第五章 网络层

5.1 网络层功能和服务

5.2 网络层互连设备

5.3 路由选择原理

5.4 拥塞控制和流量控制

5.5 IP网际协议

5.6 ARP地址解析协议

5.7 ICMP报文控制协议

5.8 IGMP组管理协议

5.9 Internet路由问题

5.1 网络层功能和服务

- 实现端到端的传递，网络层提供两种主要功能：
  - 交换：建立临时连接
  - 路由：选择最佳路径
- 交换和路由的要求：
  - 在原始数据包上附加源和目的地址(信源和信宿)
  - 这些地址和数据链路层的上、下节点地址不同
- 网络层提供任意两个网络节点的可靠通信

5.1.1 网络层的功能

- 信源到信宿的传输：将多条物理链路连接成一条传输路径；
- 逻辑寻址：为了完成从信源到信宿的传输，在数据包的头部加入源地址和目的地址；
- 路由：选择信源到信宿发送数据包的最佳路经；
- 地址转换：网络层地址和物理地址的翻译；
- 复用：同一条物理线路同时传输多个设备间的数据
- 流量和拥塞控制：调节发送流量和反馈机制；
- 网络互连：解决网络互连的有关问题。

两种服务类型

- OSI模型定义两种服务类型：
  - 面向连接的服务(CONS)：虚电路
  - 面向无连接的服务(CLNS)：数据报eg.IP数据报

5.1.2 面向连接的网络服务(CONS)

- 面向连接的网络服务为数据传输建立一条虚电路
- 该电路在整个数据传输过程中都是有效的。
- 一次数据传输过程的所有包都将按顺序沿着已建立的虚电路传输

完成步骤

- 一个面向连接的网络服务通过如下步骤完成一次传输过程：
  - 发送者发送一个连接请求包
  - 接收者使用一个连接确认包进行确认
  - 发送者传输数据
  - 发送者发送一个连接终止请求包
  - 接收者使用一个连接终止包进行确认

面向连接网络服务的优缺点

- 优点：
  - 允许一个协议包含全面的顺序、差错和流量控制。
  - 允许在流量控制上使用滑动窗口。
  - 数据包中使用了较少的协议控制信息，减少了额外开销。
- 缺点：
  - 连接建立以后，丧失路由的灵活性。如果一条链路发生阻塞或出现其他问题，后续的包不能使用其他的路径来替代。
  - 比面向非连接的网络服务速度低。因为包必须被检查，或者被确认、或者被重传。

5.1.3 面向无连接的网络服务(CLNS)

- 在面向无连接的网络服务中，一次多包传输中，每个包被当作一个独立的单元。
- 无连接协议不提供逻辑连接。
- 发送者仅仅是发送数据，不需要提醒接收者有通信即将到来。
- 中间节点根据路由信息和报头地址选择路径。

面向无连接网络服务的优缺点

- 优点：
  - 如果可靠性和排序可由上层协议来处理的话，CLNS具有速度和开销方面的优势。
  - 如果某一条路经发生阻塞或中断，包可以选择另一条路经。
  - 单个传输的各个片断可以通过不同的路径传输，从而达到最大的效率。
- 缺点：
  - CLNS不可靠，无法保证数据包顺序到达。
  - 每个包所需的开销较大，每个包必须携带完整的地址信息。

两种方式的总结(1)

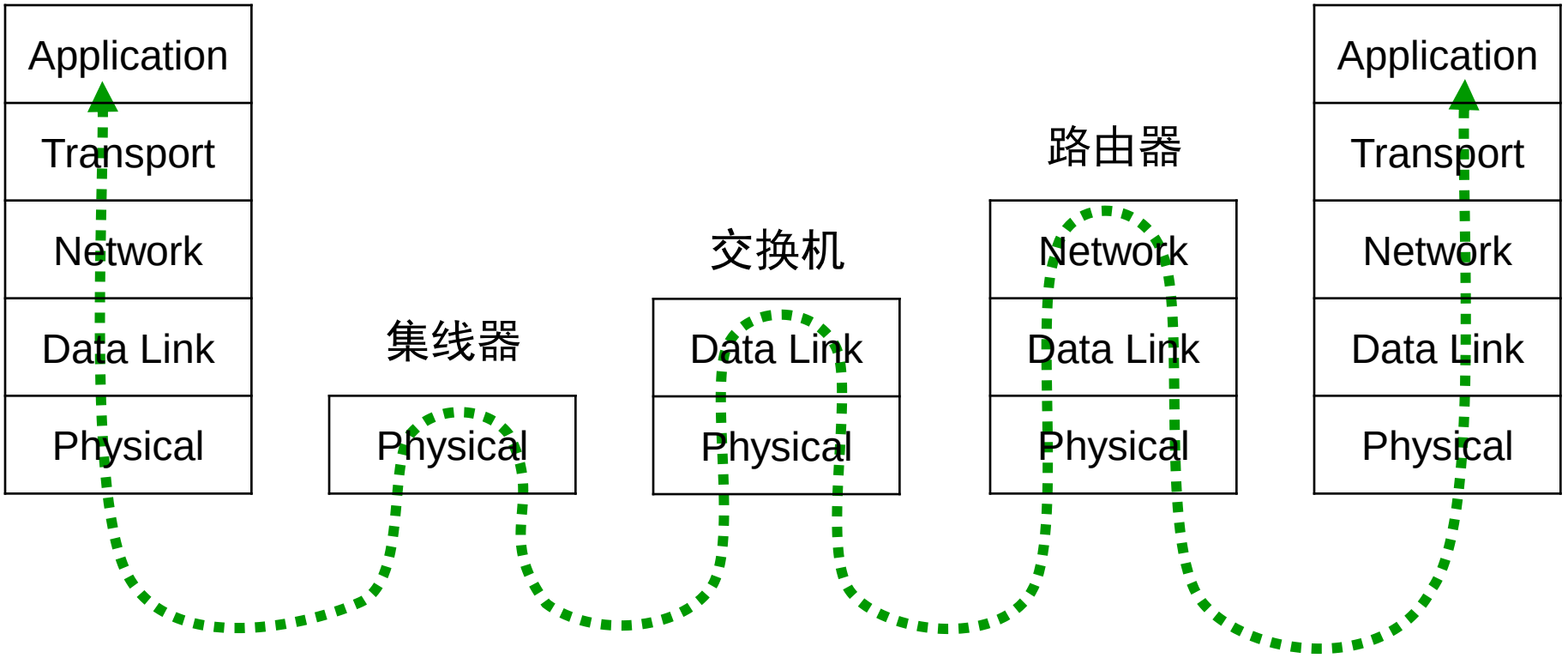
- 通信子网向端系统提供两种网络服务：虚电路，数据报。而通信子网内部的工作也有虚电路和数据报方式。
- 提供虚电路服务的通信子网内部的操作可以是虚电路方式、也可以是数据报方式，一般是提供虚电路服务。

两种方式的总结(2)

- 尽管通信子网的数据报交换是不可靠的，但两端的网络节点可以进行诸如排序、重发等工作，从而满足了可靠通信服务的要求。
  - 通信子网内部节点按数据报方式交换数据，而与端系体连接的网络节点向端系统提供虚电路服务。对于端系统来说，它的网络层与网络节点的通信像虚电路操作方式一样，先建立虚电路，再交换数据，最后拆除连接。但每个分组被网络节点分成若干数据报，附上地址分送到目的地。
  - 例如：TCP/IP协议，IP协议是无连接的，但TCP协议是面向连接的。

5.2 网络层互连设备

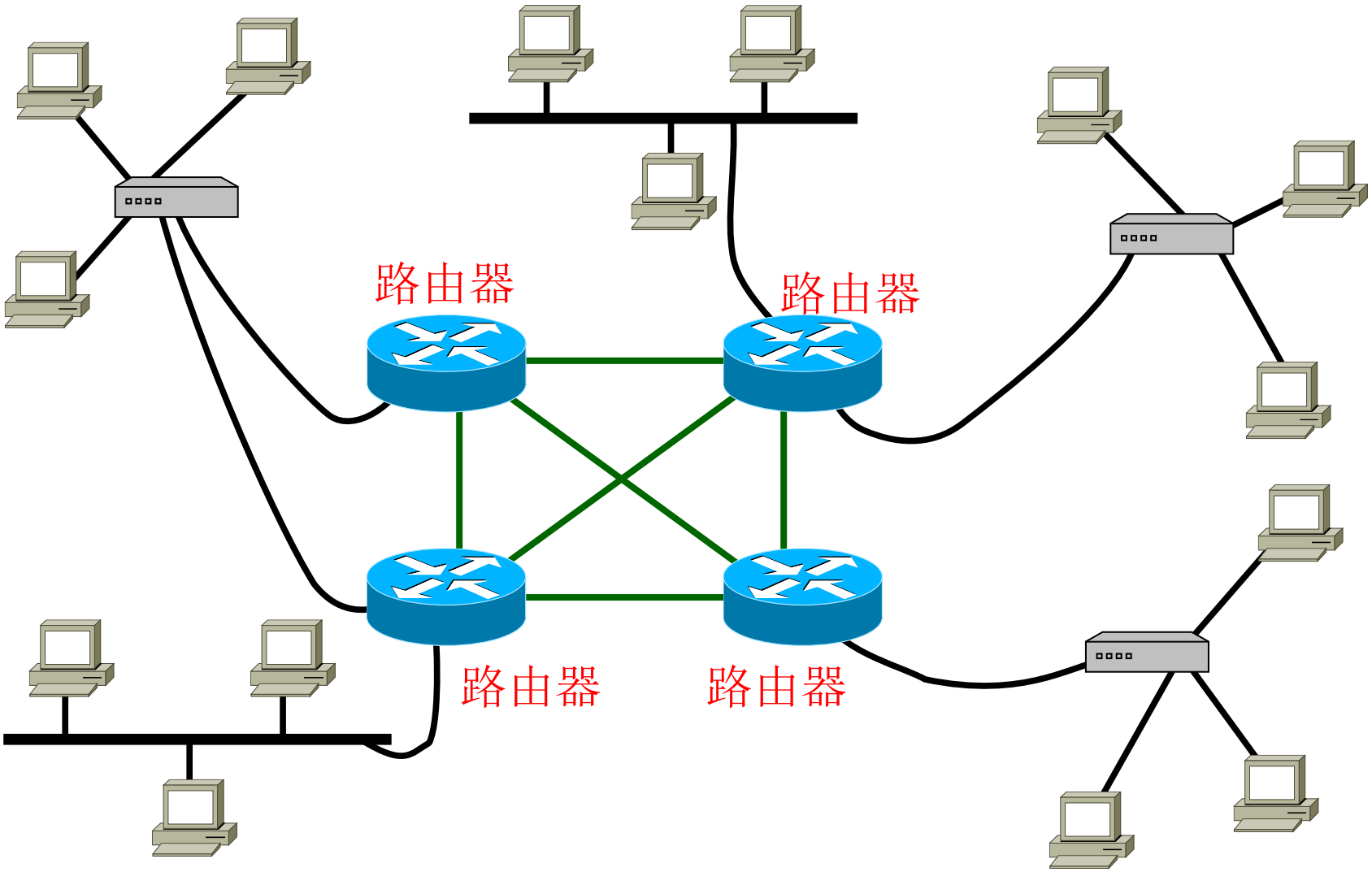
- 网络层互连设备主要是路由器



5.2.1 路由器

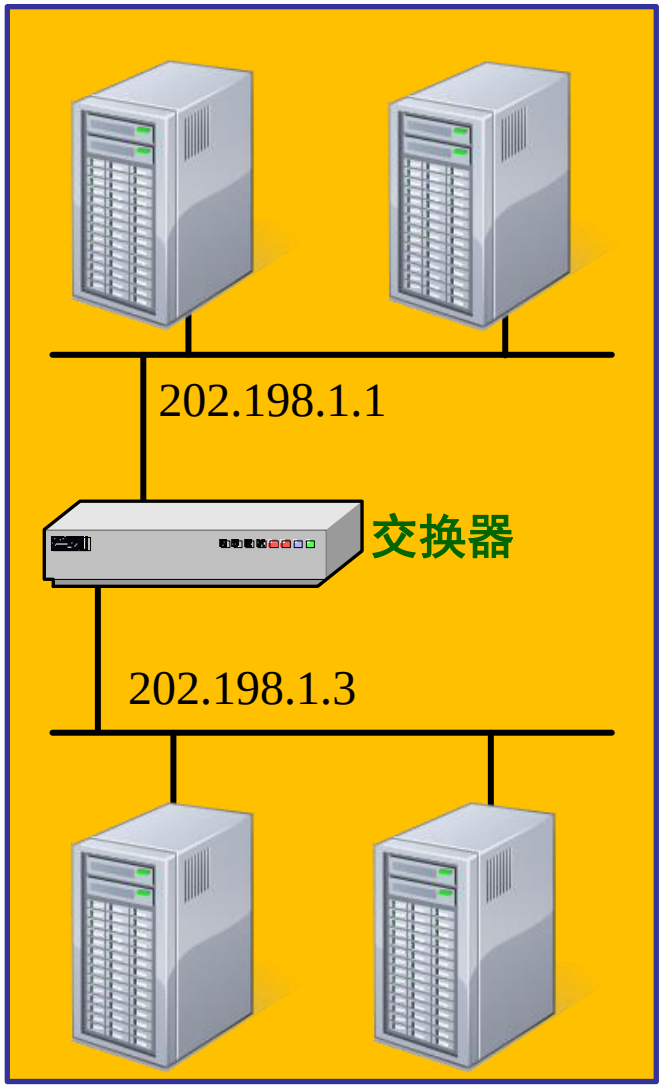
- 路由器工作在网络层
- 路由器在多个互连设备之间中继包
  - 从一个网络接口接收，从另一个网络接口发送
- 路由器对来自某个网络的包确定传输路径，发送到互连网络中任何可能的目的网络中
- 路由器通常由硬件、软件两部分构成

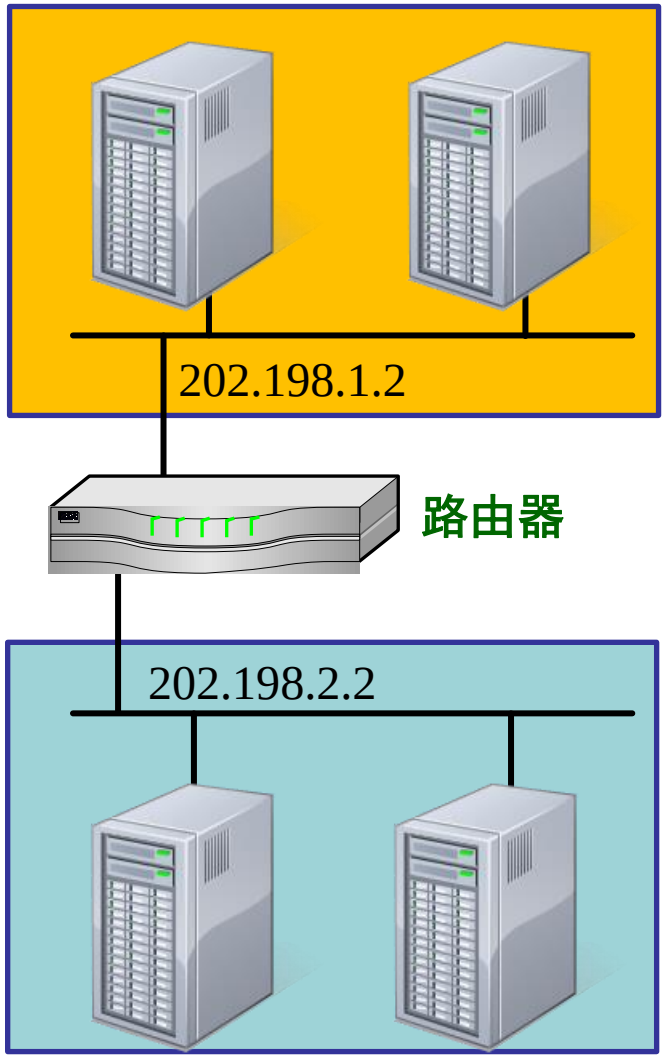
用路由器互连示例



路由器和交换机的区别

交换机连接两个局域网，把它们变成更大的一个局域网





通过路由器将这两个不同的网络互连在一起

5.2.2 第三层交换机

- 三层交换机的特征：
  - 转发基于第三层地址的业务流
  - 完全交换功能
  - 完成特殊任务，如报文过滤
  - 有路由功能

5.2.3 网关

- 网关是一个协议转换器。
- 通常是安装在路由器内部的软件。
- 可以工作在OSI的7层。

5.3 路由选择原理

- 路由选择就是网络中各个节点为到来的数据包选择一条输出链路。
- 如果网络内部使用数据报，那么就必须为每个到来的包作一次路由选择。
- 如果网络内部使用虚电路，则仅在建立一个虚电路时作一次路由选择，以后各数据包都按建立的路由传送。

路由选择的基本要求

- 正确性：路由算法必须是正确的
- 简单性：算法在计算上应该简单
- 坚定性：长时间运行不会出现系统故障
- 稳定性：算法是收敛的
- 公平性：通信节点利用信道的机会均等
- 最佳性：按一定的标准获得最好的效果

分布式路由选择策略

- 每个节点有一个路由表，并周期性地从周围相邻的节点获得网络状态信息，同时，也将本节点做出的路由周期性地通知相邻的各节点。
- 整个网络的路由选择经常处于动态变化之中。
- 路由表通常根据各结点间距离进行调整。距离可以是：
  - 链路数目、延迟时间、通信费用等等
- 典型的协议有：
  - RIP协议
  - OSPF协议

集中式路由选择策略

- 网络控制中心(NCC)负责全网状态信息的收集、路由计算、以及路由选择的实现。
- 每个节点定期向网络控制中心报告一些状态信息。
- 优点：
  - 各个节点不需要路由选择计算
  - 对网内的某种流量可调控
  - 易消除网络环路
- 缺点：
  - 中心较近的地方通信量大
  - 可靠性差
  - 网络的规模受到限制

基本的路由算法

- 距离最短的路径是最佳路径
- 距离最短的标准可以是费用最小、传输延迟最小、数据传输速率最大、以及这些因素的一种组合。
- 有两种最常用的计算最短路经的方法：
  - 距离向量路由
  - 链路状态路由

5.3.1 距离向量路由算法(RIP协议采用此算法)

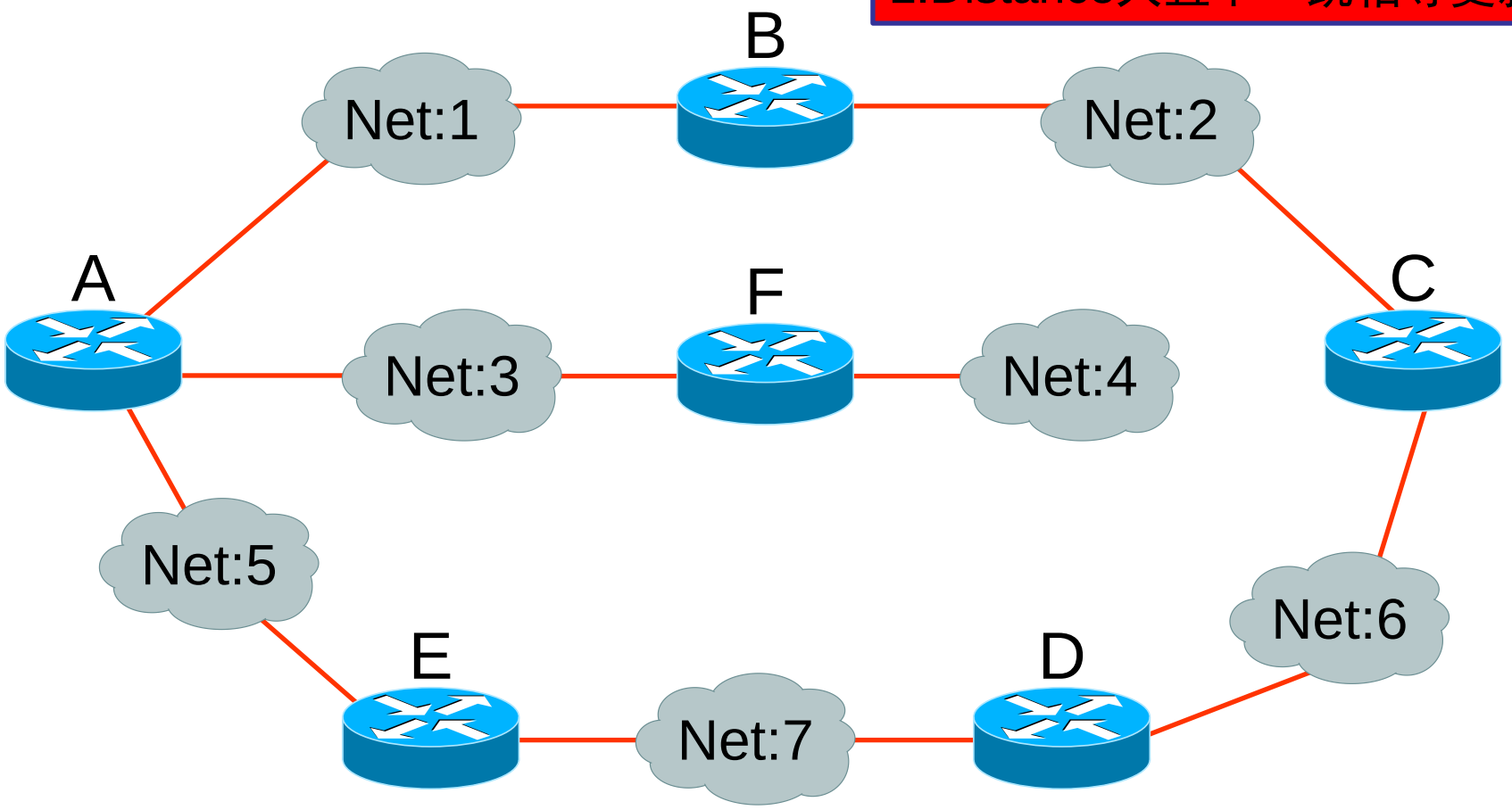
- 在距离向量路由中，每个路由器周期性的将自己关于整个网络的信息发送给它的邻居
  - 每个路由器保存关于整个网络的信息；
  - 仅仅和邻居交换网络信息；
  - 信息的交换是通过有规律的时间间隔来进行(例如每隔30秒发一次)，无论网络状态是否发生变化。
- 每个路由器依据路由表来转发数据包，其的路由表中的每一项一般具有如下的格式：

NetID : Distance : Nexthop

↓ 目的网络　↓ 网络距离　↓ 下一个跳

互连示例

更新条件：
1.Distance小更新
2.Distance大且下一跳相等更新



假设每条链路的距离为1

NetID : Distance : Nexthop

↓ ↓ ↓

目的网络 网络距离 下一个跳

例：在A的路由表中
2:2:B

路由表的创建与更新(轮次0、1)

轮次	路由器	目的网络						
		1	2	3	4	5	6	7
0	A	1:1:-	2:∞:?	3:1:-	4:∞:?	5:1:-	6:∞:?	7:∞:?
	B	1:1:-	2:1:-	3:∞:?	4:∞:?	5:∞:?	6:∞:?	7:∞:?
	C	1:∞:?	2:1:-	3:∞:?	4:∞:?	5:∞:?	6:1:-	7:∞:?
	D	1:∞:?	2:∞:?	3:∞:?	4:∞:?	5:∞:?	6:1:-	7:1:-
	E	1:∞:?	2:∞:?	3:∞:?	4:∞:?	5:1:-	6:∞:?	7:1:-
	F	1:∞:?	2:∞:?	3: 1:-	4:1:-	5:∞:?	6:∞:?	7:∞:?
1	A	1:1:-	2:2:B	3:1:-	4:2:F	5:1:-	6:∞:?	7:2:E
	B	1:1:-	2:1:-	3:2:A	4:∞:?	5:2:A	6:2:C	7:∞:?
	C	1:2:B	1:1:-	3:∞:?	4:∞:?	5: ∞ :?	6:1:-	7:2:D
	D	1:∞:?	2:2:C	3:∞:?	4:∞:?	5:2:E	6:1:-	7:1:-
	E	1:2:A	2:∞:?	3:2:A	4:∞:?	5:1:-	6:2:D	7:1:-
	F	1:2:A	2:∞:?	3:1:-	4:1:-	5:2:A	6:∞:?	7:∞:?

路由表的创建与更新(轮次2、3)

轮次	路由器	目的网络						
		1	2	3	4	5	6	7
2	A	1:1:-	2:2:B	3:1:-	4:2:F	5:1:-	6:3:B	7:2:E
	B	1:1:-	2:1:-	3:2:A	4:3:A	5:2:A	6:2:C	7:3:A
	C	1:2:B	1:1:-	3:3:B	4:∞:?	5:3:A	6:1:-	7:2:D
	D	1:3:C	2:2:C	3:3:E	4:∞:?	5:2:E	6:1:-	7:1:-
	E	1:2:A	2:3:D	3:2:A	4:3:A	5:1:-	6:2:D	7:1:-
	F	1:2:A	2:3:A	3:1:-	4:1:-	5:2:A	6:∞:?	7:3:A
3	A	1:1:-	2:2:B	3:1:-	4:2:F	5:1:-	6:3:B	7:2:E
	B	1:1:-	2:1:-	3:2:A	4:3:A	5:2:A	6:2:C	7:3:A
	C	1:2:B	1:1:-	3:3:B	4:4:B	5:3:A	6:1:-	7:2:D
	D	1:3:C	2:2:C	3:3:E	4:4:E	5:2:E	6:1:-	7:1:-
	E	1:2:A	2:3:D	3:2:A	4:3:A	5:1:-	6:2:D	7:1:-
	F	1:2:A	2:3:A	3:1:-	4:1:-	5:2:A	6:4:A	7:3:A

算法的特点

- 优点：
  - 简单
  - 适用于小规模网络
- 缺点：
  - 网络规模的伸展性差
  - 对链路状态的变化响应慢
  - 路由包文尺寸大
  - 轮数与路由器的个数成正比

距离向量路由算法典型的协议

- RIP(Routing Information Protocol)
- 在路由器上键入命令：
  - Router rip
  - Network 192.168.1.0
  - Network 192.168.2.0

5.3.2 链路状态路由算法(OSPF协议采用此算法)

- 在链路状态路由中，每个路由器和互连网络中的所有其它路由器共享关于它邻居的信息:
  - 共享关于邻居的信息
  - 共享的信息发给所有的路由器(扩散法)
  - 共享信息在有规律的时间间隔内进行(一般30分钟)
- 理解链路状态路由的关键在于它和距离向量路由的不同之处
  - 在链路状态路由中，每个路由器和互连网络中的所有其它路由器共享关于它邻居的信息。

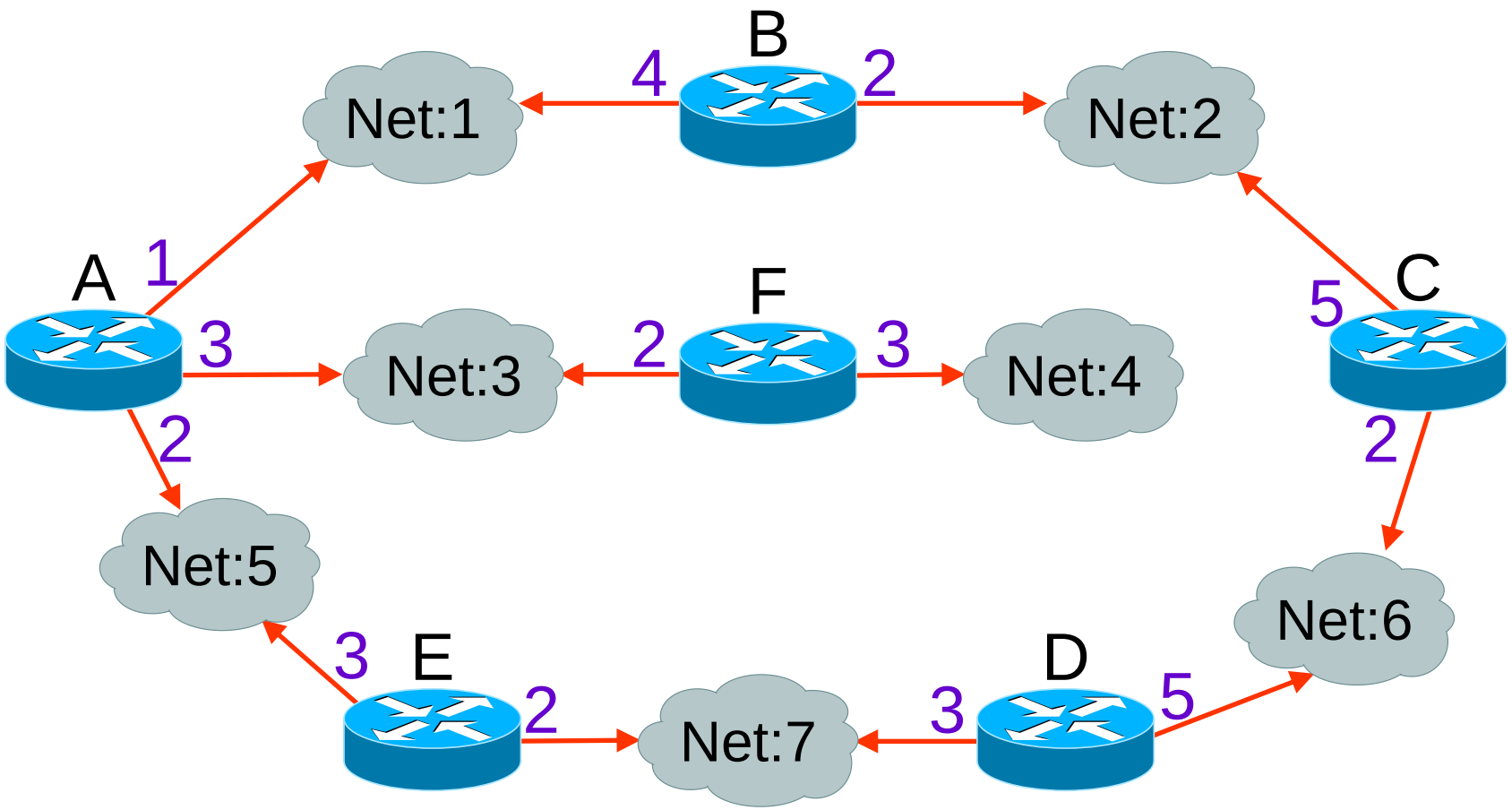
完成步骤

- 链路状态路由可分为两步完成
  - 第一步：共享链路状态信息，即每个路由器将它自己和它的所有邻居之间的链路状态信息发送给互连网络中的所有其它路由器。
  - 第二步：每个路由器根据自己所掌握的关于整个网络的链路状态信息计算到每个网路的路由。

链路状态信息共享(1)

- 路由器传输包的费用：在链路状态路由中，费用是许多因素的加权值。这些因素包括安全级别，流量和链路的传输速率等。
- 费用的计算：仅计算路由器到网络的部分，网络到路由器的费用不计。

链路状态路由中的费用



链路状态信息共享(2)

- 链路状态包：路由器通过向整个互连网络中的所有路由器发送链路状态包(LSP)，在网络中扩散关于自己邻居的信息。
- 一个LSP通常包含4个信息域：
  - 广告者的ID
  - 所影响的目标网络ID
  - 费用
  - 邻居路由器的ID

链路状态信息共享(3)

- 获得关于邻居路由器的信息：每个路由器都周期性地发送一个简短的问候包来获取关于它们邻居的信息。根据是否得到应答，做出不同反应。
- 初始化：每个路由器在启动时向它的所有邻居发送一个问候包来获取每条链路的状态信息。然后它基于这些问候的结果准备一个LSP，并将它扩散到整个网络。
  - 大家好！我是新路由器，这里有人吗?

链路状态信息共享(4)

- 链路状态数据库：每个路由器接收每个其它路由器发送来的LSP，并将它们的信息存放到一个链路状态数据库中。
  - 由于每个路由器接收相同的链路状态数据包，所以各路由器的链路状态数据库相同。

链路状态数据库

广告者	相关网络	费用	邻居
A	1	1	B
A	3	3	F
A	5	2	E
B	1	4	A
B	2	2	C
C	2	5	B
C	6	2	D
D	6	5	C
D	7	3	E
E	7	2	D
E	5	3	A
F	3	2	A
F	4	3	-

链路状态路由算法典型的协议

- 典型的协议是OSPF(Open Short Path First)
- 路由器上启动OSPF协议:

```
router  OSPF  99
Network 192.168.1.0  0.0.0.255  area 0
Network 192.168.2.0  0.0.0.255  area 0
```

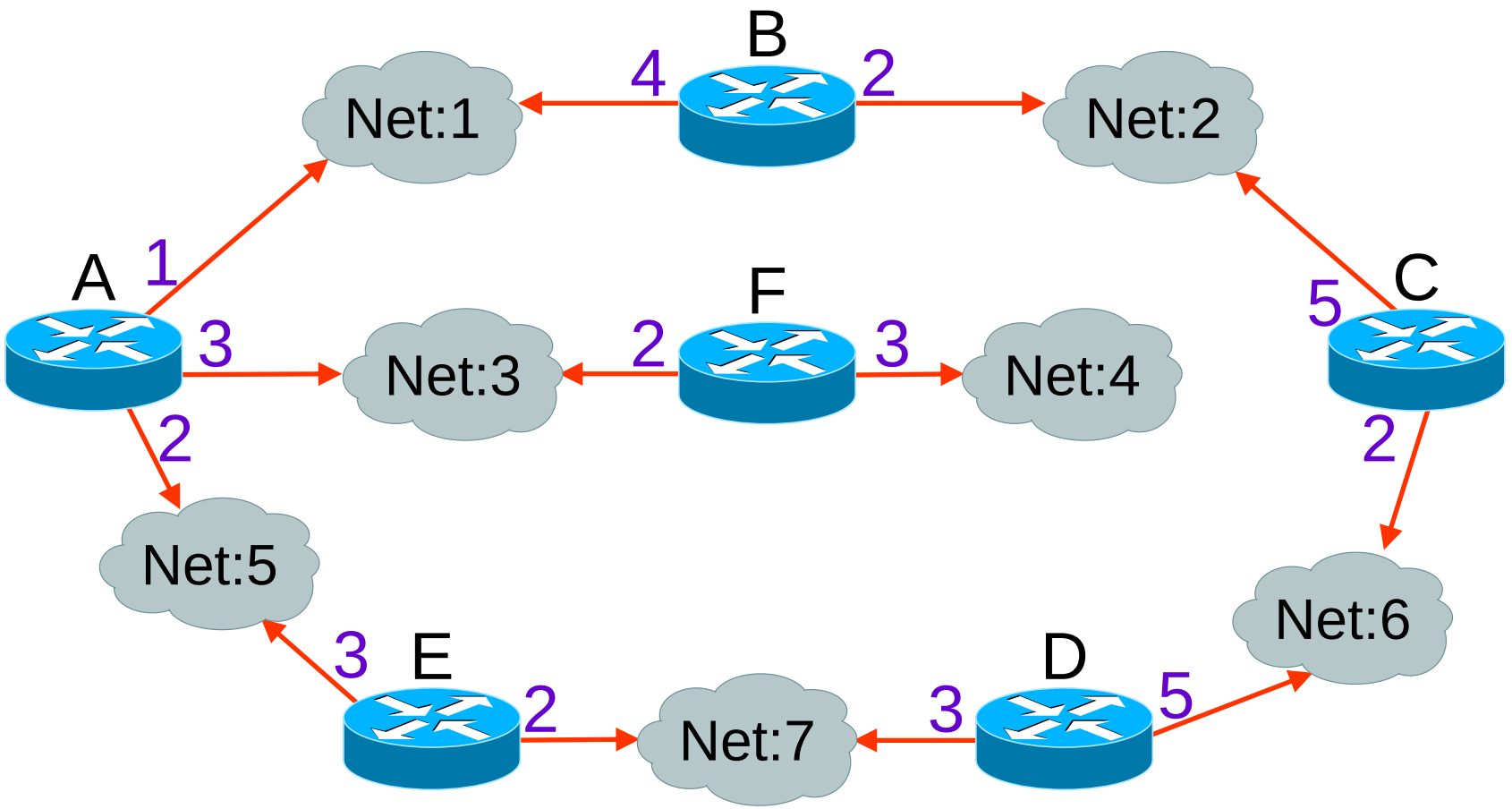
迪科斯彻(Dijkstra)算法

- Dijkstra算法(1959年)使用由节点和弧组成的图计算网络中两点之间的最短路径。
- 节点有两种类型：网络和路由器。
- 弧也有两类：路由器到网络的链路和网络到路由器的链路。
- 在Dijkstra算法中，从路由器到网络的链路的费用有效，而从网络到路由器的链路的费用总是为0。
- 每个路由器在使用Dijkstra算法时，根据下面四个步骤来形成自己的最短路经树。

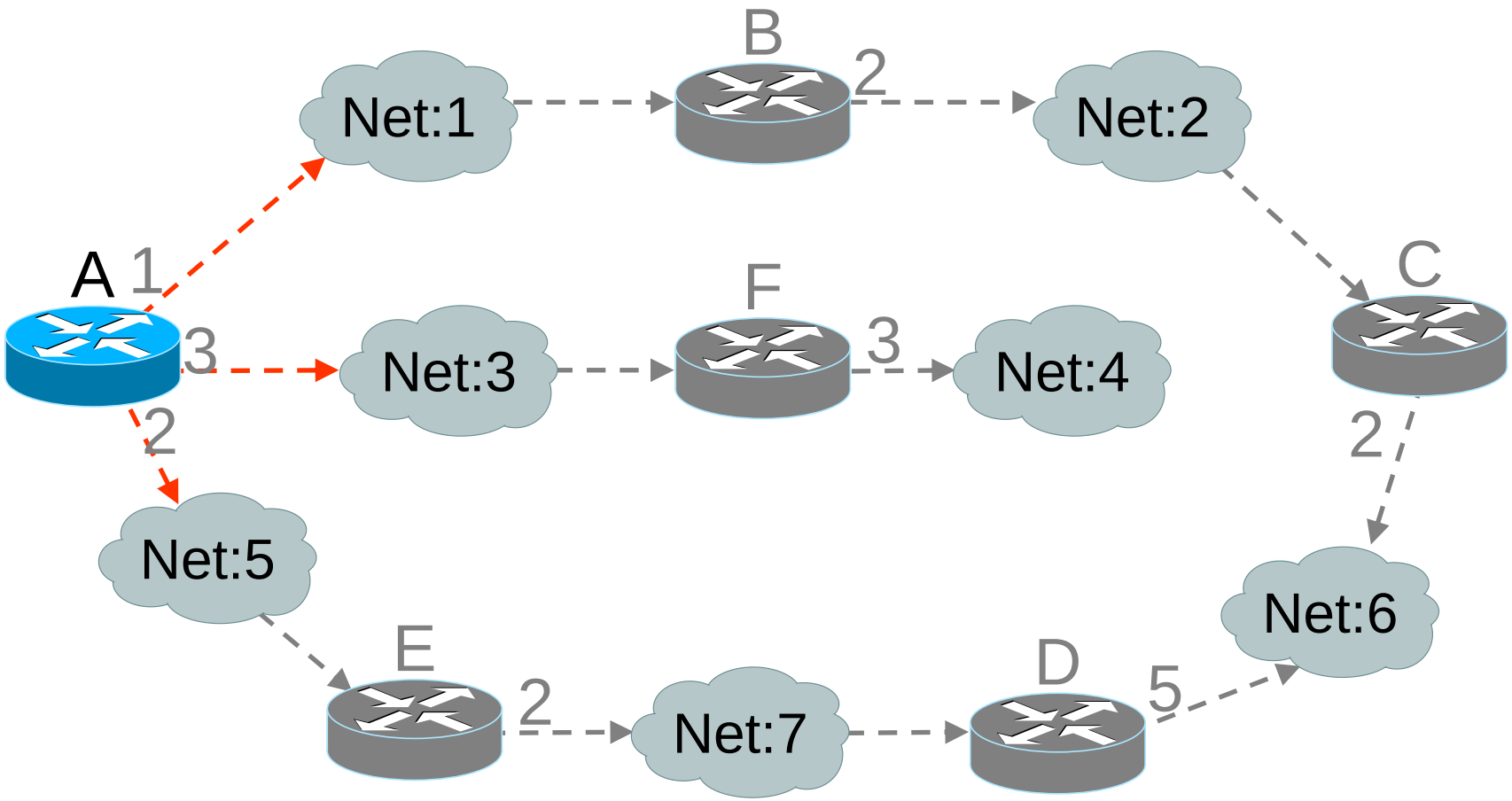
Dijkstra算法的一般步骤

1. 选择自己作为树的根，并将根标记为永久性节点，算法接着从根出发连接它所有邻居节点，这种连接是临时性的。
2. 算法比较所有的临时连接，找出费用最小的路径，这个路经上的所有弧和节点被标记为最短路经树上的永久部分。
3. 算法考察链路状态数据库，找出从这个选定的最短路经向外延伸所能连接的所有非永久性节点，将这些节点临时性的加到最短路经树上。
4. 如果所有的节点已经成为最短路经树上的永久部分，则算法结束，去掉非永久性的弧。否则，转步骤2继续执行。

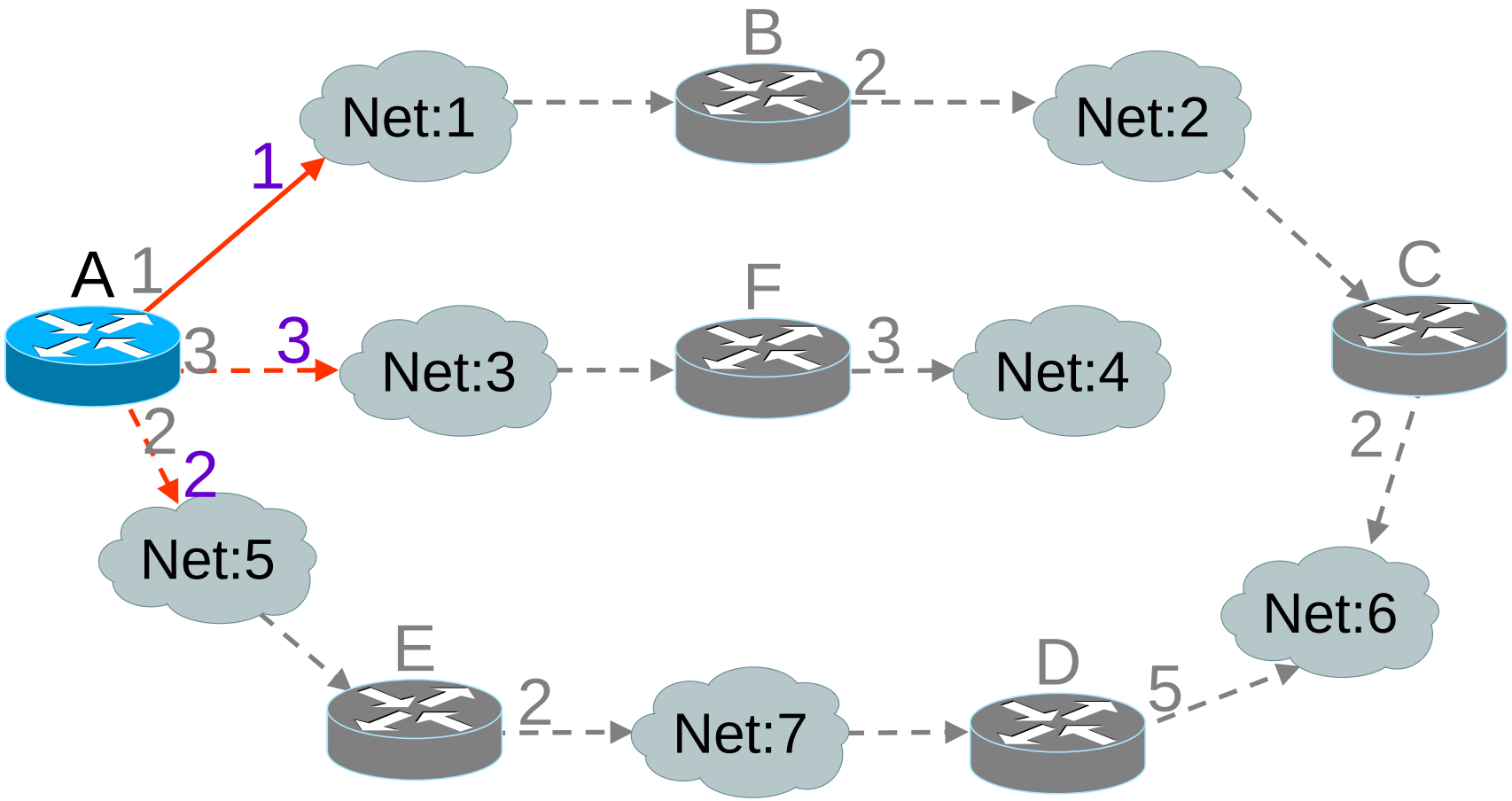
Dijkstra算法示例



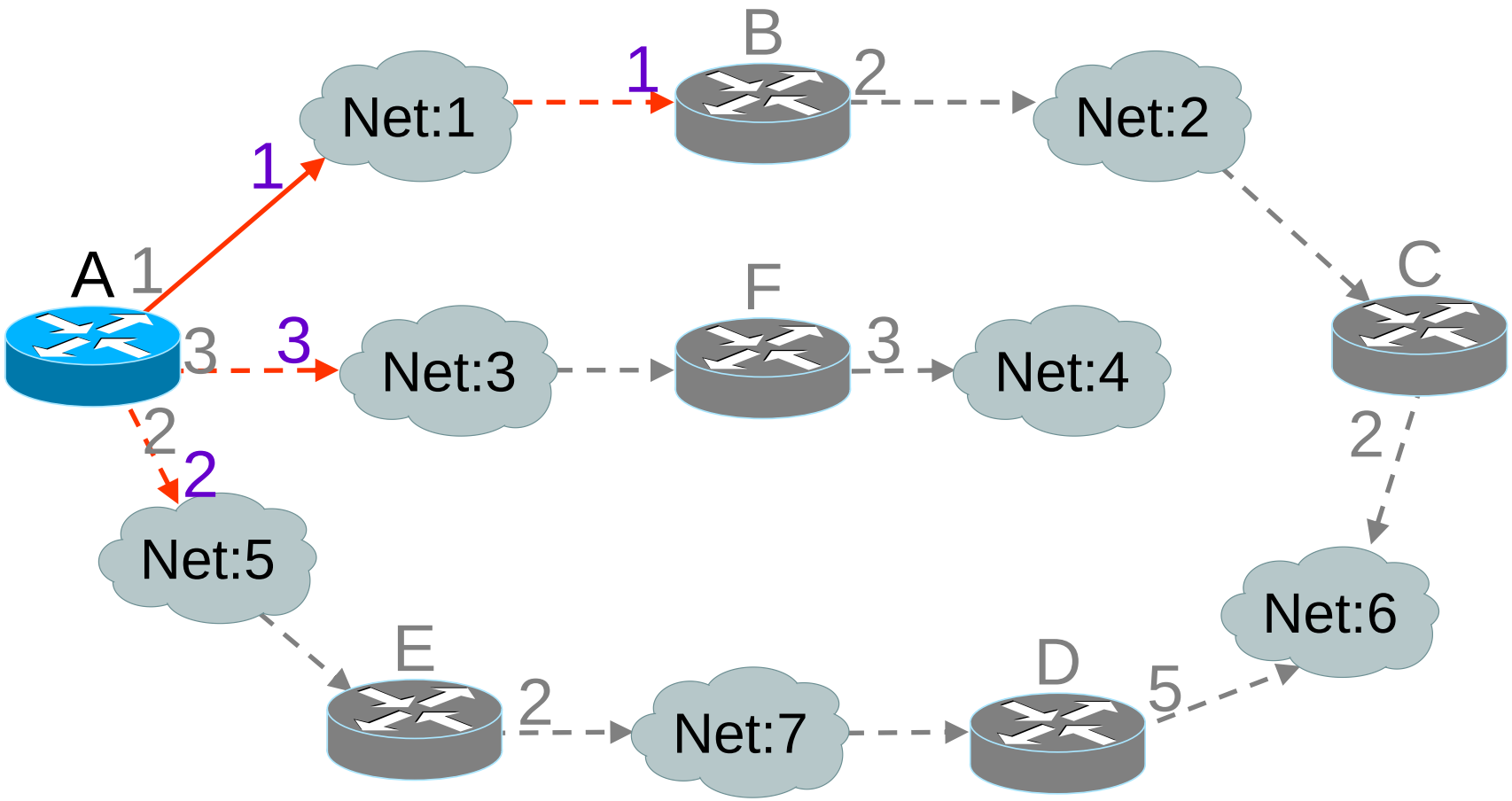
Dijkstra算法示例(1)



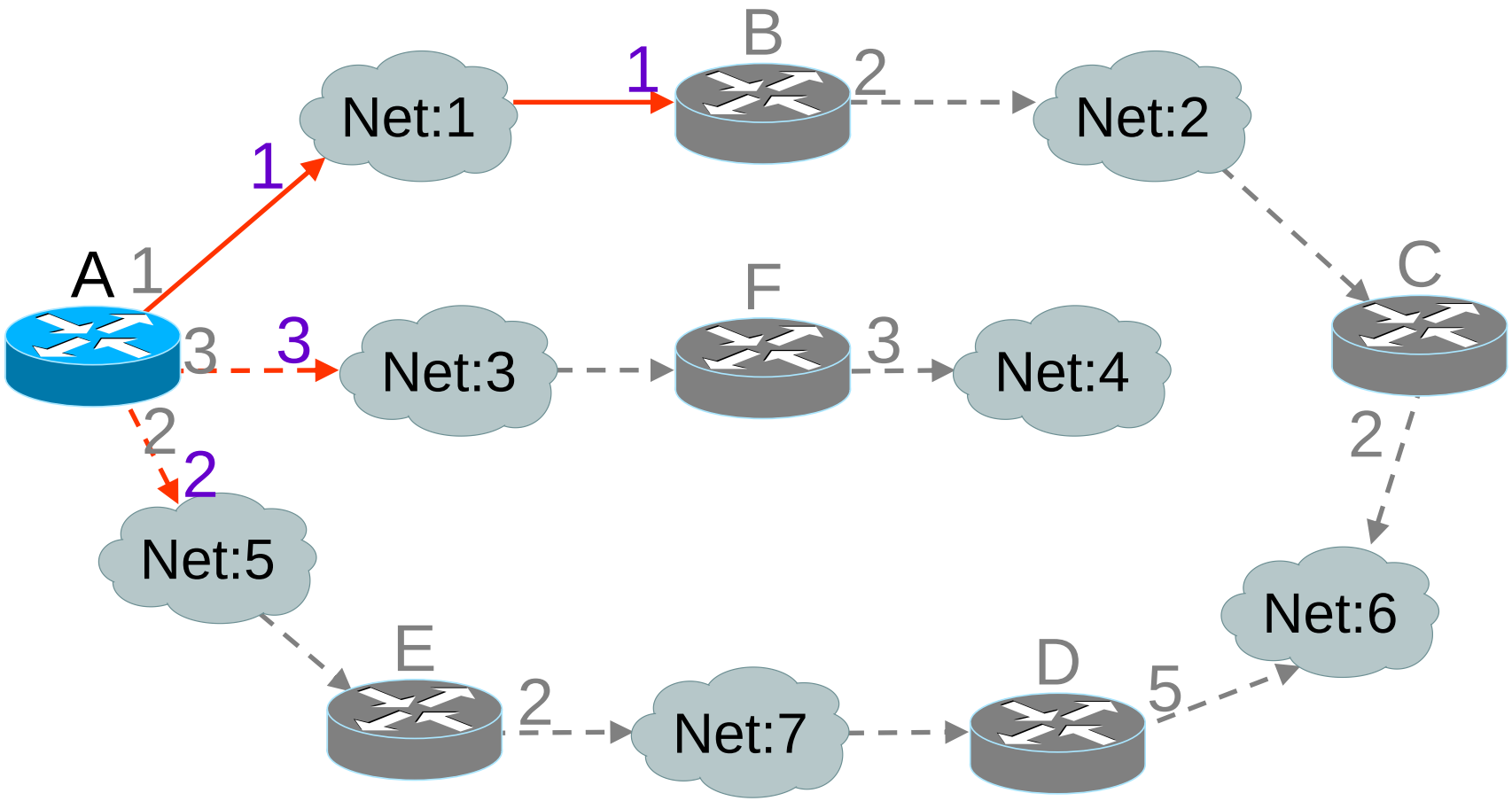
Dijkstra算法示例(2)



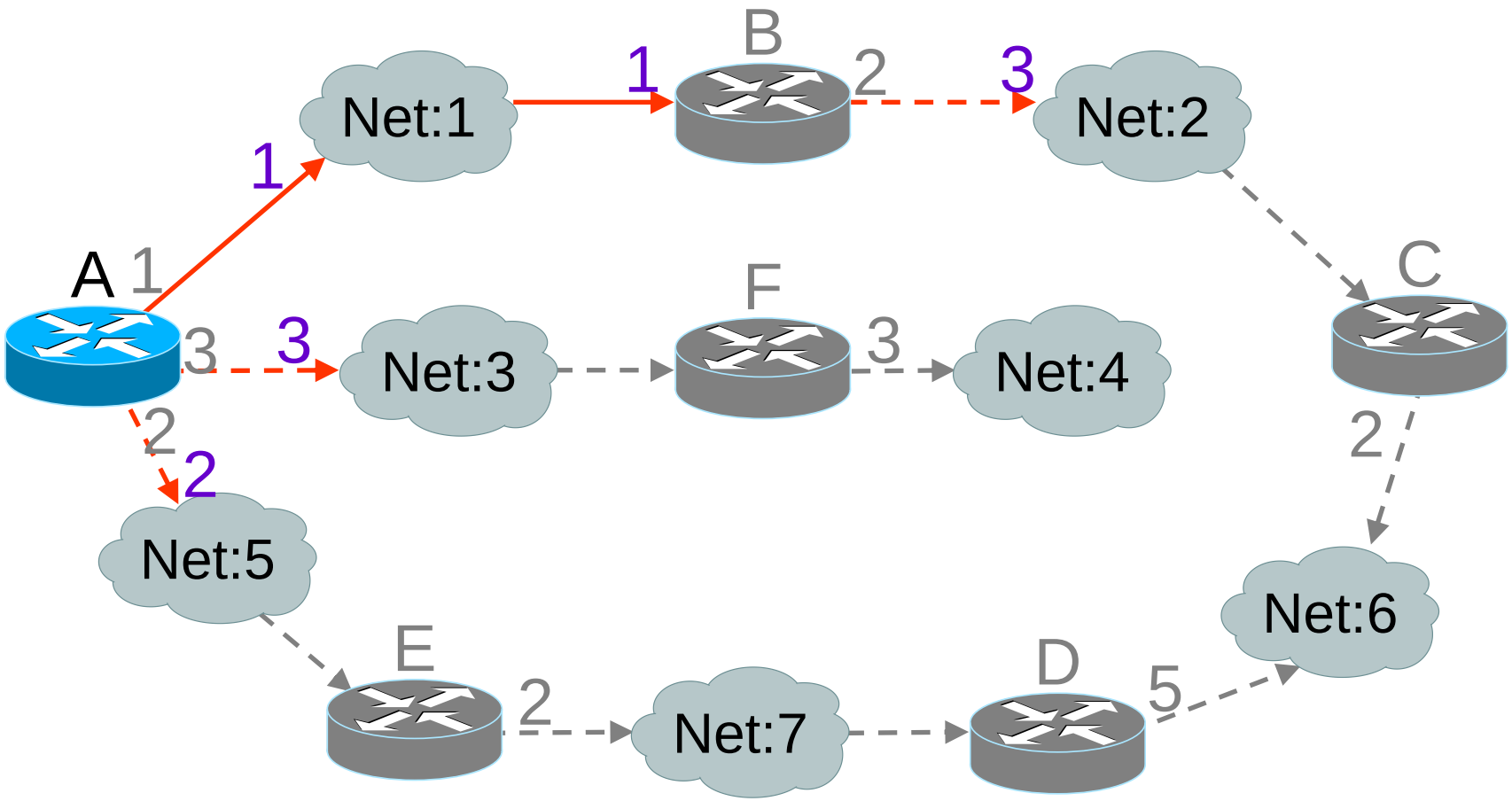
Dijkstra算法示例(3)



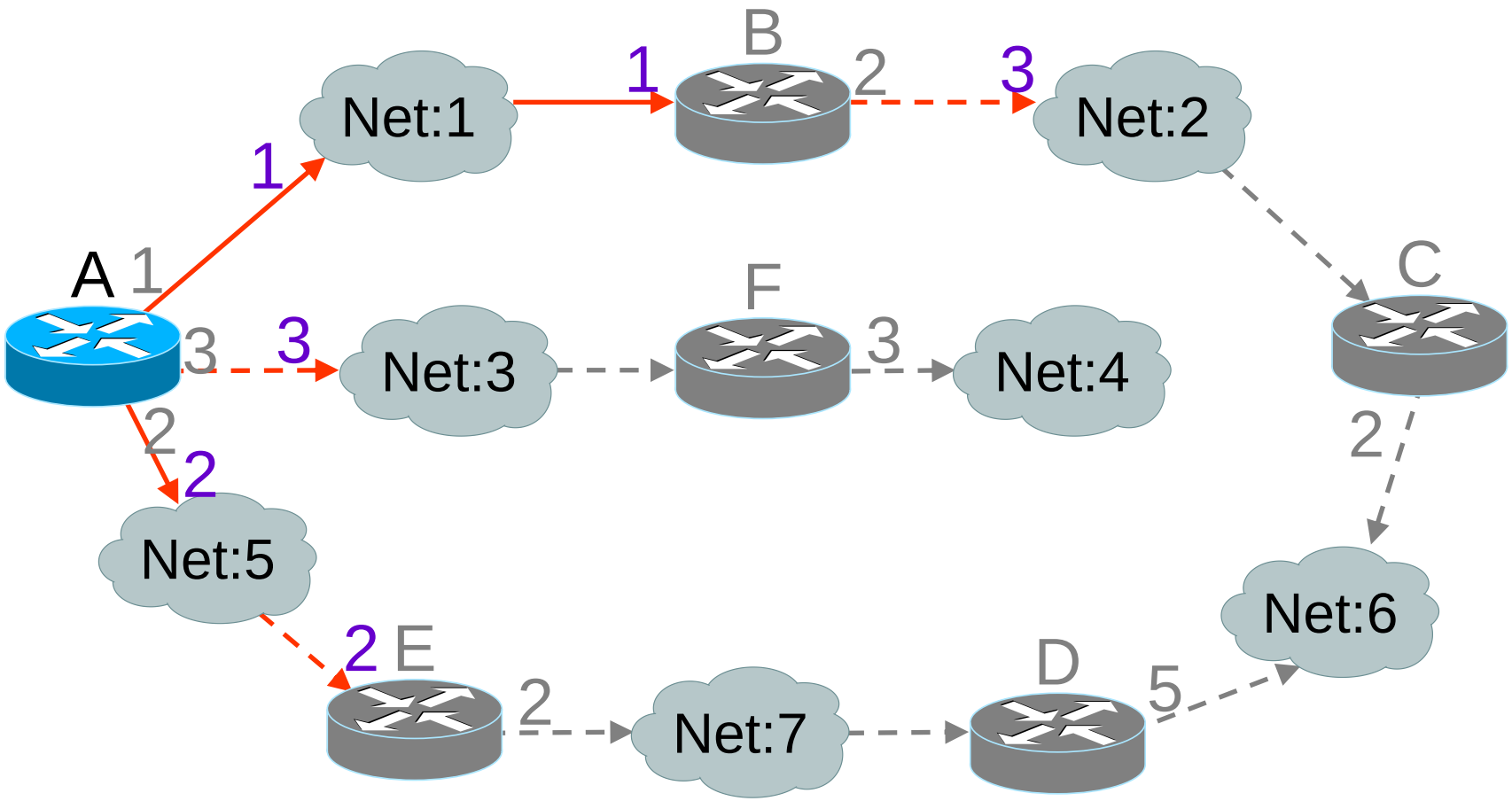
Dijkstra算法示例(4)



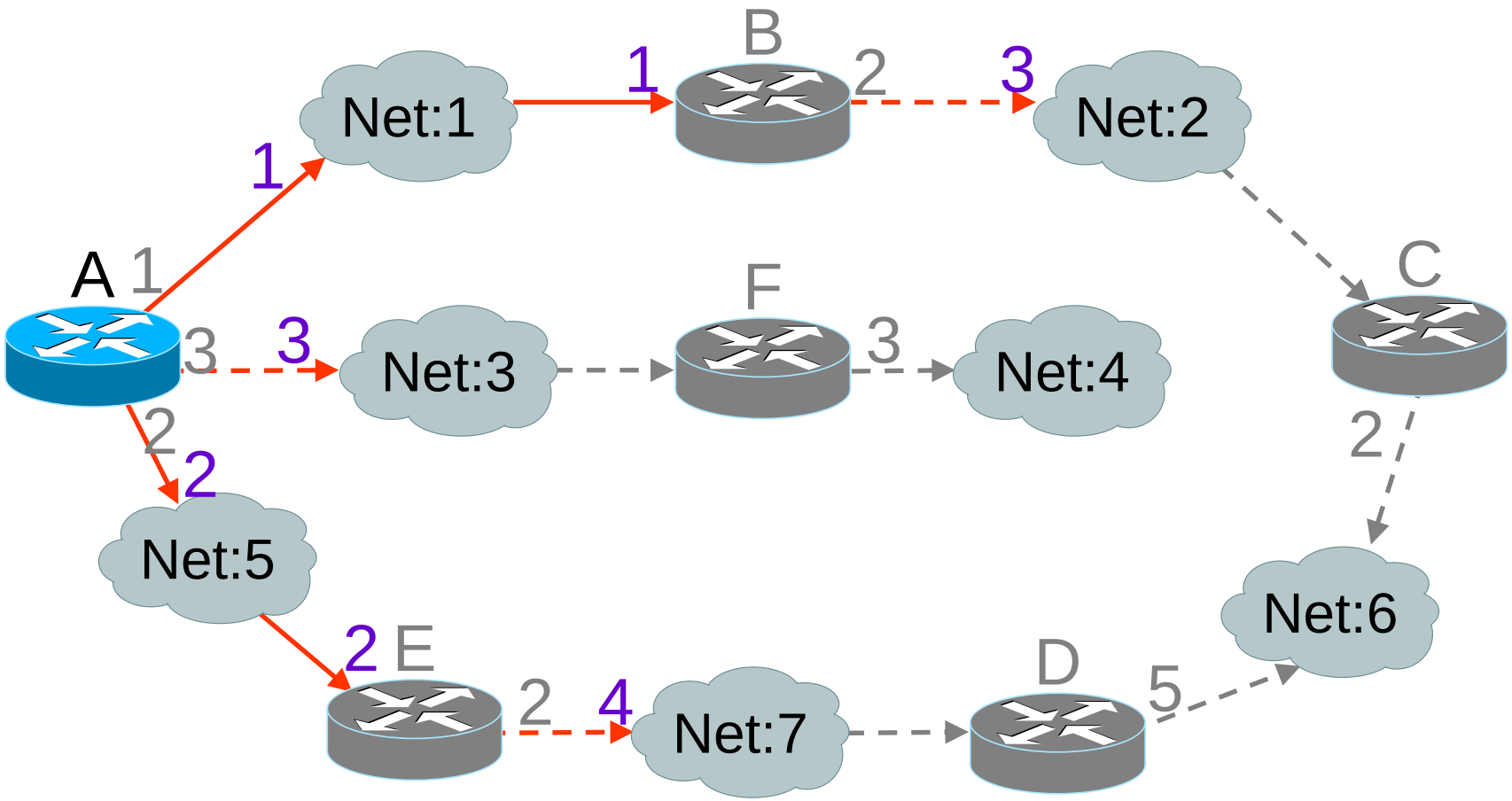
Dijkstra算法示例(5)



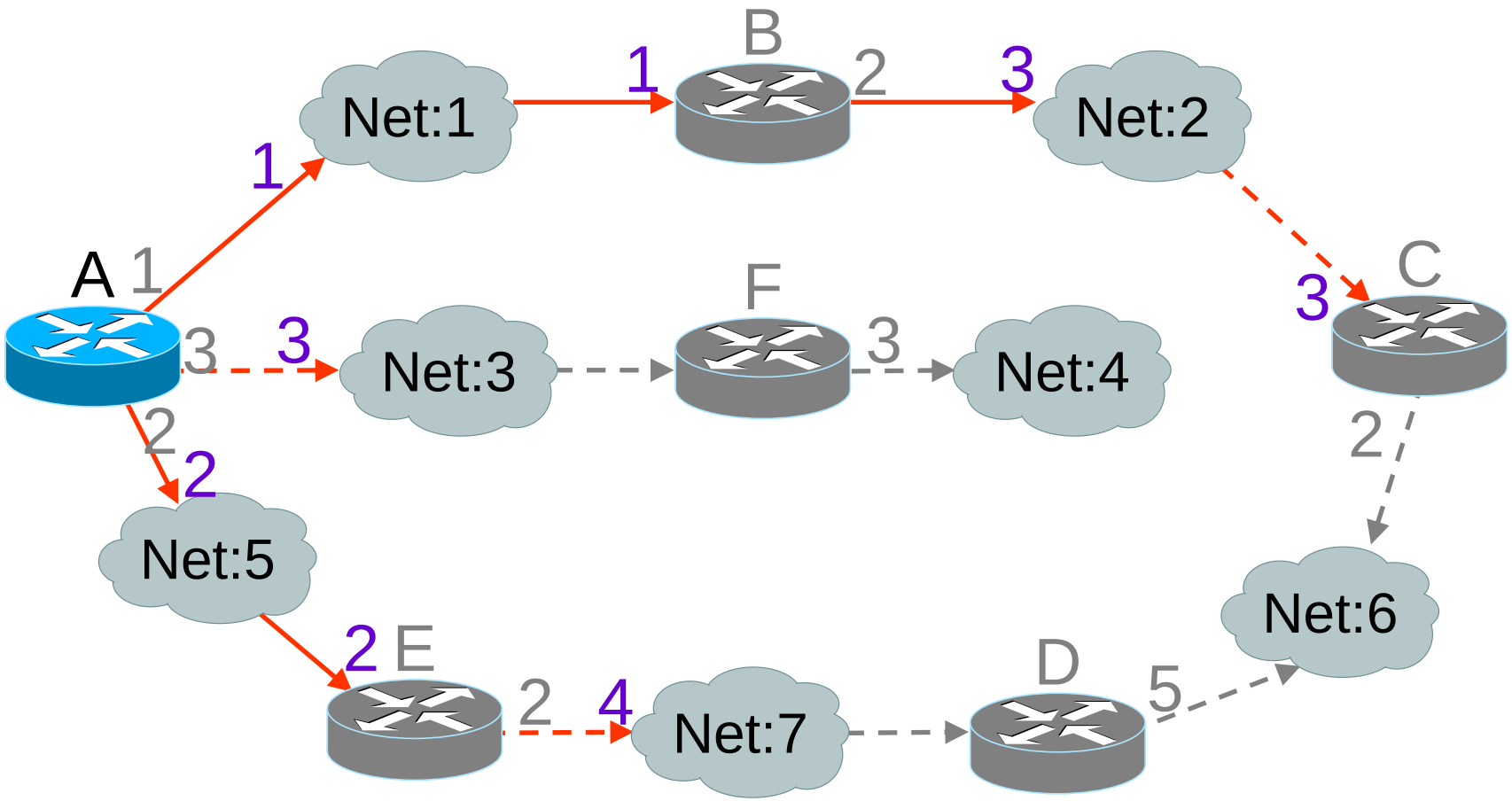
Dijkstra算法示例(6)



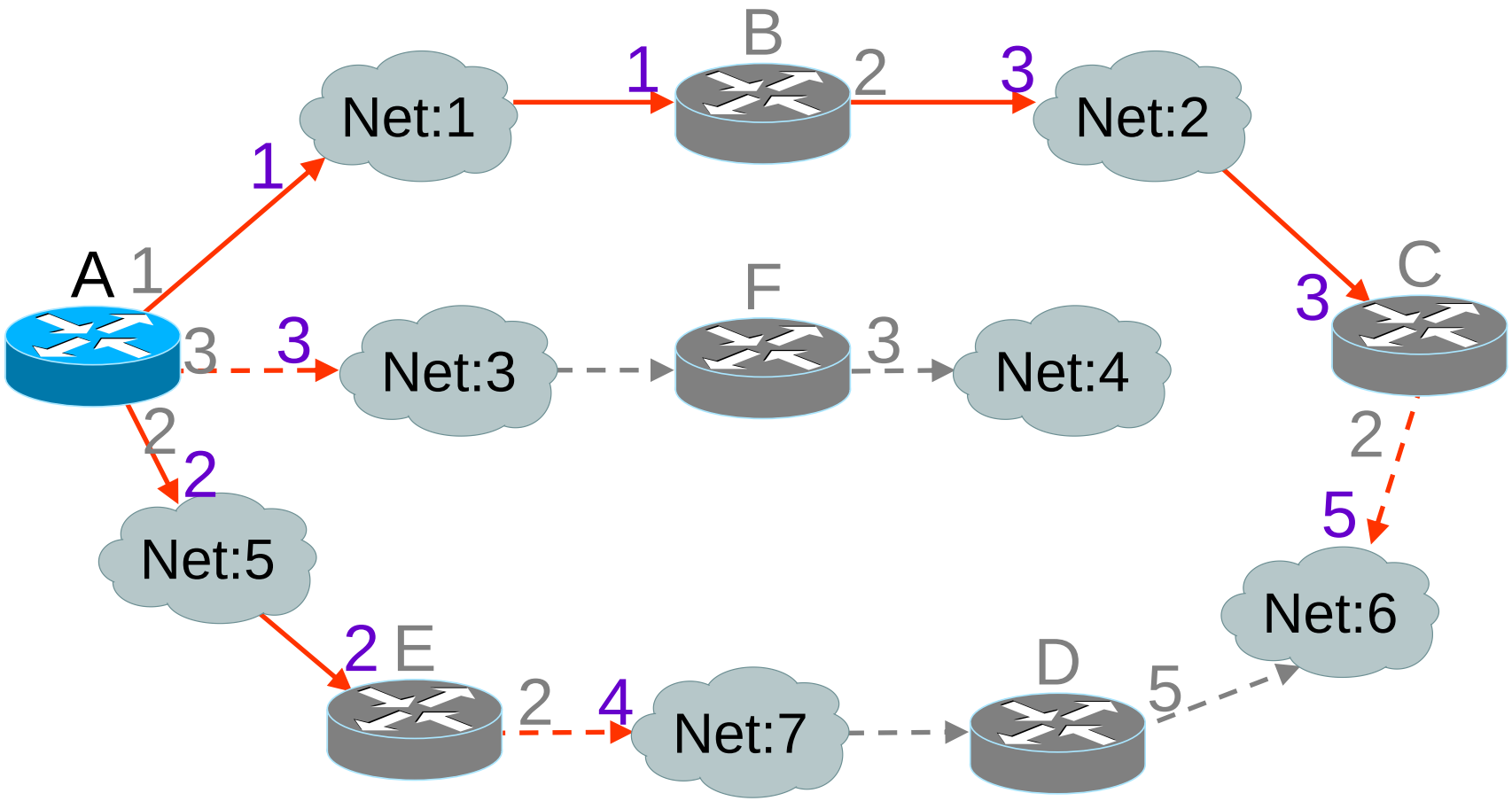
Dijkstra算法示例(7)



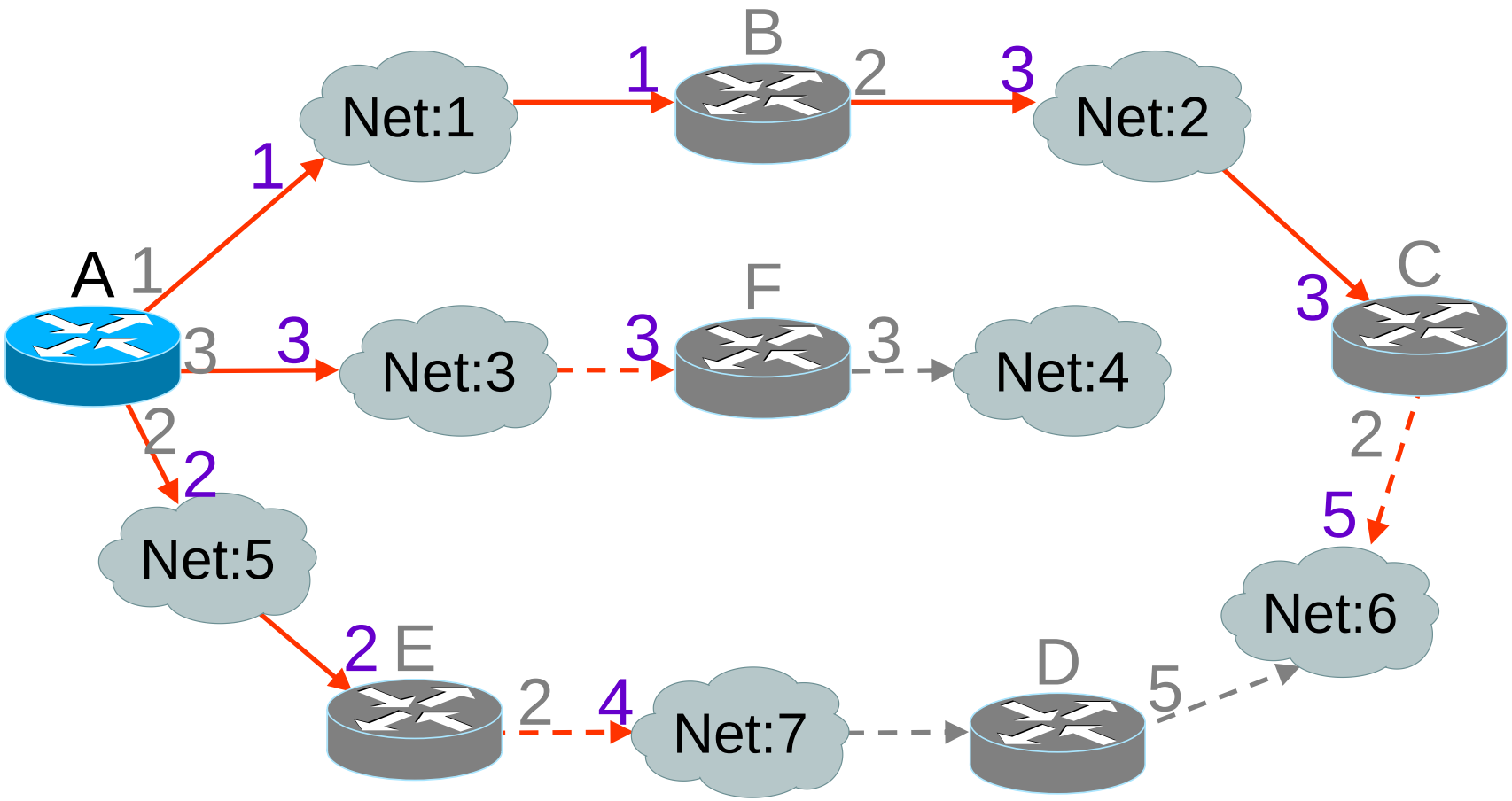
Dijkstra算法示例(8)



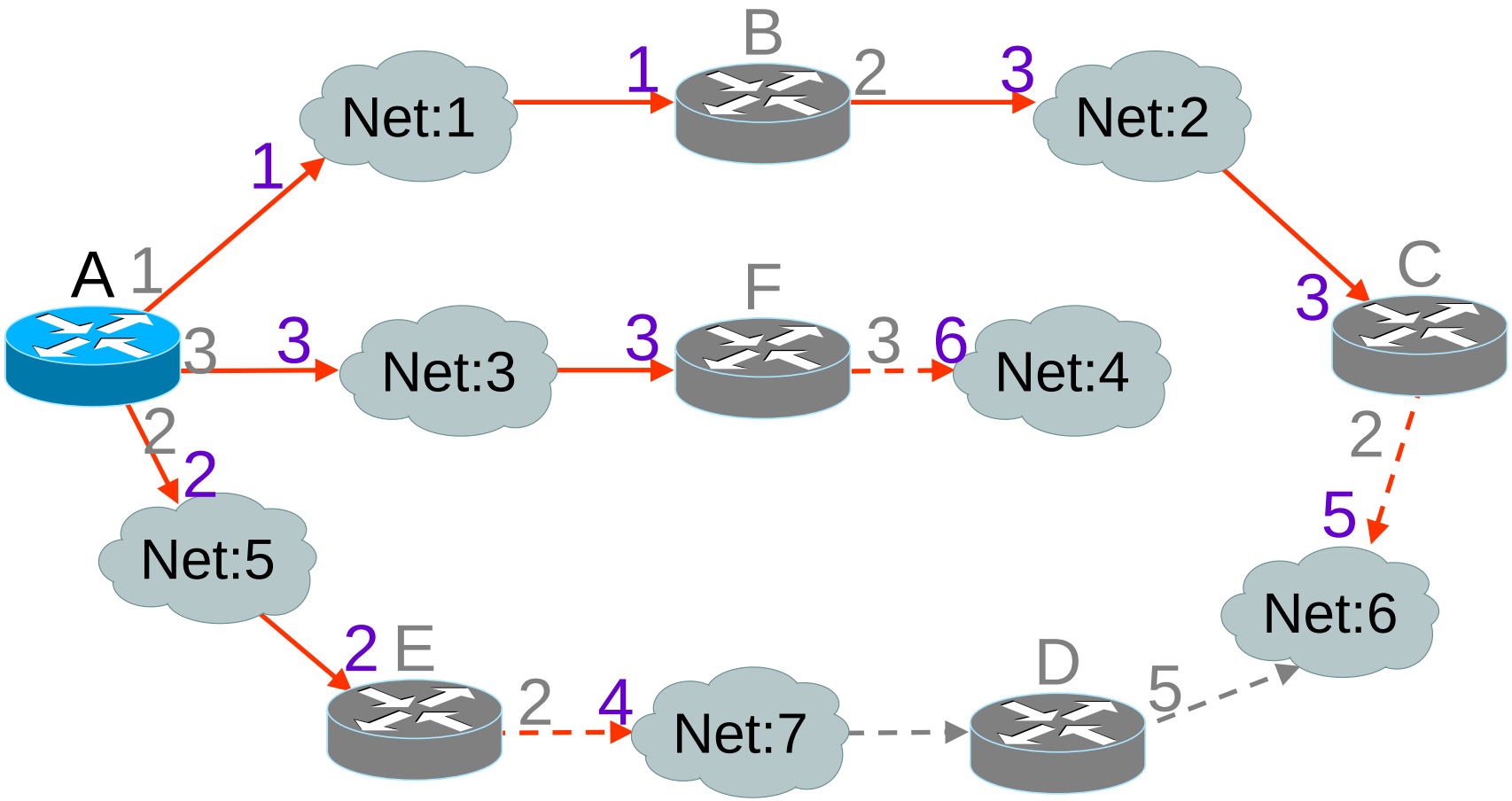
Dijkstra算法示例(9)



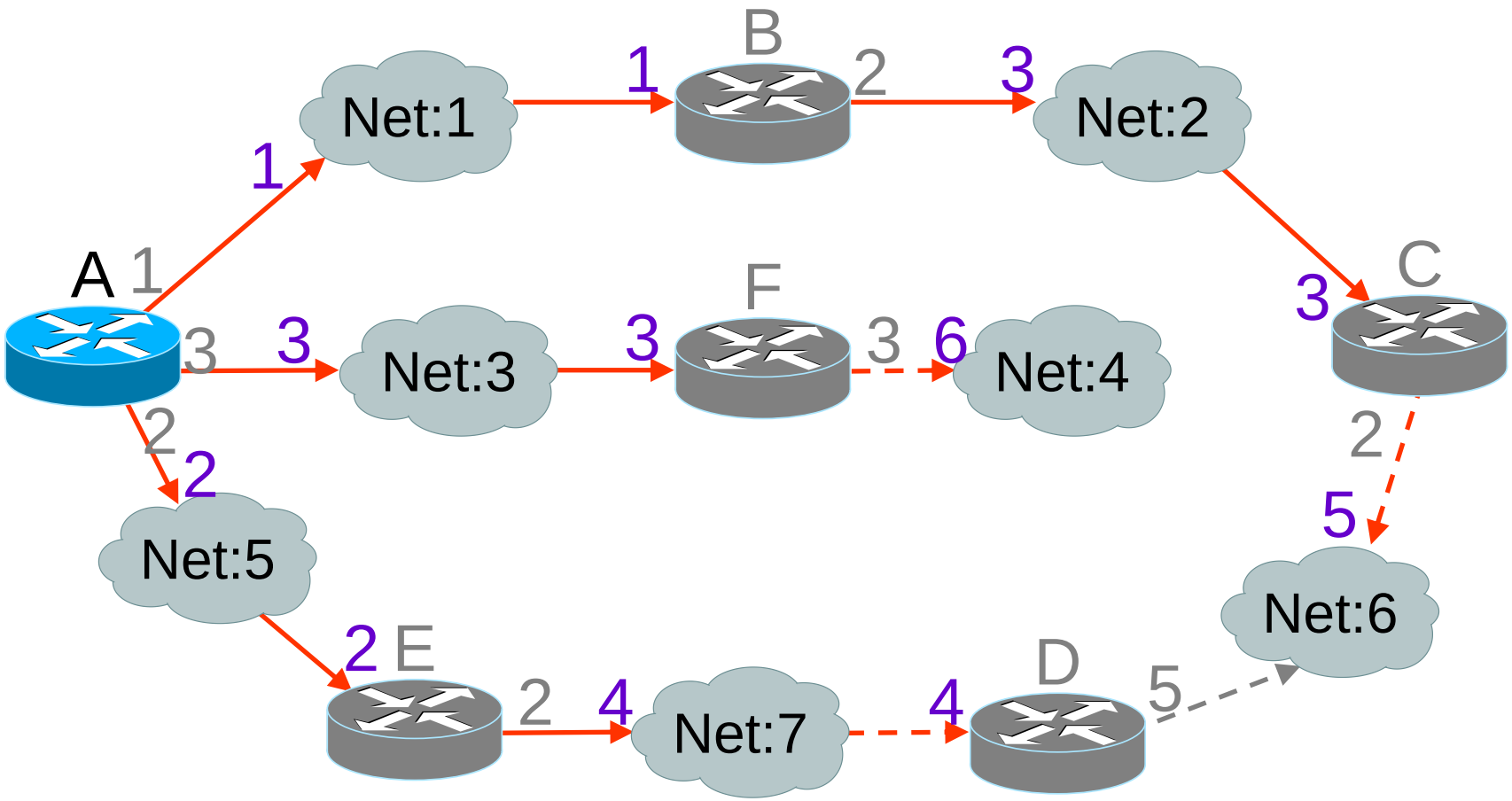
Dijkstra算法示例(10)



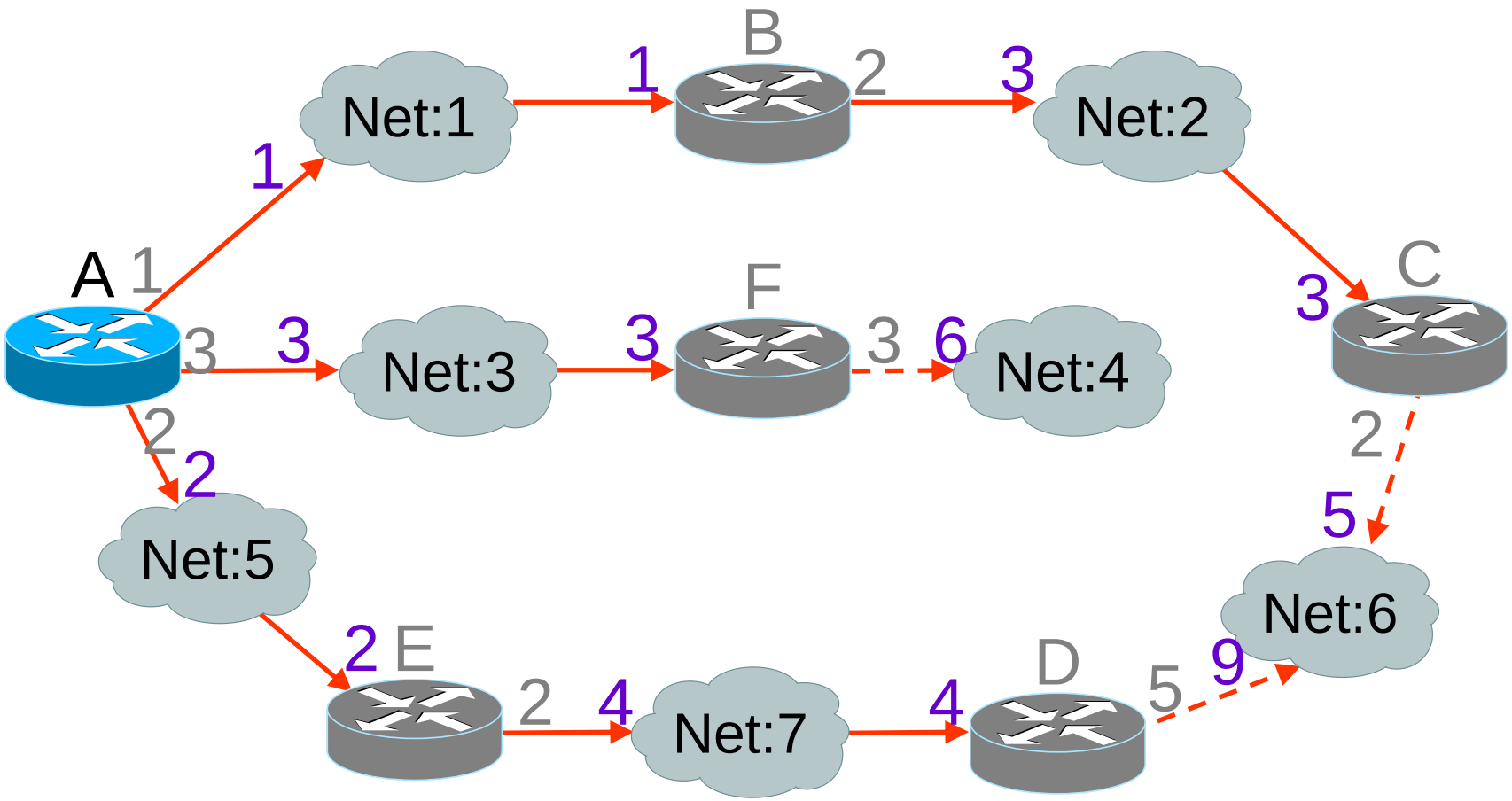
Dijkstra算法示例(11)



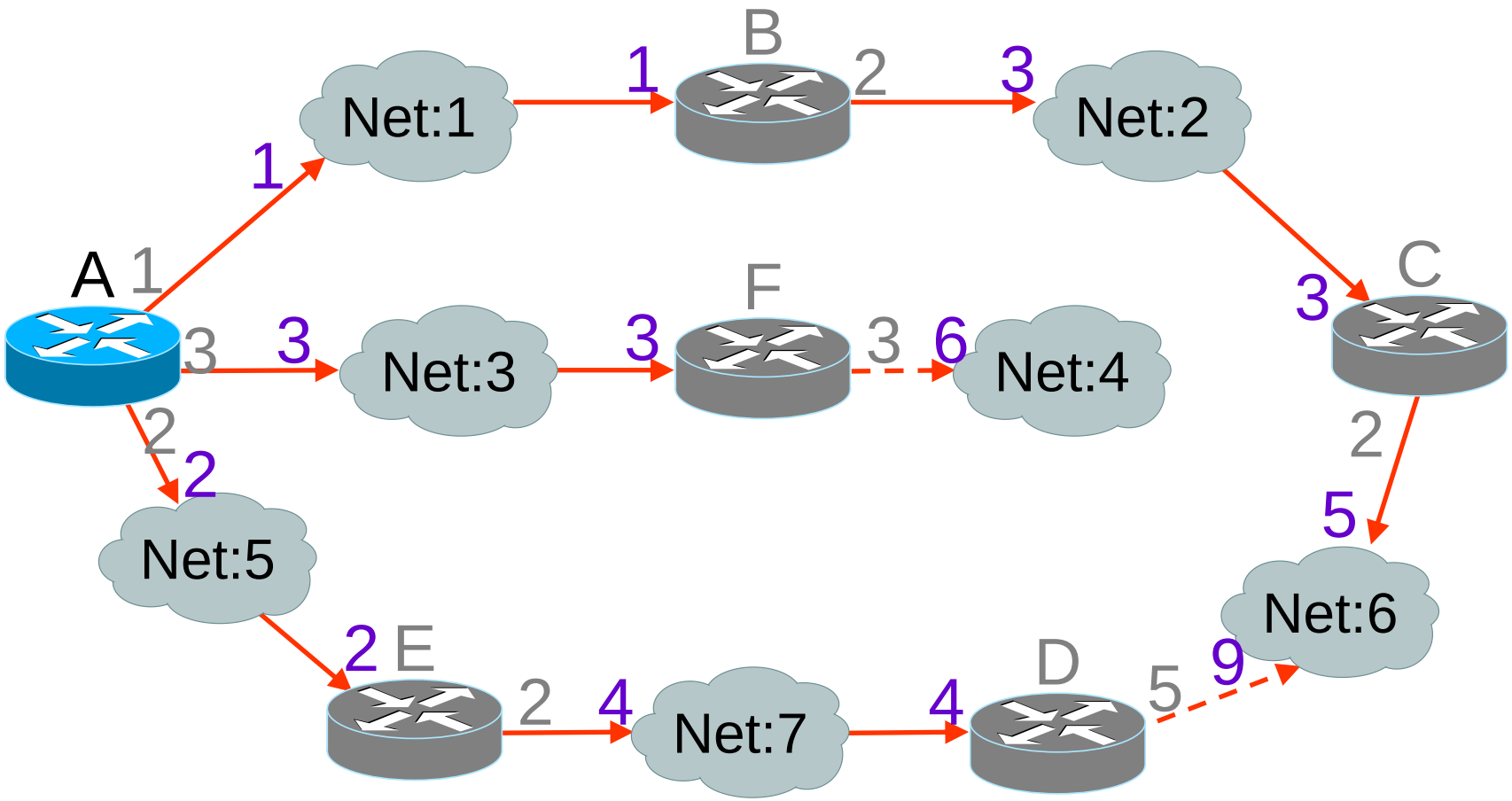
Dijkstra算法示例(12)



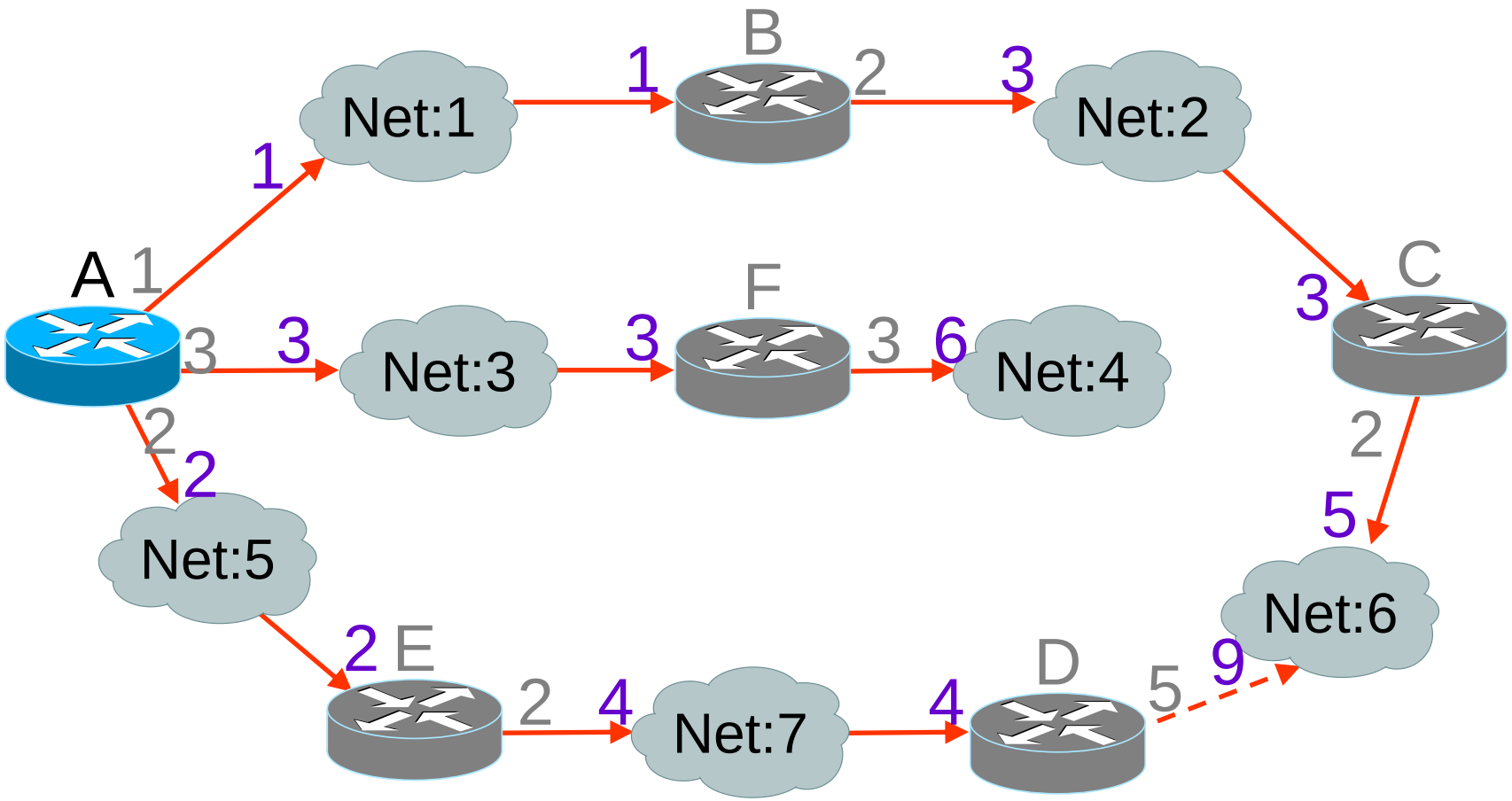
Dijkstra算法示例(13)



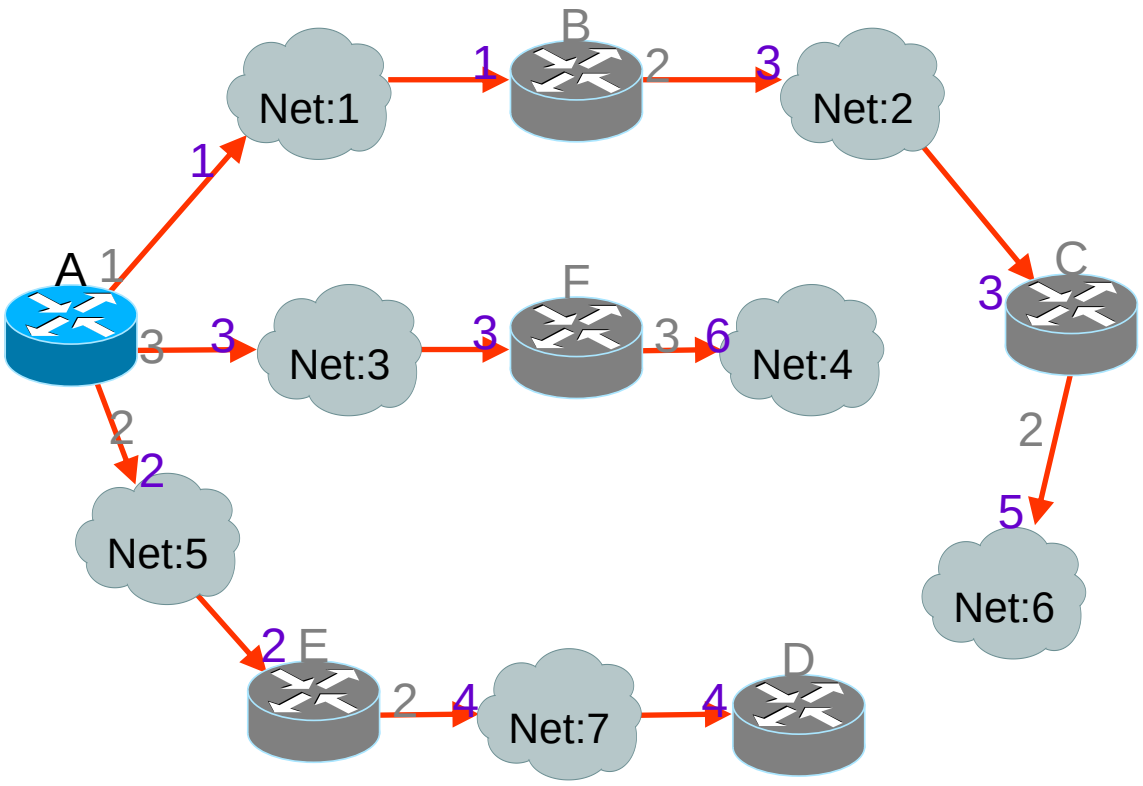
Dijkstra算法示例(14)



Dijkstra算法示例(15)



Dijkstra算法示例

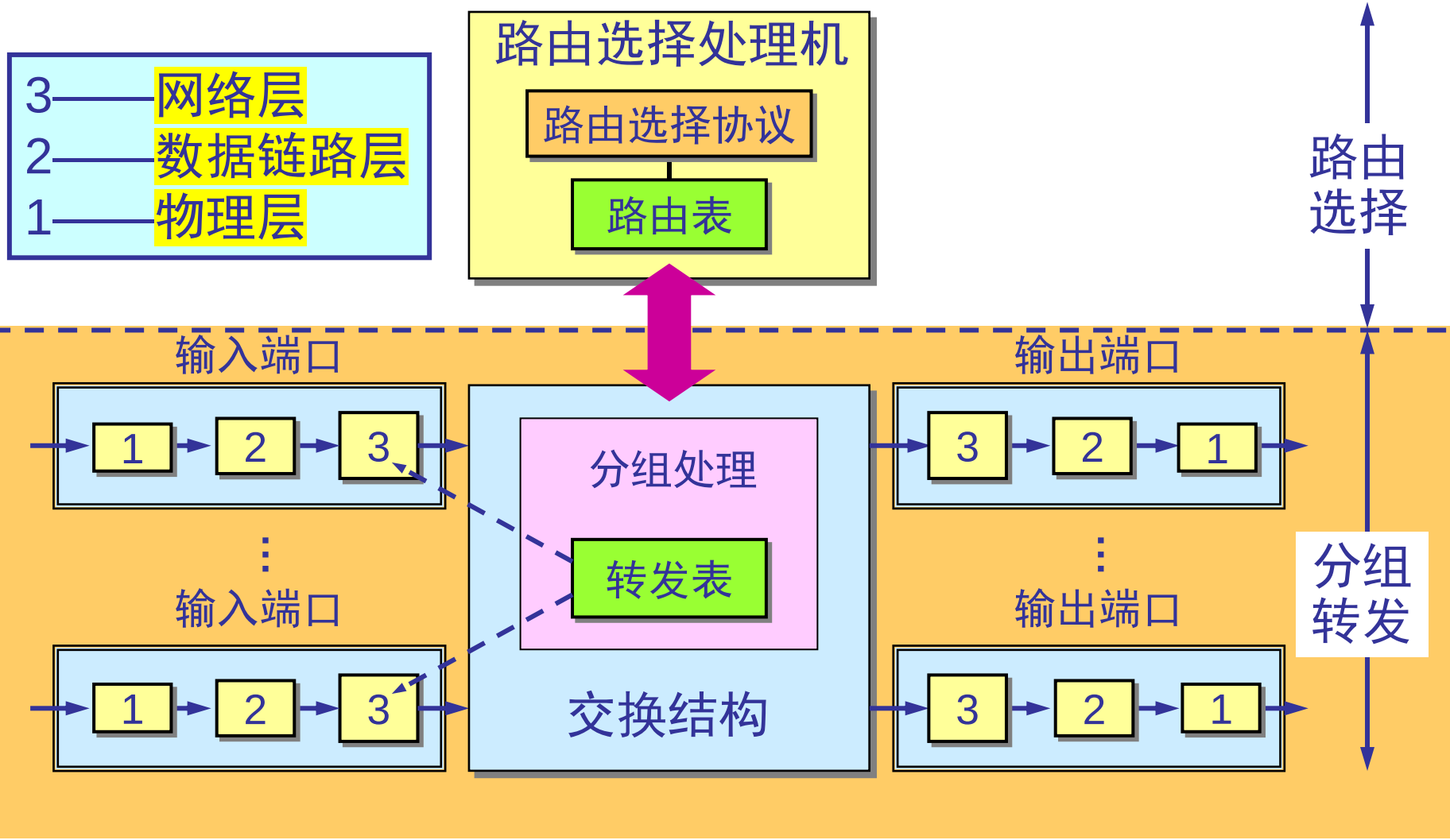


目标网络	费用	下一个路由器
1	1	-
2	3	B
3	3	-
4	6	F
5	2	-
6	5	B
7	4	E

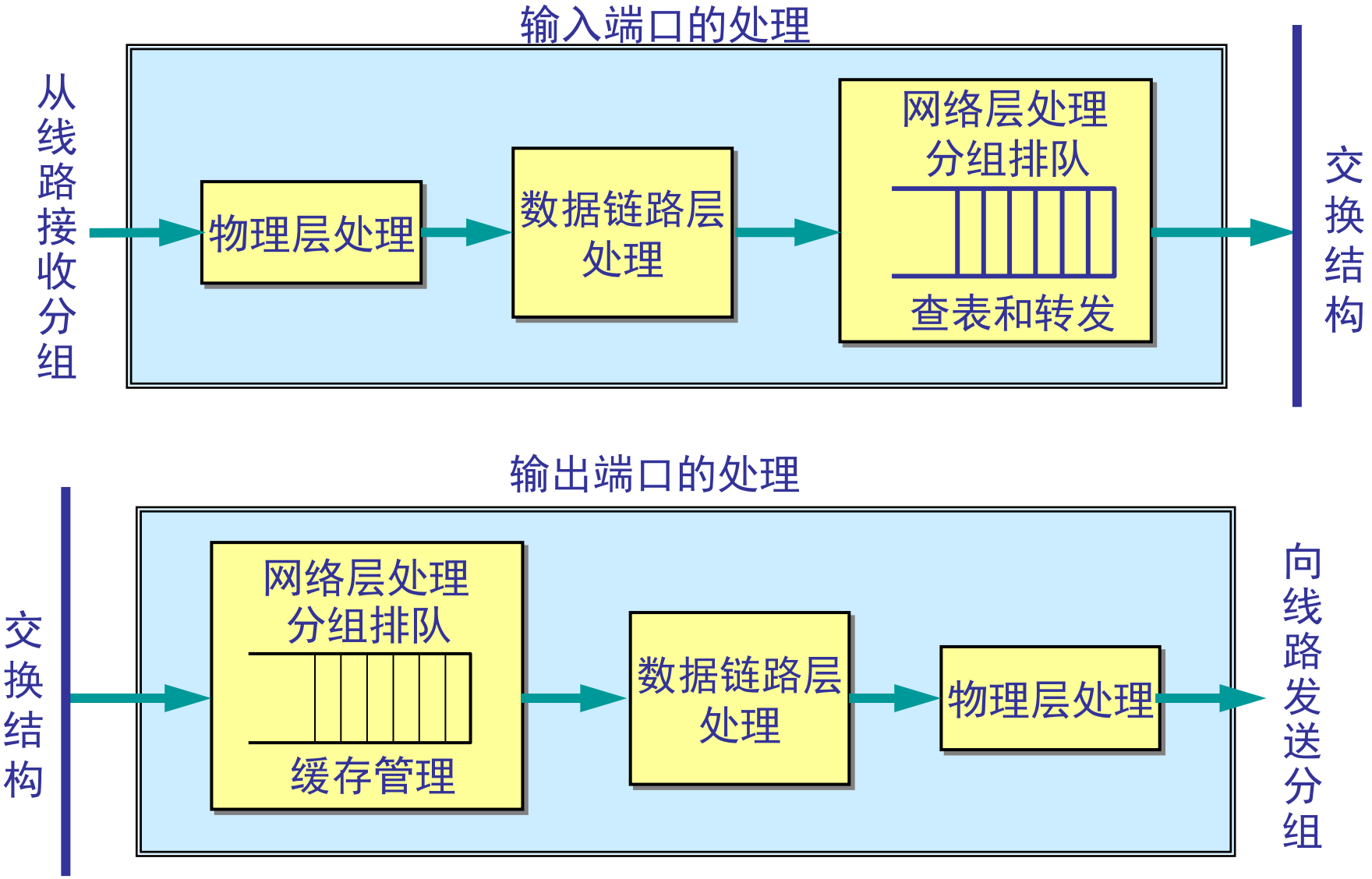
算法比较

- 算法比较
  - 距离向量路由算法
  - 链路状态路由算法

典型的路由器的结构框图



路由器输入、输出端口


输入端口的处理
从线路接收分组
物理层处理
数据链路层处理
网络层处理
分组排队
查表和转发
交换结构
输出端口的处理
交换结构
网络层处理
分组排队
缓存管理
数据链路层处理
物理层处理
向线路发送分组

5.4 拥塞控制和流量控制

- 拥塞控制：防止整个网络或网络的一部分出现过多的数据包
- 流量控制：保证发送方发送的信息量不会超过接收方的接收能力

拥塞控制的通用原则

- 开环控制(Open loop)
- 闭环控制(Close loop)
  - 显式反馈：拥塞点发警告
  - 隐式反馈：源端主动判断

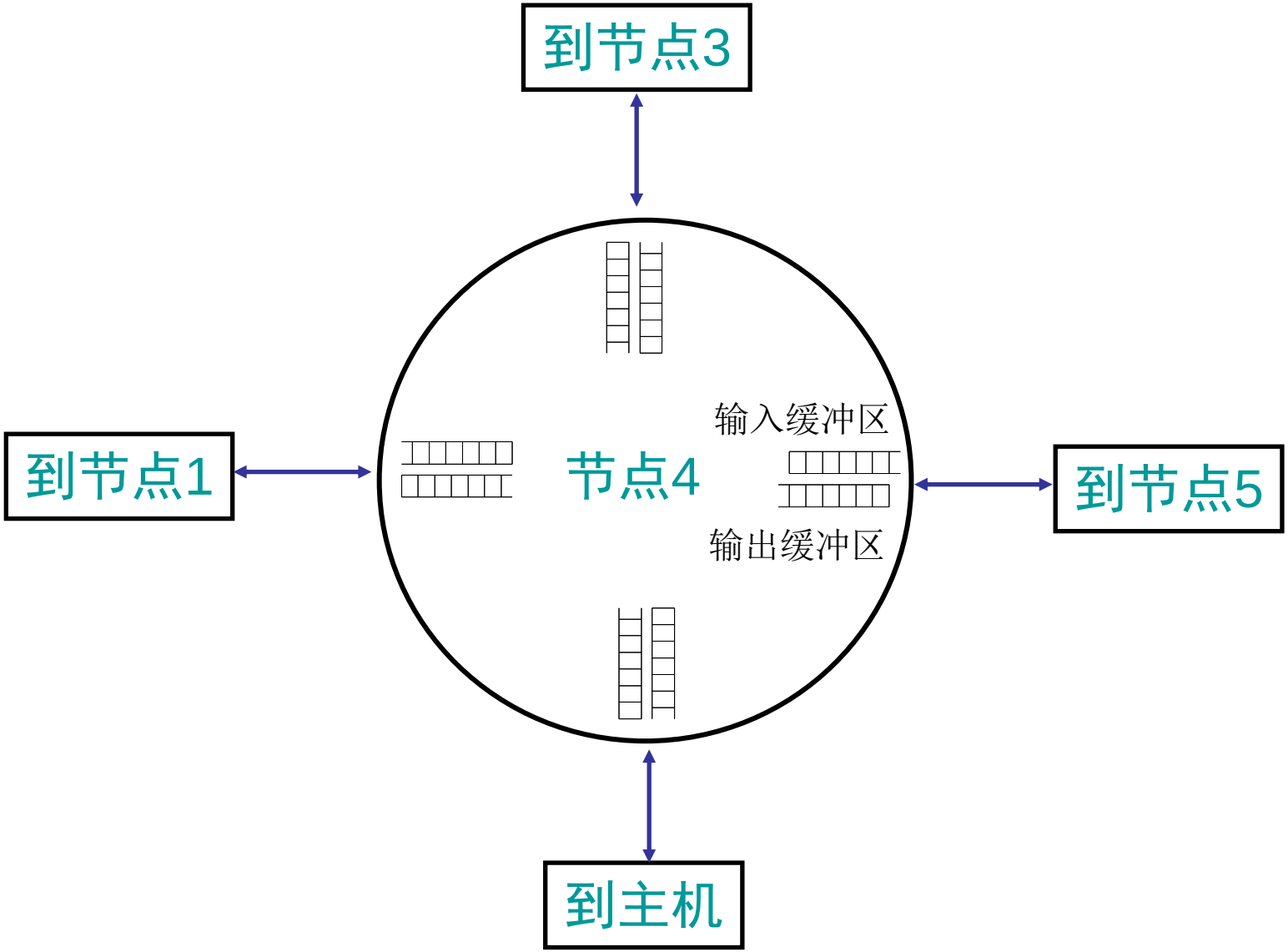
开环控制

- 通过良好的设计，避免问题的出现，确保问题在一开始就不会出现，不需要中途做修正
- 开环控制的方法：
  - 什么时候接受新的数据流
  - 什么时候开始丢弃数据报，丢弃哪些数据报？
  - 指定网络中各个节点的调度策略
- 这些方法的共同特点：做出决定的时候不考虑网络的当前状态

闭环控制

- 建立在反馈环路的基础上，由3部分组成：
  - 监视系统，检测何时、何地发生了拥塞
    - 检测的指标可以是丢包率、平均队列长度、由于超时引起的包重发、数据包延迟抖动等
  - 将检测收集的拥塞信息传递到能够采取行动的地方
    - 直接发包给相关节点
    - 利用包头中的某一位将拥塞通知邻居节点
    - 每个节点周期性地发出探测包，检查拥塞状况
  - 调整系统的运行，以改正问题

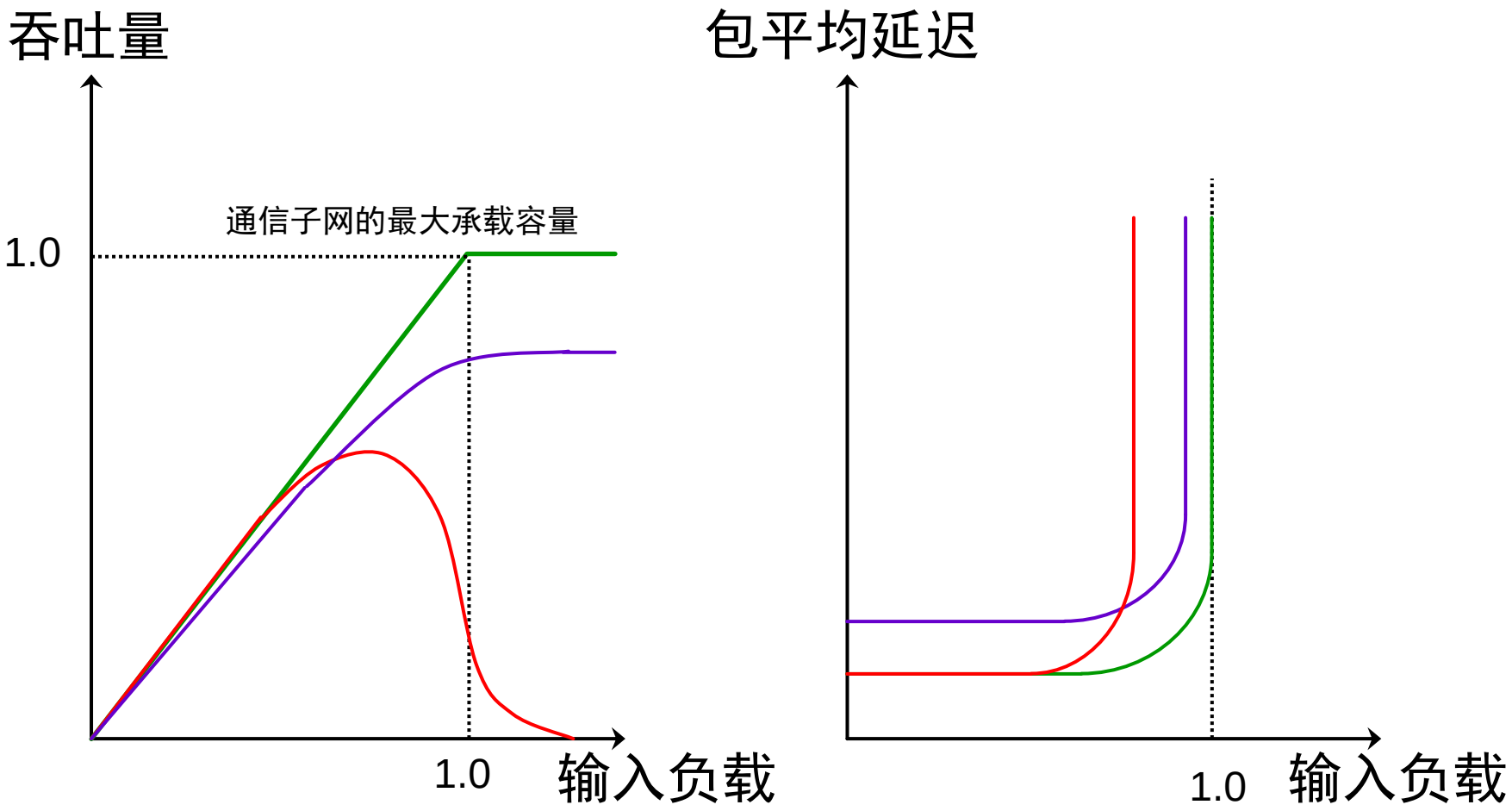
包交换结点的模型



5.4.1 拥塞控制

- 拥塞
  - 网络或其一部分出现过多的包，导致网络性能下降的现象。
- 产生的原因
  - 节点的处理速度—影响输入队列
  - 输出链路的传输速度—影响输出队列
- 对系统的影响
  - 系统吞吐量下降
  - 传输延迟增大
- 对策：增加资源，或者降低负载

拥塞对系统的影响



控制拥塞的方法(1)

- **预分配缓冲区**：常用于虚电路技术中，虚电路的建立会通知该节点为此虚电路预留缓冲区。
- **丢弃包**：节点上收到过多的包而来不及处理或无法发送出去时，可丢弃一部分包。对突发性通信造成的拥塞有效。丢包的常用机制：
  - DropTail
  - RED
- **限制网内包数量**：限制进入网内的包的数目，达到控制拥塞的目的。例如，在网内设置若干个许可证。

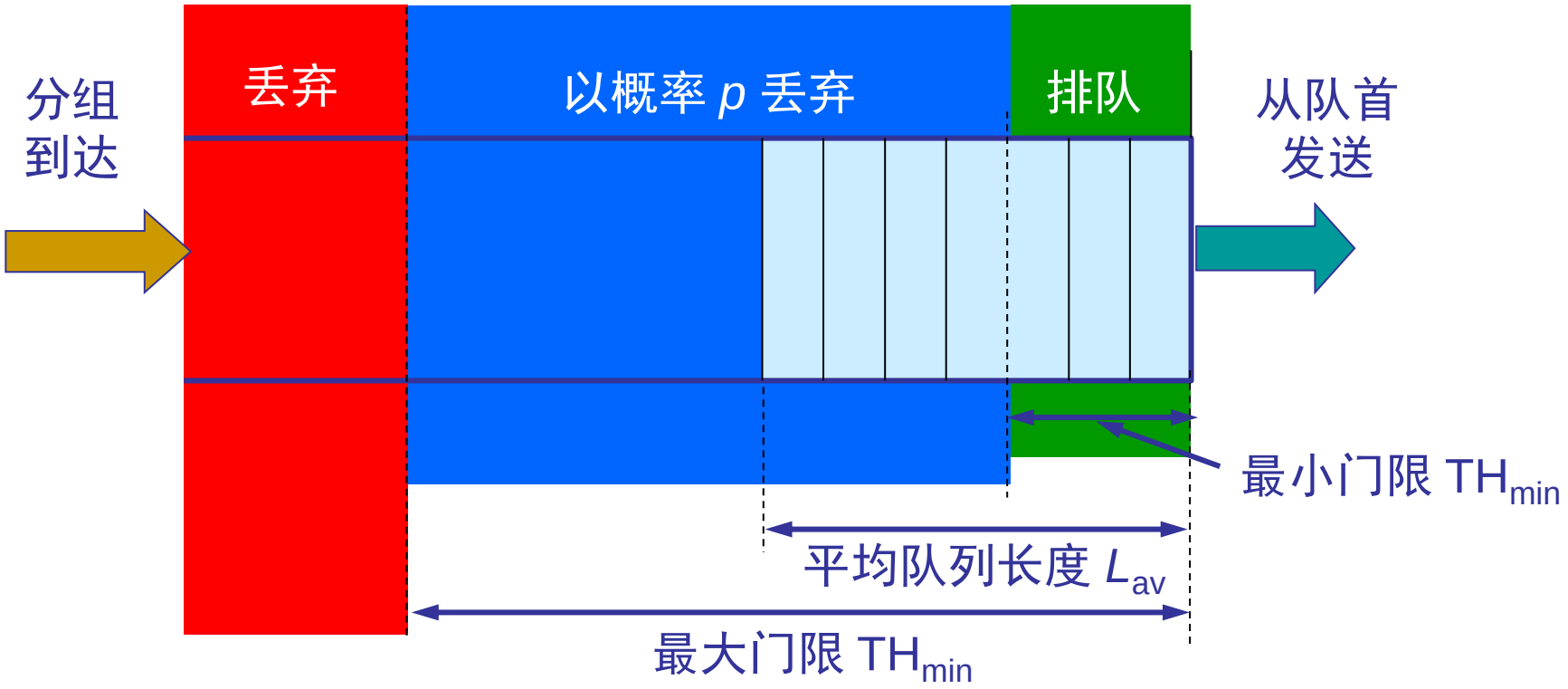
控制拥塞的方法(2)

- **流量控制**：接收端调节发送端发送数据的速率，防止到达接收端的数据速率超过接收端的处理速率。本质上流量控制和拥塞控制是不同的概念：
  - 流量控制是端到端
  - 拥塞控制涉及中间节点
- **阻塞包**：每个节点都监视其所有输出链路的使用情况。视情况决定是否向源结点发送阻塞包。

随机早期检测RED

- RED -- Random Early Detection
- 使路由器的队列维持两个参数，即队列长度最小门限 TH_{min} 和最大门限 TH_{max}。
- RED 对每一个到达的数据报都先计算平均队列长度 L_{AV}。
- 若平均队列长度小于最小门限 TH_{min}，则将新到达的数据报放入队列进行排队。
- 若平均队列长度超过最大门限 TH_{max}，则将新到达的数据报丢弃。
- 若平均队列长度在最小门限 TH_{min} 和最大门限 TH_{max} 之间，则按照某一概率 p 将新到达的数据报丢弃。

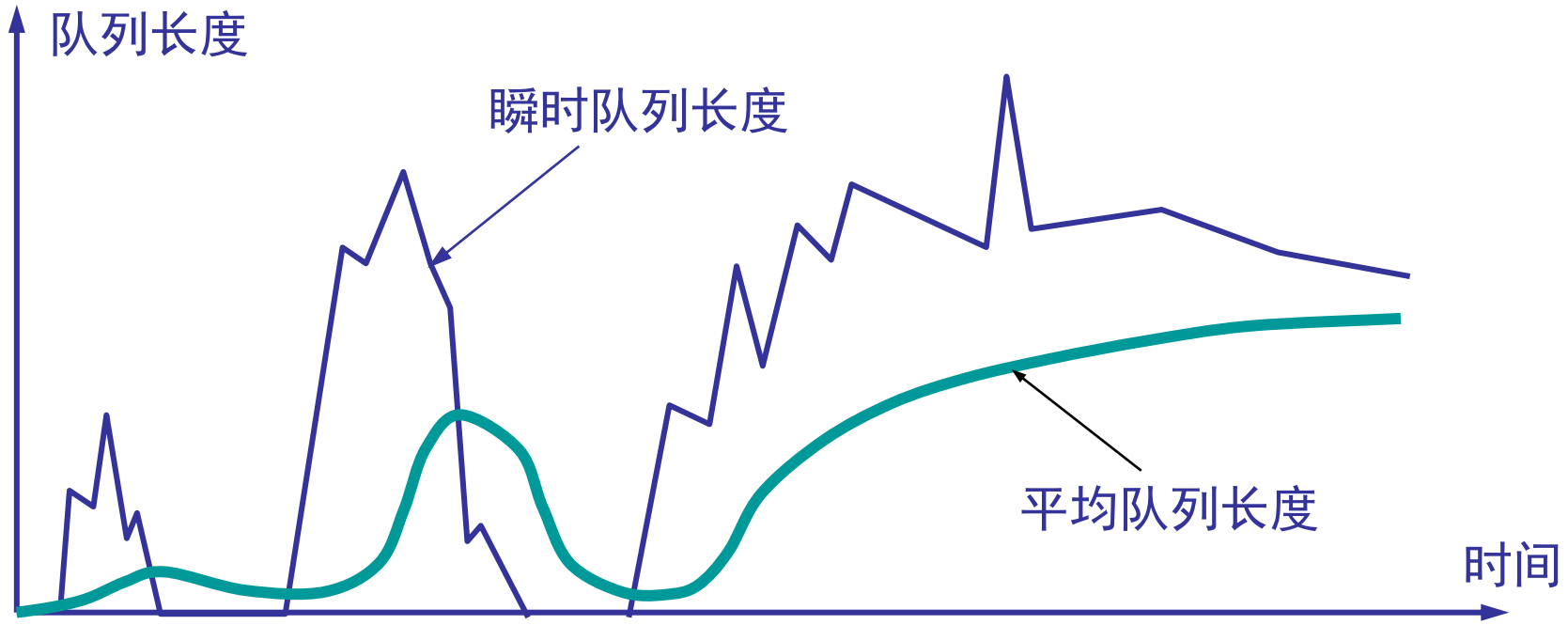
RED将到达队列划分成为三个区域



瞬时队列长度和平均队列长度

- 要谨慎调整时间尺度
  - 如果每次有2个包到达—stop
  - 如果每次路由器空闲—go

} 导致系统剧烈震荡

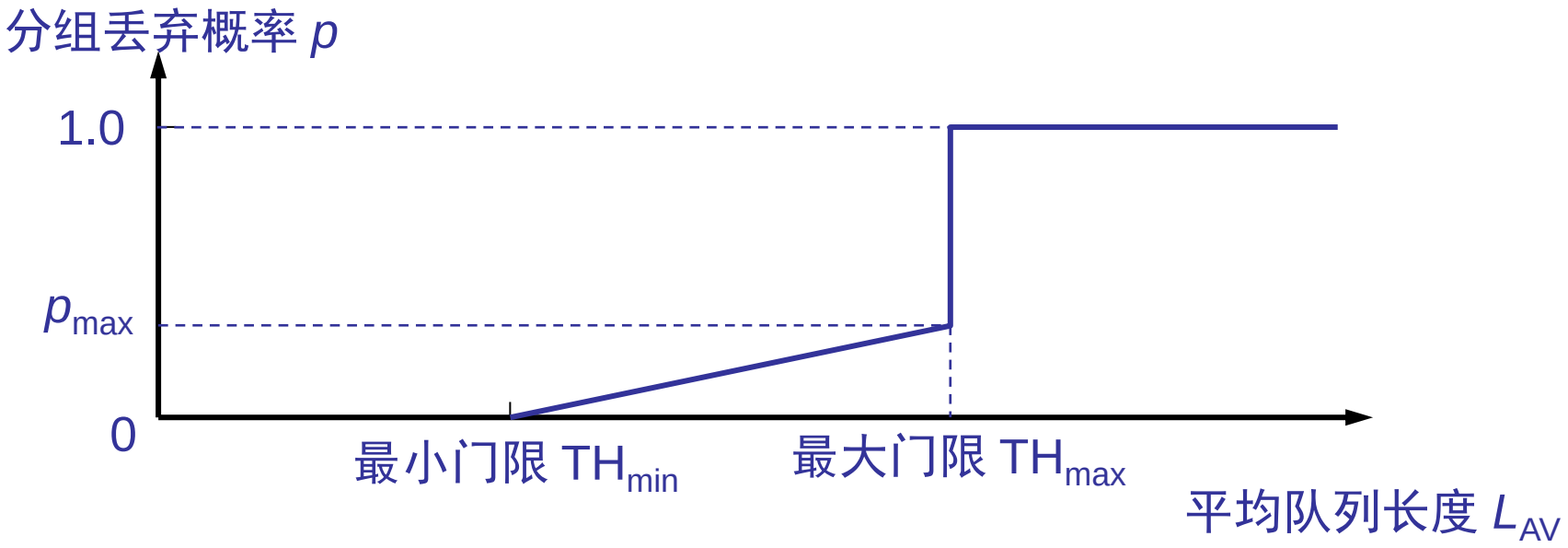


L_{AV}=(1-δ)× 旧的L_{AV} + δ×当前队列长度

丢弃概率p与TH_{min}和Th_{max}的关系

- 当 $L_{AV} < Th_{min}$ 时，丢弃概率 $p = 0$。
- 当 $L_{AV} > Th_{max}$ 时，丢弃概率 $p = 1$。
- 当 $TH_{min} < L_{AV} < TH_{max}$ 时，$0 < p < 1$。

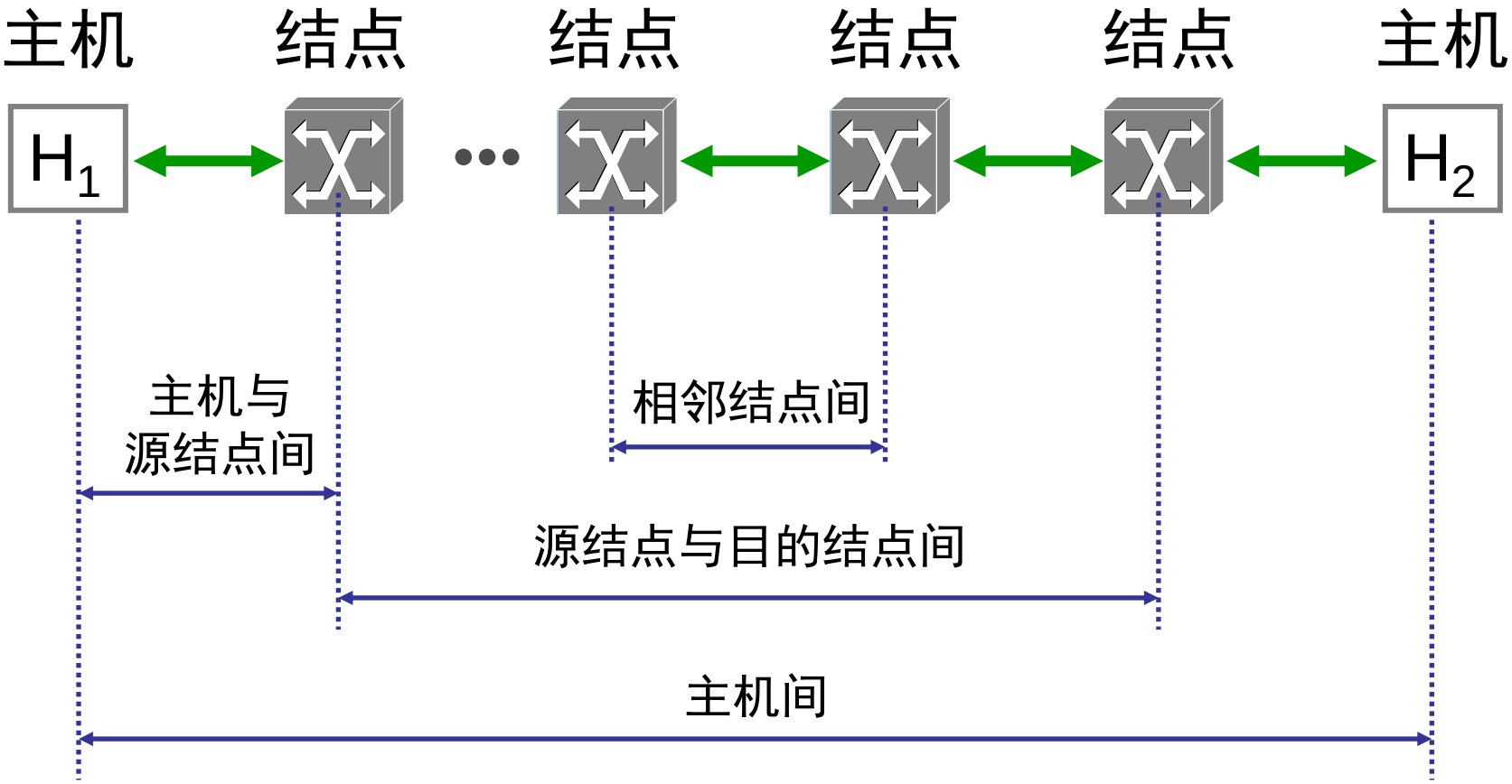
  例如，按线性规律变化，从 0 变到 p_{max}。



5.4.2 流量控制

- 流量控制是一种端到端的控制。
- 流量控制可在多个层次上进行：
  - 主机—主机间
  - 源节点—目的节点间
  - 主机—源节点间
  - 相邻节点间

流量控制层次



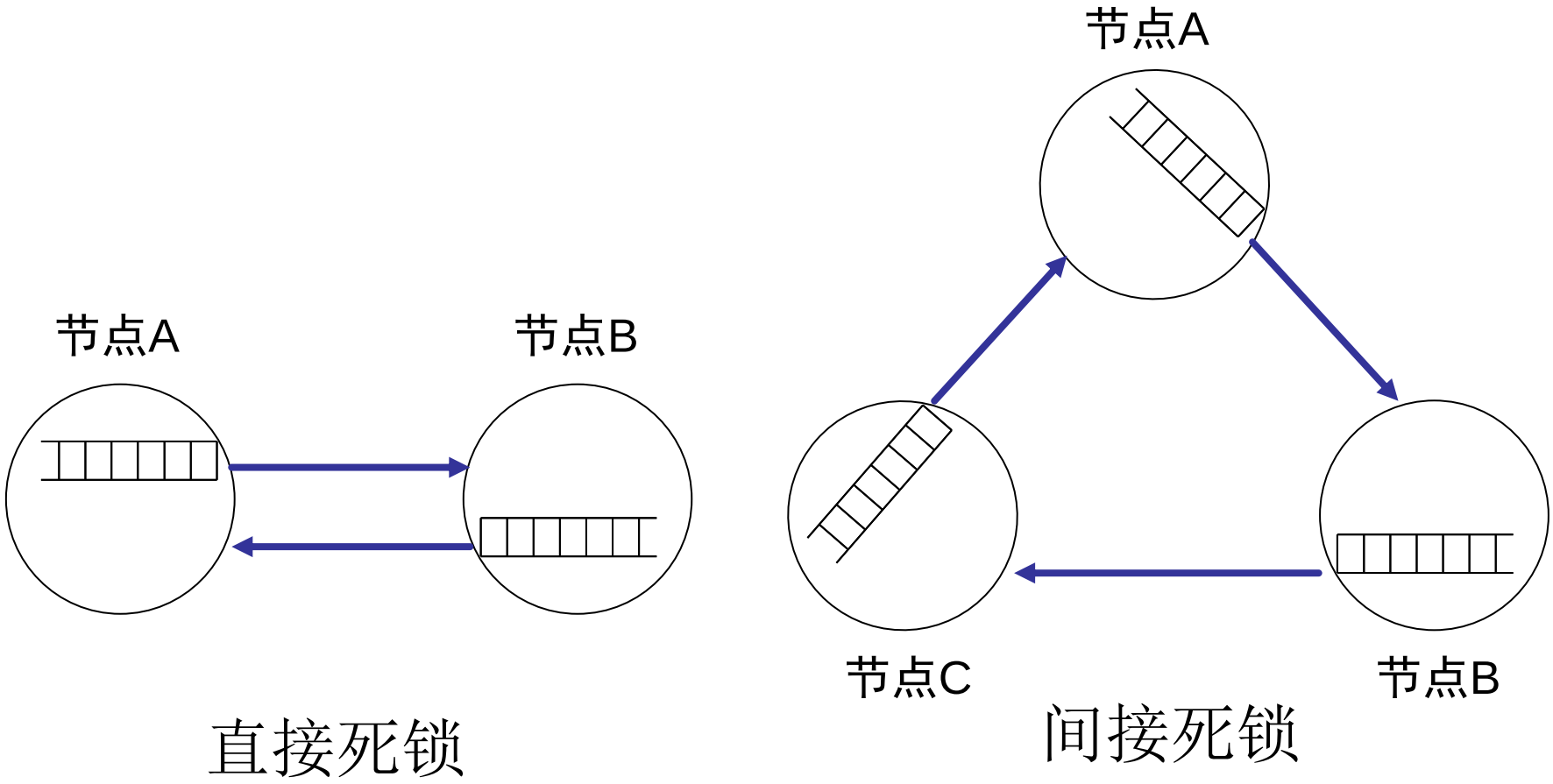
主机和源节点间的流量控制

- 通过控制进入通信子网的信息量，从而防止整个网内的缓冲区产生拥塞。
- 可基于对网络拥塞的测量采取控制手段
- 采用的主要方法：
  - 停止等待流量控制
  - 缓冲区预约
  - 许可证方案

源节点和目的节点之间的流量控制

- 其任务是和通信子网的工作方式紧密相关的。
- 如果通信子网采用虚电路工作方式，该层流量控制的任务就比较轻。因为虚电路方式本身要求有基本的缓冲区，包沿固定路径传送，且包按顺序到达目的节点。
- 如果通信子网采用数据报方式工作，而缓冲区分配采用先来先服务且全部分配的方法，则有可能产生存储转发死锁。

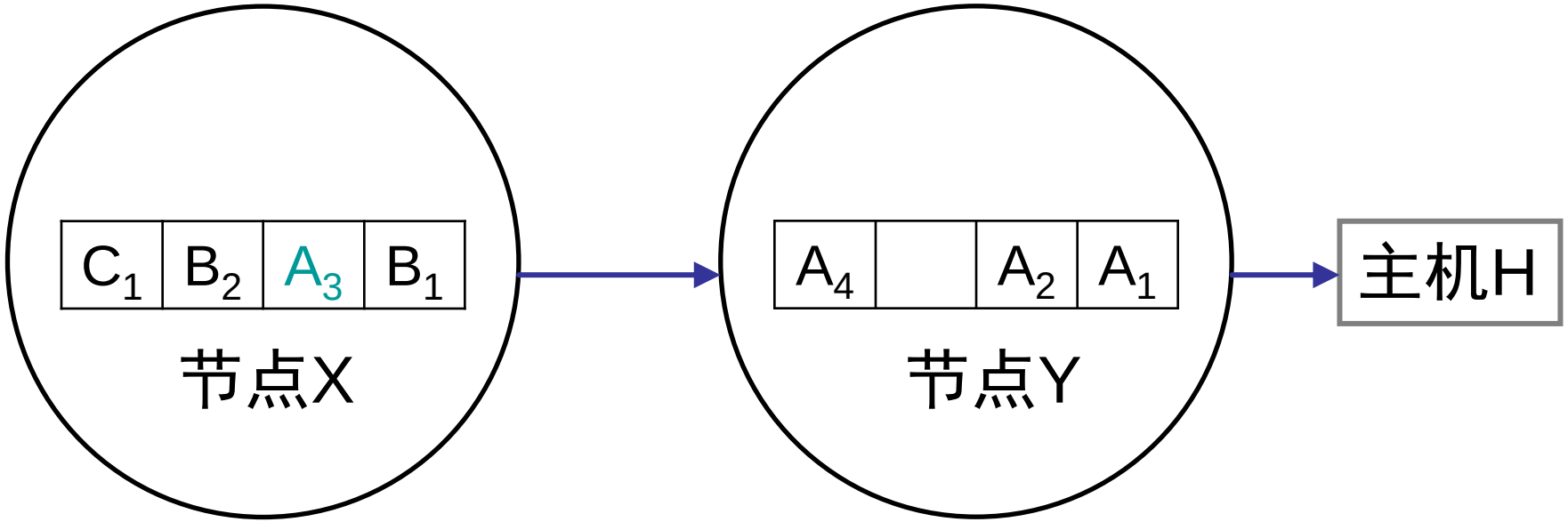
存储转发死锁



直接死锁　　间接死锁

- 解决办法：输入链路预留缓存，并允许丢包

重装死锁



- 解决办法：分配重装缓冲、并预约缓冲

5.5 IP协议

- IP协议是TCP/IP协议族中的核心协议。所有的TCP、UDP、ICMP、IGMP数据都是以IP数据报格式传输。

传输层

网络上层

- IP协议为高层提供不可靠、无连接的数据报通信。

网络层应提供什么样的服务？

- 在计算机网络领域，网络层应该向运输层提供怎样的服务？
  - 争论焦点的实质就是：在计算机通信中，可靠交付应当由谁来负责？是网络还是端系统？
- 因特网采用的设计思路
  - 网络层向上只提供简单灵活的、无连接的、尽最大努力交付的数据报服务。
  - 网络在发送分组时不需要先建立连接。每一个分组（即 IP 数据报）独立发送，与其前后的分组无关（不进行编号）。
  - 网络层不提供服务质量的承诺。即所传送的分组可能出错、丢失、重复和失序（不按序到达终点），当然也不保证分组传送的时限。

尽最大努力交付的好处

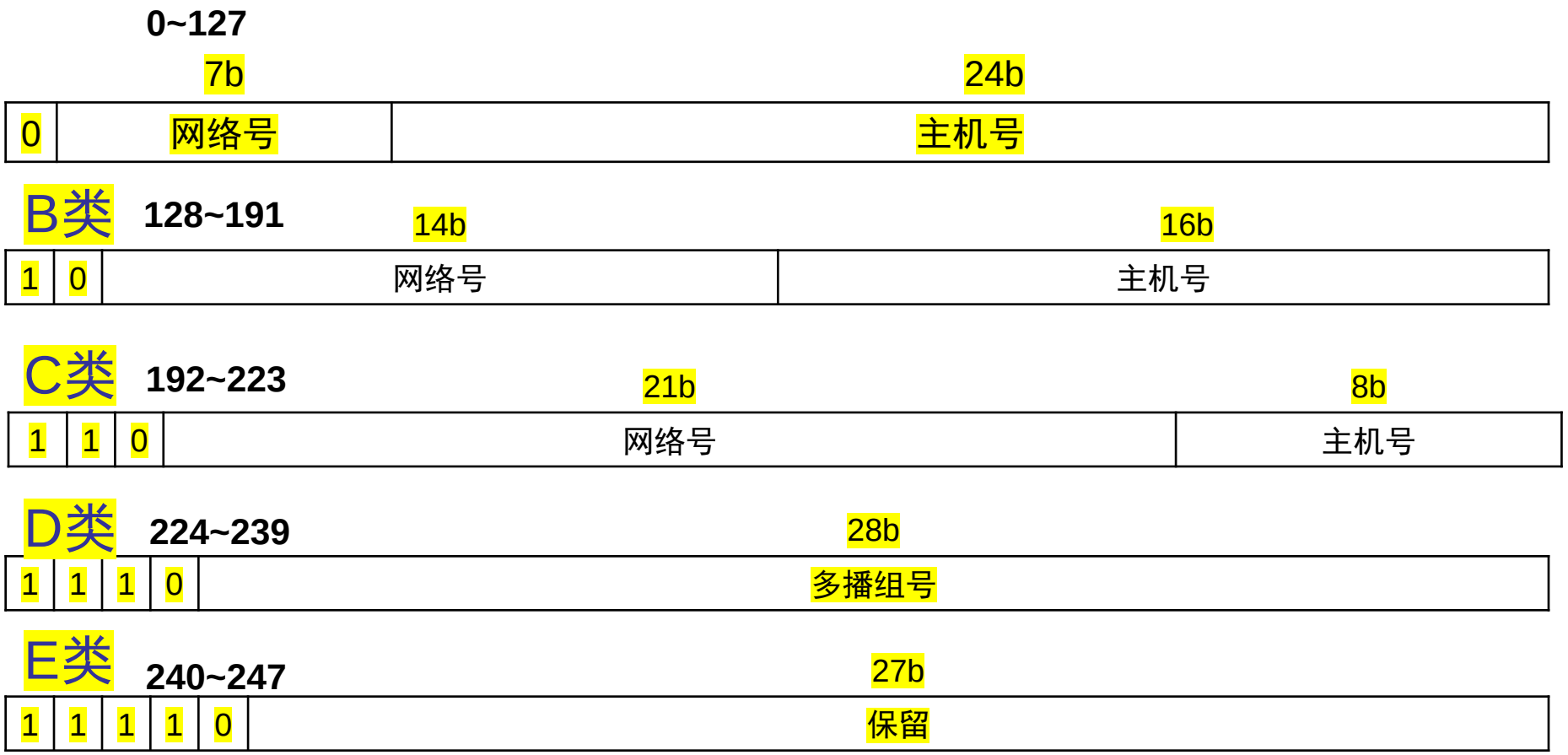
- 传输网络不提供端到端的可靠传输服务，使得网络中的路由器可以做得比较简单，而且价格低廉（与电信网的交换机相比较）。
- 如果主机（即端系统）中的进程之间的通信需要是可靠的，那么就由网络的主机中的运输层负责（包括差错处理、流量控制等）。
- 采用这种设计思路的好处是：网络的造价大大降低，运行方式灵活，能够适应多种应用。
- 因特网能够发展到今日的规模，充分证明了当初采用这种设计思路的正确性。

8.2.1 IP地址

- 网络中的每个独立主机的每个接口必须有一个唯一的Internet 地址，也称为IP地址。
- IP地址长度为32位。表示地址空间是2^{32}，或4294967296（超过40亿个）。
- IP地址的表示方法：三种常用的表示方法
  - 二进制表示方法
  - 点分十进制表示方法
  - 十六进制表示方法。

IP地址的分类

- IP地址按照层次结构划分成五类：A、B、C、D、E类。



二进制表示方法

- 在二进制表示方法中，用一个32位的比特序列表示IP地址，为了使这个地址有更好的可读性，通常在每个字节之间加上一个或多个空格做分隔。例如：

  10000001 00001110 00000110 00011111

点分十进制表示方法

- 为了使32位地址更加简洁和更容易阅读，因特网的地址通常写成用小数点把各字节分隔开的形式。每个字节用一个十进制数表示，这个数小于256。
  - 例如：129.14.6.31

十六进制表示方法

- 有时我们会见到十六进制表示方法的IP地址。每一个十六进制数字等效于4个位。

  例如：0x810E061F

各类IP地址的范围

实际上只有A类的网络号保留了全0和全1

减去主机号全0，主机号全1

类型	范围	网络数	每个网络主机数量
A	0.0.0.0 — 127.255.255.255	2^7-2	2^{24}-2
B	128.0.0.0 — 191.255.255.255	2^{14}-2	2^{16}-2
C	192.0.0.0 — 223.255.255.255	2^{21}-2	2^8-2
D	224.0.0.0 — 239.255.255.255		
E	240.0.0.0 — 247.255.255.255		
		>2百万	

特殊的IP地址

- 网络地址
- 32位全0的地址
- 广播地址
  - 直接广播地址
  - 有限广播地址
- 主机本身地址
- 环回地址
- 私有地址

特殊的IP地址—网络地址

- 网络地址： 主机号全为0的地址
  - 网络IP地址不分配给任何主机，而是作为网络本身的标识，供路由器查找路由表用。
  - 例：主机 202.198.151.136所在网段的网络地址为202.198.151.0

特殊的IP地址—32位全0的地址

- 严格说来，0.0.0.0不是一个真正意义上的IP地址。它表示的是这样一个集合：所有不清楚的主机和目的网络。这里的“不清楚”是指在本机的路由表里没有特定条目指明如何到达。
  - 对本机来说，它就是一个“收容所”，所有不认识的“三无”人员，一律送进去。如果你在网络设置中设置了缺省网关，那么系统会自动产生一个目的地址为0.0.0.0的缺省路由。
  - 还没有分配到IP地址的主机在发送IP报文时用作源IP地址。例如，用于DHCP

特殊的IP地址—广播地址

- 直接广播地址：主机地址为全“1”的IP地址不分配给任何主机，用作广播地址。
  - 例：主机 202.198.151.136所在网段的直接广播地址为202.198.151.255
- 有限广播地址：32位为全1的IP地址称为有限广播地址。
  - 例：有限广播地址为：255.255.255.255
- 两者的区别：
  - 有限广播仅限于本网广播，路由器不转发该类数据包。
  - 直接广播可以跨网广播，可以通过路由器。

特殊的IP地址—环回地址

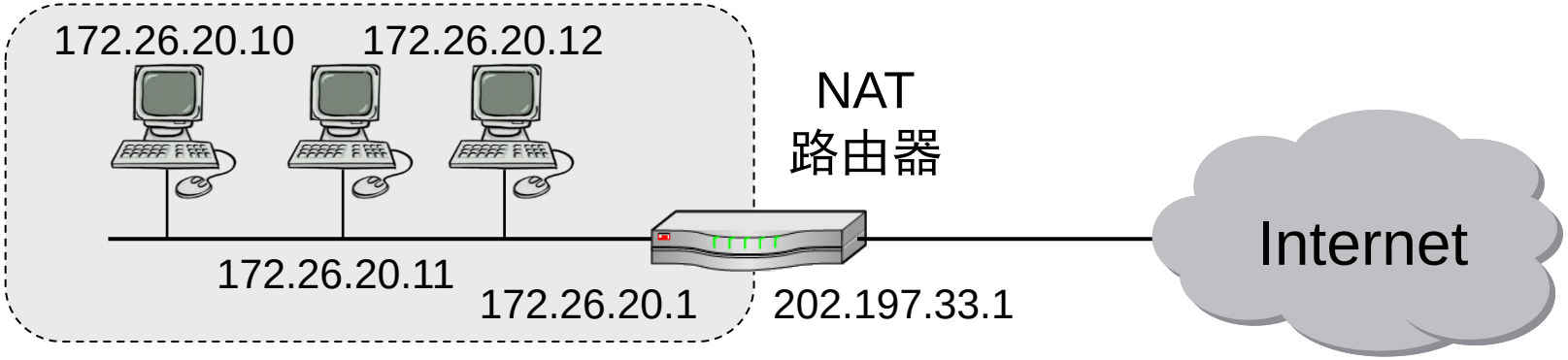
- 环回地址：第一个字节等于127的IP地址称为环回地址，用作主机或路由器的环回接口。
  - 大多数主机系统把127.0.0.1分配给环回接口，常用于本机上软件测试和本机上网络应用程序之间的通信地址。

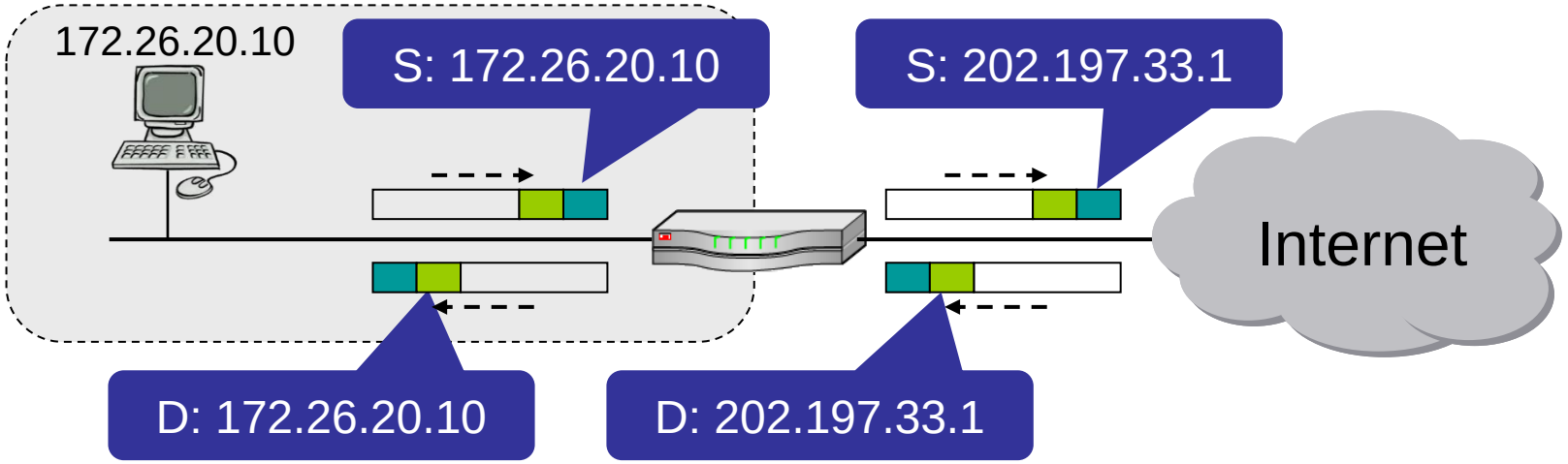
特殊的IP地址—私有地址

- 私有地址：
  - 10.0.0.0 — 10.255.255.255
  - 172.16.0.0 — 172.31.255.255
  - 192.168.0.0 — 192.168.255.255
- 企业内部网主机的IP地址可以设置成专用IP地址，进行企业内部的网络应用；并可通过NAT服务器访问Internet。这样只需要申请少量的全局IP地址，既解决了IP地址不足的问题，又解决了网络安全问题。

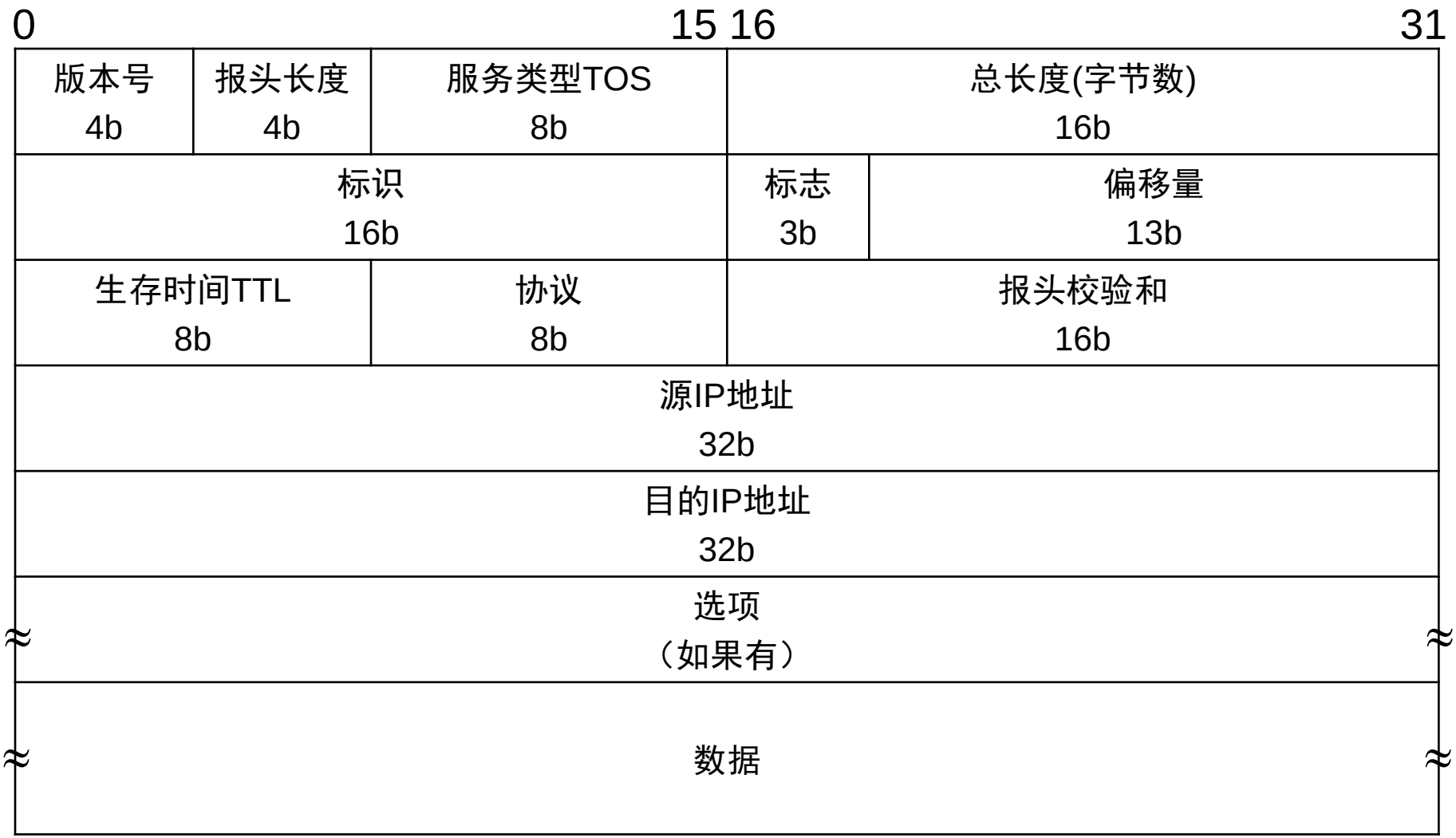
网络地址转换

- 网络地址转换NAT(Network Address Translator)





8.2.2 IP报文格式


0
15 16
31
版本号
4b
报头长度
4b
服务类型TOS
8b
总长度(字节数)
16b
标识
16b
标志
3b
偏移量
13b
生存时间TTL
8b
协议
8b
报头校验和
16b
源IP地址
32b
目的IP地址
32b
选项
（如果有）
数据

IP报文格式(1)

<table>
<tr><td>版本号 4b</td><td>报头长度 4b</td><td>服务类型TOS 8b</td><td colspan="2">总长度(字节数) 16b</td></tr>
<tr><td colspan="3">标识 16b</td><td>标志 3b</td><td>偏移量 13b</td></tr>
<tr><td>生存时间TTL 8b</td><td colspan="2">协议 8b</td><td colspan="2">报头校验和 16b</td></tr>
<tr><td colspan="5">源IP地址 32b</td></tr>
<tr><td colspan="5">目的IP地址 32b</td></tr>
</table>

- 版本号(4)：目前IP协议的版本号为4；它正逐渐地被IPv6版本所替代。
- 报头长度(4)：报头占32位的数量(一般是20字节，即该字段的值为5)。报头最长60B。
  - 报头长度字段以4字节为单位
  - 偏移量以8字节为单位

IP报文格式(2)

- 服务类型(8)：旧标准称为服务类型，但实际上一直未使用。
  - RFC2474重新定义为区分服务。3比特指明优先顺序，3比特指明标志位D/T/R，2比特未用。
    - D：Delay(低延迟)
    - T：Throughput(高吞吐率)
    - R：Reliability(高可靠性)

IP报文格式(3)

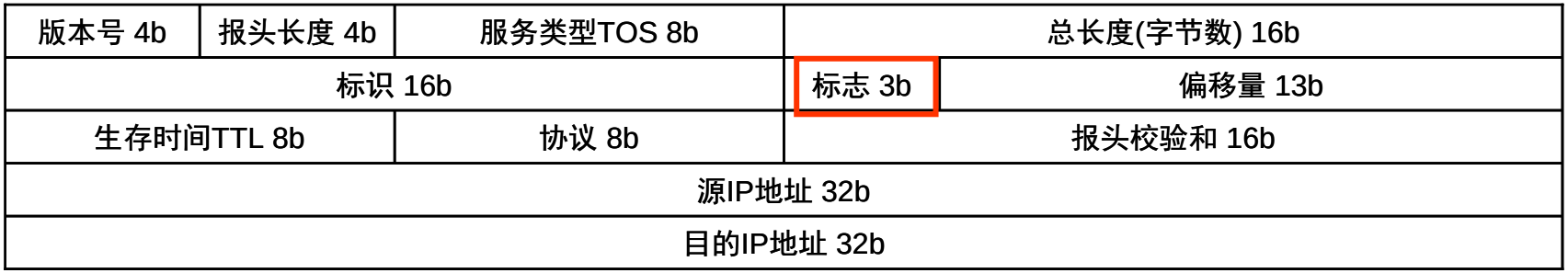
- 总长度(16)：该字段以字节为单位定义IP数据报的总长度(首部加上数据)。 最大长度可达65,536字节。
  - 分段后，该字段不是指未分段前的数据报长度，而是指各分段的长度

IP报文格式(4)

<table>
<tr><td>版本号 4b</td><td>报头长度 4b</td><td>服务类型TOS 8b</td><td colspan="2">总长度(字节数) 16b</td></tr>
<tr><td colspan="3">标识 16b</td><td>标志 3b</td><td>偏移量 13b</td></tr>
<tr><td colspan="2">生存时间TTL 8b</td><td>协议 8b</td><td colspan="2">报头校验和 16b</td></tr>
<tr><td colspan="5">源IP地址 32b</td></tr>
<tr><td colspan="5">目的IP地址 32b</td></tr>
</table>

- 标识(16)：唯一地标识主机发送的每一个数据报。通常每发送一个报文，其值自动加1。
  - 该字段不是序号
  - 如果数据包被分段以适应小型数据包的网络，那么每一个分片中都设置相同的标识号码。

IP报文格式(5)

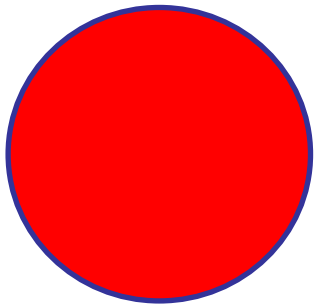


- 标志(3)：目前只有2位有意义
  - 第1位没有使用
  - 第2位为DF(Don't Fragment)位，该位被置1表示不要分段，它命令路由器不要将数据报分段，因为目的端不能重组分段。
  - 第3位是MF(More Fragments) 位，该位被置1表示该分段后还有进一步的分段，最后一个分段MF位为0
  - 是否分段与最大传送单元(MTU)有关

IP报文格式(6)

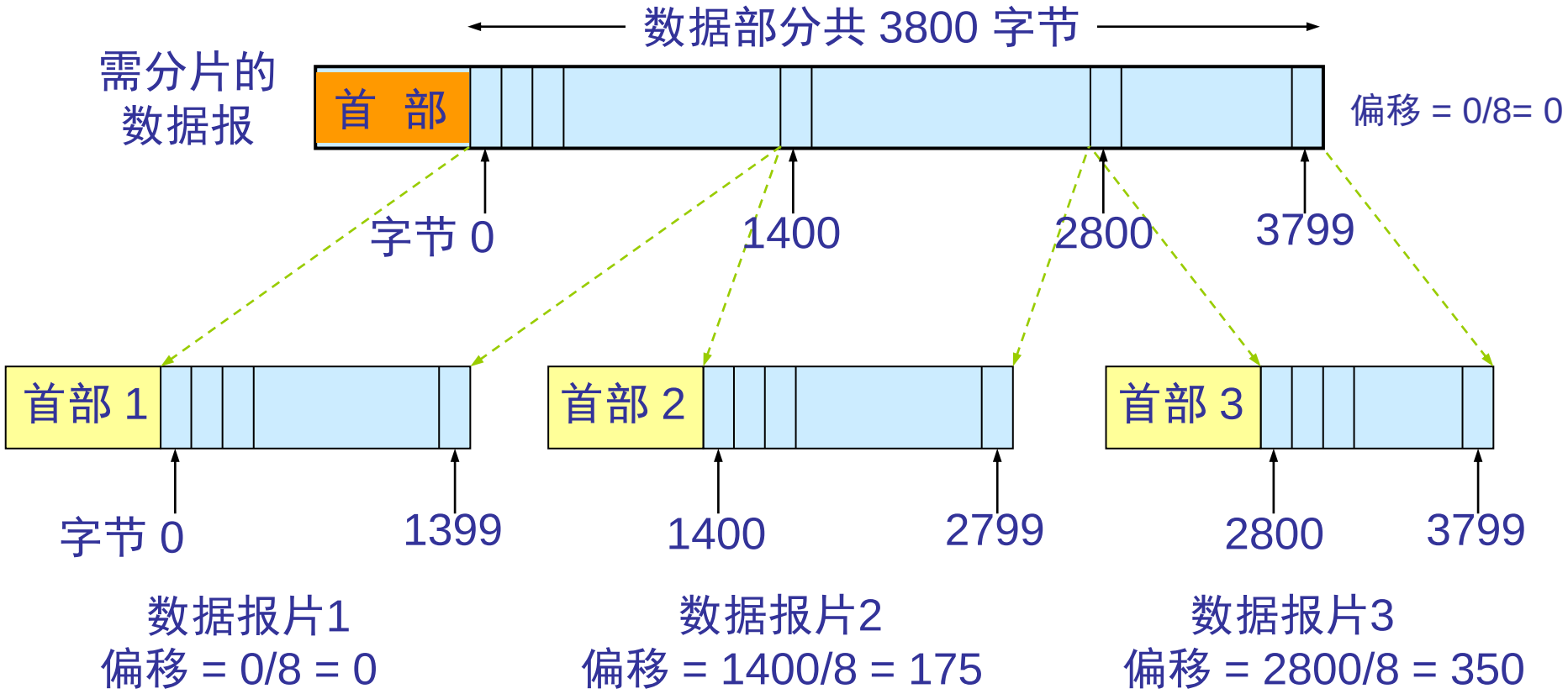
版本号 4b	报头长度 4b	服务类型TOS 8b	总长度(字节数) 16b
标识 16b		标志 3b	偏移量 13b
生存时间TTL 8b		协议 8b	报头校验和 16b
源IP地址 32b			
目的IP地址 32b			

- 偏移量(13)：分段偏移说明该分段在当前数据报的什么位置。
  - 分段偏移以 8字节为单位，这样偏移量1对应字节号8，偏移量 2对应字节号16，依此类推。
  - 数据报进行分段的主机或路由器必须选择每一个分段的长度能够被8除尽。



数据报分段示例

- 一数据报的总长度为3820字节，其数据部分为3800字节长（使用固定长度首部)，需要分片为长度不超过1420字节的数据报片。



分片后有关字段的变化

	总长度	标识	MF	DF	偏移量
原数据报	3280	12345	0	0	0
数据报片1	1420	12345	1	0	0
数据报片2	1420	12345	1	0	175
数据报片3	1020	12345	0	0	350

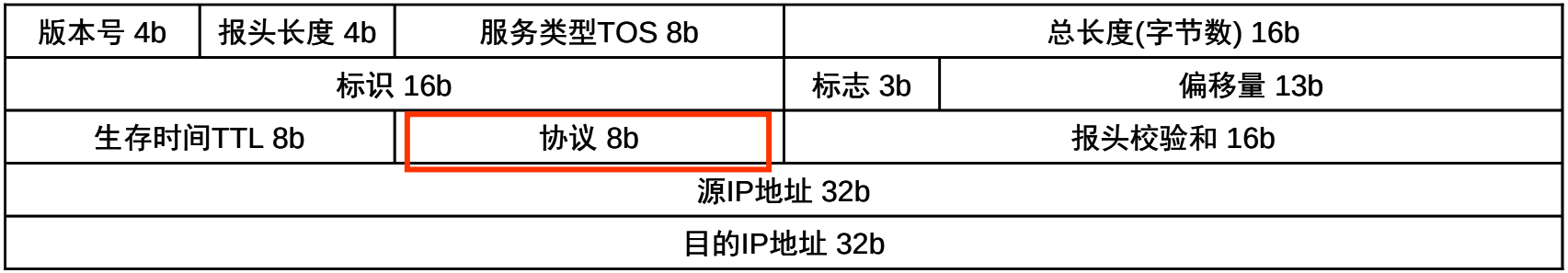
假定数据报片2经过某个网络时还需要再进行分片，分片长度不超过820字节。分片后的数据报片的总长度、标识、MF、DF和片偏移分别为是多少？
在这种情况下如何恢复原数据报？

IP报文格式(7)

<table>
<tr><td>版本号 4b</td><td>报头长度 4b</td><td>服务类型TOS 8b</td><td colspan="2">总长度(字节数) 16b</td></tr>
<tr><td colspan="3">标识 16b</td><td>标志 3b</td><td>偏移量 13b</td></tr>
<tr><td>生存时间TTL 8b</td><td colspan="2">协议 8b</td><td colspan="2">报头校验和 16b</td></tr>
<tr><td colspan="5">源IP地址 32b</td></tr>
<tr><td colspan="5">目的IP地址 32b</td></tr>
</table>

- 生存时间(8)：TTL字段设置了数据报可以经过的最多路由器数量。经过一个路由器，它的值就减去1。当该字段的值为0时，该数据报被丢弃。
  - 通常生存时间的起始值是32、64、128

IP报文格式(8)

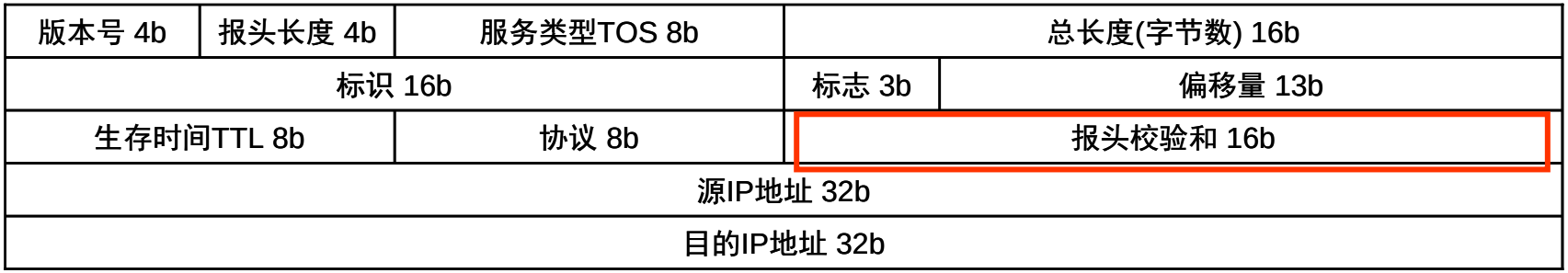


- 协议字段(8)：IP数据报的上层携带的协议。
  - 一个IP数据报能封装来自诸如TCP(17)、ICMP(1)等较高层协议的数据。

常用协议指端值

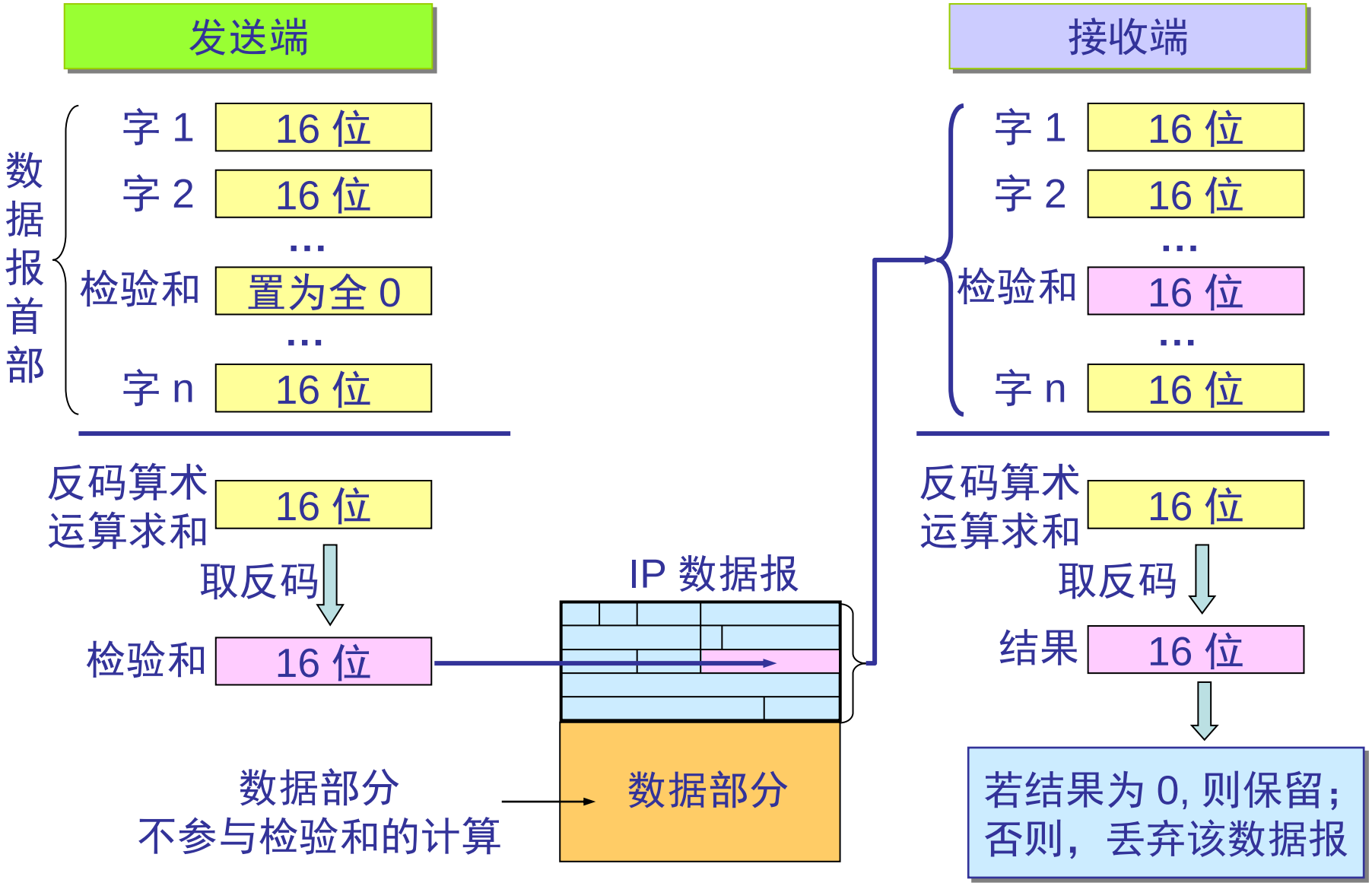
协议名	ICMP	IGMP	TCP	EGP	IGP	UDP	IPv6	OSPF
协议字段值	1	2	6	8	9	17	41	89

IP报文格式(9)



- 报头检查和(16)：根据IP报头计算的检查和，不对报头后面的数据进行计算。
  - 检验和采用16位反码求和的算法。

16位反码求和的算法


发送端
接收端
数据报首部
字 1
16 位
字 2
16 位
...
检验和
置为全 0
...
字 n
16 位
检验和
16 位
反码算术运算求和
16 位
取反码
检验和
16 位
IP 数据报
数据部分
数据部分
不参与检验和的计算
结果
16 位
若结果为 0, 则保留；
否则， 丢弃该数据报

IP报文格式(10)

- 源IP地址(32)和目的IP地址(32)：每个数据报都包含32位的源IP地址和目的IP地址。

IP报文格式(11)

- 选项字段：IP 首部的选项字段用来支持排错、测量以及安全等措施
  - 选项字段的长度可变，从 1 个字节到 40 个字节不等，取决于所选择的项目。
  - 增加首部的可变部分是为了增加 IP 数据报的功能，但同时也使得 IP 数据报的首部长度成为可变的。这就增加了每一个路由器处理数据报的开销。
  - 实际上这些选项很少被使用。

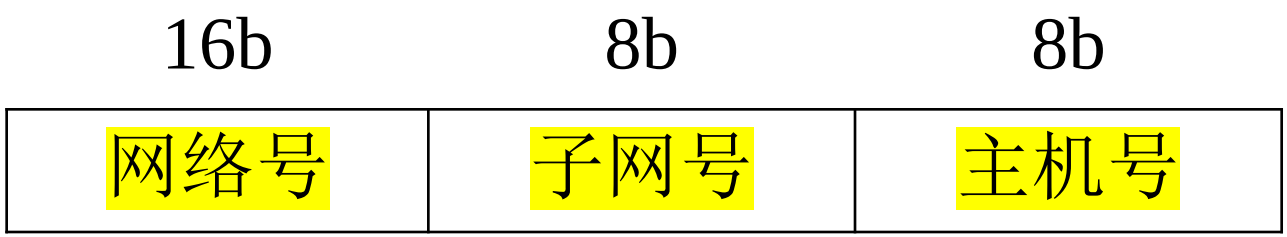
8.2.3 子网编址和子网掩码

类型	范围	网络数	每个网络 主机数量
A	0.0.0.0 — 127.255.255.255	2^7-2	>1.6千万
B	128.0.0.0 — 191.255.255.255	2^{14}-2	2^{16}-2
C	192.0.0.0 — 223.255.255.255	2^{21}-2	2^8-2
D	224.0.0.0 — 239.255.255.255		
E	240.0.0.0 — 247.255.255.255		
		>2百万	

- 早期IP 地址的设计确实不够合理：
  - IP 地址空间的利用率有时很低。
  - 给每一个物理网络分配一个网络号会使路由表变得太大因而使网络性能变坏。
  - 两级的 IP 地址不够灵活。

子网编址

- 为解决IP地址原编址方案的不足，从1985年起，在IP地址中又增加了一个“子网号字段”，使2级地址变成了3级地址。这种方法称为划分子网。
- 子网编址不是把IP地址看成由单纯的一个网络号和一个主机号组成，而是把主机号进一步划分为一个子网号和一个主机号。
- 目前所有的主机都要求支持子网编址。



B类地址的一种子网编码

子网掩码

- 从一个 IP 数据报的首部并无法判断源主机或目的主机所连接的网络是否进行了子网划分。
- 使用子网掩码(subnet mask)可以找出 IP 地址中的子网部分。
- 子网掩码是一个32比特的数值，其中值为1的比特用于网络号和子网号，为0的比特留给主机号。
  - IP地址中与子网掩码的1相对应的位构成了网络号和子网号
  - IP地址中与子网掩码的0相对应的位构成了主机号

子网掩码是一个重要属性

- 子网掩码是一个网络或一个子网的重要属性。
- 路由器在和相邻路由器交换路由信息时，必须把自己所在网络（或子网）的子网掩码告诉相邻路由器。
- 路由器的路由表中的每一个项目，除了要给出目的网络地址外，还必须同时给出该网络的子网掩码。
- 若一个路由器连接在两个子网上就拥有两个网络地址和两个子网掩码。

子网掩码作用

- 通过IP地址和子网掩码，主机就可以判断数据报的目的地址为：
  - 本子网中的主机；
  - 本网络中其他子网中的主机；
  - 其他网络上的主机。
- 例如：一个主机的IP地址为140.252.3.4，而子网掩码为255.255.255.0
  - 如果数据报的目的IP地址为140.252.7.8，我们就知道网络号是相同的，而子网号是不同的
  - 如果数据报的目的IP地址为140.252.3.9，我们就知道网络号是相同的，而且子网号也是相同的，只是主机号不同

IP地址和子网掩码

- 知道IP地址和子网掩码后可以算出：
  - 网络地址
  - 广播地址
  - 地址范围
  - 本网有几台主机

例1

- IP地址为192.168.100.5，子网掩码为255.255.255.0。算出网络地址、广播地址、地址范围、主机数?
- 192.168.100.5 ⇒11000000 10101000 01100100 00000101

  255.255.255.0 ⇒11111111 11111111 11111111 00000000
  - 子网地址：192.168.100.0
  - 广播地址： 192.168.100.255
  - 地址范围：网络地址+1至广播地址-1

    192.168.100.1 至 192.168.100.254
  - 主机的数量：2^8-2=254

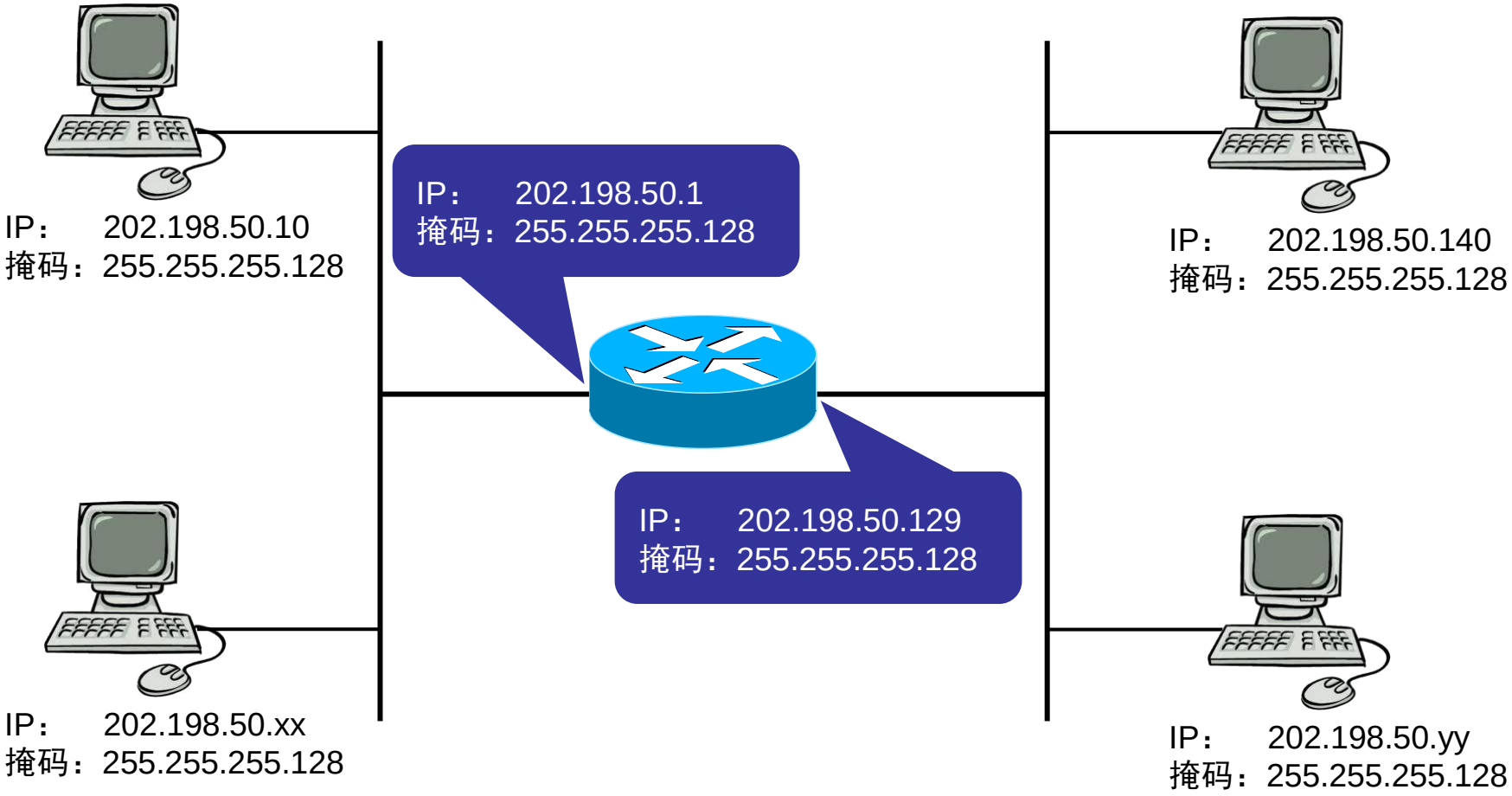
掩码中有8个0所以是2的8次方

例2

- IP地址为192.168.150.122，子网掩码为255.255.255.248。算出网络地址、广播地址、地址范围、主机数?
- 192.168.150.01111010

  255.255.255.11111000
  - 子网地址：192.168.150.120
  - 广播地址：192.168.150.127
  - 地址范围：192.168.150.121-192.168.150.126
  - 主机的数量：2^3-2=6

使用255.255.255.128子网掩码的网络



无分类编址CIDR

- 划分子网在一定程度上缓解了因特网在发展中遇到的困难。然而在 1992 年因特网仍然面临三个必须尽早解决的问题：
  - B 类地址在 1992 年已分配了近一半，眼看就要在 1994 年 3 月全部分配完毕！
  - 因特网主干网上的路由表中的项目数急剧增长（从几千个增长到几万个）。
  - 整个 IPv4 的地址空间最终将全部耗尽。

IP编址问题的演进

- 1987 年，RFC 1009 指明了在一个划分子网的网络中可同时使用几个不同的子网掩码。使用变长子网掩码 VLSM (Variable Length Subnet Mask)可进一步提高 IP 地址资源的利用率。
- 在 VLSM 的基础上又进一步研究出无分类编址方法，它的正式名字是无分类域间路由选择 CIDR (Classless Inter-Domain Routing)。

无分类编址CIDR主要特点

- CIDR 消除了传统的 A 类、B 类和 C 类地址以及划分子网的概念，因而可以更加有效地分配 IPv4 的地址空间。
- CIDR使用各种长度的“网络前缀”(network-prefix)来代替分类地址中的网络号和子网号。
- IP地址从三级编址（使用子网掩码）又回到了两级编址。

无分类的两级编址

- 无分类的两级编址的记法：

  IP地址 ≡ {<网络前缀>, <主机号>}

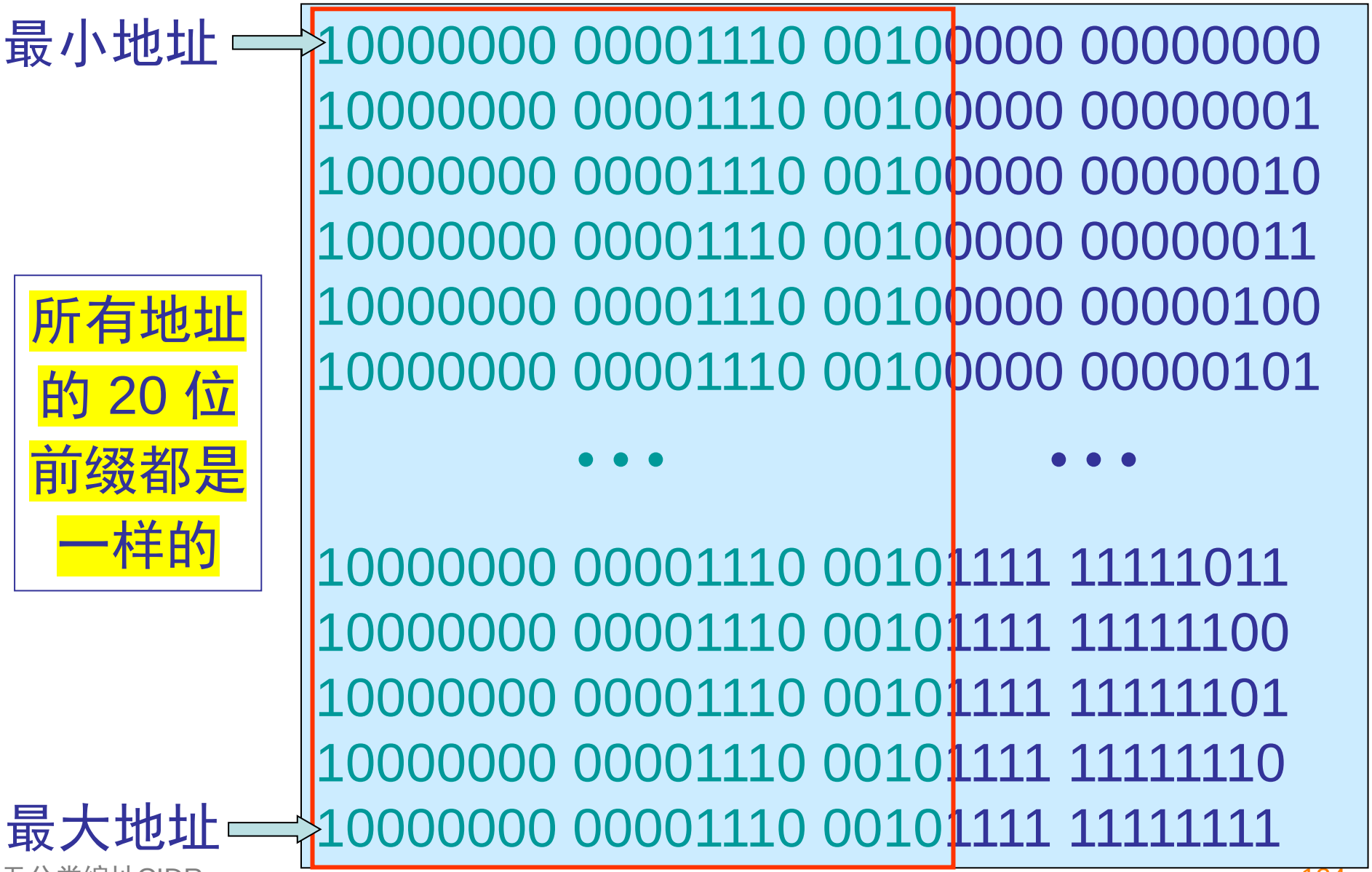
- CIDR 还使用“斜线记法”(slash notation)，它又称为CIDR记法，即在 IP 地址后面加上一个斜线“/”，然后写上网络前缀所占的位数（这个数值对应于三级编址中子网掩码中 1 的个数）。
- CIDR 把网络前缀都相同的连续的 IP 地址组成“CIDR 地址块”。

CIDR地址块

- 128.14.32.0/20 表示的地址块共有 2^{12} 个地址（因为斜线后面的 20 是网络前缀的位数，所以这个地址的主机号是 12 位）。
  - 这个地址块的起始地址是 128.14.32.0
  - 128.14.32.0/20 地址块的最小地址：128.14.32.0
  - 128.14.32.0/20 地址块的最大地址：128.14.47.255
  - 全 0 和全 1 的主机号地址一般不使用。

128.14.32.0化成二进制是10000000 00001110 00100000 00000000
有20为网络前缀，说明地址主机号是12位
地址块的最大地址即为主机号全为1，00101111正好为47，11111111为255
所以128.14.32.0/20地址块的最大地址为128.14.47.255

128.14.32.0/20 表示的地址(2^{12} 个地址)



路由聚合(route aggregation)

- 一个 CIDR 地址块可以表示很多地址，这种地址的聚合常称为路由聚合，它使得路由表中的一个项目可以表示很多个（例如上千个）原来传统分类地址的路由。
- 路由聚合也称为构成超网(supernetting)。
- CIDR 虽然不使用子网了，但仍然使用“掩码”这一名词（但不叫子网掩码）。
- 对于 /20 地址块，它的掩码是 20 个连续的 1。斜线记法中的数字就是掩码中1的个数。

CIDR记法的其他形式

- 10.0.0.0/10 可简写为 10/10，也就是把点分十进制中低位连续的 0 省略。
  - 10.0.0.0/10 隐含地指出 IP 地址 10.0.0.0 的掩码是 255.192.0.0。此掩码可表示为

11111111 11000000 00000000 00000000

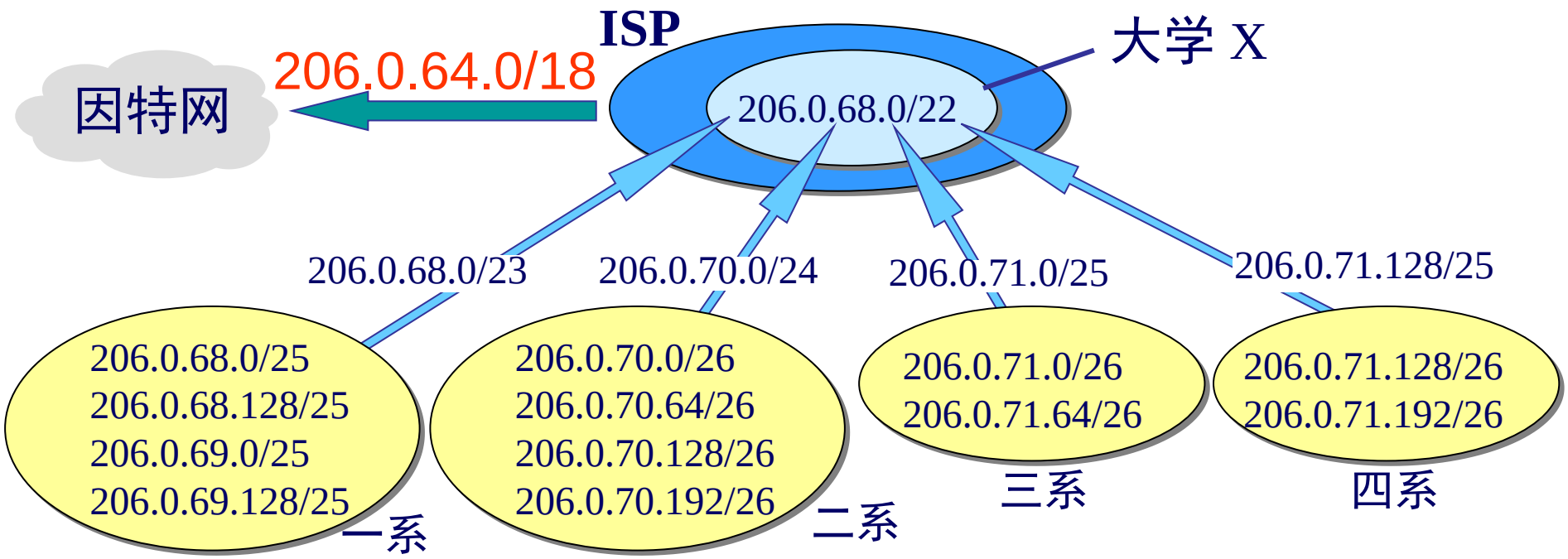
255 192 0 0

- 网络前缀的后面加一个星号 * 的表示方法：如 00001010 00*，在星号 * 之前是网络前缀，而星号 * 表示 IP 地址中的主机号，可以是任意值。

构成超网

- 前缀长度不超过 23 位的 CIDR 地址块都包含了多个 C 类地址。
- 这些 C 类地址合起来就构成了超网。
- CIDR 地址块中的地址数一定是 2 的整数次幂。
- 网络前缀越短，其地址块所包含的地址数就越多。而在三级结构的IP地址中，划分子网是使网络前缀变长。

CIDR 地址块划分举例



单位	地址块	二进制表示	地址数
ISP	206.0.64.0/18	11001110.00000000.01*	16384
大学	206.0.68.0/22	11001110.00000000.010001*	1024
一系	206.0.68.0/23	11001110.00000000.0100010*	512
二系	206.0.70.0/24	11001110.00000000.01000110.*	256
三系	206.0.71.0/25	11001110.00000000.01000111.0*	128
四系	206.0.71.128/25	11001110.00000000.01000111.1*	128

最长前缀匹配

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果。
- 应当从匹配结果中选择具有最长网络前缀的路由：最长前缀匹配(longest-prefix matching)。
- 网络前缀越长，其地址块就越小，因而路由就越具体(more specific) 。
- 最长前缀匹配又称为最长匹配或最佳匹配。

最长前缀匹配举例

收到的分组的目的地址 *D* = 206.0.71.130

路由表中的项目：206.0.68.0/22　　（大学）

206.0.71.128/25　（四系）

查找路由表中的第 1 个项目

第 1 个项目 206.0.68.0/22 的掩码 *M* 有 22 个连续的 1

M = 11111111 11111111 11111100 00000000

因此只需把 *D* 的第 3 个字节转换成二进制。

	M =	11111111	11111111	11111100 00000000
AND	*D* =	206.	0.	01000111. 0
		206.	0.	01000100. 0

与 206.0.68.0/22 匹配

最长前缀匹配举例

收到的分组的目的地址 *D* = 206.0.71.130

路由表中的项目：206.0.68.0/22　　（大学）

206.0.71.128/25　（四系）

再查找路由表中的第 2 个项目

第 2 个项目 206.0.71.128/25 的掩码 *M* 有25个连续的1

M = 11111111 11111111 11111111 10000000

因此只需把 *D* 的第 4 个字节转换成二进制。

	M =	11111111	11111111	11111111 10000000
AND	*D* =	206.	0.	71. 10000010
		206.	0.	71. 10000000

与 206.0.71.128/25 匹配

最长前缀匹配

D AND (11111111 11111111 11111100 00000000)

= 206.0.68.0/22　　匹配

D AND (11111111 11111111 11111111 10000000)

= 206.0.71.128/25　　匹配

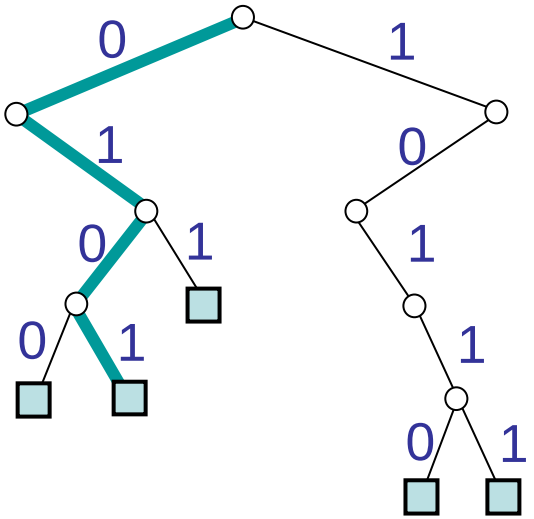
- 选择两个匹配的地址中更具体的一个，即选择最长前缀的地址。

使用二叉线索查找路由表

- 当路由表的项目数很大时，怎样设法减小路由表的查找时间就成为一个非常重要的问题。
- 为了进行更加有效的查找，通常是将无分类编址的路由表存放在一种层次的数据结构中，然后自上而下地按层次进行查找。这里最常用的就是二叉线索(binary trie)。
- IP 地址中从左到右的比特值决定了从根结点逐层向下层延伸的路径，而二叉线索中的各个路径就代表路由表中存放的各个地址。
- 为了提高二叉线索的查找速度，广泛使用了各种压缩技术。

用5个前缀构成的二叉线索

32 位的 IP 地址	唯一前缀
01000110 00000000 00000000 00000000	0100
01010110 00000000 00000000 00000000	0101
01100001 00000000 00000000 00000000	011
10110000 00000010 00000000 00000000	10110
10111011 00001010 00000000 00000000	10111



8.2.4 IP路由选择

- 对于网络中的主机来说，IP路由选择是很简单的。
  - 如果目的主机和源主机在一个共享网络上（以太网），那么IP数据报就直接送到目的主机上。
  - 否则，主机把数据报发往一个默认的路由器(网关)上，由该路由器负责转发该数据报。

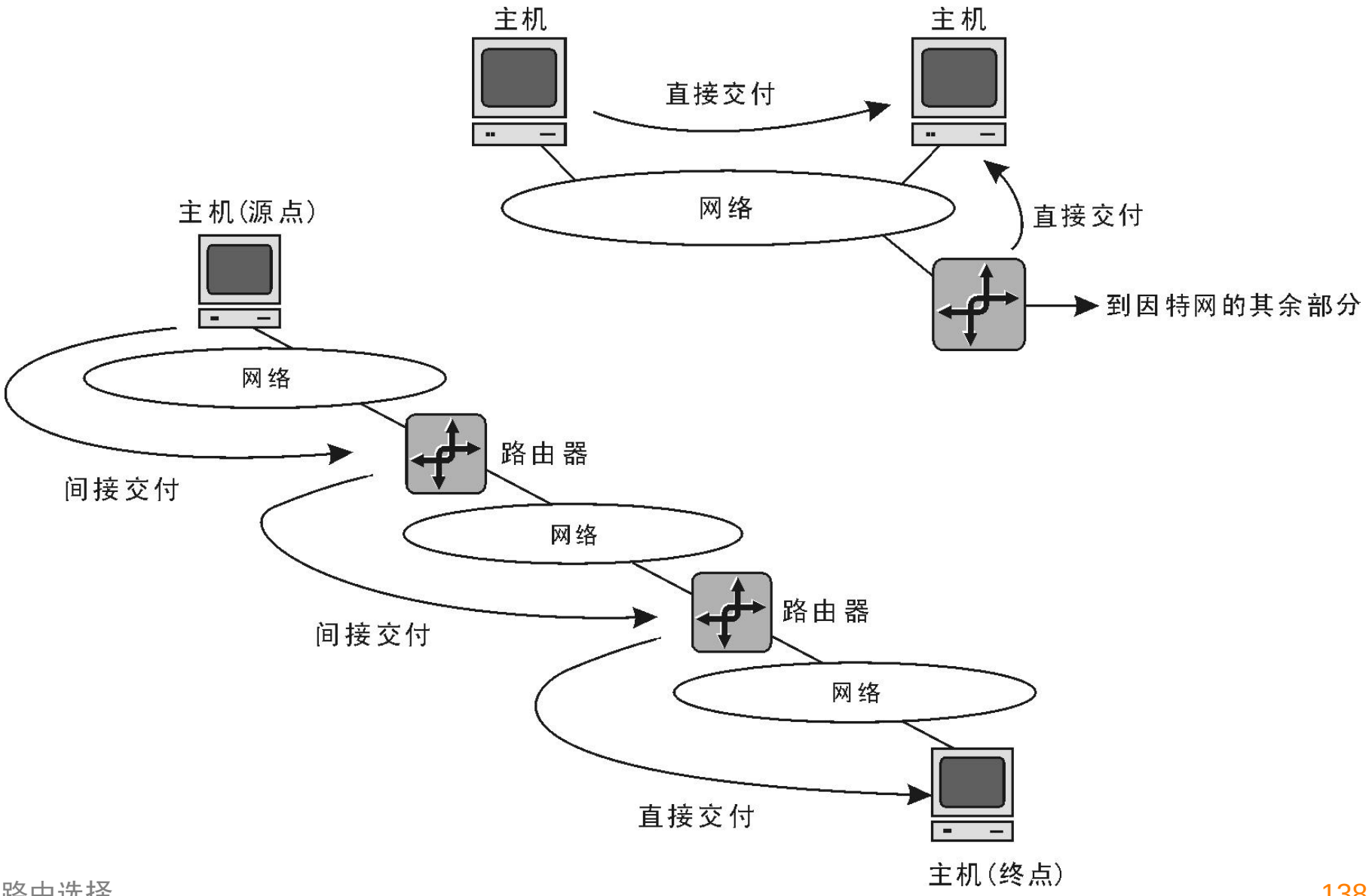
默认路由

- 路由器还可采用默认路由以减少路由表所占用的空间和搜索路由表所用的时间。
- 这种转发方式在一个网络只有很少的对外连接时是很有用的。
- 默认路由在主机发送 IP 数据报时往往更能显示出它的好处。
- 如果一个主机连接在一个小网络上，而这个网络只用一个路由器和因特网连接，那么在这种情况下使用默认路由是非常合适的。

IP交付

- 可以用两种不同的方法把一个分组交付到它最后的终点：
  - 直接交付
  - 间接交付(转发)

直接交付与间接交付


主机
主机
直接交付
网络
直接交付
主机(源点)
到因特网的其余部分
网络
路由器
间接交付
网络
路由器
间接交付
网络
直接交付
主机(终点)

IP转发

- 转发表示把分组放到去终点的路由上。进行转发就要求主机或路由器装有路由表。
- 路由器的主要功能是转发数据报，内存中维持一个路由表。当收到一个数据报并进行转发时，它都要对该路由表搜索一次，从一个接口转发到另一个接口。
- 路由表的每一行应包含下面信息：
  - 目的IP地址、下一跳路由器(next-hop router)的IP地址、为数据报的传输指定一个网络接口。

基于下一跳的路由表

主机A的路由表

终点	路由
主机B	R1，R2，主机B

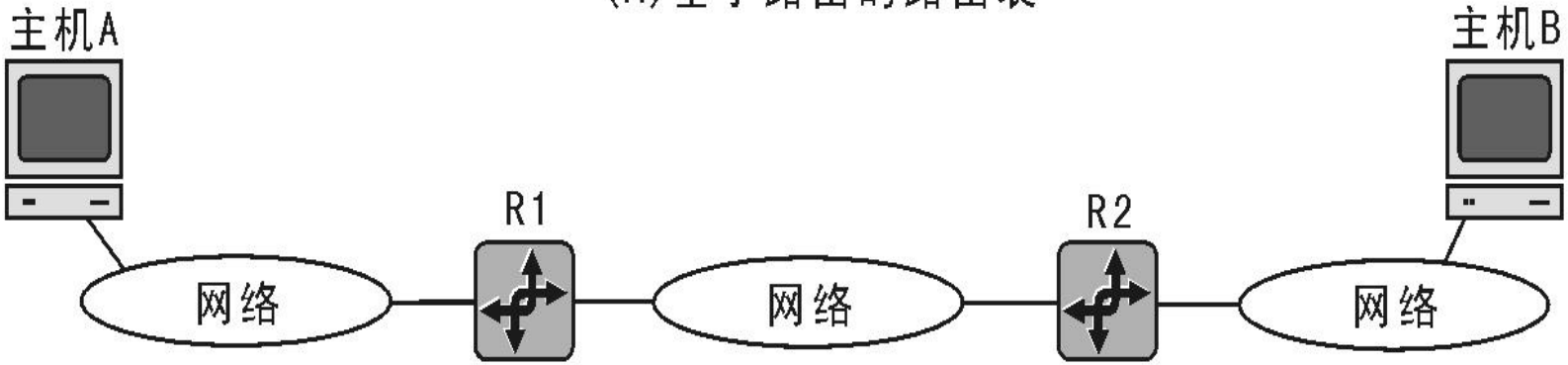
路由器R1的路由表

终点	路由
主机B	R2，主机B

路由器R2的路由表

终点	路由
主机B	主机B

(A)基于路由的路由表



主机A的路由表

终点	下一跳
主机B	R1

路由器R1的路由表

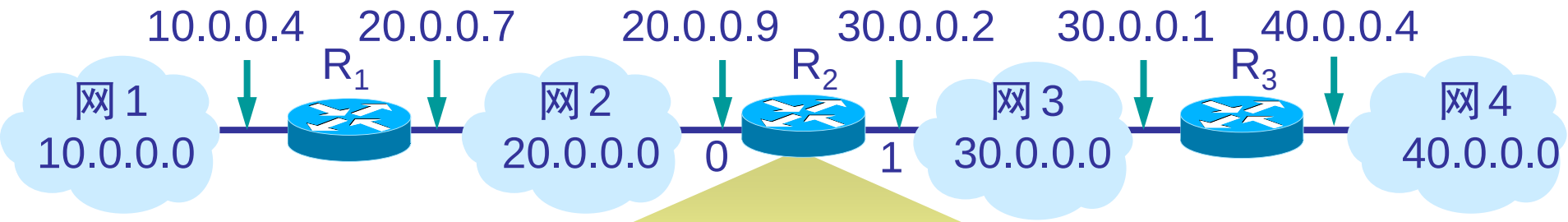
终点	下一跳
主机B	R2

路由器R2的路由表

终点	下一跳
主机B	—

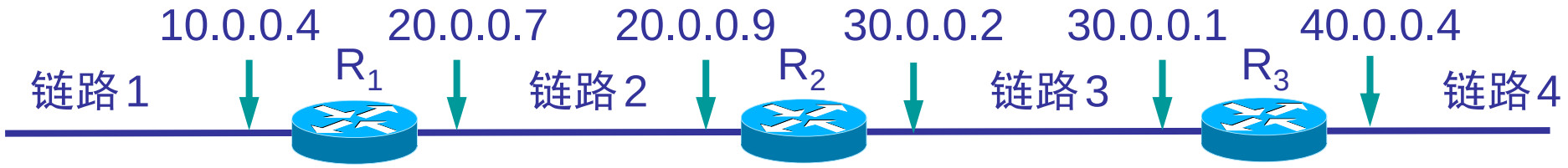
(B)基于下一跳的路由表

路由表示例



路由器 R_2 的路由表

目的主机所在的网络	下一跳地址
20.0.0.0	直接交付，接口 0
30.0.0.0	直接交付，接口 1
10.0.0.0	间接交付，20.0.0.7
40.0.0.0	间接交付，30.0.0.1



路由器转发分组的算法

1. 从数据报的首部提取目的主机的 IP 地址 D，得出目的网络地址为 N。
2. 若网络 N 与此路由器直接相连，则把数据报直接交付目的主机 D；否则是间接交付，执行(3)。
3. 若路由表中有目的地址为 D 的特定主机路由，则把数据报传送给路由表中所指明的下一跳路由器；否则，执行(4)。
4. 若路由表中有到达网络 N 的路由，则把数据报传送给路由表指明的下一跳路由器；否则，执行(5)。
5. 若路由表中有一个默认路由，则把数据报传送给路由表中所指明的默认路由器；否则，执行(6)。
6. 报告转发分组出错。

路由器转发分组的算法—划分子网

1. 从收到的分组的首部提取目的 IP 地址 D。
2. 先用各网络的子网掩码和 D 逐位相“与”，看是否和相应的网络地址匹配。若匹配，则将分组直接交付。否则就是间接交付，执行(3)。
3. 若路由表中有目的地址为 D 的特定主机路由，则将分组传送给指明的下一跳路由器；否则，执行(4)。
4. 对路由表中的每一行的子网掩码和 D 逐位相“与”，若其结果与该行的目的网络地址匹配，则将分组传送给该行指明的下一跳路由器；否则，执行(5)。
5. 若路由表中有一个默认路由，则将分组传送给路由表中所指明的默认路由器；否则，执行(6)。
6. 报告转发分组出错。

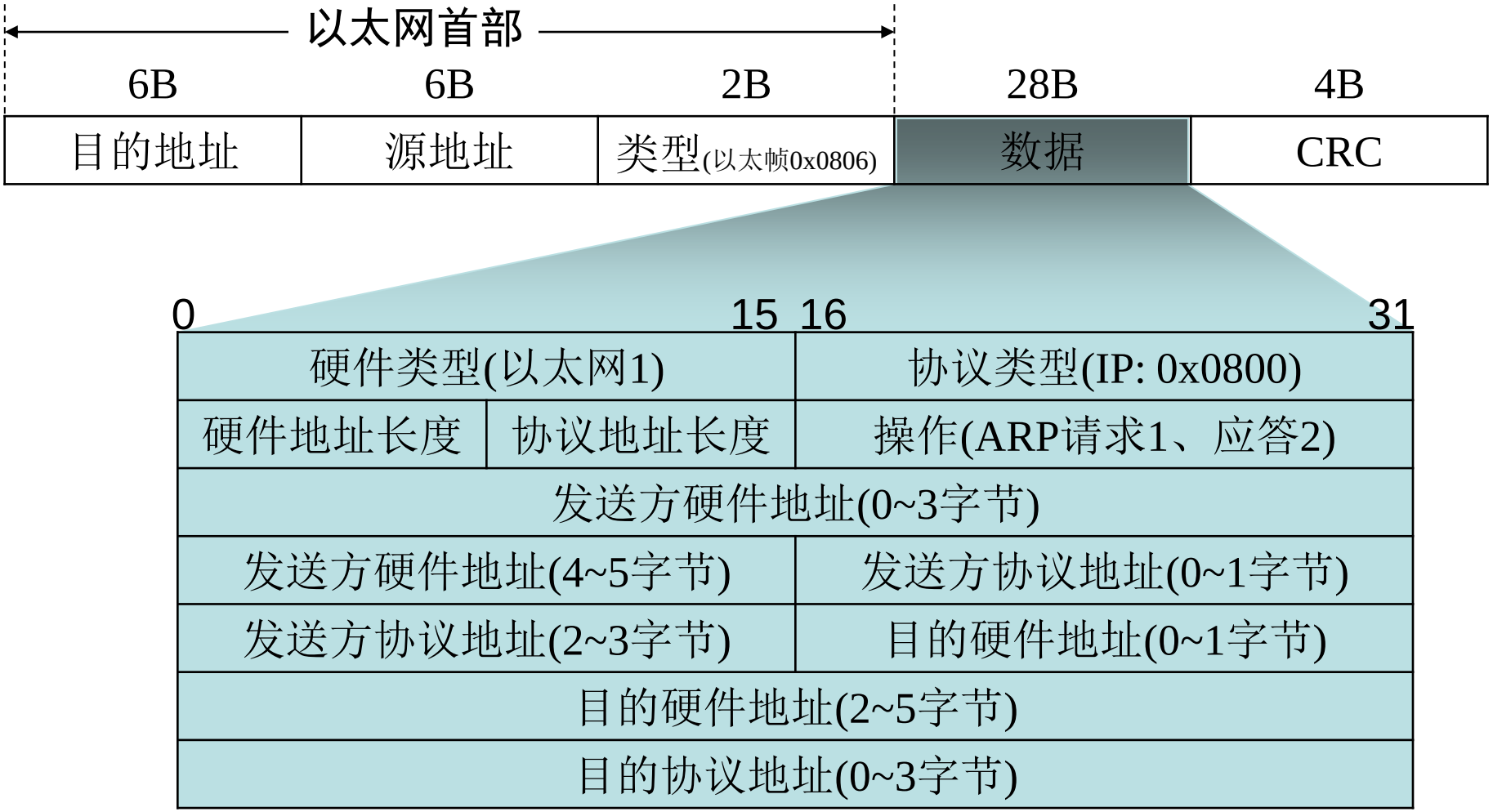
5.6 ARP协议

- ARP地址解析协议，就是将主机IP地址映射为硬件地址。
- 在局域网中，网络中实际传输的单元是“数据帧”，数据帧的首部有目的主机的MAC地址。
- 在以太网中，一个主机要和另一个主机进行直接通信，必须通过地址解析协议获得目的主机的MAC地址。
- ARP协议的基本功能就是通过目的设备的IP地址，查询目标设备的MAC地址，以保证通信的顺利进行。

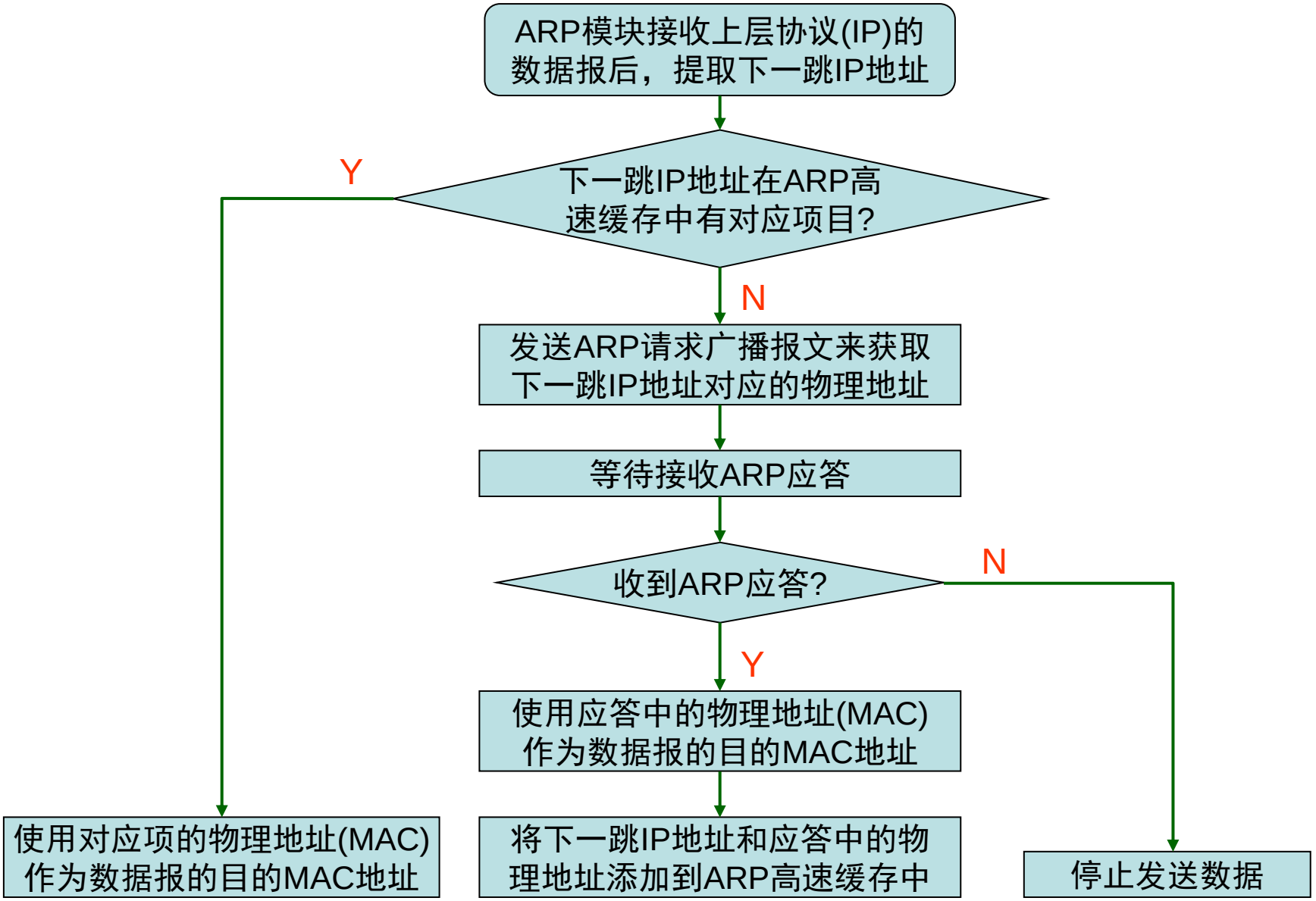
ARP协议的工作原理

- ARP协议的请求包是以广播方式发送的。
- 网段中的所有主机都会接收到这个包，如果一个主机的IP地址和ARP请求中的目的IP地址相同，该主机会对这个请求数据包作出ARP应答，将其MAC地址发送给源端。

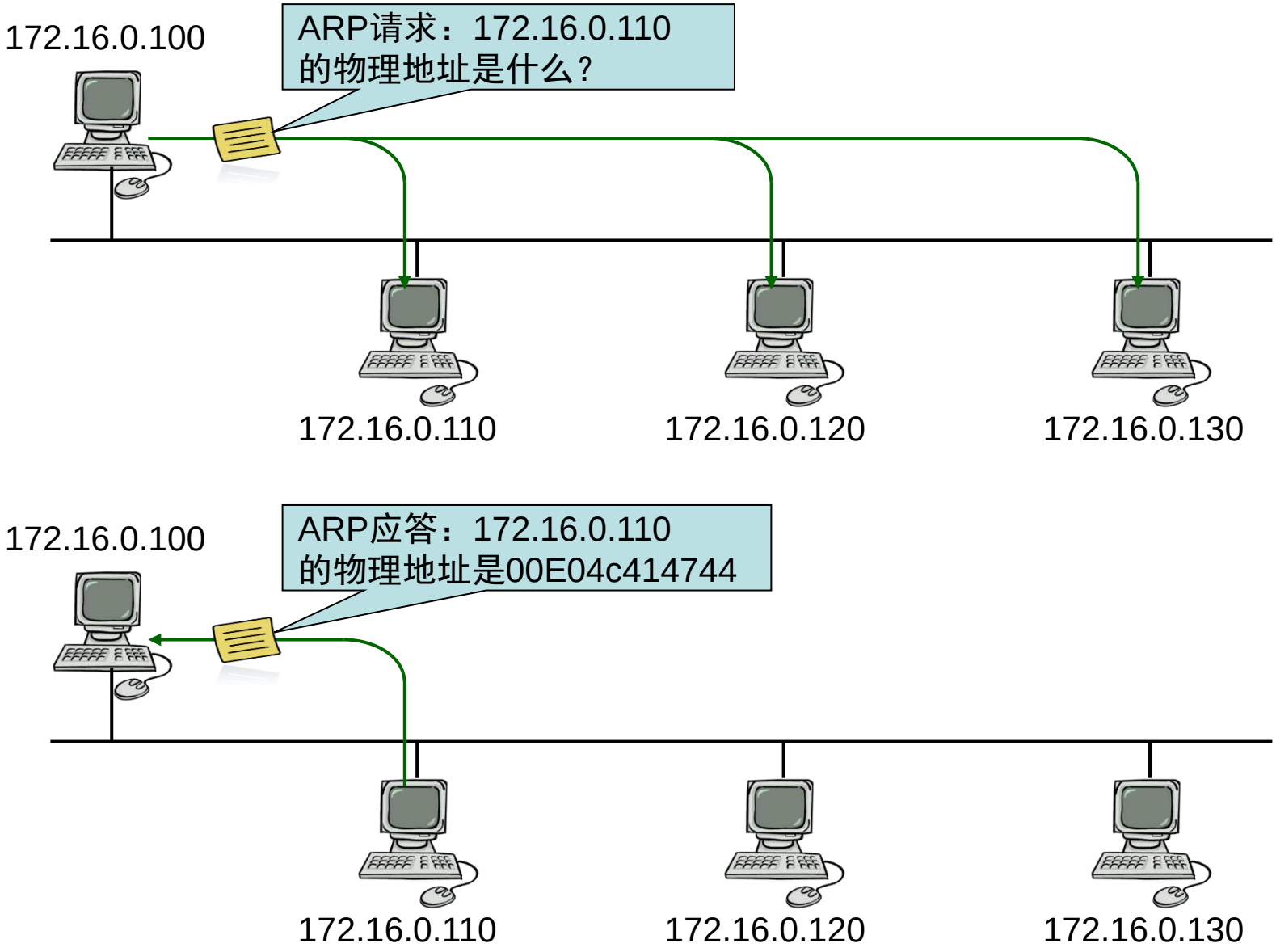
ARP封装与报文格式


以太网首部
6B
6B
2B
28B
4B
目的地址
源地址
类型(以太帧0x0806)
数据
CRC
0
15
16
31
硬件类型(以太网1)
协议类型(IP: 0x0800)
硬件地址长度
协议地址长度
操作(ARP请求1、应答2)
发送方硬件地址(0~3字节)
发送方硬件地址(4~5字节)
发送方协议地址(0~1字节)
发送方协议地址(2~3字节)
目的硬件地址(0~1字节)
目的硬件地址(2~5字节)
目的协议地址(0~3字节)

ARP工作流程


ARP模块接收上层协议(IP)的数据报后，提取下一跳IP地址
下一跳IP地址在ARP高速缓存中有对应项目?
Y
N
发送ARP请求广播报文来获取下一跳IP地址对应的物理地址
等待接收ARP应答
收到ARP应答?
N
Y
使用应答中的物理地址(MAC)作为数据报的目的MAC地址
使用对应项的物理地址(MAC)作为数据报的目的MAC地址
将下一跳IP地址和应答中的物理地址添加到ARP高速缓存中
停止发送数据

ARP工作示例


172.16.0.100
ARP请求：172.16.0.110
的物理地址是什么？
172.16.0.110
172.16.0.120
172.16.0.130
172.16.0.100
ARP应答：172.16.0.110
的物理地址是00E04c414744
172.16.0.110
172.16.0.120
172.16.0.130

ARP高速缓存

- ARP高速运行的关键是由于每个主机上都有一个ARP高速缓存。这个高速缓存存放了最近Internet地址到硬件地址之间的映射。
- 通过使用ARP高速缓存，可以在高速缓存中发现IP地址和硬件地址之间的映射，从而可以避免远程访问的开销，提高效率。

ARP命令使用示例

- 主机的ARP缓存表是可以查询的，也可以添加和修改。
- Windows系统可在命令提示符下使用。
  - arp -a 可以查看ARP缓存表中的内容

```
F:\>arp -a
Interface:192.168.1.1 on Interface 0x2
Internet Address    Physical Address      Type
192.168.1.1         00-e0-37-62-85-87     dynamic
```

  - arp -d 命令可以删除ARP表中某一行的内容
  - arp -s 可以手动在ARP表中指定IP地址与MAC地址的对应项。

使用ARP的四种典型情况

- 发送方是主机，要把IP数据报发送到本网络上的另一个主机。这时用 ARP 找到目的主机的硬件地址。
- 发送方是主机，要把 IP 数据报发送到另一个网络上的一个主机。这时用 ARP 找到本网络上的一个路由器的硬件地址。剩下的工作由这个路由器来完成。
- 发送方是路由器，要把 IP 数据报转发到本网络上的一个主机。这时用 ARP 找到目的主机的硬件地址。
- 发送方是路由器，要把 IP 数据报转发到另一个网络上的一个主机。这时用 ARP 找到本网络上的一个路由器的硬件地址。剩下的工作由这个路由器来完成。

动态主机自动配置

- 为了将软件协议做成通用的和便于移植，协议软件的编写者把协议软件参数化。这就使得在很多台计算机上使用同一个经过编译的二进制代码成为可能。
- 一台计算机和另一台计算机的区别，都可通过一些不同的参数来体现。但在软件协议运行之前，必须给每一个参数赋值。
- 在协议软件中给这些参数赋值的动作叫做协议配置。

协议配置

- 一个软件协议在使用之前必须是已正确配置的。具体的配置信息有哪些则取决于协议栈。
- 对于使用TCP/IP的主机需要配置的项目
  - IP 地址
  - 子网掩码
  - 默认路由器的 IP 地址
  - 域名服务器的 IP 地址
- RARP ⇒ BOOTP ⇒ DHCP

RARP & BOOTP

- RARP
  - 在LAN上有一个RARP服务器，存有LAN上节点的MAC-IP对应关系
  - IP节点发送一个第2层的广播请求
  - 问题：无法跨越不同网络请求地址
- BOOTP：
  - 使用IP协议与服务器通信
  - 服务器存有每台主机的IP地址
  - 客户端发送一个请求数据报，IP地址全“1”作为目的地址，全“0”地址作为源地址

DHCP协议

- DHCP--Dynamic Host Configuration Protocol
- 动态主机配置协议，它提供了一种动态指定IP地址和配置参数的机制，用于简化主机 IP 配置管理。
- 通过采用 DHCP，可以使用 DHCP 服务器为网络上启用了 DHCP 的客户端管理动态 IP 地址分配和其它相关配置细节：
  - IP地址
  - 子网掩码
  - 默认网关IP地址
  - DNS(Domain Name System)域名系统服务器IP地址

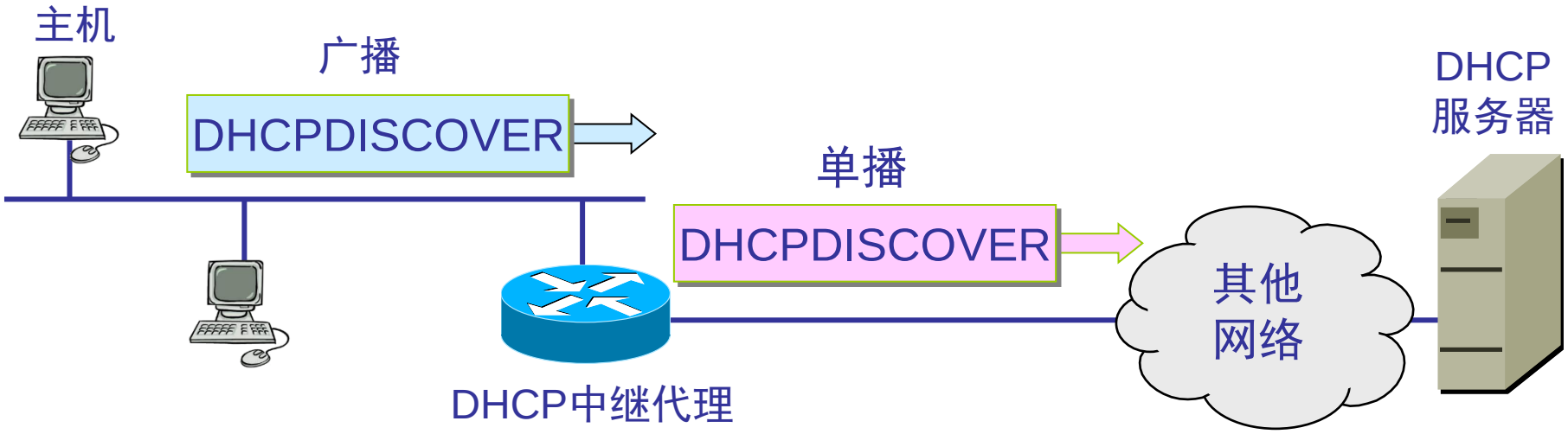
DNS协议是能完成域名到IP地址解析的协议

DHCP工作原理

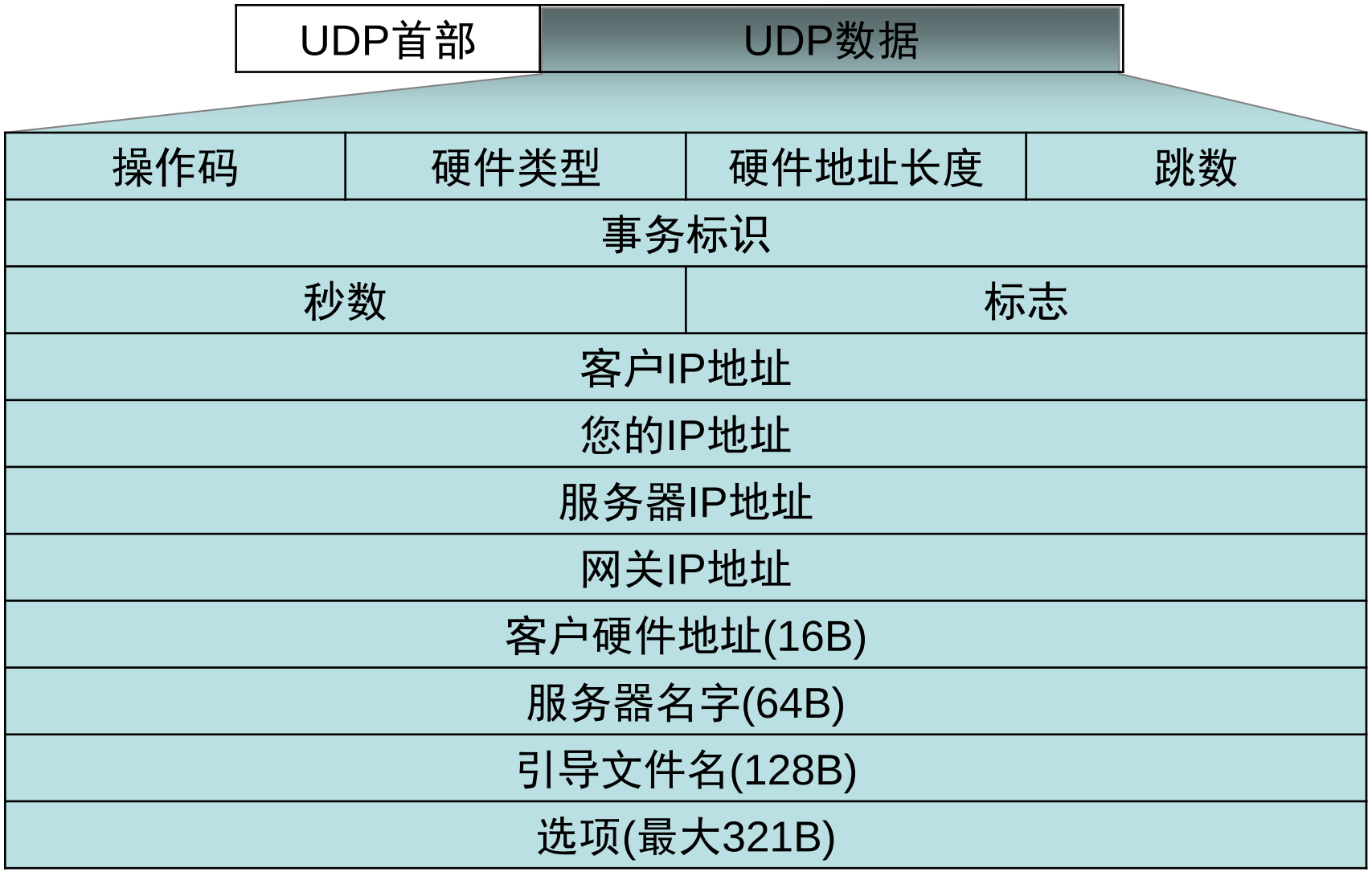
- DHCP 使用客户/服务器模型，指定的DHCP服务器负责分配IP地址，并将配置参数传送为DHCP客户。
- DHCP使用UDP协议，服务器端使用67号端口，客户端使用UDP的68号端口。
- DHCP支持3种IP地址分配机制：
  - 自动分配—DHCP服务器为DHCP客户分配一个永久IP地址
  - 动态分配—DHCP服务器为DHCP客户分配一个有租赁期的临时IP地址
  - 人工分配—DHCP客户的IP地址有管理员分配好，DHCP只负责传达

DHCP中继代理

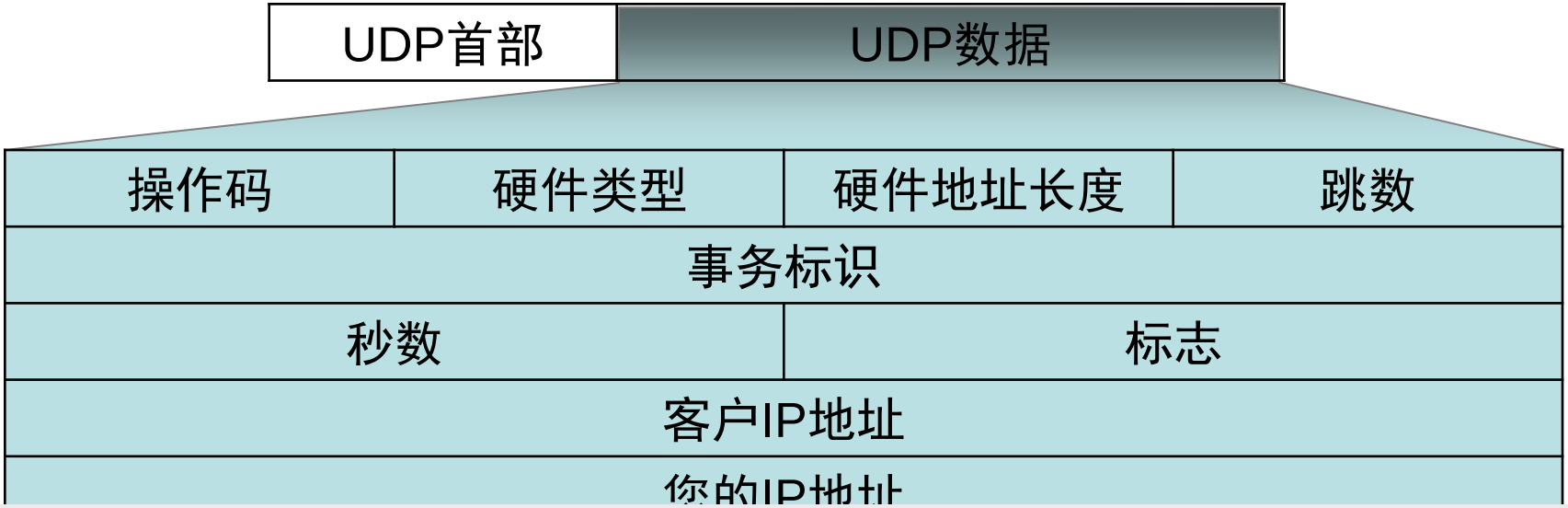
- 不是每个网络上都有 DHCP 服务器，可以设置一个 DHCP 中继代理，它配置了 DHCP 服务器的 IP 地址信息。
- 当 DHCP 中继代理收到主机发送的发现报文后，就以单播方式向 DHCP 服务器转发此报文。并等待其回答。收到 DHCP 服务器回答的提供报文后，DHCP 中继代理再将此提供报文发回给主机。



DHCP报文格式


UDP首部
UDP数据
操作码
硬件类型
硬件地址长度
跳数
事务标识
秒数
标志
客户IP地址
您的IP地址
服务器IP地址
网关IP地址
客户硬件地址(16B)
服务器名字(64B)
引导文件名(128B)
选项(最大321B)

DHCP报文格式

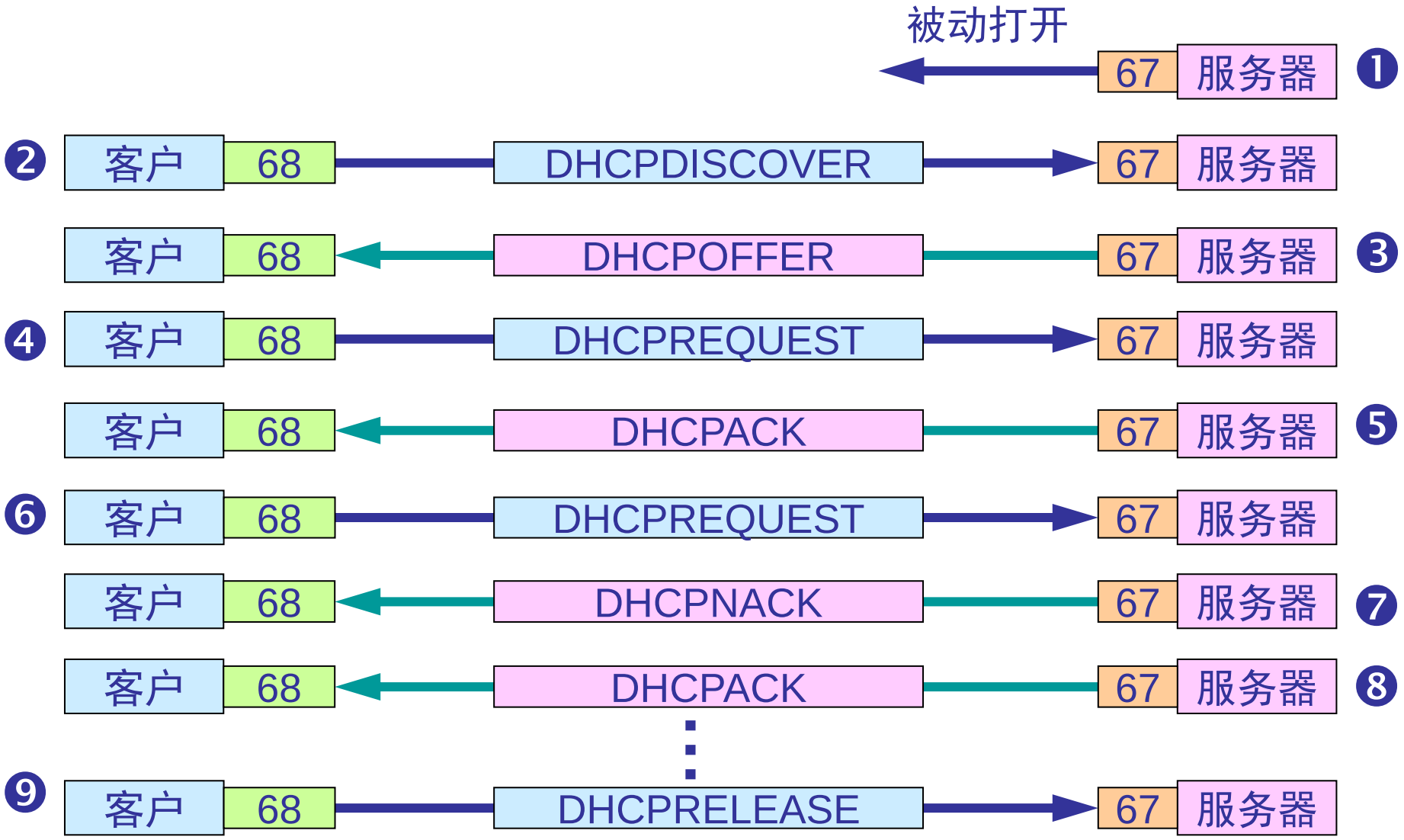


操作码：若是client送给server的封包，设为1，反向为2；
硬件类型：硬件类别，ethernet为1；
硬件地址长度：ethernet为6；
跳数：若数据包需经过router传送，每站加1，若在同一网内，为0；
事务标识：随机数，用于客户和服务器之间匹配请求和相应消息；
秒数：由用户指定的时间，指开始地址获取和更新进行后的时间；
标志：从0-15bits，最左1位为1时表示server将以广播方式传送封包给 client，其余尚未使用；

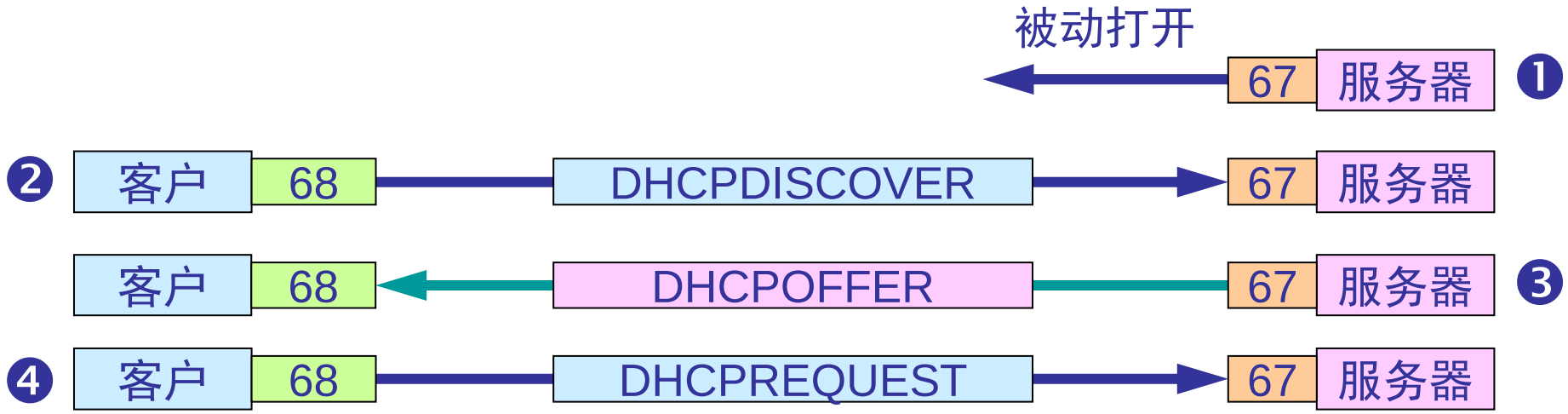
DHCP工作过程

- DHCP工作过程包括：
  - 请求IP地址
    - 发现阶段
    - 提供阶段
    - 选择阶段
    - 确认阶段
  - 续租IP地址
    - DHCP 服务器分配给 DHCP 客户的 IP 地址的临时的，DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。这段时间称为租用期。
  - 释放IP地址

DHCP 协议的工作过程



DHCP 协议的工作过程(1)



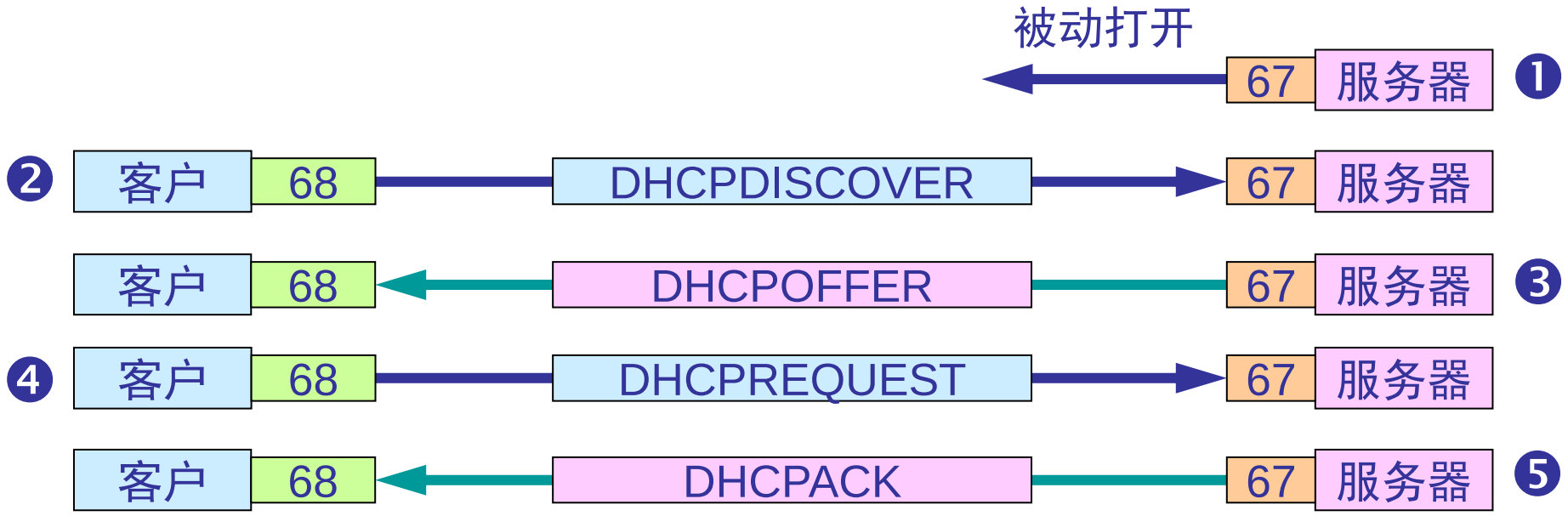
❶ DHCP服务器被动打开UDP端口67，处于等待状态。

❷ DHCP客户从UDP端口68发送DHCP发现报文。

❸凡收到DHCP发现报文的DHCP服务器都发出DHCP提供报文，因此DHCP客户可能收到多个DHCP提供报文。

❹ DHCP客户从几个DHCP服务器中选择其中的一个，并向所选择的DHCP服务器发送DHCP请求报文。

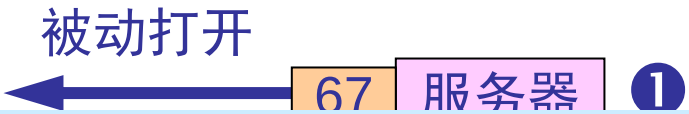
DHCP 协议的工作过程(2)



❺ 被选择的DHCP服务器发送确认报文DHCPACK，进入已绑定状态，并可开始使用得到的临时IP地址了。

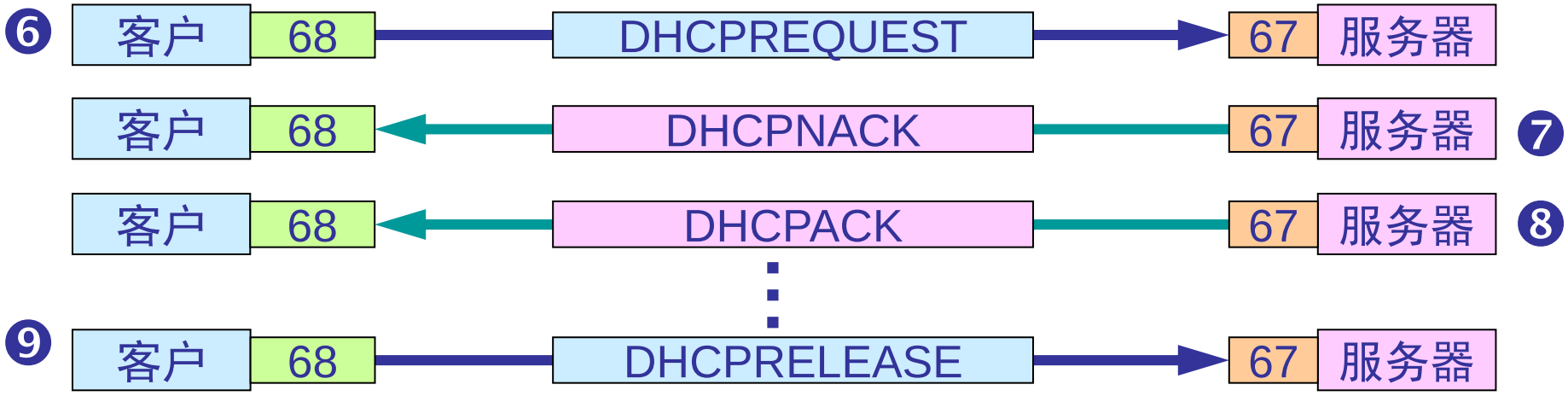
DHCP 客户现在要根据服务器提供的租用期 T 设置两个计时器 T1 和 T2，它们的超时时间分别是 0.5T 和 0.875T。当超时时间到就要请求更新租用期。

DHCP 协议的工作过程(3)



❻ 租用期过了一半(T1 时间到)，DHCP发送请求报文DHCPREQUEST要求更新租用期。

❼ DHCP 服务器若同意，则发回确认报文DHCPACK。DHCP 客户得到了新的租用期，重新设置计时器。

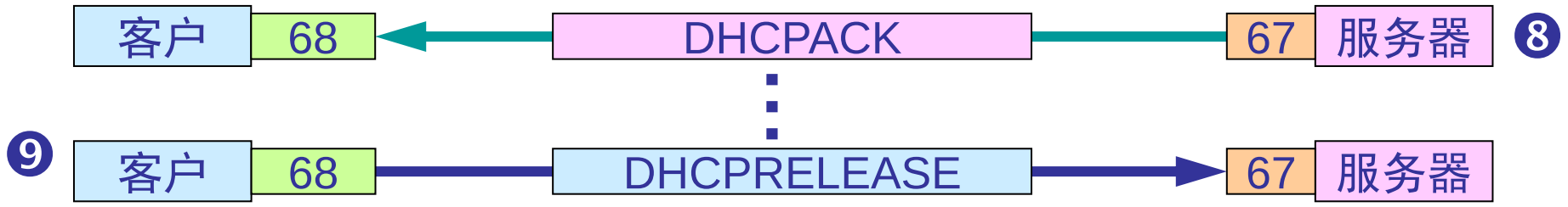


DHCP 协议的工作过程(3)

❽ DHCP服务器若不同意，则发回否认报文DHCPNACK。这时 DHCP 客户必须立即停止使用原来的 IP 地址，必须重新申请 IP 地址（回到步骤❷）。

若DHCP服务器不响应❻的请求报文DHCPREQUEST，则在租用期过了87.5%时，DHCP客户必须重新发送请求报文 DHCPREQUEST(重复❻)，然后又继续后面的步骤。

❾ DHCP客户可随时提前终止租用期，这时只需向DHCP服务器发送释放报文DHCPRELEASE 即可。



DHCP协议操作与优化

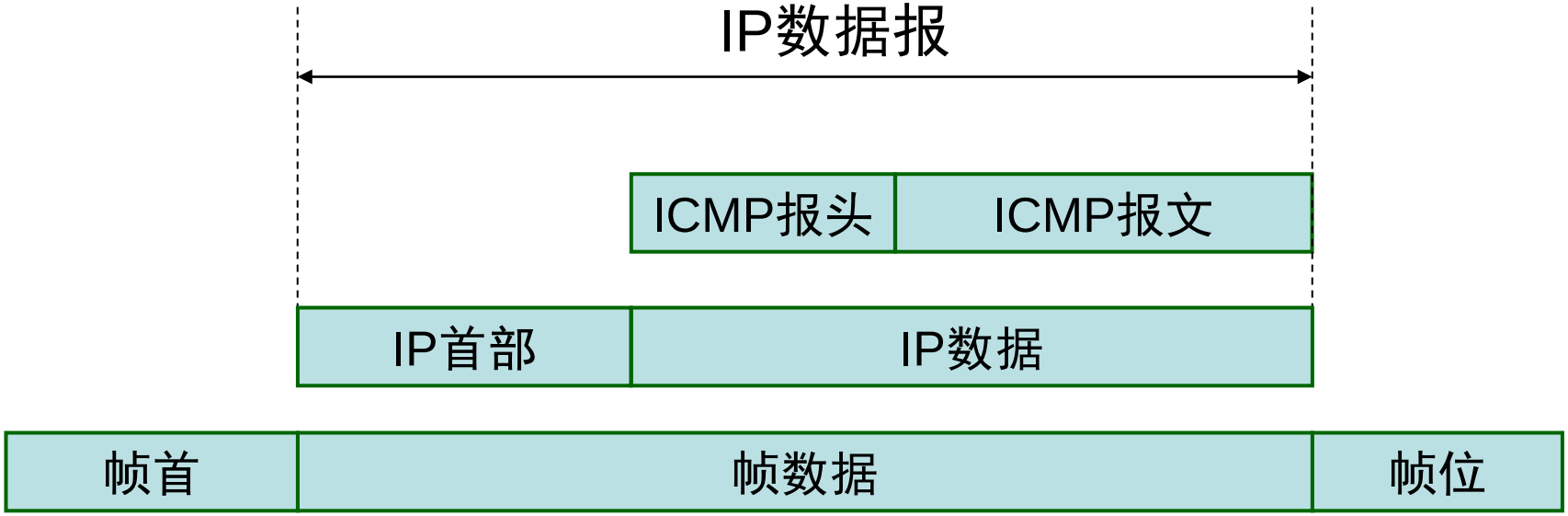
- 优化与操作的措施：
  - 分组出现丢失或重复时的回复操作
  - 对服务器地址进行高速缓存
  - 避免因同时出现大量请求而发生阻塞

5.7 ICMP

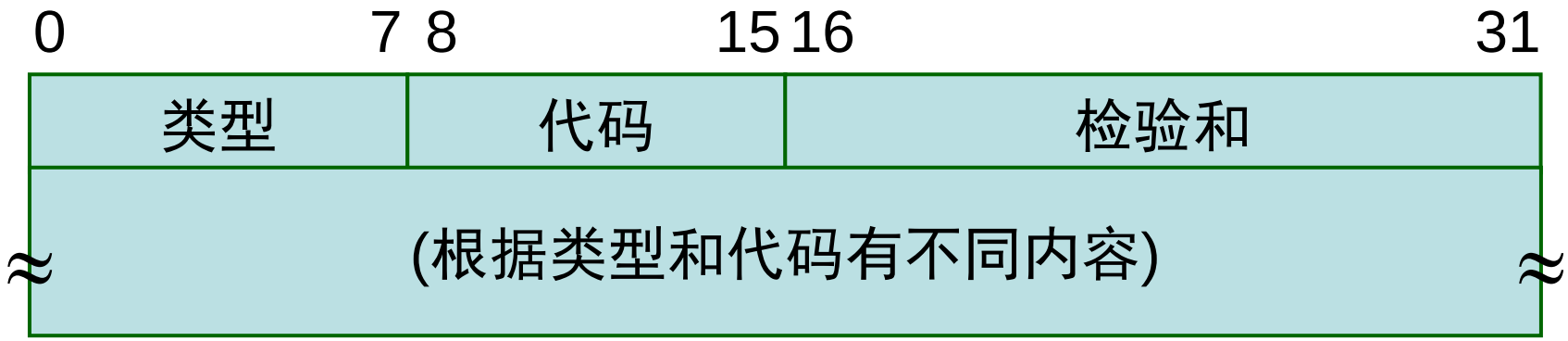
- IP协议是一种不可靠无连接的包传输。当数据包经过多个网络传输过程中，可能出现错误、目的主机不响应、包拥塞和包丢失等。为了处理这些问题，在IP层引入了一个子协议ICMP (Internet Control Message Protocol)

ICMP报文格式和类型

- ICMP数据报封装在IP数据报里传输
- ICMP报文可以被IP协议层、传输层协议（TCP或UDP）和用户进程使用。



ICMP报文格式



- 类型：占8位，有15个不同的值。用来描述特定类型的ICMP报文。
- 代码：占8位，进一步描述某类型的ICMP报文的不同功能。
- 检验和：检验和字段占16位，覆盖整个ICMP报文，包括头部和数据。

ICMP报文类型

- ICMP协议有两种报文：
  - 查询报文
  - 差错报文
- 对错误的ICMP差错报文不会产生另一个ICMP差错报文

ICMP报文的主要类型(部分)

类型	代码	描述	查询	差错
0	0	回显应答(Ping应答)	√	
3	 0 1 2 3	目的不可达 网络不可达 主机不可达 协议不可达 端口不可达		 √ √ √ √
5	0 1	对网络重定向 对主机重定向		√ √
8	0	请求回显（Ping请求）	√	
9 10	0 0	路由器通告 路由器请求	√ √	
12	0	坏的IP首部（包括各种差错）		√
13 14	0 0	时间戳请求 时间戳应答	√ √	
17 18	0 0	地址掩码请求 地址掩码应答	√ √	

检查目的站的可达性

- 为了诊断目的而设计：
  - 源主机IP软件要为数据报选路
  - 源—目的站之间的路由器必须正在运行，且正确为数据报选路
  - 目的主机必须正在运行，且ICMP和IP软件都在工作
  - 返回路径上的路由器必须有正确的路由
- 最常用的调试工具是利用回送请求和回送回答报文来测试目的站的可达性

Ping程序

- 测试网络连通性的ping程序：

```
C:\>ping www.cisco.com
Pinging e144.ca.s.tl88.net [203.110.166.170] with 32 bytes of data:
Reply from 203.110.166.170: bytes=32 time=46ms TTL=48
Reply from 203.110.166.170: bytes=32 time=44ms TTL=48
Reply from 203.110.166.170: bytes=32 time=48ms TTL=48
Reply from 203.110.166.170: bytes=32 time=45ms TTL=48

Ping statistics for 203.110.166.170:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 44ms, Maximum = 48ms, Average = 45ms
```

- ping程序利用了ICMP协议类型8的回显请求和类型0的回显应答完成。

Tracert程序

- tracert程序是测试目的主机路由线路的程序。它利用了ICMP协议的请求回显，并巧妙的利用了TTL值来获得目的

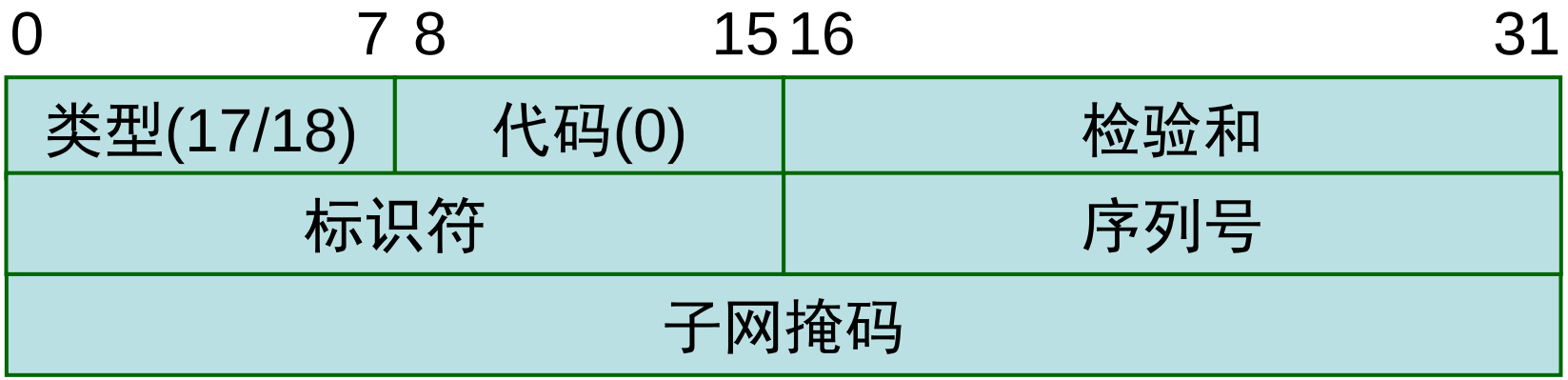
```
C:\>tracert -d bbs.jlu.edu.cn
Tracing route to bbs.jlu.edu.cn [202.198.16.92]
over a maximum of 30 hops:
  1    <1 ms    <1 ms    <1 ms  202.198.31.254
  2     1 ms    <1 ms    <1 ms  192.168.1.121
  3    <1 ms    <1 ms    <1 ms  192.168.2.134
  4    <1 ms    <1 ms    <1 ms  202.198.16.92
Trace complete.
```

时间戳请求和回答

- 两个机器(主机或路由器)可使用时间戳请求和时间戳回答报文来确定IP数据报在这两个机器之间来往所需的往返时间。
- 它也可用作两个机器中的时钟的同步。
- ICMP类型
  - 请求：类型=13
  - 回答：类型=14

ICMP地址掩码请求与响应

- 要得到主机的掩码，主机应向局域网上的路由器发送地址掩码请求报文。
  - 请求：类型=17
  - 回答：类型=18



源点抑制

- ICMP源点抑制报文(类型=4)是为了给IP增加一种流量控制而设计的。
- 当路由器或主机因拥塞而丢弃数据报时，它就向数据报的发送站发送源点抑制报文。
  - 它通知源端，数据报已被丢弃。
  - 它警告源端，在路径中的某处出现了拥塞，因而源端必须放慢发送过程。

其它类型差错报告

- 超时(类型=11)：数据报的生存时间字段值被减为0时，路由器丢弃这个数据报，并向源端发送超时报文。
- 目的不可达(类型=3)：当路由器不能够给数据报找到路由或主机，就丢弃这个数据报，然后这个路由器就向发出这个数据报的源主机发回目的端不可达报文。
- 重定向(类型=5)：路由器给主机发送的更好路由。

关于ICMP

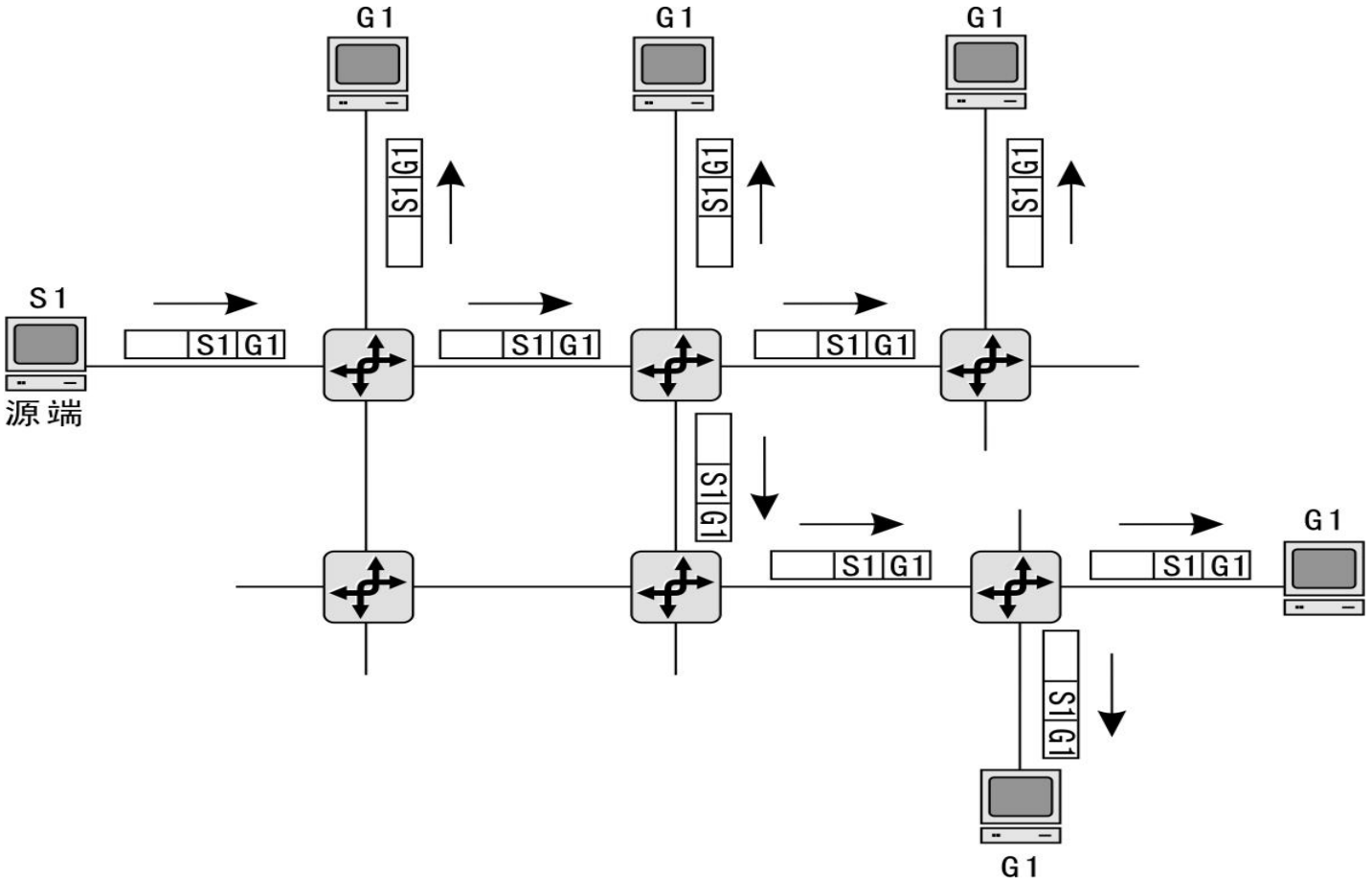
- 为什么限制ICMP只和初始源站点通信？
- 为什么ICMP报文用IP封装和发送？

5.8 Internet组管理协议IGMP

- IP地址有三种类型，分别是：
  - 单播地址
  - 广播地址
  - 多播地址
- 广播和多播地址仅应用于UDP协议，它们主要应用在将报文同时传送到多个接收者的情况。

多播的基本概念

- 多播(multicast)处于单播和广播之间：帧仅传送给属于多播组的多个主机。



IP多播

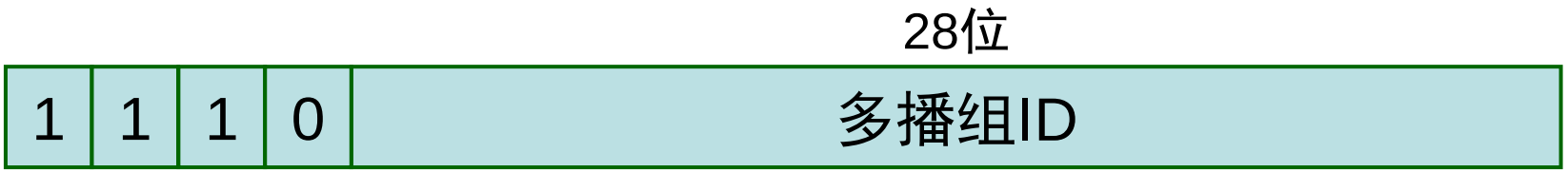
- IP多播是指在IP网中将IP报文以尽力传送的形式发送到网络中的某个确定节点子集。这个子集称为多播组。
- 基本思想：源主机只发送一份IP报文，其目的IP地址为IP多播地址，加入到该多播组的主机都可以接收到这个IP报文的拷贝。
- IP多播技术有效地解决了单点发送多点接收的问题。

IP多播的概念性组成部分

- 多播编址方法
  - 多播组用D类IP地址(224.0.0.0~239.255.255.255)标识。
  - IP首部协议字段值2(IGMP)
- 有效的通知和交付机制(网际组管理协议)
  - 通知机制：把自己参与的多播组通知路由器
  - 交付机制：路由器把多播分组传输给主机
- 有效的网络间转发工具(多播路由选择协议)
  - 有效：希望沿最短路径发送多播分组
  - 动态：允许主机任意参与或退出多播群组

多播组地址

- 多播地址只能用作目的地址，不能用作源地址

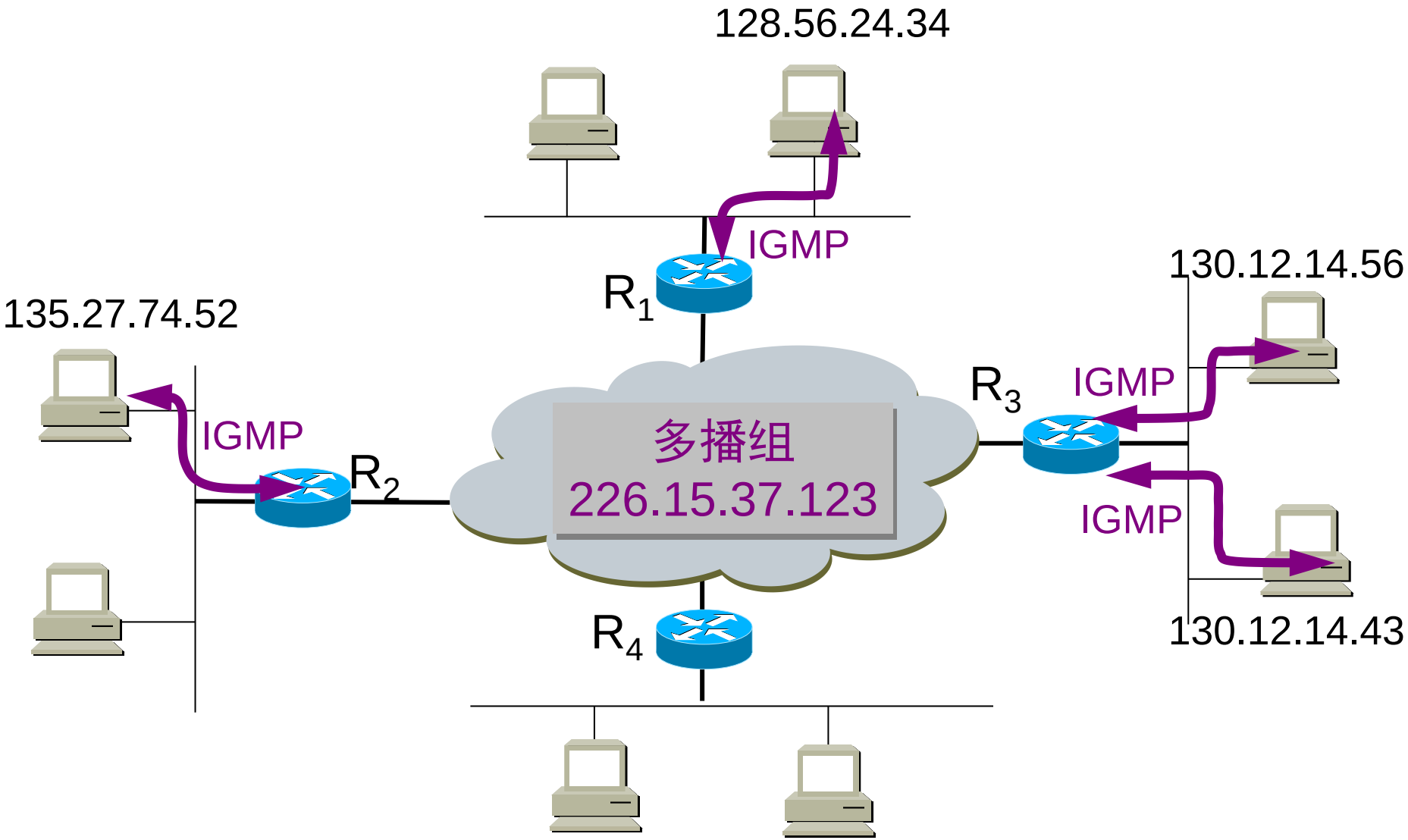


- 多播地址划分

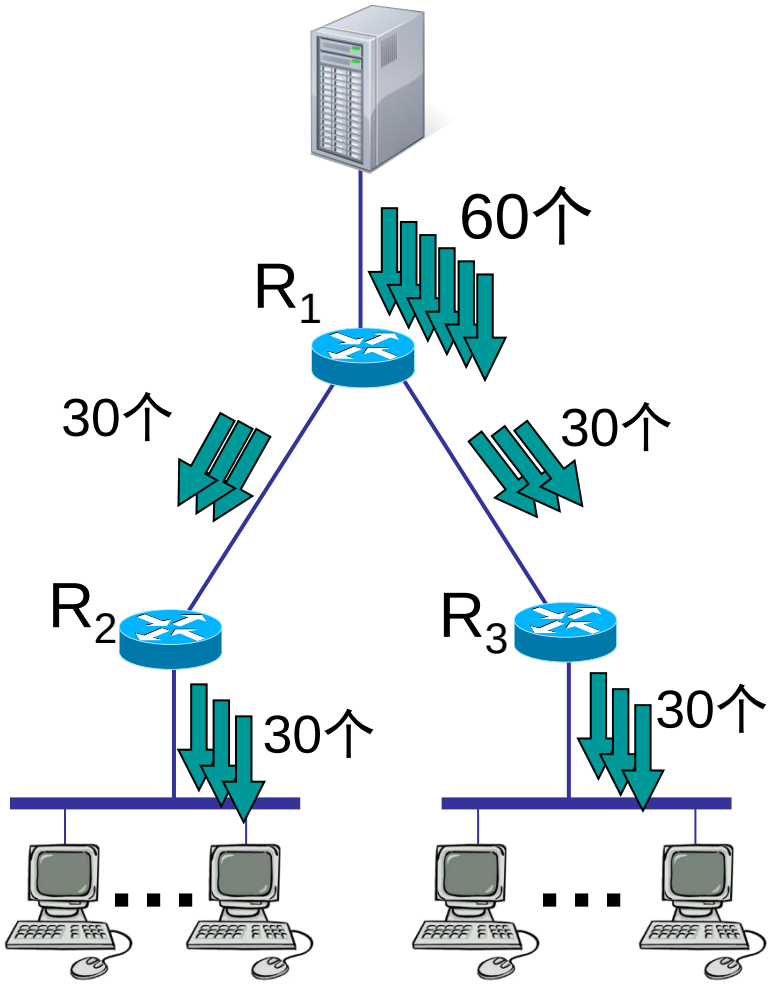
范围	用途
224.0.0.0~224.0.0.255	IANA预留
224.0.1.0~238.255.255.255	用户多播地址，全网有效
239.0.0.0~239.255.255.255	本地管理多播地址，特定范围内有效

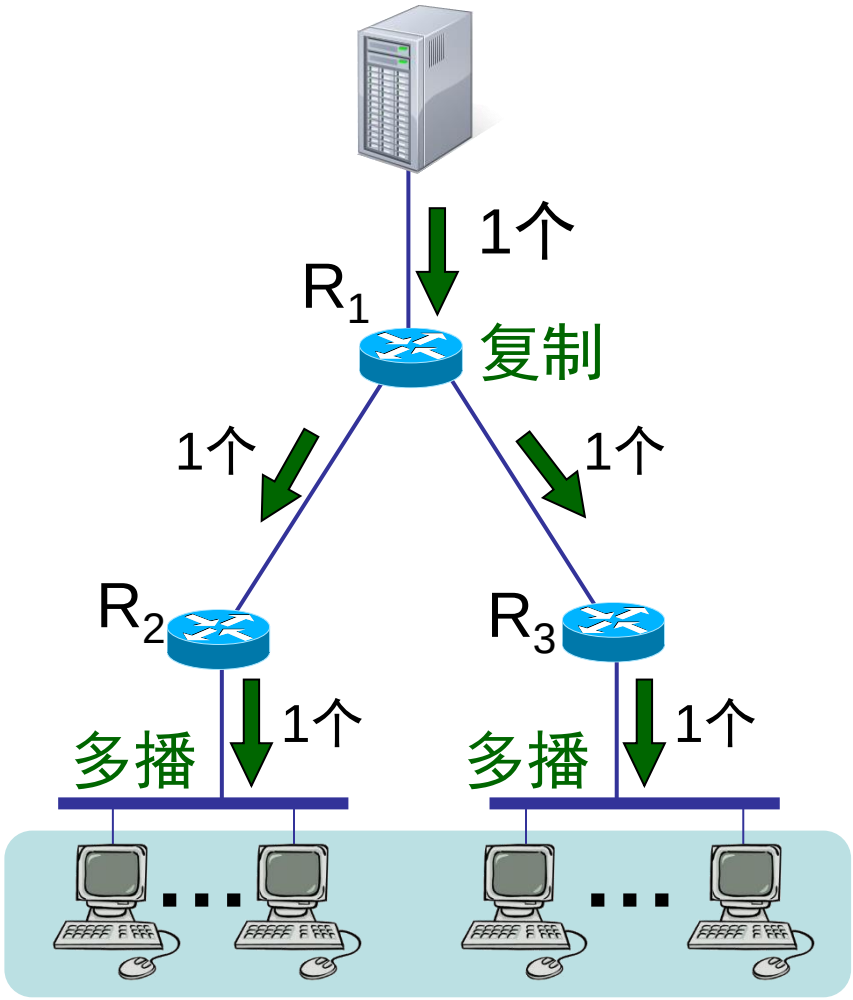
Protocol	Multicast IP address	Multicast MAC address
RIP V2	224.0.0.9	01:00:5e:00:00:09
OSPF V2	224.0.0.5	01:00:5e:00:00:05
	224.0.0.6	01:00:5e:00:00:06
EIGRP	224.0.0.10	01:00:5e:00:00:0a

多播路由器


128.56.24.34
IGMP
R1
135.27.74.52
IGMP
R2
多播组
226.15.37.123
R3
IGMP
130.12.14.56
IGMP
130.12.14.43
R4

单播与多播的比较

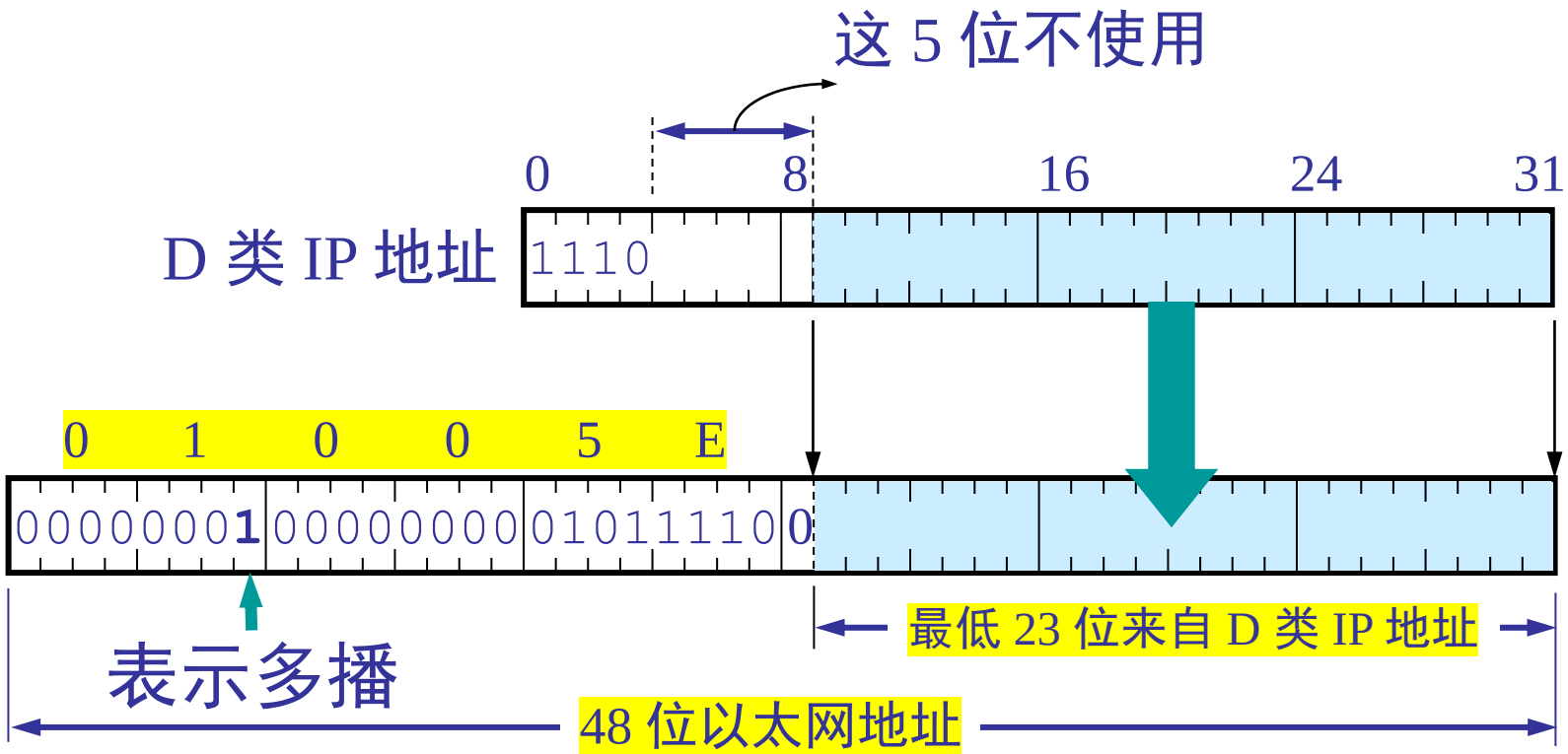




多播组成员共有 60 个

IP多播与MAC组播地址映射过程

- IANA规定，将01:00:5E:00:00:00~01:00:5E:7F:FF:FF用于IP组播地址到以太网组播地址的映射。
  - 注意：IP—MAC映射关系不是唯一的



地址映射示例

- 一台以太网主机加入组播组225.128.47.81，具有什么样的MAC地址的一个帧的到达将引起网络接口卡的中断CPU？
- 答：将IP组播地址的低24位表示为二进制：

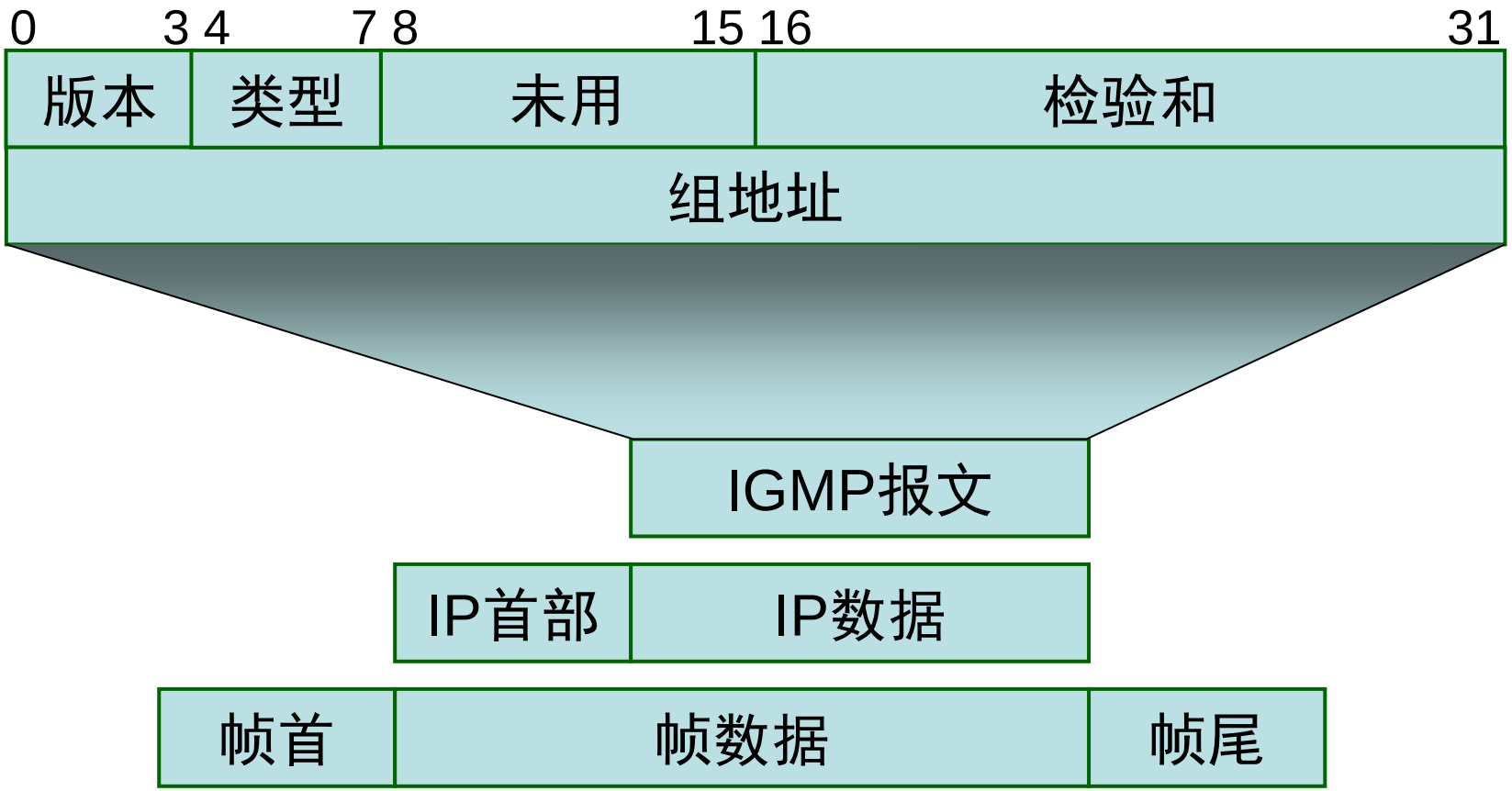
  10000000 00101111 01010001

  所以，MAC地址为01-00-5E-00-2F-51的帧将会引起中断。

IGMP协议

- IGMP是Internet Group Management Protocol（互联网组管理协议）的简称。
- IGMP是TCP/IP协议族中负责 IP多播成员管理的协议，用来在 IP主机和与其直接相邻的组播路由器之间建立、维护多播组成员关系。
- IGMP的版本：到目前为止，有3个版本
  - IGMPv1（由 RFC 1112定义）
  - IGMPv2（由 RFC 2236定义）
  - IGMPv3（由 RFC 3376定义）

IGMP v1报文格式



版本：1

类型：1-路由器发出的报文，2-主机发出的报文

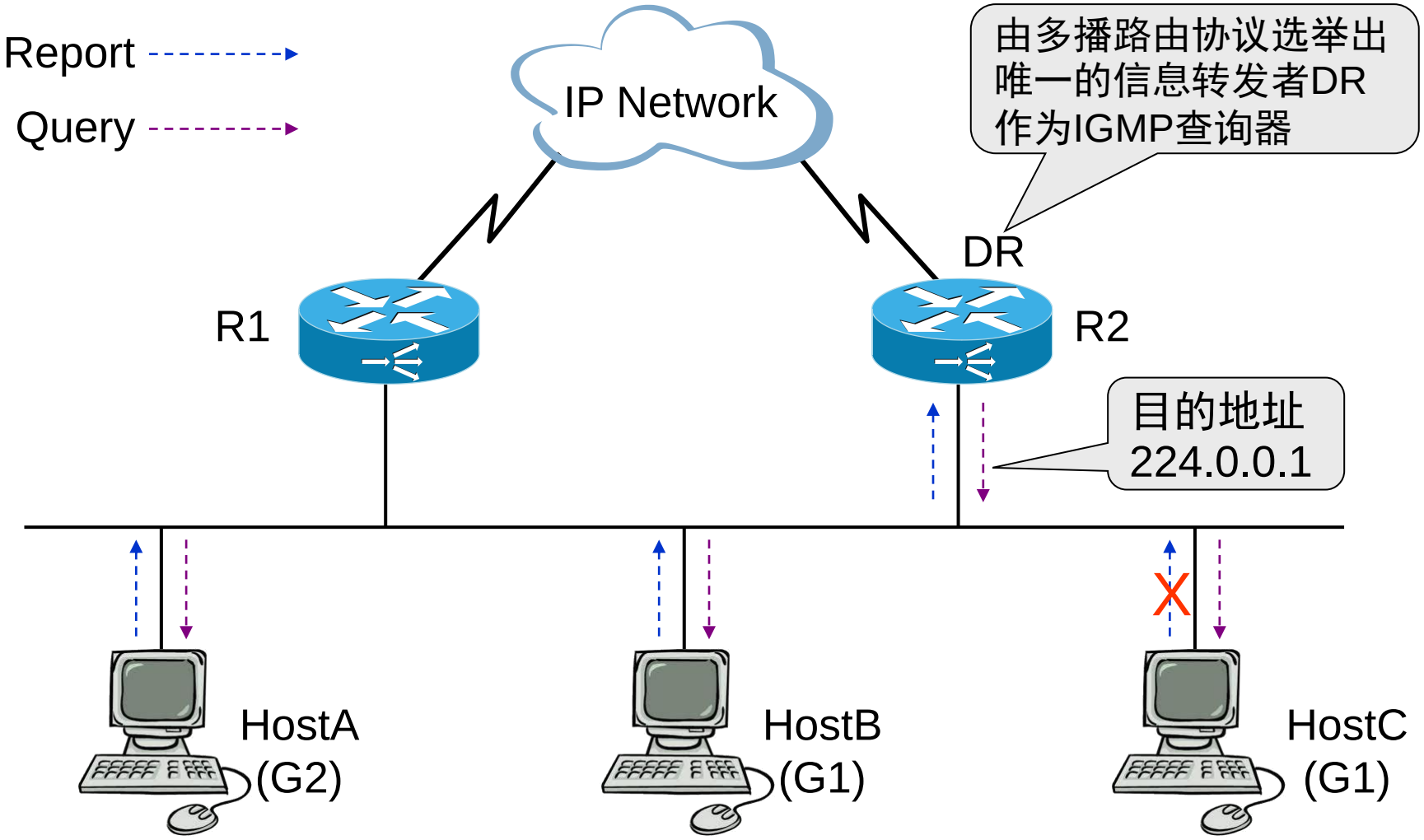
IGMP工作机制

- 第一阶段：当某个主机加入新的多播组时，该主机应向多播组的多播地址发送IGMP 报文，声明自己要成为该组的成员。本地的多播路由器收到 IGMP 报文后，将组成员关系转发给因特网上的其他多播路由器。
- 第二阶段：因为组成员关系是动态的，因此本地多播路由器要周期性地探询本地局域网上的主机，以便知道这些主机是否还继续是组的成员。
  - 只要对某个组有一个主机响应，那么多播路由器就认为这个组是活跃的。
  - 但一个组在经过几次的探询后仍然没有一个主机响应，则不再将该组的成员关系转发给其他的多播路由器。

IGMP采用的一些具体措施

- 在主机和多播路由器之间的所有通信都使用 IP 多播。
- 多播路由器在探询组成员关系时，只需要对所有的组发送一个请求信息的询问报文。
- 当同一个网络上连接有几个多播路由器时，它们能够迅速和有效地选择其中的一个来探询主机的成员关系。
- 在 IGMP 的询问报文中有一个数值 N，它指明一个最长响应时间（默认值为 10秒）。当收到询问时，主机在 0 到 N 之间随机选择发送响应所需经过的时延。对应于最小时延的响应最先发送。
- 同一个组内的每一个主机都要监听响应，只要有本组的其他主机先发送了响应，自己就可以不再发送响应了。

多播工作机制示例(IGMP v1)



IGMP v1主要基于查询和响应机制完成对多播组成员的管理

多播路由算法

- 多播路由选择协议尚未标准化
  - 一个多播组中的成员是动态变化的。多播路由选择实际上就是要找出以源主机为根结点的多播转发树。
  - 对不同的多播组对应于不同的多播转发树。同一个多播组，对不同的源点也会有不同的多播转发树。
- 几种多播路由算法：
  - 基于链路状态的路由选择
  - 基于距离向量的路由选择
  - 协议无关的组播

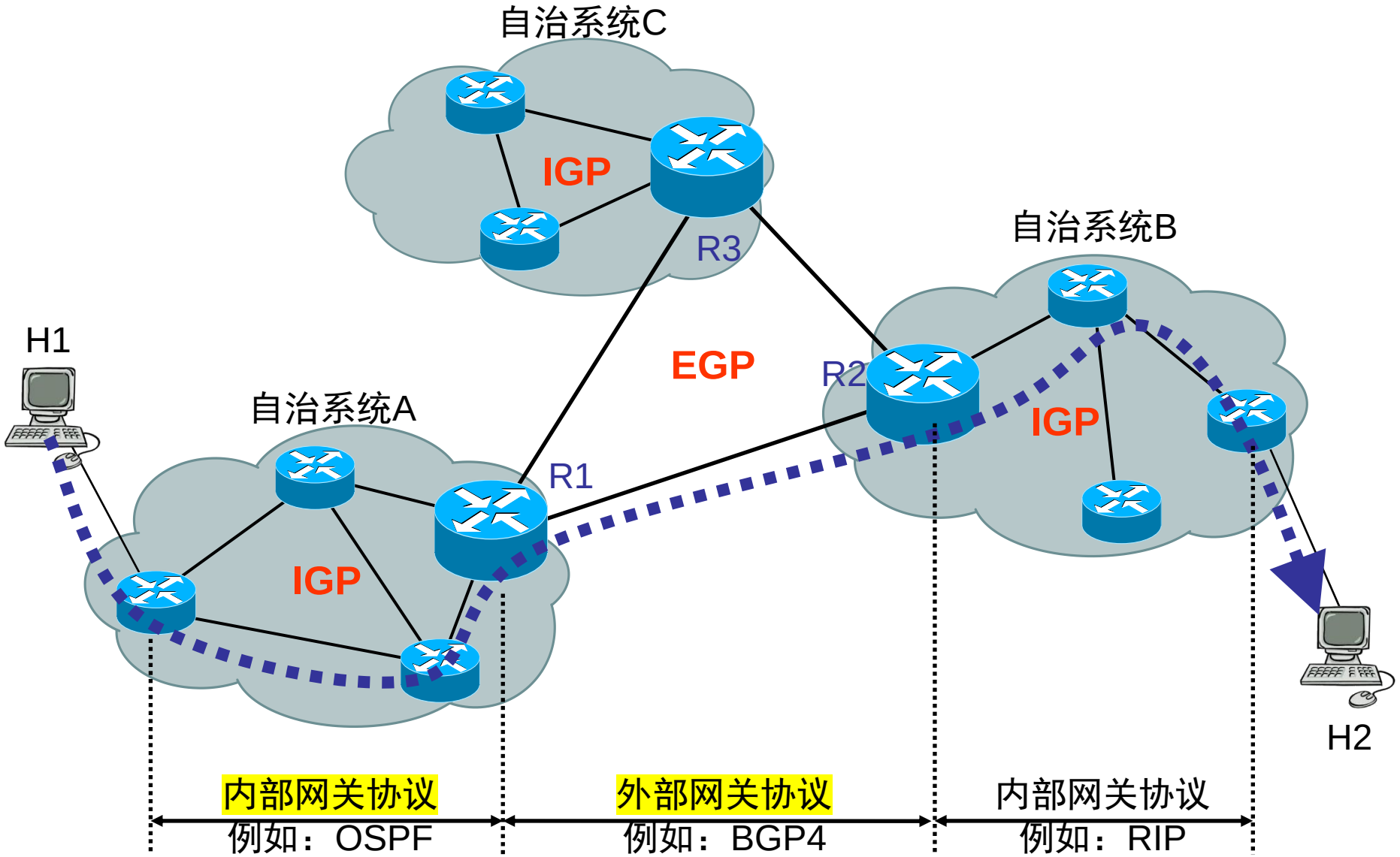
5.9 Internet路由问题

- 互连网中提供两级路由协议：
  - 内部网关协议IGP (Interior Gateway Protocol)
  - 外部网关协议EGP (Exterior Gateway Protocol)

互连网络的路由问题

- 网络互连可能需要多协议路由器，多协议路由器可以处理多种通信协议。
- 自治系统AS (Autonomous System)：一个自治系统就是处于一个管理机构控制之下的路由器和网络群组。
- 一个自治系统中的所有路由器需要相互连接，运行相同的路由协议。
- 外部网关互连会涉及更多问题。

自治系统和内部、外部网关协议



5.9.1 内部网关路由选择协议

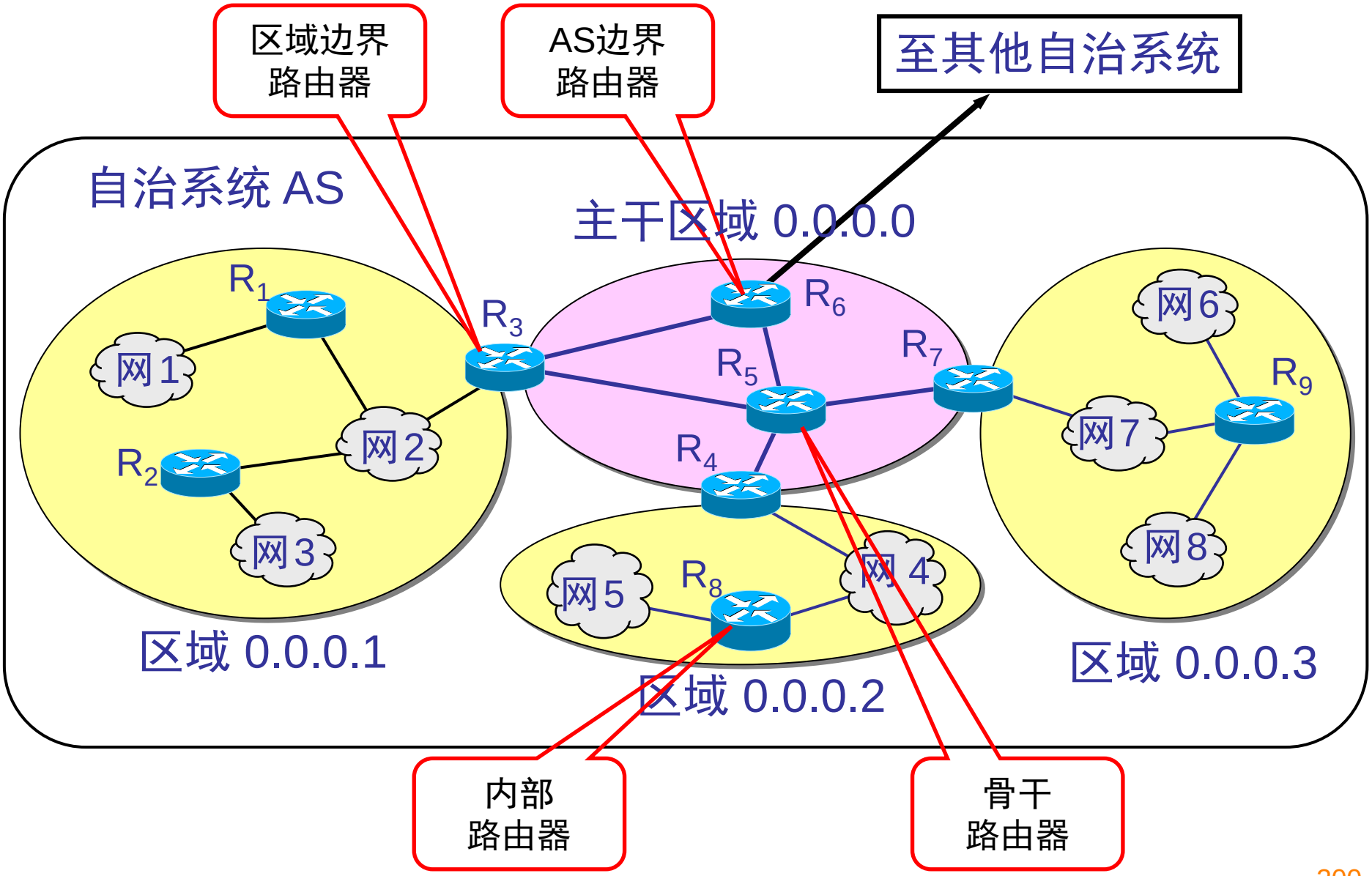
- OSPF (Open Shortest Path First)开放最短路径优先。
- OSPF路由协议是典型的链路状态路由协议，是互连网应用最广的路由协议之一。
- OSPF的SPF算法
  - SPF算法是OSPF路由协议的基础。SPF算法有时也被称为Dijkstra算法 。
- 在OSPF路由协议中，最短路径树的树干长度，称为OSPF的Cost，其算法为：

$$Cost = 100 \times 10^6 / 链路带宽(bps)$$

OSPF的三个要点

- 向本自治系统中所有路由器发送信息，这里使用的方法是洪泛法。
- 发送的信息就是与本路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。
  - “链路状态”就是说明本路由器都和哪些路由器相邻，以及该链路的“度量”(metric)。
- 只有当链路状态发生变化时，路由器才用洪泛法向所有路由器发送此信息。
  - OSPF 还规定每隔一段时间（如 30 分钟）要刷新一次数据库中的链路状态。

OSPF层次结构的区域划分



划分区域

- OSPF 使用层次结构的区域划分。在上层的区域叫作主干区域(backbone area)。主干区域的标识符规定为0.0.0.0。主干区域的作用是用来连通其它下层的区域。
- 优点：将利用洪泛法交换链路状态信息的范围局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。
- 在一个区域内部的路由器只知道本区域的完整网络拓扑，而不知道其他区域的网络拓扑的情况。

OSPF的其它特点

- OSPF 对不同的链路可根据分组的不同服务类型 TOS 而设置成不同的代价。因此，OSPF 对于不同类型的业务可计算出不同的路由。
- 如果到同一个目的网络有多条相同代价的路径，那么可以将通信量分配给这几条路径。这叫作多路径间的负载平衡。
- 所有在 OSPF 路由器之间交换的分组都具有鉴别的功能。
- 支持可变长度的子网划分和无分类编址 CIDR。
- 每一个链路状态都带上一个 32 位的序号，序号越大状态就越新。

OSPF数据包格式（24Byte）

0	7 8	15 16	31
版本(Version)	类型(Type)	分组长度(Packet length)	
路由器标识符(Router ID)			
区号(Area ID)			
校验和(Checksum)		认证类型(Authentication Type)	
认证(Authentication)			
认证(Authentication)			

类型(Type)	描述
Hello	用于发现谁是邻居
Database description	通知发送者有哪些更新
Link date request	从伙伴处请求信息
Link state update	为邻居提供发送者的开销
Link state ack	确认链路状态更新

认证类型：

- 0—不用
  - 认证填入0
- 1—口令
  - 认证填入8字符口令

5.9.2 外部网关路由选择协议

- BGP(Border Gateway Protocol)协议是一种距离向量协议。
- 使用TCP作为传输协议—本身不需要差错控制和重传机制。
- 使用增量的、触发性的路由更新，而不是一般的距离向量协议的整个路由表的、周期性的更新。它通告前往目的地的一系列自治系统号。
- BGPv4是典型的外部网关协议，完成自治系统间的路由选择问题。

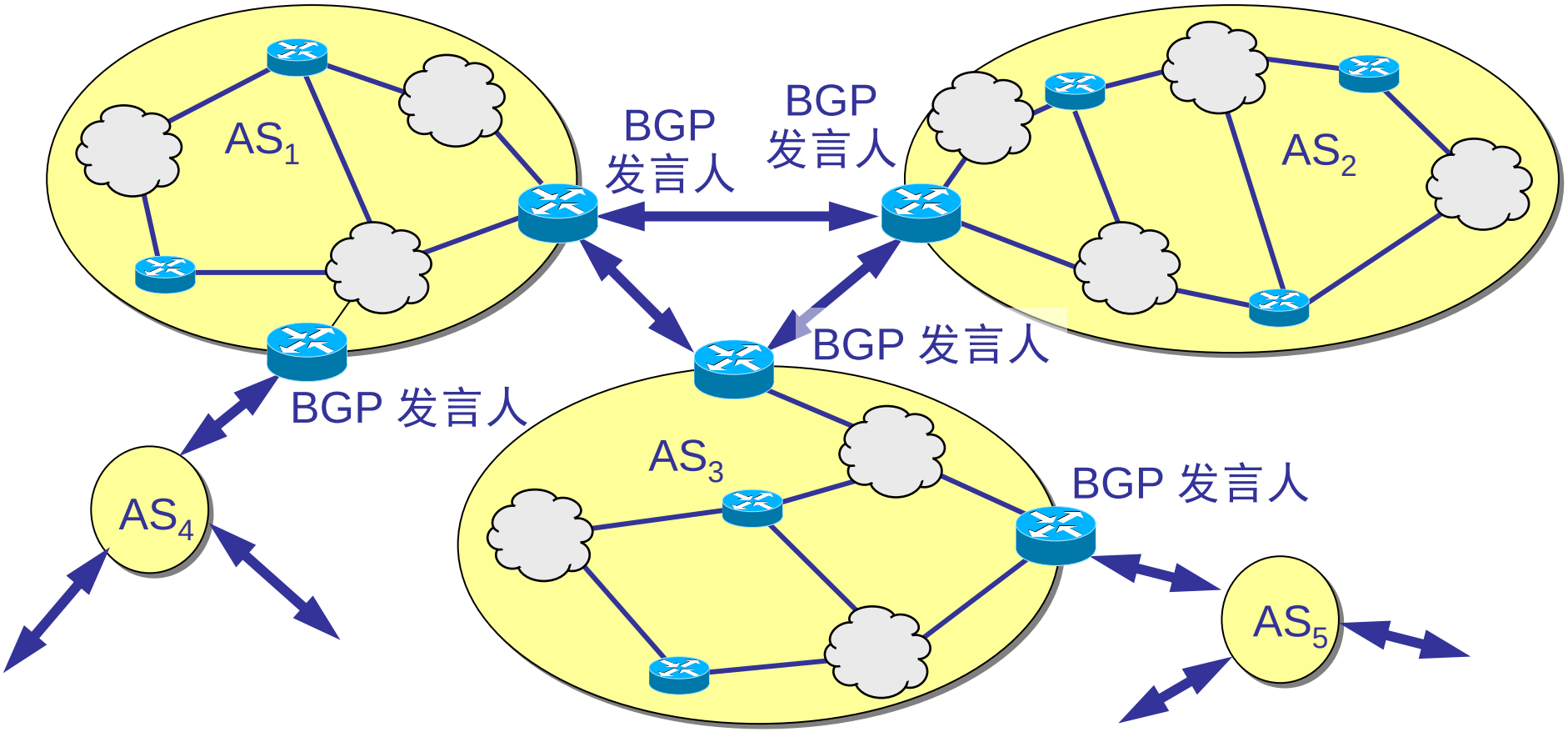
BGP使用的环境不同

- 网络的规模太大时，使得自治系统之间路由选择非常困难。对于自治系统之间的路由选择，要寻找最佳路由是很不现实的。
  - 当一条路径通过几个不同 AS 时，要想对这样的路径计算出有意义的代价是不太可能的。比较合理的做法是在 AS 之间交换“可达性”信息。
- 自治系统之间的路由选择必须考虑有关策略。
  - 教育网络不承载商业流量
  - 起止于Apple的流量不应该经过Google中转
- 边界网关协议 BGP 只能是力求寻找一条能够到达目的网络且比较好的路由（不能兜圈子），而并非要寻找一条最佳路由。

“BGP 发言人”

- 每一个自治系统的管理员要选择至少一个路由器作为该自治系统的“BGP 发言人”。
- 一般说来，两个 BGP 发言人都是通过一个共享网络连接在一起的，而 BGP 发言人往往就是 BGP 边界路由器，但也可以不是 BGP 边界路由器。

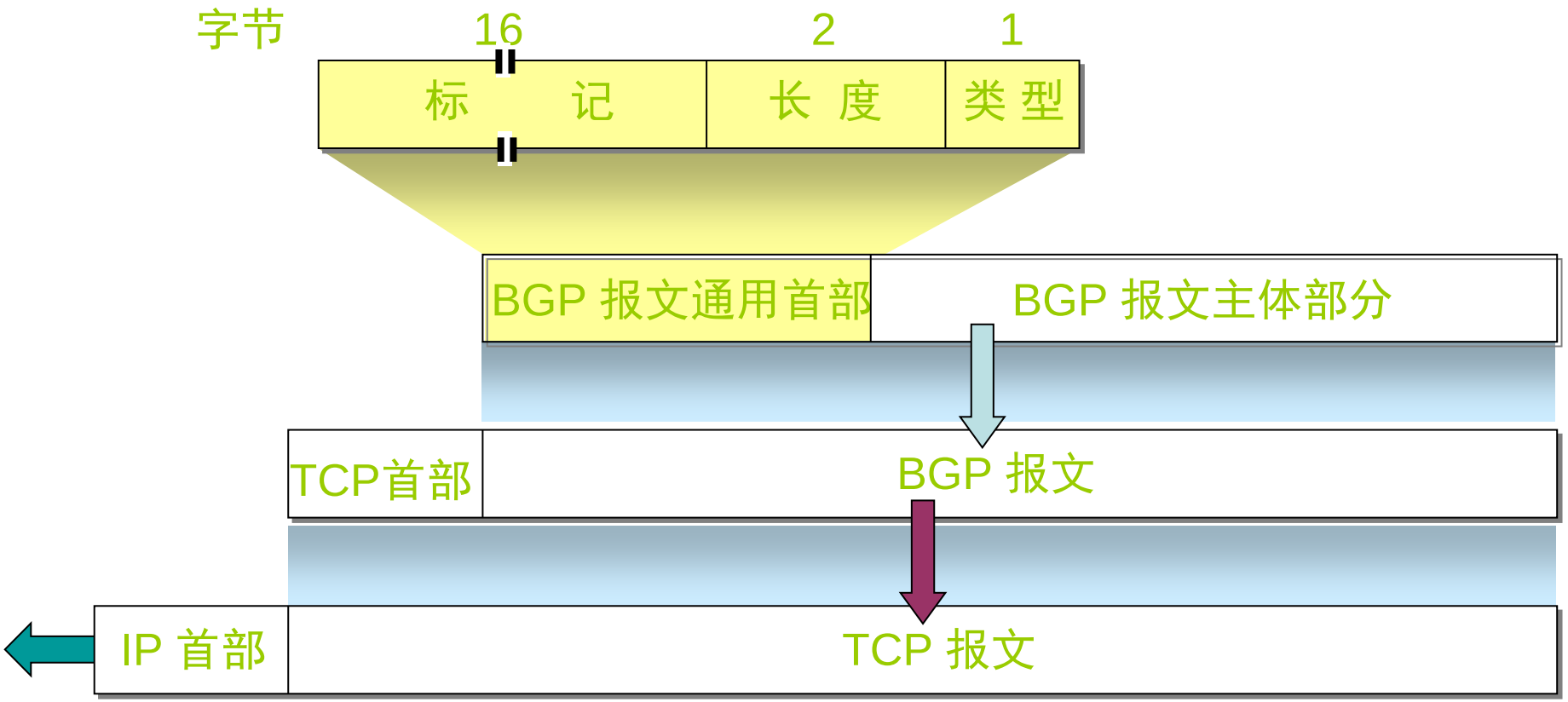
BGP发言人和自治系统 AS 的关系



BGP 协议的特点

- BGP 协议交换路由信息的结点数量级是自治系统数的量级，这要比这些自治系统中的网络数少很多。
- 每一个自治系统中 BGP 发言人（或边界路由器）的数目是很少的。这样就使得自治系统之间的路由选择不致过分复杂。
- BGP 支持 CIDR，因此 BGP 的路由表也就应当包括目的网络前缀、下一跳路由器，以及到达该目的网络所要经过的各个自治系统序列。
- 在BGP 刚刚运行时，BGP 的邻站是交换整个的 BGP 路由表。但以后只需要在发生变化时更新有变化的部分。这样做对节省网络带宽和减少路由器的处理开销方面都有好处。

BGP 报文格式



BGP-4 使用四种报文类型

- 打开(OPEN)报文：6个字段，用来与相邻的另一个BGP发言人建立关系。
- 更新(UPDATE)报文：5个字段，用来发送某一路由的信息，以及列出要撤消的多条路由。
- 保活(KEEPALIVE)报文：19字节的通用首部，用来确认打开报文和周期性地证实邻站关系。
- 通知(NOTIFICATION)报文：3个字段，用来发送检测到的差错。
- 在 RFC 2918 中增加了 ROUTE-REFRESH 报文，用来请求对等端重新通告。