50.012 Networks (2021 Term 6)

Homework 3

Network/Internet Layer Overview

Hand-out: 9 Nov

Due: 20 Nov 23:59

Name: James Raphael Tiovalen Student ID: 1004555

1. (textbook chapter 4, problem P14) Consider sending a 1,600-byte datagram into a link that has an MTU of 500 bytes. Suppose the original datagram is stamped with the identification number 291. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

Answer: Original data length = 1600 – 20 = 1580 bytes.

Maximum size of data field per fragment = 500 – 20 = 480 bytes.

Number of fragments generated = $\left\lceil \frac{1580}{480} \right\rceil = 4$ fragments.

The four fragments will have these relevant IP datagram header field values:

- Length = 500, ID = 291, fragflag = 1, offset = 0
- Length = 500, ID = 291, fragflag = 1, offset = 60
- Length = 500, ID = 291, fragflag = 1, offset = 120
- Length = 160, ID = 291, fragflag = 0, offset = 180

2. (textbook chapter 4, problem P17) Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned.

Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

a. Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.

Answer: Since all IP packets from inside/behind the NAT are sent to the outside Internet world, we can use and place a packet sniffer just outside the NAT to passively record and monitor all IP packets generated by the hosts behind the NAT. As each host generates a sequence of IP packets with sequential numbers and a distinct initial identification number (ID), which is very likely since they are randomly chosen from a large input space, we can group IP packets with consecutive IDs into a cluster. The number of clusters would be approximately or exactly equal to the number of hosts behind the NAT. While it is possible that multiple clusters will have connected consecutive IDs (and hence wrongly grouped as one clusters), and while it is also possible for consecutive ID sequences to overlap or even be the same (either partially or fully) due to the finite 16-bit space reserved for the identifier field, it is very unlikely, and this technique will still allow us to get some kind of estimate of the number of unique hosts behind the NAT.

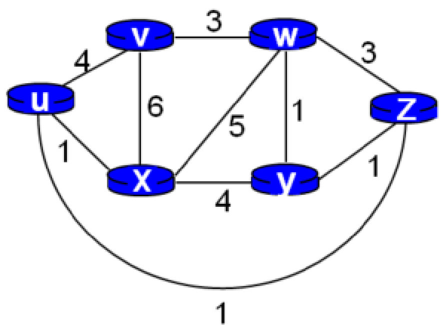
b. If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.

Answer: No, the technique specified in part (a) will not work as there will not be any clusters/groups of IP packets with sequential IDs in the sniffed data.

3. (textbook chapter 4, review problem R26): Suppose you purchase a wireless router and connect it to your cable modem. Also suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?

Answer: Usually the wireless router will include a DHCP server. Hence, DHCP is usually used to assign IP addresses to the 5 PCs and to the router interface. Yes, the wireless router also uses NAT as it obtains only 1 IP address from the ISP, which is not sufficient for all of the other devices that will be connected to the wireless router.

4. (Adapted from 2019 final exam) Consider the network in the Figure below (notice that there is a direct link between node u and z), where the numbers show the symmetrical link costs. Assume a link state routing protocol is used. **Node x** applies Dijkstra's algorithm to compute the best route to every other node. Step 0 of Dijkstra's algorithm (i.e., immediately after initialization) is shown below. Write down **all** the rows after step 0 until the algorithm completes.

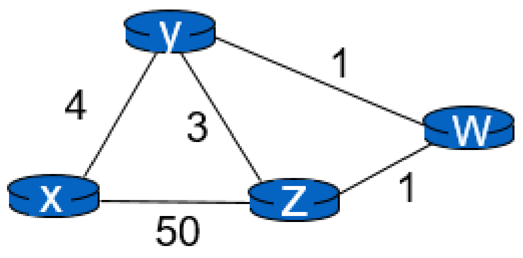


Answer:

In this solution, we assume that we break ties by preferring to choose the option with less hops (i.e., we only update the entry if the new cost is strictly less than the old cost).

Step	N'	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(y), p(y)	D(z), p(z)
0	{x}	1, x	6, x	5, x	4, x	∞, -
1	{x, u}		5, u	5, x	4, x	2, u
2	{x, u, z}		5, u	5, x	3, z	
3	{x, u, z, y}		5, u	4, y		
4	{x, u, z, y, w}		5, u			
5	{x, u, z, y, w, v}					

5. (textbook chapter 5, problem P11): Consider the network below and suppose that poisoned reverse is used in the distance-vector routing algorithm.



a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?
b. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, show the first three rounds of message exchanged among w, y, and z and how their DV change.

Answer:

a)

Router z	Informs w, $D_z(x) = \infty$
	Informs y, $D_z(x) = 6$
Router w	Informs y, $D_w(x) = \infty$
	Informs z, $D_w(x) = 5$
Router y	Informs w, $D_y(x) = 4$
	Informs z, $D_y(x) = 4$

b) Yes, there will be a count-to-infinity problem. The following table shows the first three iterations during the routing converging process. We assume that the link cost change happens at t0 and is immediately detected by y, which will update its distance vector and informs its neighbors w and z at time t1 (round 1).

time	t0	Round 1	Round 2	Round 3
Z		No change.	Updates its own table: $D_z(x) = D_w(x) + c(z,w) = 10 + 1 = 11$	Updates W that $D_z(x) = \infty$ Updates Y that $D_z(x) = 11$
W		Updates its own table: $D_w(x) = D_y(x) + c(w,y) = 9 + 1 = 10$	Updates Y that $D_w(x) = \infty$ Updates Z that $D_w(x) = 10$	No change.
Y	Y detects change of c(x, y), and hence updates its own table: $D_y(x) = D_z(x) + c(y,z) = 6 + 3 = 9$	Updates W that $D_y(x) = 9$ Updates Z that $D_y(x) = \infty$	No change.	Updates its own table: $D_y(x) = D_z(x) + c(y,z) = 11 + 3 = 14$

As shown in the table, w, y, and z forms a loop in their computation of the costs to router x. If we keep continuing the iterations, z would eventually detect that its least cost to x is 50 via its direct link with x at round 27. At round 29, w would learn about its least cost to x is 51 via z. At round 30, y would update its least cost to x to be 52 via w. Finally, at round 31, there will be no more updates, indicating that the routing algorithm has finally stabilized. Therefore, we can see that there is still a count-to-infinity problem.