each flow.
- **D.** The size of a flow record will always be smaller than the combined size of the packets in the corresponding flow.

**Answer 3** The answer is (A) and (C). (B) is false because a single flow may be split across multiple flow records, so it may be tough to distinguish when a flow begins and ends if it is idle for a long time. (D) is false; recall the example in class where we discussed that a single flow could consist of a SYN packet, while a typical flow record is 1500 bytes. ■

Suppose that you have a BGP route monitor connected as shown below. You collect traces of routing updates and tables at a BGP monitor.




**4. [2 points]:** What does a BGP routing table dump at the BGP monitor reflect? **(Circle the BEST answer)**

**A.** All routes that $R$ learns.

**B.** All routes that $R$ learns from its iBGP neighbors.

**C.** All routes that $R$ selects as its "best" routes.

**D.** The routes for which $R$ selects an iBGP-learned route as its best route.

**E.** None of the above.

**Answer 4** The answer is (E), none of the above. Think of the monitor as a regular router, in this case, if $R$ selects an iBGP-learned route, rather than an eBGP-learned route, it will not hear *any* route for that prefix (this is the same as a path visibility violation). ■

**5. [2 points]:** What do BGP routing updates logged at the BGP monitor reflect? **(Circle ALL that apply)**

**A.** All instances of $R$ changing its choice of best route.

**B.** All instances of $R$ changing from an eBGP-learned route to an iBGP-learned route.

**C.** All instances of $R$ changing from an iBGP-learned route to an eBGP-learned route.

**D.** All cases where $R$ changes from any route to *no* route (i.e., the route is globally withdrawn).

**E.** None of the above.

**Answer 5** The answer is (B) and (C), for the same reasons as mentioned in the previous question. ■

**6. [4 points]:** You are performing one-way latency measurements from some node $A$ to another node $B$. While postprocessing the data, you notice that the one-way latency decreases at time $t$ in the logs. Which of the following phenomena could explain your observation?

**(Circle ALL that apply)**

**A.** The system clock on node $A$ jumped backwards at time $t$.

**B.** The system clock on node $B$ jumped backwards at time $t$.

**C.** The clock on node $A$ jumped forward at time $t$.

**D.** The clock on node $B$ jumped forward at time $t$.

**E.** The network-layer path between nodes $A$ and $B$ changed at time $t$.

**Answer 6** The answer is (B) (C), and (E). The scenarios in (B) and (C) make $t_B - t_A$ smaller. ■

# II Potpourri

**7. [3 points]:** Recall the collection setup for DNS and TCP packet traces in the Jung *et al.* paper *DNS Performance and the Effectiveness of Caching*: the packet traces were captured *between the local resolver and the Internet* (as opposed to between the local resolver and the hosts performing lookups). Explain why this experimental setup may may result in *many* TCP connections without any DNS lookups.

**(Answer legibly in the space below.)**

**Answer 7** Many TCP connections may exist without corresponding DNS lookups because the local DNS resolver may *cache* the lookups issued by the end hosts, and these lookups will never traverse the link being traced. ■

**8. [3 points]:** Recall the "one-hop down" packet deflection rule from the Yang *et al.* paper: Packets can be deflected to a node off the shortest path if the node to which the traffic is deflected is closer to the destination than the default next-hop node.

Briefly explain why packet deflections according to the "one hop down" rule in the Yang *et al.* paper is guaranteed to never create a forwarding loop.

**(Answer legibly in the space below.)**

**Answer 8** Every hop along the path *reduces the overall distance to the destination*. By definition, then, forwarding loops are impossible (this would require increasing the distance to the destination at some point along the path). ■

**9. [2 points]:** Suppose that the path between two endpoints, $P(t)$, oscillates in a way that is defined as a function of time $t$, as follows:

$$\mathrm{P}(t) = \begin{cases} P_1, & \text{if} \quad (\lfloor t \rfloor \mod 4) = 0 \\ P_2, & \text{otherwise} \end{cases}$$

Which (if either) path is more *prevalent*? Which path (if either) is more *persistent*?

**(Answer legibly in the space below.)**

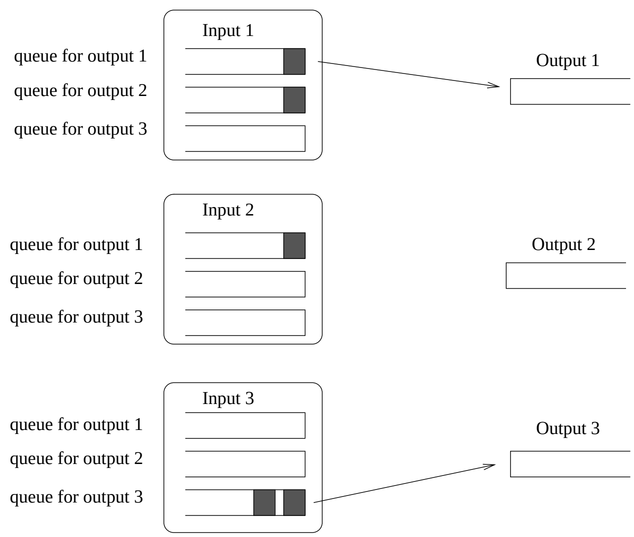
**Answer 9** $P_2$ is both more prevalent and more persistent. It is both more likely to be the path at any instance in time (prevalence) and, when it is selected, it is less likely to switch to an alternate path (persistence). ■

**Initials:**

# III Routing and Switching

**10. [6 points]:** Consider a router with $3 \times 3$ crossbar switch with virtual output queues: each input interface has 3 input queues and each output interface has one output queue. The figure shows three queues for each input, one corresponding to each output.

Suppose the queued packets shown arrived in one timestep.




Suppose that all interfaces have the same throughput.

**a.** (2 points) Briefly explain why the matching above shown above is not maximal.

**b.** (2 points) Sis Cohacker claims that, if the crossbar switch has a speedup of 2, the above matching need not degrade throughput for this case. Take the initial matching above and combine it with a second matching in a way that pushes all queued packets through the crossbar in two matchings.

**c.** (2 points) Explain what "speedup" means and why crossbar speedup helps router throughput.

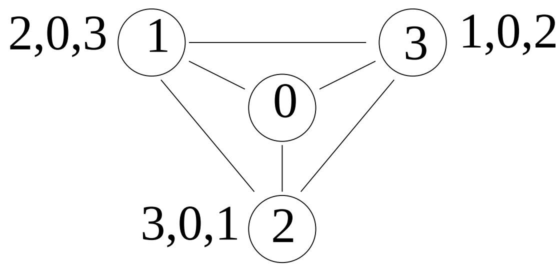
**(Answer legibly in the space below.)**

**Answer 10** (a) The above matching scheme is not maximal because the first input has packets queued for the second output, and the second input has packets queued for the first output. Matching the first input to the second output and the second input to the first output would be maximal.
(b) The second such mapping could be the matching from part (a), for example. (c) If the router crossbar has a speedup of 2, then the crossbar can push packets from input to output at twice the rate at which they arrive at the input queues. This allows the opportunity for *two* matchings to push the available packets across the crossbar. ■

**Initials:**

**11. [8 points]:** Consider the following topology, where each node in the graph denotes an AS. Each node's preferences for routing order are written next to the node in order of most preferred to least preferred.

Each node selects preferences only based on the next-hop AS: each node prefers to route indirectly through the node in counter-clockwise direction, *regardless of the length of the path*. Each node's second most preferred path is the direct path, and its least preferred paths are those through the node in the clockwise direction.



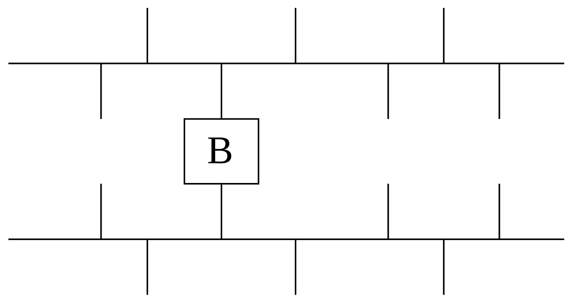

**a.** (3 points) Provide one stable solution for this routing system (there are at least three). In other words, write down a path assignment for which no node can switch to a more preferred path (*e.g.*, Write something of the form "Node 1 selects path '1 2 0', Node 2 selects '2 0', etc.).

**b.** (5 points) **Very Hard!** Describe an initial state and an activation sequence where the routing system will oscillate (there are at least three solutions to this problem, too).

**Answer 11** (a) There are three stable solutions to this routing system: path assignments '1 3 2 0', '3 2 1 0', and '1 3 2 0' are all stable path assignments because no AS can switch to a path that is both (i) more preferred than its current path and (ii) consistent with the path selections currently being made by other nodes.
(b) Here is one solution: Begin with initial path assignment (10, 20, 320). Activate 2, which then switches to what it thinks is still '2 1 0' . Activate 1, which then switches to using '1 3 2 0'. Activate 3, which switches to using '3 2 1 0'. Active 2, which must now switch to using the direct path (its previous path through 1 now includes itself, so it must switch to the direct path). Similarly, activate 1, which must also switch to the direct path. Activating 3 brings us back to 3 selecting '3 2 0'. ■

**Initials:**

**12. [4 points]:** Augment the layer 2 topology shown below ($B$ denotes a bridge between two LANs). Show that, without a spanning tree algorithm and significant latency between the two LANs, ethernet frames would *double* at every time step. Briefly explain why this exponential increase occurs.

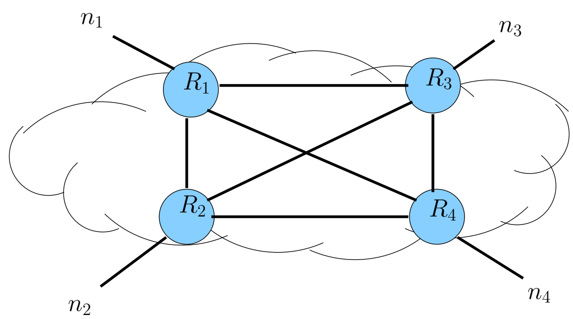



**Answer 12** Each packet that is advertised on one output port arrives on two other input ports (and is subsequently re-broadcast), resulting in a per-packet doubling. ■

**Initials:**

# IV Traffic Monitoring and Engineering

Consider a network shown below. All IGP links in the graph have unit cost (in both directions). Assume that the iBGP is "fully meshed" (*i.e.*, each router $R_i$ has an iBGP session with every other router).




The network has the following traffic matrix, for *external* sources and destinations, $n_1$, $n_2$, ... (*i.e.*, the sources and destination IP addresses refer to global source and destination IP addresses, which may not be routers in the local network).

Suppose that external sources only send traffic to the other external nodes (*i.e.*, not to routers). The traffic matrix below reflects the *actual* number of bytes destined between each source-destination pair of external nodes, *in thousands of bytes*.

$$\begin{bmatrix} 0 & 6667 & 2241 & 8665 \\ 7517 & 0 & 2753 & 3338 \\ 1164 & 6245 & 0 & 4926 \\ 710 & 7205 & 3491 & 0 \end{bmatrix}$$

For example, the matrix indicates that there were $6,667,000$ bytes flowing *from* node $n_1$ *to* node $n_2$.

**13. [2 points]:** Suppose that the path between $R_1$ and $R_4$ is dominated by HTTP traffic. Which of the end nodes—$n_1$ or $n_4$—is more likely a Web server, and why?

**Answer 13** $n_4$ is more likely a Web server. HTTP responses (Web pages) are larger than the requests for those pages, so more bytes will flow from the server to the client than vice versa. ■

**Initials:**

Suppose now that all routers $R_i$ have flow monitoring enabled, and that they maintain flow statistics by sampling each packet uniformly at random with $p = 1/1000$.

**14. [2 points]:** Suppose that the traffic from $n_2$ to $n_1$ consisted of 1,000-byte packets from one source port to one destination port (*i.e.*, that all traffic on this path can be considered as one "flow"). Also assume that the traffic in the matrix above is not divided across flow records.

What is the *expected value* for the number of bytes in a flow record for traffic with source IP $n_2$ and destination IP $n_1$?

**Answer 14** $7,000$. To compute the expected number of packets, use the standard Bernoulli equation: $np = 7000 \cdot 0.001 = 7$. $1,000$ bytes per packet yields $7,000$ bytes. ■

**15. [5 points]: Hard!** Anita Perator has decided that the links from $R_3$ to $R_1$, $R_4$ to $R_1$, and $R_2$ to $R_3$ are too lightly loaded. Increase *two* unidirectional IGP link weights to divert traffic from $n_2$ along these two lightly loaded links. (*Hint:* My solution to this took advantage of equal-cost multipath.)

**Answer 15** Increasing the $R_1 - R_2$ link weight to 5, and the $R_1 - R_3$ link weight to 2 creates *two* equal cost shortest paths between $R_1$ and $R_2$: ($R_2$, $R_3$, $R_1$) and ($R_2$, $R_3$, $R_4$, $R_1$). This will increase the traffic on the two under-loaded links. ■

**16. [5 points]: Harder!** Suppose that the flow from $R_2$ to $R_1$ comprised packets that were all 1000 bytes (*i.e.*, the flow had $7,517$ packets). What is the probability that the number of bytes in the flow record for traffic from $R_2$ to $R_2$ will be $6,000$ or less? **Your answer doesn't need to be an exact number. It is fine to express your answer as an mathematical expression (*i.e.*, sums of probabilities).**

**Answer 16** Given equal-sized packets, the question is essentially asking the probability of sampling $6,000$ packets or less, out of a total flow size of $N = 7.517 \times 10^3$. Use the binomial distribution to compute the probability of sampling exactly $k$ packets from this flow:

$$P(k) = \binom{N}{k}(0.001)^k(0.999)^{N-k}$$

The probability that $k \leq 6000$ is simply:

$$\sum_{k=0}^{6000} P(k)$$

■

**Initials:**

## V Bonus: Anonymous Course Feedback

**This page is anonymous.** Rip this off from your exam, and turn it in separately if you like. You'll get five points for simply ripping off the last page of the exam, but I'd prefer if you fill it out and hand it in in a separate stack.

What are the things you like most about the course so far?

What are the things you like least about the course so far?

Should we have more "coffee talks"? What should the topic be? What else is missing (topics in syllabus you'd like to see, things you'd like to learn more about, etc.)?

**Initials:**