Name: Zouyan Song

Title: HW4

Course number: EE450

**Chapter 4**

**P8**. 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

表格

描述已自动生成

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

|  |  |
| --- | --- |
| Prefix | Link I/F |
| 11100000 00 | 0 |
| 11100000 01000000 | 1 |
| 1110000 | 2 |
| 11100001 1 | 3 |
| otherwise | 3 |

1. 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

* First address:

**11001**000 10010001 01010001 01010101 -> not start with 11100, matches interface **3**

Second address:

**1110000**1 01000000 11000011 00111100 -> starts with 1110000, matches interface **2**

Third address:

**11100001 1**0000000 00010001 01110111 -> starts with 11100001 1, matches interface **3**

**P9**. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table

表格

描述已自动生成

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 | I/F |
| 00000000 through 00111111 | 0 |
| 01000000 through 01011111 | 1 |
| 01100000 through 01111111 | 2 |
| 10000000 through 10111111 | 2 |
| 11000000 through 11111111 | 3 |

* # of addresses:

Interface 0:

Interface 1:

Interface 2:

Interface 3:

**P10.** Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

图表

中度可信度描述已自动生成

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 | I/F |
| 11000000 through 11011111 | 0 |
| 10000000 through 10111111 | 1 |
| 11100000 through 11111111 | 2 |
| 00000000 through 01111111 | 3 |

* # of addresses:

Interface 0:

Interface 1:

Interface 2:

Interface 3:

**P11.** 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

* For Subnet 1,

since , the address could be given as: **223.1.17.0/26**

* For Subnet 2,

since , the address could be given as: **223.1.17.128/25**

* For Subnet 3,

since , the address could be given as: **223.1.17.192/28**

**P12**. In Section 4.2.2, an example forwarding table (using longest prefix matching) is given. Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

文本

中度可信度描述已自动生成

* Rewritten in CIDR notation:

|  |  |
| --- | --- |
| Prefix | I/F |
| 200.23.16/21 | 0 |
| 200.23.24/24 | 1 |
| 200.23.24/21 | 2 |
| Otherwise | 3 |

**P13.** In Problem P8, you are asked to provide a forwarding table (using longest prefix matching). Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

|  |  |
| --- | --- |
| Prefix | I/F |
| 224.0.0.0/10 | 0 |
| 224.64.0.0/16 | 1 |
| 224.0.0.0/8 | 2 |
| 225.128.0.0/9 | 3 |
| Otherwise | 3 |

**P14.** Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network. Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

* **128.119.40.128** could be assigned to this network (128.119.40.128/26)
* Those four subnets could be given as:

**128.119.40.64/28**

**128.119.40.80/28,**

**128.119.40.96/28**

**128.119.40.112/28**

**P15.** Consider the topology shown in Figure 4.20. 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.

图示

描述已自动生成

1. 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.

* Those subnets could be given as:

**Subnet A**: **214.97.255/24**  ()

**Subnet B**: **214.97.254.0/25**  ()

**Subnet C**: **214.97.254.128/25** ()

**Subnet D**: **214.97.254.0/31** ()

**Subnet E**: **214.97.254.2/31** ()

**Subnet F**: **214.97.254.4/30** ()

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

* Router 1:

|  |  |
| --- | --- |
| Prefix | I/F |
| 11010110 01100001 11111111 | I/F to Subnet A |
| 11010110 01100001 11111110 0000000 | I/F to Subnet D |
| 11010110 01100001 11111110 000001 | I/F to Subnet F |

Router 2:

|  |  |
| --- | --- |
| Prefix | I/F |
| 11010110 01100001 11111110 0 | I/F to Subnet B |
| 11010110 01100001 11111111 0000000 | I/F to Subnet D |
| 11010110 01100001 11111110 0000001 | I/F to Subnet E |

Router 3:

|  |  |
| --- | --- |
| Prefix | I/F |
| 11010110 01100001 11111110 1 | I/F to Subnet C |
| 11010110 01100001 11111110 0000001 | I/F to Subnet E |
| 11010110 01100001 11111111 000001 | I/F to Subnet F |

**Chapter 5**

**P3.** 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 5.1

图示

描述已自动生成

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | N’ | D(y), p(y) | D(z), p(z) | D(v), p(v) | D(w), p(w) | D(t), p(t) | D(u), p(u) |
| 0 | x | 6, x | 8, x | 3, x | 6, x | ∞ | ∞ |
| 1 | x,v | 6, x | 8, x |  | 6, x | 7, v | 6, v |
| 2 | x,v,y |  | 8, x |  | 6, x | 7, v | 6, v |
| 3 | x,v,y,w |  | 8, x |  |  | 7, v | 6, v |
| 4 | x,v,y,w,u |  | 8, x |  |  | 7, v |  |
| 5 | x,v,y,w,u,t |  | 8, x |  |  |  |  |
| 6 | x,v,y,w,u,t,z |  |  |  |  |  |  |

**P5.** Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.

图示

描述已自动生成

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Cost to | | | | |
| u | v | x | y | z |
| From | v | ∞ | ∞ | ∞ | ∞ | ∞ |
| x | ∞ | ∞ | ∞ | ∞ | ∞ |
| z | ∞ | 6 | 2 | ∞ | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Cost to | | | | |
| u | v | x | y | z |
| From | v | 1 | 0 | 3 | ∞ | 6 |
| x | ∞ | 3 | 0 | 3 | 2 |
| z | 7 | 5 | 2 | 5 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Cost to | | | | |
| u | v | x | y | z |
| From | v | 1 | 0 | 3 | 3 | 5 |
| x | 4 | 3 | 0 | 3 | 2 |
| z | 6 | 5 | 2 | 5 | 0 |