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

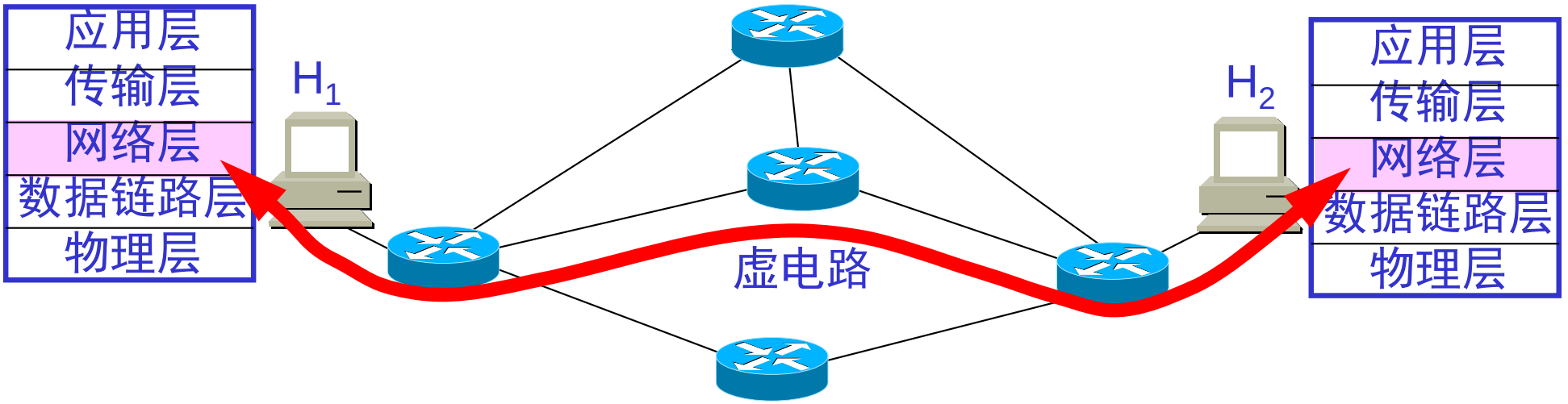
4.1 网络层提供的两种服务
4.2 网际协议IP
4.3 划分子网和构造超网
4.4 网际控制报文协议ICMP
4.5 路由算法及协议
4.6 IP组播
4.7 网络地址转换NAT和虚拟专用网VPN

4.1 网络层提供的两种服务

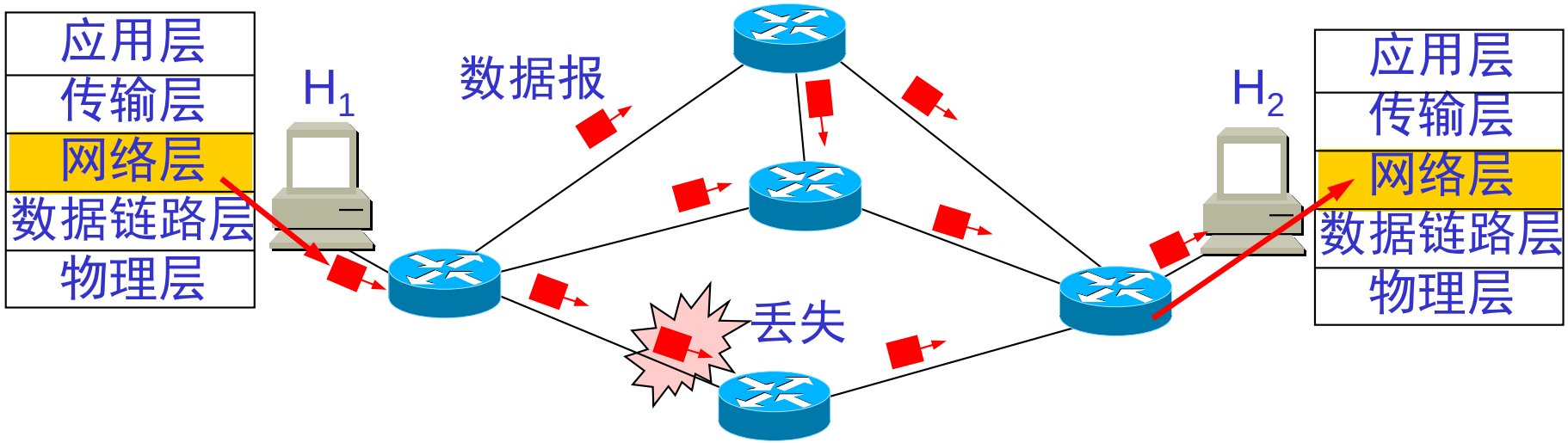
4.1 网络层提供的两种服务

- 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



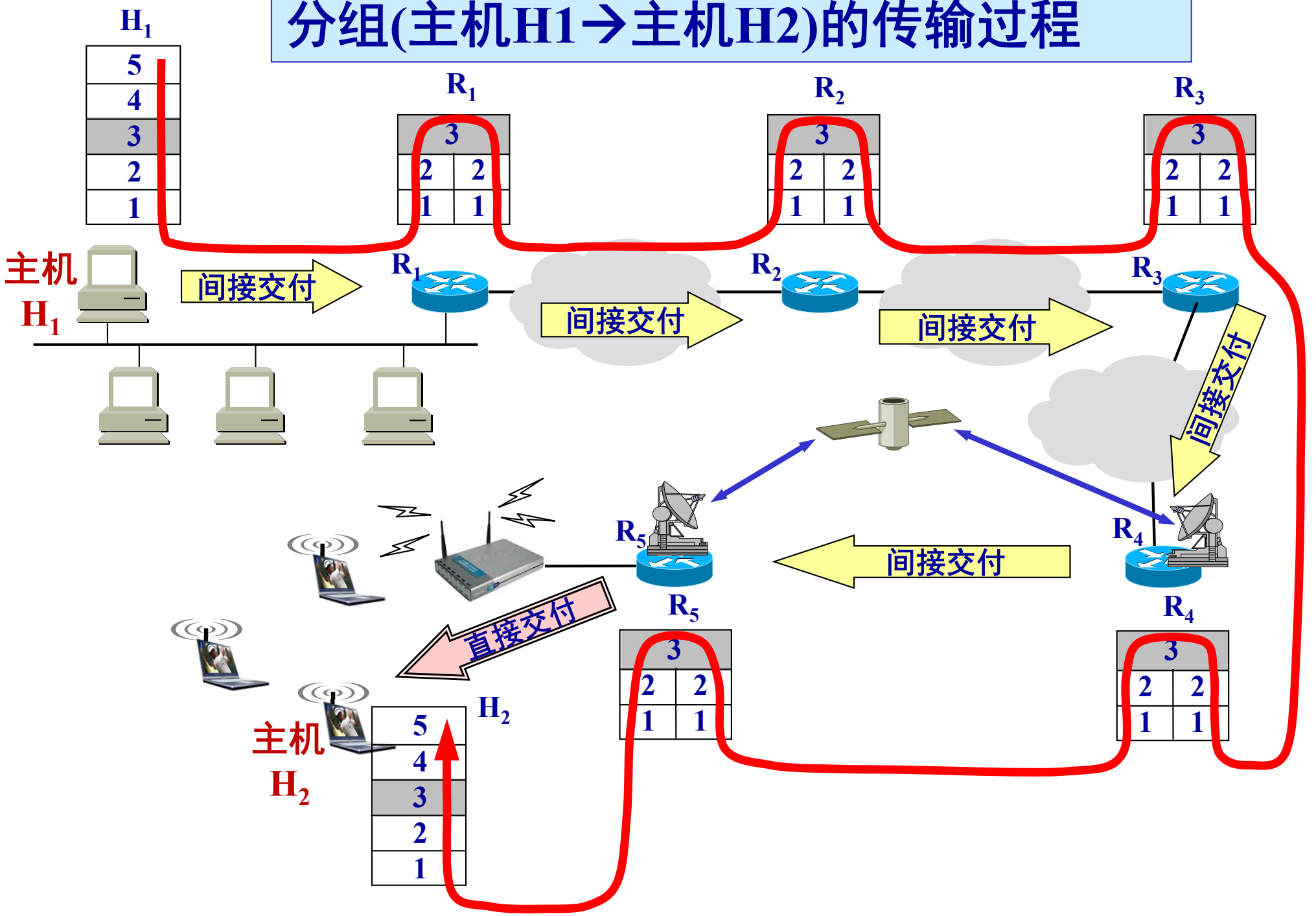
数据报：**H1** 发送给 **H2** 的分组可能沿着不同路径传送

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

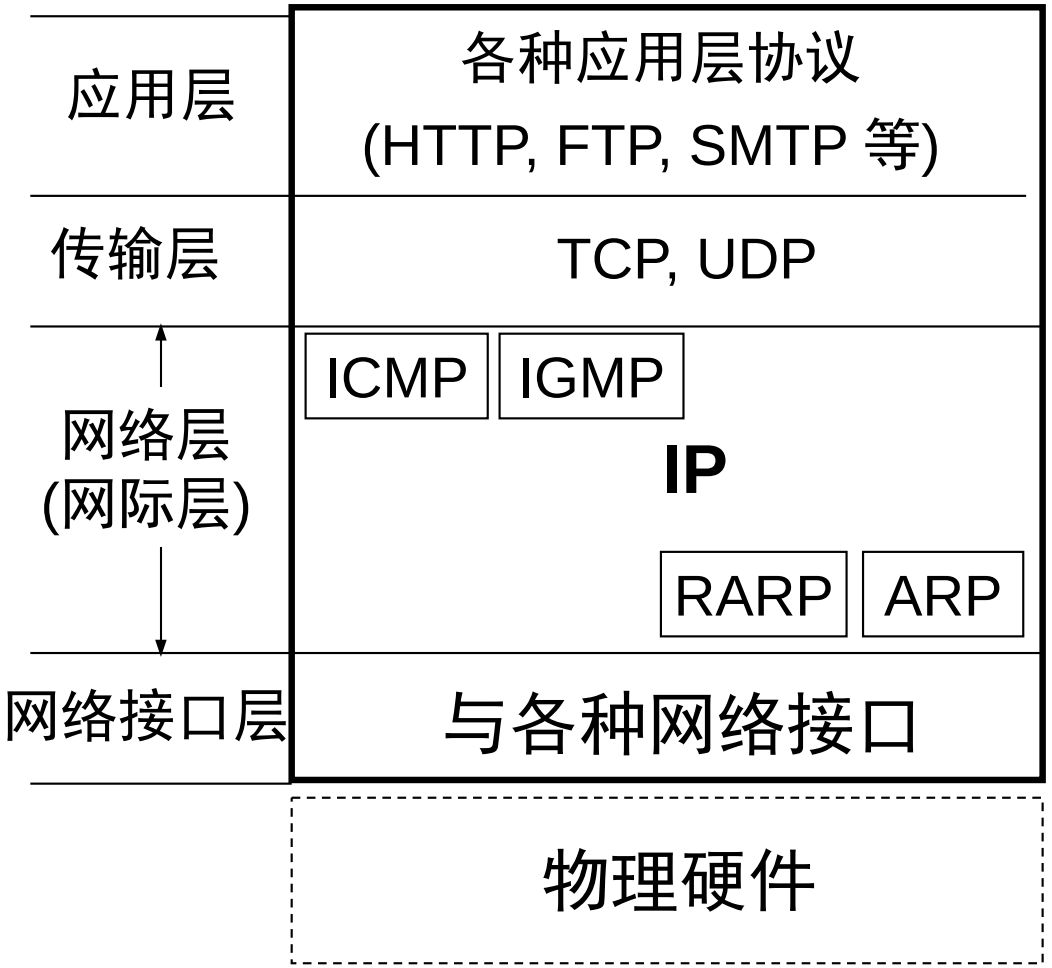


4.2 网际协议 IP

4.2 网际协议 IP

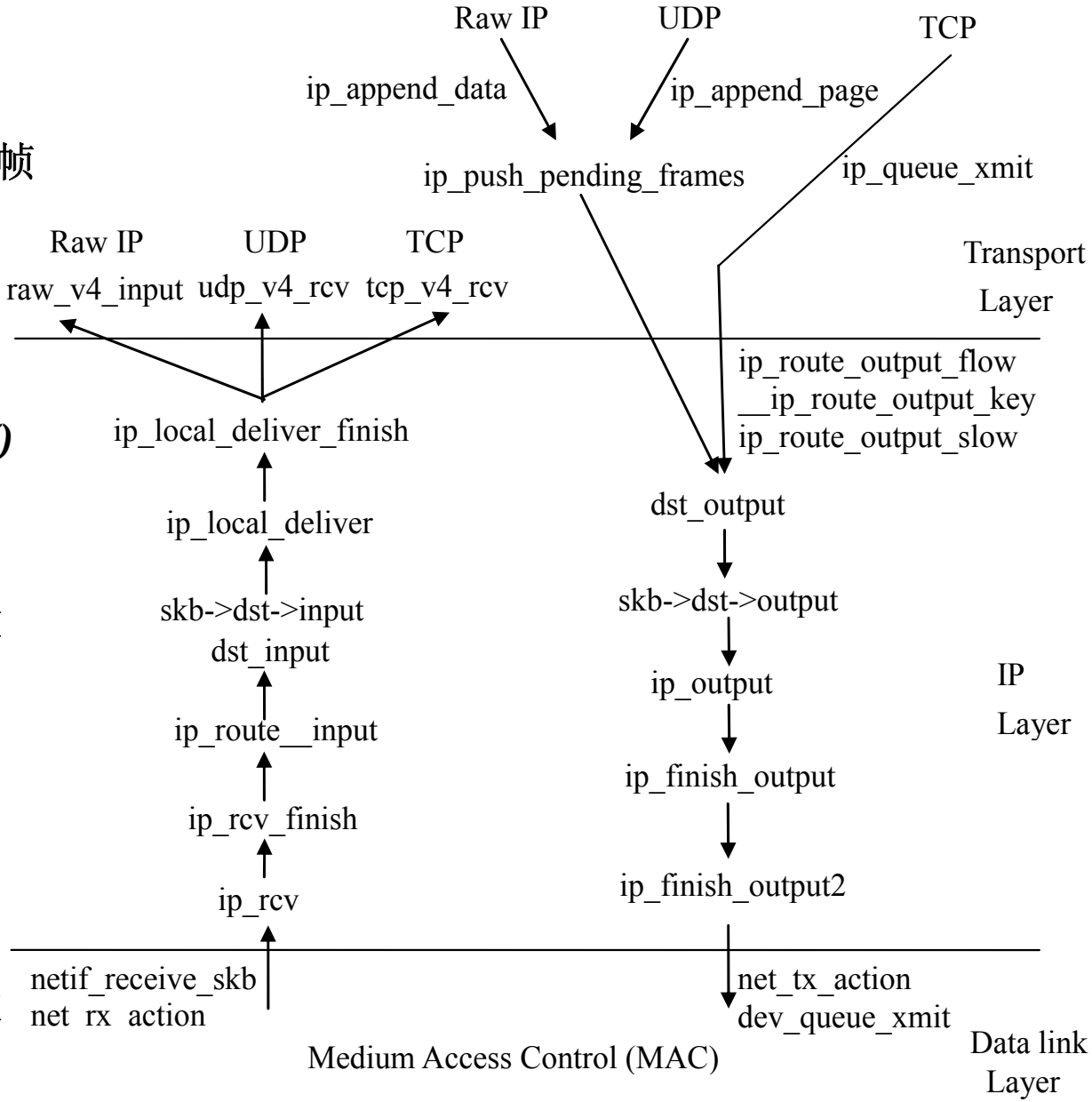
一、IP(Internet Protocol)简介

- 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

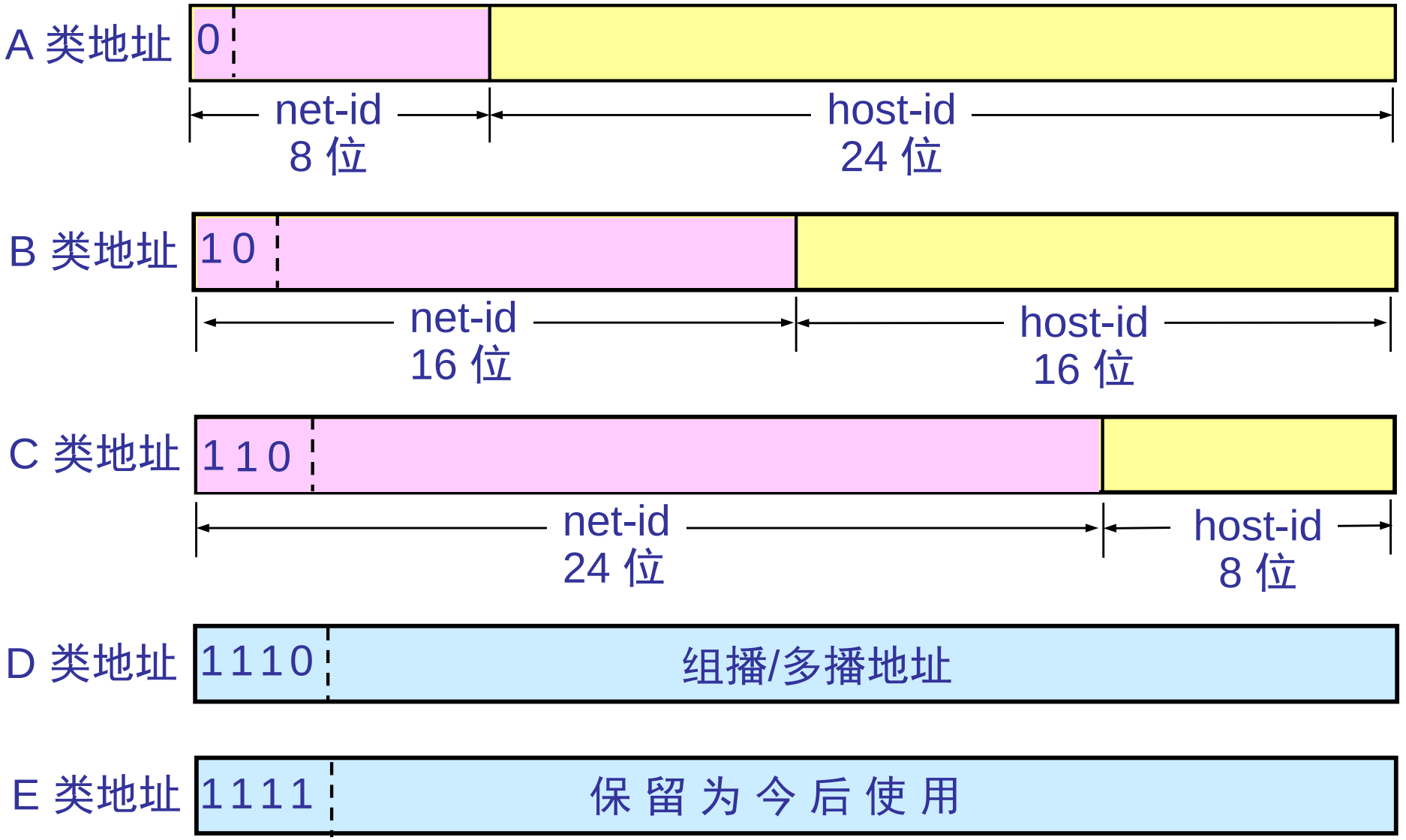
- **IP 地址**
  - 分配给主机或路由器的标识符，目前使用的**IPv4**为**32**位**IP**地址
  - **IP** 地址的分配由**ICANN (Internet Corporation for Assigned Names and Numbers)**负责
- **IP**地址的编址方法经历了三个阶段：
  - 分类的 **IP** 地址：最基本的编址方法，**1981** 年通过标准
  - 子网的划分：最基本编址方法的改进， **1985** 年成为标准**[RFC 950]**
  - 构成超网：比较新的无分类编址方法，**1993** 年提出

  } 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

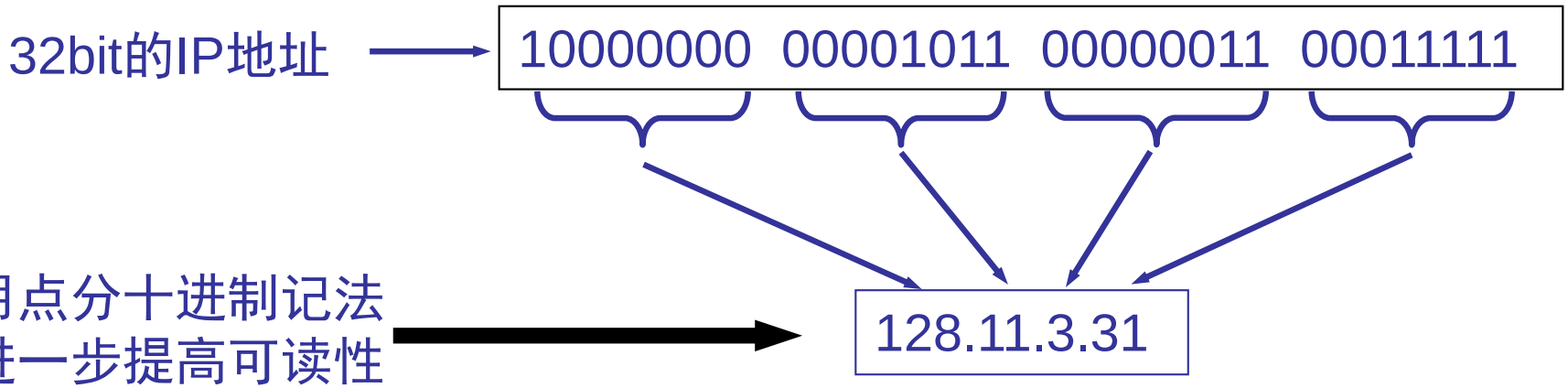
    IP 地址 ::= { <网络号>, <主机号>}
  - 不同类的**IP**地址区别主要是网络号、主机号的长度不同

4.2 网际协议 IP

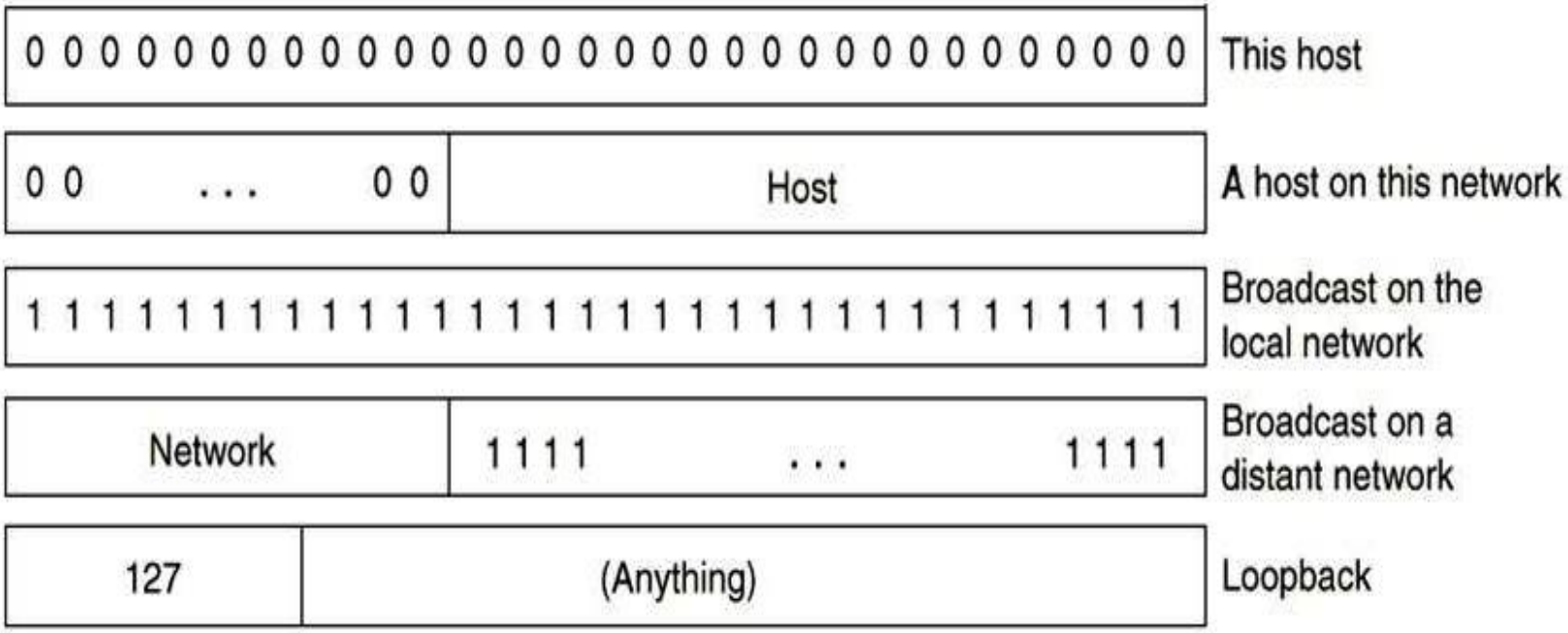
IP 地址中的网络号字段和主机号字段



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



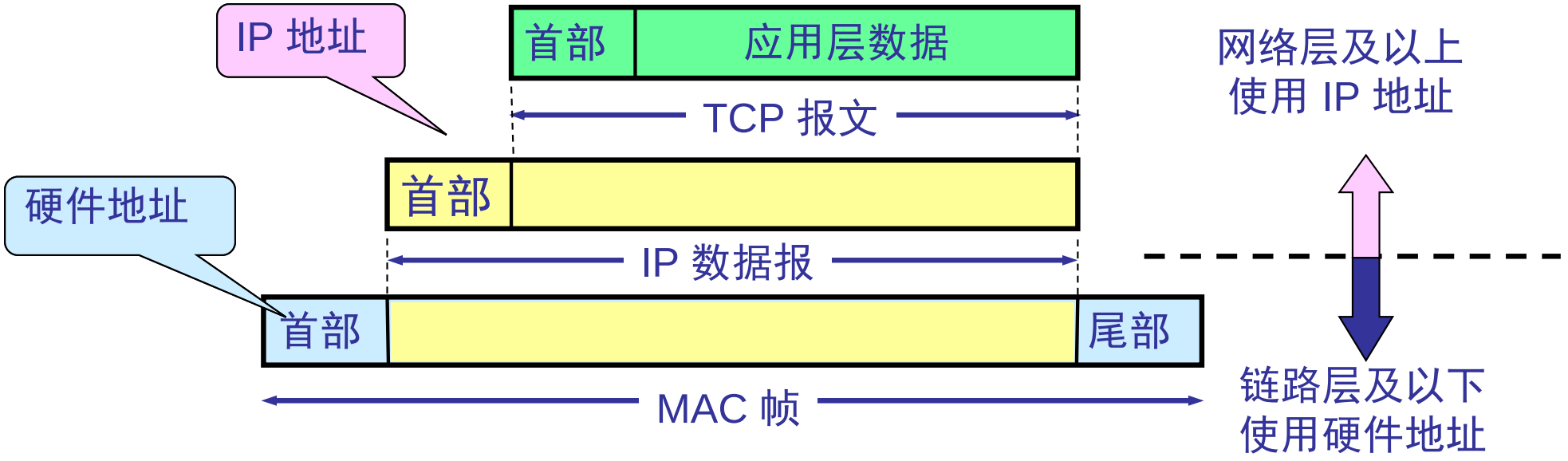
- 全0、全1的IP地址有特殊含义
  - 全0表示本网络或本主机
  - 全1表示广播地址



4.2 网际协议 IP

三、 IP 地址与硬件地址

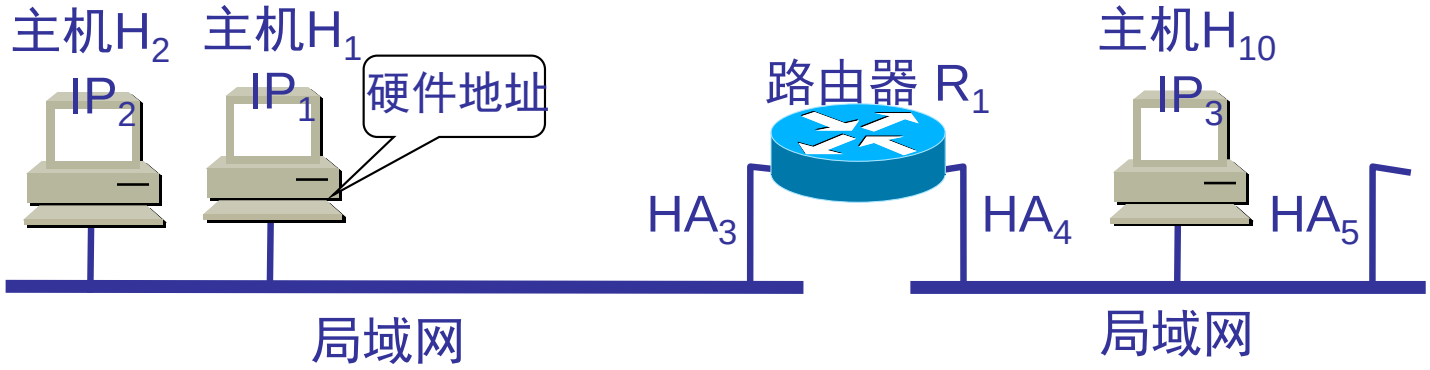
- **IP地址**
  - 网络层及以上各层使用的地址，是一种逻辑地址
  - 存放在**IP**包头部
- 物理地址
  - 数据链路层及物理层使用的地址
  - 存放在数据链路层的帧中
    - 问题：帧中有无**IP**地址？

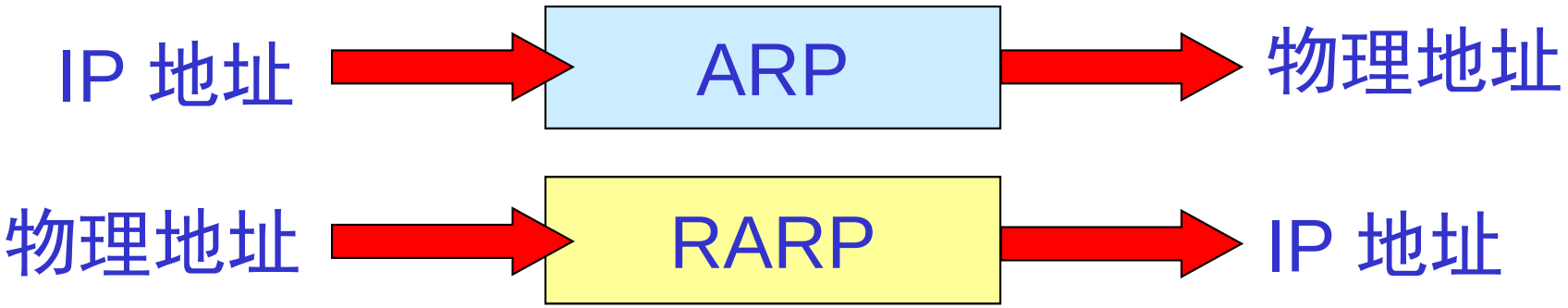


4.2 网际协议IP

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**

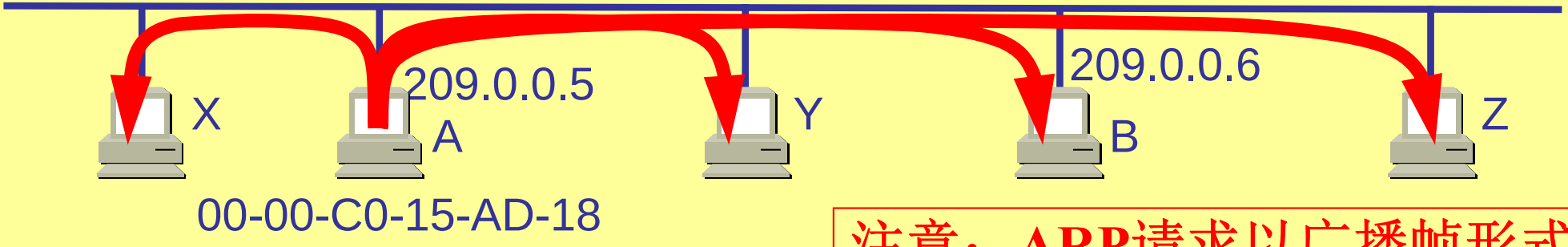




主机 A 广播发送
ARP 请求分组

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

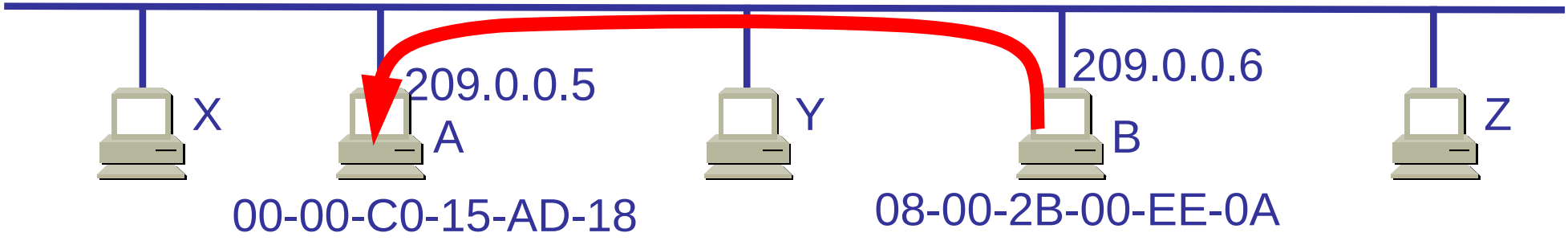


注意：ARP请求以广播帧形式发送

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

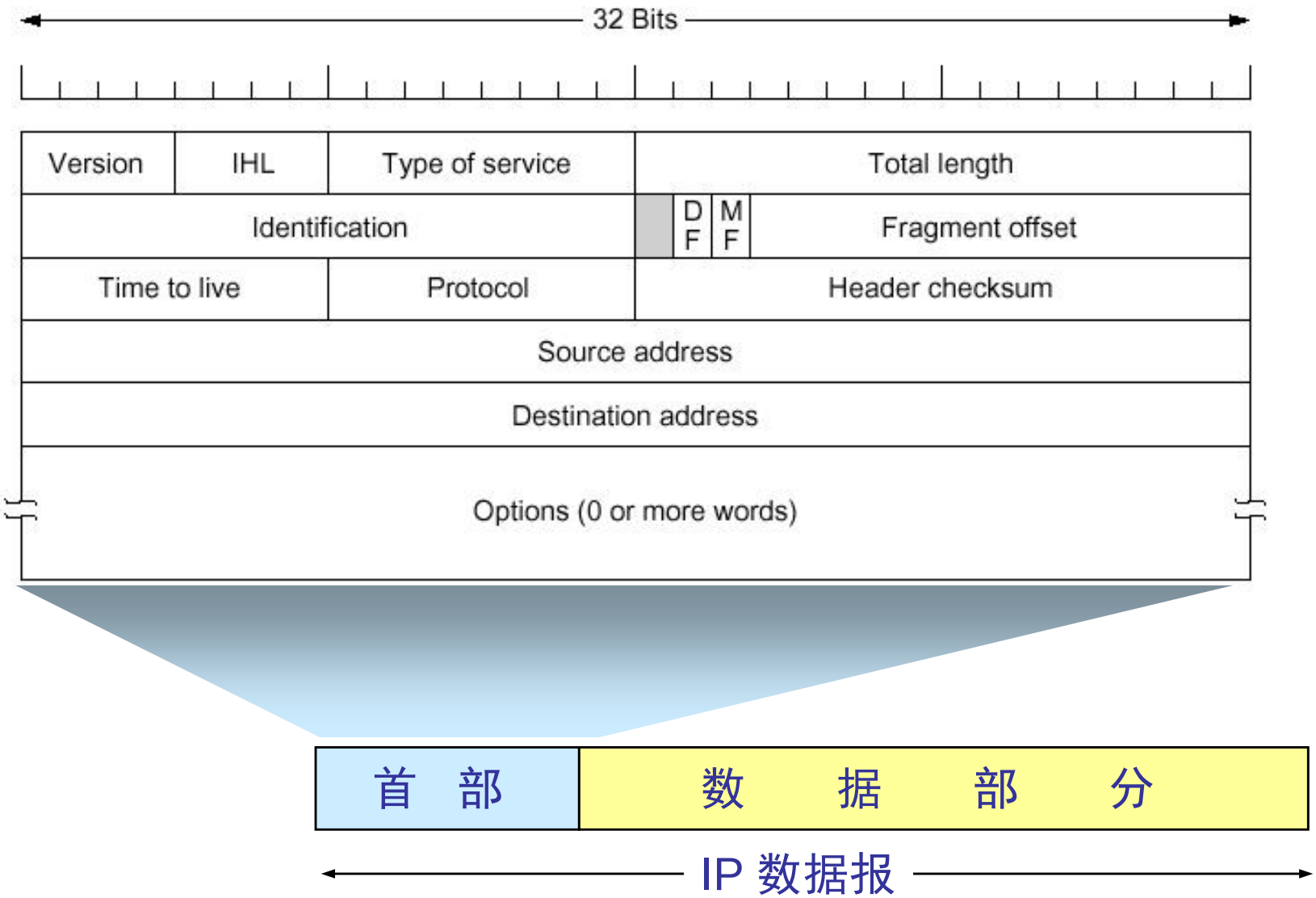
```
C:\>arp -a

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

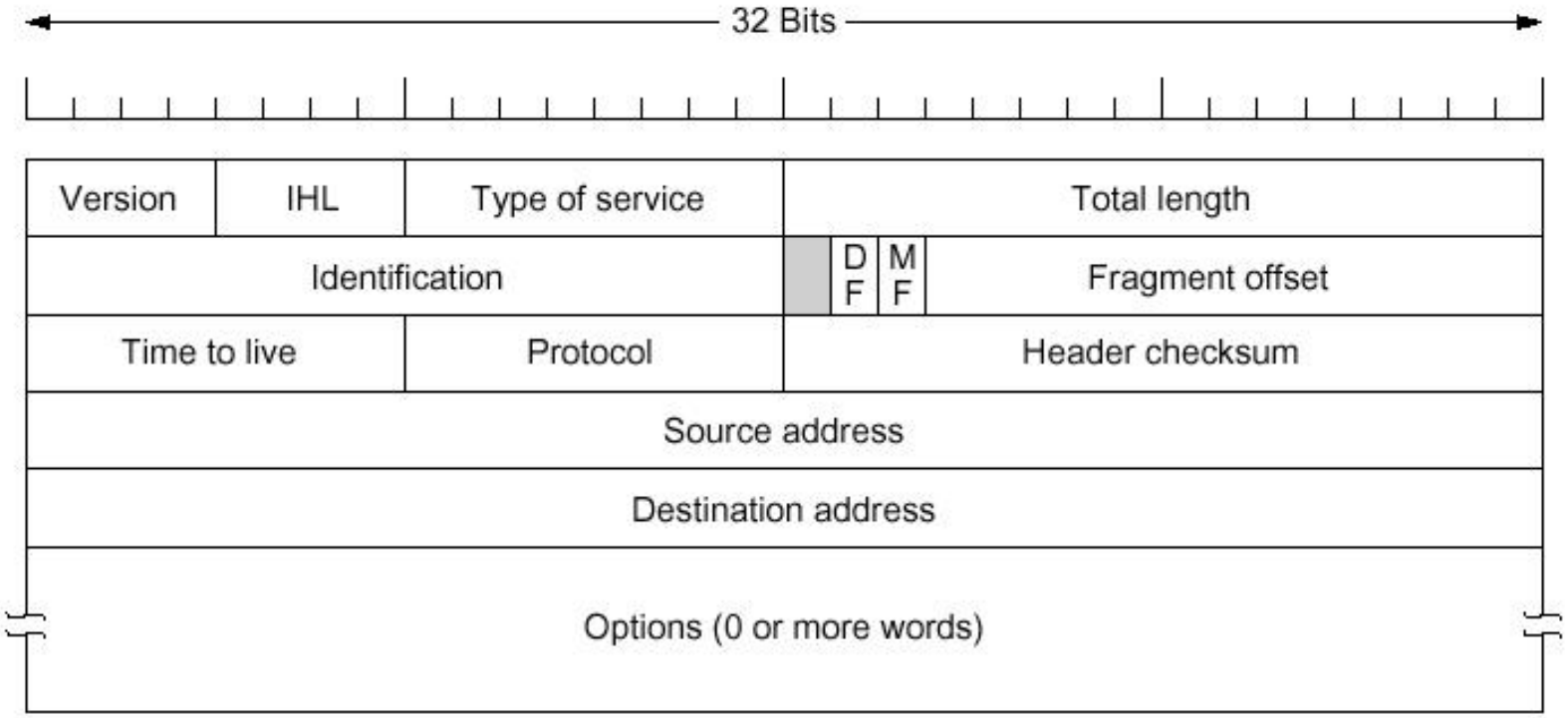
五、IP数据报格式

- 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



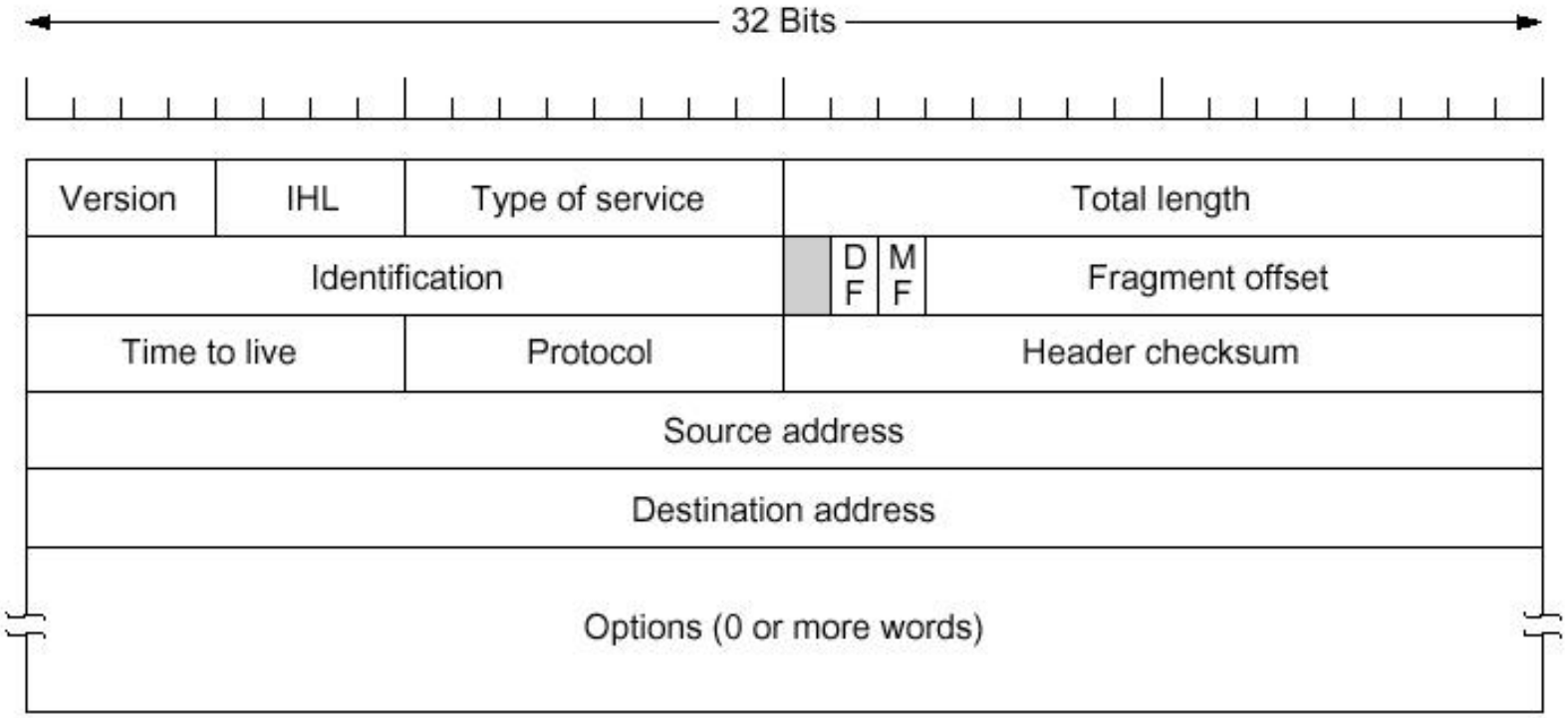
五、IP数据报格式(cont.)

- **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**



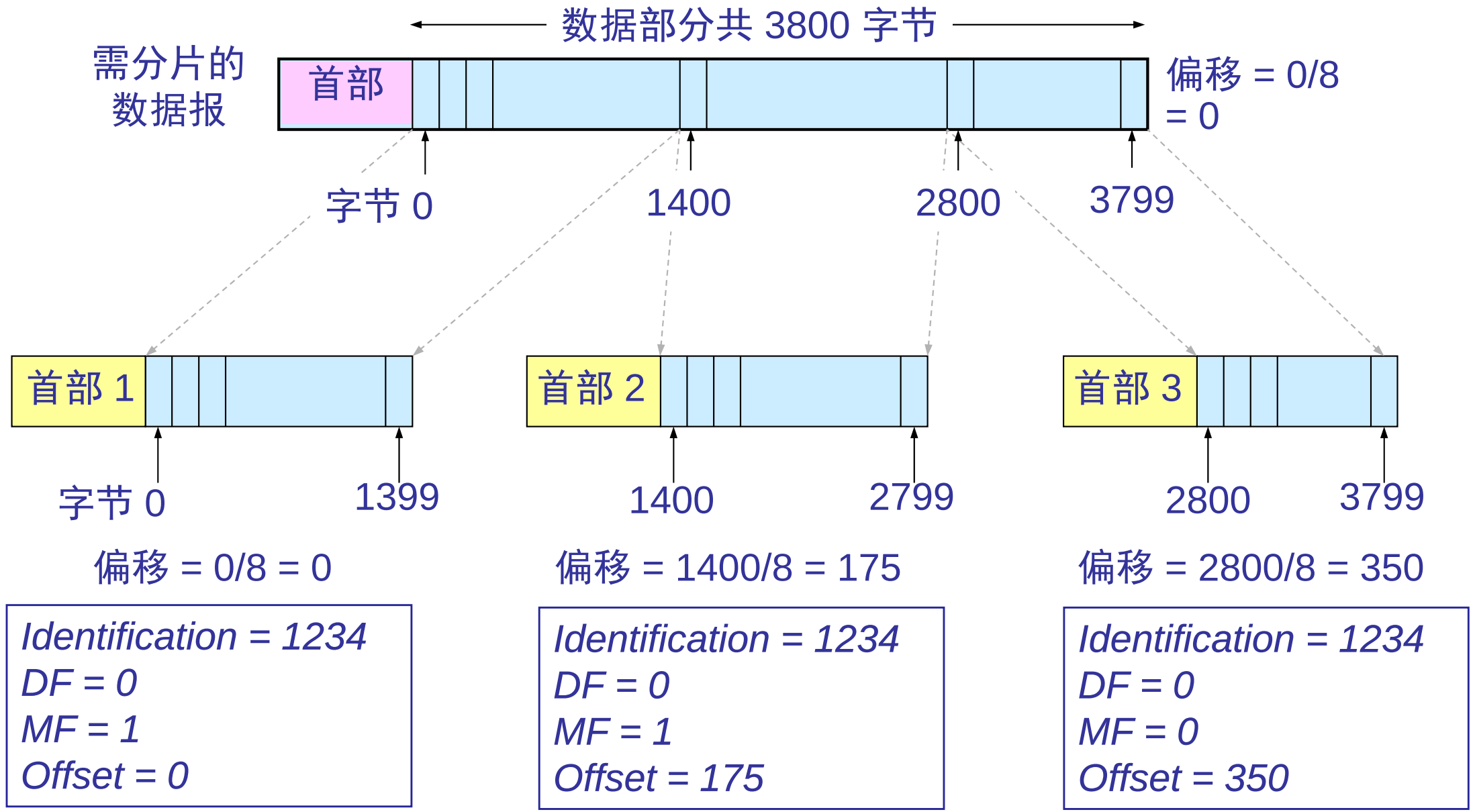
五、IP数据报格式(cont.)

- **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.)

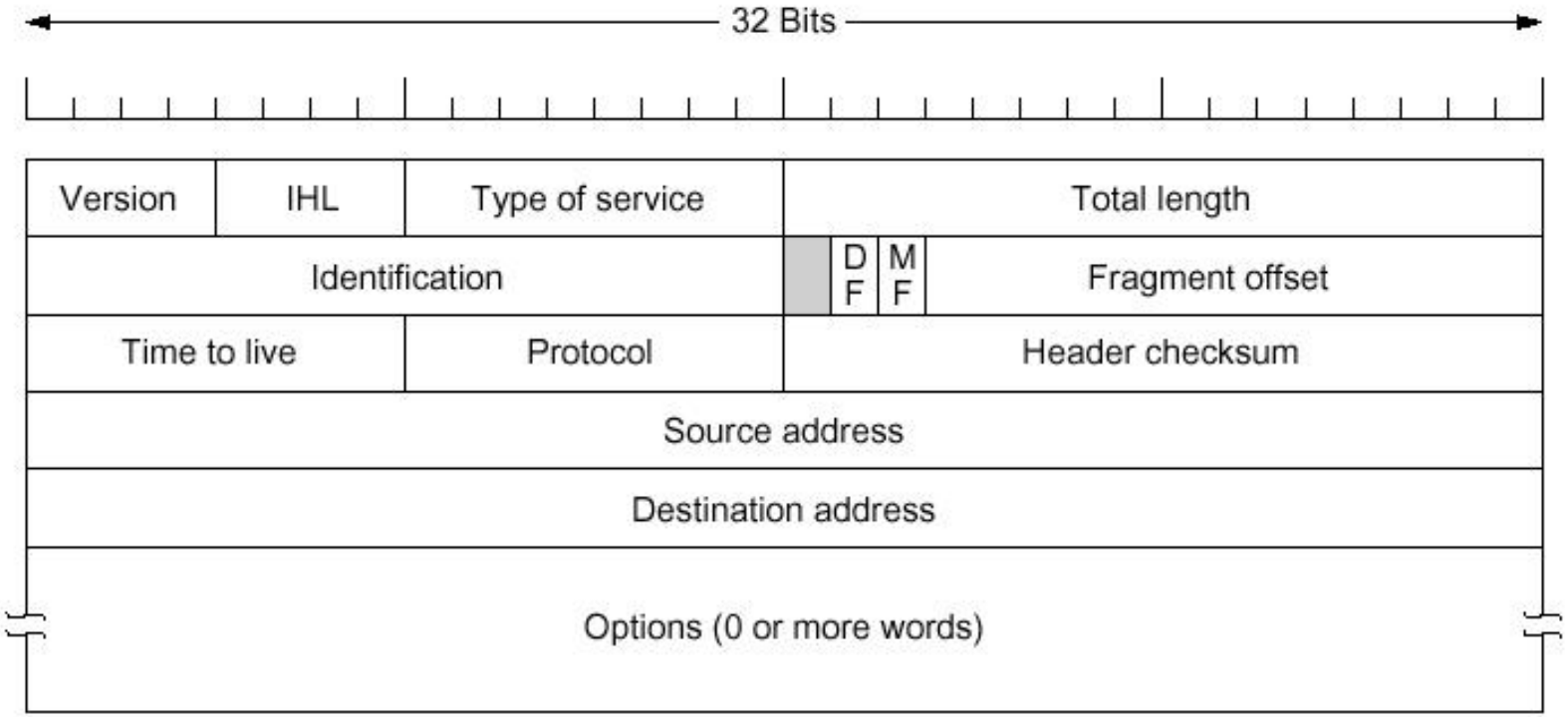
- 分片举例(假设数据链路层一帧的载荷长度≤1420字节)



五、IP数据报格式(cont.)

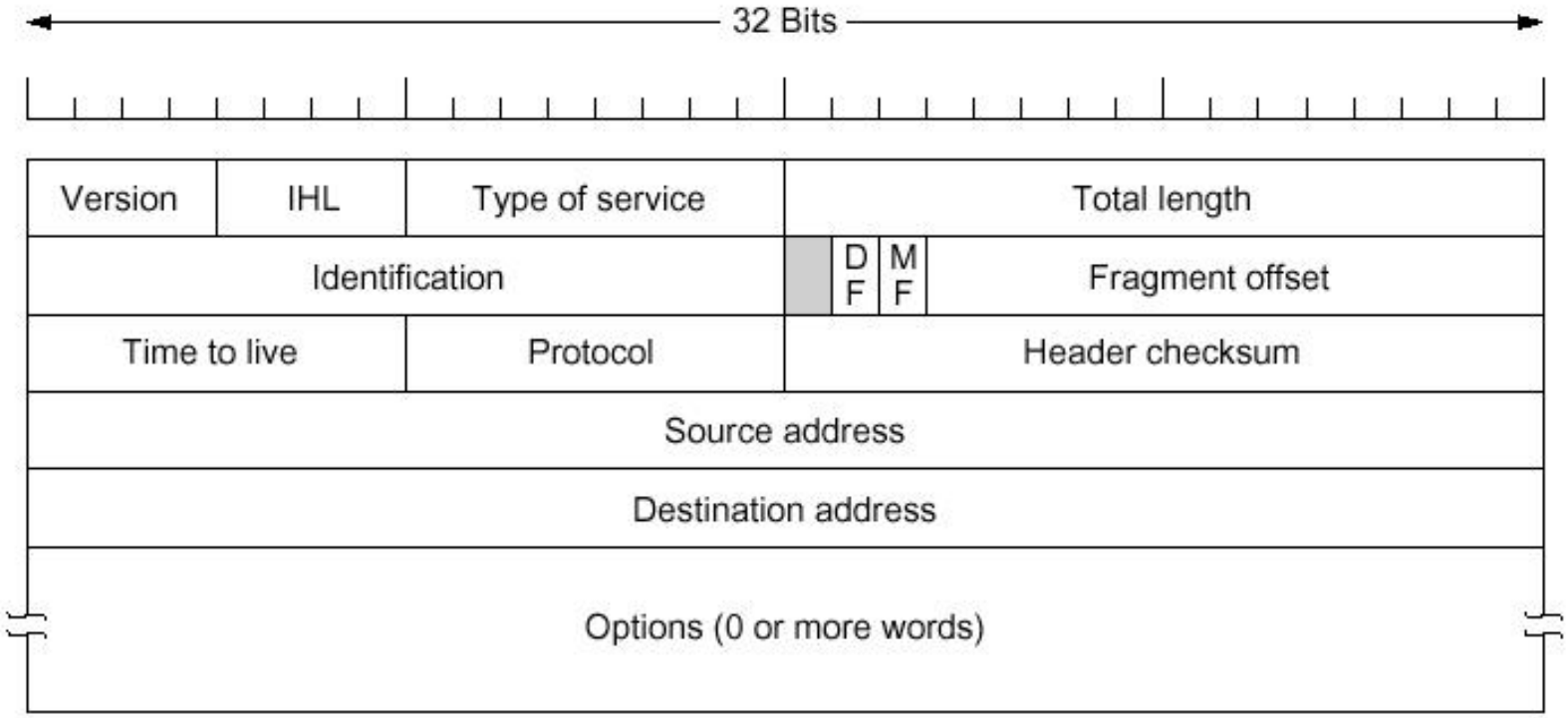
IP包每经过一个路由器俗称为“一跳(hop)”，经过的路由器个数俗称为“跳数(number of hops)”

- **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.)

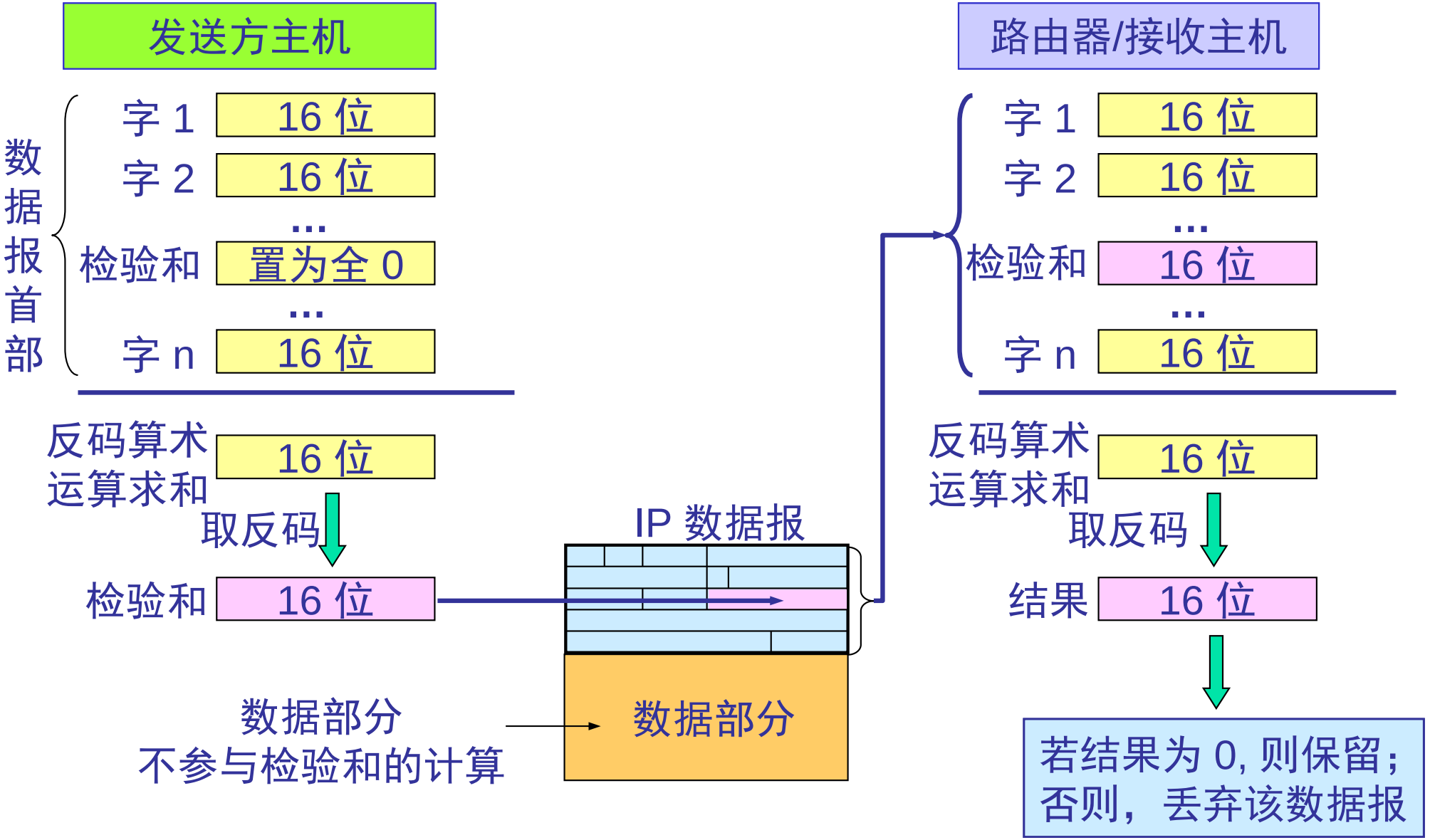
- **Protocol：8bit，协议字段，该包中数据部分的协议类型，即上层协议类型 → 该字段决定了该包将交由哪里**
- **Header checksum：2字节，包头校验和(注意：只针对包头)**
- **Source address：4字节，源IP地址**
- **Destination address：4字节，目的IP地址**
- **选项字段：以4字节为单位，最长40字节。实际网络中很少使用**



五、IP数据报格式(cont.)

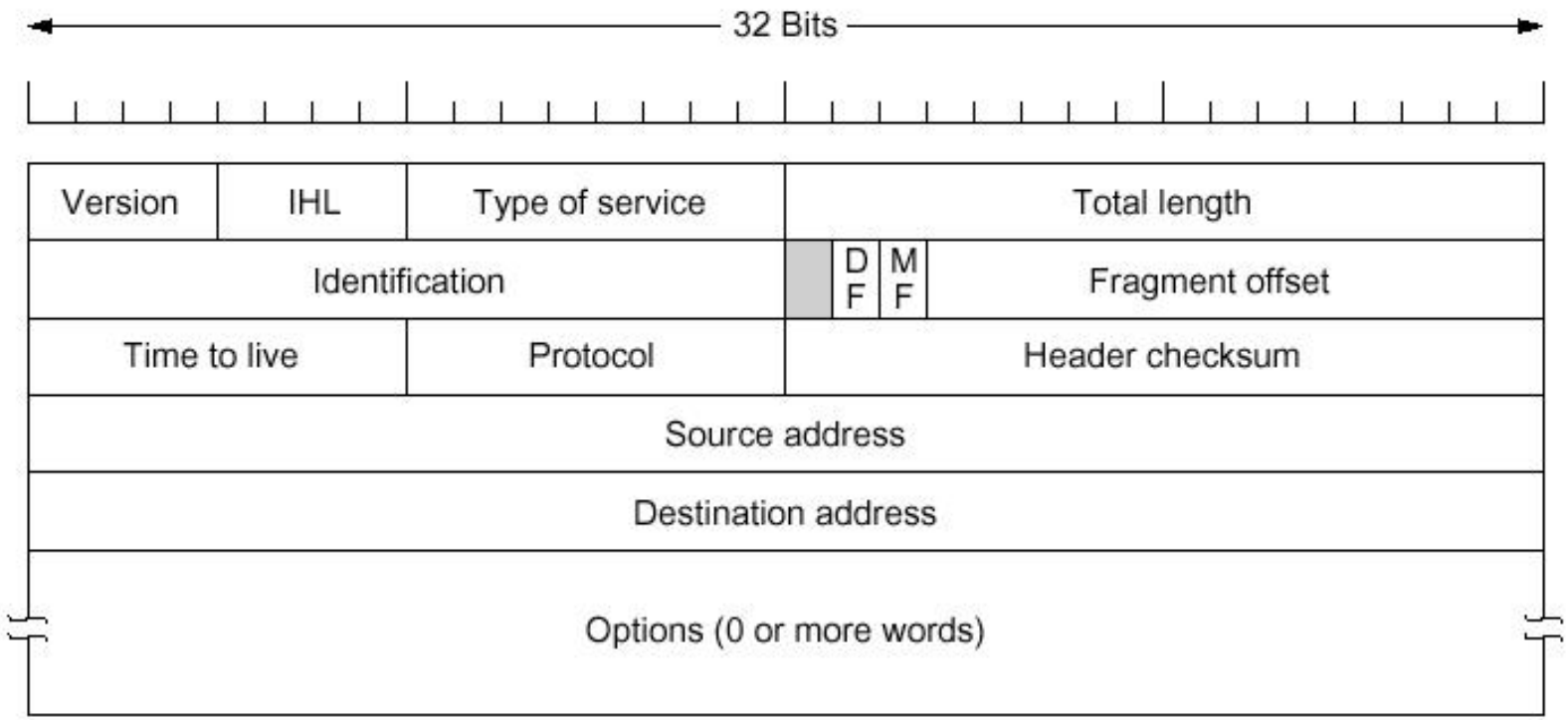
- 校验和算法：对IP包头，每16位求反，循环相加(进位加在末尾)，和再求反

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

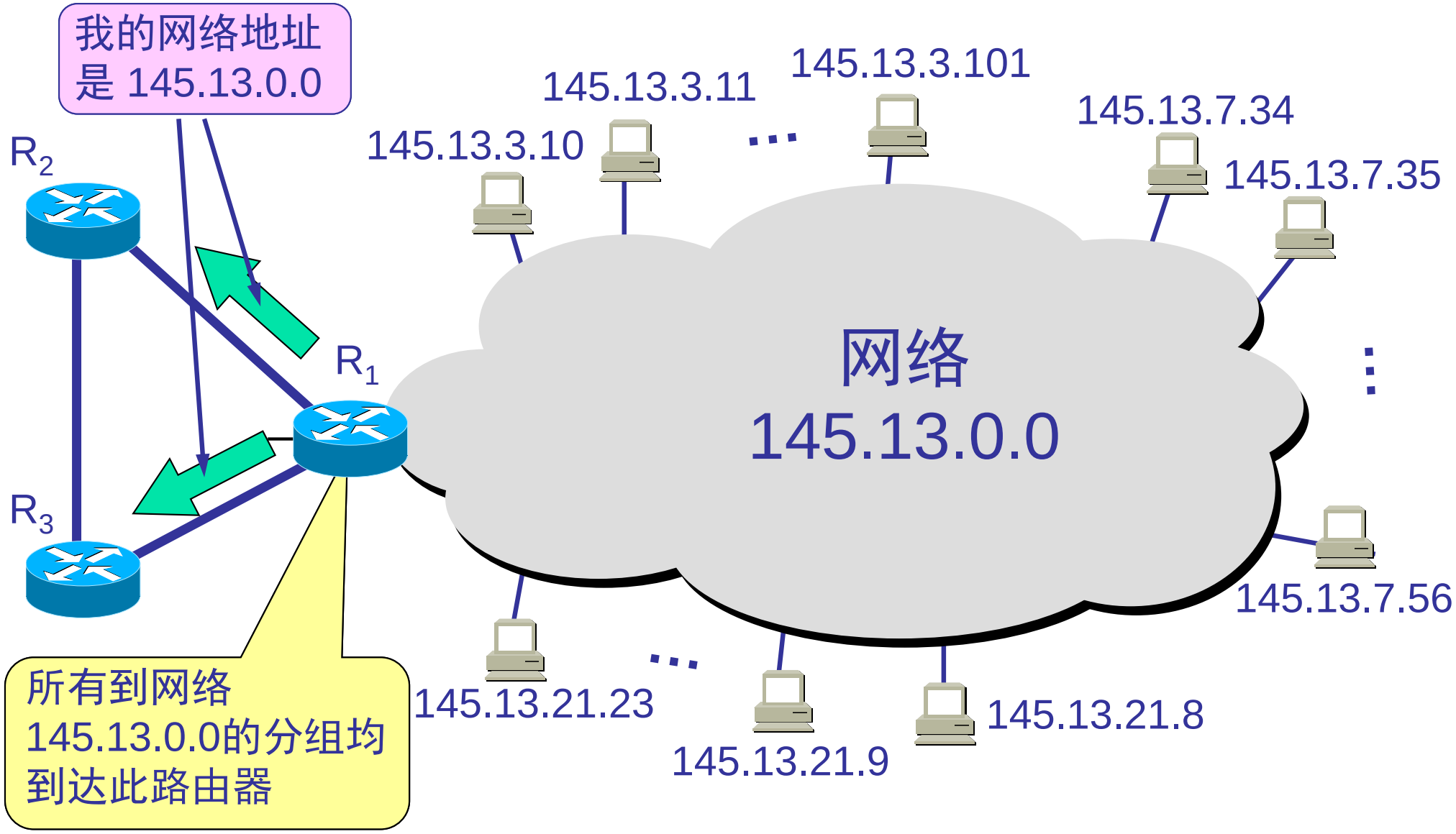
一、划分子网

- 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：

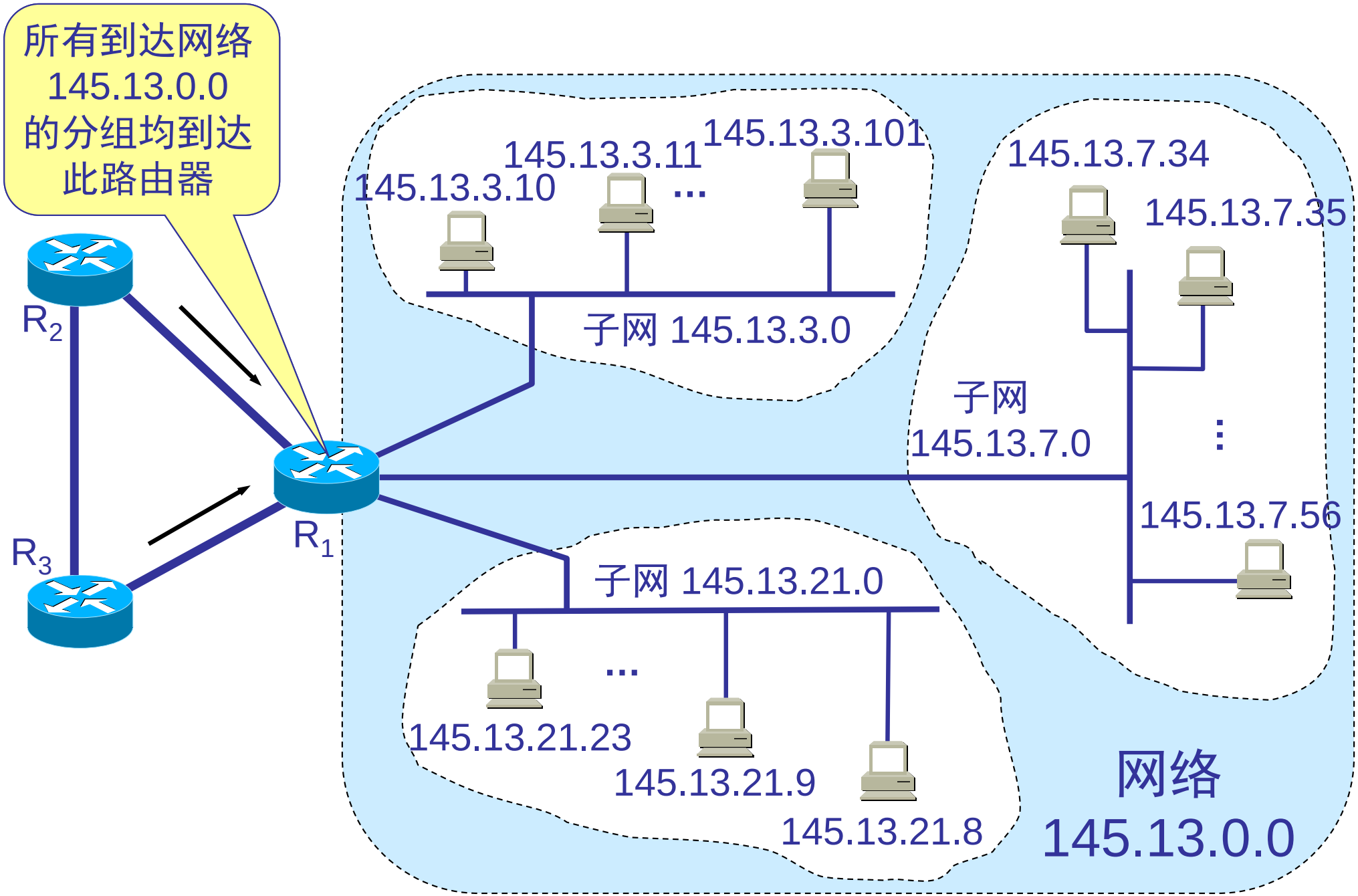
IP地址 ::= {<网络号>, <子网号>, <主机号>}

  - 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



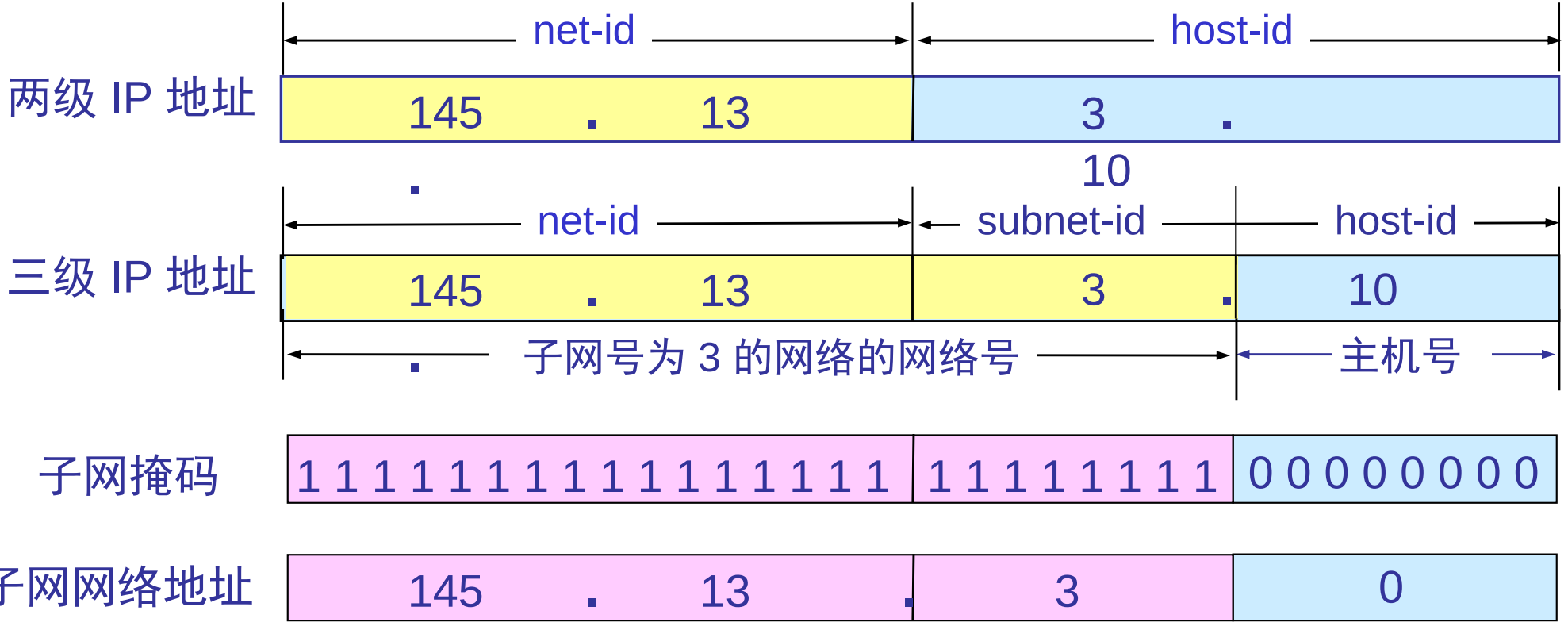
划分为三个子网后


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

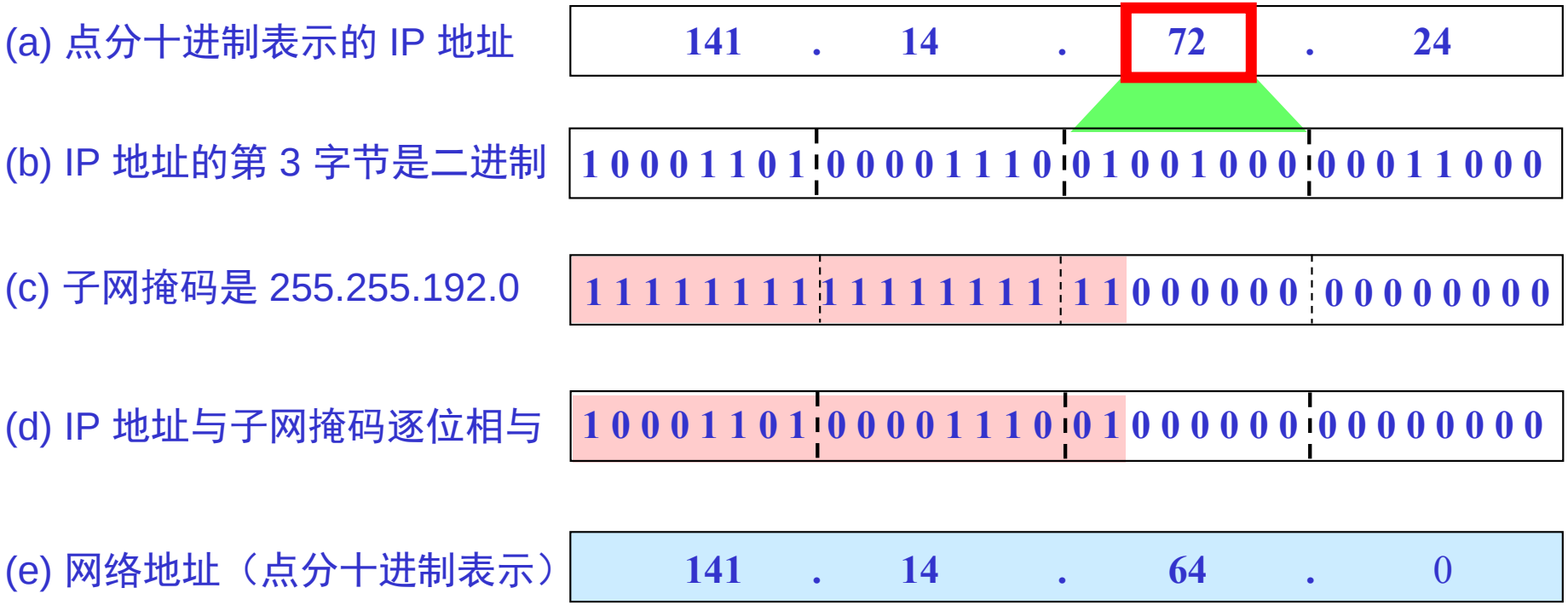
- 子网掩码
  - **Internet**标准规定，所有网络都必须使用子网掩码
  - 子网掩码是一个网络或一个子网的重要属性
  - 路由器的路由表中的每个表项除了包含目的网络地址外，还要有子网掩码栏目
  - 如一个路由器连接在两个子网上，就拥有两个网络地址和两个子网掩码

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

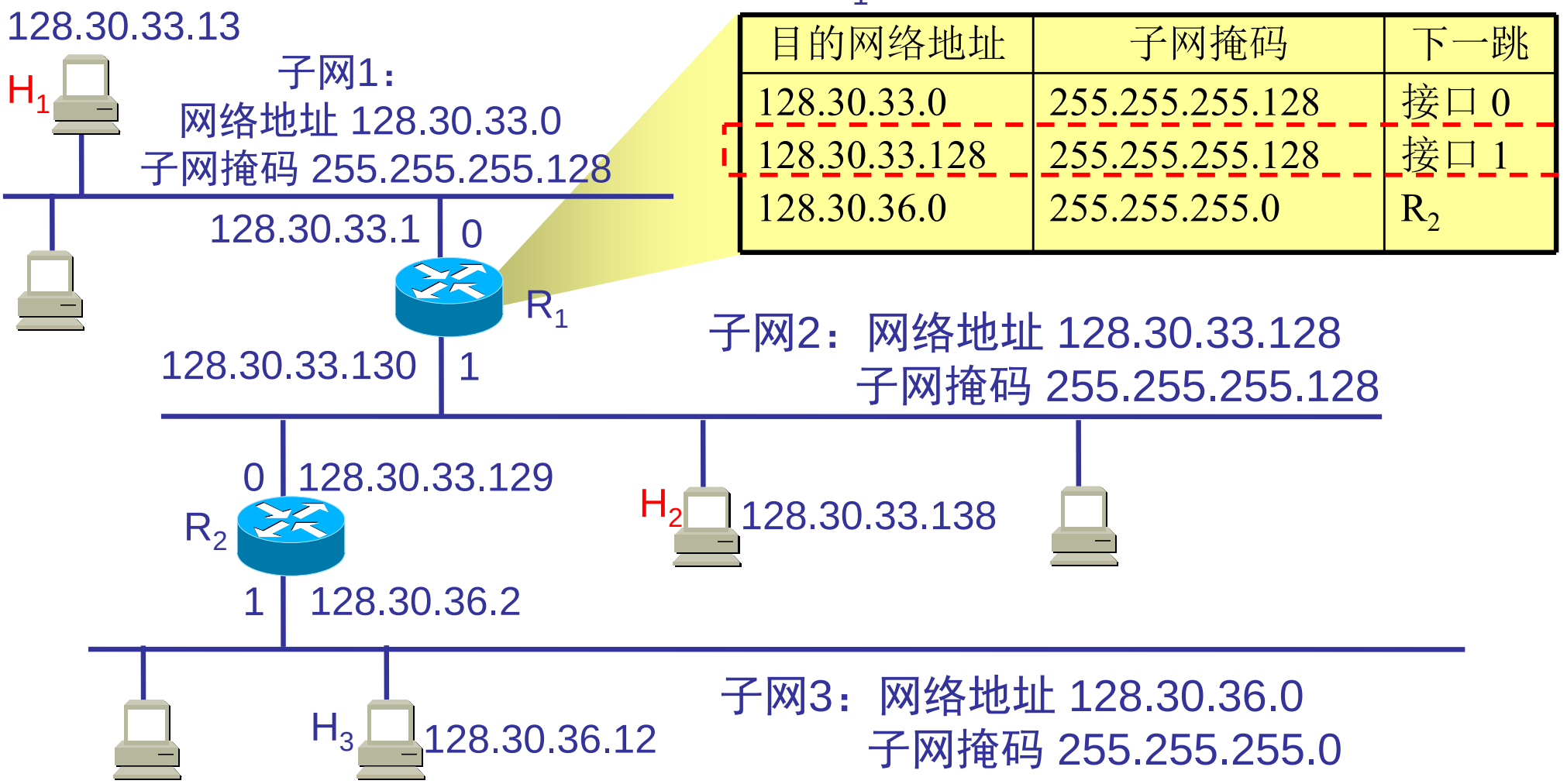
② 主机H1将数据包发给路由器R1(注意局域网内可能有ARP查询过程)

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

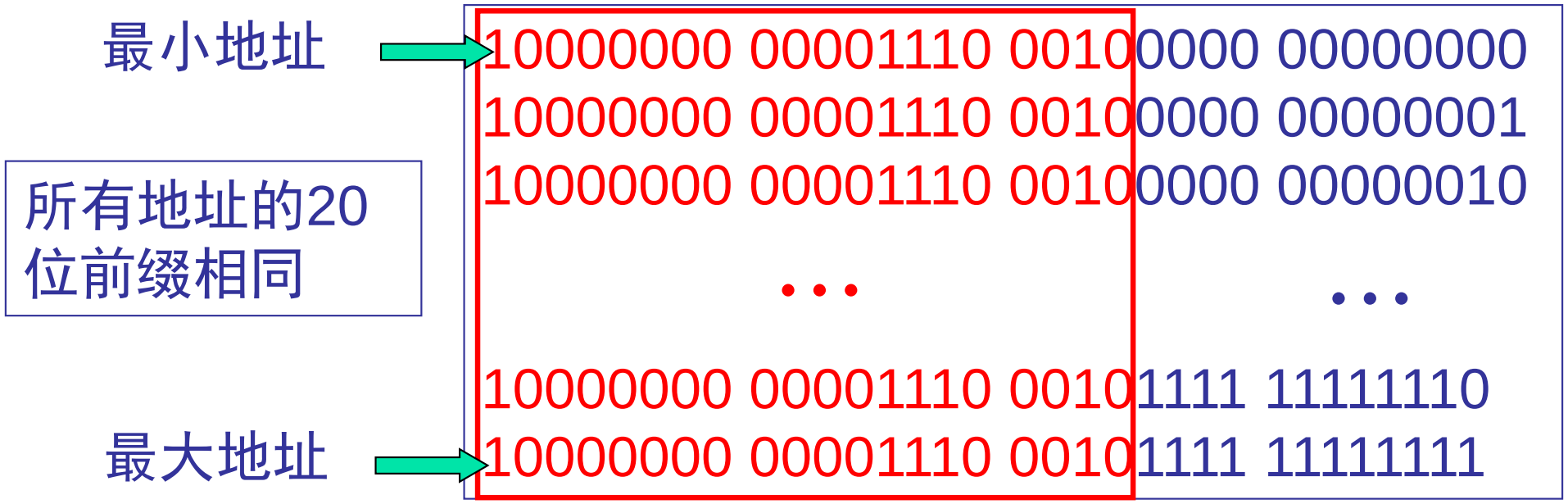
三、无分类编址 CIDR

- 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



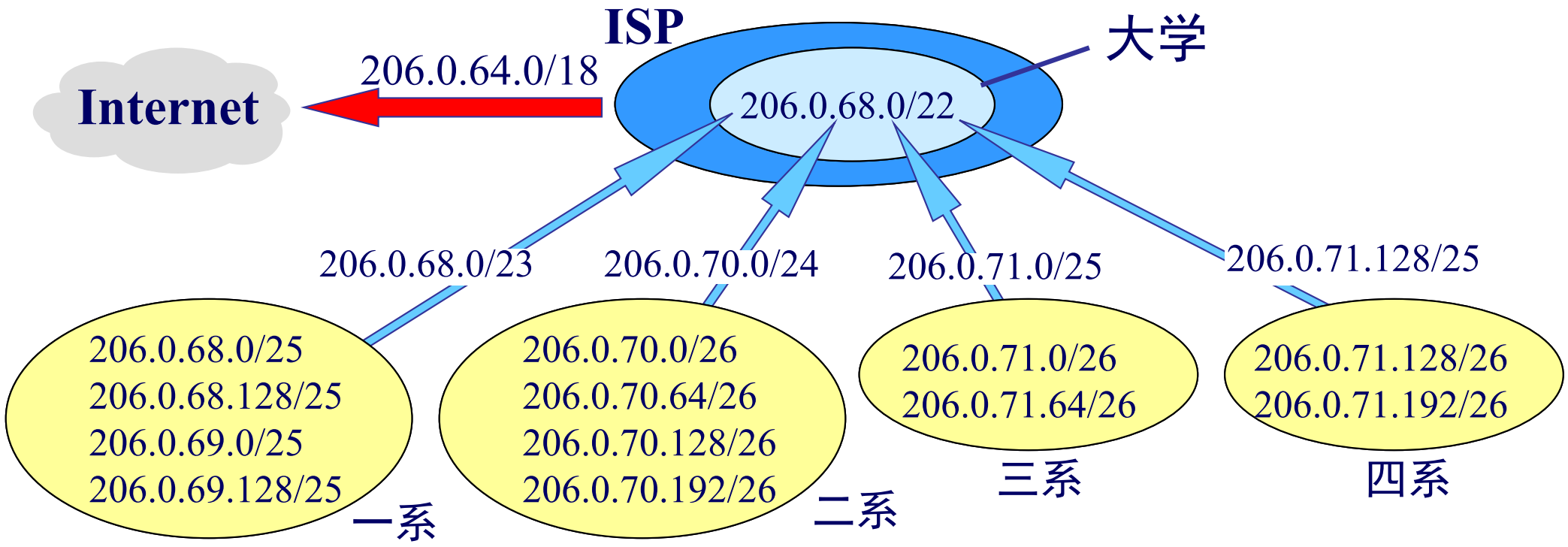
4.3 划分子网和构造超网

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