Problem 1

Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router.

(a) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a shared bus?

(b) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses switching via memory?

(c) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a crossbar?

Write your solution to Problem 1 in this box

a) No, it is not possible to forward two packets through the switching fabric via a shared bus at the same time. This is because the shared bus only allows one packet to be on the bus at one time.

b) No, it is not possible to forward two packets through the switching fabric via memory at the same time. This is because there is are two buses connecting the memory to the input and output ports respectively, limiting the possible number of packets being forwarded to one.

c) Yes, it is possible to forward two packets through the switching fabric via a crossbar or interconnected network at the same time. This is because each combination of input and output ports is connected. So, we can forward more than one packet at a time.

Problem 2

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 8 interfaces. Provide three subnet addresses (of the form a.b.c.d/x) that satisfy the constraints. You may use the following link to help verify your result: `http://jodies.de/ipcalc`.

Write your solution to Problem 2 in this box

Subnet 1: 223.1.17/26
This is because we can support 2^(32 - 26) - 2 possible hosts. The result is 62, which is greater than 60. We subtract 2 because two addresses are reserved for the network ID and the broadcast ID.

Subnet 2: 223.1.17/25
This is because we can support 2^(32 - 25) - 2 possible hosts. The result is 126, which is greater than 90.

Subnet 3: 223.1.17/28
This is because we can support 2^(32 - 28) - 2 possible hosts. The result is 14, which is greater than 8. Notice we cannot use 223.1.17/29, because although there are 8 addresses in total, two are reserved.

Problem 3

Consider sending a datagram with total length 2400 B into a link that has an MTU (maximum transmission unit) of 800 B. Suppose the original datagram is stamped with the identification number 421.

(a) How many fragments are generated?

(b) What are the values in the various fields (header length, total length, identification, MF flag, fragment offset, and IP payload size) in the IP datagram(s) generated related to fragmentation?

Write your solution to Problem 3 in this box

a) There are 4 segments.

Segment	Header(B)	Total(B)	Payload	Offset
1	20	796	776	0
2	20	796	776	97
3	20	796	776	194
4	20	92	72	291

b)

Segment 1:
- Header length: 5 (counted in number of words where each words is 4 Bytes)
- Total length: 796B
- Identification: 422
- MF flag: 1
- Fragment offset: 0
- Payload Size: 776B

Segment 2:
- Header length: 5
- Total length: 796B
- Identification: 422
- MF flag: 1
- Fragment offset: 97
- Payload Size: 776B

Segment 3:
- Header length: 5
- Total length: 796B
- Identification: 422
- MF flag: 1
- Fragment offset: 194
- Payload Size: 776B

Segment 4:
- Header length: 5
- Total length: 92B
- Identification: 422
- MF flag: 0
- Fragment offset: 291
- Payload Size: 72B

Problem 4

In this problem we will explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

Write your solution to Problem 4 in this box

We cannot devise such a technique if there is no application-specific NAT configuration. This is because NAT will drop packets from unknown hosts, unless configured otherwise by the NAT configuration. This is an issue during the initial 3-way handshake phase, as the NAT on Bernard's side does not contain a configuration that allows packets from Arnold's NAT. Thus, Bernard's NAT will drop the packet.

Problem 5

Consider the SDN OpenFlow network shown as follows. Suppose that the desired forwarding behavior for datagrams arriving at `s3` is as follows:

- Any datagrams arriving on input port 4 from hosts `h3` or `h4` that are destined to hosts `h1` or `h2` should be forwarded over output port 3;
- Any datagrams arriving on input port 3 from hosts `h1` or `h2` that are destined to hosts `h3` or `h4` should be forwarded over output port 4;
- Any arriving datagrams on input ports 3 or 4 and destined to hosts `h5` or `h6` should be delivered to the host specified;
- Host `h5` and `h6` should be able to send datagram to each other.

Specify the flow table entries in `s3` that implement this forwarding behavior.

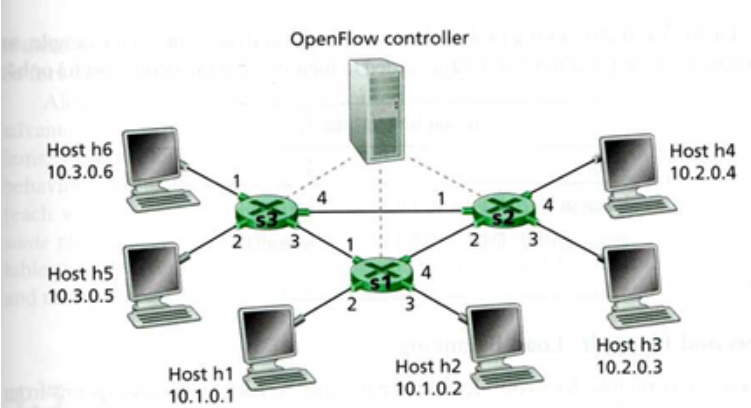

Figure 4.30 ♦ OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Problem 5 continued on next page...

Write your solution to Problem 5 in this box

S3 Forwarding Table:

Match	Action
Ingress Port: 4 IP src: 10.2.0.* IP dest: 10.1.0.*	forward(3)
Ingress Port: 3 IP src: 10.1.0.* IP dest: 10.2.0.*	forward(4)
Ingress Port: 3 IP src: 10.1.0.* IP dest: 10.3.0.5	forward(2)
Ingress Port 3: IP src: 10.1.0.* IP dest: 10.3.0.6	forward(1)
Ingress Port 4: IP src: 10.2.0.* IP dest: 10.3.0.5	forward(2)
Ingress Port 4: IP src: 10.2.0.* IP dest: 10.3.0.6	forward(1)
Ingress Port 1: IP src: 10.3.0.6 IP dest: 10.3.0.5	forward(2)
Ingress Port 2: IP src: 10.3.0.5 IP dest: 10.3.0.6	forward(1)