CS 176A: Homework #4

<u>Instructions</u>:

1. All parts of the assignment are to be completed independently.
2. Answer each question (not sub-question) on a fresh page. Convert your document into a PDF and upload on Gradescope under HW4. When you do this, identify where each question AND sub-question is (follow the Gradescope prompts on this).

Part 1: Complete the questions on the following pages. (70 points total)

Part 2: Wireshark Lab (30 points total)

Homework 4 Questions

1. Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11010000 00000000 00000000 00000000 through 11010000 11111111 11111111 11111111	0
11010001 00000000 00000000 00000000 through 11010001 00000000 11111111 11111111	1
11010001 10000010 00000000 00000000 through 11010001 10000011 11111111 11111111	2
Otherwise	3

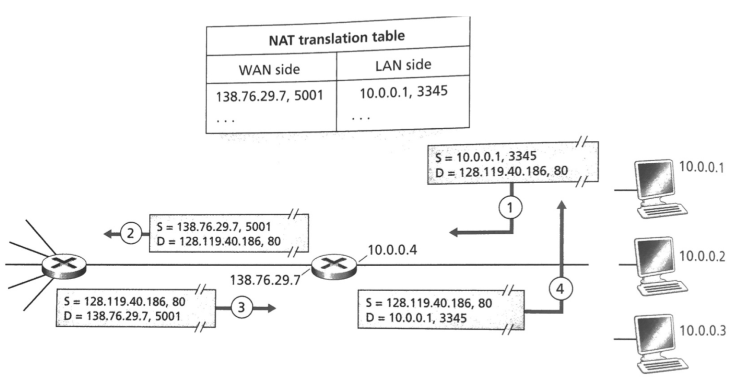
(a) Provide the corresponding forwarding table that has four entries, uses longest-prefix matching, and forwards packets to the correct link interfaces. In other words, what is the prefix that would correspond to each of the address ranges?

(b) What is the correct interface to use for each of the following addresses:

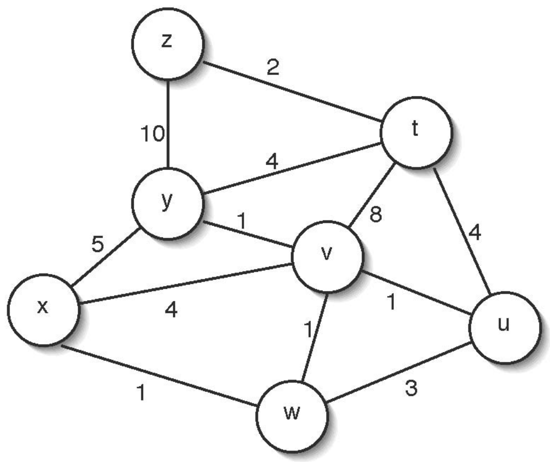
- 10011000 10010001 01010001 01010101
- 11010001 00000000 11000011 00111100
- 11010001 10000000 00010001 01110111

(c) Rewrite the forwarding table you created for part (a) using the a.b.c.d/x notation instead of the binary string notation.

2. Consider the network figure below (Figure 4.25 of your textbook). Suppose that the ISP instead assigns the router the address 128.100.85.28 (the left interface of the center router) and that the network address of the home network is 192.168.110/24 (the network on the right side of the router).
(a) Assign addresses to all interfaces in the home network (the right interface of the router, and the three hosts).
(b) Suppose each host has two ongoing TCP connections, all to port 80 at host 91.84.100.233. Provide the six corresponding entries in the NAT translation table in the router.



3. Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1 in your book.

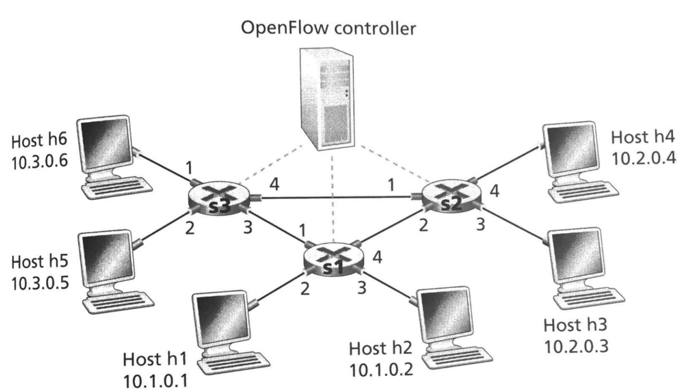


4. Consider the three node topology shown in Figure 5.6 of your book. Rather than having the link costs as shown in the figure, use the link costs of c(x,y)=10, c(y,z)=5, and c(x,z)=3. Compute the distance tables after the initialization step and after each iteration of a synchronous version of the distance-vector algorithm (as shown in the book for this figure).

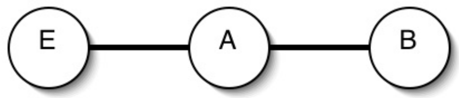
5. Consider the below SDN OpenFlow network. Suppose that the desired behavior for datagrams arriving from hosts h2 or h4 at s2 is as follows:

- any datagrams arriving from host h2 and destined for h5 or h6 should be forwarded to s3
- any datagrams arriving from host h4 and destined for h1, h2, or h5 should be forwarded to s1

Specify the flow table entries in s2 that implement this forwarding behavior.



6. Consider the following simple network, in which A and B exchange distance-vector routing information.



All links have cost 1. Suppose the A-E link fails. Give a sequence of routing table updates that leads to a routing loop between A and B.

Wireshark Lab: NAT

In this lab, we'll investigate the behavior of a NAT router. This lab will be different from our other Wireshark labs, where we've captured a trace file at a single Wireshark measurement point. Because we're interested in capturing packets at *both* the input and output sides of the NAT device, we'll need to capture packets at *two* locations. Also, because many students don't have easy access to a NAT device or to two computers on which to take Wireshark measurements, this isn't a lab that is easily done "live" by a student. So, in this lab, you'll use Wireshark trace files that we've captured for you. This should be a relatively short and easy lab since the concepts behind NAT aren't difficult, but it'll be good nonetheless to observe NAT in action. Before beginning this lab, you'll probably want to review the material on NAT in section 4.3.3 in the text.

NAT Measurement Scenario

In this lab, packets were captured that contain a simple HTTP GET request message from a client inside a home network to a remote server, and the corresponding HTTP response from that server. Within the home network, the home network router provides a NAT service, as discussed in Chapter 4. Figure 1 shows the Wireshark trace-collection scenario. Packets were captured in *two* locations, and thus this lab has *two* trace files:

- Packets being received at the local area network (LAN) side of the NAT router were captured. All devices in this LAN have addresses in 192.168.10/24. This file is named *nat-inside-wireshark-trace1-1.pcapng*.
- Because we're also interested in analyzing packets being forwarded (and received) by the NAT router on its Internet-facing side, a second trace file was captured on the Internet side of the router, as shown in Figure 1. Packets captured by Wireshark at this point that were sent from a host on the right to the server on the left will have undergone NAT translation by the time they reach this second measurement point. This file is named *nat-outside-wireshark-trace1-1.pcapng*.

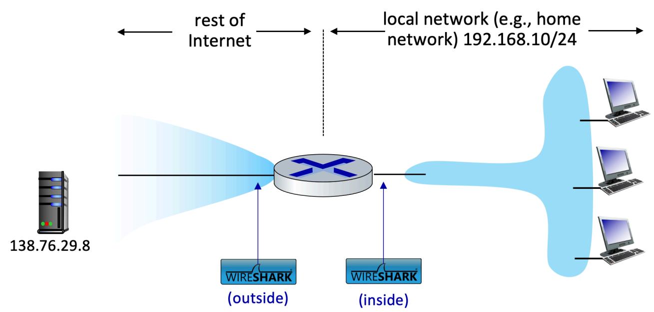


Figure 1: NAT packet capture scenario

In the scenario shown in Figure 1, one of the hosts within the LAN sent an HTTP GET request to the web server at IP address 138.76.29.8, which responded back to the requesting host. Of course, we're not really interested in the HTTP GET request itself, but rather how the NAT router changes the IP addresses and port numbers of the datagram containing the GET request on the LAN side (inside) to addresses and port numbers in the forwarded outgoing datagram on the Internet side (outside) of the NAT router.

Let's first take a look at what's happening on the LAN side of the NAT router. Open the *nat-inside-wireshark-trace1-1.pcapng* trace file. In this file, you should see an HTTP GET request addressed to the external web server at IP address 138.76.29.8, as well as the subsequent HTTP response message ("200 OK"). Both of these messages in the trace file were captured on the LAN side of the router.

Answer the following questions.

1. What is the IP address of the client that sends the HTTP GET request in the *nat-inside-wireshark-trace1-1.pcapng* trace? What is the source port number of the TCP segment in this datagram containing the HTTP GET request? What is the destination IP address of this HTTP GET request? What is the destination port number of the TCP segment in this datagram containing the HTTP GET request?
2. At what time[1] is the corresponding HTTP 200 OK message from the webserver forwarded by the NAT router to the client on the router's LAN side?

In the following we'll focus on these two HTTP messages (GET and 200 OK). Our goal below will be to locate these two HTTP messages in the trace file *nat-outside-wireshark-trace1-1.pcapng*, captured on the Internet-side link between the router and the ISP. Because the captured packets heading towards the server will have already been forwarded through the NAT router, some of the IP address and port numbers will have been changed as a result of NAT translation.

[1] Specify time using the time since the beginning of the trace (rather than absolute, wall-clock time).

Open the trace file *nat-outside-wireshark-trace1-1.pcapng*. Note that the time stamps in this file and the *nat-inside-wireshark-trace1-1.pcapng* file are not necessarily synchronized.

In the *nat-outside-wireshark-trace1-1.pcapng* trace file, find the HTTP GET message that corresponds to the HTTP GET message that was sent from the client to the 138.76.29.8 server at time *t*=0.27362245, where *t*=0.27362245 is the time at which this message was sent, as recorded in the *nat-inside-wireshark-trace1-1.pcapng* trace file.

3. At what time does this HTTP GET message appear in the *nat-outside-wireshark-trace1-1.pcapng* trace file?
4. What are the source and destination IP addresses and TCP source and destination port numbers on the IP datagram carrying this HTTP GET (as recorded in the *nat-outside-wireshark-trace1-1.pcapng* trace file)?
5. Which of these four fields are different than in your answer to question 1 above?
6. Are any fields in the HTTP GET message changed?
7. Which of the following fields in the IP datagram carrying the HTTP GET are changed from the datagram received on the local area network (inside) to the corresponding datagram forwarded on the Internet side (outside) of the NAT router: Version, Header Length, Flags, Checksum?

Let's continue to look at the *nat-outside-wireshark-trace1-1.pcapng* trace file. Find the HTTP reply containing the "200 OK" message that was received in response to the HTTP GET request you just examined in questions 4-8 above.

8. At what time does this message appear in the *nat-outside-wireshark-trace1-1.pcapng* trace file?
9. What are the source and destination IP addresses and TCP source and destination port numbers on the IP datagram carrying this HTTP reply ("200 OK") message (as recorded in the *nat-outside-wireshark-trace1-1.pcapng* trace file)?

Lastly, let's consider what happens when the NAT router receives this datagram that you examined in questions 9 and 10, performs NAT translation, and finally forwards that datagram to the destination host on the LAN side. Based on your answers to questions 1 through 10 above and your knowledge of how NAT works, you should be able to answer the following question without actually looking at the *nat-inside-wireshark-trace1-1.pcapng* trace file:

10. What are the source and destination IP addresses and TCP source and destination port numbers on the IP datagram carrying the HTTP reply ("200 OK") that is forwarded from the router to the destination host in the right of Figure 1?

Just to make sure you understand NAT, you should now use Wireshark to peek into the *nat-inside-wireshark-trace1-1.pcapng* trace file at look at the HTTP reply ("200 OK"). Do your answers to question 11 above match what you see in the *nat-inside-wireshark-trace1-1.pcapng* trace file? [Hopefully, your answer is yes ☺].

That's it!