第六章 网络层：路由协议

崔勇

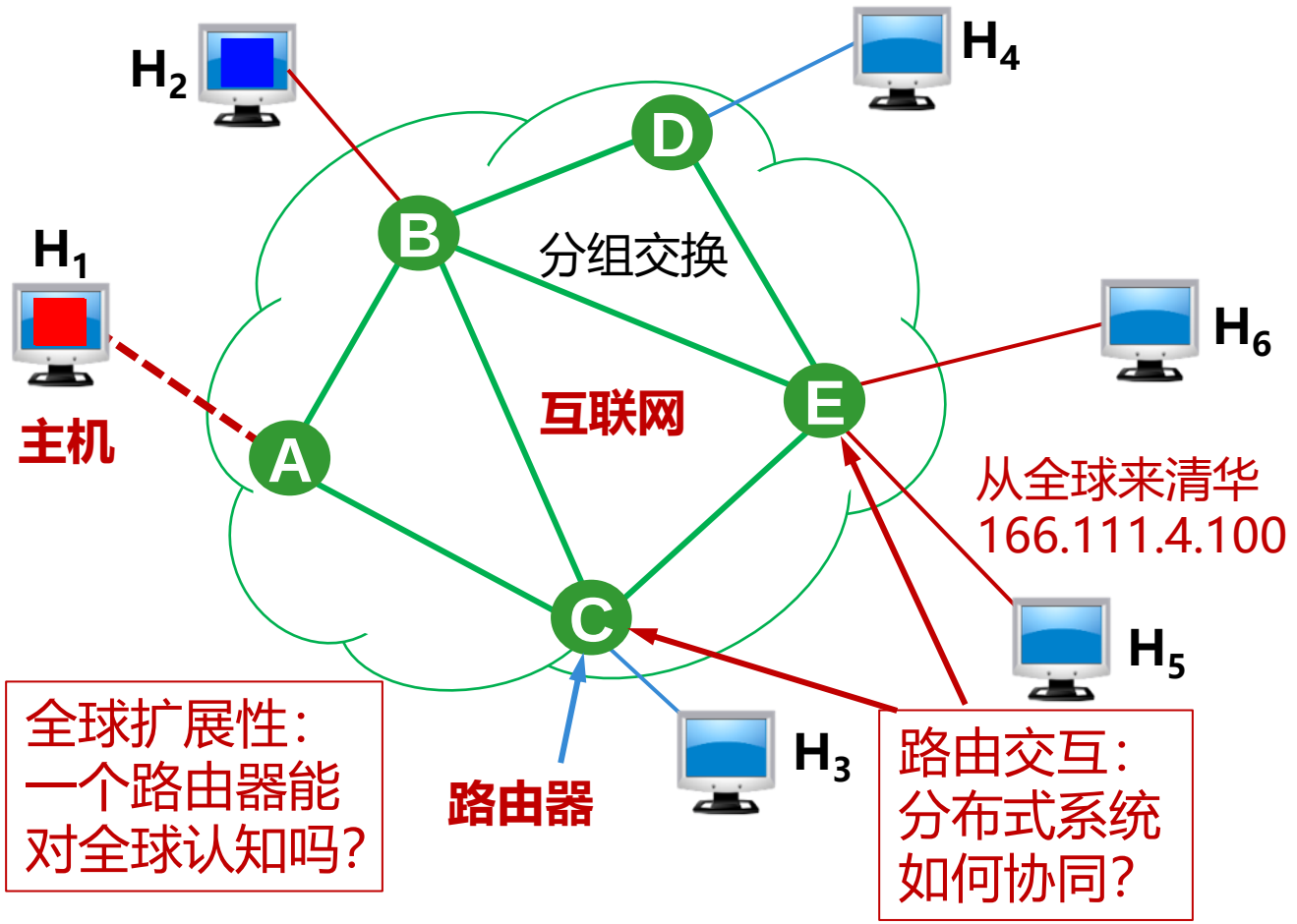
清华大学

计算机网络
教案社区

致谢社区成员	
首都师范大学 陈文龙	武汉大学 吴黎兵
西安邮电大学 谢晓燕	仲恺农业工程学院 邹莹
枣庄学院 李旭宏	辽宁大学 曲大鹏
西南民族大学 方诗虹	浙江理工大学 舒挺
内蒙古农业大学 白云莉	荆楚理工学院 余琨
华为技术有限公司 李振斌	

思考与展望

- 网络层服务
  - 两大功能：路由 & 转发
  - 发明：IP协议（编址）
  - 发明：DHCP，ARP，ICMP
- 大规模路由（Routing）
  - 静态配置？动态学习！
  - 汶川地震和911事件
- 问题分解
  - 玩命化简：路由器间协调计算
  - 扩展性：规模大点，再到全球



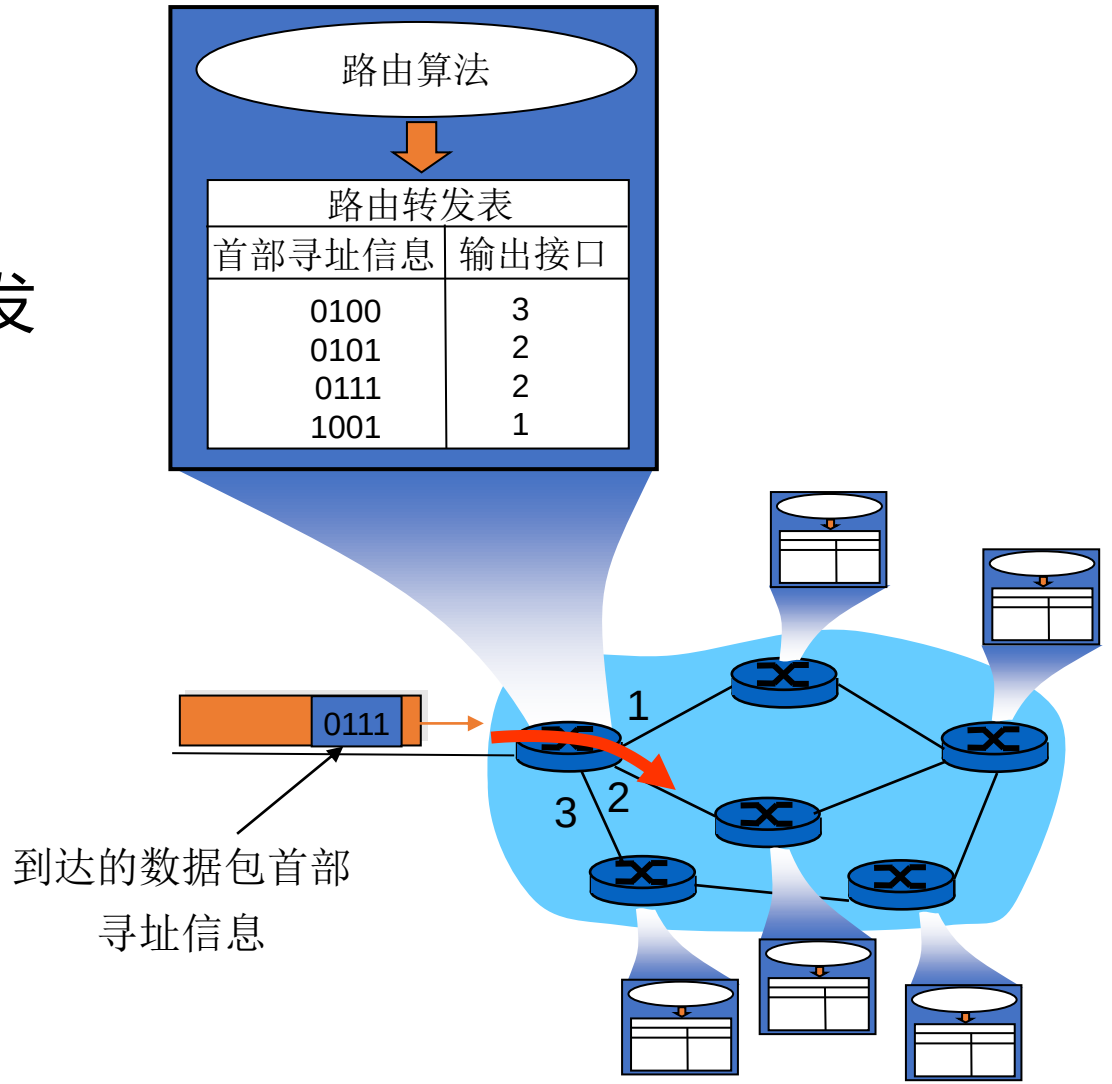
路由协议

- 网络层的基本需求
  - 控制平面：路由协议产生路由表
  - 数据平面：基于路由表进行分组转发

静态路由 v.s. **动态路由**

- 动态路由需要满足的特性
  - 正确性、简单性、鲁棒性
  - 稳定性、高效性、公平性、有效性

复杂性 v.s. 可扩展性



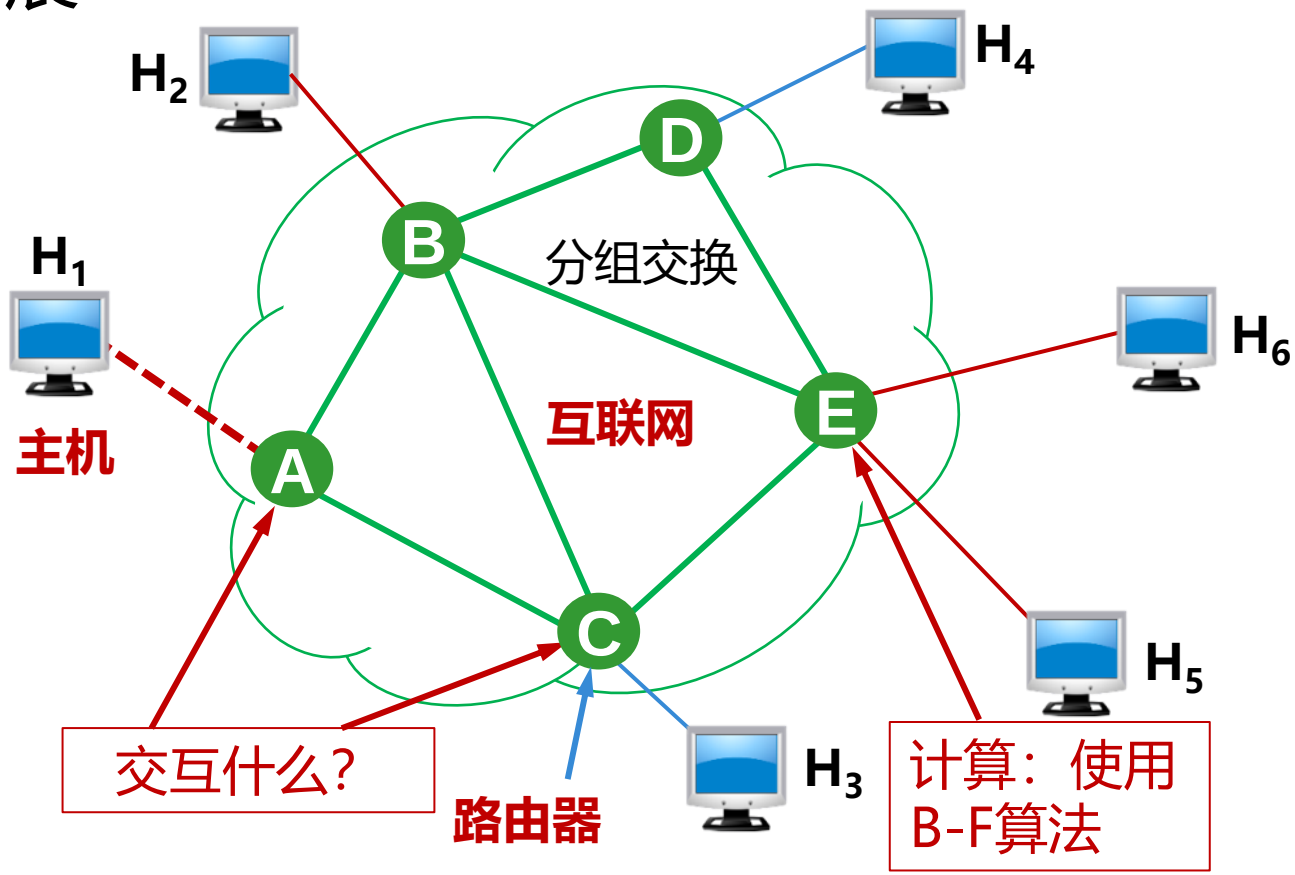
本节课程目标

三大核心路由协议

1. 掌握距离矢量路由算法，掌握路由协议RIP
2. 掌握链路状态路由算法，掌握路由协议OSPF
3. 掌握层次路由概念，掌握路由协议BGP

思考与发明

- 设计思想：最简开始，逐步扩展
- 最短路径问题
  - 回顾：离散数学最短路
  - Bellman-Ford算法
- 基本思路
  - 仅解决必须解决的问题
  - 集中式算法->分布式算法
  - 理论->实际：节点/前缀……
  - 算法->协议：设备间交互方式
  - 发现实际问题（路由回路等），进一步优化



本节内容

6.3 距离向量算法和 RIP

6.4 链路状态算法和 OSPF

6.5 层次路由和域间路由协议 BGP

1. Bellman-Ford
2. 距离向量路由
3. RIP 路由协议

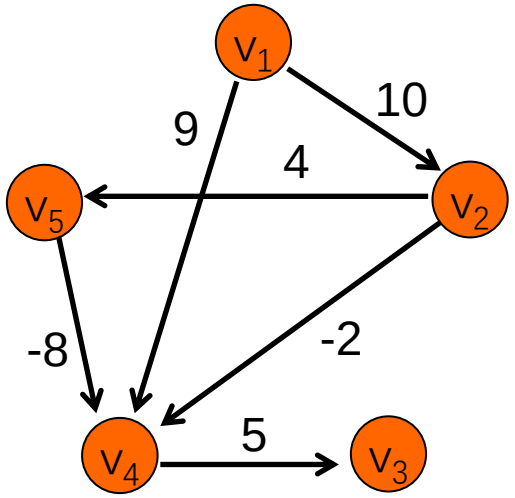
回顾Bellman-Ford

- Bellman-Ford 算法

已知全局信息

  - π(y) 是v_1到 y 最小代价路径的代价值
  - w_{ji}是 j 和 i 之间道路的代价值
  - 1 初始令 $\pi(1)=0, \pi(i)=\infty, i=2,3,\cdots,n$
  - 2 i从2到n循环，令
    $\pi(i) \leftarrow \min[\pi(i), \min_{j\in\Gamma_i^-}(\pi(j)+w_{ji})]$
  - 3 若全部π(i)都没变化，结束；否则转2

π(1)	π(2)	π(3)	π(4)	π(5)	k
0	∞	∞	∞	∞	0
0	10	∞	8	14	1
0	10	13	6	14	2
0	10	11	6	14	3
0	10	11	6	14	4



集中式算法能分布式协作运行吗?

v_4告诉v_3什么内容？v_3如何计算?

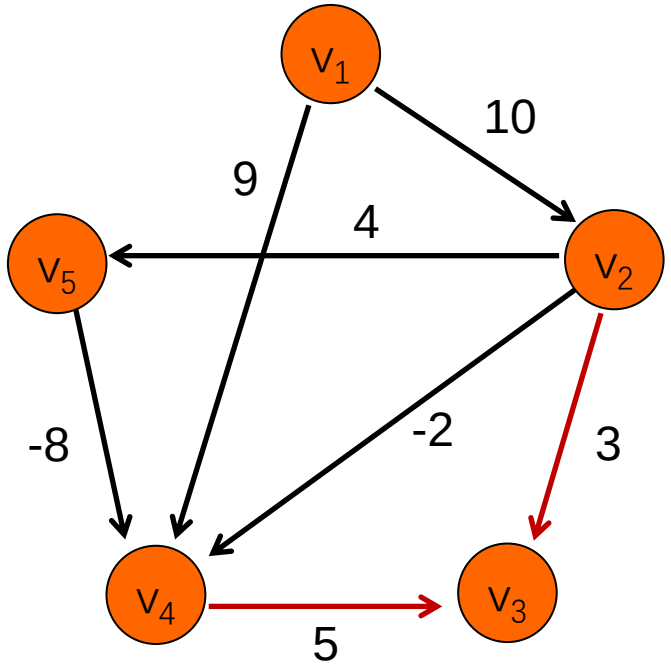
分布式Bellman-Ford

- 路由器v_3如何计算到v_1的最短路
  - v_2 和 v_4 要传输本地$\pi_i(v_1)$ 给 v_3
  - v_3据算法，天才宝宝二选一： 最小的$w_{3i} + \pi_i(v_1)$

$$\pi(i) \leftarrow \min[\pi(i), \min_{j \in \Gamma_i^-}(\pi(j) + w_{ji})]$$

- 分布式 Bellman-Ford 算法
  - 各节点维护到达每个节点的最优<距离,向量>
  - 各节点与邻居交互距离向量，并按算法更新
  - 收敛条件?

玩命简化：持续计算和交互，总会收敛



从v_3的视角看

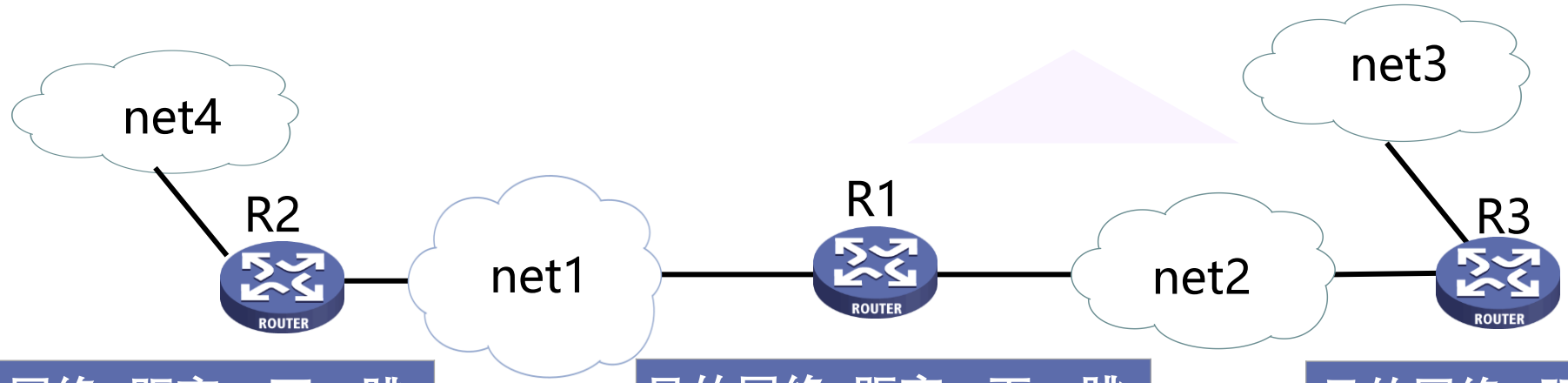
$\pi_2(v_3) = 3$; $w_{12} = 10$; $10+3 = 13$;

$\pi_4(v_3) = 5$; $w_{14} = 9$; $5 + 9 = 14$;

v_3形成到v_1的<距离,向量>：<13,v_2>

距离向量路由

- 路由器启动时初始化自己的路由表
  - 初始路由表包含所有直接相连的网络路径，距离均为0



目的网络	距离	下一跳
net1	0	--
net4	0	--

直连路由

目的网络	距离	下一跳
net1	0	--
net2	0	--

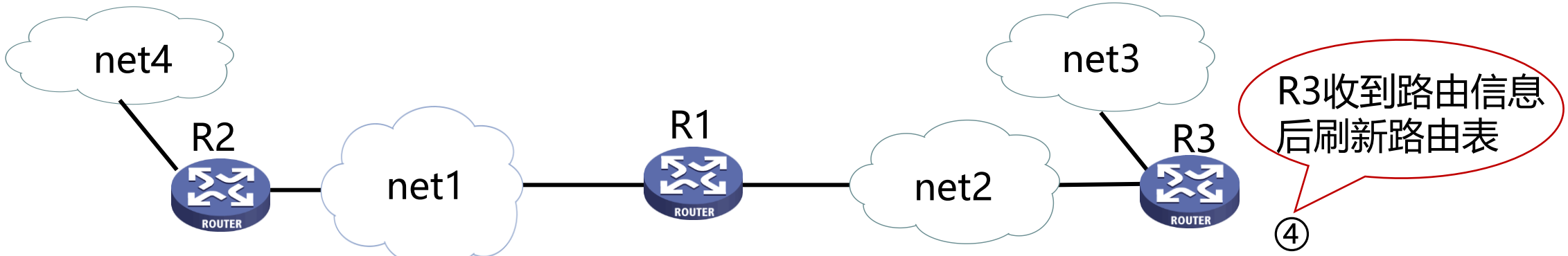
直连路由

目的网络	距离	下一跳
net2	0	--
net3	0	--

直连路由

距离向量路由

- 路由器周期性地向其相邻路由器，广播自己知道的路由信息
- 相邻路由器根据收到的路由信息，修改和刷新自己的路由表



目的网络	距离	下一跳
net1	0	--
net4	0	--

①

R2将路由信息发送给R1

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2

②

R1收到路由信息后刷新路由表

③

R1将路由信息发送给R3

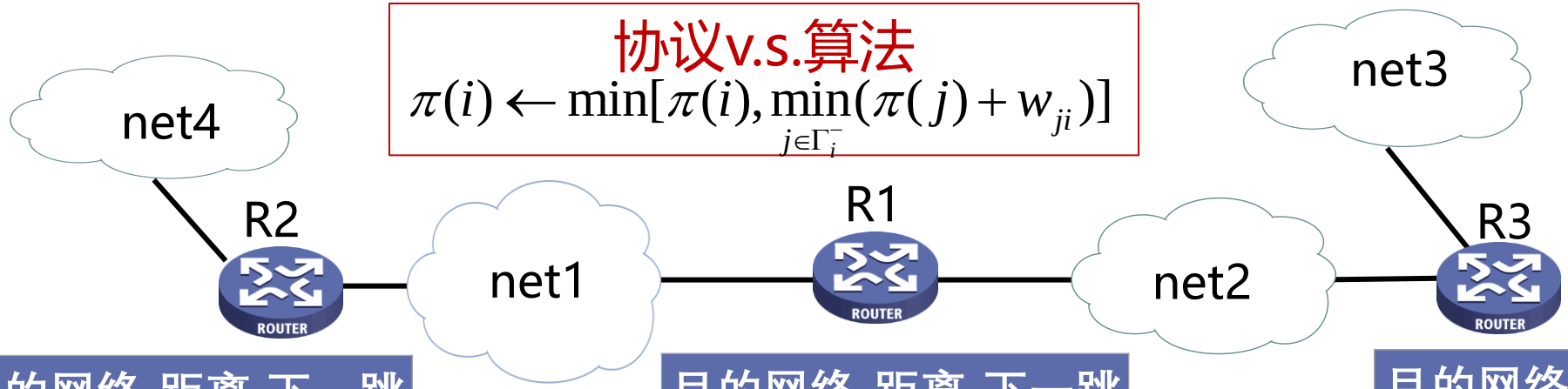
目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

距离向量路由

- 路由器经过若干次更新后，最终都会知道到达所有网络的最短距离
- 所有路由器逐步得到正确路由信息，路由“收敛”（convergence）

协议v.s.算法

$$\pi(i) \leftarrow \min[\pi(i), \min_{j \in \Gamma_i^-}(\pi(j) + w_{ji})]$$



目的网络	距离	下一跳
net1	0	--
net4	0	--
net2	1	R1
net3	2	R1

直接交付

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2
net3	1	R3

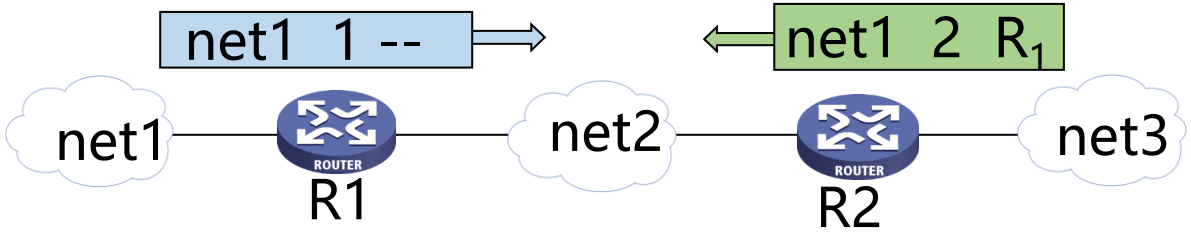
间接交付

目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

距离向量路由

- 发现神奇现象

正常情况

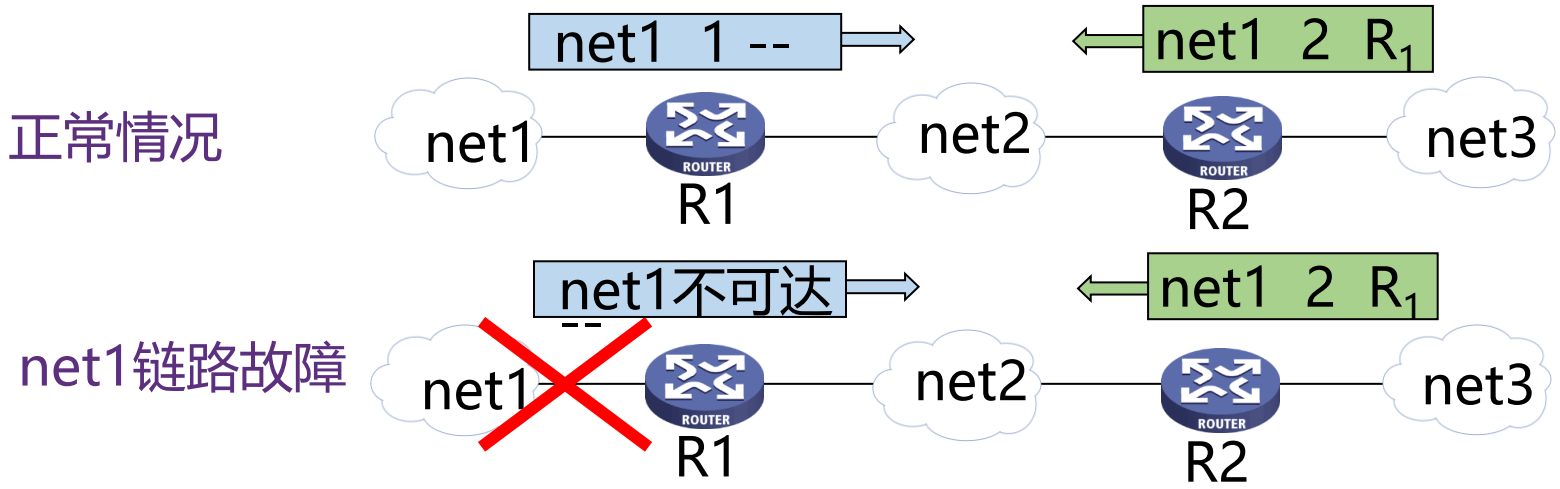


R_1 说：“我到net1 的距离是 1，是直接交付。”

R_2 说：“我到net1 的距离是 2，是经过 R_1。”

距离向量路由

➢ 发现神奇现象



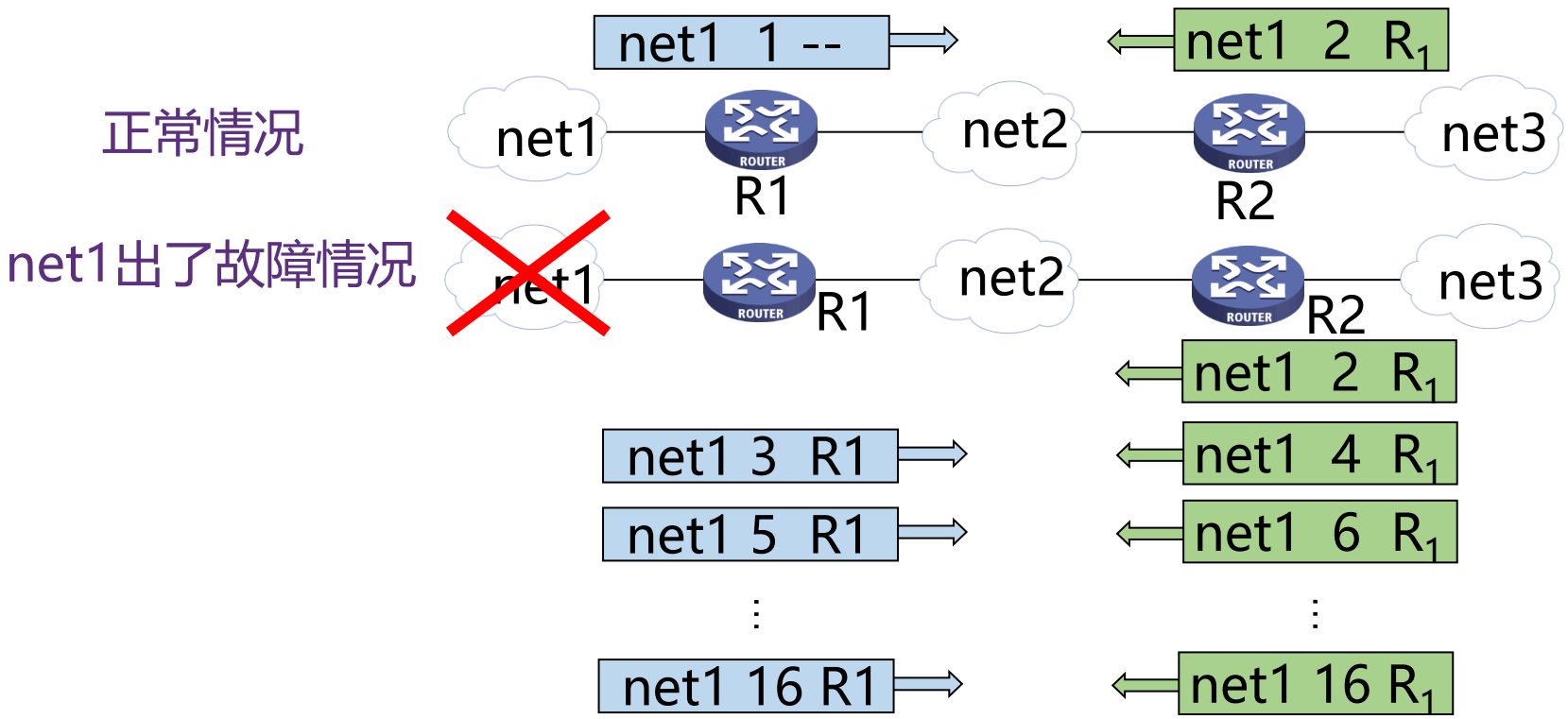
R_1 说：“我到net1 无法到达”

神奇现象

但 R_2 在收到 R_1 的更新报文之前，还发送原来的报文，因为这时 R_2 并不知道 R_1 出了故障。

距离向量路由

- 无穷计算问题（The Count-to-Infinity Problem）



如何解决无穷计算问题？

增加规定：到16 时，认为不可达

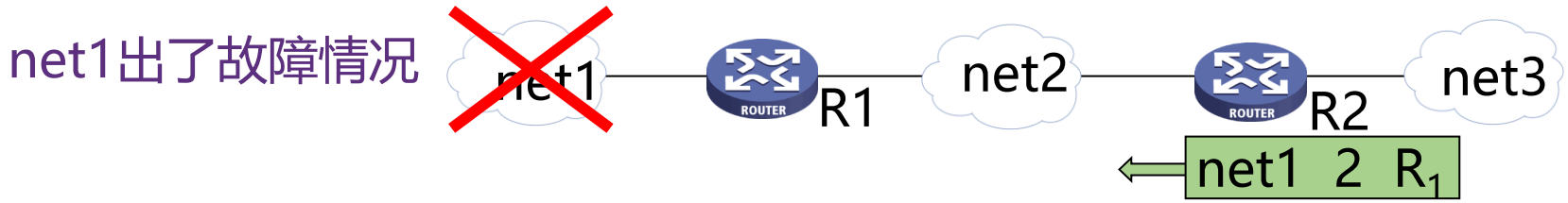
透过现象看本质？

- 好消息传播快，坏消息传播慢，是距离向量路由的一个主要缺点

距离向量路由算法（续）

- 无穷计算问题，透过现象看本质?
- 问题出现在哪个行为 || 步骤上?

如果R2一定要发，
R1有优化策略吗?



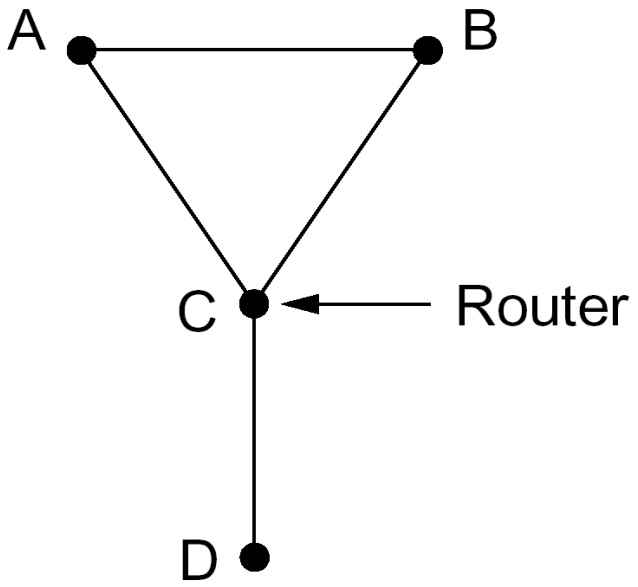
- 原本更靠近目的地的节点，被更远的节点欺骗了
- 核心问题：路由表不加选择地洪泛和使用

R2依赖于向R1的接口转发
因此不应该反向声称转发能力

修改策略：全部洪泛->
过滤依赖于本接口转发的路由

距离向量路由算法（续）

- 水平分裂算法（Split Horizon）
  - 从邻居学到的路由，不向邻居结点报告
  - 避免上述无穷计算问题
- 毒性逆转（Poison Reverse）
  - 改进的水平分裂算法
  - 从邻居学到的路由，向邻居节点报告时， 用距离16(即不可达的度量值)将它广播出去
  - 虽然增加了路由表的大小，但有助于消除路由循环
- 其他优化：触发更新、中毒路由

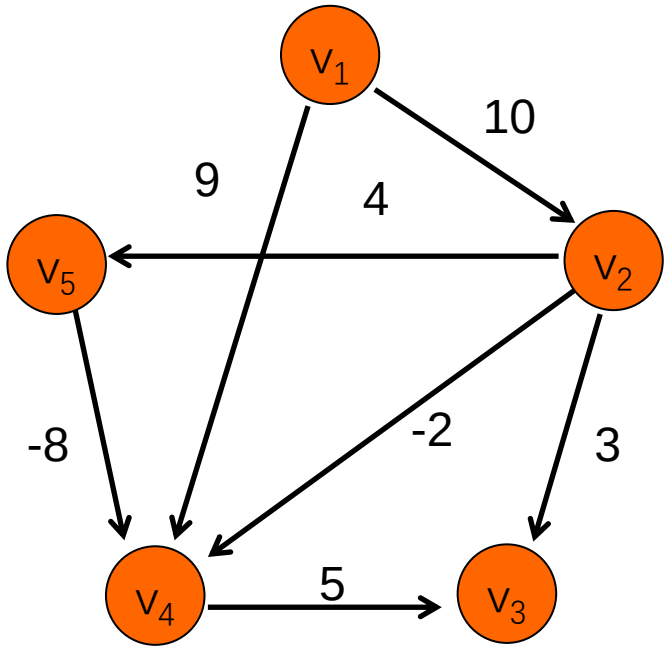


RFC2080：推荐实现
能够按照接口配置策略
No SH | SH | PR

从算法到协议设计

- 分布式Bellman-Ford算法很好，但是……
- 概念和实际的映射
  - 图论：节点、边(物理链路)、边权
  - IP->前缀？邻居发现？边权->开销度量？
- 协议设计：路由信息传播
  - 用什么协议承载路由信息？
  - 路由信息传给谁？
  - 什么时候传播？

思路
玩命简化

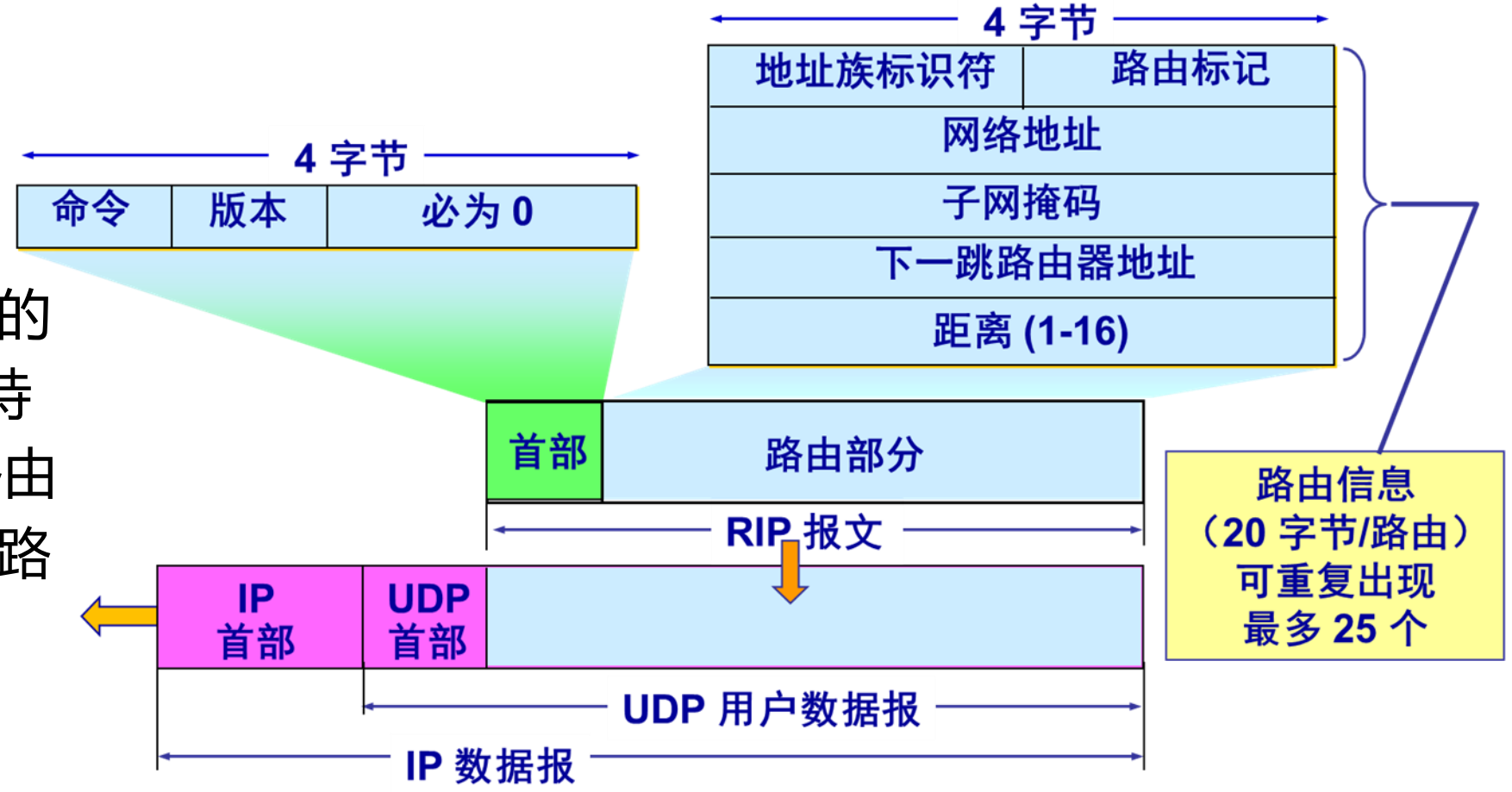


RIP-概述

- RIP协议的基本思想（Routing Information Protocol）
  - 仅和相邻路由器交换信息，使用Bellman-Ford算法
  - 路由器交换的内容是自己计算出来的路由表
- 玩命简化的RIP
  - 最为简单的路由协议，基于距离矢量，使用Bellman-Ford算法
  - 目的地是什么？什么是最佳路由？网络动态性，传输可靠性？
  - 目的地是：标准的IP地址（即A/B/C类地址）
  - 使用跳数衡量的距离，即好路由就是跳数少（增加线路网络变差☹）
  - 通过周期性更新（30s），解决可靠性和动态性问题

RIP-报文格式(V2)

- 协议封装
  - UDP，520
- 版本演进
  - V1：使用标准的IP地址，不支持CIDR和子网路由
  - V2：支持子网路由、身份验证、多播



RIP-报文格式

- 请思考
  - 这个报文是什么报文?
  - 目的地址是多少?
  - RIP版本?
  - 路由表项是什么?
  - RIP的下层协议?
  - 如何实现可靠性?

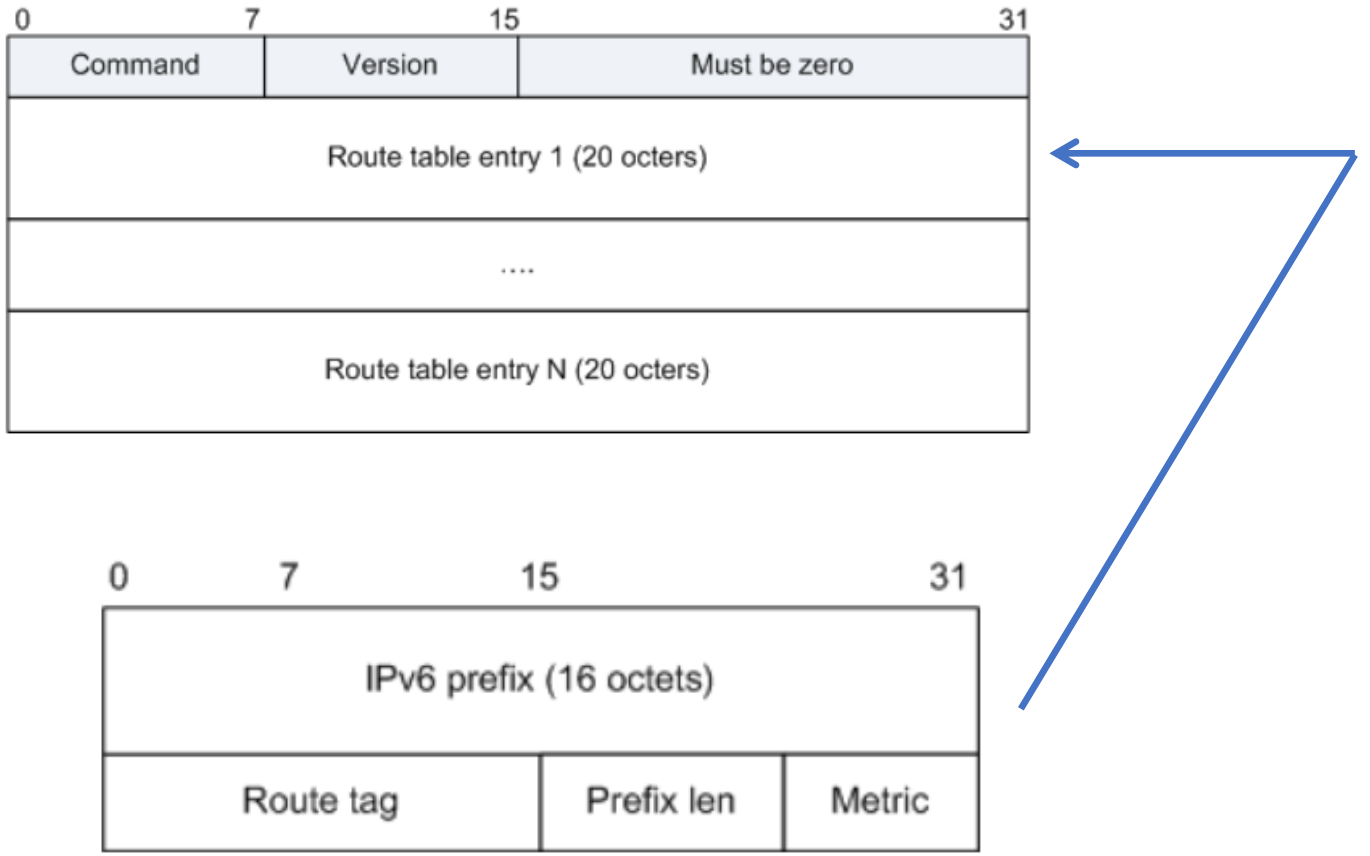
No.	Time	Source	Destination	Protocol	Length	Info
36	481.203000	10.0.12.2	224.0.0.9	RIPv2	86	Response
37	490.125000	10.0.12.1	224.0.0.9	RIPv2	86	Response
38	510.375000	10.0.12.2	224.0.0.9	RIPv2	86	Response
39	525.343000	10.0.12.1	224.0.0.9	RIPv2	86	Response
40	540.671000	10.0.12.2	224.0.0.9	RIPv2	86	Response
41	559.609000	10.0.12.1	224.0.0.9	RIPv2	86	Response
42	572.921000	10.0.12.2	224.0.0.9	RIPv2	86	Response
43	589.875000	10.0.12.1	224.0.0.9	RIPv2	86	Response
44	607.203000	10.0.12.2	224.0.0.9	RIPv2	86	Response
45	625.203000	10.0.12.1	224.0.0.9	RIPv2	86	Response
46	639.359000	10.0.12.2	224.0.0.9	RIPv2	86	Response
47	657.484000	10.0.12.1	224.0.0.9	RIPv2	86	Response
48	674.718000	10.0.12.2	224.0.0.9	RIPv2	86	Response
49	684.671000	10.0.12.1	224.0.0.9	RIPv2	86	Response
50	704.984000	10.0.12.2	224.0.0.9	RIPv2	86	Response
51	709.921000	10.0.12.1	224.0.0.9	RIPv2	86	Response

```
> Frame 39: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: HuaweiTe_db:36:bb (54:89:98:db:36:bb), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
> Internet Protocol Version 4, Src: 10.0.12.1, Dst: 224.0.0.9
> User Datagram Protocol, Src Port: 520, Dst Port: 520
v Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
  v IP Address: 10.0.1.0, Metric: 1
      Address Family: IP (2)
      Route Tag: 0
      IP Address: 10.0.1.0
      Netmask: 255.255.255.0
      Next Hop: 0.0.0.0
      Metric: 1
  > IP Address: 10.0.13.0, Metric: 1
```

```
0000  01 00 5e 00 00 09 54 89  98 db 36 bb 08 00 45 c0   ··^···T· ··6···E·
0010  00 48 00 27 00 00 0e 11  b5 b4 0a 00 0c 01 e0 00   ·H·'···· ········
0020  00 09 02 08 02 08 00 34  e3 61 02 02 00 00 00 02   ·······4 ·a······
0030  00 00 0a 00 01 00 ff ff  ff 00 00 00 00 00 00 00   ········ ········
0040  00 01 00 02 00 00 0a 00  0d 00 ff ff ff 00 00 00   ········ ········
0050  00 00 00 00 00 01                                 ......
```

RIPng-报文格式

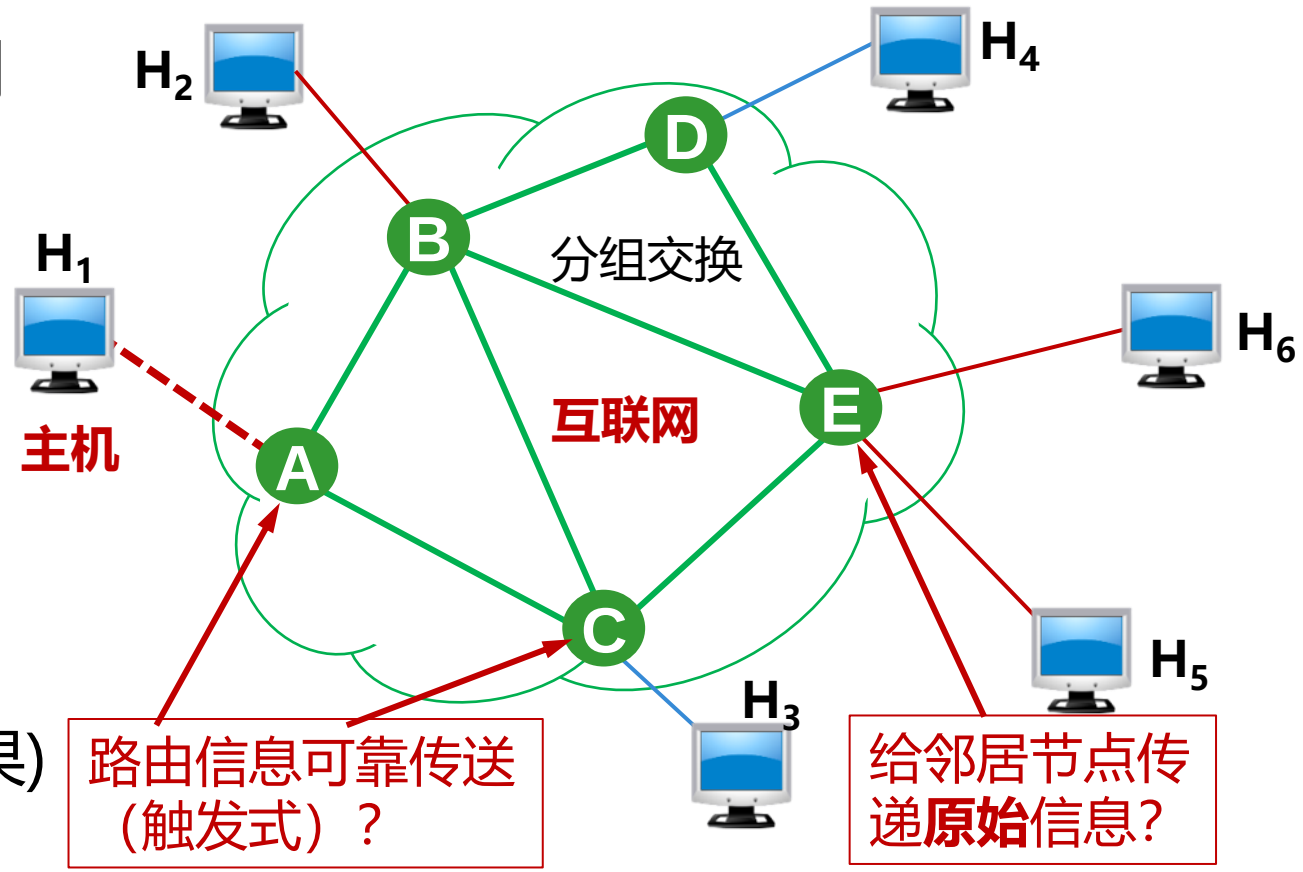
- 协议封装
  - UDP，521
- IPv6适配版（大实验）
  - 协议总体思想和逻辑与IPv4版本完全一致
  - 从32位IPv4升级到128位IPv6地址
  - 细节适配（更长的前缀等）



RIP-扩展性问题

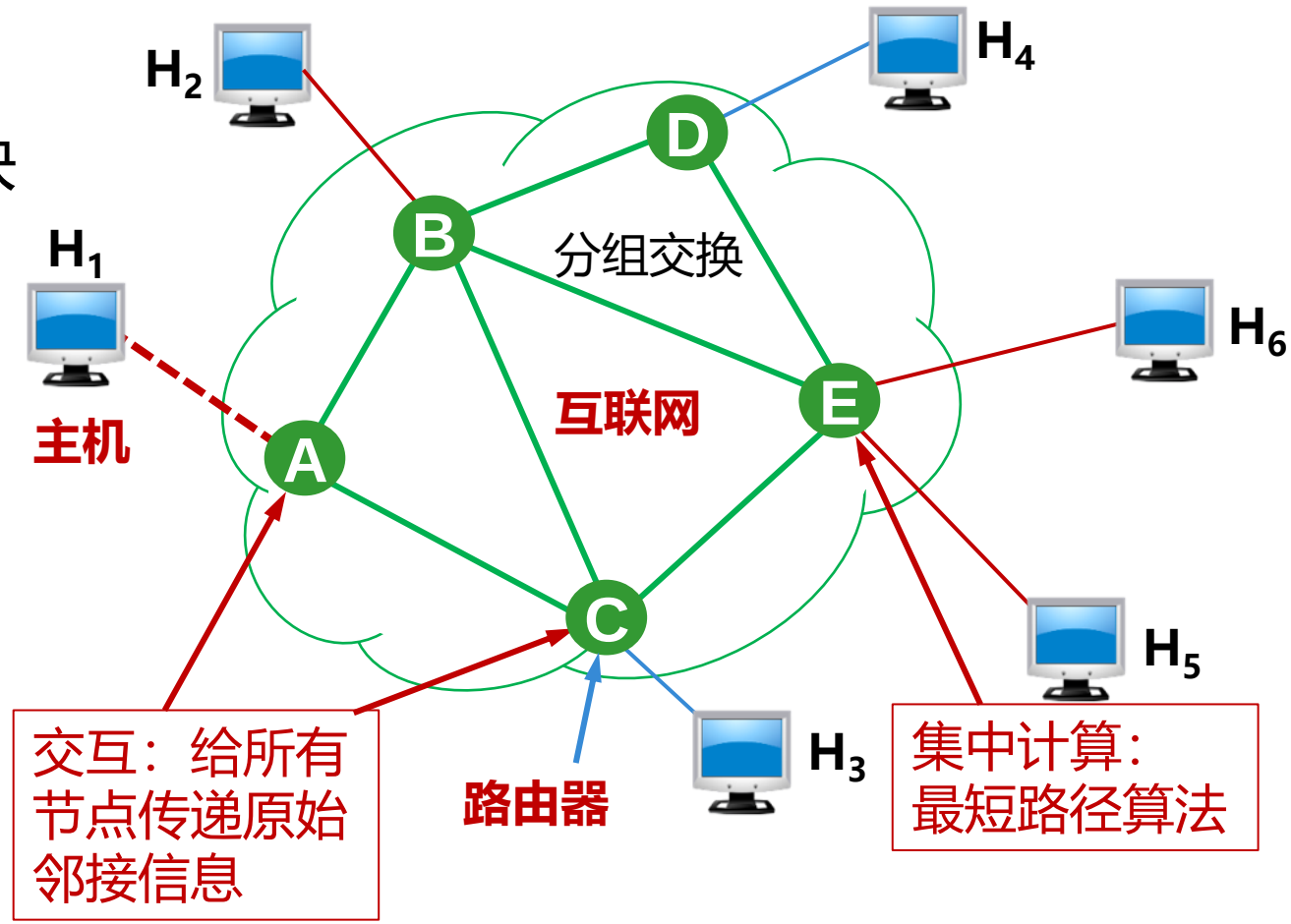
- 分布式带来多种问题
  - 无穷计算：毒性反转、水平分割
  - 收敛慢：依赖邻居计算结果
  - 可靠性：相同信息被反复传输
  - 开销大：周期性重复计算
- RIP协议仅适合中小型网络

  如何设计新协议?
- RIP问题本质
  - 传递信息过少(基于邻居计算结果)
  - 路由信息可靠确认



思考与发明

- 分布式收敛慢->集中式计算
  - 回顾：Dijkstra算法
  - 优势：计算则立刻收敛，速度快
  - 问题：仅有邻接信息无法计算
- 思路：Dijkstra + 集中式
  - 每个路由器都知道全局信息
  - 每个路由器独立计算最短路径（不依赖其他路由器计算）
  - 可靠路由交互
  - 如何支持大规模扩展性?



本章内容

6.3 距离向量算法和 RIP

6.4 链路状态算法和 OSPF

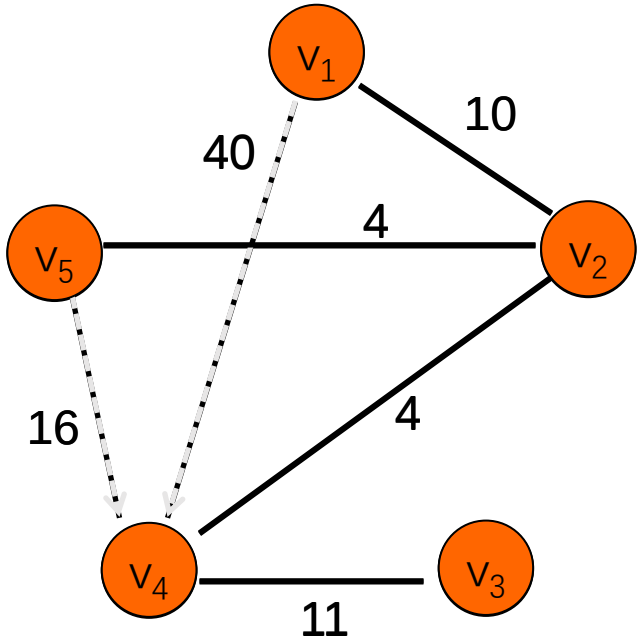
6.5 层次路由和域间路由协议 BGP

1. 最短路径算法Dijkstra
2. 链路状态路由
3. OSPF 路由协议

最短路径算法Dijkstra

回顾

- 使用Dijkstra算法求v_1到其余各点的最短路径
- 已知全局的原始信息，在单个节点上完整计算（不依赖其他节点的计算）



π(2)	π(3)	π(4)	π(5)	-S	访问
10	∞	40	∞	2,3,4,5	2
10	∞	14	14	3,4,5	4
10	25	14	14	3, 5	5
10	25	14	14	3	3
10	25	14	14	Φ	-

构成最短路径树：沿着树到达任何节点均为最短路径（从v_1开始）

最短路径

- Dijkstra算法（1959）

1. 置 $\overline{S}=\{2,3,\cdots,n\}, \pi(1)=0, \pi(i)=\begin{cases} l_{1i} & i\in\Gamma_1^+ \\ \infty & other \end{cases}$

   $\overline{S}$为尚未找到最短路径的节点集

2. 在 $\overline{S}$ 中寻找j满足 $\pi(j)=\min_{i\in\overline{S}}\pi(i), \overline{S}\leftarrow\overline{S}-\{j\}$

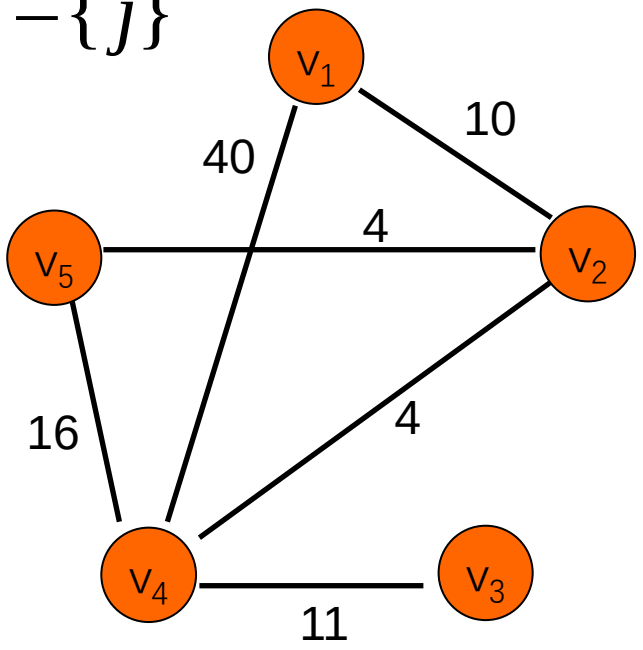
   若 $\overline{S}=\Phi$，结束；否则转3。

3. 对全部 $i\in\overline{S}\cap\Gamma_j^+$ 置：

   $\pi(i)\leftarrow\min(\pi(i),\pi(j)+l_{ji})$ 转2

   更新j的直接后继(各i)的距离并记录前驱

访问最近节点j



Dijkstra算法复杂度？ $O(n^2)$

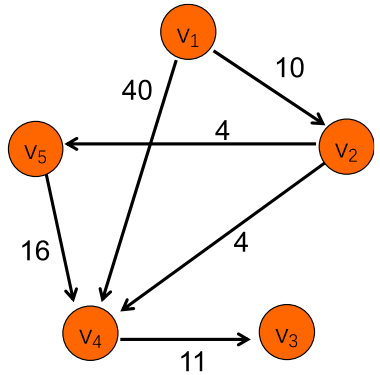
从算法到协议设计

- 算法的基础是协议交互
- 理论到实际
  - 节点还是IP前缀
  - 邻居发现？到邻居的链路开销？

- 路由信息传播
  - 用什么协议承载？
  - 如何实现全局洪泛，可靠性呢？
  - 周期性广播产生冗余？

RIP思路：玩命简化

边权(开销)：跳数
邻居发现：乱喊
承载协议：UDP(IP)
如何洪泛：邻居计算
什么时候：定时器

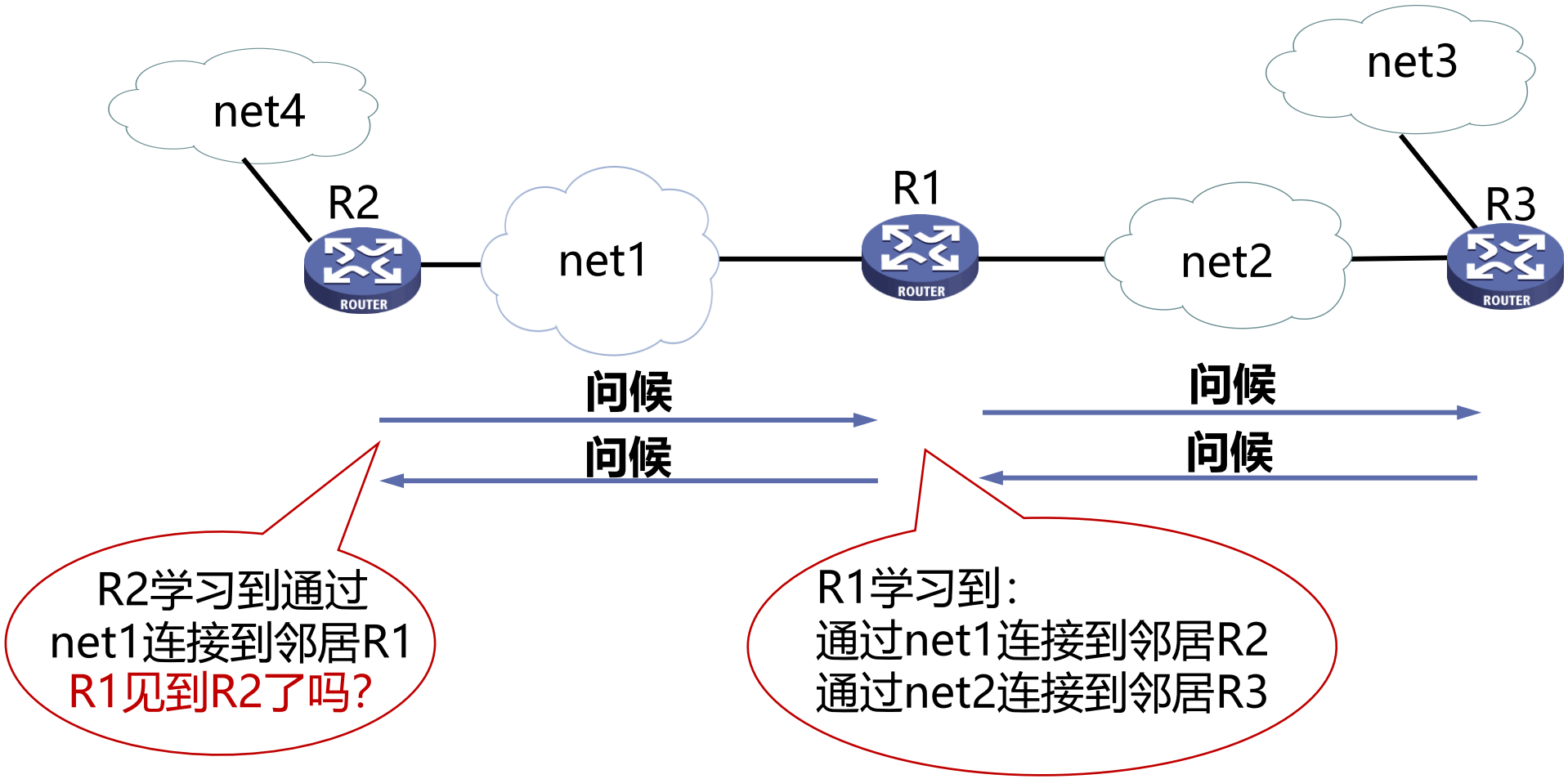


OSPF思路：规模扩展

边权(开销)：选路优化
邻居发现：主动发现
承载协议：IP+可靠
如何洪泛：邻居转发
什么时候：增量更新

链路状态路由

- 1.发现邻居，了解他们的网络地址



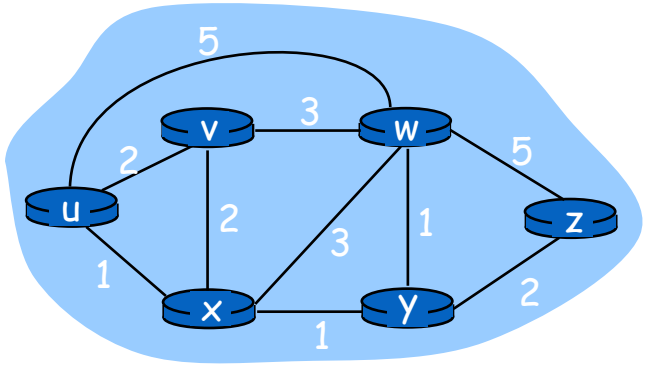
链路状态路由

- 2.设置到每个邻居的成本度量
  - 开销/度量/代价：
    - 自动发现设置或人工配置
    - 度量：带宽、跳数、延迟、负载、可靠性等
  - 常用度量：链路带宽（反比）
    - 例如：1-Gbps以太网的代价为1，100-Mbps以太网的代价为10
  - 可选度量：延迟
    - 发送一个echo包，另一端立即回送一个应答
    - 通过测量往返时间RTT，可以获得一个合理的延迟估计值
  - 服务质量路由?

链路状态路由

- 3.构造链路状态信息，包含刚收到的所有信息
  - 构造链路状态分组（link state packet，LSP）
    - 发送方标识
    - 邻居列表、序列号、年龄
    - 每个LSP分组包含一个不断递增的序列号

链路状态漂泊在外，哪个是最新的?



u	
Seq.	
Age	
v	2
x	1
w	5

v	
Seq.	
Age	
u	2
x	2
w	3

x	
Seq.	
Age	
u	1
v	2
w	3
y	1

y	
Seq.	
Age	
x	1
w	1
z	2

w	
Seq.	
Age	
u	5
v	3
x	3
y	1
z	5

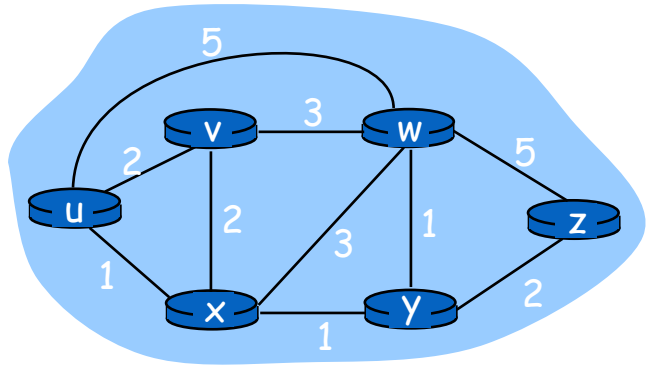
z	
Seq.	
Age	
w	5
y	2

z	
Seq.	
Age	
w	5

如何避免重复?

链路状态路由

- 4.将LSP分组发送给其他的路由器
  - 路由器记录所收到的所有（源路由器、序列号）对
  - 当一个新分组到达时，路由器根据记录判断：
    - 如果是新分组，洪泛路由信息
    - 如果是重复分组，丢弃
    - 如果是过时分组，拒绝



- 交互原始信息：链路两端IP + 链路度量
- 除非链路改变，否则原始信息长期有效
- 通过邻居，告诉世界（不依赖邻居计算）

链路状态路由

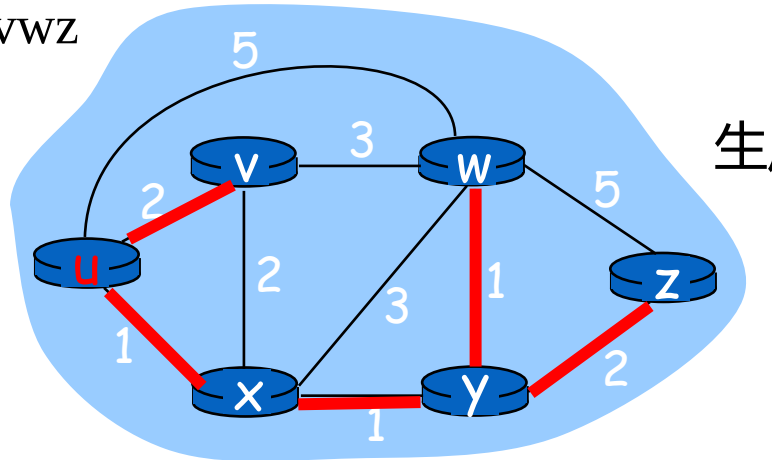
5.计算到其他路由器的最短路径：Dijkstra算法示例

- D(k)：从计算节点到目的节点k当前路径代价
- p(k)：从计算节点到目的节点k的路径中k节点的前继节点

步骤	集合N	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

生成最短路径树

u为树根?



生成路由表

目的	下一跳	代价
v	v	2
w	x	3
x	x	1
y	x	2
z	x	4

链路状态路由

- 链路状态（Link State）路由可分为五个部分：
  - 1. 发现邻居，了解他们的网络地址
  - 2. 设置到每个邻居的成本度量
  - 3. 构造分组，分组中包含刚收到的所有信息
  - 4. 将此分组发送给其他所有路由器？通过邻居！
  - 5. 每个路由器独立计算到其他路由器的最短路径

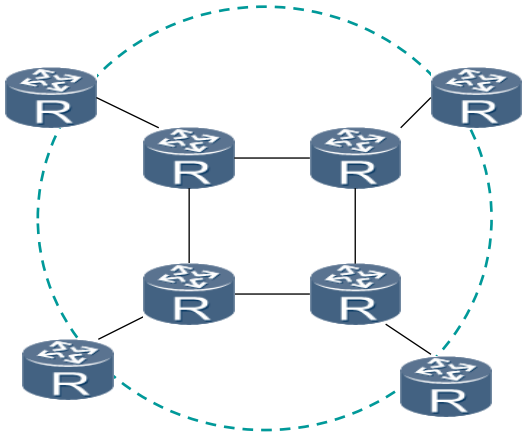
初始: 习得邻接关系和成本度量（协议）

交互: 给所有节点传递邻接信息（协议）

计算: Dijkstra（算法）

OSPF-概述

- OSPF（Open Shortest Path First）
  - 开放最短路径优先协议于1989 年开发
  - 目标：采用链路状态算法建立路由表
  - 核心需求：和谁交互，交互什么，减小开销



- OSPF协议的基本思想
  - 每个路由器与邻居，建立邻接关系
  - 每个路由器通过邻居帮忙，向区域内所有路由器洪泛路由状态信息
  - 每个路由器获知并维护区域内的拓扑结构图
  - 每个路由器采用Dijkstra算法独立计算：自己到全世界的路由器

OSPF-链路状态

- 链路状态LS
  - LS说明本路由器都和哪些路由器相邻，以及该链路的“度量”(metric)
  - OSPF度量值一般包括费用、距离、时延、带宽等
- 同步链路状态数据库
  - 各路由器间交换链路状态信息，使得所有路由器最终都能建立链路状态数据库LSDB，其交互过程称为链路状态数据库的同步
  - LSDB构成网络拓扑结构图，它在区域内是一致的

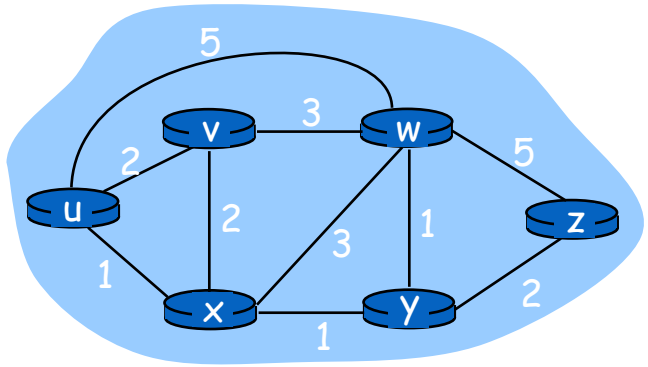
不同路由器的LSDB会不一致吗?

OSPF-五种报文

- 1 Hello 报文
  - 用于发现、维护邻居关系
- 2 数据库描述（Database Description）报文
  - 用于描述自己的LSDB
  - 内容包括LSDB 中每一条LSA 的Header 头部，对端路由器根据LSA Header 就可以判断出是否已有这条LSA
- 3 链路状态请求（LSA Request，LSR）报文
  - 用于请求缺少的LSA，内容包括所需要的LSA 的摘要
- 4 链路状态更新（LSA Update，LSU）报文
  - 用于向对端路由器发送所需要的LSA，内容是多条LSA（全部内容）的集合
- 5 链路状态确认（Link State Acknowledgment，LSACK）报文
  - 用来对接收到的LSU 报文进行确认

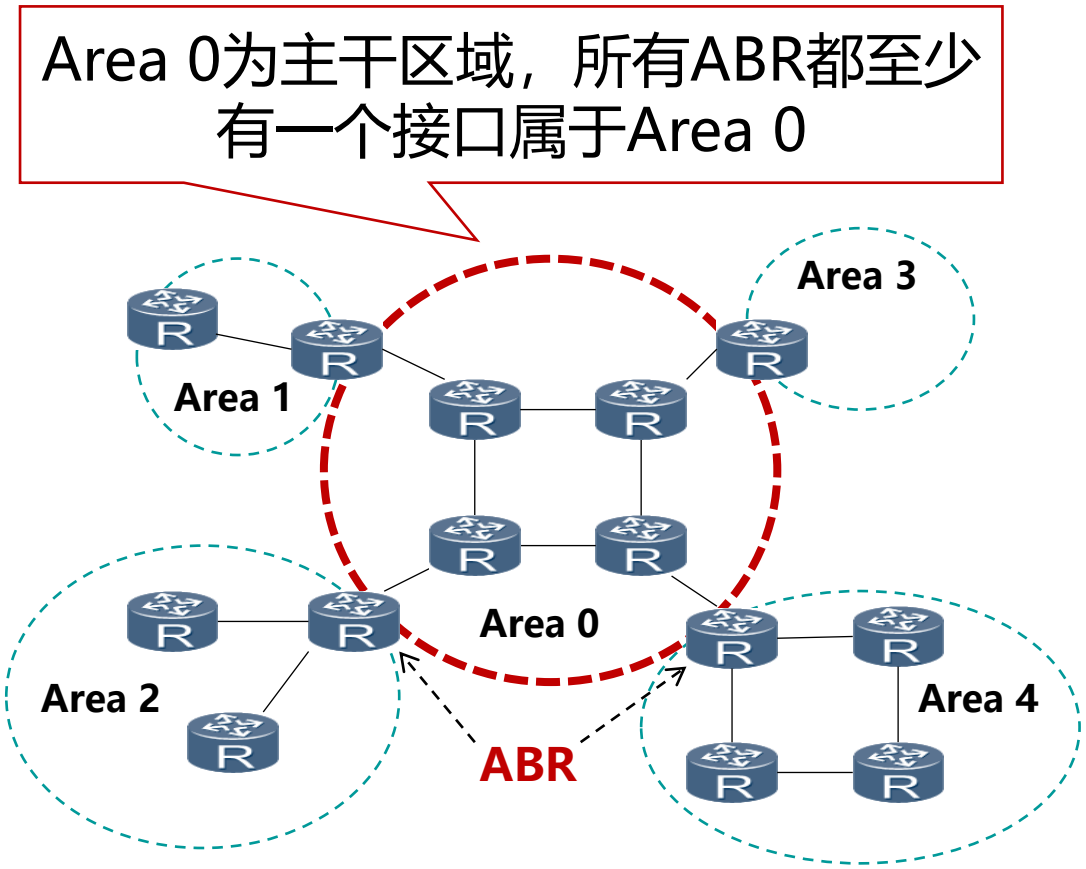
邻居发现&成本度量

如何减少路由更新信息
10000条路由?
如v如何获得z的路由信息?



OSPF-区域的概念

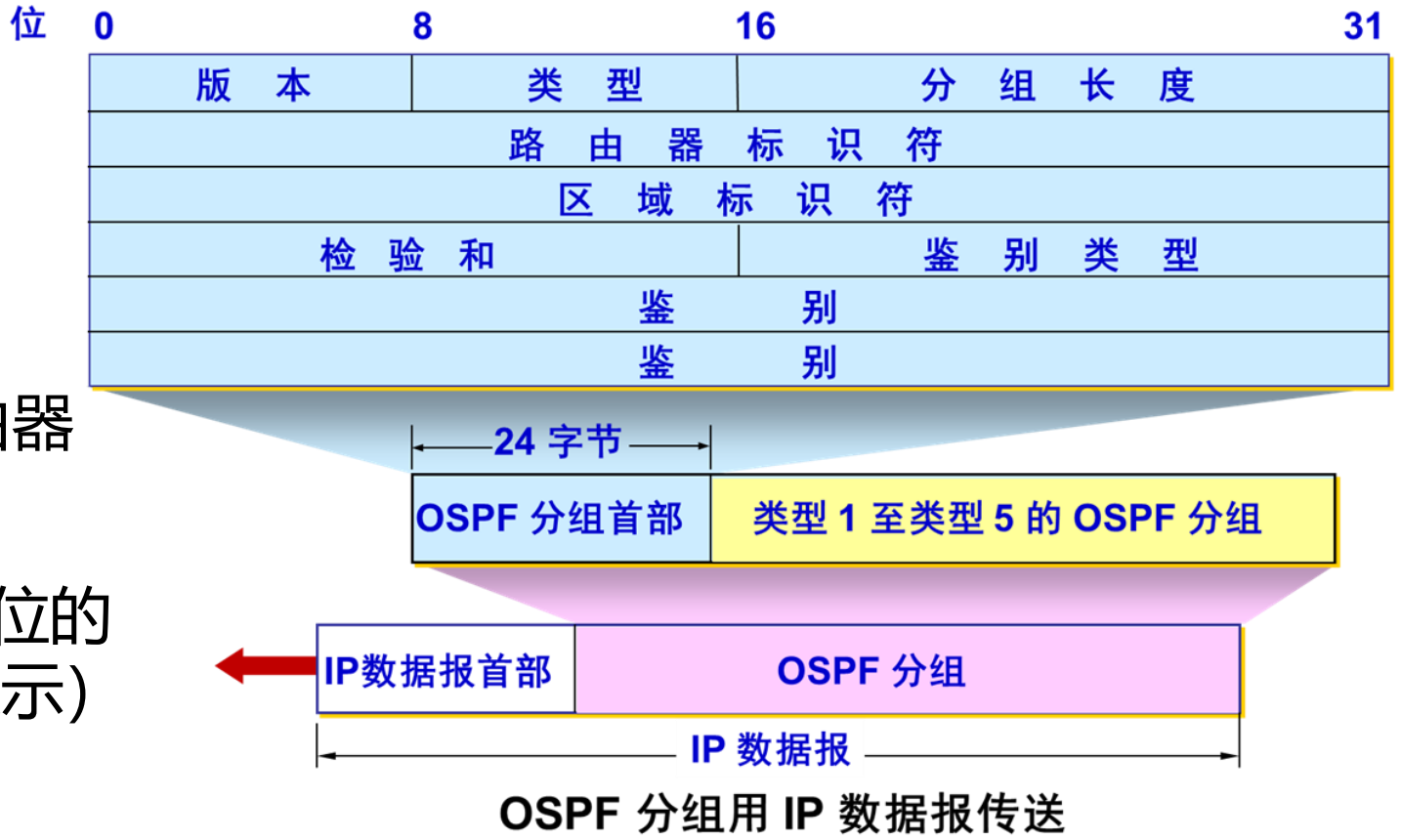
- OSPF扩展性问题
  - 计算、传输开销大
  - 节点的计算不能相互分摊
  - 参考RIP，优化OSPF
- OSPF划分区域
  - 主干区域、非主干区域
- 路由器角色
  - OSPF内部路由器
  - OSPF区域边界路由器ABR (Area Bounder Router)



详细链路状态信息仅在域内传递
区域间传递抽象路由信息（距离向量）（RIP的思想）

OSPF-报文格式

- 协议封装
  - 运行于IP协议之上
  - IP协议号89
- 路由器标识符
  - 32位，唯一标识一个路由器
- 区域标识符
  - 每一个区域都有一个 32 位的标识符（用点分十进制表示）
  - 主干区域为0.0.0.0

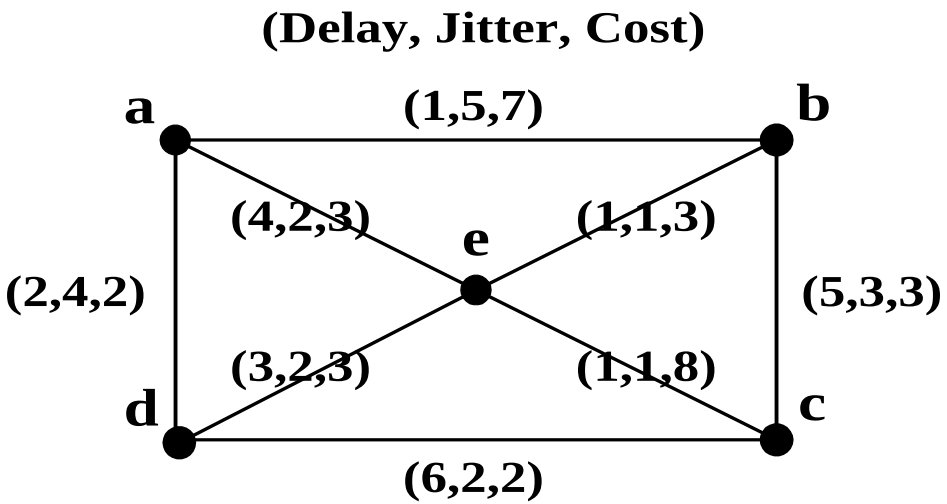


OSPF-报文格式

- 请观察
  - 这是什么报文?
  - 承载协议?
  - 目的IP?
  - TTL?
  - OSPF分组类型?
  - OSPF域号码?

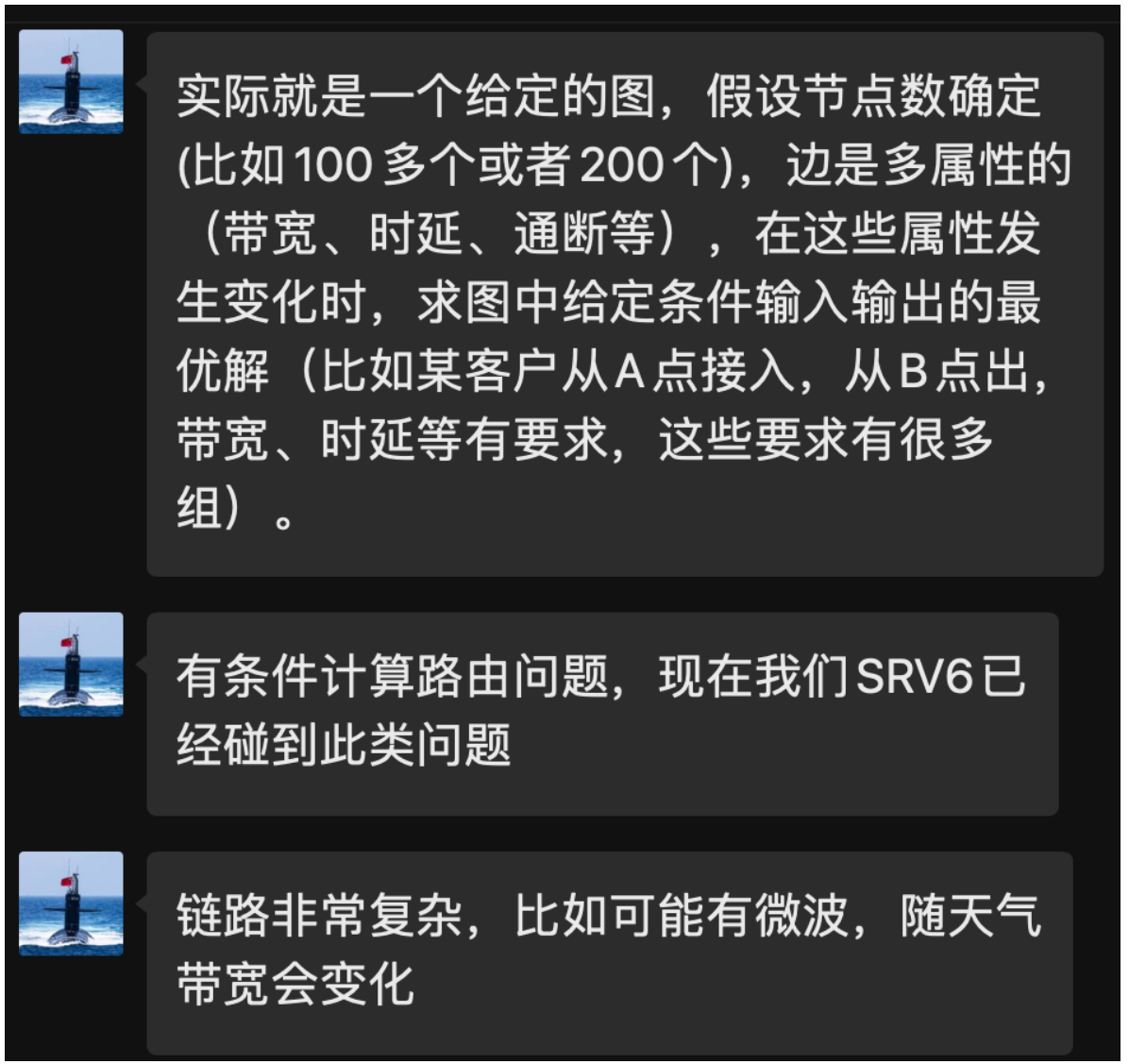
```
Internet Protocol Version 4, Src: 12.12.12.1 (12.12.12.1), Dst: 224.0.0.5 (224.0.0.5)
    Version: 4
    Header Length: 20 bytes
  ⊞ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT
    Total Length: 76
    Identification: 0x0002 (2)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
  ⊞ Header checksum: 0xc085 [validation disabled]
    Source: 12.12.12.1 (12.12.12.1)
    Destination: 224.0.0.5 (224.0.0.5)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Open Shortest Path First
  OSPF Header
      Version: 2
      Message Type: Hello Packet (1)
      Packet Length: 44
      Source OSPF Router: 1.1.1.1 (1.1.1.1)
      Area ID: 0.0.0.0 (0.0.0.0) (Backbone)
      Checksum: 0xea9c [correct]
      Auth Type: Null (0)
      Auth Data (none): 0000000000000000
```

服务质量路由



寻找路径(a⇒c)满足约束c=(7,8,9)?
即w(a⇒c)≤ (7,8,9)

权的可加性?



OSPF-小结

- OSPF的特点
  - 支持无类域间路由（CIDR）
  - 无路由自环（本地计算无自环）
  - 收敛速度快
  - 使用IP组播收发协议数据
  - 支持多条等值路由
  - 支持协议报文的认证
  - 可使用区域概念优化协议交互
- OSPF适合大中型网络

- OSPF缺点
  - 使用Dijkstra算法，每个路由器独立，计算导致计算量大
  - 协议复杂（状态机维护），路由交互传输压力较大
  - 基于IP，人工实现“可靠传输”

优化思路：按照区域组织OSPF，区域间不传输全部的原始信息

区域思想推广：或许互联网可以分区域解决路由问题？

距离向量路由 vs 链路状态路由

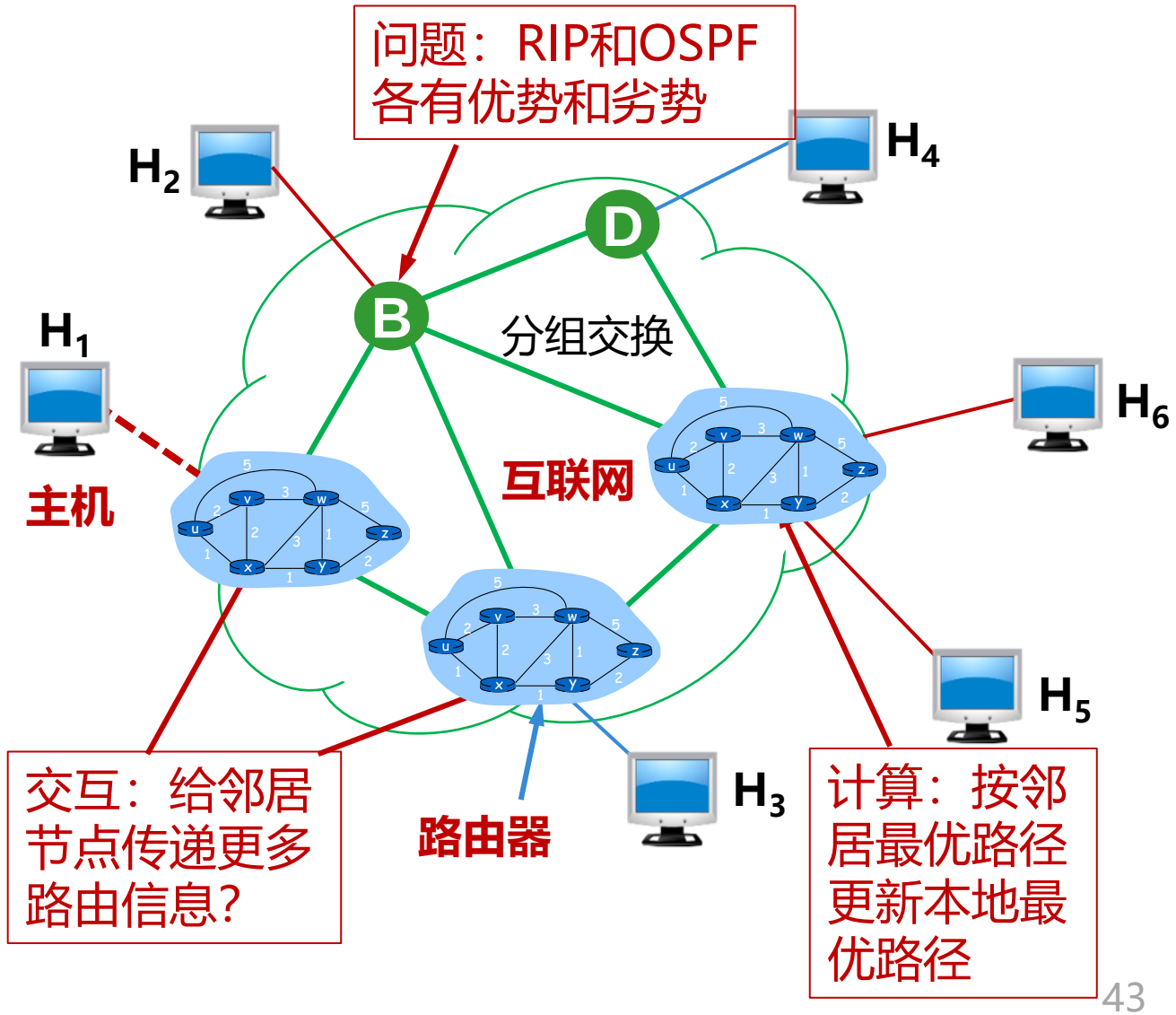
- RIP -距离向量DV：将世界的(最佳)信息告诉邻居
- OSPF -链路状态LS：将自己的信息通过邻居告诉世界

➢ 距离向量和链路状态算法比较

- 网络状态信息交换的范围
  - DV:邻居间交换
  - LS:全网扩散
- 网络状态信息的可靠性
  - DV:部分道听途说
  - LS:使用原始状态信息
- 健壮性:
  - DV:计算结果传递，健壮性差
  - LS: 各自计算，健壮性好
- 收敛速度:
  - DV: 慢，有无穷计算问题
  - LS: 快

思考与发明

- 全球互联？巨人呢？
  - OSPF独立计算导致计算压力大
  - RIP计算互相帮助但有回路
  - 参考OSPF分层区域概念
  - 可靠传输仅在变化时计算
- 扩展性全球路由设计思路
  - 网络分层：推广区域概念，将网络看成节点
  - 收敛问题：基于距离向量利用更多信息？
  - 无回路的可靠分层RIP？



本节内容

6.3 距离向量算法和 RIP

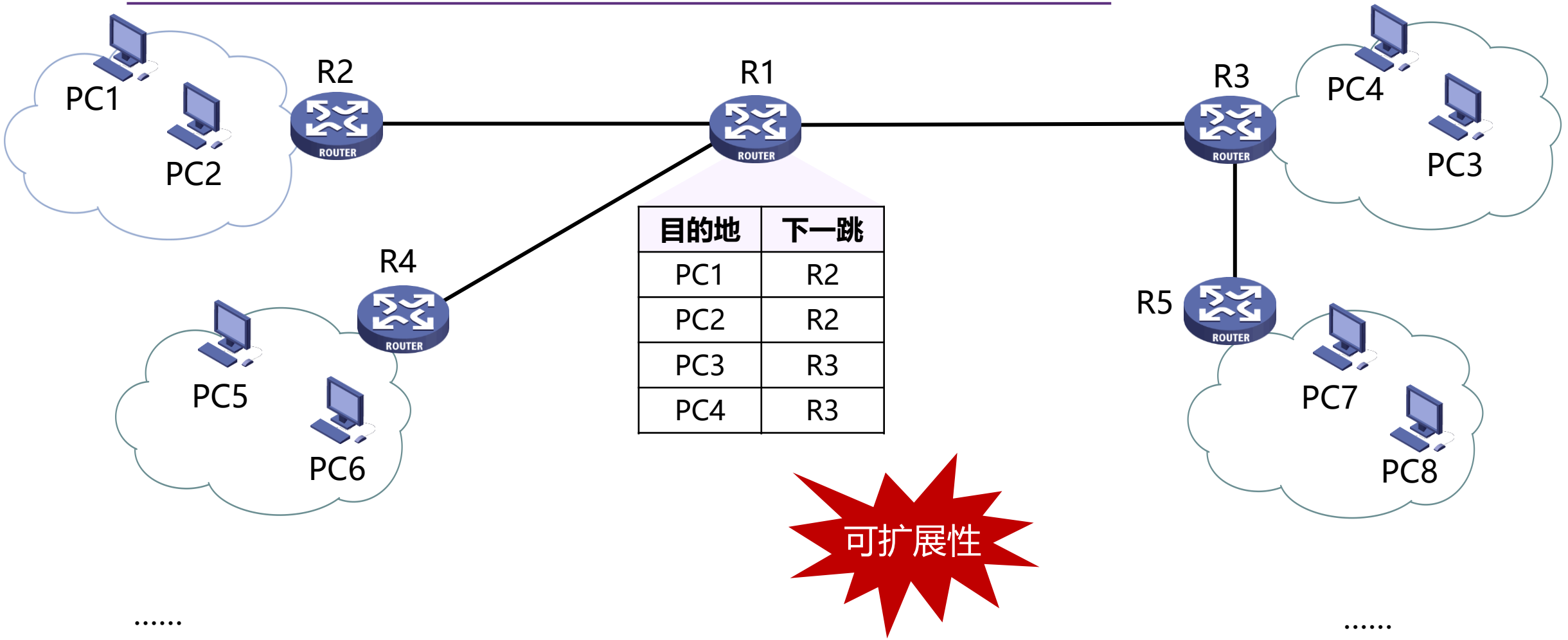
6.4 链路状态算法和 OSPF

6.5 层次路由和域间路由协议 BGP

6.6 标签交换和MPLS

1. 层次路由概念
2. BGP 路由协议

层次路由-产生原因

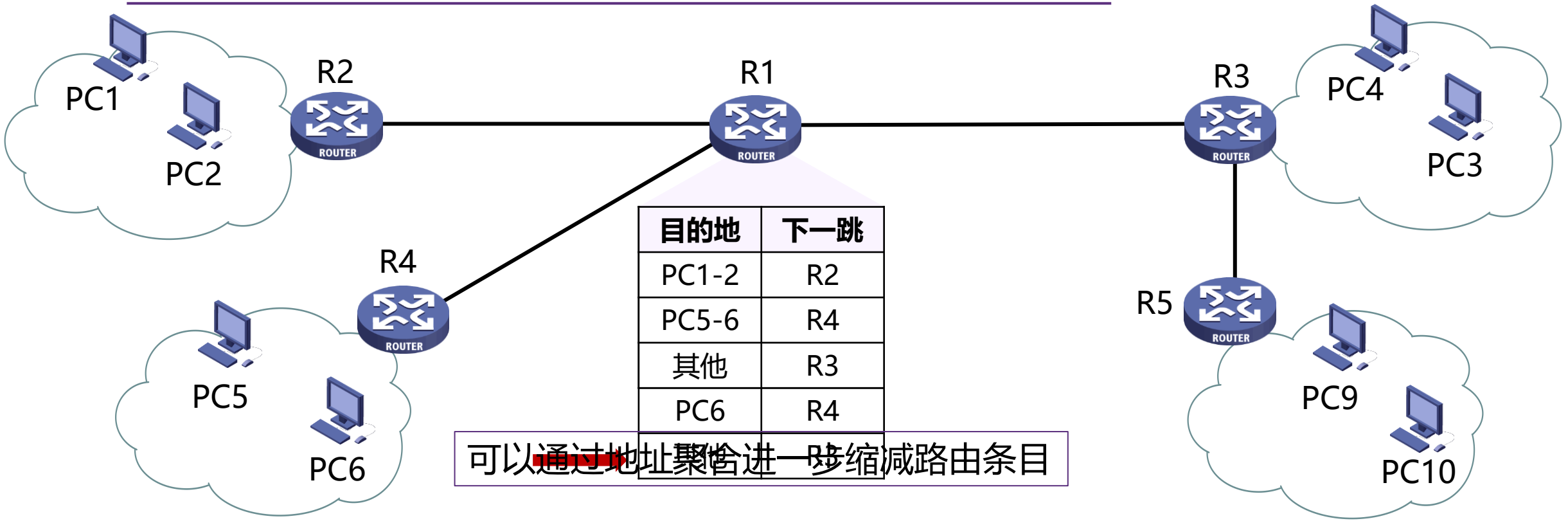


目的地	下一跳
PC1	R2
PC2	R2
PC3	R3
PC4	R3

可扩展性

过于庞大的路由表存储、查找困难，路由信息交互开销高

层次路由-产生原因



目的地	下一跳
PC1-2	R2
PC5-6	R4
其他	R3
PC6	R4
……	……

可以通过地址聚合进一步缩减路由条目

地址聚合依赖于连续的地址空间及拓扑结构

减少路由表存储空间
需要缩减路由条目

层次路由-效果

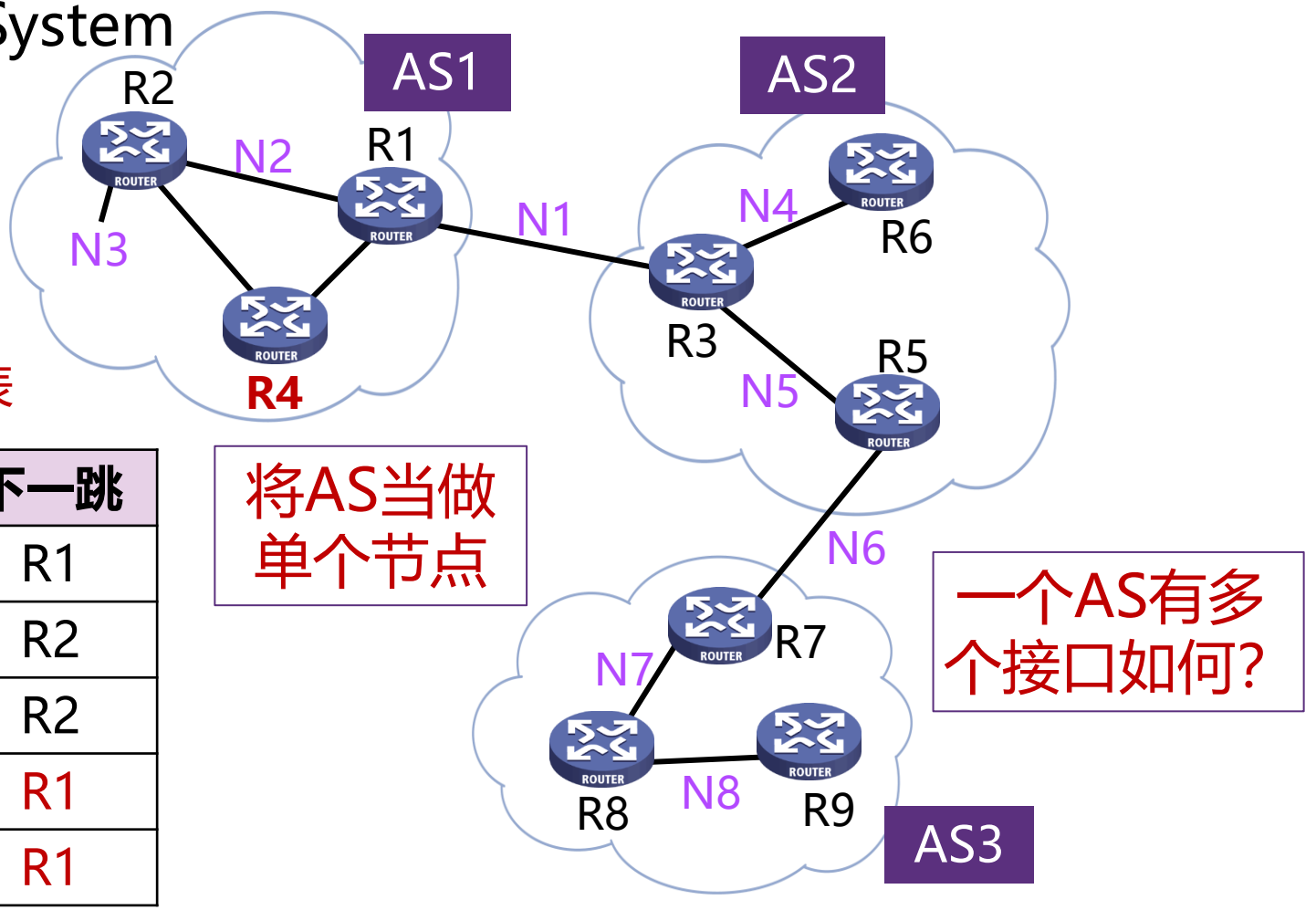
自治系统AS: Autonomous System

路由不分层情况下
R4路由表

目的地	下一跳
N1	R1
N2	R2
N3	R2
N4	R1
N5	R1
N6	R1
N7	R1
N8	R1

R4的层次表

目的地	下一跳
N1	R1
N2	R2
N3	R2
AS2	R1
AS3	R1

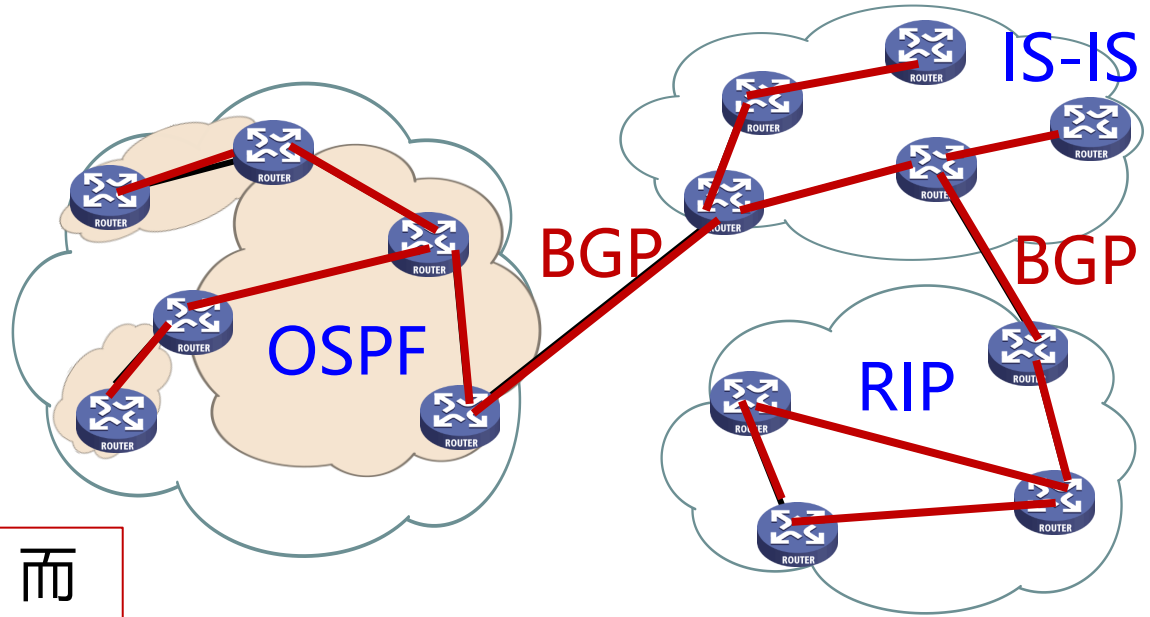


将AS当做
单个节点

一个AS有多
个接口如何?

层次路由-基本思路

- 自治系统内：使用内部网关协议IGP
  - IGP：Interior Gateway Protocols，如OSPF，RIP，IS-IS……
  - 区域边界路由器ABR(Area Bounder Router)
    - 每个AS内路由算法协议相同
- 自治系统间：使用外部网关协议
  - Exterior Gateway Protocols
  - 各自治系统域之间的路由需统一
  - 典型外部网关协议：BGP
  - 自治系统边界路由器ASBR (AS Bounder Router)



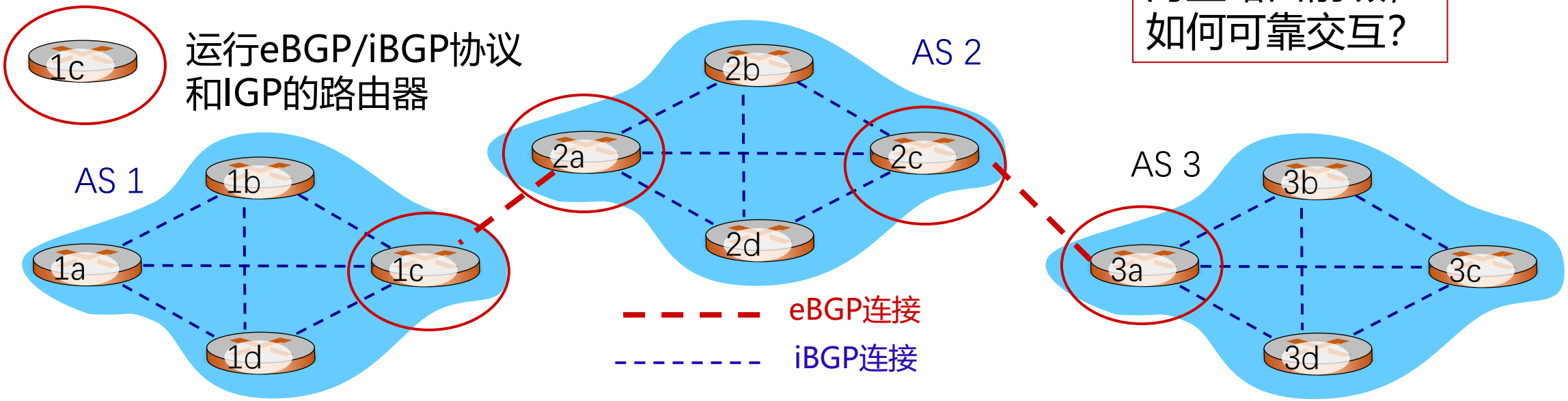
区域边界路由器ABR是跨越区域的，而自治系统边界路由器ASBR则不跨越AS.

每个AS有一个全球唯一的ID号：AS ID

BGP-外部网关路由协议

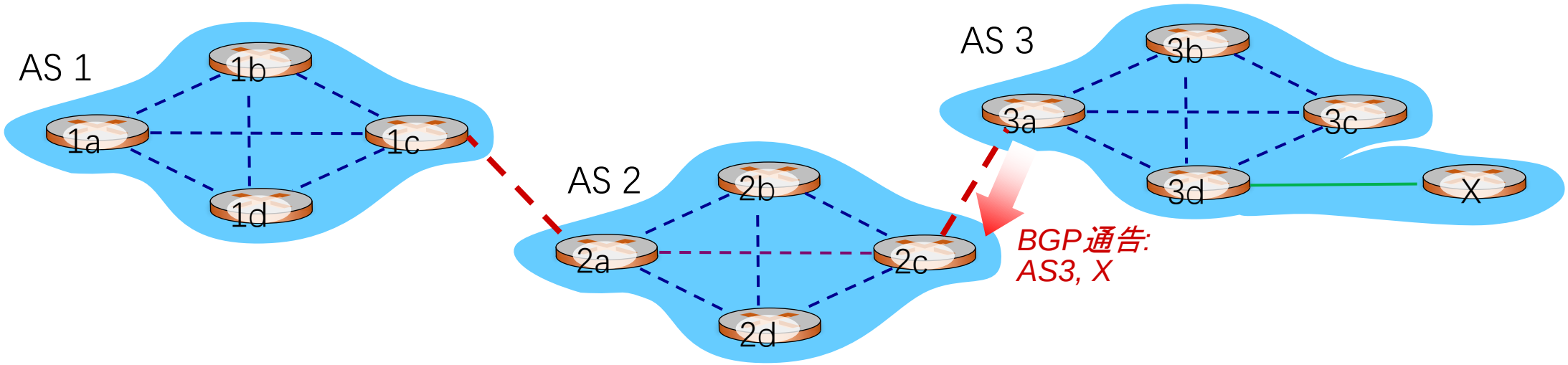
- 边界网关协议BGP (Border Gateway Protocol)
  - 目前互联网中唯一实际运行的自治域间的路由协议
- BGP功能
  - eBGP：从相邻的AS获得网络可达信息；iBGP： 将网络可达信息传播给AS内的路由器
  - 基于网络可达信息和策略决定到其他网络的“最优”路由

海量路由前缀，
如何可靠交互？

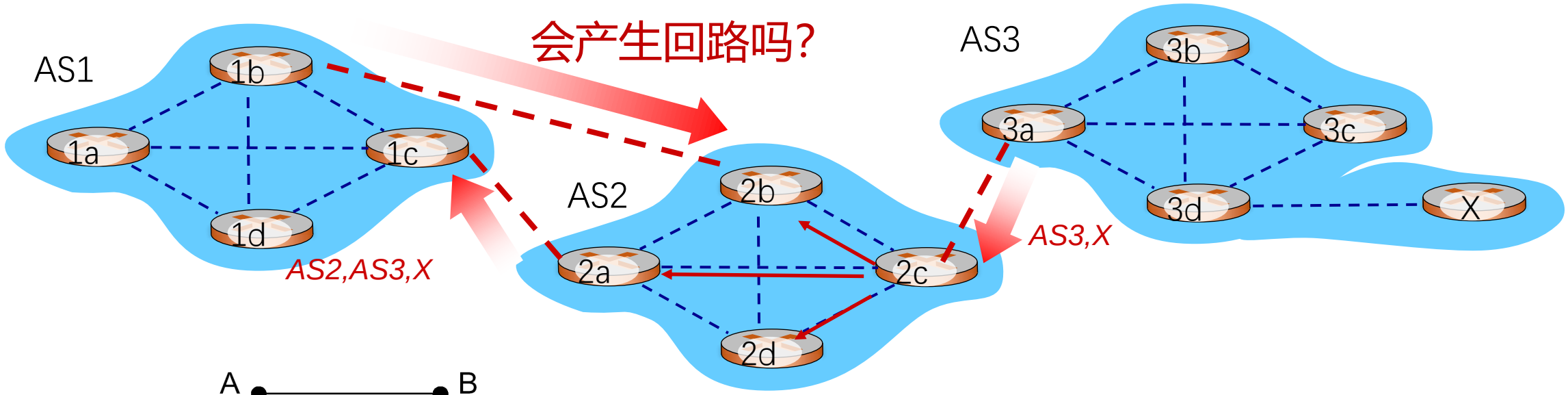


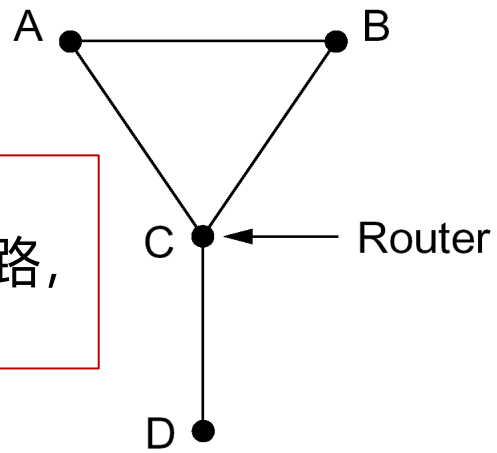
BGP基础

- BGP如何保证路由交互的可靠性?
  - 两个BGP路由器通过TCP连接交换BGP报文
  - 通告到不同网络前缀的路径，即路径向量协议
- 路由通告的意义：提供分组转发服务
  - 当AS3的路由器3a向AS2的路由器2c通告路径AS3的路由X时，AS3向AS2承诺它会向X转发数据包



BGP路径通告

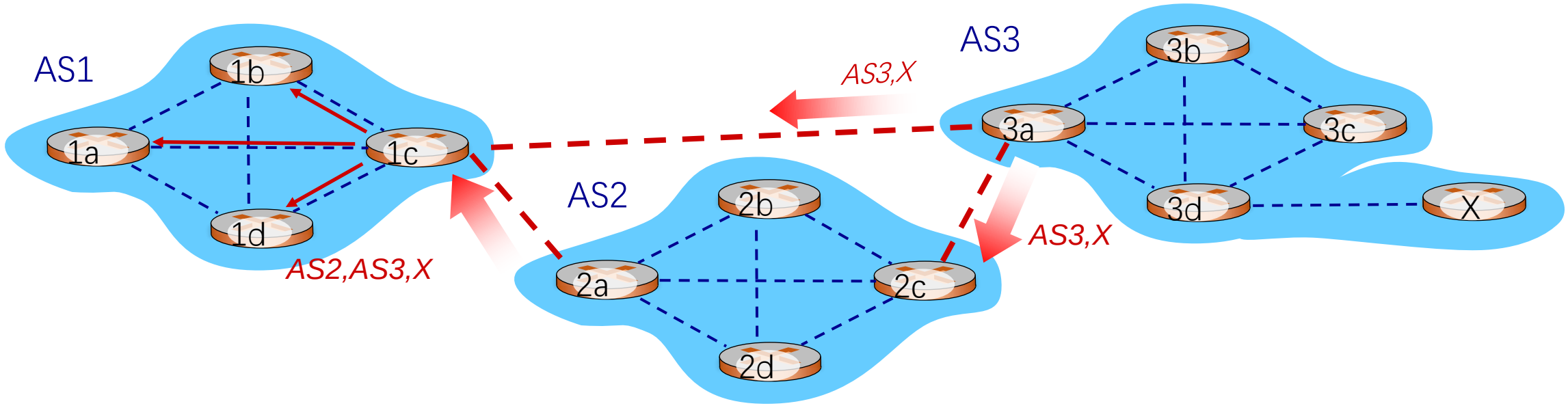




设计BGP
一定要避免回路，
如何避免?

- AS2的路由器2c从AS3的路由器3a接收到路径AS3, X
- 根据AS2的策略，AS2的路由器2c接受路径AS3, X，通过iBGP传播给AS2的所有路由器
- 根据AS2策略，AS2的路由器2a通过eBGP向AS1的路由器1c通告从AS3的路由器3a接收到路径
- 从距离向量升级为路径向量（X：AS1, AS2, AS3；距离3）

BGP路径通告



路由器可能会学到多条到目的网络的路径

- AS1的路由器1c从2a学到路径*AS2, AS3, X*
- AS1的路由器1c从3a学到路径*AS3, X*
- 由策略：AS1路由器1c可能选择路径*AS3, X*，并在AS1中通过iBGP通告路径

最短路径是最短吗？将AS当节点

BGP协议的可扩展性

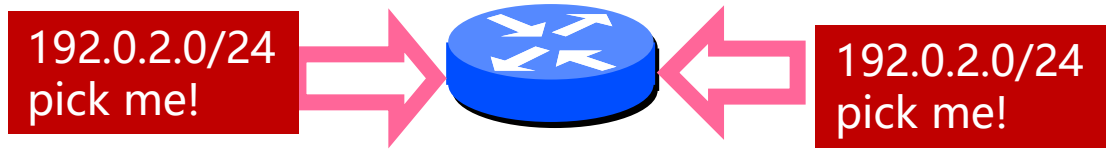
- BGP计算路由的基础是自治系统数目
  - 路径向量（X：AS1, AS2, AS3；距离3）
  - 在 BGP 刚刚运行时，BGP 的邻站是交换整个的 BGP 路由表；以后只需要在发生变化时更新有变化的部分
  - BGP通过TCP的179端口交换报文，实现路由信息传输的可靠性
- BGP为每个AS提供
  - 从邻居AS获取网络可达信息（eBGP协议）
  - 传播可达信息给所有的域内路由器（iBGP协议）
  - 根据“可达信息”和“策略”决定路由

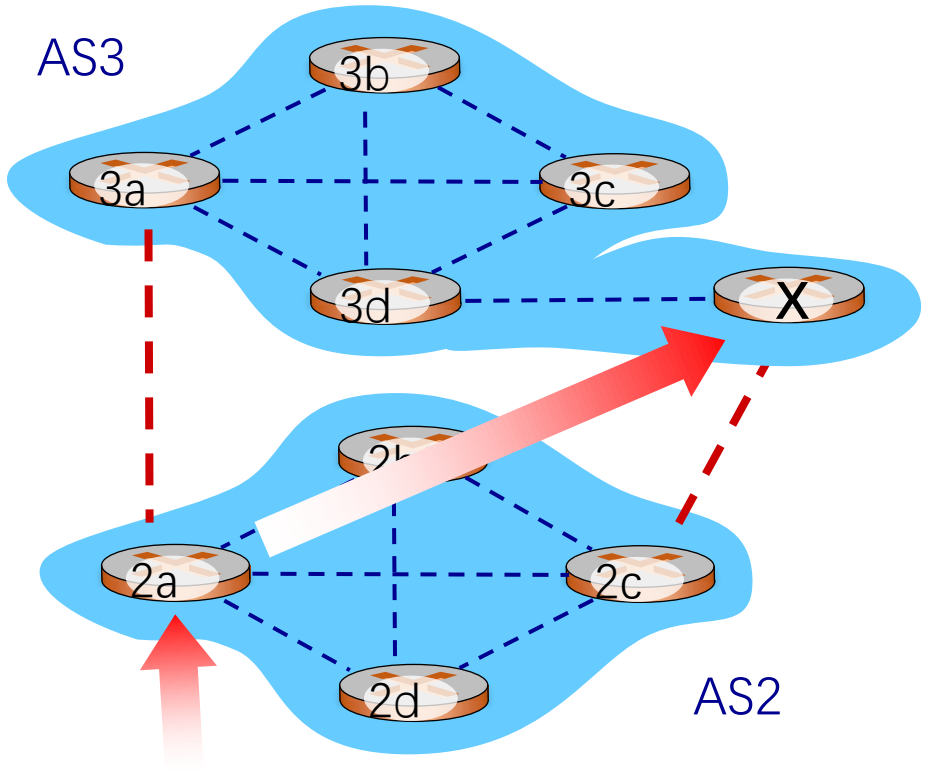
BGP报文与路径属性

- BGP报文类型
  - Open报文：用于建立BGP对等体（peer）之间的会话连接，协商BGP参数（该过程需要认证）
  - Update报文：用于在对等体之间交换路由信息
  - Keepalive报文：用于保持BGP会话连接
  - Notification报文：用于差错报告和关闭BGP连接
- BGP发布的前缀信息包括BGP属性(BGP attributes)
  - 路由“route”=prefix + attributes
- 两个重要属性
  - AS路径（AS-PATH）：IP前缀通过经过的所有AS号，如：AS 67, AS 17
  - 下一跳（NEXT-HOP）：说明路由信息对应的下一跳IP地址
- 路由器接收到路由通告时，通过既定策略采纳或拒绝

BGP 路由选择

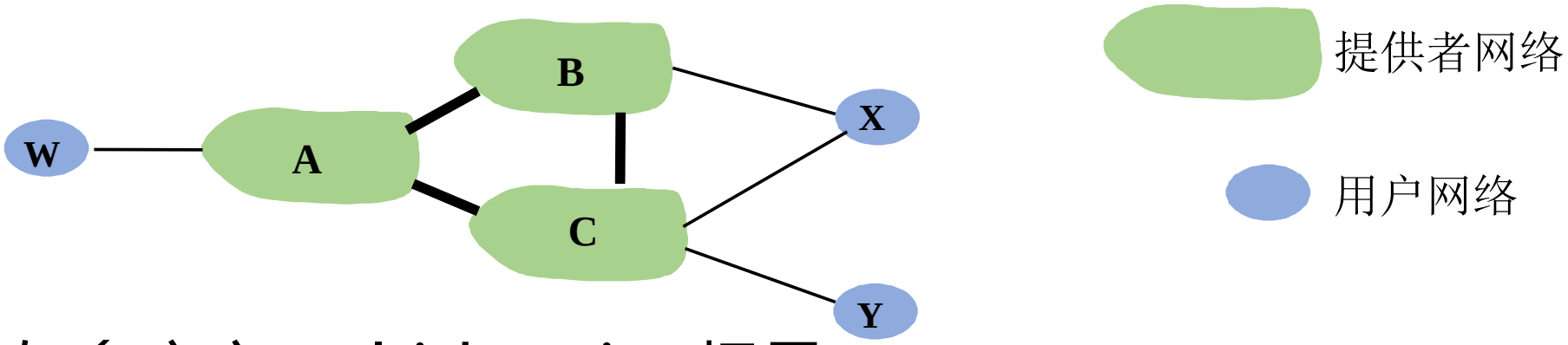
- 最佳路由选择
  - 路由器可能从多个对等体收到针对同一目的IP的路由
- 最佳路由选择规则
  - 本地偏好值属性：政策决策
  - 最短的AS-PATH
  - 最近的NEXT-HOP路由器
  - 附加标准...
  - 最低路由器ID
- 热土豆问题





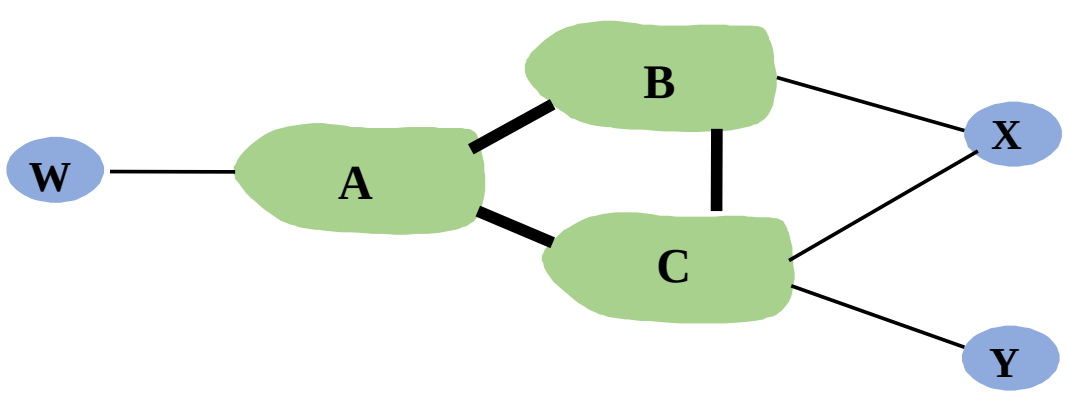
BGP路由策略

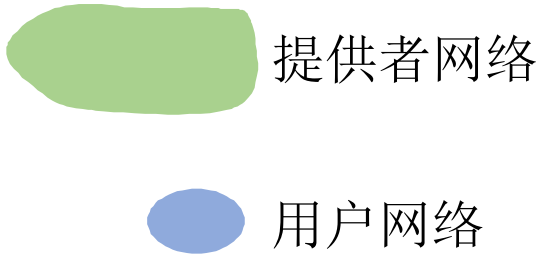
- BGP策略
  - 路由器使用策略决定接受或拒绝接收到的路由通告
  - 路由器也会基于策略决定是否向其他相邻AS通告路径信息



- 如多宿主multi-homing场景
  - X连接到两个提供者网络
  - X为用户网络，X不希望从B到C的数据包经过X，怎么办？
  - X则不向B通告到C的路由

BGP路由策略





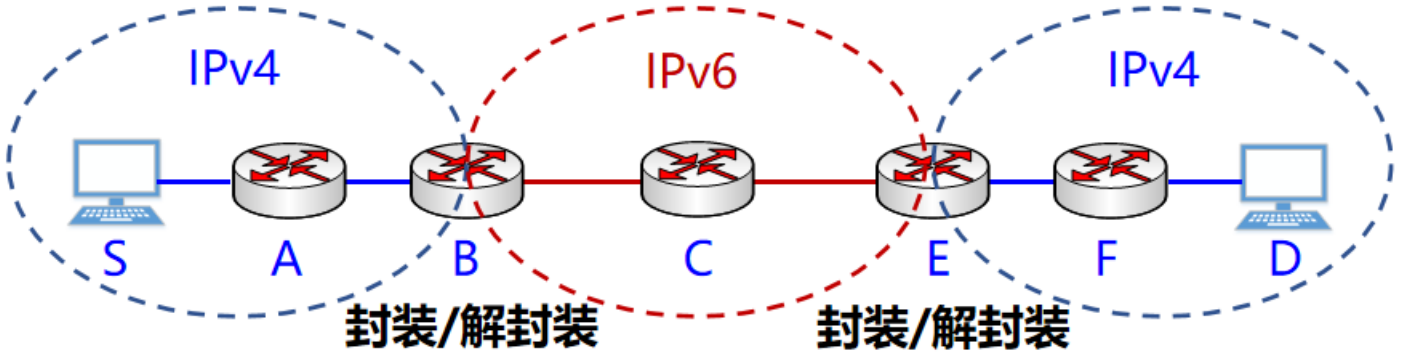
- A向B通告路径*AW*
- B向X通告到目的W的路径为*BAW*
- B是否向C通告路径*BAW*？
  - 由于W和C都不是B的用户，B要迫使C通过A路由到W
  - B只路由（服务）来自于或到达其**用户**的数据包

总结

- 路由协议的基本需求
  - 控制平面：路由协议产生路由表
  - 数据平面：基于路由表进行分组转发
  - 高效收敛且无回路，尽量简单，可扩展，支持多种策略
- 思考与发明：RIP->OSPF->BGP->MPLS
  - RIP：距离向量，邻居交互最优下一跳，分布式计算；收敛速度较慢
  - OSPF：链路状态，洪泛原始信息，集中计算最短路；计算开销大
  - BGP：层次路由 + 邻居交互路径向量；支撑全球路由，结合商业运营

下周预告

- 路由器体系结构
  - 发明了好路由协议，多个路由协议如何相互配合？
  - 如何实现为硬件设备
- IPv4地址耗尽，IPv6是网络层的未来
  - 抛弃现有软硬件？设计共存机制？
- 在尽力而为的IP上，能实现服务质量保证吗？



作业

- 《Computer Networks-5th Edition》章节末习题
  - CHAPTER 5：31（路由表更新）
  - 补充习题（见下页）
  - 小实验3（IP数据包格式观察，见网络学堂附件）
  - 思考：RIP、OSPF和BGP分别采用什么下层协议，为什么这样设计?
- 截止时间：下下周三晚11:59，提交网络学堂

作业

02. 在某个使用 RIP 的网络中，B 和 C 互为相邻路由器，其中表 1 为 B 的原路由表，表 2 为 C 广播的距离向量报文<目的网络，距离>。

表 1

目的网络	距离	下一跳
N1	7	A
N2	2	C
N6	8	F
N8	4	E
N9	4	D

表 2

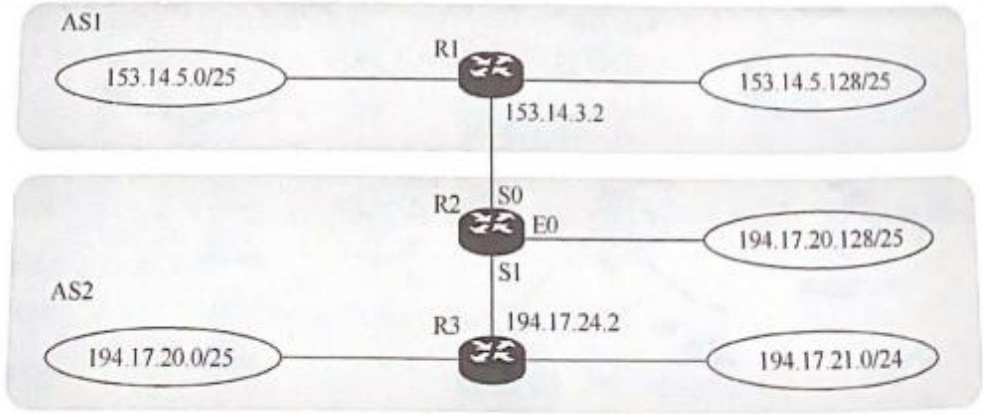
目的网络	距离
N2	15
N3	2
N4	8
N8	2
N7	4

1）试求路由器 B 更新后的路由表并说明主要步骤。

2）当路由器 B 收到发往网络 N2 的 IP 分组时，应该做何处理？

作业

04. 【2013 统考真题】假设 Internet 的两个自治系统构成的网络如下图所示，自治系统 AS1 由路由器 R1 连接两个子网构成；自治系统 AS2 由路由器 R2、R3 互联并连接 3 个子网构成。各子网地址、R2 的接口名、R1 与 R3 的部分接口 IP 地址如下图所示。



请回答下列问题：

1）假设路由表结构如下表所示。利用路由聚合技术，给出 R2 的路由表，要求包括到达图中所有子网的路由，且路由表中的路由项尽可能少。

目的网络	下一跳	接口

2）若 R2 收到一个目的 IP 地址为 194.17.20.200 的 IP 分组，R2 会通过哪个接口转发该 IP 分组？

3）R1 与 R2 之间利用哪个路由协议交换路由信息？该路由协议的报文被封装到哪个协议的分组中进行传输？

致谢社区本章贡献者

贡献者姓名	单 位	贡献内容
陈文龙	首都师范大学	本章统稿 5.5 5.9(IPv6协议)
吴黎兵	武汉大学	5.6 5.7
谢晓燕	西安邮电大学	5.8
邹莹	仲恺农业工程学院	5.4.1 5.4.2 5.4.4
李旭宏	枣庄学院	5.1.2 5.1.3 5.2.3 5.2.4 5.2.6 5.4.3
曲大鹏	辽宁大学	5.3.1 5.3.2
方诗虹	西南民族大学	5.1.4 5.3.5 5.3.6 5.3.7 5.3.8
舒挺	浙江理工大学	5.1.1
白云莉	内蒙古农业大学	5.3.3 5.3.4
余琨	荆楚理工学院	5.2.1 5.2.2 5.2.5
李振斌	华为技术有限公司	5.9(SRv6)

致谢社区本章贡献者

陈文龙

首都师范大学

5.路由器工作原理
9.IPv6技术

吴黎兵

武汉大学

6.拥塞控制算法
7.服务质量

谢晓燕

西安邮电大学

8.三层交换和VPN

邹莹

仲恺农业工程学院

4.Internet路由协议

李旭宏

枣庄学院

1.网络层服务
2.Internet网际协议
4.Internet路由协议

《计算机网络：自顶向下方法》(原书第7版)，库罗斯 罗斯，机械工业出版社，2018年06月
《计算机网络（第5版）》，Tanenbaum & Wetherall，清华大学出版社，2012年3月
《计算机网络（第7版）》，谢希仁，电子工业出版社，2017年01月
《计算机网络教程（第6版）》，吴功宜，电子工业出版社，2018年03月
《计算机网络（第3版）》，徐敬东、张建忠，清华大学出版社，2013年6月1日

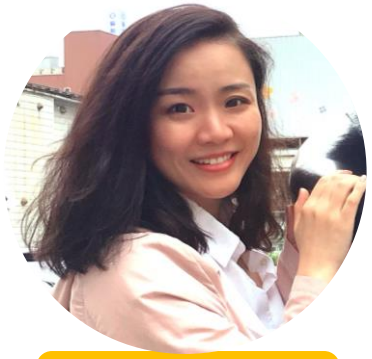
特别致谢：
部分内容取材于此

致谢社区本章贡献者

曲大鹏

辽宁大学

3.路由算法



方诗虹

西南民族大学

1.网络层服务
3.路由算法

舒挺

浙江理工大学

1.网络层服务

白云莉

内蒙古农业大学

3.路由算法

余琨

荆楚理工学院

2.Internet网际协议

李振斌

华为技术公司

9.IPv6技术

《计算机网络：自顶向下方法》(原书第7版)，库罗斯 罗斯，机械工业出版社，2018年06月
《计算机网络（第5版）》，Tanenbaum & Wetherall，清华大学出版社，2012年3月
《计算机网络（第7版）》，谢希仁，电子工业出版社，2017年01月
《计算机网络教程（第6版）》，吴功宜，电子工业出版社，2018年03月
《计算机网络（第3版）》，徐敬东、张建忠，清华大学出版社，2013年6月1日

特别致谢：
部分内容取材于此