第四章 网络层

刘 轶
北京航空航天大学 计算机学院

本章内容

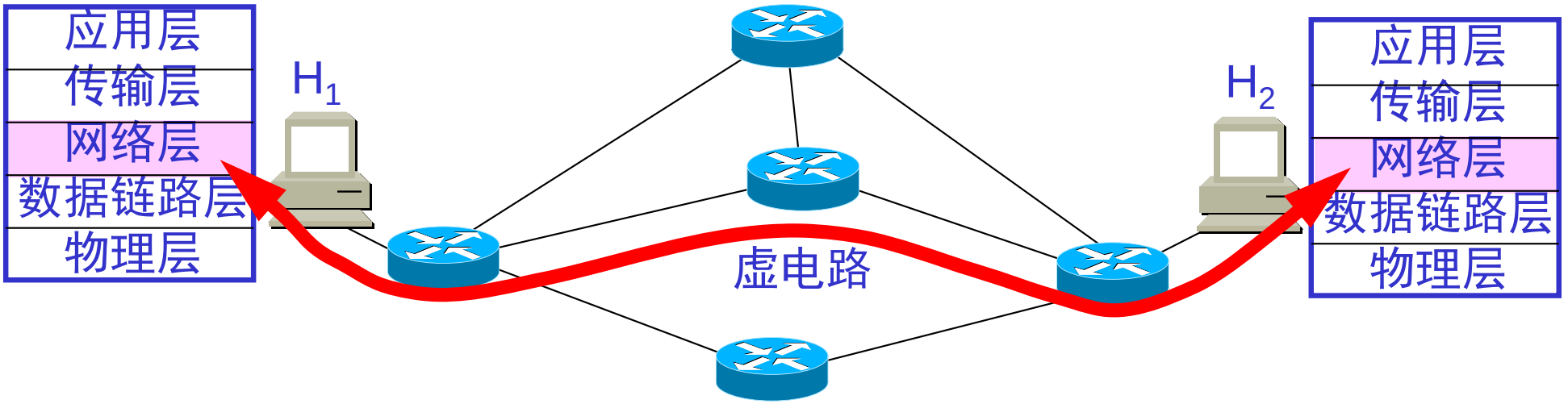
4.1 网络层提供的两种服务
4.2 网际协议IP
4.3 划分子网和构造超网
4.4 网际控制报文协议ICMP
4.5 路由算法及协议
4.6 IP组播
4.7 网络地址转换NAT和虚拟专用网VPN

4.1 网络层提供的两种服务

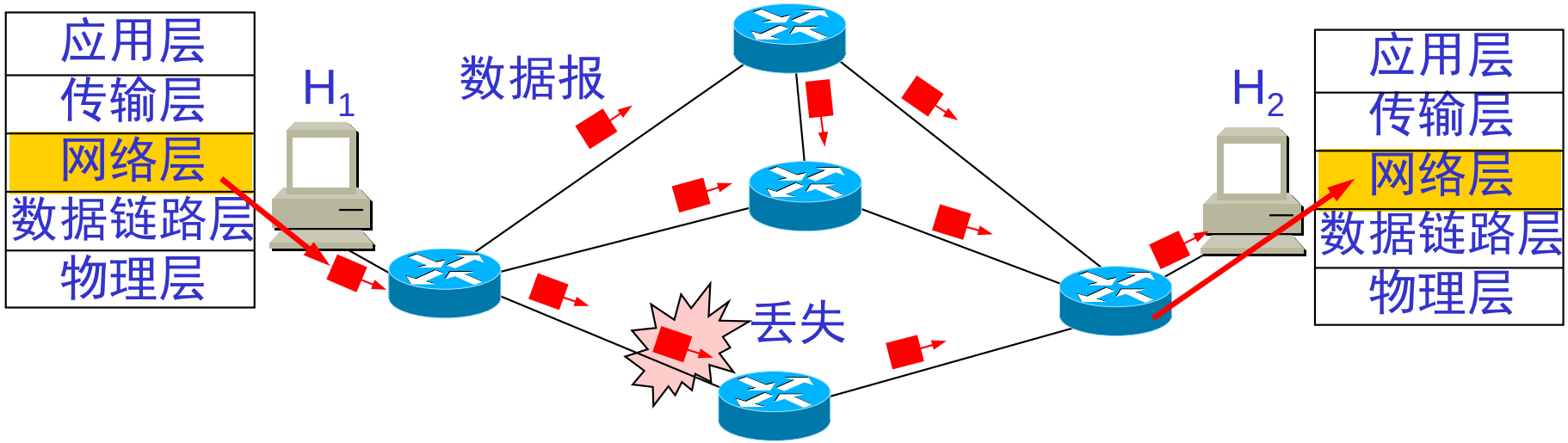
4.1 网络层提供的两种服务

- 网络层应该向传输层提供怎样的服务?
  - 两种选择：面向连接 or 无连接
  - 曾引起了长期的争论
  - 争论的实质：数据的可靠传输应该由网络还是端系统来负责？
- 面向连接的服务，即虚电路(virtual circuit)
  - 通信双方在开始数据传输前，先由网络建立连接，之后的数据均通过该连接进行，由网络保证数据传输的可靠性
  - 虚电路只是一种逻辑连接，分组沿着这条逻辑连接按照存储转发方式传送，而并不是真正建立了一条物理连接
  - 支持方：以电信公司为代表的一派
- 无连接的服务，即数据报(datagram)
  - 网络在发送数据时不需要先建立连接，每一个分组在网络中独立传送
  - 网络层不保证服务质量，分组可能出错、丢失、重复和失序，也不保证分组传送的时限
  - 支持方：以Internet为代表的一派
- TCP/IP采用数据报服务

packet：分组、数据包



虚电路：**H1** 发送给 **H2** 的所有分组都沿着同一条虚电路传送



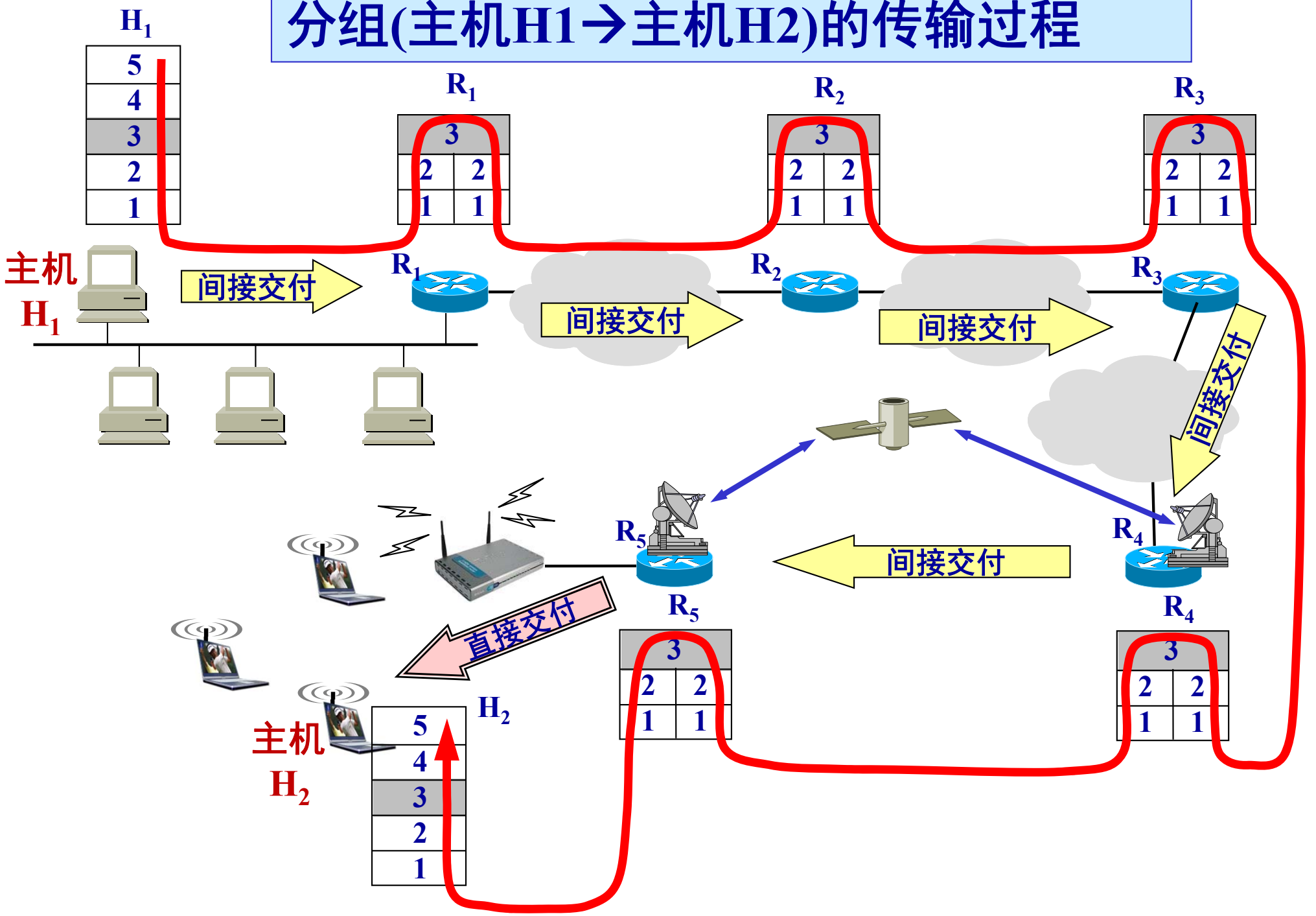
数据报：**H1** 发送给 **H2** 的分组可能沿着不同路径传送

4.1 网络层提供的两种服务

虚电路与数据报的比较

对比的方面	虚电路服务	数据报服务
思路	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
分组的转发	属于同一条虚电路的分组均按照同一路由进行转发	每个分组独立选择路由进行转发
当结点出故障时	所有通过出故障的结点的虚电路均不能工作	出故障的结点可能会丢失分组，一些路由可能会发生变化
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
端到端的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

分组(主机H1→主机H2)的传输过程

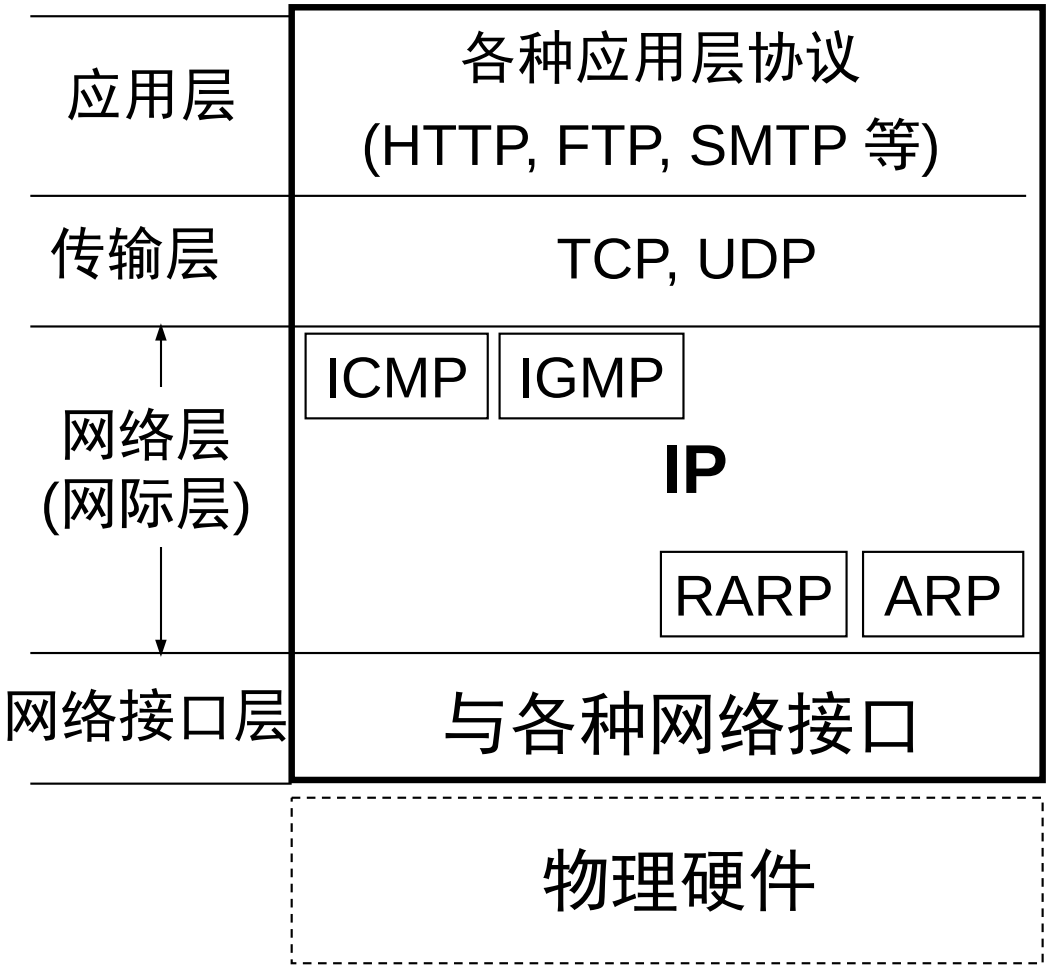


4.2 网际协议 IP

4.2 网际协议 IP

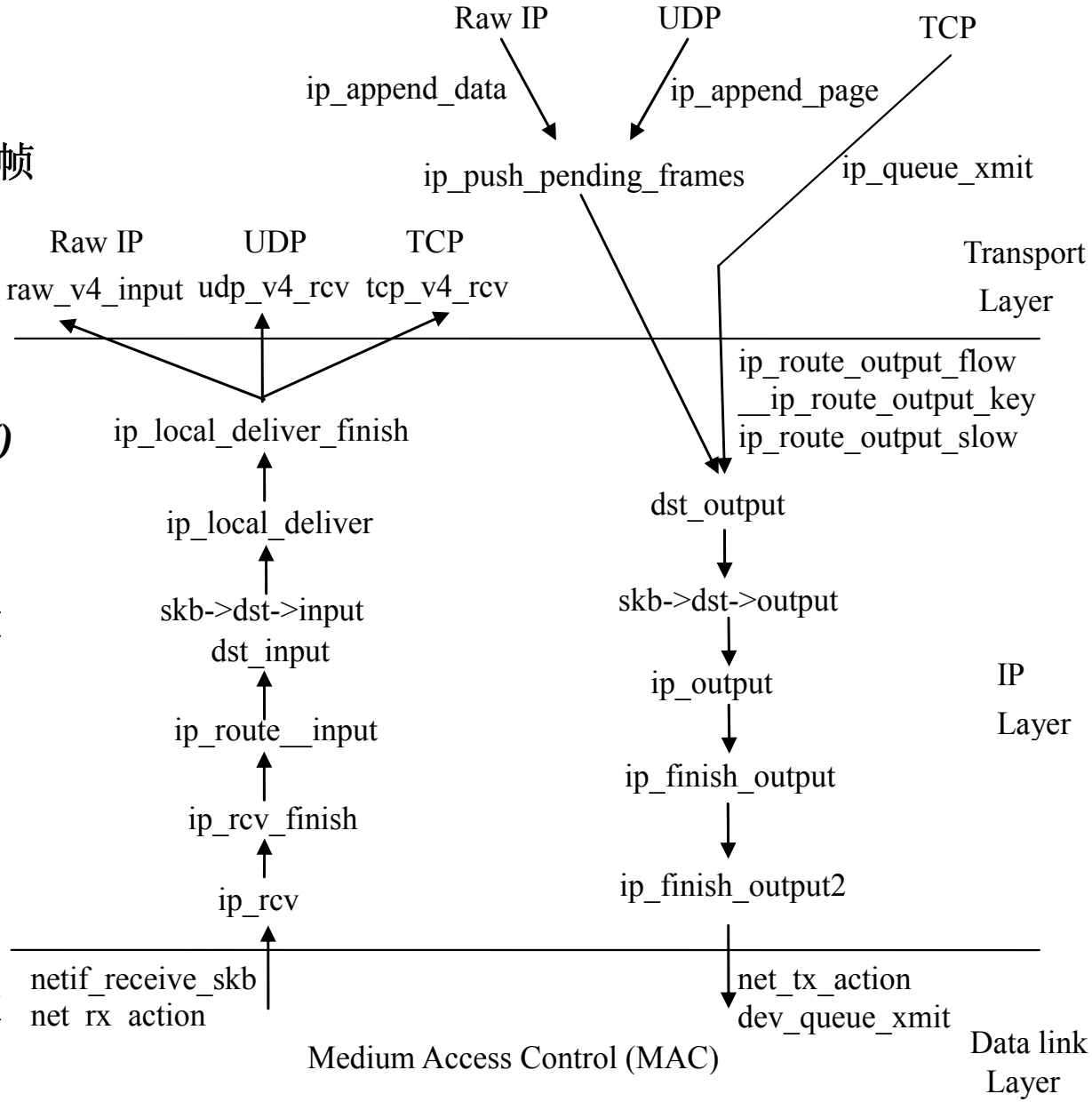
一、IP(Internet Protocol)简介

- 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四个协议：
  - 地址解析协议ARP (Address Resolution Protocol)
  - 逆地址解析协议RARP (Reverse Address Resolution Protocol)
  - 网际控制报文协议ICMP (Internet Control Message Protocol)
  - 网际组管理协议IGMP (Internet Group Management Protocol)



Linux中接收/发送包的调用图

- 包接收：
  - 网络接口卡(NIC)收到帧后触发中断
  - 中断服务程序调用***net_rx_action()***接收帧
  - 调用网络层接口函数***netif_receive_skb()***将帧中数据交给网络层
  - 包被注册到***sk_buff***中以便后续处理
  - 如为IP协议包则调用***ip_rcv()***作协议处理
  - 如包是发给本机的，则调用***ip_local_deliver()***和***ip_local_deliver_finish()***将数据交给传输层
- 包发送：
  - 根据传输层协议不同，分别调用接口函数***ip_append_data()***、***ip_append_page()***或***ip_queue_xmit()***将数据交给传输层
  - 调用***dst_output()***，将包注册到***sk_buff***
  - 如为IP包，则调用***ip_output()***
  - 如不分片，则***ip_finish_output2()***调用***net_tx_action()***将包交给数据链路层
  - 调用网卡驱动程序接口函数发送帧，帧发送完毕后通常会产生中断通知上层



注：**sk_buff是Linux中用于存储和处理包的数据结构，通过使用sk_buff，无需在各层间和程序模块间复制数据，而只需传递指针。采用双向链表结构**

4.2 网际协议 IP

二、分类的IP 地址

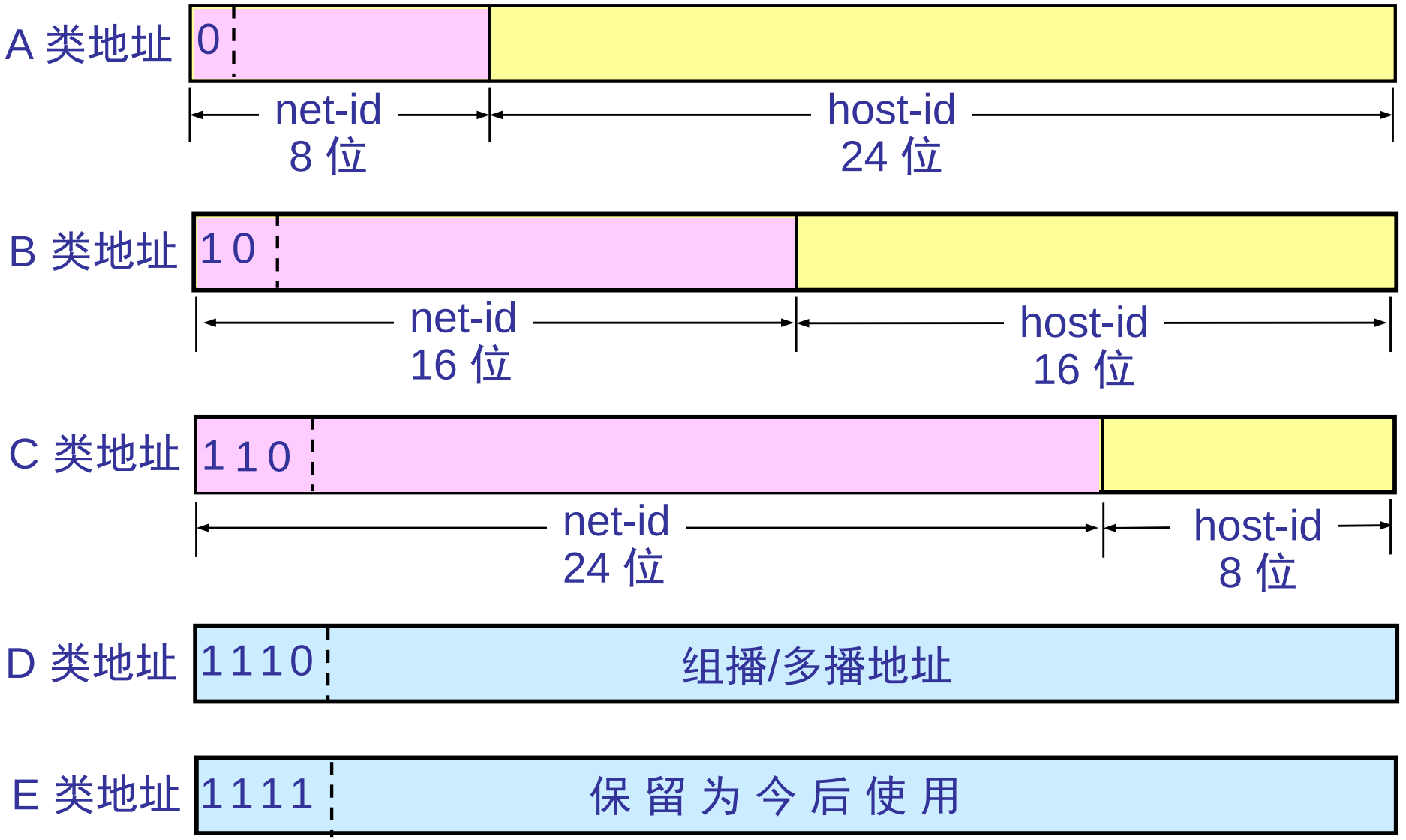
- **IP 地址**
  - 分配给主机或路由器的标识符，目前使用的**IPv4**为**32**位**IP**地址
  - **IP** 地址的分配由**ICANN (Internet Corporation for Assigned Names and Numbers)**负责
- **IP**地址的编址方法经历了三个阶段：
  - 分类的 **IP** 地址：最基本的编址方法，**1981** 年通过标准
  - 子网的划分：最基本编址方法的改进， **1985** 年成为标准**[RFC 950]**
  - 构成超网：比较新的无分类编址方法，**1993** 年提出

  } 4.3节介绍
- 分类的**IP**地址
  - **IP**地址被分为***A, B, C, D, E***五类，每一类地址都包含网络号**(net-id)**和主机号**(host-id)**两个字段

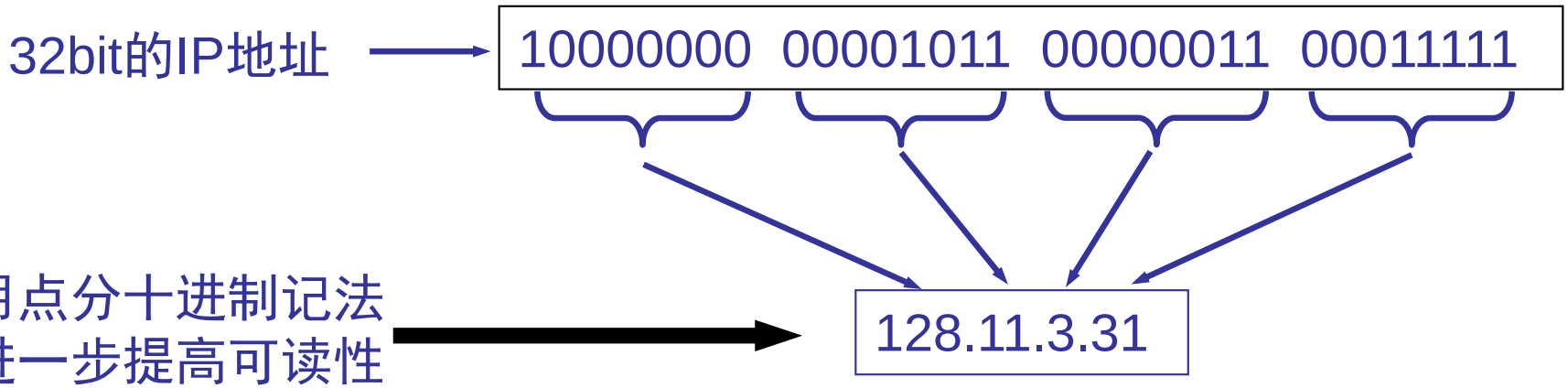
    IP 地址 ::= { <网络号>, <主机号>}
  - 不同类的**IP**地址区别主要是网络号、主机号的长度不同

4.2 网际协议 IP

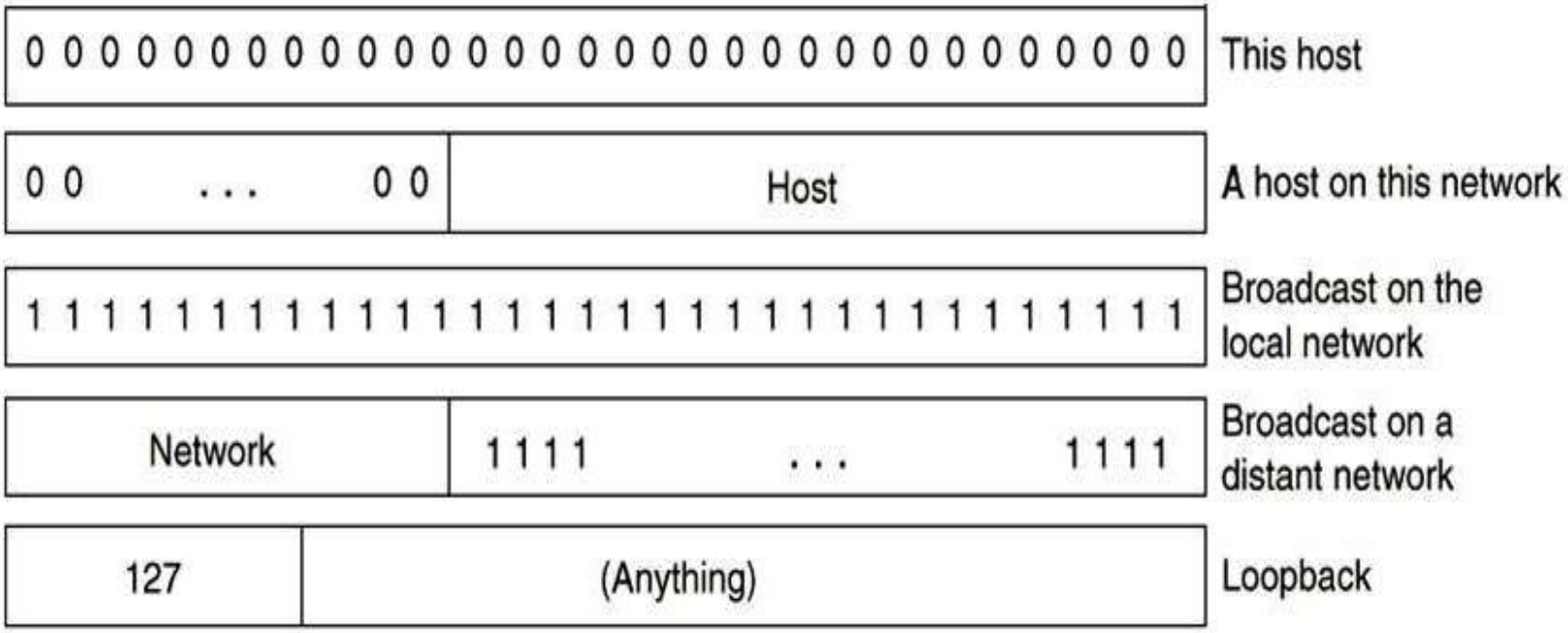
IP 地址中的网络号字段和主机号字段



IP 地址的表示方法：点分十进制记法(dotted decimal notation)



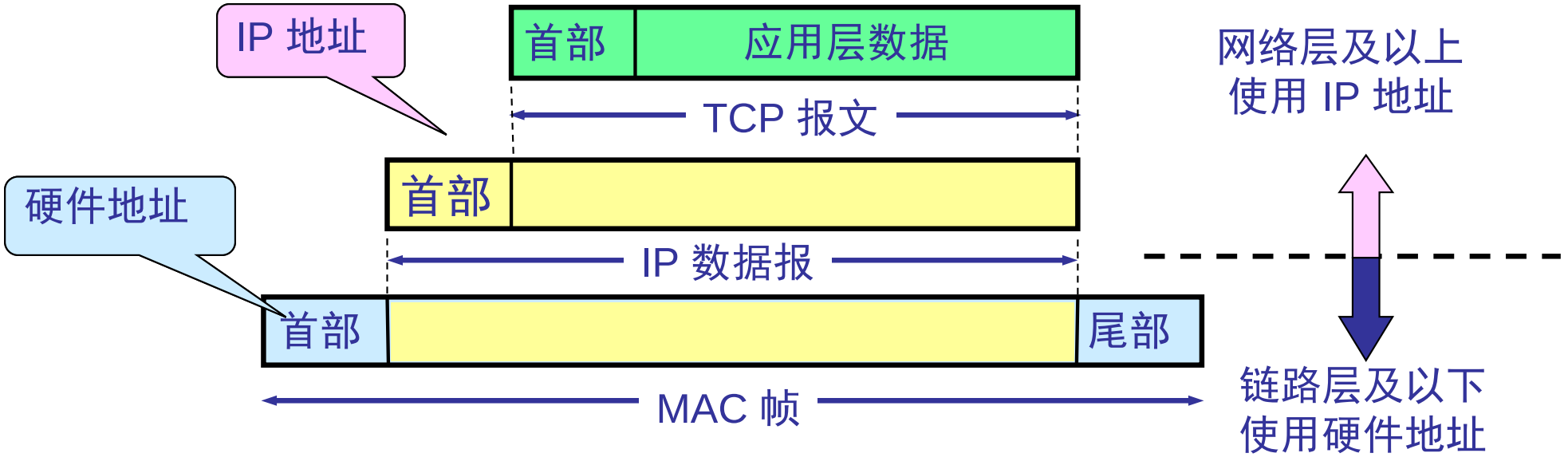
- 全0、全1的IP地址有特殊含义
  - 全0表示本网络或本主机
  - 全1表示广播地址



4.2 网际协议 IP

三、 IP 地址与硬件地址

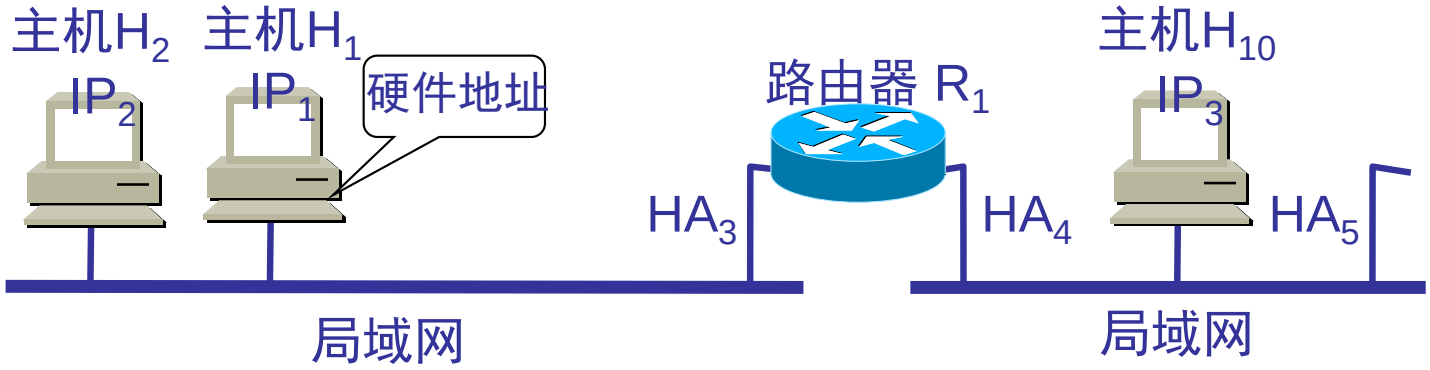
- **IP地址**
  - 网络层及以上各层使用的地址，是一种逻辑地址
  - 存放在**IP**包头部
- 物理地址
  - 数据链路层及物理层使用的地址
  - 存放在数据链路层的帧中
    - 问题：帧中有无**IP**地址？

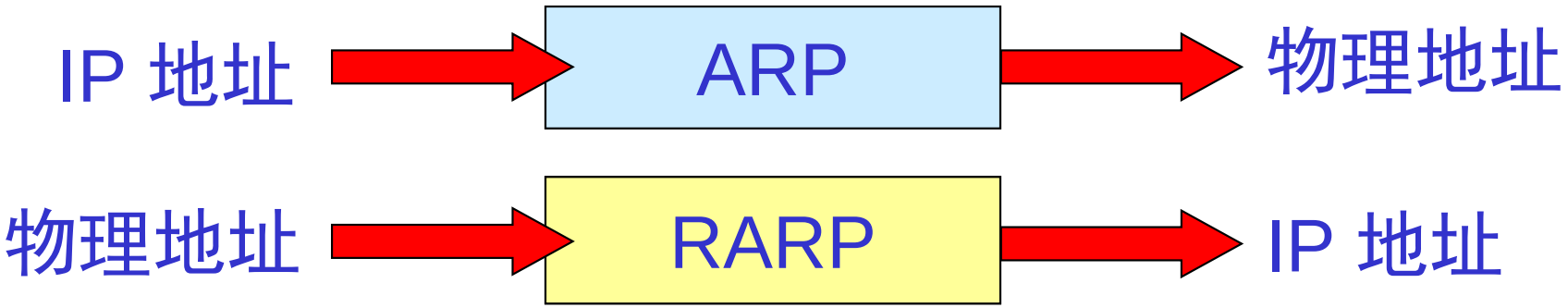


4.2 网际协议IP

四、ARP与RARP协议

- **IP 地址与物理地址的相互转换问题**
  - 例：如下图，主机H_{10}向主机H_1发送了IP包，路由器R_1要想在局域网中将IP包发送给主机H_1，需知道H_1的物理地址
- **RFC 826: An Ethernet Address Resolution Protocol**

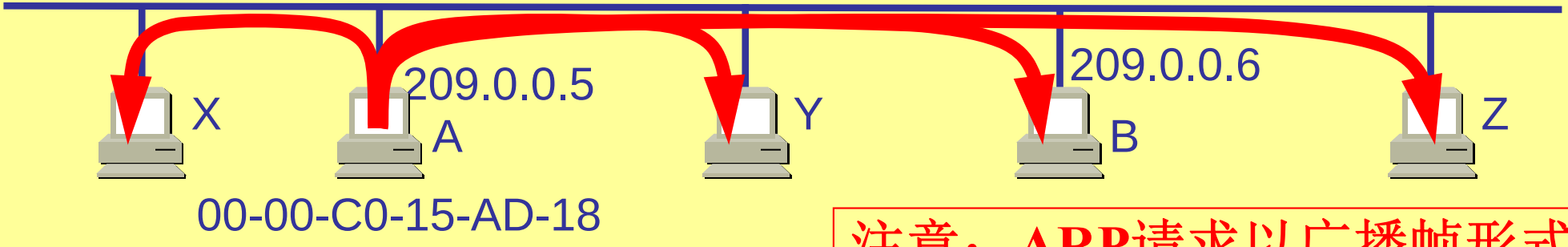




主机 A 广播发送
ARP 请求分组

我是 209.0.0.5，硬件地址是 00-00-C0-15-AD-18
我想知道主机 209.0.0.6 的硬件地址

ARP 请求

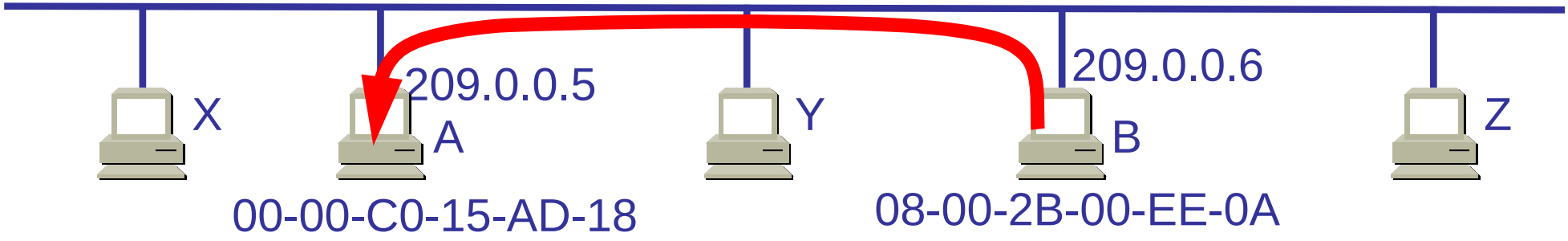


注意：ARP请求以广播帧形式发送

主机 B 向 A 发送
ARP 响应分组

我是 209.0.0.6
硬件地址是 08-00-2B-00-EE-0A

ARP 响应



4.2 网际协议IP

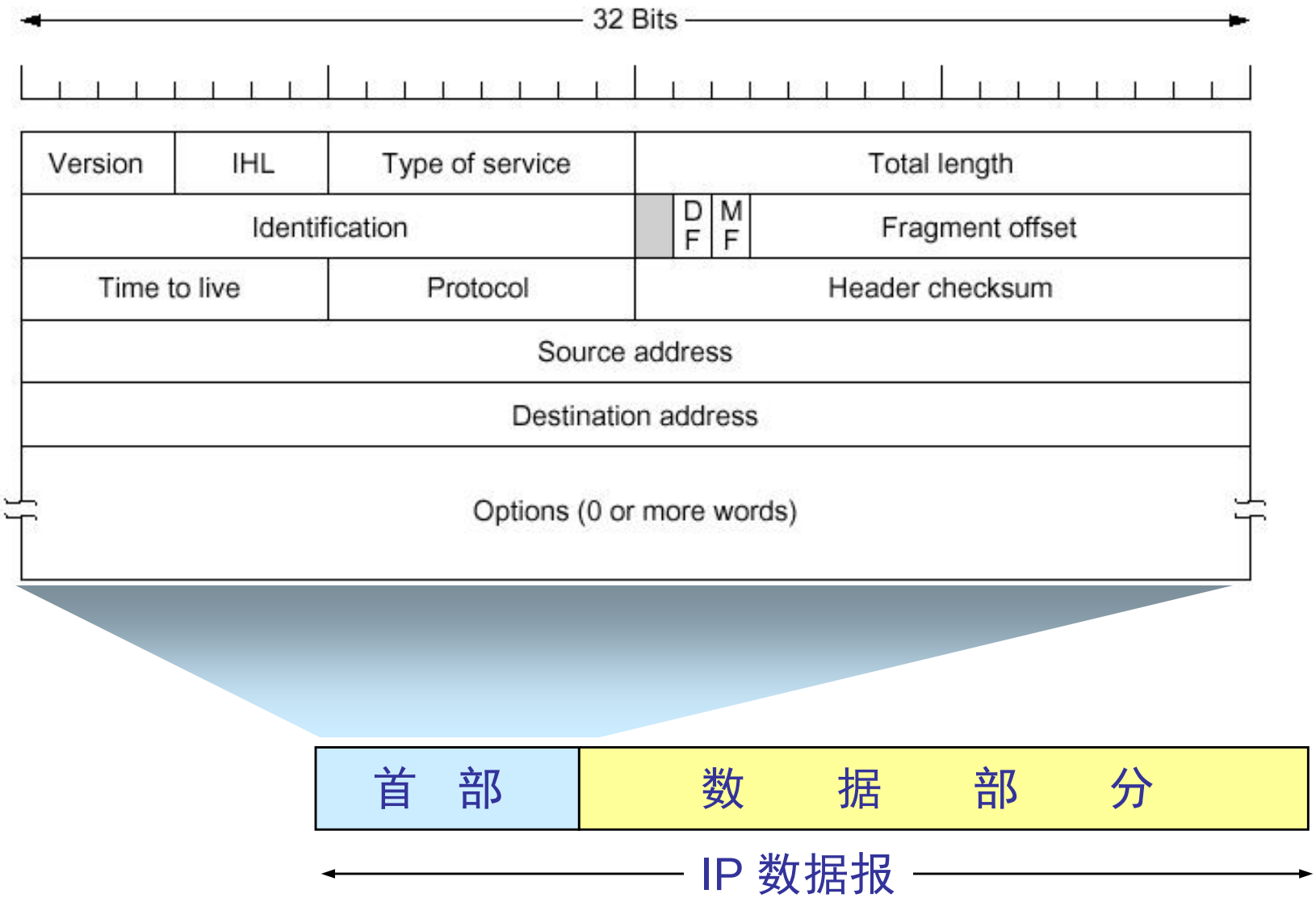
```
C:\>arp -a

Interface: 192.168.1.103 --- 0x10004
  Internet Address      Physical Address      Type
  192.168.1.1           70-a8-e3-e0-ba-f8     dynamic
```

四、ARP与RARP协议

- **ARP协议(Address Resolution Protocol)**
  - 主机设有一个ARP高速缓存(ARP cache)，存有本地局域网上各主机和路由器的 IP 地址与硬件地址的映射表
  - 当主机 A 欲向本局域网上的主机B发送IP包时
    - ① 先在其ARP高速缓存中查看有无主机B的IP地址
    - ② 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，通过局域网发送
    - ③ 如无，则在网络中广播一个ARP请求
    - ④ 当主机B收到ARP请求后，向主机A返回一个ARP应答，告知自己的物理地址
  - 注意：
    - ARP解决同一局域网中的主机或路由器的 IP 地址和硬件地址的映射问题
    - 如果目的主机不在本局域网内，IP包需经由路由器转发
      - 此时在局域网内要完成的是路由器IP与物理地址的映射

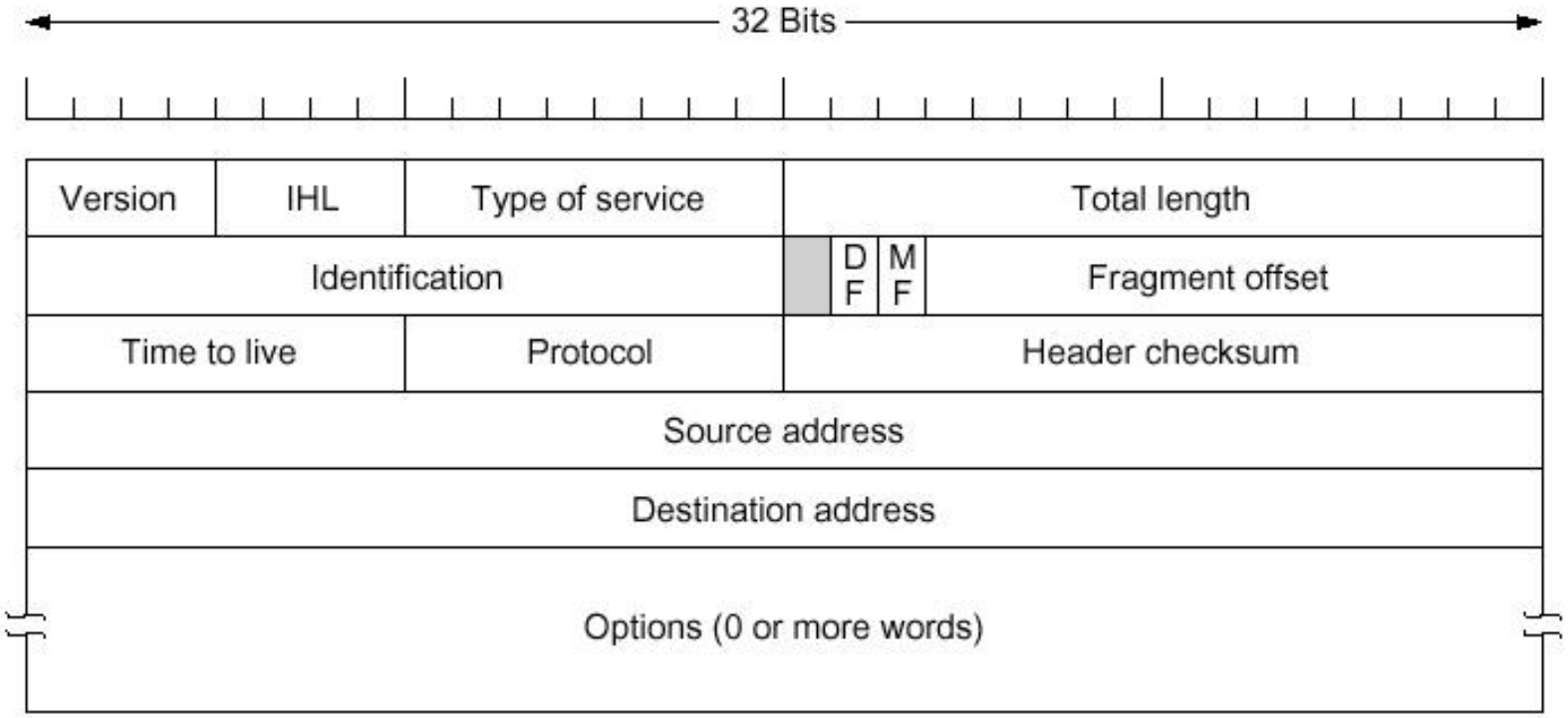
五、IP数据报格式

- 一个 IP包由头部和数据两部分组成
- 头部：20字节的固定字段 + 0到多个可选字段



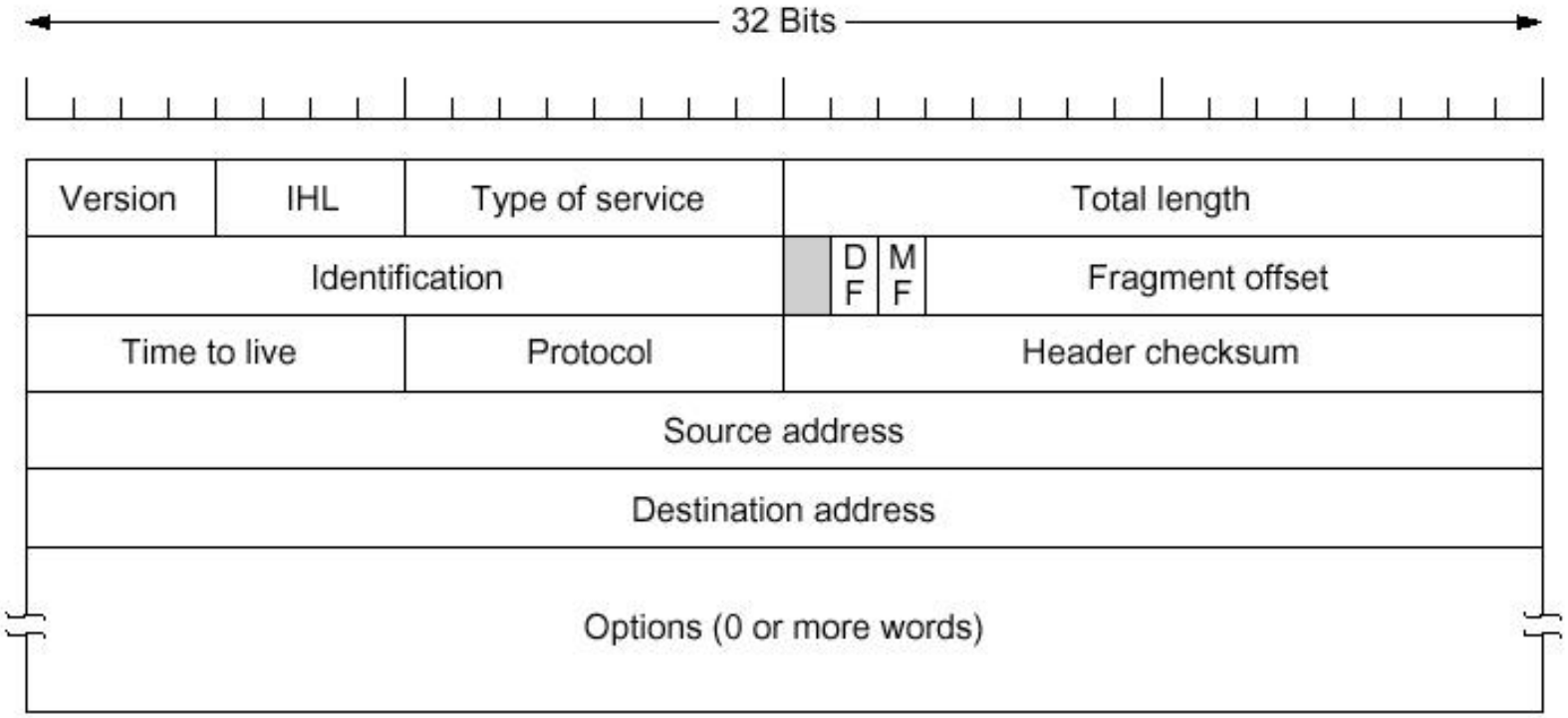
五、IP数据报格式(cont.)

- **Version字段：4bit，IP 协议的版本，目前的 IP 协议版本号为 4（即 IPv4)**
- **IHL：4bit，IP包头长度，最小5，最大15，单位为word(32bit)。因此 IP包头最长60 字节**
- **Type of service：1字节，服务类型，目前很多路由器忽略该字段**
- **Total Length：2字节，IP包总长度(含头部和数据)，单位为字节。因此IP包的最大长度为 65535 字节**



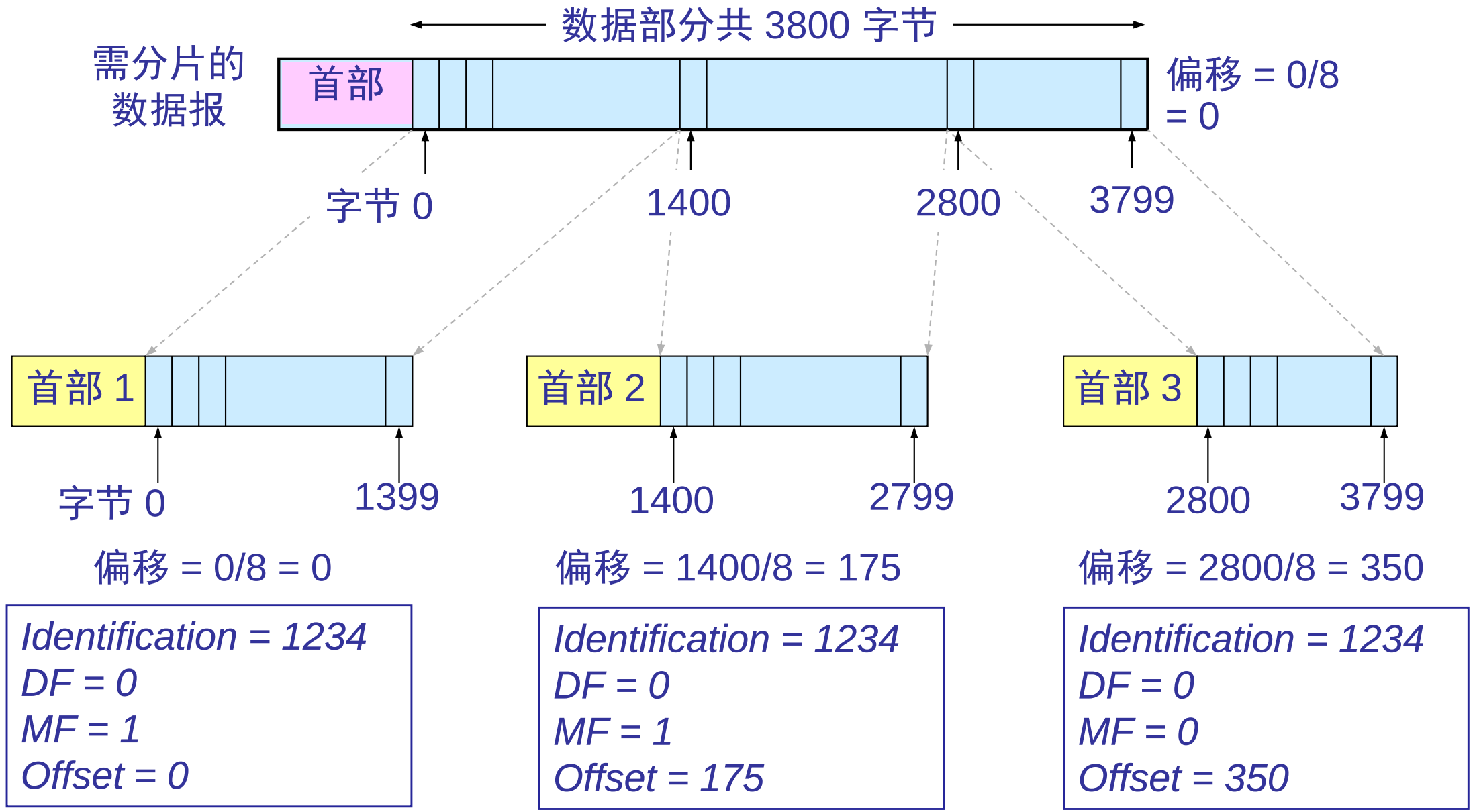
五、IP数据报格式(cont.)

- **Identification：2字节，标识，是一个计数器，用来产生IP包的标识**
  - **超过数据链路层MTU(Maximum Transmission Unit)的IP包要分片传输**
  - **分片的多个包具有相同的标示，便于接收端重组**
- **DF：1bit，Don’t Fragment，当 DF=0 时允许分片**
- **MF：1bit，More Fragment，MF=1表示后面“还有分片”；MF=0表示最后一个分片**
- **Fragment offset：13bit，片偏移，较长的包在分片后，某片在原分组中的相对位置，以8字节为单位**



五、IP数据报格式(cont.)

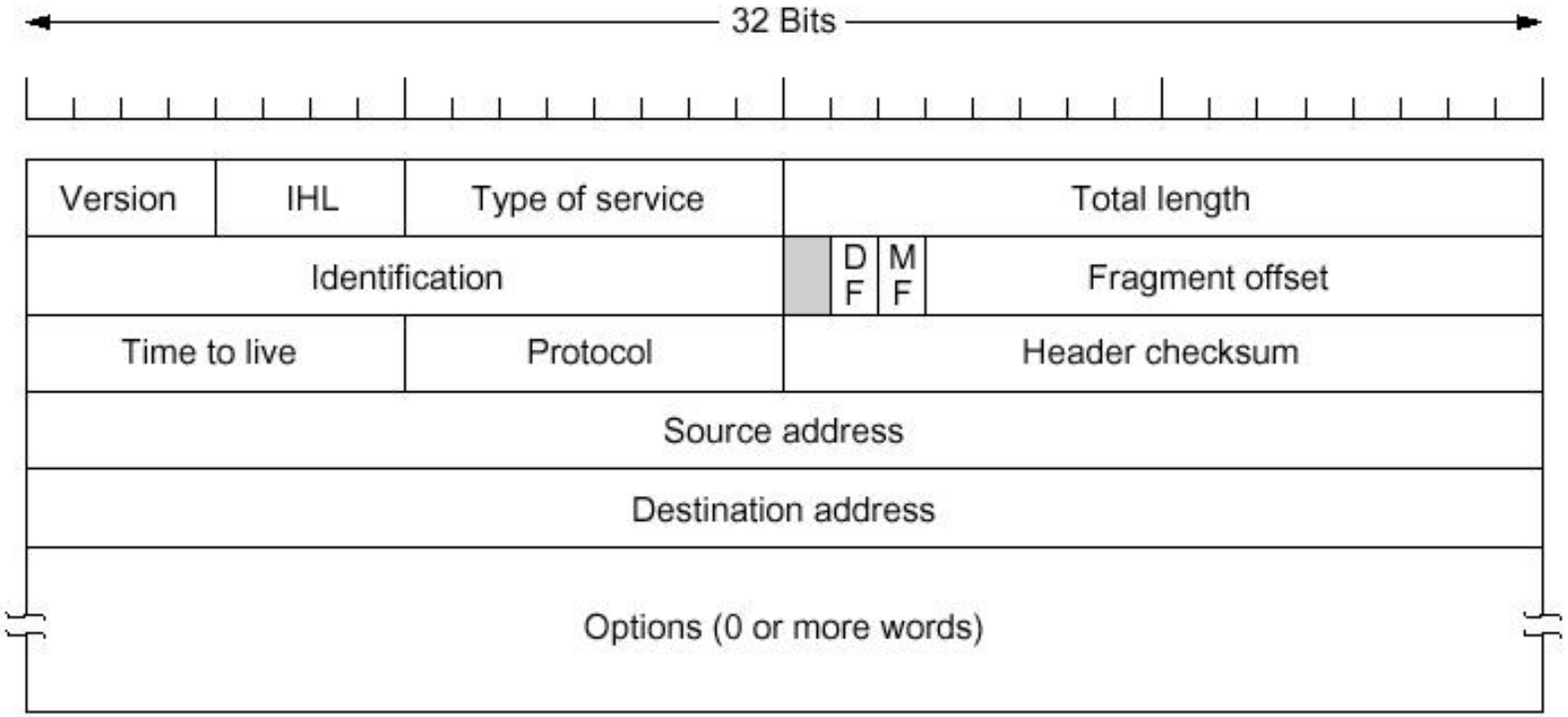
- 分片举例(假设数据链路层一帧的载荷长度≤1420字节)



五、IP数据报格式(cont.)

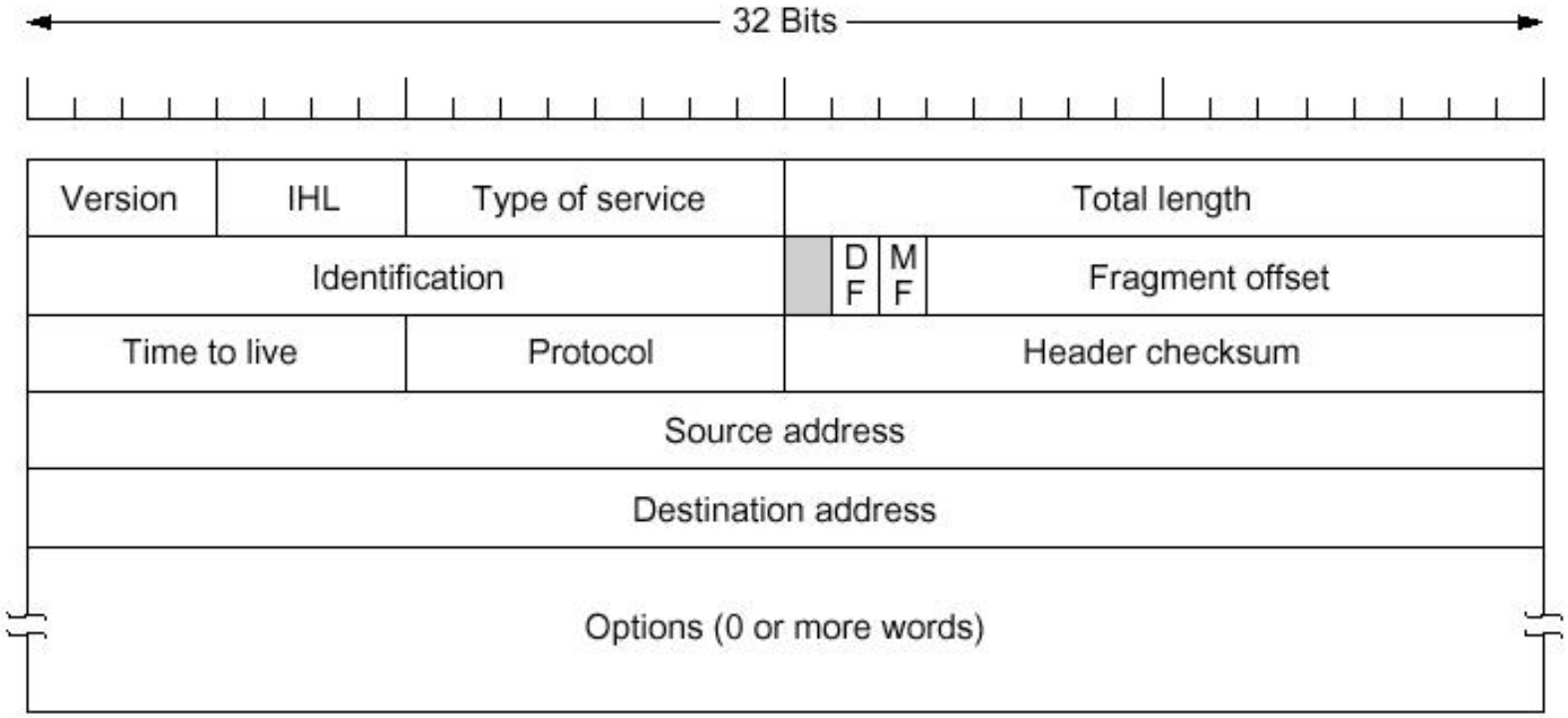
IP包每经过一个路由器俗称为“一跳(hop)”，经过的路由器个数俗称为“跳数(number of hops)”

- **Time to live(TTL)**：1字节，生存时间，IP包在网络中可通过的路由器个数的最大值
  - 实际实现中，IP包每经过一个路由器TTL减1，为0则丢弃，并向源主机发送一个告警包
  - 最大值为255，由源主机设定初始值，Windows操作系统一般为128，UNIX操作系统一般为255，Linux一般为64



五、IP数据报格式(cont.)

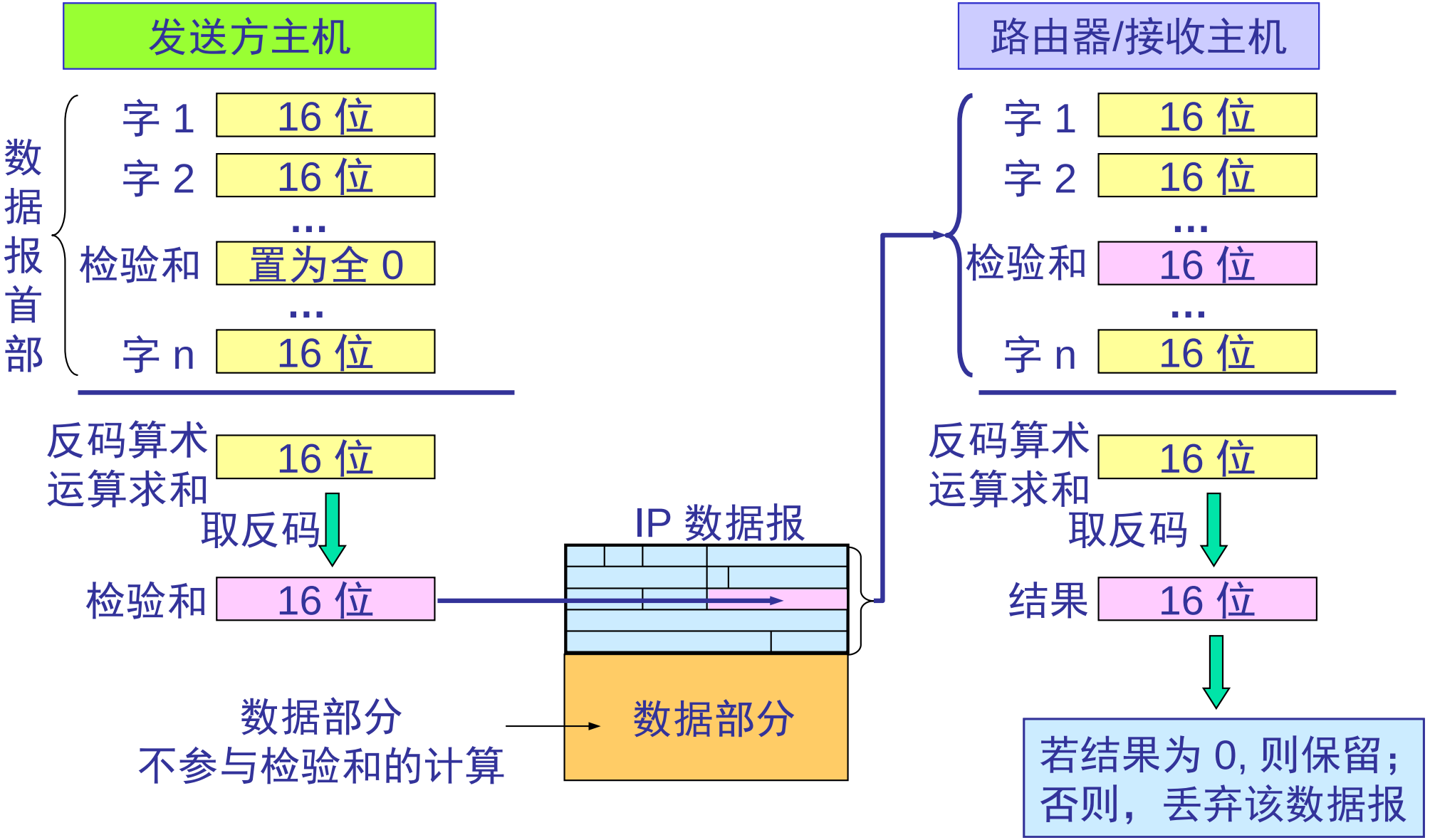
- **Protocol：8bit，协议字段，该包中数据部分的协议类型，即上层协议类型 → 该字段决定了该包将交由哪里**
- **Header checksum：2字节，包头校验和(注意：只针对包头)**
- **Source address：4字节，源IP地址**
- **Destination address：4字节，目的IP地址**
- **选项字段：以4字节为单位，最长40字节。实际网络中很少使用**



五、IP数据报格式(cont.)

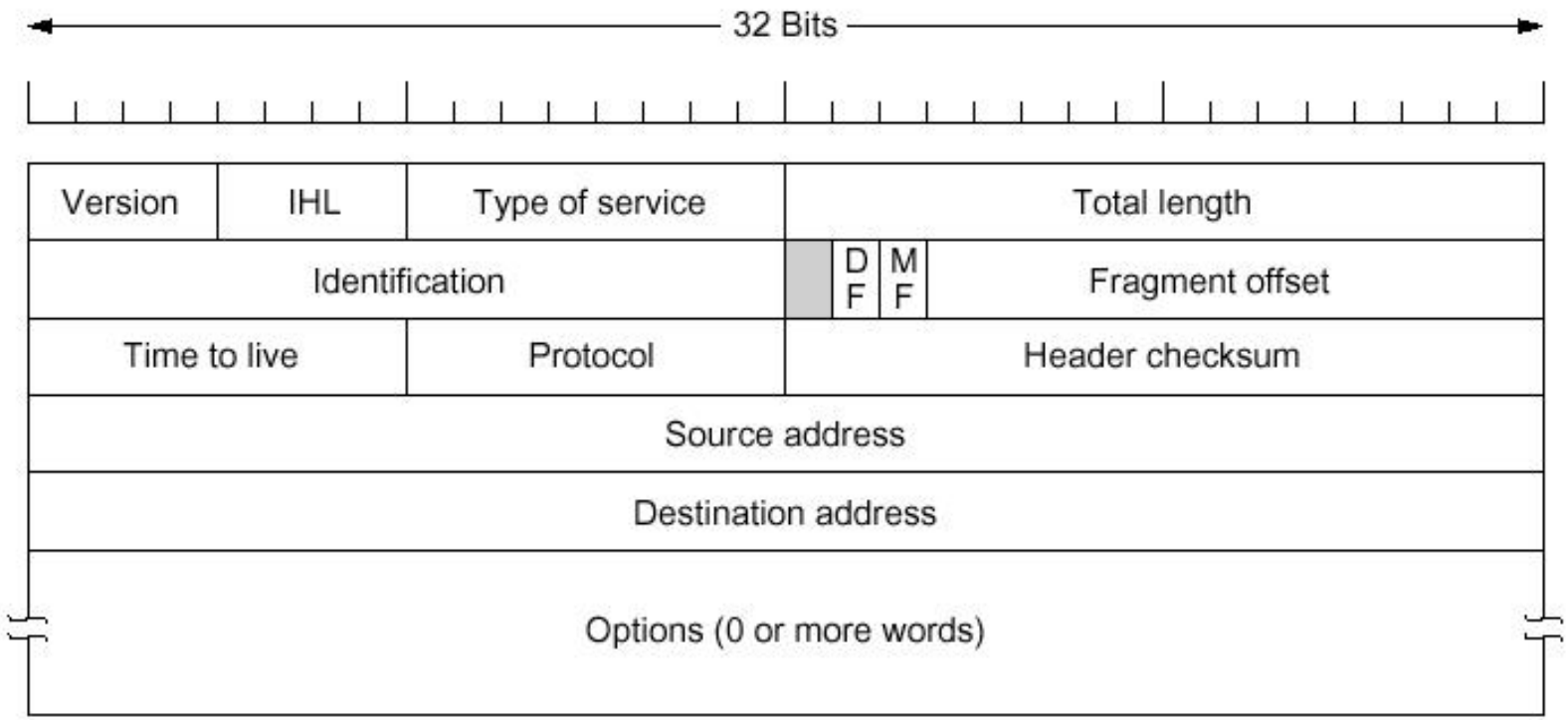
- 校验和算法：对IP包头，每16位求反，循环相加(进位加在末尾)，和再求反

问：IP包在网络中传输过程中，何时检查校验和？何时计算填写校验和？



IP包示例

注意包中整数的字节顺序




```
00 d0 d0 72 a4 b4 00 16 ea c3 8c 6c 08 00 45 00  .行r亡..管宣..E.
00 3c 00 d6 00 00 80 01 b6 8f c0 a8 01 0a c0 a8  .<.?.€.稅括..括
01 01 08 00 47 5c 02 00 04 00 61 62 63 64 65 66  ....G\....abcdef
67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

4.2 网际协议 IP

```
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:       000. ....   = routine
IP:       ...0 ....   = normal delay
IP:       .... 0...   = normal throughput
IP:       .... .0..   = normal reliability
IP:       .... ..0.   = ECT bit - transport protocol will ignore the CE bit
IP:       .... ...0   = CE bit - no congestion
IP: Total length    = 60 bytes
IP: Identification  = 214
IP: Flags           = 0X
IP:       .0.. ....   = may fragment
IP:       ..0. ....   = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live    = 128 seconds/hops
IP: Protocol        = 1 (ICMP)
IP: Header checksum = B68F (correct)
IP: Source address      = [192.168.1.10], X301
IP: Destination address = [192.168.1.1]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 475C (correct)
ICMP: Identifier = 512
ICMP: Sequence number = 1024
ICMP: [32 bytes of data]
```

```
00000000: 00 d0 d0 72 a4 b4 00 16 ea c3 8c 6c 08 00 45 00  .行r亡..昏窘..E.
00000010: 00 3c 00 d6 00 00 80 01 b6 8f c0 a8 01 0a c0 a8  .<.?.€.稜括..括
00000020: 01 01 08 00 47 5c 02 00 04 00 61 62 63 64 65 66  ....G\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

Linux中IP包头定义

```
struct  iphdr {
        #if defined(_LITTLE_ENDIAN_BITFIELD)
            __u8  ihl: 4,
                  version: 4;
        #elif defined(__BIG_ENDIAN_BITFIELD)
            __u8  version: 4,
                  ihl: 4;
        #else
        #error "Please fix <asm/byteorder.h>"
        #endif
            __u8  tos;
            __be16  tot_len;
            __be16  id;

            __be16  frag_off;
            __u8  ttl;
            __u8  protocol;
            __sum16  check;
            __be32   saddr;
            __be32   daddr;
            /*The options start here.*/
};
```

4.3 划分子网和构造超网

4.3 划分子网和构造超网

一、划分子网

- 分类IP地址的缺点
  - IP地址空间的利用率有时很低
    - A类地址的主机数超过1000万，B类地址也超过6万
  - 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
- 1985年起，增加子网字段，形成三级IP地址
  - RFC 950: Internet Standard Subnetting Procedure

4.3 划分子网和构造超网

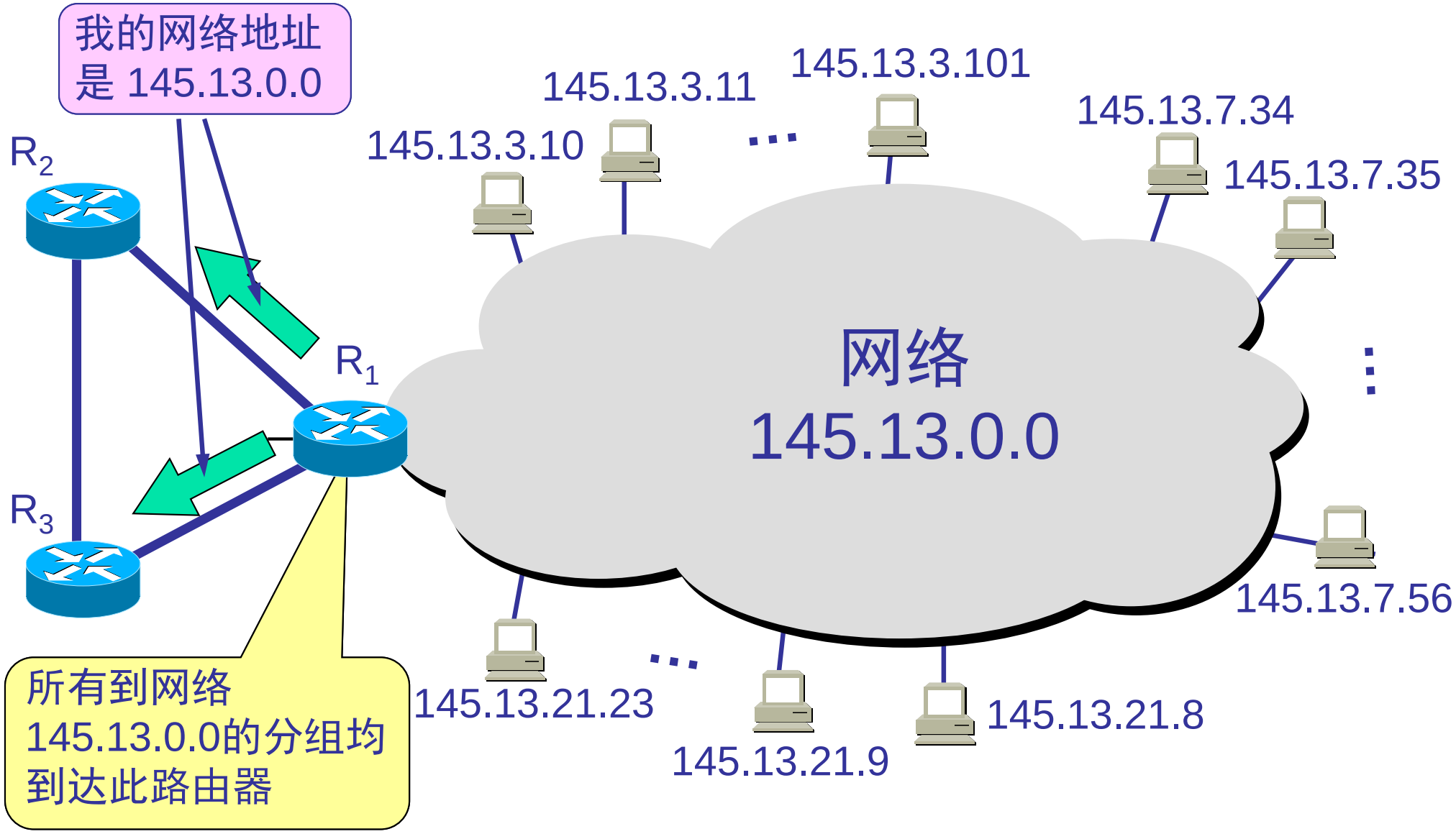
一、划分子网

- 划分子网的基本思路
  - 拥有多个物理网络的单位可按物理网络划分为若干个子网(subnet)
  - 从主机号借用若干位作为子网号subnet-id，三级IP地址记为：

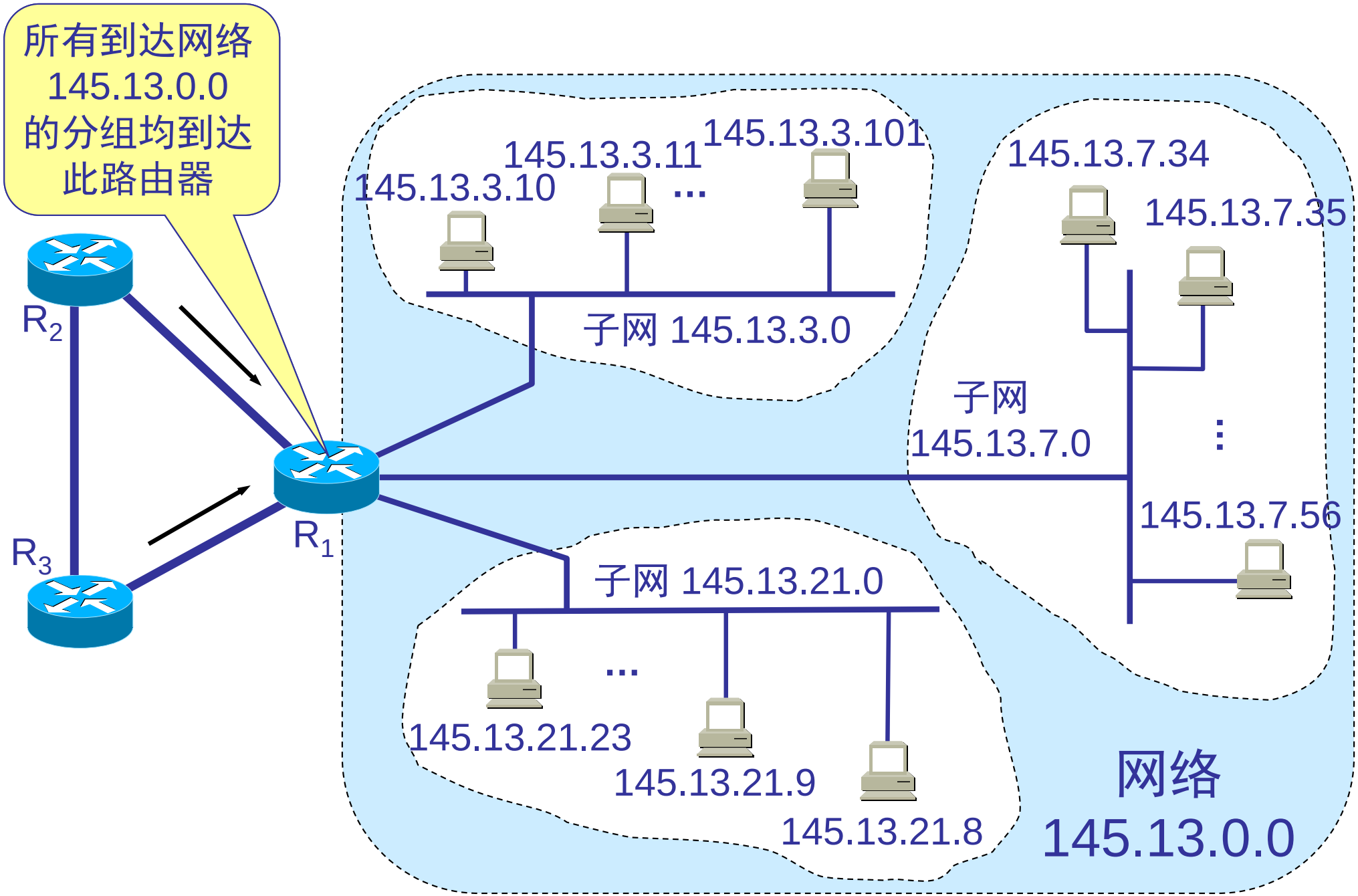
IP地址 ::= {<网络号>, <子网号>, <主机号>}

  - 从其他网络发来的IP数据报，仍然根据IP数据报的目的网络号net-id，找到本网络的路由器，此路由器收到IP数据报后，再按目的网络号net-id和子网号subnet-id找到目的子网

一个未划分子网的 B 类网络145.13.0.0



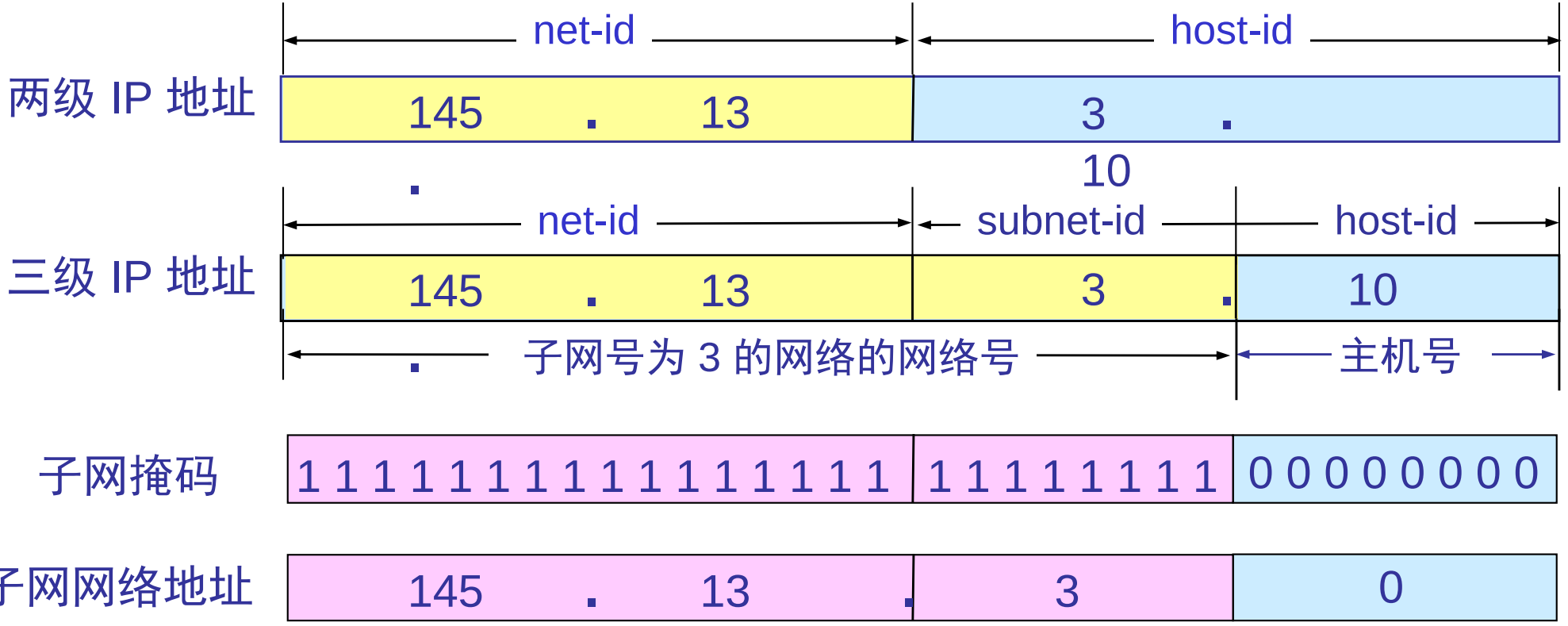
划分为三个子网后


所有到达网络
145.13.0.0
的分组均到达
此路由器
R2
R3
R1
145.13.3.10
145.13.3.11
...
145.13.3.101
子网 145.13.3.0
145.13.7.34
145.13.7.35
子网
145.13.7.0
145.13.7.56
子网 145.13.21.0
...
145.13.21.23
145.13.21.9
145.13.21.8
网络
145.13.0.0

4.3 划分子网和构造超网

一、划分子网

- 子网掩码的提出
  - 在划分子网后，路由器需要把数据报转发给不同的子网，从何处得知子网划分信息？IP地址中并未明确包含这部分信息
  - 解决方法：使用子网掩码(subnet mask)
- 对目的IP地址和子网掩码执行“按位与”操作，即得到子网地址



4.3 划分子网和构造超网

一、划分子网

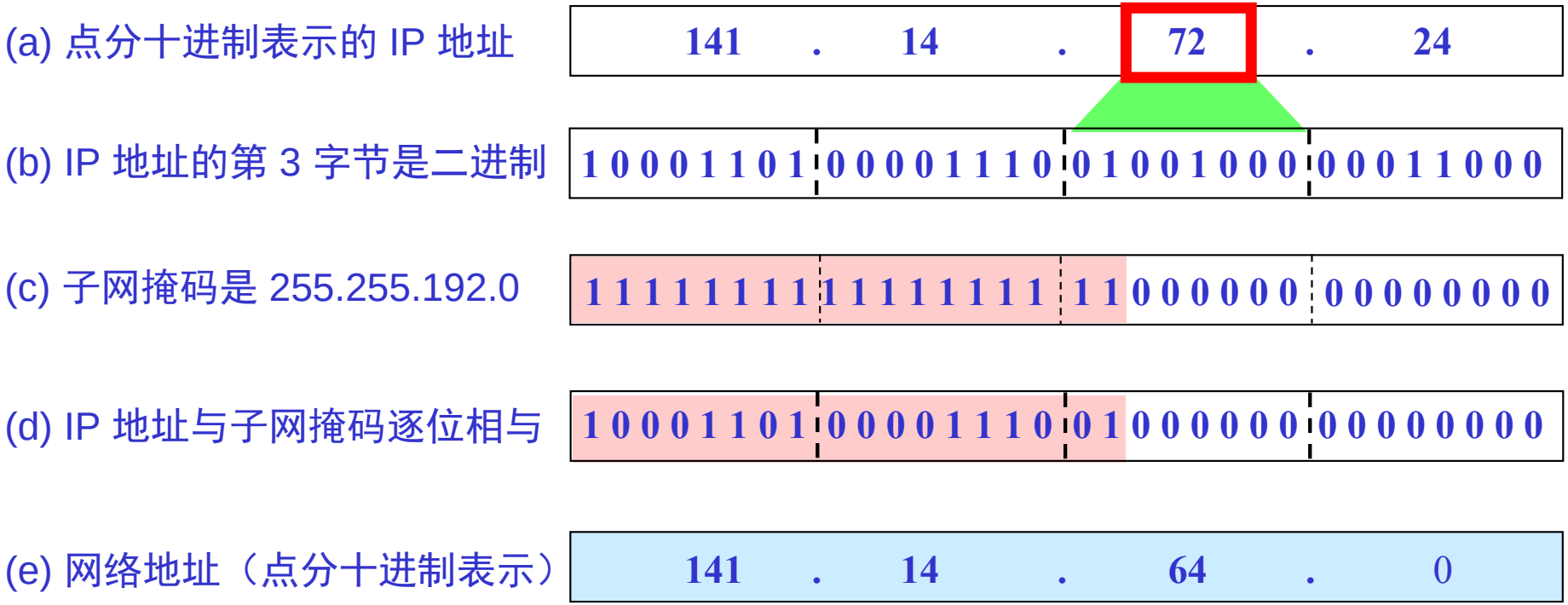
- 子网掩码
  - **Internet**标准规定，所有网络都必须使用子网掩码
  - 子网掩码是一个网络或一个子网的重要属性
  - 路由器的路由表中的每个表项除了包含目的网络地址外，还要有子网掩码栏目
  - 如一个路由器连接在两个子网上，就拥有两个网络地址和两个子网掩码

4.3 划分子网和构造超网

例：**IP address：141.14.72.24**

subnet mask：255.255.192.0

求网络地址



二、使用子网掩码的分组转发过程

- 路由器中路由表项包含三项基本信息：
  - 目的网络地址、子网掩码、下一跳地址

目的网络	子网掩码	下一跳
…	…	…

- 转发流程：

① 从收到的分组的首部提取目的IP地址***D***

② 先用与该路由器直接相连各网络的子网掩码和***D***逐位相“与”，看是否和相应的网络地址匹配，若匹配，则将分组直接交付；否则就是间接交付，执行③

③ 若路由表中有目的地址为***D***的特定主机路由，则将分组传送给指明的下一跳路由器；否则执行④

④ 对路由表中的每一行的子网掩码和 ***D*** 逐位相“与”，若其结果与该行的目的网络地址匹配，则将分组传送给该行指明的下一跳路由器；否则执行⑤

⑤ 若路由表中有一个默认路由，则将分组传送给路由表中所指明的默认路由器；否则执行⑥

⑥ 报告转发分组出错

注意：路由表中仅包含局部路由信息，使用默认路由以确保任意分组的转发

核心操作：将目的IP地址与路由表中子网掩码“与”，并判断是否与目的网络匹配

例：考虑主机H1向H2发送数据包后的转发过程

① 主机H1根据自身设置判断目的地址是否在本地子网

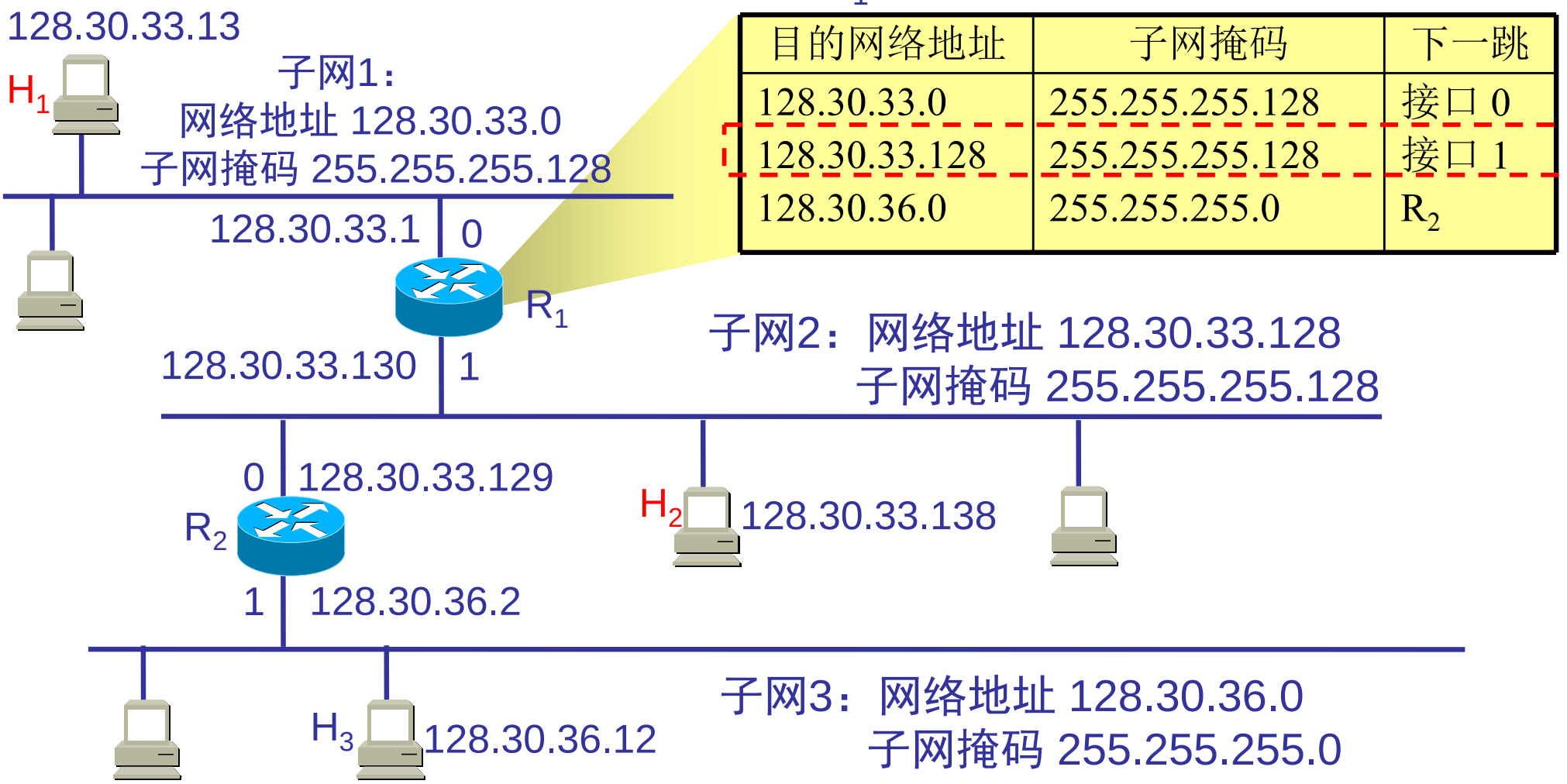
② 主机H1将数据包发给路由器R1(注意局域网内可能有ARP查询过程)

③ 路由器R1收到数据包后，在路由表中逐项根据子网掩码计算匹配项

④ 路由器R1将数据包通过子网2发给主机H2 (注意局域网内可能有ARP查询过程)

R_1 的路由表（未给出默认路由器）

目的网络地址	子网掩码	下一跳
128.30.33.0	255.255.255.128	接口 0
128.30.33.128	255.255.255.128	接口 1
128.30.36.0	255.255.255.0	R_2



4.3 划分子网和构造超网

针对上例的说明：

- 主机发送数据包是判断目的地址是否在本地子网的方法：

```
if ( ( 目的地址 & subnet mask ) == ( 主机地址 & subnet mask ) )
        目的地址在本地子网，直接交付；
else
        数据包发往gateway
```

- 路由器查找路由表进行表项匹配的过程：

```
if ( ( 目的地址 & subnet mask ) == 目的网络地址 )
        数据包发往该表项的网络出口；
```

- 在子网内直接交付过程：

```
查找ARP缓存，是否有目的IP地址对应的MAC地址
if (目的MAC地址在ARP缓存中)
        将IP数据包封装成帧后，在局域网内向目的MAC地址直接发送帧
else
        在子网内广播发送ARP请求，目的主机收到请求后返回ARP应答，由此
            得知目的主机MAC地址
```

4.3 划分子网和构造超网

三、无分类编址 CIDR

- **CIDR(Classless Inter-Domain Routing)无分类域间路由**
- **CIDR的主要特点**
  - 消除传统A类、B类和C类地址以及划分子网的概念
  - 使用各种长度的“网络前缀”(network-prefix)来代替分类地址中的网络号和子网号
  - IP地址从三级编址(使用子网掩码)又回到了两级编址

    IP地址 ::= {<网络前缀>, <主机号>}
  - CIDR还使用“斜线记法”(slash notation)，又称为CIDR记法
    - IP地址后加一个斜线“/”，后跟网络前缀所占的位数

例：128.14.35.7/20 表示该地址的高20位是网络前缀

4.3 划分子网和构造超网

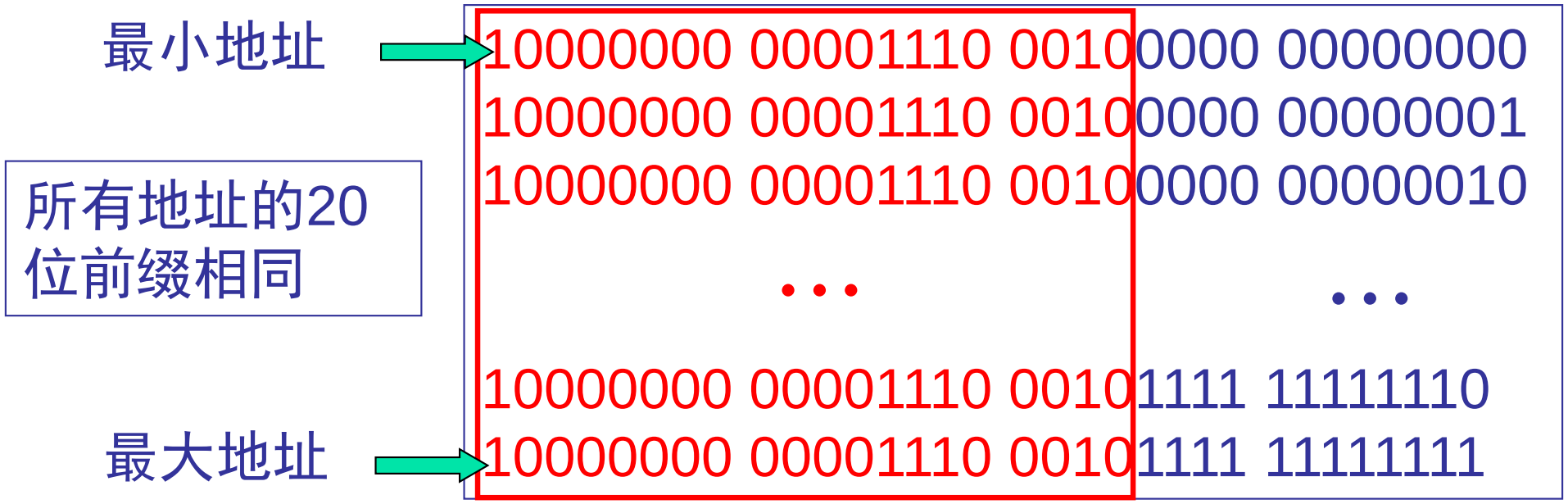
三、无分类编址 CIDR

- 网络前缀都相同的连续的 IP 地址组成“CIDR地址块”

例：128.14.32.0/20表示的CIDR地址块共有2^{12}个地址

地址块的起始地址：128.14.32.0

地址块的最大地址：128.14.47.255



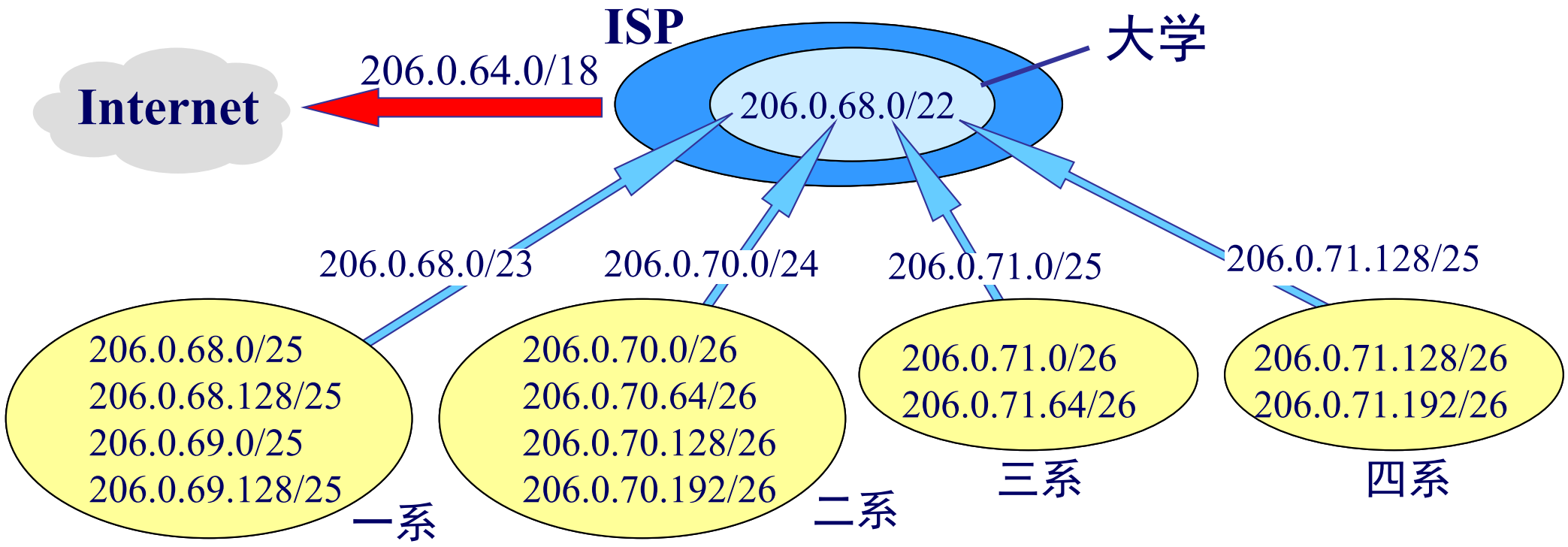
4.3 划分子网和构造超网

三、无分类编址 CIDR

- 路由聚合(route aggregation) ← CIDR带来的好处
  - 一个 CIDR 地址块可以表示很多地址，这种地址的聚合称为路由聚合
  - 路由聚合的好处：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，可以减少路由表中表项个数，并减少路由器间交换的路由信息量
  - 路由聚合也称为构成超网(supernetting)
    - 称为超网是由于CIDR地址块大多包含多个C类地址
- 关于地址掩码
  - CIDR不使用子网，但仍使用“地址掩码”这一名词
  - 例：/20的地址掩码是：11111111 11111111 11110000 00000000
- CIDR记法的其他形式
  - 10.0.0.0/10 可简写为 10/10，即省略点分十进制中低位连续的0
  - 网络前缀后跟星号 * 的表示方法

    如 00001010 00*，星号 * 之前为网络前缀，星号 *为任意主机号

CIDR 地址块划分举例



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

该ISP拥有64个C类地址，采用CIDR技术只需一个路由器表项，不采用则需要64个表项(所有相邻路由器中)

4.3 划分子网和构造超网

三、无分类编址 CIDR

- 最长前缀匹配
  - 使用**CIDR**时，路由表中的表项中的“目的网络地址”由固定长度变成了变长的“网络前缀”
  - 在查找路由表时可能会得到不止一个匹配结果
  - 最长前缀匹配**(longest-prefix matching)**原则
    - 从匹配结果中选择具有最长网络前缀的路由
    - 网络前缀越长，其地址块就越小，因而路由就越具体**(more specific)**
    - 最长前缀匹配又称为最长匹配或最佳匹配

最长前缀匹配举例

收到的分组的目的地址 D = 206.0.71.128

路由表中的项目：206.0.68.0/22　　(ISP)

　　206.0.71.128/25　(四系)

第 1 个表项206.0.68.0/22 的掩码 M 有 22 个连续的 1

	M =	11111111	11111111	11111100	00000000
AND	D =	206.	0.	01000111.	128
		206.	0.	01000100.	0

匹配！

第 2个表项 206.0.71.128/25的掩码 M 有 25 个连续的 1　√

	M =	11111111	11111111	11111111	10000000
AND	D =	206.	0.	01000111.	10000000
		206.	0.	01000111.	10000000

匹配！

4.3 划分子网和构造超网

2010年的一道考研题：

某网络的IP地址空间为192.168.5.0/24，采用长子网划分，子网掩码为255.255.255.248，则该网络的最大子网个数、每个子网内的最大可分配地址个数为(　　)

A、32，8　　B、32，6

C、8，32　　D、8，30

2011年的一道考研题：

在子网192.168.4.0/30中，能接收目的地址为192.168.4.3的IP分组的最大主机数是 (　　)

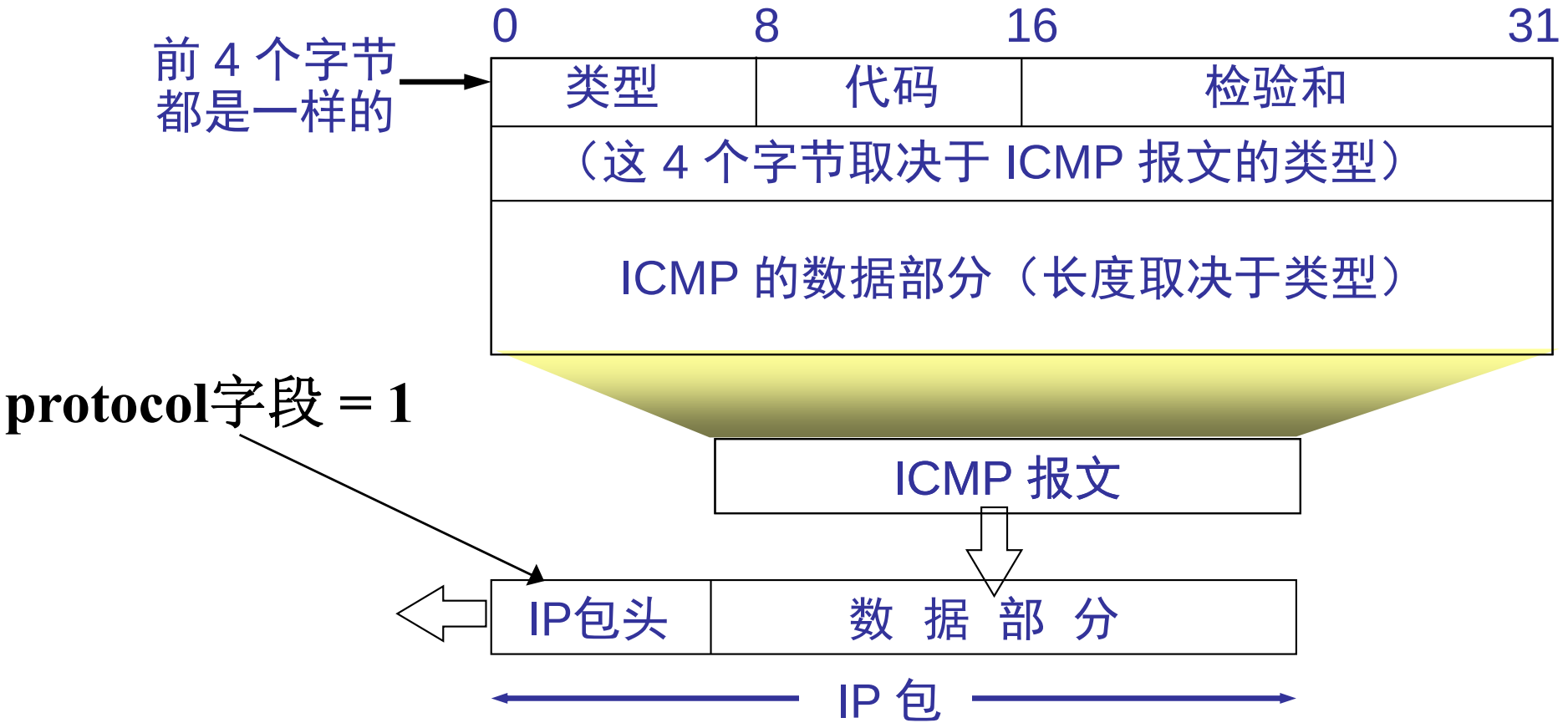
A、0　　B、1　　C、2　　D、4

4.4 网际控制报文协议 ICMP

4.4 网际控制报文协议 ICMP

一、ICMP简介

- **RFC 792: Internet Control Message Protocol**
- **主要用于报告出错和测试等控制信息**
- **ICMP位于IP层，ICMP报文是封装在IP包中传输的**



4.4 网际控制报文协议 ICMP

二、ICMP报文类型

- ICMP 报文有9种，可分为差错报告报文和询问报文两类

	Message type	Description
差错报告	Destination unreachable	Packet could not be delivered
	Time exceeded	Time to live field hit 0
	Parameter problem	Invalid header field
	Source quench	Choke packet
	Redirect	Teach a router about geography
询问	Echo request	Ask a machine if it is alive
	Echo reply	Yes, I am alive
	Timestamp request	Same as Echo request, but with timestamp
	Timestamp reply	Same as Echo reply, but with timestamp

4.4 网际控制报文协议 ICMP

二、ICMP报文类型

- 差错报告报文
  - **Destination unreachable**：终点不可达，路由器或主机无法传输报文时向源主机发送此报文
  - **Source quench**：源点抑制，路由器或主机由于拥塞丢弃报文时，向源主机发送此报文，使其放慢发送速度
  - **Time exceeded**：超时，路由器收到TTL字段为0的报文时，向源主机发送此报文
  - **Parameter problem**：参数问题，路由器或主机收到的报文中头部有非法字段时，丢弃数据包，并向源主机发送此报文
  - **Redirect**：重定向，路由器向主机发送此报文告知路由改变，主机下次发送数据报给另外的路由器
- 询问报文
  - **Echo request / reply**：回声探测，用于测试网络连通性
  - **Timestamp request / reply**：请求时间，可用于时间同步

4.3 划分子网和构造超网

2010年的一道考研题：

若路由器R因为拥塞丢弃IP分组，则此时R可向发出该IP分组的源主机的ICMP报文的类型是(　)

A.路由重定向　　B. 目的不可达

√ C. 源抑制　　D. 超时

三、ICMP应用举例(1/2)

- **Ping**
  - 用来测试两个主机之间的连通性
  - 采用**ICMP echo request / reply**报文
    - 向目的主机发送**ICMP echo request**报文，对方收到后会回应**ICMP echo reply**，根据能否收到应答后判断两台主机之间是否连通

```
D:\>ping news.sina.com.cn

Pinging hydra.sina.com.cn [218.30.108.63] with 32 bytes of data:

Reply from 218.30.108.63: bytes=32 time=11ms TTL=54
Reply from 218.30.108.63: bytes=32 time=12ms TTL=54
Reply from 218.30.108.63: bytes=32 time=12ms TTL=54
Reply from 218.30.108.63: bytes=32 time=12ms TTL=54

Ping statistics for 218.30.108.63:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 12ms, Average = 11ms
```

三、ICMP应用举例(2/2)

- **Traceroute / Tracert**
  - 用来测试到另一台主机所经过的路由信息
  - 采用ICMP超时报告报文
    - 逐个发出UDP报文，其IP包头中的TTL字段分别设为1, 2, 3, …，直到到达目的主机
    - 报文路由路径上的路由器会返回ICMP超时报文，从该报文即可得知路由器IP地址

```
D:\>tracert -d news.sina.com.cn

Tracing route to hydra.sina.com.cn [218.30.108.67]
over a maximum of 30 hops:

  1     2 ms    <1 ms    <1 ms  192.168.1.1
  2    12 ms     9 ms    13 ms  219.143.128.1
  3    11 ms     9 ms     9 ms  219.141.142.29
  4    11 ms    10 ms    10 ms  219.141.131.13
  5    11 ms    10 ms     9 ms  219.141.130.110
  6    12 ms    11 ms    10 ms  219.142.1.70
  7    12 ms    10 ms    21 ms  220.181.16.130
  8    16 ms    15 ms    14 ms  218.30.25.237
  9    14 ms    11 ms    12 ms  218.30.28.70
 10    12 ms    10 ms    10 ms  218.30.104.42
 11    11 ms    10 ms    10 ms  218.30.108.67

Trace complete.
```

4.5 路由算法及协议

Routing： 路由

Router： 路由器

Routing table ： 路由表

4.5 路由算法及协议

一、路由算法简介(1/4)

- 在网络中，路由器依据路由信息(路由表)转发分组，路由信息是路由协议生成的，路由算法是路由协议的基础和核心，注意：
  - 路由协议是用来生成路由信息(路由表)的，不是转发分组的
- 理想的路由算法应具备的特性
  - ① 必须是正确的和完整的：按照得出的路由能够进行正确寻址
  - ② 在计算上应简单：不增加过多开销
  - ③ 有自适应性：能根据通信量和网络拓扑的变化调整路由
  - ④ 应具有稳定性：通信量和拓扑稳定时，能快速收敛
  - ⑤ 应是公平的：对所有用户公平
  - ⑥ 应是最佳的：能找出最好的路由(时延最小、吞吐量最大)
- 不同情况下的路由需求可能各有侧重，且网络环境不断变化，因此不存在一种绝对的最佳路由算法

4.5 路由算法及协议

一、路由算法简介(2/4)

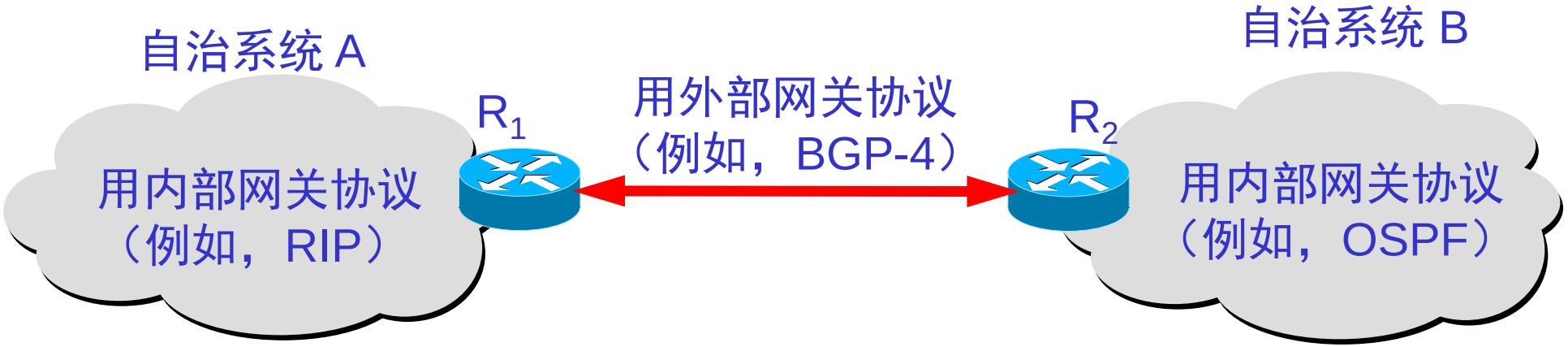
- 静态路由与动态路由
  - 静态路由选择策略
    - 非自适应路由选择
    - 简单和开销较小，但不能及时适应网络状态的变化
    - 适用于小规模且变化较少的网络，由人工设置路由
  - 动态路由选择策略
    - 自适应路由选择
    - 能较好地适应网络状态的变化，但实现起来较为复杂，开销也比较大
    - 适用于较大规模、频繁变化的网络，通过专门的算法和协议进行路由的计算
- 两类典型的动态路由算法
  - **距离向量(distance vector)路由算法**
  - **链路状态(link state)路由算法**

4.5 路由算法及协议

一、路由算法简介(3/4)

- Internet采用分层次的路由，原因：
  - Internet规模庞大，任何一台路由器都不可能获取和存储整个Internet的路由信息
  - 接入Internet的众多网络(管理方)不希望外界了解自己网络的内部信息
- 自治系统(AS—Autonomous System)的概念
  - 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
  - 一个 AS可使用多种内部路由选择协议和度量，但对其他 AS 表现出的是单一的和一致的路由选择策略

- 在自治系统背景下，**Internet**路由协议可以分为两类：
  - 内部网关协议 **IGP(Interior Gateway Protocol)**
    - 自治系统内部使用的路由选择协议
    - 这类路由协议使用得最多，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，当数据报传到一个自治系统的边界时，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，这样的协议就是外部网关协议 **EGP**。
    - 应用最为广泛的外部网关协议：**BGP-4**
- 自治系统之间的路由选择又称为域间路由选择**(interdomain routing)**
- 自治系统内部的路由选择又称为域内路由选择**(intradomain routing)**



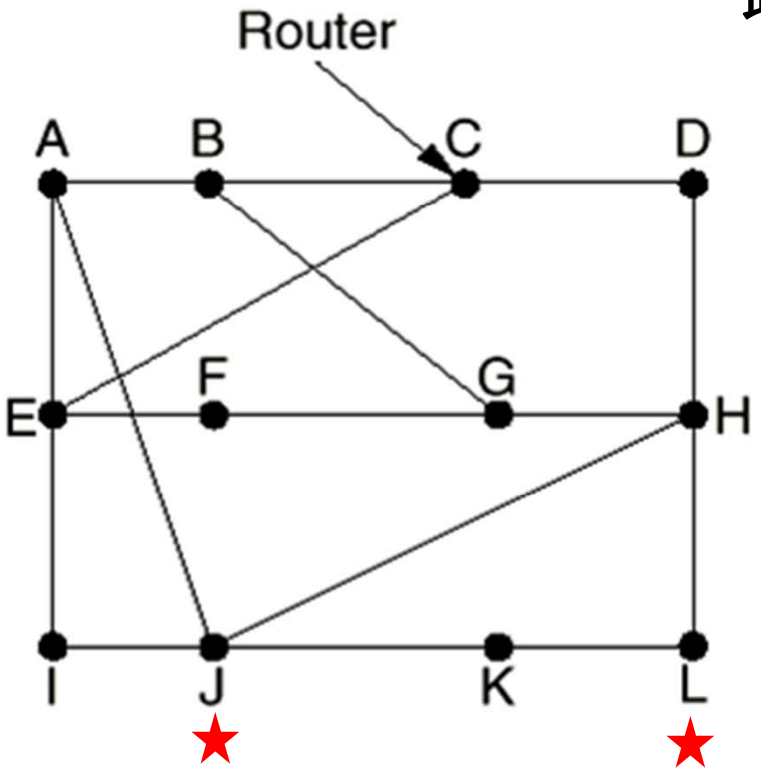
4.5 路由算法及协议

问：如何衡量距离？

二、距离向量路由与RIP协议(1/7)

- 距离向量路由(distance vector routing)简介
  - 属于动态路由算法
  - 也称为Bellman-Ford路由算法和Ford-Fulkerson算法
  - 最初用于ARPANET，被RIP协议采用
- 距离向量路由的基本思想
  - 每个路由器维护一张表，表中给出了到每个目的地的已知最佳距离和线路，并通过与相邻路由器交换距离信息来更新表
  - 路由器周期性地向所有相邻路由器发送它的距离表，同时它也接收每个邻居结点发来的距离表；
  - 相邻路由器*X*发来的表中，*X*到路由器*i*的距离为*Xi*，本路由器到*X*的距离为*m*，则路由器经过*X*到*i*的距离为*Xi* + *m*。根据不同邻居发来的信息，计算*Xi* + *m*，并取最小值，更新本路由器的路由表。

距离向量路由示例



J如何计算到L的距离？

JA+AL = 8 + 29 = 37

JI +IL = 10 + 33 = 43

JH+HL = 12 + 9 = 21

JK+KL = 6 + 9 = 15

J到L的下一跳为K，距离为15

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

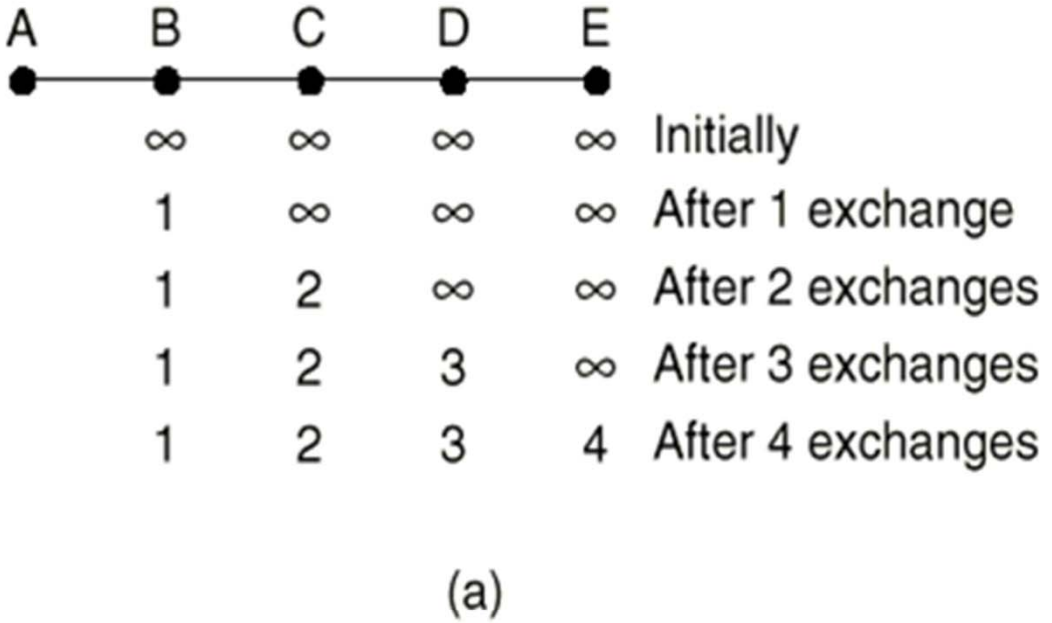
Vectors received from J's four neighbors

New routing table for J

结点J从相邻结点收到的距离向量

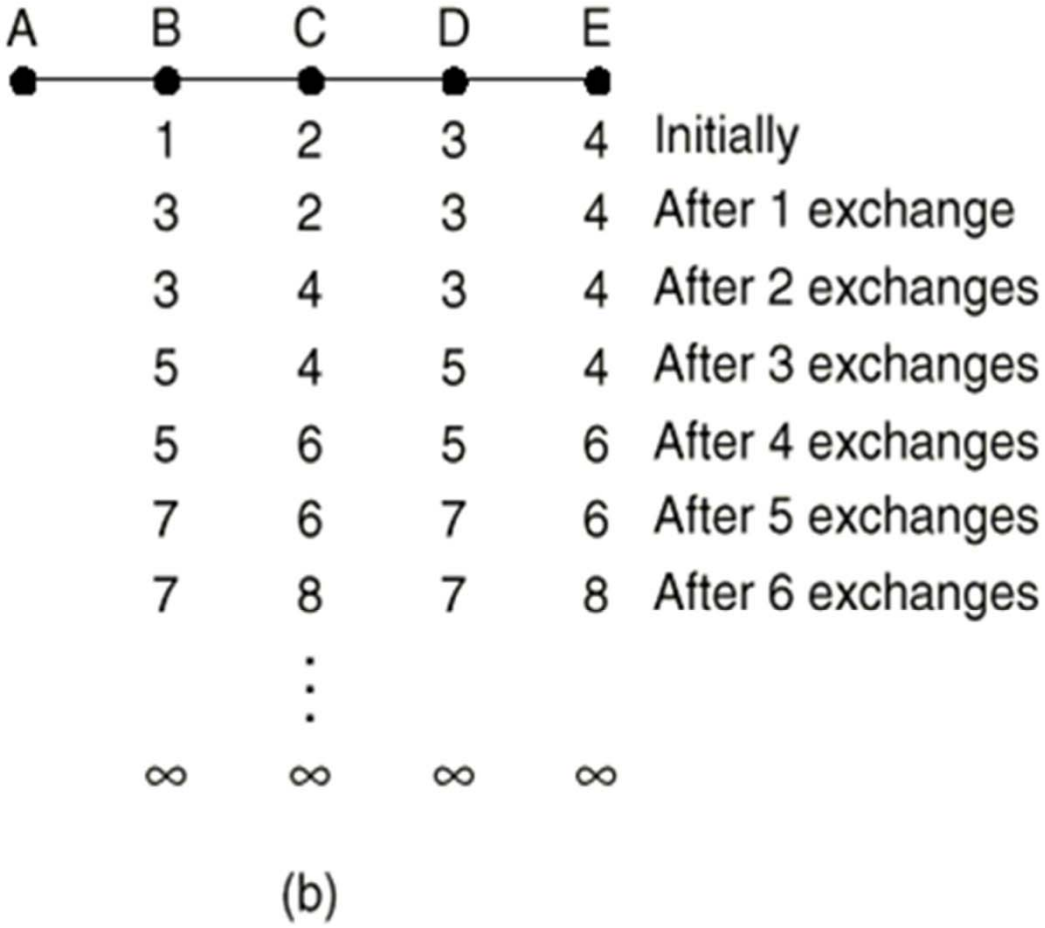
二、距离向量路由与RIP协议(3/7)

- 距离向量路由的无穷计算问题
  - 算法的缺陷：对好消息反应迅速，对坏消息反应迟钝



(a)

A由停机→开机时



(b)

A由开机→停机时

4.5 路由算法及协议

二、距离向量路由与RIP协议(4/7)

- **RIP(Routing Information Protocol)协议**
  - 属于距离向量路由协议
  - **RFC1058: Routing Information Protocol**
  - 特点：简单，适用于小规模网络中的路由
  - **RIP中的距离定义：路由器跳数(hop count)**
    - 优点：便于计算；缺点：？
    - 最长距离**15**，即允许的最长路径中最多包含**15**个路由器
      - 距离为**16**表示不可达
    - 直接连接的距离为**0**
  - **RIP的三个要点**
    - 仅和相邻路由器交换信息
    - 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
    - 按固定的时间间隔交换路由信息(典型值：每隔 **30** 秒)

4.5 路由算法及协议

2010年的一道考研题：

某自治系统采用RIP协议，若该自治系统内的路由器R1收到其邻居路由器R2的距离矢量中包含的信息<net1，16>，则可能得出的结论是：（　　）

A. R2可以经过R1到达net1，跳数为17

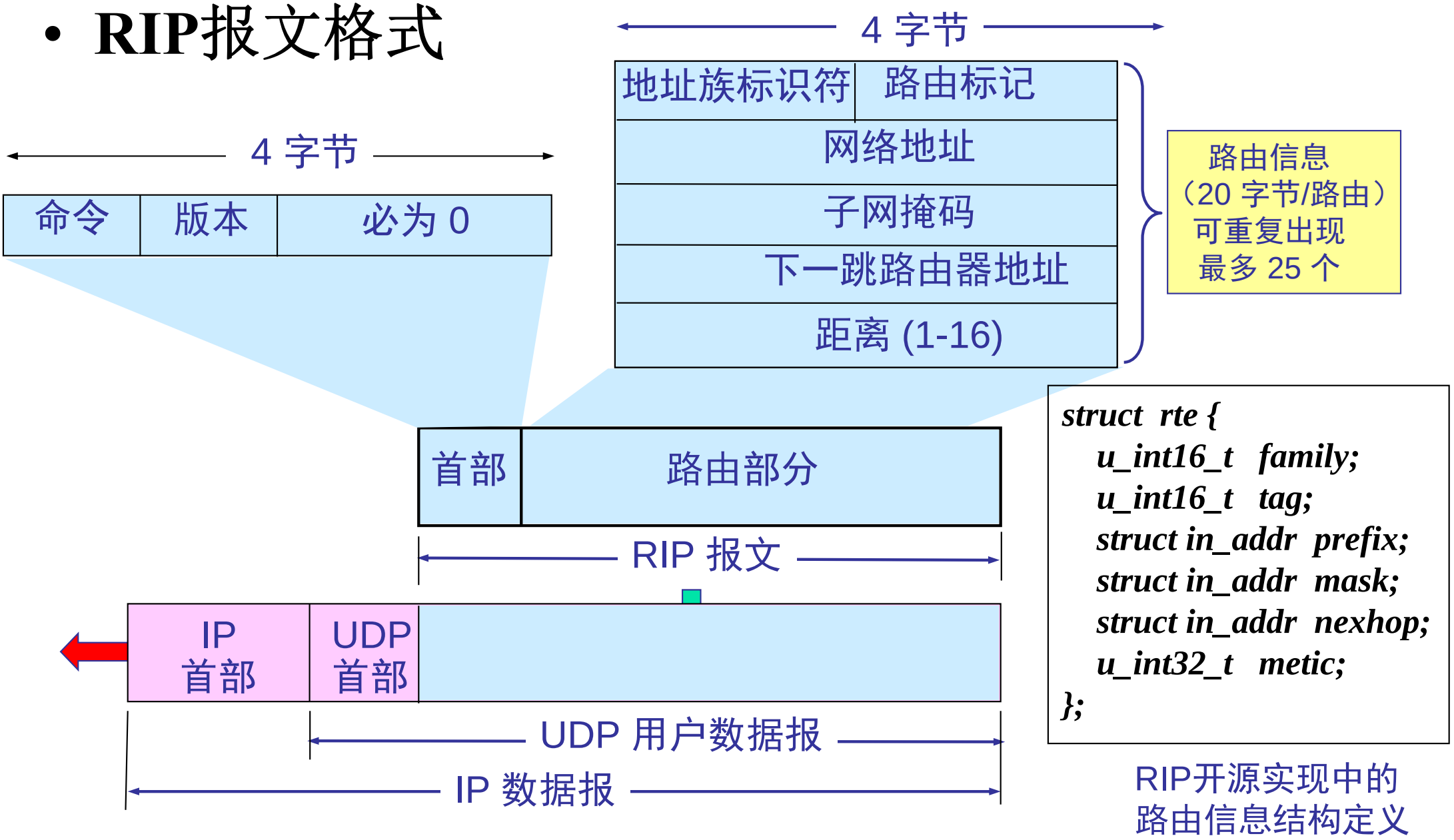
B. R2可以到达net1，跳数为16

C. R1可以经过R2到达net1，跳数为17

D. R1不能经过R2到达net1

4.5 路由算法及协议

- RIP报文格式




```
struct  rte {
    u_int16_t   family;
    u_int16_t   tag;
    struct in_addr  prefix;
    struct in_addr  mask;
    struct in_addr  nexhop;
    u_int32_t   metic;
};
```

RIP开源实现中的
路由信息结构定义

收到相邻路由器发来RIP报文后的处理流程

(注意它是按照距离向量算法工作的)

收到相邻路由器(其地址为 *X*)的一个 RIP 报文：

(1) 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 *X*，并把所有的“距离”字段的值加 1。每个项目有三个关键数据：目的网络*N*、距离*d*、下一跳路由器*X*。

(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若路由表中没有目的网络*N*，则把该项目加到路由表中。

否则

若下一跳字段给出的路由器地址是*X*，则用收到的项目替换原有项目。

否则

若收到项目中的距离*d*小于路由表中的距离，则进行更新，

否则，什么也不做。

(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为16（距离为16表示不可达）。

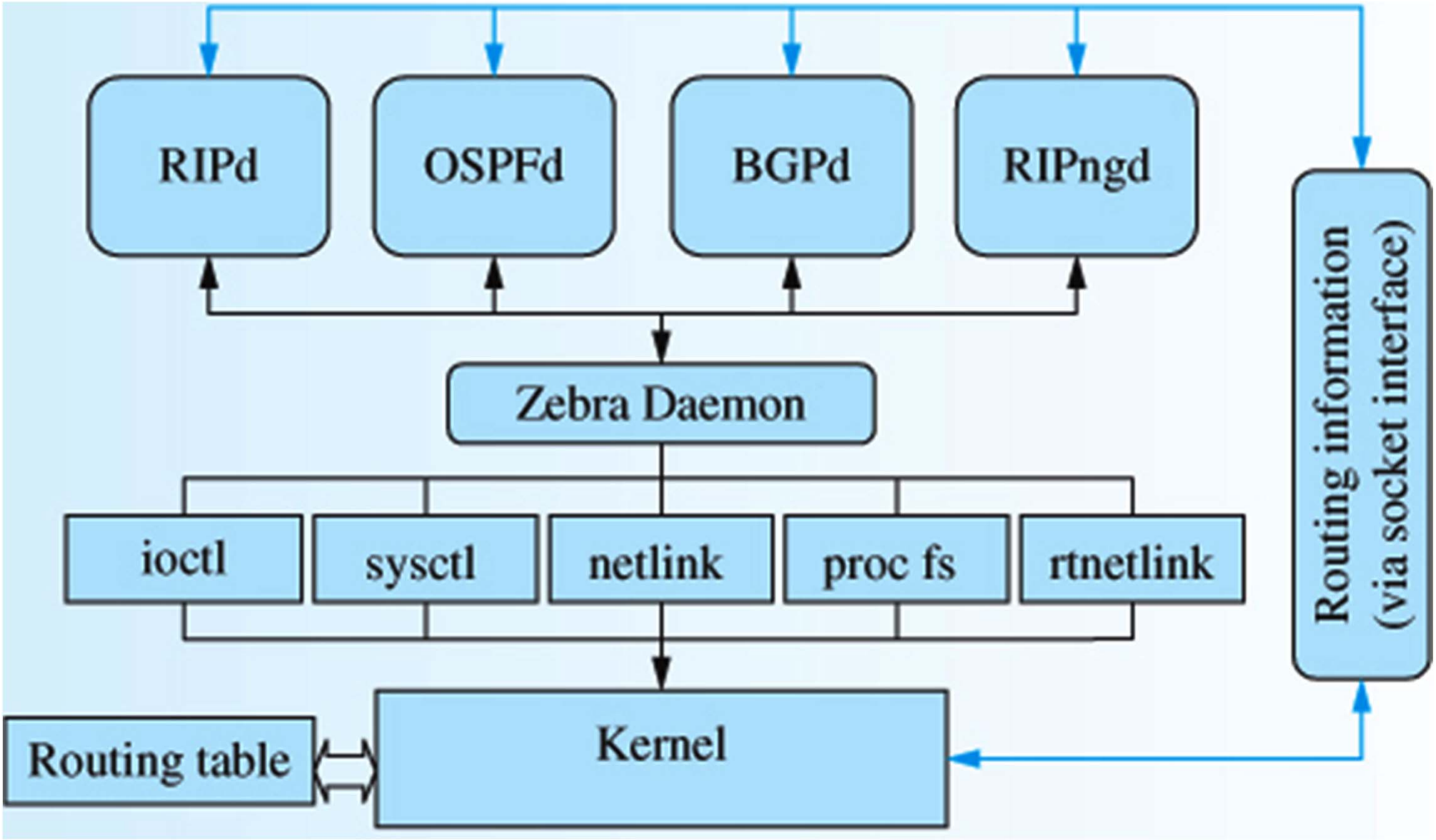
(4) 返回。

问：对前面的无穷计算实例，按交换周期30秒、3分钟超时计算，何时变为不可达？

二、距离向量路由与RIP协议(7/7)

- **RIP优缺点**
  - 优点
    - 实现简单，开销较小
  - 缺点
    - 当网络出现故障时，要经过比较长的时间才能将此信息传送到所有的路由器(距离向量路由的固有缺点)
    - 支持的网络规模有限，最大距离为 **15(16** 表示不可达)
    - 路由器之间交换的路由信息是路由器中的完整路由表，随着网络规模的扩大，开销随之增加

开源路由软件Zebra的架构



4.5 路由算法及协议

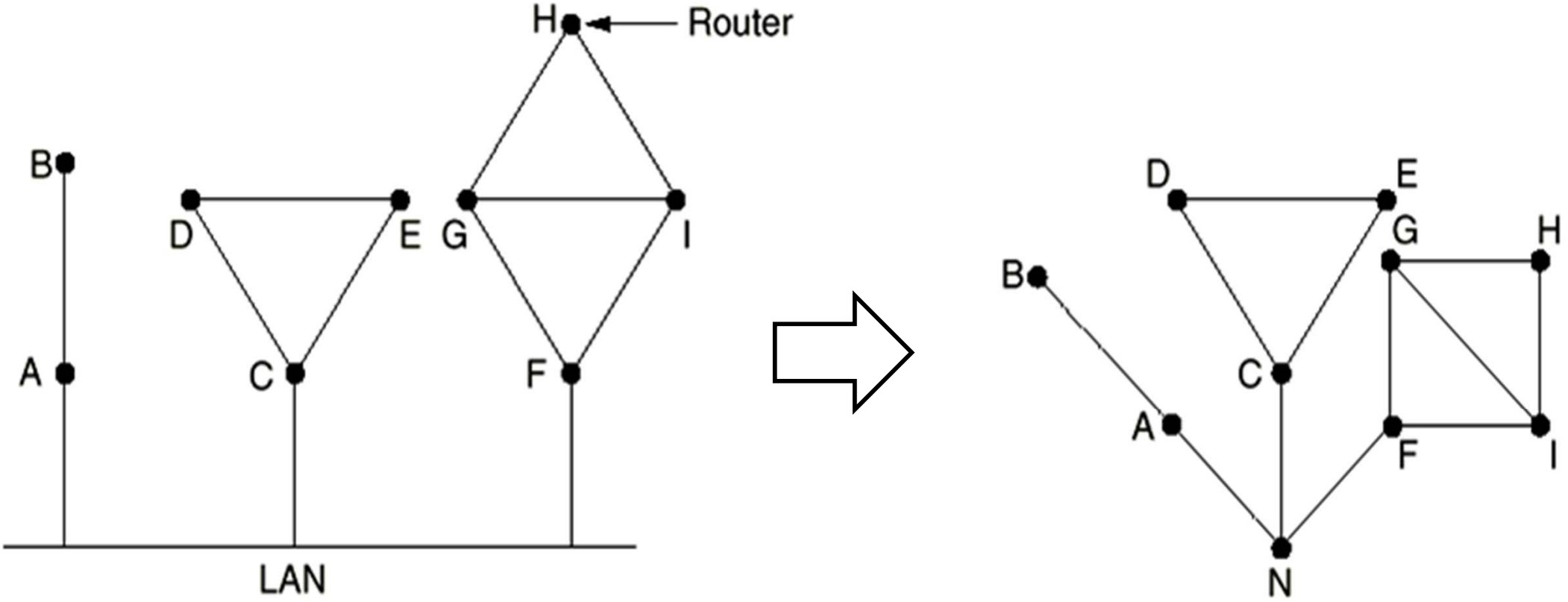
三、链路状态路由与OSPF协议(1/7)

- 链路状态路由(link-state routing)的基本思想
  - 每个路由器完成5步工作:
    ① 发现它的邻居结点，并学习其网络地址
    ② 测量到各邻居结点的延迟或开销
    ③ 构造一个分组，其中包含所有它刚刚知道的信息
    ④ 将这个分组发送给其他所有路由器
    ⑤ 计算出到每一个其他路由器的最短路径
  - 各路由器之间动态交换链路状态信息，每个路由器都建立一个链路状态数据库
    - “链路状态”即本路由器都和哪些路由器相邻，以及该链路的“度量”(metric)
  - 链路状态数据库实际上是全网的拓扑结构图，它在全网范围内是一致的(即链路状态数据库的同步)
  - 各路由器根据网络拓扑使用Dijkstra算法计算从本路由器到其他结点的最佳路径，构成路由表

4.5 路由算法及协议

三、链路状态路由与OSPF协议(2/7)

- 发现邻居结点，并学习它们的网络地址
  - 路由器启动后，通过发送HELLO包发现邻居结点
  - 两个或多个路由器连在一个LAN时，引入人工结点



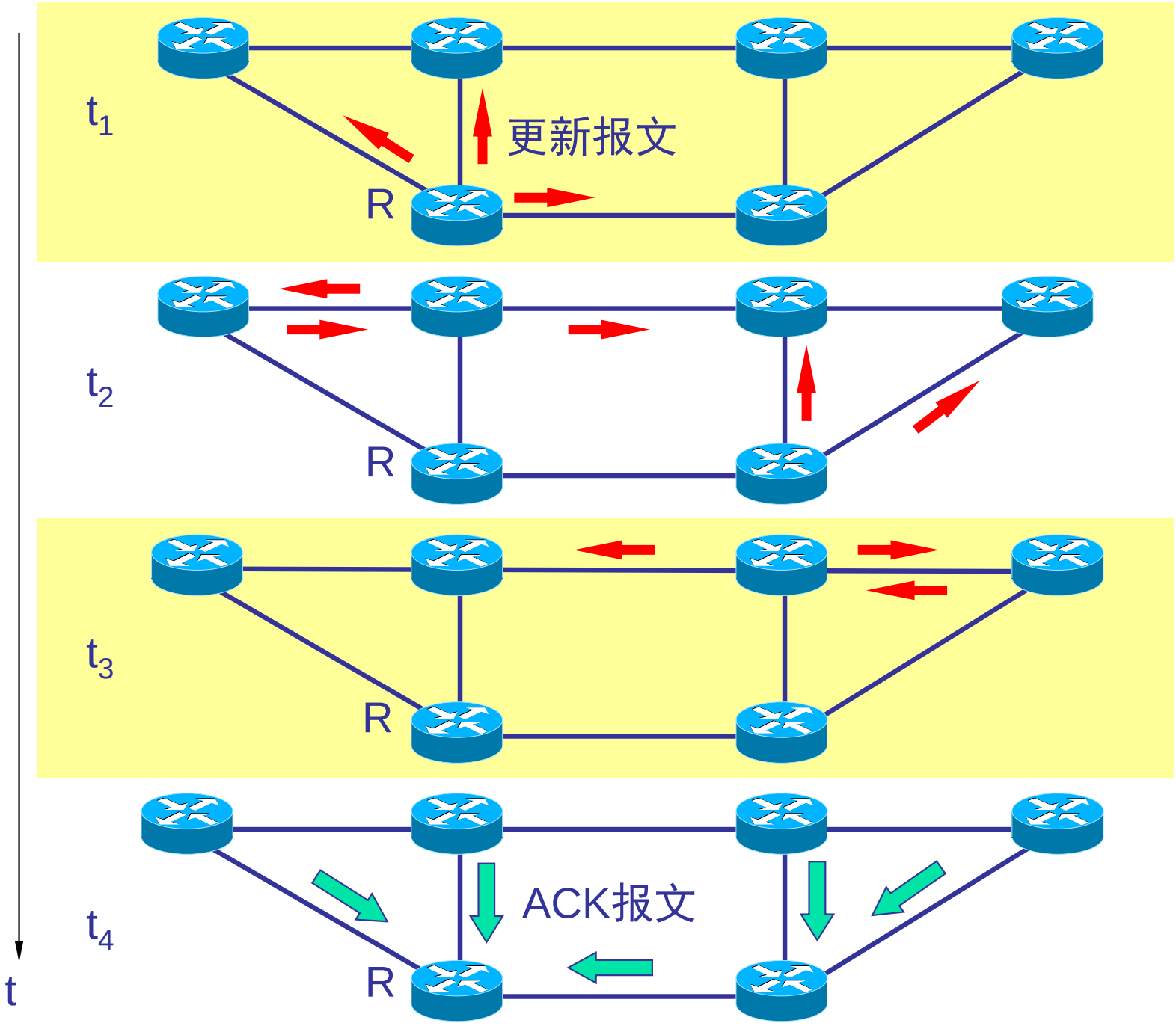
多个路由器连接到同一局域网时的拓扑转换

三、链路状态路由与OSPF协议(3/7)

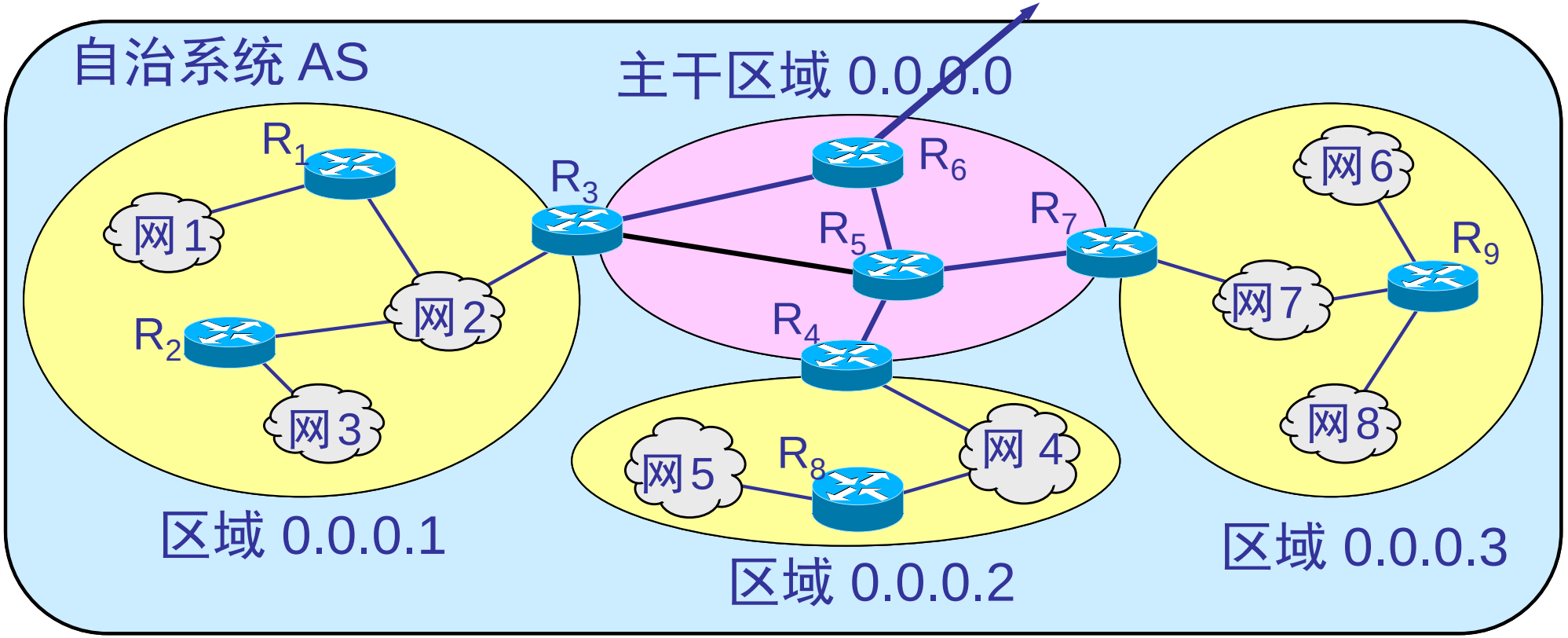
- **OSPF(Open Shortest Path First)协议简介**
  - 属于链路状态路由协议
  - **RFC2328: OSPF Version 2**
  - 优点：
    - 收敛速度快
      - 当链路状态发生变化时，结点会以洪泛**(flooding)**方式告知所有网络中其他所有结点
    - 适用于较大规模的网络
      - 仅在链路状态发生变化时发送洪泛信息，不会产生很大通信量

洪泛**(flooding):** 结点收到分组时，向除输入链路外的所有其他链路发送出去

OSPF洪泛发送

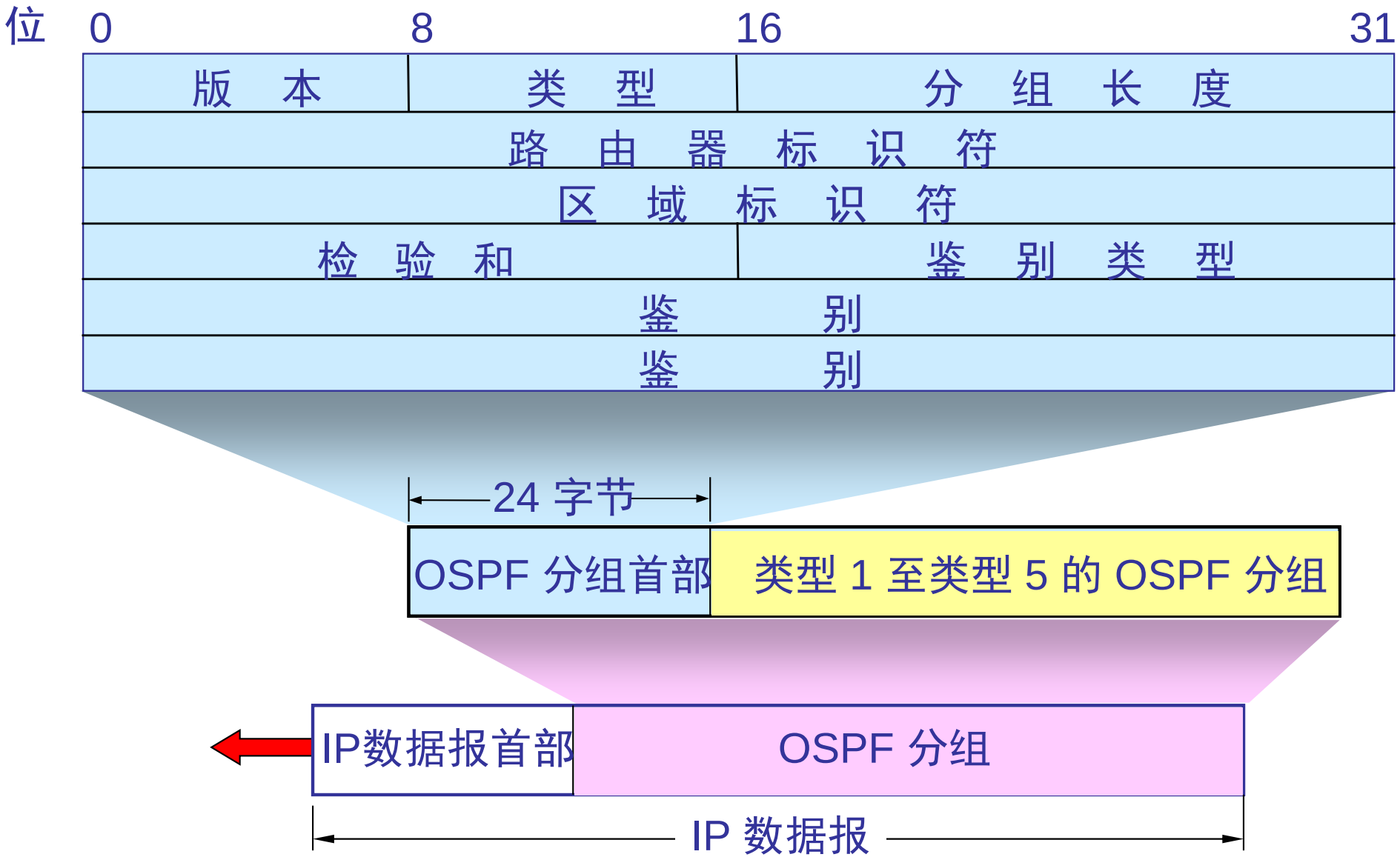


- 为了管理大规模的网络，**OSPF** 将一个自治系统再划分为若干个区域**(area)**
  - 每一个区域都有一个 **32** 位的区域标识符**(**用点分十进制表示**)**
- 划分区域的好处：利用洪泛法交换链路状态信息的范围局限于一个区域而不是整个的自治系统，减少了整个网络上的通信量
- 一个区域内的路由器只知道本区域的完整网络拓扑，而不知道其他区域的网络拓扑
- 层次结构的区域划分，上层的区域称为主干区域**(backbone area)**，主干区域的作用是用来连通其他在下层的区域
  - 主干区域的标识符规定为**0.0.0.0**。



OSPF分组结构

(OSPF 不用 UDP 而是直接用 IP 数据报传送)

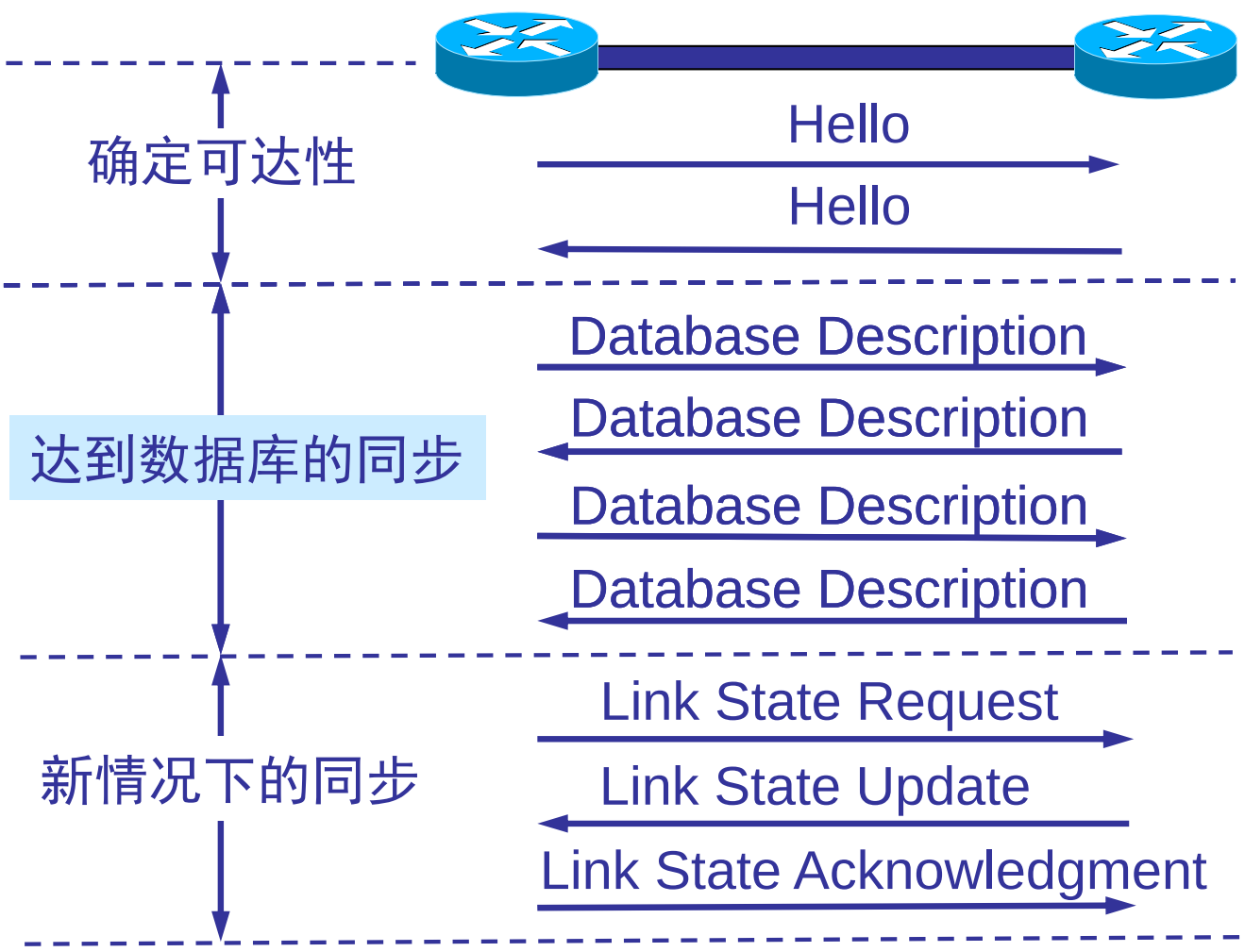


4.5 路由算法及协议

OSPF的5种分组

- 类型1：Hello分组
- 类型2：Database Description分组
- 类型3：Link State Request分组
- 类型4：Link State Update分组，用洪泛法对全网更新链路状态
- 类型5：Link State Acknowledgment分组

OSPF操作



4.5 路由算法及协议

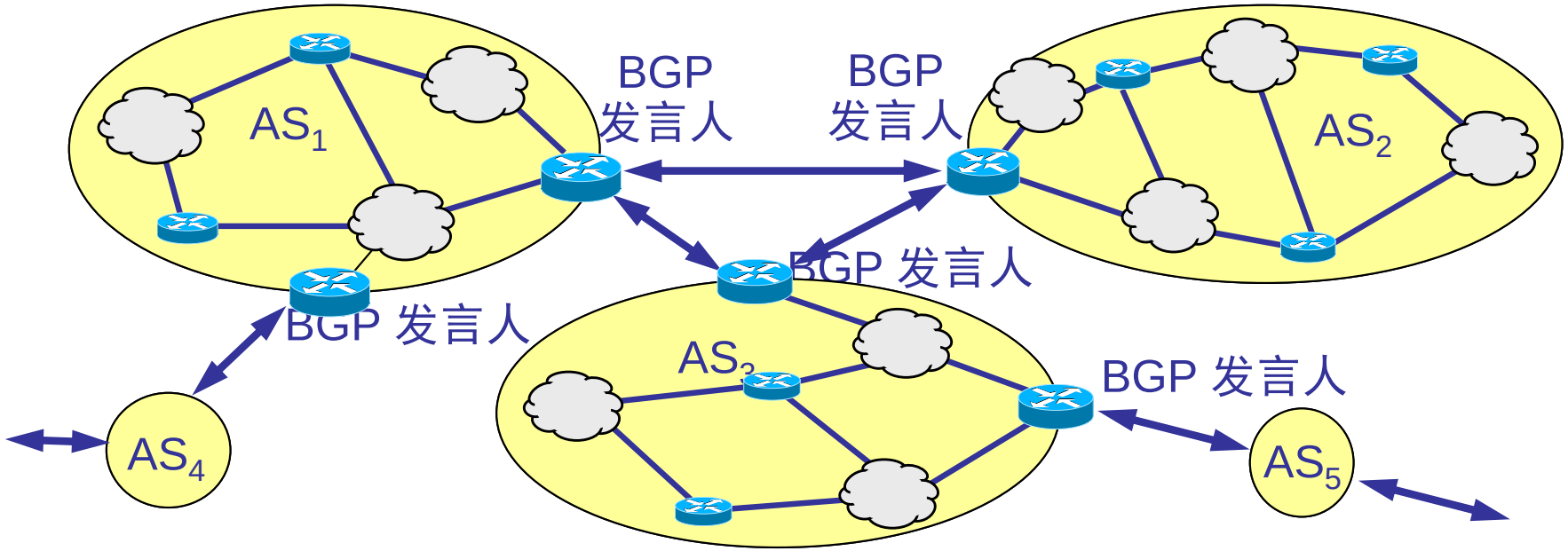
亚太网络信息中心2009年7月统计：
- IPv4路由器可见的AS为3.2万个，平均每月新增200多个
- 路由表58.7万项；转发表29.9万项

四、外部网关路由协议BGP

- BGP(Border Gateway Protocol)简介
  - BGP是不同自治系统的路由器之间交换路由信息的协议
  - BGP-4， RFC4271～4278，2006年
- BGP使用环境的特殊性
  - Internet的规模很大，AS之间的路由选择非常困难
    - Internet主干网路由器中的网络前缀达到数万条
    - 对于AS之间的路由选择，要寻找最佳路由是不现实的
    - 不同AS对路径度量标准各不相同
  - AS之间的路由选择需考虑多种策略
    - 例1：AS1到AS2的数据报经过AS3在技术上最佳，但AS3可能不同意
    - 例2：出于国家安全的考虑，可能不希望数据报流经某些国家
- BGP难以找到最佳路由，只能寻找一条能够到达目的网络且比较好的路由(不能兜圈子)

4.5 路由算法及协议

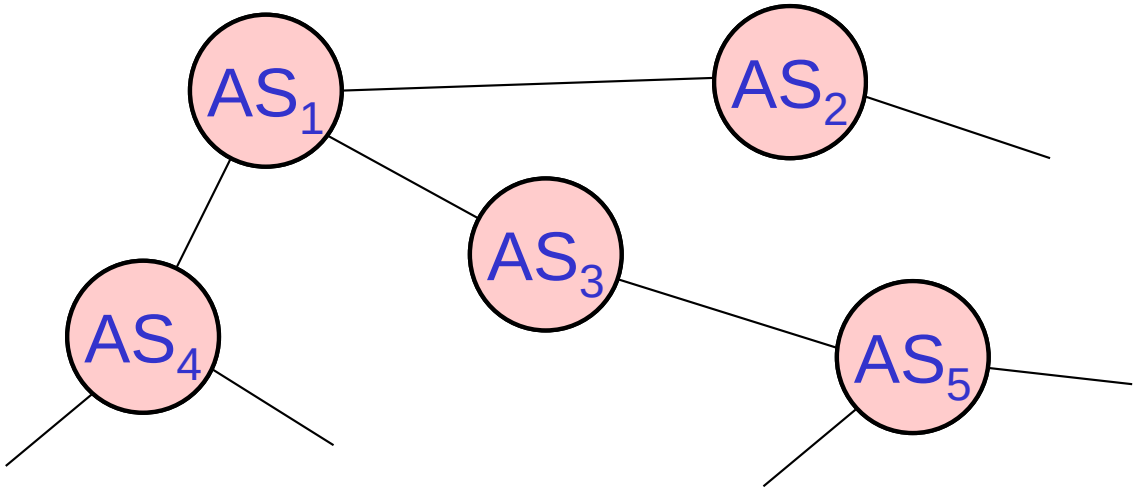
- **BGP发言人(speaker)**
  - 每个AS要选择至少一个路由器作为“BGP speaker”
  - 两个BGP发言人通过一个共享网络连接在一起
  - BGP发言人一般是AS的边界路由器，但也可以不是
  - BGP 发言人与其他自治系统中的BGP发言人交换路由信息
  - 路由信息的交换通过TCP协议进行，两个发言人在TCP连接上交换BGP报文以建立BGP会话(session)，利用 BGP 会话交换路由信息
  - BGP发言人除了运行BGP协议外，还要运行AS的内部网关路由协议



4.5 路由算法及协议

四、外部网关路由协议BGP

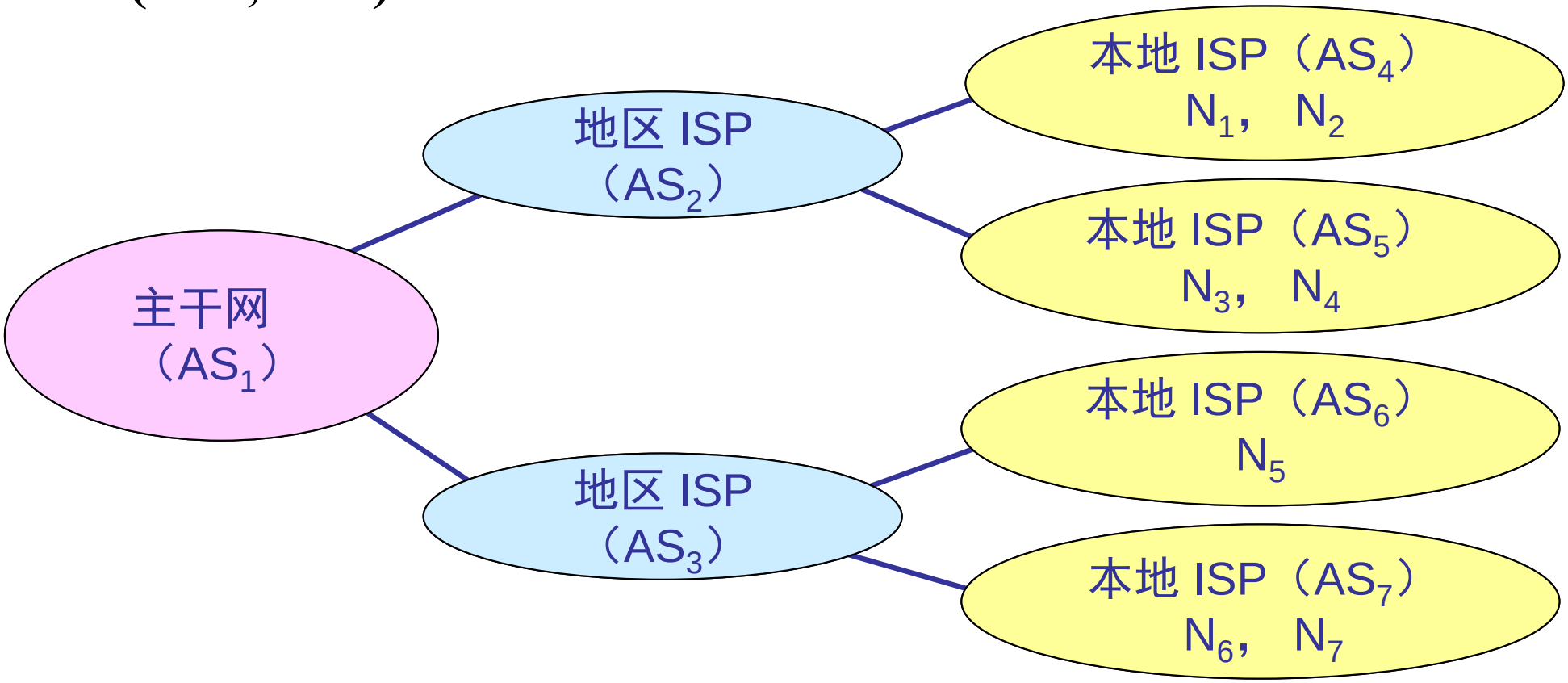
- 信息交换与路由生成
  - **BGP**所交换的网络可达性的信息就是要到达某个网络所要经过的一系列 **AS**
    - 称为路径向量**(path vector)**
  - **BGP** 发言人互相交换了网络可达性的信息后，就根据所采用的策略从其中找出到达各 **AS** 的较好路由



根据路由信息计算得到的连通图示例(注意无回路)

4.5 路由算法及协议

- 示例
  - **AS2的BGP发言人通知主干网的BGP发言人：要到达网络 N1, N2, N3 和 N4 可经过 AS2**
  - 主干网还可发出通知：“要到达网络 **N5, N6** 和 **N7** 可沿路径 **(AS1, AS3)**



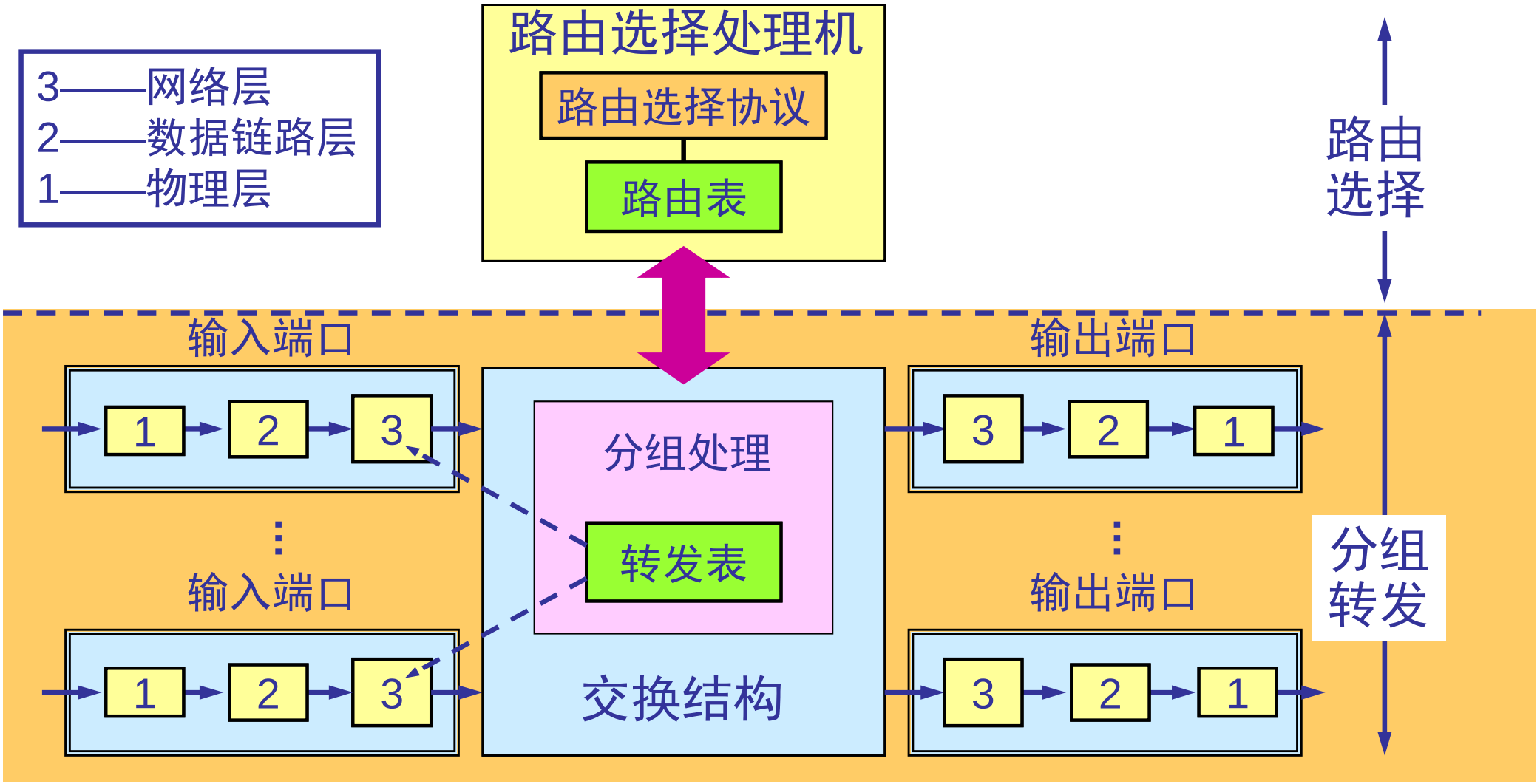
4.5 路由算法及协议

- BGP-4使用四种报文
  - ① *OPEN*报文：用来与相邻的另一个BGP发言人建立关系
  - ② *UPDATE*报文：用来发送某一路由的信息，以及列出要撤消的多条路由
  - ③ *KEEPALIVE*报文：用来确认打开报文和周期性地证实邻站关系
  - ④ *NOTIFICATION*报文：用来发送检测到的差错
  - BGP发言人通过*OPEN*报文与另一发言人建立会话，对方如同意则回应*KEEPALIVE*报文
  - 双方周期性交换*KEEPALIVE*报文，以确认会话未中断
  - 路由信息变化时，发送*UPDATE*报文
- BGP协议特点
  - 交换路由信息的结点数量级与自治系统个数相当，大大少于这些自治系统中的网络数
  - 每个自治系统中BGP发言人(或边界路由器)的数目很少，使得自治系统间的路由选择不致过分复杂
  - BGP支持CIDR，因此BGP路由表包括目的网络前缀、下一跳路由器，以及到达该目的网络要经过的各个自治系统序列
  - 在BGP刚运行时，邻站交换完整的 BGP 路由表，以后只需要在发生变化时更新变动部分，有利于节省网络带宽和减少处理开销

五、路由器(Router)

- 具有多个输入/输出端口的专用计算机，其任务是转发分组

路由器组成
- ① 路由选择部分：按照路由选择协议工作，构建路由表
- ② 分组转发部分：交换结构(switching fabric) + 一组输入/端口
  - 交换结构根据转发表进行分组的快速转发，转发表从路由表得到



4.5 路由算法及协议

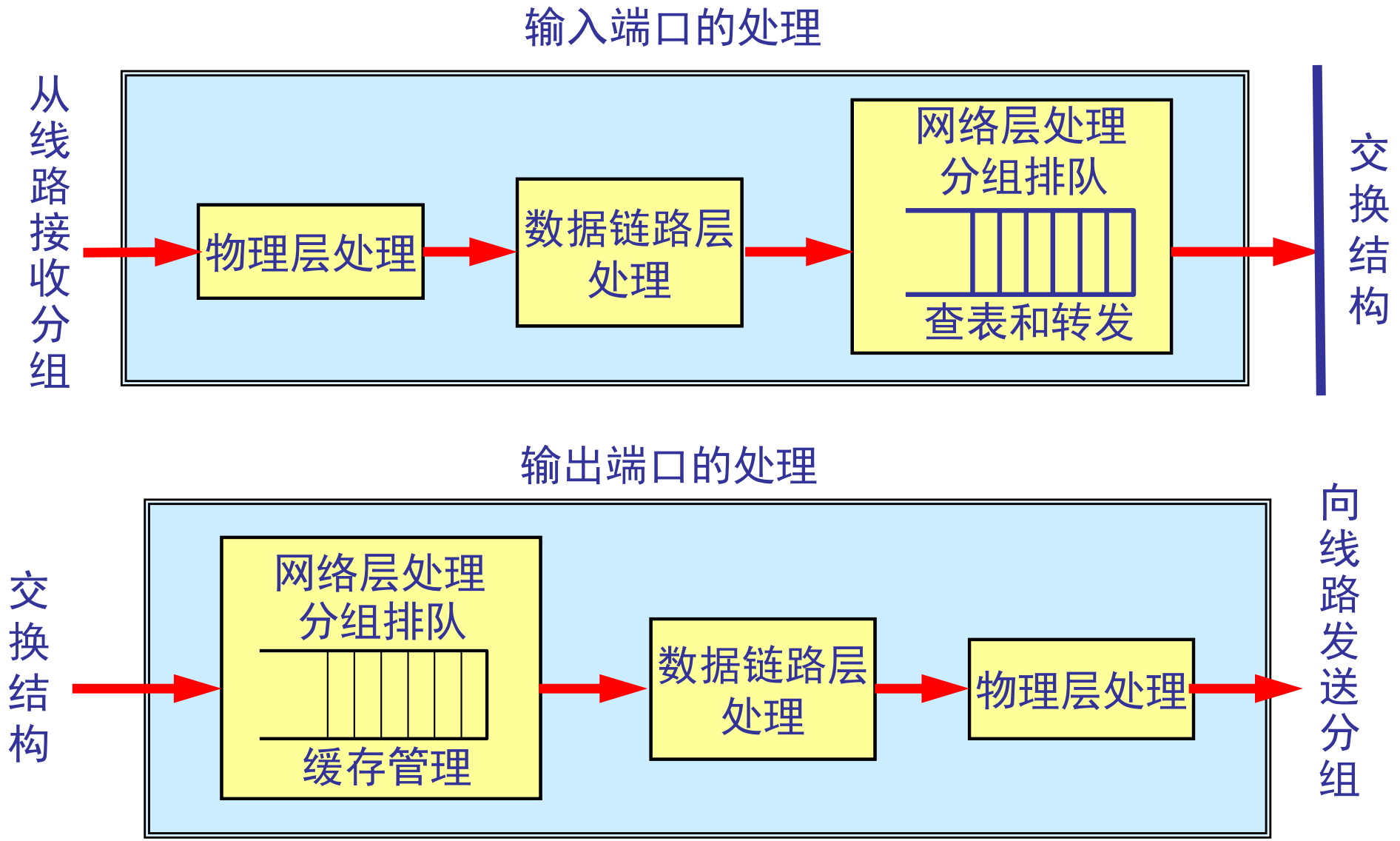
五、路由器(Router)

- 路由器设计的关键之一是分组的高速转发
  - 决定报文转发速率的因素有：
    - 输入/输出端口的处理性能
    - 交换机构的性能
  - 线速(line speed 或 wire speed): 分组的处理速率能够达到线路上分组的传输速率
  - 衡量路由器性能的指标：pps(packet per second)
  - 如oc-48链路(2.5Gb/s)，如分组长度256字节，则线速转发意味着每秒处理分组数>100万

4.5 路由算法及协议

注意：当排队的分组数量过多时，可能导致路由器内部缓冲区溢出丢包

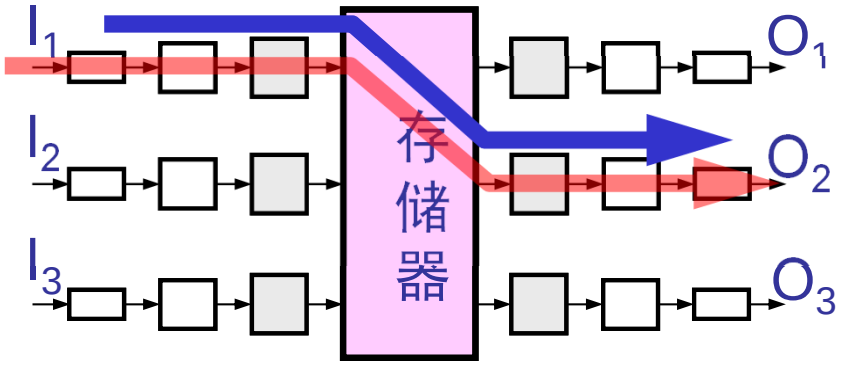
- 输入和输出端口的处理



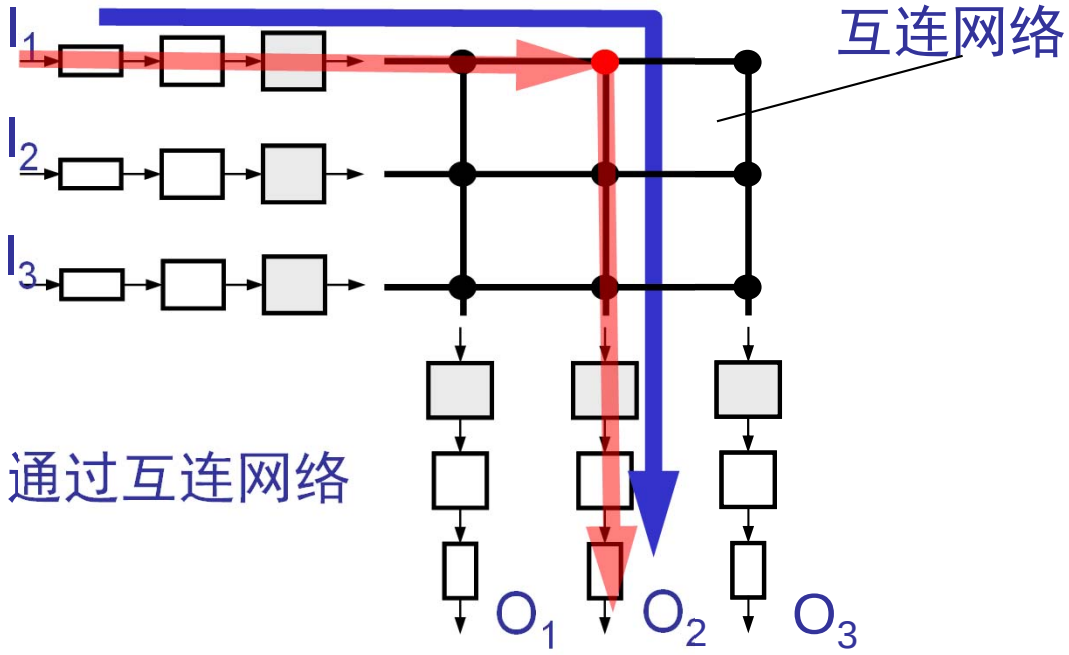
4.5 路由算法及协议

五、路由器(Router)

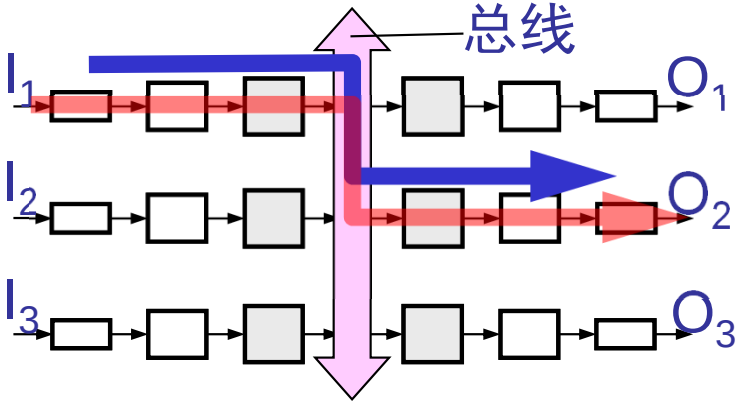
- 交换结构



(a) 通过存储器



(c) 通过互连网络



(b) 通过总线

4.5 路由算法及协议

2011年的一道考研题：

在下列关于IP路由器功能的描述中，正确的是：（　　）

I. 运行路由协议，设置路由表

II. 监测到拥塞时，合理丢弃IP分组

III. 对收到的IP分组头进行差错校验，确保传输的IP分组不丢失

IV. 根据收到的IP分组的目的IP地址，将其转发到合适的输出线路上

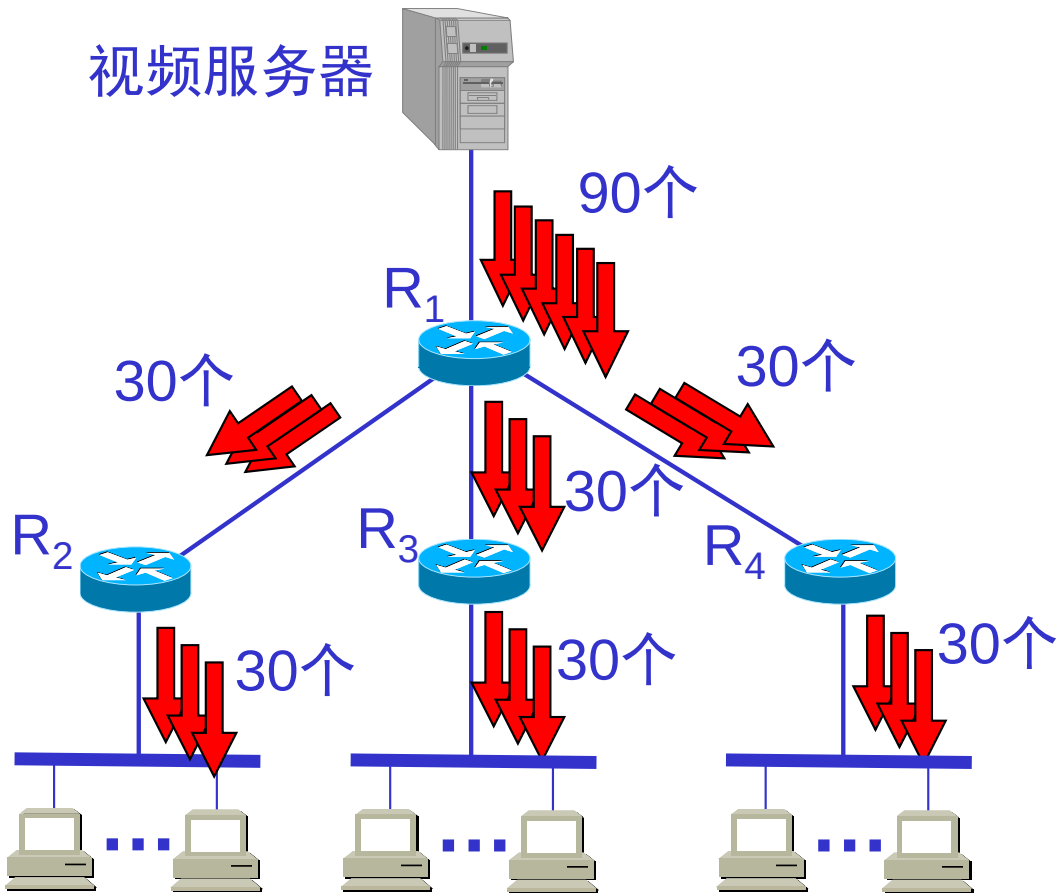
A. 仅III、IV　　B. 仅I、II、III

C. 仅I、II、IV　　D. I、II、III、IV

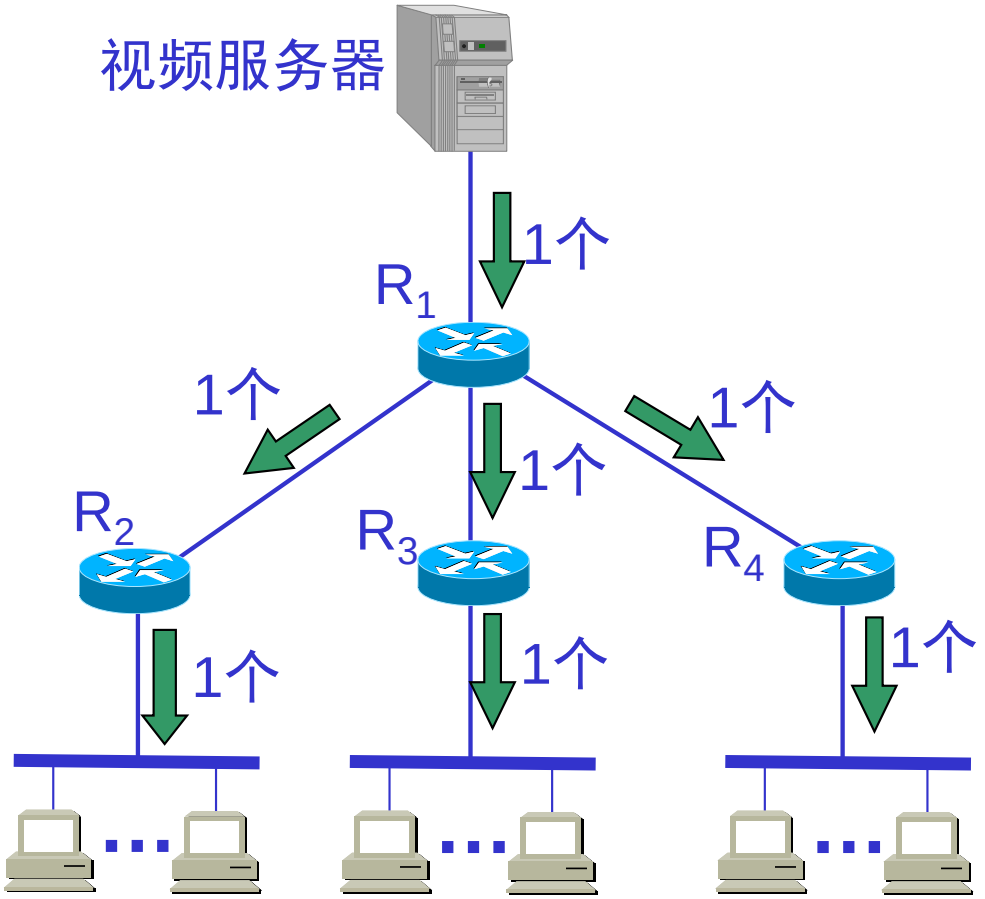
4.6 IP 组播

4.6 IP 组播

- 组播(multicast)又称为多播，用于实现一点对多点的数据传输
- 对于一些网络应用，采用组播可大大减少网络流量
- 组播的典型应用示例：网络视频服务



使用单播技术实现



使用组播技术实现

4.6 IP 组播

- 使用D类IP地址作为组播地址
  - **224.0.0.0—239.255.255.255**
  - 组播数据报：IP包头中的目的地址为D类地址，协议类型为2(IGMP协议)
- 组播地址即为特定组播组的标识符，主机通过加入组播组来接收组播数据
- 组播可分为两种
  - 在局域网中的硬件组播
    - 将MAC地址中的特定地址段作为组播地址，并与IP组播地址形成对应关系
  - 在Internet中的组播
    - 路由器需支持组播，即组播路由器
    - 主机通过IGMP协议与组播路由器通信，加入/退出某个组播组
    - 组播路由器之间通过组播路由协议实现组播数据报的传输

4.7 网络地址转换 NAT和虚拟专用网 VPN

4.7 虚拟专用网 VPN 和网络地址转换 NAT

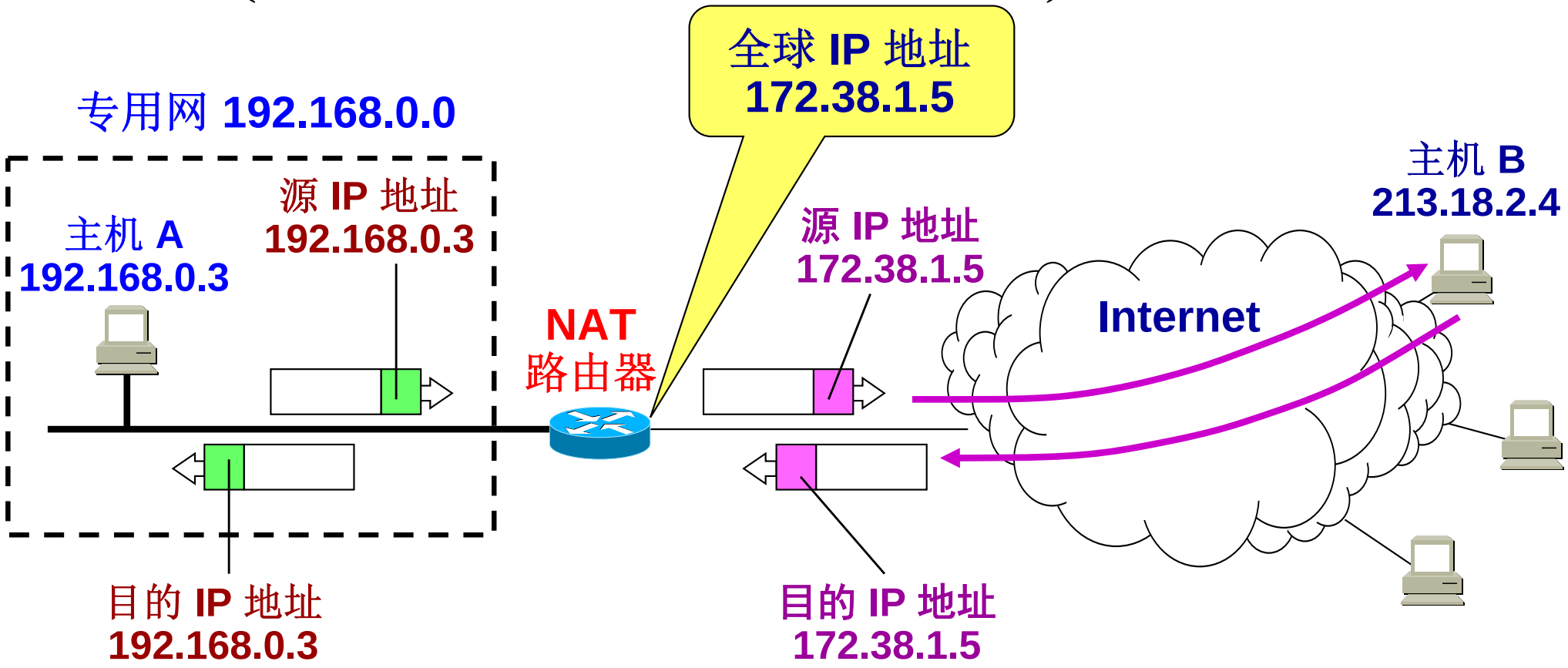
一、网络地址转换 NAT

- 专用地址(保留地址)
  - **RFC1918中定义了一系列专用地址(private address)**
    - ① **10.0.0.0 ～ 10.255.255.255 (CIDR记法: 10/8)**
    - ② **172.16.0.0 ～ 172.31.255.255 (CIDR记法: 172.16/12)**
    - ③ **192.168.0.0 ～ 192.168.255.255 (CIDR记法: 192.168/16)**
  - 这些地址只能用于机构的内部
  - **Internet中的路由器不转发目的地址为专用地址的包**
- 企业/机构内部网络使用专用地址的优点
  - 减少IP地址空间的占用
  - 可提高安全性

4.7 虚拟专用网 VPN 和网络地址转换 NAT

一、网络地址转换 NAT

- 当内部网络使用专用地址时，与Internet的通信需要通过NAT(Network Address Translation)

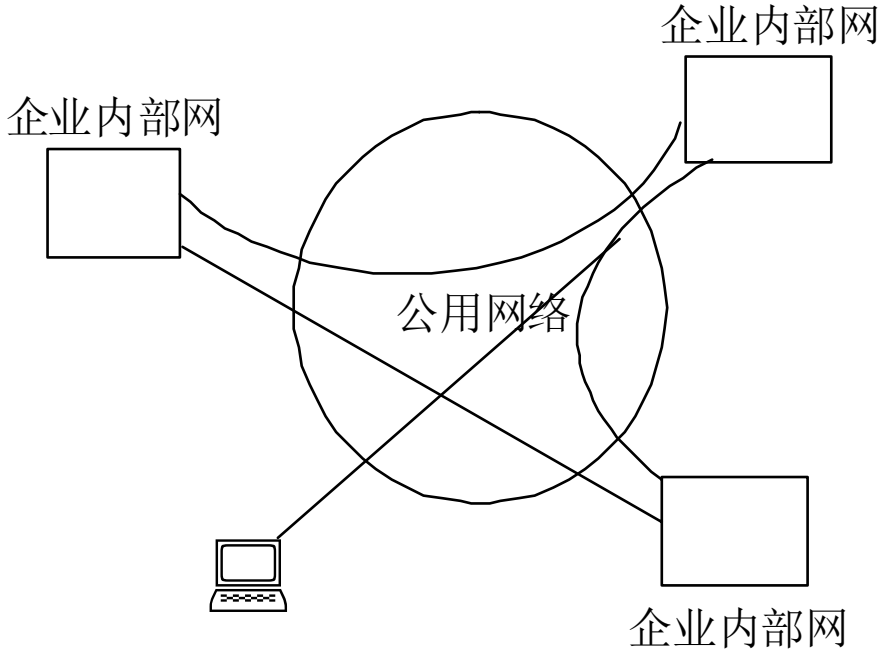


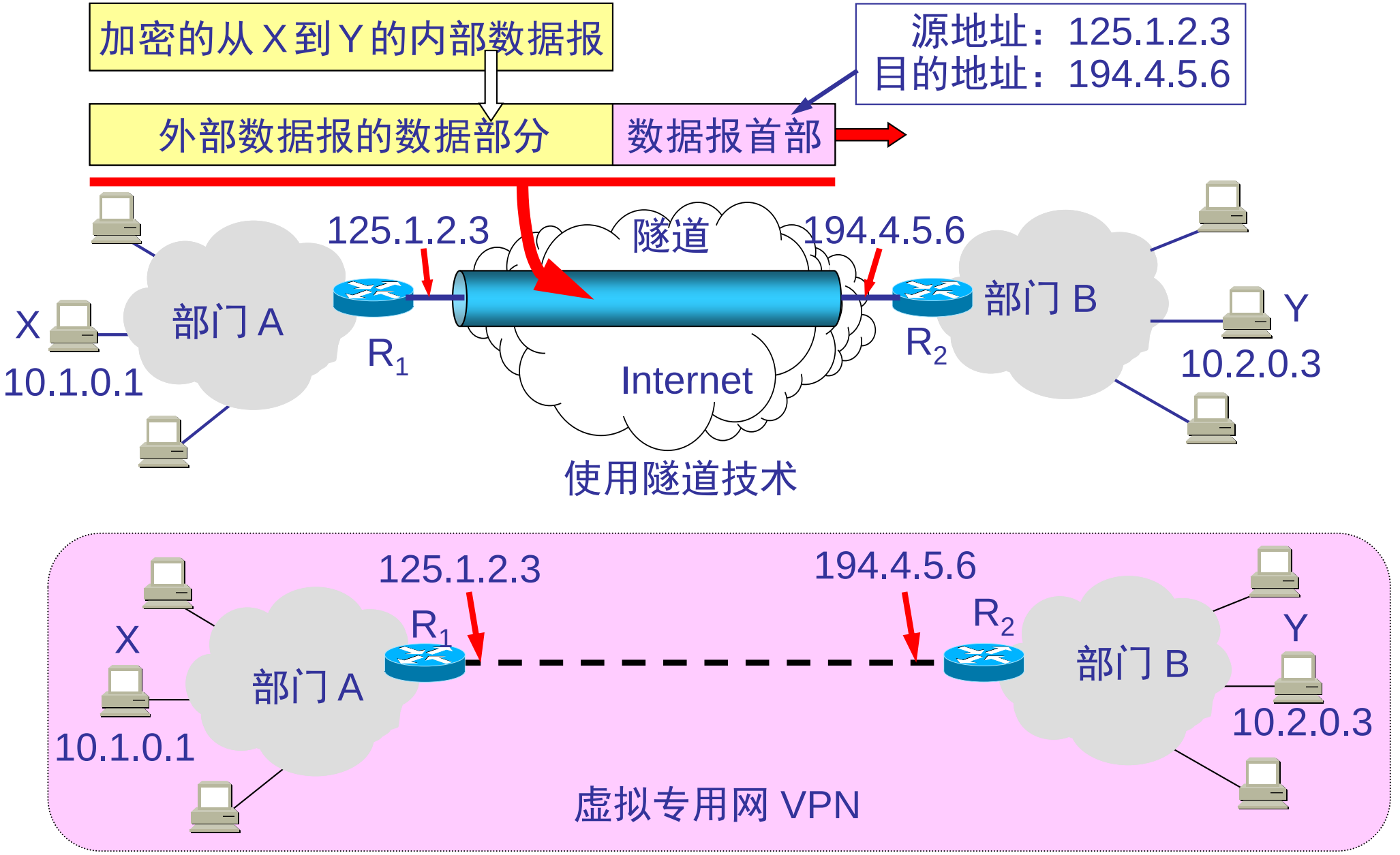
- NAT路由器内部使用TCP/UDP端口号实现外网数据包向内网地址的转换

4.7 虚拟专用网VPN 和网络地址转换NAT

二、虚拟专用网VPN

- **VPN---Virtual Private Network 虚拟专用网**
- 多个企业/机构的内部网络之间互连的实现方法
  - ① 租用专用线路，形成专用网，成本高昂
  - ② 基于公用网络(如Internet)，形成VPN
- **VPN涉及的技术包括：隧道(tunnel)、加密、身份认证等**
- 基于Internet建立VPN的两种情形
  - 内部网络通过Internet互连
    - 可以采用IPSec的ESP隧道模式
  - 远程用户访问内部网(remote access VPN)
    - 拨号虚拟专用网(VPDN---Virtual Private Dialup Network)
    - 基于SSL的VPN技术，优点：客户端无须安装或配置软件





- 机构内部网络又称为**Intranet(内联网)**
- 外部网络称为**Extranet(外联网)**

通过公共网络传输的数据内容均经过加密，外部仅能通过**IP**分组头得知**R1**在**R2**在通信