

## 第 5 章 网络互连：网络层

**5.8** 考虑一个路由器收到一个 IP 数据报，其中 IP 头部长度为 20 字节，总长度为 1000 字节，DF 和 MF 标志都为 0，分段偏移为 0。该数据报要往一个 MTU 为 512 字节的链路转发。

- (a) 该数据报需要分成几个分段？并给出这些分段的 IP 头部中总长度、DF 和 MF 标志、分段偏移的取值
- (b) 假设收到的数据报的 DF 标志为 1，路由器会如何处理？

答：

a.  $MTU = 512$ ，最多携带  $512 - 20(\text{IP header}) = 492$  字节，但是 492 不是 8 的倍数，因此前面的分段携带的数据为小于 492 的最大一个 8 的倍数 = 488

原始数据报为 1000 字节，携带  $1000 - 20 = 980$  字节的数据，可以分为三段，分别为前面 488、488 和  $980 - 2 \times 488 = 4$  个字节。

第一个分段：总长度 =  $488 + 20 = 508$ ，DF/MF = 0/1，分段偏移 = 0

第二个分段：总长度 =  $488 + 20 = 508$ ，DF/MF = 0/1，分段偏移 =  $488/8 = 61$

第三个分段：总长度 =  $4 + 20 = 24$ ，DF/MF = 0/0，分段偏移 =  $2 \times 61 = 122$

b. 收到一个 DF=1 的数据报，路由器往前转发需要进行分段但是不许分段会丢弃该数据报，并且发送一个不许分段的 ICMP 不可达（ICMP Unreachable with don't fragment）消息给源。

**5.11** 考虑一个机构拥有 IP 地址块 101.101.176.0/22，机构内部有 4 个部门，每个部门要求支持的主机数目分别为 318、235、108、69，请给出一种 IP 地址块分配方案，采用类似于 101.101.176.0/22 的形式来描述地址块。

答： 可有多个回答，比如：

318 台主机需要用 9 个比特来表示主机部分，可以采用地址块： 101.101.176.0/23

235 台主机需要用 8 个比特来表示主机部分，可以采用地址块： 101.101.178.0/24

108 台主机需要用 7 个比特来表示主机部分，可以采用地址块： 101.101.179.0/25

69 台主机需要用 7 个比特来表示主机部分，可以采用地址块： 101.101.179.128/25

题目改动一下，使得其更加严格一些。

**5.12** 下面是一个路由器的路由表(的一部分)，请尽可能把路由表项汇集起来以使得路由表项的个数最少，注意路由表项应该按照正确的顺序排列（出现多个表项匹配时第一个匹配项为最终的匹配）。

子网	物理端口
----	------

10.25.16.0 / 24	A
10.25.17.0 / 24	B
10.25.18.0 / 24	A
10.25.19.0 / 24	A

答：

能够汇集的表项必须是同一个物理端口，18 的二进制表示为 1010，19 的二进制表示为 1011，因此可以合并为 10.25.18.0/23 A，所以最终结果为：

10.25.16.0/24 A

10.25.17.0/24 B

10.25.18.0/23 A

**5.13** 一个支持 CIDR 的路由器的路由表如下所示，假设路由器收到多个分组，其目的地地址分别为：

1. 196.94.19.135

2. 196.94.34.9

3. 94.67.145.18

4. 196.106.49.49

请问这些分组分别转发给哪个路由器？

网络/掩码	下一跳
196.80.0.0/12	A
196.94.16.0/20	B
196.96.0.0/12	C
196.104.0.0/14	D
128.0.0.0/1	E
64.0.0.0/2	F
0.0.0.0/2	G

答：

1. 转发给 B

2. 转发给 A

3. 转发给 F

4. 转发给 D

**5.17** 假设主机 10.0.0.1 访问 Internet 上 Web 服务器，其 IP 地址为 21.1.78.5，10.0.0.1 需要穿越一个 NAT 设备才能访问 Internet，请分别给出 HTTP 请求和响应在 NAT 设备内外的 IP 地址和端口号情况。假设采用 NAT，NAT 设备使用公网 IP 地址 138.76.29.7。

答： x 为内部主机选取的一个端口号，而 y 为 NAT 设备选取的一个端口号

	源 IP：源端口号	目的 IP：目的端口号
C→S: NAT 内部	10.0.0.1:x	21.1.78.5:80
C→S: NAT 外部	138.76.29.7:y	21.1.78.5:80
S→C: NAT 外部	21.1.78.5:80	138.76.29.7:y
S→C: NAT 内部	21.1.78.5:80	10.0.0.1:x

# 第 5 章 网络互连: IPv6 和路由协议

- 5.39 请对下列 IPv6 地址进行压缩：
- a) 0000:0000:0F53:6382:AB00:67DB:BB27:7332
  - b) 2819:00AF:0000:0000:0000:0035:0CB2:B271

答：

a. ::F53:6382:AB00:67DB:BB27:7332

b. 2819:AF::35:CB2:B271

5.19 在图 5.48 中，计算 A 到各个节点的最短路径树。

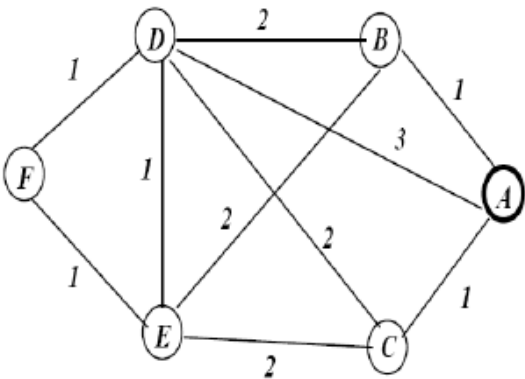


图 5.48

答：（注意有多个正确答案）

S 为当前规约的节点集，D(v)表示算法求得的当前从源 A 到 v 的路径花费，P(v)表示当前从源到 v 的路径中 v 的上一跳节点

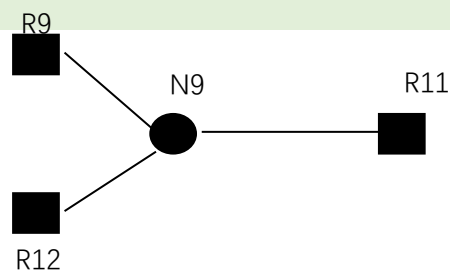
步骤	S	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
0（初始）	A	1,A	1,A	Inf	Inf	Inf
1	AB		1,A	3,B	3,B	Inf
2	ABC			3,B	3,B	Inf
3	ABCD				3,B	4,D
4	ABCDE					4,D
5	ABCDEF					
6						

A 出发到其他节点的最短路径和路径花费如下

- AB: 1
- AC: 1
- ABD:3
- ABE:3
- ABDF:4

5.22 在 OSPF 中，一个广播网络的状态信息是由连在该网络中的选取路由器负责传播的。考虑教材中图 5.27 的配置，网络 N9 的链路状态消息中包含什么？**用一个拓扑图或表格形式来表示。**

**答：**  
**网络的链路状态信息给出了该网络所连接的路由器的信息，即 N9 上连接了路由器 R9, R11, R12。**  
**如图所示：**



5.24 有一个网络如图 5.51 所示，图中的节点为运行距离向量路由算法的路由器，边上的权值为链路的距离。对于下面两种情况，给出算法达到稳定状态后节点 C 的路由表：

(1) 不采用水平分割时节点 C 的路由表；

(2) 采用带反向抑制的水平分割时节点 C 的路由表。

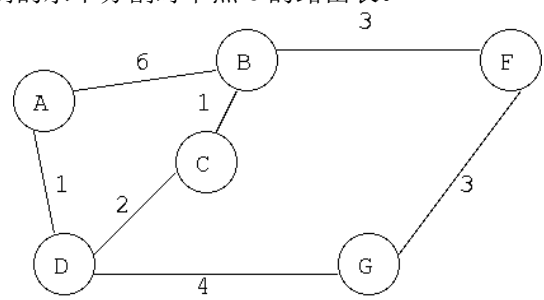


图 5.51

**答：**  
**(1) 不采用水平分割时节点 C 收到来自于 B 和 D 的路由更新消息，然后花费更小的路径作为当前的路由表，如下所示（用下划线字描述，下同）：**

目的地	通过 B	通过 D
A	5	<u>3</u>
B	<u>1</u>	5
D	4	<u>2</u>
F	<u>4</u>	8
G	7	<u>6</u>

**(2) 采用水平分割算法，节点 C 的路由表如下所示：**

目的地	通过 B	通过 D
A	$\infty$	<u>3</u>
B	<u>1</u>	$\infty$
D	$\infty$	<u>2</u>



7.1									

答案：

时刻	节点 1, 到 5		节点 2, 到 5		节点 3, 到 5		节点 4, 到 5		
	通过 2	通过 4	通过 1	通过 3	通过 2	通过 4	通过 1	通过 3	通过 5
0.1	<u>3+6=9</u>	7+3=10	inf	<u>2+4=6</u>	inf	<u>1+3=4</u>	7+9=16	inf	<u>3</u>
1.0	<u>3+6=9</u>	7+3=10	inf	<u>2+4=6</u>	inf	<u>1+3=4</u>	7+9=16	inf	<u>10</u>
1.1	<u>3+6=9</u>	7+10=17	inf	<u>2+4=6</u>	inf	<u>1+10=11</u>	7+9=16	inf	<u>10</u>
2.1	<u>3+6=9</u>	7+10=17	inf	<u>2+11=13</u>	inf	<u>1+10=11</u>	7+9=16	inf	<u>10</u>
3.1	<u>3+13=16</u>	7+10=17	inf	<u>2+11=13</u>	inf	<u>1+10=11</u>	7+9=16	inf	<u>10</u>
4.1	<u>3+13=16</u>	7+10=17	inf	<u>2+11=13</u>	inf	<u>1+10=11</u>	7+16=23	inf	<u>10</u>
5.1									

t=3.1 时刻进入稳定状态，t=4.1 确认稳定状态。确认处于稳定状态指的是上一轮的转发表（路由表）与当前轮的转发表一样，即到目的地的路径花费一样。