**Lab #11: Networks**

Due: Tuesday, April 29, 2014, beginning of class

## Instructions

* You will need your WFU-issued ThinkPad for this lab.
* Submit your answers to the assignment in Sakai using this document and any other material you are asked to submit in the exercises.

## Part 0: Background

The purpose of this lab is to reinforce what you have learned about computer networks. Background comes from Chapter 15 and from lectures.

* Create a folder called ***Lab 11*** on your desktop.
* From the assignment on Sakai, right-click and *save link as* the ***Networks\_Lab\_Applet\_Doc*** into your Lab 11 folder
* From the assignment on Sakai, right-click and *save link as* the ***Applets\_for\_Networks\_Lab.zip***  into your Lab 11 folder. Unzip this file.
* Download the lab manual and report.

## Part 1: TCP/IP

* Read the applet documentation then complete the exercises below.
* Note each exercise calls for you to start a particular applet. To do this, go into the appropriate applets sub-folder in your Lab 11 folder. In that sub-folder, double click on the ***applet\_frame.htm*** file. Once the applet frame starts, depending on your security settings, you may need to explicitly allow the applet to run. Note that if the applet does not work with Internet Explorer, right click on applet\_frame.htm and select ‘open with’ and select Firefox.

**Exercise 1.**

Start the “TCP/IP” (reliable connection simulator) applet. In the text field labeled “Your message:”, type the following:

**Computer networking is essential in our world today.**

Then press the button “Send a message.”

* 1. Watch the entire sequence of packets that are sent for the sample message. How many DAT packets were sent? How many ACK packets were sent?

6 DAT; 6 ACK

* 1. Each character or blank in a packet counts for one character. A packet *header* consists of 13 characters and all DAT packets carry 10 characters (if the actual data is less than 10 characters, then blanks are added). ACK packets have no data characters. So, a DAT packet with data “Computer n,” will have a total of 23 characters in it (13 in the header + 10 data). An ACK packet will always have 13 characters. Count up how many characters were sent in total.

6\* (13 + 10) + 6 \* 13 = 216

1.3) There are 52 characters in the Example message, including blanks and punctuation. Subtract 52 from the total number of characters sent in both directions in all packets. (Do not damage or delete packets this time.) Divide this number by the total number of characters to get the overhead, expressed as a percentage.

(216-52)/208 = 72%

* 1. Imagine that you have a million-character message to send, perhaps a large file. How many characters total would be sent in all packets necessary to move it from node 0 to node 1?

1,000,000 /10 = 100,000 DAT packets + 100,000 ACK packets. 100,000 \* (13+10) + 100,000 \* 13 = 3,600,000 characters

* 1. What would be an obvious way to decrease the overhead? Why might this solution backfire? Under what conditions?

Increase the amount of data sent within each packet. Solution might backfire if the same data has to be sent multiple times due to packets being damaged or never arriving.

**Exercise 2.**

Start the “TCP/IP” (reliable connection simulator) applet. In the text field labeled “Your message:”, type the following:

**Computer networking is essential in our world today.**

Then press the button “Send a message.”

2.1) Select “Delete packets that are touched” from the drop-down menu. Delete some data packets by clicking on them as they move along the wire and watch the re-transmission after timeout. What happens if you delete the re-transmitted packet? Does the “TCP/IP” applet need to take any special action?

The sender keeps sending until the sender receives an ACK from the receiver that the packet arrived.

2.2) Now try deleting some ACK packets. (Are any NAK packets sent?) What happens?

The sender keeps sending until the sender receives an ACK from the receiver that the packet arrived.

2.3) Select “Damage packets that are touched” from the drop-down menu. Damage some data packets by clicking on them as they move along the wire and watch the re-transmission after timeout. (Check the margin note in the applets documentation.) What happens if you damage a packet?

The receiver sends a NAK packet, and the sender resends the data packet.

**Exercise 3.**

3.1) See if you can compute the checksum as TCP/IP does. Run the applet, using any message. Select a data packet, but don’t select the last packet because it might be too short—less than 10 characters. Count up the characters, including blanks. If they do not equal 10, assume there are blanks at the end so that the character count is 10.

Using an ASCII chart (you should be able to find one), add up the values for each of the characters. Then take the modulus of this number using 256. The modulus operator is available on some computer calculators and is represented by mod, but you can easily compute it by dividing the total by 256 and saving only the remainder. For instance, suppose the total is 7452. Since 7452 ÷ 256 = 29.109375, but we only want the remainder.

Multiply 0.109375 by 256, which gives us 28. Or: 7452 (29 256) = 28.

Compare your computed checksum against what the applet shows for the packet. Do they agree?

Message = “aaaaaaaaaa” ASCII code for “a” is 97. 97\*10 = 970. 970%256 = 202 which agrees to the program. Note: the checksum only is compute on the data.

3.2) Now damage your packet by altering one character. Re-compute the checksum. Do you see how TCP/IP can spot errors?

Yes. The transmitted checksum does not agree with the checksum compute by the receiver.

3.3) Think of a way that a packet can be damaged and still have the same checksum as the undamaged version. (There are several possibilities. Imagine that two or more bytes are altered at the same time.)

Easy example is if send message “bbbbbbbbbb” checksum = 212. Corrupted message “abcbbbbbbb” has the same checksum.

## Part 2: Network routing

**Exercise 4.**

Start the “Network router” applet. Select Example 3, the ring network. Become familiar with the applet. Move a few nodes around by dragging them.

4.1) Double-click on the node 37.61.25.46. List the nodes it is directly connected to.

159.121.66.98, 138.92.0.5

4.2) If 37.61.25.46 wants to send packets to a node that is not directly connected, to which node will it first send the packets? (Check the routing table by double clicking on the node.)

138.92.0.5

4.3) Run the applet for a while, letting it generate packets continuously. Double-click on 37.61.25.46 again and look at its statistics. How many packets were sent? Received? Forwarded?

Sent = 0, received = 0, forward = 9

4.4) Who else is sending messages, and to whom?

159.121.2.13 is sending a message to 138.92.0.5.

138.92.6.17 is sending a message to 159.121.66.98

4.5) Click on 138.92.6.17. Write down its statistics.

Sent = 32, received = 0, forwarded= 32, dropped = 0

**Exercise 5.**

Start the “Network router” applet. Select Example 4, the star network.

5.1)Look at the routing and connection tables for the center node and several other nodes. Describe any pattern you can see in these tables.

138.92.0.5 Center: multiple connections. Default for routing.

8.10.20.25: one connection to center. Default for routing.

Non-center nodes all connected to center.

5.2) How is the connection table for the center node different from the other nodes?

Has multiple entries

5.3) Select Generate when I click on a node from the pull-down menu. This means a user at this computer wants to send packets to the 126.14.5.46 computer. If you double-click on 159.121.2.13, you will see that its destination node is 126.14.5.46. Run the applet, click on 159.121.2.13, and watch the packets go. What color does the sending computer turn briefly? What color does the destination computer turn? What does it mean if a node flashes green?

Yellow = send

Red = receive

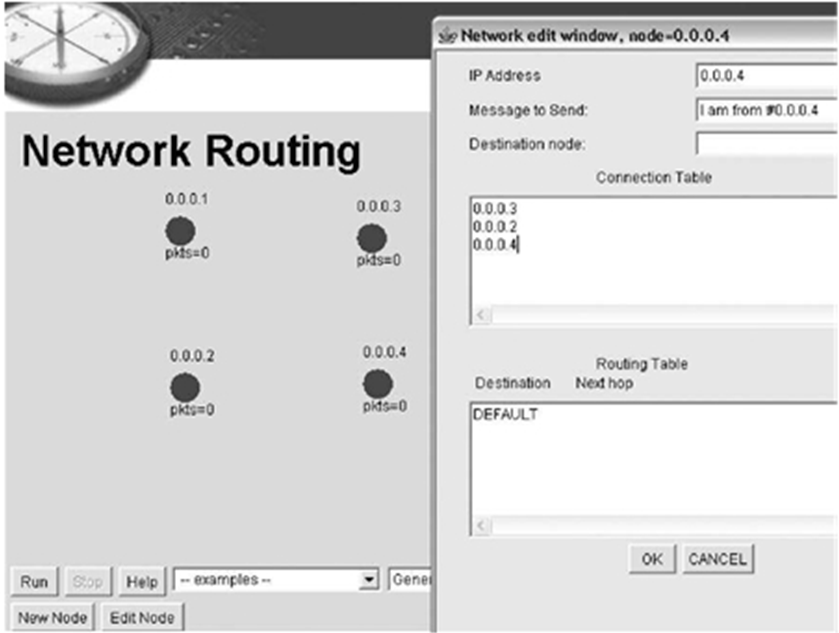
Yellow = forwarded

5.4) Many early computer networks used the star topology (Example 4 on the applet). What will happen if the center node in this type of network dies?

Gee, no one will be able to communicate.

**Exercise 6.**

Start the “Network router” applet but don’t select an example network. Click on the “New Node” button four times, which will place four nodes on your screen (one on top of the other). Move them around into a square. Then double click on node 0.0.0.1. When the edit window appears, type the IP addresses of the other nodes into the Connection Table, as shown:



6.1) Now edit nodes 0.0.0.2, 0.0.0.3, and 0.0.0.4. In each case, add every other node to their Connection Tables. When you are done, every node will have a direct wire to every other node. This is what is called a directly connected network. It is also called a fully connected network, for obvious reasons. Every node in the network has a direct connection to every other node. Is there a need for a routing table in this network? Why or why not?

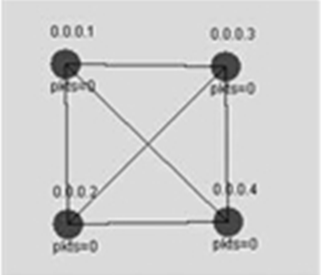
Nope, it is directly connected to every possible destination

6.2) On a separate piece of paper, sketch out a network of 2 nodes and make it directly connected. Do not bother to assign address numbers to the nodes. Just draw circles with lines between them. Now sketch out a network of 3 nodes and directly connect every node. (You do not have to hand in your sketches.) Fill in the table below for 2, 3, and 4 nodes, which compares the number of nodes in a directly connected network to the total number of wires (connections) in the network. Do not double count wires between the same nodes. That is, since there is a wire between 0.0.0.1 and 0.0.0.2, there is also a wire going the other direction, from 0.0.0.2 to 0.0.0.1. Count this as just one wire, not 2.

|  |  |
| --- | --- |
| **Number of nodes** | **Total number of wires** |
| 2 | **1** |
| 3 | **3** |
| 4 | **6** |
| 5 | **10** |
| 10 | **45** |
| 100 | **4950** |

6.3) Look up the number of connections (wires) for a fully-connected network (also called a fully-connected graph). If the number of nodes is N, write the formula for the number of connections, C. Check the answer from the formula with your table entries for 2, 3, and 4. Complete the table above for 5, 10, and 100 using the formula.

**C = n\*(n-1)/2**



**Exercise 7.**

7.1) List one major advantage of a directly connected network.

No single point failure. No routing needed. Very fast since only on hop.

7.2) List one major disadvantage of a directly connected network.

Cost of so many connections.

## Part 3: Open for anything we want to add.

## Part 3: Some probability things about networks

**Exercise 8**

8.1) Suppose that there are 2 nodes that want network access on an Ethernet at the same time. Assume that on a collision, each node generates an integer random wait time of from 1 to 10 milliseconds (ms). What is the probability that the two nodes will generate the same random wait time? Hint: Think about how many possibilities there are and how many of these have both times the same. Show you work. A table is useful.

100 possibilities. 10 of them have collisions. E.g., (1,1), (2,2), etc. Probability of a collision is 10/100 = 0.1

8.2) The method described in 8.1 may not be very efficient because the nodes may waste a lot of time waiting. That is, they could generate wait times of say 7 and 9 ms, and the wait time may be unused. You could help the situation by changing the integer wait times to 1 to 3 milliseconds, but the chance of collision goes up, so you have to wait again. It can get complicated when there are lots of nodes. If you do choose 1 to 3 milliseconds, what is the probability of 2 nodes generating the same wait time? Same hint as in 8.1.

9 possibilities. 3 are collision. Probability is 3/9 = 0.333333…

8.3) Repeat 8.2 for 3 nodes. In this case, you want the probability that all 3 nodes generate the same wait time, or any two nodes generate the same wait time. Hint: Like in 8.1, consider all of the possibilities. Making a table of the possibilities is useful.

27 possibilities. 21 are collisions. You can look at the table or use the fact that not to have a collision that all three times must be different. There are 6 ways this can happen leaving 21 that are collision. Probability is 21/27 = 7/9 = 0.77777…