

第 4 章 网络层

网络层的基本功能

IP、ICMP等协议、路由选择

网络层概述

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 为什么需要网络层?
- 网络层提供哪两种服务?

为什么需要网络层

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 数据链路层解决了同一局域网计算机间帧的传输问题，没有解决以下问题：
 - 异构网络互联，即跨局域网连接和资源共享；
 - 互联网络中主机标识问题；
 - 互联网中主机间路由选择问题（最佳路径）；
 - 互联网中数据转发的问题（分组转发）。

局域网：属于资源子网；广域网：属于通信子网。

网络层提供的两种服务

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

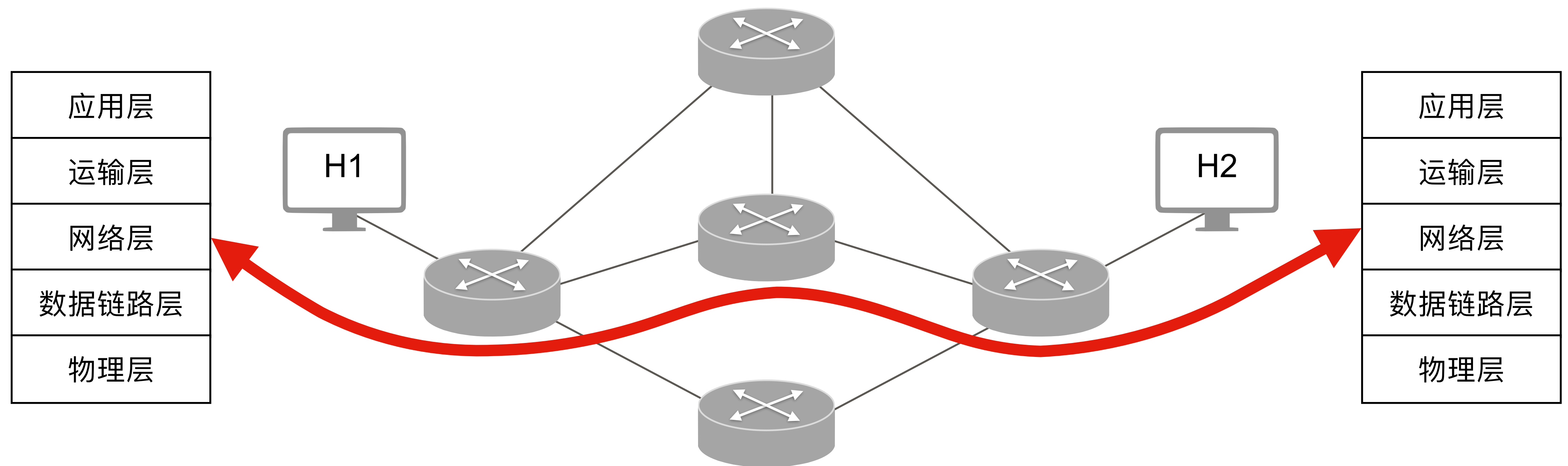
- 在计算机网络领域，曾引起了长期的争论的问题：网络层应该向运输层提供怎样的服务？
 - “面向连接”；
 - “无连接”。
 - 争论焦点的实质就是：
 - 在计算机通信中，可靠交付应当由谁来负责？是网络还是端系统？

一种观点：让网络负责可靠交付

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 这种观点认为，应借助于电信网的成功经验，让网络负责可靠交付，计算机网络应模仿电信网络，使用面向连接的通信方式：
 - 通信之前先建立虚电路 (Virtual Circuit)，以保证双方通信所需的一切网络资源；
 - 如果再使用可靠传输的网络协议，就可使所发送的分组无差错按序到达终点，不丢失、不重复。

虚电路服务



H₁ 和 H₂ 之间的所有分组都沿着**同一条虚电路**传送。

电路交换与虚电路的区别

- **工作层次不同：**

- 电路交换属于物理层概念；
- 虚电路属于网络层概念，是分组交换。

- **虚电路：**结合了电路交换和无连接分组交换的优点。

- **链路占用方式不同：**

- 电路交换的通信双方有**一条物理通路**，被通信双方独占；
- 虚电路是在一条物理线路上**虚拟出多个逻辑的通道**，此时该物理线路上可以有多条通信量（共享），每条通信**独占一条虚拟电路**。多个虚拟电路通过**时分/频分**等多路复用方式复用到一条物理链路上，实现了通信双方**虚拟的“点到点的”连接**。

另一种观点：网络提供数据报服务

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 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。

互联网的先驱者提出了一种崭新的网络设计思路。

两种观点代表

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- OSI:
 - 根据唯一的网络设备地址路由数据包，提供流量和拥塞控制以防止网络资源的损耗（虚电路服务）。
- TCP/IP:
 - 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（数据报服务）。

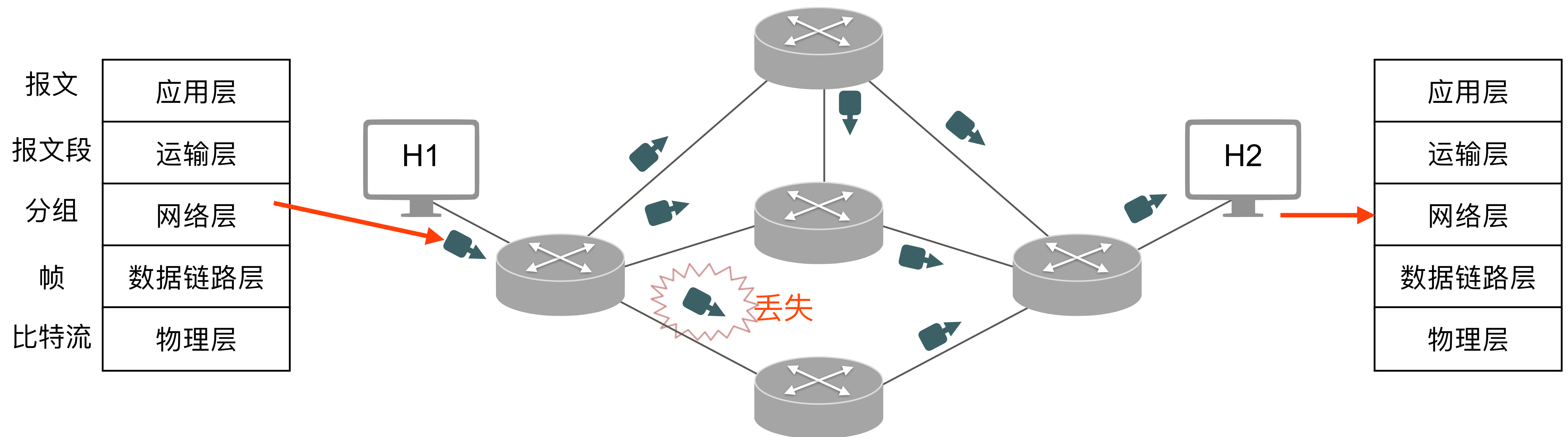
尽最大努力交付

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 由于传输网络不提供端到端的可靠传输服务，网络中的路由器可以做得比较简单，而且价格低廉：
 - 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可靠交付（包括差错处理、流量控制等）；
 - 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，适应多种应用。

为什么不让网络层保证可靠性？

数据报服务：无连接



H1 发送给 H2 的分组可能沿着不同路径传送

网络层传输数据单位：IP数据报、分组

虚电路服务与数据报服务的对比

对比的方面	虚电路服务	数据报服务
可靠性	可靠通信应当由网络来保证	可靠通信应当由用户主机来保证
连接的建立	必须有	不需要
终点地址	仅在连接建立阶段使用，每个分组使用短的虚电路号	每个分组都有终点的完整地址
路由选择	属于同一条虚电路的分组按照同一路由进行转发	每个分组独立选择路由进行转发
网络故障适应性	所有通过出故障的结点的虚电路均不能工作	故障的结点会丢失分组，一些路由会发生变化
分组的顺序	总是按发送顺序到达终点	到达终点时不一定按发送顺序
差错处理和流量控制	可以由网络负责，也可以由用户主机负责	由用户主机负责

小结

- 网络层
 - 网络层概述
 - 网络层提供服务
 - 两种观点
 - 尽最大努力交付

- 网络层的基本功能。
- 网络层提供的两种服务：
 - 虚电路；
 - 数据报；
 - 两种服务的对比。

网际协议 IP

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备

- 虚拟互连网络。
- 分类的 IP 地址。
- IP 地址与硬件地址。
- 地址解析协议 ARP。
- IP 数据报的格式。
- IP 层转发分组的流程 。

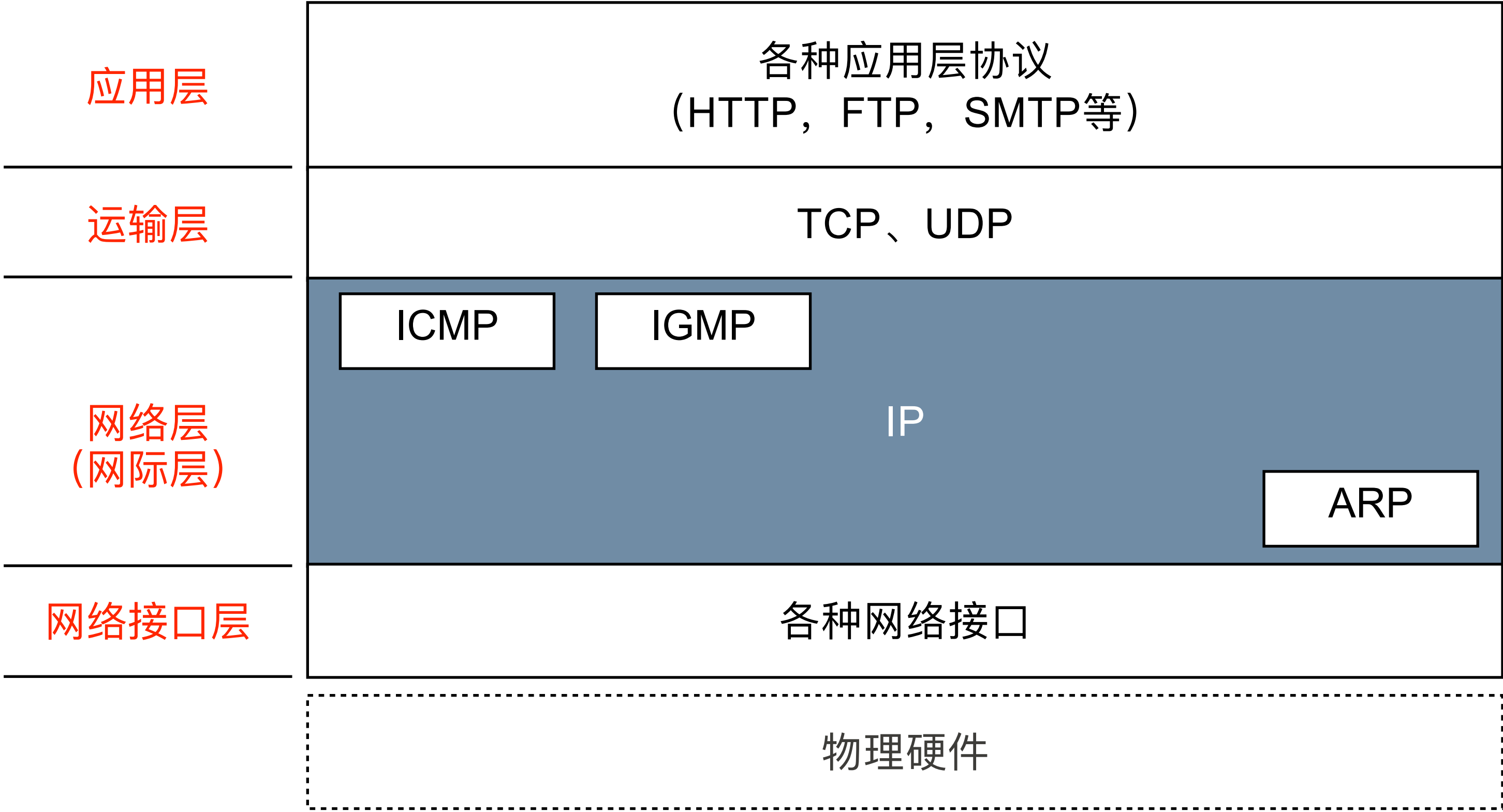
网际协议 IP

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备

- 网际协议 IP 是 TCP/IP 体系中两个最主要的协议之一。
- 与 IP 协议配套使用的还有三个协议：
 - 地址解析协议 ARP；
 - 网际控制报文协议 ICMP；
 - 网际组管理协议 IGMP。

网际层的 IP 协议及配套协议

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备



虚拟互连网络

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备

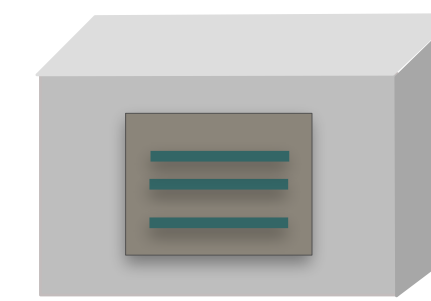
- 不同的寻址方案；
- 不同的最大分组长度；
- 不同的网络接入机制；
- 不同的超时控制；
- 不同的差错恢复方法。
- 不同的状态报告方法；
- 不同的路由选择技术；
- 不同的用户接入控制；
- 不同的服务（面向连接服务和无连接服务）；
- 不同的管理与控制方式等。

将网络互连并能够互相通信，会遇到许多问题需要解决。

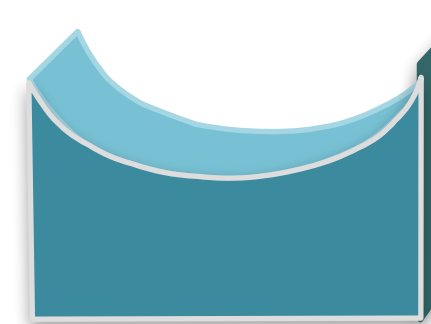
使用一些中间设备进行互连

- 有以下五种不同的中间设备：
 - **物理层**中继系统：转发器 (repeater, 中继器);
 - **数据链路层**中继系统：网桥 或 桥接器 (bridge);
 - **网络层**中继系统：路由器 (router);
 - 网桥和路由器的**混合物**：桥路器 (brouter);
 - **网络层**以上的中继系统：网关 (gateway)。

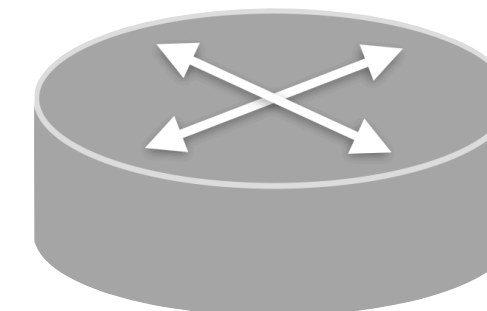
将网络互相连接起来要使用一些中间设备。
中间设备又称为**中间系统或中继** (relay)系统。



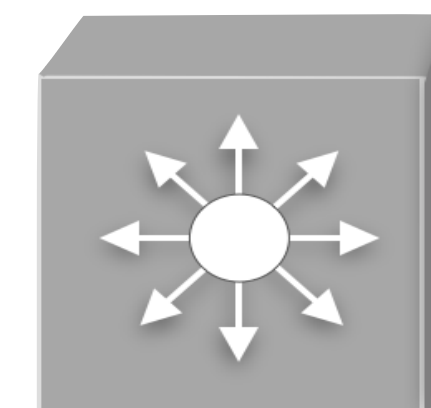
中继器



网桥



路由器



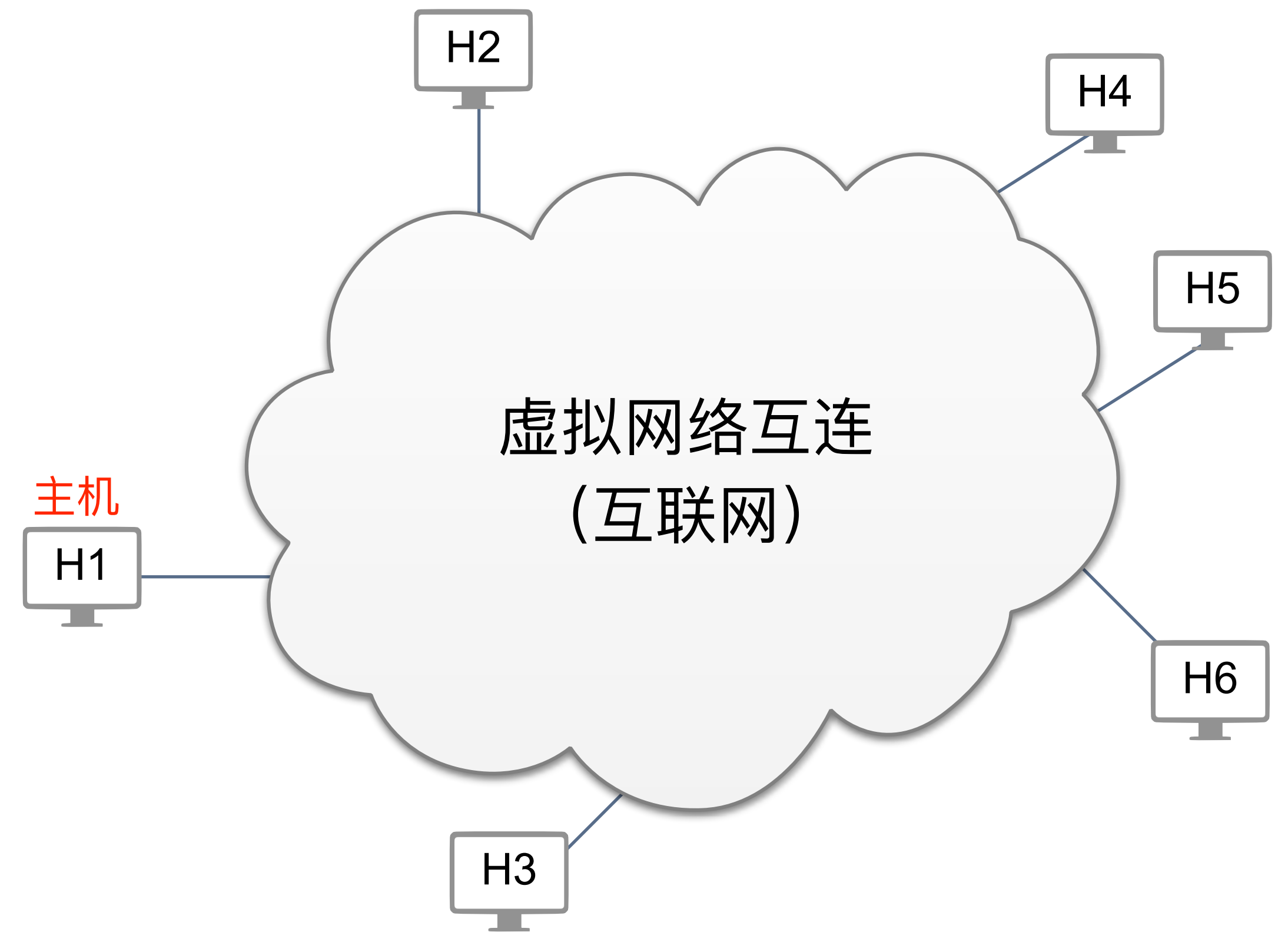
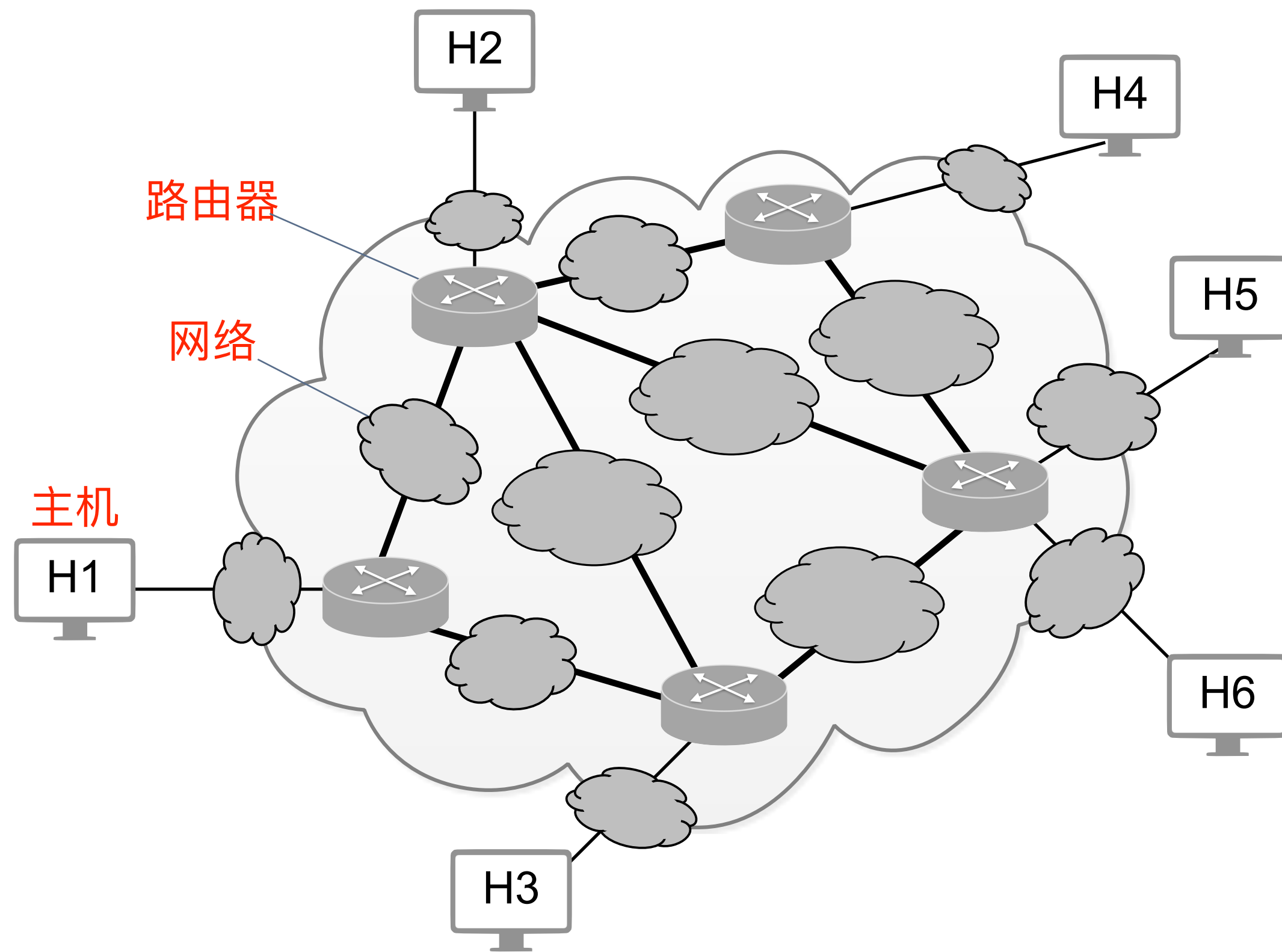
三层交换机

网络互连使用路由器

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备

- 当中继系统是转发器或网桥时，一般并不称之为网络互连，仅仅是把一个网络扩大了，而这仍然是一个网络（局域网）。
- 网关由于比较复杂，目前使用得较少。
- 由于历史的原因，许多有关 TCP/IP 的文献将网络层使用的路由器称为网关。
- 网络互连都是指用路由器进行网络互连和路由选择。

互连网络与虚拟互连网络

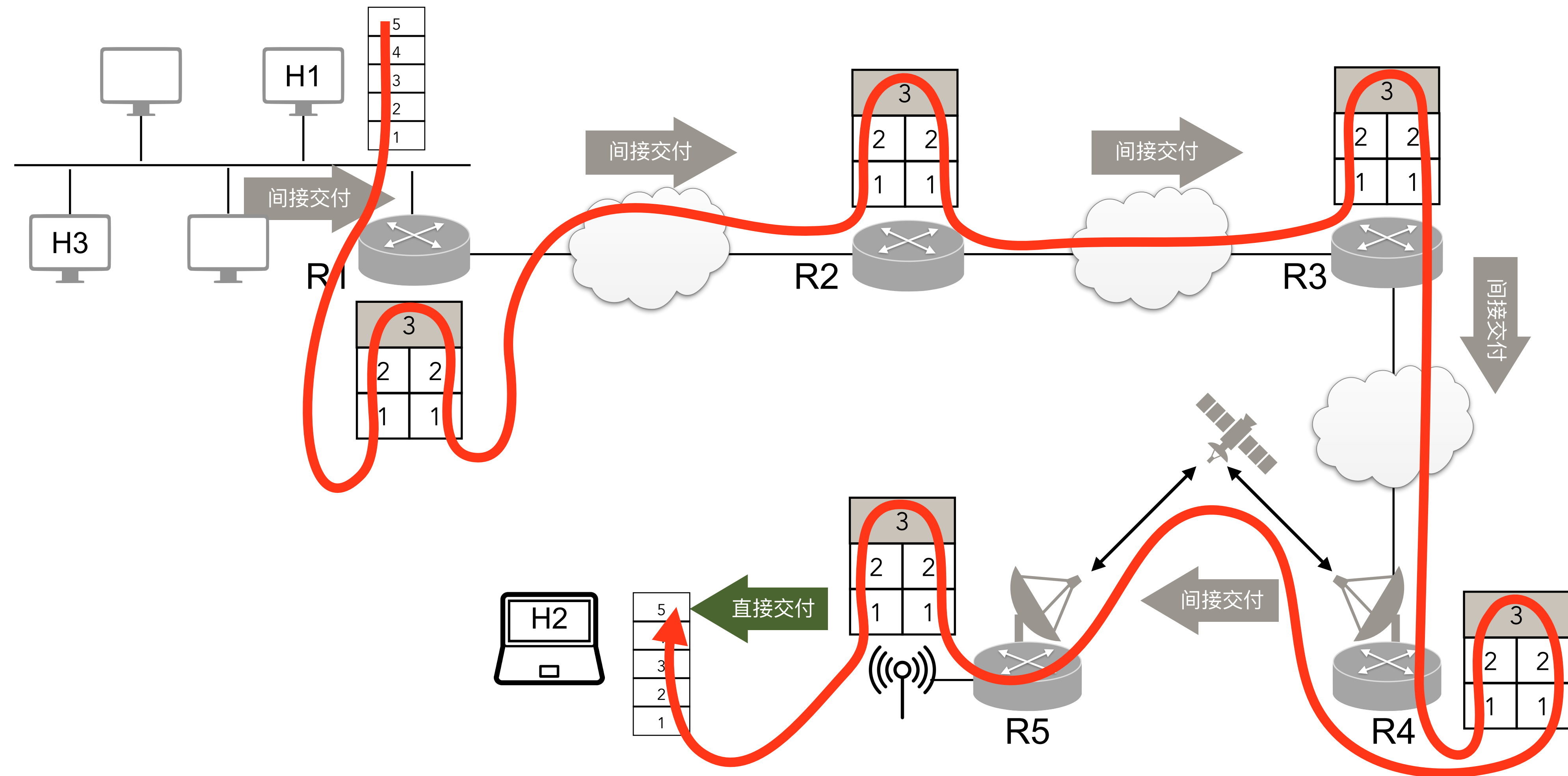


虚拟互连网络的意义

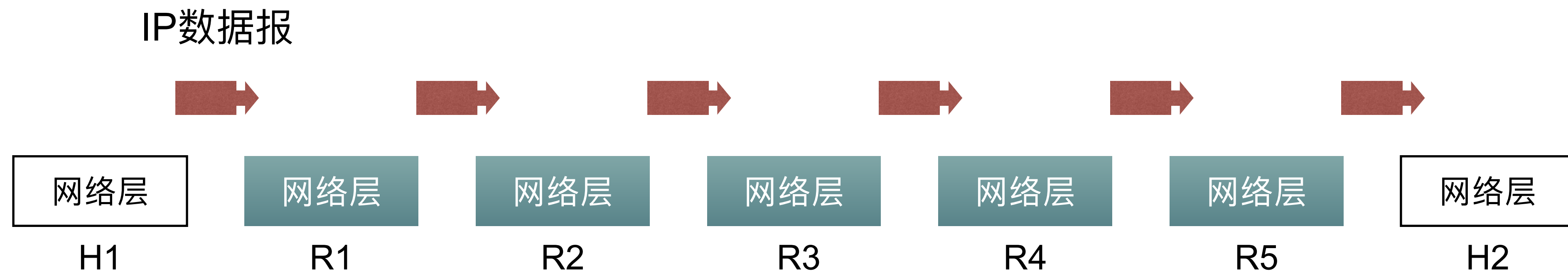
- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备

- 所谓虚拟互连网络也就是逻辑互连网络：
 - IP 协议可以使这些性能各异的网络从用户看起来好像是一个统一的网络；
 - 使用 IP 协议的虚拟互连网络可简称为 IP 网。
- 使用虚拟互连网络的好处是：
 - 当互联网上的主机进行通信时，就好像在一个网络上通信一样，而看不见互连的各具体的网络异构细节。

分组在互联网中的传送



从网络层看 IP 数据报的传送



如果从网络层考虑问题，IP 数据报就可以想象是在网络层中传送。

小结

- 网络层
 - 网际协议IP
 - 配套协议
 - 虚拟网络互连
 - 网络互连设备
 - 分组传输过程

- 虚拟互连网络。
- 网络互连中继设备：
 - 中继器、网桥（交换机）、路由器。
- 网络层协议：
 - ICMP、IGMP、IP、ARP。
 - 分组在网络中的传输过程。

IP 地址

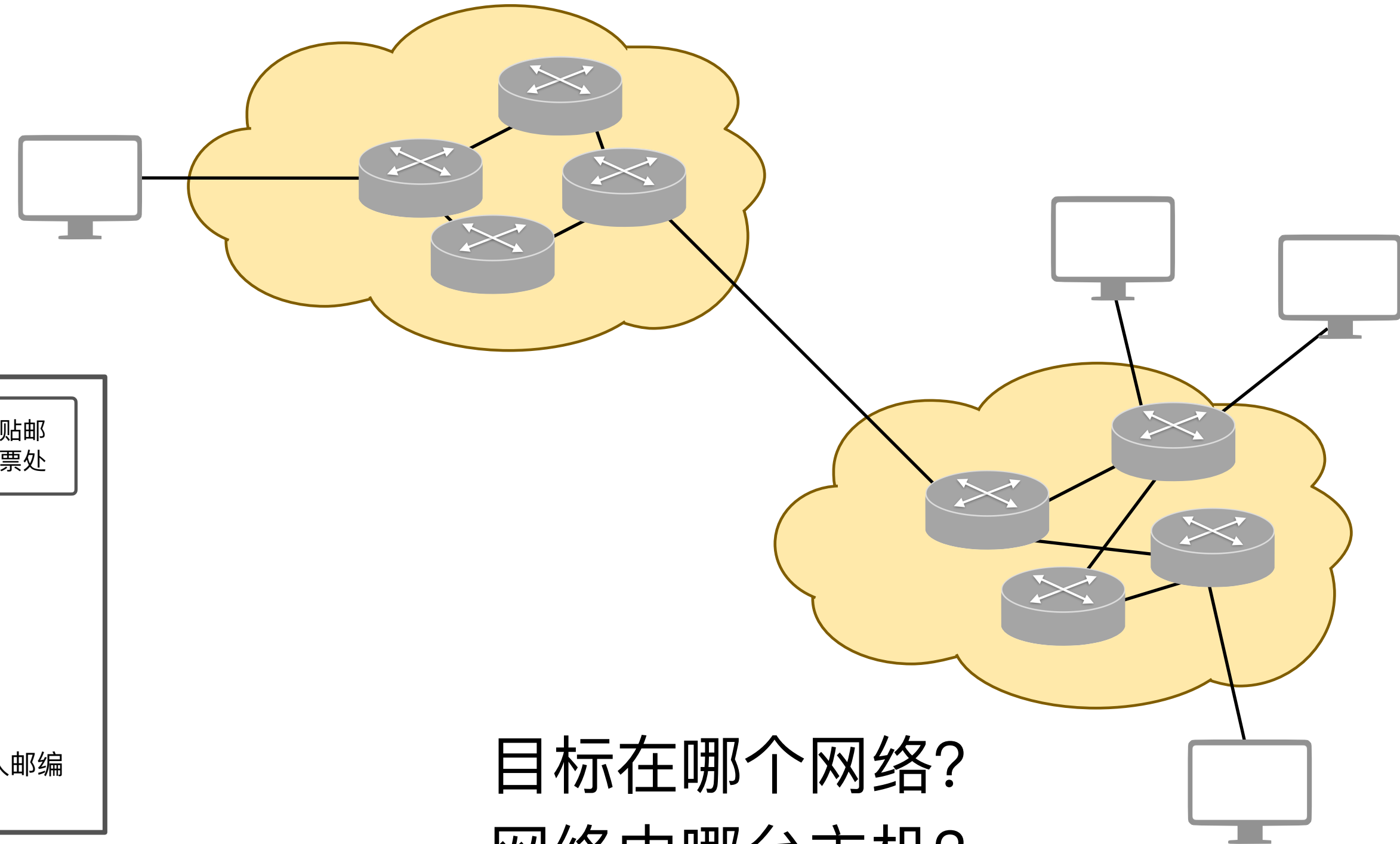
- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

- IP 地址及其表示方法。
- 常用的三种类别的 IP 地址。

为什么需要IP地址

寄往哪里？
什么人收？

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	贴邮 票处
收信人邮编							
收信人地址							
收信人姓名							
寄信人地址、姓名							
寄信人邮编							
邮政编码：							



目标在哪个网络？
网络中哪台主机？

IP 地址及其表示方法

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

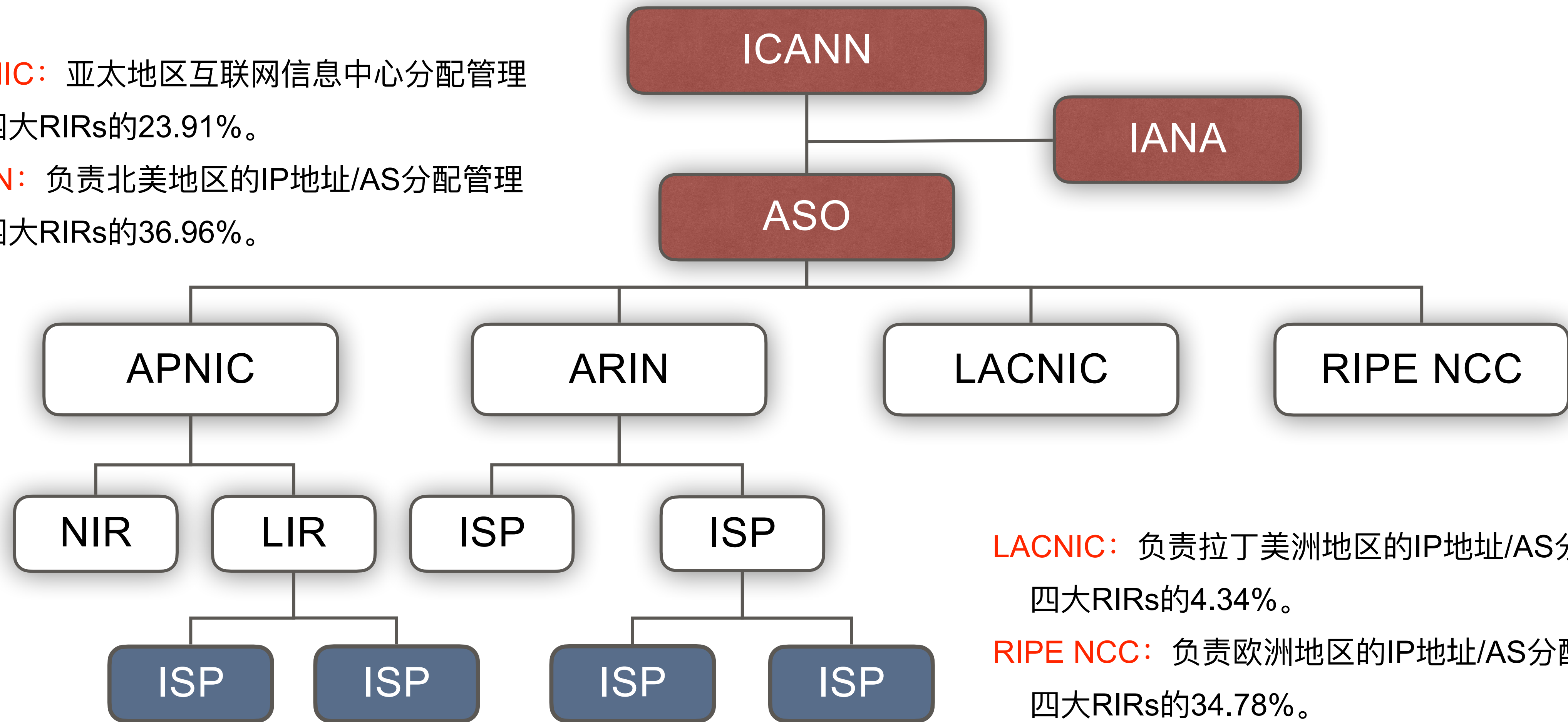
- 我们把整个因特网看成为一个单一的、抽象的网络：
 - IP 地址就是给每个连接在互联网上的主机（或路由器）分配一个在全世界范围是唯一的 32 位的标识符；
 - IP 地址由互联网名字和数字分配机构ICANN (Internet Corporation for Assigned Names and Numbers)进行分配。

<http://www.icann.org>

ICANN (2004年)

APNIC: 亚太地区互联网信息中心分配管理
四大RIRs的23.91%。

ARIN: 负责北美地区的IP地址/AS分配管理
四大RIRs的36.96%。



LACNIC: 负责拉丁美洲地区的IP地址/AS分配管理
四大RIRs的4.34%。

RIPE NCC: 负责欧洲地区的IP地址/AS分配管理
四大RIRs的34.78%。

RIRs: 区域互联网注册管理机构

CNNIC

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

- 北京计算机应用研究所于1993年6月4日从APNIC处获得了一个C类的IP地址。
- CNNIC以国家互联网络注册机构（NIR）的身份于1997年1月成为APNIC的联盟会员，成立了以CNNIC为召集单位的分配联盟，已为多家ISP提供IP地址：
 - 中国联合通信有限公司、中国卫星通信集团公司、总参通信部、中国科技网络中心、中国国际电子商务中心、中国铁通集团公司等单位的全部IP地址；
 - 中国电信集团公司、中国网络通信集团公司、中国移动通信集团公司的部分下属单位的IP地址。

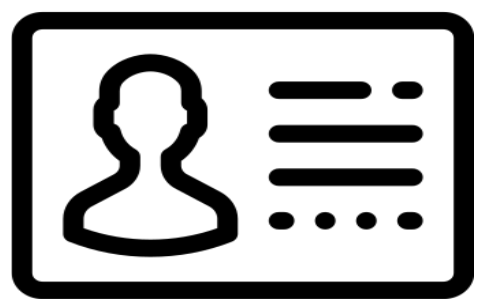
<http://www.cnnic.net.cn>

IP 地址的编址方法

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

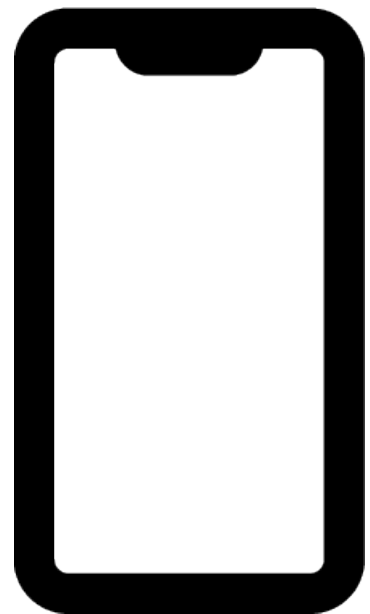
- 分类的 IP 地址：
 - 最基本的编址方法，在 1981 年就通过了相应的标准协议。
- 子网的划分：
 - 对最基本的编址方法的改进，其标准[RFC 950] 在 1985 年通过。
- 构成超网：
 - 比较新的无分类编址方法。1993 年提出后很快就得到推广应用。

分类IP地址



身份证号码

110000 19XX0328 030 8
行政区代码 出生日期码 顺序码 校验码



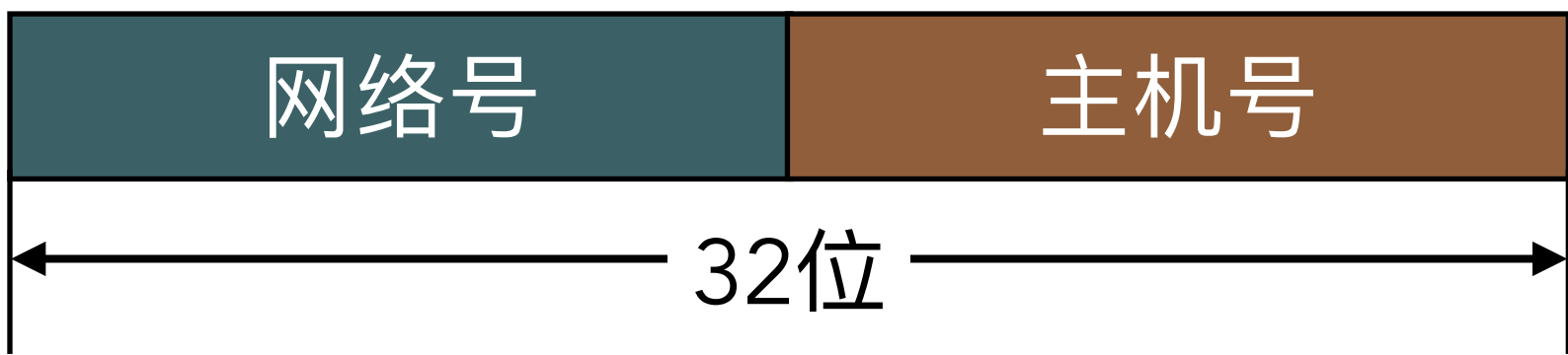
手机号码

137 0001 XXXX
网络号 区号 手机号



IP地址

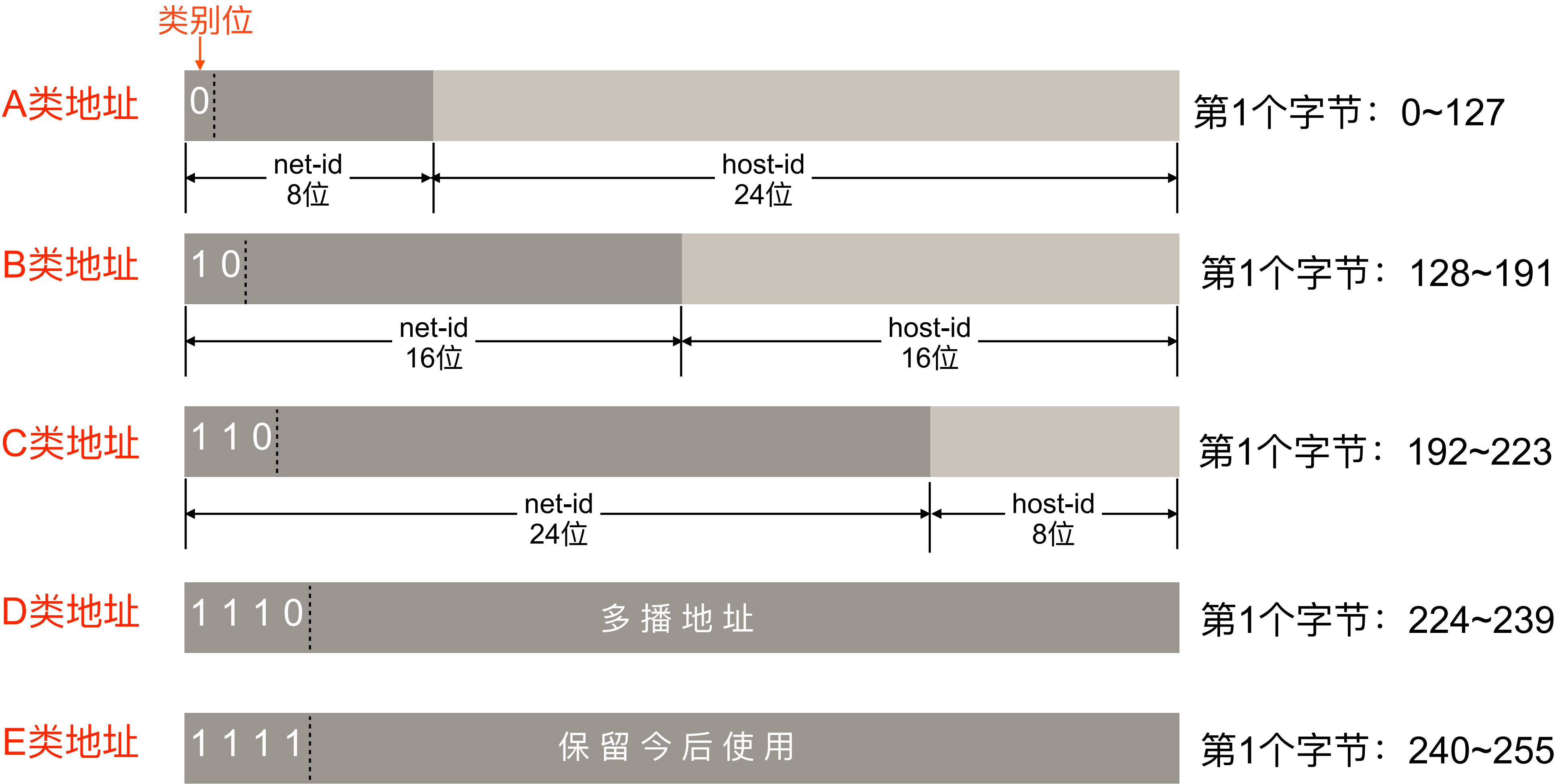
192 160 100 1
11000000 10100000 01100100 00000001
网络号 主机号



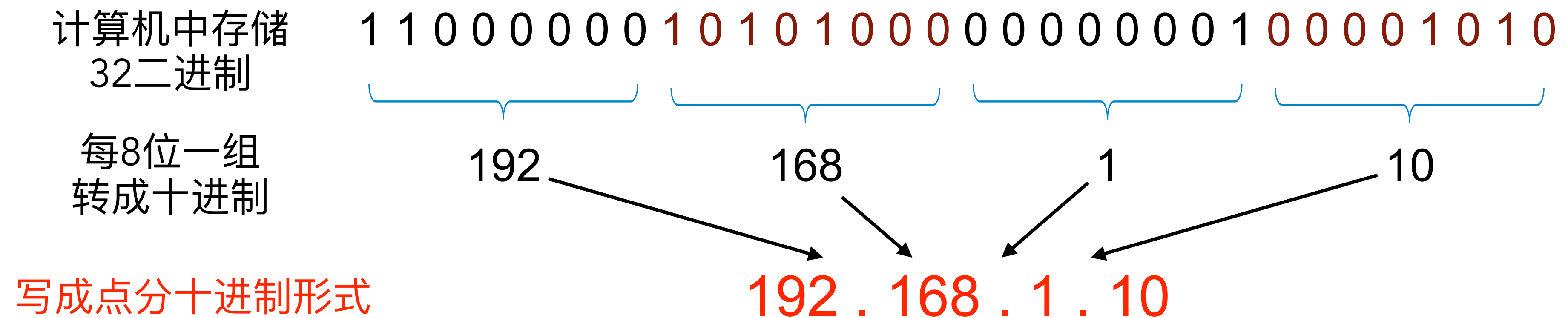
将IP地址划分为若干个固定类

IP地址 ::= {<网络号>, <主机号>}

各类 IP 地址的网络号字段和主机号字段



点分十进制记法



采用点分十进制记法，可以提高可读性

八位二进制与十进制间转换

序号	7	6	5	4	3	2	1	0
二进制数位	1	1	1	1	1	1	1	1
十进制数	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1

10000000 → 128

11000000 → 192

11100000 → 224

11110000 → 240

11111000 → 248

11111100 → 252

11111110 → 254

11111111 → 255

221 = 128 + 64 + 0 + 16 + 8 + 4 + 0 + 1

1 1 0 1 1 1 0 1

255 - 221 = 34 = 32 + 2 (对应位置0, 其余为1)

常用的三类类别的 IP 地址

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

网络类别	网络数	第1个网络号	最后一个网络号	每个网络中主机数
A	126 ($2^7 - 2$)	1	126	16777214
B	16383 ($2^{14} - 1$)	128.1	191.255	65534
C	2097151 ($2^{21} - 1$)	192.0.1	223.255.255	254

不指派的网络：全0的网络； 127.0.0.0； 128.0.0.0； 192.0.0.0。

一般不使用的特殊的 IP 地址

网络号	主机号	源地址使用	目的地址使用	代表的意思
0	0	可以	不可以	在本网络上的本主机
0	host-id	可以	不可以	本网络上的某台主机
全1	全1	不可以	可以	在本网络上进行广播，路由器不转发
net-id	全0	不可以	不可以	网络地址，表示一个网络
net-id	全1	不可以	可以	对net-id上的所有主机广播
127	非全0或全1的任何数	可以	可以	用作本地软件环回测试用

<https://tools.ietf.org/html/rfc1122#page-29>, 3.2.1.3

Specified host on this network. It MUST NOT be sent, except as a source address as part of an initialization procedure by which the host learns its full IP address.

分类IP地址的优点

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

- IP 地址是一种分等级的地址结构。分两个等级的好处是：
 - 第一，IP 地址管理机构在分配 IP 地址时只分配网络号，而剩下的主机号则由得到该网络号的单位自行分配。这样就方便了 IP 地址的管理；
 - 第二，路由器仅根据目的主机所连接的网络号来转发分组（而不考虑目的主机号），这样就可以使路由表中的项目数大幅度减少，从而减小了路由表所占的存储空间。

IP 地址的一些重要特点

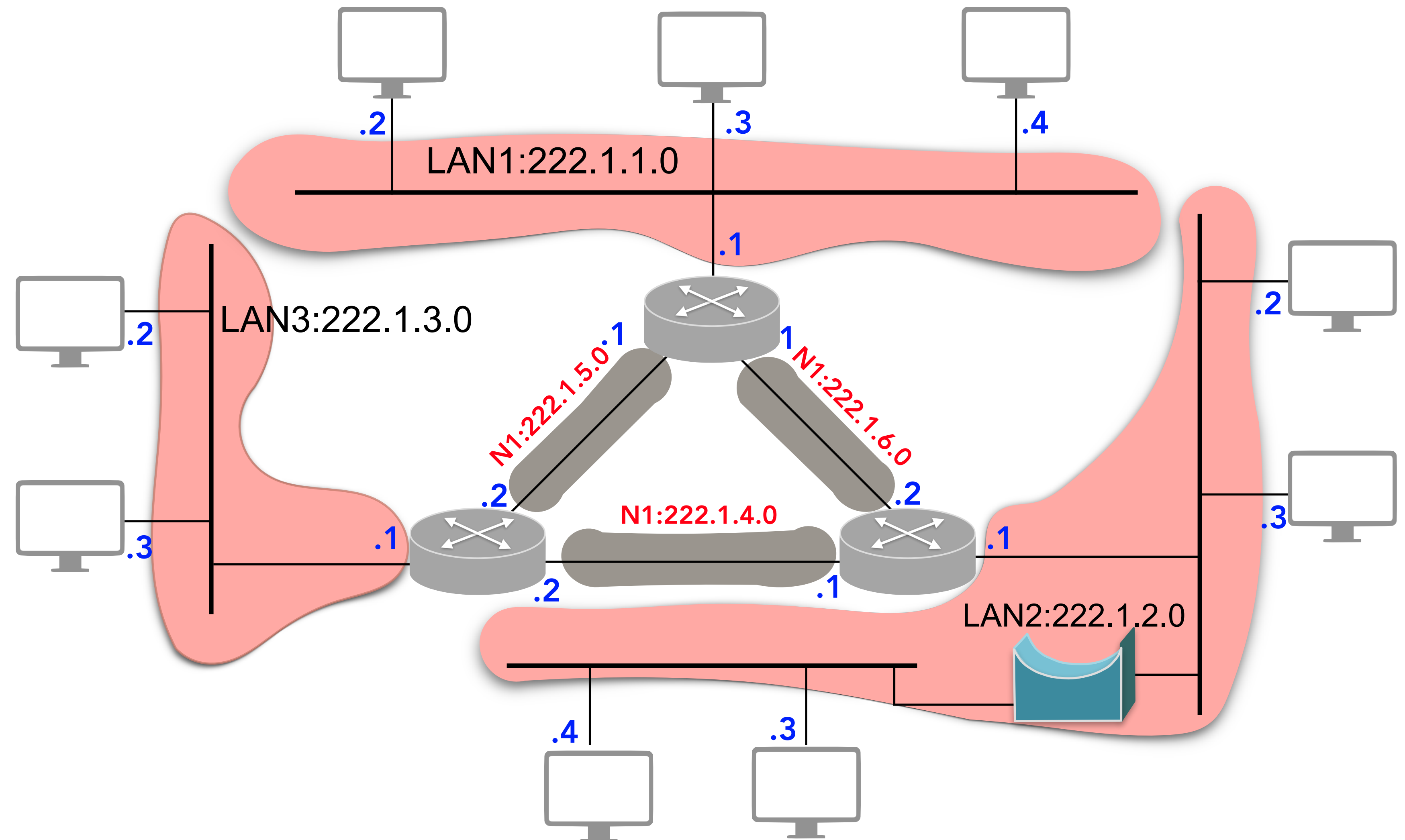
- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址
 - IP地址的特点
 - 互联网上的IP地址

- 一个主机同时连接到两个网络上时，该主机必须具有两 IP 地址，其网络号 net-id 必须是不同的。这种主机称为多归属主机 (multihomed host):
 - 由于一个路由器至少应当连接到两个网络（这样它才能将 IP 数据报从一个网络转发到另一个网络），因此一个路由器至少应当有两个不同的 IP 地址；
 - 用转发器或网桥连接起来的若干个局域网仍为一个网络，因此这些局域网都具有同样的网络号 net-id；
 - 所有分配到网络号 net-id 的网络，无论是范围很小的局域网，还是可能覆盖很大地理范围的广域网，都是平等的。

实际上 IP 地址是标志一个主机（或路由器）和一条链路的接口

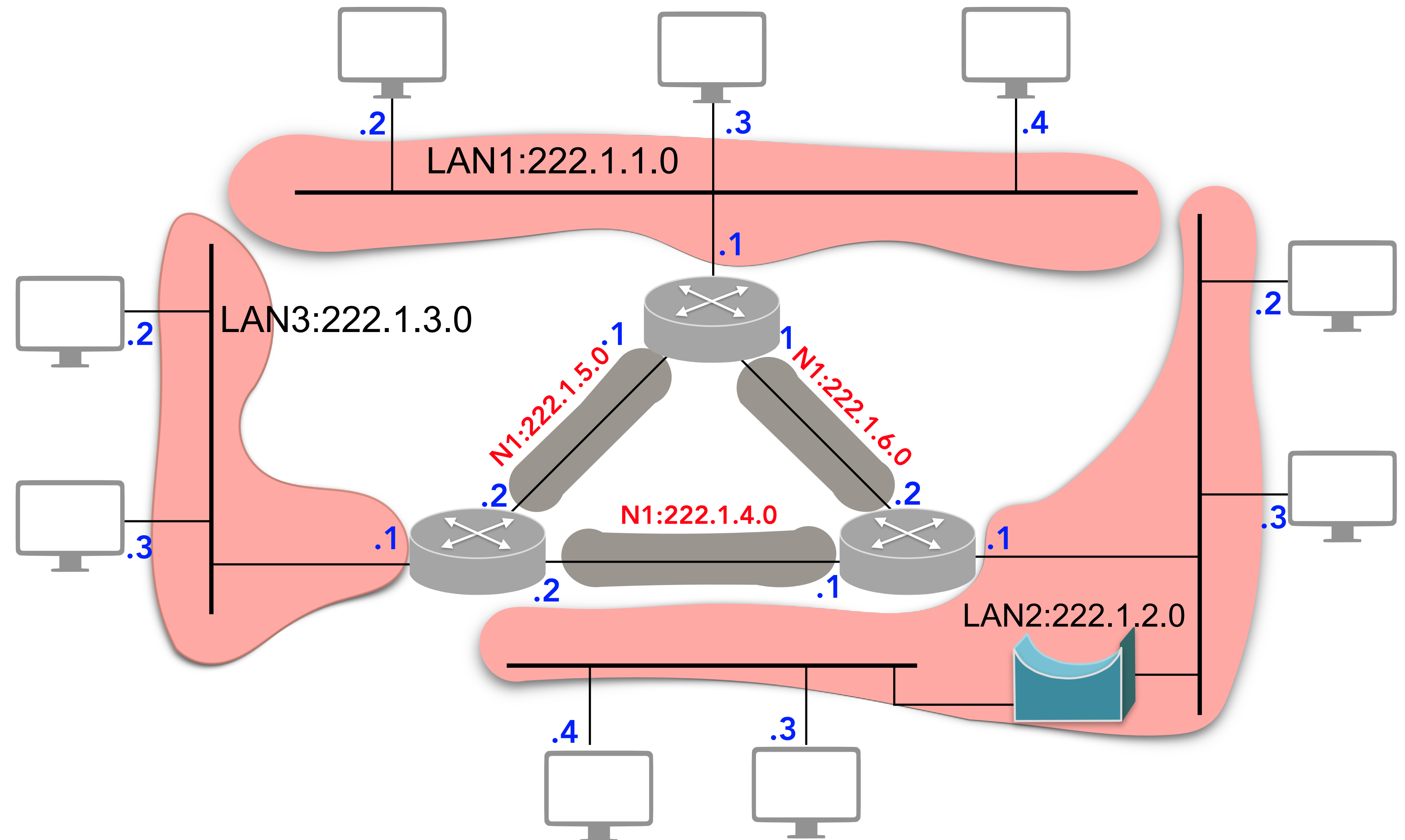
互联网中的 IP 地址

- 同一个局域网上的主机或路由器的IP 地址中的网络号必须是一样的。
- 图中的网络号就是 IP 地址中的 net-id。



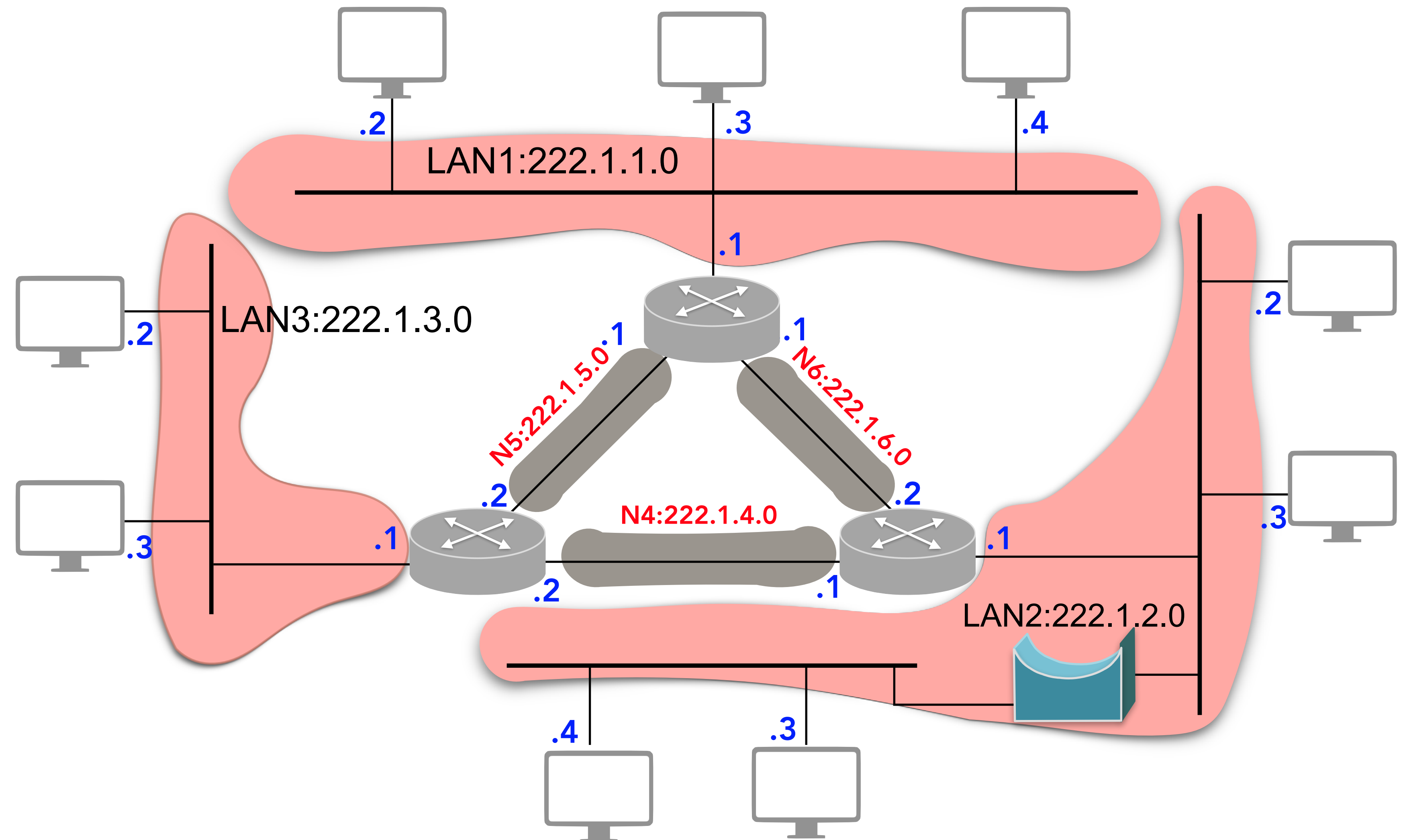
互联网中的 IP 地址

- 路由器总是具有两个或两个以上的IP地址，并且每一个接口都有一个不同网络号的IP地址：
 - 222.1.1.1
 - 222.1.5.1
 - 222.1.6.1



互联网中的 IP 地址

- 两个路由器直接相连的接口，可指明也可不指明IP地址。如指明IP地址，则这一段连线就构成了一种只包含一段线路的**特殊“网络”**。



小结

- 网络层
 - IP地址概述
 - 分配机构
 - 编址方法
 - 分类IP地址

- IP地址的编号方法：
 - 分类IP地址、子网的划分、构成超网。
- 分类IP地址：
 - A类、B类、C类地址范围；
 - 点分十进制的IP地址表示。
- 特殊IP地址。
- IP地址的一些特点：
 - 分等级的地址结构；
 - 标志一个主机和一条链路的接口；
 - 中继器或网桥连接起来的网络仍为一个网络（网络号相同）；
 - 拥有net-id的网络都是平等的。

IP地址与硬件地址

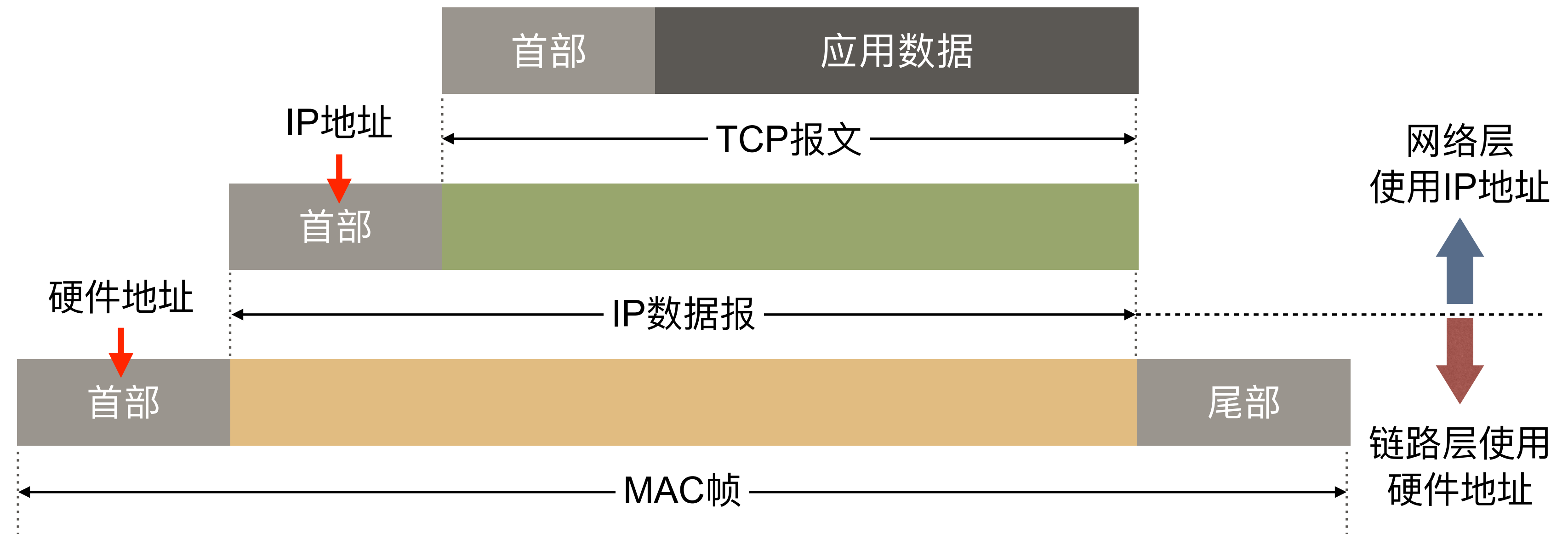
- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP缓存
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况

- 为什么不用硬件地址直接通信？
 - 硬件地址用于直接相连的网络（同一个二层广播网络），用于找到局域网中的主机；
 - 互联网中很多局域网是异构的，硬件地址不同，需要地址转换；
 - IP地址是软件地址或逻辑地址，用于定位主机所处的网络（网络号不同的网络），连接到互联网上的主机都有一个唯一的IP地址。

- 硬件地址与物理位置无关（不同国别身份证号码编码方式不同）。
- IP地址（非保留IP地址，用于Internet中的地址），与“物理位置”相关（通信地址）。

IP地址与硬件地址

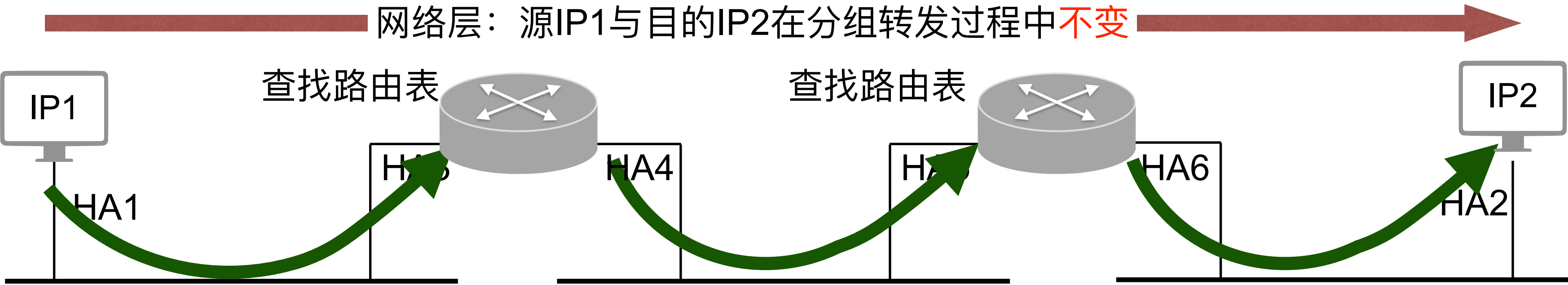
- 网络层
 - IP地址硬件地址
- ARP协议
- ARP缓存
- ARP报文格式
- ARP工作流程
- 注意问题
- 使用ARP的四种情况



IP 地址放在 IP 数据报的首部，而硬件地址则放在 MAC 帧的首部

IP地址与硬件地址

网络层IP数据报流动



数据链路层数据帧流动

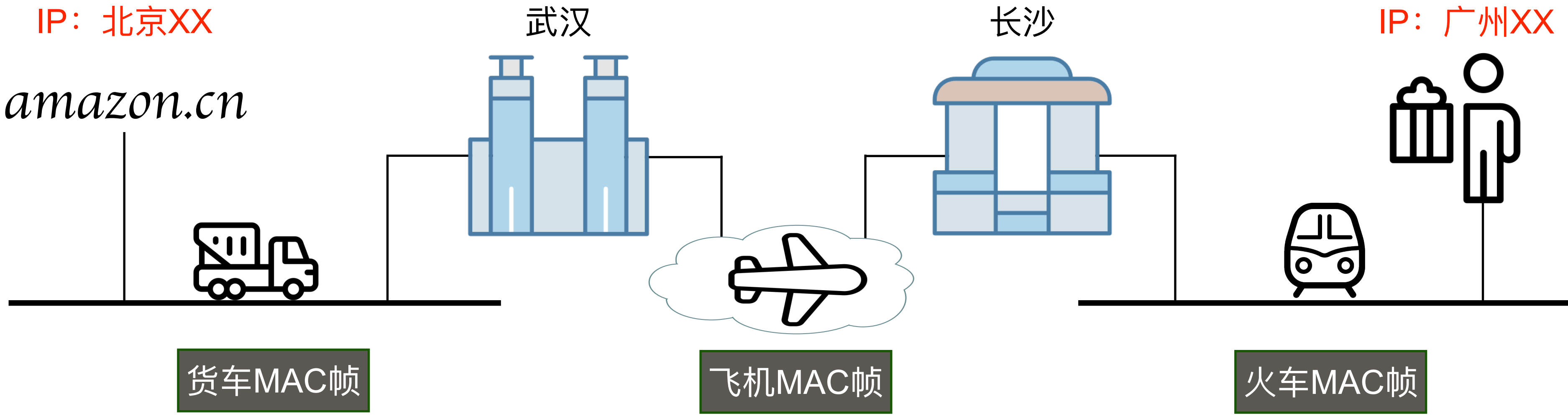
MAC：从H1到H3

MAC：从H4到H5

MAC：从H6到H2

数据链路层：源MAC与目的MAC地址在转发过程中发生**变化**

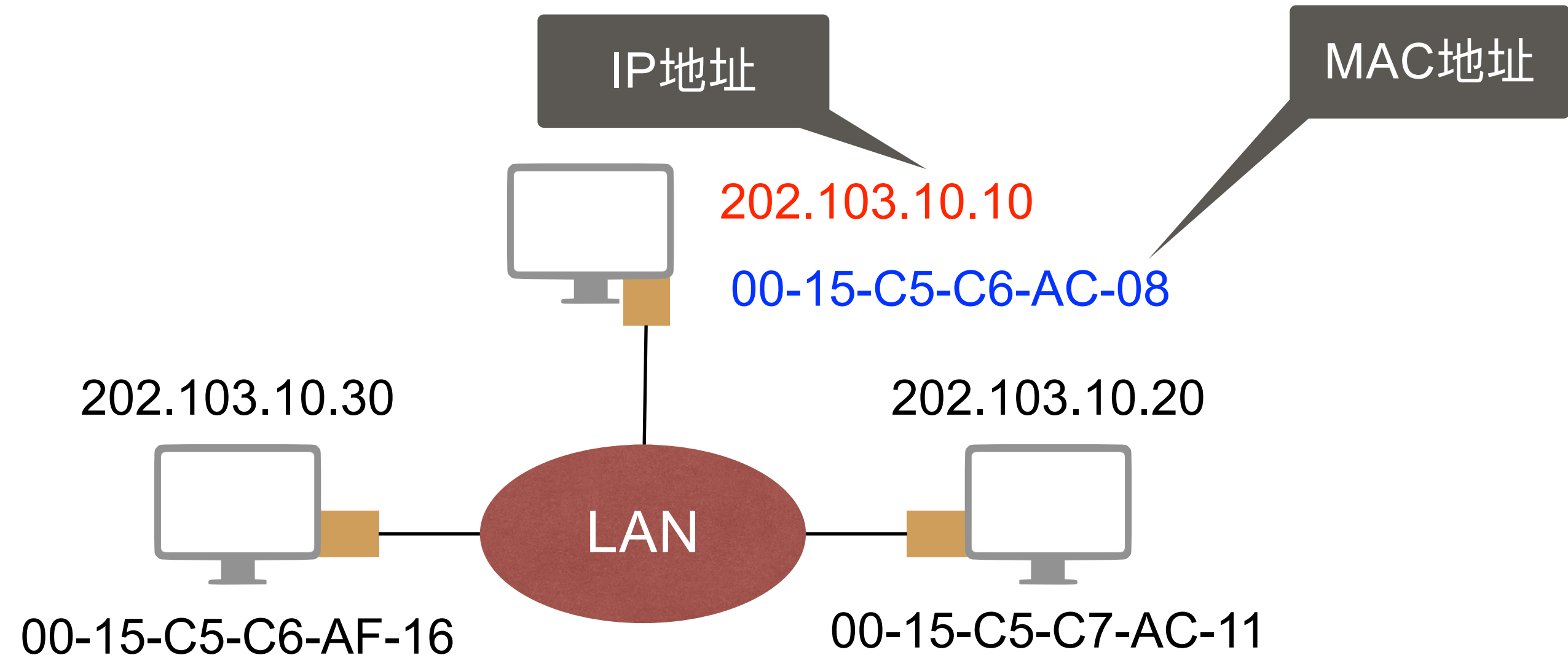
货物在各物流分公司间转发过程



IP地址与硬件地址

- 网络层
 - IP地址硬件地址
- ARP协议
- ARP缓存
- ARP报文格式
- ARP工作流程
- 注意问题
- 使用ARP的四种情况

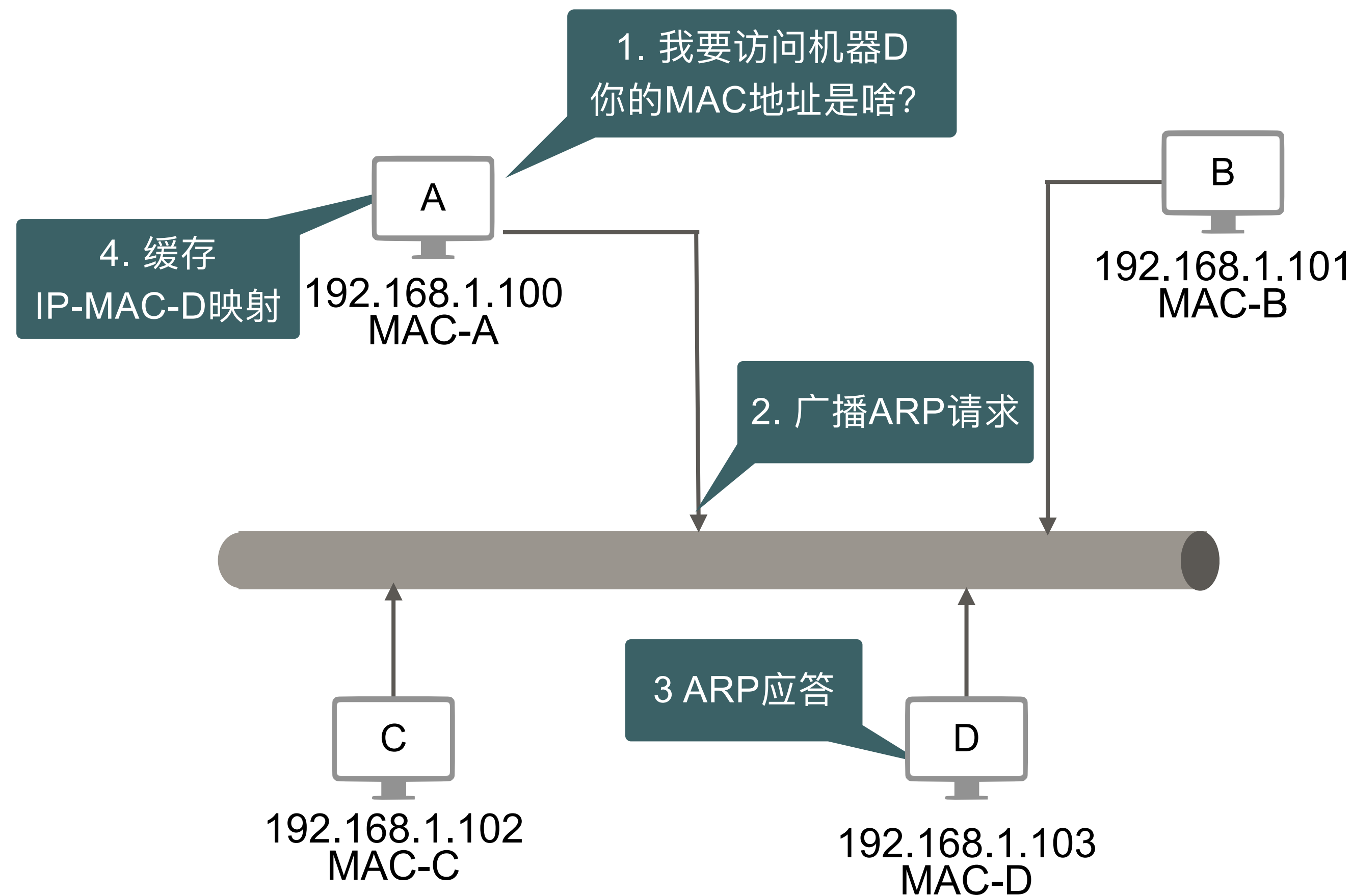
- 计算机通信时使用了两个地址：
 - IP 地址（网络层地址）；
 - MAC 地址（数据链路层地址）。



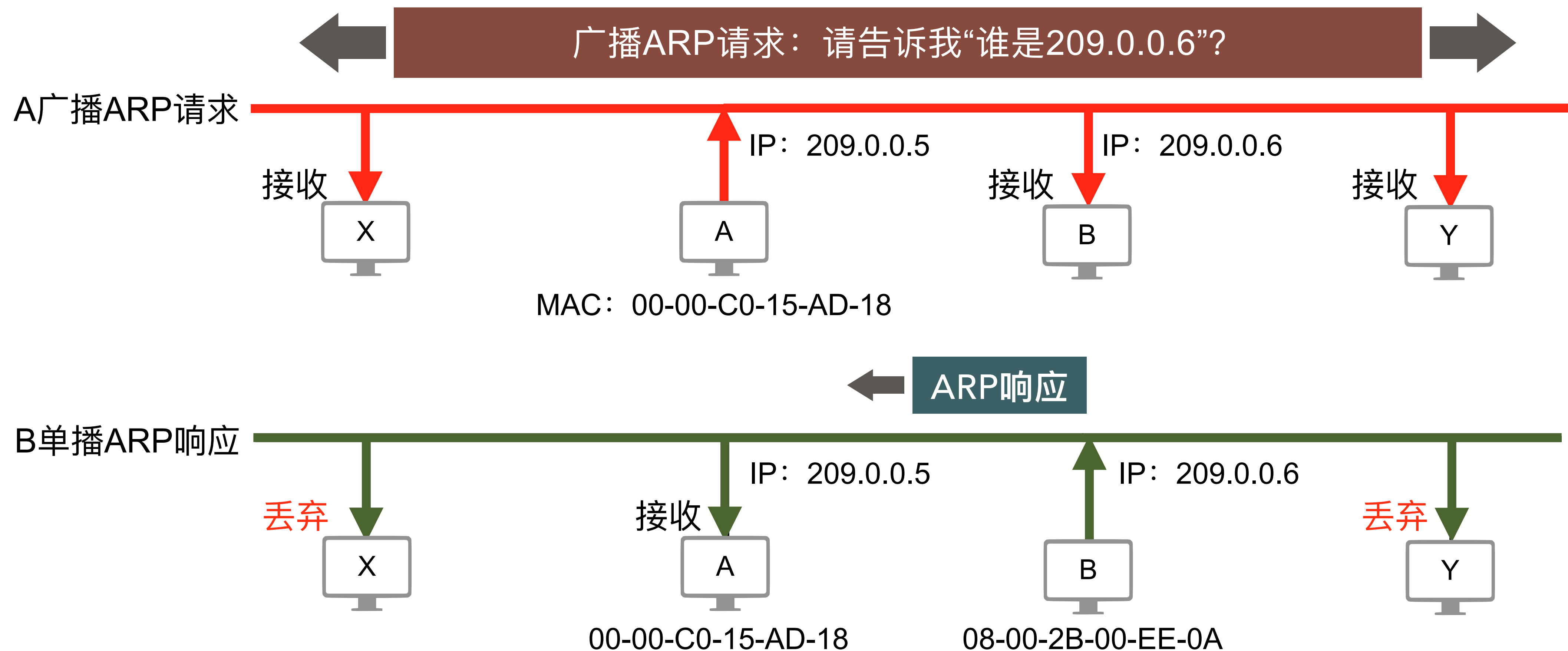
ARP协议的作用

- 根据网络层使用的 **IP 地址**，解析出在数据链路层使用的**硬件地址**。
- ARP的**作用范围**：直连的网络（同一个二层广播网络）。

我的是硬件地址： MAC-A	机器DMAC是啥？
我的IP地址： 192.168.1.100	目标IP为： 192.168.1.103



ARP协议的作用



ARP请求以广播帧形式发送，所有主机接收该帧；
ARP响应以单播帧形式发送，只有目标主机接收。

ARP 高速缓存的作用

- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP缓存
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况
- 存放最近获得的 IP 地址到 MAC 地址的绑定，以减少 ARP 广播的数量：
 - 为了减少网络上的通信量，主机 A 在发送其 ARP 请求分组时，就将自己的 IP 地址到硬件地址的映射写入 ARP 请求分组；
 - 当主机 B 收到 A 的 ARP 请求分组时，就将主机 A 的这一地址映射写入主机 B 自己的 ARP 高速缓存中。这对主机 B 以后向 A 发送数据报时就更方便了。

ARP缓存实例

- 网络层
 - IP地址硬件地址
 - ARP协议
 - **ARP缓存**
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况

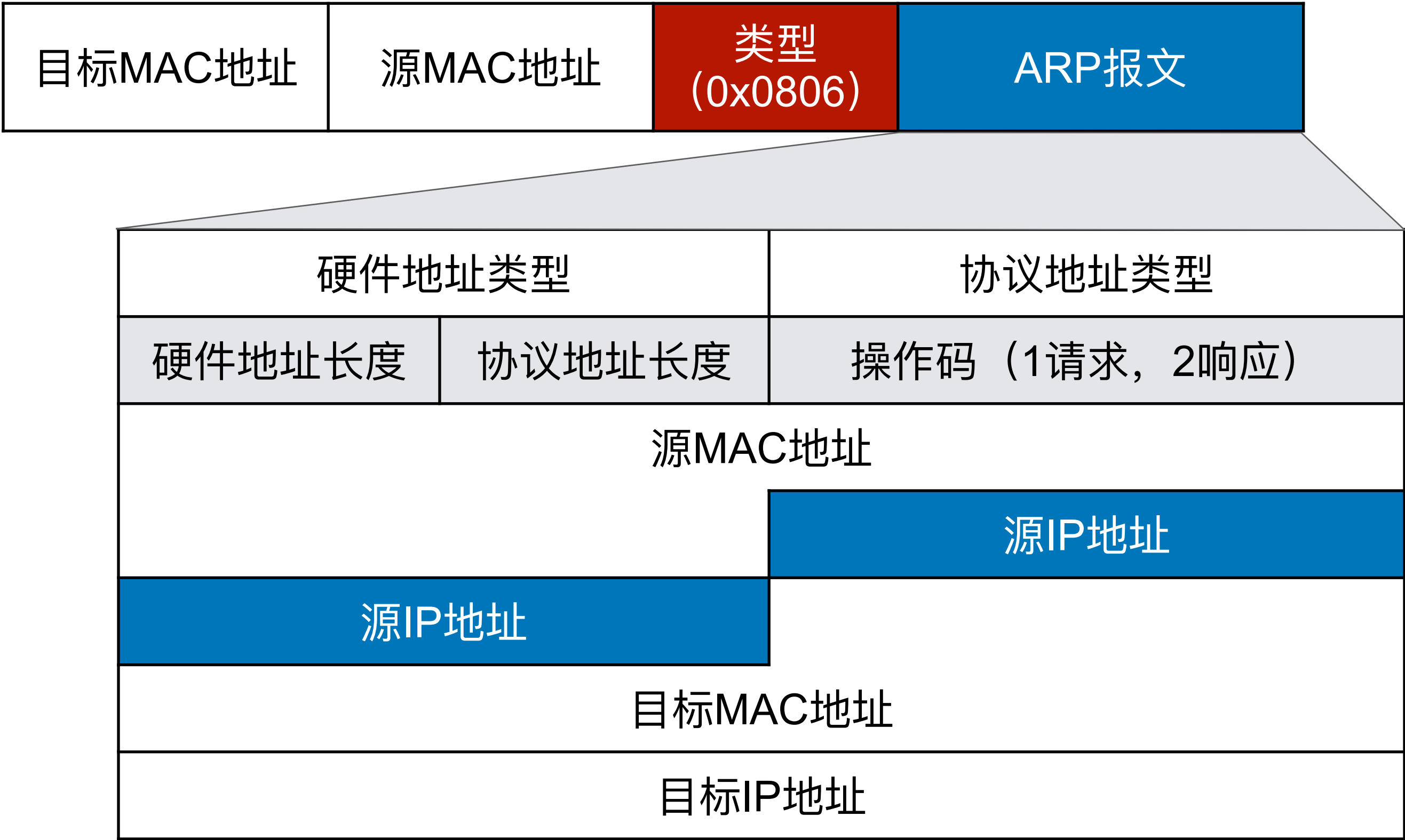
```
Mac-mini-2:~ li$ arp -a
? (192.168.1.1) at d4:41:65:ee:5c:c0 on en0 ifscope [ethernet]
? (192.168.1.4) at 80:d6:5:16:c5:7a on en0 ifscope [ethernet]
? (192.168.1.8) at 8c:fe:57:39:8b:1b on en0 ifscope [ethernet]
? (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
```

```
C:\Users\Administrator>arp -a
```

接口: 192.168.1.10 --- 0xa		
Internet 地址	物理地址	类型
192.168.1.1	d4-41-65-ee-5c-c0	动态
192.168.1.255	ff-ff-ff-ff-ff-ff	静态
224.0.0.22	01-00-5e-00-00-16	静态
224.0.0.252	01-00-5e-00-00-fc	静态
239.255.255.250	01-00-5e-7f-ff-fa	静态
255.255.255.255	ff-ff-ff-ff-ff-ff	静态

ARP报文格式（语法、语义）

- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP缓存
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况



ARP协议运行过程

- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP缓存
 - ARP报文格式
 - **ARP运行流程**
 - 注意问题
 - 使用ARP的四种情况

- 当主机 A 欲向本局域网上的某个主机 B 发送 IP 数据报时，先在其 ARP 高速缓存中查看有无主机 B 的 IP 地址：
 - **如有**，就可查出其对应的硬件地址，再将此硬件地址写入 MAC 帧，然后通过局域网将该 MAC 帧发往此硬件地址；
 - **如没有**，ARP 进程在本局域网上**广播发送一个 ARP 请求分组**。收到 ARP 响应分组后，将得到的 IP 地址到硬件地址的映射写入 ARP 高速缓存。

ARP协议请求实例

- Frame 136: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
- Ethernet II, Src: Elitegro_4f:47:d2 (74:27:ea:4f:47:d2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)

↑
MAC广播帧

Hardware type: Ethernet (1) ← 硬件地址类型: 以太网

Protocol type: IPv4 (0x0800) ← 协议地址类型: IP地址

Hardware size: 6 ← 硬件地址长度: 6字节

Protocol size: 4 ← 协议地址长度: 4字节

Opcode: request (1) ← 操作码: 1表示请求

Sender MAC address: Elitegro_4f:47:d2 (74:27:ea:4f:47:d2) ← 发送方硬件地址

Sender IP address: 172.20.28.40 ← 发送方IP地址

Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00) ← 目标硬件地址: 未知

Target IP address: 172.20.31.254 ← 目标IP地址

ARP协议响应实例

- Frame 137: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface
- Ethernet II, Src: Hangzhou_e1:c9:06 (00:0f:e2:e1:c9:06), Dst: Elitegro_4f:47:d2 (74:27:ea:4f:47:d2)
- Address Resolution Protocol (reply)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2) ← 操作码: 2表示响应

Sender MAC address: Hangzhou_e1:c9:06 (00:0f:e2:e1:c9:06)

Sender IP address: 172.20.31.254

Target MAC address: Elitegro_4f:47:d2 (74:27:ea:4f:47:d2)

Target IP address: 172.20.28.40

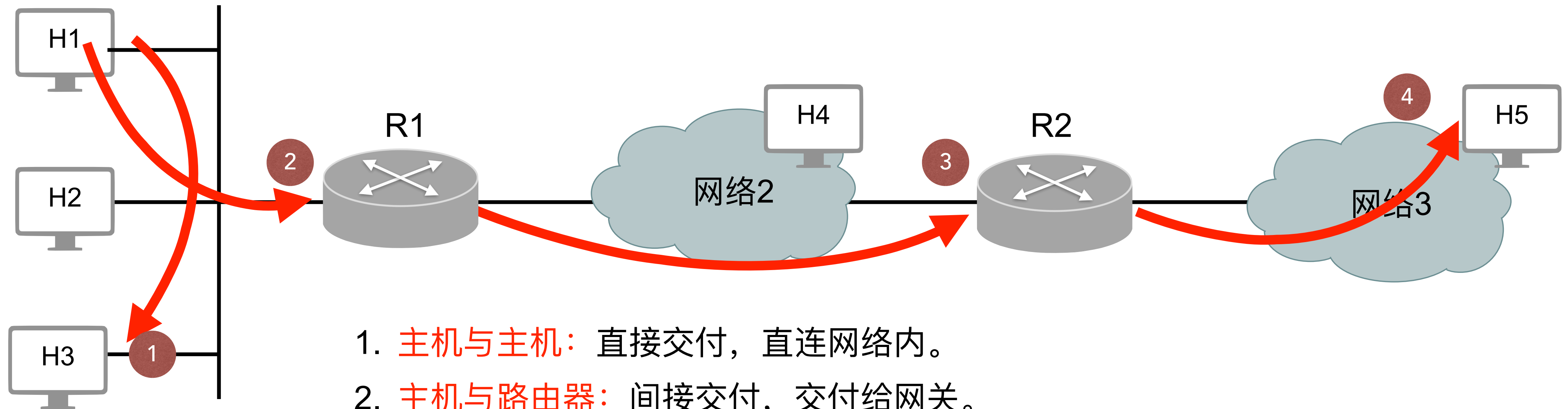
↑
MAC单播帧

注意问题

- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP缓存
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况

- **ARP协议不能穿透路由器：**
 - ARP 用于解决**同一个局域网**上的主机或路由器的 IP 地址和硬件地址的映射问题；
 - 如果所要找的主机和源主机**不在同一个局域网**上，那么就要通过 ARP 找到一个位于本局域网上的某个路由器的硬件地址，然后把分组发送给这个路由器，让这个路由器把分组转发给下一个网络；
 - 从 IP 地址到硬件地址的解析是**自动进行**的，主机的用户对这种地址解析过程是不知道的。

使用 ARP 的四种典型情况



1. **主机与主机**：直接交付，直连网络内。
2. **主机与路由器**：间接交付，交付给网关。
3. **路由器与路由器**：间接交付，路由器间转发。
4. **路由器与主机**：直接交付，路由器直接交付目标主机。

从 IP 地址到硬件地址的**解析是自动进行的**，主机的用户对这种地址解析过程是不知道的。

小结

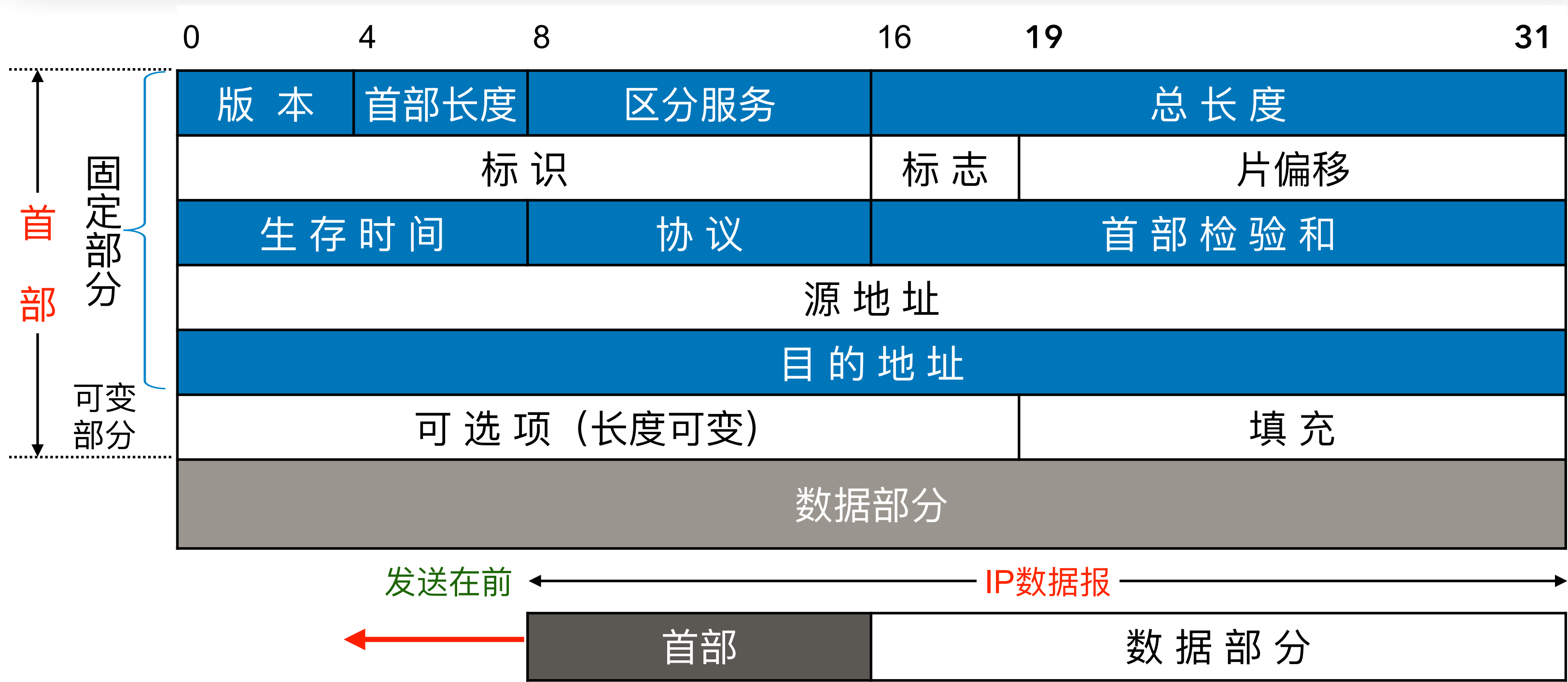
- 网络层
 - IP地址硬件地址
 - ARP协议
 - ARP报文格式
 - ARP工作流程
 - 注意问题
 - 使用ARP的四种情况

- 为什么不用硬件地址直接通信？
- IP地址与硬件地址的关系：
 - 查找路由使用IP地址，实现路由使用硬件地址。
- 如何根据IP地址获取硬件地址？
 - ARP协议（自动进行）。
- ARP高速缓存的作用：
 - 减少ARP广播。
- ARP协议：
 - 语法、语义、同步。
 - 使用ARP协议的四种情况：
 - 主机与主机、主机与路由器、路由器与路由器、路由器与主机。

IP 数据报的格式（语法、语义）

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

- IP 数据报由首部和数据两部分组成：
 - 首部的前一部分是固定长度，共 20 字节；
 - 在首部的固定部分的后面是一些可选字段，其长度是可变的。



IP 数据报的格式（语法、语义）

0	4	8	16	19	31
版 本		首部长度	区分服务	总 长 度	
标 识			标 志	片偏移	
生 存 时 间		协 议	首 部 检 验 和		
源 地 址					
目的地址					
可 选 项 （长度可变）				填 充	
数据部分					

- 区分服务：指明期望获得哪种类型的服务。
- 总长度：占 16 位，首部 + 数据，单位为字节，数据报的最大长度为 65535 字节。注意总长度与 MTU的关系。

- 版本：占 4 位，指 IP 协议的版本。目前的 IP 协议版本号为 4 (即 IPv4)。
- 首部长度：占 4 位，可表示的最大数值是 15 个单位(一个单位为 4 字节)。IP 的首部长度的最大值是 60 字节。一般IP首部仅有固定部分20字节。
- 故通常该字段对应的值为：
0101。

IP 数据报的格式（语法、语义）

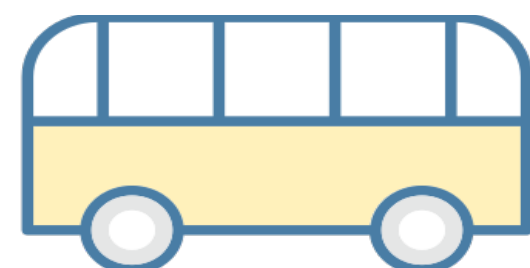
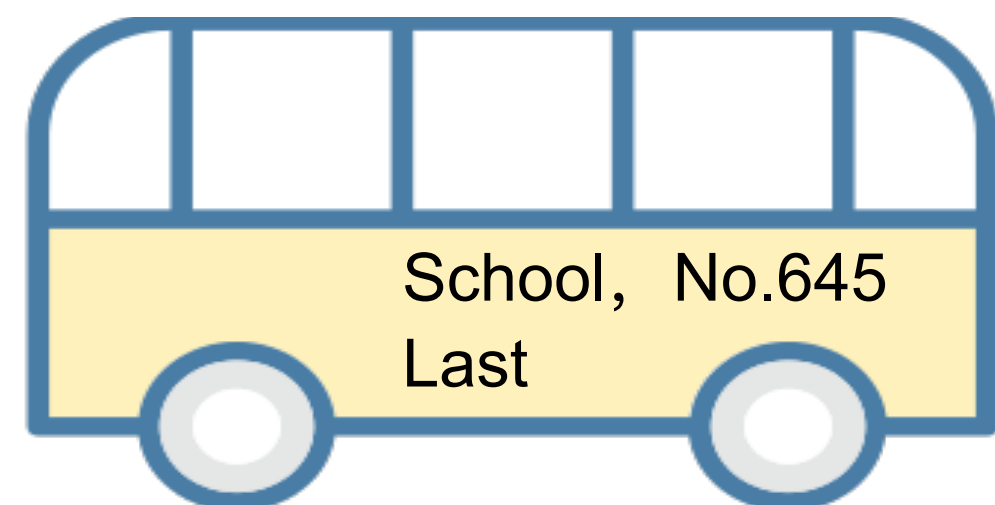
0	4	8	16	19	31
版 本		首部长度	区分服务	总 长 度	
标 识			标 志	片偏移	
生 存 时 间		协 议	首 部 检 验 和		
源 地 址					
目 的 地 址					
可 选 项 （长度可变）				填 充	
数据部分					

这三个字段与IP分片有关，IP数据报总长度超过MTU时，IP需要分片。

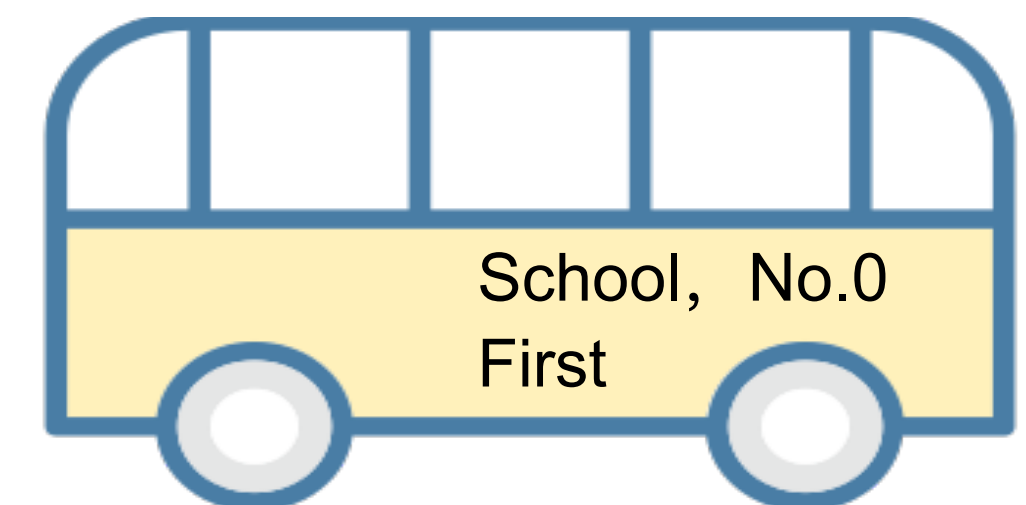
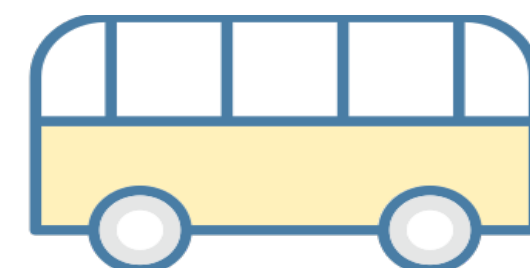
- 标识：是一个计数器，用来产生IP数据报标识。所有IP分片的标识与原始IP标识一致，便于接收方还原原始IP数据报。
- 标志：占3位，最高位无意义：
 - 中间位DF（Don't Fragment）：
 - DF=1，不允许分片；
 - DF=0，允许分片。
 - 最低位MF（More Fragment）：
 - MF=1，后面还有分片；
 - MF=0，这是最后一片。
- 片偏移：占13位，某片在原始IP中的相对位置，以8字节为单位。

IP分片实例

- 某学校655名学生全部到另一学校参观：
 - 假设每辆大巴车可乘坐15人，学校需要一次性安排44辆大巴车来运送学生，其中前43辆每辆安排15人，第44辆安排10人；
 - 每辆大巴车上都贴上学校的标签（标识）、标志（后面是否还有大巴车）和大巴车里第1个同学的编号（片偏移）。



.....

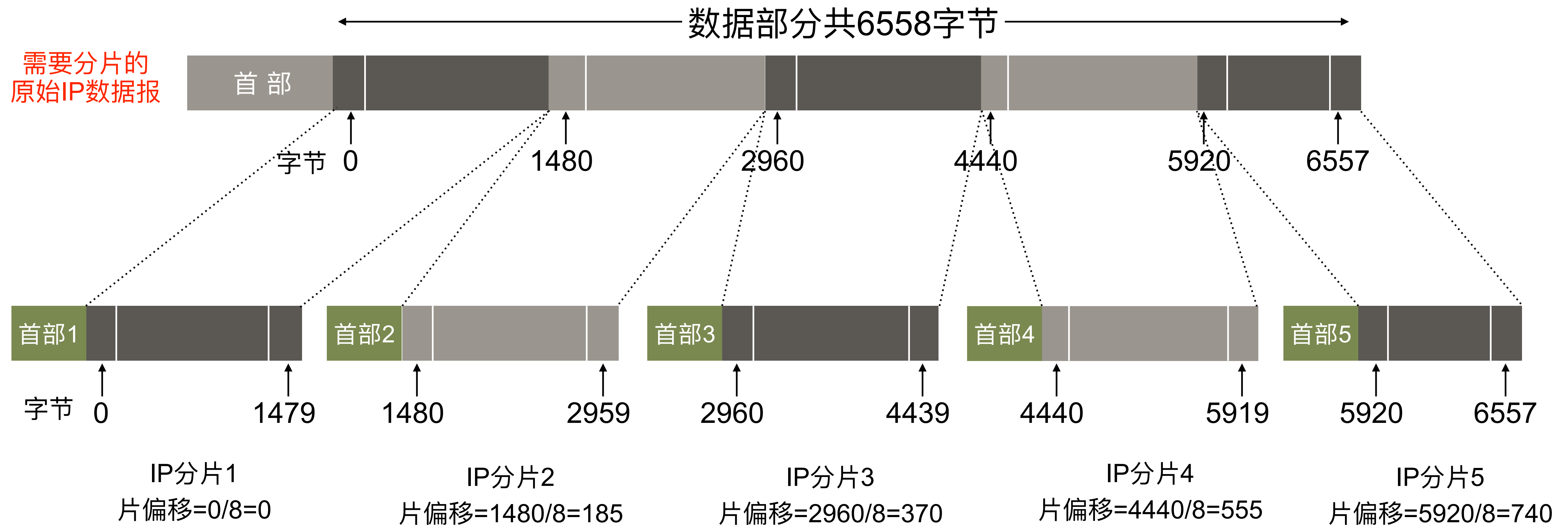


IP分片实例

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

- 原始IP数据报总长度为6578字节，则原始IP数据报中携带的数据为6558字节（固定首部20字节）。
- 假设该IP分组需通过以太网帧传输（MTU=1500字节），则需要分为5个IP分片：
 - 前4个IP分片每片携带1480字节（加上IP分片首部20字节，刚好1500字节）；
 - 最后1个IP分片携带638字节。

IP数据报分片示意图



IP数据报分片总结

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

IP	总长度（20字节首部）	标识	MF标志	片偏移
原始IP	6558+20	942	0	0
分片1	1480+20	942	1	0
分片2	1480+20	942	1	185
分片3	1480+20	942	1	370
分片4	1480+20	942	1	555
分片5	638+20	942	0	740

Ethernet II, Src: 00:0c:29:41:3b:83, Dst: 00:50:56:c0:00:08 #IP封装到以太网

Internet Protocol Version 4, Src: 172.16.25.130, Dst: 172.16.25.1 #始终未变

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

#首部长度

Differentiated Services Field: 0x00

Total Length: 1500

#总长度, 首部+数据

Identification: 0x03ae (942)

#原始IP的标识: 942, 5个分片相同

Flags: 0x2000, More fragments

0... = Reserved bit: Not set

.0.. = Don't fragment: Not set

..1. = More fragments: Set

#后面还有分片标志

...0 0000 0000 0000 = Fragment offset: 0

#片偏移

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0x86cf

Source: 172.16.25.130

Destination: 172.16.25.1

Reassembled IPv4 in frame: 8

Data (1480 bytes)

Data: 08005f020200a3006162636465666768696a6b6c6d6e6f70...

第1个分片

Ethernet II, Src: 00:0c:29:41:3b:83, Dst: 00:50:56:c0:00:08

Internet Protocol Version 4, Src: 172.16.25.130, Dst: 172.16.25.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00

Total Length: 1500

Identification: 0x03ae (942)

#原始IP标识: 942

Flags: 0x20b9, More fragments

0... = Reserved bit: Not set

.0.. = Don't fragment: Not set

..1. = More fragments: Set

#后面还有分片

...0 0000 1011 1001 = Fragment offset: 185

#前一片已传了0~1479字节数据,

片偏移为 $1480/8 = 185$

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0x8616

Source: 172.16.25.130

Destination: 172.16.25.1

Reassembled IPv4 in frame: 8

Data (1480 bytes)

#1480字节数据

Data: 6162636465666768696a6b6c6d6e6f707172737475767761...

第2个分片

[Length: 1480]

Ethernet II, Src: 00:0c:29:41:3b:83, Dst: 00:50:56:c0:00:08
Internet Protocol Version 4, Src: 172.16.25.130, Dst: 172.16.25.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00

Total Length: 1500

Identification: 0x03ae (942)

#原始IP标识

Flags: 0x2172, More fragments

0... = Reserved bit: Not set

.0.. = Don't fragment: Not set

..1. = More fragments: Set

#后面还有分片

...0 0001 0111 0010 = Fragment offset: 370

#前面已传输0~2959字节的数据,

片偏移=2960/8=370

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0x855d

Source: 172.16.25.130

Destination: 172.16.25.1

Reassembled IPv4 in frame: 8

Data (1480 bytes)

Data: 696a6b6c6d6e6f7071727374757677616263646566676869...

第3个分片

[Length: 1480]

#1480字节数据

Ethernet II, Src: 00:0c:29:41:3b:83, Dst: 00:50:56:c0:00:08
Internet Protocol Version 4, Src: 172.16.25.130, Dst: 172.16.25.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00

Total Length: 1500

Identification: 0x03ae (942)

#原始IP标识

Flags: 0x222b, More fragments

0... = Reserved bit: Not set

.0.. = Don't fragment: Not set

..1. = More fragments: Set

#后面还有分片

...0 0010 0010 1011 = Fragment offset: 555

#前面已传输0~4439字节的数据,
片偏移为4440/=555

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0x84a4

Source: 172.16.25.130

Destination: 172.16.25.1

Reassembled IPv4 in frame: 8

Data (1480 bytes)

#1480字节数据

Data: 717273747576776162636465666768696a6b6c6d6e6f7071...

[Length: 1480]

第4个分片

第5个分片

Ethernet II, Src: 00:0c:29:41:3b:83, Dst: 00:50:56:c0:00:08

Internet Protocol Version 4, Src: 172.16.25.130, Dst: 172.16.25.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00

Total Length: 658

Identification: 0x03ae (942)

Flags: 0x02e4

0... = Reserved bit: Not set

.0.. = Don't fragment: Not set

..0. = More fragments: Not set

...0 0010 1110 0100 = Fragment offset: 740

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0xa735

Source: 172.16.25.130

Destination: 172.16.25.1

#总长度: 20+8+630, IP分片中的数据为638字节

#原始IP标识

#这是最后一个IP分片

#前面已传输0~5919字节的数据,
片偏移为5920/8=740

[5 IPv4 Fragments (6558 bytes): #4(1480), #5(1480), #6(1480), #7(1480), #8(638)] #5个分片情况

IP 数据报格式（语义）

0	4	8	16	19	31
版 本	首部长度	区分服务	总 长 度		
标 识			标 志	片偏移	
生 存 时 间	协 议		首 部 检 验 和		
源 地 址					
目 的 地 址					
可 选 项 （长度可变）				填 充	
数据部分					

- 00：表示IP协议。

01：ICMP协议。

06：表示TCP协议。

17：表示UDP协议。
- macOS：
/etc/procotols

• Windows 7：
C:\Windows\System32\drivers\etc\procotol

• **生存时间**：占8 位，记为 TTL (Time To Live)，指示数据报在网络中可通过的路由器数的最大值。路由器收到IP数据报后，将TTL减1，减1之后TTL值若**变为0**，路由器**丢弃该IP数据报**。并向源端报超时错误。

• **协议**：占8 位，指出此数据报携带的数据**使用何种协议**，以便目的主机的 IP 层将数据部分，上交给那个处理过程。

TTL超时实例

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

```
li@ubuntu1604:~$ ping -c 2 www.baidu.com
```

```
PING www.wshifen.com (103.235.46.39) 56(84) bytes of data.
```

```
64 bytes from 103.235.46.39: icmp_seq=1 ttl=39 time=361 ms
```

```
64 bytes from 103.235.46.39: icmp_seq=2 ttl=39 time=361 ms
```

```
--- www.wshifen.com ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
```

```
rtt min/avg/max/mdev = 361.532/361.618/361.704/0.086 ms
```

向主机www.baidu.com发送2个ICMP请求报文，收到2个响应报文。

TTL超时实例

- 网络层
 - IP数据报格式
 - IP分片
 - **TTL**
 - 检验和

```
li@ubuntu1604:~$ ping -c 4 -t 4 www.baidu.com
PING www.wshifen.com (103.235.46.39) 56(84) bytes of data.
From 202.193.156.213 icmp_seq=1 Time to live exceeded
From 202.193.156.213 icmp_seq=2 Time to live exceeded
From 202.193.156.213 icmp_seq=3 Time to live exceeded
--- www.wshifen.com ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3004ms
```

将封装的IP数据报中的TTL置为4（从源到目标不止经过4个路由器）
IP地址为202.193.156.213的路由器报告“**Time to live exceeded**”错误。

```
C:\Users\Administrator>ping -i 3 www.baidu.com
正在 Ping www.a.shifen.com [14.215.177.39] 具有 32 字节的数据:
请求超时。
macOS中IP默认的TTL为64
```

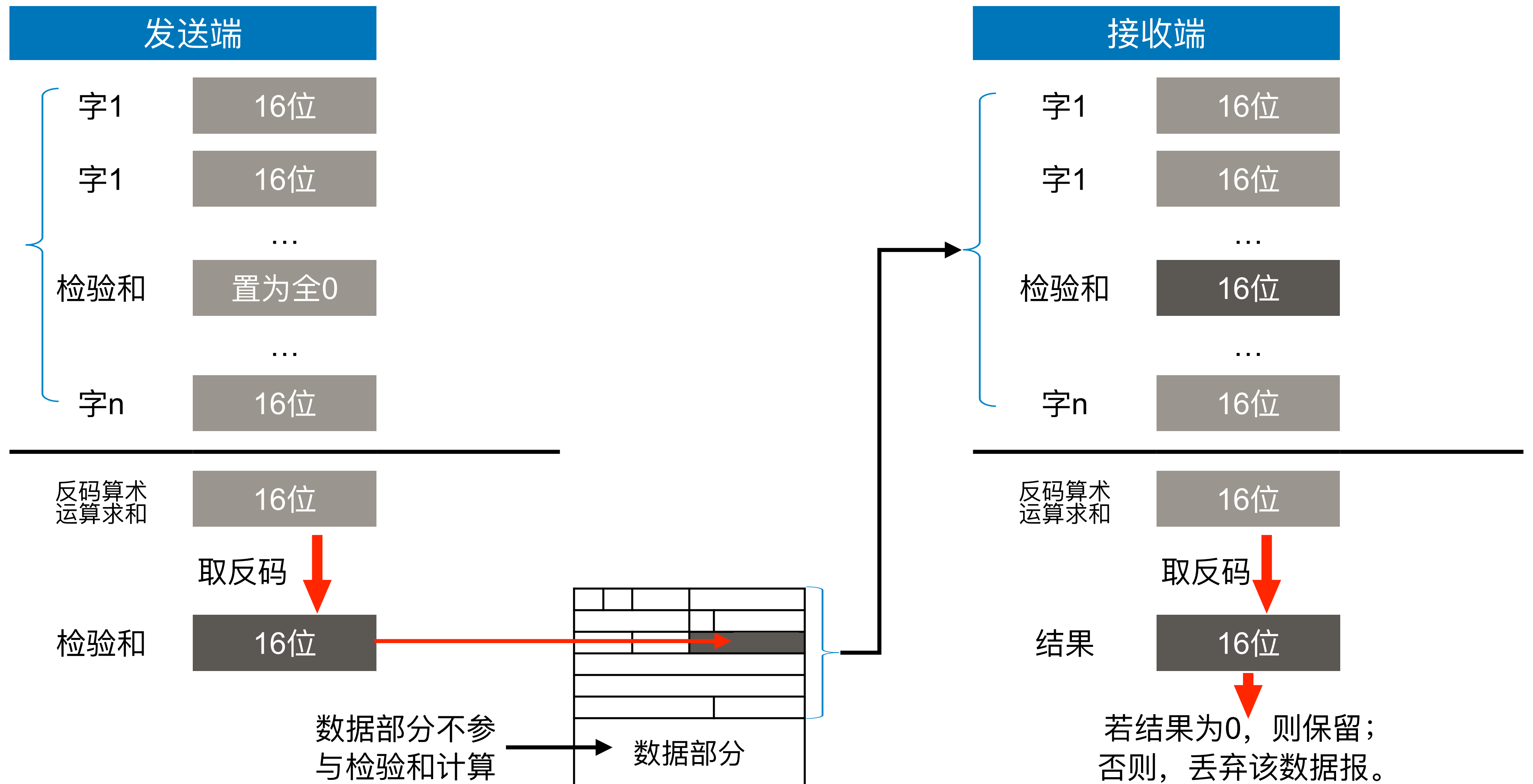
IP 数据报格式 (语义)

0	4	8	16	19	31
版 本	首部长度	区分服务	总 长 度		
标 识			标 志	片偏移	
生 存 时 间	协 议		首 部 检 验 和		
源 地 址					
目 的 地 址					
可 选 项 （长度可变）				填 充	
数据部分					

- 首部检验和：占16 位，只检验数据报的首部，不检验数据部分。这里不采用 CRC 检验码而采用简单的计算方法。
- 地址：源地址和目的地址都各占 4 字节。

为什么不对数据部分进行检验？

IP 数据报首部检验和的计算采用 16 位二进制反码求和算法



二进制反码求和实例

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

- 原始数据为：1100, 1010, 0000（检验和）；
- 将原始数相加： $1100+1010+0000=10110$ ；
- 高位有进位加到低位： $0110+1=0111$ ；
- 检验和为：1000。

- 发送的数据为：1100, 1010, 1000（检验和）；
- 接收端验证： $1100+1010+1000=1111$ ；
- 取反为：0000（没有错误）。

IP 数据报首部的可变部分

0	4	8	16	19	31		
版 本		首部长度		区分服务		总 长 度	
标 识				标 志		片偏移	
生 存 时 间		协 议		首 部 检 验 和			
源 地 址							
目的地址							
可 选 项 （长度可变）					填 充		
数据部分							

填充： 使IP数据报为4字节的整数倍。

- **选项：** 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。实际上这些选项很少被使用：
- IP 首部的可变部分就是一个选项字段，用来支持排错、测量以及安全等措施，**内容很丰富**；
- 可变部分增加了 IP 数据报的功能。这就增加了每一个路由器处理数据报的**开销**。

小结

- 网络层
 - IP数据报格式
 - IP分片
 - TTL
 - 检验和

- IP数据报格式（语法、语义）：
 - 20字节固定部分，40字节可选部分；
 - 重要字段：首部长、总长度、标识、标志、片偏移、生存时间TTL；
 - 检验和计算方法。

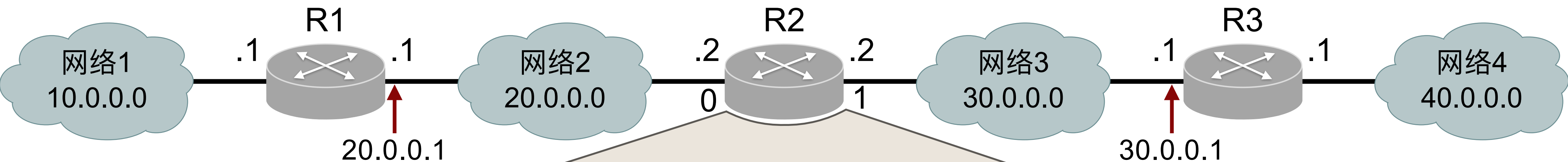
IP 层转发分组的流程

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

- 假设：有四个 A 类网络通过三个路由器连接在一起。每一个网络上都可能有成千上万个主机：
 - 按目的主机号来制作路由表，每一个路由表就有 4 万个项目，即 4 万行（每一行对应于一台主机），则所得出的路由表就会过于庞大；
 - 若按主机所在的网络地址来制作路由表，那么每一个路由器中的路由表就只包含 4 个项目（每一行对应于一个网络），这样就可使路由表大大简化。

物流公司不可能为每一个人保留一份路由表。

IP 层转发分组的流程



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	20.0.0.1
40.0.0.0	30.0.0.1

在路由表中，对每一条路由，最主要的是：
(目的网络地址，下一跳地址)

查找路由表

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

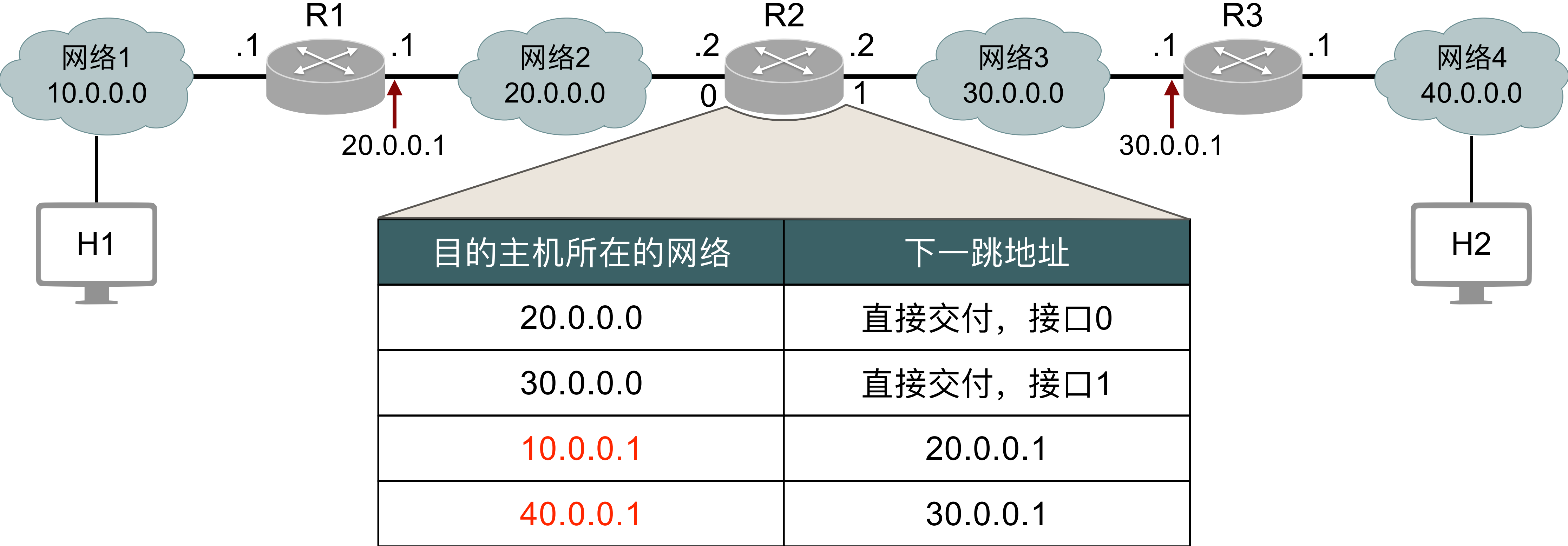
- 根据目的网络地址就能确定下一跳路由器，这样做的结果是：
 - IP 数据报最终一定可以找到目的主机所在目的网络上的路由器（可能要通过多次的间接交付）；
 - 只有到达最后一个路由器时，才试图向目的主机进行直接交付。

特定主机路由

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

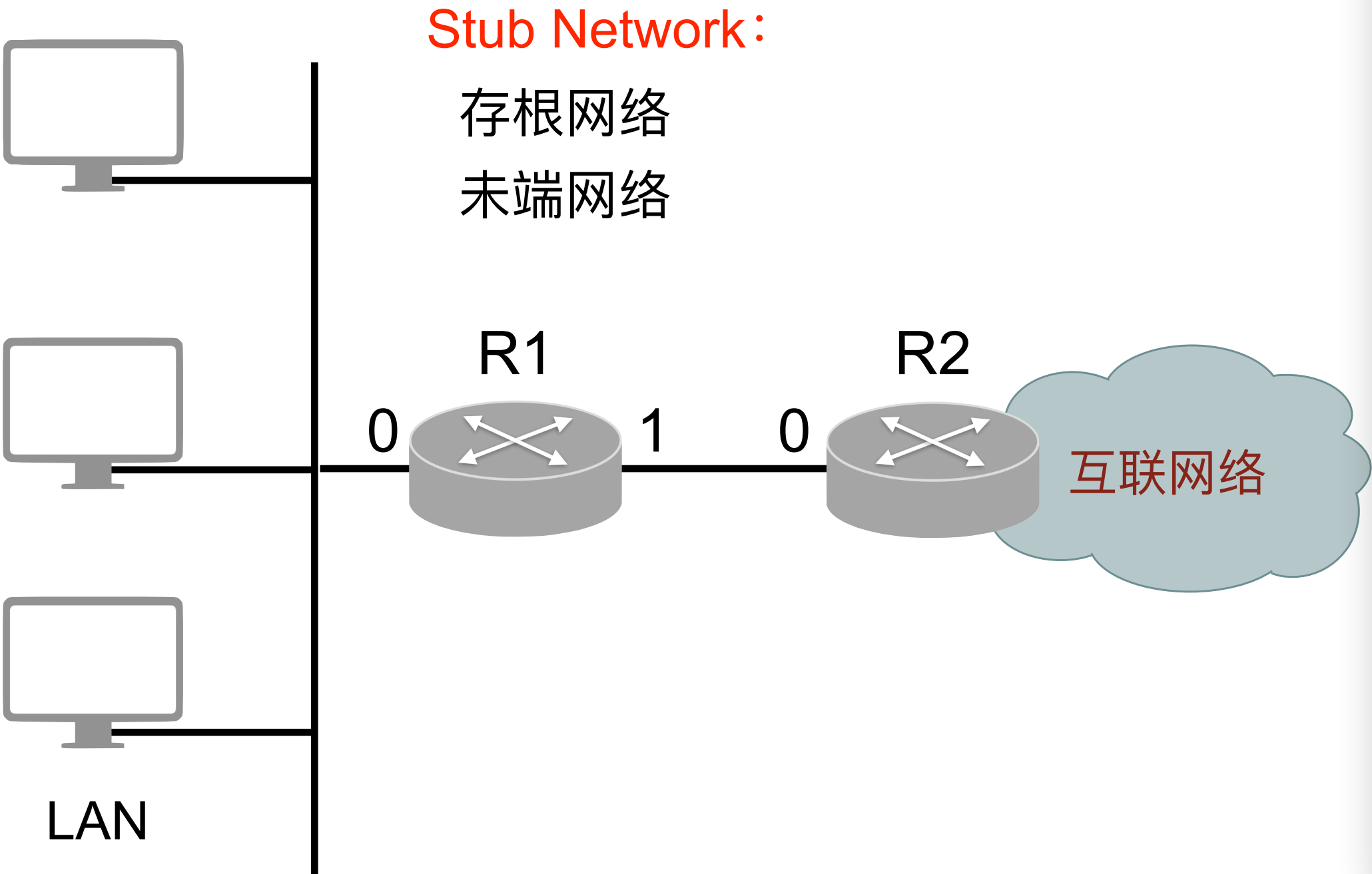
- 互联网所有的分组转发都是基于目的主机所在的网络，但允许有这样的特例：
 - 即为特定的目的主机指明一个路由；
 - 采用特定主机路由可使网络管理人员能更方便地控制网络和测试网络，同时也可在需要考虑某种安全问题时采用这种特定主机路由。

特定主机路由



(目的IP地址, 下一跳地址)

默认路由 (default route)

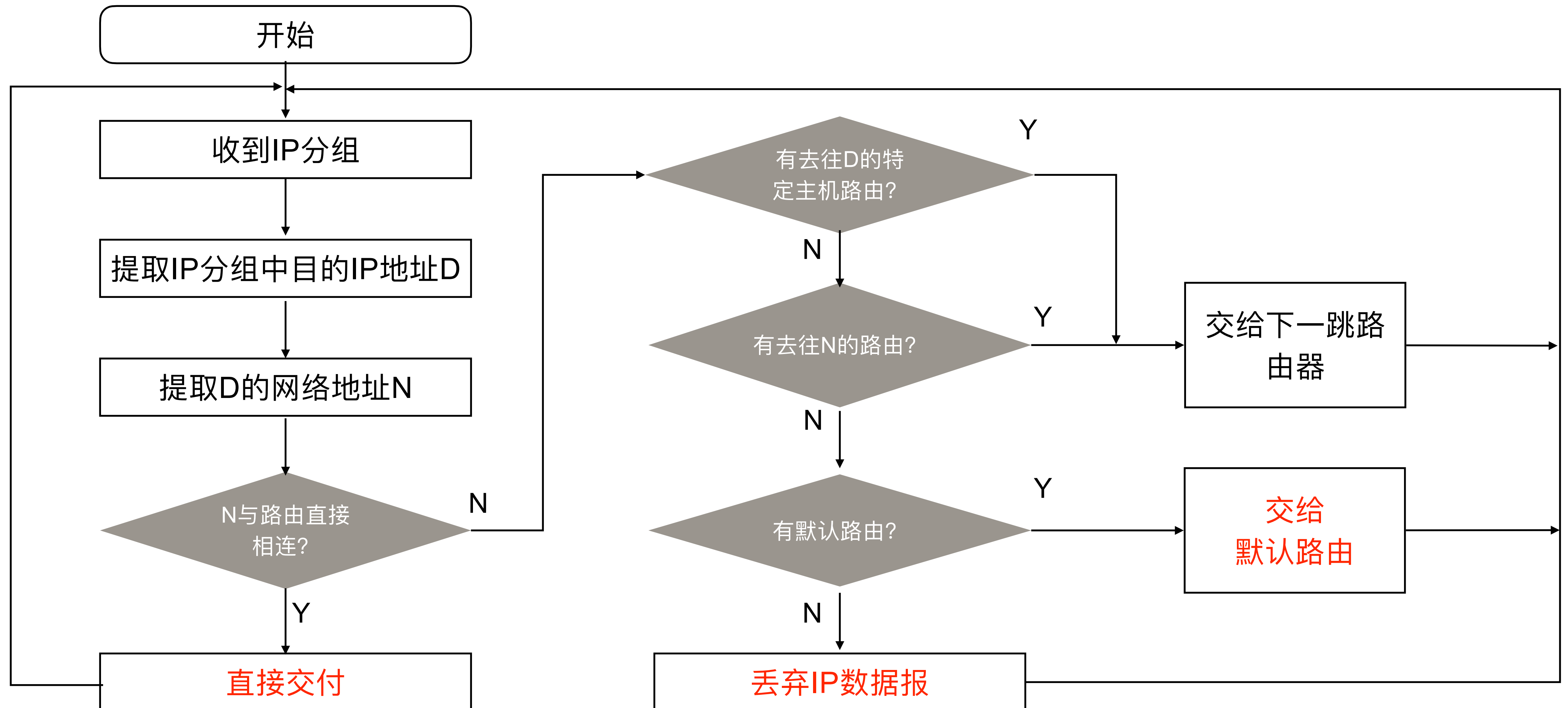


- 路由器采用默认路由以减少路由表所占用的空间和搜索路由表所用的时间：
- 网络只用一个路由器和互联网连接，这种情况下使用默认路由；
- LAN中的主机，访问互联网的默认路由为R1的接口0。这些主机使用ARP协议获取R1接口0的MAC地址。

```
li@ubuntu1604:~$ route
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	202.XXX.XXX.XXX	0.0.0.0	UG	0	0	0	ens32

路由器分组转发算法



注意

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

- IP 数据报的首部中没有地方可以用来指明“下一跳路由器的 IP 地址”：
 - 当路由器收到待转发的数据报，不是将下一跳路由器的 IP 地址填入 IP 数据报，而是送交下层的网络接口软件；
 - 网络接口软件使用 ARP 根据下一跳路由器的 IP 地址获取硬件地址，并将此硬件地址作为 MAC 帧的目的 MAC 地址，然后将该 MAC 帧交付给下一跳路由器。

不管是直接交付还是间接交付，路由器都会调用 ARP 协议获取对端的 MAC 地址。

关于路由表

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

- 路由表没有给分组指明到某个网络的完整路径。**路由表指出：**
 - 到某个网络应当先到某个路由器（**即下一跳路由器**）；
 - 到达下一跳路由器后，**继续查找其路由表**，再下一步应当到哪一个路由器；
 - 这样一步一步地查找下去，直到**最后到达目的网络**。

小结

- 网络层
 - 分组转发流程
 - 查找路由表
 - 特定主机路由
 - 默认路由
 - 分组转发算法

IP层转发分组的流程、分组转发算法（同步）：

- 间接交付、直接交付；
- 特定主机路由；
- 默认路由。

分组转发算法。

IP地址编址方法

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

划分子网

使用子网时分组的转发

IP地址编址方法

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

- 1985 年起，IP 地址中增加了“子网号字段”，使两级的 IP 地址变成三级的 IP 地址。这种做法叫作划分子网 (subnetting)：
 - 划分子网已成为互联网的正式标准协议；
 - 当没有划分子网时，IP 地址是两级结构；
 - 划分子网后 IP 地址就变成了三级结构。

划分子网是把 IP 地址的主机号 host-id 进行再划分，而不改变 IP 地址的网络号 net-id。

为什么要划分子网？

- 网络层
- IP地址编址方法
- 划分子网
- 子网掩码
- 分组转发流程

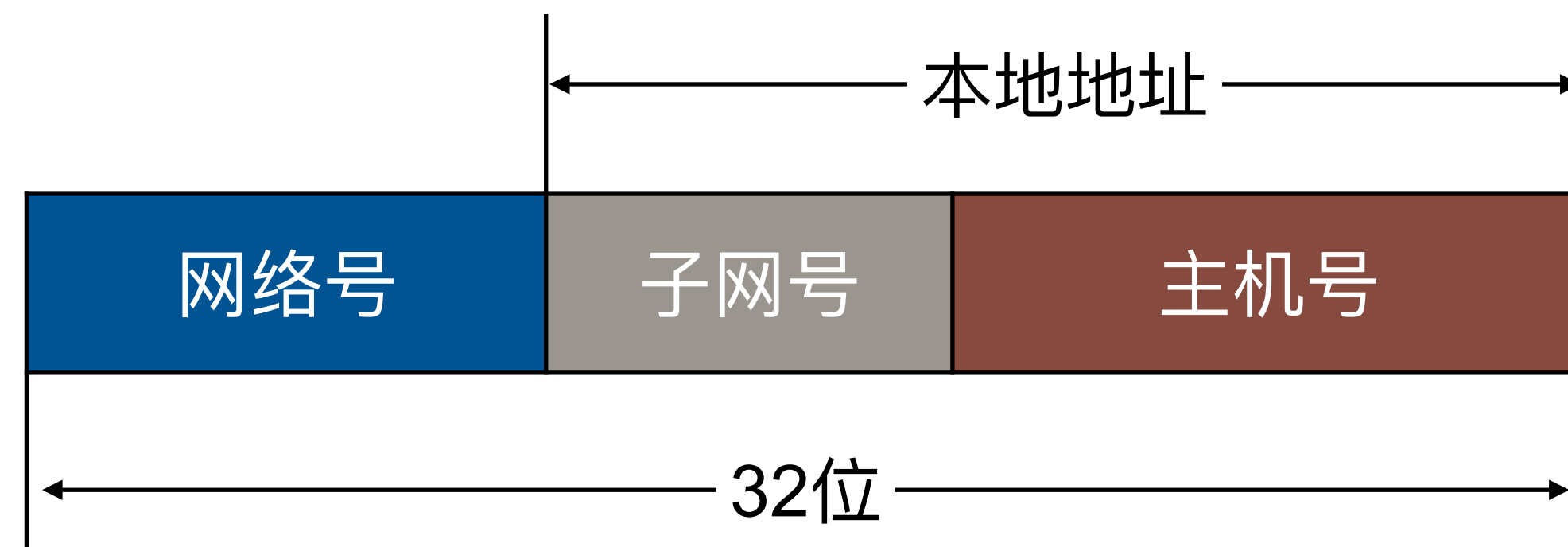
- 某单位有 4 个部门，每个部门 30 人左右：
 - 若每个部门分配一个A、B或C类网络地址，大量IP地址被浪费；
 - 大量计算机在同一个IP网络中，存在安全隐患；
 - 增加一个部门，没有得到IP网络之前不能连入互联网。

- 网络层划分子网：
 - 把大的广播域划分为较小的广播域；
 - 提高IP 地址空间的利用率；
 - 保证网络的安全性；
 - 提高网络的灵活性。

划分子网的基本思路

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

- 划分子网纯属一个单位内部的事情。单位对外仍然表现为没有划分子网的网络。具体方法：
 - 从主机号借用若干个位作为子网号 subnet-id, 主机号 host-id 相应减少了若干个位。



IP地址 :: = {<网络号>, <子网号>, <主机号>}

划分子网的生活实例

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

序号	网络号	子网号	主机号	第1个可用的号码	最后1个可用的号码
0	139	0000	0000	13900000000	13900009999
1	139	0001	0000	13900010000	13900019999
2	139	0002	0000	13900020000	13900029999
...
9999	139	9999	0000	13999990000	13999999999

身份证号码、邮政编码、学生学号等，凡是编号问题都会采取这种办法。

子网划分方法

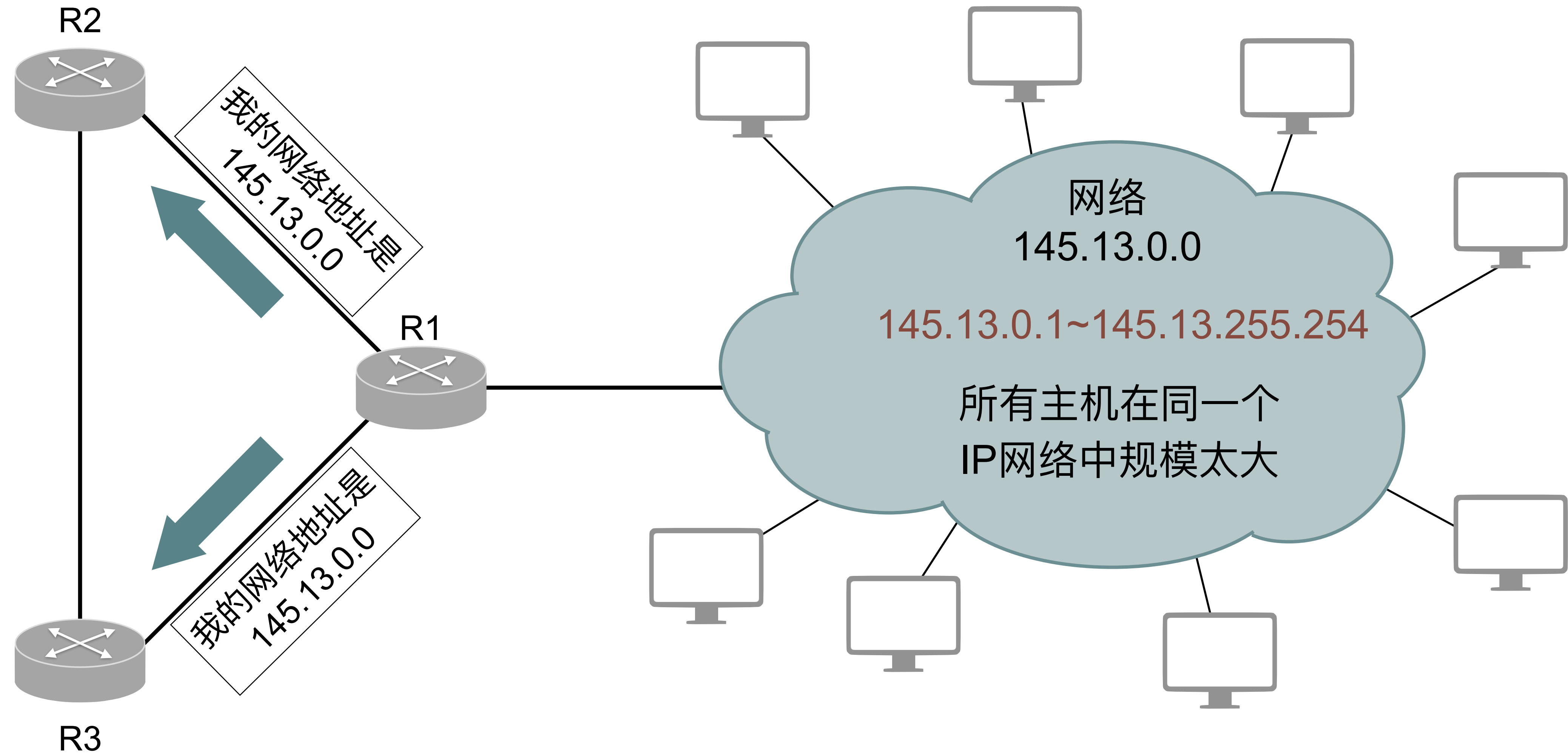
- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

- 有固定长度子网和变长子网两种子网划分方法：
 - 在采用固定长度子网时，所划分的所有子网的子网掩码都是相同的；
 - 划分子网增加了灵活性，但却减少了能够连接在网络上的主机总数。

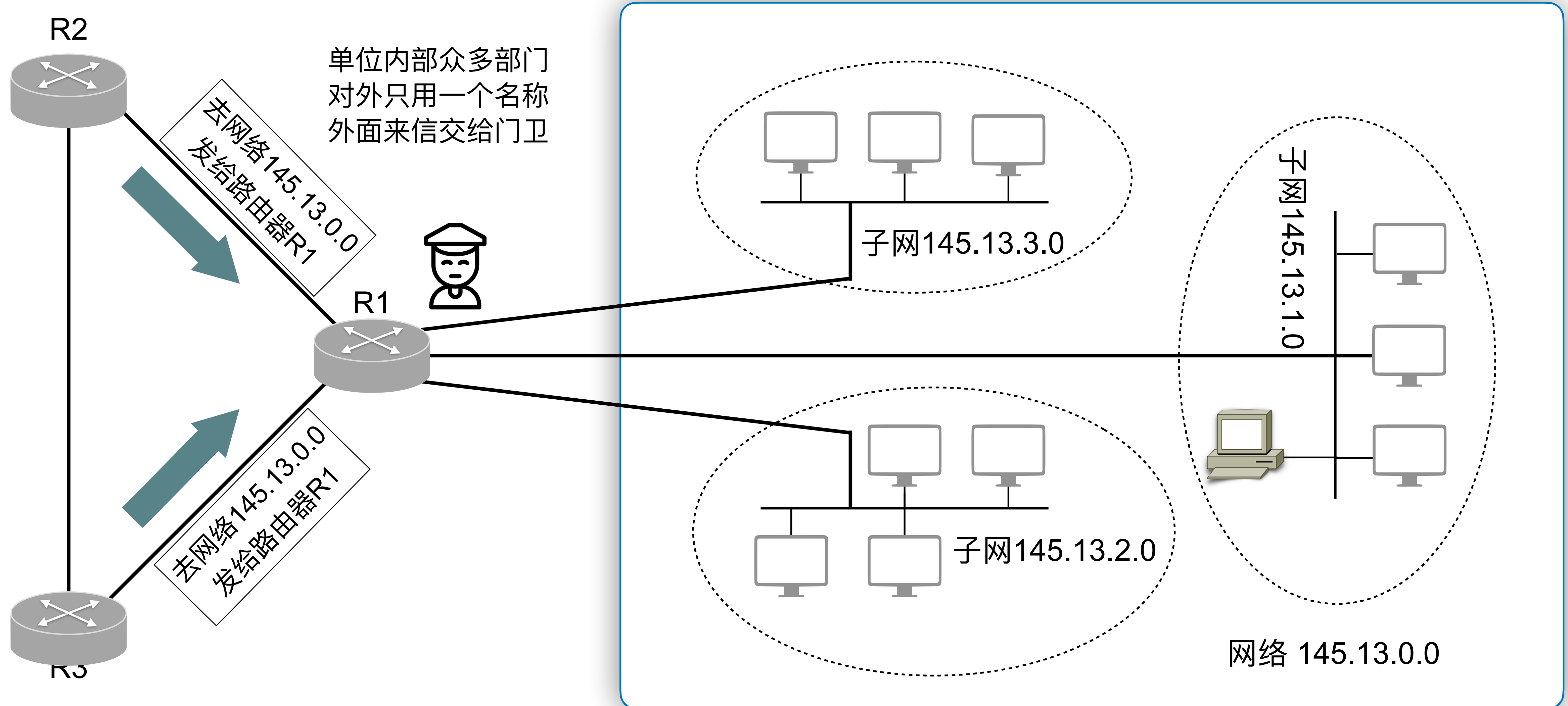
划分子网的一个实例

192.168.1.0							
每次增加二进制1		每次增加32					
序号	网络号	子网号 (十进制)	第1个可用的IP地址	最后1个可用的IP地址			
0	192.168.1	000	00000(0)	192.168.1.00000001	192.168.1.1	192.168.0.00011110	192.168.1.30
1	192.168.1	001	00000(32)	192.168.1.00100001	192.168.1.33	192.168.0.00111110	192.168.1.62
2	192.168.1	010	00000(64)	192.168.1.01000001	192.168.1.65	192.168.0.01011110	192.168.1.94
3	192.168.1	011	00000(96)	192.168.1.01100001	192.168.1.97	192.168.0.01111110	192.168.1.126
4	192.168.1	100	00000(128)	192.168.1.10000001	192.168.1.129	192.168.0.10011110	192.168.1.158
5	192.168.1	101	00000(160)	192.168.1.10100001	192.168.1.161	192.168.0.10111110	192.168.1.190
6	192.168.1	110	00000(192)	192.168.1.11000001	192.168.1.193	192.168.0.11011110	192.168.1.222
7	192.168.1	111	00000(224)	192.168.1.11100001	192.168.1.225	192.168.0.11111110	192.168.1.254
主机号全0表示网络				主机号全1表示广播地址			

一个未划分子网的 B 类网络145.13.0.0



划分子网后对外仍是一个网络



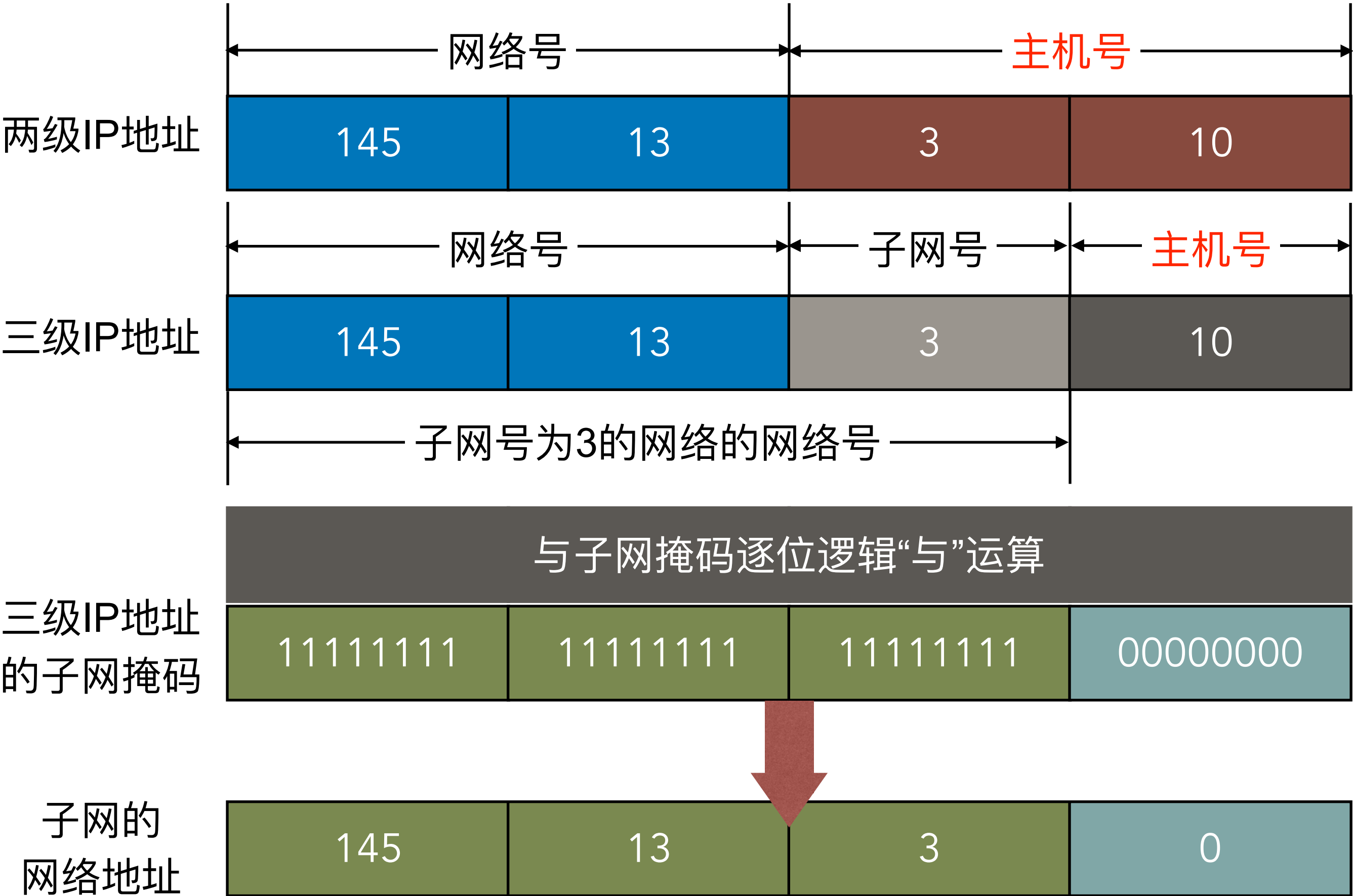
子网掩码

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

- 路由器R1必须知道子网划分情况，路由器使用子网掩码计算IP地址的网络号（划分子网之后）：
 - 子网掩码长度 = 32位；
 - 子网掩码左边部分的一连串 1：对应位为网络号和子网号；
 - 子网掩码右边部分的一连串 0：对应位为主机号。

门卫（路由器R1）如何把信件分发给单位内部各部门？
门卫必须知道来信的子网分配情况（有哪些部门）。

(IP 地址) AND (子网掩码) =网络地址



逻辑“与”的真值表

P	Q	P AND Q
0	0	0
0	1	0
1	0	0
1	1	1

1保持原样不变，0把一切变0

主机位全“0”，代表主机所在的网络号

默认子网掩码

A类地址

网络号

主机号全为0

子网掩码

255.0.0.0

11111111

00000000

00000000

00000000

B类地址

网络号

主机号全为0

子网掩码

255.255.0.0

11111111

11111111

00000000

00000000

C类地址

网络号

主机号全为0

子网掩码

255.255.255.0

11111111

11111111

11111111

00000000

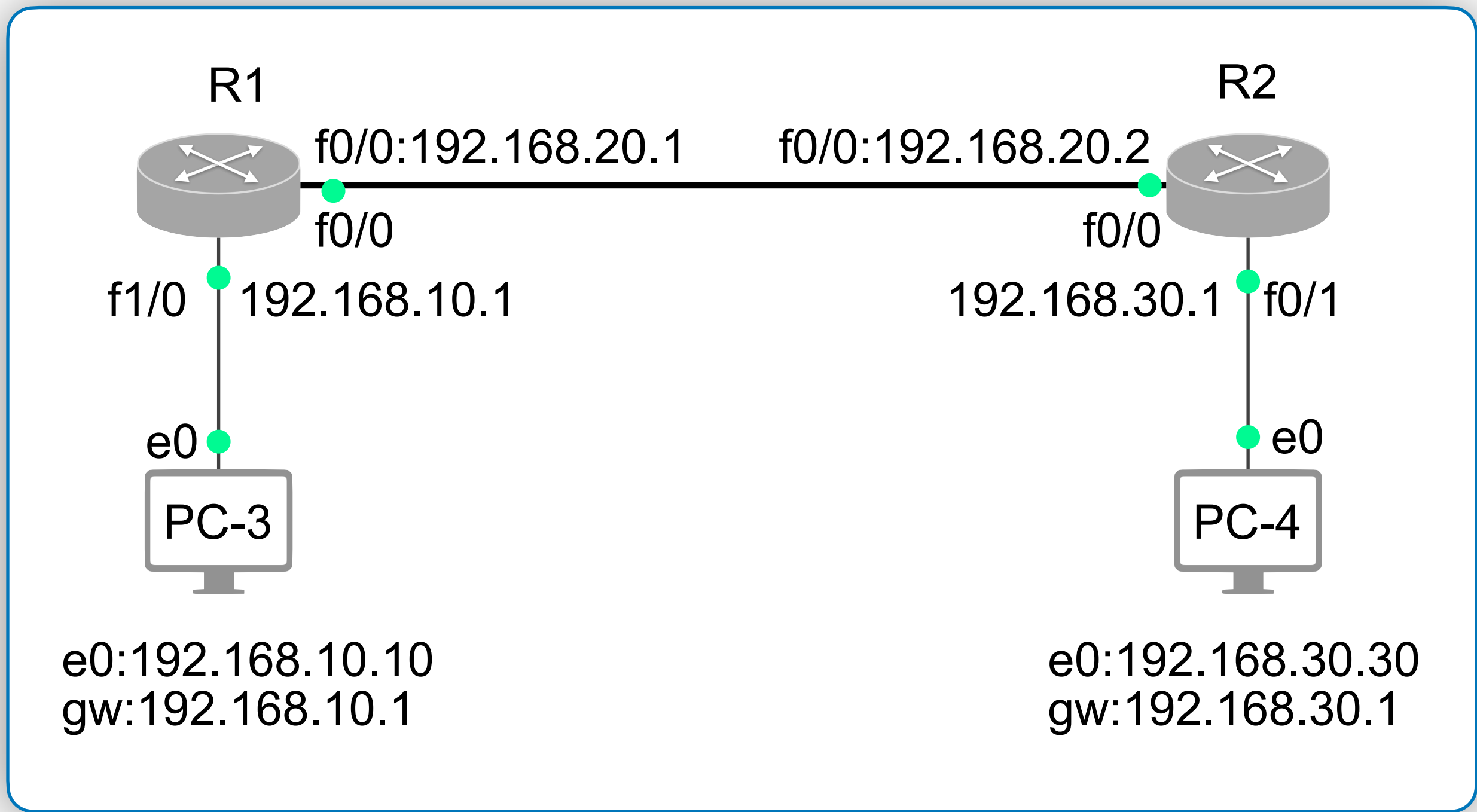
子网掩码是一个重要属性

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程

- 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。

目的网络	子网掩码	下一跳路由器
------	------	--------

路由器路由表实例



PC-3> ping 192.168.30.30 #验证网络连通性

84 bytes from 192.168.30.30 icmp_seq=1 ttl=62 time=59.038 ms
84 bytes from 192.168.30.30 icmp_seq=2 ttl=62 time=25.999 ms
84 bytes from 192.168.30.30 icmp_seq=3 ttl=62 time=26.725 ms
84 bytes from 192.168.30.30 icmp_seq=4 ttl=62 time=34.983 ms
84 bytes from 192.168.30.30 icmp_seq=5 ttl=62 time=29.415 ms

R2#show ip route

C 192.168.30.0/24 is directly connected, FastEthernet0/1
C 192.168.20.0/24 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 192.168.20.1

默认路由

直连网络

R1#show ip route

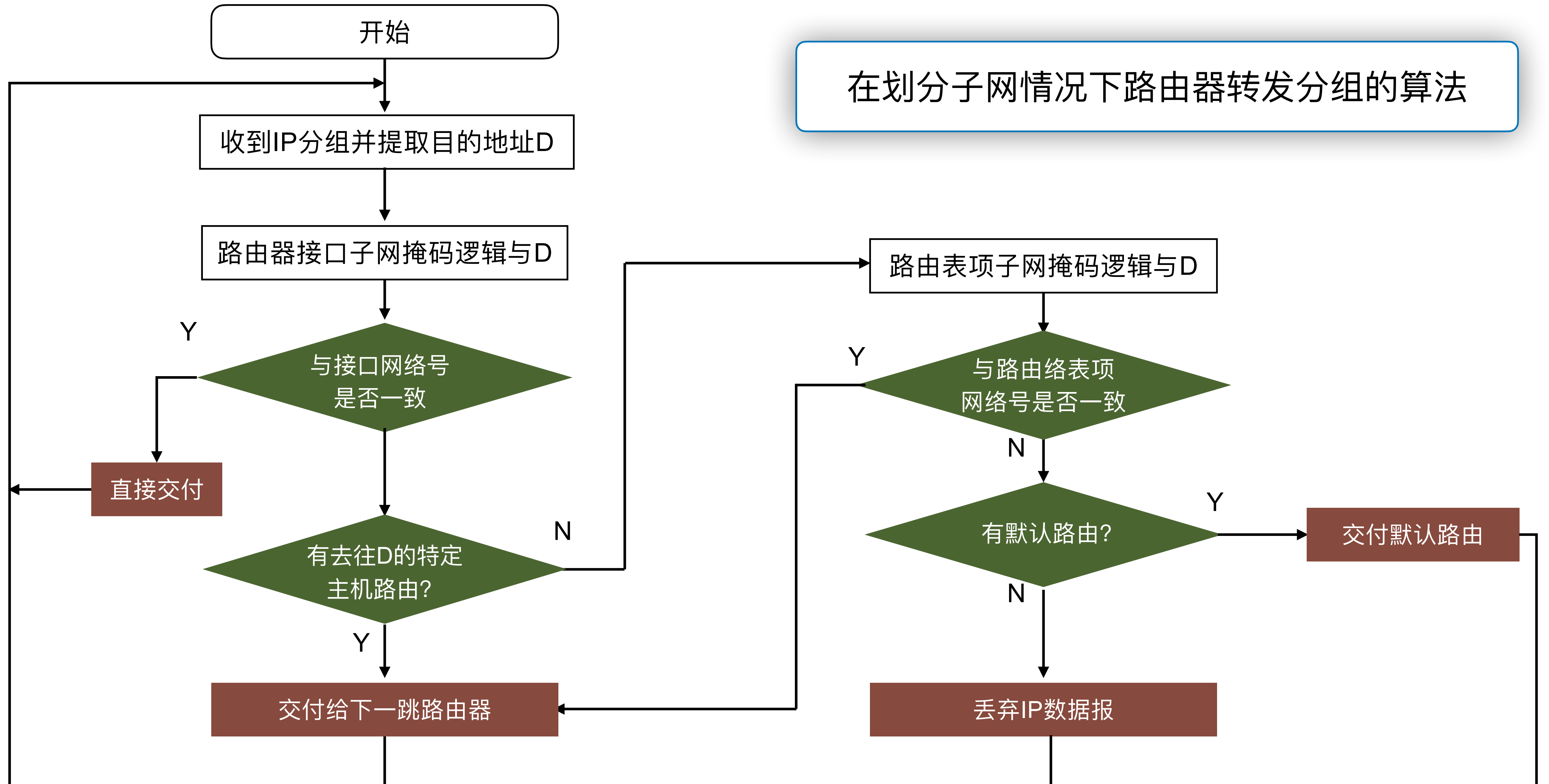
S 192.168.30.0/24 [1/0] via 192.168.20.2
C 192.168.10.0/24 is directly connected, FastEthernet1/0
C 192.168.20.0/24 is directly connected, FastEthernet0/0

子网掩码

B类子网划分

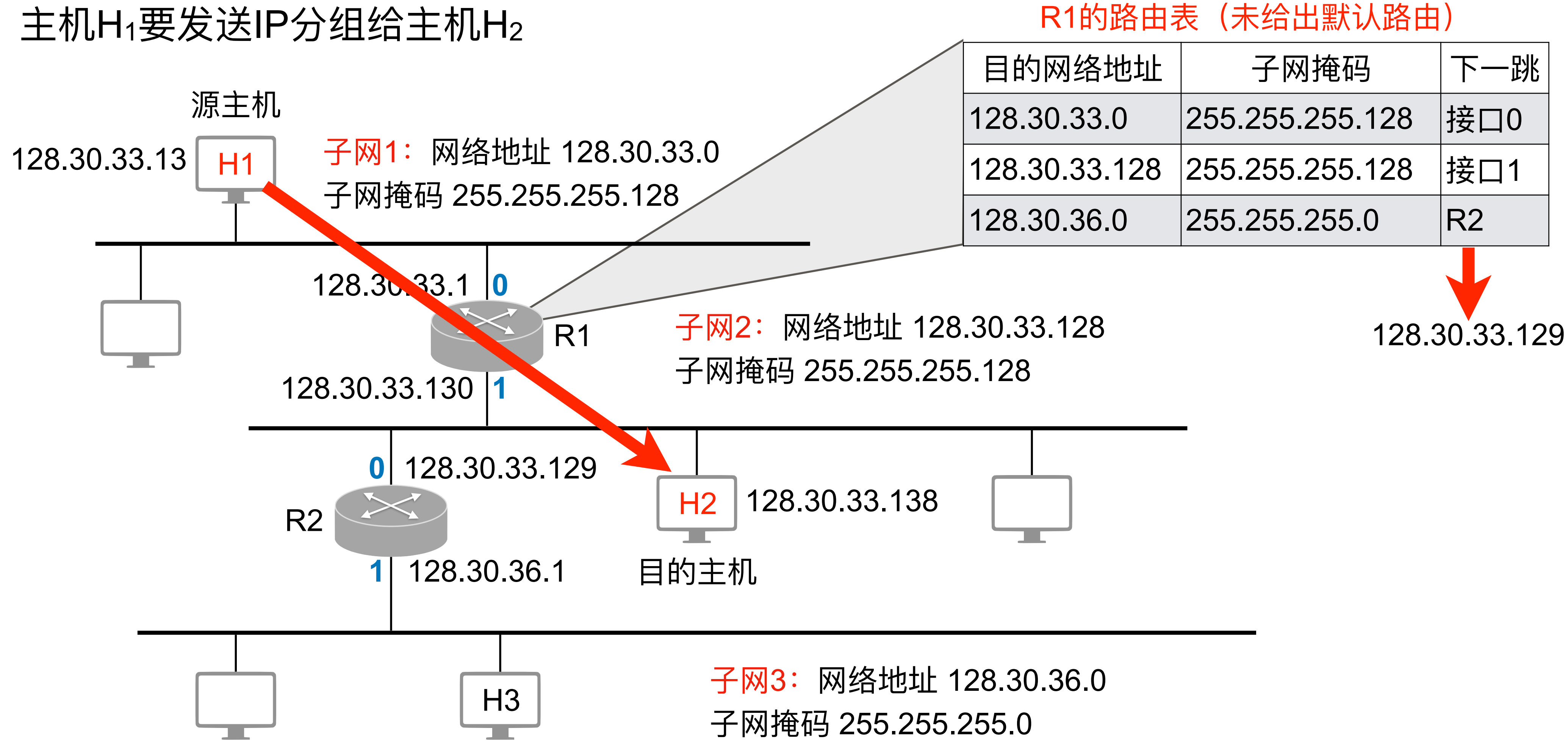
B 类地址的子网划分选择（使用固定长度子网）			
子网号的位数	子网掩码	子网数	每个子网的主机数
1	255.255.128.0	2	32766
2	255.255.192.0	4	16382
3	255.255.224.0	8	8190
4	255.255.240.0	16	4094
5	255.255.248.0	32	2046
6	255.255.252.0	64	1022
7	255.255.254.0	128	510
8	255.255.255.0	256	254
9	255.255.255.128	512	126
10	255.255.255.192	1024	62
11	255.255.255.224	2048	30
12	255.255.255.240	4096	14
13	255.255.255.248	8192	6
14	255.255.255.252	16384	2

在划分子网情况下路由器转发分组的算法

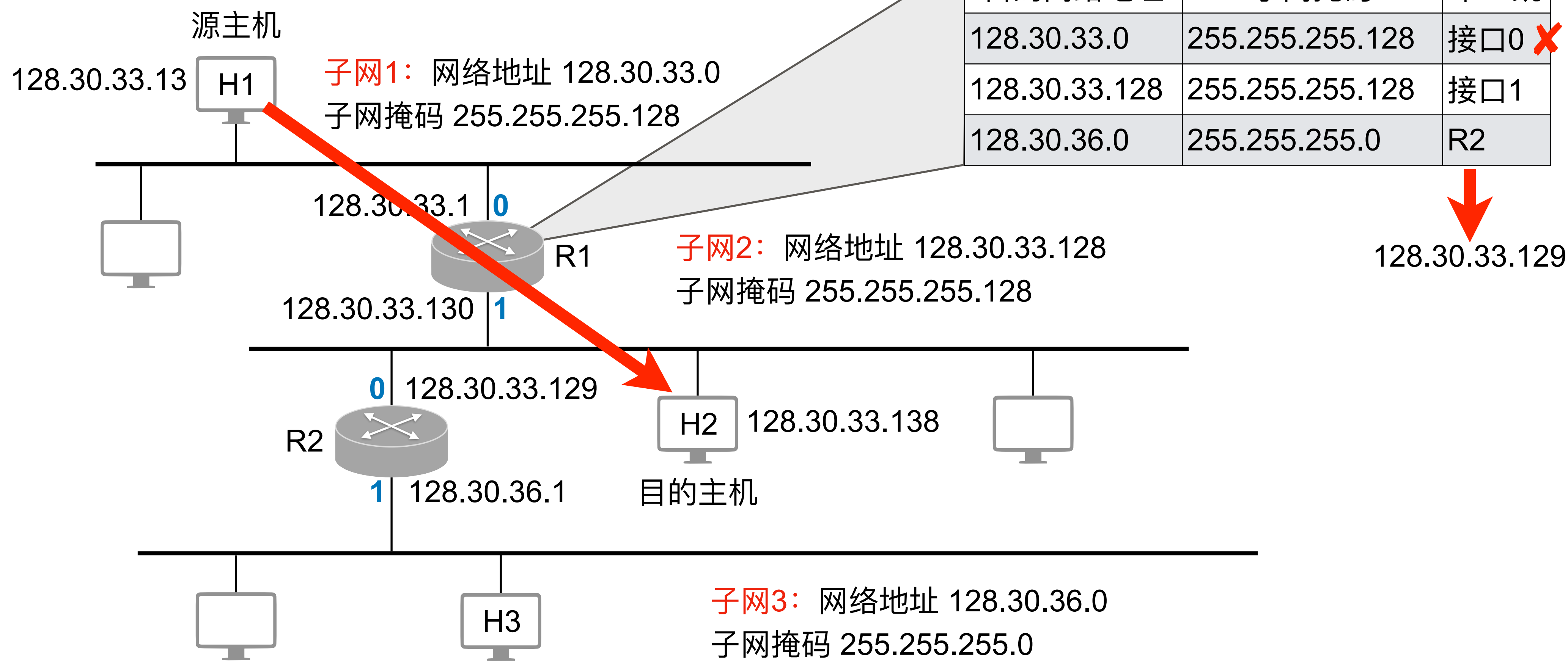


使用子网时分组转发的例子

主机H₁要发送IP分组给主机H₂



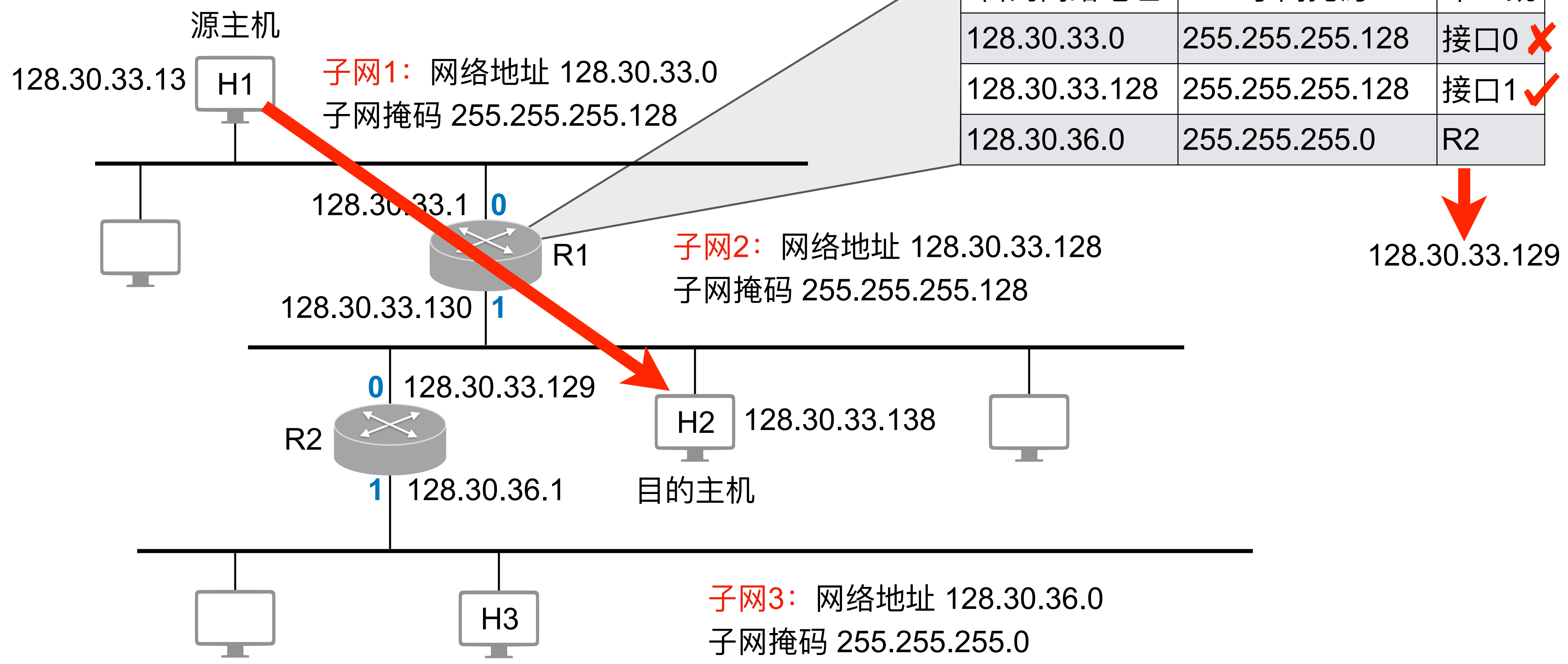
主机H₁要发送IP分组给主机H₂



H1首先检查H2是否与自己同处一个IP网络：
H1用自己的子网掩码与目标IP逻辑“与”运算，
结果：128.30.33.128与H1不在同一个IP网络
H1交付给路由器R1（H1的网关，间接交付）。

```
128. 30. 33. 1 0 0 0 1 0 1 0
255.255.255. 1 0 0 0 0 0 0 0
-----
128. 30. 33. 1 0 0 0 0 0 0 0
```

主机H₁要发送IP分组给主机H₂



路由器R1用自己接口地址的子网掩码逐条与目的IP地址进行逻辑“与”运算：
接口0与主机H1在同一网络，结果不匹配；
接口1的网络号与目标主机网络号一致（直接交付）。

128.	30.	33.	1	0	0	1	0	1	0
255.	255.	255.	1	0	0	0	0	0	0
<hr/>									
128.	30.	33.	1	0	0	0	0	0	0

已知 IP 地址是 141.14.72.24，子网掩码是 255.255.192.0。试求网络地址

点分十进制IP地址	141	14	72	24
第3字节转为二进制	141	14	0 1 0 0 1 0 0 0	24
AND				
子网掩码255.255.192.0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0
IP地址的网络地址	141	14	64	0

将子网掩码改为255.255.224.0、255.255.240.0，计算结果一样
请注意，它们的网络位数不一样，网络规模不一样。

小结

- 网络层
 - IP地址编址方法
 - 划分子网
 - 子网掩码
 - 分组转发流程
- 两级IP地址结构变为三级。
- 划分子网的基本思路（借主机位）。
- 优点：
 - 减少IP地址浪费；
 - 网络组织更加灵活；
 - 便于管理和维护。
- 子网掩码的作用（用于计算网络地址）。
- 使用子网时分组转发流程。

无分类编址CIDR

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- 新问题:
 - 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；
 - 互联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。
- 另一情况：
 - 某单位有4个部门，每个部门90人左右，该单位至少需要两个C类网络地址，才能满足需求。若用默认子网掩码，这两个C类网络不能直接通信。
- 可不可以按需分配IP地址？

IP 编址问题的演进

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- RFC 1009 规定，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)，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 CIDR (Classless Inter-Domain Routing)。

CIDR 最主要的特点

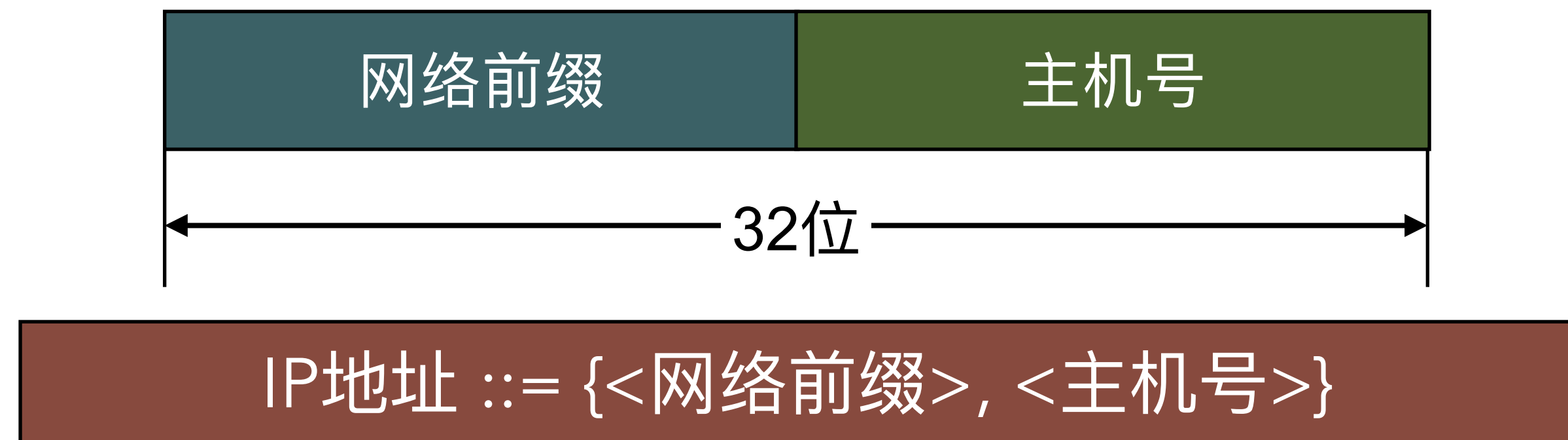
- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- CIDR 消除了A类、B类和C类地址以及划分子网的概念，更加有效地分配 IPv4 的地址空间：
 - CIDR变长的“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

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- 无分类的两级编址的记法：
 - CIDR 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) ;
 - 例如: 220.78.168.0/24。



CIDR 地址块

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- 128.14.32.0/20 表示的地址块共有 2^{12} 个地址（斜线后面的 20 是网络前缀的位数，这个地址的主机号是 12 位）：
 - 地址块的起始地址是 128.14.32.0；
 - 在不需要指出地址块的起始地址时，也可将这样的地址块简称为“/20 地址块”；
 - 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 的主机号地址不使用。

CIDR 把网络前缀都相同的连续的 IP 地址组成“CIDR 地址块”

128.14.32.0/20 表示的地址数目等价于16个C类网络

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

20位网络位不变			12位主机位		
最小IP地址					
10000000	00001110	00100000	00000000		128.14.32.0
10000000	00001110	00100000	00000001		128.14.32.1
10000000	00001110	00100000	00000010		128.14.32.2
10000000	00001110	00100000	00000011		128.14.32.3
10000000	00001110	00100000	00000100		128.14.32.4
...		
10000000	00001110	00101111	11111100		128.14.47.252
10000000	00001110	00101111	11111101		128.14.47.253
10000000	00001110	00101111	11111110		128.14.47.254
10000000	00001110	00101111	11111111		128.14.47.255
最大IP地址					

路由聚合 (route aggregation)

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- 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原来传统分类地址的路由：
 - 路由聚合减少路由器之间的路由选择信息的交换，提高了整个互联网的性能；
 - 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。

构成超网

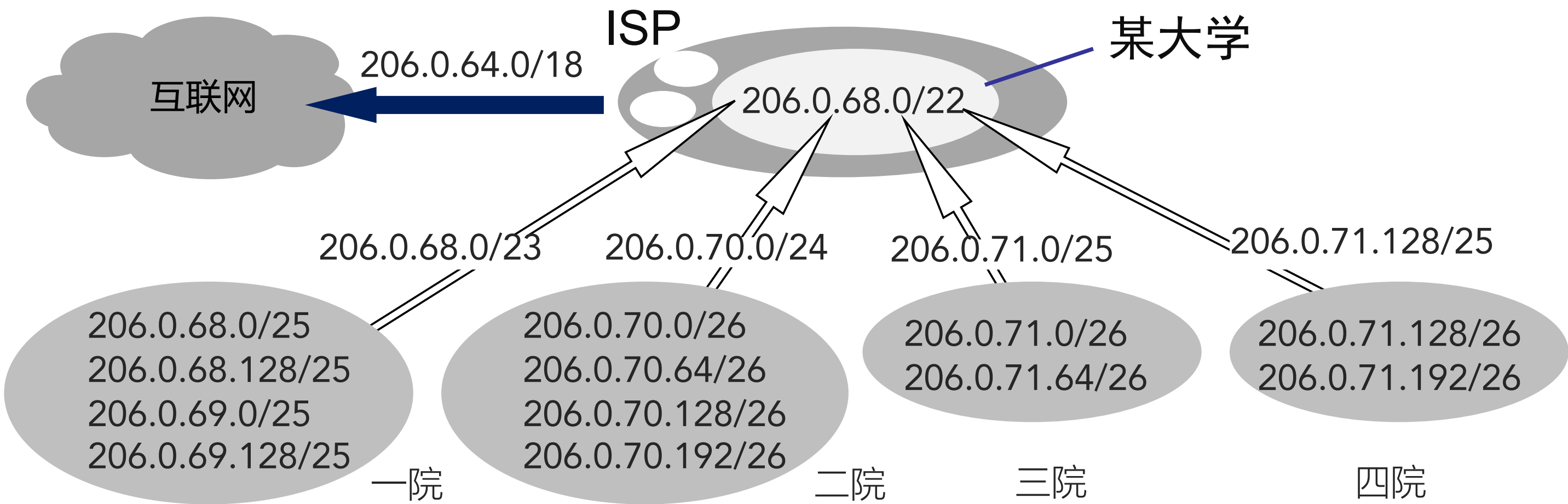
- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

- 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 的一个好处是：可以更加有效地分配 IPv4 的地址空间，可根据客户的需要分配适当大小的 CIDR 地址块。

CIDR 前缀长度	点分十进制	包含的地址数	相当于包含分类的网络数
/13	255.248.0.0	512 K	8个B类或2048个C类
/14	255.252.0.0	256 K	4个B类或1024个C类
/15	255.254.0.0	128 K	2个B类或512个C类
/16	255.255.0.0	64 K	1个B类或256个C类
/17	255.255.128.0	32 K	128个C类
/18	255.255.192.0	16 K	64个C类
/19	255.255.224.0	8 K	32个C类
/20	255.255.240.0	4 K	16个C类
/21	255.255.248.0	2 K	8个C类
/22	255.255.252.0	1 K	4个C类
/23	255.255.254.0	512	2个C类
/24	255.255.255.0	256	1个C类
/25	255.255.255.128	128	1/2个C类
/26	255.255.255.192	64	1/4个C类
/27	255.255.255.224	32	1/8个C类

CIDR的一个实例

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配



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
一院	206.0.68.0/23	11001110.00000000.0100010*	512
二院	206.0.70.0/24	11001110.00000000.01000110.*	256
三院	206.0.71.0/25	11001110.00000000.01000111.0*	128
四院	206.0.71.128/25	11001110.00000000.01000111.1*	128

ISP 共有 64 个 C 类网络。如果不采用 CIDR 技术，则在与该 ISP 的路由器交换路由信息的每一个路由器的路由表中，就需要有 64 个项目。但采用地址聚合后，只需用路由聚合后的 1 个项目 206.0.64.0/18 就能找到该 ISP。

最长前缀匹配

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 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

收到的分组的地址 $D = 206.0.71.130$

路由表中的项目：
206.0.68.0/22 1
206.0.71.128/25 2

网络206.0.68.0/22, 第1条路由表项, 掩码: 255.255.252.0
网络206.0.71.128/25, 第2条路由表项, 掩码: 255.255.255.128

目的地址与第1条路由的掩码逻辑“与”运算:

206.	0.	01000111.130	
<u>11111111 11111111 11111100 00000000</u>			
206.	0.	68.	0

结果与路由表项第1条网络号匹配

目的地址与第2条路由的掩码逻辑“与”运算:

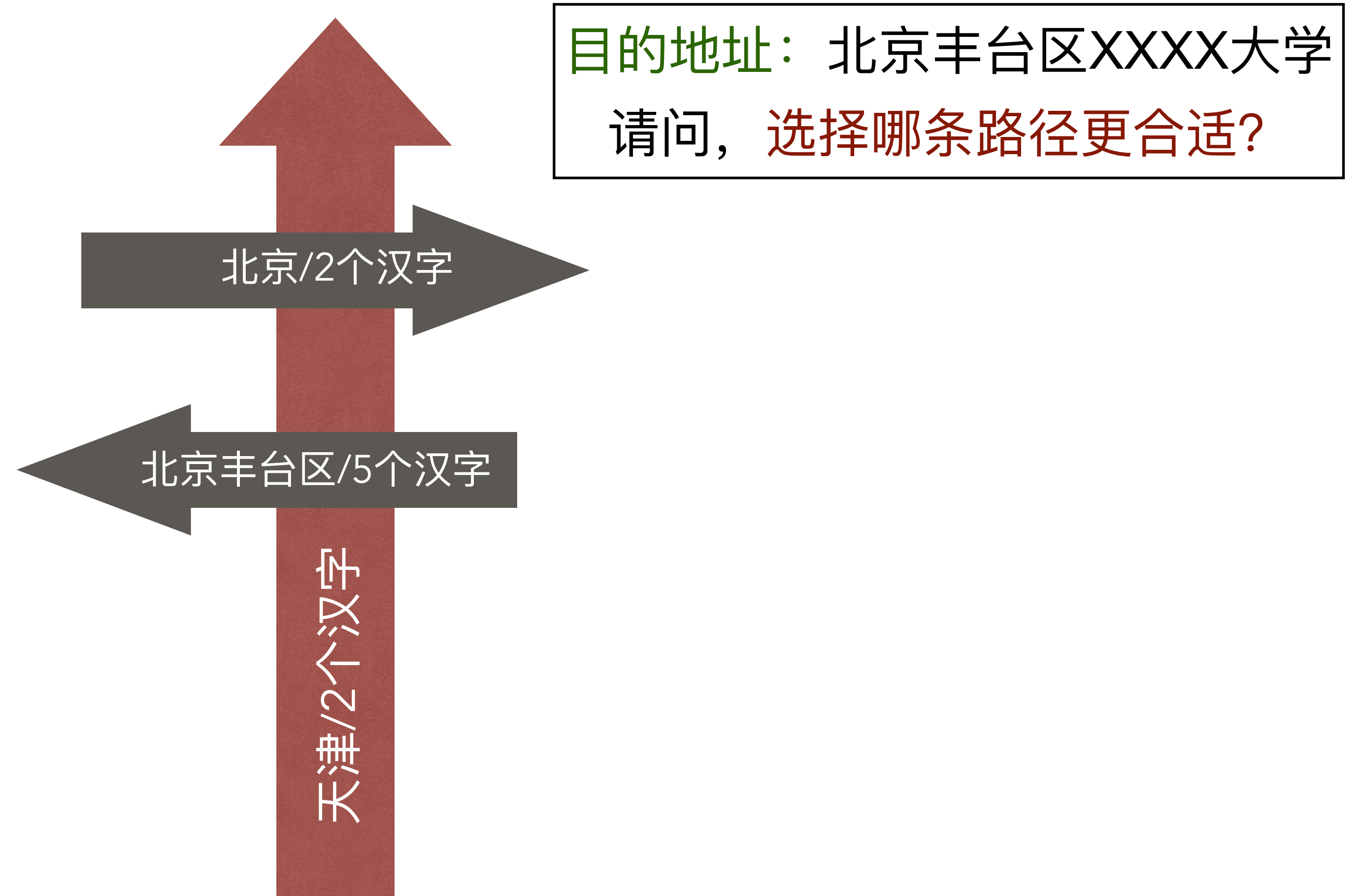
206.	0.	01000111.10000010	
<u>11111111 11111111 11111111 10000000</u>			
206.	0.	71.	128

结果与路由表项第2条网络号匹配

最终路由器选择两个匹配的地址中更具体的一个, 即第2条路由表项。

最长前缀匹配实例

- 网络层
 - CIDR概述
 - 路由聚合
 - 超网的概念
 - CIDR应用实例
 - 最长前缀匹配

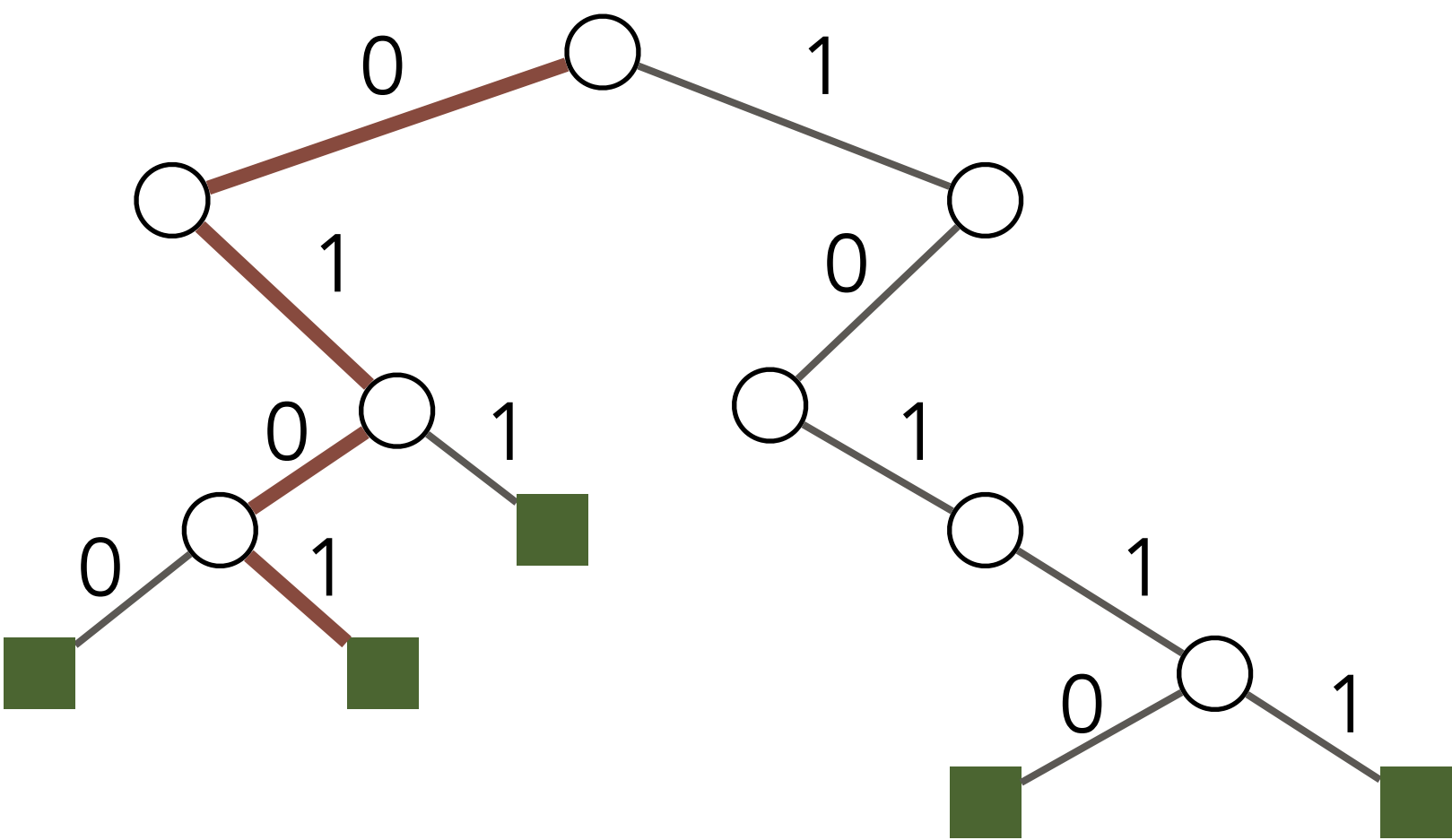


使用二叉线索查找路由表

- 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)。

32 位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



- IP地址存入二叉线索的规则：先检查IP地址左边的第一位，如为 0，则第一层的节点在根节点的左下方；如为 1，则在右下方。然后再检查地址的第二位，构造出第二层的节点。依此类推，直到唯一前缀的最后一位。

小结

- 网络层
- CIDR概述
- 路由聚合
- 超网的概念
- CIDR应用实例
- 最长前缀匹配

- 无分类编址CIDR：
 - 网络前缀；
 - 变长子网掩码。
- CIDR的特点：
 - IP地址从三级又回到两级；
 - 斜线记法。
- CIDR地址块：
 - 路由聚合；
 - 构成超网。
- 二叉线索查找路由表。
- 最长匹配原则。

网际控制报文协议 ICMP (Internet Control Message Protocol)

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

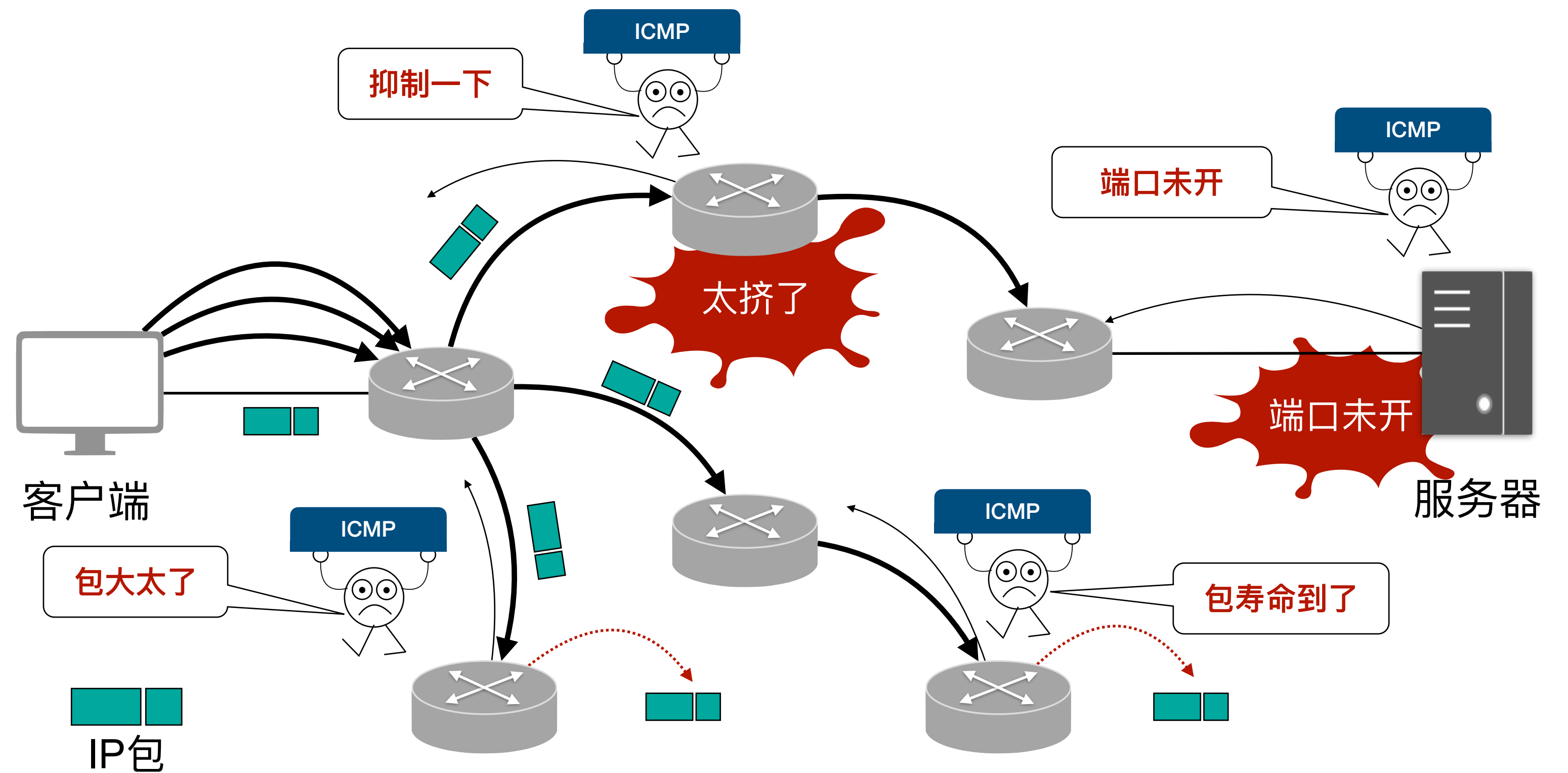
- 为什么需要ICMP协议？
 - 为了更有效地转发 IP 数据报和提高交付成功的机会，在网际层使用了网际控制报文协议 ICMP；
 - ICMP 使主机或路由器报告差错情况和提供有关异常情况的报告；
 - ICMP 不是高层协议（看起来好像是高层协议，因为 ICMP 报文是装在 IP 数据报中，作为其中的数据部分），是 IP 层的协议。

ICMP分担IP的一部分功能：

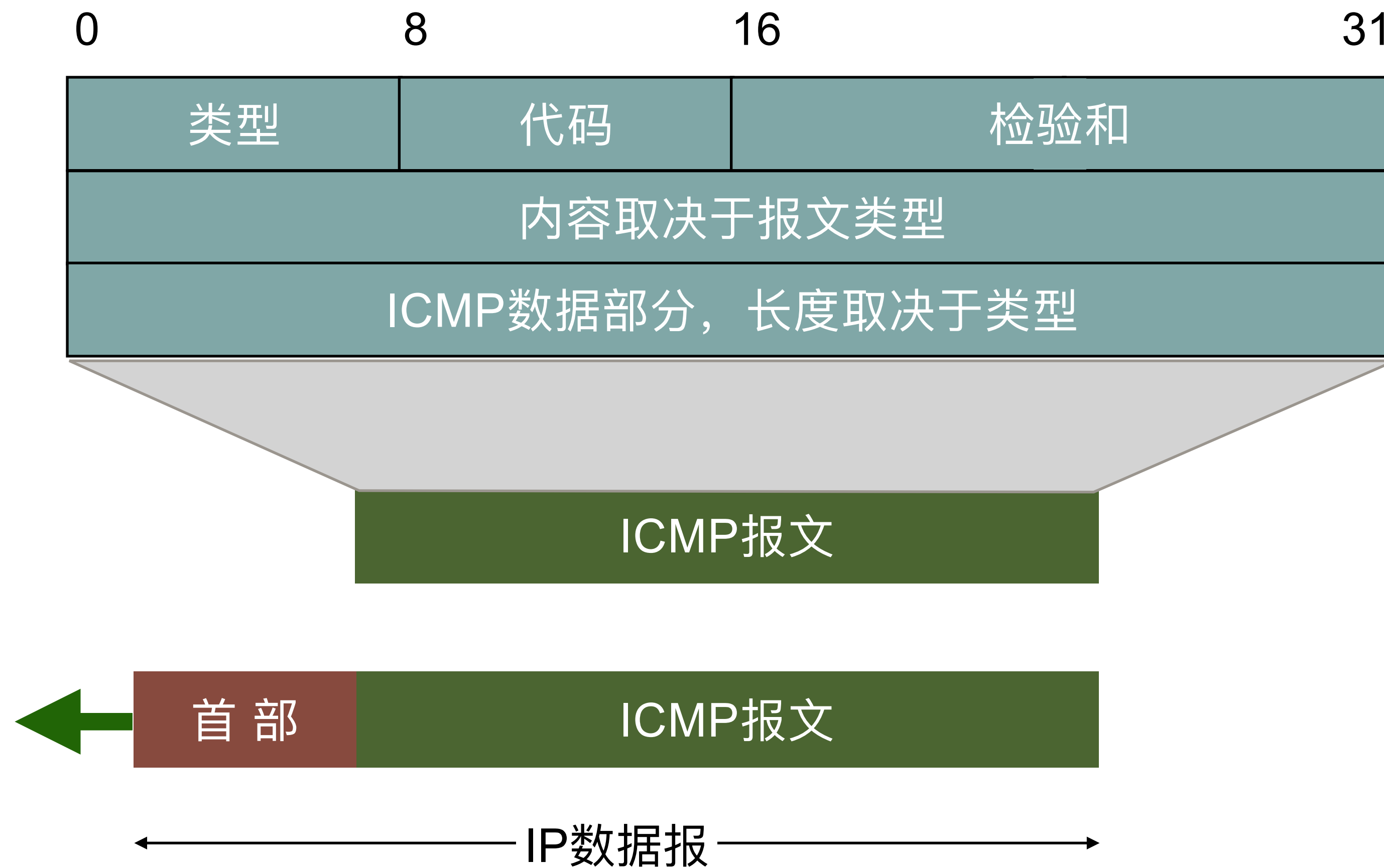
- 差错报告：目的不可达、源站抑制、超时、参数问题、重定向；
- 查询：回送请求/应答、地址掩码请求/应答、时间戳请求/应答。

ICMP 的作用

- 网络层
 - ICMP协议
 - **ICMP的作用**
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例



ICMP报文的格式



ICMP前4个字节都是一样的。

ICMP报文类型：

- 差错报告报文；
- 询问应答报文。

类型与代码的组合：

- 表示需要报告的详细信息。

Checksum检验和：

- 对整个ICMP报文进行检验。

ICMP 差错报告报文共有 4 种

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - **ICMP报文类型**
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

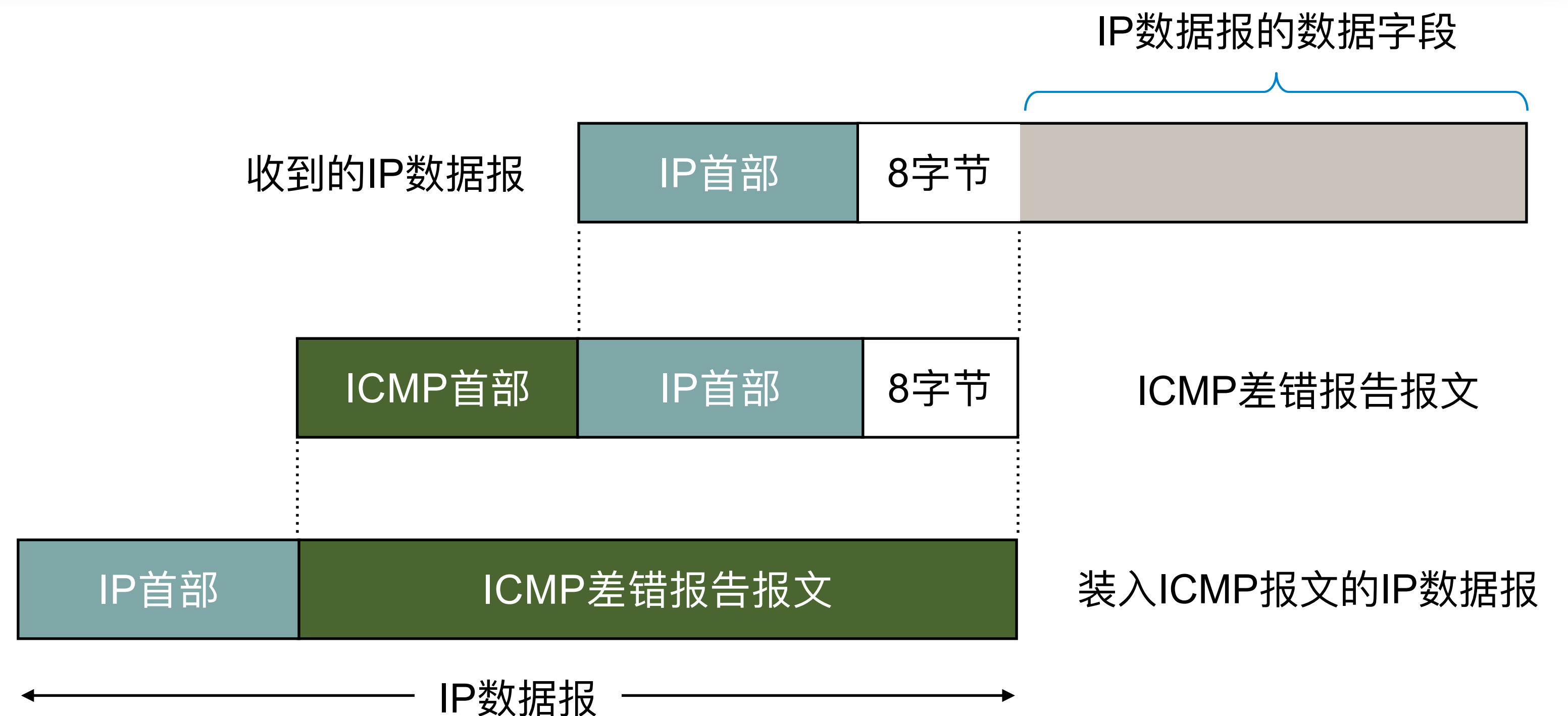
- 终点不可达。
- 时间超过 。
- 参数问题 。
- 改变路由（重定向）(Redirect) 。

Type	Code	Description	Query	Error
0	0	Echo Reply: 回送回答 (ping应答)	√	
3	0	Network Unreachable: 网络不可达		√
3	1	Host Unreachable: 主机不可达		√
3	2	Protocol Unreachable: 协议不可达		√
3	3	Port Unreachable: 端口不可达		√
3	5	Source routing failed: 源站选路失败		√
3	6	Destination network unknown: 目的网络未知		√
3	7	Destination host unknown: 目的主机未知		√
5	1	Redirect for host: 主机重定向		√
8	0	Echo request: 回送请求 (ping请求)	√	
11	0	TTL equals 0 during transit: 传输期间生存时间为0		√
12	0	IP header bad (catchall error): 坏的IP首部 (包括各种差错)		√
17	0	Address mask request: 地址掩码请求	√	
18	0	Address mask reply: 地址掩码应答	√	

ICMP 差错报告报文的数据字段的内容

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - **ICMP报文类型**
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

- ICMP数据字段为什么需要**加上原始出错IP**中的数据字段**前8个字节**？
 - 告诉源IP数据报发送端，是哪一个进程发送的IP数据报出错。



不应发送 ICMP 差错报告报文的几种情况

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - **ICMP报文类型**
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

- 对 ICMP **差错报告报文** 不再发送 ICMP 差错报告报文。
- 对 **第一个分片的数据报分片** 的所有后续数据报片都不发送 ICMP 差错报告报文。
- 对具有 **多播地址** 的数据报都不发送 ICMP 差错报告报文。
- 对具有 **特殊地址**（如127.0.0.0 或 0.0.0.0）的数据报不发送 ICMP 差错报告报文。

ICMP 询问报文有两种

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - **ICMP询问报文**
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

- **回送请求和回答报文。**
 - 请求: type = 8, code = 0;
 - 回答: type = 0, code = 0。
- **时间戳请求和回答报文。**
 - 请求: type = 13, code = 0;
 - 回答: type = 14, code = 0。
- **下面的几种 ICMP 报文不再使用:**
 - 信息请求与回答报文;
 - 掩码地址请求和回答报文;
 - 路由器询问和通告报文;
 - 源点抑制报文。

ICMP的应用举例：回送请求和回答报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

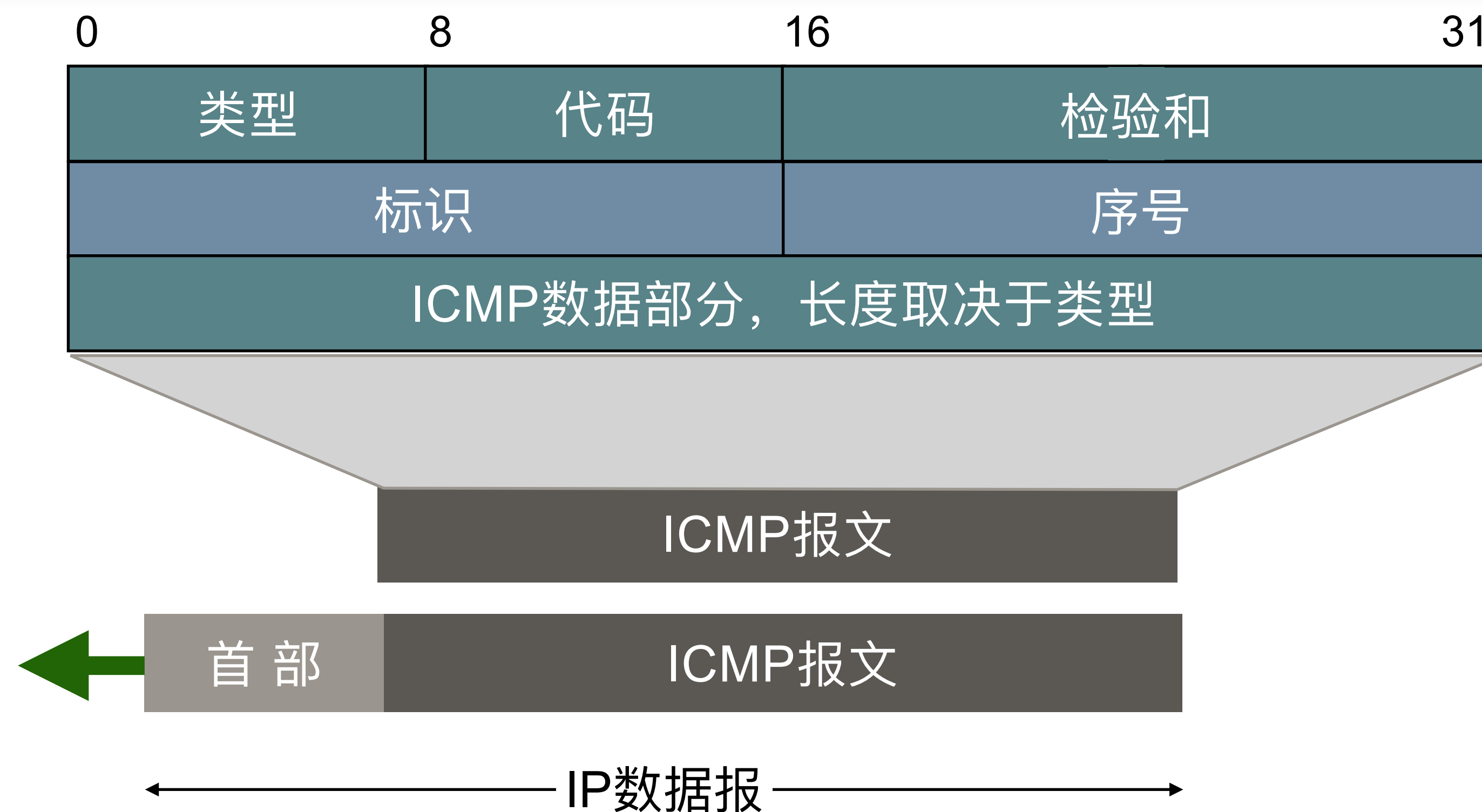
PING (Packet InterNet Groper) :

- PING 用来测试两个主机之间的连通性；
- PING 使用了 ICMP 回送请求与回送回答报文；
- PING 是应用层直接使用网络层 ICMP 的例子，它没有通过运输层的 TCP 或UDP。

ping采用的ICMP协议格式

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

- 标识和序号是用来实现类似TCP协议或UDP协议的端口号功能，用以区分“询问和应答”对。



ping命令实例

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

```
li@ubuntu1604:~$ ping -c 3 www.baidu.com
PING www.wshifen.com (103.235.46.39) 56(84) bytes of data.
64 bytes from 103.235.46.39: icmp_seq=1 ttl=38 time=387 ms
64 bytes from 103.235.46.39: icmp_seq=2 ttl=38 time=470 ms
64 bytes from 103.235.46.39: icmp_seq=3 ttl=38 time=472 ms

--- www.wshifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 387.418/443.393/472.366/39.596 ms
li@ubuntu1604:~$
```

ICMP回送请求报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

#类型值为8

Code: 0

#代码值为0, 询问报文

Checksum: 0x5ff9 [correct]

#检验和

Identifier (BE): 49169 (0xc011)

#用于标识ping进程

Identifier (LE): 4544 (0x11c0)

#用于标识ping进程

Sequence number (BE): 1 (0x0001)

#用于标识ping进程

Sequence number (LE): 256 (0x0100)

#用于标识ping进程

Data (56 bytes)

#数据

ICMP回送回答报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

#类型值为0

Code: 0

#代码值为0, 应答报文

Checksum: 0x67f9 [correct]

#检验和

Identifier (BE): 49169 (0xc011)

#用于标识ping进程

Identifier (LE): 4544 (0x11c0)

#用于标识ping进程

Sequence number (BE): 1 (0x0001)

#用于标识ping进程

Sequence number (LE): 256 (0x0100)

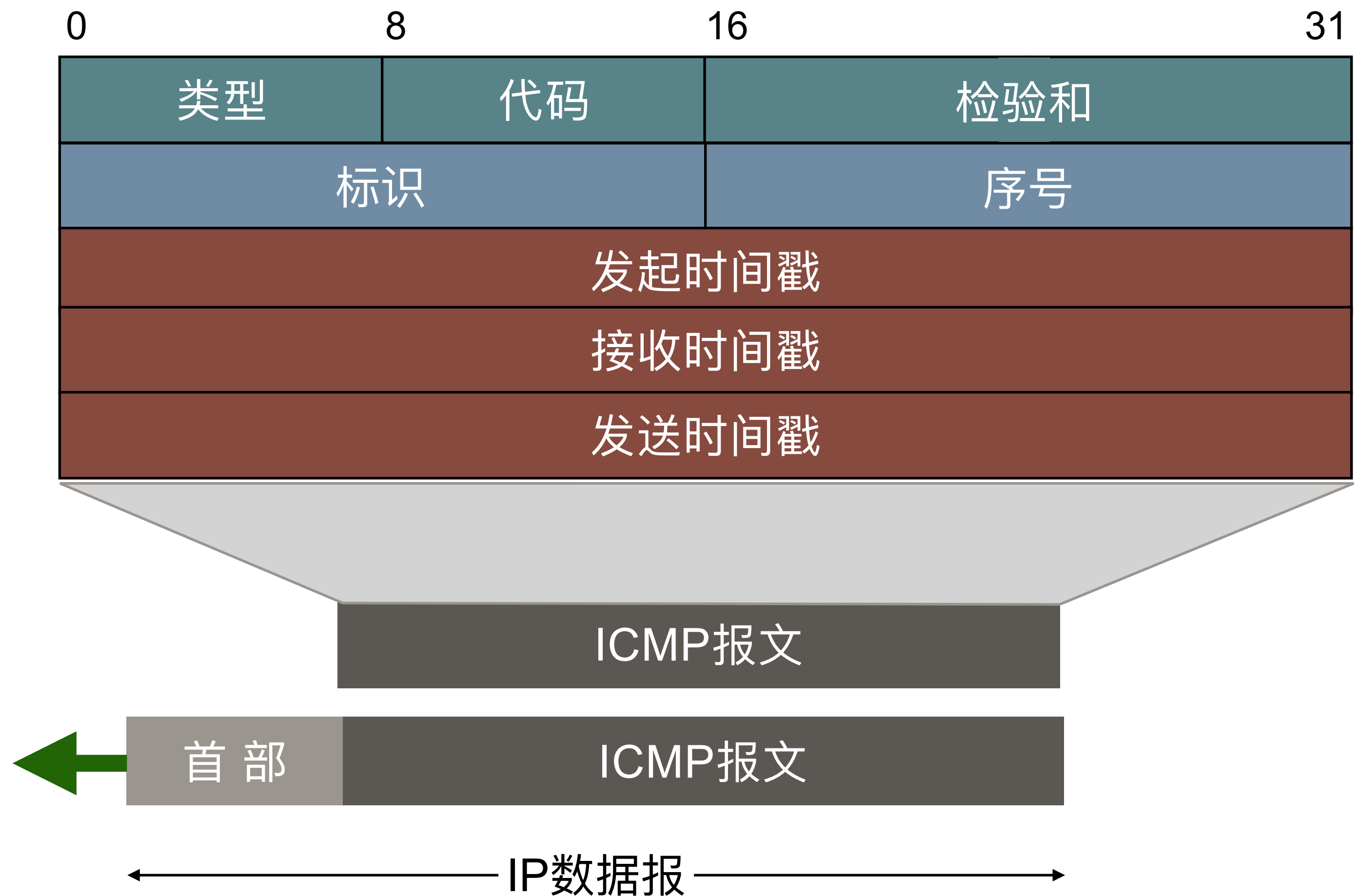
#用于标识ping进程

Data (56 bytes)

#数据

ICMP的应用举例：时间戳请求和回答报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例



ICMP时间戳请求报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

Internet Protocol Version 4, Src: 192.168.1.11, Dst: 192.168.1.1

Internet Control Message Protocol

Type: 13 (Timestamp request)

Code: 0

Checksum: 0xf453 [correct]

[Checksum Status: Good]

Identifier (BE): 0 (0x0000)

Identifier (LE): 0 (0x0000)

Sequence number (BE): 0 (0x0000)

Sequence number (LE): 0 (0x0000)

Originate timestamp: 27262446

Receive timestamp: 27262447

Transmit timestamp: 27262447

ICMP时间戳回答报文

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.11

Internet Control Message Protocol

Type: 14 (Timestamp reply)

Code: 0

Checksum: 0xae4d [correct]

[Checksum Status: Good]

Identifier (BE): 0 (0x0000)

Identifier (LE): 0 (0x0000)

Sequence number (BE): 0 (0x0000)

Sequence number (LE): 0 (0x0000)

Originate timestamp: 27262446

Receive timestamp: 56434357

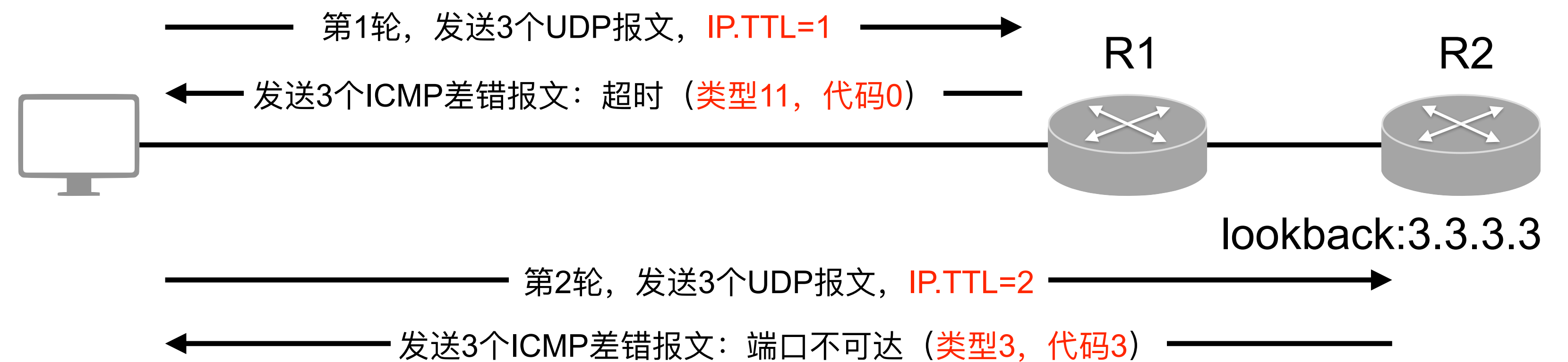
Transmit timestamp: 56434357

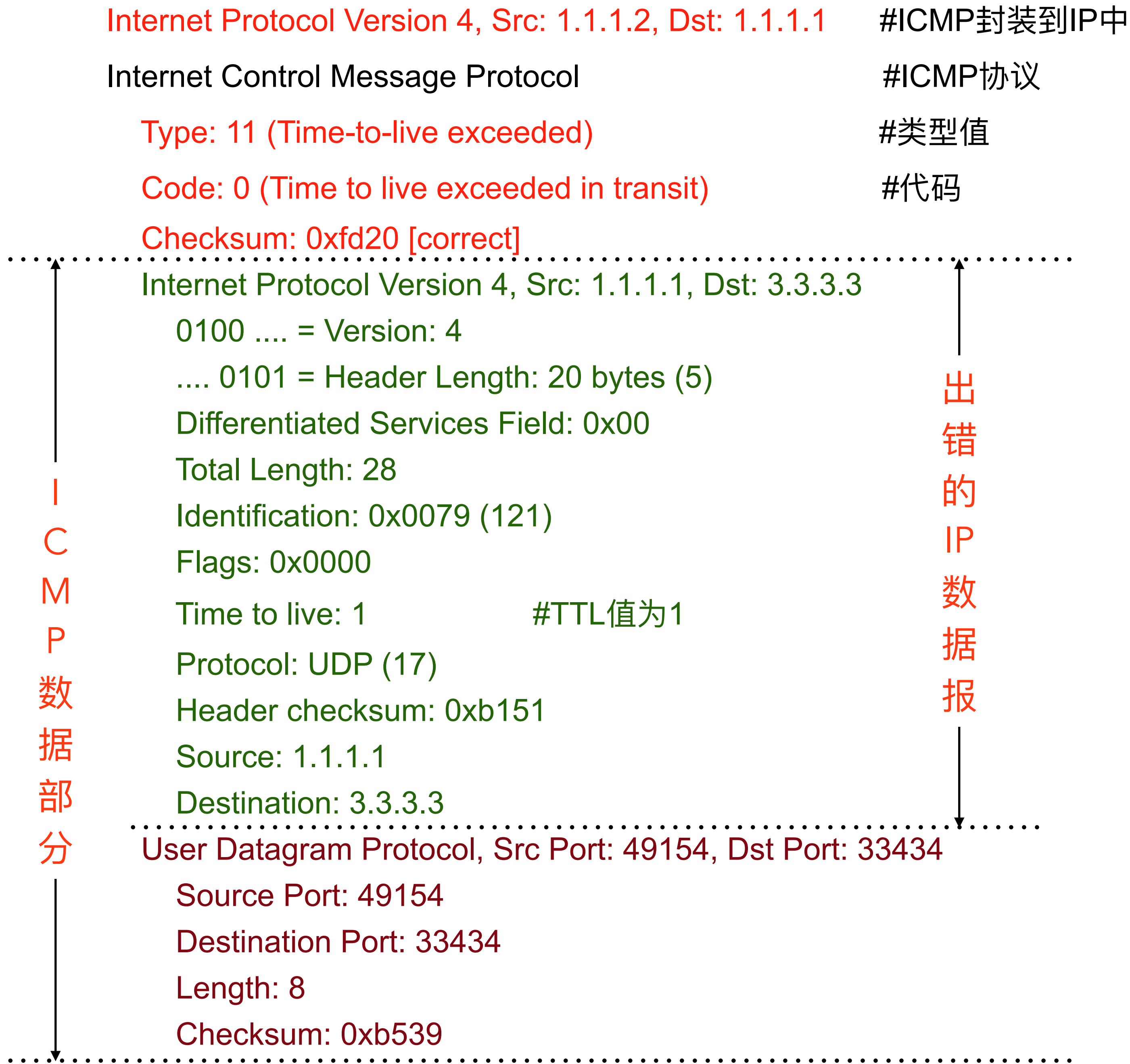
Traceroute 的应用举例 (Windows中: tracert)

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

用来跟踪一个分组从源点到终点的路径。

- 利用 IP 数据报中的 TTL 字段和 ICMP 时间超过差错报告报文实现对从源点到终点的路径的跟踪。





目标端口不可达差错报告报文

Internet Protocol Version 4, Src: 2.2.2.2, Dst: 1.1.1.1	#ICMP报文封装到IP中
Internet Control Message Protocol	#ICMP报文
Type: 3 (Destination unreachable)	#目标不可达
Code: 3 (Port unreachable)	#端口不可达
Checksum: 0x051e [correct]	
Unused: 00000000	
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 3.3.3.3	#出错的IP数据报
User Datagram Protocol, Src Port: 49159, Dst Port: 33439	#出错IP数据报数据部分
Source Port: 49159	
Destination Port: 33439	#目标主机中该端口没有开启
Length: 8	
Checksum: 0xb52f	

小结

- 网络层
 - ICMP协议
 - ICMP的作用
 - ICMP报文格式
 - ICMP报文类型
 - ICMP询问报文
 - 回送请求/回答
 - 时间戳请求/回答
 - ICMP应用举例

- ICMP的作用。
- 两类ICMP报文：
 - 询问应答报告报文；
 - 差错报告报文。
- ICMP差错报文格式：
 - 类型、代码；
 - Ping的ICMP报文格式；
 - 不发送 ICMP 差错报告报文的几种情况。
- 两个应用程序：
 - ping；
 - traceroute。

互联网的路由选择协议

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 路由选协议的几个概念。
- 内部网关协议 RIP 。
- 内部网关协议 OSPF 。
- 外部网关协议 BGP。
- 路由器的构成。

理想的路由算法

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 算法必须是正确的和完整的。
- 算法在计算上应简单。
- 算法应能适应通信量和网络拓扑的变化，要有自适应性。
- 算法应具有稳定性。
- 算法应是公平的。
- 算法应是最佳的。

关于“最佳路由”

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 不存在一种绝对的最佳路由算法：
 - 所谓“最佳”只能是相对于某一种特定要求下得出的较为合理的选择而已。
 - 实际的路由选择算法，应尽可能接近于理想的算法。
- 路由选择是个非常复杂的问题：
 - 它是网络中的所有结点共同协调工作的结果；
 - 路由选择的环境往往是不不断变化的，而这种变化有时无法事先知道。

从路由算法的自适应性考虑

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 静态路由选择策略：
 - 即非自适应路由选择，其特点是简单和开销较小，但不能及时适应网络状态的变化。
- 动态路由选择策略：
 - 自适应路由选择，其特点是能较好地适应网络状态的变化，但实现起来较为复杂，开销也比较大。

分层次的路由选择协议

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 互联网为什么采用分层次的路由选择协议：
 - 互联网的规模非常大。如果让所有的路由器知道所有的网络应怎样到达，则这种路由表将非常大，处理起来也太花时间。而所有这些路由器之间交换路由信息所需的带宽就会使互联网的通信链路饱和；
 - 许多单位不愿意外界了解自己单位网络的布局细节和本部门所采用的路由选择协议（这属于本部门内部的事情），但同时还希望连接到互联网上。

自治系统 AS(Autonomous System)

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 自治系统 **AS** 的定义：
 - 在**单一的技术管理下的一组路由器**，而这些路由器使用一种 AS 内部的路由选择协议和**共同的度量**以确定分组在该 AS 内的路由，同时还使用一种 AS 之间的路由选择协议用以确定分组在**AS之间的路由**。

现在对自治系统 AS 的定义是**强调下面的事实**：

- 尽管一个 AS **使用了多种内部路由选择协议和度量**，但重要的是一个 AS 对其他 AS 表现出的是一个**单一的和一致的路由选择策略**。一个规模较大的ISP就是一个AS。

一个自治系统(AS)是一个有权自主地决定在本系统中应采用何种路由协议的小型单位。一个自治系统有时也被称为是一个**路由选择域** (routing domain) 。

互联网有两大类路由选择协议

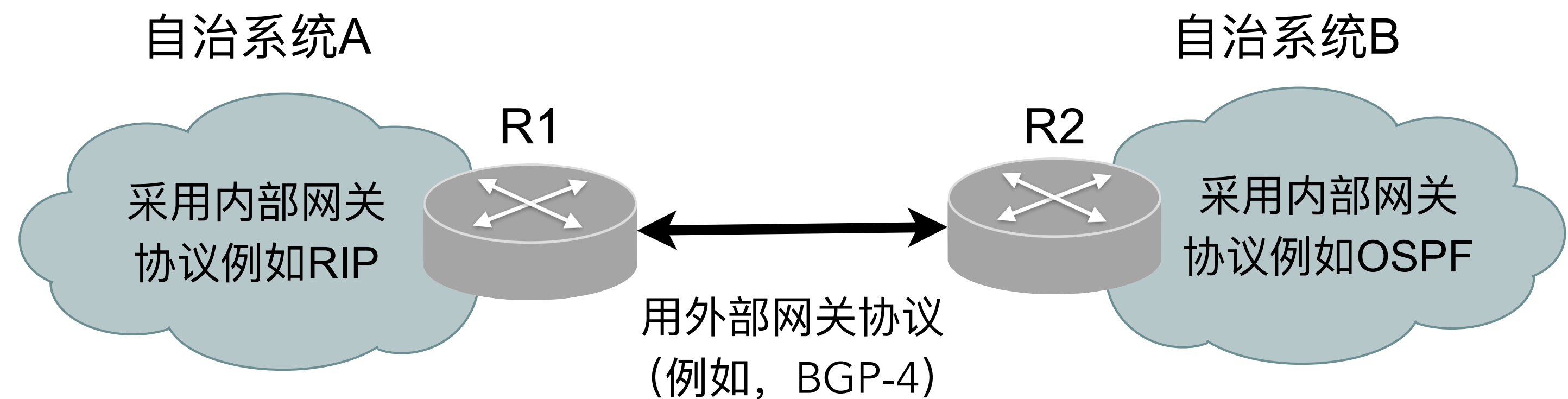
- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 内部网关协议 IGP (Interior Gateway Protocol) :
 - 在一个自治系统内部使用的路由选择协议;
 - 目前这类路由选择协议使用得最多, 如 RIP 和 OSPF 协议。
- 外部网关协议 EGP (External Gateway Protocol) :
 - 若源站和目的站处在不同的自治系统中, 就需要使用一种协议将路由选择信息传递到另一个自治系统中, 这种协议就是外部网关协议 EGP;
 - 在外部网关协议中目前使用最多的是 BGP-4。

自治系统和内部网关协议、外部网关协议

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
- 自治系统

- 自治系统之间的路由选择也叫作域间路由选择 (interdomain routing)
- 在自治系统内部的路由选择叫作域内路由选择 (intradomain routing)



小结

- 网络层
 - 路由选择协议
 - 路由算法要求
 - 路由算法分类
 - 自治系统

- 理想的路由选择算法。
- 最佳路由。
- 静态路由选择策略。
- 动态路由选择策略。
- 分层次路由选择。
- 自治系统AS。

内部网关协议 RIP

- 网络层
- **RIP路由选择协议概述**
- RIP的特点
- 距离向量算法
- RIP的优缺点

- **简单介绍：**
 - RIP 是一种分布式的、基于距离向量的路由选择协议；
 - 运行RIP协议的路由器，维护从它自己到其他每一个目的网络的距离记录。
- **距离的定义：**
 - 从一个路由器到直接连接的网络的距离定义为 1（实际为0）；
 - 从一个路由器到非直接连接的网络的距离定义为所经过的路由器数加 1；
 - “距离”也称为“跳数”(hop count)，因为每经过一个路由器，跳数就加 1；
 - 这里的“距离”实际上指的是“最短距离”。

内部网关协议 RIP

- 网络层
- RIP路由选择协议概述
- RIP的特点
- 距离向量算法
- RIP的优缺点

- RIP 认为一个好的路由就是它通过的**路由器的数目少**，即“距离短”。
- RIP 允许一条路径**最多只能包含 15 个路由器**。
- “距离”的最大值为 16 时即相当于不可达。可见 RIP 只适用于**小型互联网**。
- RIP **不能**在两个网络之间同时**使用多条路由**。
- RIP 选择一个具有最少路由器的路由（即最短路由），哪怕还存在另一条高速(低时延)但路由器较多的路由。

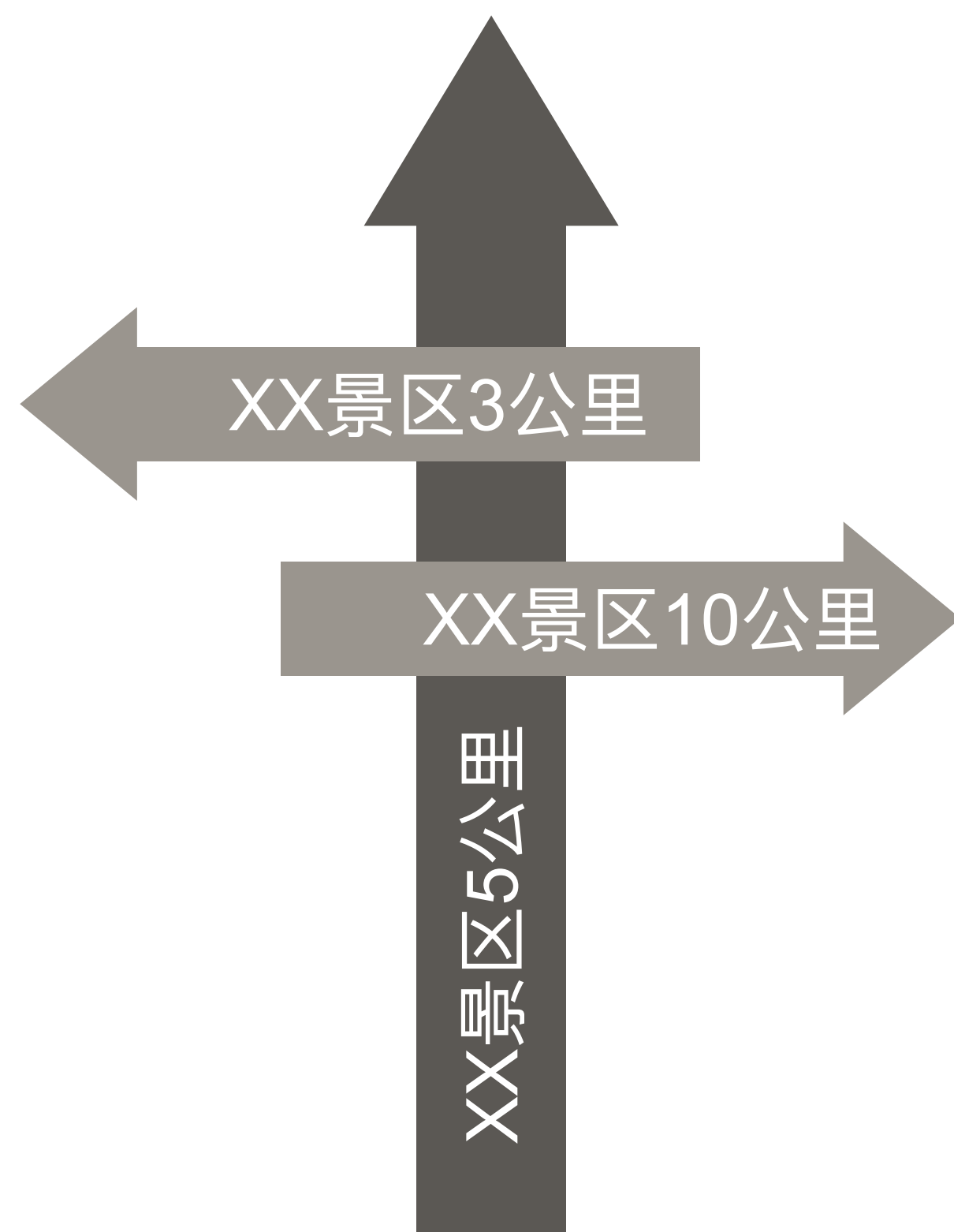
RIP 协议的三个特点

- 网络层
- RIP路由选择协议概述
- **RIP的特点**
- 距离向量算法
- RIP的优缺点

- **和谁交换信息**：仅和相邻路由器交换信息。
- **交换什么信息**：交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
- **什么时候交换**：按固定的时间间隔交换路由信息，例如，每隔 30 秒。当网络拓扑发生变化时，路由器也及时向相邻路由器通告拓扑变化后的路由信息。

WWW: Who? 、What? 、When?

路由表的建立



- 路由器刚开始工作时，只知道到**直接连接的网络的距离**（此距离定义为1）。
- 以后，每一个路由器也只**和相邻路由器交换**并更新路由信息。
- **经过若干次更新后**，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。
- **RIP路由表项**：目的网络，距离，下一跳。

RIP 协议的收敛 (convergence) 过程较快。

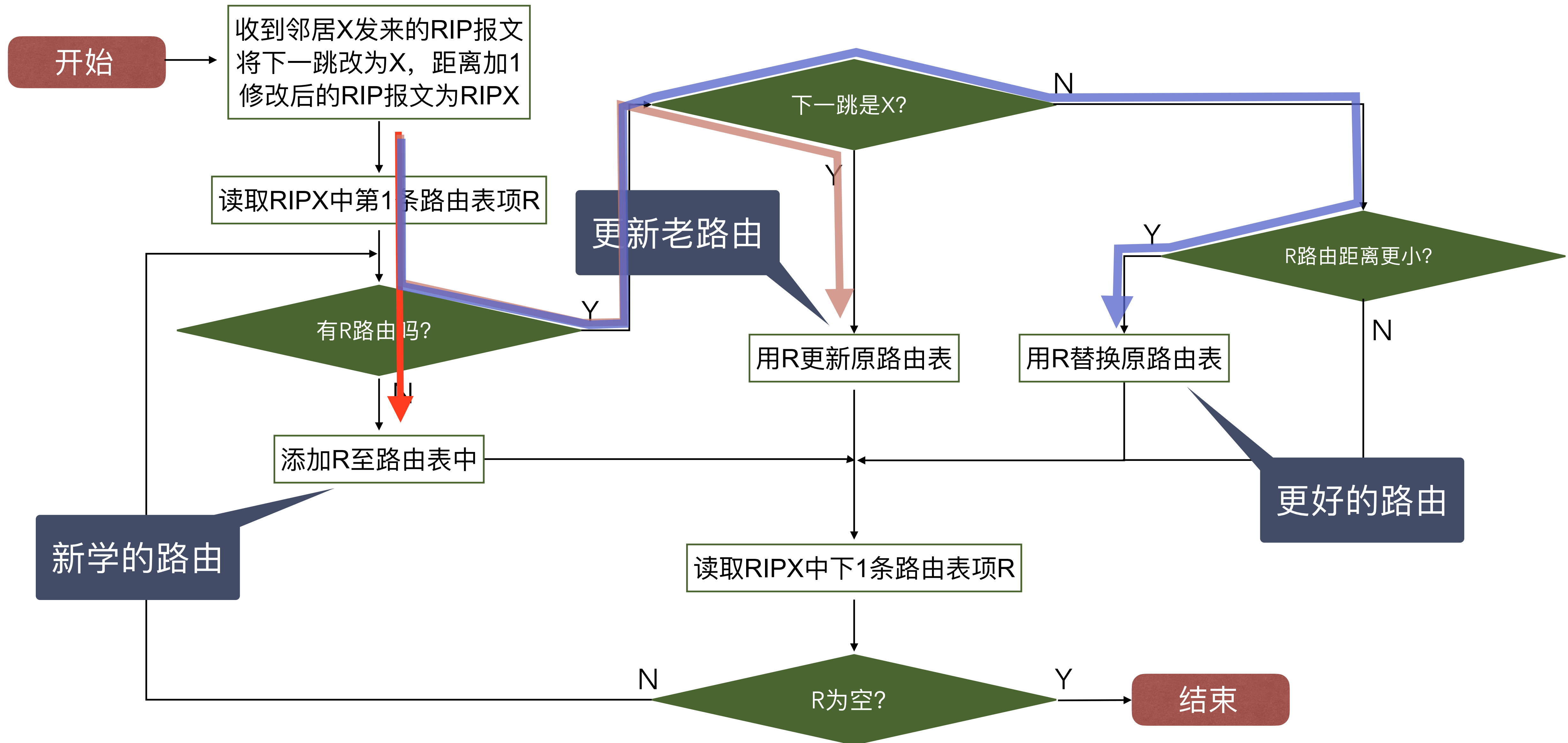
“收敛”：自治系统中所有的结点都得到正确的路由选择信息的过程。

距离向量算法

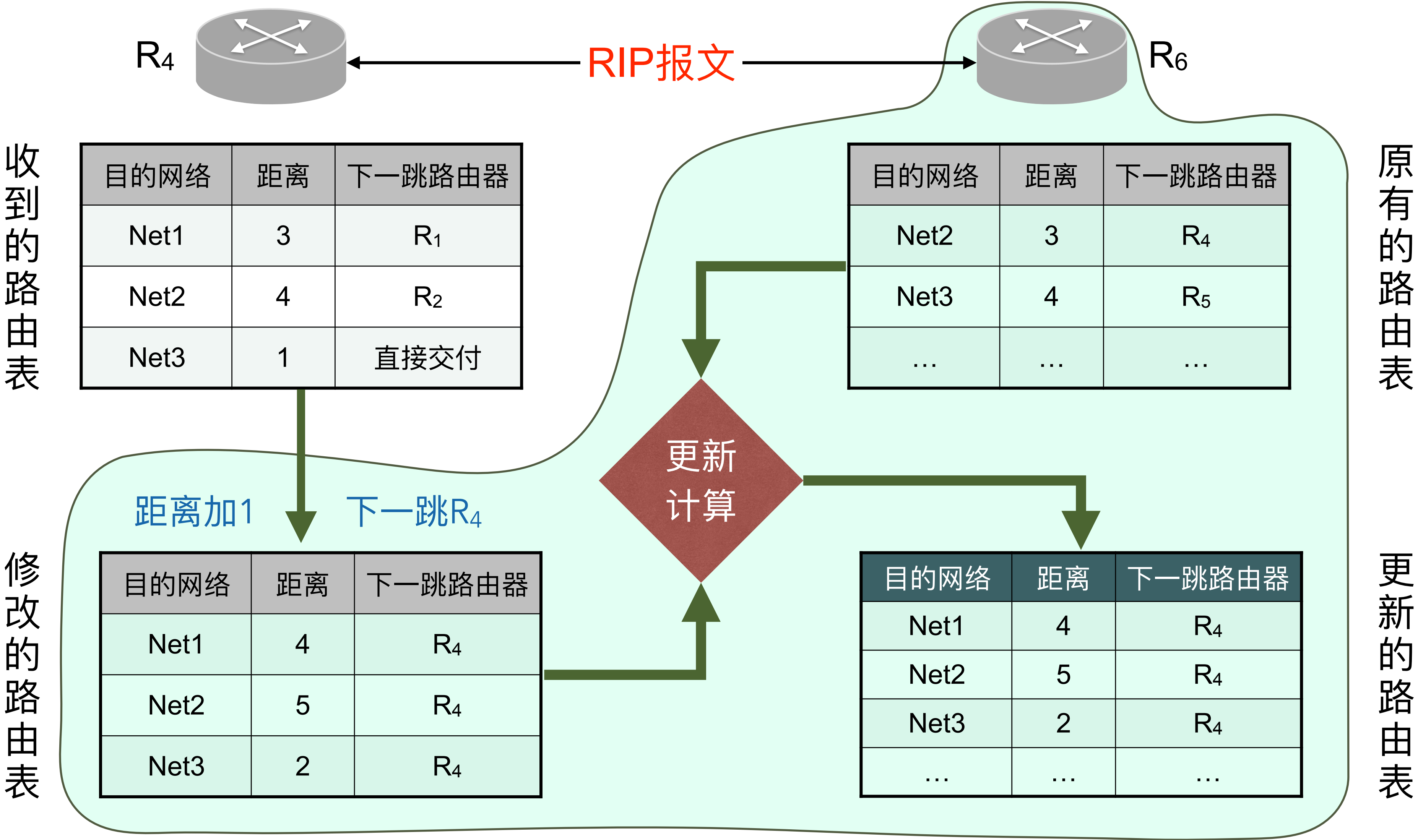
- 网络层
 - RIP路由选择协议概述
 - RIP的特点
 - 距离向量算法
 - RIP的优缺点

- 距离向量算法的基础就是 **Bellman-Ford 算法**（或 Ford-Fulkerson 算法）。这种算法的要点是这样的：
 - 设X是结点 A 到 B 的最短路径上的一个结点；
 - 若把路径 $A \rightarrow B$ 拆成两段路径 $A \rightarrow X$ 和 $X \rightarrow B$ ，则每一段路径 $A \rightarrow X$ 和 $X \rightarrow B$ 也都分别是结点 A 到 X 和结点 X 到 B 的最短路径。

距离向量算法



距离向量算法

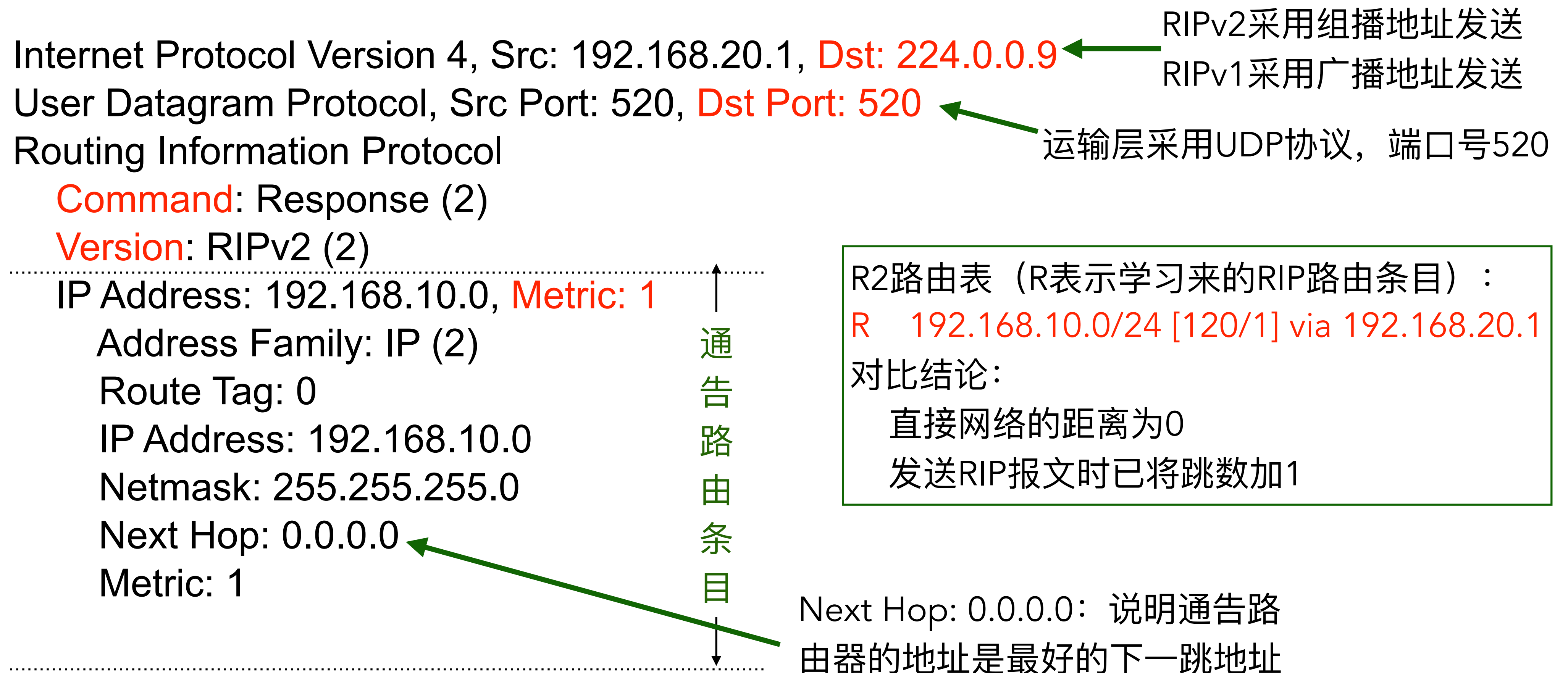


RIPv2报文格式

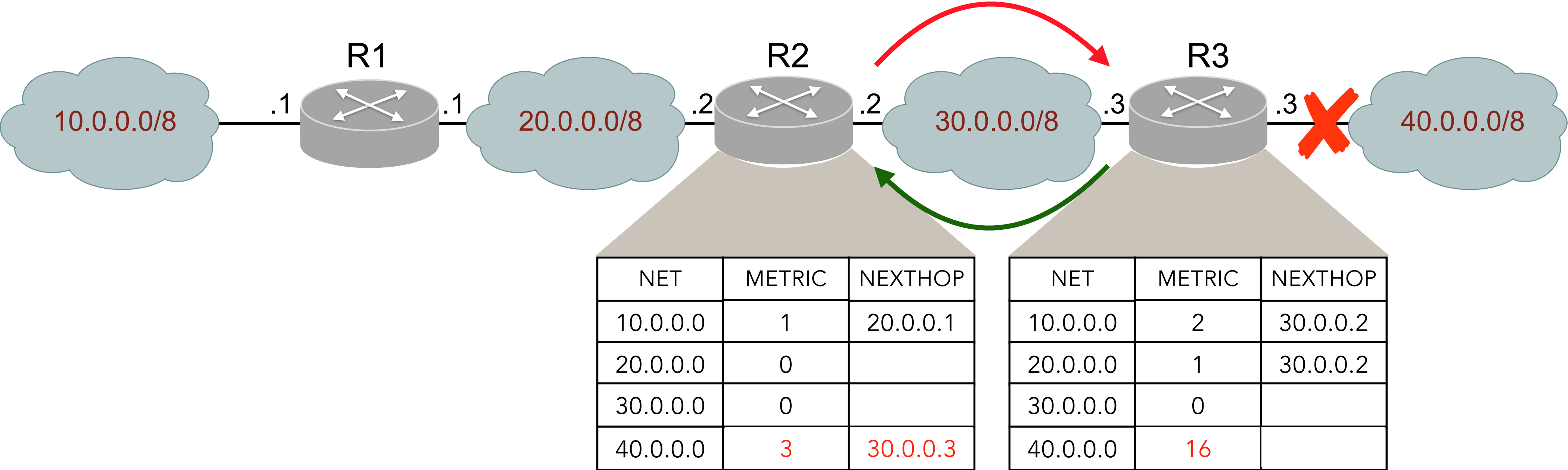
- 命令：1为请求，2为响应。
- 版本：对于RIPv2，值为2。
- 地址簇标识：又称地址类别，标志使用的地址协议，IP协议该值为2。
- 路由标记：自治系统号。
- 网络地址：路由条目的目的地址。
- 子网掩码：确定网络和子网部分的32掩码。
- 下一跳路由器地址：去往目的网络的下一跳路由。
- 距离：1~16之间的跳数，16为目的网络不可达。
- RIP最大报文长度：4 + 25 × 20 = 504字节。
- 如果使用鉴别功能：第 1 条路由用作鉴别最多通告24条路由。
- RIP封装到UDP中，端口520。



RIPv2实例



好消息传播得快，坏消息传播得慢



RIP协议特点：好消息传播得快，坏消息传播得慢。当网络出现故障时，要经过比较长的时间 (例如数分钟) 才能将此信息传送到所有的路由器。

RIP 协议的优缺点

- 网络层
- RIP路由选择协议概述
- RIP的特点
- 距离向量算法
- **RIP的优缺点**

- **优点：**
 - 实现简单，开销较小。
- **缺点：**
 - RIP 限制了网络的规模，它能使用的最大距离为 15（16 表示不可达）；
 - 路由器之间交换的是完整路由表，因而随着网络规模的扩大，开销增加；
 - “坏消息传播得慢”，使更新过程的收敛时间过长。

小结

- 网络层
- RIP路由选择协议概述
- RIP的特点
- 距离向量算法
- RIP的优缺点

- 距离向量算法。
- 距离的定义。
- RIP报文格式。

RIP的特点：

- 实现简单、开销小；
- 仅和邻居交换信息、交换自己的路由表；
- 固定时间交换（拓扑变化时），适用于小规模网络；
- 好消息传播得快，坏消息传播得慢。

内部网关协议 OSPF

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - OSPF基本操作
 - 指定路由器

- 开放最短路径优先 OSPF (Open Shortest Path First)是为克服RIP 的缺点在1989年开发出来的。OSPF 的原理很简单，但实现起来却较复杂。
- 特点：
 - “开放”表明 OSPF 协议不是受某一家厂商控制，而是公开发表的；
 - “最短路径优先”是因为使用了 Dijkstra 提出的最短路径算法 SPF
 - 采用分布式的链路状态协议 (link state protocol)；
 - OSPF 一个协议的名字，并不表示其他的路由选择协议不是“最短路径优先”。

OSPF的三个要点

- 网络层
- OSPF路由选择协议概述
- 三个要点
- 链路状态数据库
- 区域的概念
- OSPF的特点
- OSPF报文格式
- OSPF基本操作
- 指定路由器

- **和谁交换信息**：使用洪泛法向本自治系统中所有路由器发送信息。
- **交换什么信息**：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, 费用、距离、时延、带宽等)。
- **何时交换信息**：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。

链路状态数据库 (link-state database)

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - OSPF基本操作
 - 指定路由器

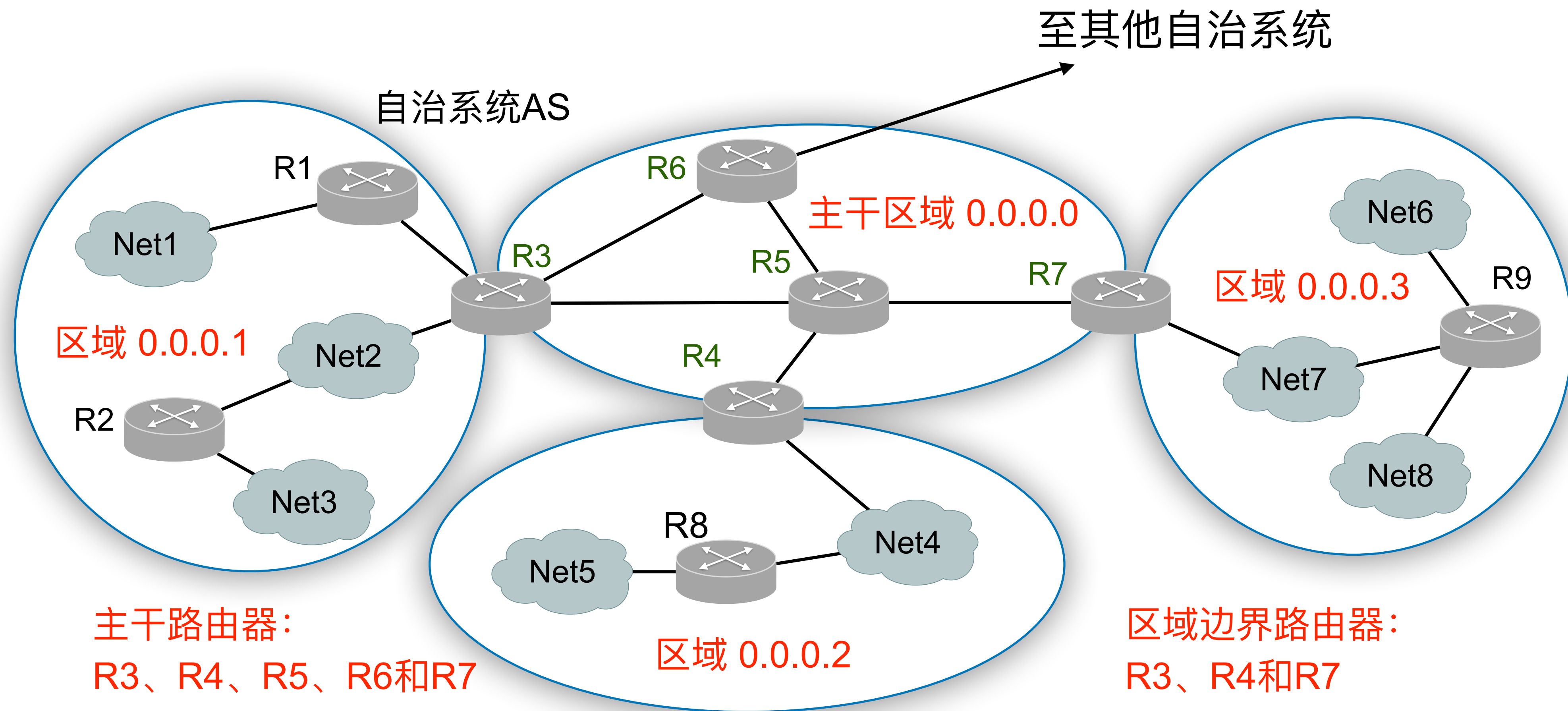
- 所有的路由器最终都能建立一个链路状态数据库。这个数据库实际上就是全网的拓扑结构图，它在全网范围内是一致的（这称为链路状态数据库的同步）。
- OSPF 的链路状态数据库能较快地进行更新，使各个路由器能及时更新其路由表。
- OSPF 的更新过程收敛得快是其重要优点。

各国向其他邻居国家“洪泛”本国的地图，最终各国都有一张一样“世界地图”，各国一样的链路状态数据库。

OSPF 的区域 (area)

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - OSPF基本操作
 - 指定路由器
- 为了使 OSPF 能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫做区域。
 - 每一个区域都有一个 32 位的区域标识符（用点分十进制表示）。
 - 区域也不能太大，在一个区域内的路由器最好不超过 200 个。
 - 为什么要划分区域：
 - 将洪泛法交换链路状态信息的范围局限于每一个区域，减少了网络上的通信量；
 - 同一区域内的路由器只知道本区域的完整网络拓扑，不知道其他区域的网络拓扑；
 - 上层的区域叫作主干区域，其标识符规定为0.0.0.0，主干区域的作用是用来连通其他区域的。

OSPF 划分为两种不同的区域



OSPF 直接用 IP 数据报传送

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - **OSPF的特点**
 - OSPF报文格式
 - OSPF基本操作
 - 指定路由器

- OSPF不用UDP而是**直接用 IP 数据报传送**。
- OSPF构成的数据报很短。可**减少路由信息的通信量**。
- 数据报短的另一好处是可以**不必**将长的数据报**分片传送**。
- 但分片传送的数据报只要丢失一个，就无法组装成原来的数据报，而整个数据报就必须重传。

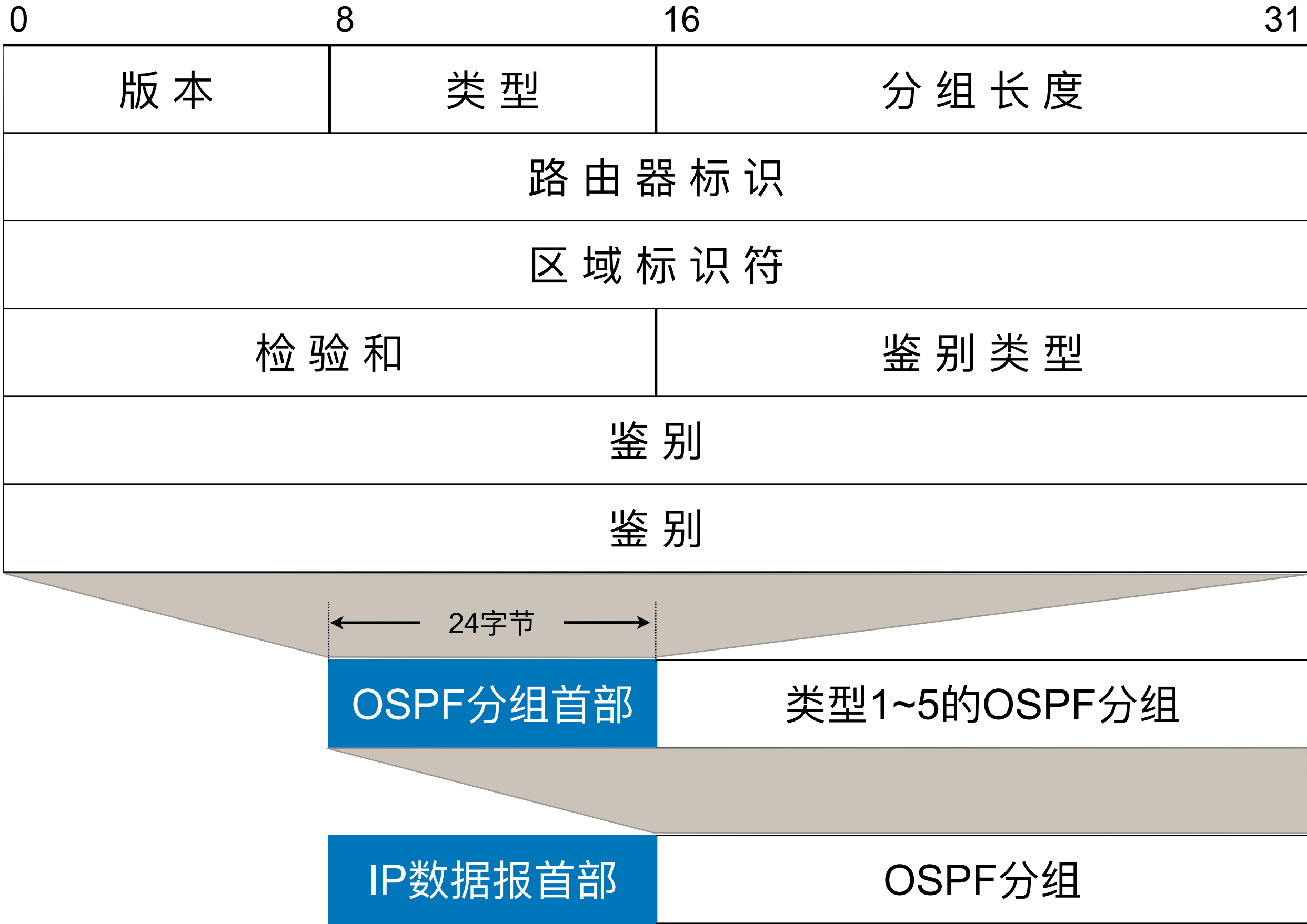
OSPF 的特点

- 网络层
- OSPF路由选择协议概述
- 三个要点
- 链路状态数据库
- 区域的概念
- **OSPF的特点**
- OSPF报文格式
- OSPF基本操作
- 指定路由器

- OSPF 对不同的链路可根据 IP 分组的不同服务类型 TOS 而设置成不同的代价，**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**负载均衡**。
- 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。
- **与网络规模无关**：由于路由器的链路状态只涉及到与相邻路由器的连通状态，因而与整个互联网的规模并无直接关系。当互联网规模很大时，**OSPF协议要比距离向量协议RIP好得多**。

OSPF 的报文格式

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - **OSPF报文格式**
 - OSPF基本操作
 - 指定路由器



OSPF 的报文格式

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - **OSPF报文格式**
 - OSPF基本操作
 - 指定路由器

字段	长度 (bit)	作用
版本	8	OSPFv2适用于IPv4，OSPFv3适用于IPv6
类型	8	1为Hello，2为DD，3为LSR，4为LSU，5为LSACK
分组长度	16	OSPF分组总长度
路由器标识符	32	始发LSA的路由器的ID
区域标识符	32	始发LSA的路由器所在的区域ID
检验和	16	对整个报文的检验和
鉴别类型	16	认证字段：0为不认证，1为简单明文，2为MD5
鉴别	64	验证信息：0没有，1为明文密码，2为key id

LSA: Link-State Advertisement (链路状态通告)

OSPF 的五种分组类型（链路状态路由算法）

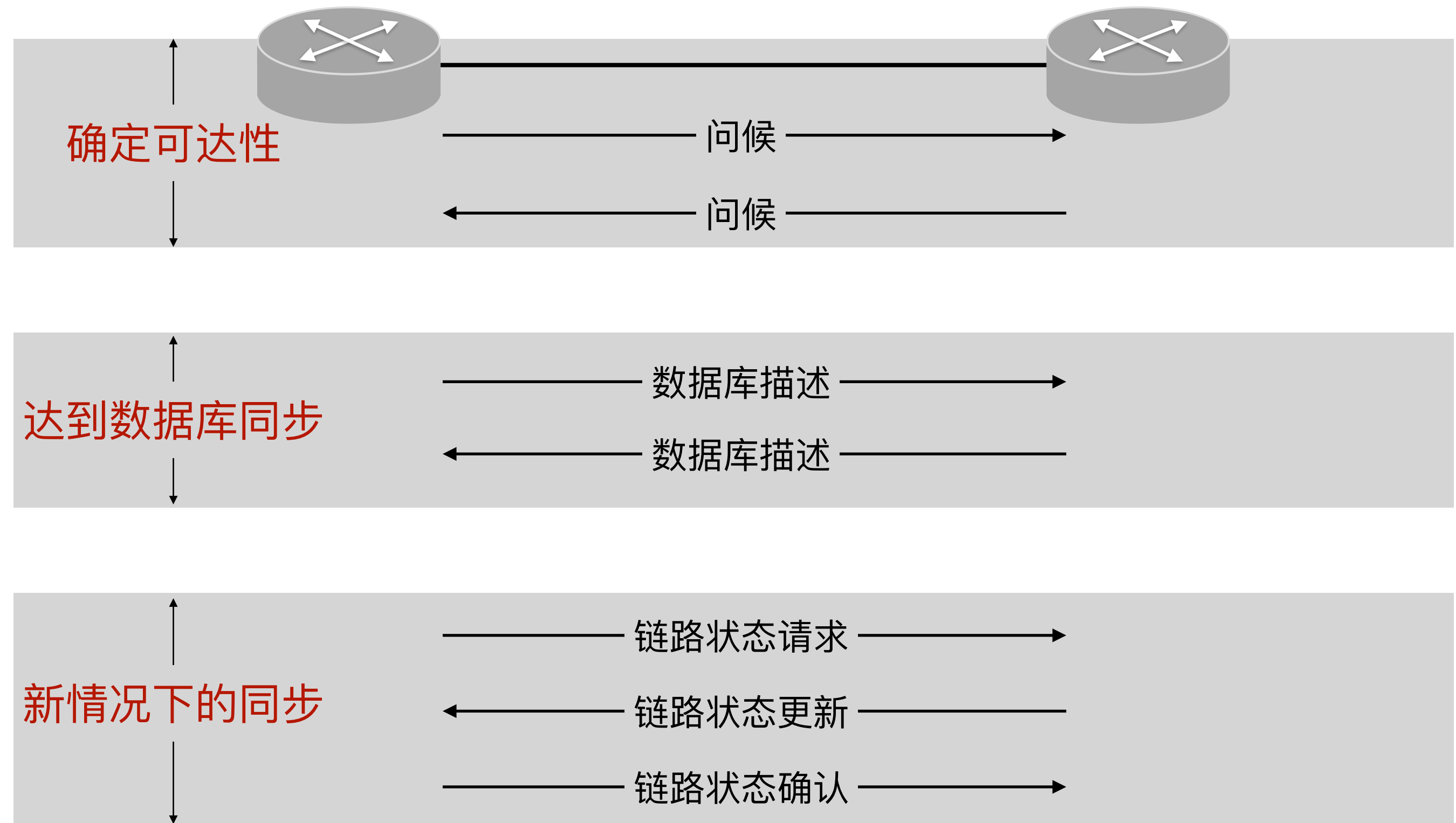
- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - **OSPF报文格式**
 - OSPF基本操作
 - 指定路由器

类型	含义	作用
1	问候 (Hello)	发现邻居结点，设置到每个邻居的metric
2	数据库描述 (Database Description)	向邻居发送链路状态摘要信息
3	链路状态请求 (Link State Request)	向邻居请求自己没有的链路状态信息
4	链路状态更新 (Link State Update)	用洪泛法对全网更新链路状态
5	链路状态确认 (Link State Acknowledgment)	向第1个发送更新的路由器发送确认

使用Dijkstra算法，根据自己的链路状态数据库构造到其他结点的最短路径。

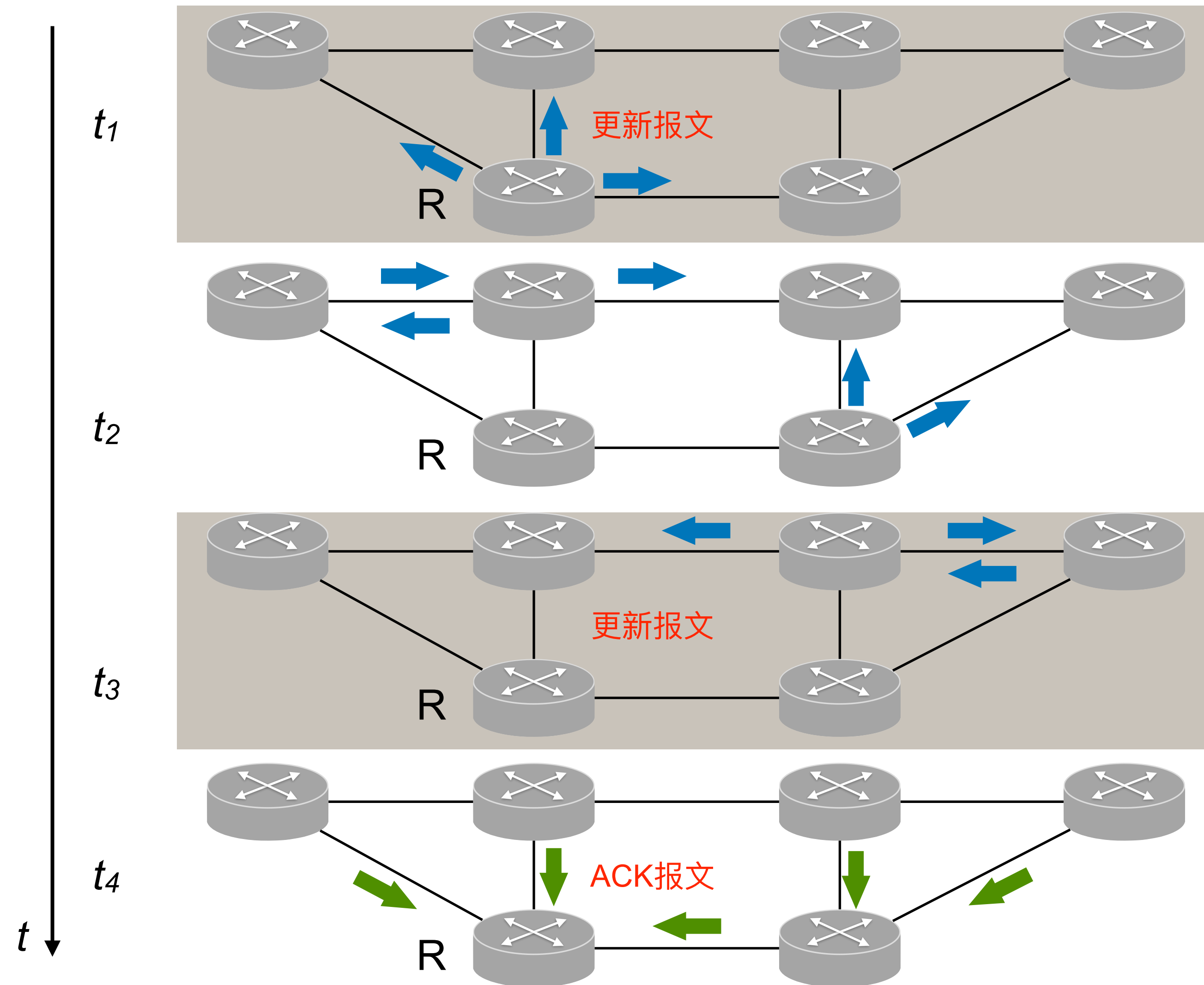
OSPF 的基本操作

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - **OSPF基本操作**
 - 指定路由器



OSPF 的基本操作

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - **OSPF基本操作**
 - 指定路由器



- OSPF 使用**可靠的洪泛法**发送更新分组。
- 向最早向其发送LSA的路由器发送确认。

OSPF 数据库刷新时间

- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - **OSPF基本操作**
 - 指定路由器

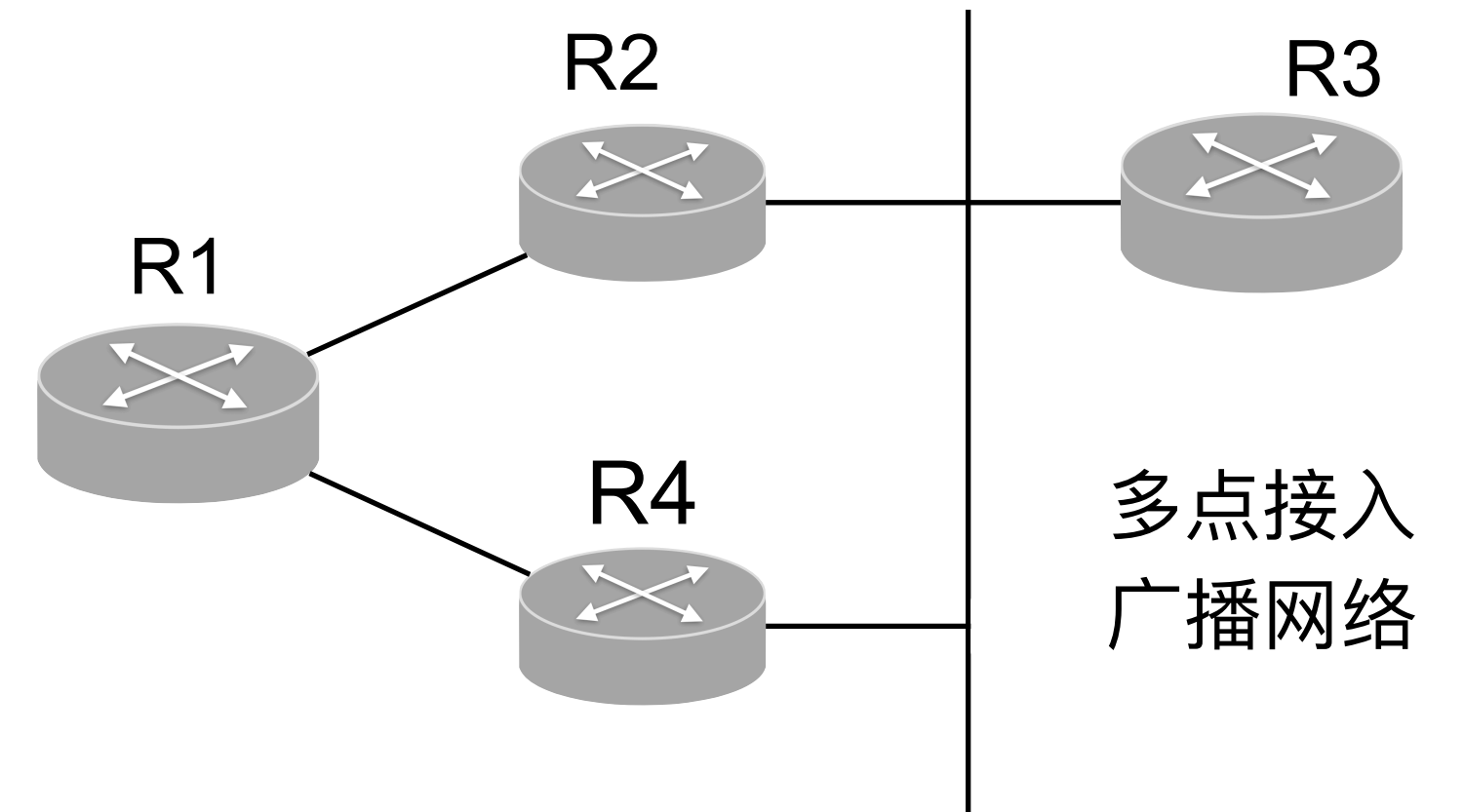
- OSPF 规定**每隔一段时间**，如 30 分钟，要刷新一次数据库中的链路状态。
- 由于一个路由器的链路状态只涉及到与相邻路由器的连通状态，因而与整个互联网的规模并无直接关系。因此当互联网规模很大时，OSPF 协议要比距离向量协议 RIP 好得多。
- OSPF **没有“坏消息传播得慢”**的问题，据统计，其响应网络变化的时间小于 100 ms。

指定路由器

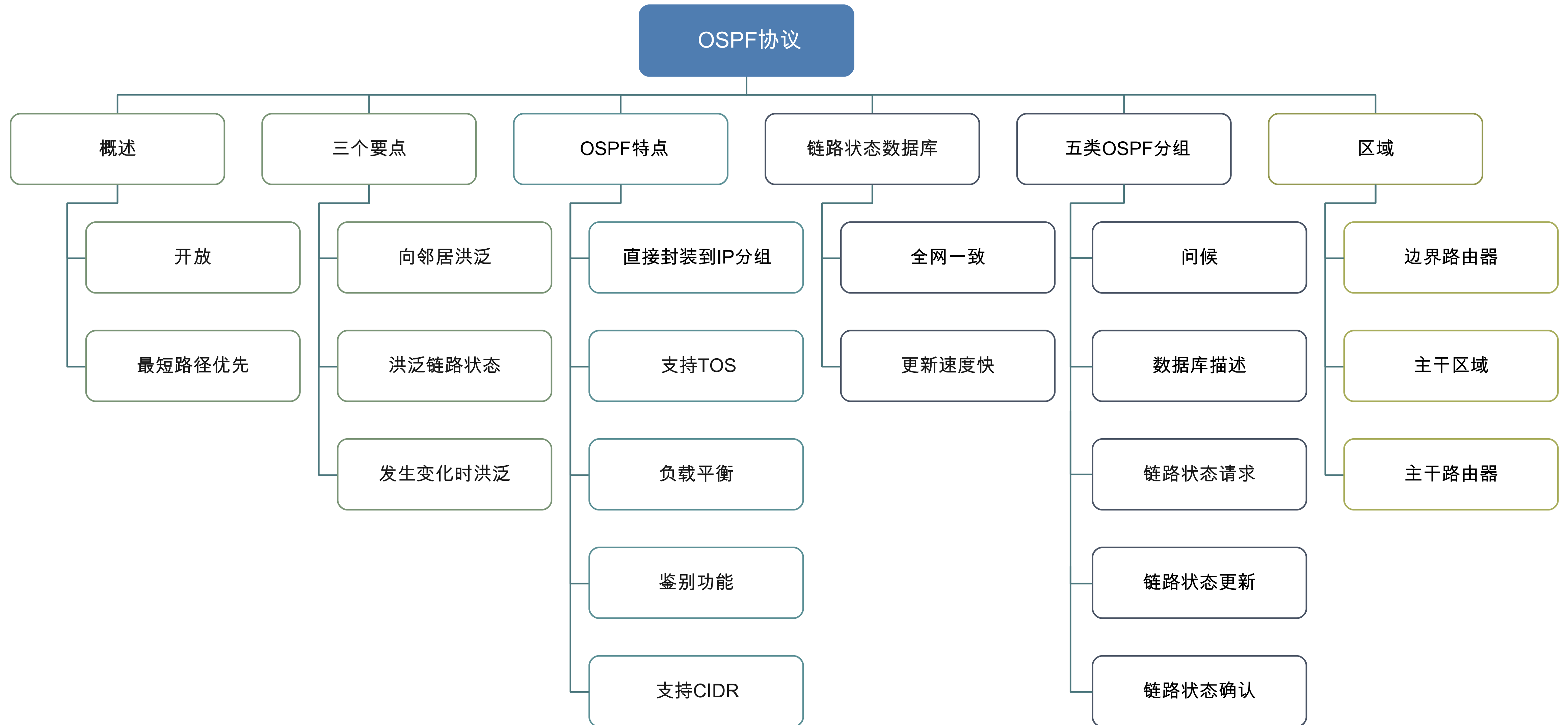
- 网络层
 - OSPF路由选择协议概述
 - 三个要点
 - 链路状态数据库
 - 区域的概念
 - OSPF的特点
 - OSPF报文格式
 - OSPF基本操作
 - 指定路由器

- 多点接入的局域网采用了**指定的路由器** (designated router) 的方法, 使广播的**信息量大大减少**。
- 指定的路由器**代表该局域网上所有的链路**向连接到该网络上的各路由器发送状态信息。

R1、R2、R3、R4运行OSPF路由选择协议, 且在同一OSPF区域, R2、R3、R4**多点接入属于同一广播域中**, 它们选出一个DR, 发送**链路状态信息**。



小结



外部网关协议 BGP

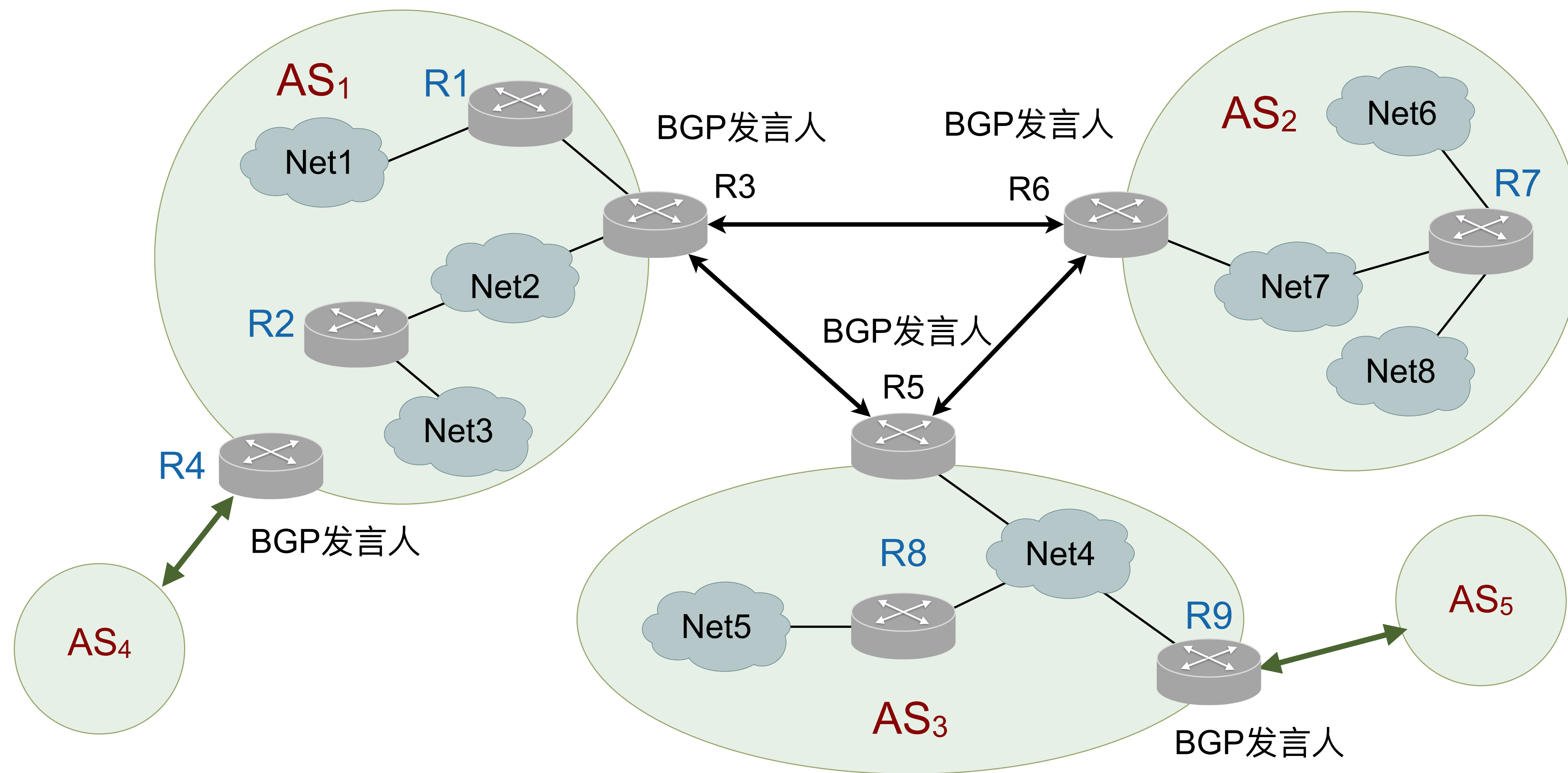
- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析
- BGP 是不同自治系统的路由器之间交换路由信息的协议。
- 互联网的规模太大，使得自治系统之间路由选择非常困难。对于自治系统之间的路由选择，要寻找最佳路由是很不现实的：
 - 当一条路径通过几个不同 AS 时，要想对这样的路径计算出有意义的代价是不太可能的；
 - 比较合理的做法是在 AS 之间交换“可达性”信息。
- 自治系统之间的路由选择必须考虑有关策略：
 - BGP 寻找一条能够到达目的网络且比较好的路由（不能兜圈子），不是寻找一条最佳路由。

BGP发言人信息交换

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析
- 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**” (BGP speaker) 。
- 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。
- **和谁交换信息**：与邻居AS的BGP发言人交换信息。
- **交换什么信息**：交换的网络可达性的信息，即到达某个网络所要经过的一系列 AS。
- **什么时候交换**：网络拓扑发生变化时，更新有变化的部分。

BGP 发言人和自治系统AS的关系

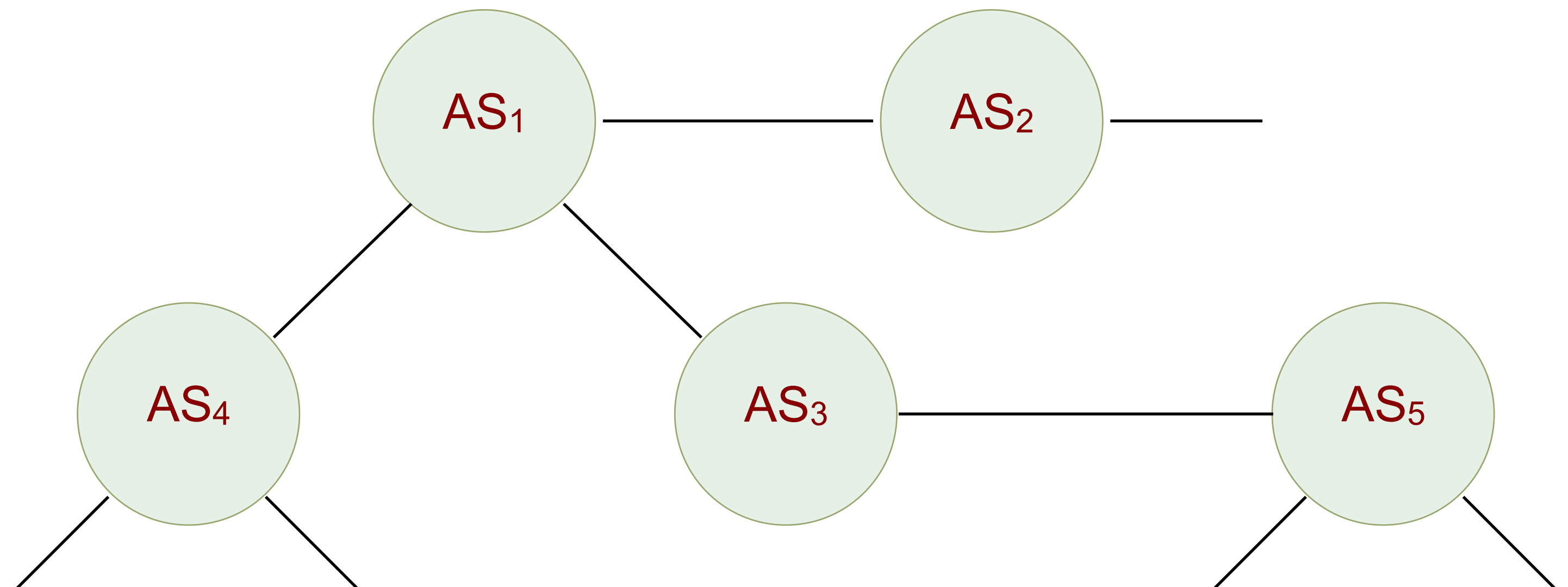
北京 → 石家庄 → 郑州 → 武汉 → 长沙 → 广州



BGP协议交换信息的过程

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

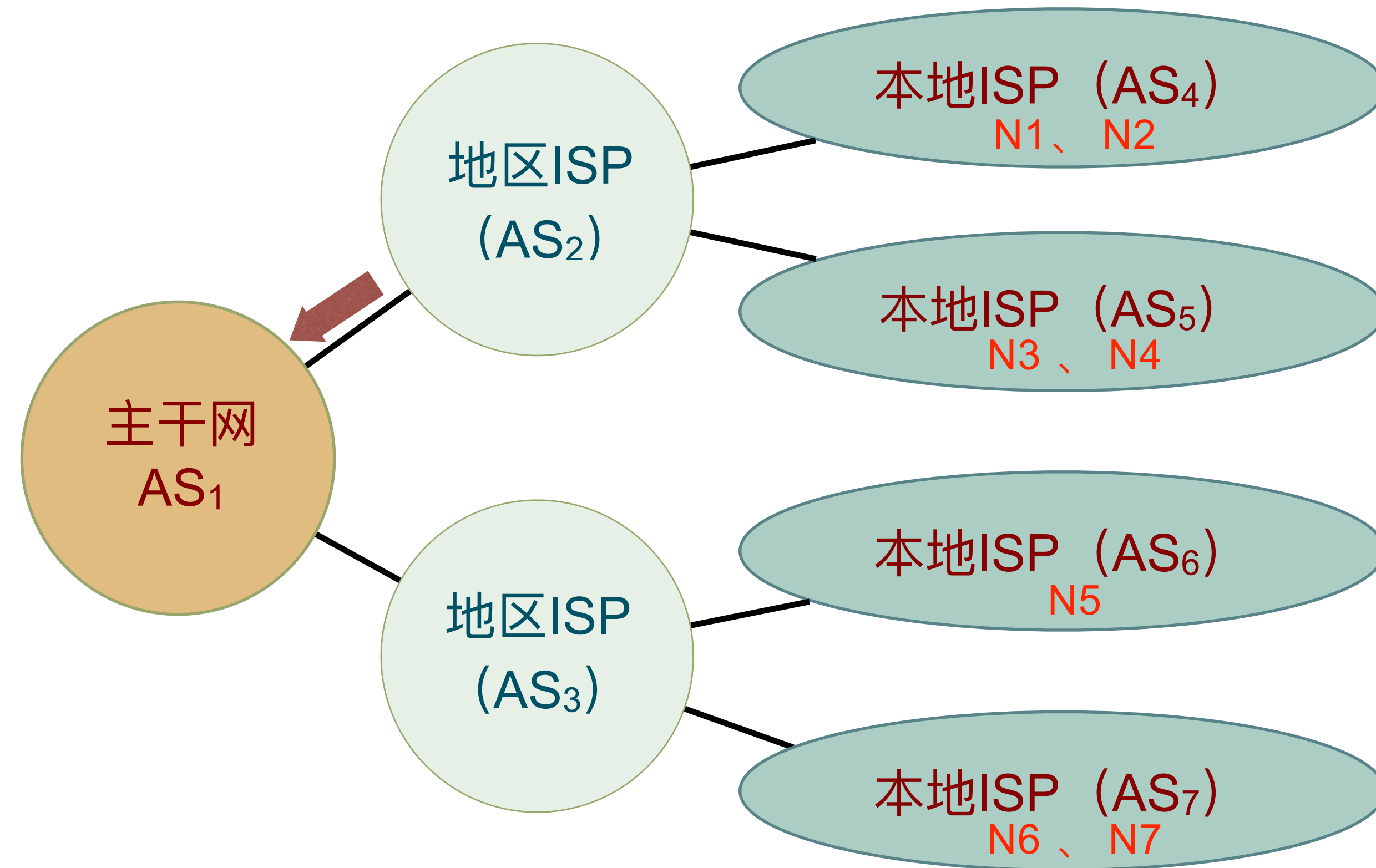
- BGP 所交换的网络可达性的信息就是要到达某个网络所要经过的一系列 AS。
- 当 BGP 发言人互相交换了网络可达性的信息后，各 BGP 发言人就根据所采用的策略从收到的路由信息中找出到达各 AS 的较好路由。



BGP 发言人交换路径向量

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

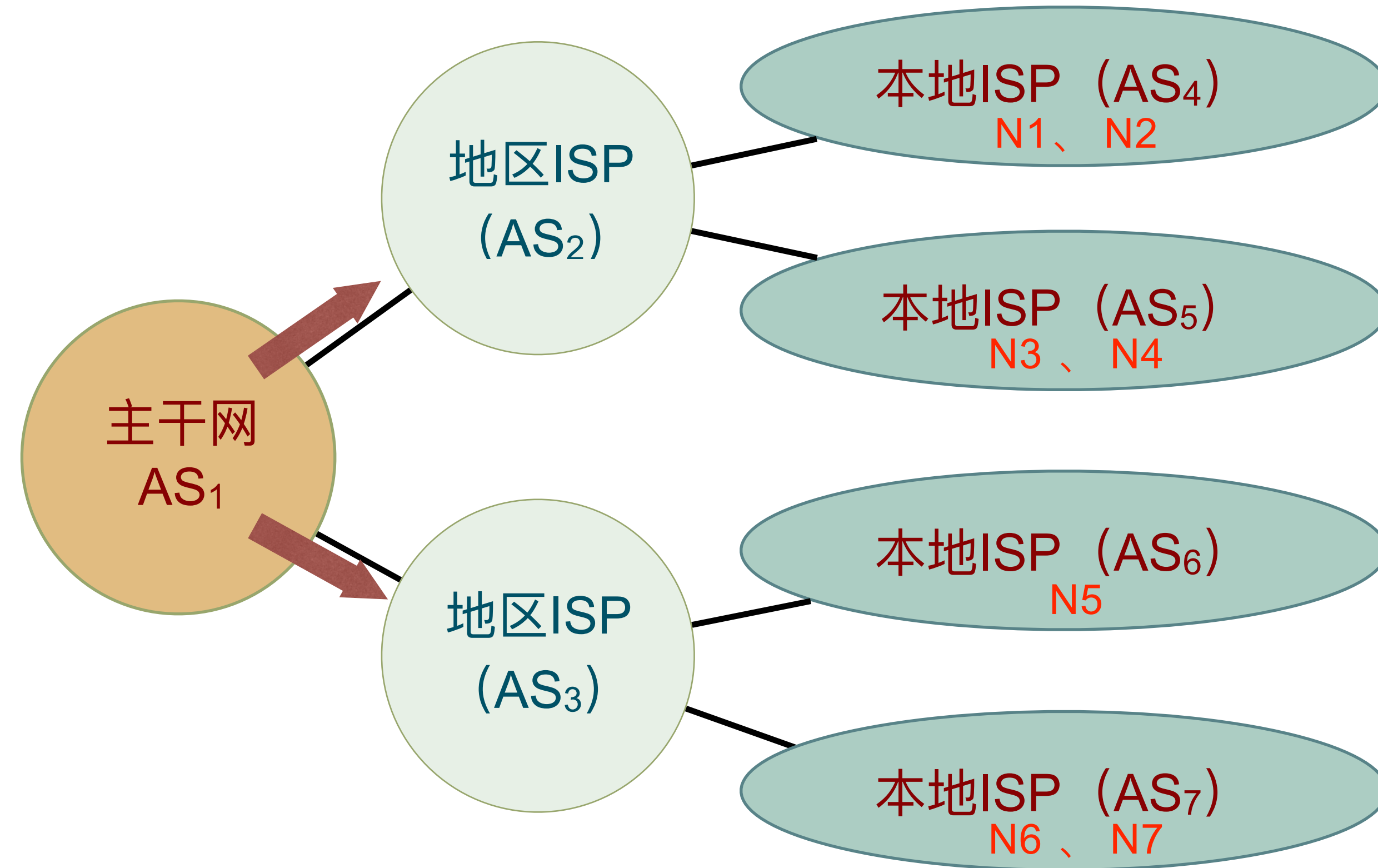
- 自治系统 AS2 的 BGP 发言人通知主干网 AS1 的 BGP 发言人：
 - “要到达网络 N1、N2、N3 和 N4 可经过 AS2。”



BGP 发言人交换路径向量

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

- 主干网还可发出通知：
 - 要到达网络 N5、N6 和 N7，可沿路径 (AS1, AS3)。



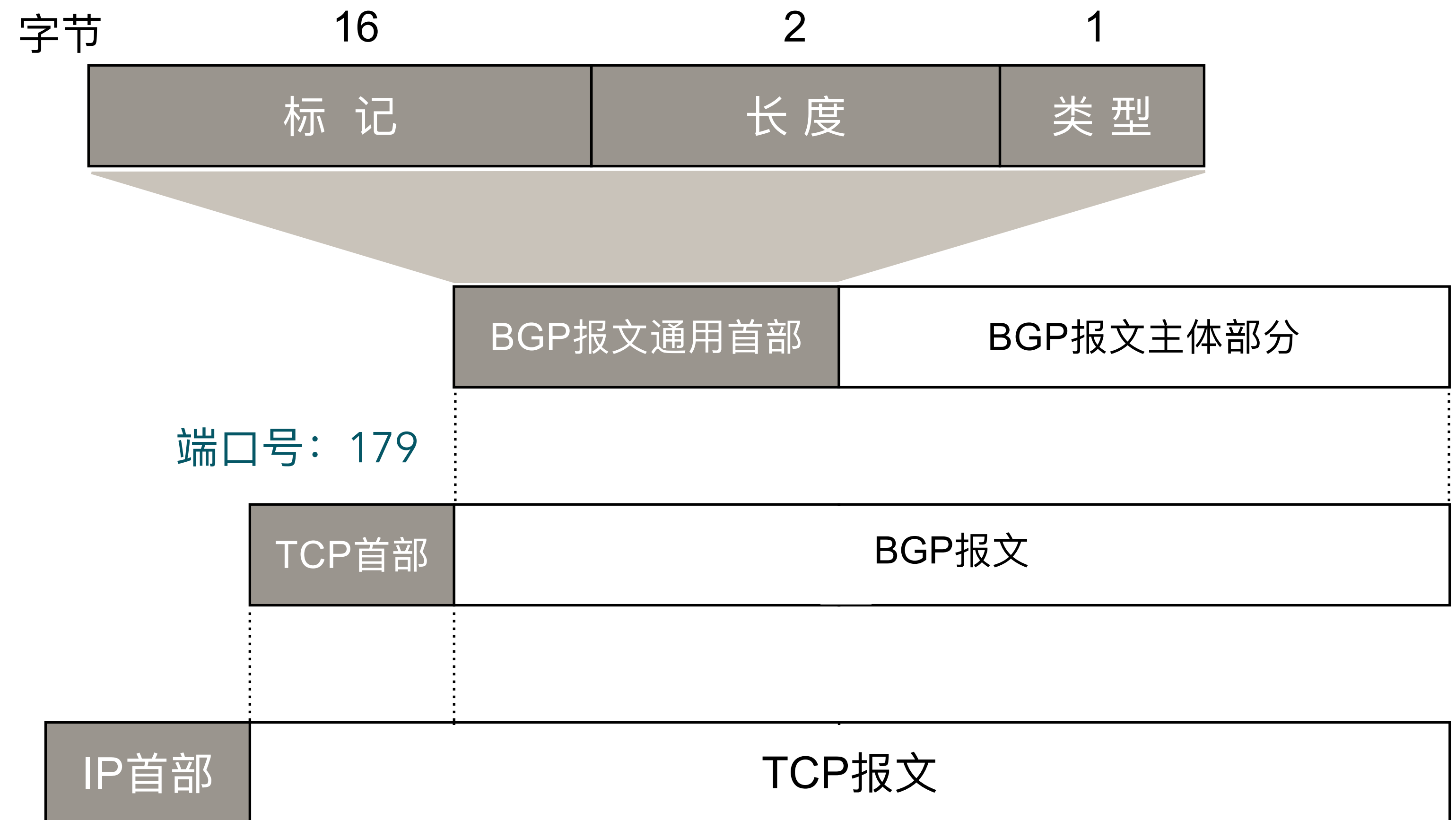
BGP 协议的特点

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - **BGP协议的特点**
 - BGP报文格式
 - BGP实例分析

- 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。
 - 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**。

BGP 报文格式

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - **BGP报文格式**
 - BGP实例分析



通用首部

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

- **标记**：用于检查BGP对等体的同步信息是否完整，以及用于BGP验证的计算（用于鉴别）。
- **长度**：整个BGP报文的长度，单位字节最小19，最大4096。
- **类型**：1~4，分别对应四种类型的BGP报文。



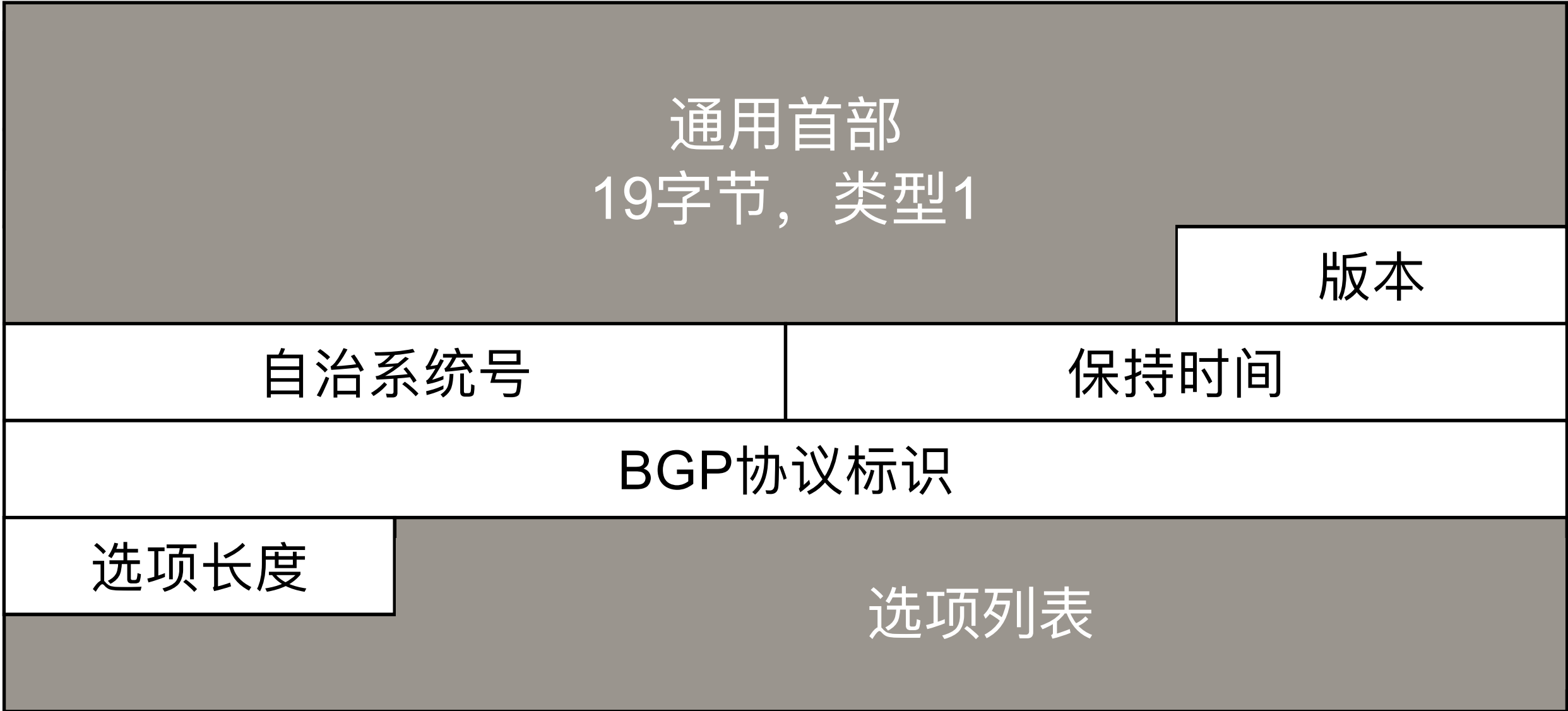
BGP-4 共使用五种报文

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析
- 打开 (OPEN) 报文：用来与相邻的另一个BGP发言人建立关系。
- 更新 (UPDATE) 报文：用来发送某一路由的信息，以及列出要撤销的多条路由。
- 保活 (KEEPALIVE) 报文：用来确认打开报文和周期性地证实邻站关系。
- 通知 (NOTIFICATION) 报文：用来发送检测到的差错。
- 以上四种报文在RFC4271中定义。
- 刷新(REFRESH)，刷新报文：在RFC2918中定义。

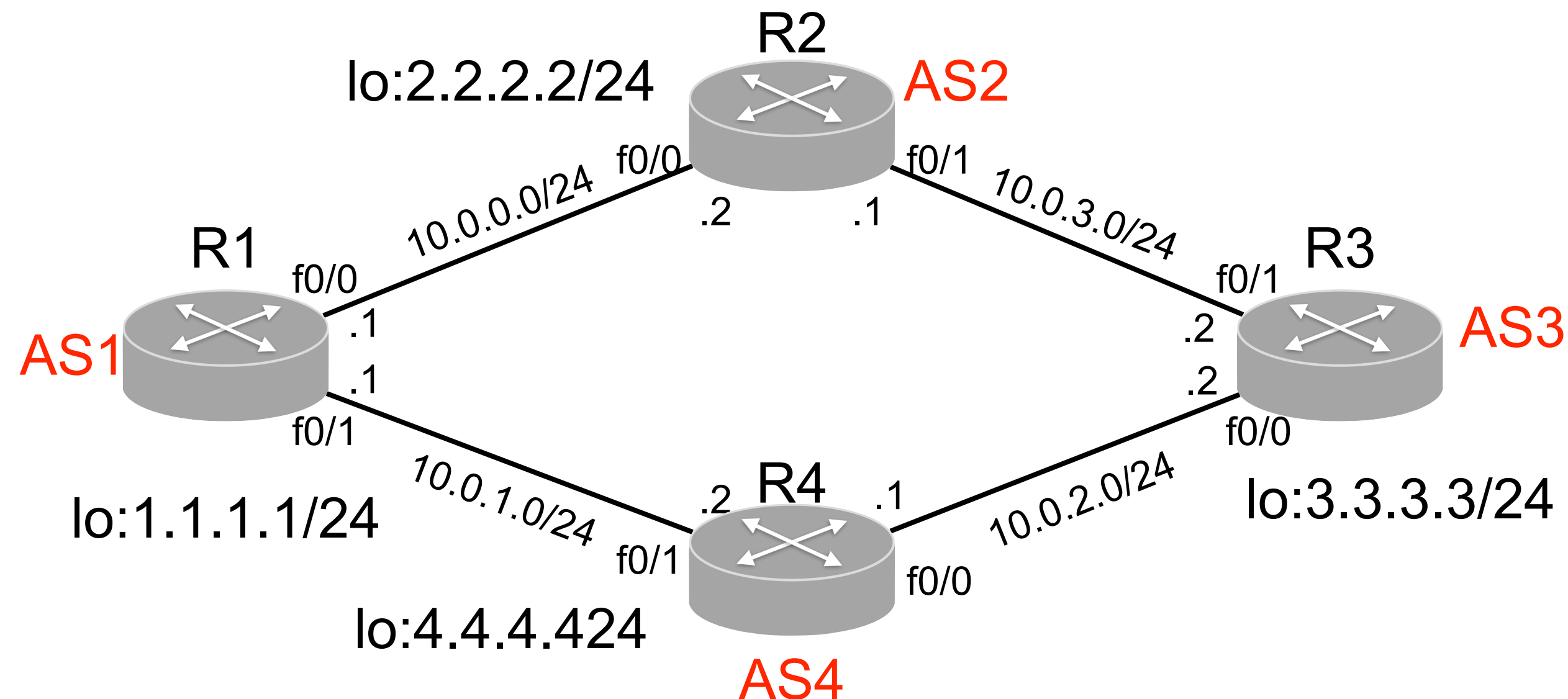
例如：BGP的OPEN报文，类型为1

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

- 版本：协议版本号，现BGP版本为4。
- 自治系统号：自己的AS号。
- 保活时间：设置自己的Hold Time时间，默认180秒。
- BGP协议标识：发送者路由器ID。
- 选项长度：可选项长度。
- 选项列表：可选参数列表。



BGP运行实例



R1(config)#int lo0 ← #配置环回接口, 模拟一个网络

R1(config-if)#ip address 1.1.1.1 255.255.255.0

R1(config-if)#int f0/0 ← #配置f0/0接口

R1(config-if)#ip add 10.0.0.1 255.255.255.0

R1(config-if)#no shut

R1(config-if)#int f0/1

R1(config-if)#ip add 10.0.1.1 255.255.255.0

R1(config-if)#no shut

R1(config)#router bgp 1 ← #开启BGP进程, AS为1

R1(config-router)#network 10.0.0.0 mask 255.255.255.0 ← #宣告网络到BGP进程

R1(config-router)#network 10.0.1.0 mask 255.255.255.0

R1(config-router)#network 1.1.1.0 mask 255.255.255.0

R1(config-router)#neighbor 10.0.0.2 remote-as 2 ← #手动配置R2为邻居

R1(config-router)#neighbor 10.0.1.2 remote-as 4

BGP的OPEN（打开消息）

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
Transmission Control Protocol, Src Port: 43254, Dst Port: 179, Seq: 1, Ack: 1, Len: 45
Border Gateway Protocol - OPEN Message

通用首部

Marker: ffffffffffffffffffffffff (标记)
Length: 45 (长度)
Type: OPEN Message (1) (类型)

打开消息

Version: 4 (版本)
My AS: 1 (自治系统号)
Hold Time: 180 (保活时间)
BGP Identifier: 1.1.1.1 (BGP协议标识)
Optional Parameters Length: 16 (选项长度)
Optional Parameters (可选项列表)
Optional Parameter: Capability
Optional Parameter: Capability
Optional Parameter: Capability

BGP的UPDATE

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1

Transmission Control Protocol, Src Port: 179, Dst Port: 25745, Seq: 226, Ack: 139, Len: 51

Border Gateway Protocol - UPDATE Message

Marker: ffffffffffffffffffffffffffffffff

Length: 51

Type: UPDATE Message (2)

Withdrawn Routes Length: 0 (要撤销的路由列表)

Total Path Attribute Length: 20 (更新的路由属性列表总长度)

Path attributes (要更新的路由属性列表)

Path Attribute - ORIGIN: IGP

Path Attribute - AS_PATH: 2 3

Flags: 0x40, Transitive, Well-known, Complete

Type Code: AS_PATH (2)

Length: 6

AS Path segment: 2 3

Path Attribute - NEXT_HOP: 10.0.0.2

Network Layer Reachability Information (NLRI)

BGP的KEEPALIVE

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析

保活报文只有通用首部：

Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2

Transmission Control Protocol, Src Port: 25745, Dst Port: 179, Seq: 139, Ack: 277, Len: 19

Border Gateway Protocol - KEEPALIVE Message

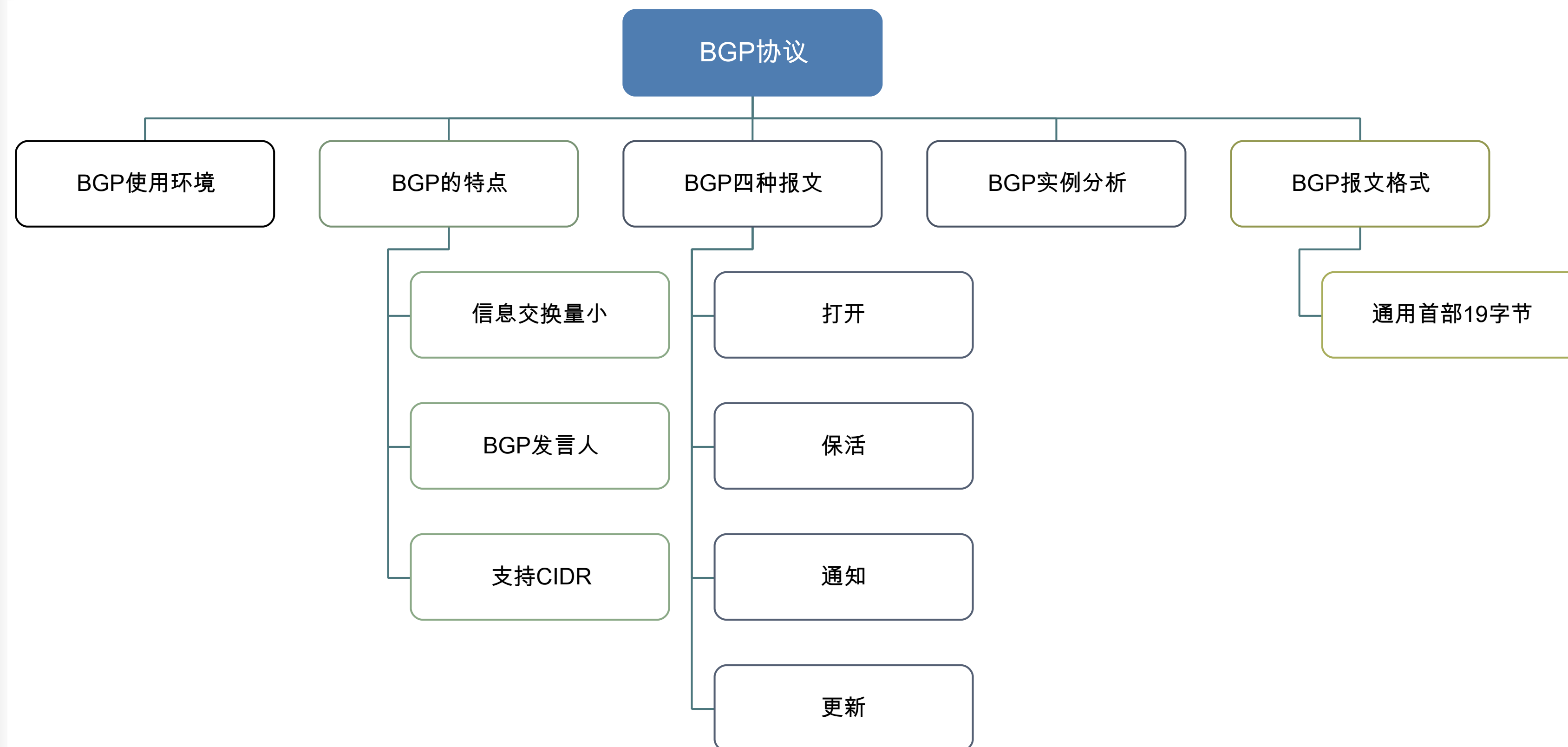
Marker: ffffffffffffffffffffffffffff

Length: 19

Type: KEEPALIVE Message (4)

小结

- 网络层
 - BGP网关协议
 - 发言人信息交换
 - BGP协议的特点
 - BGP报文格式
 - BGP实例分析



路由器简单介绍

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃
- 路由器是互联网中的关键设备：
 - 连通不同的网络。
 - 选择信息传送的线路：选择通畅快捷的“近路”，能大大提高通信速度，减轻网络系统通信负荷，节约网络系统资源，提高网络系统畅通率。
 - 路由器是多个输入端口和多个输出端口的专用计算机，其任务是转发分组（转发给下一跳路由器）。
 - 下一跳路由器也按照这种方法处理分组，直到该分组到达终点为止。

路由器的结构

- 路由器结构分为两部分：

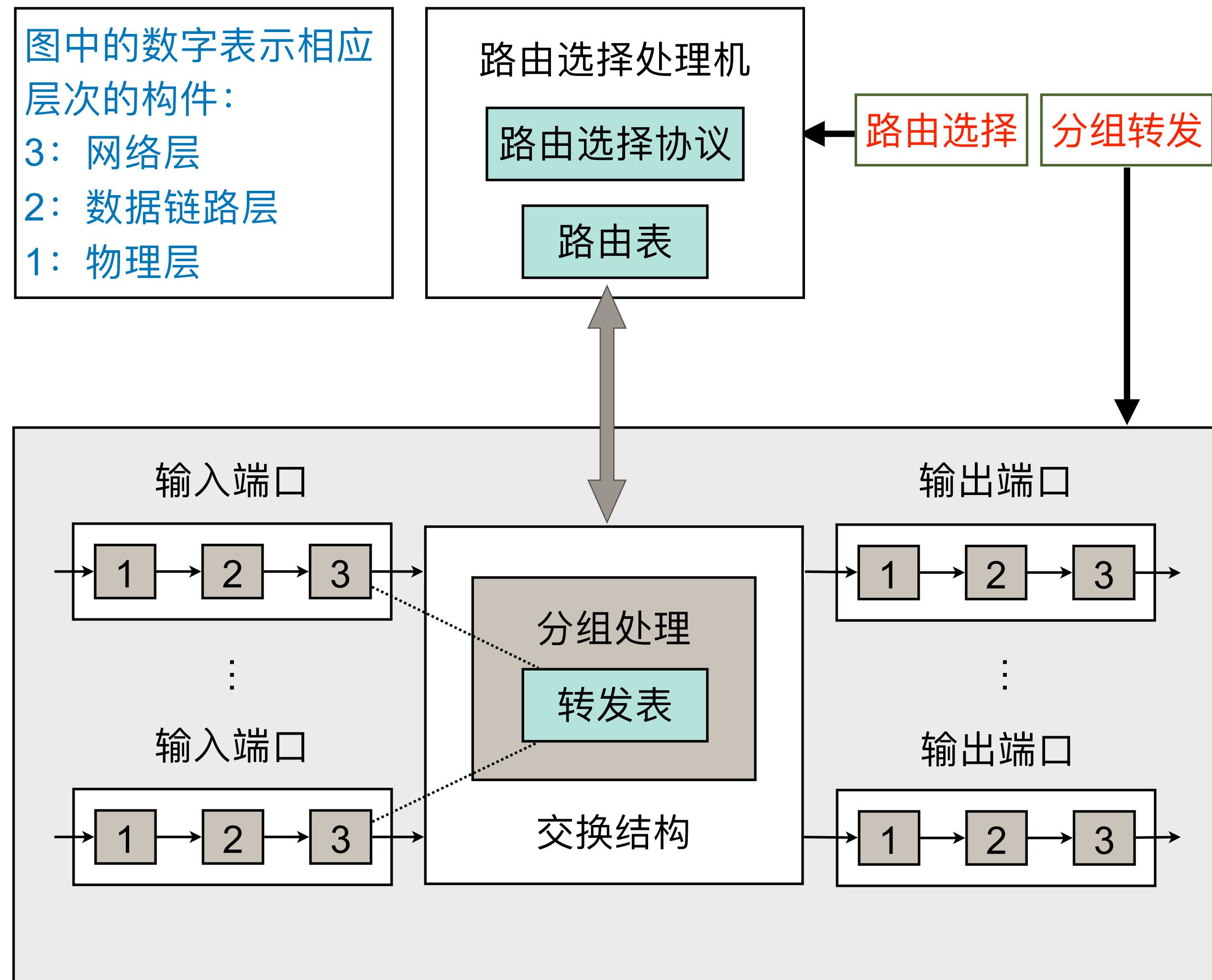
- 路由选择部分；
- 分组转发部分。

- 路由选择部分：

- 也称控制部分，其核心构件是路由选择处理机；
- 任务是根据所路由选择协议构造出路由表，并和相邻路由器交换路由信息，不断地更新和维护路由表。

图中的数字表示相应层次的构件：

3：网络层
2：数据链路层
1：物理层



路由器的结构

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃
- 分组转发部分由三部分组成：
 - 交换结构 (switching fabric): 又称为交换组织, 其作用是根据转发表 (forwarding table) 对分组进行处理;
 - 一组输入端口;
 - 一组输出端口;
 - 请注意: 这里的端口就是硬件接口。

注意转发表与路由表的区别

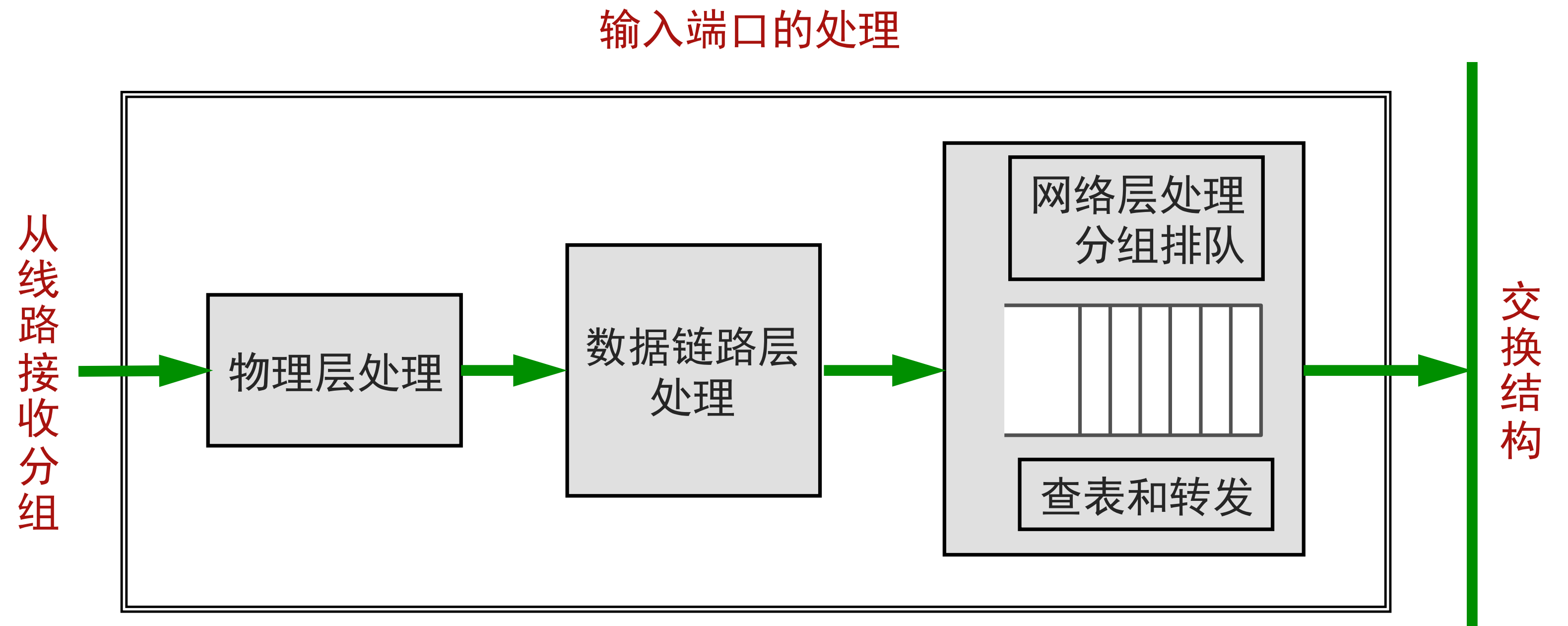
- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃

- “转发”(forwarding) :
 - 路由器根据转发表将用户的 IP 数据报从合适的端口转发出去。
- “路由选择”(routing):
 - 按照分布式算法, 根据从各相邻路由器得到的关于网络拓扑的变化情况, 动态地改变所选择的路由;
 - 路由表是根据路由选择算法得出的。而转发表是从路由表得出的;
 - 在讨论路由选择的原理时, 往往不去区分转发表和路由表的区别。

查找和转发功能在路由器的交换功能中是最重要的。

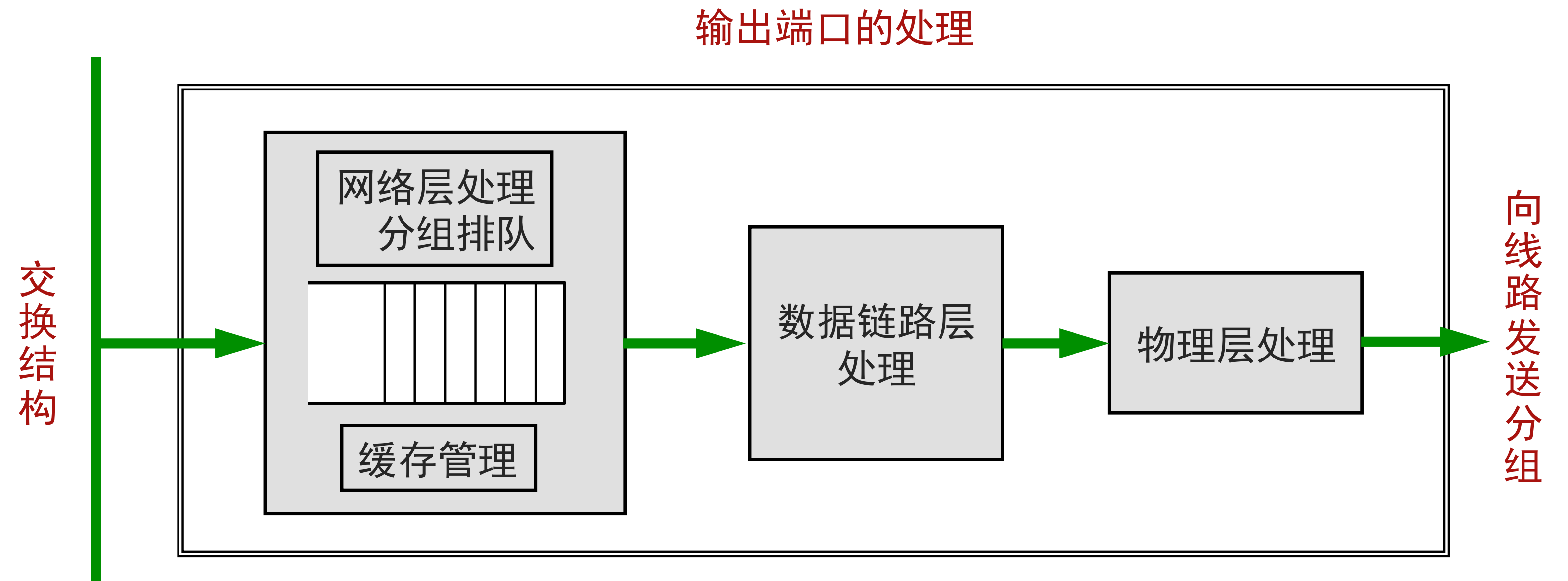
输入端口对线路上收到的分组的处理

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃



输出端口将交换结构传送来的分组发送到线路

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃



分组丢弃

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃

- 若路由器处理分组的速率赶不上分组进入队列的速率，则队列的存储空间最终必定减少到零，这就使后面再进入队列的分组由于没有存储空间而只能被丢弃。
- 路由器中的输入或输出队列产生溢出是造成分组丢失的重要原因。

小结

- 网络层
 - 路由器简介
 - 路由器结构
 - 转发表与路由表
 - 输入端口
 - 输出端口
 - 路由器分组丢弃

