**Chapter 4**

**P1**

a.

|  |  |
| --- | --- |
| Destination IP Address | Output port |
| IP address of H3 | 3 |

b.

|  |  |
| --- | --- |
| Input port && Destination IP address | Output port |
| Input port 1 && IP address of H3 | 3 |
| Input port 2 && IP address of H3 | 4 |

**P6**

Procedure:

Let’s see entry “011”, the range is , that is 96~127, the port is 2;

Entry “010”, the range is, that is 64~95, the port is 1;

Entry “11”, the range is , that is 192~255, the port is 3;

Entry “10”, the range is ,that is 128~191, the port is 2;

Entry “00”, the range is , that is 0~63, the port is 0;

Result :

For port 0, the range is 0~63, the number of addresses is 64;

For port 1, the range is 64~95, the number of addresses is 32;

For port 2, the range is 96~191, the number of addresses is 96;

For port 3, the range is 192~255, the number of addresses is 64;

**P8.**

The network address of subnet 1: 223.1.17.64/26 ( support 64 interfaces)

The network address of subnet 2: 223.1.17.128/25 ( support 128 interfaces)

The network address of subnet 3: 223.1.17.0/26 (support 64 interfaces)

**P14**

The number of fragments: 4.

There are 20 bytes for header.

First fragment:

Length: 700 ID: 422 Flag: 1 Offset: 0( According to IPv4’s corresponding RFC, the offset = 0/8= **0**)

Second fragment:

Length: 700 ID: 422 Flag: 1 Offset: 680( According to IPv4’s corresponding RFC, the offset = 680/8= **85**)

Third fragment:

Length: 700 ID: 422 Flag: 1 Offset: 1360 ( According to IPv4’s corresponding RFC, the offset = 0/8=**170 )**

Fourth fragment:

Length: 360 ID: 422 Flag: 0 Offset: 2040(According to IPv4’s corresponding RFC, the offset= 2040/8= **255**)

**P19:**

|  |  |
| --- | --- |
| S2 Flow Table (Match) | Action |
| Ingress port = 1; IP Src = 10.3.\*.\*; IP Dst = 10.1.\*.\* | Forward(2) |
| Ingress port = 2; IP Src = 10.1.\*.\*; IP Dst = 10.3.\*.\* | Forward(1) |
| IP Dst = 10.2.0.4 | Forward(4) |
| IP Dst = 10.2.0.3 | Forward(3) |

**Chapter 5:**

**P3:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | N | D(z),P(z) | D(y),P(y) | D(v),P(v) | D(w),P(w) | D(u),P(u) | D(t),P(t) |
| 0 | X | 8, x | 6, x | 3, x | 6, x | ∞ | ∞ |
| 1 | x, v | 8, x | 6, x |  | 6, x | 6, v | 7, v |
| 2 | x, v, y | 8, x |  |  | 6, x | 6, v | 7, v |
| 3 | x, v, y, w | 8, x |  |  |  | 6, v | 7, 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 |  |  |  |  |  |  |

**P7:**

(Please notice the difference between the “ inform its neighbors of a new distance vector” and “inform its neighbors of a new minimum-cost path to u”, I have listed every possible results).

1. Distance vector of x:

D(x,x): 0;

D(x,w): 2;

D(x,y): 4;

D(x,u): 7;

1. If we change c(x,w), whatever we change it to, x will inform a new minimum-cost path to u.

If we change c(x,y),

If c(x,y)>=1 && c(x,y) <4, x will inform a new minimum-cost path to **y** to its neighbors.

If c(x,y)<1, x will inform a new minimum-cost path to **u** to its neighbors.

1. If we change c(x,w),

whatever the result is, x will definitely inform its neighbors of a new minimum-cost path to u.

If we change c(x,y),

if c(x,y)>=4, x will not inform its neighbors of a new distance vector.

If c(x,y)>=1, x will not inform its neighbors of a new minimum-cost path to u.

**P14:**

1. Router 3c learns about prefix x from 4c, 4c learns from 4a: RIP -- eBGP.

Thus router 3c learns x from eBGP.

1. Router 3a learns about prefix x from 3b, 3b learns x from 3c. RIP-eBGP-iBGP

Thus router 3a learns x from iBGP.

1. Router 1c learns x from 3a. The route is: RIP-eBGP-iBGP-eBGP.

Thus router 1c learns x from eBGP.

1. Router 1d learns prefix x from 1c. The route is: RIP-eBGP-iBGP-eBGP-iBGP.

Thus router 1d learns x from iBGP.

**P19:**

A should advertise A-W to B

A should advertises A-V to B and C.

B advertise B-A-W to C, Such that C receive A-V, B-A-W;

**Chapter 8:**

**R19:**

Bob needs to use Alice’s public key to decrypt the digital signature of that message. Then he needs to use the Hash function to calculate the message digest of that message body. Finally he compare the two results and determine whether this message is sent by Alice and integrated.

No, PGP doesn’t use MAC for message integrity.

**P4:**

1. 00000101 repeated eight times
2. 00000101 repeated seven times + 10000101
3. First loop: 10100001 + 10100000 repeated 7 times

Second loop : 10100000 repeated 7 times + 10100001

Third loop: 101000001 + 10100000 repeated 7 times