**CSCE 560 Homework 4**

**Chapter 4 – Network Layer**

**Fall 18**

**Marvin Newlin**

**Assigned: Monday, 5 Nov**

**Due: Monday, 19 Nov, 1400**

**Problem 1**. Chapter 4, R2

We noted that network layer functionality can be broadly divided into data plane functionality and control plane functionality. What are the main functions of the data plane? Of the control plane?

**Sol’n:**

The Control plane runs the routing algorithms that fill up the forwarding table. The data plane looks at the forwarding table to determine what to do with a datagram.

**Problem 2**. Chapter 4, R3

We made a distinction between the forwarding function and the routing function performed in the network layer. What are the key differences between routing and forwarding?

**Sol’n:**

Routing is the act of calculating a full path from point A to point B and running the routing algorithm to determine it (link state or distance vector) and fill up the forwarding table. Forwarding is looking up an entry in the table to determine the next hop for a datagram.

**Problem 3**. Chapter 4, R17

Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to some other upper-layer protocol?

**Sol’n:** The type of service field in the IP datagram header determines the layer 4 protocol that is being used.

**Problem 4**. Chapter 4, R18

What field in the IP header can be used to ensure that a packet is forwarded through no more than N routers?

**Sol’n:** The Time to Live (TTL) field.

**Problem 5**. Chapter 4, R21

Do routers have IP addresses? If so, how many?

**Sol’n:** Yes, they do have IP addresses and every interface of a router has an IP address.

**Problem 6**. Chapter 4, R22

What is the 32-bit binary equivalent of the IP address 223.1.3.27?

**Problem 7**. Chapter 4, R24

Suppose there are three routers between a source host and a destination host. Ignoring fragmentation, an IP datagram sent from the source host to the destination host will travel over how many interfaces? How many forwarding tables will be indexed to move the datagram from the source to the destination?

**Sol’n:**

It will travel through 8 interfaces, 1 each at the source and destination and 2 at each router (entry and exit).

Three forwarding tables will be indexed, one at each router.

**Problem 8**. Chapter 4, R25

Suppose an application generates chunks 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

**Sol’n:** The TCP and IP headers are 20 bytes each so for each datagram, 40 bytes of app data + 40 bytes of overhead so there is 50% overhead per datagram.

**Problem 9**. Chapter 4, 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?**Sol’n:**

The IP addresses are assigned to each device utilizing DHCP and the IP’s assigned are from the private range of IP addresses (in my case 192.168.1.1 is my default gateway). The router does use NAT, otherwise I would need 6 public facing IP addresses, 1 for the router and 5 for my devices.

**Problem 10**. Chapter 4, R32

How does generalized forwarding differ from destination-based forwarding?

**Sol’n:**

Destination based is the traditional way and forwards based only on destination IP address

Generalized forwarding forwards based on any set of header values.

**Problem 11**. Chapter 4, P5

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

11100000 00000000 00000000 00000000

through 0 224.0.0.0 to 224.63.255.255

11100000 00111111 11111111 11111111

11100000 01000000 00000000 00000000

through 1 224.64.0.0 to 224.64.255.255

11100000 01000000 11111111 11111111

11100000 01000001 00000000 00000000

through 2 224.65.0.0 to 224.255.255.255

11100001 01111111 11111111 11111111 225.0.0.0 to 225.127.255.255

Otherwise 3

a. Provide a forwarding table that has four entries, uses longest-prefix matching, and forwards packets to the correct link interfaces. "Otherwise" does not count as one of the four entries. In other words, you should have four entries with prefixes in your table in addition to "Otherwise" for a total of five rows/entries.

**Sol’n:**

|  |  |
| --- | --- |
| **IP Address/Mask** | **Interface** |
| 224.0.0.0/10 | 0 |
| 224.64.0.0/16 | 1 |
| 224.65.0.0/8 | 2 |
| 225.0.0.0/9 | 2 |
| Otherwise | 3 |

b. Describe how your forwarding table determines the appropriate link interface for datagrams with the following destination addresses. Be sure to actually list the interface used for each datagram.

11001000 10010001 01010001 01010101 (200.145.81.85) 3  
11100001 01000000 11000011 00111100 (225.64.195.60) 2  
11100001 10000000 00010001 01110111 (225.128.17.119) 3

11100000 01000000 00010001 01110111 (224.64.17.119) 1

11100000 00010001 01010001 01010101 (224.17.81.85) 0  
223.0.16.1 (11011111 00000000 00010000 00000001) 3

224.253.1.1 (11100000

The process here is to do a binary AND with the subnet mask for however many bits the value will match for each entry in the forwarding table and then take the one that matches the longest prefix of bits.

**Sol’n:**

The first one doesn’t match any of the entries in the table, so it goes out on interface 3

The second one does match the 9 bit mask of 225.0.0.0/9 and it’s the only 225 entry so its interface 2

The third one doesn’t match the 9 bit mask of 225.0.0.0/9 and since it’s the only 225 mask it goes on interface 3

The 4th one matches the 16-bit mask of 224.64.0.0/16 and is longer than the 224.0.0.0/10 mask so it goes on interface 1

The 5th one doesn’t match the 16 bit mask but does match the 10 bit 224 mask so it goes out on interface 0

The 6th one doesn’t match any prefix so it falls into the otherwise category of interface 3

The 7th one matches the 224.65.0.0/8 and doesn’t match the other 224 entries so it is interface 2.

**Problem 12**. Chapter 4, P6

Consider a datagram network using 8-bit host addresses (e.g., 10101111); we are not using 32-bit IP addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match Interface

00 0

010 1

011 2

10 2

11 3

For each of the four interfaces, give the associated range of destination host addresses in binary and decimal and the total number of addresses for each interface.

**Sol’n:**

00000000 🡪 00111111 (0-63) All match only the prefix for interface 0

01000000 🡪 01011111 (64-95) All only match the prefix for interface 1

01100000 🡪 01111111 (96-127) All match only the (011) prefix for interface 2

10000000 🡪 10111111 (128-191) All match only the (10) prefix for interface 2

11000000 🡪 11111111 (192-255) Match the prefix for interface 3

|  |  |
| --- | --- |
| 00000000 🡪 00111111 (0-63) | Interface 0 |
| 01000000 🡪 01011111 (64-95) | Interface 1 |
| 01100000 🡪 10111111 (96-191) | Interface 2 |
| 11000000 🡪 11111111 (192-255) | Interface 3 |

**Problem 13**. Chapter 4, P11

Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network. Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

**Sol’n:**

128.119.40‬.1000000/26

255.255.255.1100000 (AND with mask)

Results in a max IP of 128.119.40.10111111 which is 128.119.40.191 so we can assign 128.119.40.145 to this network.

We have available 128.119.40.128-128.119.40.191 for this block. We need 2 bits more for the subnet now so we need 28 bits overall for the subnet. These are the subnets :

128.119.40.128/28

128.119.40.144/28

128.119.40.160/28

128.119.40.176/28

**Problem 14**. Chapter 5, R4

Compare and contrast link state and distance vector routing algorithms. What information is transmitted and to whom by each router?

**Sol’n:**

Link State sends what nodes it is attached to and the associated cost of each link to every other node at each iteration.

Distance Vector computes the shortest path to every node it knows about and sends that to its neighbors. It recomputes this anytime it receives an update from its neighbors.

**Problem 15**.

Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance vector algorithm and show the distance table entries at node z. Be sure to show how ALL node entries change over time. Finally, show the forwarding table for node z after the algorithm completes.



|  |  |
| --- | --- |
| Indicates updates to other distance vectors | Indicates updates to own distance vector |











**Problem 16**.

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 4.3. Use the table given below; do not change the column headings. You may not need all rows shown. Finally, show the forwarding table for node x after the algorithm completes; remember the forwarding table is derived from the table used in completing the algorithm.



**Sol’n:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | N' | D(s),p(s) | D(t),p(t) | D(u),p(u) | D(v),p(v) | D(w),p(w) | D(y),p(y) | D(z),p(z) |
| 0 | x | ∞ | ∞ | ∞ | 3, x | 6, x | 6, x | ∞ |
| 1 | xv |  | 7, v | 6, v |  | 6, x | 4, v |  |
| 2 | xvy |  | 7, v |  |  |  |  | 16, y |
| 3 | xvyt | 8, t |  | 6, v |  |  |  | 12, t |
| 4 | xvytu | 8, t |  |  |  | 6, x |  |  |
| 5 | xvytuw |  |  |  |  |  |  |  |
| 6 | xvytuws |  |  |  |  |  |  |  |
| 7 | xvytuwsz |  |  |  |  |  |  |  |

Indicates that there were no neighbor nodes not in N’ so picked minimum remaining element.

Forwarding table:

The last column is technically not a part of the forwarding table, but it will help you determine the appropriate output link.

|  |  |  |
| --- | --- | --- |
| Destination | Link | Path from destination back to x |
| s | v | s, t, v |
| t | v | t, v |
| u | v | u, v |
| v | v | v |
| w | w | w |
| y | v | y,v |
| z | v | z, t, v |