第五单元 网络层
-路由协议

本节内容

- 有类网中的IP路由选择
- 无类网中的IP路由选择

IP路由选择

- 路由协议
- 自治系统

一些概念

- 距离向量算法
- RIP协议
- RIP协议的问题

RIP协议

- 链路状态算法
- OSFP协议
- OSPF协议的特点

OSPF协议

- BGP协议

BGP协议

- IP多播
- 逆向路径多播
- IGMP协议
- 协议无关多播-稀疏模式

IP多播协议

IP路由选择-有类网

R60的路由表

目的网络	下一跳	接口
12	R12	1
166.132	R34	2
195.42.21	R41	3
202.118.201	-	2
default	R5	4

默认路由

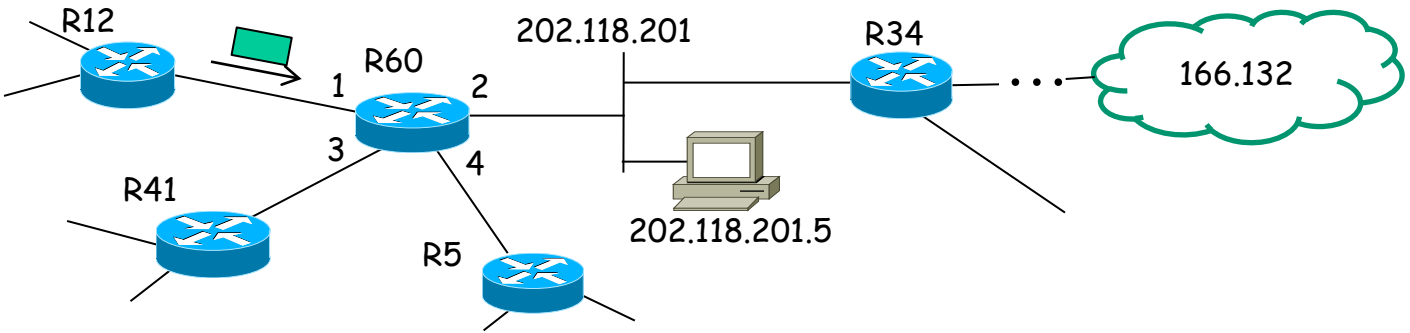
IP分组的目的地址：

166.132.1.1

202.118.201.5

211.1.1.1

如何转发？



- 如果目的网络为直连网络，则下一跳(next hop)为空。
- 路由表(routing table)有时也被称为转发表(forwarding table)。

有类网的路由选择算法：

利用IP数据报中的目的地址得到网络号，然后用它查询路由表：

(1) 如果查询的结果为直连网，

则直接把数据报从查出的接口转发到目的主机。

(2) 否则，如果查询得到下一跳(路由器)，则把数据报转发给下一跳，

如果没有查到任何匹配项，则把数据报转发给默认路由器，

如果没有设置默认路由，则丢弃该数据报。

IP路由选择-无类网

128 10000000
131 10000011

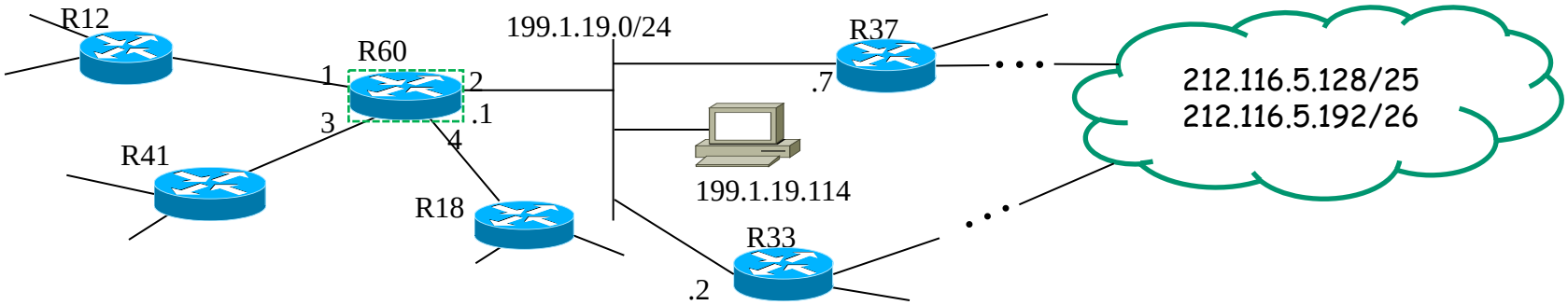
R60的路由表

子网号	子网掩码	下一跳	接口
212.116.5.128	255.255.255.128	R37	2
199.1.19.0	255.255.255.0	-	2
212.116.5.192	255.255.255.192	R33	2
0.0.0.0	0.0.0.0	R18	4

default route

目的地址：
212.116.5.131
199.1.19.114
212.116.5.221
66.5.10.100
如何转发？

- **匹配方法:** 目的地址 and 子网掩码 = 子网号?
- **最长匹配原则**(The longest match rule): 当有多条路由都匹配时选择子网掩码最长的路由。

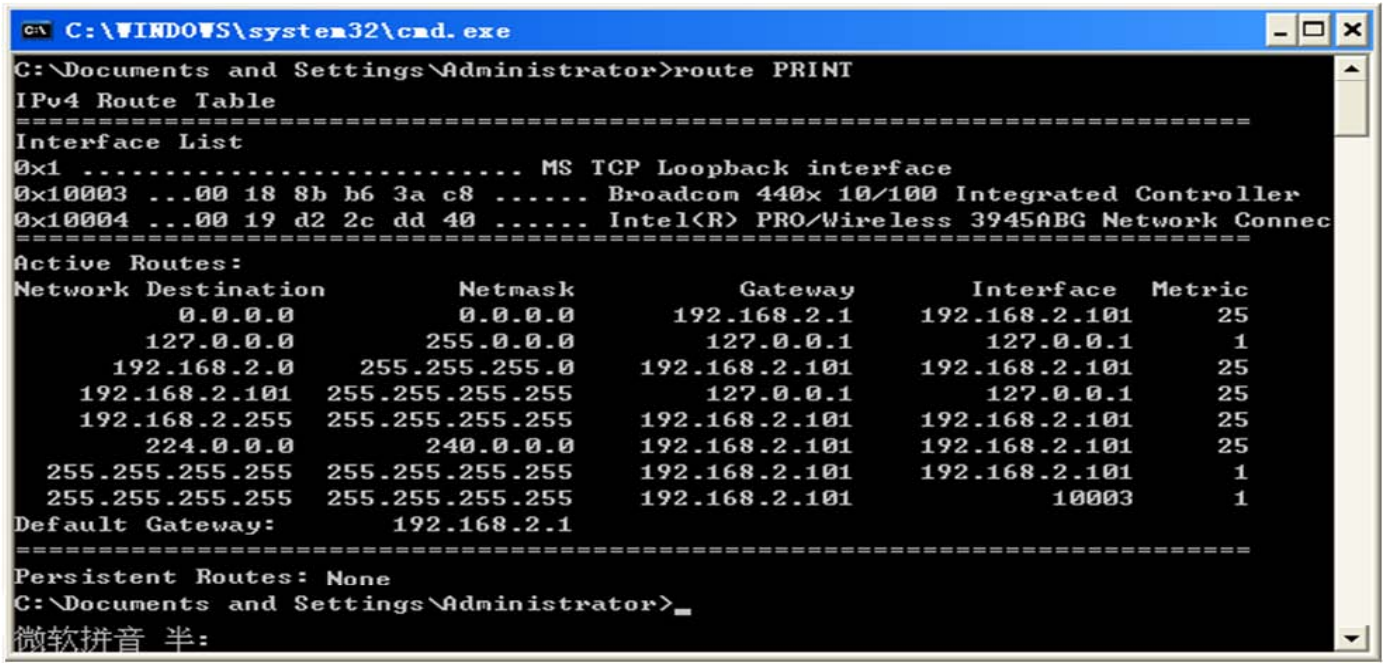


无类网的路由选择算法：

用收到的IP数据报中的目的地址查询路由表:

对于路由表的每个表项<子网号, 子网掩码, 下一跳>:
  如果 目的地址 and 子网掩码 =子网号 （匹配）
    如果下一跳是直连网
      则直接把数据报发往目的主机
    否则
      把数据报发往下一跳
如果没有任何匹配项，则丢弃该数据报

Windows 7 的路由表



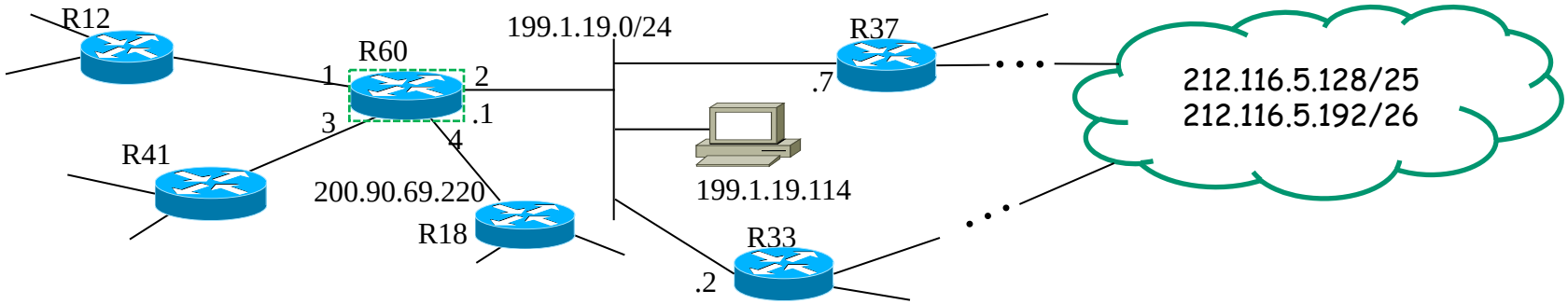
如果存在多个匹配项并且子网掩码长度相同，选择开销(metric)更小的。如果开销也相同，则它们一起被选中。

数据报转发

R60的路由表				
子网号	子网掩码	下一跳		接口
212.116.5.128	255.255.255.128	199.1.19.7	(R37)	2
199.1.19.0	255.255.255.0	199.1.19.1	(-)	2
212.116.5.192	255.255.255.192	199.1.19.2	(R33)	2
0.0.0.0	0.0.0.0	200.90.69.220	(R18)	4

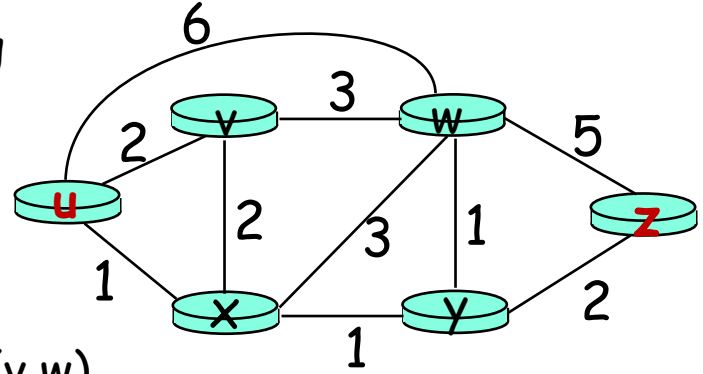
default route

目的地址：
212.116.5.131
199.1.19.114
212.116.5.221
66.5.10.100
如何转发？



路由协议

- 路由表(routing tables)可以由管理员手工建立，也可以由**路由协议**(routing protocols)自动建立，所建的路由分别称为**静态路由**和**动态路由**。
- 建立动态路由所用的算法称为路由算法(Routing algorithm)。路由算法一般采用最短路径算法，例如：距离向量算法和链路状态算法。



图G = (N,E)

- 右图是一个从网络转化而来的图：

节点集合N = {routers} = { u, v, w, x, y, z }

边集合E = {links} = { (u,v), (u,x), (u,w), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

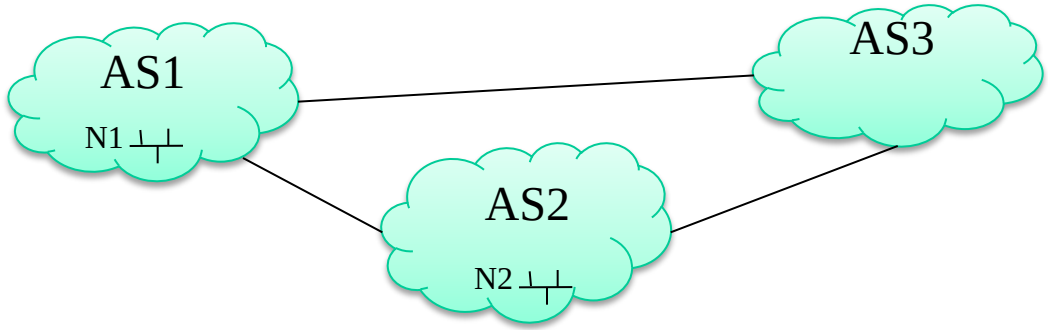
$c(x_i,x_j)$为链路(x_i,x_j)的开销(cost)，例如，c(w,z) = 5,c(u,z)=∞

路径 $(x_1, x_2, x_3,\ldots, x_p)$的开销 = $c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

例如：路径(u,v,w,z)的开销=c(u,v)+c(v,w)+c(w,z)=2+3+5=10

问题: 结点u和z之间最短路径(具有最小开销的路径)是什么？

自治系统



- 规模庞大的因特网实际上是由很多自治系统构成的。每个自治系统(autonomous systems, AS)都是在同一个机构管理之下的。
- 用于在AS内(Intra-AS)建立动态路由的路由协议称为**内部网关协议**(Interior Gateway Protocols，IGP)。例如， RIP协议和OSPF协议。用于在AS之间(Inter-AS)建立动态路由的路由协议称为**外部网关协议**(Exterior Gateway Protocol，EGP)。例如，BGP协议。
- 在一个AS中可以使用多个IGP协议。AS中使用同一个IGP协议的连通区域称为一个路由选择域(routing domain)。
- 每个中转AS(transit AS)都需要由ICANN分配一个AS号(1~65535)，而末端AS(stub AS)不需要AS号。一个末端AS可以连入多个中转AS，称为多穴AS(multi-homed AS)。

距离向量算法

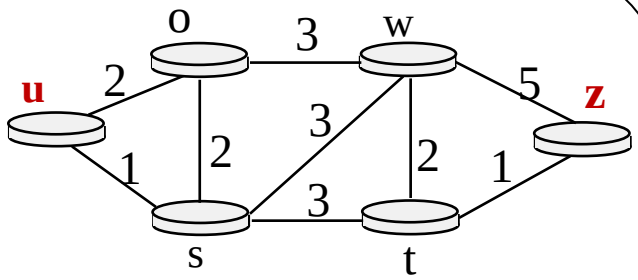
距离向量算法(Distance vector algorithm)是一种最短路径算法，也称为Bellman-Ford算法或Ford-Fulkerson算法。其基本思想如下：

- 初始时或距离向量改变时，每个节点把它的距离向量发送给所有邻居。
- 当一个节点(x)收到邻居(n)的距离向量时，保存它，并根据下面的Bellman-Ford方程更新它自己的距离向量$[D_x(y): y \in N]$:

$$D_x(y) = \min_n\{c(x, n) + D_n(y)\}$$

其中，n为邻居，$D_x(y)$ 为x到y的当前最短路径开销(距离)。

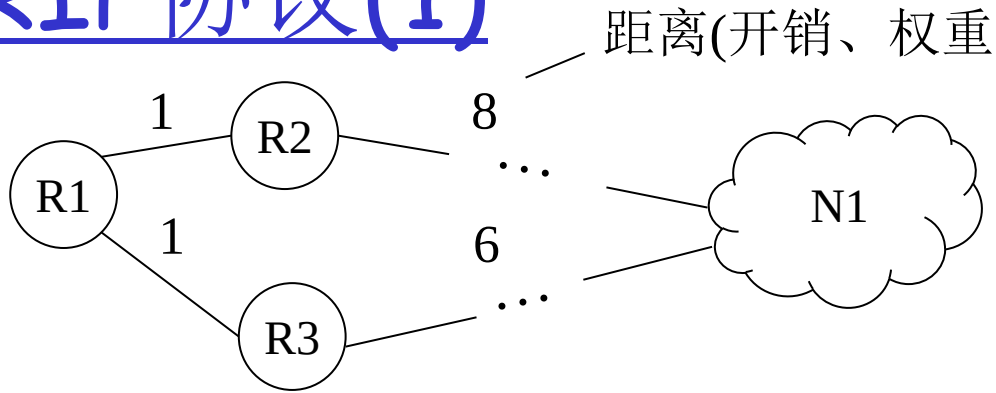
- 在自然条件下，$D_x(y)$会收敛到真正的从x到y的最短路径开销$d_x(y)$ 。



u的初始**距离向量**:

$$\begin{pmatrix} D_u(u) \\ D_u(o) \\ D_u(w) \\ D_u(s) \\ D_u(t) \\ D_u(z) \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ \infty \\ 1 \\ \infty \\ \infty \end{pmatrix}$$

RIP协议(1)



R1的路由表

目的地	距离	下一跳
N1	7	R3

- RIP协议(Routing Information Protocol)是一种采用距离向量算法的路由协议，它利用邻居的路由表得到自己的路由表。
- 到目的网络的距离以跳为单位。最大距离为15。距离16表示无穷大，即目的网络不可达。
- 初始时每个路由器只有到直连网的路由，它们的距离为1。
- 每隔30秒路由器把它的路由表发送给邻居。具体实现时会错开发送给每个邻居，30秒的时间也会随机变化一点。

RIP协议(2)

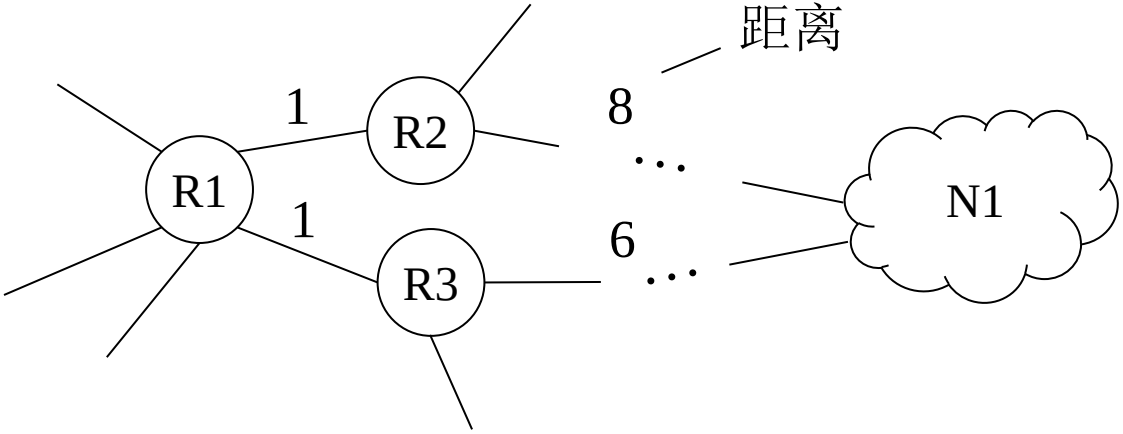
- 当一个路由器收到邻居发来的路由表(update packet)，它将用其中的路由更新它的路由表 <目的网络, 距离,下一跳>:

  (1) 它把收到的每条路由的距离都加上它到邻居的开销(默认为1) 。

  (2) 利用上面修改后的路由更新它的路由表:

  - 对于路由表中不存在的路由，直接把它加入路由表，下一跳设为邻居。
  - 对于路由表中存在而且距离更小的路由，则更新距离和下一跳。
  - 对于路由表中存在而且下一跳就是该邻居的路由，必须修改距离（即使距离比以前更大）。
  - 只要路由存在，就必须重置失效定时器。

- RIP路由表的每一项都有TTL(Time-To-Live)，用失效定时器(invalid timer)计时，超时则让该路由失效（距离改为无穷大）。



R1的路由表

目的地	距离	下一跳
N1	7	R3
N2	5	R2
N3	2	R4
N4	1	-

R2的路由表

目的地	距离	下一跳
N1	4	R5
N2	6	R6
N3	2	R4
N4	2	R1
N5	3	R6

R1收到R2的路由表之后

目的地	距离	下一跳

RIPv1的数据报格式

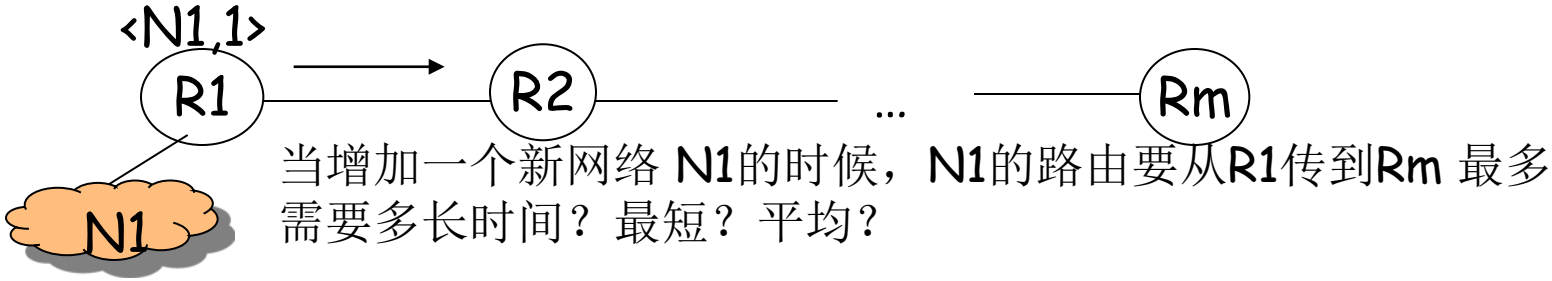
http://tools.ietf.org/html/rfc1058

- RIPv1数据包用UDP数据报封装 (端口号为520)，并且采用广播方式发送给邻居。RIPv1只能发布有类网，因此，对于存在非邻接子网的情况，很可能会发生错误。

- ✓ 如果请求的网络地址为0.0.0.0，则用整个路由表进行响应，否则，用该网络地址的距离进行响应。
- ✓ 每30秒和触发更新都是发送响应分组。如果项目超过25项，则可以发送多个响应分组。
- ✓ RIPv2可以用于无类网。每条路由增加了子网掩码和下一跳，还增加了认证方式和多播方式(224.0.0.9)。

RIP协议中存在的问题

- 慢收敛问题(Slow Convergence)



当增加一个新网络 N1的时候，N1的路由要从R1传到Rm 最多需要多长时间？最短？平均？

- 计数到无穷问题(Count to Infinity)

当N1失效时，R1把N1路由的距离改为16后会发生什么？

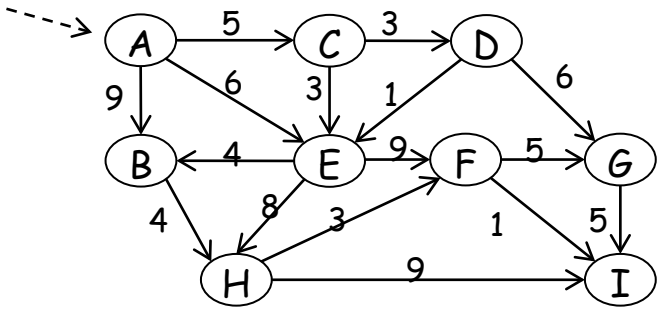
- 水平分割技术(split horizon):
  从一个接口学来的路由不会从该接口发回去。
- 触发更新

链路状态算法(link state algorithm)

(1) 每个节点把自己的链路状态扩散给其它节点。利用这些链路状态就可以把图构造出来。

(2) 每个节点利用单源最短路径算法(Dijkstra最短路经算法)求出到所有其它节点的最短路径。

(3) 每个节点利用这些最短路径上的下一个节点作为下一跳就得到它的转发表。



节点A的链路状态：c(A,B)=9
c(A,C)=5
c(A,E)=6

节点A的转发表

目的地	链路	距离
B	(A,B)	9
C	(A,C)	5
D	(A,C)	8
E	(A,E)	6
F	(A,B)	14
G	(A,C)	14
H	(A,B)	13
I	(A,E)	16

OSFP协议(1)

http://tools.ietf.org/html/rfc2328

- **OSPF** 协议(Open Shortest Path First)采用链路状态算法动态建立路由表，它是在大型企业中使用最广泛的内部网关协议。
- OSPF路由器（运行OSPF协议的路由器）：

  (1) 每10秒向邻居发Hello分组。从接口收到Hello分组就可以得知接口的链路状态；

  (2) 每30分钟或链路状态变化时用接口的链路状态形成**链路状态通告**(Link State Advertisement，LSA)，并把它扩散给AS中的所有路由器；

  (3) 把扩散来的LSA放入链路状态数据库中；

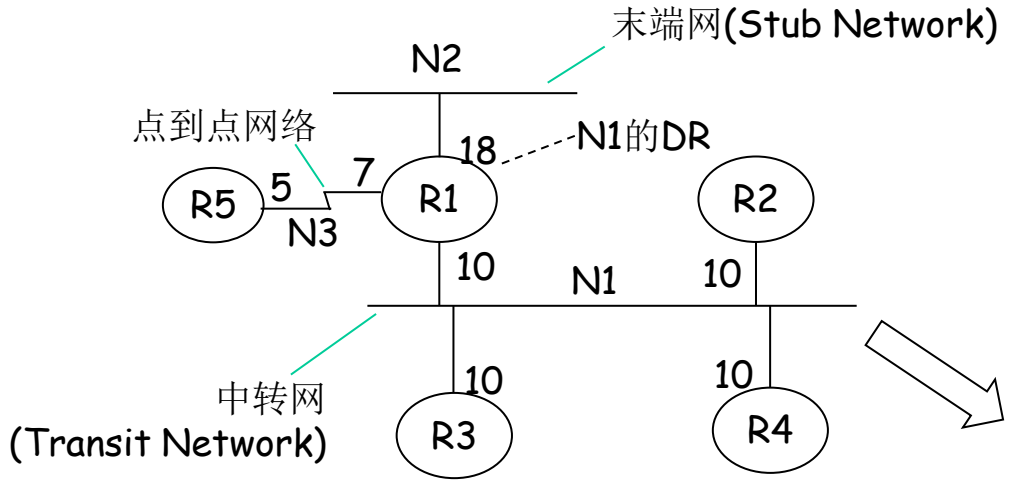
  (4) 利用链路状态数据库中的LSA建立整个AS的拓扑结构图；

  (5) 利用Dijkstra最短路径算法求出该路由器到AS中所有网络的最短路径；

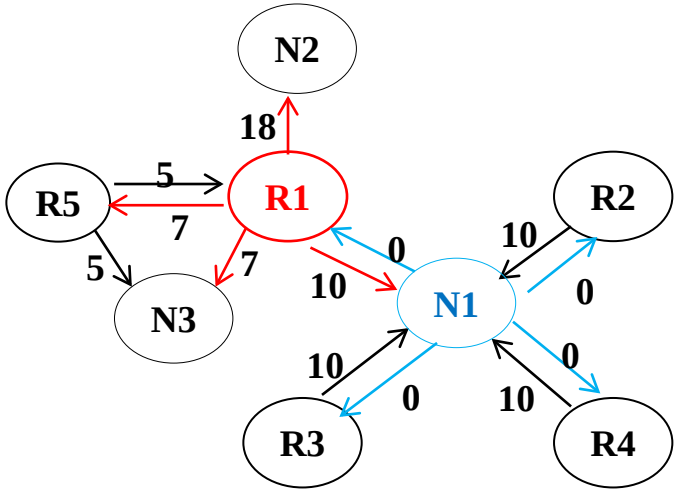
  (6) 利用这些最短路径上的下一个路由器作为下一跳建立这些网络的路由表。

  问题：如何形成LSA？如何得到AS的拓扑结构图？扩散时如何防止形成回路？

OSFP协议(2)



- N1和N2是多路访问网络，例如：以太网。
- N3为点到点网络，例如：ppp。
- 用路由器ID(RID)区分每个路由器，这里是用R1、R2，...表示。
- 每个路由器都要形成和扩散一个Router LSA。



R1's Router LSA:

	R1 (From)
N1	10
N2	18
R5	7
N3	7

N1's Network LSA:

	N1(From)
R1	0
R2	0
R3	0
R4	0

- 对于每个中转网，要选举一个直连路由器作为其指定路由器 (designated router, DR)，由它负责收集和扩散Network LSA。
- 如果图中点到点网络没有配置IP地址，则不要节点N3。

- 每个LSA包含发通告路由器ID、LSA类型和序号三个字段。新旧LSA，防止出现回路。
- 链路的开销：1000/带宽(Mbps)。
- 链路状态数据库同步

N1's Network LSA:

	N1(From)
R1	0
R2	0
R3	0
R4	0

R1's Router LSA:

	R1 (From)
N1	10
N2	18
R5	7
N3	7

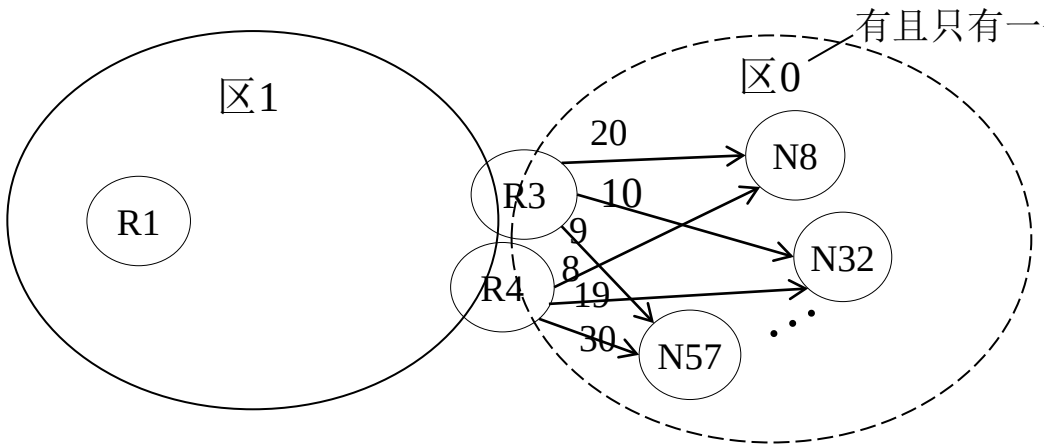
用链路状态数据库中的LSA形成图

	R1	R2	R3	R4	R5	N1	(From)
R1							0
R2							0
R3							0
R4							0
R5	7						
N1	10						
N2	18						
N3	7						

- R2~R5 的Router LSAs也将被加入到链路状态数据库中
- 用链路状态数据库形成AS拓扑结构图的邻接矩阵。

OSPF协议(3)

- OSPF协议是分区的，使用OSPF协议的AS中至少要有主干区，其它区必须与主干区相邻。同时属于多个区的路由器称为区边界路由器(Area Border Router，ABR)，ABR需要分别建立这些区的链路状态数据库，并独立计算最短路径。其它位于区内的路由器称为内部路由器。
- ABR负责把主干区中已知网络的链路状态注入到另一个区或相反。一个区的拓扑结构图要把所有区外网络作为节点包含进来，这样就可以利用最短路径算法得到包含了AS中所有网络的路由表。



R3's Network Summary LSA:

	R3 (From)
N8	20
N32	10
	...
N57	9

OSPF协议的特点

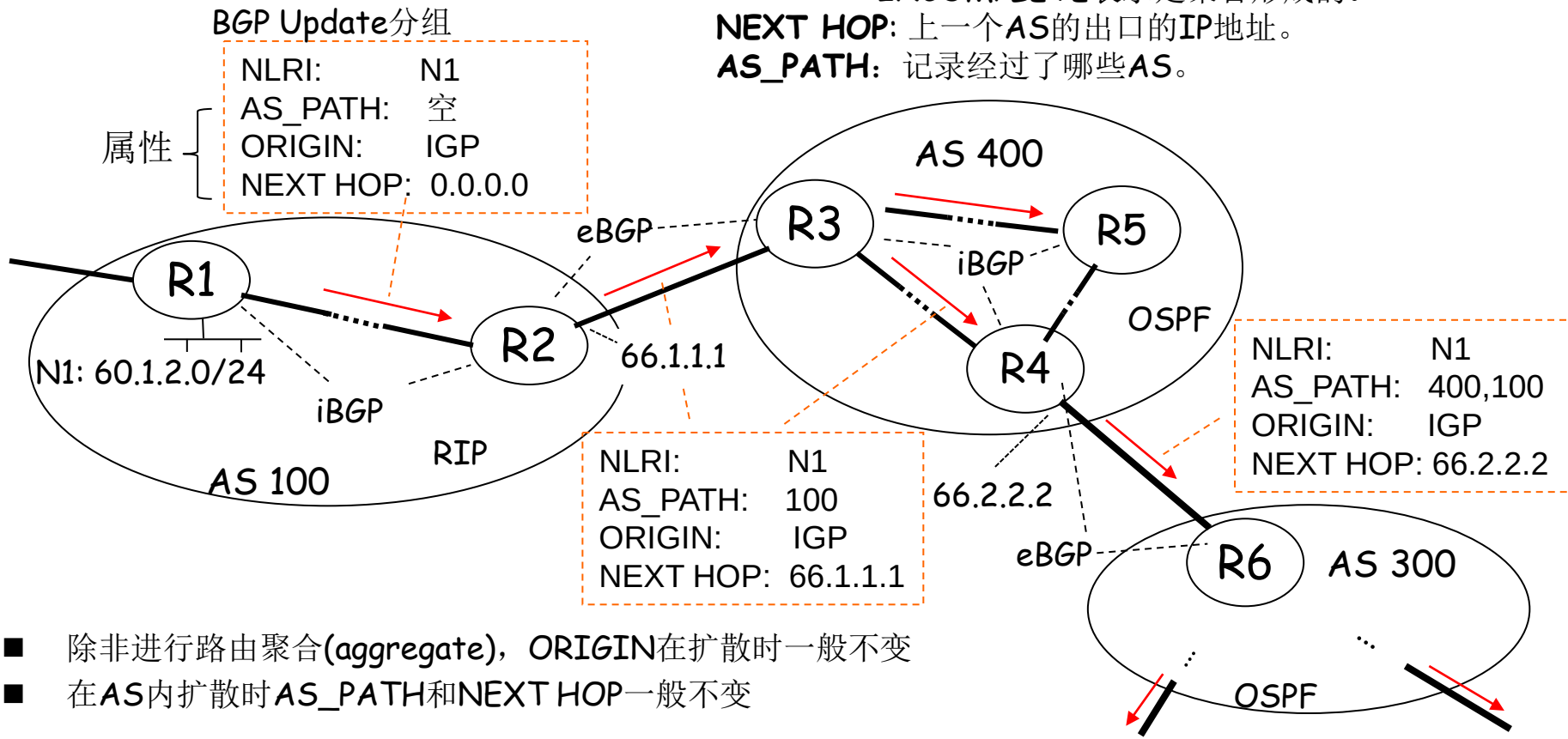
- 所有的OSPF消息都要认证 (防止恶意入侵)。
- 路由表中允许多个相同开销的路经存在(RIP只允许一条路经)，可以实现负载均衡。
- 对于每条链路，允许同时有多个(TOS)开销。
- 多播OSPF (MOSPF)使用与OSPF相同的链路状态数据库
- 在大型路由选择域中OSPF可以分区。
- 比RIP收敛快而且更安静。
- 实现起来更复杂，需要更多的计算开销。

BGP协议(1)

- BGP协议(Border Gateway Protocol)是基于路径向量的外部网关协议。
- BGP协议采用可靠扩散(reliable flooding)的方法把AS内的网络号(或网络前缀)发布到AS外，并传遍整个因特网。这里的网络号也称为**网络层可达信息**(Network Layer Reachablility Information，NLRI)。
- BGP路由器只在建立网络时才扩散包含该网络NLRI的更新分组。一般来说，只有管理员指定并且在IGP路由表中有效的网络才会被BGP协议扩散出去。在网络失效时，BGP协议会扩散撤销该网络的更新分组。
- BGP路由器采用更新分组通过相邻关系(邻居)扩散NLRI。AS内和AS之间的两个BGP路由器之间建立的相邻关系分别称为**iBGP(interior BGP)和eBGP(exterior BGP)相邻关系**。
- BGP路由器之间是通过TCP连接(端口号为179)建立相邻关系的,与谁建立相邻关系是由AS管理员指定的。

BGP协议(2)

ORIGIN：NLRI的起源。IGP表示是由管理员加入的，INCOMPLETE表示是聚合形成的。
NEXT HOP：上一个AS的出口的IP地址。
AS_PATH：记录经过了哪些AS。

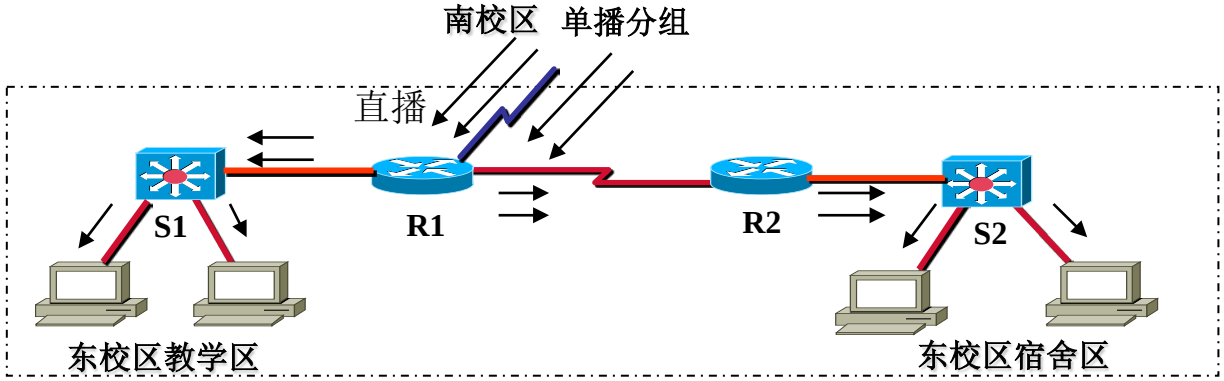


- 除非进行路由聚合(aggregate)，ORIGIN在扩散时一般不变
- 在AS内扩散时AS_PATH和NEXT HOP一般不变

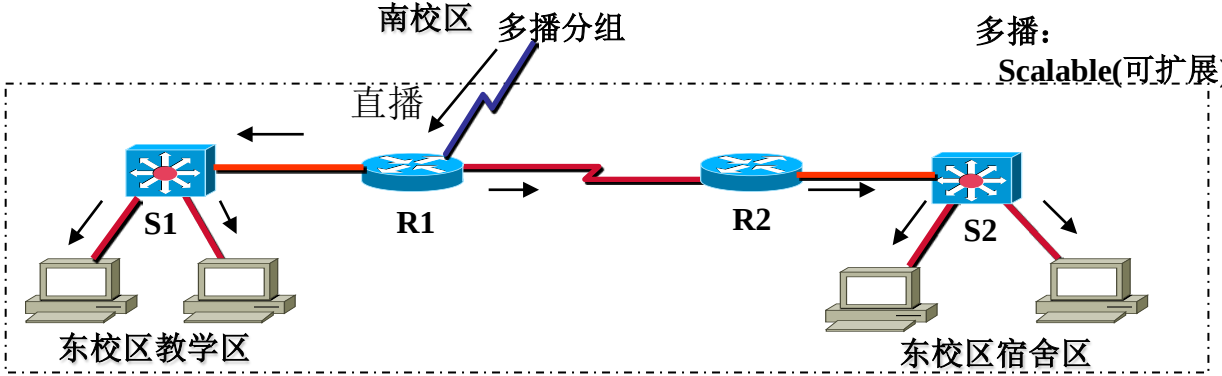
BGP协议(3)

- 为了防止NLRI在AS之间扩散时形成回路，BGP路由器会丢弃所收到的AS_PATH中包含当前AS号的NLRI。
- 为了防止NLRI在AS内部扩散时形成回路，BGP路由器不会把从iBGP邻居收到的NLRI转发给iBGP邻居。
- 为了减少路由数量，BGP路由器会聚合若干NLRI网络形成一个新的NLRI。
- 如果从多条路径收到同一个NLRI，在默认情况下选择AS-PATH最小的路径。
- BGP路由器根据NLRI的属性NEXT HOP查询IGP路由表得到匹配项的NEXT HOP，就可以形成BGP路由。如果查不到匹配项，则丢弃该NLRI。
- 如果设置了IGP同步且IGP路由表中没有该NLRI的匹配项，该NLRI不能转发给eBGP邻居。

IP多播



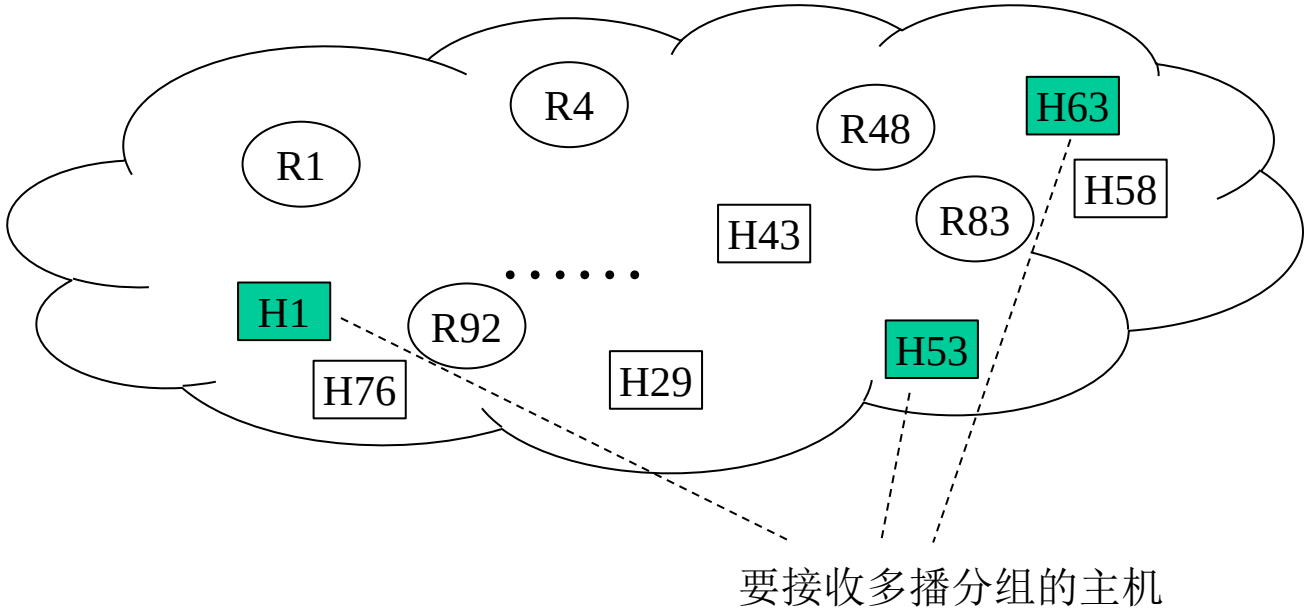
- 单播：
  浪费带宽
  增加CPU负担
  扩展性差



多播：
Scalable(可扩展)

- 多播地址为D类地址：
  224.0.0.0~239.255.255.255
- 对于点到点网络，直接封装成帧。对于以太网，用IP多播地址的低23位替换地址01-00-5E-00-00-00的低23位得到多播MAC地址，然后封装成帧。
- 多播也称为组播

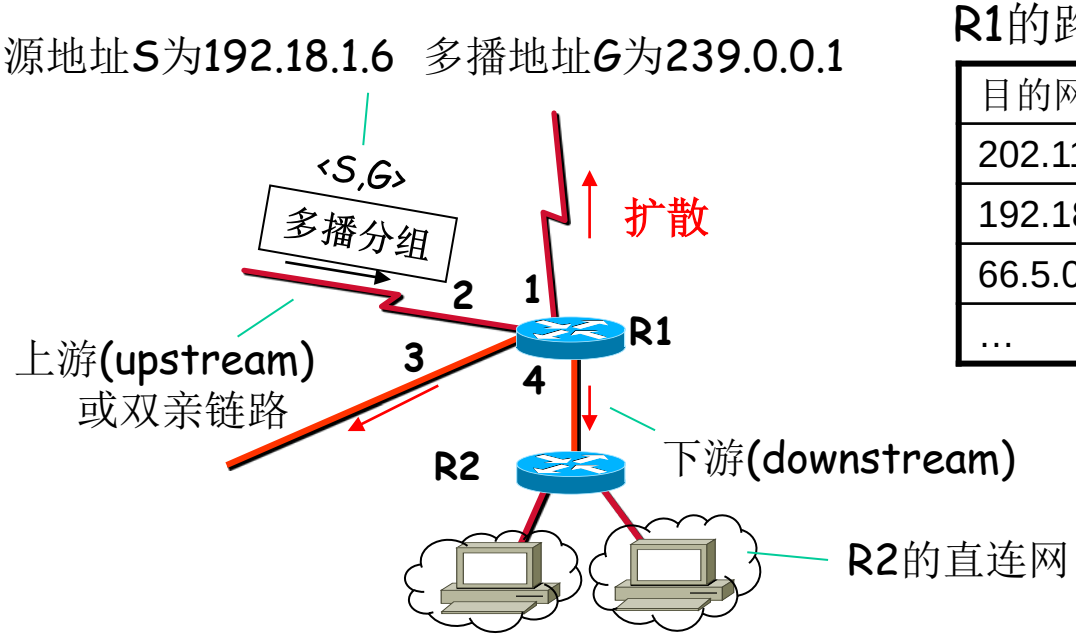
- 路由器怎么知道哪里有要接收多播分组的主机？如何转发？



逆向路径多播
(Reserve Path Multicasting)

- 如果每台主机都要接收多播分组，则可以采用扩散的方法，但是，简单扩散会产生回路，有什么解决方法呢？逆向路径广播。

- **逆向路径广播**规定：当一个路由器收到一个源地址为S发往组G的多播分组<S,G>时，仅当接收该分组的接口位于从该路由器到源主机S的最短路径上时，该路由器才扩散(flooding)该分组。

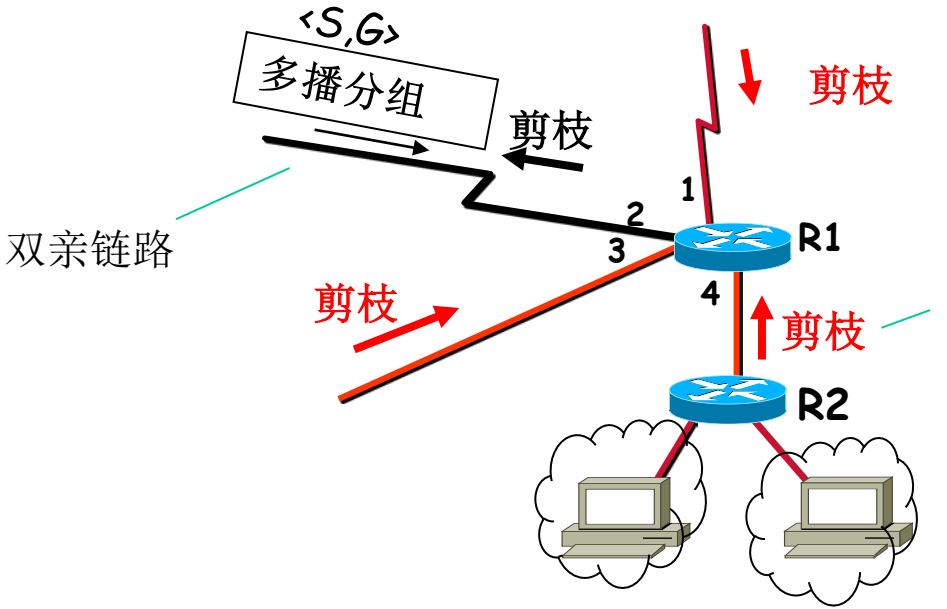


R1的路由表

目的网络号	接口
202.116.164.0/24	1
192.18.1.0/24	2
66.5.0.0/16	3
…	

- 多路访问网络要选举指定路由器进行扩散。

- 对于基于一个源地址的多播流，如果路由器的所有下游接口均没有该组成员或已被剪枝，则它通过其双亲链路向上发送剪枝消息 (Prune Message)。路由器不会把多播分组从剪枝接口转发出去。

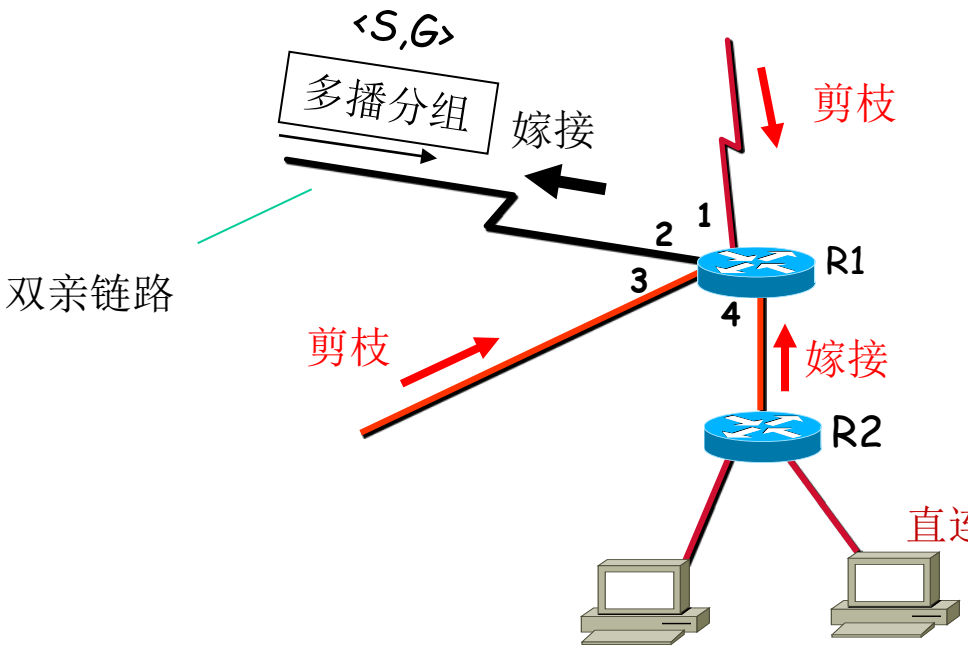


对于每个多播地址G，每个源形成一棵树，即源特定组播树(source-specific multicast tree)。

下游全部被剪枝或无组成员(直连网)

直连网无组成员

如果被剪枝的网络中新增了该组成员，怎么办？



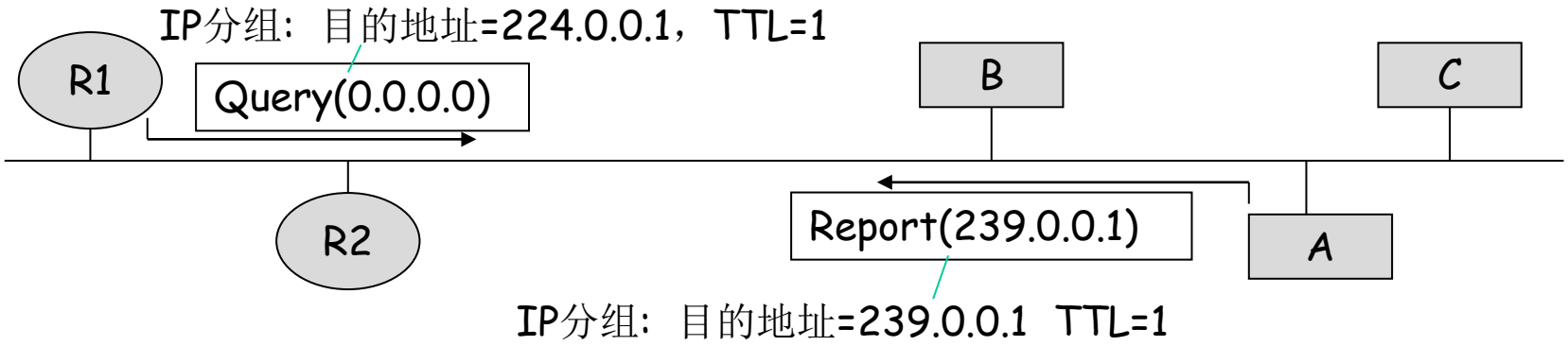
- 通过嫁接消息(Graft Message)逐级向上通知直到某个未被剪枝的接口或者根路由器。
- 为了防止嫁接消息丢失引起转发受阻，路由器会定期取消所有剪枝。

上面通过剪枝就实现了**逆向路径多播**，它适用于整个网络中有很多组成员的情况。 距离向量多播路由协议 (DVMRP)和稠密模式下的协议无关多播协议 (PIM-DM) 都采用了逆向路径多播方法。

DVMRP	Distance Vector Multicast Routing Protocol	利用RIP协议
PIM-DM	Protocol Independent Multicast-Dense Mode	

IGMP协议

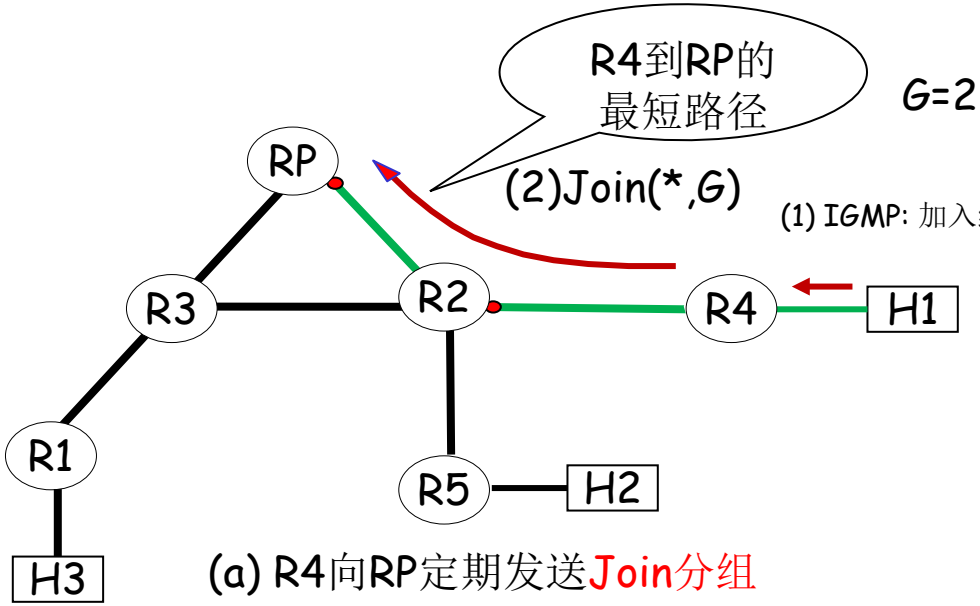
- IGMP协议(Internet Group Management Protocol)用于路由器查询与它直连的网络上是否存在组成员。
- 下图是IGMPv1的工作原理图。IGMPv1协议只能对某个接口查询所有组，如果三次查询在十秒内都没有收到响应报告，则认为该接口没有任何组成员。
- IGMPv2协议可以直接针对某个组进行查询，而且主机加入组和离开组都要发通告。



IP头部(协议号=2)	IGMP分组

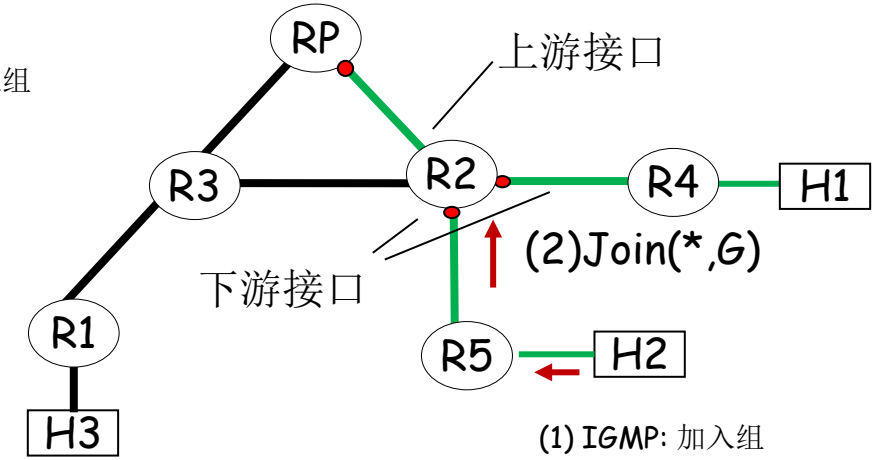
协议无关多播-稀疏模式 (PIM-Sparse Mode)

- 如果一个网络中组成员分布很密集，大部分直连网都有，我们称为稠密模式，否则，称为稀疏模式。在稠密模式下，扩散是一种效率很高的方法，通过剪枝辅助一下可以进一步提高效率。但是在稀疏模式下这会造成带宽极大的浪费，在稀疏模式下需要更加精准的转发。
- 在稠密模式下我们讲了逆向路径多播算法，这个算法被DVMRP和PIM-DM这两种多播路由协议所用，DVMRP协议是在RIP协议的基础上设计的，而PIM-DM协议是与使用什么内部网关协议无关的。
- PIM-SM协议是稀疏模式下的协议无关多播协议，没有采用扩散的方法，而是采用了先确定转发路由然后进行转发的算法。



(a) R4向RP定期发送Join分组

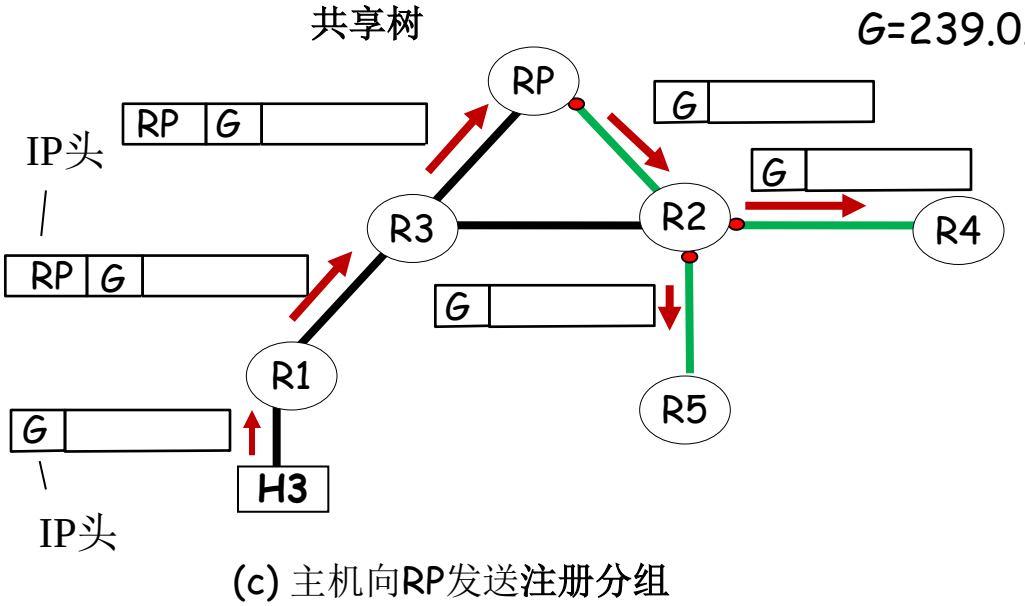
(*,G):任意源到组G



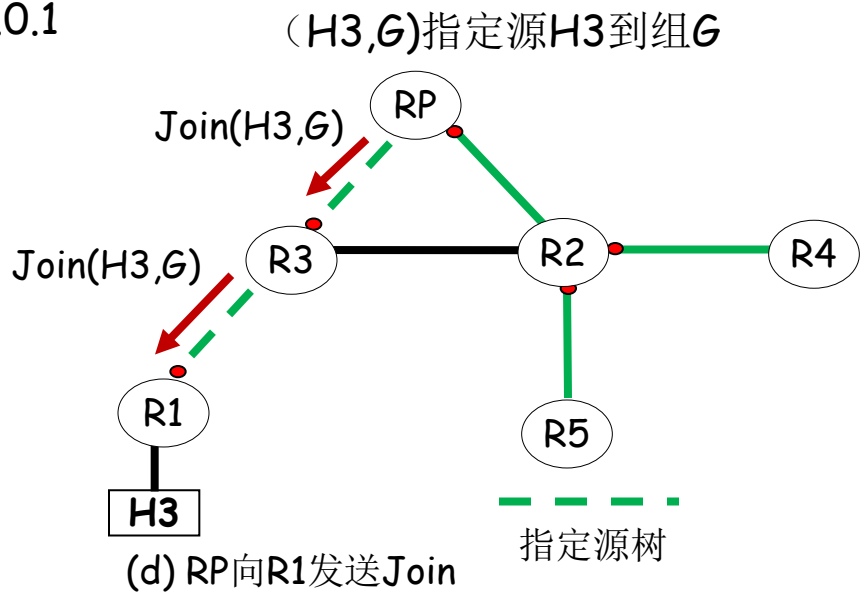
(b) R5定期向RP发送Join分组。R2标明接口后停止继续转发。

RP = 汇集点 (Rendezvous point)

共享源树(Shared tree)

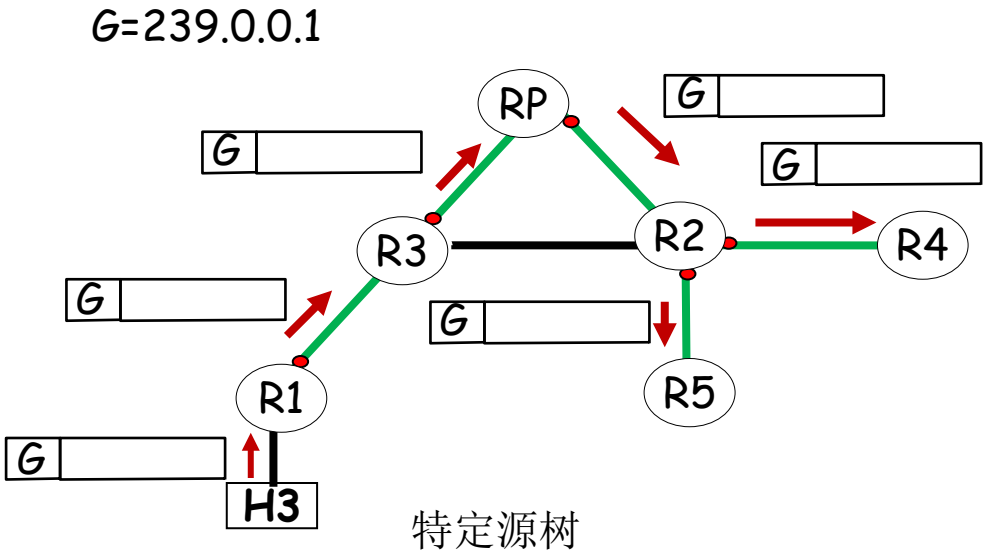


(c) 主机向RP发送**注册分组**



(d) RP向R1发送Join

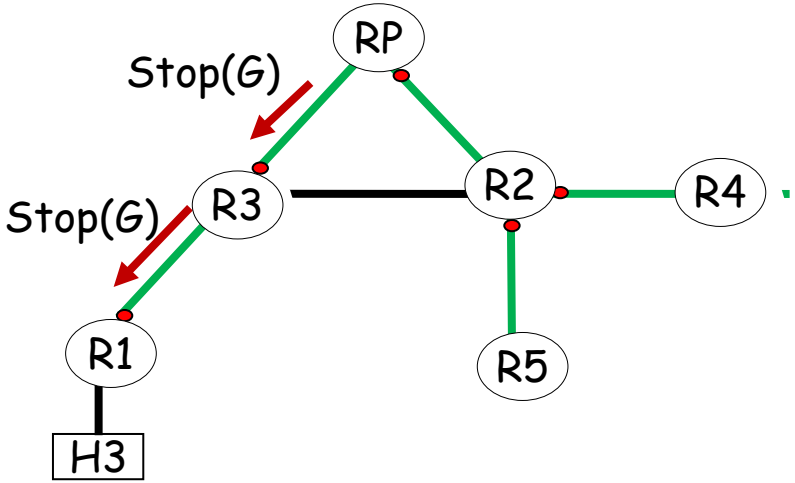
- 源站点通过注册消息进行封装后把多播分组发送给RP，RP解封装后在把该多播分组转发出去。
- 多路访问网络选择IP最大的路由器为指定路由器，用来转发多播分组。
- 要传送的数据量较大时，采用(d)(e)(f)进行优化。



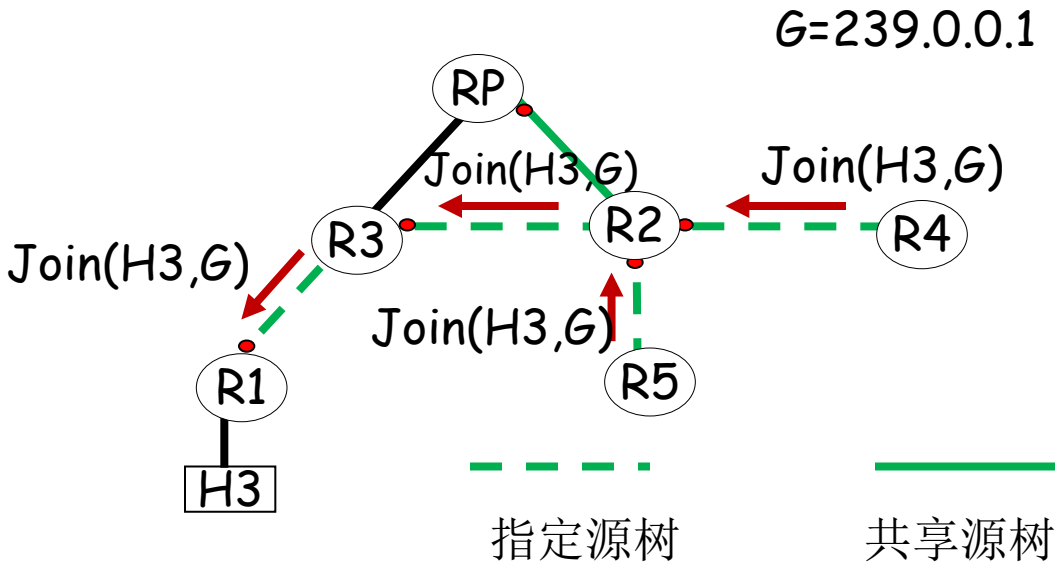
特定源树

Source-specific tree for source R1

(e) 主机直接向RP向发送多播分组；同时还要发送注册消息。



(f) RP向R1发送注册停止消息。R1收到该消息后将停止发送注册消息。



(g) 还可以做进一步改进，在收到多播分组之后叶子路由器就可以向源主机发送Join分组，建立不通过RP的指定源多播树。

总结

- 有类网中的IP路由选择
- 无类网中的IP路由选择

IP路由选择

- 路由协议
- 自治系统

一些概念

- 距离向量算法
- RIP协议
- RIP协议的问题

RIP协议

- 链路状态算法
- OSFP协议
- OSPF协议的特点

OSPF协议

- BGP协议

BGP协议

- IP多播
- 逆向路径多播
- IGMP协议
- 协议无关多播-稀疏模式

IP多播协议