CS 168 Introduction to the Internet
Fall 2017 Scott Shenker

Discussion 8 NAT & Midterm Review

1 NAT

(1) **True/False**

i. NAT allows multiple hosts to share the same IP address in a way that conforms to the E2E principle.
Solution: False. NAT is an example of middleboxes, which add additional functionalities besides packet forwarding to the network.

ii. Port numbers are used to multiplex among hosts behind the same NAT.
Solution: True.

(2) **Short Answer**
Consider a host A behind a NAT, trying to communicate with a remote host B. When a packet headed for B leaves host A, its source port is 1, destination port is 2, source IP is 1.2.3.4, destination IP is 4.3.2.1. When a packet headed for A leaves host B, its source port is 2, destination port is 3, source IP is 4.3.2.1, destination IP is 5.6.7.8. Based on the information above, answer the following questions:

i. The packet from host A to host B arrives at the NAT. What are the fields in the packet after the NAT has altered it?

Source IP: 5.6.7.8

Source Port: 3

Destination IP: 4.3.2.1

Destination Port: 2

ii. The packet from host B to host A arrives at the NAT. What are the fields in the packet after the NAT has altered it?

Source IP: 4.3.2.1

Source Port: 2

Destination IP: 1.2.3.4

Destination Port: 1

2 Midterm Potpourri

(1) **Short Answer**

(a) In Link State routing, routers always know the path along which their packets will travel.
Solution: False - routers have a good guess at what the packet's path will be, but, as in most distributed algorithms, there is no guarantee of a synchronized worldview.

(b) Consider a routing protocol that always computes valid paths, where the forwarding decision at router X can depend on (i) the destination address and (ii) the interface through which the packet arrived at router X. Assume that the routing algorithm has converged, and consider the routing entries that take a packet from Host A to Host B. Which of these statements are true? (Mark all that apply.)
Solution: The path can visit the same router twice. This is because it is *not* considered a loop if a packet enters the same router coming from a different source. A different routing decision may be made for that (source, destination) tuple.

(c) A router splits a 3240 byte packet (which uses no IP options) into four fragments before sending it out over a link that has an MTU of 1000 bytes. There is no error in the fragmentation process, but the implementation chooses to break up the packet into fragments with different sizes. We will tell you the fragment sizes, but not the order in which they are sent. The fragment sizes are, from smallest to largest: 620, 800, 900, and 980 bytes. In the following, you can decide if this is enough information to answer certain question. Which of these fragments (identified by their size), does not have the MF flag set (i.e., which is the fragment containing the last byte of the original packet)?
Solution: Note first that no fragmentation choice is specified (we don't necessarily greedily stuff into the first n packets). It's pretty unclear what to do then - any decision we make would be arbitrary. But, remember that fragmentation actually requires that byte sizes are divisible by 8. If that is not the case, then that packet must be sent last (so that we do not need to specify an offset for it). (900-20) mod 8 = 0, (620-20) mod 8 = 0, (980-20) mod 8 = 0, but (800 - 20 mod 8 != 0), so therefore, the 800 packet must be sent last.

(2) **Reliability** Consider a design that uses cumulative ACKs (where an ACK of the fifth packet means all five lowest packets have been received, but not the sixth). Whenever a packet is sent, a timer is set for it, and if not ACKed when the timer goes off, the packet is resent. The algorithm starts by sending the last five packets (i.e., the highest numbered), and then sends an additional packet (the highest-numbered previously unsent packet), whenever one of its previously sent packets is ACKed for the first time. Is this reliable?
Solution: Assume the algorithm sends packets 10-15 first. ACK 15 communicates that all packets ¡ 15 have been received, but the sender has not sent packets 0-10, so the sender won't send any new packets and the receiver won't be ACKing any.