### 第五章作业

#### 5-5

|  |  |  |  |
| --- | --- | --- | --- |
| to router | from B | from D | from E |
| A | 11 | 19 | 12 |
| B | 6 | 15 | 11 |
| C(itself) | - | - | - |
| D | 18 | 3 | 14 |
| E | 12 | 12 | 5 |
| F | 8 | 13 | 9 |

For each row, choose column router with the minimum delay value to be the outgoing line. Result routing table:

|  |  |
| --- | --- |
| outgoing line | expected delay |
| B | 11 |
| B | 6 |
| -(Itself) | 0 |
| D | 3 |
| E | 5 |
| B | 8 |

#### 5-20

Data that IP should transmit: 920 byte. Assume a 20 byte IPv4 header.

A-R1: Maximum IP packet size = 1024-14-20 byte > 920 byte, so we only need 1 frame to carry all the data.

R1-R2: Maximum IP packet size = 512-8-20 byte < 920 byte, so there will be fragmentation.

R2-B: Maximum IP packet size = 512-12-20 = 480 byte. No fragmentation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | total length | identification | DF | MF | fragment offset |
| A->R1 | 940byte | 0000 0000 0001 0101 | 0 | 0 | 0 |
| R1->R2 | 960byte | 0000 0000 0001 0101 | 0 | 1 | 0 |
|  | 960byte | 0000 0000 0001 0101 | 0 | 0 | 60 |
| R2->B | 960byte | 0000 0000 0001 0101 | 0 | 1 | 0 |
|  | 960byte | 0000 0000 0001 0101 | 0 | 0 | 60 |

#### 5-27

A: allocate hosts - 4096, so host partition 12 bits, network partition 20 bits

B: allocate hosts - 2048, so host partition 11 bits, network partition 21 bits

C: allocate hosts - 4096, so host partition 12 bits, network partition 20 bits

D: allocate hosts - 8192, so host partition 13 bits, network partition 19 bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| organization | first address | last address | prefix | mask |
| A | 198.16.0.0 | 198.16.15.255 | 198.16.0.0/20 | 255.255.240.0 |
| B | 198.16.16.0 | 198.16.23.255 | 198.16.16.0/21 | 255.255.248.0 |
| C | 198.16.32.0 | 198.16.47.255 | 198.16.32.0/20 | 255.255.240.0 |
| D | 198.16.64.0 | 198.16.95.255 | 198.16.64.0/19 | 255.255.224.0 |

#### 5-28

their mask: 255.255.248.0, so we need to split the third part of their IP address

|  |  |
| --- | --- |
| IP address |  |
| 57.6.96.0 | 57.6.0110 0000.0 |
| 57.6.104.0 | 57.6.0110 1000.0 |
| 57.6.112.0 | 57.6.0111 0000.0 |
| 57.6.120.0 | 57.6.0111 1000.0 |

They can be aggregated by 57.6.96.0/19.

#### 5-29

Not necessary to split up the aggregated address. We can use CIDR to send the packets with the longest matching prefix IP address. Just add block 29.18.60.0/22 to the table.

#### 5-30

(a) 135.46.0011 1111.10, longest matching prefix with 135.46.60.0/22, so transfer to Interface 1

(b) 135.46.0011 1001.14, longest matching prefix with 135.46.56.0/22, so transfer to Interface 0

(c) 135.46.0011 0100.2, doesn't match 135.46.60.0/22 or 135.46.56.0/22, so transfer to default, which is Router 2

(d) 192.53.40.7, longest matching prefix with 192.53.40.0/23, so transfer to Router 1

(e) 192.53.0011 1000.7, doesn't match with 192.53.0010 1000.0/23, so transfer to default, which is Router 2