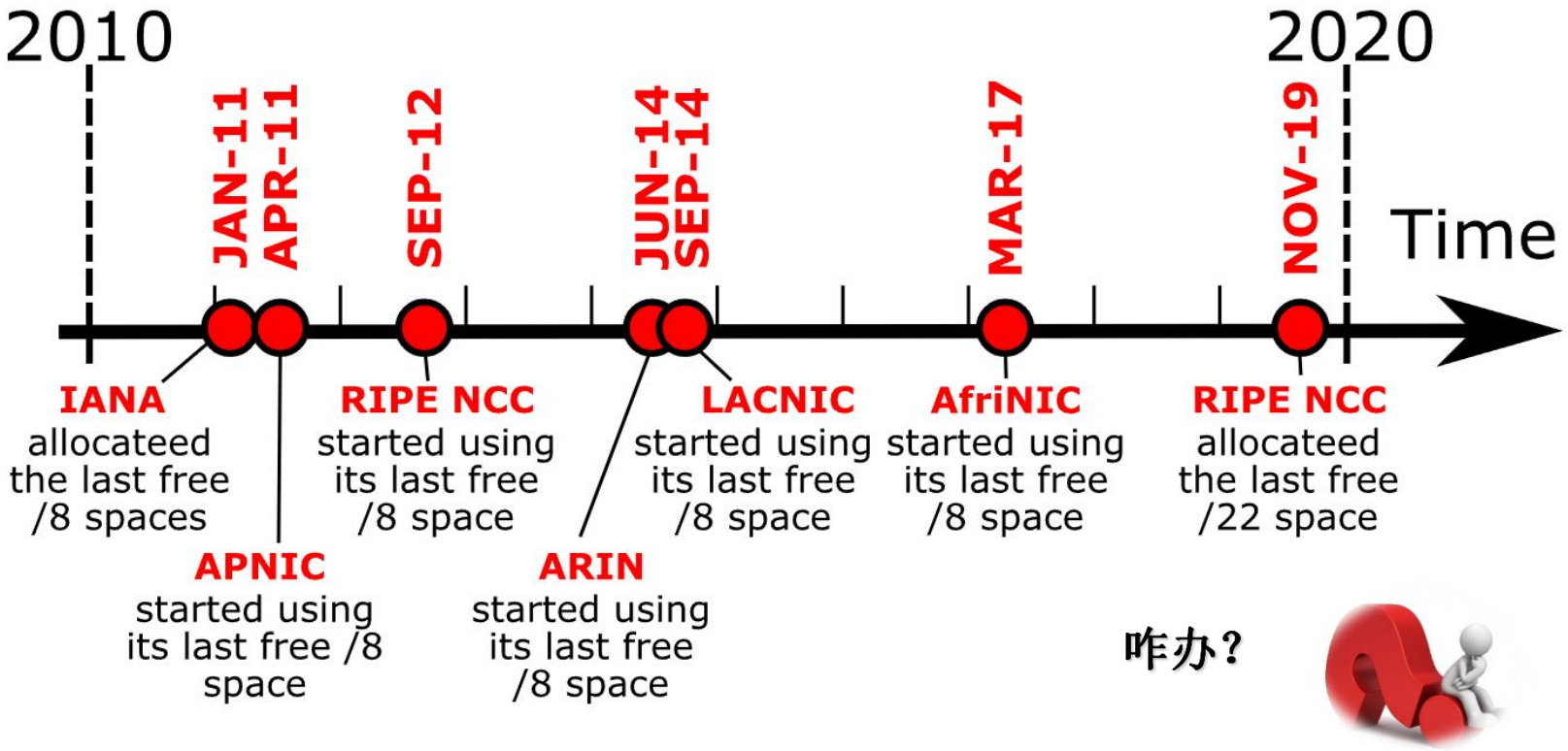
中南大學
CENTRAL SOUTH UNIVERSITY

第五章 网络层

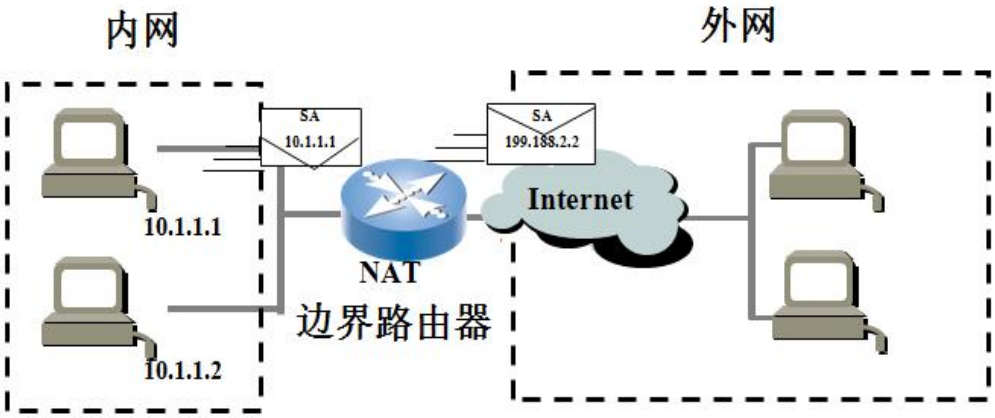
IPv4地址池耗尽



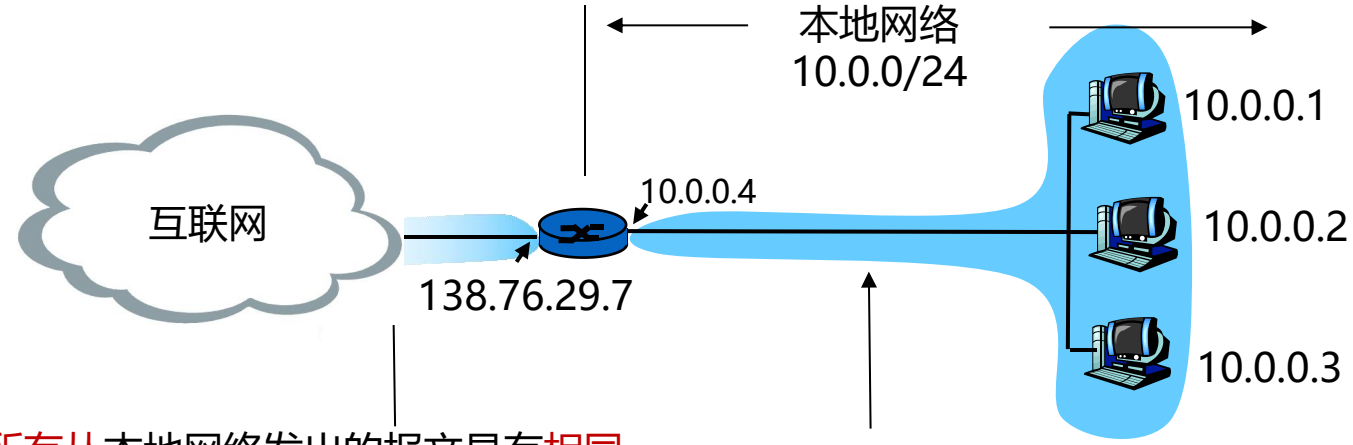
来源：IPv4 address exhaustion
https://en.wikipedia.org/wiki/IPv4_address_exhaustion

网络地址转换

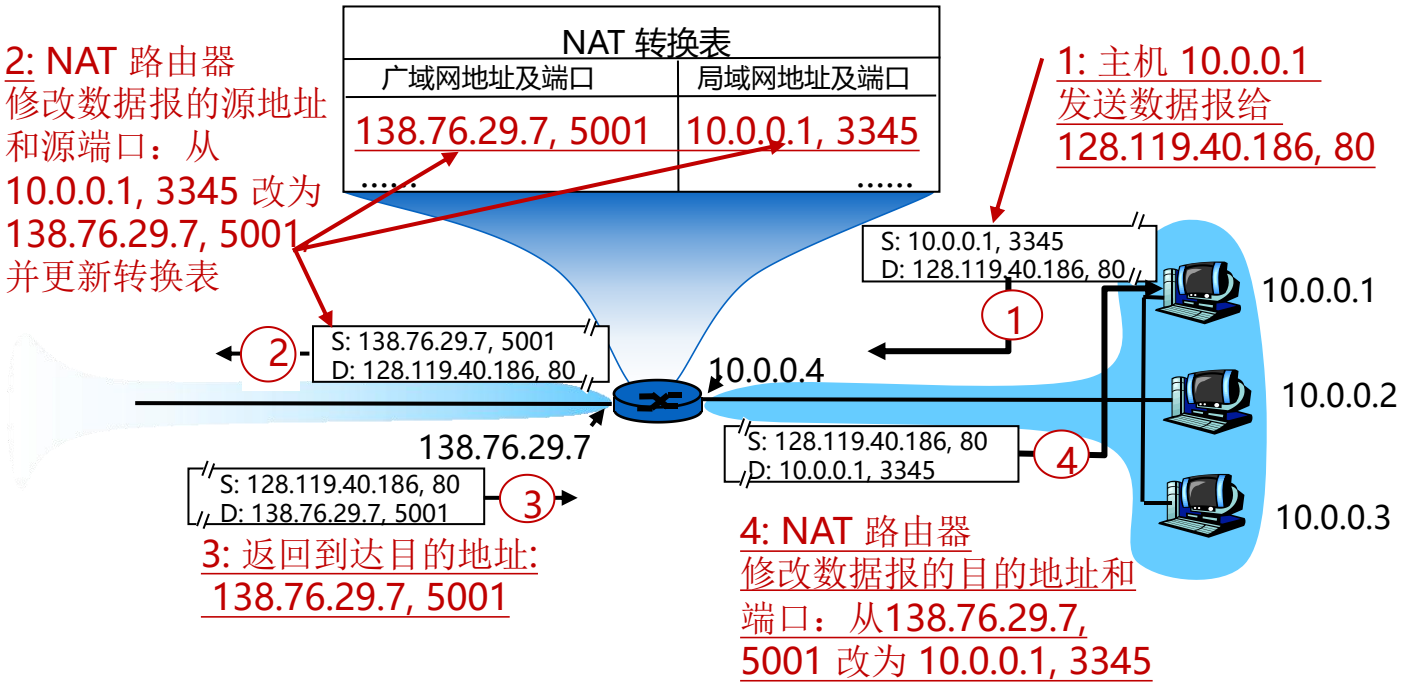
- 网络地址转换(NAT)用于解决IPv4地址不足的问题，是一种将私有（保留）地址转化为公有IP地址的转换技术
- 私有IP地址：
  - A类地址：10.0.0.0--10.255.255.255
  - B类地址：172.16.0.0--172.31.255.555
  - C类地址：192.168.0.0--192.168.255.255



NAT工作机制

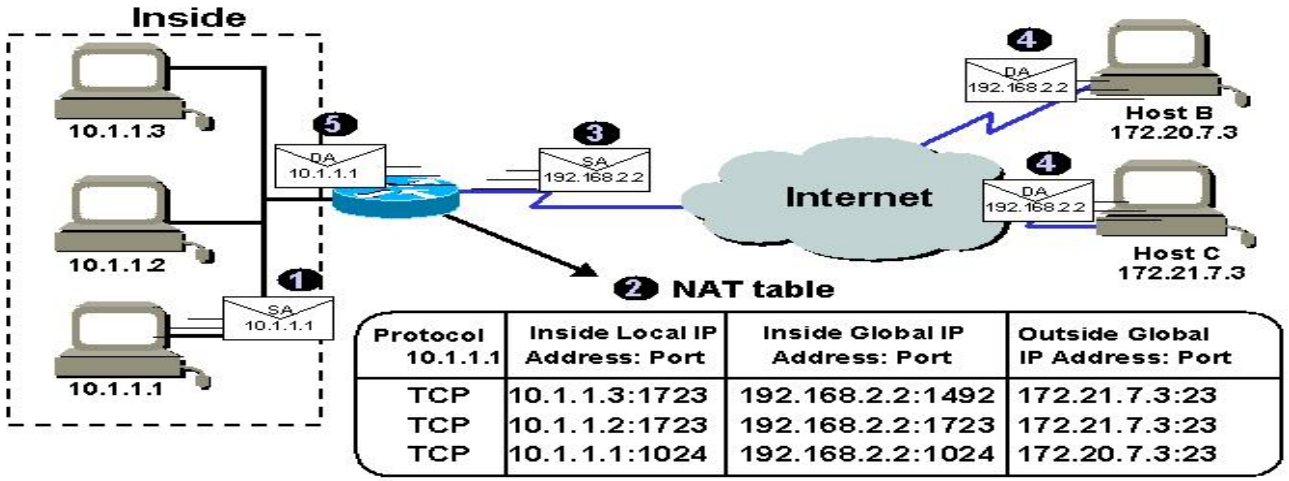

本地网络
10.0.0/24
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
互联网
138.76.29.7
所有从本地网络发出的报文具有相同的单一源IP地址: 138.76.29.7
报文的源地址(发出时)或目的地址(接收时)都属于10.0.0.0/24的地址
不同的源端口号

NAT工作机制


NAT 转换表
广域网地址及端口
局域网地址及端口
138.76.29.7, 5001
10.0.0.1, 3345
……
……
2: NAT 路由器
修改数据报的源地址
和源端口：从
10.0.0.1, 3345 改为
138.76.29.7, 5001
并更新转换表
1: 主机 10.0.0.1
发送数据报给
128.119.40.186, 80
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
10.0.0.4
138.76.29.7
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
4
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
3
10.0.0.1
10.0.0.2
10.0.0.3
3: 返回到达目的地址:
138.76.29.7, 5001
4: NAT 路由器
修改数据报的目的地址和
端口：从138.76.29.7,
5001 改为 10.0.0.1, 3345

NAT工作机制

- 出数据报：外出数据报用 NAT IP地址(全局), 新port # 替代 源IP地址(私有), port #
- NAT转换表：每个 (源IP地址, port #)到(NAT IP地址, 新port #) 映射项
- 入数据报：对每个入数据报的地址字段用存储在NAT表中的(源IP地址, port #)替代对应的 (NAT IP地址, 新port #)



Protocol 10.1.1.1	Inside Local IP Address: Port	Inside Global IP Address: Port	Outside Global IP Address: Port
TCP	10.1.1.3:1723	192.168.2.2:1492	172.21.7.3:23
TCP	10.1.1.2:1723	192.168.2.2:1723	172.21.7.3:23
TCP	10.1.1.1:1024	192.168.2.2:1024	172.20.7.3:23

网络地址转换

- NAT根据不同的IP上层协议进行NAT表项管理
  - TCP，UDP，ICMP
- 传输层TCP/UDP拥有16-bit 端口号字段
  - 所以一个WAN侧地址可支持60,000个并行连接

➢ NAT的优势
- 节省合法地址，减少地址冲突
- 灵活连接Internet
- 保护局域网的私密性

➢ 问题或缺点
- 违反了IP的结构模型，路由器处理传输层协议
- 违反了端到端的原则
- 违反了最基本的协议分层规则

IPv4地址如何获取

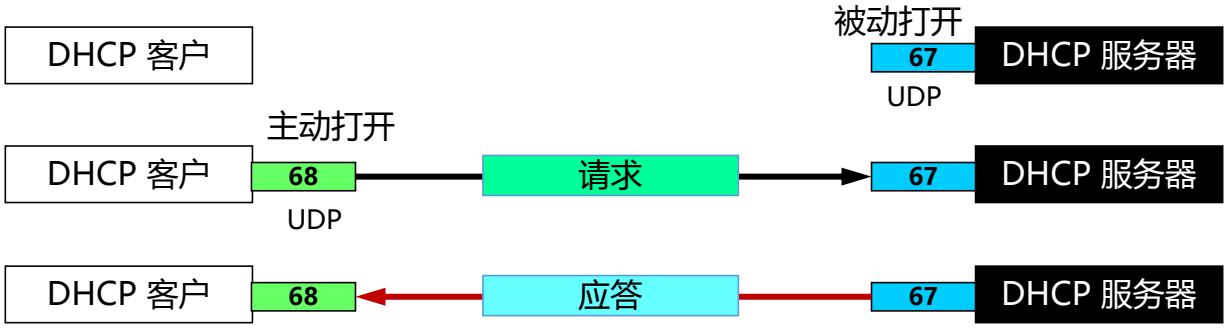
- **公有IP地址要求全球唯一**
  - ICANN（Internet Corporation for Assigned Names and Numbers）即互联网名字与编号分配机构向ISP分配，ISP再向所属机构或组织逐级分配
- **静态设定**
  - 申请固定IP地址，手工设定，如路由器、服务器
- **动态获取**
  - 使用DHCP协议或其他动态配置协议
  - 当主机加入IP网络，允许主机从DHCP服务器动态获取IP地址
  - 可以有效利用IP地址，方便移动主机的地址获取

DHCP动态主机配置协议

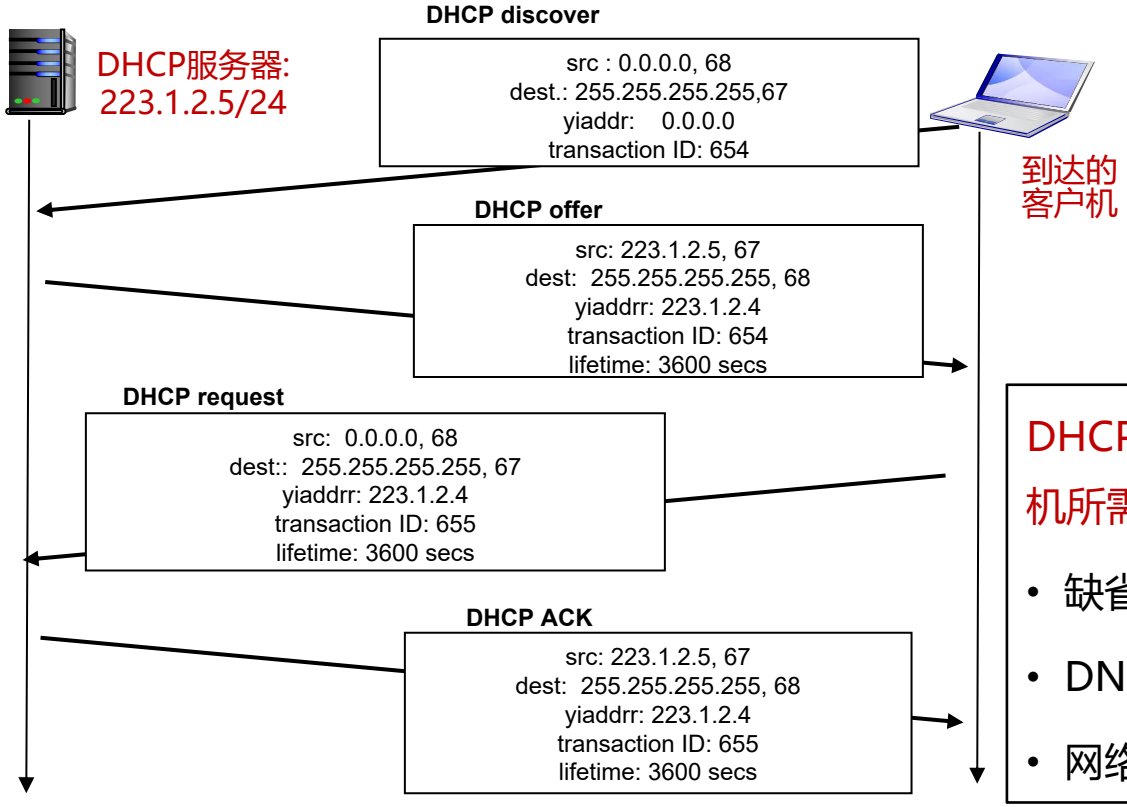
- DHCP：动态主机配置协议
  - 当主机加入IP网络，允许主机从DHCP服务器动态获取IP地址
  - 可以有效利用IP地址，方便移动主机的地址获取

➢ 工作模式：客服/服务器模式（C/S）

- 基于 UDP 工作，服务器运行在 67 号端口， 客户端运行在 68 号端口



DHCP 工作过程



DHCP服务不只返回客户机所需的IP地址，还包括:

- 缺省路由器IP地址
- DNS服务器IP地址
- 网络掩码

DHCP 工作过程

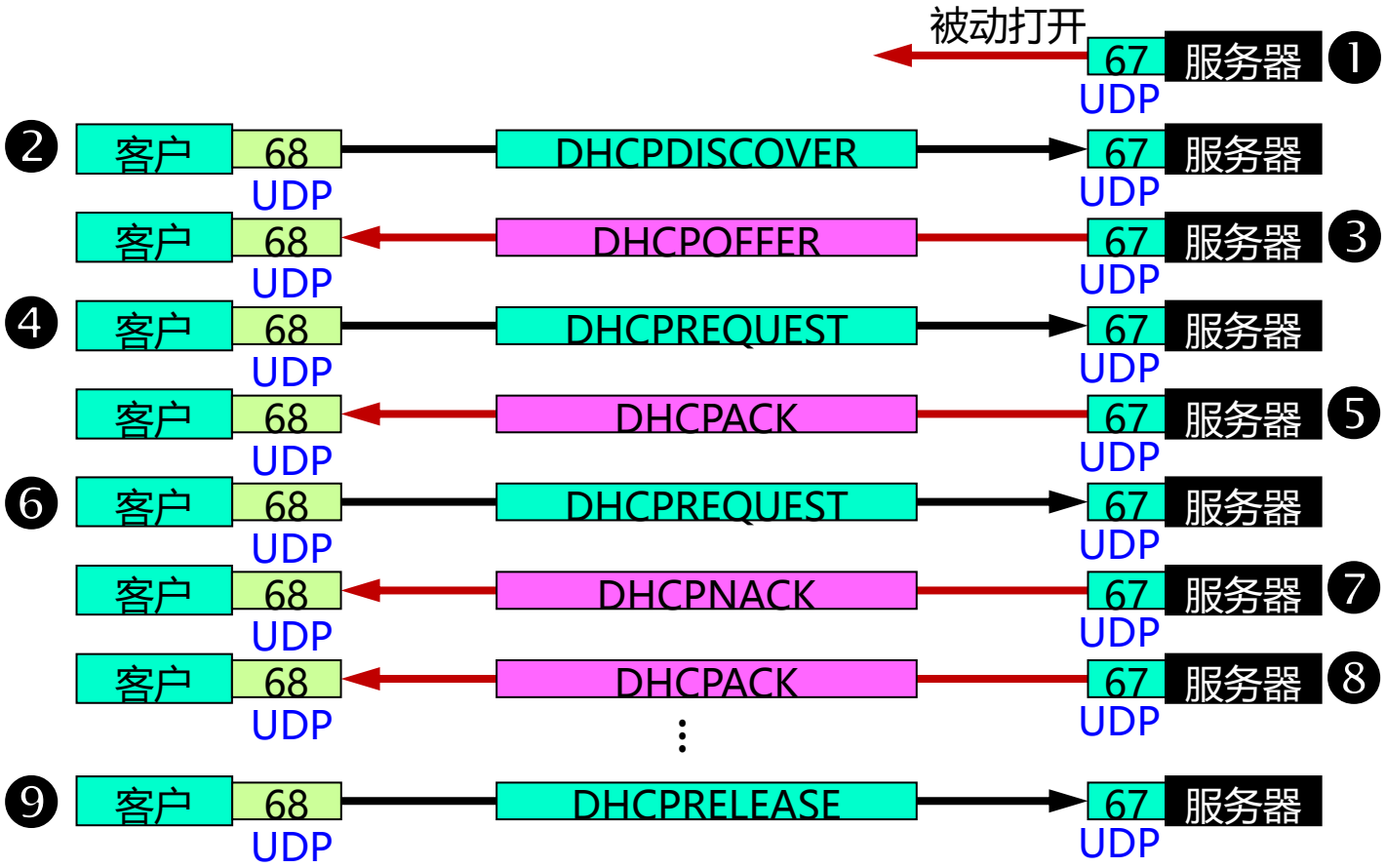
- DHCP 客户从UDP端口68以广播形式向服务器发送发现报文（DHCPDISCOVER）
- DHCP 服务器单播发出提供报文（DHCPOFFER）
- DHCP 客户从多个DHCP服务器中选择一个，并向其以广播形式发送DHCP请求报文（DHCPREQUEST）
- 被选择的DHCP服务器单播发送确认报文（DHCPACK）

DHCP 工作过程

- 小结

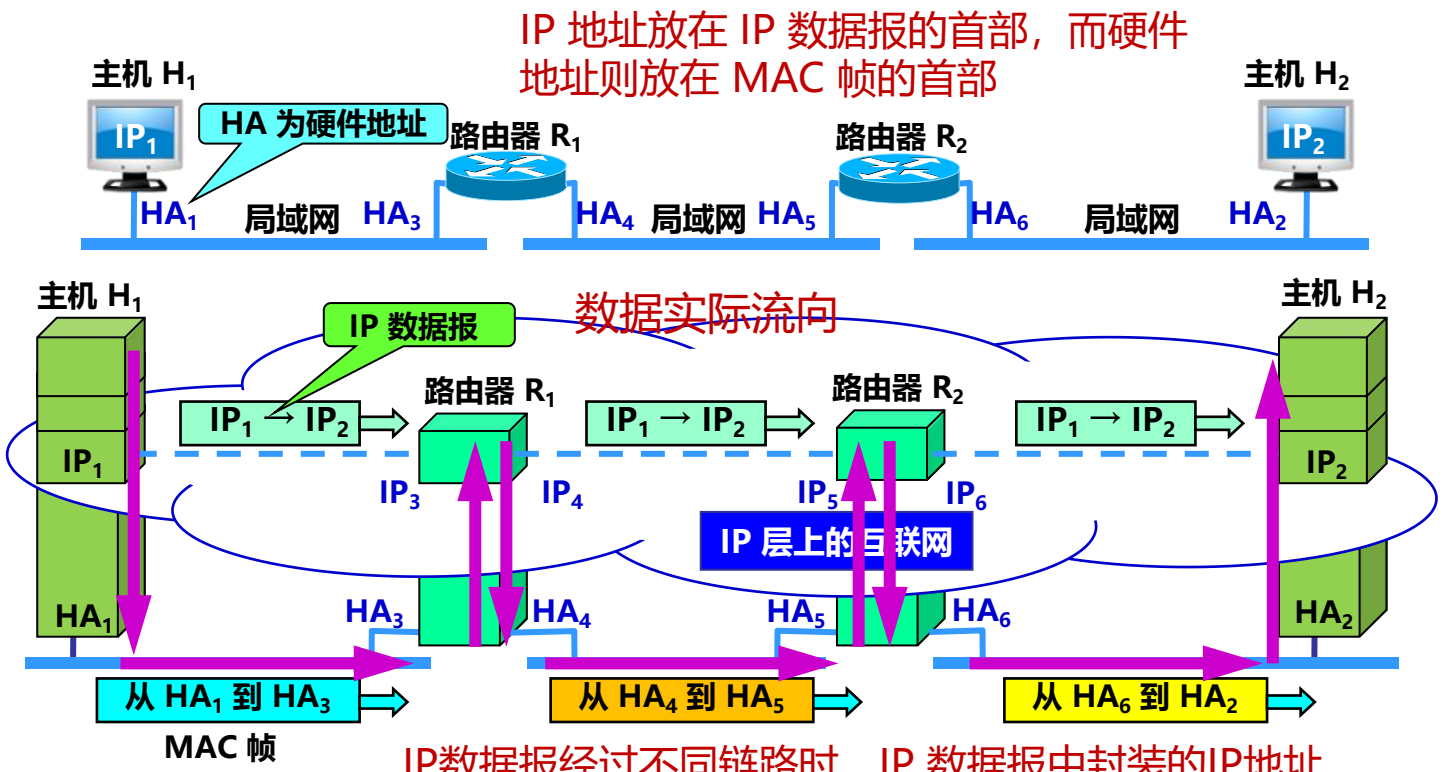
阶段	源MAC	目标MAC	源IP	目标IP	传输形式
Discover	PC机的MAC	全FF	0.0.0.0	255.255.255.255	广播
Offer	DHCP服务器或者中继器路由的MAC	DHCP客户机的MAC	DHCP服务器或中继路由器的IP地址	255.255.255.255	单播
Request	PC机的MAC	全FF	0.0.0.0	255.255.255.255	广播
Ack	DHCP服务器或者中继器路由的MAC	DHCP客户机的MAC	DHCP服务器或中继路由器的IP地址	255.255.255.255	单播

DHCP 完整工作过程



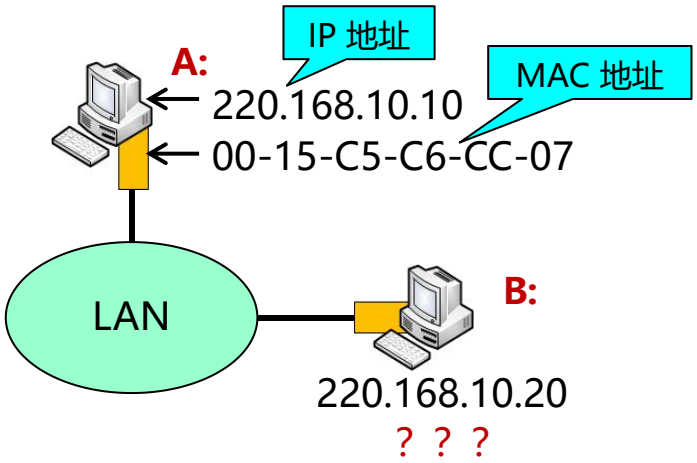
注：说明见备注

IP 与 MAC地址

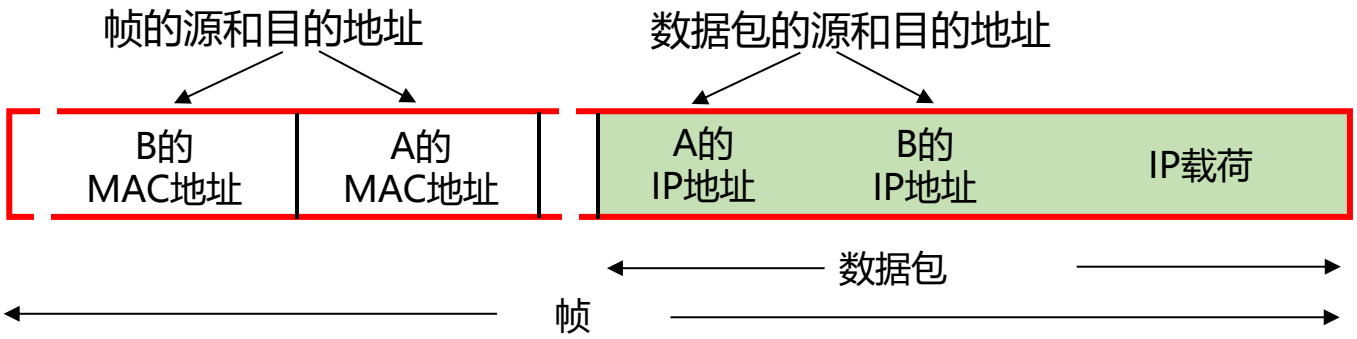

IP 地址放在 IP 数据报的首部，而硬件地址则放在 MAC 帧的首部
主机 H1
主机 H2
IP1
IP2
HA 为硬件地址
路由器 R1
路由器 R2
HA1
局域网
HA3
HA4
局域网
HA5
HA6
局域网
HA2
数据实际流向
IP 数据报
IP1 → IP2
IP1 → IP2
IP1 → IP2
IP3
IP4
IP5
IP6
IP 层上的互联网
从 HA1 到 HA3
从 HA4 到 HA5
从 HA6 到 HA2
MAC 帧
IP数据报经过不同链路时，IP 数据报中封装的IP地址不发生改变，而Mac帧中的硬件地址是发生改变的

ARP地址解析协议

- IP数据包转发：从主机A到主机B
  - 检查目的IP地址的网络号部分
  - 确定主机B与主机A属相同IP网络
  - 将IP数据包封装到链路层帧中，直接发送给主机B



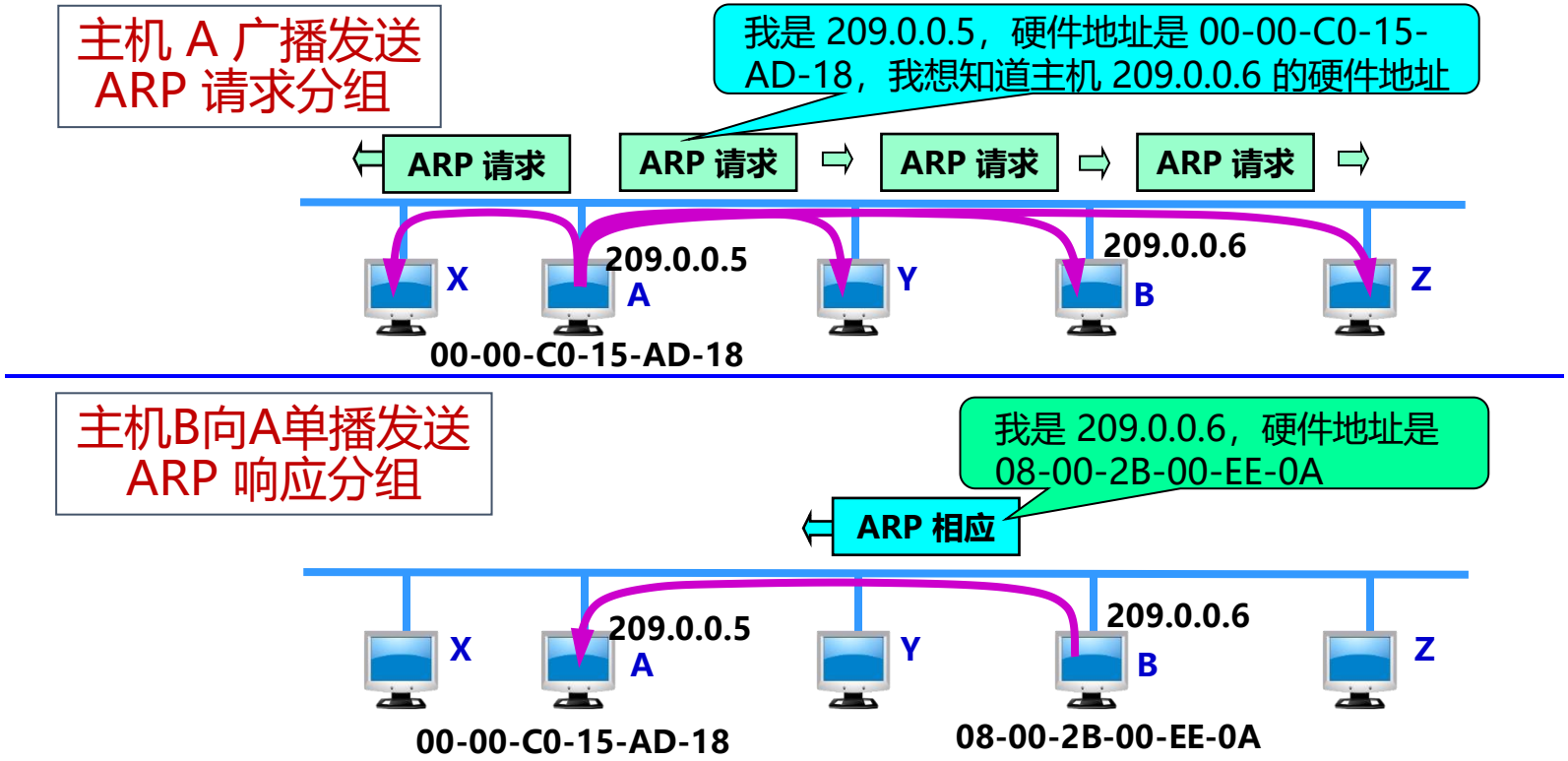
问题：给定B的IP地址，如何获取B的MAC地址?



ARP地址解析协议

- A已知B的IP地址，需要获得B的MAC地址（物理地址）
- 如果A的ARP表中缓存有B的IP地址与MAC地址的映射关系，则直接从ARP表获取
- 如果A的ARP表中未缓存有B的IP地址与MAC地址的映射关系，则A广播包含B的IP地址的ARP query分组
  - 在局域网上的所有节点都可以接收到ARP query
- B接收到ARP query分组后，将自己的MAC地址发送给A
- A在ARP表中缓存B的IP地址和MAC地址的映射关系
  - 超时时删除

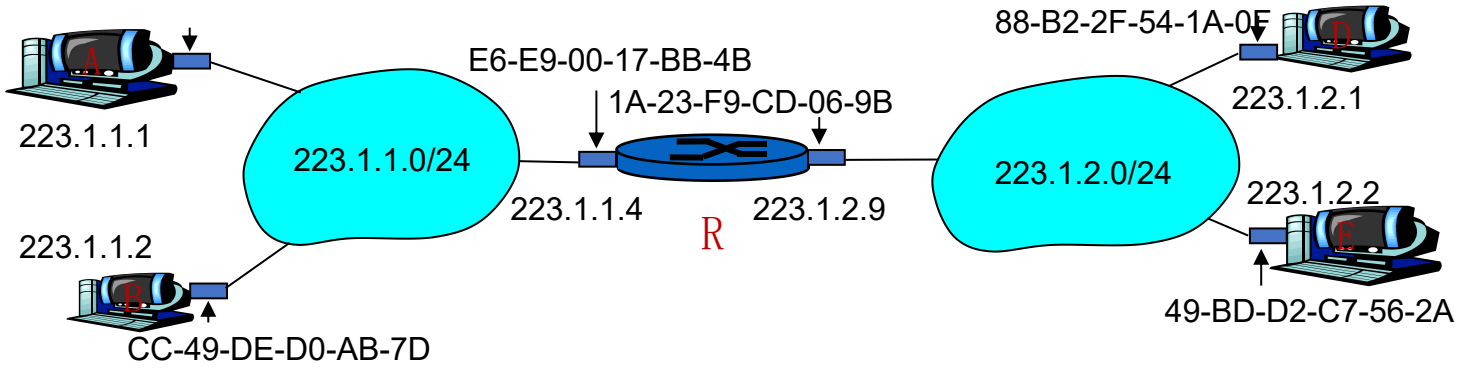
ARP协议工作过程


主机 A 广播发送
ARP 请求分组
我是 209.0.0.5，硬件地址是 00-00-C0-15-AD-18，我想知道主机 209.0.0.6 的硬件地址
ARP 请求
ARP 请求
ARP 请求
ARP 请求
X
209.0.0.5
A
Y
209.0.0.6
B
Z
00-00-C0-15-AD-18
主机B向A单播发送
ARP 响应分组
我是 209.0.0.6，硬件地址是 08-00-2B-00-EE-0A
ARP 相应
X
209.0.0.5
A
Y
209.0.0.6
B
Z
00-00-C0-15-AD-18
08-00-2B-00-EE-0A

路由到另一个局域网

- A创建IP数据包（源为A、目的为E）
- 在源主机A的路由表中找到路由器R的IP地址223.1.1.4
- A根据R的IP地址223.1.1.4，使用ARP协议获得R的MAC地址
- A创建数据帧（目的地址为R的MAC地址）
- 数据帧中封装A到E的IP数据包
- A发送数据帧，R接收数据帧

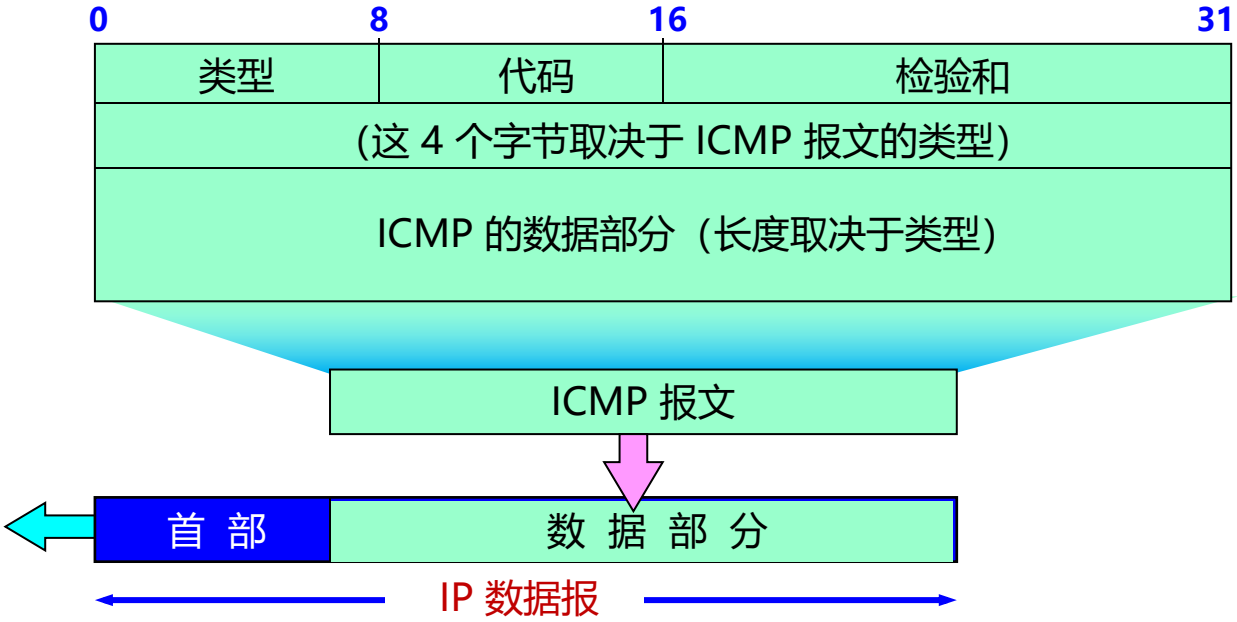
例：从A经过R到E



ICMP协议

- ICMP: 互联网控制报文协议
  - ICMP 允许主机或路由器报告差错情况和提供有关异常情况的报告
  - 由主机和路由器用于网络层信息的通信
  - ICMP 报文携带在IP 数据报中： IP上层协议号为1
- ICMP报文类型
  - ICMP 差错报告报文
    - 终点不可达：不可达主机、不可达网络，无效端口、协议
  - ICMP 询问报文
    - 回送请求/回答 (ping使用)

ICMP 报文格式



- ICMP报文的前 4 个字节包含格式统一的三个字段：类型、代码、检验和
- 相邻的后四个字节内容与ICMP的报文类型有关

ICMP报文类型及功能

ICMP报文类型	类型值	功能描述
差错报告报文	3	终点不可达
	5	改变路由(Redirect)
	11	时间超时
	12	参数问题
询问报文	8或0	回
	13或14	时间戳

类型	代码	功能描述
0	0	回送应答 (ping)
3	0	目的网络不可达
3	1	目的主机不可达
3	2	目的协议不可达
3	3	目的端口不可达
3	6	目的网络未知
3	7	目的主机未知
8	0	回送请求 (ping)
9	0	路由通告
10	0	路由发现
11	0	TTL过期
12	0	坏的IP首部

注：根据RFC 6633（2012年5月）规定，已不再使用类型值为4、10或9、15或16、17或18的报文。

Ping和ICMP

- PING (Packet InterNet Groper)
  - PING 用来测试两个主机之间的连通性
  - PING 使用了 ICMP 回送请求与回送回答报文

```
C:\Users\DELL>ping bing.com

正在 Ping bing.com [13.107.21.200] 具有 32 字节的数据:
来自 13.107.21.200 的回复: 字节=32 时间=59ms TTL=111
来自 13.107.21.200 的回复: 字节=32 时间=60ms TTL=111
来自 13.107.21.200 的回复: 字节=32 时间=60ms TTL=111
来自 13.107.21.200 的回复: 字节=32 时间=60ms TTL=111

13.107.21.200 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 59ms, 最长 = 60ms, 平均 = 59ms
```

连通性　　往返时延　　单向转发跳数

思考：

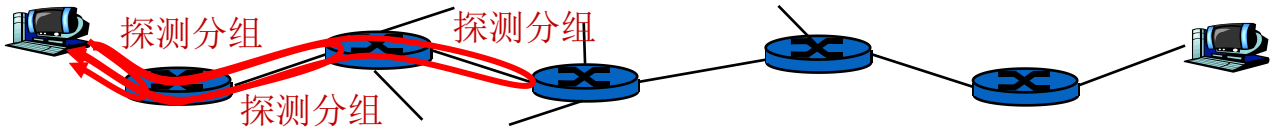
如何利用Ping命令返回的TTL值(报文剩余跳数)，来判断对方主机操作系统的类型?

默认操作系统的TTL值:

1、WINDOWS NT/2000　TTL：128
2、WINDOWS 95/98　TTL：32
3、UNIX　TTL：255
4、LINUX　TTL：64
5、WIN7　TTL：64
6、WIN10　TTL：128

Traceroute和ICMP

- 如何知道整个路径上路由器的地址？ 使用TraceRT命令



- 源向目的地发送一系列UDP段(不可能的端口号)
  - 第一个 TTL =1
  - 第二个 TTL=2, 等
- 当第n个数据报到达第n和路由器:
  - 路由器丢弃数据报
  - 并向源发送一个ICMP报文 (类型 11, 编码0)
  - 报文的源IP地址就是该路由器的IP地址

Traceroute和ICMP

C:\>tracert www.taobao.com

通过最多 30 个跃点跟踪
到 www.taobao.com.danuoyi.tbcache.com
[27.211.197.171] 的路由:

1	3 ms	2 ms	4 ms	192.168.3.1
2	3 ms	6 ms	3 ms	SMBSHARE [192.168.1.1]
3	6 ms	9 ms	5 ms	27.215.136.1
4	6 ms	11 ms	7 ms	61.162.199.89
5	7 ms	18 ms	21 ms	61.162.199.9
6	23 ms	29 ms	22 ms	61.156.223.69
7	28 ms	31 ms	20 ms	112.230.160.54
8	19 ms	27 ms	36 ms	60.208.64.230
9	15 ms	15 ms	16 ms	119.164.254.86
10	19 ms	13 ms	13 ms	27.211.197.171

跟踪完成。

- 当源收到ICMP报文，计算 RTT
- Tracert针对同一RTT值执行上述过程3次

停止条件

- UDP段最终到达目的地主机
- 目的地返回ICMP “端口不可达”分组 (类型3, 编码3)
- 当源得到该ICMP，停止

本章内容

5.1 网络层服务

5.2 Internet 网际协议

5.3 路由算法

5.4 Internet路由协议

5.5 路由器工作原理

5.6 拥塞控制算法

5.7 服务质量

5.8 三层交换与VPN

5.9 IPv6技术

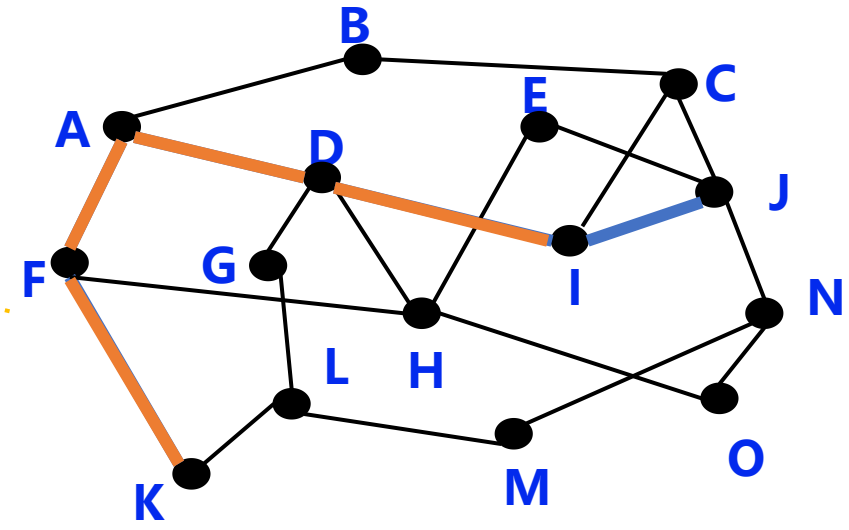
1. 优化原则
2. 最短路径算法
3. 距离向量路由
4. 链路状态路由
5. 层次路由
6. 广播路由（可选）
7. 组播路由
8. 选播路由（可选）

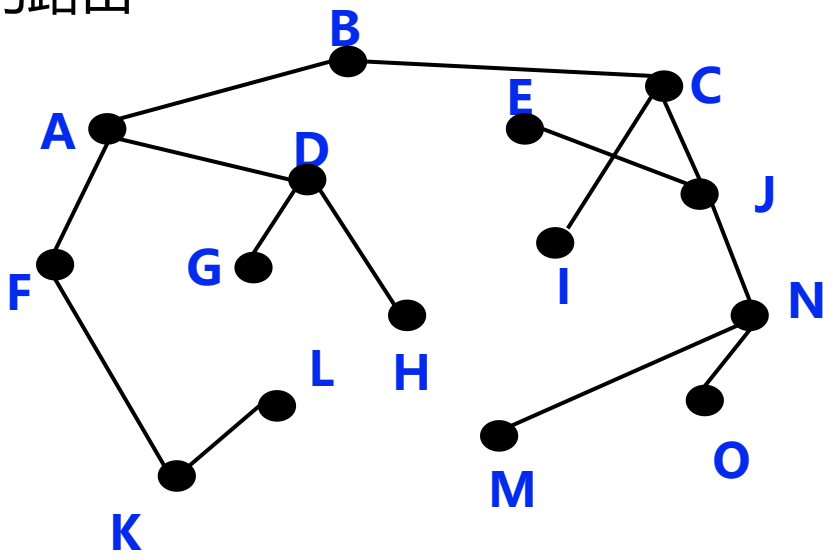
路由算法

- 路由算法须满足的特性：
  - 正确性
  - 简单性
  - 鲁棒性
  - 稳定性
  - 公平性
  - 有效性
- 根据路由算法是否随网络的通信量或拓扑自适应划分
  - 静态路由选择策略（非自适应路由选择）
  - 动态路由选择策略（自适应路由选择）

优化原则

- 汇集树(Sink Tree)
  - 所有的源节点到一个指定目标节点的最优路径的集合构成一棵以目标节点为根的树
  - 一棵路由器B的汇集树（距离度量单位：步长数）
  - 最优路径：若路由器I在从路由器J到路由器K的最优路径上，则从I到K的最优路径也必定遵循同样的路由



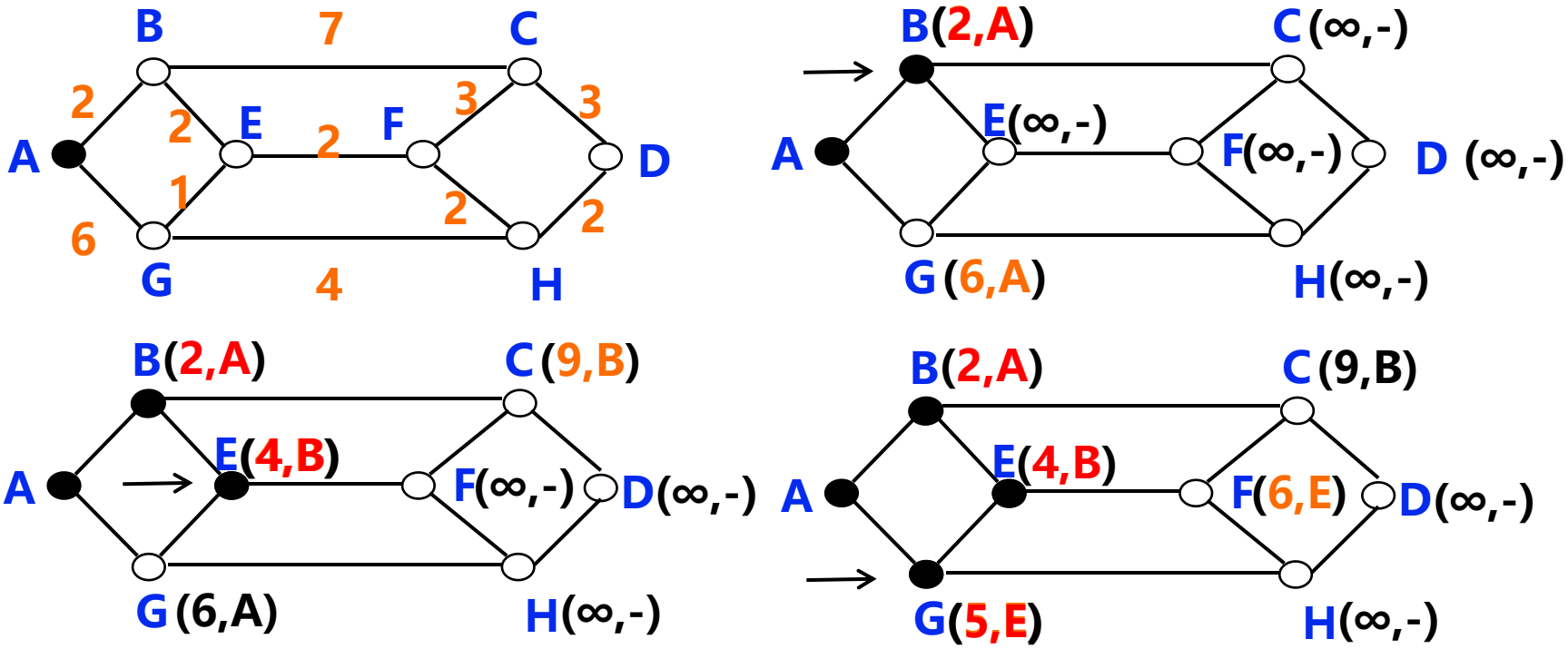


最短路径算法

- **定义**
  - 用于计算一个节点到其他所有节点的最短路径，主要特点是以起始点为中心向外逐层扩展，直到扩展到终点为止
- **Dijkstra算法思想**
  - 建立网络图
    - 节点表示路由器
    - 边表示通信线路/链路
    - 链路代价表示链路上的距离、信道宽度或通信开销等参数
  - 根据算法在网络图上为某一对路由器找之间的最短路径

最短路径算法

- 具体例子（A到D的最短路径过程）

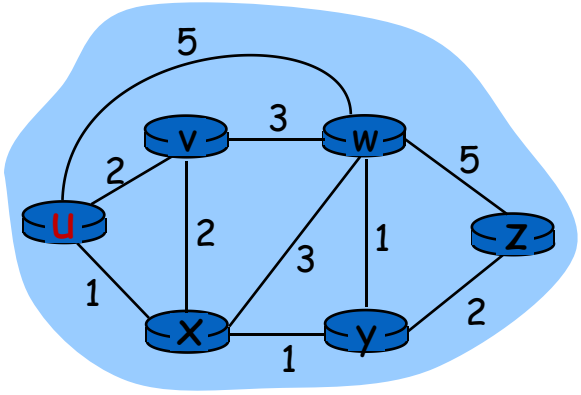


距离向量路由

- **距离向量（Distance Vector）算法基本思想:**
  - 每个节点周期性地向邻居发送它自己到某些节点的距离向量;
  - 当节点x接收到来自邻居的新DV估计，它使用B-F方程更新其自己的DV：

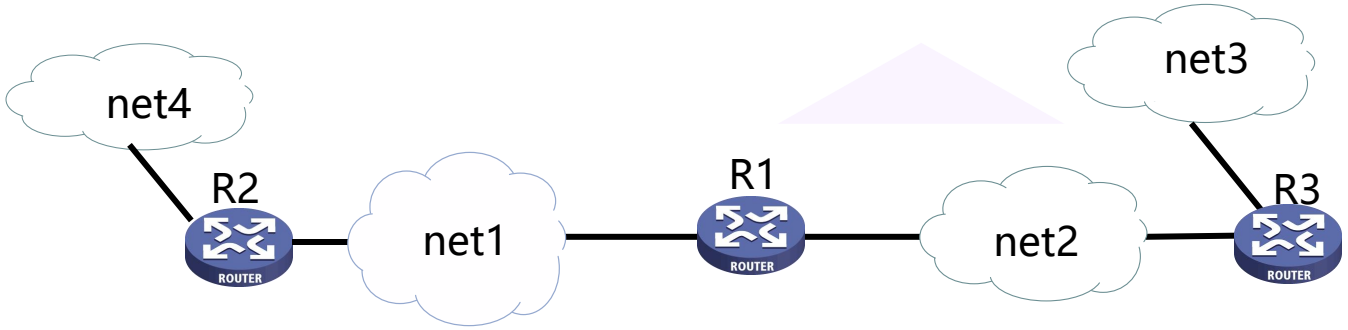
$$D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\} \quad for\ each\ node\ y\ \epsilon\ N$$

  - 上述过程迭代执行，$D_x(y)$收敛为实际最小费用 $d_x(y)$
- **距离向量算法特点：迭代的、分布式的**
  - 每次本地迭代由下列引起: 本地链路费用改变、邻居更新报文
  - 分布式:各节点依次计算，相互依赖



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



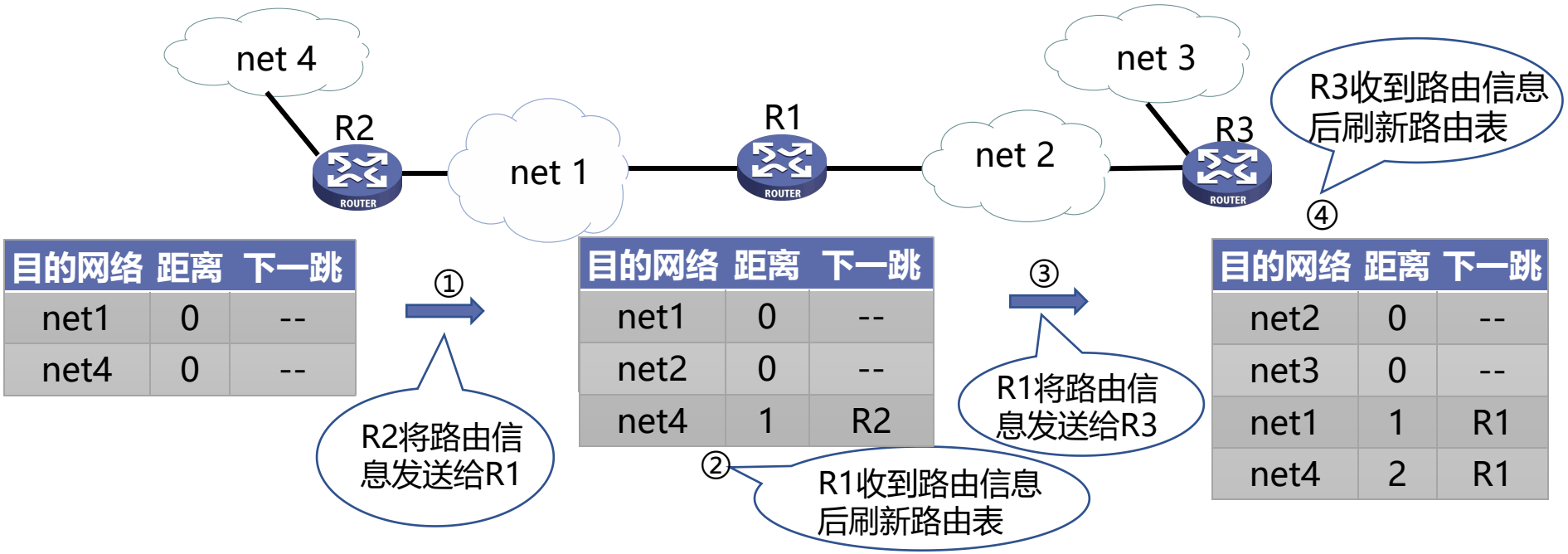
目的网络	距离	下一跳
net1	0	--
net4	0	--

目的网络	距离	下一跳
net1	0	--
net2	0	--

目的网络	距离	下一跳
net2	0	--
net3	0	--

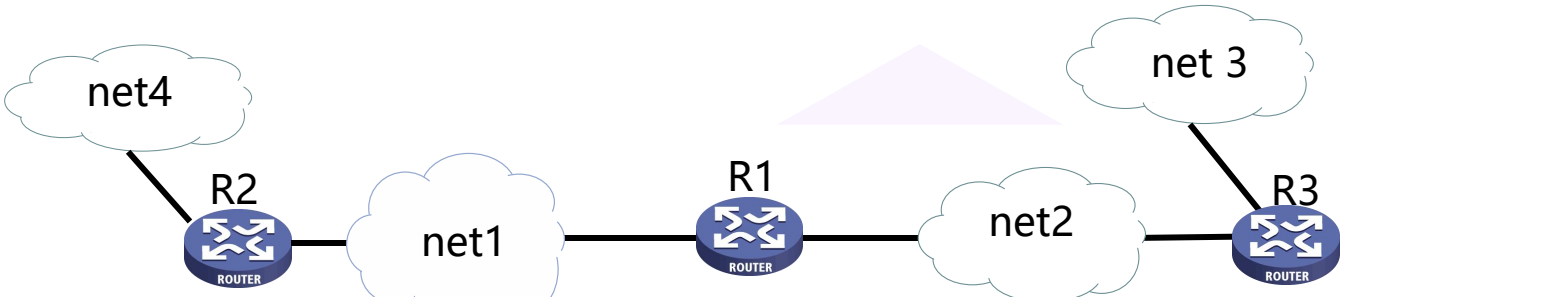
距离向量路由

- 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
- 相邻路由器可以根据收到的路由信息修改和刷新自己的路由表



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
- 所有的路由器都得到正确的路由选择信息时网络进入“收敛”（convergence）状态



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

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

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

距离矢量路由-课堂练习

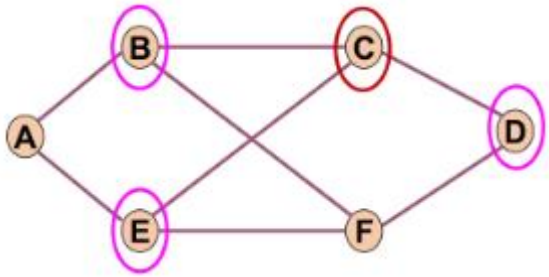
- 一个网络拓扑如下，某个时刻，路由器C接收到三个邻居发过来的矢量如下：

A B C D E F

From B:(5, 0, 8, 12, 6, 2)

From D:(16, 12, 6, 0, 9, 10)

From E:(7, 6, 3, 9, 0, 4)

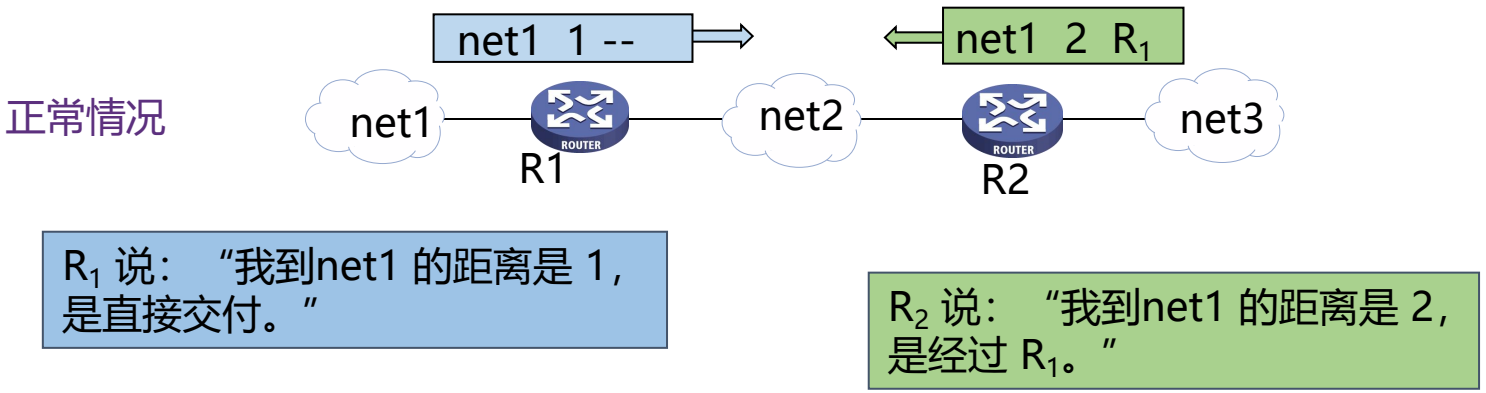


- 现在，C到B,D和E的代价分别是6、3和5，试回答C更新后的路由表是？

A	B	C	D	E	F
(11，B)	(6,B)	(0,C)	(3,D)	(5,E)	(8,B)

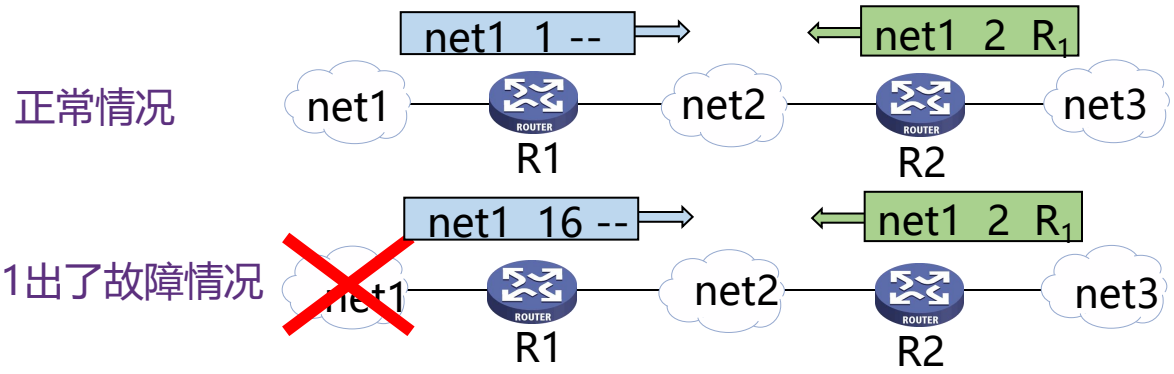
距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）



距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）

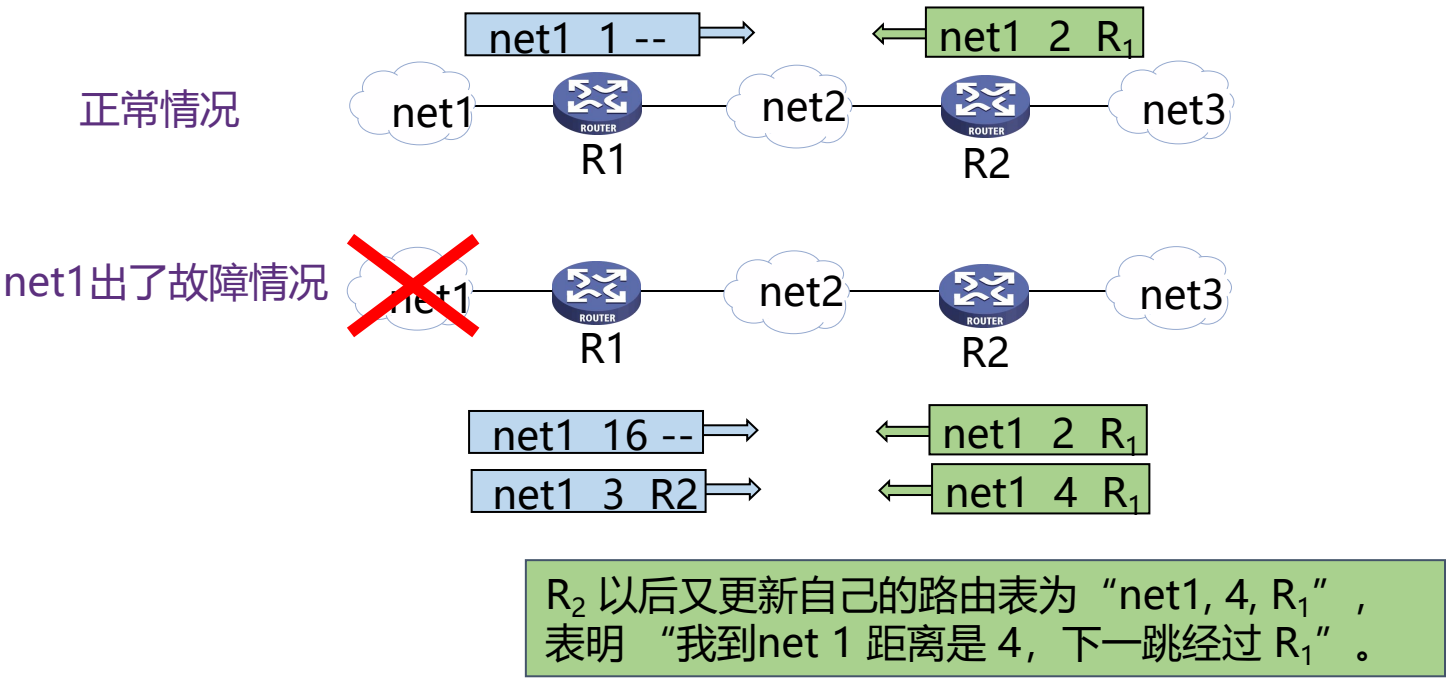


R_1 说：“我到net1 的距离是 16（表示无法到达），是直接交付。”

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 Net_1 出了故障。

距离向量路由

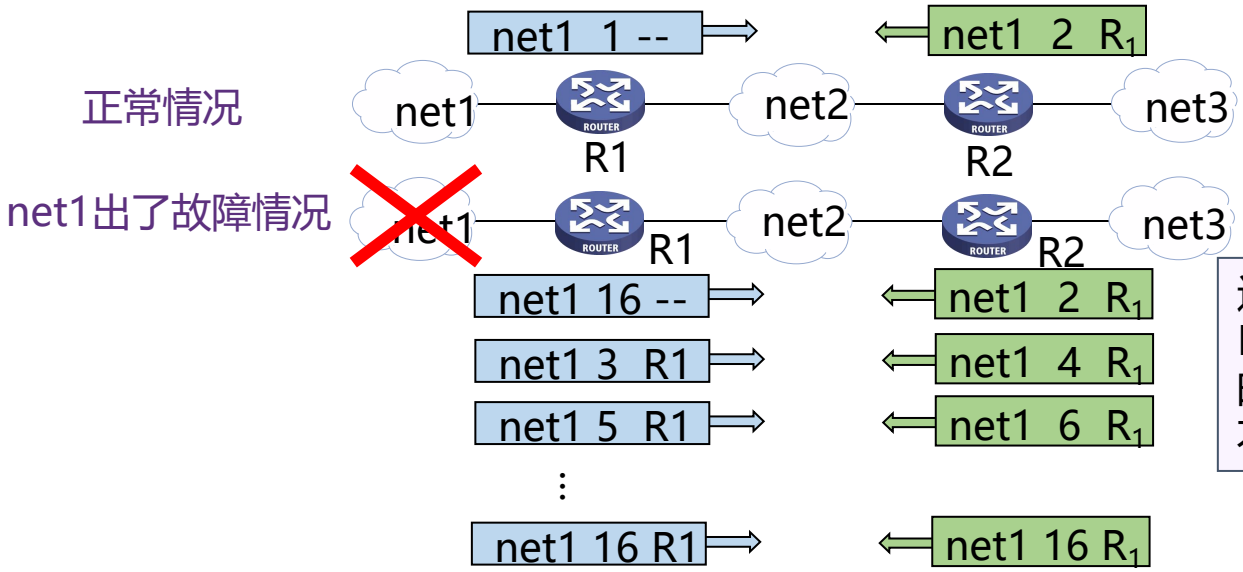
- 计数到无穷问题（The Count-to-Infinity Problem）



R_2 以后又更新自己的路由表为“net1, 4, R_1”，表明 “我到net 1 距离是 4，下一跳经过 R_1”。

距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）



这样不断更新下去，直到 R_1 和 R_2 到net 1 的距离都增大到16时， R_1 和 R_2才知道net 1 是不可达的。

➢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
(问题本质：局部信息共享，缺乏全局拓扑)