

# 计算机网络

胡亮 等编著

# 第5章 网络层

---

5.1 网络层基本原理

5.2 IPv4协议

5.3 IPv6协议

5.4 互联网路由问题

5.5 软件定义网络

5.6 多协议标签交换与段路由协议

5.7 本章总结

## 5.1 网络层基本原理

---

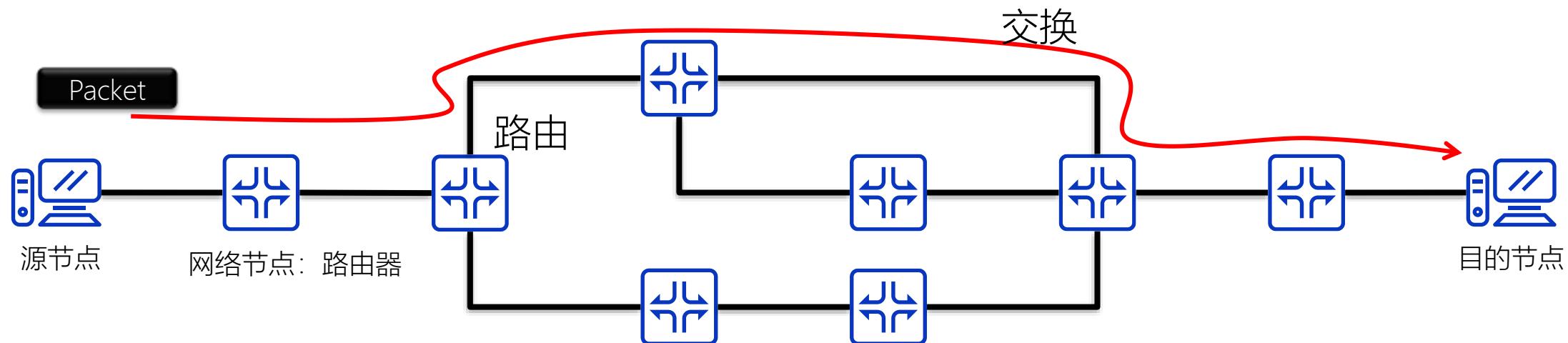
- 网络层功能和服务
- 网络层互连设备
- 路由选择原理
- 拥塞控制和服务质量

# 网络层的功能和服务

- 提供任意两个网络节点之间的通信
- 实现端到端的传递，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径
- 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



# 网络层的服务（1）

## ■ OSI模型定义两种服务类型

### 面向连接的服务 (CONS)

- 类似固定电话的虚电路模型
- 一次完整的数据传输需要建立连接，传输数据，拆除连接等三个步骤
- 来自源节点的所有数据包都以相同的方式路由(路由器需要保存状态)
- 典型的网络技术：ATM，帧中继，X.25

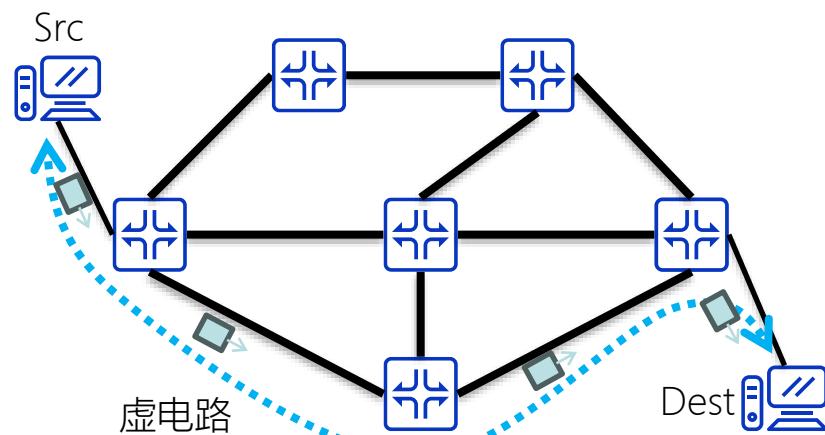
### 无连接的服务 (CLNS)

- 类似邮政的投递服务
- 无需建立无连接
- 尽最大努力或不可靠的服务，即网络层不能保证数据包传送到目的节点
- 来自源节点的每个数据包都独立路由
- 典型的网络技术：IP

# 网络层的服务 (2)

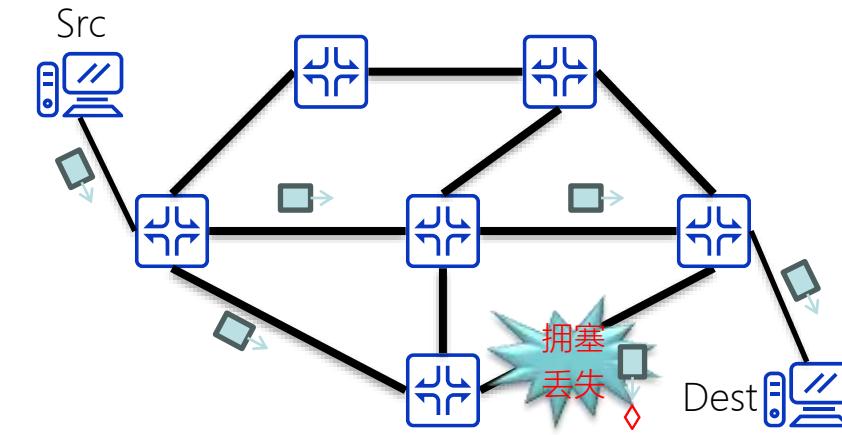
## 面向连接的服务 (CONS)

- Src发送给Dest的所有数据包都沿着同一条虚电路传送
- 一个节点出现故障时，所有通过该结点的虚电路均不能工作
- 数据包按照发送的顺序到达目的节点



## 无连接的服务 (CLNS)

- Src发送给Dest的数据包可能沿着不同路径传送
- 一个节点出现故障时，后续数据包重新路由到其他节点
- 数据包到达目的节点可能出现乱序



# 面向连接网络服务的优缺点

## 面向连接的服务 (CONS)

- 优点：
  - 允许一个协议包含顺序、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
  - 比无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

## 无连接的服务 (CLNS)

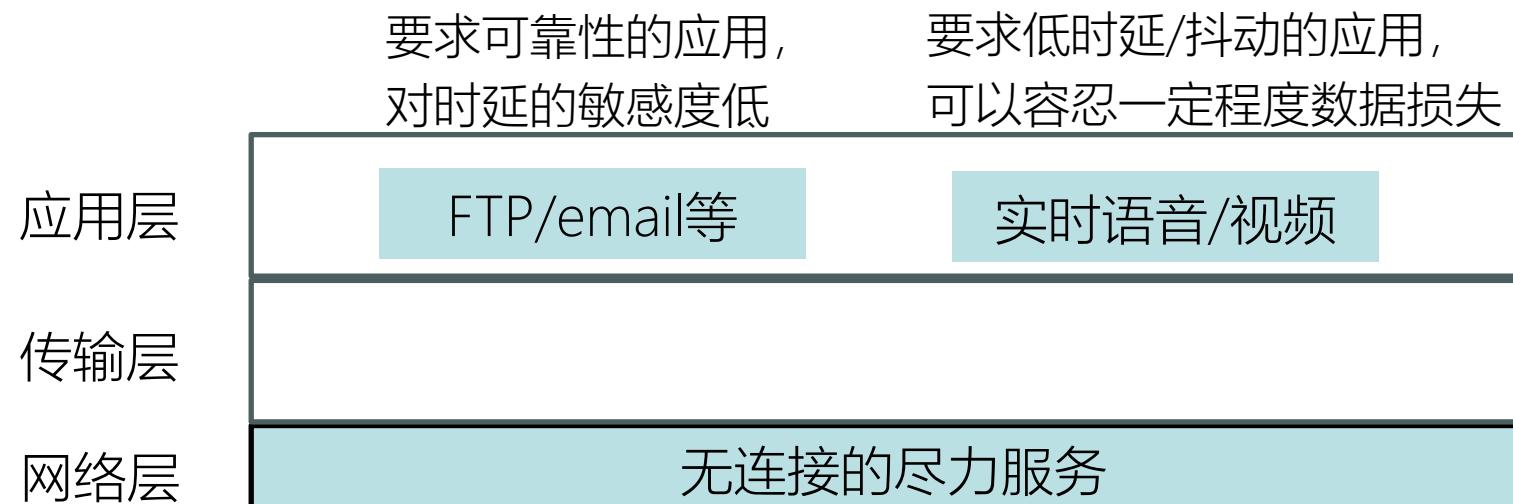
- 优点
  - 如果可靠性和排序可由上层协议来处理的话，具有速度和开销方面的优势
  - 如果某一条路经发生阻塞或中断，包可以选择另一条路经
  - 单个传输的各个片断可以通过不同的路径传输，从而达到最大的效率
- 缺点
  - 不可靠，无法保证数据包顺序到达
  - 每个包所需的开销较大，每个包必须携带完整的地址信息

# 网络层应提供什么样的服务

- 在计算机网络领域，网络层应该向传输层提供怎样的服务？
  - 焦点：可靠交付应当由谁来负责？是网络还是端系统？
- TCP/IP的网络层设计思路
  - 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 IP 数据报独立发送，与其前后的 IP 数据报无关（不进行编号）
  - 网络层不提供服务质量的承诺。即所传送的IP 数据报可能出错、丢失、重复和失序（不按序到达终点），当然也不保证IP 数据报传送的时限

# IP层为何选择无连接的服务

- 网络层应该只提供在各种应用中广泛使用的一般服务，而具体的应用功能应该在较高的层在终端主机上实现
- 应用程序在可靠性、吞吐量和延迟方面有不同的需求



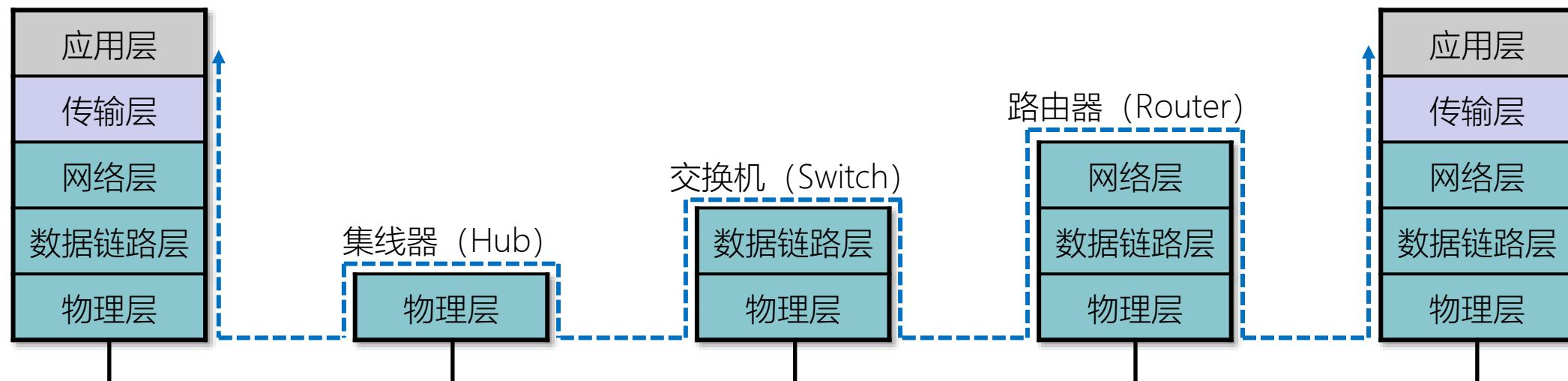
- 在网络层提供虚电路服务不能有效地支持那些不需要连接或可靠性的应用，甚至可能排除那些受时延增加影响的应用（如语音的实时传递）
- 终端主机的高层协议无法在网络层虚电路的基础上构建高效的数据报服务，网络层引入的时延和开销无法在更高的层中消除

# 无连接服务的优点

- 网络层不提供端到端的可靠传输服务，网络中的设备不必维护和保存海量的连接状态
  - 可以降低网络设备的技术复杂度，从而降低网络设备的成本及价格（与传统电网的设备相比较）
- 提供无连接服务的网络层可以满足所有应用的需求，并提高网络的生存能力，降低网络的造价
- 对于需要可靠性交付的应用，传输层在网络层提供的无连接服务的基础上，通过差错处理、流量控制、超时重传等机制实现可靠的通信，并在终端上实现
- 因特网发展到今日的规模，充分证明了当初采用分层设计思想和端到端设计原则的正确性

# 网络层互连

- 网络层互连设备主要是路由器

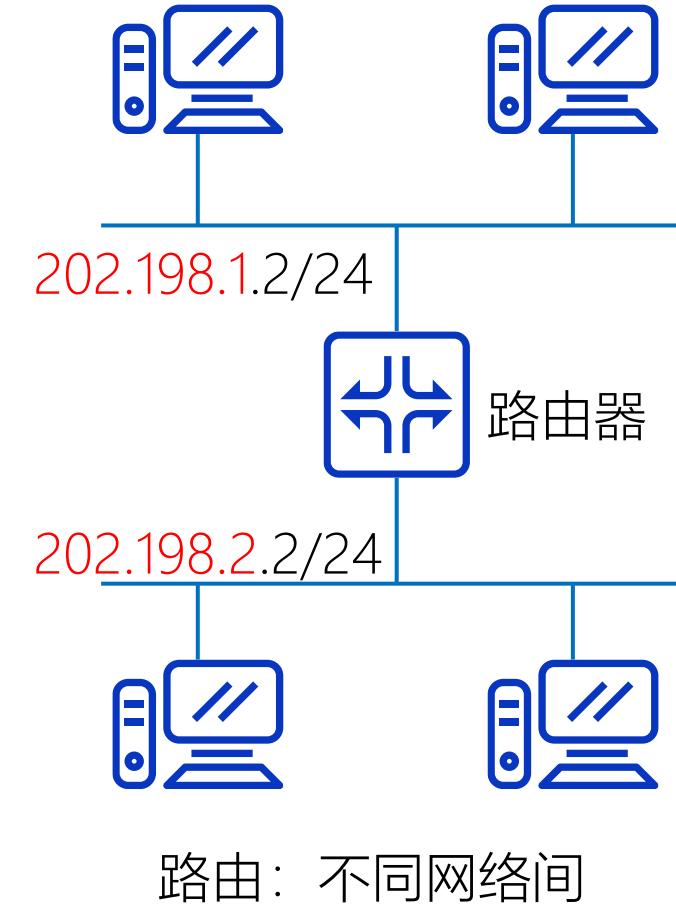
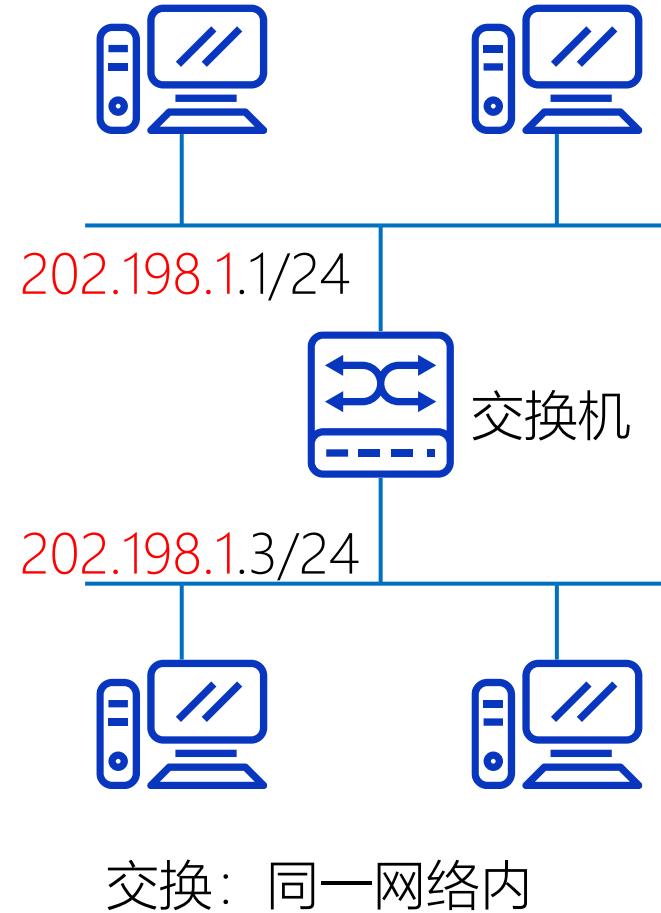


# 路由器

---

- 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

# 路由器与交换机的区别



# 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

# 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的所有层

# 路由选择原理

## ■ 路由选择

- 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
  - 费用最小
  - 传输延迟最小
  - 数据传输速率最大
  - 多因素的一种组合
- 两种最常用的计算最短路径的方法
  - 距离向量路由
  - 链路状态路由

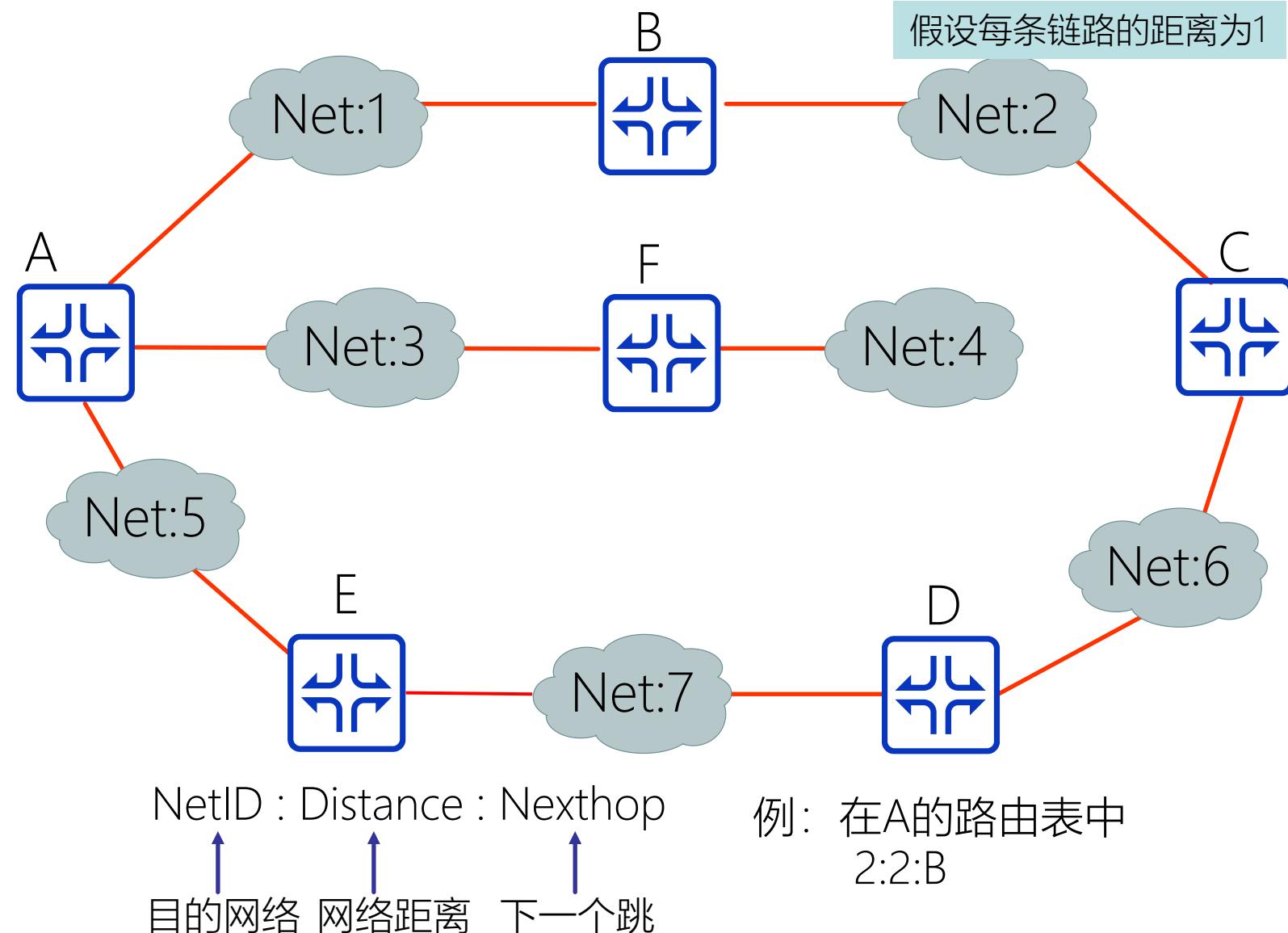
# 距离向量路由算法

- 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，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

- 路由器并不了解网络的拓扑
- 路由器只知道
  - 自己的直连网络
  - 自身与目的网络之间的距离
  - 应该往哪个方向使用哪个接口转发数据包



# 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: $\infty$ :?	3:1:-	4: $\infty$ :?	5:1:-	6: $\infty$ :?	7: $\infty$ :?
	B	1:1:-	2:1:-	3: $\infty$ :?	4: $\infty$ :?	5: $\infty$ :?	6: $\infty$ :?	7: $\infty$ :?
	C	1: $\infty$ :?	2:1:-	3: $\infty$ :?	4: $\infty$ :?	5: $\infty$ :?	6:1:-	7: $\infty$ :?
	D	1: $\infty$ :?	2: $\infty$ :?	3: $\infty$ :?	4: $\infty$ :?	5: $\infty$ :?	6:1:-	7:1:-
	E	1: $\infty$ :?	2: $\infty$ :?	3: $\infty$ :?	4: $\infty$ :?	5:1:-	6: $\infty$ :?	7:1:-
	F	1: $\infty$ :?	2: $\infty$ :?	3:1:-	4:1:-	5: $\infty$ :?	6: $\infty$ :?	7: $\infty$ :?
1	A	1:1:-	2:2:B	3:1:-	4:2:F	5:1:-	6: $\infty$ :?	7:2:E
	B	1:1:-	2:1:-	3:2:A	4: $\infty$ :?	5:2:A	6:2:C	7: $\infty$ :?
	C	1:2:B	1:1:-	3: $\infty$ :?	4: $\infty$ :?	5: $\infty$ :?	6:1:-	7:2:D
	D	1: $\infty$ :?	2:2:C	3: $\infty$ :?	4: $\infty$ :?	5:2:E	6:1:-	7:1:-
	E	1:2:A	2: $\infty$ :?	3:2:A	4: $\infty$ :?	5:1:-	6:2:D	7:1:-
	F	1:2:A	2: $\infty$ :?	3:1:-	4:1:-	5:2:A	6: $\infty$ :?	7: $\infty$ :?

# 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: $\infty$ :?	5:3:A	6:1:-	7:2:D
	D	1:3:C	2:2:C	3:3:E	4: $\infty$ :?	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: $\infty$ :?	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

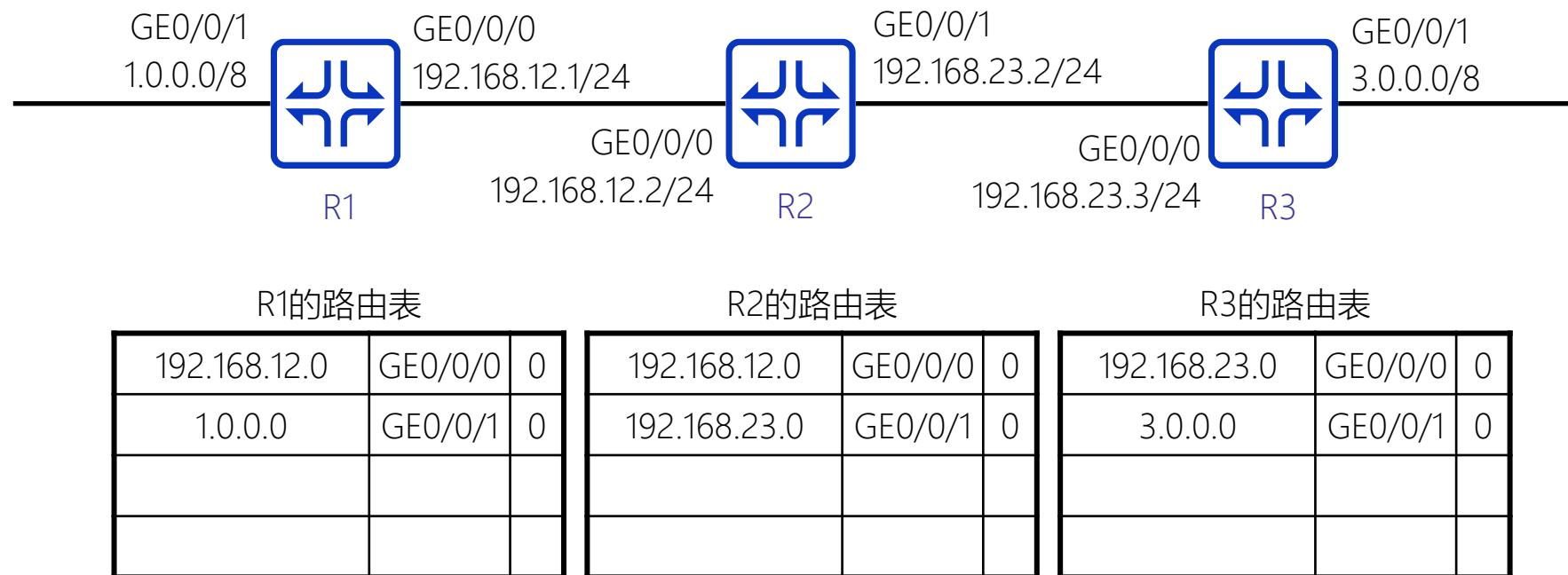
- 缺点：

- 网络规模的伸展性差
  - 对链路状态的变化响应慢
  - 路由报文尺寸大
  - 轮数与路由器的个数成正比

# 距离向量路由选择协议—以IP协议为例

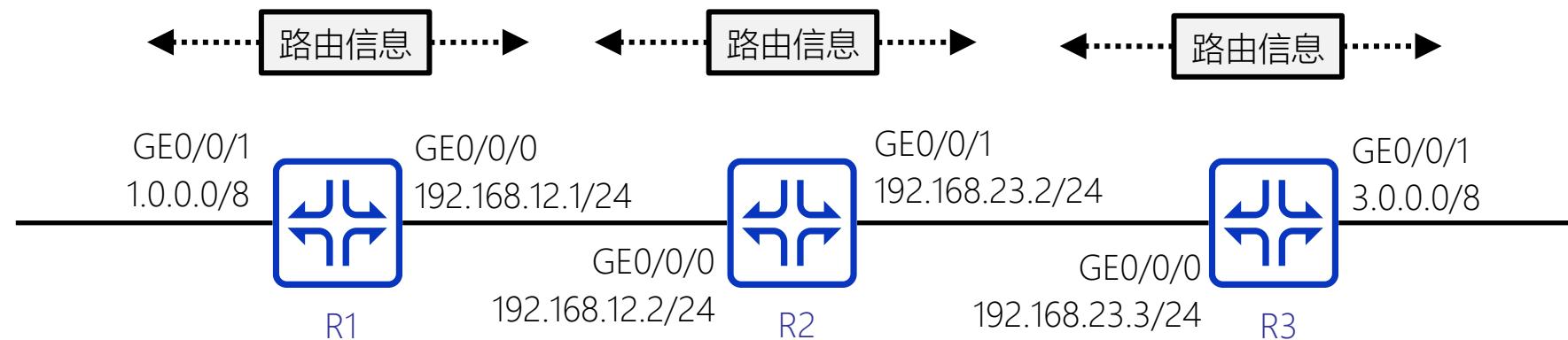
## ■ 路由器初始启动

- 最初的网络发现直连路由写入路由表



# 距离矢量路由选择协议—以IP协议为例

## ■ 初次路由信息交换



R1的路由表

192.168.12.0	GE0/0/0	0
1.0.0.0	GE0/0/1	0
192.168.23.0	GE0/0/0	1

R2的路由表

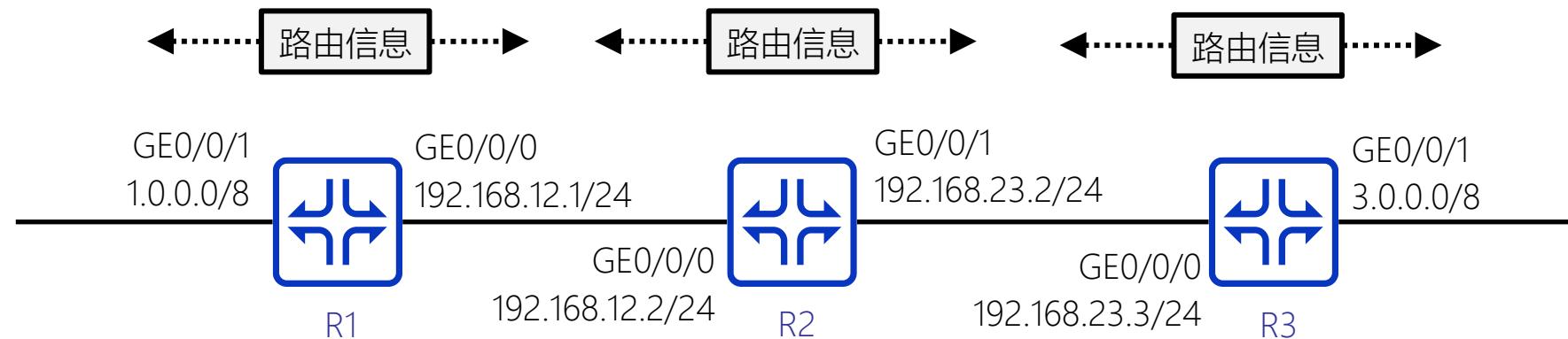
192.168.12.0	GE0/0/0	0
192.168.23.0	GE0/0/1	0
1.0.0.0	GE0/0/0	1
3.0.0.0	GE0/0/1	1

R3的路由表

192.168.23.0	GE0/0/0	0
3.0.0.0	GE0/0/1	0
192.168.12.0	GE0/0/0	1

# 距离矢量路由选择协议—以IP协议为例

## ■ 路由收敛完成



R1的路由表

192.168.12.0	GE0/0/0	0
1.0.0.0	GE0/0/1	0
192.168.23.0	GE0/0/0	1
3.0.0.0	GE0/0/0	2

R2的路由表

192.168.12.0	GE0/0/0	0
192.168.23.0	GE0/0/1	0
1.0.0.0	GE0/0/0	1
3.0.0.0	GE0/0/1	1

R3的路由表

192.168.23.0	GE0/0/0	0
3.0.0.0	GE0/0/1	0
192.168.12.0	GE0/0/0	1

# 链路状态路由算法

- 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：
  - 每个链路状态描述路由器本身能直接可达（邻居）的路由器/路由
  - 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)
  - 所有的链路状态组成链路状态数据库
- 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络的路由

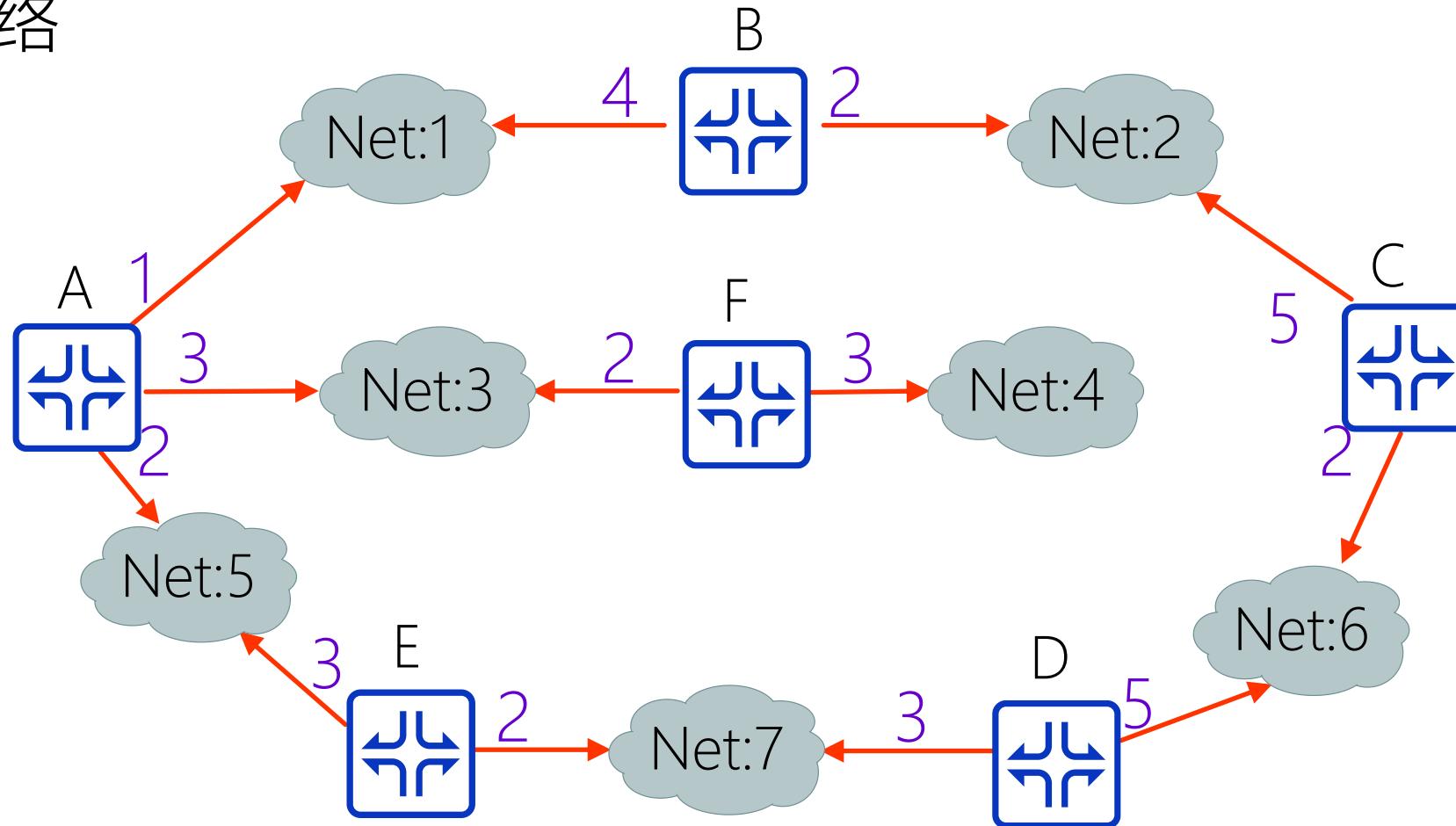
# 链路状态信息共享 (1)

---

- 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

# 链路状态信息共享 (2)

## ■ 示例网络



## 链路状态信息共享(3)

- 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(4)

- 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(5)

- 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	-

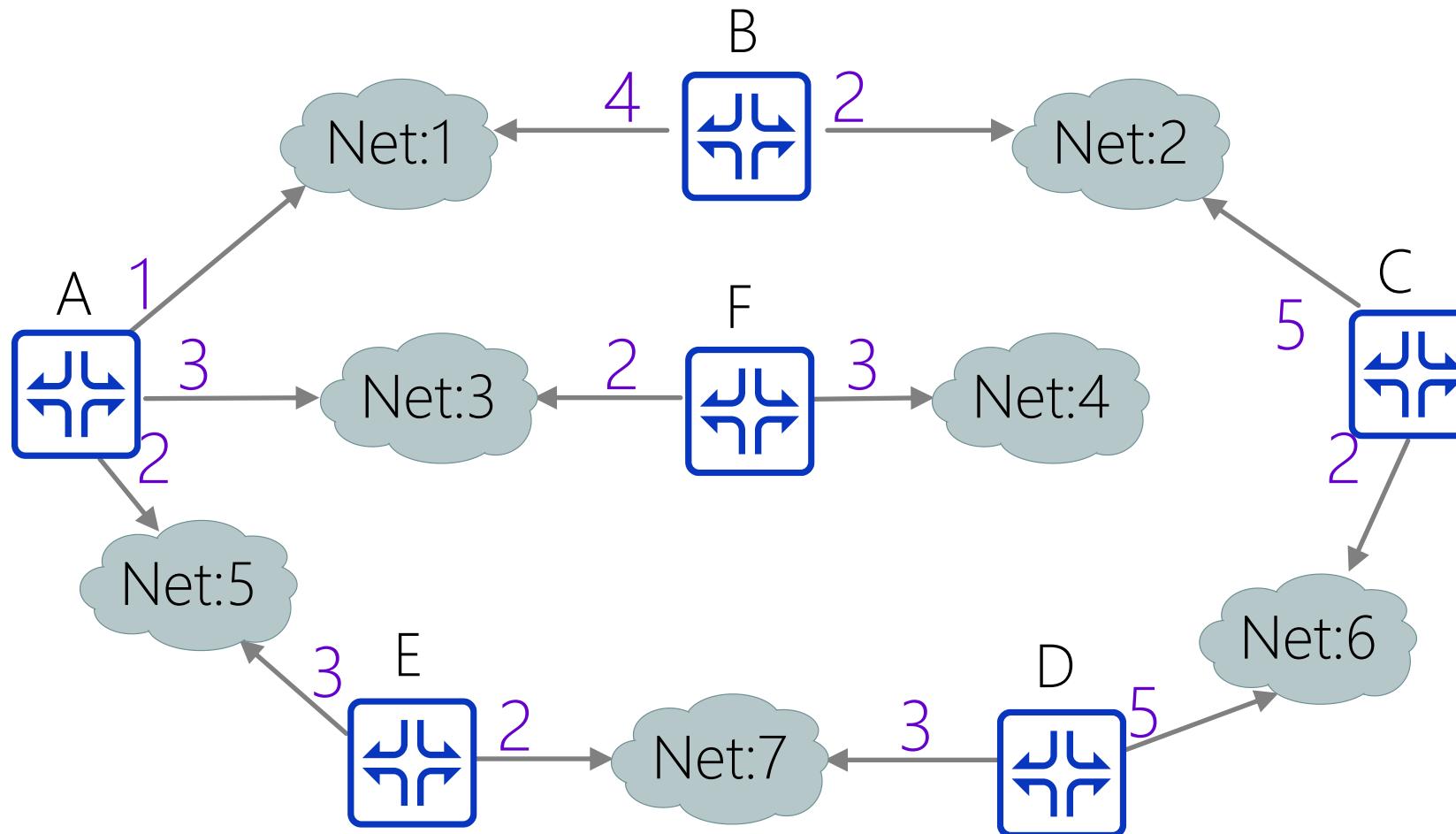
# 迪科斯彻(Dijkstra)算法

- 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径树

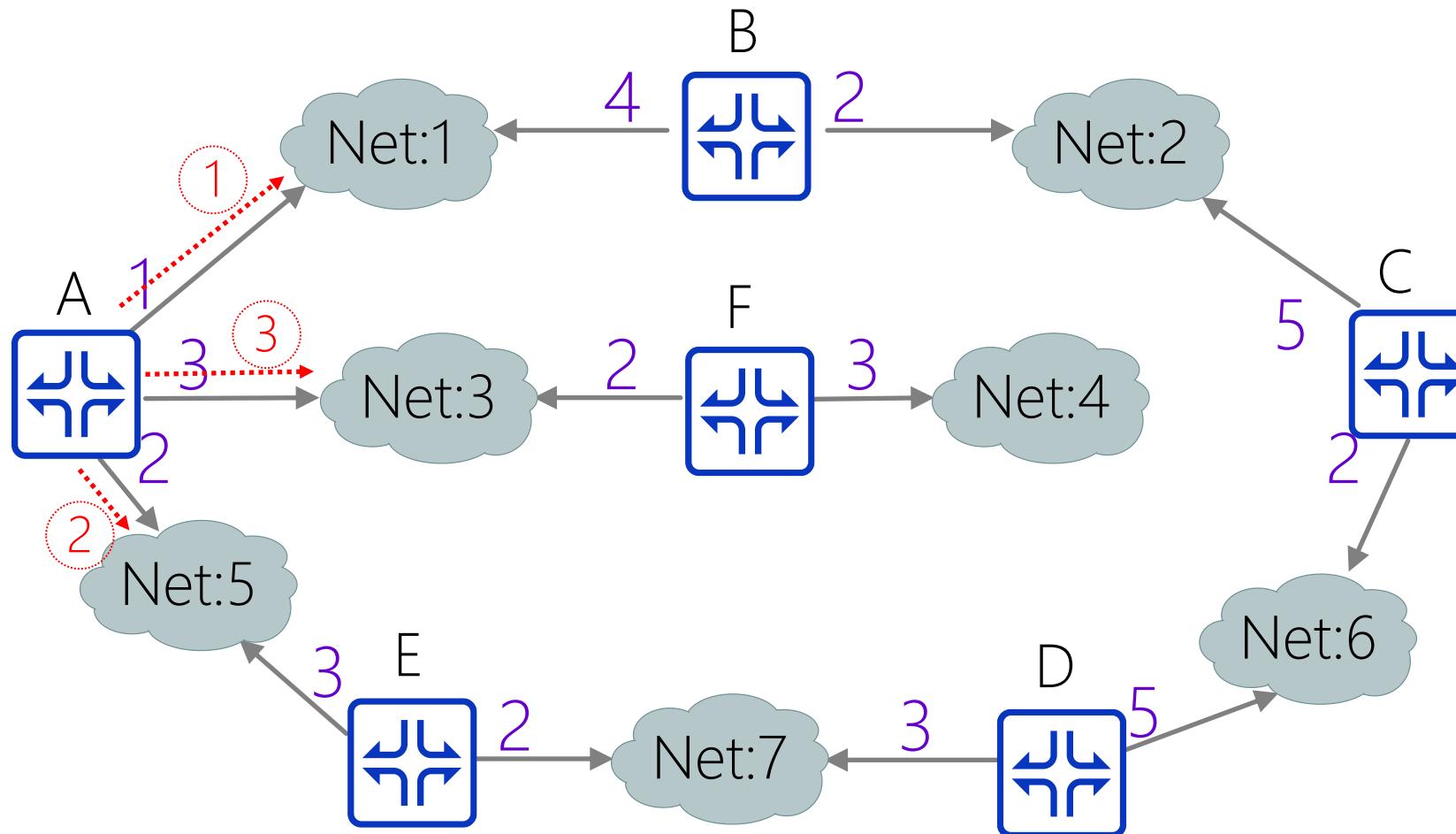
# Dijkstra算法的一般步骤

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，这个路径上的所有弧和节点被标记为最短路径树上的永久部分
3. 算法考察链路状态数据库，找出从这个选定的最短路径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径树上
4. 如果所有的节点已经成为最短路径树上的永久部分，则算法结束，去掉非永久性的弧。否则，转步骤2继续执行

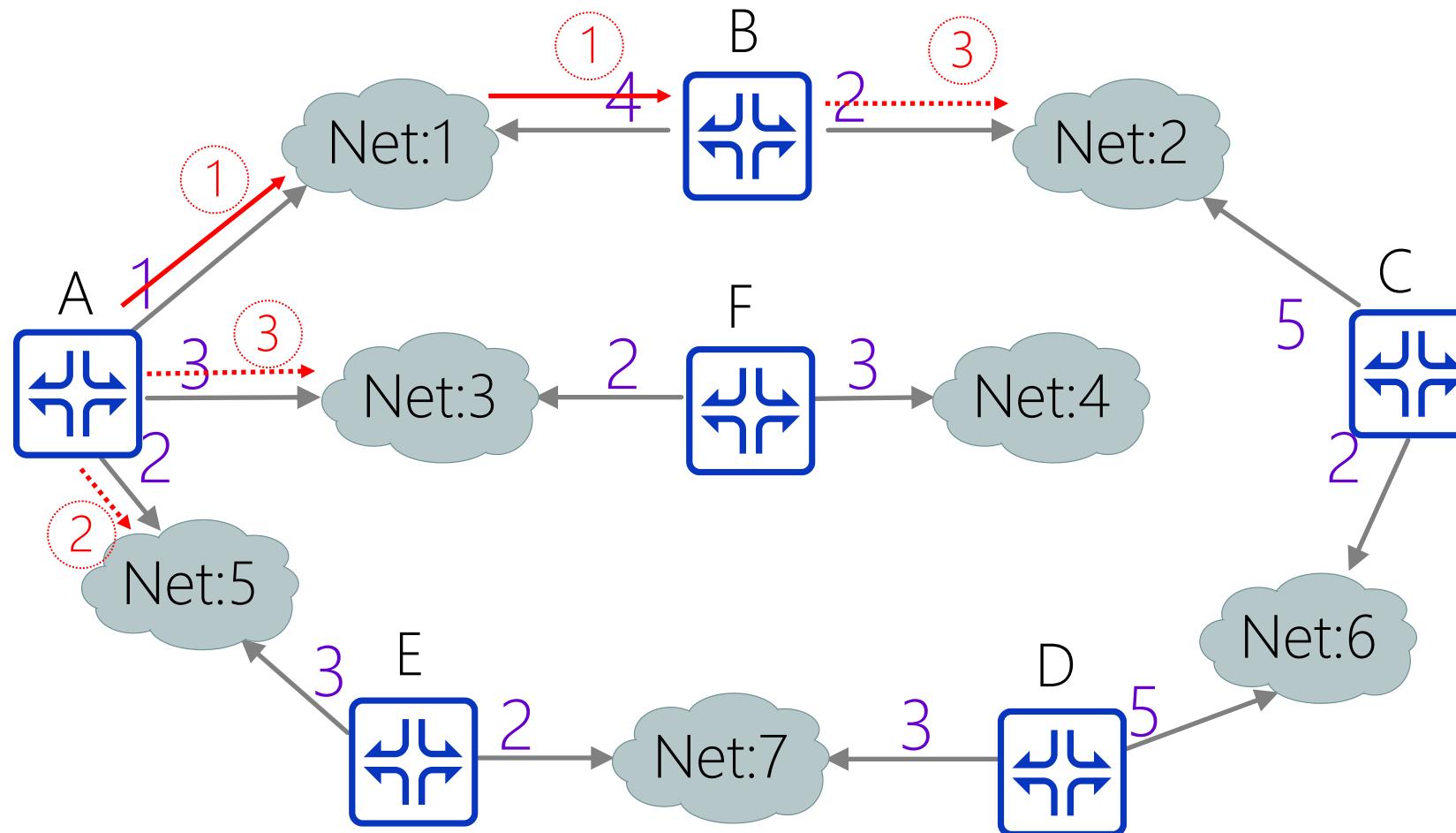
# Dijkstra算法示例



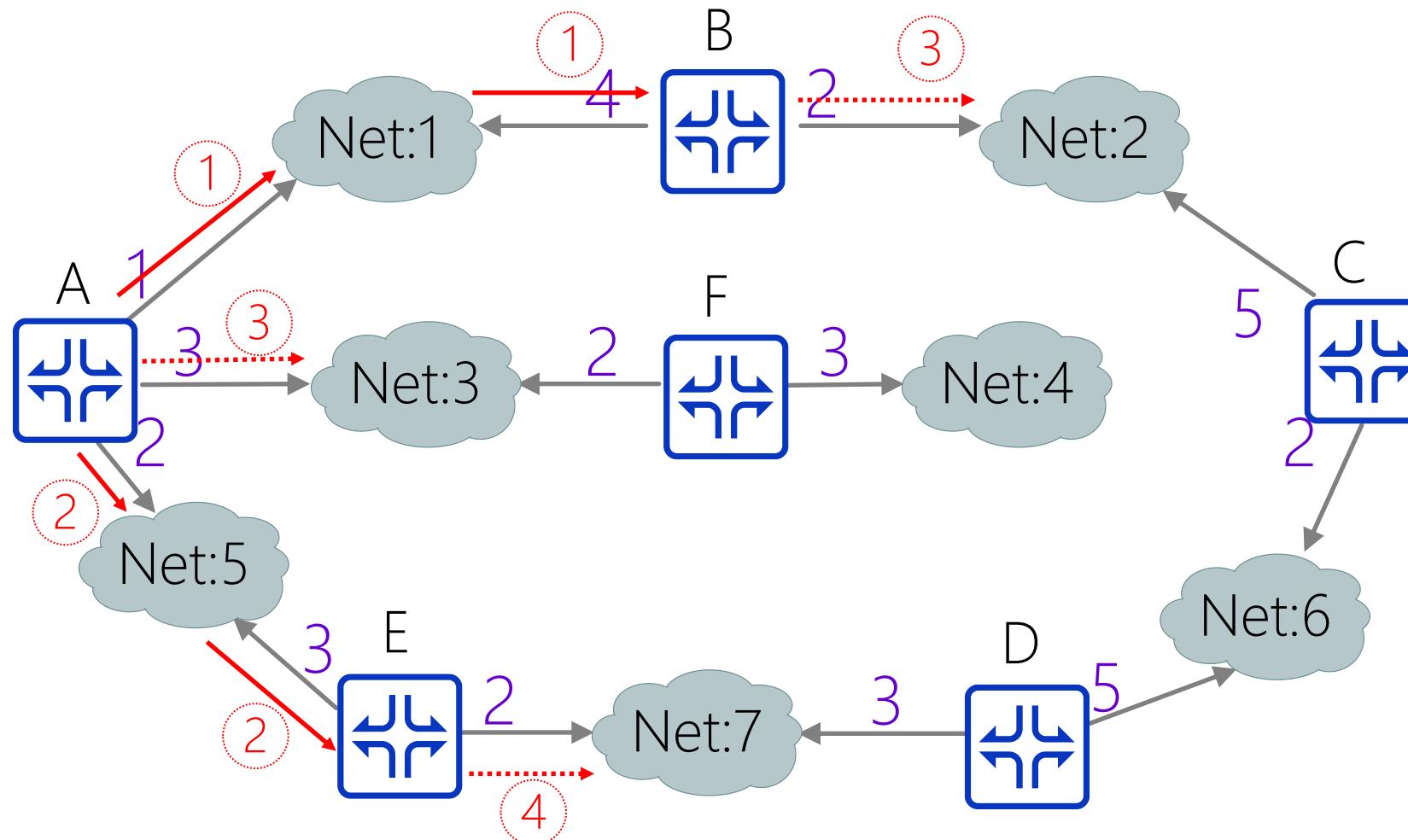
# Dijkstra算法示例



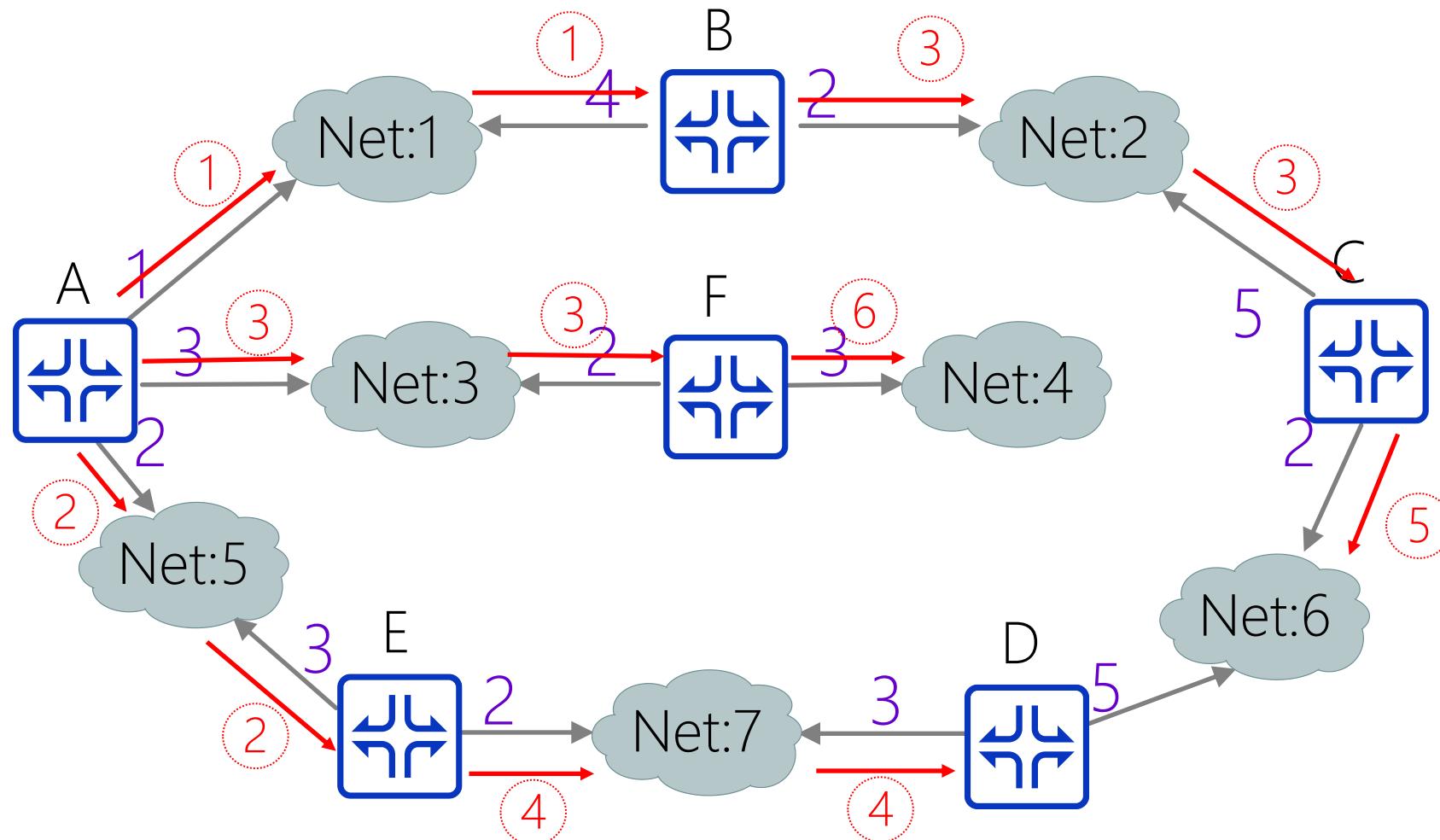
# Dijkstra算法示例



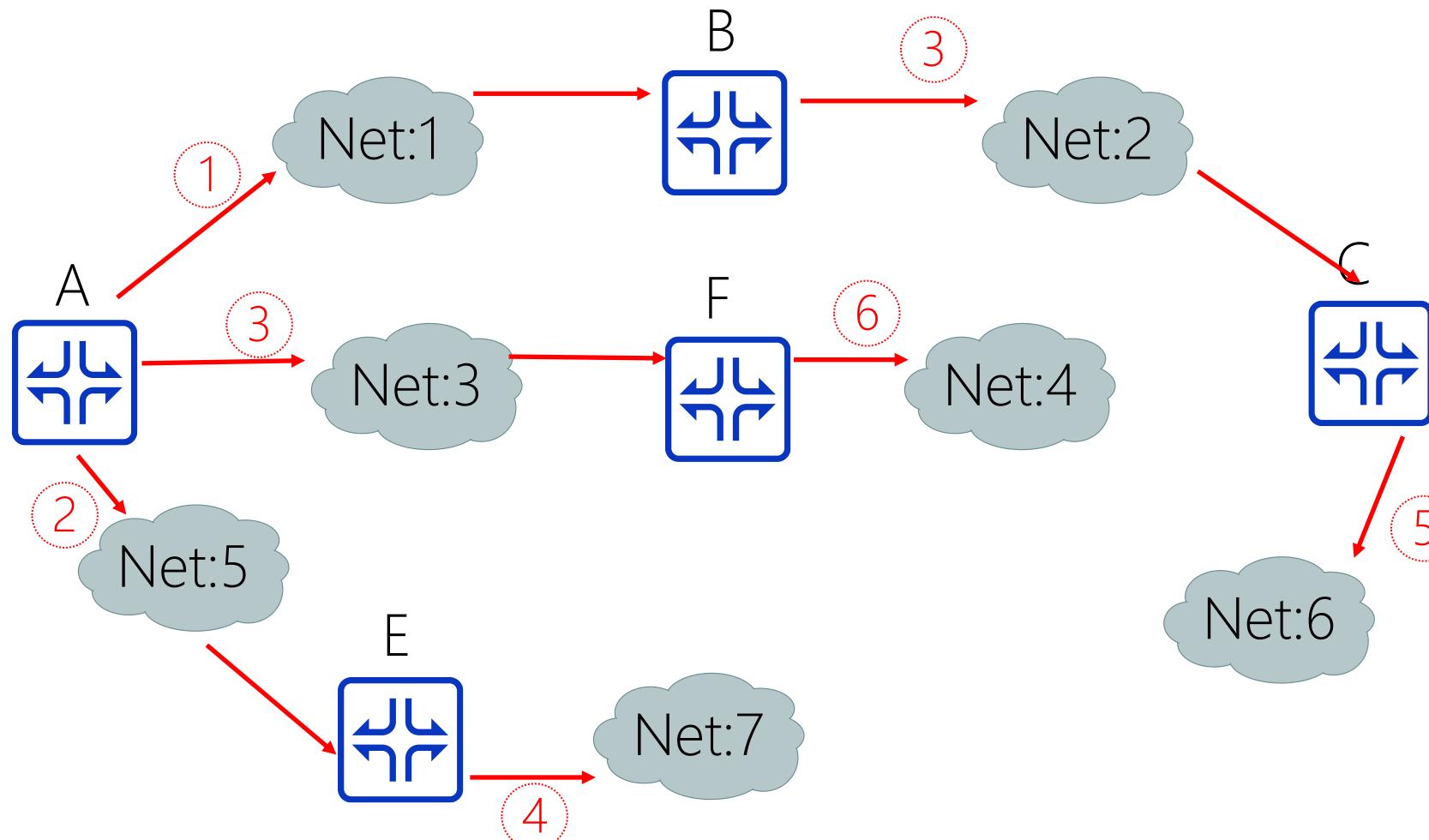
# Dijkstra算法示例



# Dijkstra算法示例



# Dijkstra 算法示例



# 算法比较

## 距离向量路由算法

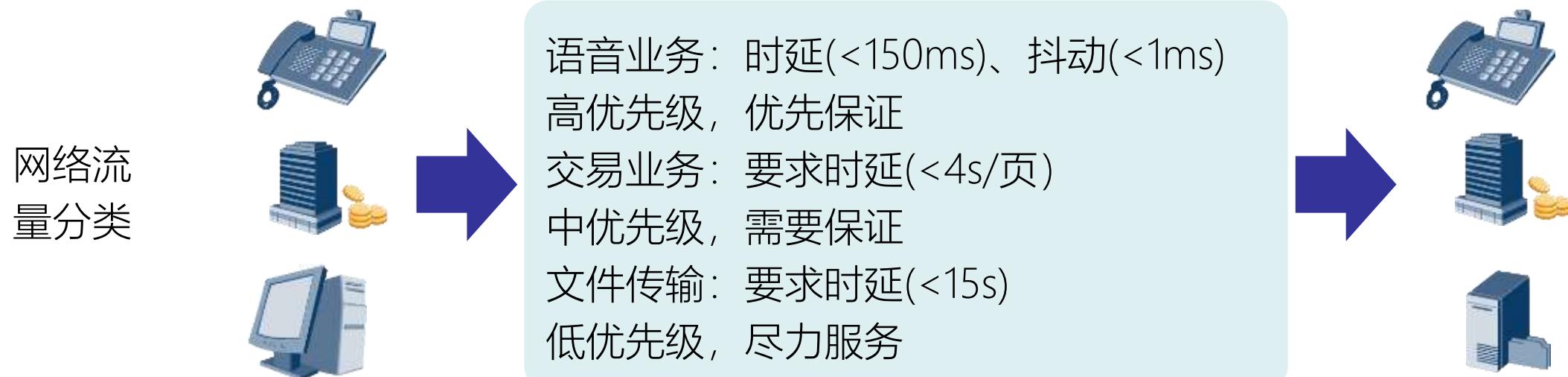
- 每个路由器周期性的将自己关于整个网络的信息发送给它的邻居
  - 每个路由器保存关于整个网络的信息
  - 仅仅和邻居交换网络信息
  - 信息的交换是通过有规律的时间间隔来进行(例如每隔30秒发一次),无论网络状态是否发生变化

## 链路状态路由算法

- 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
  - 每个链路状态描述路由器本身能直接可达(邻居)的路由器/路由
  - 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)
  - 所有的链路状态组成链路状态数据库

# 拥塞控制和服务质量

- 为什么需要QoS
- QoS: Quality of Service, 服务质量
  - 针对各种业务的不同需求，为其提供端到端的服务质量保证
- QoS技术在当今的互联网中应用越来越多，其作用越来越重要
- 无QoS技术，业务的服务质量就无法保证



## 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
- 拥塞控制是一个全局性的过程，涉及到所有的主机、所有的路由器，以及与降低网络传输性能有关的所有因素

## 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
- 流量控制往往指在给定的发送端和接收端之间的点对点通信量的控制。流量控制所要做的就是抑制发送端发送数据的速率，以便接收端来得及接收

# 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

---

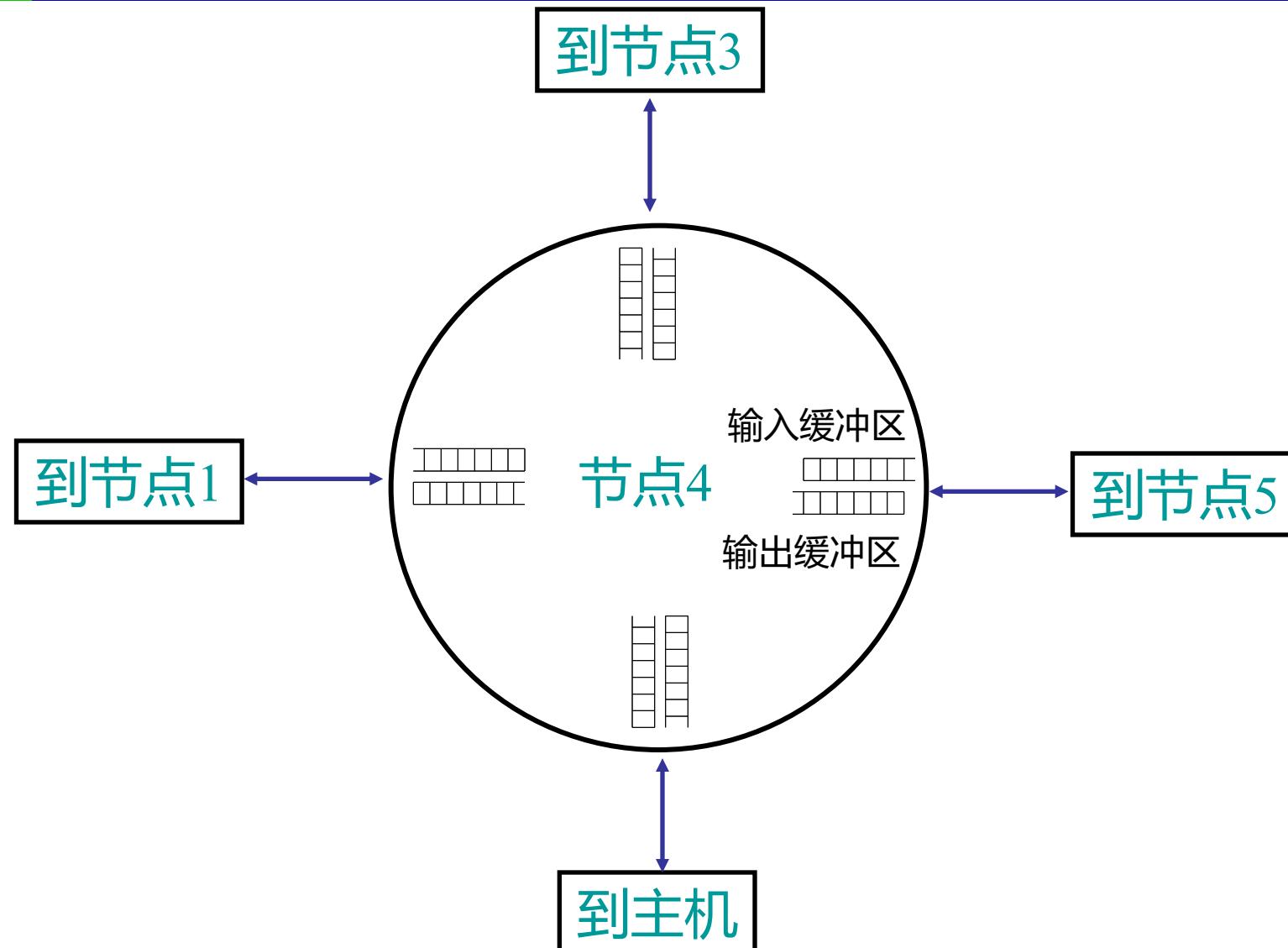
- 通过良好的设计，避免问题的出现，确保问题在一开始就不会出现，不需要中途做修正
- 开环控制的方法：
  - 什么时候接受新的数据流
  - 什么时候开始丢弃数据报，丢弃哪些数据报？
  - 指定网络中各个节点的调度策略
- 这些方法的共同特点：做出决定的时候不考虑网络的当前状态

# 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



# 拥塞对系统的影响

吞吐量

1.0

通信子网的最大承载容量

包平均延迟

1.0

输入负载

1.0

输入负载

理想情况

无拥塞控制

有拥塞控制

# 控制拥塞的方法

- 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

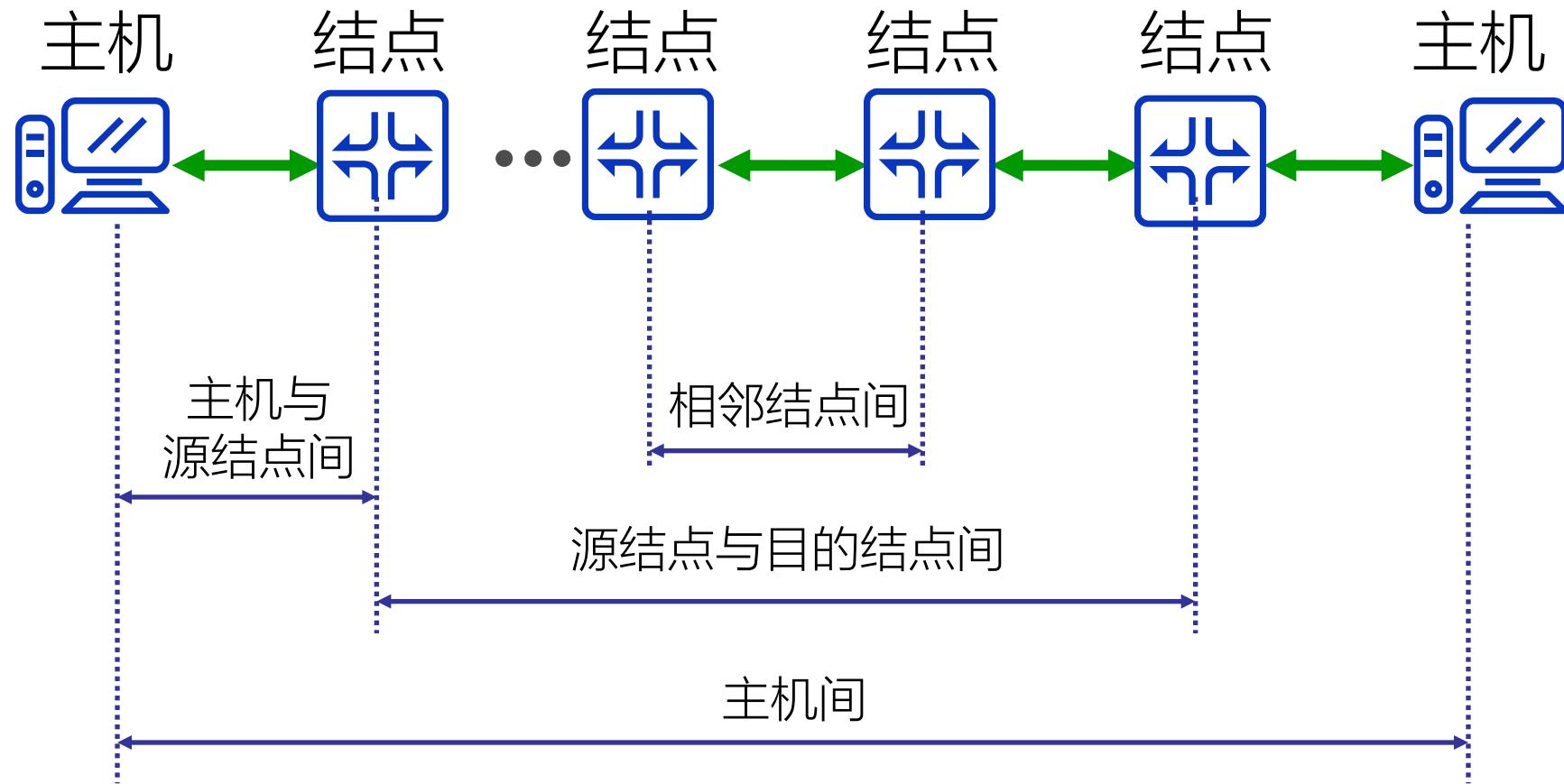
- 流量控制：接收端调节发送端发送数据的速率，防止到达接收端的数据速率超过接收端的处理速率。本质上流量控制和拥塞控制是不同的概念：
  - 流量控制是端到端
  - 拥塞控制涉及中间节点
- 阻塞包：每个节点都监视其所有输出链路的使用情况。视情况决定是否向源结点发送阻塞包

# 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

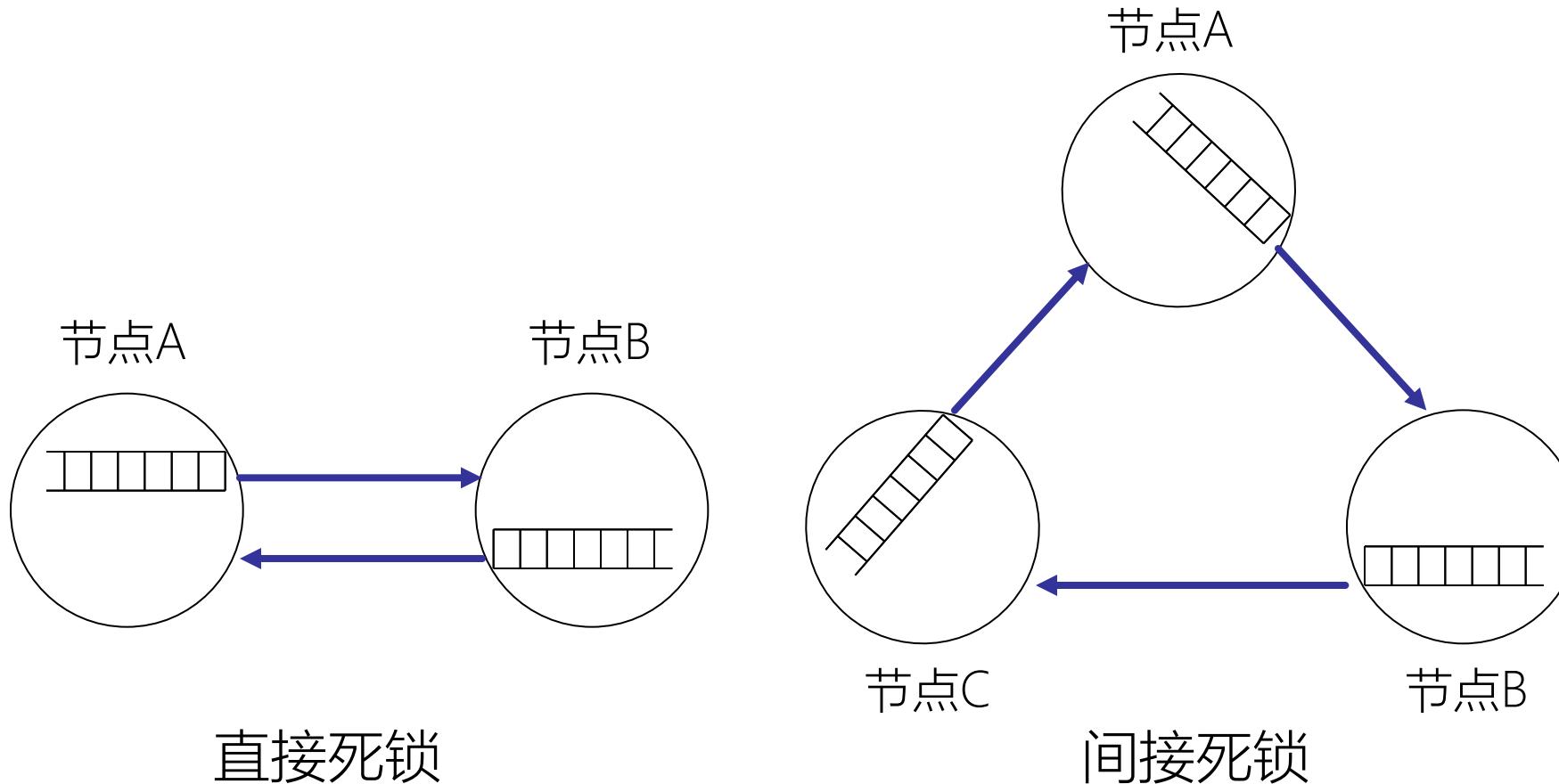
---

- 通过控制进入通信子网的信息量，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

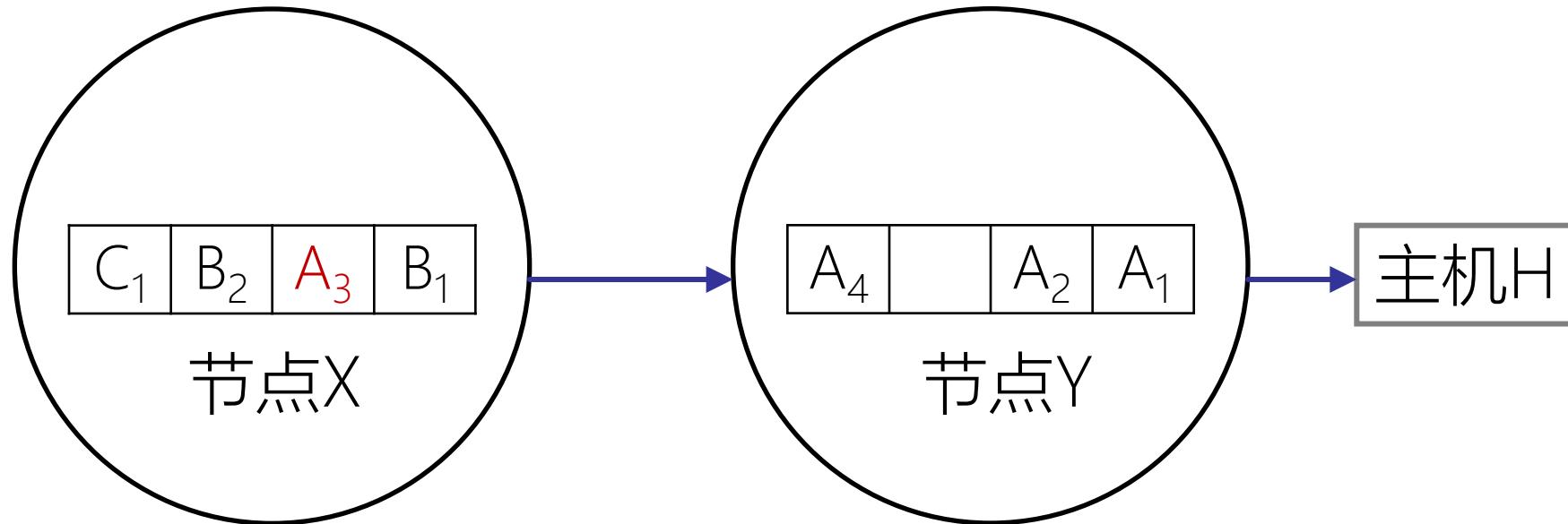
- 其任务是和通信子网的工作方式紧密相关的
- 如果通信子网采用虚电路工作方式，该层流量控制的任务就比较轻。因为虚电路方式本身要求有基本的缓冲区，包沿固定路径传送，且包按顺序到达目的节点
- 如果通信子网采用数据报方式工作，而缓冲区分配采用先来先服务且全部分配的方法，则有可能产生存储转发死锁

# 存储转发死锁



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

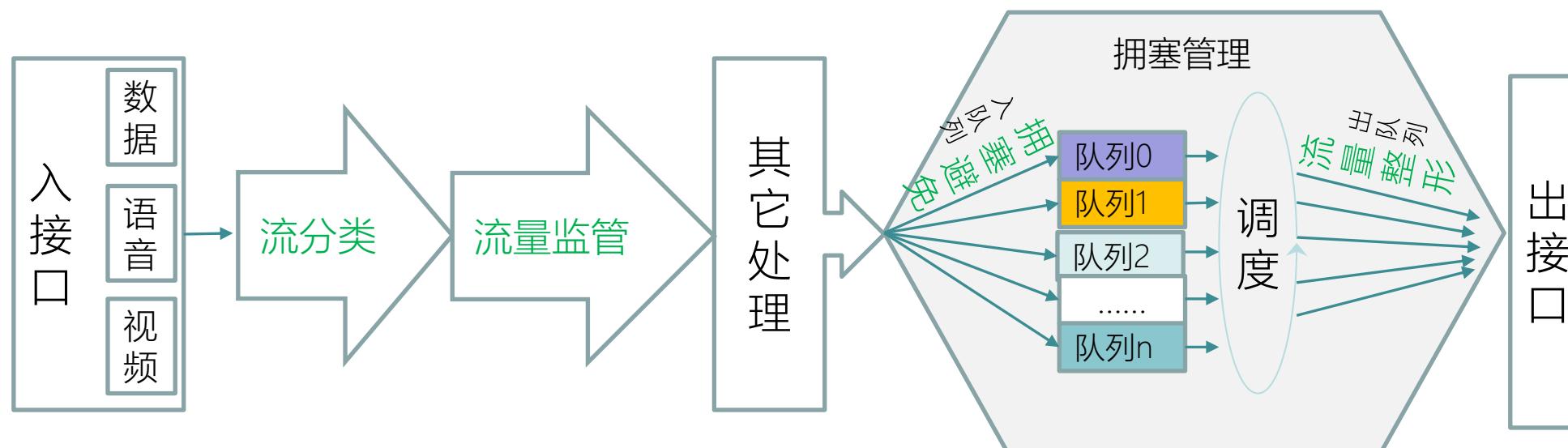
# 服务质量

---

- 为关键业务提供服务质量保证，使其获得可预期的服务水平
- QoS度量指标
  - 带宽
  - 时延
  - 时延变化（抖动）
  - 丢包率

# 服务质量的保证方法

- 过度配置：建设一个有足够的容量的网络
- 流量整形：调节网络数据流平均速率和突发性速率
  - SLA服务等级约定（流分类）
  - 流量监管（漏桶算法、令牌桶算法）
- 数据包调度：在流的数据包之间或竞争流之间分配路由器资源

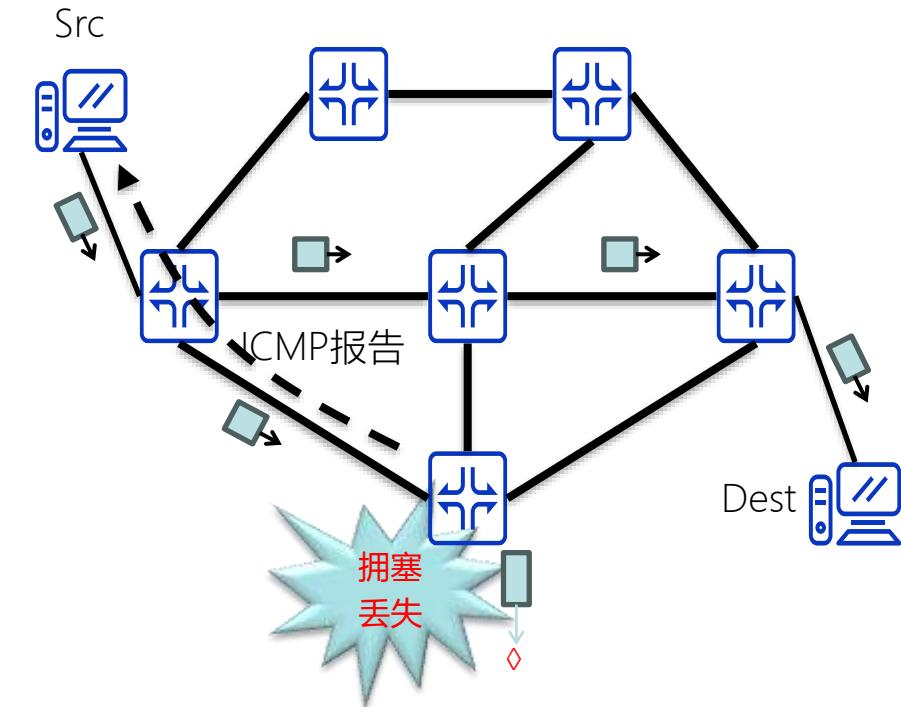


# QoS服务模型

- 网络应用都是端到端的通信，两个主机进行通信，中间可能要跨越多个物理网络，经过多个路由器，要实现端到端的QoS，就必须从全局考虑
- QoS模型是端到端的QoS设计方案，确定如何在网络中通过部署来保证QoS的度量指标在一定的合理范围内，提高网络的服务质量
- 网络中的两个主机通信时，中间可能会跨越各种各样的设备。只有当网络中所有设备都遵循统一的QoS服务模型时，才能实现端到端的质量保证
- 三种服务模型：
  - 尽力而为 (Best-Effort) 服务模型
  - 综合服务 (Integrated Service, 简称IntServ) 模型
  - 差分服务 (Differentiated Service, 简称DiffServ) 模型

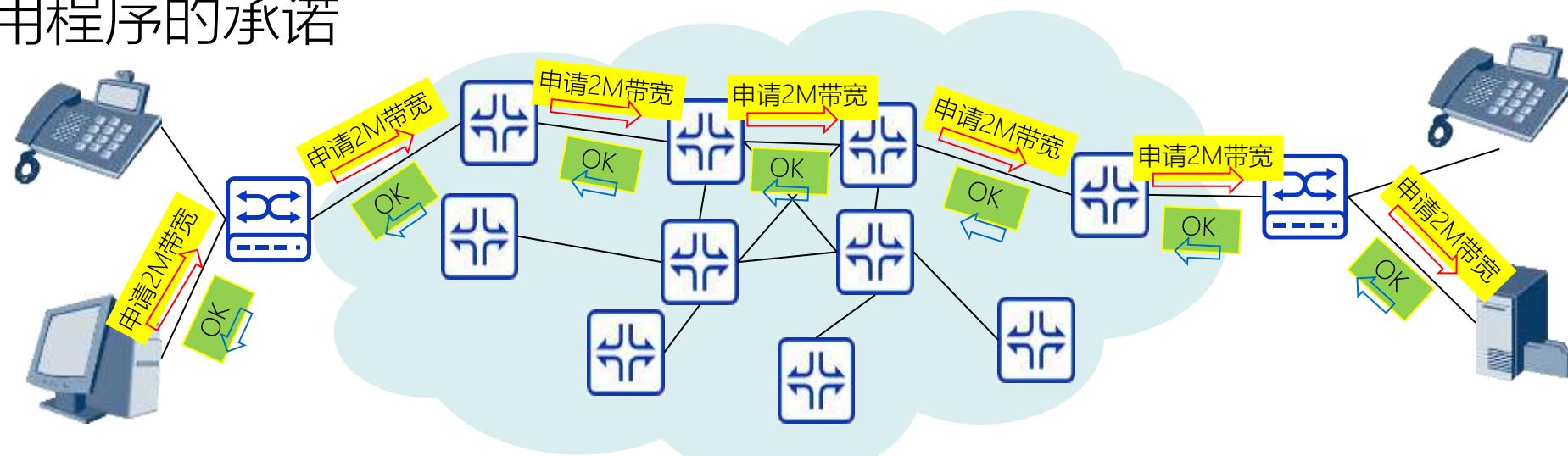
# Best-Effort服务模型

- Best-Effort是Internet的缺省服务模型
- Best-Effort是最简单的QoS服务模型：
  - 应用程序可以在任何时候，发出任意数量的报文，而且不需要通知网络
  - 网络尽最大的可能性来发送报文，但对时延、可靠性等性能不提供任何保证
- Best-Effort服务模型适用于对时延、可靠性等性能要求不高的业务，适用于绝大多数网络应用，如FTP、E-Mail等



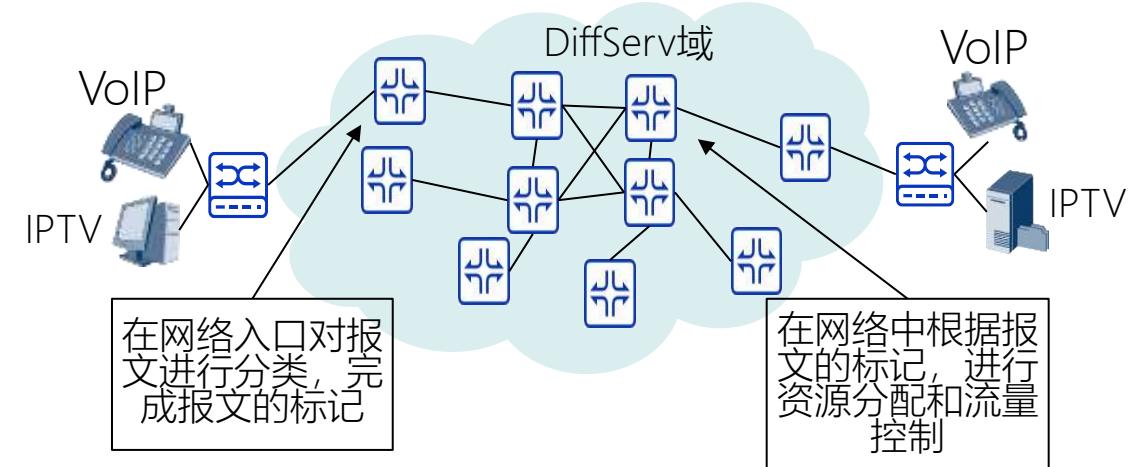
# IntServ模型

- 应用程序发送报文前，需要通过信令（signaling）向网络描述它的流量参数，申请特定QoS服务
- 网络在流量参数描述的范围内，预留资源以承诺满足该请求
- 在收到确认信息，确定网络已经为这个应用程序的报文预留了资源后，应用程序才开始发送报文，并将发送的报文控制在流量参数描述的范围内
- 网络节点为每个流维护一个状态，并基于这个状态执行相应的QoS动作，来满足对应用程序的承诺



# DiffServ模型

- DiffServ模型是当前网络中的主流QoS服务模型：
  - 将网络中的流量分成多个类，每个类享受不同的处理，尤其是网络出现拥塞时不同的类会享受不同的优先处理
  - 同一类的业务在网络中会被聚合起来统一发送，保证相同的延迟、抖动、丢包率等QoS指标
- DiffServ是一种基于报文流的QoS模型
  - 业务流分类和汇聚工作在网络边缘由边缘路由器完成
  - 边界路由器可以通过多种条件（比如报文的源地址和目的地址、ToS域中的优先级、协议类型等）灵活地对报文进行分类，对不同的报文设置不同的标记字段
  - 其他路由器只需要简单地识别报文中的这些标记，进行资源分配和流量控制
- DiffServ模型充分考虑了IP网络本身灵活性、可扩展性强的特点，将复杂的服务质量保证通过报文自身携带的信息转换为单跳行为，从而大大减少了信令的工作
- DiffServ模型不但适合运营商环境使用，而且也大大加快了IP QoS在实际网络中应用的进程



## 5.2 IPv4协议

- 在计算机网络领域，网络层应该向传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
  - 网络在发送数据包时不需要先建立连接。每一个 IP 数据包独立发送，与其前后的数据包无关（不进行编号）
  - 网络层不提供服务质量的承诺。即所传送的IP 数据包可能出错、丢失、重复和失序（不按序到达终点），当然也不保证IP 数据包传送的时限

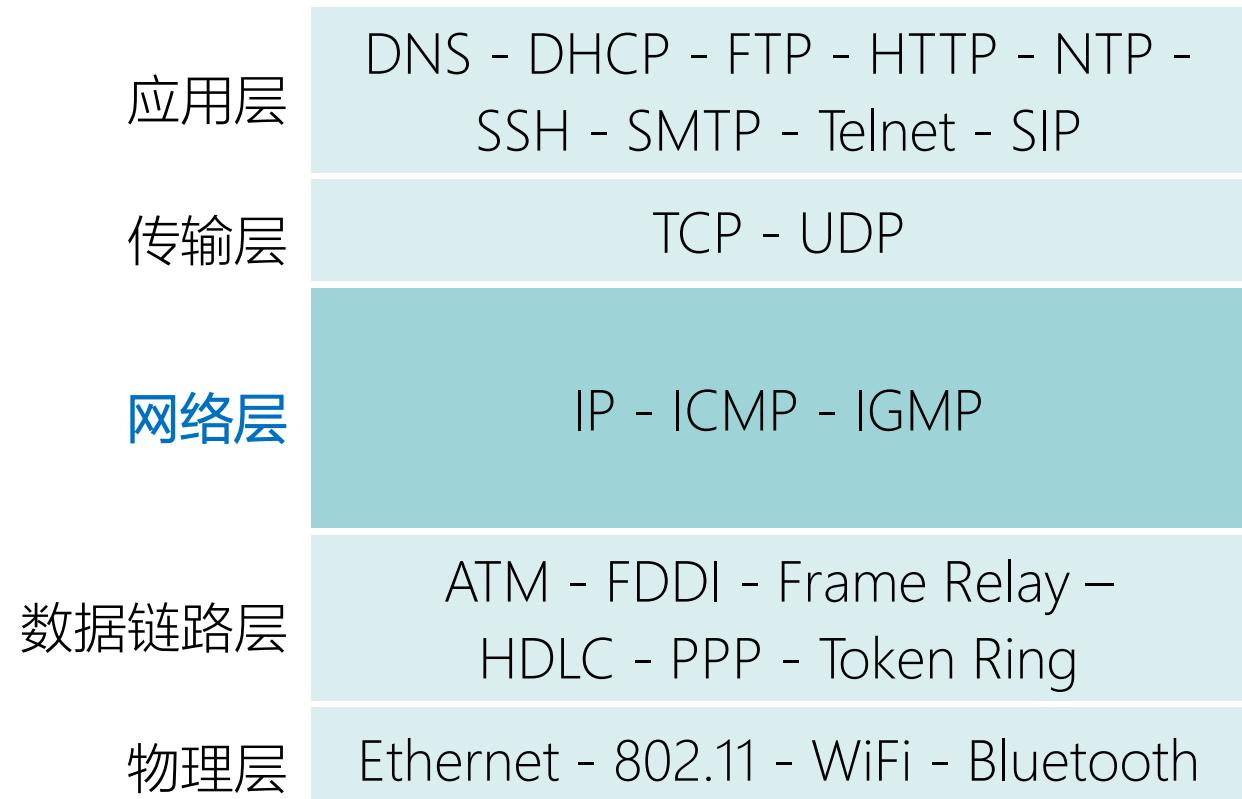
# 尽最大努力交付的好处

---

- 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传输层负责（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

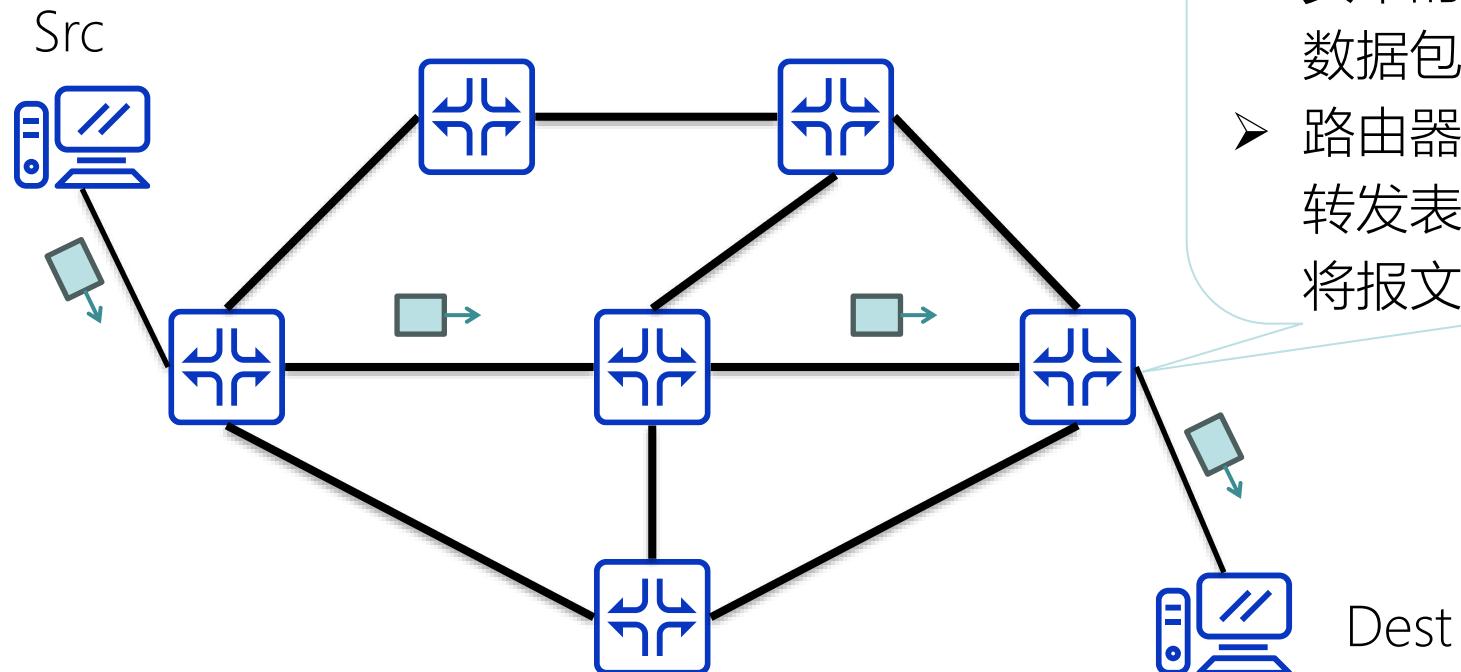
# 网络层协议概览

- IP协议是TCP/IP协议族中的核心协议
- IP协议为高层提供不可靠、无连接的数据报通信
- 所有的TCP、UDP、ICMP、IGMP数据都是以IP数据报格式传输
- 与IP协议配套使用的还有三个协议：
  - 地址解析协议(Address Resolution Protocol – ARP)
  - Internet控制消息协议(Internet Control Message Protocol – ICMP)
  - Internet组管理协议(Internet Group Management Protocol – IGMP)



# IP地址

## ■ 网络层数据平面如何识别一台主机？



- 路由器通过**网络层报文头**中的目的IP地址确定数据包的目的地
- 路由器使用目的IP查找转发表确定从哪个接口将报文转发出去

# IP地址

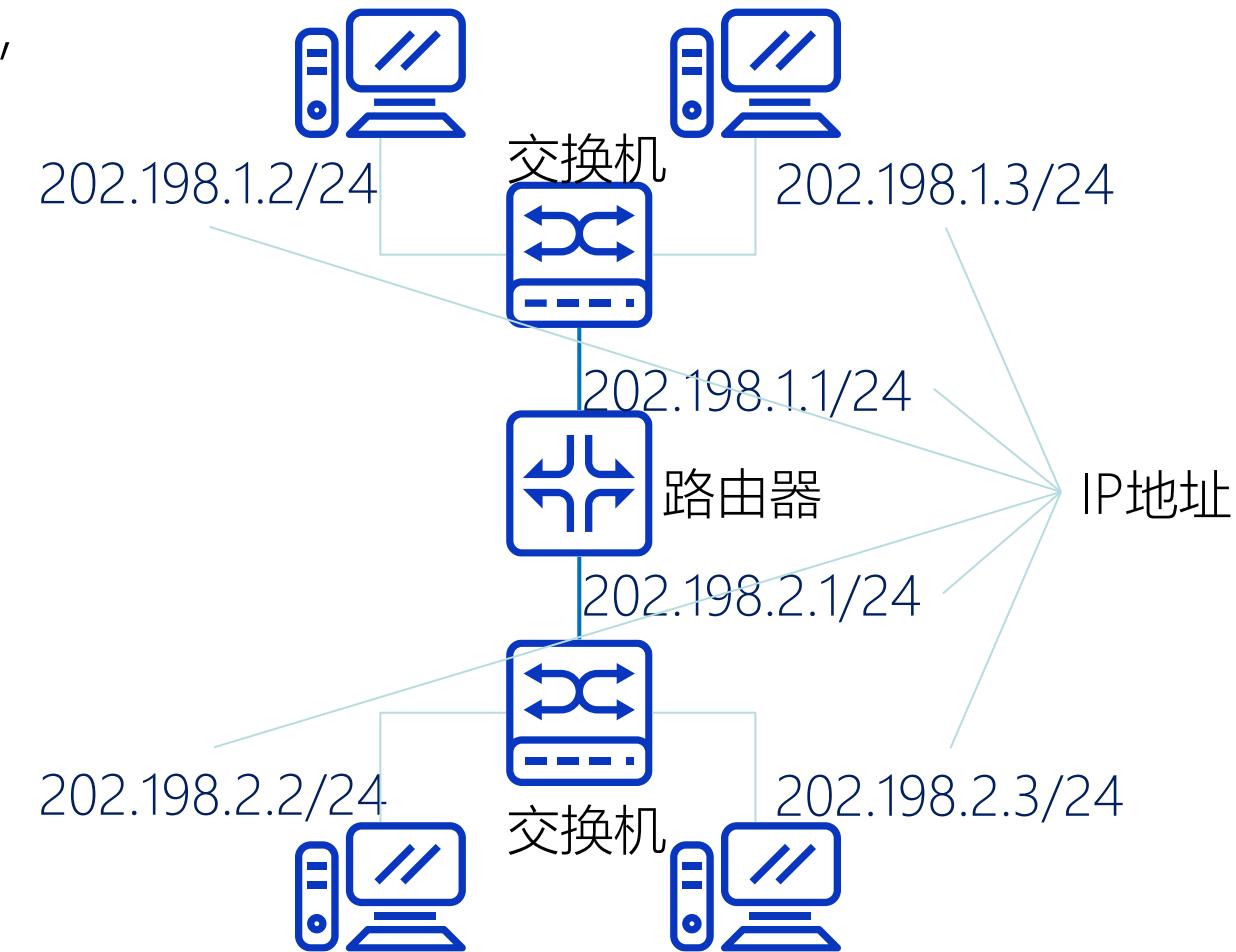
■ 网络中每个独立主机/路由器的每个接口必须有一个唯一的Internet 地址，即IP地址

- 通常路由器有多个接口，高端/核心路由器多至几十个接口
- 主机通常有1个或2个接口（如以太网接口和WiFi）

■ IP地址长度为32位（4个字节）。表示地址空间是 $2^{32}$ （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



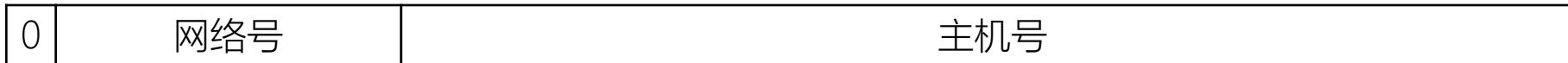
# IP地址的分类

- IP地址按照层次结构划分成五类：A、B、C、D、E类

A类

7b

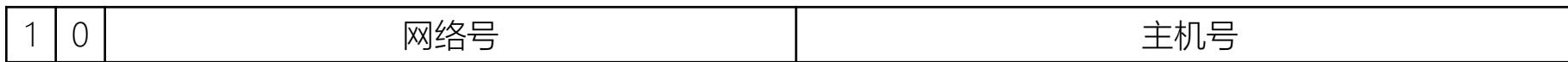
24b



B类

14b

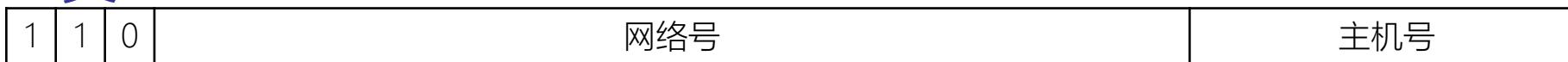
16b



C类

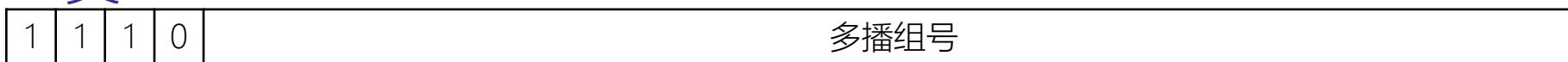
21b

8b



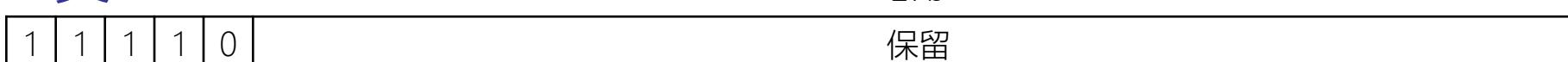
D类

28b



E类

27b



# IP地址的分类

- IP地址采用两级结构，分为网络标识和主机标识



- IP地址被划分为若干个固定分类
- 每类IP地址由两个固定长度的字段组成：
  - 网络号 Network：标识主机（或路由器）所连接到的网络
  - 主机号 Host：在所连接到的网络中唯一标识该主机（或路由器）
- IP 地址在整个互联网范围内是唯一的

# IP地址的表示方法

## ■ 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，IP地址通常写成用小数点把各字节分隔开的形式。每个字节用一个小于256的十进制数表示

## ■ 二进制表示方法

- 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

## ■ 十六进制表示方法

- 每一个十六进制数字等效于4 bits

	第1字节	第2字节	第3字节	第4字节
10 进制	129	.	16	.
2 进制	10000001	00001110	00000110	00011111
16 进制	81	.	0E	.
网络号				主机号

# 各类IP地址的范围

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

# 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
  - 受限广播地址
- 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

- 0.0.0.0不是一个真正意义上的IP地址
- 表示一个集合：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
- 受限广播地址：32位为全1的IP地址称为受限广播地址
  - 例：受限广播地址为：255.255.255.255
- 两者的区别：
  - 受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

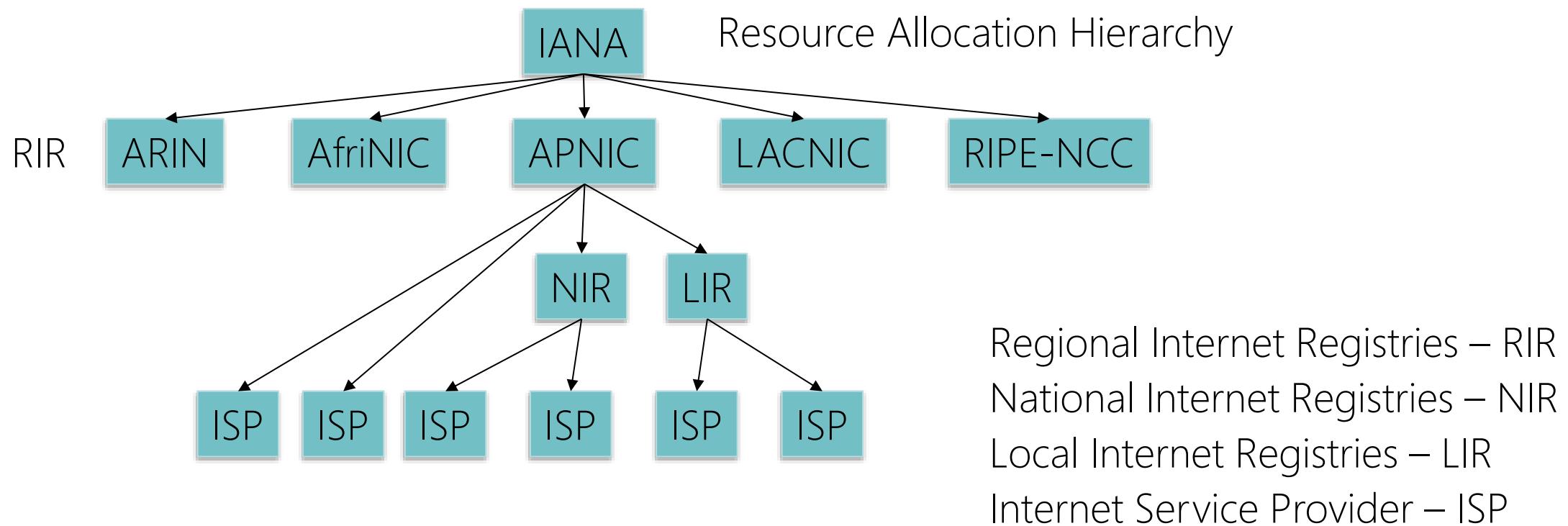
- 环回地址：第一个字节等于127的IP地址称为环回地址，用作主机或路由器的环回接口
  - 大多数主机系统把127.0.0.1分配给环回接口，常用于本机上软件测试和本机上网络应用程序之间的通信地址

# 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地址可以设置成私有IP地址，进行企业内部的网络应用
- 通过NAT服务器访问Internet
  - 只需要申请少量的全局IP地址
  - 解决了IP地址不足的问题
  - 解决了网络安全问题

# IP网络地址如何分配

- IP 地址由互联网名字和数字分配机构ICANN (Internet Corporation for Assigned Names and Numbers)进行分配



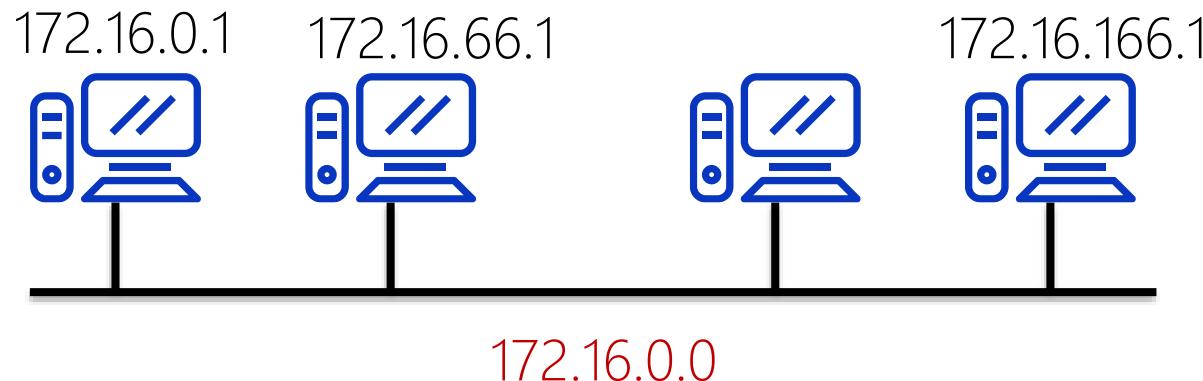
# 早期IP地址分类设计缺陷

- 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
- 提出子网编址、 CIDR等技术

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

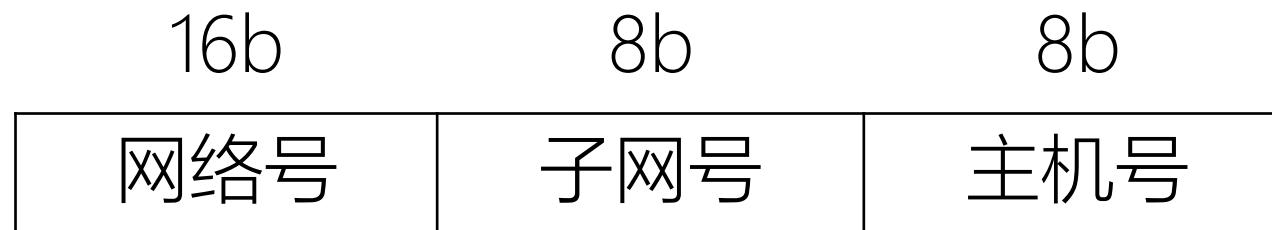
# 为什么要划分子网

- “有类编址”的地址划分过于死板，划分的粒度太大，会有大量的主机号不能被充分利用，从而造成了大量的IP地址资源浪费
- 一个B类地址用于一个广播域，地址浪费
- 广播域太庞大，一旦发生广播网络不堪重负



# 子网编址

- 划分子网：为解决IP地址分类编址方案的不足，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
- 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 数据报的头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

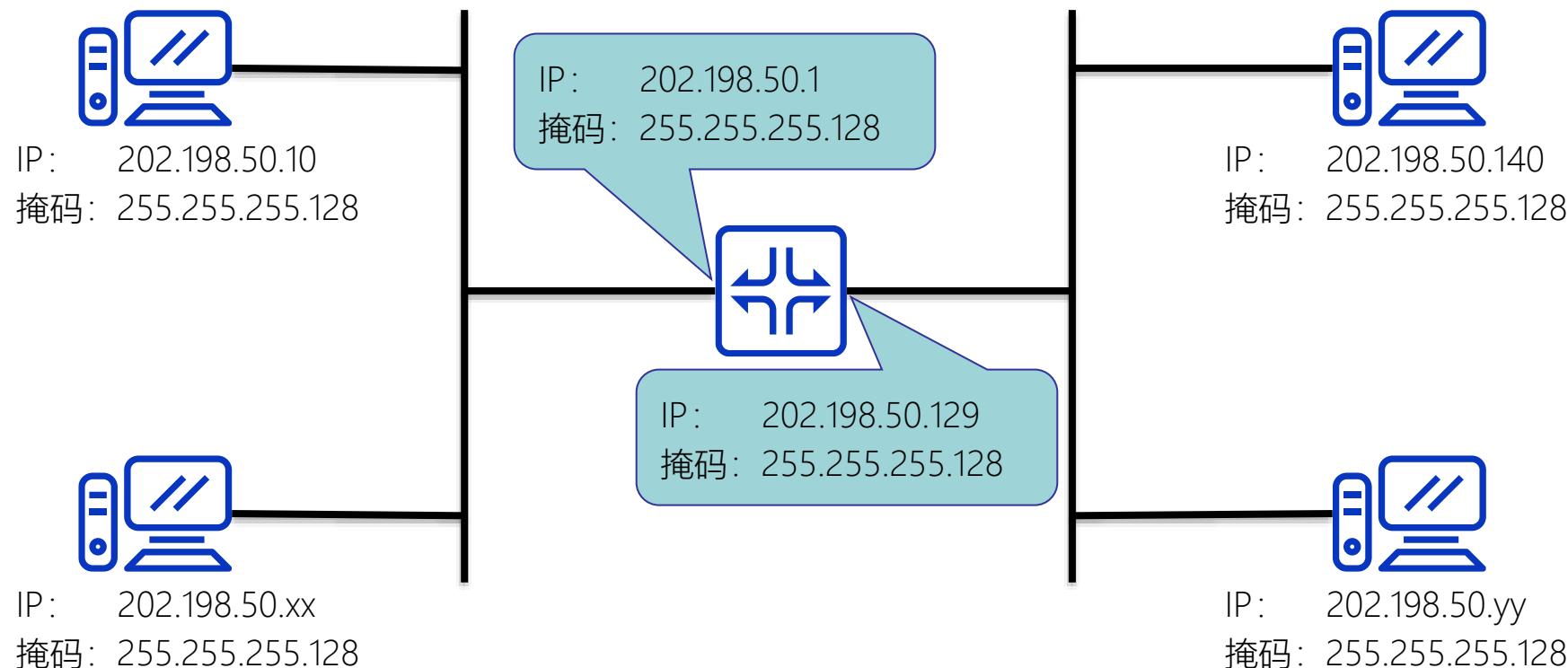
# 例1

- 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？
- $192.168.100.5 \Rightarrow 11000000\ 10101000\ 01100100\ 00000101$   
 $255.255.255.0 \Rightarrow 11111111\ 11111111\ 11111111\ 00000000$ 
  - 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： $2^8 - 2 = 254$

## 例2

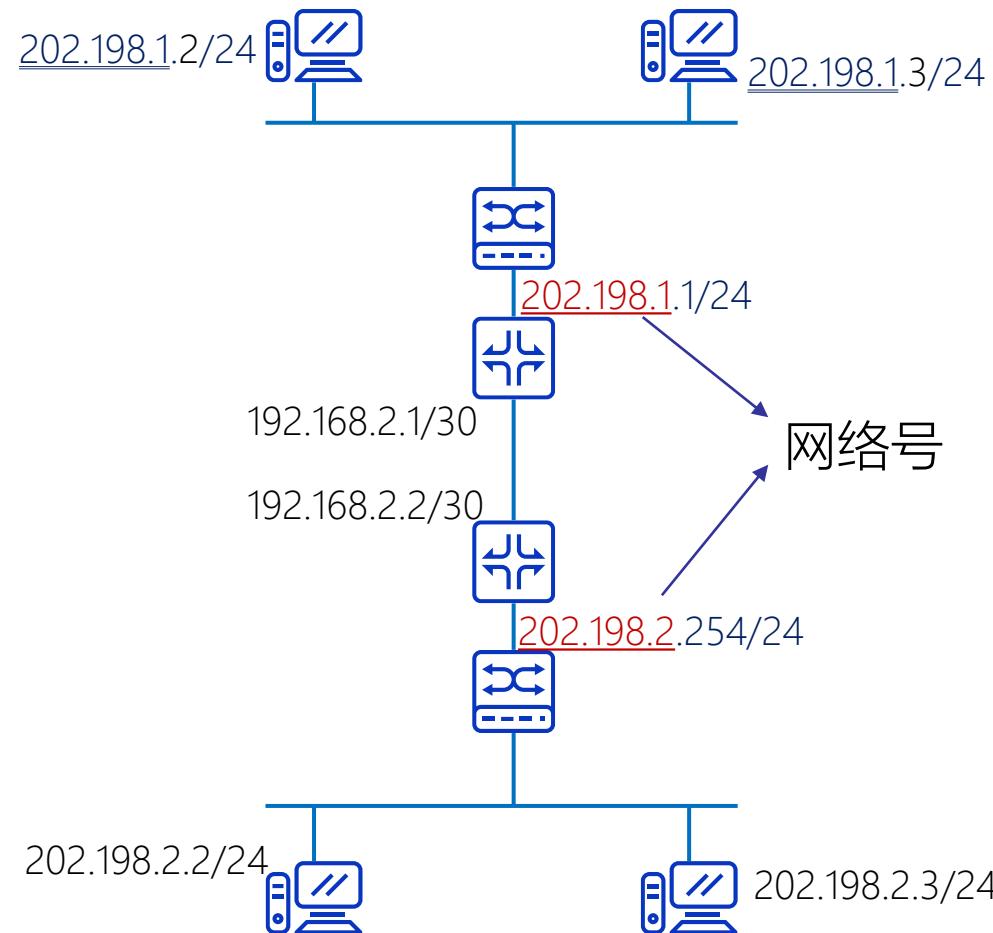
- IP地址为192.168.150.122，子网掩码为255.255.255.248。算出网络地址、广播地址、地址范围、主机数？
- 192.168.150.01111010  
255.255.255.11111000
  - 子网地址：192.168.150.120
  - 广播地址：192.168.150.127
  - 地址范围：192.168.150.121-192.168.150.126
  - 主机的数量： $2^3-2=6$

# 使用255.255.255.128子网掩码的网络



# 网络中IP地址使用示例

- 路由器通常有2个以上的接口
  - 分别连接到不同的网络
  - 每个接口有不同网段的IP地址
- 两台路由器连接的端口，通常使用30位掩码的网络地址，减少网络地址的浪费



- 一个局域网内的所有主机、路由器接口的IP地址必须是同一个网段IP地址
- 网络号通常由管理机构分配，网络内部主机号的分配由使用者自行分配

# 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。但仍然面临问题
  - IP地址资源部分浪费
  - 路由表爆炸的严重问题
- 减小路由表大小的一个方法就是无类别域间路由（Classless Inter-Domain Routing, CIDR）
- CIDR通过把多个地址块组合到一个路由表项，组成一个CIDR地址块，来减少路由表的表项数量，从而减少由核心路由器承载的路由选择信息的数量。同时，CIDR方法也延缓了IP地址资源枯竭的时间

# IP编址问题的演进

1987 年，RFC 1009  
制定了划分子网的  
方案

使用变长子网掩码 VLSM  
(Variable Length Subnet  
Mask)进一步提高 IP 地  
址资源的利用率

无分类编址方法 – 无分类  
域间路由选择 CIDR  
(Classless Inter-Domain  
Routing)



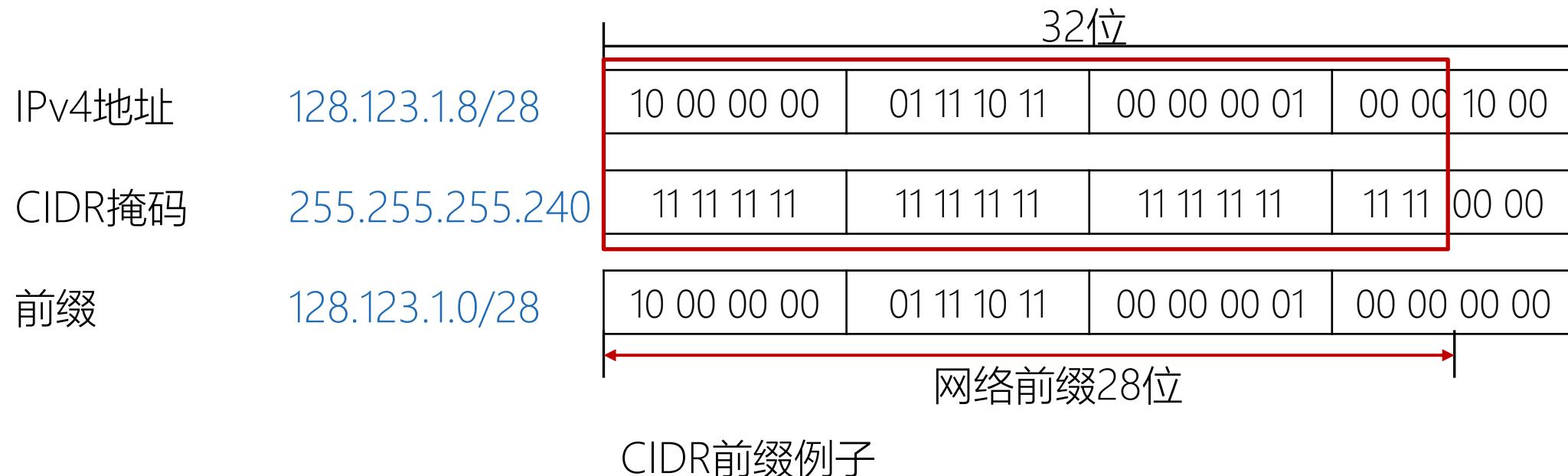
# 无分类的两级编址

## ■ 无分类的两级编址的记法：



- CIDR引入了一种类似于子网掩码的掩码，称之为 CIDR 掩码 或者 掩码
- CIDR 使用“斜线记法”(slash notation)，又称为CIDR记法
  - 在 IP 地址后加上一个斜线“/”，然后写上网络前缀所占的位数
  - IP地址/掩码长度
  - 斜线记法中的数字就是掩码中1的个数
    - 例如，88.166.0.1/19，它的掩码是 19 个连续的 1
- CIDR 把网络前缀都相同的连续的 IP 地址组成“CIDR 地址块”

# CIDR地址表示示例



前缀 (CIDR表示法)	前缀 (二进制表示法)	地址范围
128.0.0.0/1	10000000 00000000 00000000 00000000	128.0.0.0 ~ 255.255.255.255
128.123.1.0/28	10000000 01111011 00000001 00000000	128.123.1.0 ~ 128.123.1.15

二进制表示中，加粗的部分为网络前缀

# 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  
          \u21d3        \u21d3        \u21d3        \u21d3  
      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

# CIDR主要特点

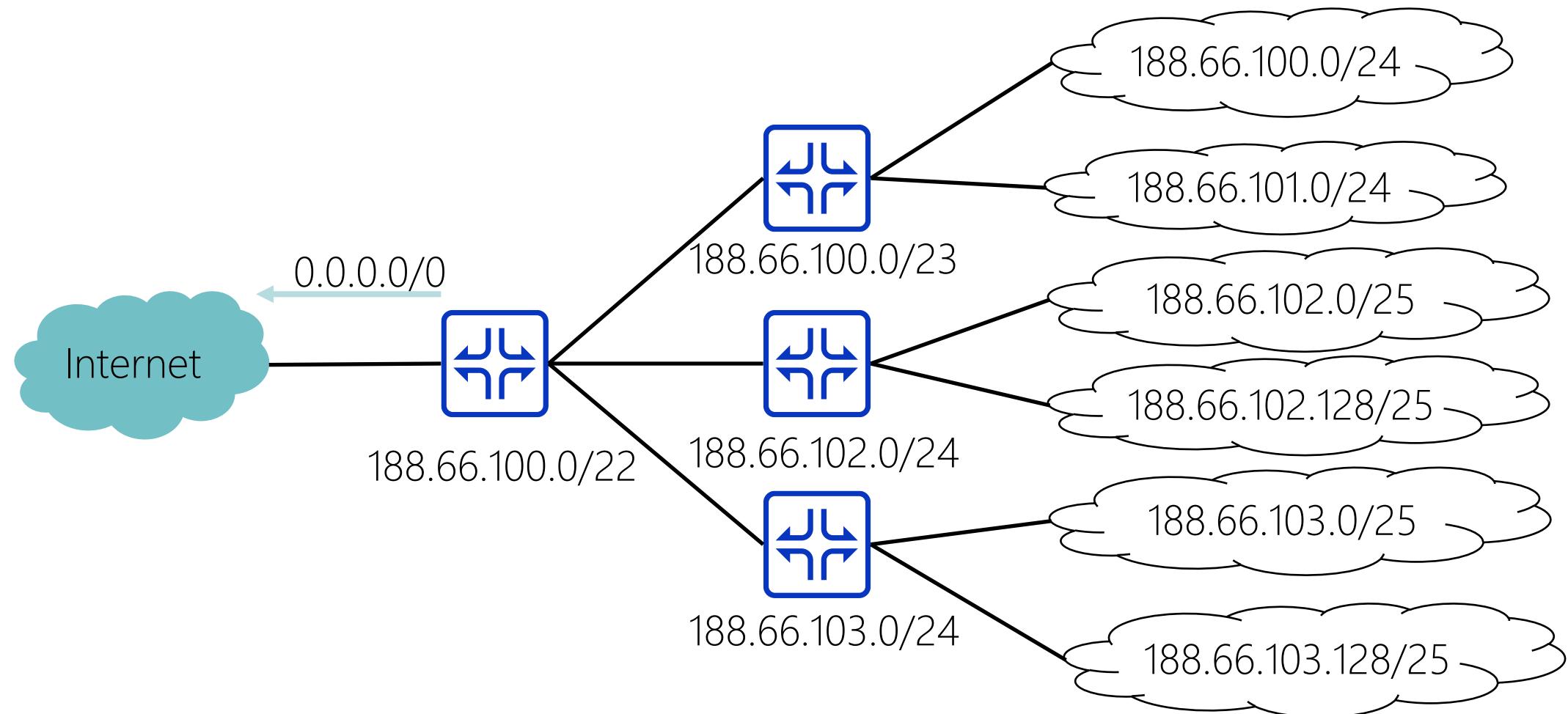
---

- 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

# CIDR地址块

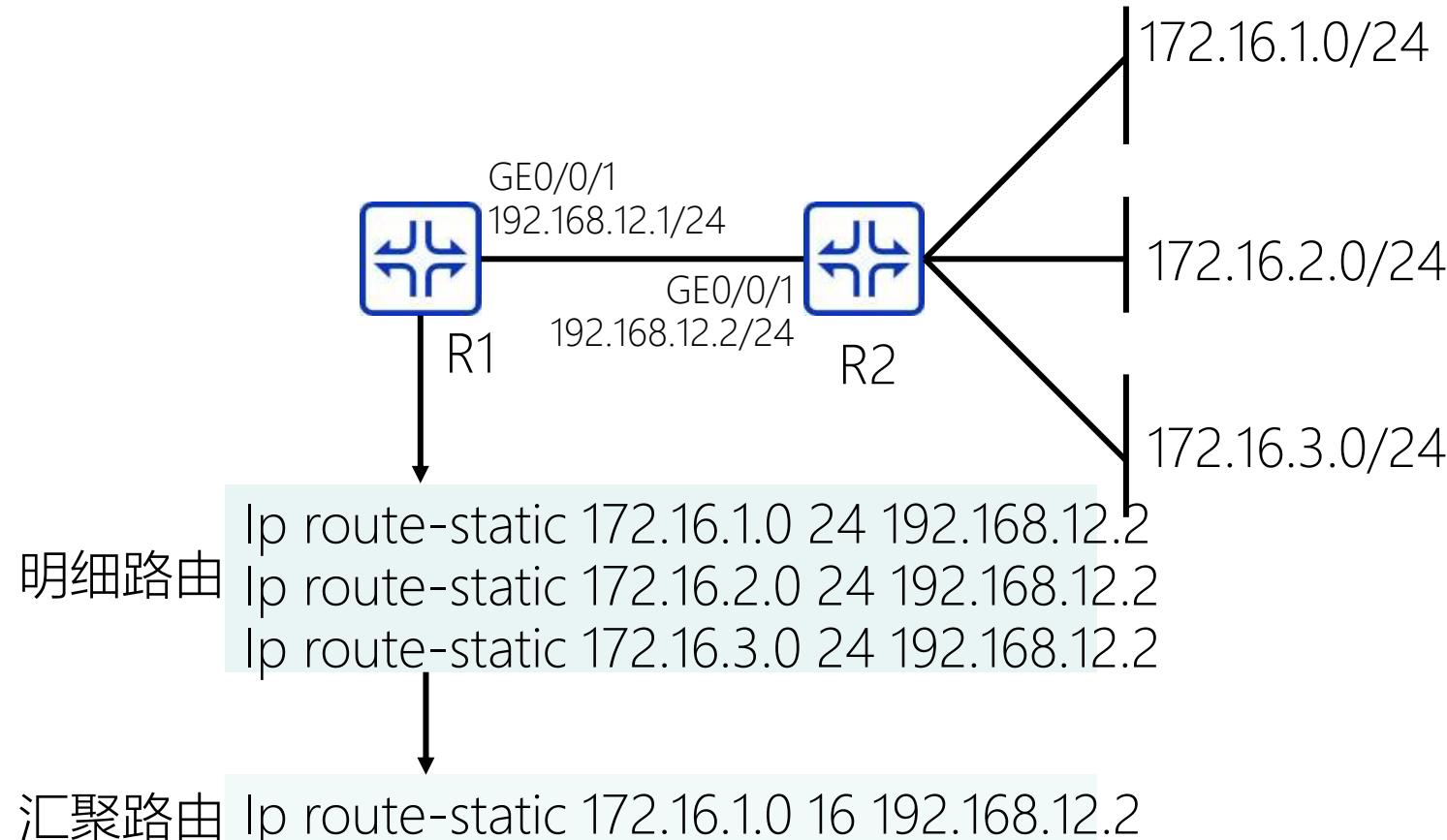
- 128.14.32.0/20 表示的地址块共有 212 个地址（斜线后面的 20 是网络前缀的位数，所以这个地址的主机号是 12 位）
  - 这个地址块的起始地址是 128.14.32.0
  - 128.14.32.0/20 地址块的最小地址：128.14.32.0
  - 128.14.32.0/20 地址块的最大地址：128.14.47.255
  - 全 0 和全 1 的主机号地址一般不使用

# CIDR 地址块划分举例



# 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 (super-netting)
- 为了减小路由表的规模，对于某些属于一个更大网段的子网所对应的路由，可以使用聚合的方法，不发布那些具体的子网路由，代之以发布那个更大网段的路由



# 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

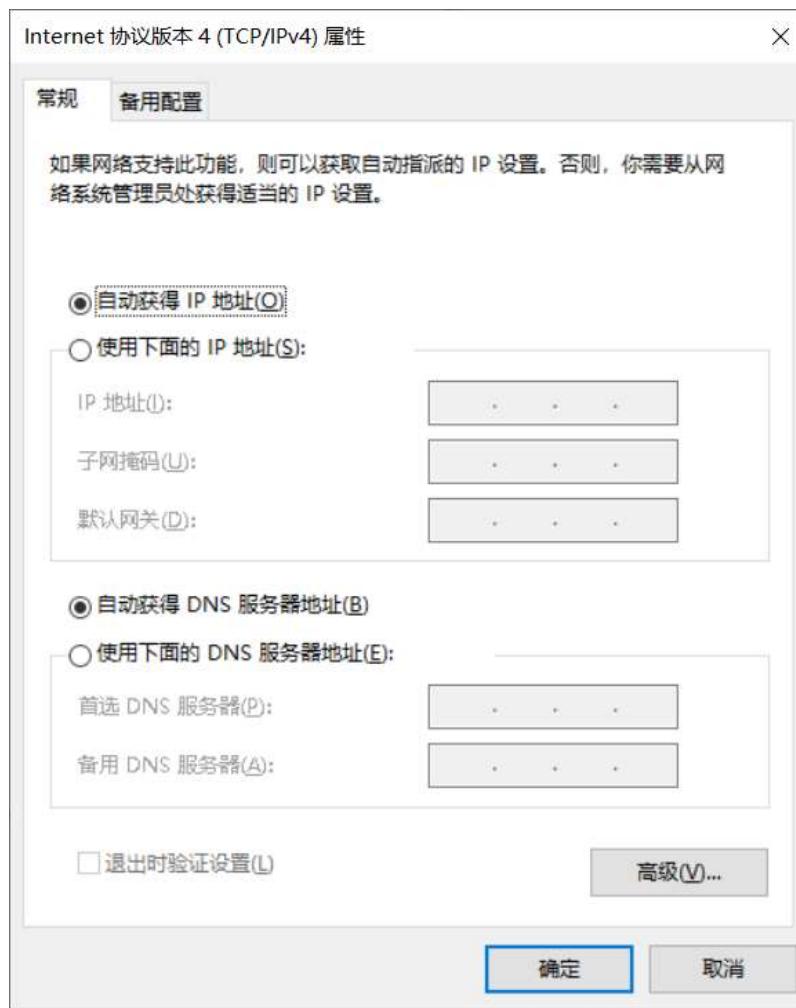
# 如何给主机配置IP地址

- 每台连接到计算机网络的主机均需要配置IP地址等数据，以便与其他主机进行通信：
  - IP地址
  - 网络掩码
  - 默认网关地址
  - 域名服务器（DNS）地址
  - 其他相关配置参数
- 两种方式配置主机的IP地址

	静态配置	动态配置
IP地址规划分配	由网络管理人员统一规划、分配IP地址	由协议自动分配IP地址
设置方式	手工设置到每一台主机	由协议自动配置到主机
IP地址复用	每台主机一个固定IP地址，无法复用	IP地址复用性(一定时间不再使用的IP地址可分配给其他使用者)
IP地址冲突	IP地址容易冲突	避免了IP地址冲突
配置工作量	大	小，只需要配置协议服务器

# 静态配置主机IP地址

## ■ Windows



## ■ Linux

### ■ 命令:

ip [ OPTIONS ] OBJECT { COMMAND | help }

或

ifconfig [-v] [-a] [-s] [interface]

### ■ 编辑配置文件

# 动态主机自动配置

---

- 为了通用性和便于移植，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。但在协议软件运行之前，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

---

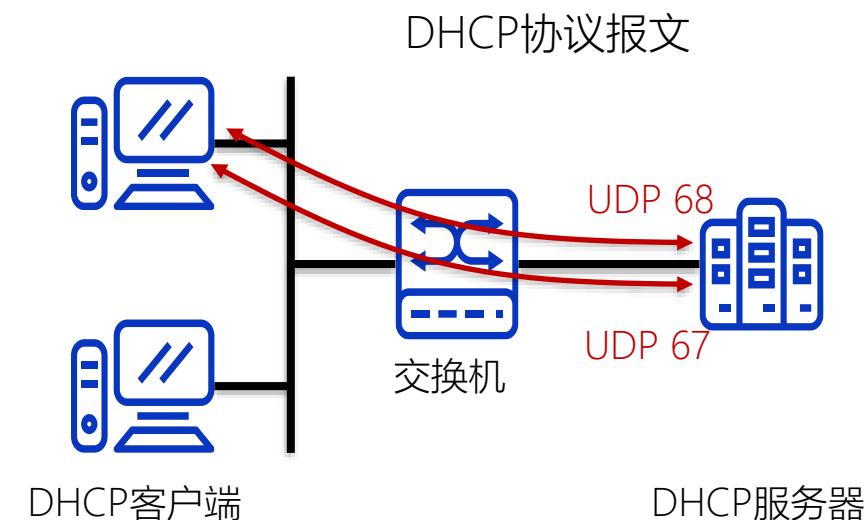
- 一个软件协议在使用之前必须是已正确配置的。具体的配置信息有哪些则取决于协议栈
- 对于使用TCP/IP的主机需要配置的项目
  - IP 地址
  - 子网掩码
  - 默认路由器的 IP 地址
  - 域名服务器的 IP 地址
- RARP ⇒ BOOTP ⇒ DHCP

# 动态主机配置协议--DHCP

- 动态主机配置协议DHCP (Dynamic Host Configuration Protocol, RFC 2131) 是一种网络管理协议，用于集中对用户IP地址进行动态管理和配置
- DHCP协议的优点：
  - 使网络管理员能从中心结点监控和分配IP地址
  - 当某台计算机移到网络中的其它位置时，能自动收到新的IP地址
  - 实现的自动化分配IP地址不仅降低了配置和部署设备的时间，同时也降低了发生配置错误的可能性
  - DHCP服务器可以管理多个网段的配置信息，当某个网段的配置发生变化时，管理员只需要更新DHCP服务器上的相关配置即可，实现了集中化管理

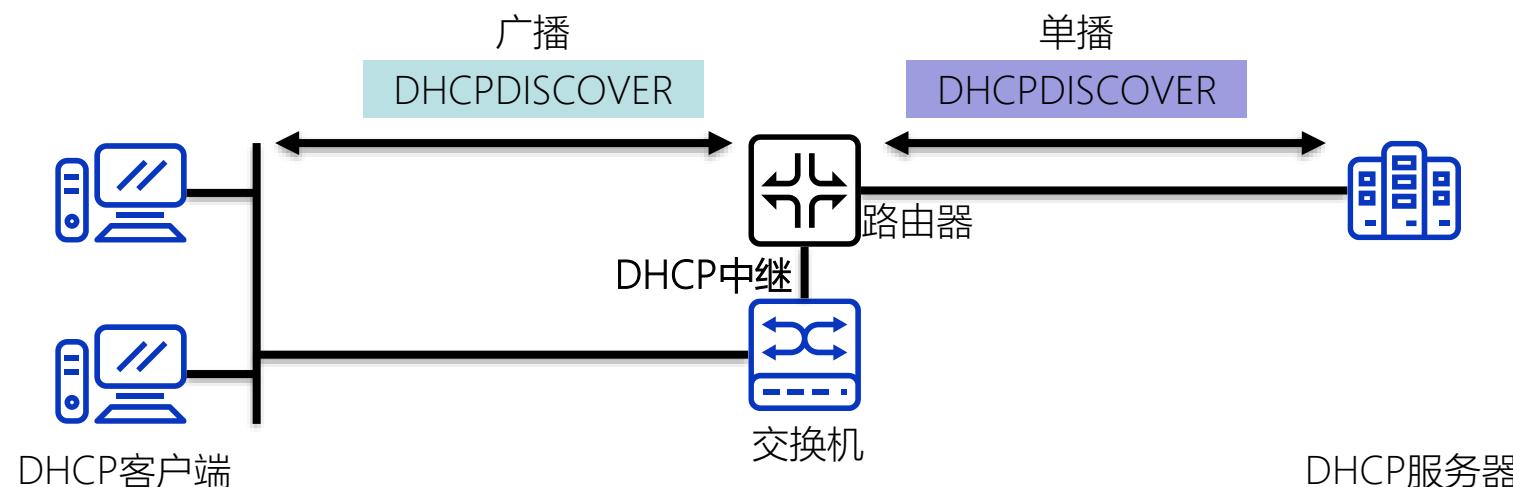
# DHCP工作原理—局域网

- DHCP协议采用客户端/服务器通信模式：
  - 客户端（DHCP Client）向服务器（DHCP Server）发送配置请求
  - DHCP服务器为网络上的每个设备动态分配相关配置参数，以便客户端可以与其他主机通信
- DHCP协议采用UDP作为传输协议
  - DHCP客户端发送请求消息到DHCP服务器的67号端口
  - DHCP服务器回应应答消息给DHCP客户端的68号端口
- 只有跟DHCP客户端在同一个网段的DHCP服务器才能收到DHCP客户端广播的DHCP DISCOVER报文



# DHCP工作原理—跨网段

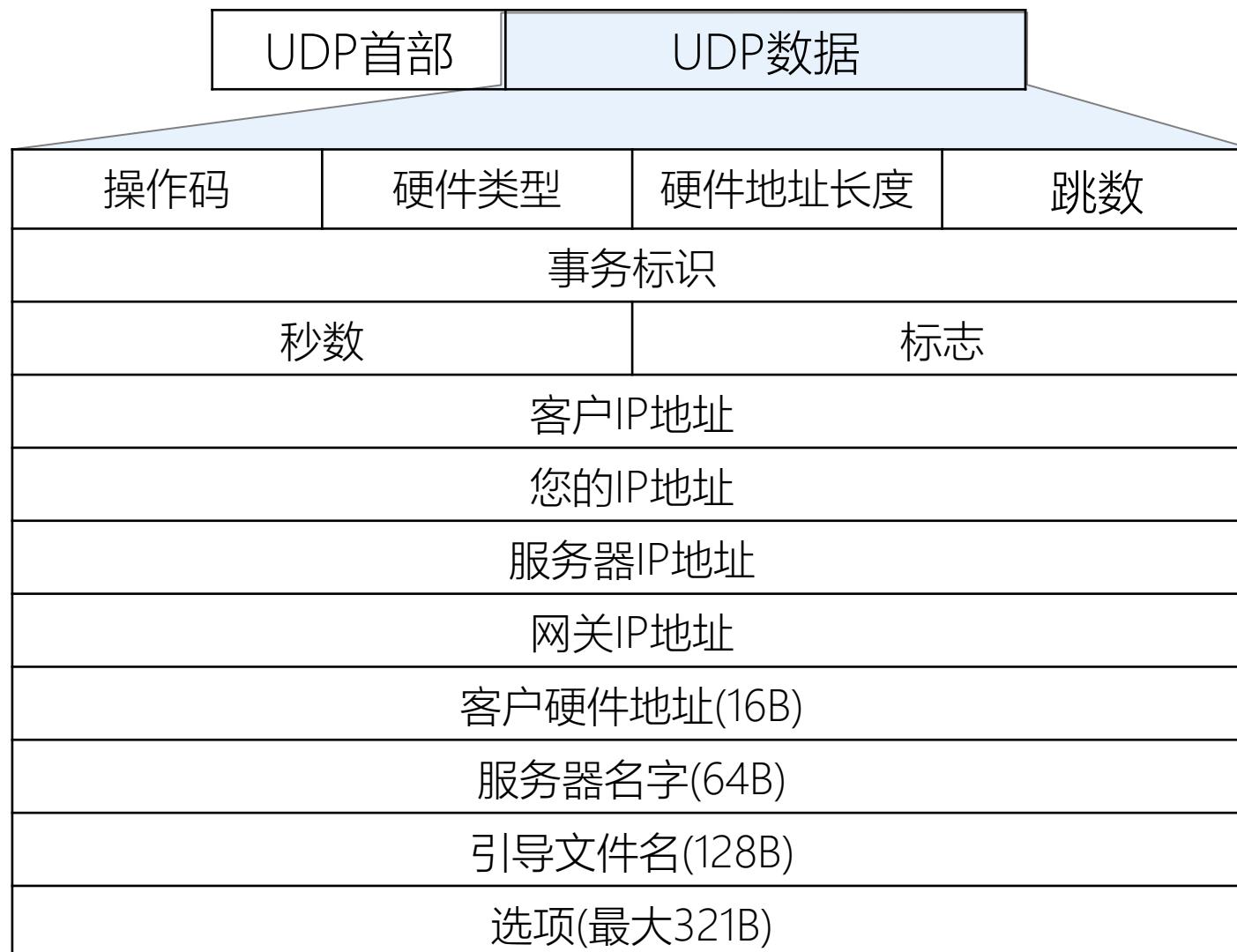
- 一个组织可以只部署一台DHCP服务器来管理多网段的配置，同时设置 DHCP 中继
- 当DHCP客户端与DHCP服务器不在同一个网段时，必须部署DHCP中继来转发DHCP客户端和DHCP服务器之间的DHCP报文：
- 在DHCP客户端看来，DHCP中继就像DHCP服务器，DHCP 中继收到主机发送的发现报文后，以单播方式向 DHCP 服务器转发此报文
- 在DHCP服务器看来，DHCP中继就像DHCP客户端，DHCP 中继收到服务器回复的报文后，将此报文发回给主机



# DHCP的地址分配机制

- 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
  - 在C/S架构的软件中，服务端软件通常需要一个固定的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
  - 在C/S架构的软件中，客户端软件通常分配一个临时的IP地址
  - 这种地址，计算机重启后可能会发生变化
- 人工分配—DHCP客户的IP地址由管理员分配好，DHCP只负责传达

# 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

- 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 地址是临时的
  - 租用期：DHCP 客户只能在一段有限的时间内使用分配到的 IP 地址
- 释放IP地址

# DHCP工作过程—发现阶段

- 首次接入网络的DHCP客户端不知道DHCP服务器的IP地址
- 为了学习到DHCP服务器的IP地址，DHCP客户端以广播方式发送DHCP DISCOVER报文（目的IP地址为255.255.255.255）给同一网段内的所有设备（包括DHCP服务器或中继）
- DHCP DISCOVER报文中携带了：
  - 客户端的MAC地址（chaddr字段）
  - 需要请求的参数列表选项
  - 广播标志位（flags字段）等信息



# DHCP工作过程—提供阶段

- 与DHCP客户端位于同一网段的所有DHCP服务器都会接收到DHCP DISCOVER报文
- DHCP服务器选择跟接收DHCP DISCOVER报文接口的IP地址处于同一网段的地址池，并且从中选择一个可用的IP地址
- 然后通过DHCP OFFER报文发送给DHCP客户端



# DHCP工作过程—选择阶段

- 如果有多个DHCP服务器向DHCP客户端回应DHCP OFFER报文，则DHCP客户端一般只接收第一个收到的DHCP OFFER报文，然后以广播方式发送DHCP REQUEST报文，该报文中包含客户端想选择的DHCP服务器标识符和客户端IP地址
- DHCP客户端通知所有的DHCP服务器，它将选择某个DHCP服务器提供的IP地址，其他DHCP服务器可以重新将曾经分配给客户端的IP地址分配给其他客户端



# DHCP工作过程—确认阶段

- DHCP服务器收到客户端发送的DHCP REQUEST报文后，DHCP服务器回应DHCP ACK报文，表示DHCP REQUEST报文中请求的IP地址分配给客户端使用
- DHCP客户端收到DHCP ACK报文，使用此地址，过程结束

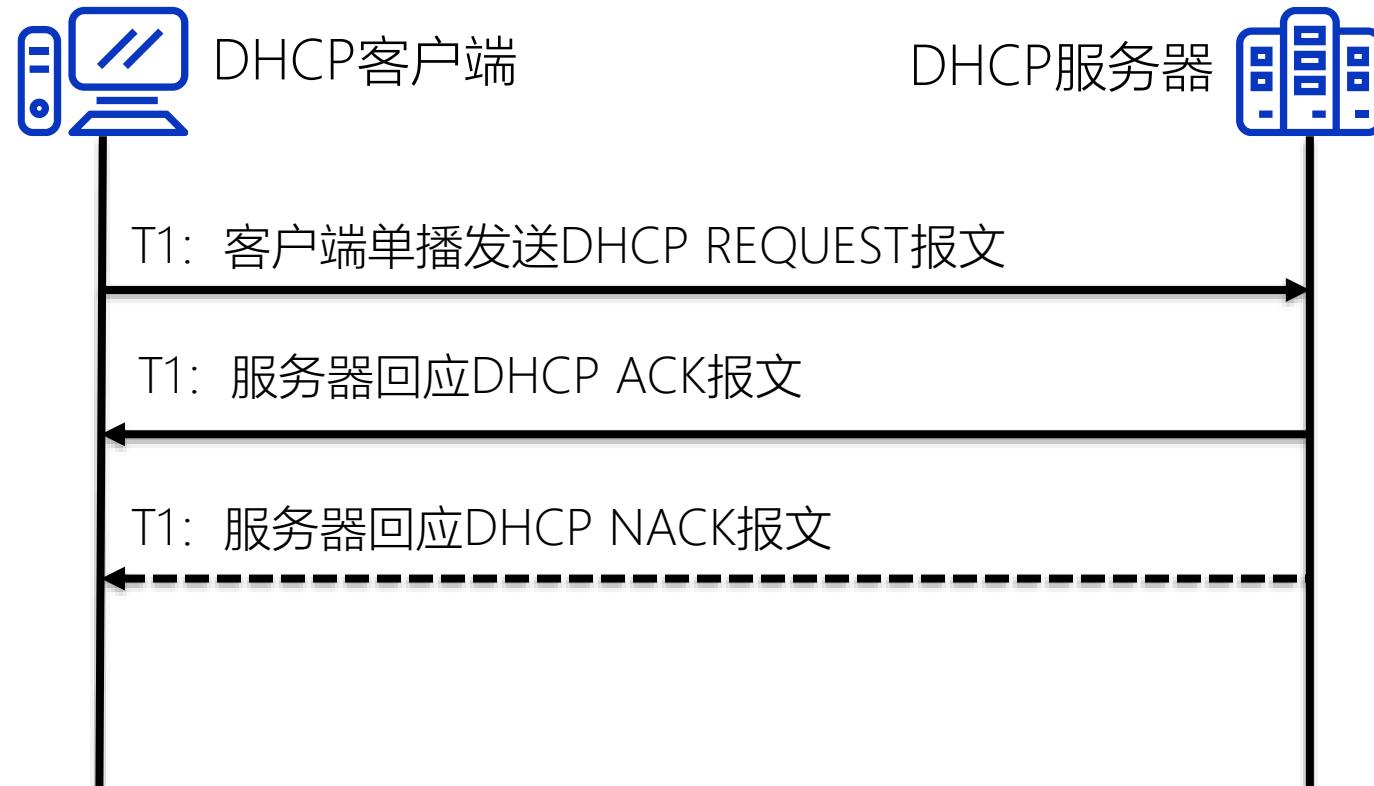


# DHCP工作过程—更新租期

当租期达到50% (T1) 时

- DHCP客户端会自动以单播的方式向DHCP服务器发送DHCP REQUEST报文，请求更新IP地址租期

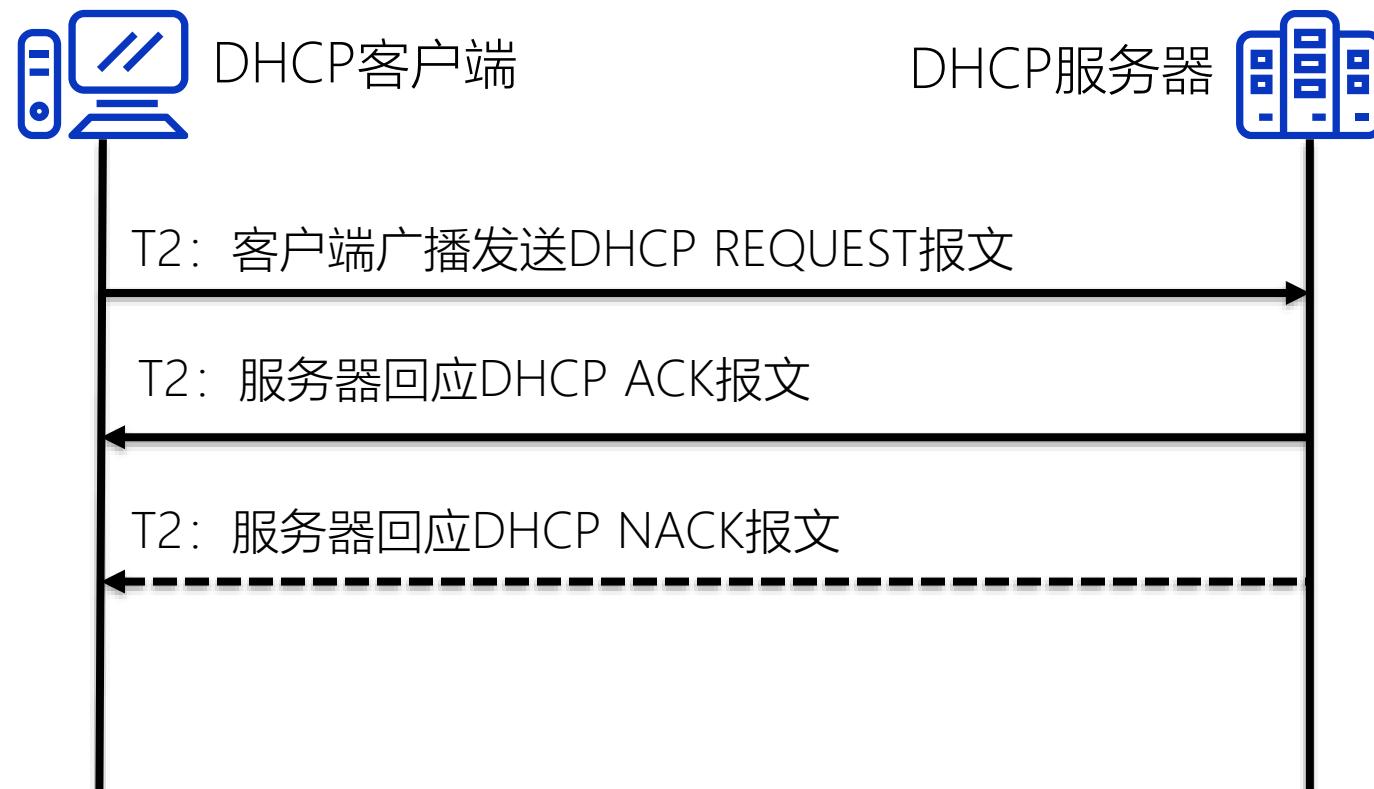
- 如果收到DHCP服务器回应的DHCP ACK报文，则租期更新成功（即租期从0开始计算）；
- 如果收到DHCP NACK报文，则重新发送DHCP DISCOVER报文请求新的IP地址



# DHCP工作过程—更新租期

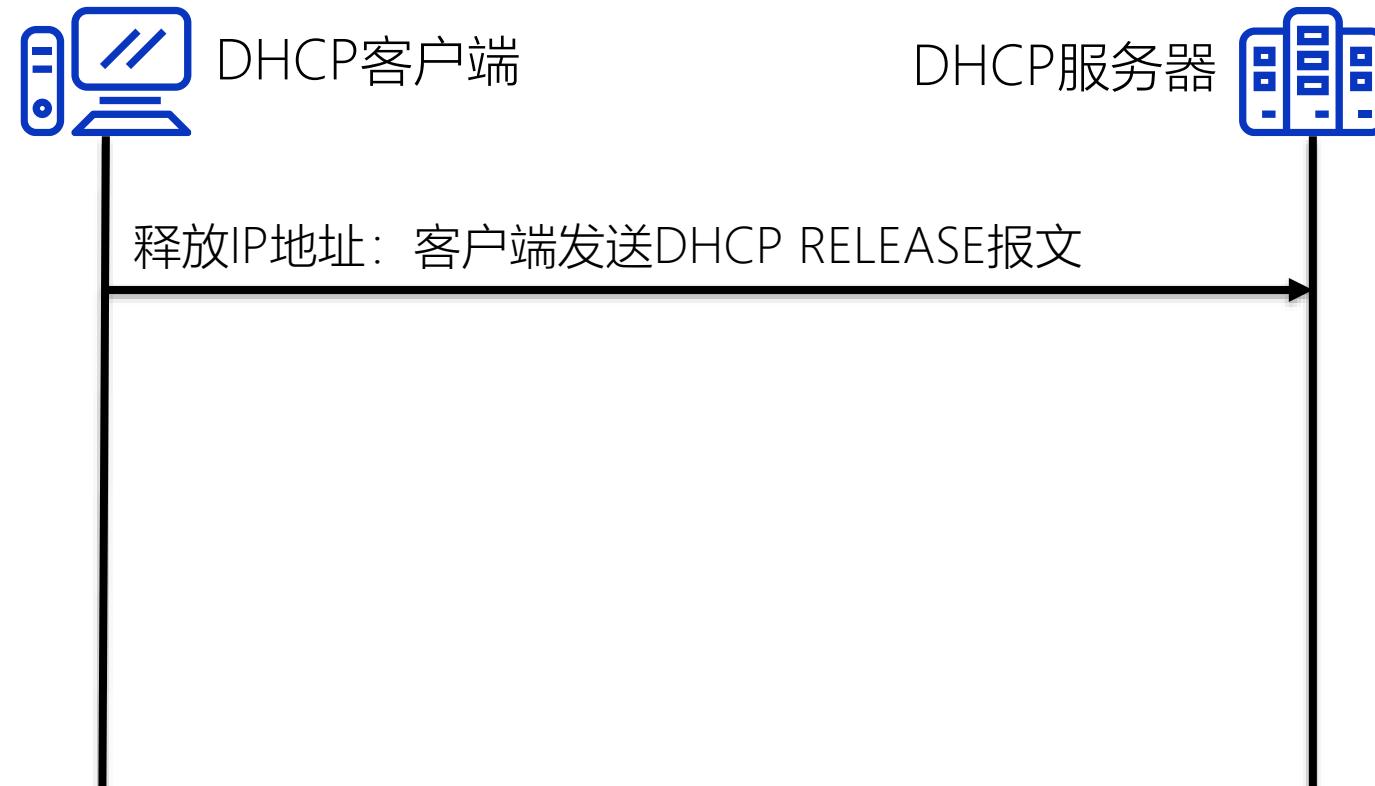
- 当租期达到87.5% (T2) 时:

- 如果仍未收到DHCP服务器的应答，DHCP客户端会自动以广播的方式向DHCP服务器发送DHCP REQUEST报文，请求更新IP地址租期
- 如果收到DHCP服务器回应的DHCP ACK报文，则租期更新成功（租期从0开始计算）
- 如果收到DHCP NACK报文，则重新发送DHCP DISCOVER报文请求新的IP地址



# DHCP工作过程——释放地址

- 客户端在租期时间到之前，如果用户不想使用分配的IP地址（例如客户端网络位置需要变更），会触发DHCP客户端向DHCP服务器发送DHCP RELEASE报文，通知DHCP服务器释放IP地址的租期
- DHCP服务器会保留这个DHCP客户端的配置信息，将IP地址列为曾经分配过的IP地址中，以便后续重新分配给该客户端或其他客户端



# DHCP协议操作与优化

## ■ 优化与操作的措施：

- 数据包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

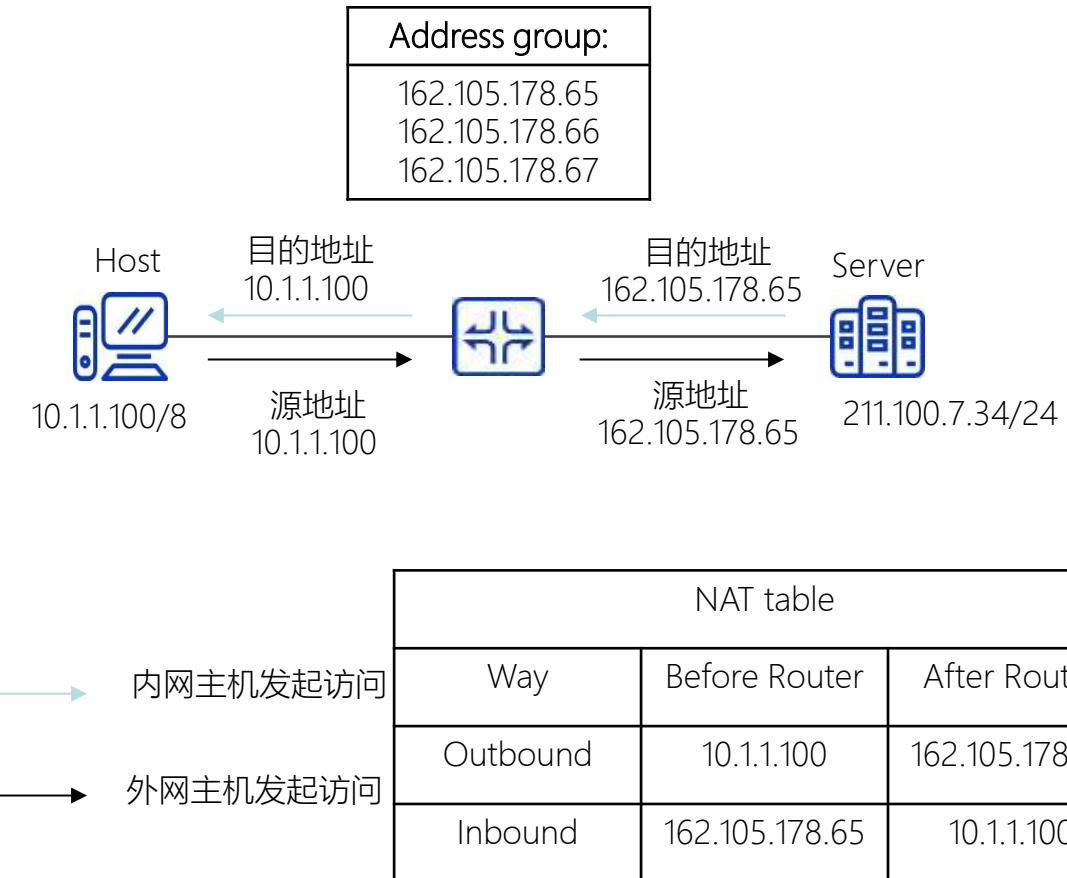
- 网络地址转换 (Network Address Translation – NAT) 是一种地址转换技术，它可以将IP数据报文头中的IP地址转换为另一个IP地址，并通过转换端口号达到地址重用的目的
- NAT主要用于实现内部网络（简称内网，使用私有IP地址）访问外部网络（简称外网，使用公有IP地址）的功能。当内网的主机要访问外网时，通过NAT技术可以将其私网地址转换为公网地址，可以实现多个私网用户共用一个公网地址来访问外部网络，这样既可保证网络互通，又节省了公网地址
- NAT作为一种缓解IPv4公网地址枯竭的过渡技术，由于实现简单，得到了广泛应用

# NAT原理 – Basic NAT

■ Basic NAT方式属于一对一的地址转换，在这种方式下只转换IP地址，而不处理TCP/UDP协议的端口号，一个公网IP地址不能同时被多个私网用户使用

■ 地址转换过程如：

- Router收到内网侧Host发送的访问公网侧Server的报文，其源IP地址为10.1.1.100
- Router从地址池中选取一个空闲的公网IP地址，建立与内网侧报文源IP地址间的NAT转换表项（正反向），并依据查找正向NAT表项的结果将报文转换后向公网侧发送，其源IP地址是10.1.1.100，目的IP地址是162.105.178.65
- Router收到公网侧的回应报文后，根据其目的IP地址查找反向NAT表项，并依据查表结果将报文转换后向私网侧发送，其源IP地址是162.105.178.65，目的IP地址是10.1.1.100



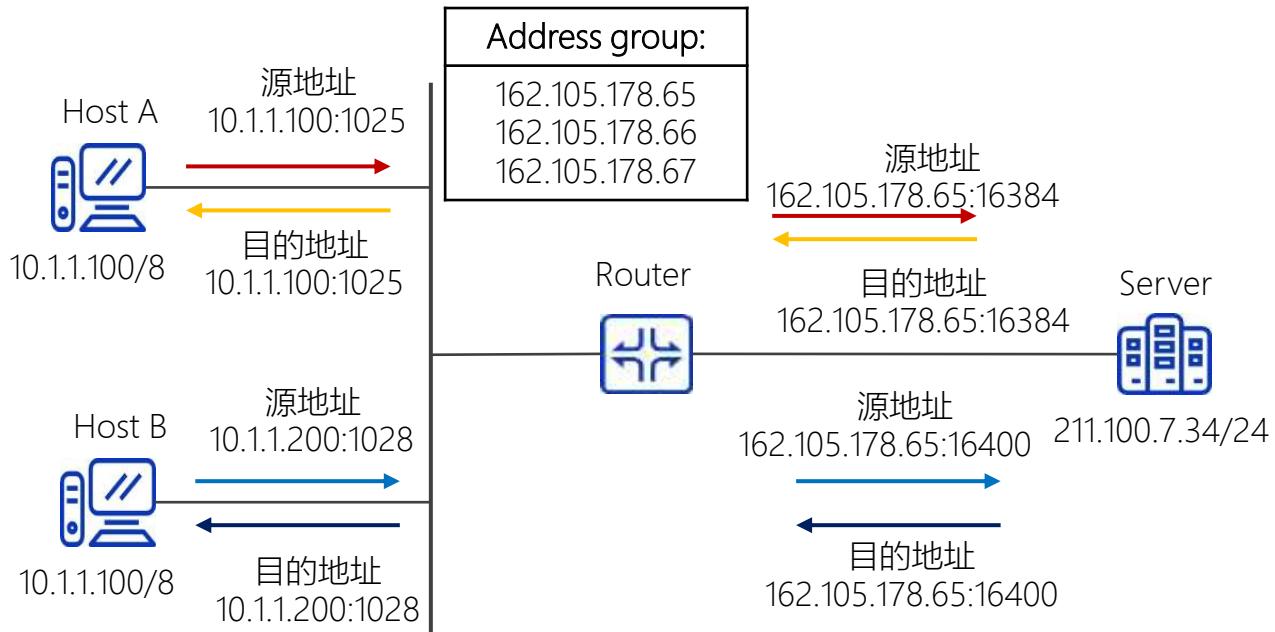
由于Basic NAT这种一对一的转换方式并未实现公网地址的复用，不能有效解决IP地址短缺的问题，因此在实际应用中并不常用

# NAT原理 – NAPT

- 网络地址端口转换NAPT (Network Address Port Translation) 通过使用“IP地址 + 端口号”的形式进行转换，使多个私网用户可共用一个公网IP地址访问外网

- 地址转换过程如下：

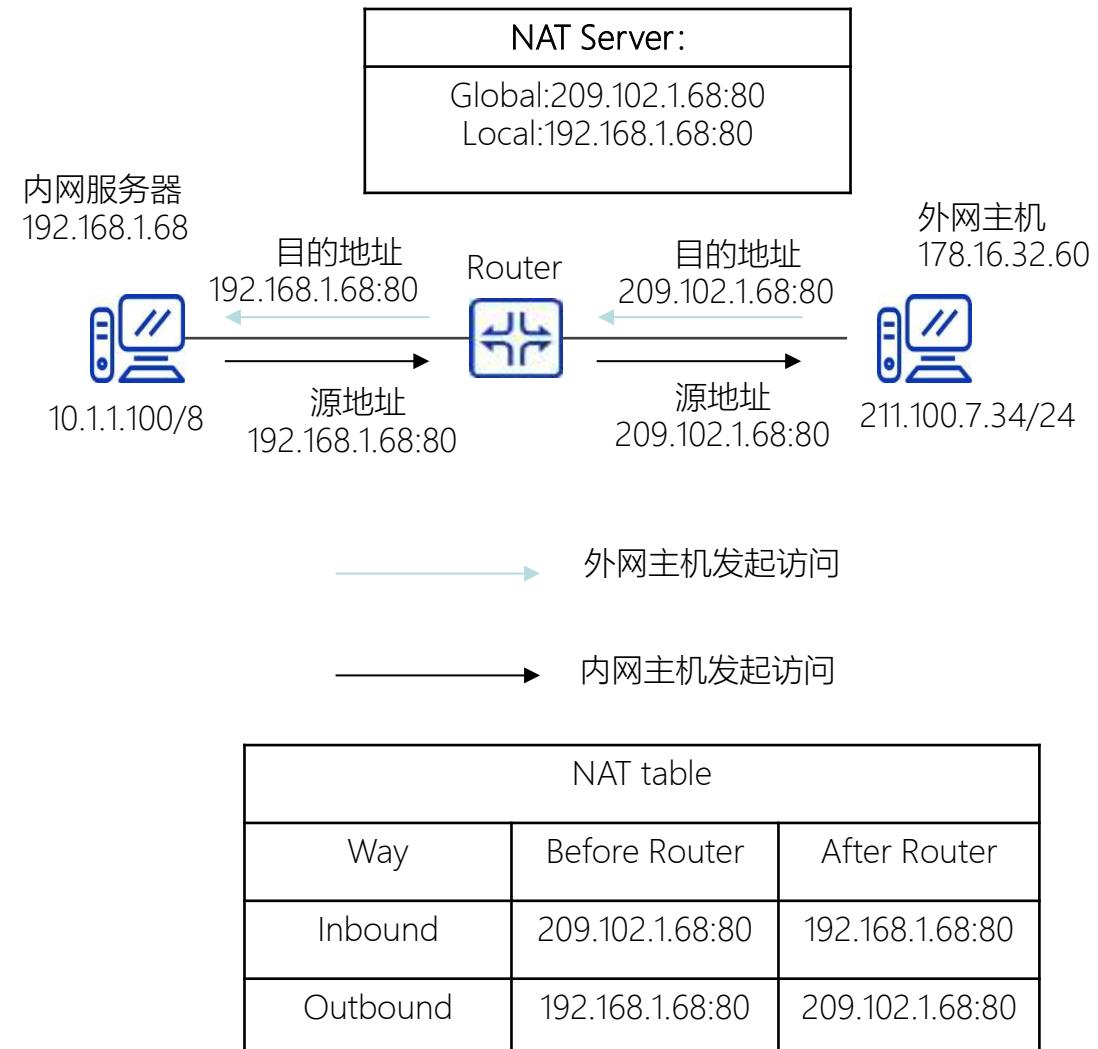
- Router收到内网侧Host发送的访问公网侧Server的报文。比如收到Host A报文的源地址是10.1.1.100，端口号1025
- Router从地址池中选取一对空闲的“公网IP地址 + 端口号”，建立与内网侧报文“源IP地址 + 源端口号”间的NAPT转换表项（正反向），并依据查找正向NAPT表项的结果将报文转换后向公网侧发送。比如Host A的报文经Router转换后的报文源地址为162.105.178.65，端口号16384
- Router收到公网侧的回应报文后，根据其“目的IP地址 + 目的端口号”查找反向NAPT表项，并依据查表结果将报文转换后向私网侧发送。比如Server回应Host A的报文经Router转换后，目的地址为10.1.1.100，端口号1025



NAT table		
Way	Before Router	After Router
Outbound	10.1.1.100:1025	162.105.178.65:16384
Inbound	162.105.178.65:16384	10.1.1.100:1025
Outbound	10.1.1.200:1028	162.105.178.65:16400
Inbound	162.105.178.65:16400	10.1.1.200:1028

# NAT原理 – NAT Server

- NAT具有“屏蔽”内部主机的作用，但有时内网需要向外网提供服务。这种情况下需要内网的服务器不被“屏蔽”，外网用户可以随时访问内网服务器
- NAT Server：外网用户访问内网服务器时，通过事先配置好的“公网IP地址+端口号”与“私网IP地址+端口号”间的映射关系，将服务器的“公网IP地址+端口号”根据映射关系替换成对应的“私网IP地址+端口号”
- 地址转换过程：
  - 在Router上配置NAT Server的转换表项
  - Router收到公网用户发起的访问请求，设备根据该请求的“目的IP+端口号”查找NAT Server转换表项，找出对应的“私网IP+端口号”，然后用查找结果替换报文的“目的IP+端口号”
  - Router收到内网服务器的回应报文后，根据该回应报文的“源IP地址 + 源端口号”查找NAT Server转换表项，找出对应的“公网IP+端口号”，然后用查找结果替换报文的“源IP地址 + 源端口号”



# NAT的优缺点

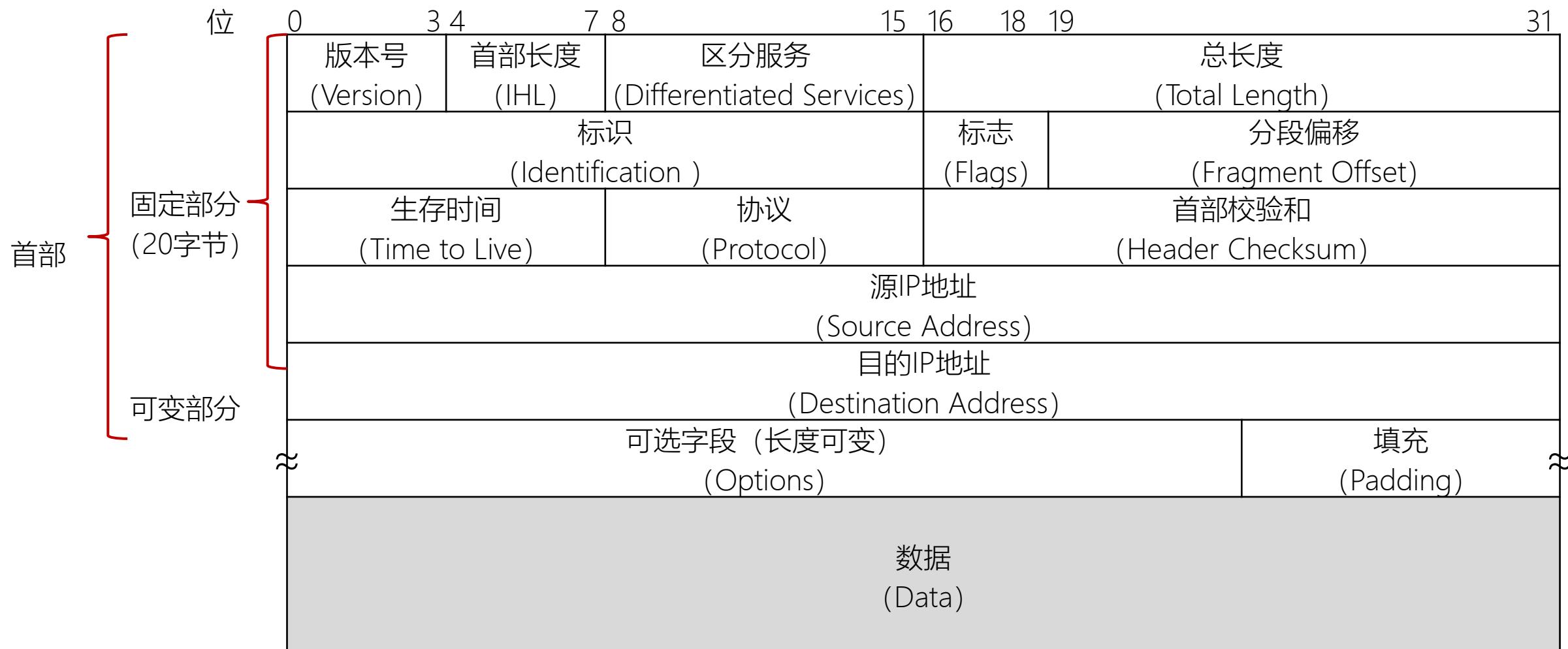
## ■ NAT的优点：

- 作为减缓IP地址枯竭的一种过渡方案，NAT通过地址重用的方法来满足IP地址的需要，可以在一定程度上缓解IP地址空间枯竭的压力
- 有效避免来自外网的攻击，可以很大程度上提高网络安全性
- 控制内网主机访问外网，同时也可以控制外网主机访问内网，解决了内网和外网不能互通的问题

## ■ 缺点：

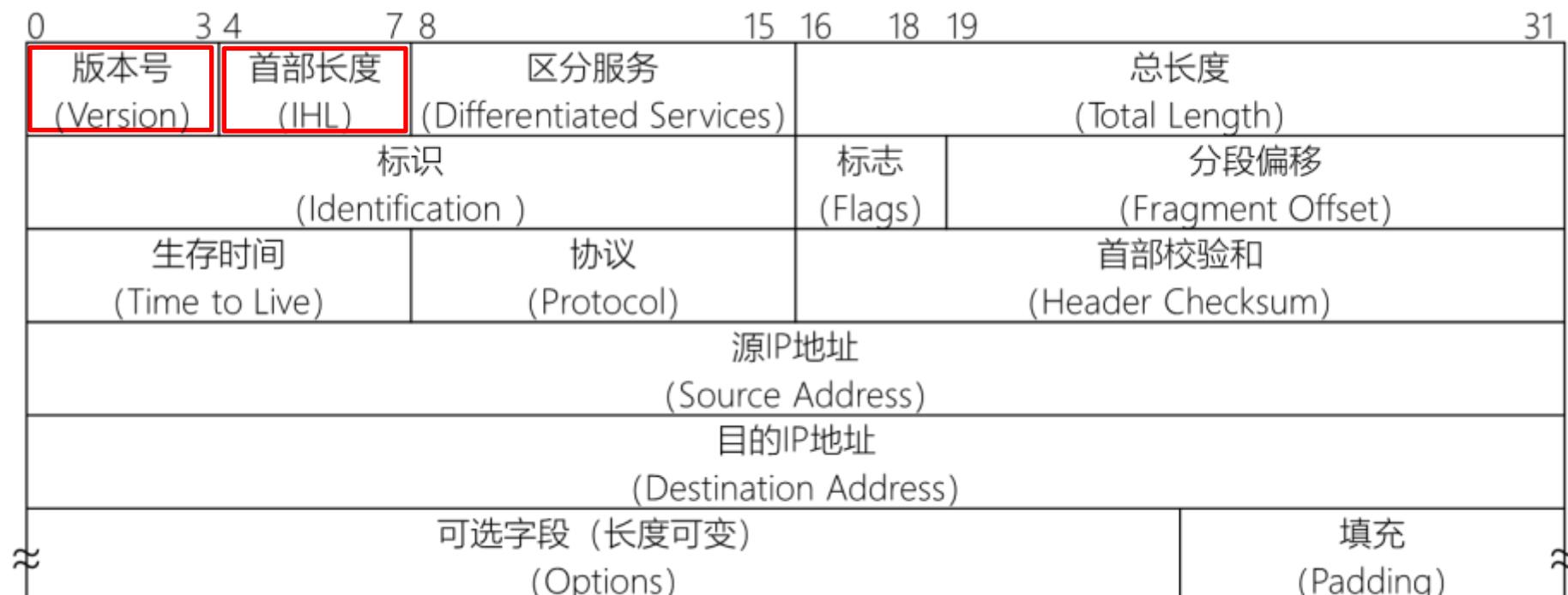
- NAT和NAPT只对IP报文的头部地址和TCP/UDP头部的端口信息进行转换。对于一些协议，例如FTP等，在报文的数据部分可能包含IP地址信息或者端口信息，需要NAT实现中支持应用层网关ALG（Application Level Gateway）功能

# IP报文格式



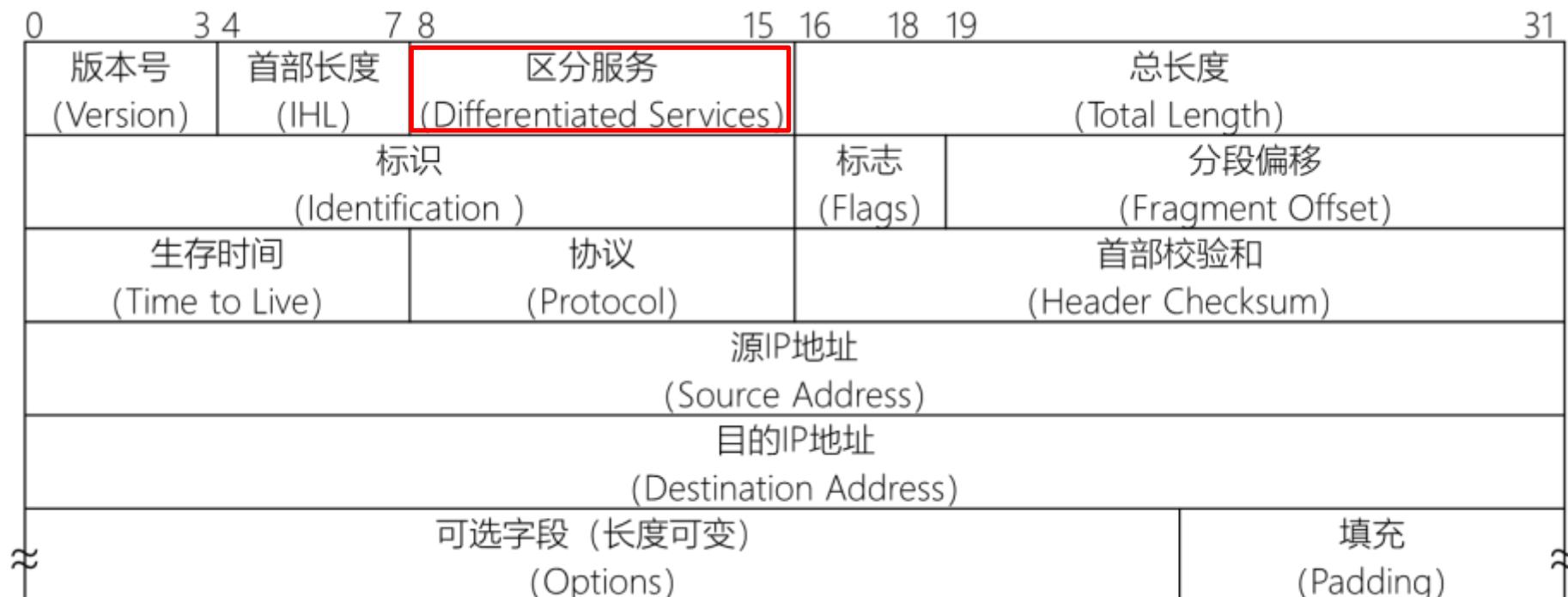
# IP报文格式

- 版本号：IPv4协议的版本号为4
- 报头长度：报头占32位的数量
  - 无选项时是20字节，即该字段的值为5
  - 报头最长60B（包括可选字段）



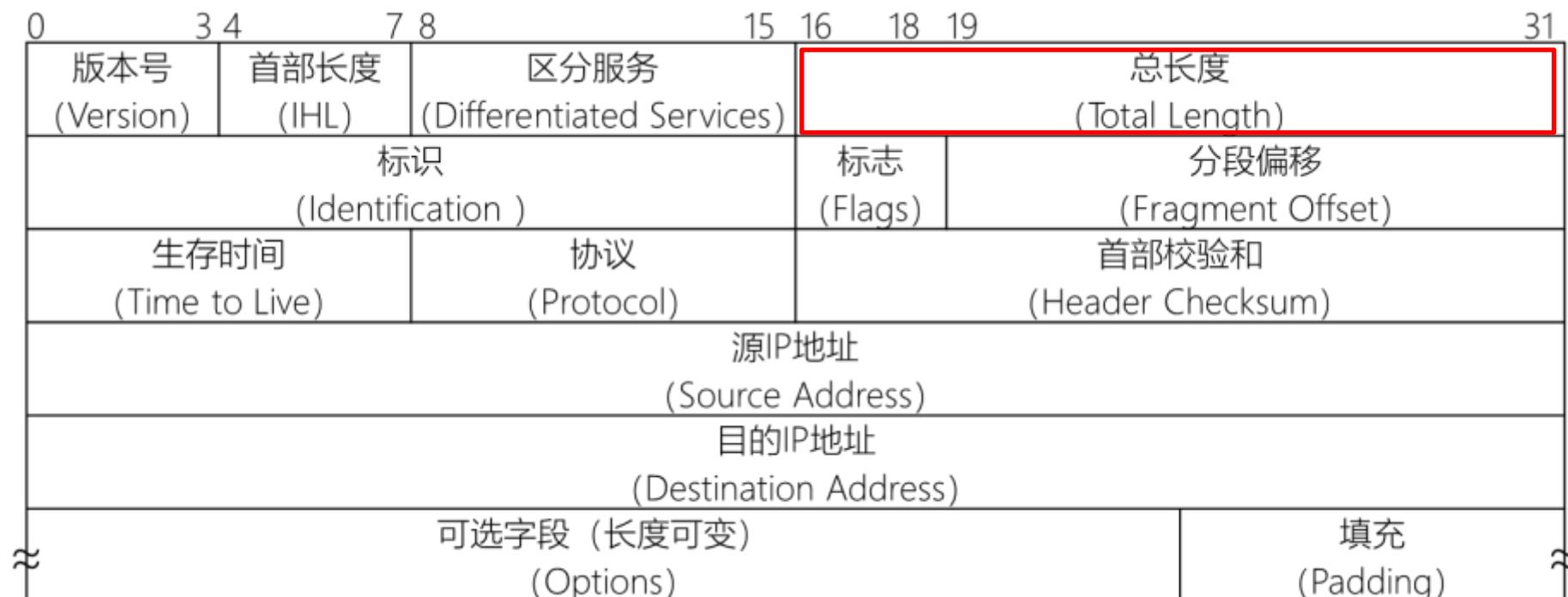
# IP报文格式

- 区分服务：该字段最初称为服务类型（TOS），但未用过
- 1998年，IETF重新启用了这个字段，并更名为区分服务：
  - 前6位用来标记数据报的服务类别
  - 后2位用来携带显示拥塞通知信息



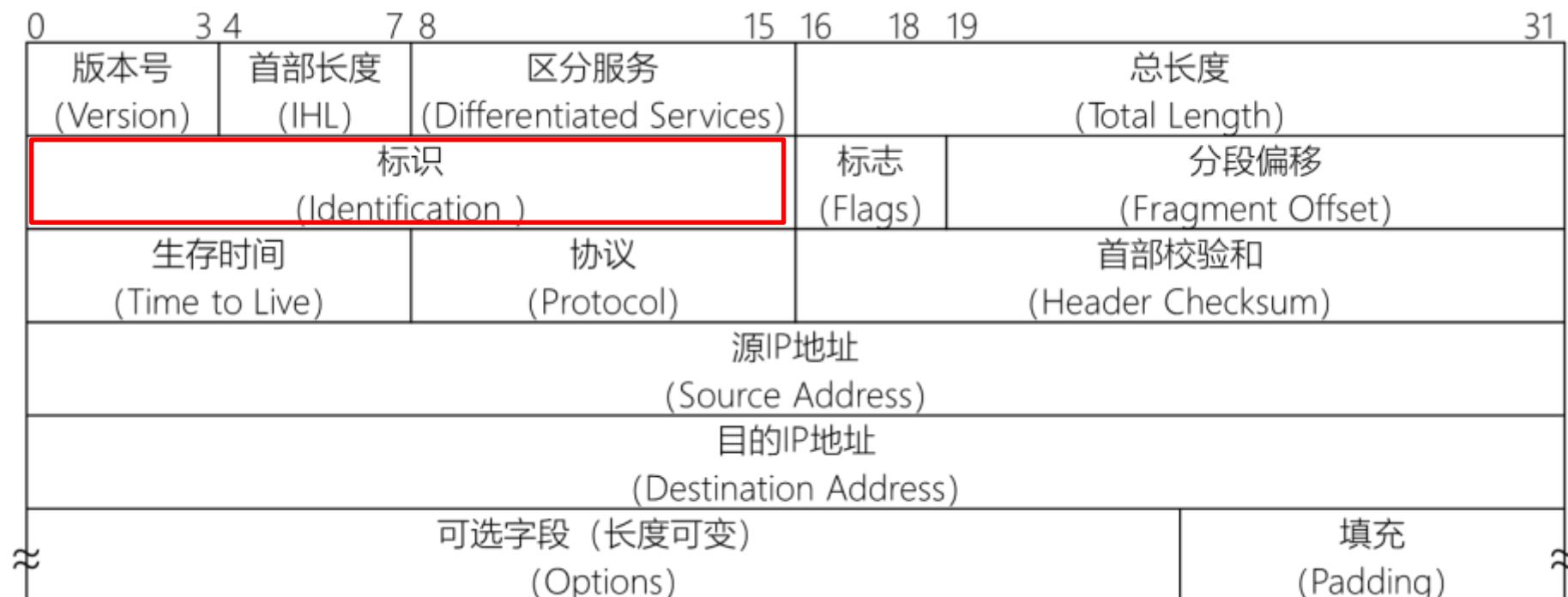
# IP报文格式

- 总长度：指整个IP数据报的长度，以字节为单位（IP首部加上数据）
- 由于该字段长度为16比特，所以IP数据报的最大长度可达65536字节
- 当数据报被分段处理时，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

- 标识：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
- 如果数据包被分段以适应小型数据包的网络，那么同一数据报的每一个分片中都使用相同的标识号



# IP报文格式

## ■ 标志：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

版本号 (Version)	首部长度 (IHL)	区分服务 (Differentiated Services)	总长度 (Total Length)
标识 (Identification)		标志 (Flags)	分段偏移 (Fragment Offset)
生存时间 (Time to Live)	协议 (Protocol)	首部校验和 (Header Checksum)	
源IP地址 (Source Address)			目的IP地址 (Destination Address)
可选字段 (长度可变) (Options)		填充 (Padding)	

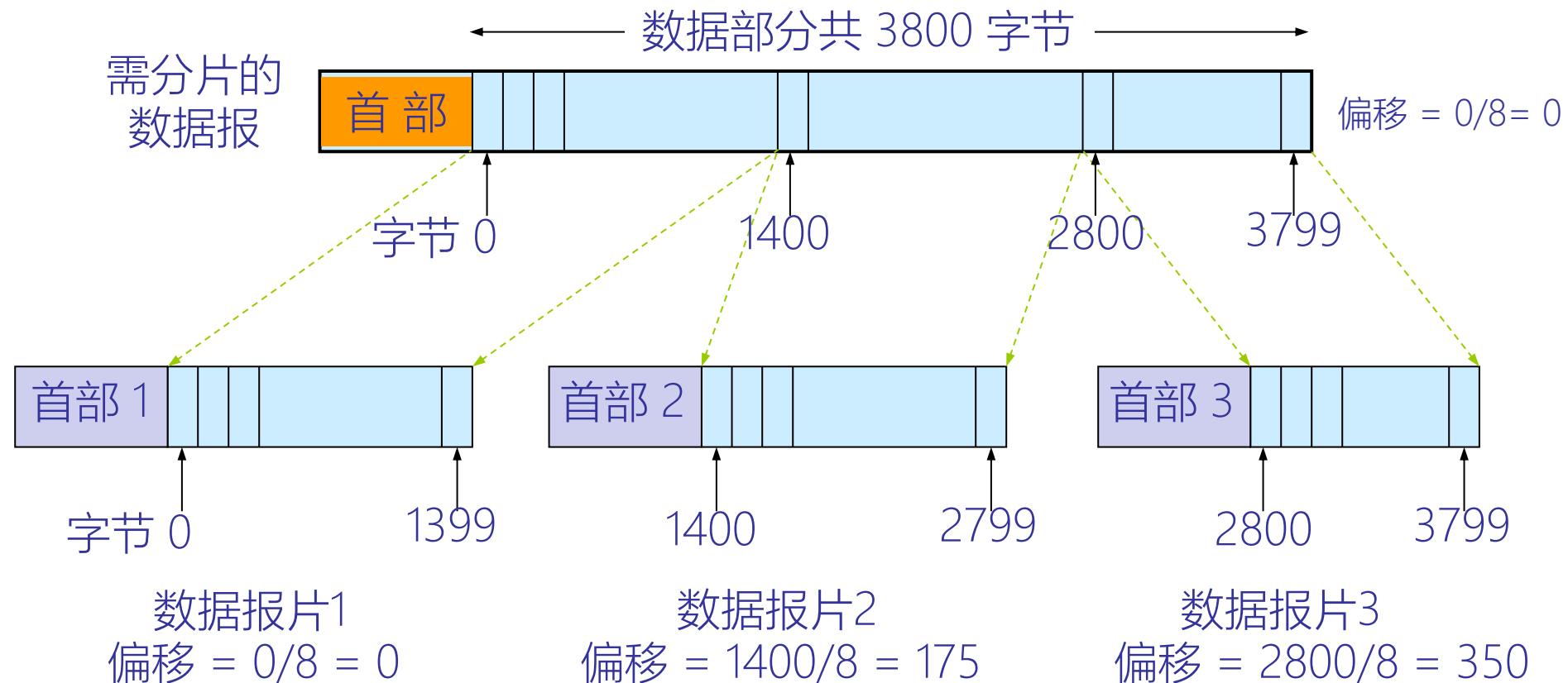
# IP报文格式

- 偏移量：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

0	3 4	7 8	15 16	18 19	31
版本号 (Version)	首部长度 (IHL)	区分服务 (Differentiated Services)		总长度 (Total Length)	
		标识 (Identification )	标志 (Flags)	分段偏移 (Fragment Offset)	
生存时间 (Time to Live)		协议 (Protocol)		首部校验和 (Header Checksum)	
			源IP地址 (Source Address)		
			目的IP地址 (Destination Address)		
≈	可选字段 (长度可变) (Options)			填充 (Padding)	≈

# 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



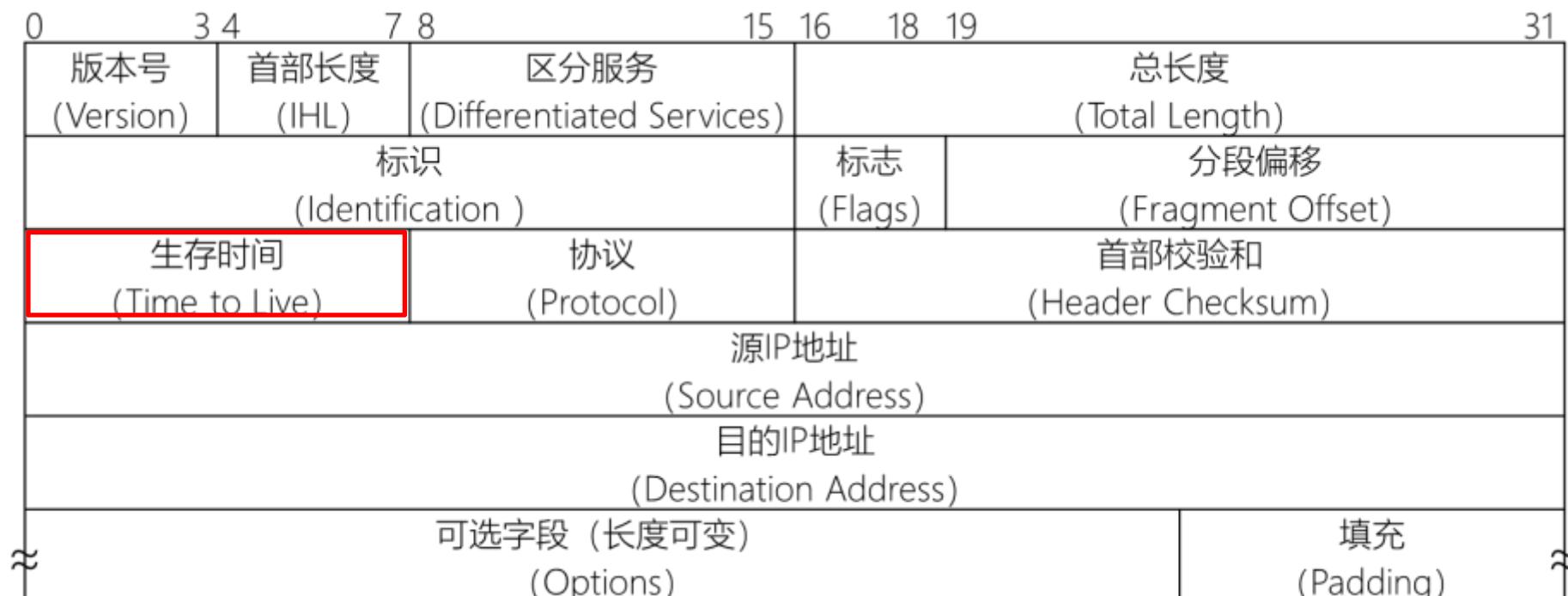
假定数据报片2经过某个网络时还需要再进行分片，分片长度不超过820字节。分片后的数据报片的总长度、标识、MF、DF和片偏移分别为是多少？

在这种情况下如何恢复原数据报？

# IP报文格式

## ■ 生存时间：设置数据报可以经过的最多路由器数量

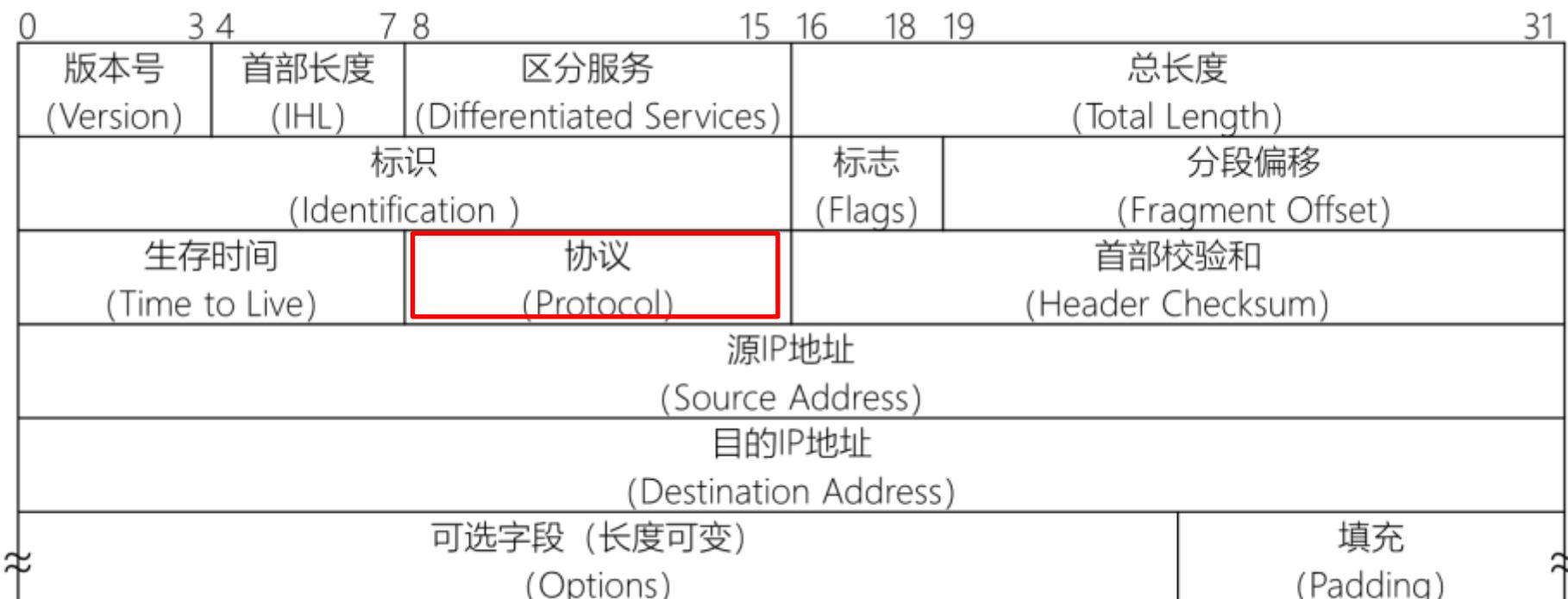
- 初始值由源主机设置，通常生存时间的起始值是32或64
- 经过一个路由器，它的值减去1
- 当该字段的值为0时，该数据报被丢弃



# IP报文格式

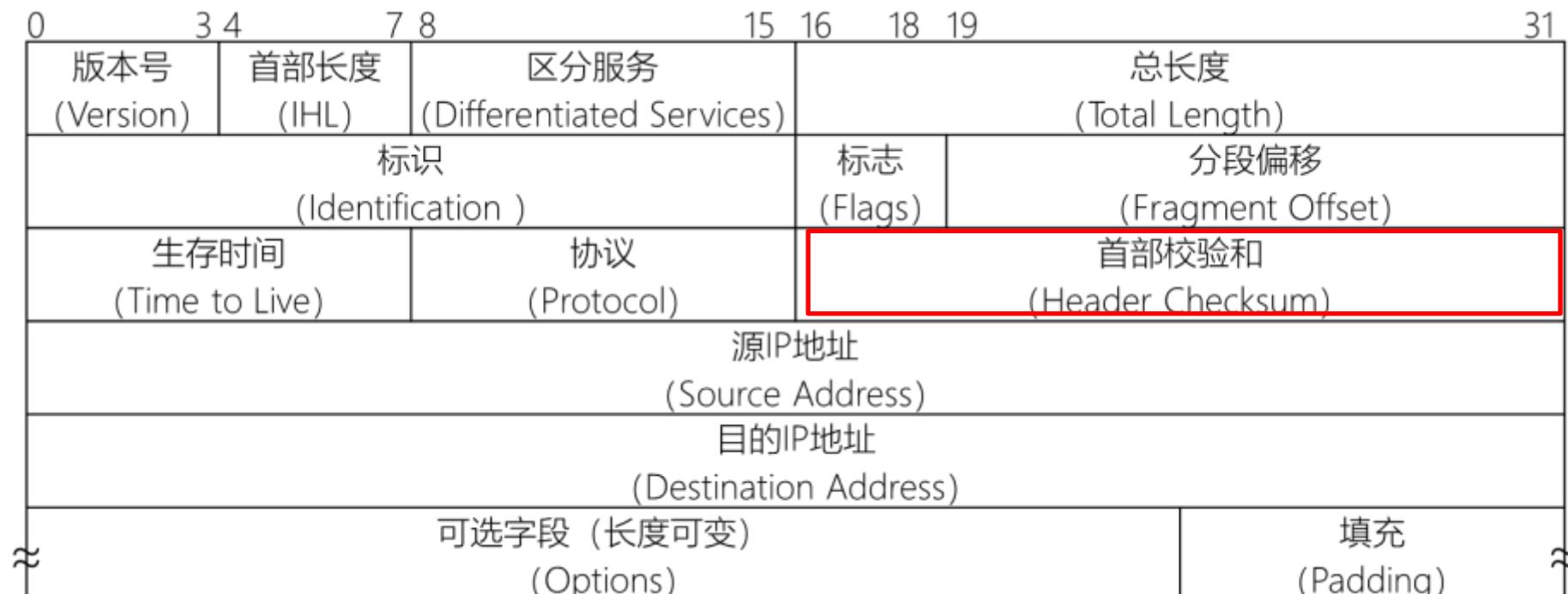
- 协议字段：上层所使用的协议。使用协议字段对数据报进行分类，根据它的值可以确定是哪个协议向IP发送数据报

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

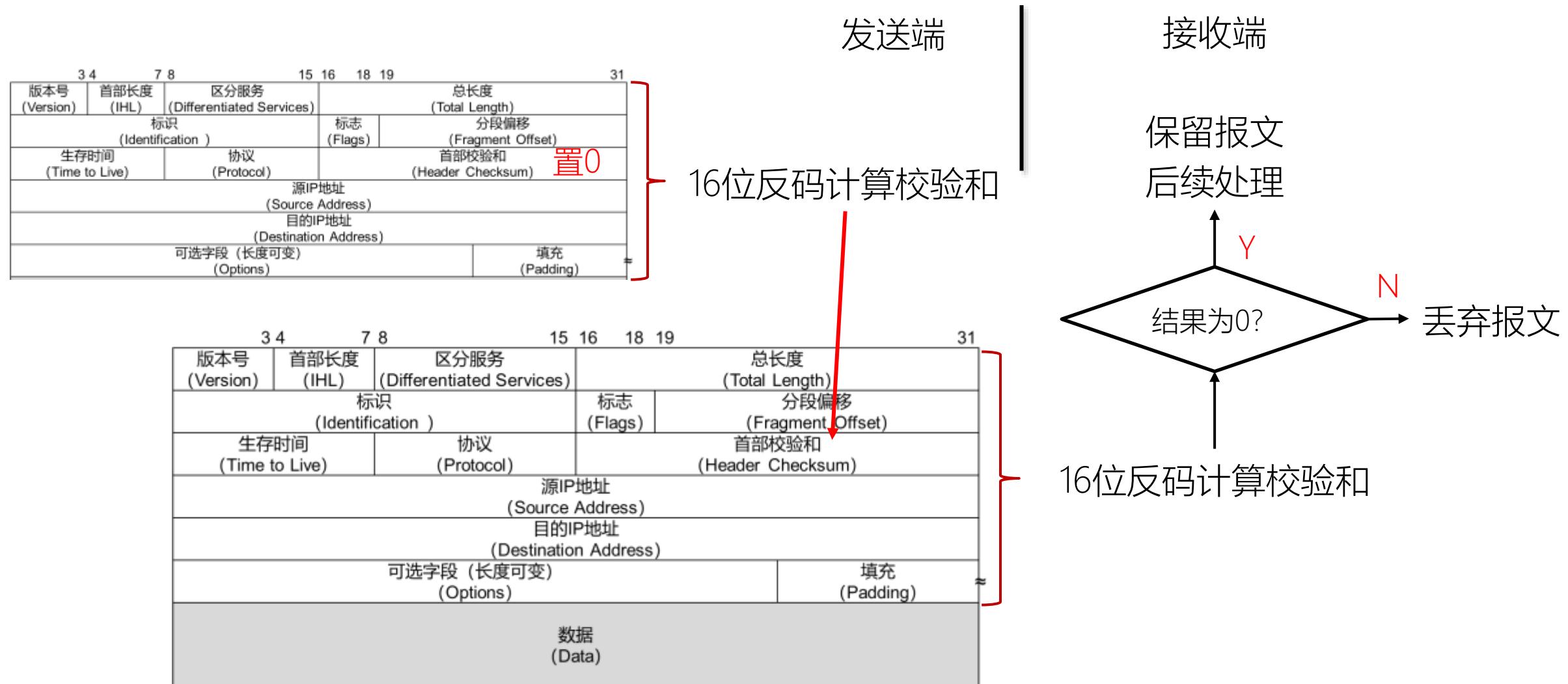


# IP报文格式

- 报头检查和：根据IP报头计算的检查和，不对报头后面的数据进行计算
  - 如果检查和错误，那么IP丢弃该数据报，但不生成差错报文，由上层去发现丢失的数据报并进行重传
  - 检验和采用16位反码求和的算法，RFC1141给出了增量式修改检查和的方法

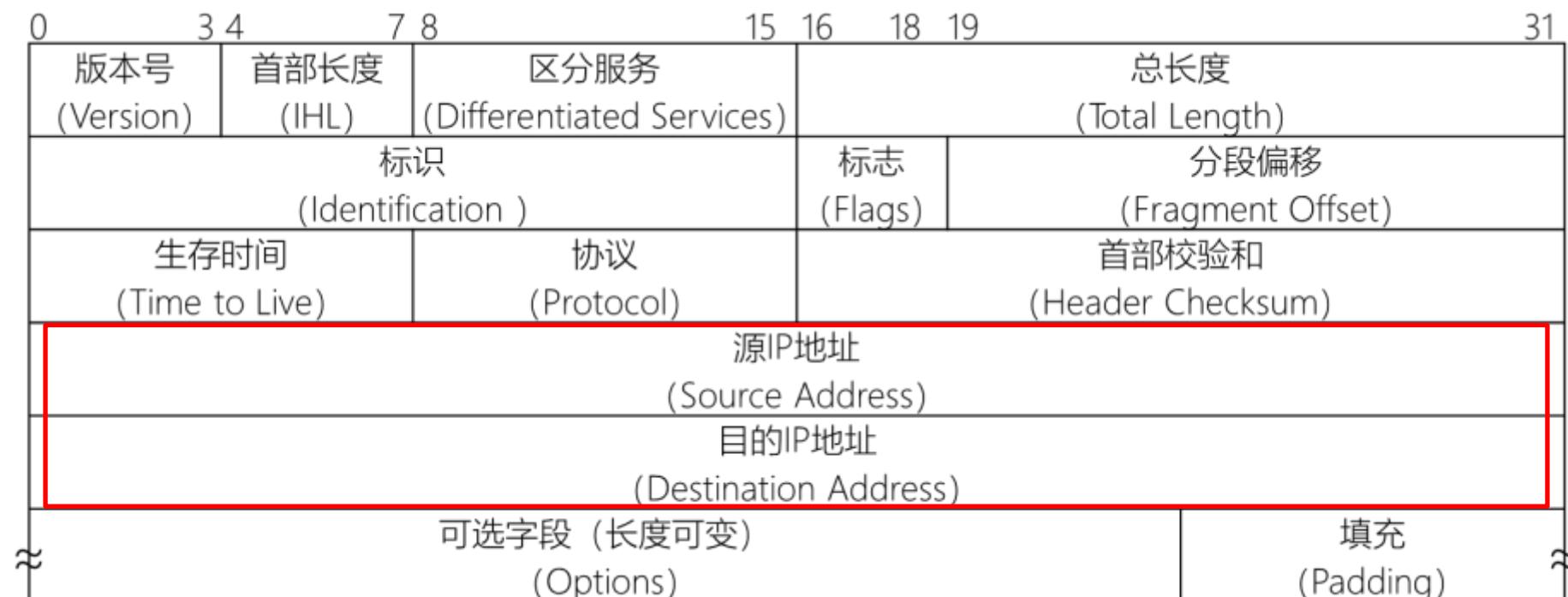


# 16位反码求和的算法



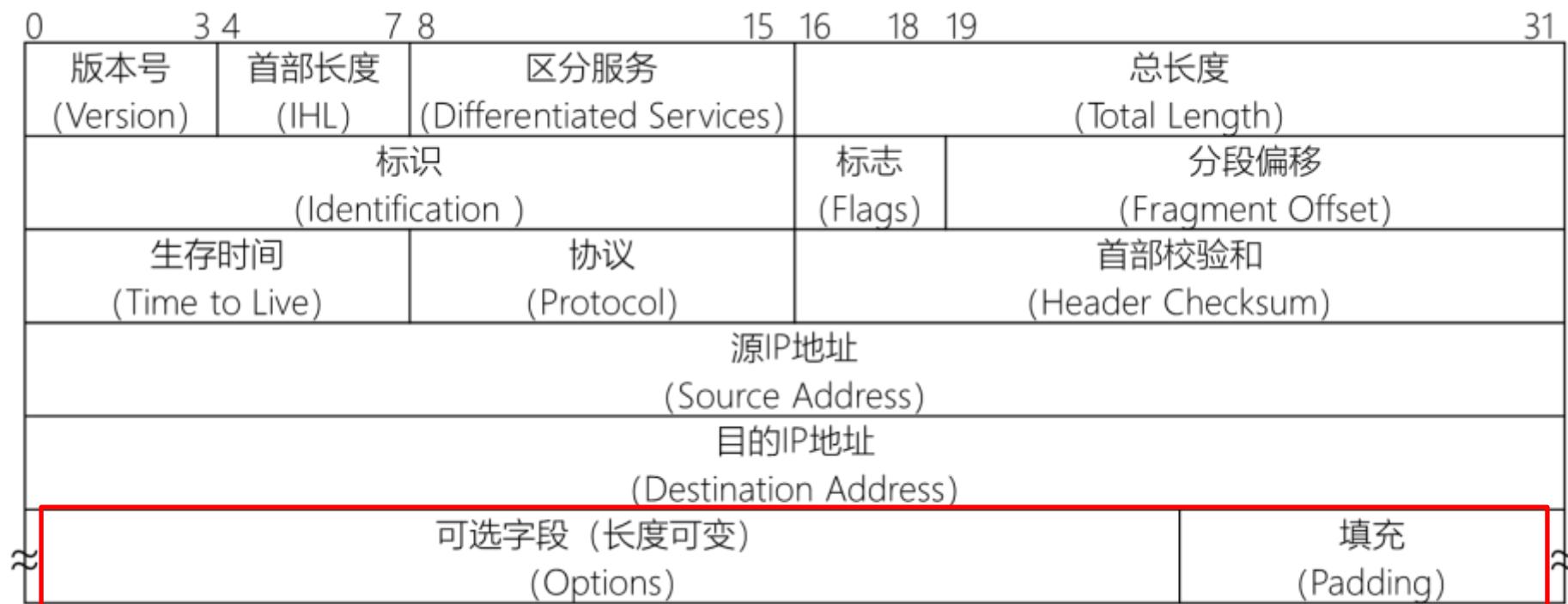
# IP报文格式

- 源IP地址和目的IP地址：各32比特
- 每个IP数据报都包含源IP地址和目的IP地址



# IP报文格式

- 可选字段：可变长度的可选信息，0~40字节。这些选项有：
  - 安全和处理限制、记录路径、时间戳、宽松的源站选路、严格的源站选路
  - IPv4协议很少使用这些选项功能（不是所有的主机和路由器都支持这些选项）
- 填充：选项字段以32比特作为界限，在必要的时候插入值为0的填充字节，保证IP首部始终是32比特的整数倍（这是首部长度字段要求的）



# 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

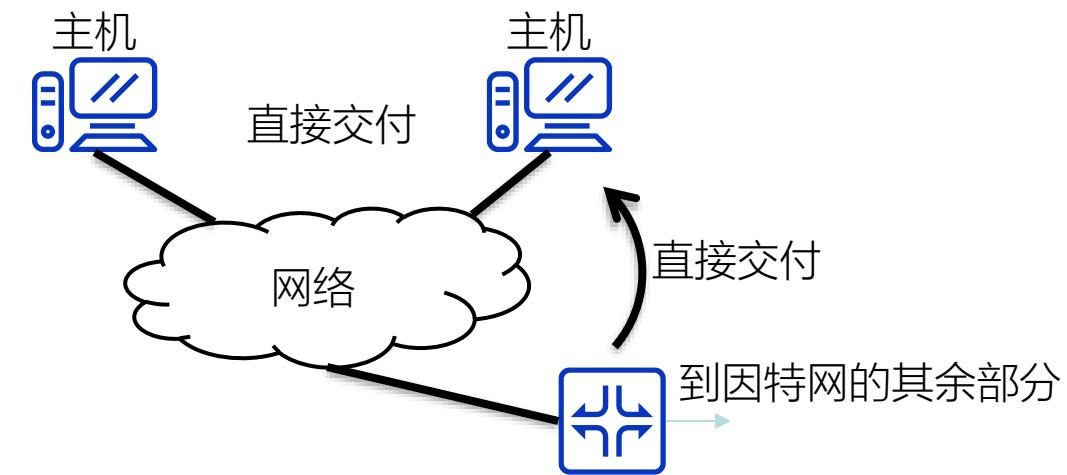
# 默认路由

---

- 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分组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

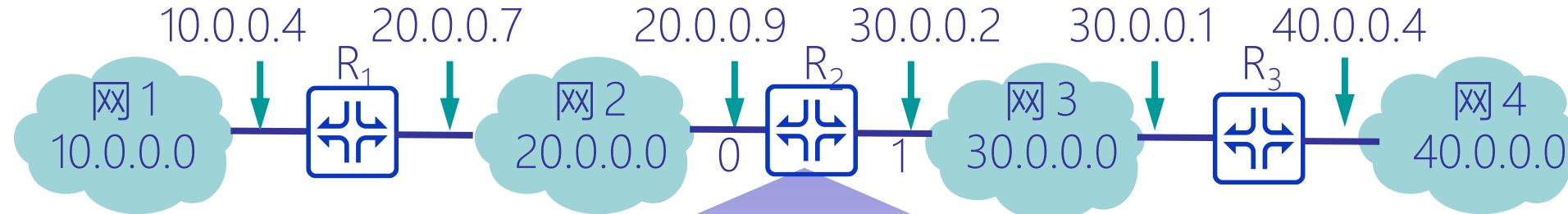
# IP交付

- 配置了IP地址后，主机之间即可  
以进行通信
- 网络可以用两种不同的方法把一  
个数据包交付到它最后的终点：
  - 直接交付
  - 间接交付(转发)



- 转发表示把分组放到去终点的路由上。进行转发要求主机或路由器具有路由表
- 路由器的主要功能是转发分组，内存中维持一个路由表。当收到一个分组并进行转发时，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、为分组传输指定一个网络接口

# IP转发



路由器 R<sub>2</sub> 的路由表

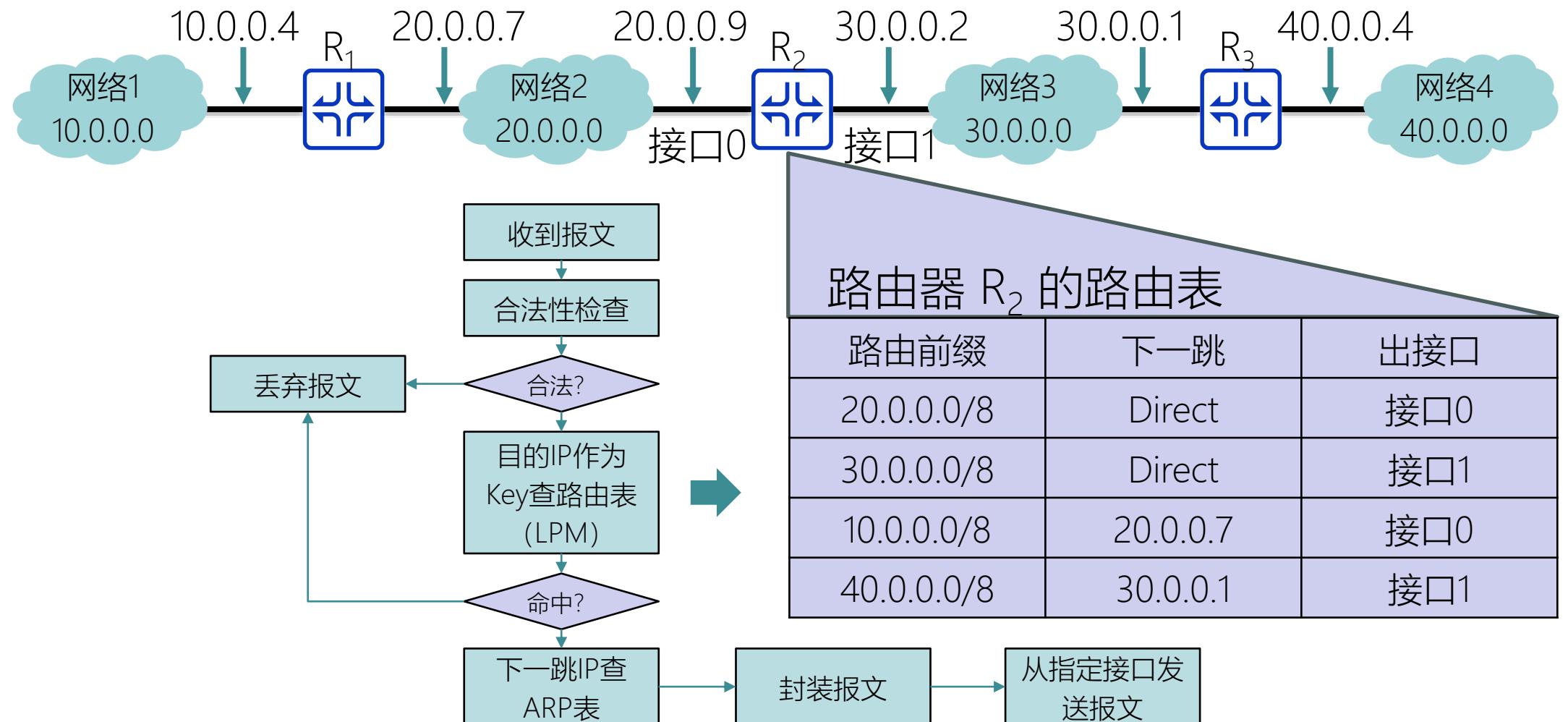
路由前缀	下一跳	出接口
20.0.0.0/8	Direct	接口0
30.0.0.0/8	Direct	接口1
10.0.0.0/8	20.0.0.7	接口0
40.0.0.0/8	30.0.0.1	接口1

- 路由器的主要功能是转发数据包，路由器内存中维护一个路由表。收到一个数据包后，提取数据包的目的IP地址作为Key，搜索路由表，基于检索结果将数据包从指定的接口转发出去

路由表的每一行应包含下面信息：

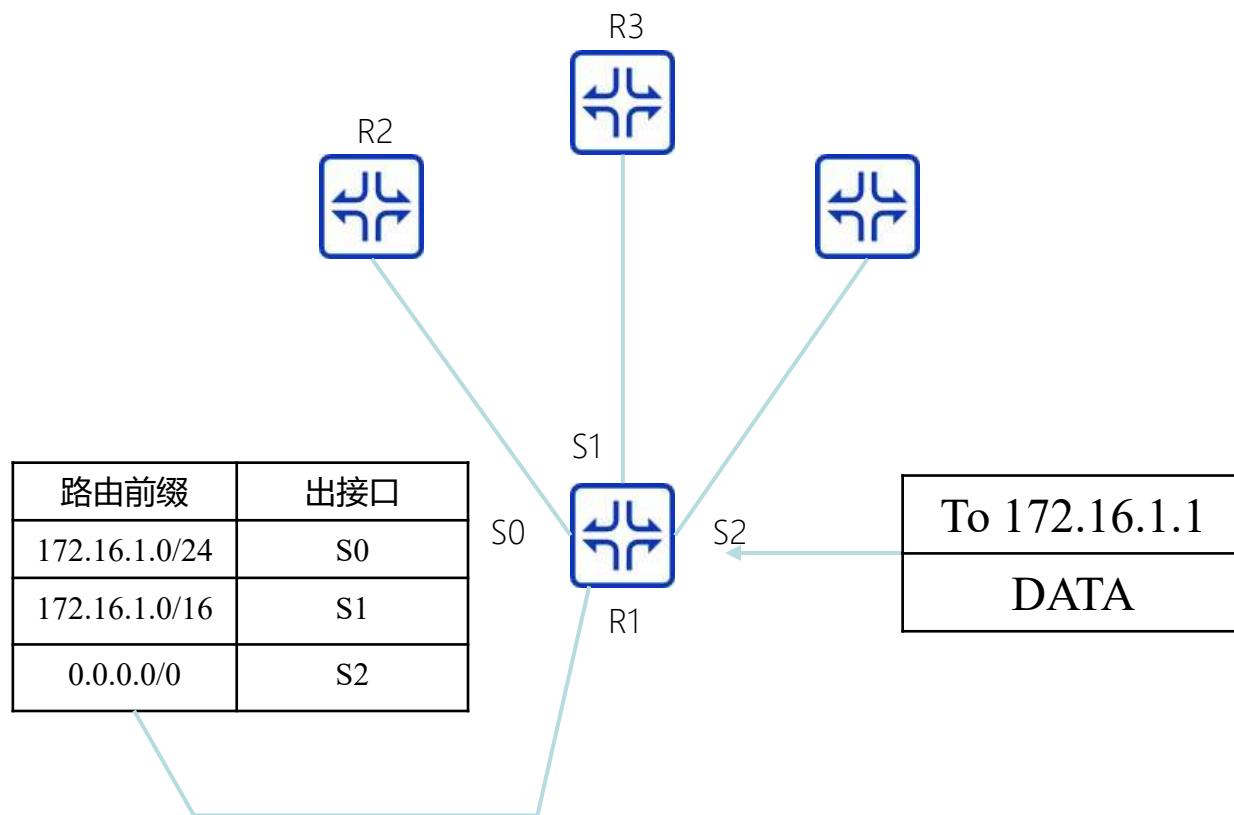
- 目的IP地址，即路由前缀
- 下一跳路由器的IP地址
- 为数据报的传输指定一个网络接口

# IP转发



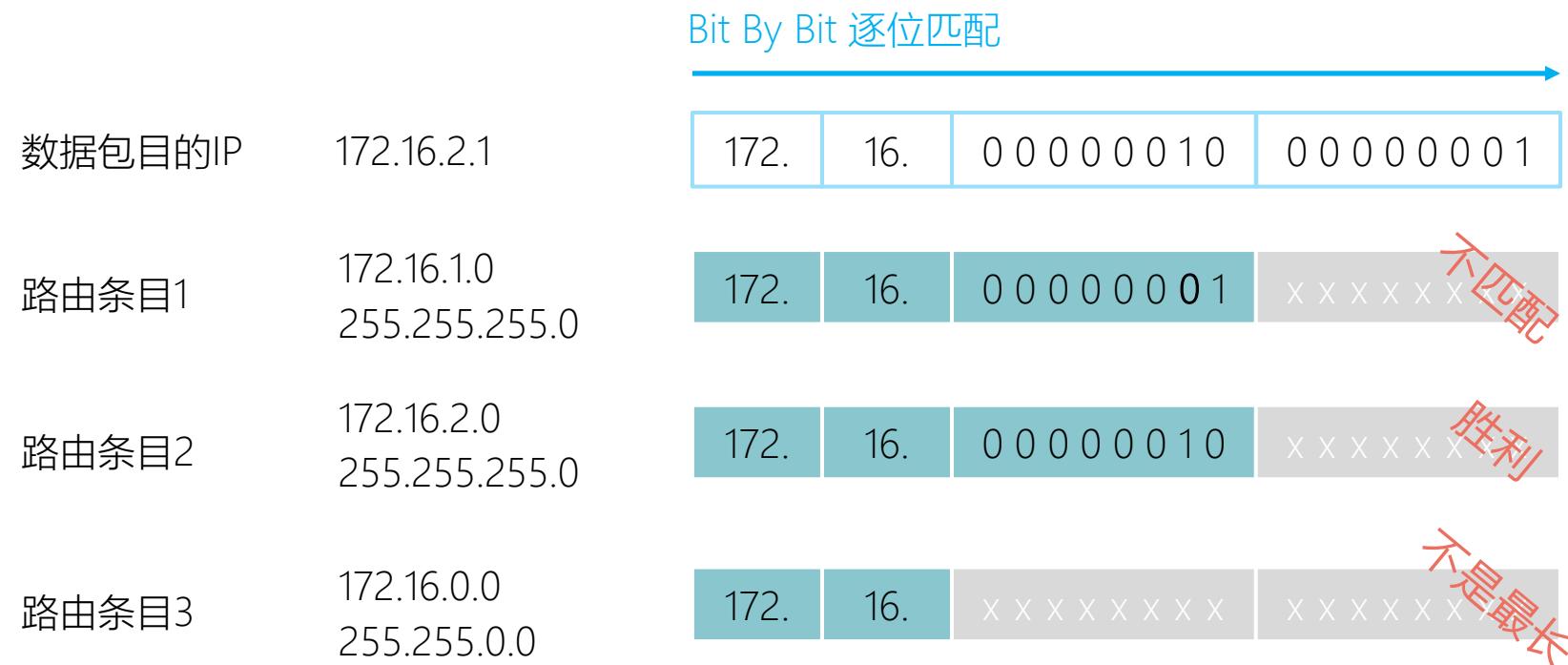
# 最长前缀匹配

- 在路由器中，路由表中的每条路由的基本信息包括“网络前缀”、“下一跳地址”和/或“出接口”组成
- 网络前缀越长，其地址块就越小，因而路由就越具体



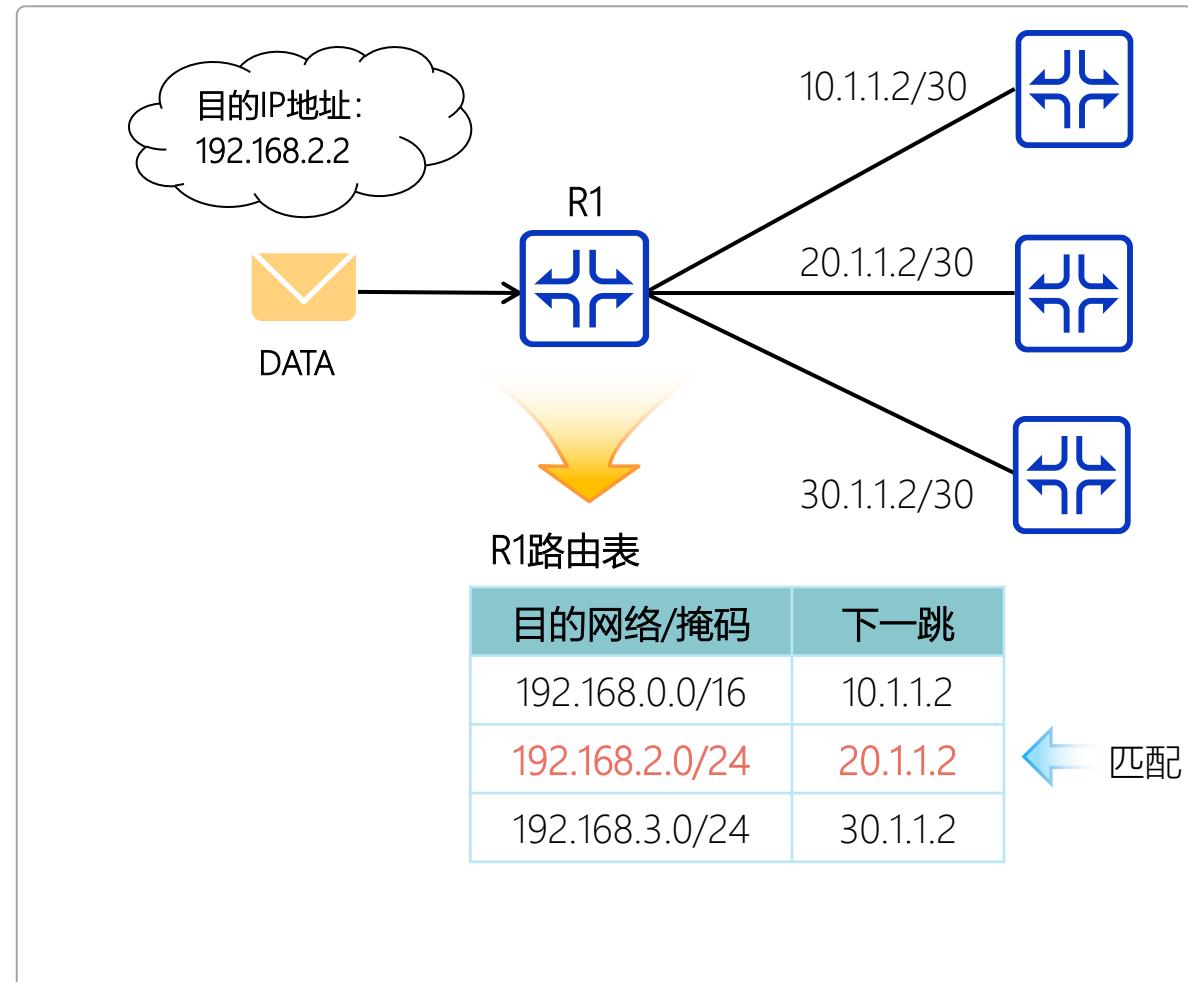
# 最长前缀匹配

- 当路由器收到一个IP数据包时，会将数据包的目的IP地址与自己本地路由表中的所有路由表项进行逐位（Bit-By-Bit）比对，直到找到匹配度最长的条目，这就是最长前缀匹配机制



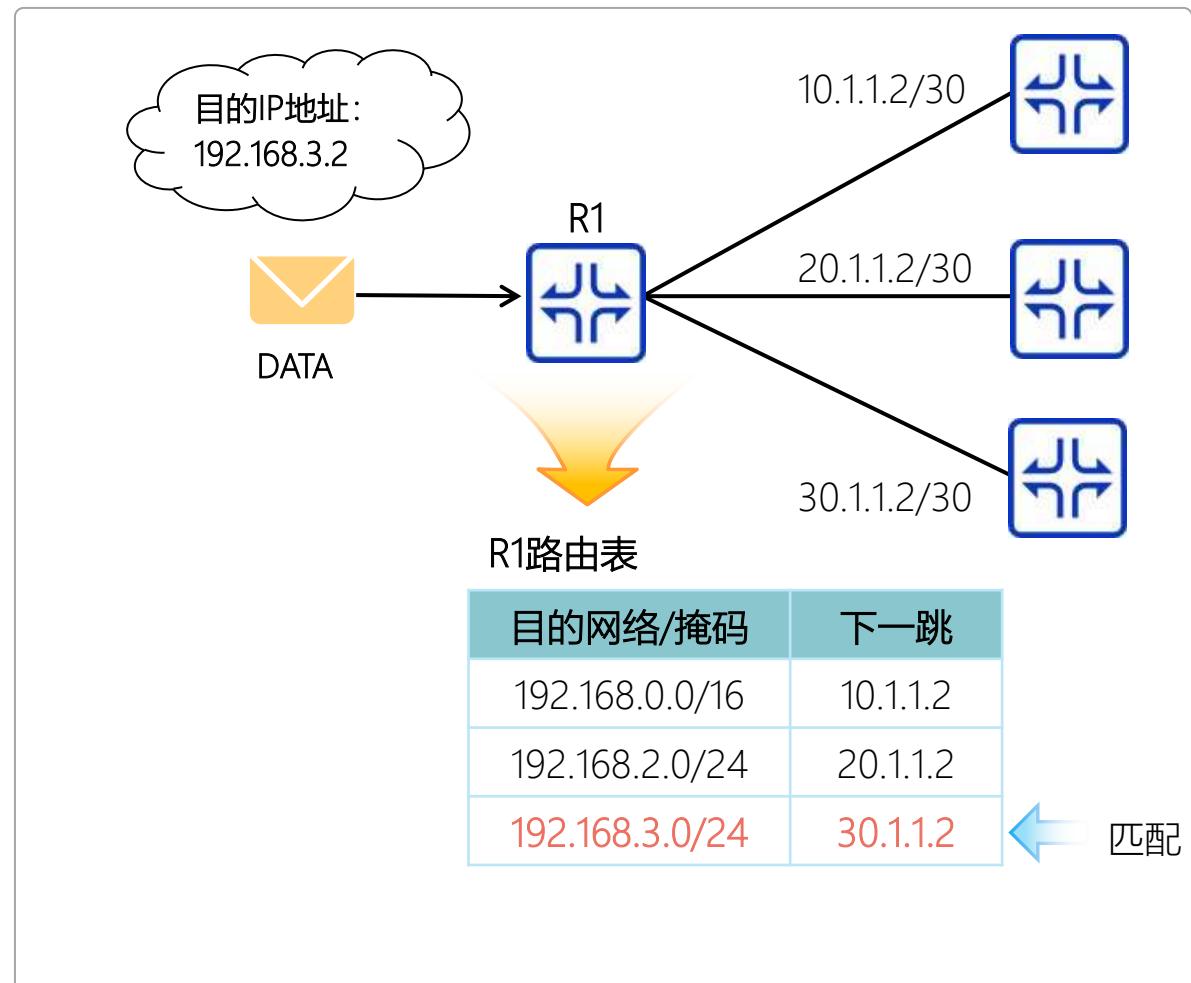
# 最长匹配示例

- 根据最长匹配原则进行匹配，能够匹配192.168.2.2的路由存在两条
- 路由的掩码长度中，一个为16 bit，另一个为24 bit
- 掩码长度为24 bit的路由满足最长匹配原则
- 被选择用于指导发往192.168.2.2的报文转发



# 最长匹配示例

- 根据最长匹配原则匹配，能够匹配到192.168.3.2的路由只有一条，此路由为最终转发依据



# 路由器转发分组的算法

1. 从分组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，则把分组直接交付目的主机 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，则把分组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，则把分组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

# ARP协议

---

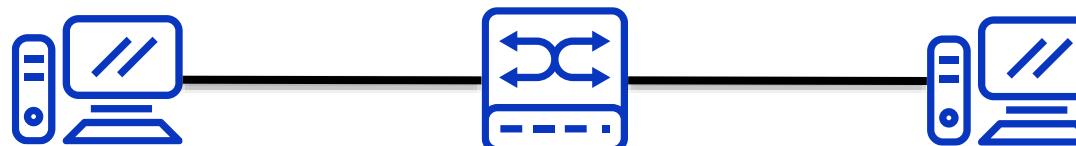
- ARP地址解析协议，就是将主机IP地址映射为硬件地址
- 在局域网中，网络中实际传输的单元是“数据帧”，数据帧的首部有目的主机的MAC地址
- 在以太网中，一个主机要和另一个主机进行直接通信，必须通过地址解析协议获得目的主机的MAC地址
- ARP协议的基本功能就是通过目的设备的IP地址，查询目的设备的MAC地址，以保证通信的顺利进行

# ARP协议

- IP协议设计的目的是将不同的物理网络，如以太网、FDDI、令牌环网、帧中继、X.25等连接在一起
- 在一个物理网络中（如以太网），当主机有数据要发送给另一台主机时，需要知道对方的网络层地址，即IP地址
- IP报文必须封装成帧才能通过物理网络发送，发送方需要知道接收方的物理地址（在以太网中是MAC地址），这就需要一个通过IP地址获取物理地址的协议，以完成从IP地址到MAC地址的映射
- 在以太网络中，地址解析协议ARP即用于实现将IP地址解析为MAC地址

# 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
- 主机或三层网络设备上会维护一张ARP表，用于存储IP地址和MAC地址的映射关系，一般ARP表项包括动态ARP表项和静态ARP表项



Host-1

172.16.1.10/16

0025-9301-0001

Host-2

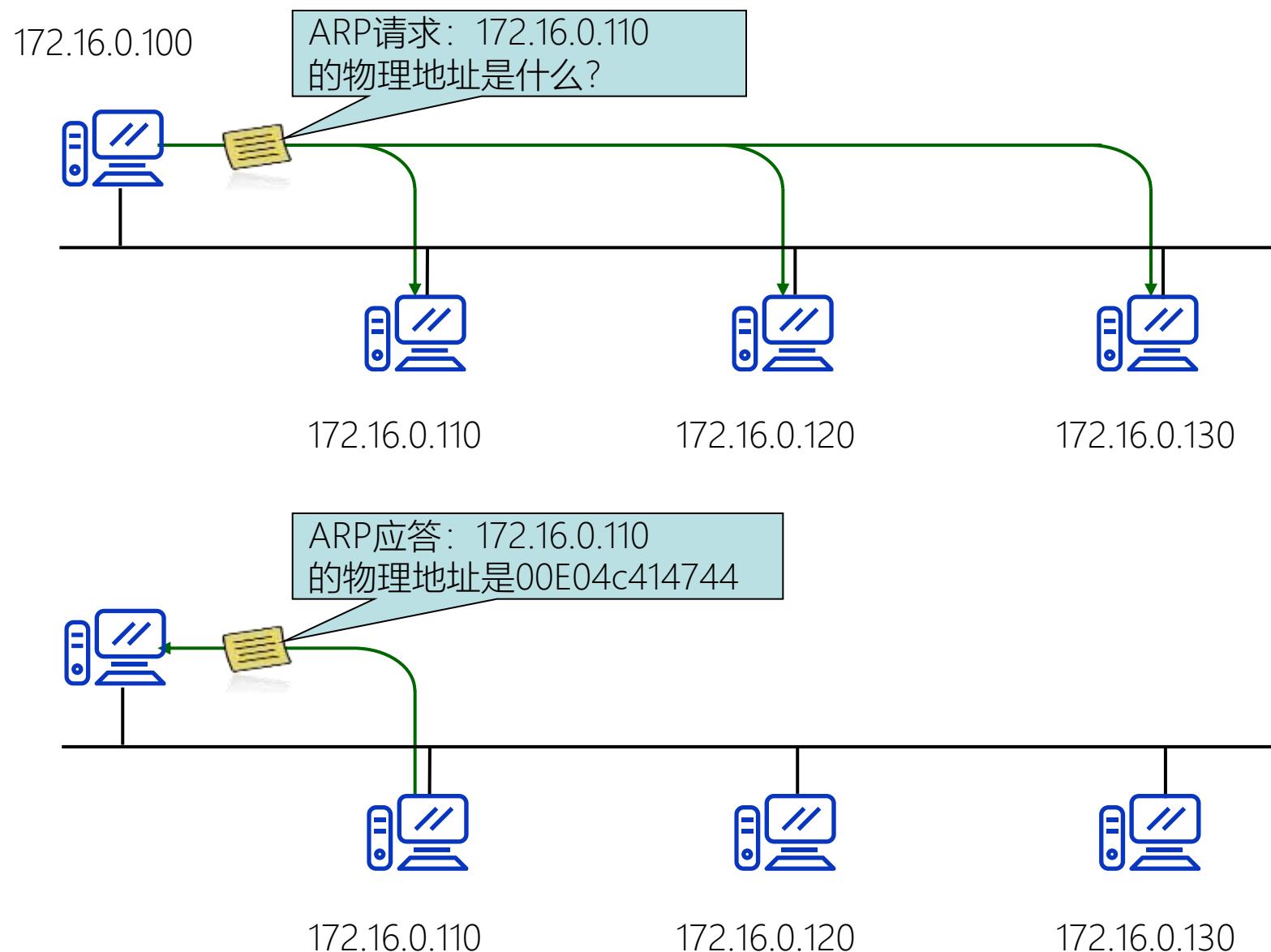
172.16.1.20

0025-9301-0002

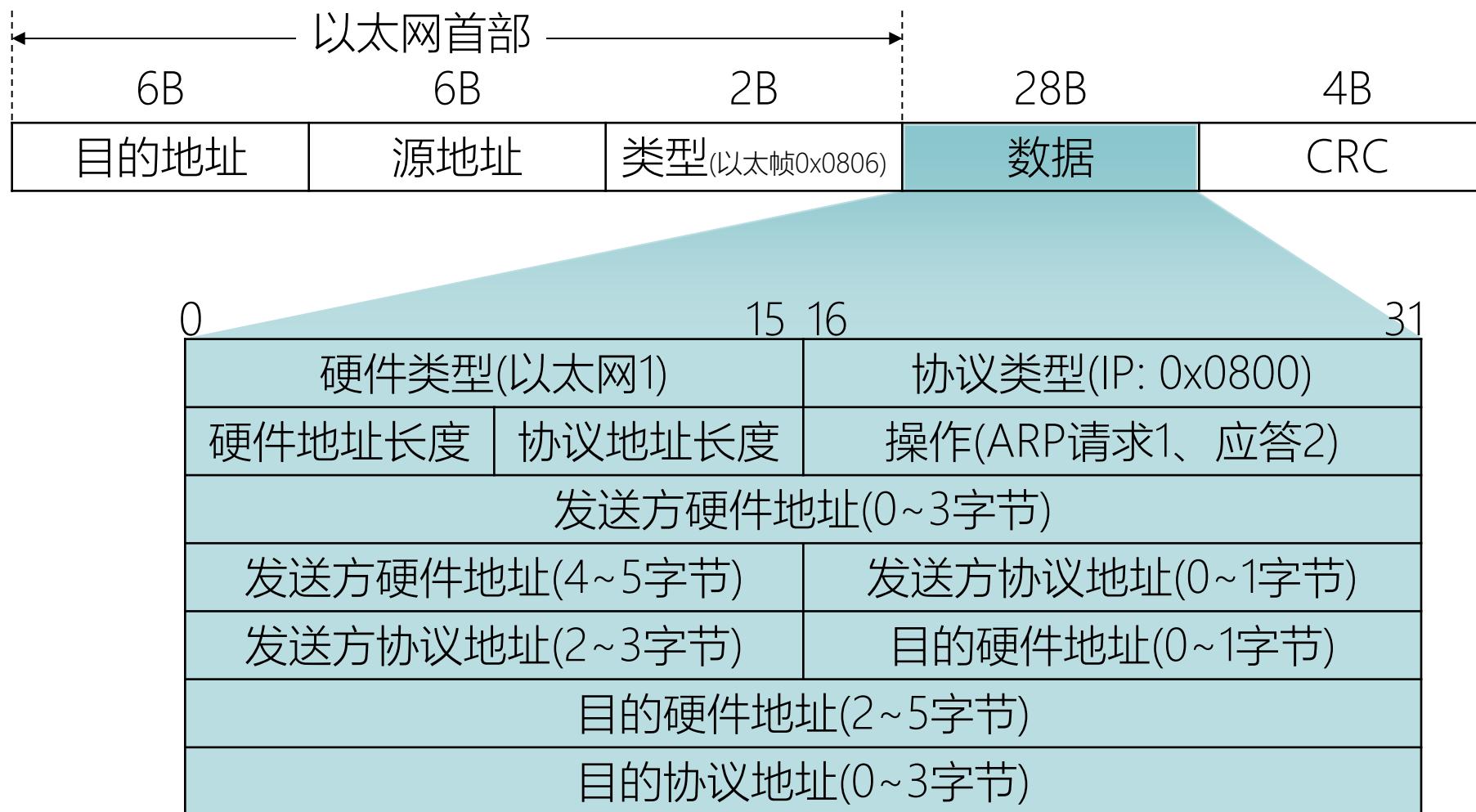
IP地址	MAC地址
172.16.1.20	0025-9301-0002

IP地址	MAC地址
172.16.1.10	0025-9301-0001

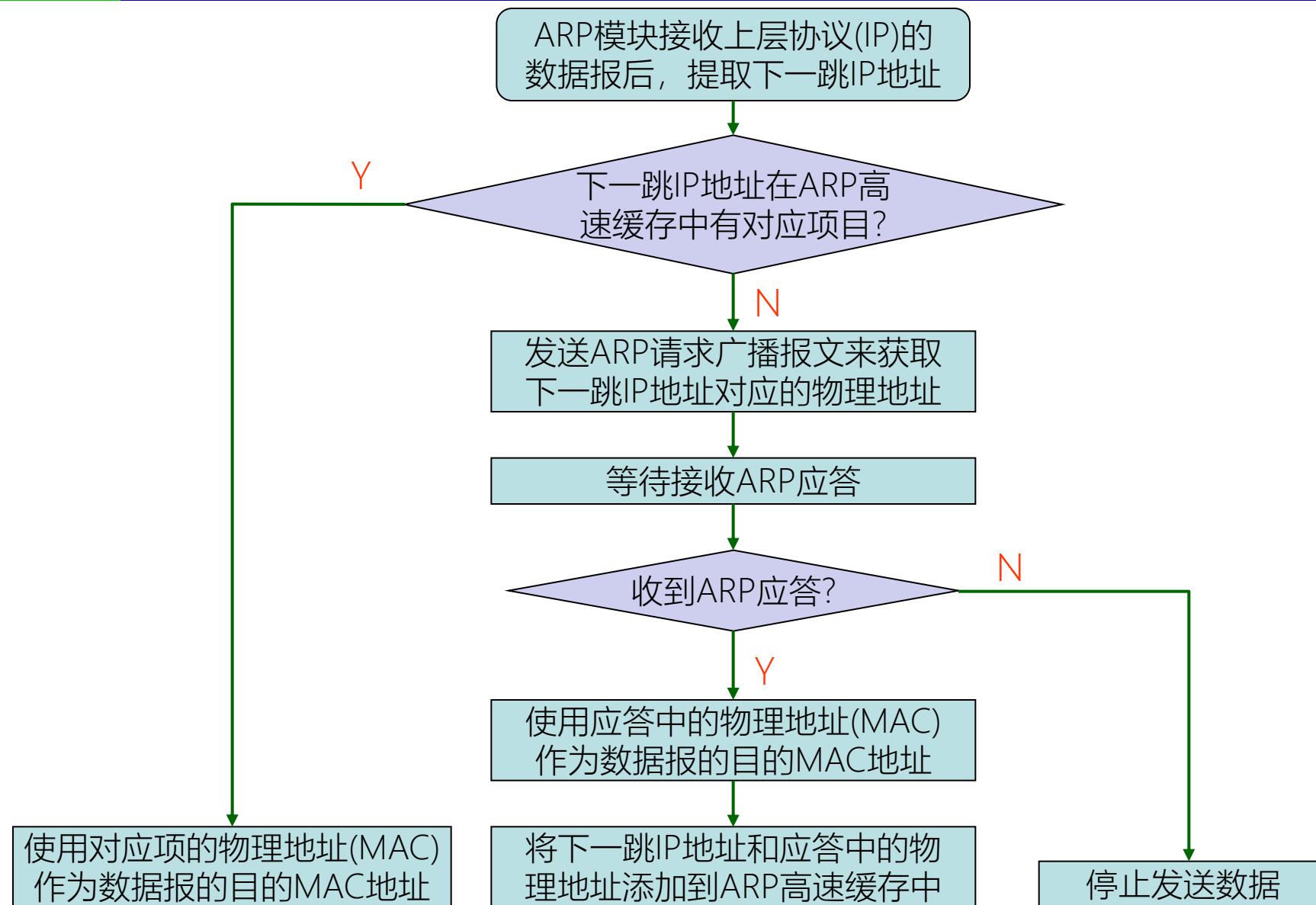
# ARP工作示例



# ARP封装与报文格式



# ARP工作流程



# 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IP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类型：
  - 动态ARP：动态ARP表项由ARP协议通过ARP报文自动生成和维护，可以被到期删除，可以被新的ARP报文更新，也可以被静态ARP表项覆盖
  - 静态ARP：静态ARP表项是由网络管理员手工建立的IP地址和MAC地址之间固定的映射关系。静态ARP表项不会被到期删除，不会被动态ARP表项覆盖

# 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、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
  - 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

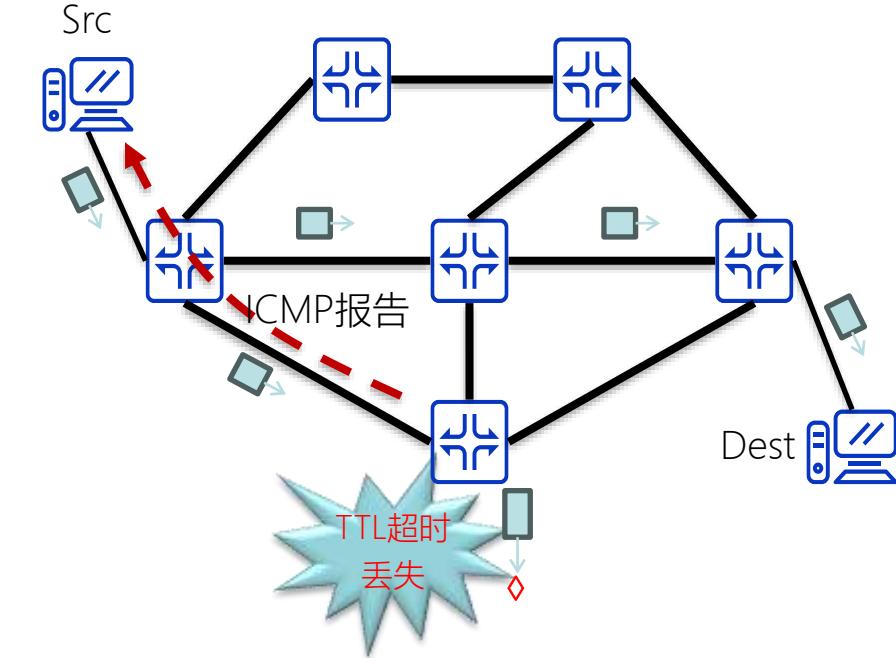
```
C:\>arp -a
接口: 192.168.0.17 --- 0xe
      Internet 地址      物理地址      类型
      192.168.0.1      3c-06-a7-8b-5e-3f    动态
      192.168.0.3      18-f2-2c-e0-54-27    动态
      192.168.0.255    ff-ff-ff-ff-ff-ff    静态
      224.0.0.2        01-00-5e-00-00-02    静态
      224.0.0.22       01-00-5e-00-00-16    静态
```

# 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

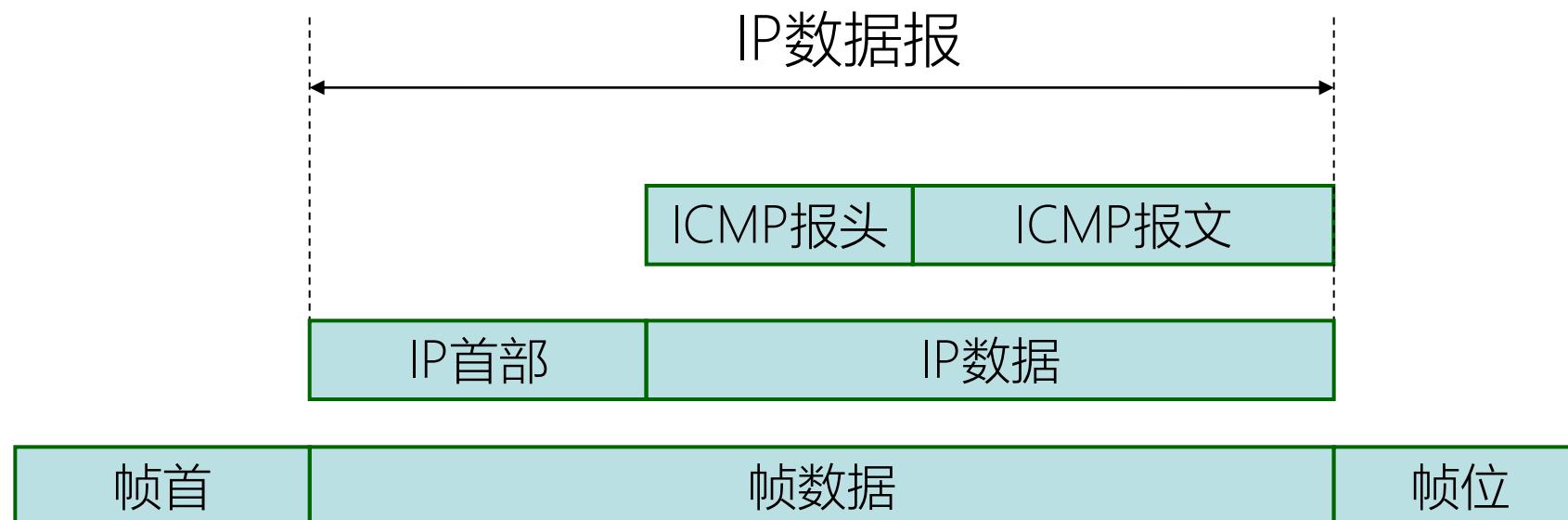
# ICMP

- 在数据传输的过程中，IP提供尽力而为的服务，不对目的主机是否收到数据包进行验证，无法进行流量控制和差错控制
- 数据包在网络中传输时，产生各种错误在所难免，如目的主机不响应、包拥塞和丢失等
- 为处理这些问题，更有效地转发IP数据包和提高数据包交付成功的机会，在IP层引入了ICMP协议
- 使用ICMP，当网络中数据包传输出现问题时，主机或设备就会向上层协议报告差错情况和提供有关异常情况的报告，使得上层协议能够通过自己的差错控制程序来判断通信是否正确，以进行流量控制和差错控制，从而保证服务质量



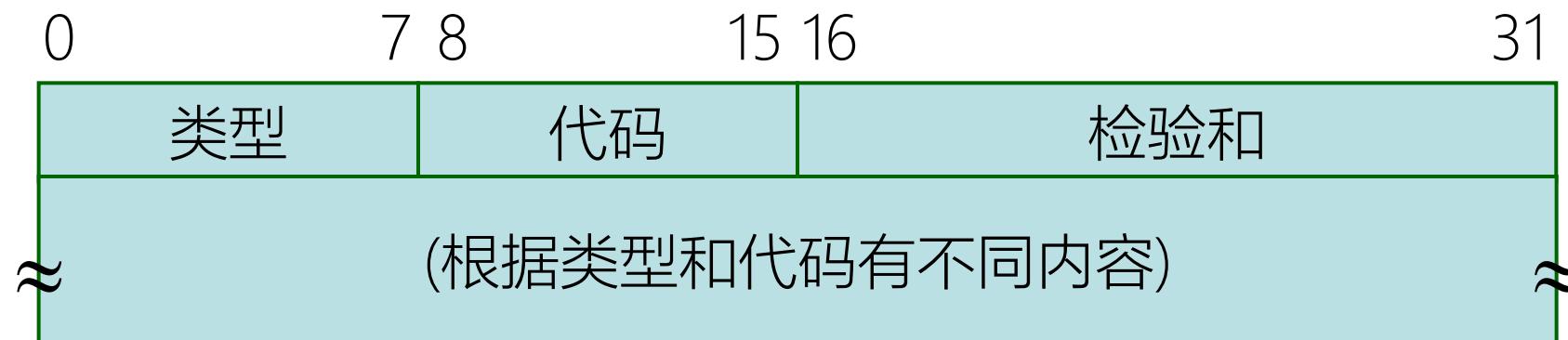
# ICMP报文格式和类型

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

# 检查目的站的可达性

---

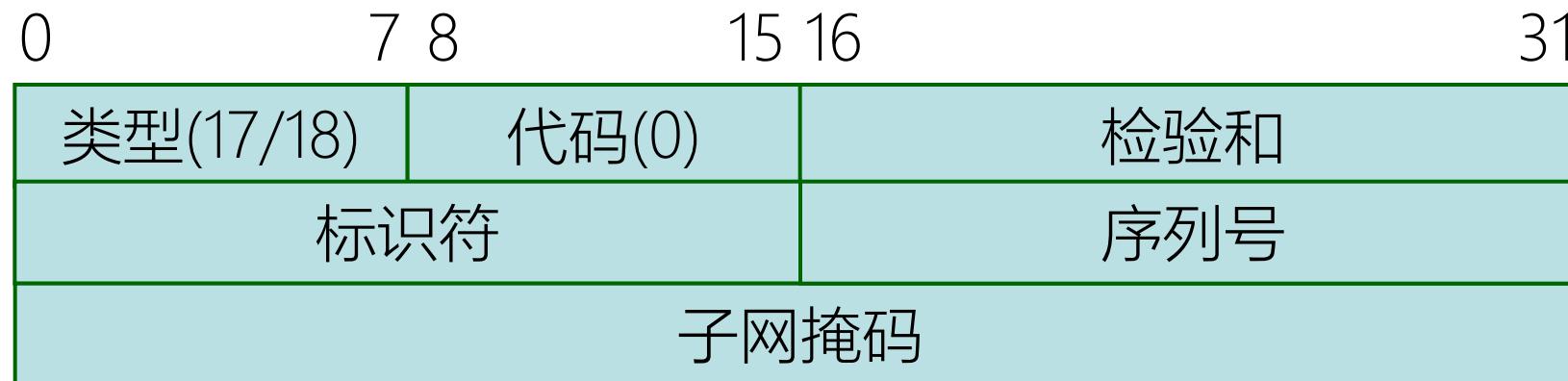
- 为了诊断目的而设计：
  - 源主机IP软件要为数据报选路
  - 源—目的站之间的路由器必须正在运行，且正确为数据报选路
  - 目的主机必须正在运行，且ICMP和IP软件都在工作
  - 返回路径上的路由器必须有正确的路由
- 最常用的调试工具是利用回送请求和回送回答报文来测试目的站的可达性

# 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

- 向局域网上的路由器发送地址掩码请求报文，得到主机的掩码
  - 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

# ICMP典型应用—ping程序

- Ping程序是最常见的用于检测网络设备是否可达的调试手段
  - 远程设备是否可达
  - 与远程主机通信的来回旅程（round-trip）的延迟
  - 报文包的丢失情况
- 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

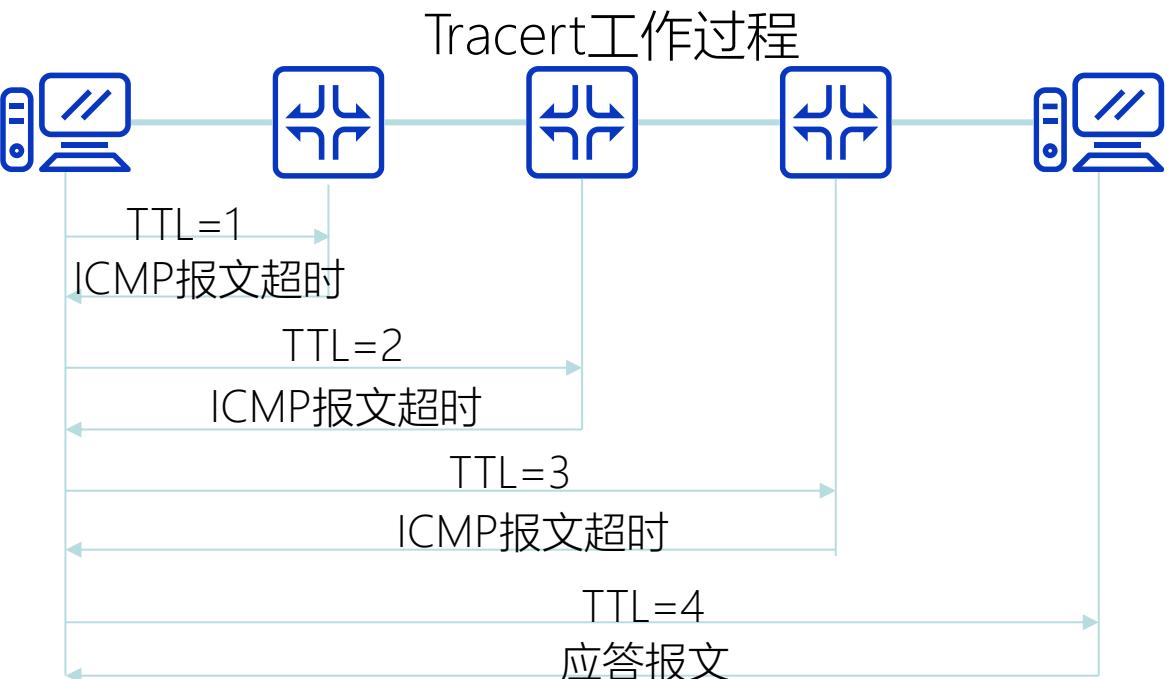
```
C:\>ping www.cisco.com
Pinging e144.ca.s.tl88.net [203.110.166.170] with 32 bytes of data:
Reply from 203.110.166.170: bytes=32 time=46ms TTL=48
Reply from 203.110.166.170: bytes=32 time=44ms TTL=48
Reply from 203.110.166.170: bytes=32 time=48ms TTL=48
Reply from 203.110.166.170: bytes=32 time=45ms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

# ICMP典型应用—tracert程序

- 主要用于查看数据包从源端到目的端的路径信息，从而检查网络连接是否可用。当网络出现故障时，用户可以使用该命令定位故障点
- 利用ICMP超时信息和目的不可达信息来确定从一个主机到网络上其他主机的路由，并显示IP网络中每一跳的延迟
- 利用了ICMP协议的请求回显，并巧妙的利用了TTL值来获得路径信息

```
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.
```



# 互联网组管理协议

- 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(multicast)处于单播和广播之间：报文仅传送给属于多播组的多个主机
- 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多播组成部分

- 多播编址方法
  - 多播组用D类IP地址(224.0.0~239.255.255.255)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

28位

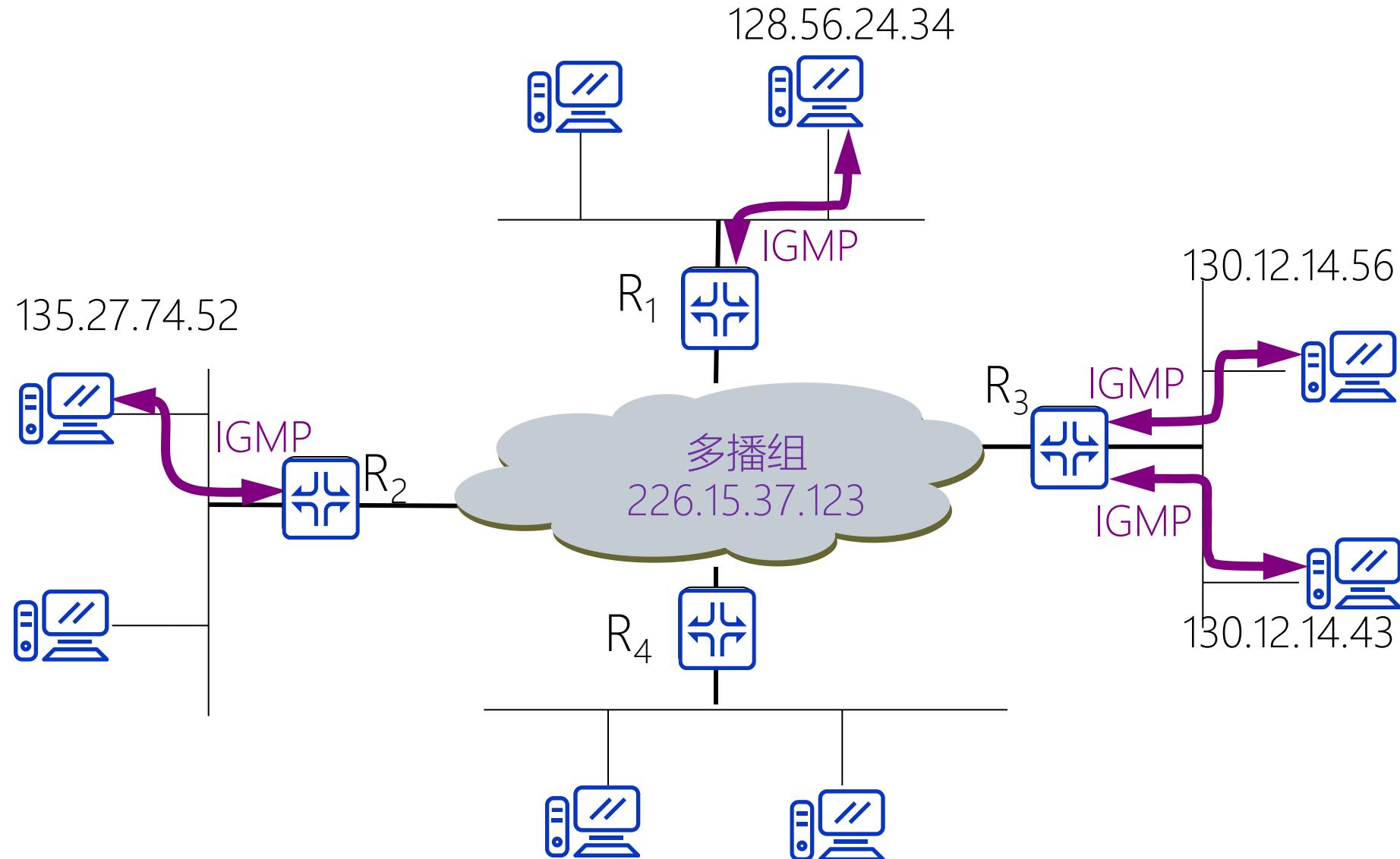


- 多播地址划分

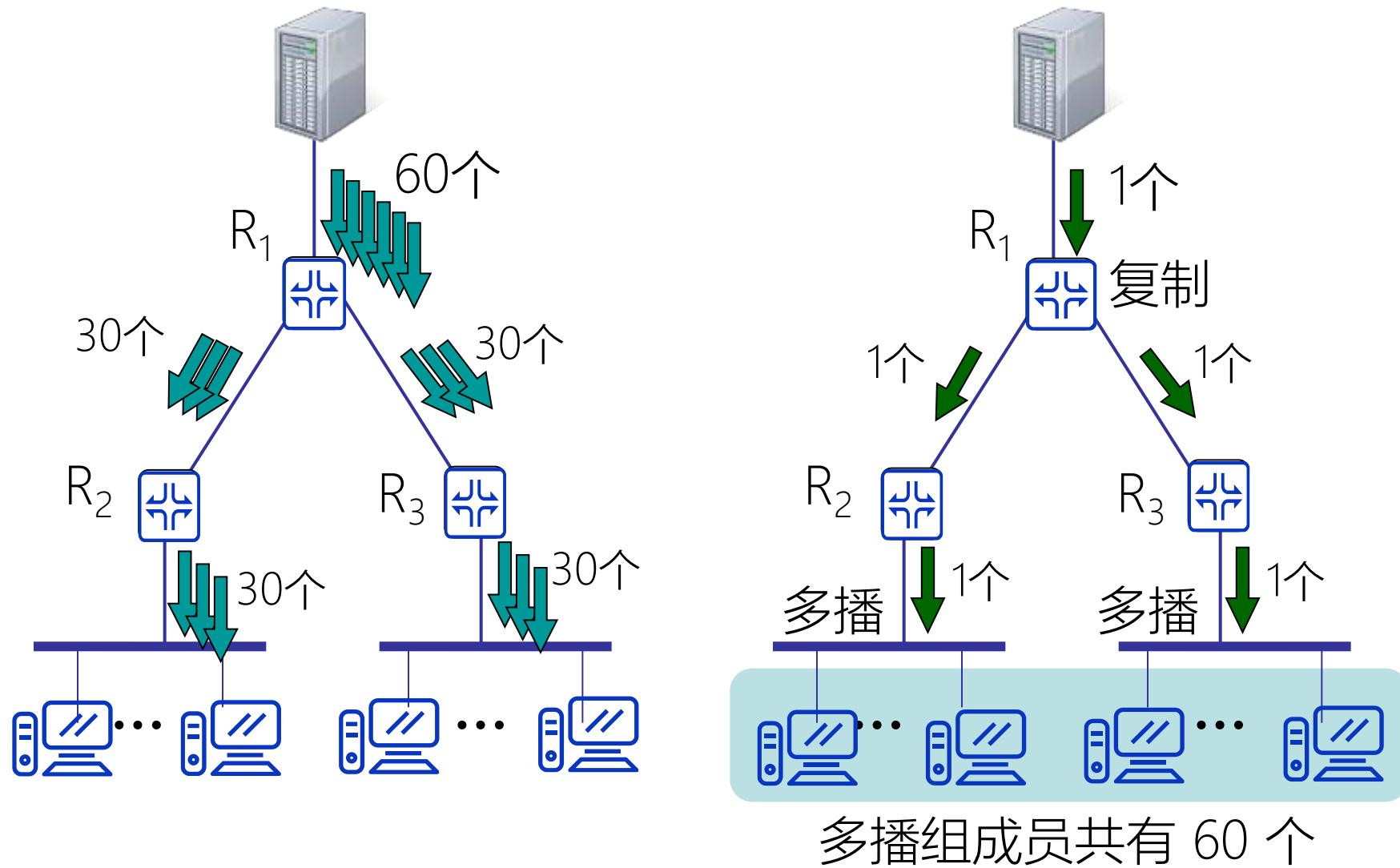
范围	用途
224.0.0.0~224.0.0.255	IANA预留
224.0.1.0~238.255.255.255	用户多播地址，全网有效
239.0.0.0~239.255.255.255	本地管理多播地址，特定范围内有效

协议	组播IP地址	组播MAC地址
RIP V2	224.0.0.9	01:00:6e:00:00:09
OSPF V2	224.0.0.5	01:00:6e:00:00:05
	224.0.0.6	01:00:6e:00:00:06
EIGRP	224.0.0.10	01:00:6e:00:00:0a

# 多播路由器

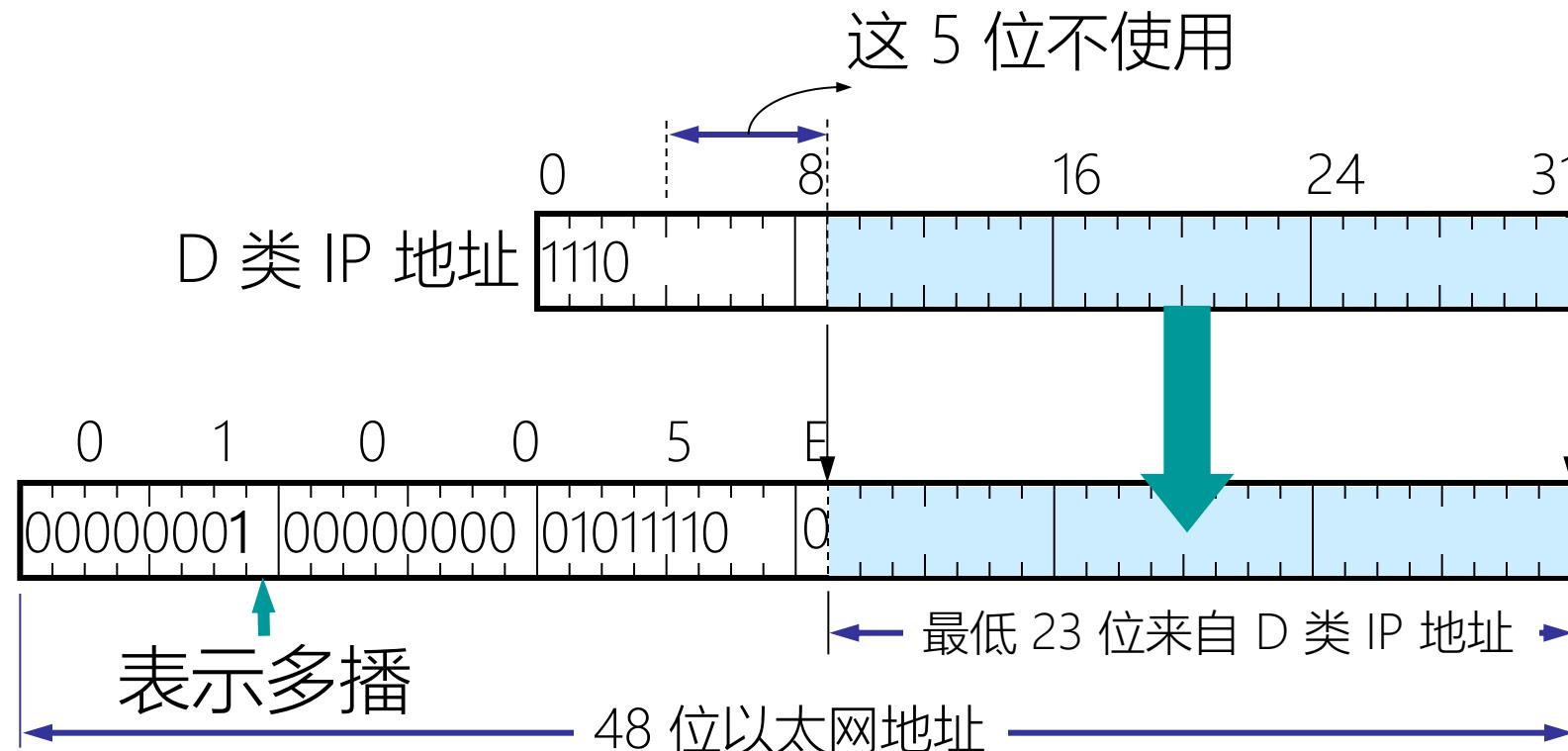


# 单播与多播的比较



# 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：

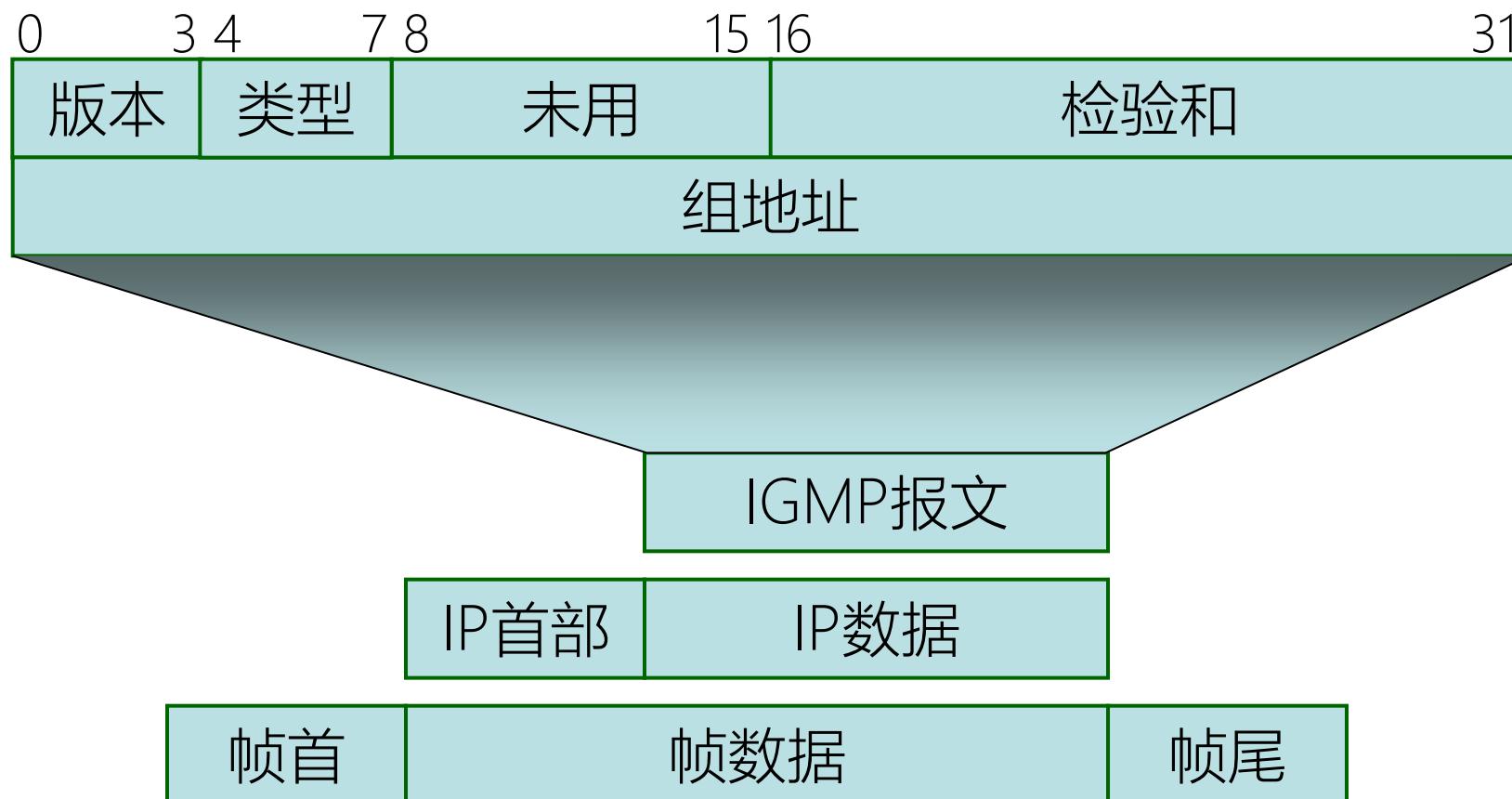
10000000 00101111 01010001

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

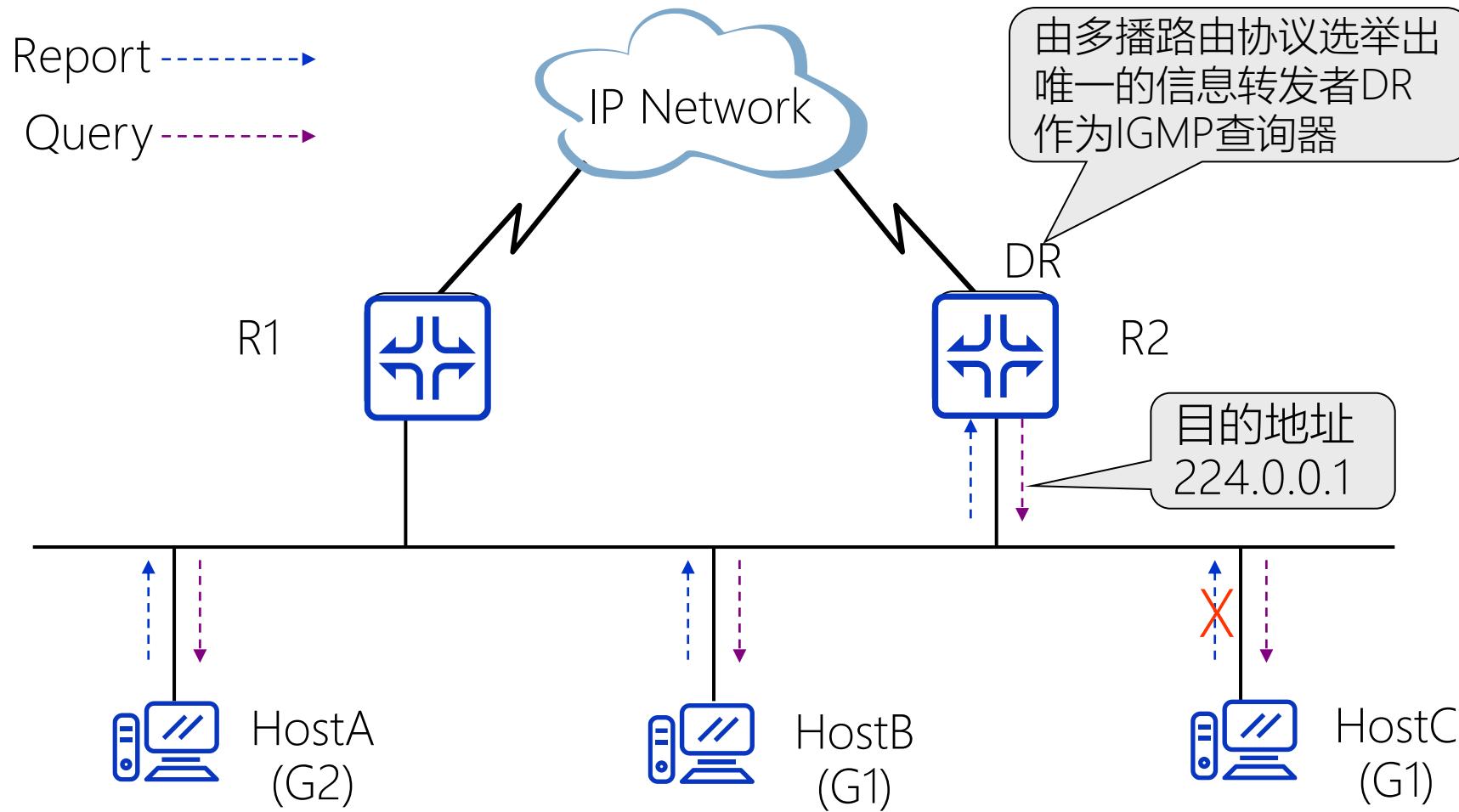
# IGMP工作机制

- 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—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

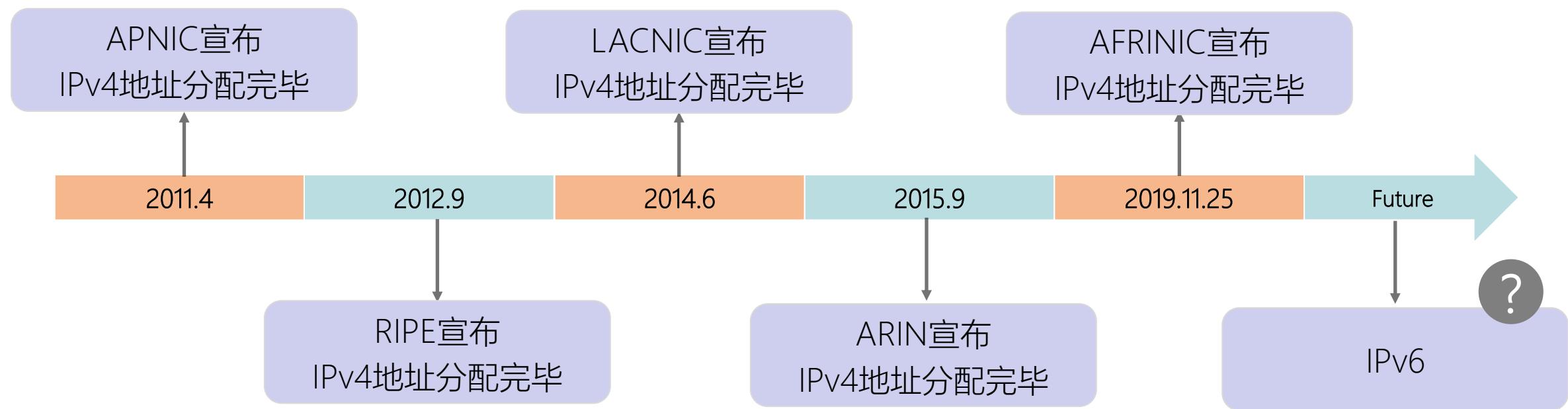
# 多播工作机制示例(IGMP v1)



IGMP v1主要基于查询和响应机制完成对多播组成员的管理

## 5.3 IPv6协议

- 2011年2月3日，IANA (Internet Assigned Numbers Authority) 宣布将其最后的468万个IPv4地址平均分配到全球5个RIR (Regional Internet Registry)，此后IANA再没有可分配的IPv4公网地址资源



# IPv6的优势

## “无限”地址空间

地址长度为128 bit，海量的地址空间，满足物联网等新兴业务、有利于业务演进及扩展

## 层次化的地址结构

IPv6地址的分配更加规范，利于路由聚合（缩减IPv6路由表规模）、路由快速查询

## 即插即用

IPv6支持无状态地址自动配置（SLAAC），终端接入更简单

## 简化的报文头部

简化报文头，提高效率；通过扩展报文头支持新应用，利于网络设备转发处理

## 安全特性

IPsec、真实源地址认证等保证端到端安全；避免NAT破坏端到端通信的完整性

## 移动性

对移动网络实时通信有较大改进，整个移动网络性能有比较大的提升

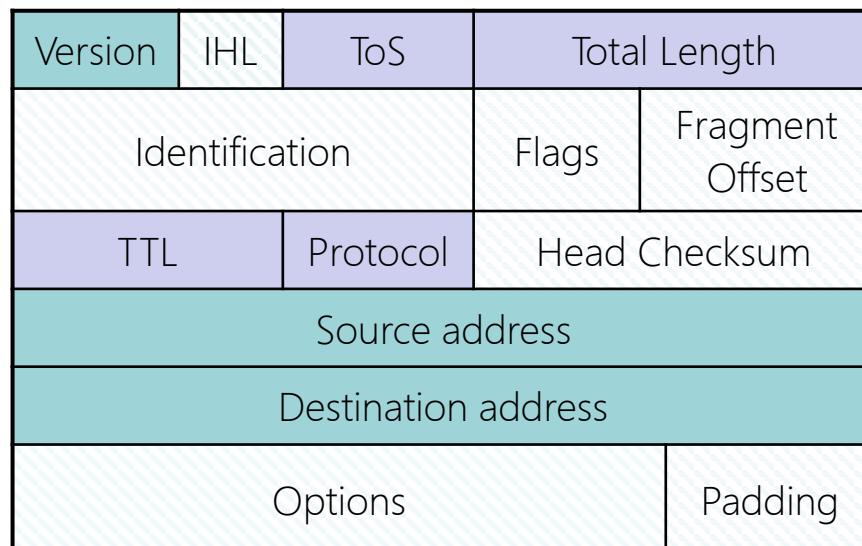
## 增强的QoS特性

额外定义了流标签字段，可为应用程序或者终端所用，针对特殊的服务和数据流，分配特定的资源

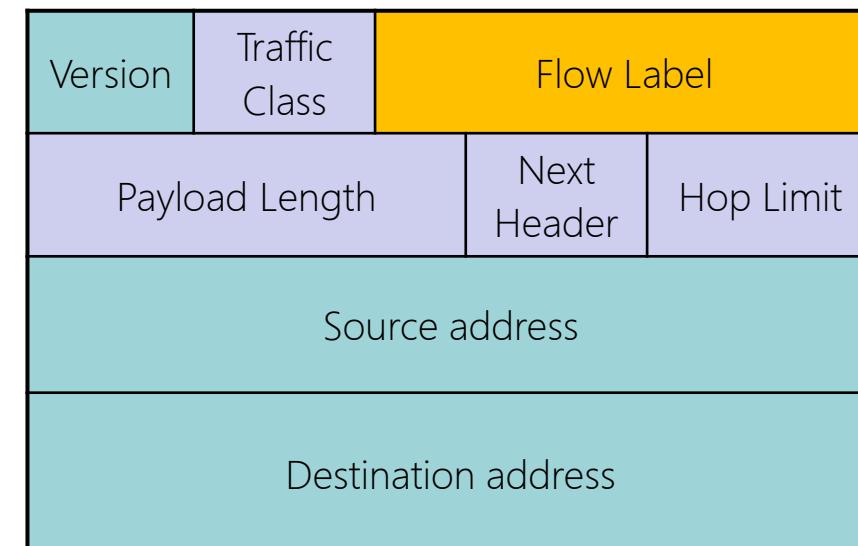
# IPv6基本头部

- IPv6报文头部由一个IPv6基本头部（必须存在）和多个扩展头部（可选）组成
- 基本头部提供报文转发的基本信息，会被转发路径上的所有设备解析

IPv4报文头部 (20 Byte ~ 60 Byte )



IPv6基本头部 (40 Byte)



保留的字段

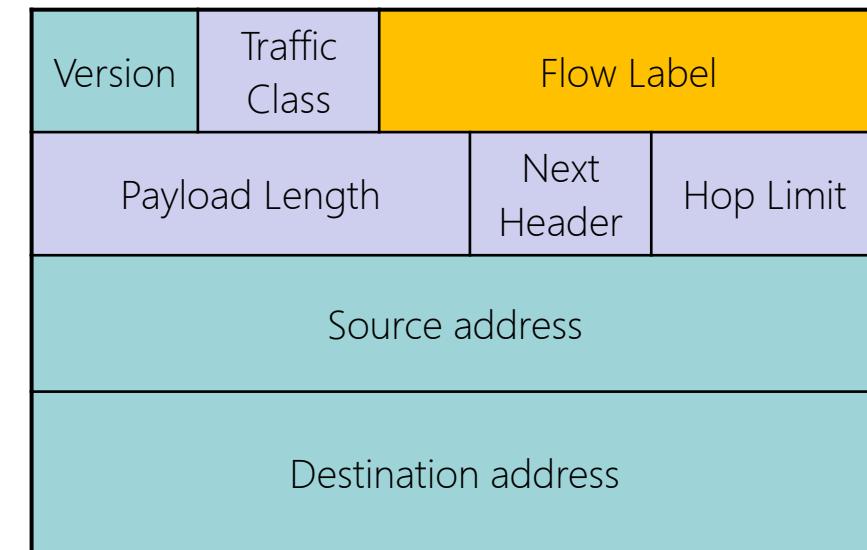
取消的字段

名字/位置变化

新增字段

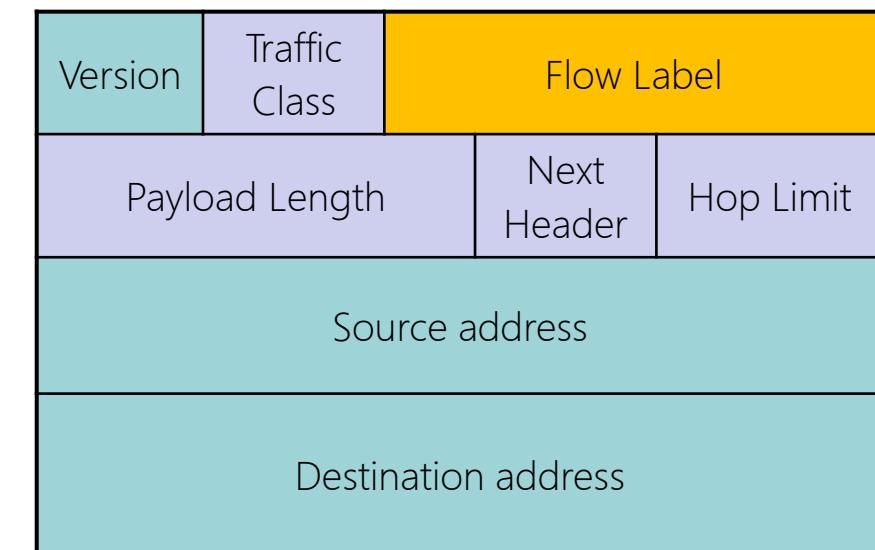
# IPv6基本头格式

- Version: 版本号(4 bit)
  - 对于IPv6，该值为6
- Traffic Class: 流类别(8 bit)
  - 类似IPv4中的DS字段
  - 表示IPv6数据包的类或优先级，主要应用于QoS
- Flow Label: 流标签(20 bit)
  - 用于区分实时流量，不同的流标签+源地址可以唯一确定一条数据流，中间网络设备可以根据这些信息更加高效率的区分数据流



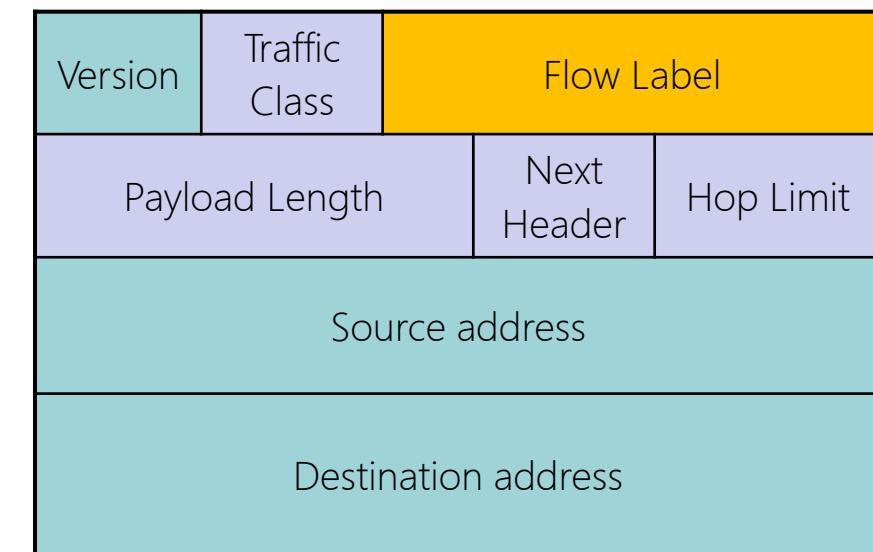
# IPv6基本头格式

- Payload Length: 有效载荷长度(16 bit)
  - 有效载荷是指紧跟IPv6包头的数据包的其它部分（即扩展报文头和上层协议数据单元）
- Next Header: 下一个报文头(8 bit)
  - 类似于IPv4的Protocol字段
  - 该字段定义紧跟在IPv6报文头后面的第一个扩展报文头（如果存在）的类型，或者上层协议数据单元中的协议类型
- Hop Limit: 跳数限制(8 bit)
  - 该字段类似于IPv4中的TTL
  - 它定义了IP数据包所能经过的最大跳数。每经过一个路由器，该数值减去1，当该字段的值为0时，数据包将被丢弃

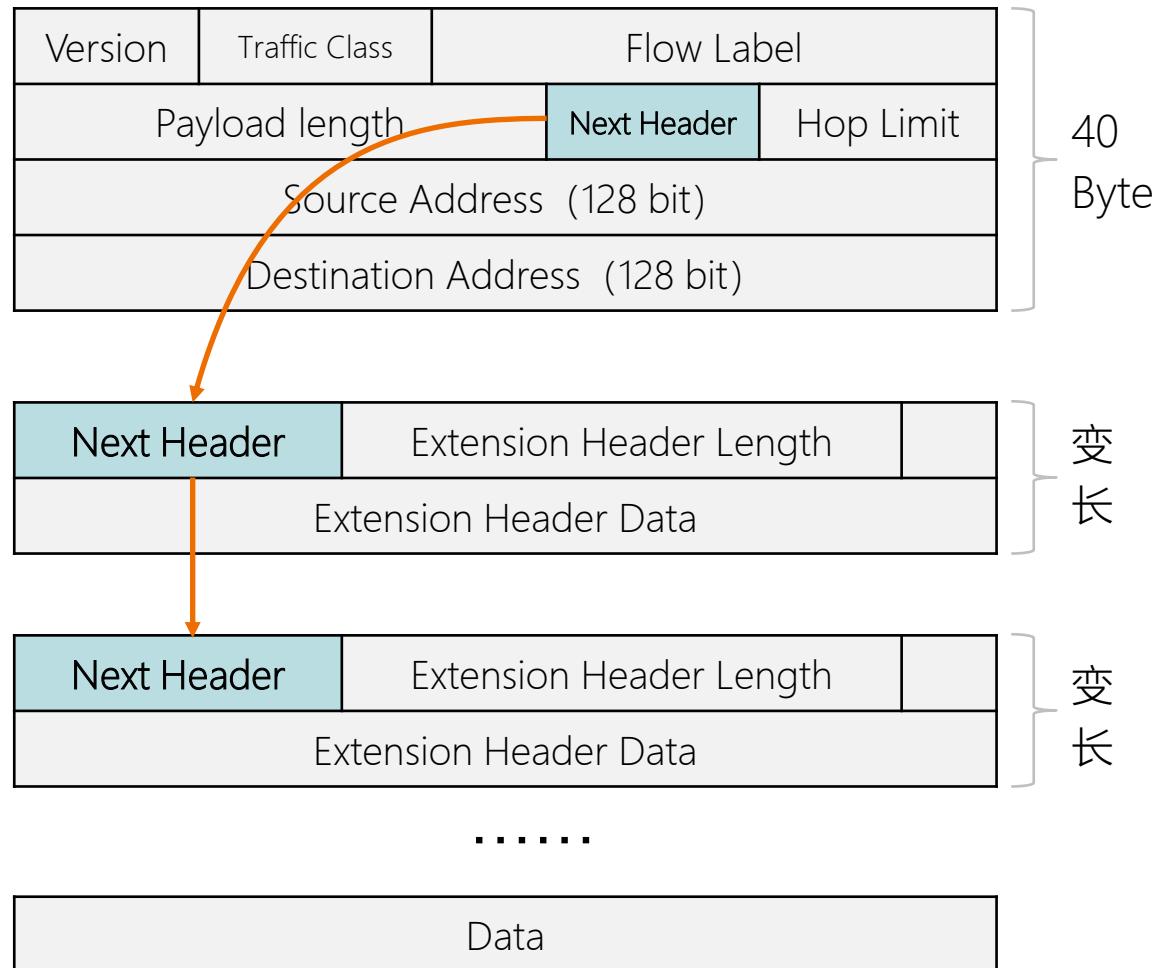


# IPv6基本头格式

- Source Address: 源地址(128 bit), 表示发送方的地址
- Destination Address: 目的地址, (128 bit)。表示接收方的地址



# IPv6扩展头



- Extension Header Length: 扩展报文头长度，长度为8 bit。表示扩展报文头的长度（不包含Next Header字段）
- Extension Header Data: 扩展报文头数据，长度可变。扩展报文头的内容，为一系列选项字段和填充字段的组合

# IPv6扩展头示例

示例1



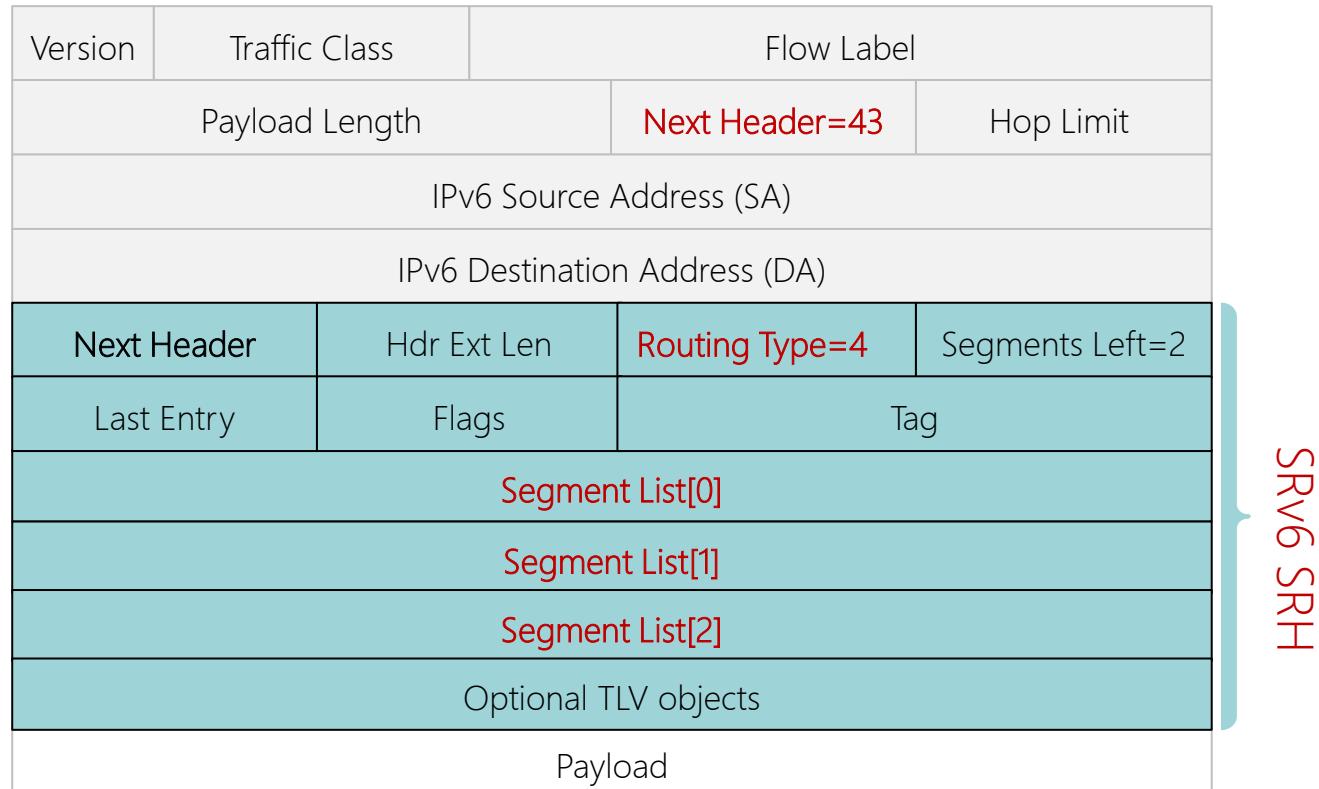
示例2



# IPv6扩展头

名称	Next Header协议号	作用
逐跳选项扩展报文头HBH (Hop-by-Hop Options Header)	0	用来携带需要被转发 <b>路径上的每一跳</b> 路由器去处理的信息
目的选项扩展报文头DOH (Destination Options Header)	60	用于携带需要由当前 <b>目的地址对应的节点</b> 去处理的信息
路由扩展报文头RH (Routing Header)	43	用来指明一个报文在网络内需要 <b>依次经过的路径点</b> ，用于源路由方案
分片扩展报文头 (Fragment Header)	44	携带了 <b>各个分片的识别信息</b>
认证扩展报文头AH (Authentication Header)	51	通常用于IPSec <b>认证</b> 。能提供3种安全功能：无连接的完整性验证、IP报文来源认证和重放攻击防护
封装安全有效载荷扩展报文头ESP (Encapsulating Security Payload Header)	50	通常用于IPSec <b>认证与加密</b> 。能提供无连接的完整性验证，数据来源认证，重放攻击防护，以及数据加密等安全功能
上层协议报文 (Upper-Layer Header)	ICMPv6 58 TCP 6/UDP 17	上层协议数据，比如ICMPv6等

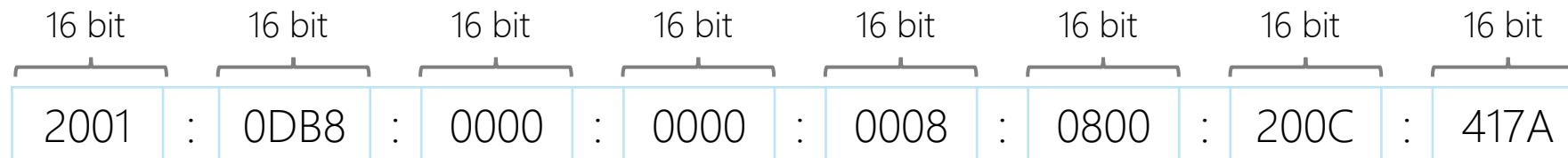
# IPv6扩展头典型应用：SRv6



- 为了基于IPv6转发平面实现 Segment Routing, IPv6路由扩展头 (Routing Header, RH) 新增加一种类型，称作SRH (Segment Routing Header) 扩展头，该扩展头指定一个IPv6 的显式路径，存储的是IPv6的 Segment List信息
- 当IPv6基本报文头的Next Header取值为43，表明下层报文头是RH；当Routing Type取值为4，表明RH的具体类型是SRH

# IPv6地址

- IPv6地址的长度为128 bit。一般用冒号分割为8段，每一段16 bit，每一段内用十六进制表示。地址中的字母大小写不敏感



- IPv6用“IPv6地址/掩码长度”的方式来表示IPv6地址
  - 例如： 2001:0DB8:2345:CD30:1230:4567:89AB:CDEF/64
  - 网络前缀： 2001:0DB8:2345:CD30::/64

# IPv6地址缩写规范

2001	:	0DB8	:	0000	:	0000	:	0008	:	0800	:	200C	:	417A
------	---	------	---	------	---	------	---	------	---	------	---	------	---	------

每组16 bit单元中的前导0可以省略，如果16 bit单元的所有比特都为0，那么至少要保留一个“0”；拖尾的0不能被省略



2001	:	DB8	:	0	:	0	:	8	:	800	:	200C	:	417A
------	---	-----	---	---	---	---	---	---	---	-----	---	------	---	------

一个或多个连续的16 bit单元为0时，可用“::”表示，但整个IPv6地址缩写中只允许有一个“::”。若缩写后的IPv6地址出现两个“::”，会导致无法还原为原始IPv6地址



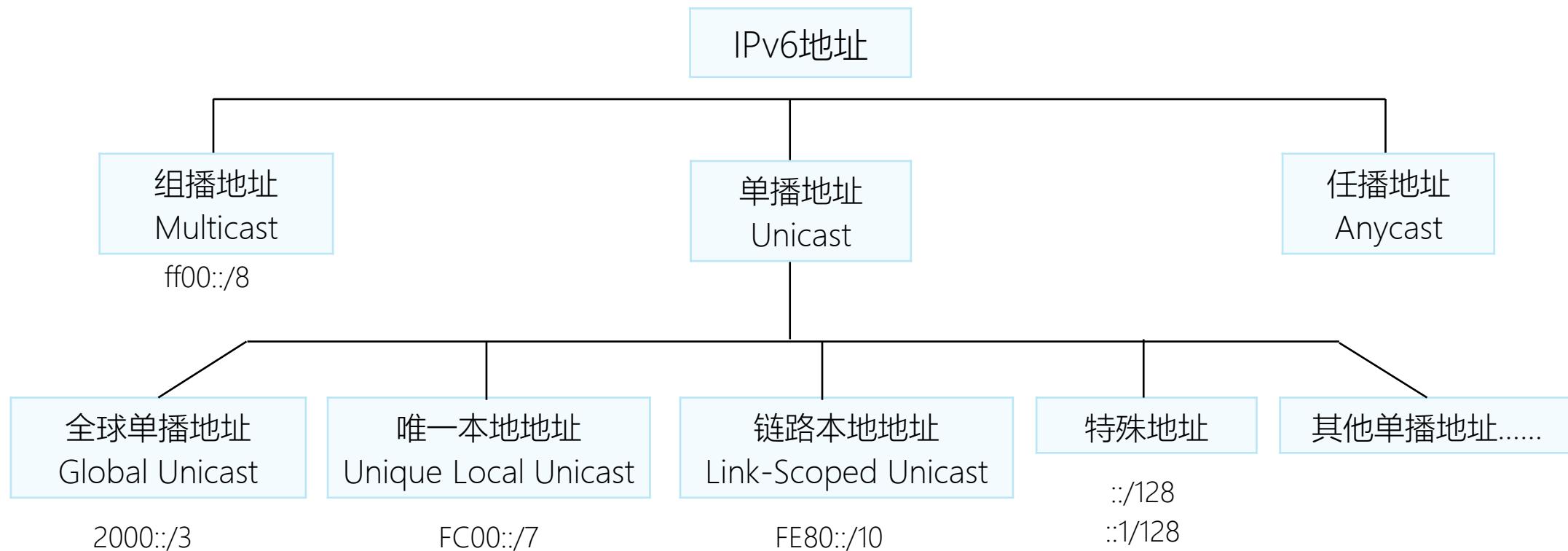
2001	:	DB8		::		8	:	800	:	200C	:	417A
------	---	-----	--	----	--	---	---	-----	---	------	---	------

# IPv6地址缩写示例

- 缩写前 0000:0000:0000:0000:0000:0000:0000:0001  
缩写后 ::1
- 缩写前 2001:0DB8:0000:0000:FB00:1400:5000:45FF  
缩写后 2001:DB8::FB00:1400:5000:45FF
- 缩写前 2001:0DB8:0000:0000:0000:2A2A:0000:0001  
缩写后 2001:DB8::2A2A:0:1
- 缩写前 2001:0DB8:0000:1234:FB00:0000:5000:45FF  
缩写后 2001:DB8::1234:FB00:0:5000:45FF  
或 2001:DB8:0:1234:FB00::5000:45FF

# IPv6地址分类

- IPv6地址分为为单播地址、组播地址和任播地址
- IPv6没有定义广播地址



[IANA: Internet Protocol Version 6 Address Space](#)

## IPv6单播地址结构

- 一个IPv6单播地址可以分为如下两部分
    - 网络前缀 (Network Prefix) : n bits, 相当于IPv4地址中的网络ID
    - 接口标识 (Interface Identify) : (128-n) bits , 相当于IPv4地址中的主机ID
  - 常见的IPv6单播地址, 如全球单播地址, 通常网络前缀和接口标识为64 bits



# IPv6单播地址接口标识

- 接口ID可通过3种方法生成：手工配置、系统自动生成，或基于IEEE EUI-64规范生成
- 基于IEEE EUI-64规范自动生成最为常用。EUI-64规范将接口的MAC地址转换为IPv6接口标识

MAC地址 (16进制)

08-70-5A-90-1A-01

MAC地址 (2进制)

第7bit=0表示该MAC地址是全局管理地址

00001000 – 01110000 – 01011010 – 10010000 – 00011010 - 00000001

② 第7bit取反

1

在这个位置插入FFFF

00001000 – 01110000 – 01011010 – 11111111 - 11111110 - 10010000 – 00011010 - 00000001

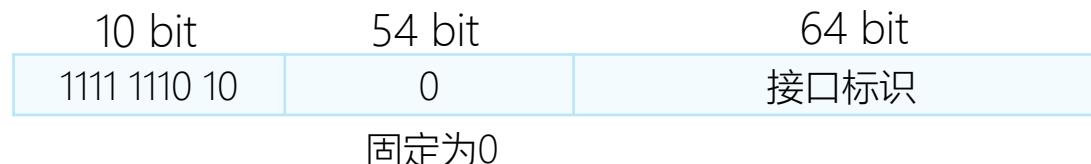
00001010 – 01110000 – 01011010 – 11111111 - 11111110 - 10010000 – 00011010 - 00000001

EUI-64规范的接口ID

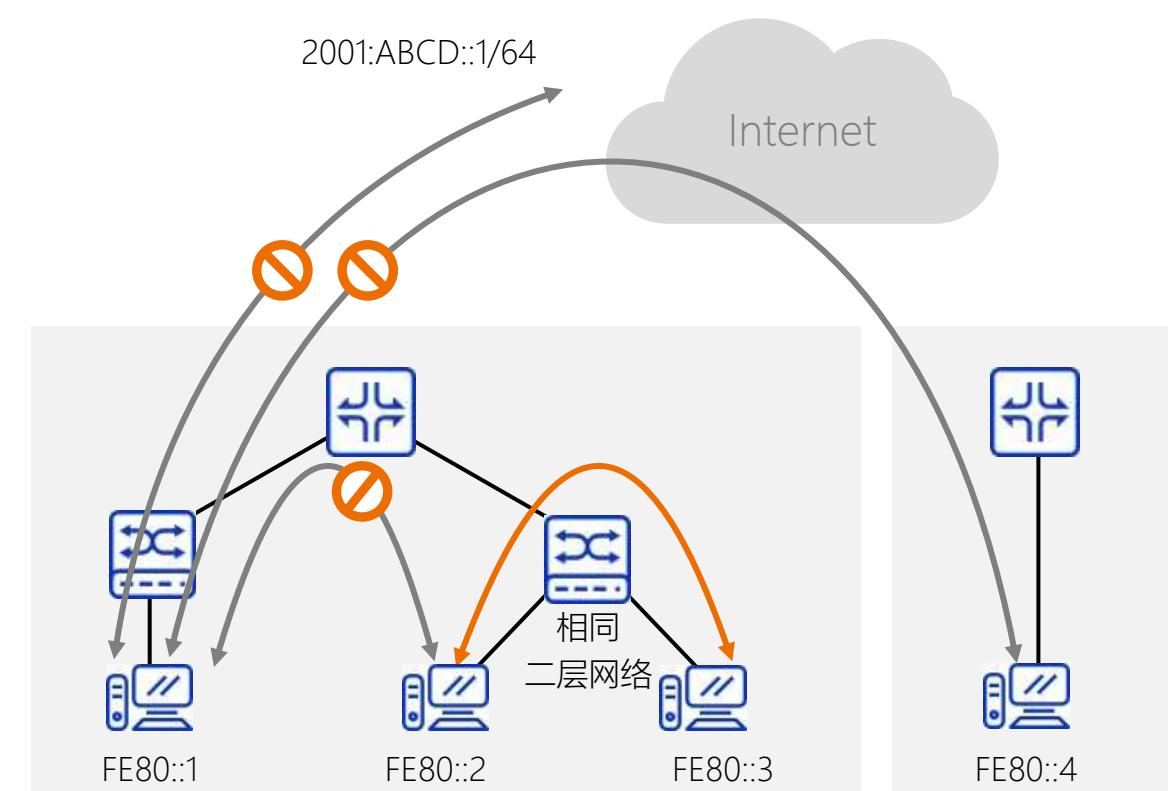
0A – 70 – 5A – FF - FE - 90 – 1A - 01

# IPv6链路本地地址

- 链路本地地址 (Link-Local Address, LLA) 是IPv6中另一种应用范围受限制的地址类型
- LLA的有效范围是本地链路，前缀为FE80::/10

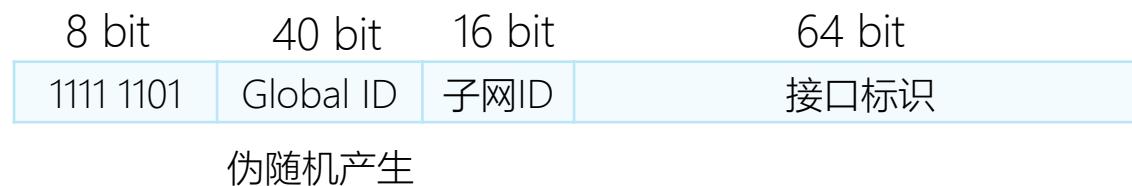


- LLA用于一条单一链路层面的通信，例如IPv6地址无状态自动配置、动态路由协议邻居关系建立等
- 源或目的IPv6地址为链路本地地址的数据包将不会被转发到链路之外
- 每一个IPv6接口都必须具备一个链路本地地址
- 当接口获得一个IPv6单播地址后，设备会自动为该接口配置链路本地地址

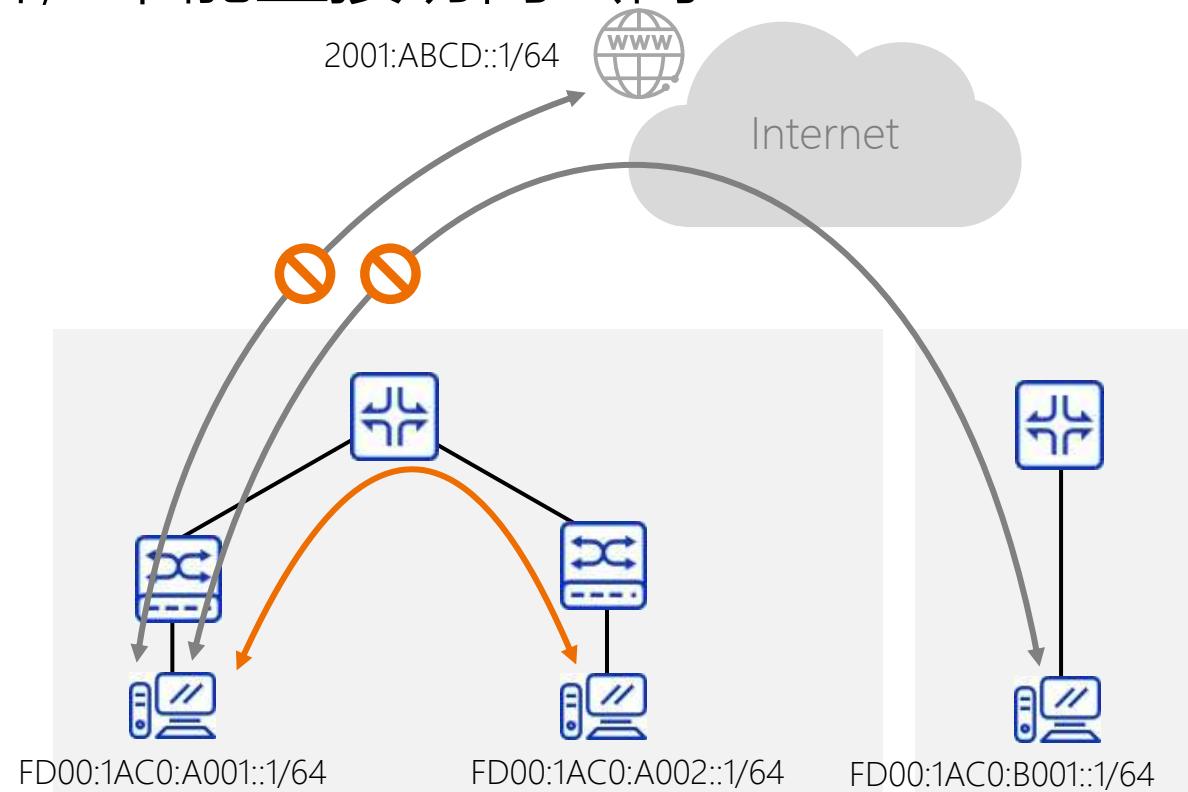


# IPv6唯一本地地址

- 唯一本地地址 (Unique Local Address, ULA) 是IPv6私网地址，只能够在内网中使用
- 该地址空间在IPv6公网中不可被路由，不能直接访问公网



- 唯一本地地址使用FC00::/7地址块，目前仅使用了FD00::/8地址段。FC00::/8预留为以后拓展用
- ULA虽然只在有限范围内有效，但也具有全球唯一的前缀（随机方式产生，但是冲突概率很低）

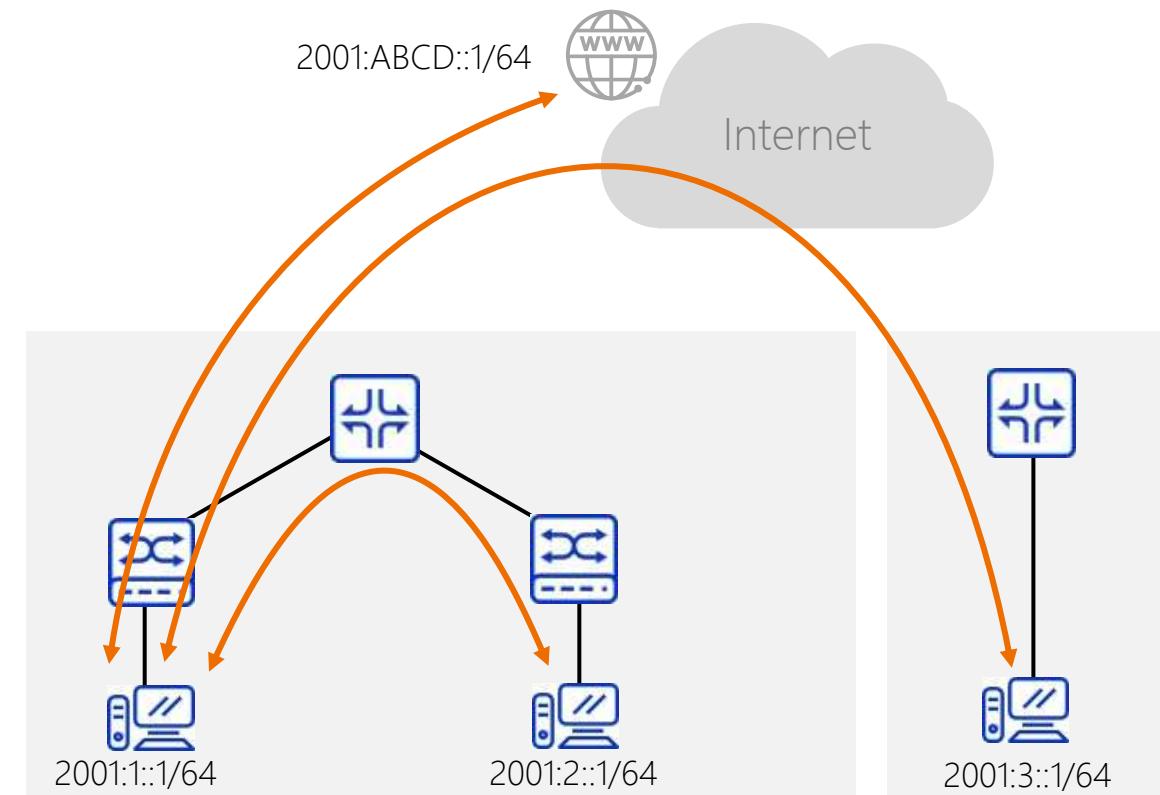


# IPv6全球单播地址

- 全球单播地址(Global Unicast Address, GUA)也称为可聚合全球单播地址
- 该类地址全球唯一，用于有互联网访问需求的主机，相当于IPv4的公网地址



- 通常GUA的网络部分长度为64 bit，接口标识也为64 bit
- 全局路由前缀：通常由服务提供商分配给一个组织机构（例如企业、学校等），一般至少为45 bit
- 子网ID：组织机构根据自身网络需求划分子网
- 接口标识：用来标识一个设备（的接口）

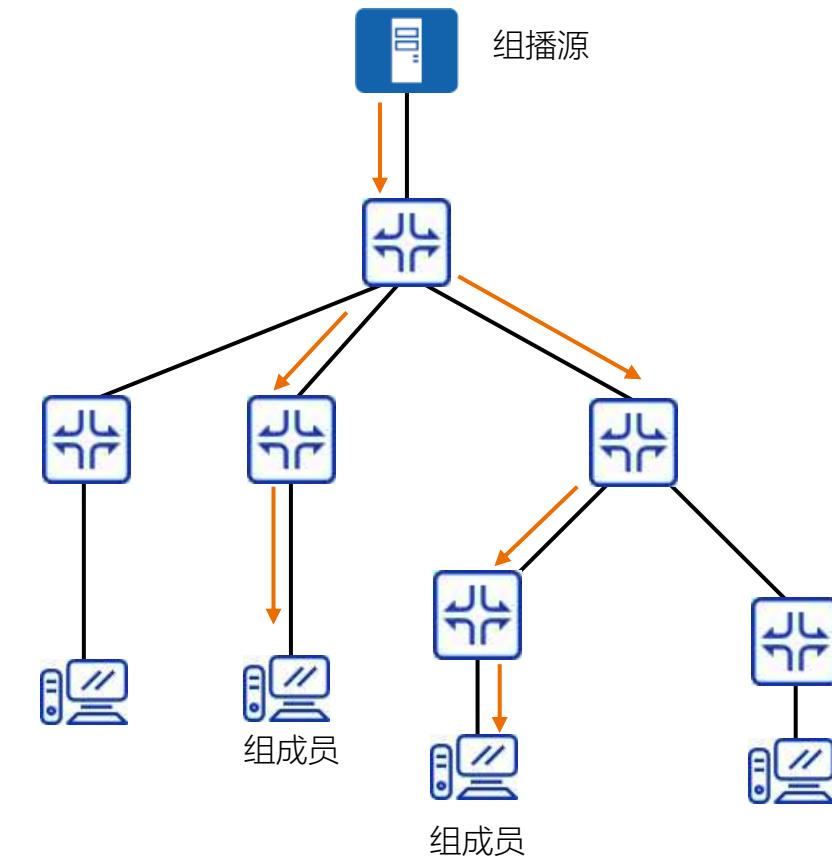


# IPv6组播地址

- IPv6组播地址标识多个接口，一般用于“一对多”的通信场景
- IPv6组播地址只可以作为IPv6报文的目的地址

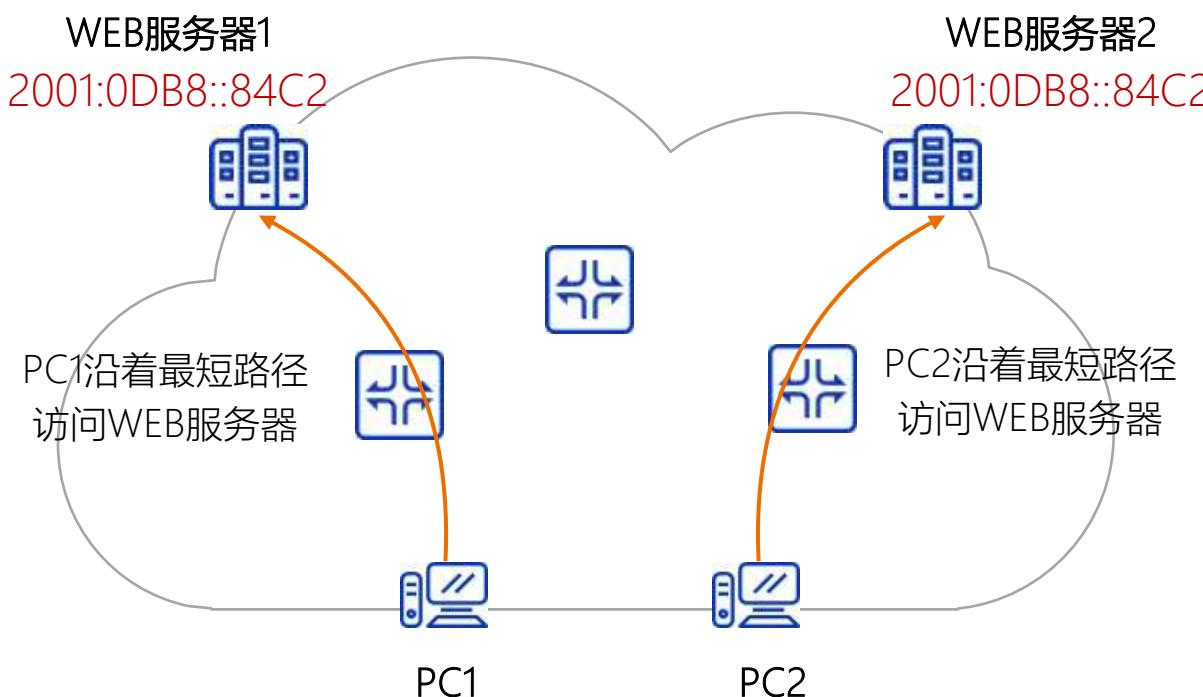


- Flags: 4bit, 目前只使用了最后1个bit (前3bit必须置0)
- 当Flags=0000时, 表示该组播地址是由IANA所分配的一个永久分配地址
- 当Flags=0001时, 表示该组播地址是一个临时组播地址 (非永久分配地址)
- Scope: 用来限制组播数据流在网络中发送的范围。例如取值为2标识链路本地范围
- Group ID: 组播组ID

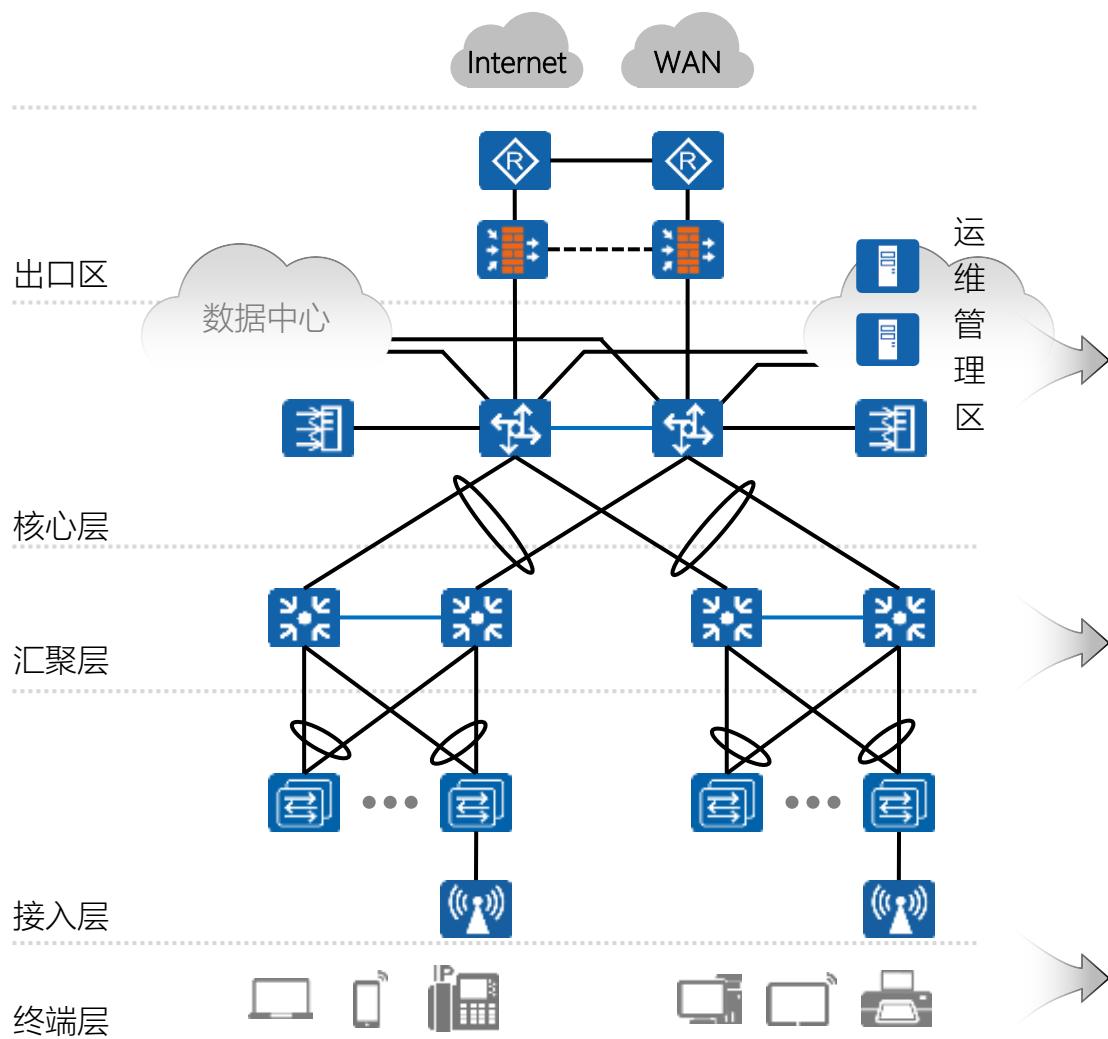


# IPv6任播地址

- 任播地址标识一组网络接口（通常属于不同的节点）
- 任播地址可以作为IPv6报文的源地址，也可以作为目的地址
- 主要为DNS、HTTP提供服务
  - 在为多个主机或节点提供相同服务的前提下条件下提供冗余和负载分担



# 典型园区网络中的地址配置需求



运维管理区的服务器长期固定，且随时需要被网络管理员访问，因此通常配置静态IPv6地址

网络设备承载业务数据，且自身也存在被管理的需求，因此也需要IPv6地址，通常配置静态IPv6地址

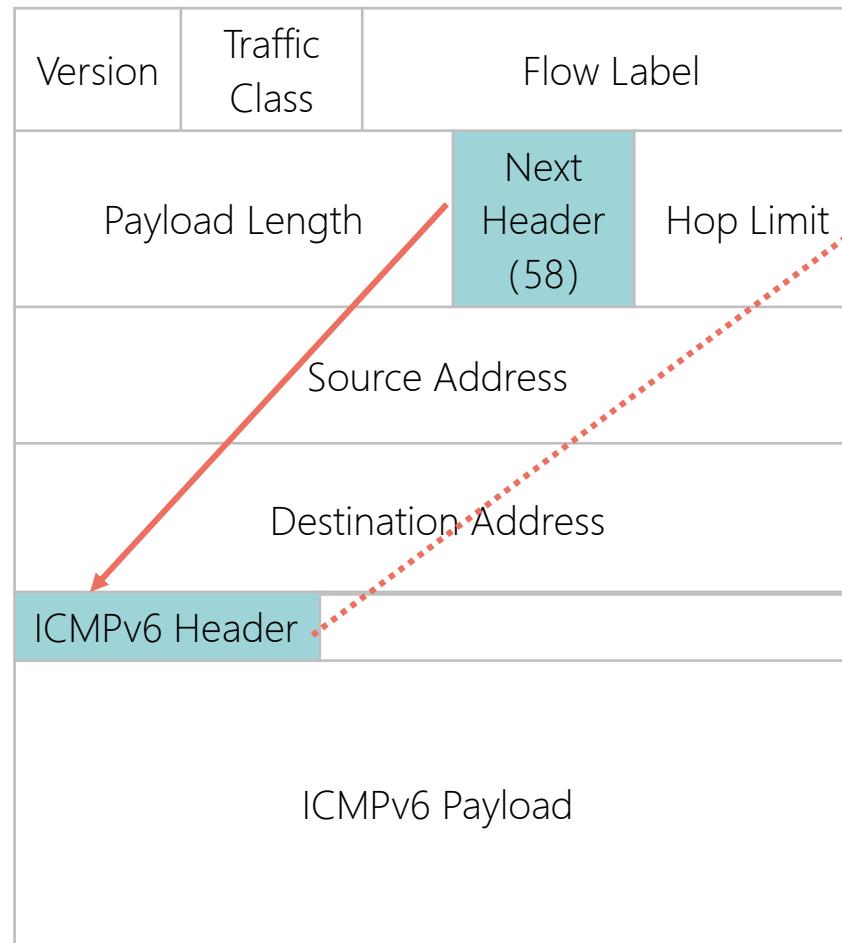
用户终端、物联网终端等需要获取IPv6地址接入园区网络，往往数量多，且具移动性，通常采用自动化配置地址，部分场景亦可静态配置地址

# ICMPv6

---

- ICMPv6是IPv6的基础协议之一
- 在IPv4中，Internet控制报文协议ICMP向源节点报告关于向目的地传输IP数据包过程中的错误和信息。它为诊断、信息和管理目的定义了一些消息
  - 目的不可达、数据包超长、超时、回应请求和回应应答.....
- ICMPv6除了提供ICMPv4常用的功能之外，还是其它一些功能的基础，
  - 邻接点发现、无状态地址配置（包括重复地址检测）、PMTU发现.....
- ICMPv6的协议号（即IPv6报文中的Next Header字段的值）为58

# ICMPv6报文格式



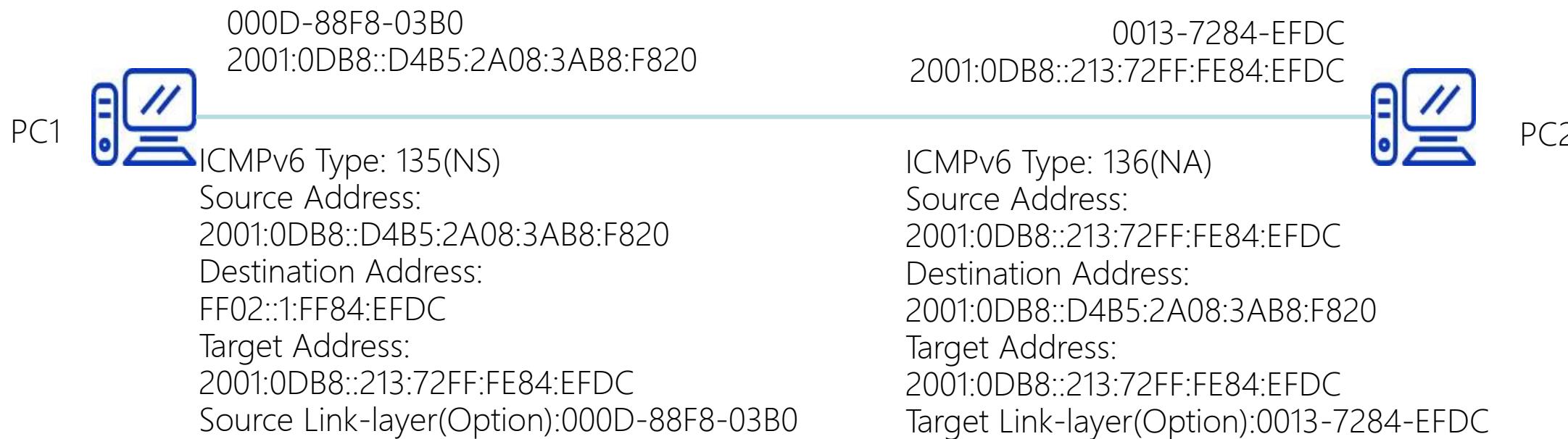
ICMPv6 Header

Type	Code	Checksum
------	------	----------

消息类型	Type	名称	Code
差错消息	1	目的不可达	0 无路由
			1 因管理原因禁止访问
			2 未指定
			3 地址不可达
差错消息	2	数据包过长	4 端口不可达
			0
			0 跳数到0
差错消息	3	超时	1 分片重组超时
			0 错误的包头字段
			1 无法识别的下一包头类型
信息消息	4	参数错误	2 无法识别的IPv6选项
			128 Echo Request 0
			129 Echo Reply 0

# NDP协议

- Neighbor Discovery Protocol (NDP) — 邻居发现协议在IPv6中，取消了 IPv4中的ARP协议，使用NDP所定义的邻居请求报文（NS）及邻居通告报文（NA）来实现地址解析功能
- 被请求节点组播地址： FF02::1:FFXX:XXXX/104
- IPv6组播MAC地址以0x3333开头，低32位为IPv6组播地址的低32位，最终形成48比特的组播MAC地址



# 网卡启动过程



# 生成链路本地地址



## ■ Link-Local Address

- 当网卡启动时会根据某种算法自动生成链路本地地址。具体生成的地址，与操作系统有关，不统一
    - 根据MAC地址换算而来 (EUI-64)  
例如：7007.1234.5678 → FE80::7207:12FF:FE34:5678
    - 随机生成
  - 链路本地地址是范围为fe80::/10的单播地址
  - 链路本地只在同一个二层内传播，不会被路由器转发
- ## ■ 链路本地地址作用
- 地址自动配置
  - 邻居发现协议
  - 路由转发（可以作为下一跳地址）

# 生成被请求节点多播地址



- Solicited-Node multicast address
  - 每生成一个单播IP地址，无论什么类型，都会对应生成一个“被请求节点多播地址”
  - 组成方式：FF02::1:FF00:0/104 + 单播地址的最后24bit
- “被请求节点多播地址”用途：地址解析
  - IPv4中用ARP做地址解析，ARP是基于广播的
  - IPv6没有广播，只有多播，使用一个多播地址解析
- 工作原理：当想解析MAC地址时
  - 发送一个“地址解析请求包”到这个多播地址，然后属于该多播地址的成员会收到该数据包，
  - 返回MAC地址给对方
- “被请求节点”含义
  - 别人请求解析“我”的地址，“我”就是被请求的节点
  - “我”生成“被请求节点多播地址”是让别人能够请求到我

# 多播成员报告



## ■ 多播成员报告

- 对外声明“我要加入某某多播组”
- 成员报告是单向的，不会收到回应包
- 使用MLDv2协议（多播控制协议）

## ■ 为什么要进行“多播成员报告”

- 成员报告的目的是为了减少网络中的多播流量
- 多播的工作机制：只要生成多播地址，就要进行成员报告

## ■ 要报告的是哪个成员

- 要报告的成员不是单播地址，而是多播地址
- 例如：“我要加入ff02::1:**ff00:2**多播组”

# 重复地址检测



- Duplicate Address Detection, 简称DAD
  - 为了防止IP地址冲突，每生成一个单播地址，都会进行一次重复地址检测
  - 此刻就是对Step1生成的“链路本地地址”进行检测
- 何时进行“重复地址检测”
  - 在生成单播地址并完成发送一次“MLDv2成员报告”后，会随机延时一小段时间进行检测
  - 是否进行检测可通过内核参数设置
- 重复地址检测的工作原理
  - 发送一个地址解析包（Neighbor Solicitation, NS），请求解析的地址就是自己的地址，并等待回应
  - 若超时仍未得到回应（Neighbor Advertisement, 简称NA），即可认为地址可用

# 无状态地址自动配置

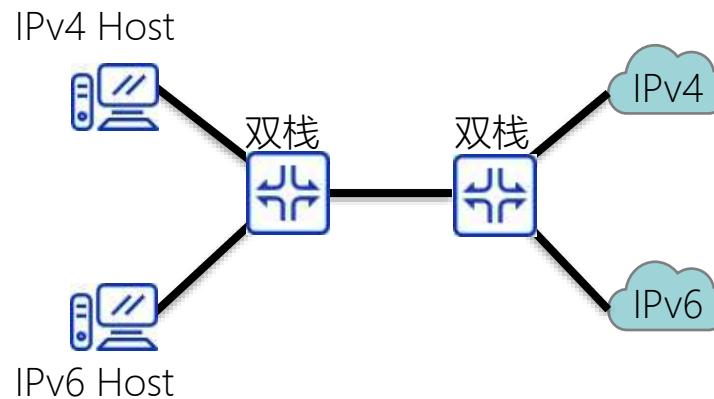


- SLACC: Stateless Address Autoconfiguration
  - 自动配置IP地址、自动配置网关
- 工作原理：涉及到这2种报文
  - “路由器请求” (Router Solicitation, RS)
  - “路由器通告” (Router Advertisement, RA)
  - 当收到路由器回应的RA报文后，就会根据报文中的IP前缀信息，自动生成IP地址，并将网关指向该路由器的“链路本地地址”
- 收到RA报文的2种办法
  - 路由器定期发送RA报文
  - 自己主动发送RS报文，路由器收到后就会立刻回应RA报文

# IPv4/IPv6互通

## IPv4/IPv6双栈

- IPv4/IPv6在网络中并存、独立部署。对现有IPv4业务影响较小
- 演进方案简单、易理解。网络规划设计工作量相对更少



## 隧道技术

- IPv6 over IPv4隧道：将IPv6流量封装在IPv4隧道中，在IPv4网络中实现IPv6孤岛互通

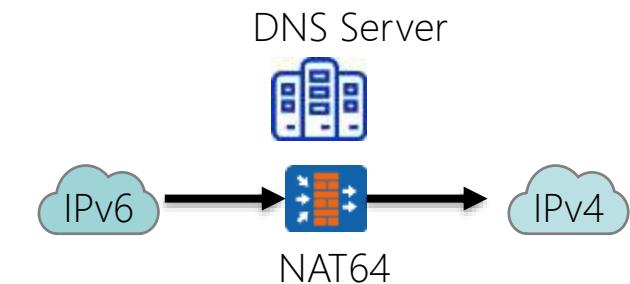


- IPv4 over IPv6隧道：将IPv4流量封装在IPv6隧道中，在IPv6网络中实现IPv4孤岛互通



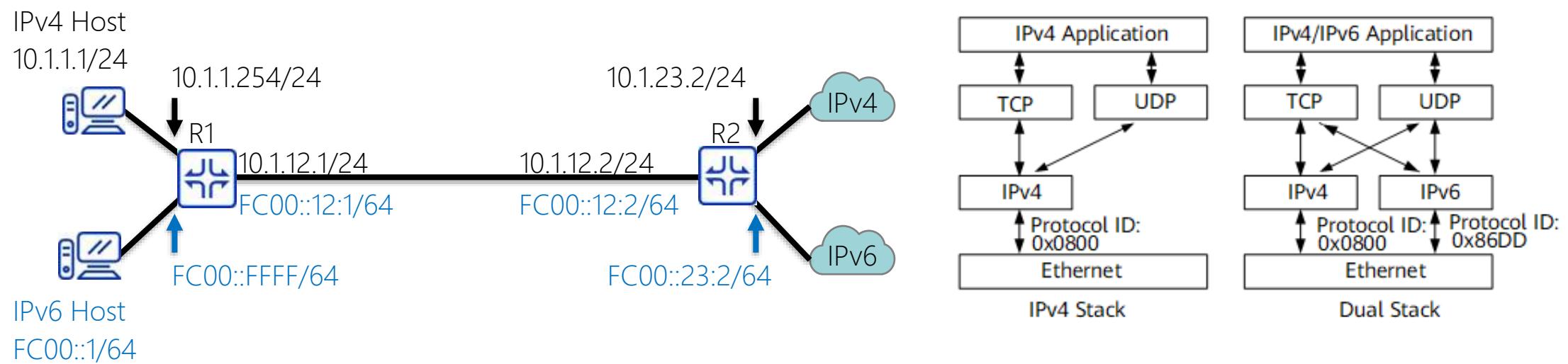
## 转换技术

- 将IPv4流量与IPv6流量进行协议转换。适用于纯IPv4网络与纯IPv6网络之间的通信，需要在网络中部署NAT、DNS等



# IPv4/IPv6双栈

- IPv4/IPv6双栈是最基本的过渡机制
- 网络中的设备同时支持IPv4和IPv6协议栈
- 源节点根据目的节点的不同选择不同的协议栈，而网络设备根据报文的协议类型选择不同的协议栈进行数据处理



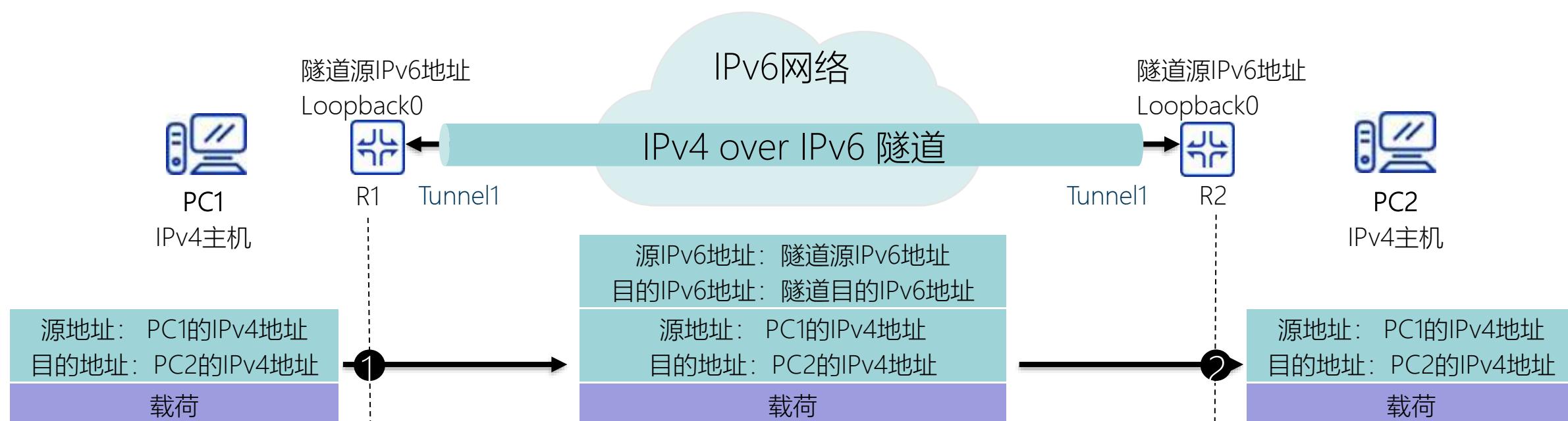
# 隧道技术：IPv6 over IPv4隧道

- IPv6 over IPv4隧道的源IP地址必须为手工配置，而目的IPv4地址有手工配置和自动获取两种方式
- 根据隧道目的IPv4地址的获取方式不同，可以将IPv6 over IPv4隧道分为手工隧道和自动隧道



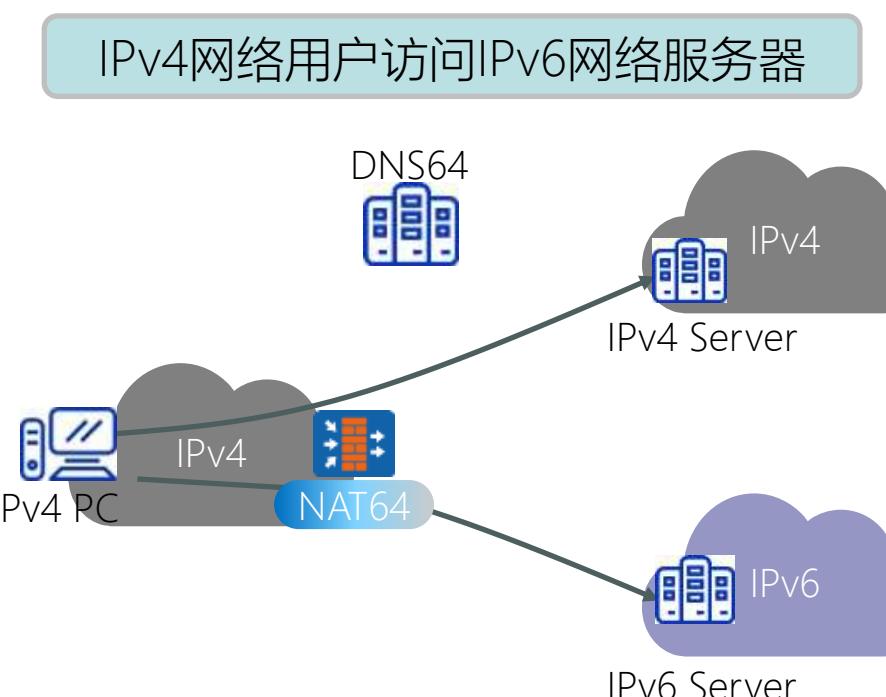
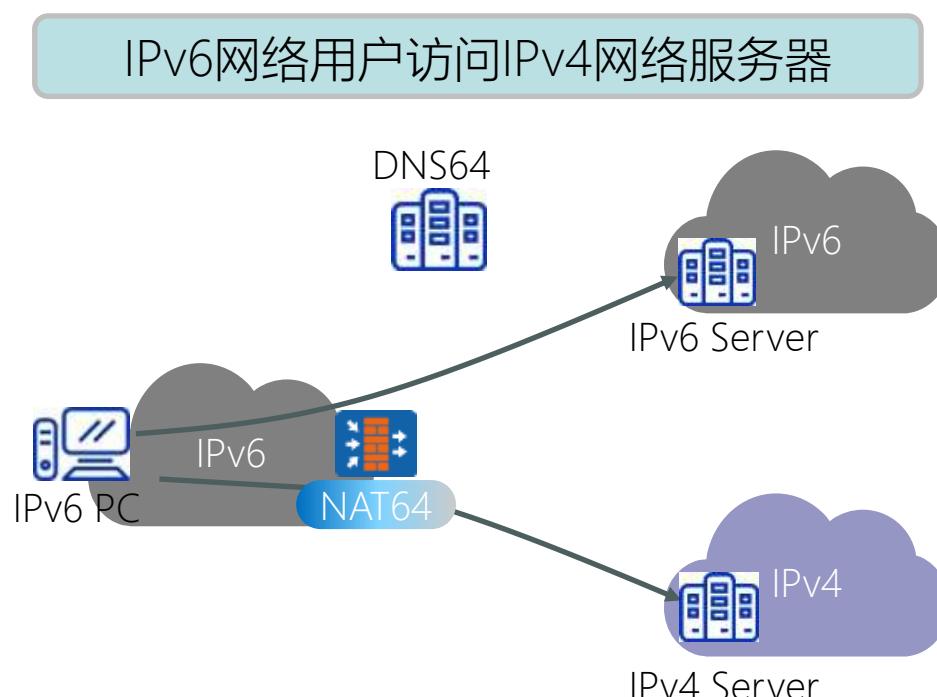
# 隧道技术：IPv4 over IPv6隧道

- 在IPv4网络向IPv6网络过渡后期，IPv6网络已被大量部署，而IPv4网络只是被IPv6网络隔离开的局部网络
- 利用隧道技术可以在IPv6网络上创建隧道，使IPv4网络能通过IPv6网络互通，这种隧道称为IPv4 over IPv6隧道



# 转换技术：NAT64

- NAT64是一种在IPv6与IPv4之间实现协议转换的NAT技术
- 当IPv4网络的节点需要直接与IPv6网络的节点进行通信时，默认情况下是行不通的，因为两个协议栈无法兼容。但是借助一台NAT64设备，由该设备来实现IPv6与IPv4的转换，那么上述通信就可以实现

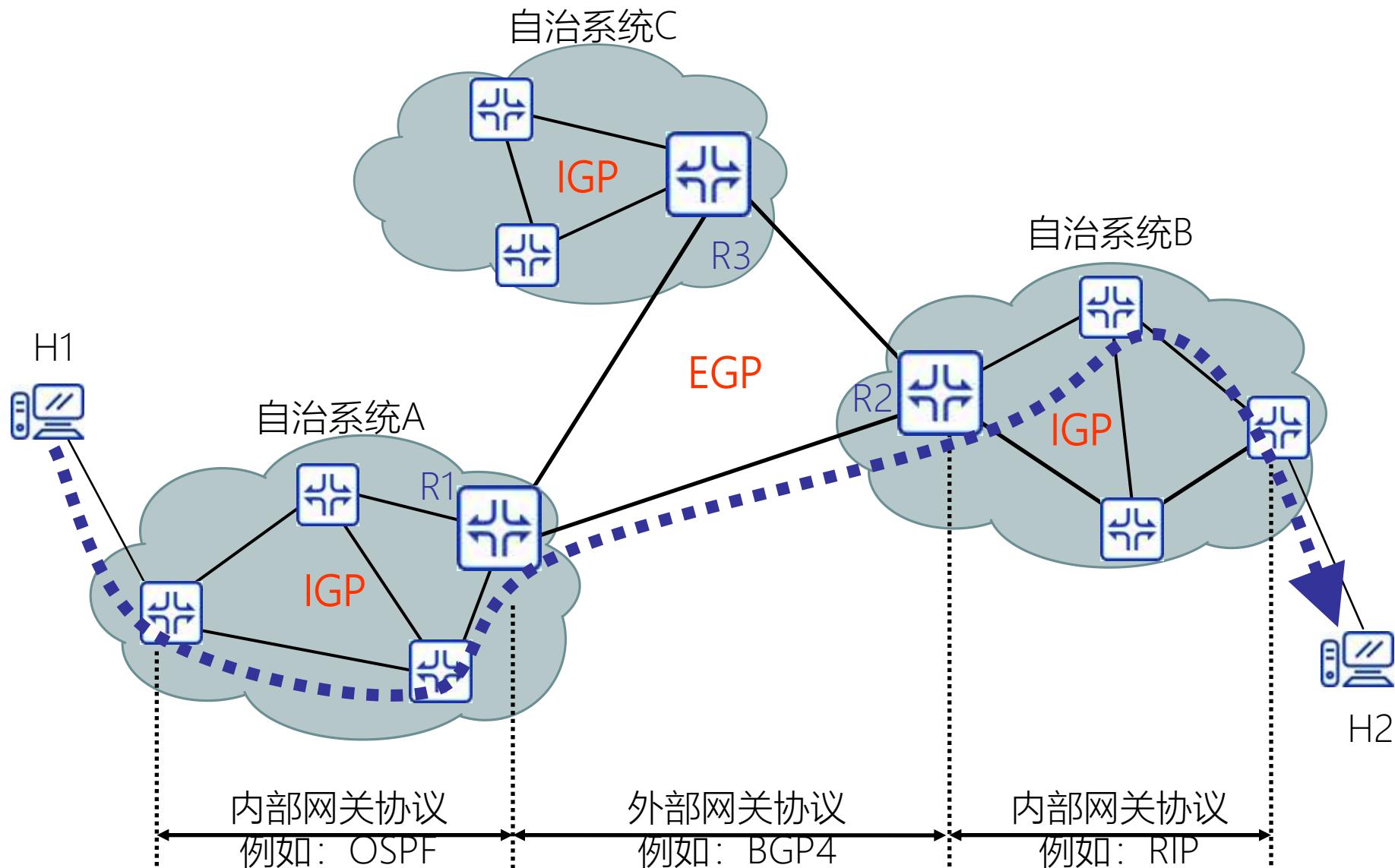


## 5.4 互连网路由问题

---

- 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
- 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)

# 自治系统和内部、外部网关协议



# 内部网关路由选择协议

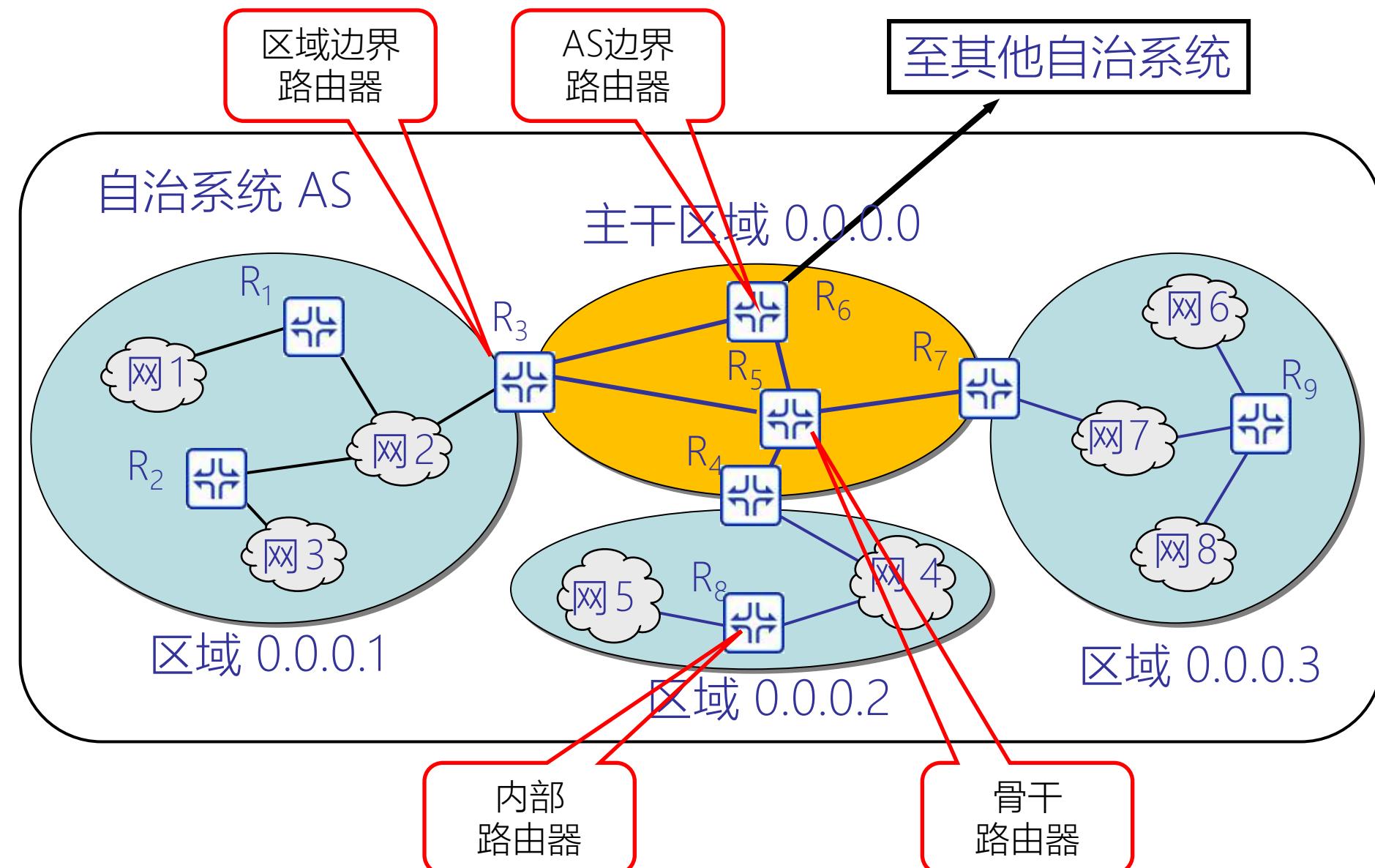
- OSPF (Open Shortest Path First)开放最短路径优先
- OSPF路由协议是典型的链路状态路由协议，是互连网应用最广的路由协议之一
- OSPF的SPF算法
  - SPF算法是OSPF路由协议的基础。SPF算法有时也被称为Dijkstra算法
- 在OSPF路由协议中，最短路径树的树干长度，称为OSPF的Cost，其算法为：

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

# 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数据包类型

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 state 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

# 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

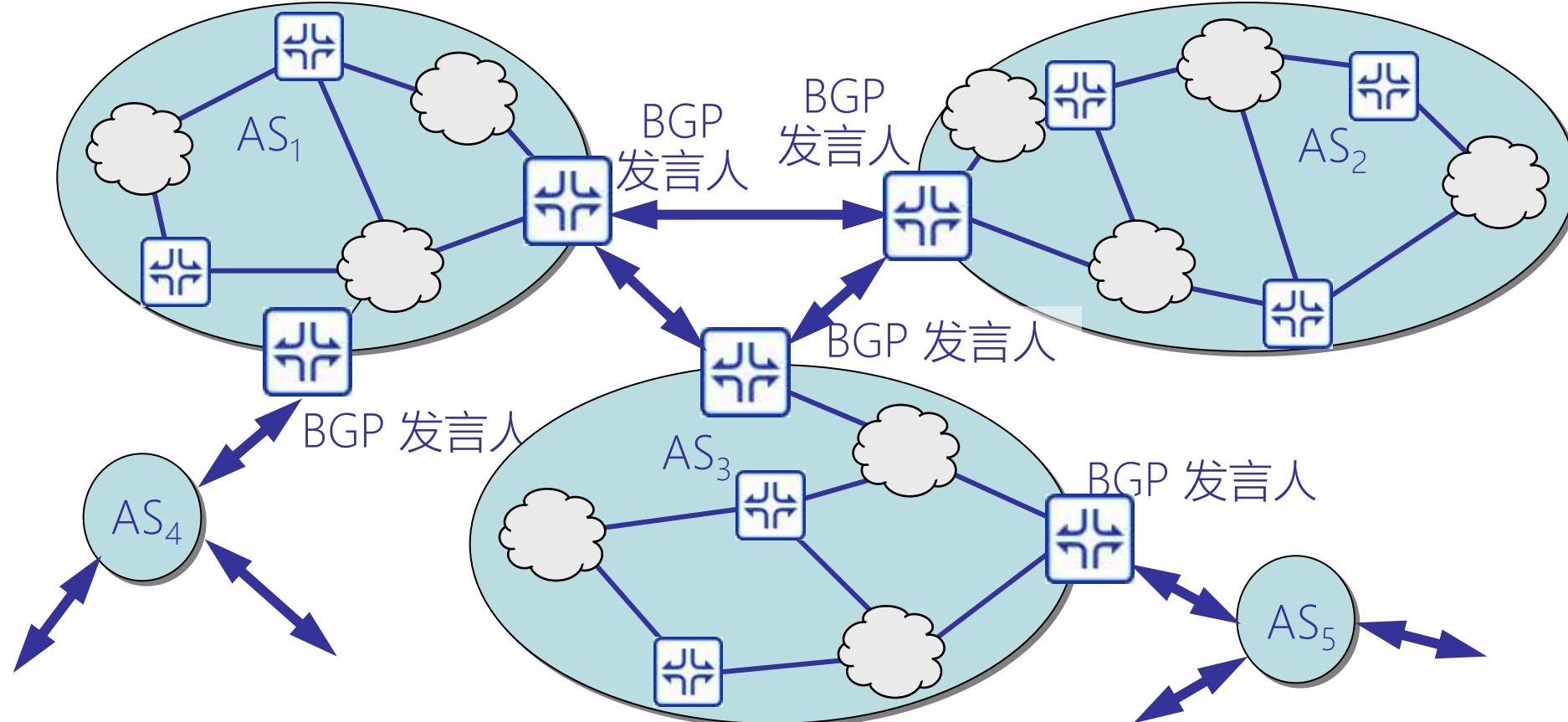
# BGP使用的环境不同

- 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

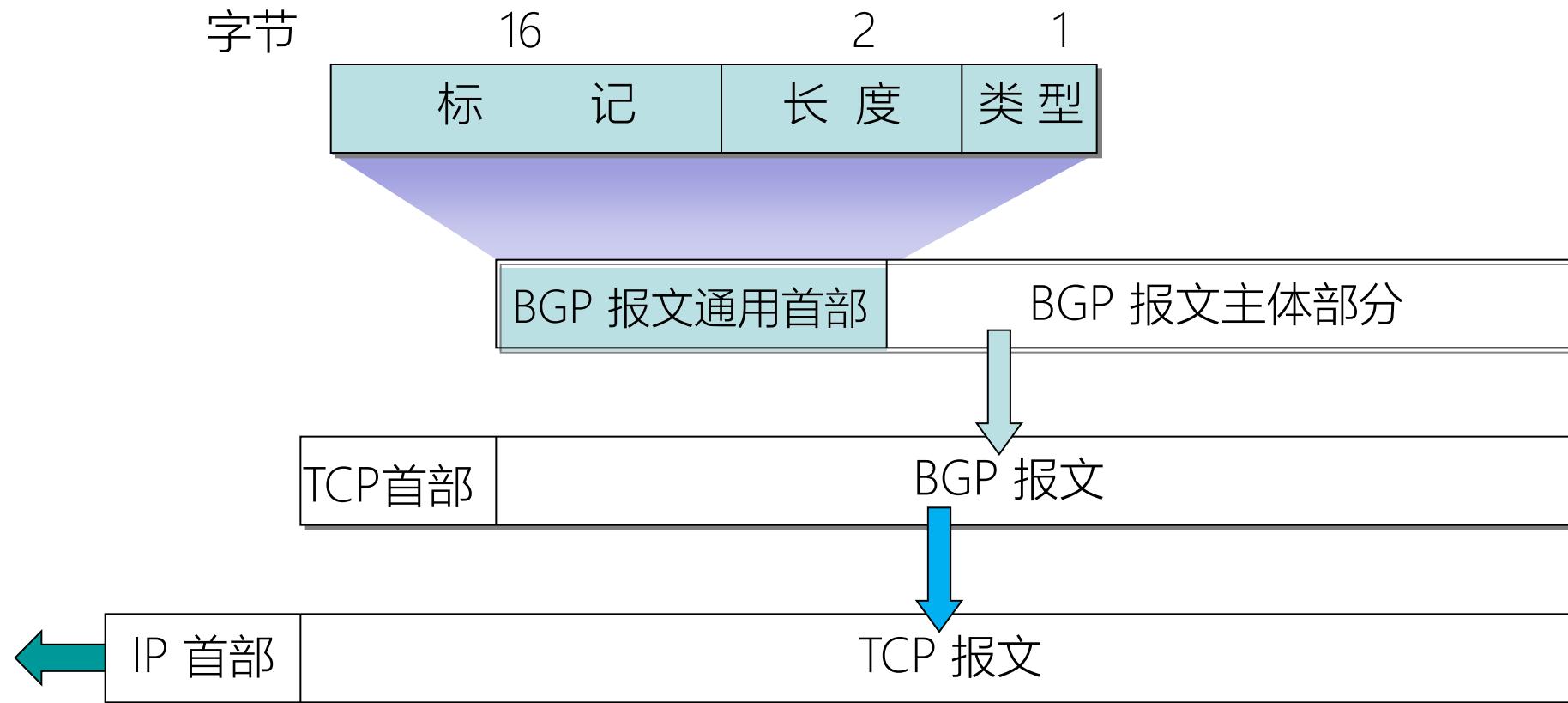
# BGP发言人和自治系统 AS 的关系



# 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

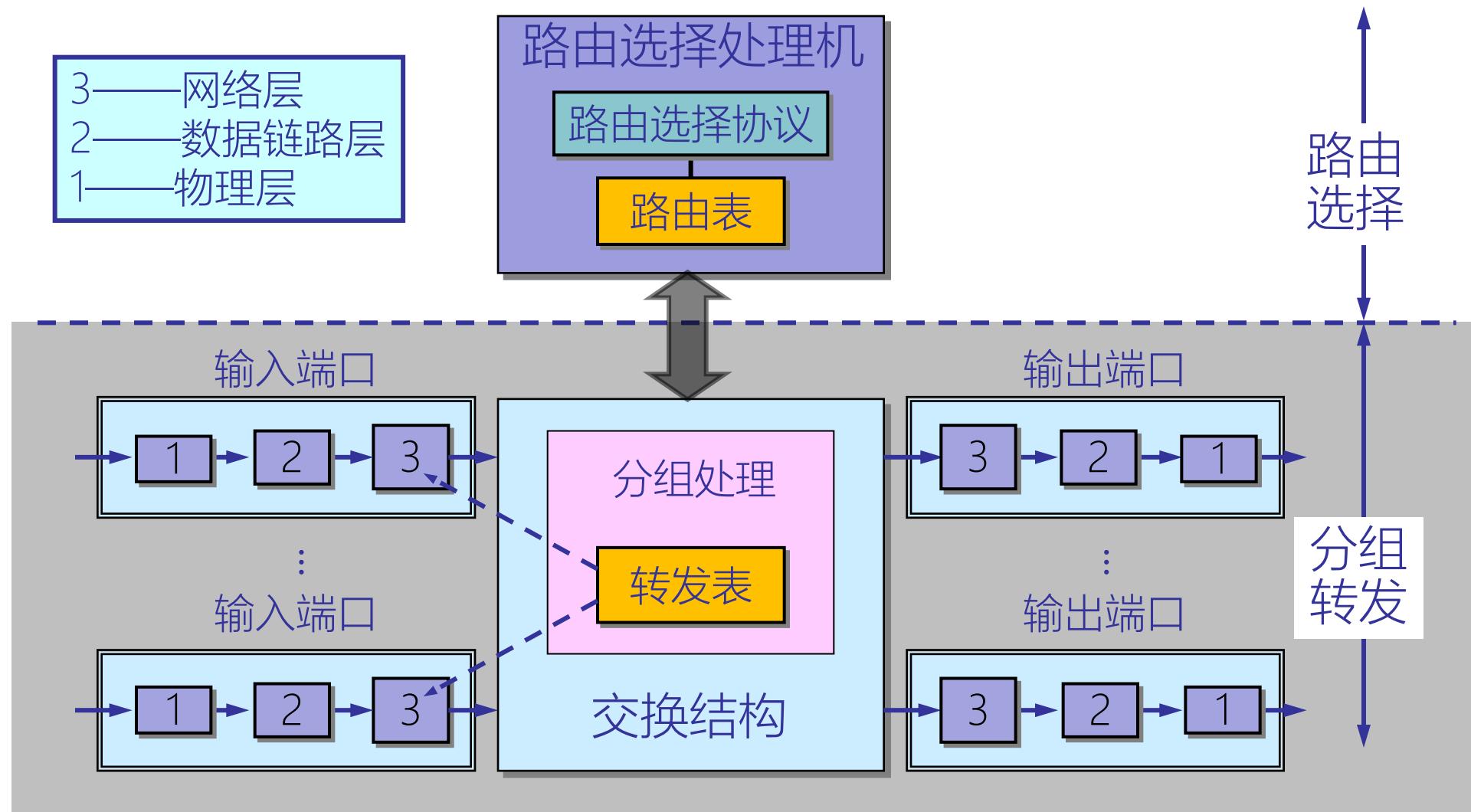


# BGP-4 使用四种报文类型

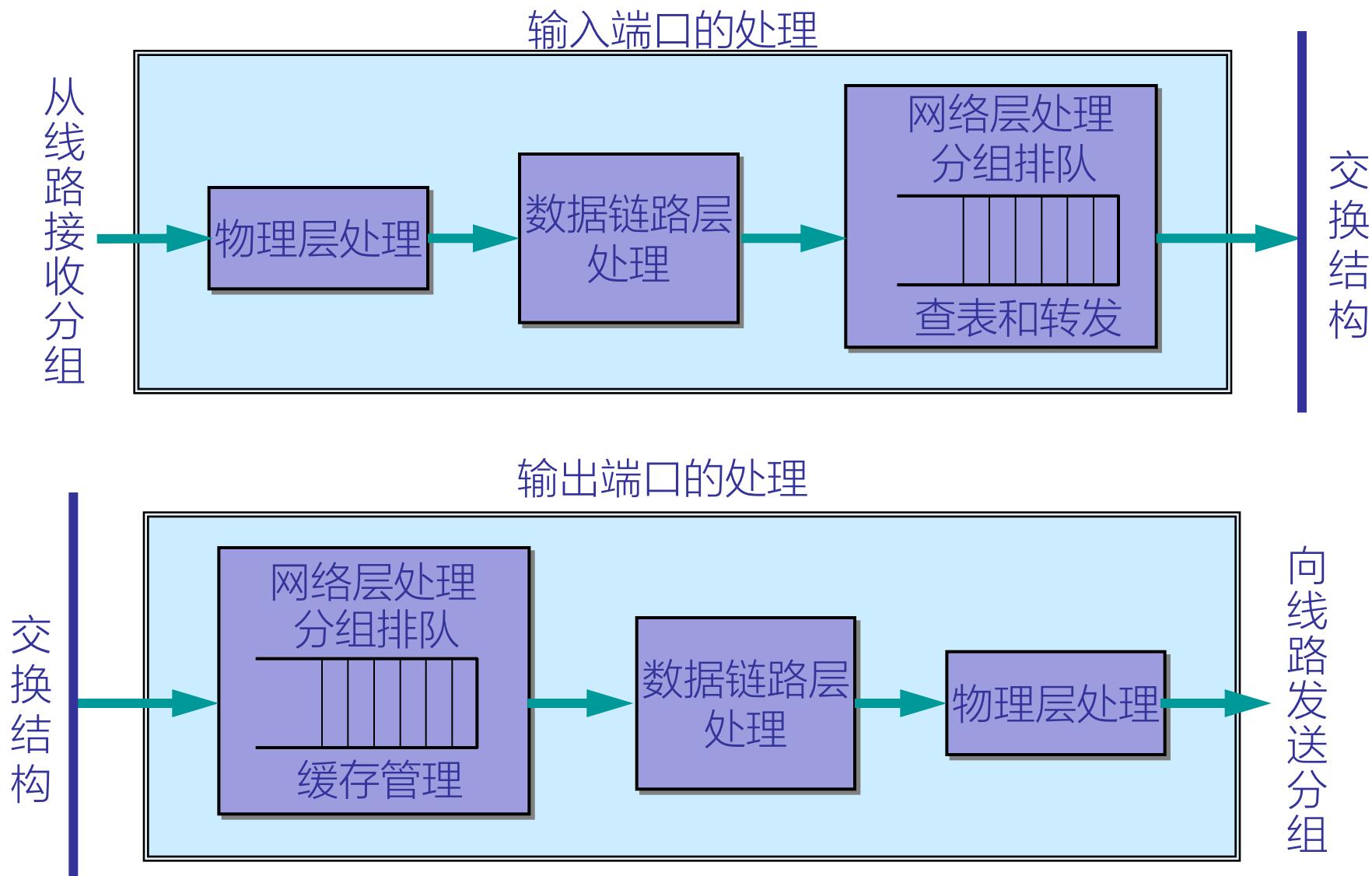
---

- 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

# 路由器体系结构



# 路由器输入、输出端口

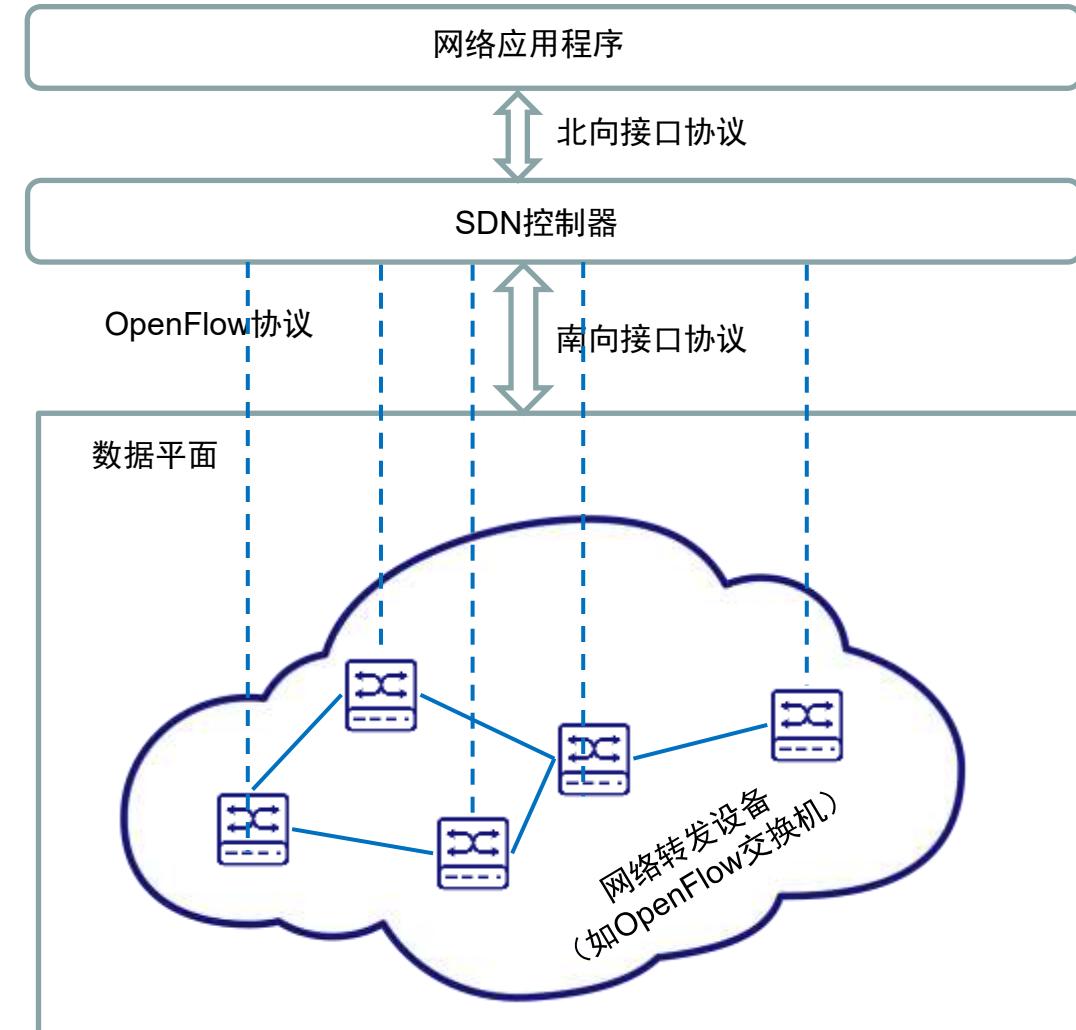


## 5.5 软件定义网络

- SDN: Software-defined

Networking技术是一种网络管理方法，它支持动态可编程的网络配置，提高了网络性能和管理效率，使网络服务能够像云计算一样提供灵活的定制能力

- SDN将网络设备的转发面与控制面解耦，通过控制器负责网络设备的管理、网络业务的编排和业务流量的调度，具有成本低、集中管理、灵活调度等优点



SDN网络控制平面和数据平面的分离

# SDN的优点

- 网络可编程：网络设备提供应用编程接口（API），管理人员能够通过编程语言向网络设备发送指令
- 网络抽象化：底层的硬件设备抽象为虚拟化的资源池，应用和服务不再与硬件紧密耦合
- 业务灵活调度：通过开放的南北向接口，实现了将计算机语言到配置命令行的翻译，解决了传统网络业务调度不灵活的问题
- 集中管理：管理员可以直接感知整个网络的状态，及时调整带宽和优化策略，便于进行整网的管理
- 开放性：开放的API支持云编排、SaaS等多种应用程序，也可以通过Openflow控制不同供应商的硬件

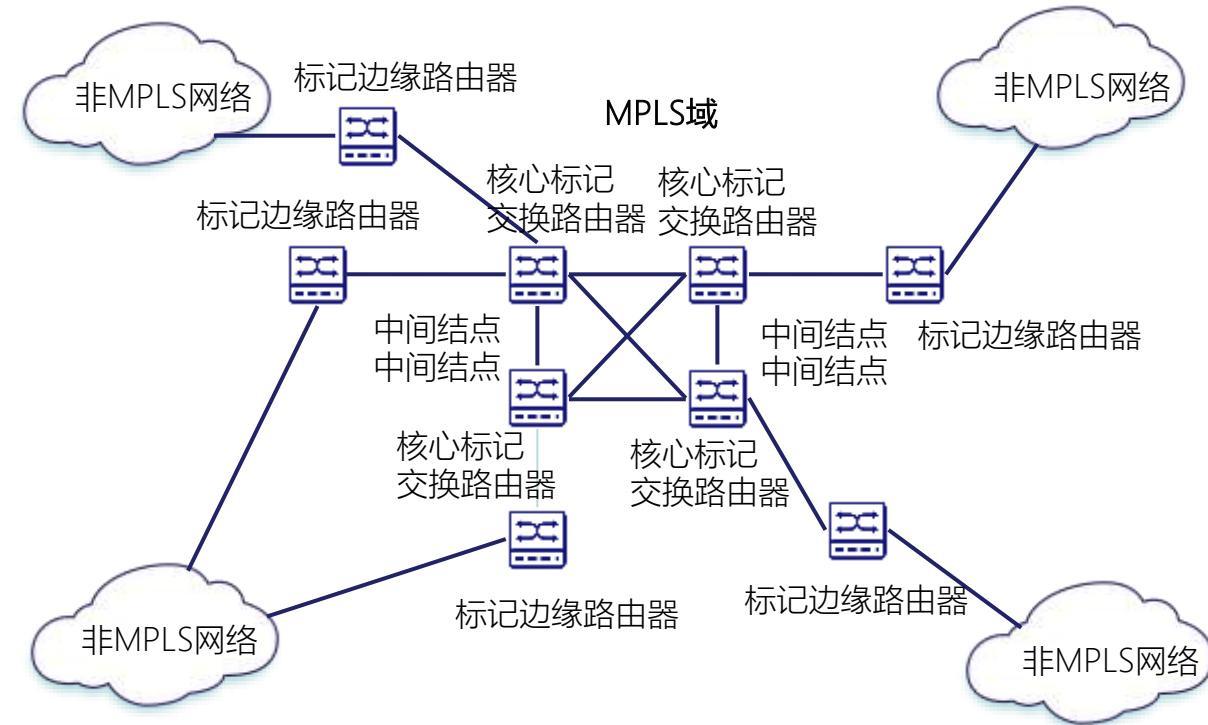
# SDN网络工作流程

- 控制器和转发设备之间建立控制通道，通常使用传统的IGP/VLAN
- 控制器和转发设备建立连接后，从转发设备收集网络资源信息：
  - 设备信息、接口信息、标签信息等
  - 通过拓扑收集协议收集网络拓扑信息
- 控制器利用网络拓扑信息和网络资源信息计算网络内部的交换路径，将信息下发给转发设备
- 转发设备接收控制器下发的网络内部交换路径转发表数据和业务路由转发表数据，并依据这些转发表进行报文转发
- 当网络状态发生变化时，SDN控制器会实时感知网络状态，并重新计算网络内部交换路径和业务路由，以确保网络能够继续正常提供业务

# 5.6 多协议标签交换与段路由协议

## ■ 多协议标签交换MPLS基本思想：

- 在 IP 数据包进入 MPLS 网络时，MPLS 入口的标记边缘路由器分析 IP 数据包的内容并为这些 IP 数据包添加合适的标记
- 所有 MPLS 网络中的标记交换路由器都根据标记转发数据
- 当该 IP 数据包离开 MPLS 网络时，标记由出口标记边缘路由器弹出

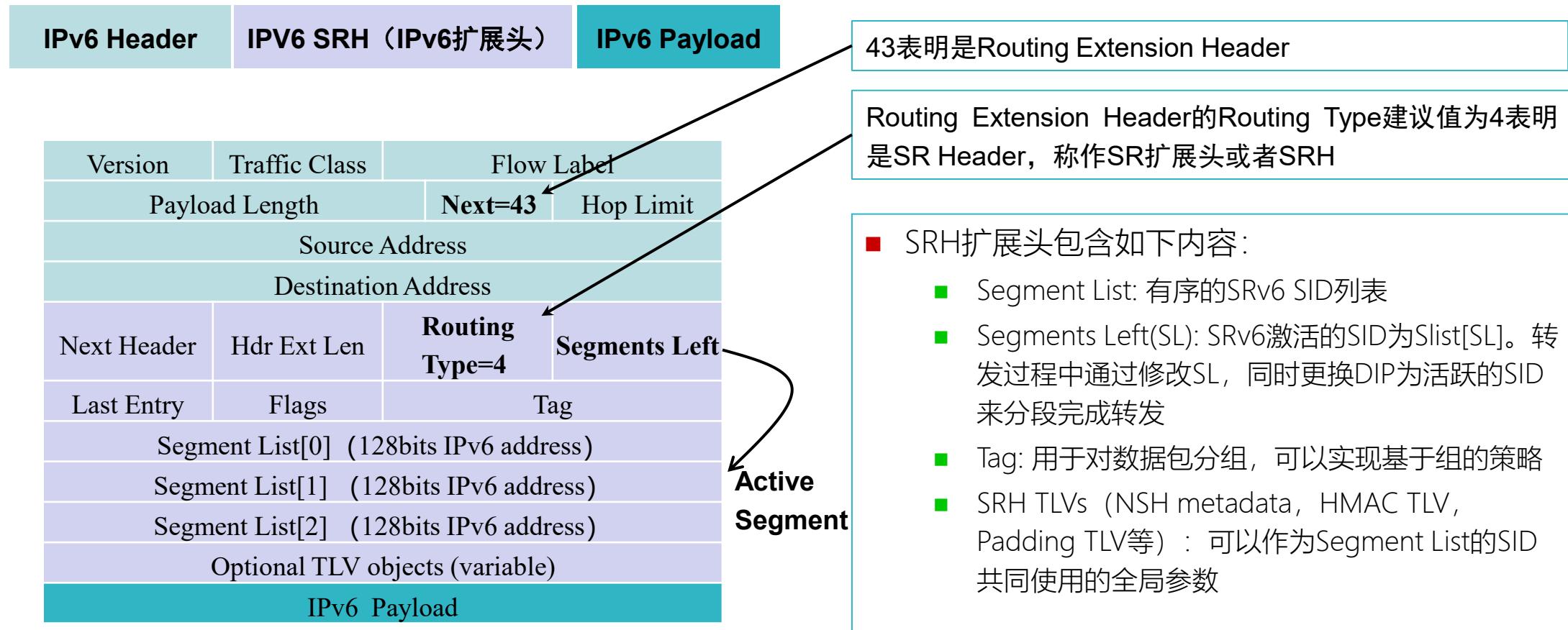


# 段路由协议SRv6

## ■ 段路由SRv6基本思想：

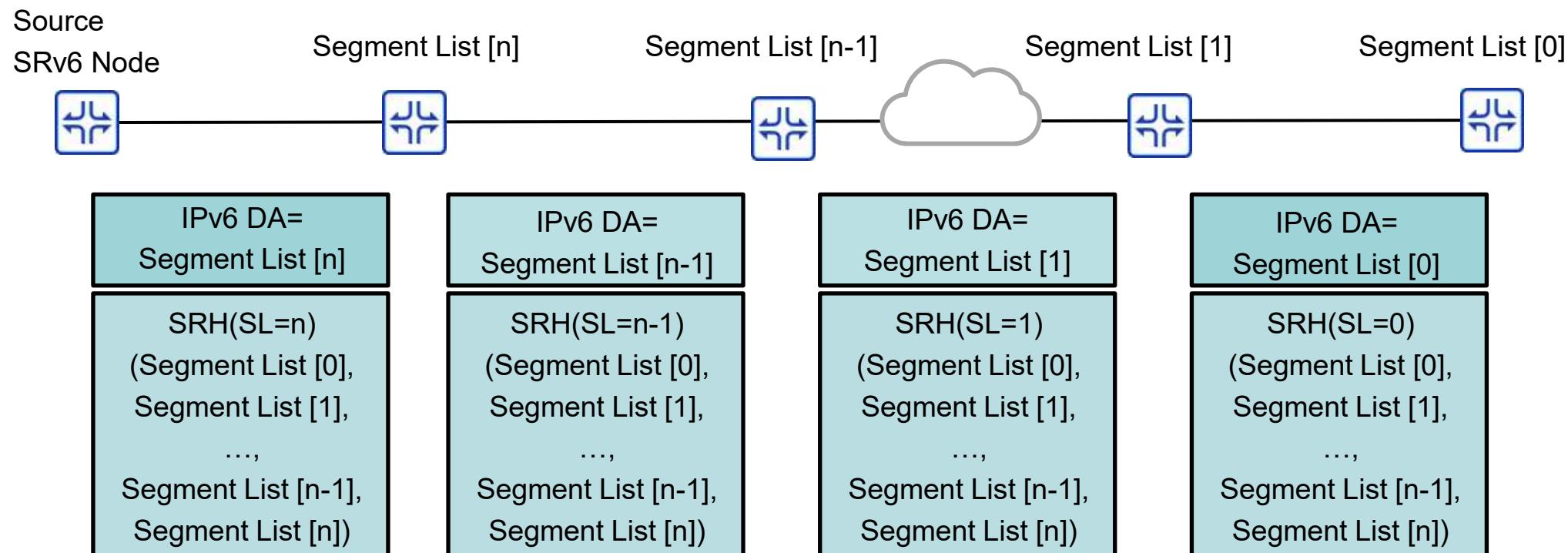
- 将数据包转发路径切割为不同的分段
- 在路径头结点处向数据包中插入分段信息
- 中间结点只需要按照数据包所携带的分段信息转发

# IPv6段路由扩展报头 (SRH)



# SRv6报文转发

- 在SRv6转发过程中每经过一个SRv6 节点，Segments Left (SL) 字段减1，IPv6 DA信息变换一次。Segments Left和Segment List字段共同决定IPv6 DA信息
- SRv6 SRH是从下到上逆序操作，SRH中的Segment在经过节点后也不会被弹出。因此SRv6报头可以做路径回溯



## 5.7 本章总结

- 网络层主要负责将数据包从信源传送到信宿，即完成信源与信宿之间端到端的报文传递
- 网络层提供了两种主要功能：交换和路由。交换是在两个或多个设备之间建立临时连接，路由选择从一点到另一点发送数据包的最佳路径
- 网络层可以为传输层提供面向连接的服务和无连接服务。在TCP/IP体系中，网络层仅向传输层提供不可靠、无连接、尽最大努力交付的数据报服务
- 网络层的主要互连设备是路由器，路由器具有多个输入端口和多个输出端口，任务是路由和转发分组。路由器的结构分为路由选择和分组转发两大部分

# 本章总结

- 理想的路由选择算法应符合正确性、简单性、坚定性、稳定性、公平性和最佳性的基本要求
- 在路由中，具有最短距离的路径是最佳的路径。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、以及这些因素的组合
- 最常用的计算最短路径的方法有两种：距离向量路由和链路状态路由。网络中实际运行的路由选择协议大多基于这两种算法（如内部网关协议RIP、OSPF，外部网关协议BGP）
- 流量控制和拥塞控制技术用于减少拥塞并提高网络性能
- 服务质量保证的技术包括过度配置、流量整形、数据包调度、资源预留等；提供服务质量保证的规范性方案包括综合服务和区分服务

# 本章总结

- IP协议为高层提供不可靠、无连接的数据报通信服务。所有的TCP、UDP、ICMP、IGMP数据都以IP数据报格式传输
- 网络中每个独立主机和路由器的每个接口必须有一个唯一的IP地址
- 早期IPv4 地址的设计不合理，造成了地址浪费以及路由表的膨胀。CIDR和路由聚合技术的使用，大大减少了路由表的条目数，提高了路由器的转发效率
- DHCP提供了主机IP地址动态分配的有效机制，NAT提供了主机内网IP地址到互联网地址的转换机制。DHCP和NAT技术的使用节约了IP地址资源，延缓了IPv4地址资源枯竭时间
- IP 协议有三个配套协议：ARP，ICMP，IGMP

# 本章总结

- IPv6具有地址空间巨大、可扩展性好等诸多优势。IPv6的配套协议ICMPv6以其强大的功能取代了ARP、IGMP和ICMPv4。随着IPv4地址耗尽，从IPv4向IPv6迁移已经成为各网络当前的工作重点
- SDN提供了可编程的网络、全局的网络视野、集中的网络控制等便于网络管理的新功能，促进了网络管理的开放化、标准化和智能化
- MPLS是为了提高网络设备转发速度提出的技术，只在网络边缘分析IP报文首部，节约了处理时间
- SRv6是基于IPv6转发平面，利用IPv6简洁易扩展头部结构，具有源路由、无状态、集中控制的特点，体现出简单、高效、易扩展的特性