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?

(a) No, you can only transmit one packet at a time over a shared bus.

(b) No, you can only transmit one packet because only one read/write can be done at a time.

(c) Yes, as discussed in the text, as long as the two packets use different input buses and different output buses, they can be forwarded in parallel.

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`.

- 223.1.17.0/26
- 223.1.17.128/25
- 223.1.17.192/28

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, and fragment offset) in the IP datagram(s) generated related to fragmentation?

The maximum size of data field in each fragment = 776 (because there are 20 bytes IP header and it should be divisible by 8). Thus the number of required fragments is 4. Each fragment will have identification number 421. Each fragment except the last one will be of size 796 bytes (including IP header), with payload length 776. The last datagram will be of size 72 bytes (including IP header), with payload length 52. The offsets of the 4 fragments will be 0, 97, 194, 291. Each of the first 3 fragments will have flag=1; the last fragment will have flag=0.

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.

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 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.

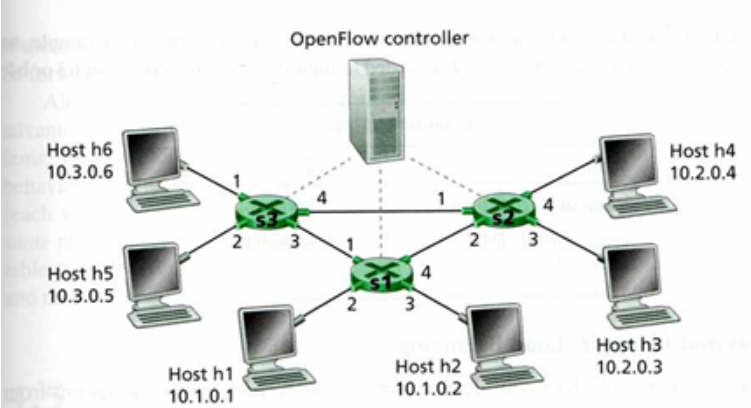

Figure 4.30 ♦ OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Problem 5 continued on next page...

S3 Flow Table	
Match	Action
Ingress Port = 4; IP Src = 10.2.*.*; IP Dst = 10.1.*.*	Forward (3)
Ingress Port = 3; IP Src = 10.1.*.*; IP Dst = 10.2.*.*	Forward (4)
Ingress Port = 3; IP Dst = 10.3.0.5	Forward (2)
Ingress Port = 4; IP Dst = 10.3.0.5	Forward (2)
Ingress Port = 3; IP Dst = 10.3.0.6	Forward (1)
Ingress Port = 4; IP Dst = 10.3.0.6	Forward (1)
Ingress Port = 1	Forward (2)
Ingress Port = 2	Forward (1)

Note: You can write one asterick in src address, e.g. Src=10.2.0.* for first two rules. For last two rules, you can combine filtering on src address together with filtering on ingress port number.