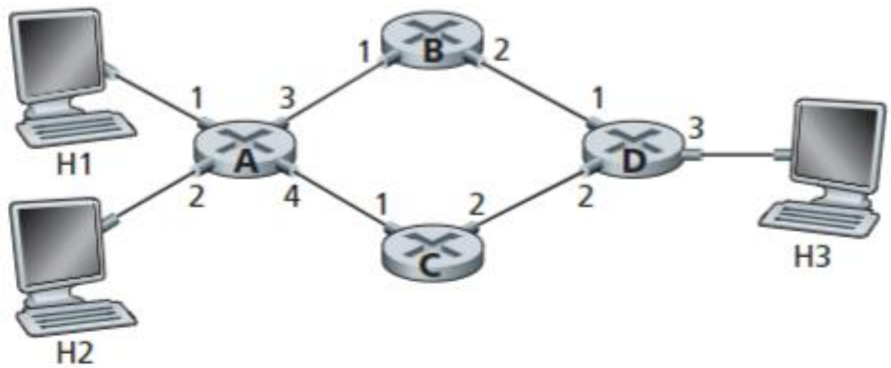
Ramon Veal

P4. Consider the network below.




a. Suppose that this network is a datagram network. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.

| Destination Address | Link Interface |
|---|---|
| H3 | 3 |

b. Suppose that this network is a datagram network. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (Hint: this is a trick question.)

Not possible as only destination address is valid

P10. 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<br>through<br>11100000 00111111 11111111 11111111 | 0 |

| | |
|---|---|
| 11100000 01000000 00000000 00000000<br>through<br>11100000 01000000 11111111 11111111 | 1 |
| 11100000 01000001 00000000 00000000<br>through<br>11100001 01111111 11111111 11111111 | 2 |
| otherwise | 3 |

a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

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

b. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11001000 10010001 01010001 01010101
11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111

Prefix match for 1st address is 5th entry: link interface 3

Prefix match for 2nd address is 3nd entry: link interface 2

Prefix match for 3rd address is 4th entry: link interface 3

P12. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

| Prefix Match | Interface |
|---|---|
| 1 | 0 |
| 10 | 1 |
| 111 | 2 |
| otherwise | 3 |

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

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

P13. 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 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

Subnet 1:

60 < 2^6

32 – 6 = 26 1's

11111111.11111111.11111111.11000000

255.255.255.192

223.1.17.0/26

Subnet 2:

90 < 2^7

32 – 7 = 25

11111111.11111111.11111111.10000000

255.255.255.128

223.1.17.128/25

Subnet 3:

12 < 2^4

32 – 4 = 28

11111111.11111111.11111111.11110000

255.255.255.240

223.1.17.192/28

P17. Consider the topology shown in Figure 4.17. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

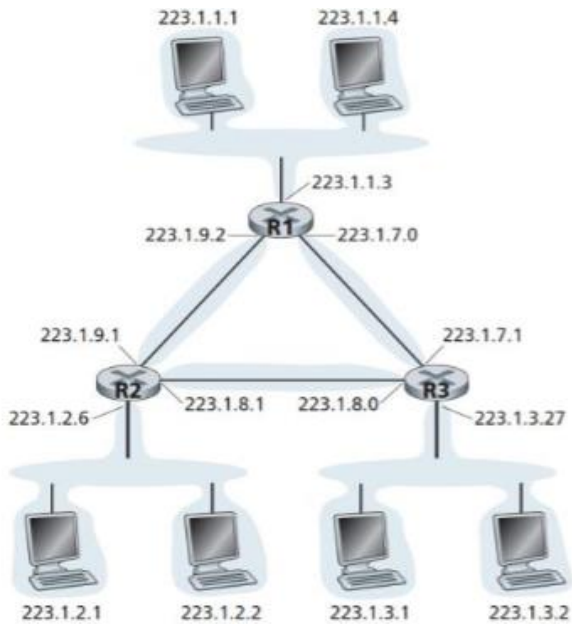



**Figure 4.17** ♦ Three routers interconnecting six subnets

a. Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces. For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x – e.f.g.h/y.

A: 250 < 2^8 -> 24 1's -> 214.97.254.0/24

B: 120 < 2^7 -> 25 1's -> 214.97.254.0/25

C: 120 < 2^7 -> 25 1's -> 214.97.254.128/25

D: 2 < 2^1 -> 31 1's -> 214.97.254.0/31

E: 2 < 2^1 -> 31 1's -> 214.97.254.2/31

F: 2 < 2^1 -> 31 1's -> 214.97.254.4/31

b. Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

Router 1

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 | A |
| 11010110 01100001 11111110 0000000 | D |
| 11010110 01100001 11111110 000001 | F |

Router 2

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 0000000 | D |
| 11010110 01100001 11111110 0 | B |
| 11010110 01100001 11111110 0000001 | E |

Router 3

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 000001 | F |
| 11010110 01100001 11111110 0000001 | E |
| 11010110 01100001 11111110 1 | C |

P19. Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

2400/(700-20) = 3.5 -> 4

F1: ID = 422, fragment = 1, length = 700, offset = 0

F2: ID = 422, fragment = 1, length = 700, offset = (700-20)/8 = 85

F3: ID = 422, fragment = 1, length = 700, offset = 170

F4: ID = 422, fragment = 0, length = 360, offset = 255

P21. Consider the network setup in Figure 4.22. Suppose that the ISP instead assigns the router the address 24.34.112.235 and that the network address of the home network is 192.168.1/24.

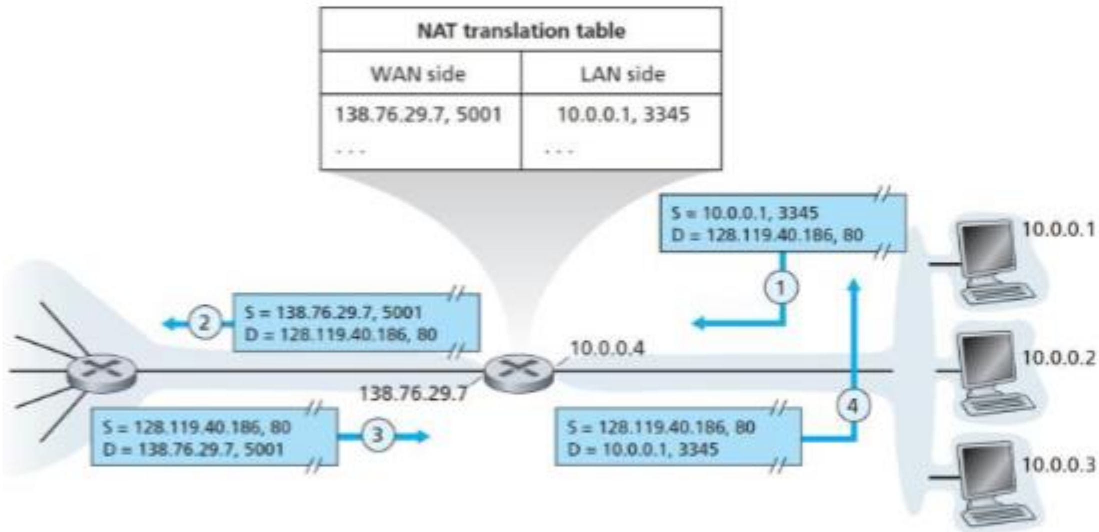



**Figure 4.22** ♦ Network address translation

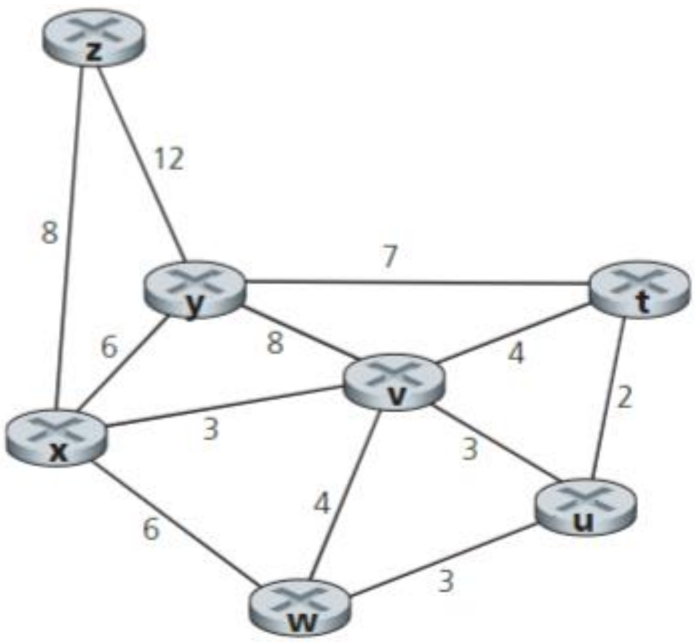
a. Assign addresses to all interfaces in the home network.

192.168.1.1, 192.168.1.2, 192.168.1.3, 192.168.1.4

b. Suppose each host has two ongoing TCP connections, all to port 80 at host 128.119.40.86. Provide the six corresponding entries in the NAT translation table.

| WAN | LAN |
|---|---|
| 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 |

P26. 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.




| step | N' | D(v),p(v) | D(w),p(w) | D(x),p(x) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2,u | 5,u | 1,u | ∞ | ∞ |
| 1 | ux | 2,u | 4,x | | 2,x | ∞ |
| 2 | uxy | 2,u | 3,y | | | 4,y |
| 3 | uxyv | | 3,y | | | 4,y |
| 4 | uxyvw | | | | | 4,y |
| 5 | uxyvwz | | | | | |

**Table 4.3** ♦ Running the link-state algorithm on the network in Figure 4.27

| step | N' | D(t), p(t) | D(u), p(u) | D(v), p(v) | D(w), p(w) | D(x), p(x) | D(y), p(y) | D(z), p(z) |
|---|---|---|---|---|---|---|---|---|
| 0 | X | inf | inf | 3,x | 6,x | | 6,x | 8,x |
| 1 | Xv | 7,v | 6,v | **3,x** | 6,x | | 6,x | 8,x |
| 2 | Xvu | 7,v | **6,v** | 3,x | 6,x | | 6,x | 8,x |
| 3 | Xvuw | 7,v | 6,v | 3,x | **6,x** | | 6,x | 8,x |
| 4 | Xvuwy | 7,v | 6,v | 3,x | 6,x | | **6,x** | 8,x |
| 5 | Xvuwyt | **7,v** | 6,v | 3,x | 6,x | | 6,x | 8,x |
| 6 | xvuwytz | 7,v | 6,v | 3,x | 6,x | | 6,x | **8,x** |