Assignment 4

Name: Muhammad Mustafa Manga
Roll #: 17K-3795

Problem 1:

1. Data destined to host H3 is forwarded through interface 3 Destination Address Link Interface H3
2. No, because forwarding rule is only based on destination address.

Problem 2:

1. No, you can only transmit one packet at a time over a shared bus.
2. No, as discussed in the text, only one memory read/write can be done at a time over the shared system bus.
3. No, in this case the two packets would have to be sent over the same output bus at the same time, which is not possible.

Problem 3:

1. (n-1)D
2. (n-1)D
3. 0

Problem 4:

The minimal number of time slots needed is 3. The scheduling is as follows.
Slot 1: send X in top input queue, send Y in middle input queue.
Slot 2: send X in middle input queue, send Y in bottom input queue.
Slot 3: send Z in bottom input queue.
Largest number of slots is still 3.
Actually, based on the assumption that a non-empty input queue is never idle, we see that the first time slot always consists of sending X in the top input queue and Y in either middle or bottom input queue, and in the second time slot, we can always send two more datagram, and the last datagram can be sent in third time slot.

Problem 5:

1)

Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

2)

Prefix match for first address is 5th entry: link interface 3 Prefix match for second address is 3nd entry: link interface 2 Prefix match for third address is 4th entry: link interface 3

Problem 6:

Destination Address Range	**Link Interface**
00000000 through 00111111	0
01000000 through 01011111	1
01100000 through 01111111	2
10000000 through 10111111	2
11000000 through 11111111	3

Number of addresses for interface 0 = 2^6 = 64

Number of addresses for interface 1 = 2^5 = 32

Number of addresses for interface
2 = 2^6 + 2^5 = 64 + 32 = 96
Number of addresses for interface
3 = 2^6 = 64

Problem 7:

Destination Address Range	Link Interface
11000000 Through (32 add) 11011111	0
10000000 Through (64 add) 10111111	1
11100000 Through (32 add) 11111111	2
00000000 Through (128 add) 01111111	3

Problem 8:

223.1.17.0/26
223.1.17.128/25
223.1.17.192/28

Problem 9:

Destination Address	Link Interface
200.23.16/21	0
200.23.24/24	1
200.23.24/21	2
Otherwise	3

Problem 10:

Port numbers are meant to be used for addressing processes, not for addressing hosts. This violation can indeed cause problems for servers running on the home network, since Server processes wait for incoming requests at well-known port numbers and peers in a P2P protocol need to accept incoming connections when acting as servers. Technical solutions to these problems include NAT traversal tools [RFC 5389] and Universal Plug and Play (UPnP), a protocol that allows a host to discover and configure a nearby NAT.

Problem 11:

Any IP address in range 192.168.56.128 to 192.168.56.191 can be assigned to the network

Problem 12:

From 214.97.254/23,
Possible assignments are four equal
size subnets will be:

192.168.56.32/28
192.168.56.48/28
192.168.56.64/28
192.168.56.80/28

a)

Subnet A: 214.97.255/24 (256 addresses)
Subnet B: 214.97.254.0/25 - 214.97.254.0/29 (128-8 = 120 addresses)
Subnet C: 214.97.254.128/25
(128 addresses) Subnet D:
214.97.254.0/31 (2 addresses)
Subnet E: 214.97.254.2/31 (2
addresses) Subnet F:
214.97.254.4/30 (4 addresses)

b)

To simplify the solution, assume that no datagrams have router interfaces as ultimate destinations. Also, label D, E, F for the upper-right, bottom, and upper-left interior subnets, respectively.

Router 1

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111	Subnet A
11010110 01100001 11111110 0000000	Subnet D
11010110 01100001 11111110 000001	Subnet F

Router 2

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111 0000000	Subnet D
11010110 01100001 11111110 0	Subnet B
11010110 01100001 11111110 0000001	Subnet E

Router 3

Longest Prefix Match	**Outgoing Interface**
11010110 01100001 11111111 000001	Subnet F
11010110 01100001 11111110 0000001	Subnet E
11010110 01100001 11111110 1	Subnet C

Problem 13:

Three Services are provided by Internet Protocol Security, which are as follow:

Message Confidentiality: Protect against unauthorized data disclosure. Accomplished by the use of encryption mechanisms.

Traffic Analysis Protection: A person monitoring network traffic does not know which parties are communicating, how often communications are occurring, or how much data is being exchanged. Provided by concealing IP datagram details such as source and destination address.

Message Integrity: IPsec can determine if data has been changed (intentionally or unintentionally) during transit. The integrity of data can be assured by generating a message authentication code (MAC) value, which is a cryptographic checksum of the data. If the data is altered and the MAC is recalculated, the old and new MACs will differ.

Problem 14:

1. The maximum size of data field in each fragment = 680 (because there are 20 bytes IP header).Thus the number of required fragments ceil(2400-20/680) = 4

2. Each fragment will have Identification number 422. Each fragment except the last one will be of size 700 bytes (including IP header). The last datagram will be of size 360 bytes (including IP header). The offsets of the 4 fragments will be 0, 85, 170, 255. Each of the first 3 fragments will have flag=1; the last fragment will have flag=0.

Problem 15:

MP3 file size = 5 million bytes. Assume the data is carried in TCP segments, with each TCP segment also having 20 bytes of header. Then each datagram can carry 1500-40=1460 bytes of the MP3 file.

Number of datagrams required = ceil (5x10^6 /1460)= 3425 . All but the last datagram will be 1,500 bytes; the last datagram will be 960+40 = 1000 bytes. Note that here there is no fragmentation

The source host does not create datagrams larger than 1500 bytes, and these datagrams are smaller than the MTUs of the links.

Problem 16:

a) Home addresses: 192.168.1.1, 192.168.1.2, 192.168.1.3 with the router interface being 192.168.1.4

b)

NAT Translation Table

WAN Side	**LAN Side**
24.34.112.235, 4000	192.168.1.1,3345
24.34.112.235, 4001	192.168.1.1, 3346
24.34.112.235, 4002	192.168.1.2, 3445
24.34.112.235, 4003	192.168.1.2, 3446
24.34.112.235, 4004	192.168.1.3, 3545
24.34.112.235, 4005	192.168.1.3, 3546

Problem 17:

1. Since all IP packets are sent outside, so we can use a packet sniffer to record all IP packets generated by the hosts behind a NAT. As each host generates a sequence of IP packets with sequential numbers and a distinct (very likely, as they are randomly chosen from a large space) initial identification number (ID), we can group IP packets with consecutive IDs into a cluster. The number of clusters is the number of hosts behind the NAT. For more practical algorithms, see the following papers.
   A Technique for Counting NATted Hosts, by Steven M.Bellovin, appeared in IMW'02, Nov. 6-8, 2002, Marseille, France.
   Exploiting the IPID field to infer network path and end-system characteristics. Weifeng Chen, Yong Huang, Bruno F.Ribeiro, Kyoungwon Suh, Honggang Zhang, Edmundo de Souza e Silva, Jim Kurose, and Don Towsley.
   PAM'05 Workshop, March 31 - April 01, 2005. Boston, MA, USA.

2. However, if those identification numbers are not sequentially assigned but randomly assigned, the technique suggested in part (a) won't work, as there won't be clusters in sniffed data.

Problem 18:

It is not possible to devise such a technique. In order to establish a direct TCP connection between Arnold and Bernard, either Arnold or Bob must initiate a connection to the other. But the NATs covering Arnold and Bob drop SYN packets arriving from the WAN side. Thus neither Arnold nor Bob can initiate a TCP connection to the other if they are both behind NATs.

Problem 19:

S2 Flow Table	
Match	**Action**
Ingress Port = 1; IP Src = 10.3.*. *; IP Dst = 10.1.*. *	Forward (2)
Ingress Port = 2; IP Src = 10.1.*. *; IP Dst = 10.3.*. *	Forward (1)
Ingress Port = 1; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 2; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 1; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 2; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 4	Forward (3)
Ingress Port = 3	Forward (4)

Problem 20:

***S2 Flow* Table**	
Match	**Action**
Ingress Port = 3; IP Dst = 10.1.*. *	Forward (2)
Ingress Port = 3; IP Dst = 10.3.*. *	Forward (2)
Ingress Port = 4; IP Dst = 10.1.*. *	Forward (1)
Ingress Port = 4; IP Dst = 10.3.*. *	Forward (1)

Problem 21:

S1 Flow Table	
Match	**Action**
IP Src = 10.2.*. *; IP Dst = 10.1.0.1	Forward (2)
IP Src = 10.2.*. *; IP Dst = 10.1.0.2	Forward (3)
IP Src = 10.2.*. *; IP Dst = 10.3.*. *	Forward (1)

S3 Flow Table	
Match	**Action**
IP Src = 10.2.*. *; IP Dst = 10.3.0.6	Forward (1)
IP Src = 10.2.*. *; IP Dst = 10.3.0.5	Forward (2)
IP Src = 10.2.*. *; IP Dst = 10.1.*. *	Forward (3)

Problem 22:

S2 Flow Table	
Match	**Action**
IP Src = 10.1.0.1; IP Dst = 10.2.0.3	Forward (3)
IP Src = 10.1.0.1; IP Dst = 10.2.0.4	Forward (4)
IP Src = 10.3.0.6; IP Dst = 10.2.0.3	Forward (3)
IP Src = 10.3.0.6; IP Dst = 10.2.0.4	Forward (4)

S2 Flow Table	
Match	**Action**
IP Src =.*.*.*. *; IP Dst = 10.2.0.3; port = TCP	Forward (3)
IP Src =.*.*.*. *; IP Dst = 10.2.0.4; port = TCP	Forward (4)

S2 Flow Table	
Match	**Action**
IP Src =.*.*.*. *; IP Dst = 10.2.0.3	Forward (3)

S2 Flow Table	
Match	**Action**
IP Src = 10.1.0.1; IP Dst = 10.2.0.3; port = UDP	Forward (3)