# 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. Only one packet can cross the bus at a time.

(b) No. Only one memory read/write can be done at a time over the shared system bus.

(c) Yes. The packets are to be forwarded to different output ports, so crossbar will assign them different input and output busses. Crossbar is non-blocking when output ports are different.

# 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.

The first 24 bits are fixed. So we have flexibility in the last 8 bits. Subnet 1 requires a block of log60=6 bits. Subnet 2 requires 7 bits. Subnet 3 requires log8=3 bits. Thus, the following will work:

Assign 223.1.17.128/ 25 to subnet2, so that it has control of the last 7 bits.
Assign 223.1.17.64/26 to subnet1, so that it has control of the last 6 bits.
Assign 223.1.17.0/29 to subnet 3, so that it has control of the last 3 bits.

# 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?

(a) we have in total 2400-20=2380 bytes to be sent. Since the fragment offset is 13 bits long, we want it to be multiple of 8 bytes, thus the maximum size of the fragment is 796 byte, and each packet has a payload of 776 bytes. Thus in total we need 2380/776= 4 fragments.

(b)

| Header length (bytes) | Total length (bytes) | Identification number | MF flag | Fragment offset | IP payload size |
|---|---|---|---|---|---|
| 20 | 796 | 421 | 1 | 0 | 776 |
| 20 | 796 | 421 | 1 | 97 | 776 |
| 20 | 796 | 421 | 1 | 194 | 776 |
| 20 | 72 | 421 | 0 | 291 | 52 |

# 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.

There will be difficulties. Users behind NAT has to initiate the request first, users outside NAT cannot connect to users behind NAT directly. When Arnold's request arrive at Bernard's NAT router, since Bernard has not initiate any request to Arnold's NAT router, the NAT translation table of Bernard's NAT router will not recognize the IP address of Arnold's NAT router. Thus the packet will be dropped at Bernard's NAT router.

# 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.

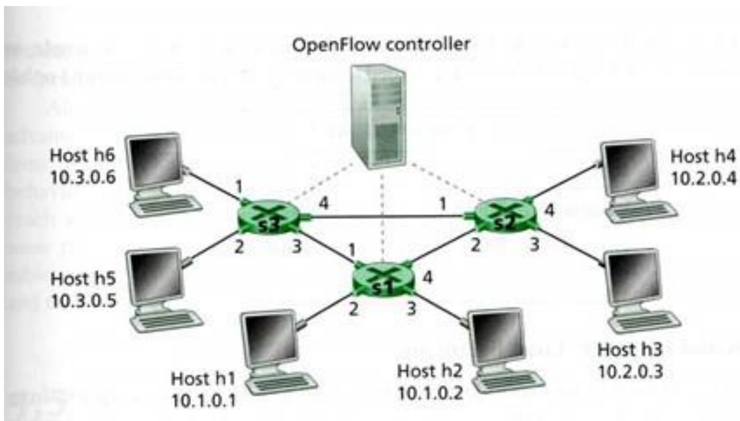


**Figure 4.30** ♦ OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Problem 5 continued on next page. . . 

| | | | |
|---|---|---|---|
| Ingress port=4 | IPSrc=10.2.*.* | IPDst=10.1.*.* | Forward(3) |
| Ingress port=3 | IPSrc=10.1.*.* | IPDst=10.2.*.* | Forward(4) |
| Ingress port=3 | | IPDst=10.3.0.6 | Forward(1) |
| Ingress port=4 | | IPDst=10.3.0.6 | Forward(1) |
| Ingress port=3 | | IPDst=10.3.0.5 | Forward(2) |
| Ingress port=4 | | IPDst=10.3.0.5 | Forward(2) |
| | IPSrc=10.3.0.5 | IPDst=10.3.0.6 | Forward(1) |
| | IPSrc=10.3.0.6 | IPDst=10.3.0.5 | Forward(2) |