

# 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**

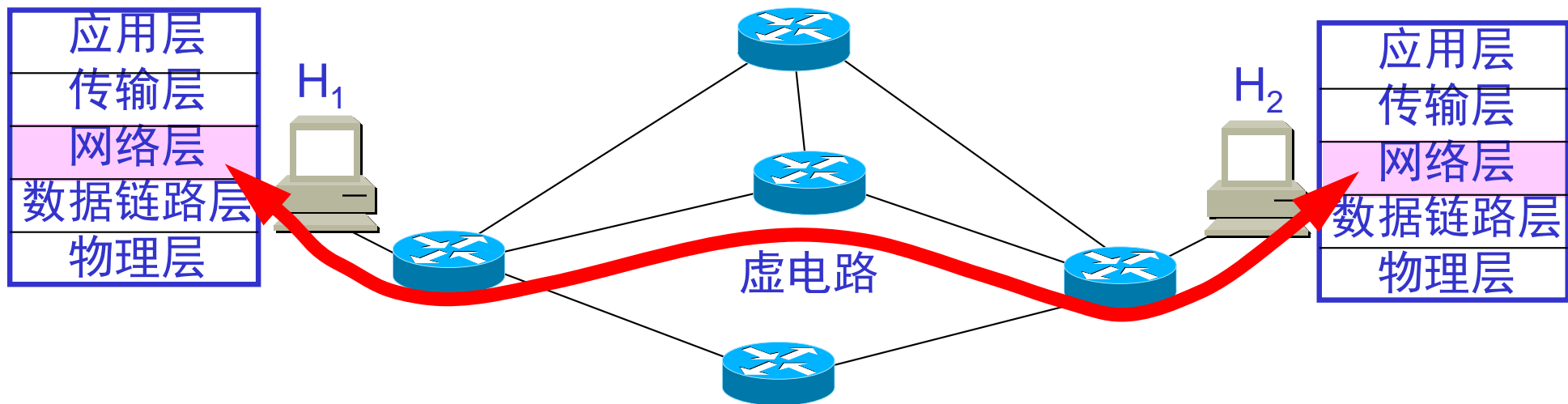
## **4.1 网络层提供的两种服务**

## 4.1 网络层提供的两种服务

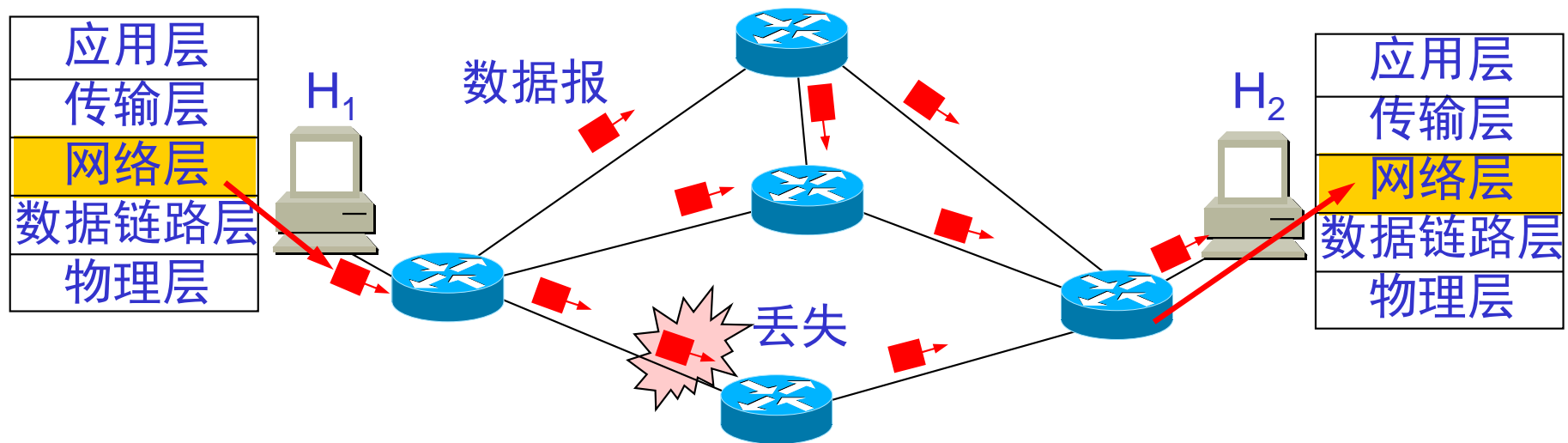
- 网络层应该向传输层提供怎样的服务？
  - 两种选择：面向连接 or 无连接
  - 曾引起了长期的争论
  - 争论的实质：数据的可靠传输应该由网络还是端系统来负责？
- 面向连接的服务，即虚电路(virtual circuit)
  - 通信双方在开始数据传输前，先由网络建立连接，之后的数据均通过该连接进行，由网络保证数据传输的可靠性
  - 虚电路只是一种逻辑连接，分组沿着这条逻辑连接按照存储转发方式传送，而并不是真正建立了一条物理连接
  - 支持方：以电信公司为代表的一派
- 无连接的服务，即数据报(datagram)
  - 网络在发送数据时不需要先建立连接，每一个分组在网络中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



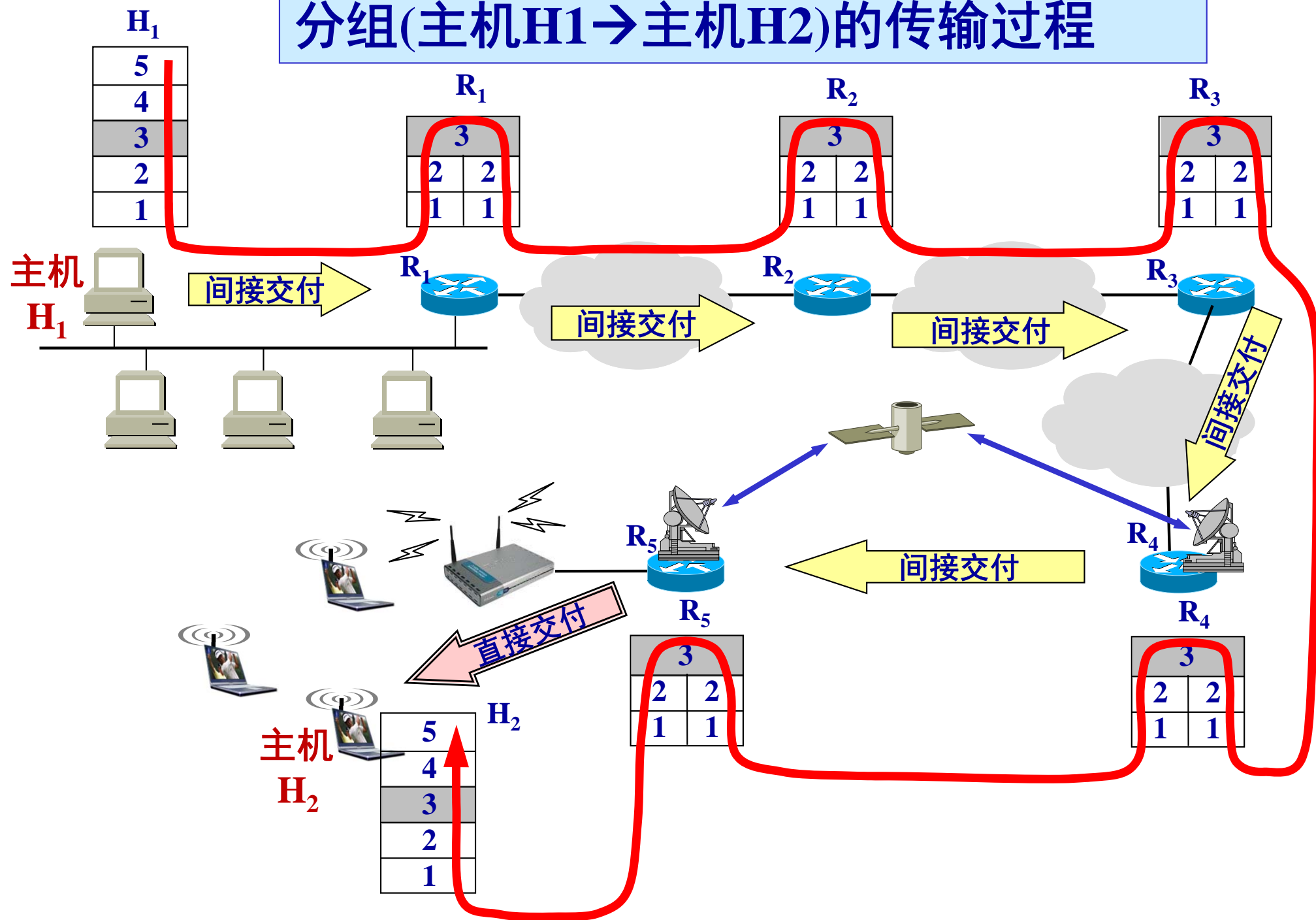
数据报：H1 发送给 H2 的分组可能沿着不同路径传送

# 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



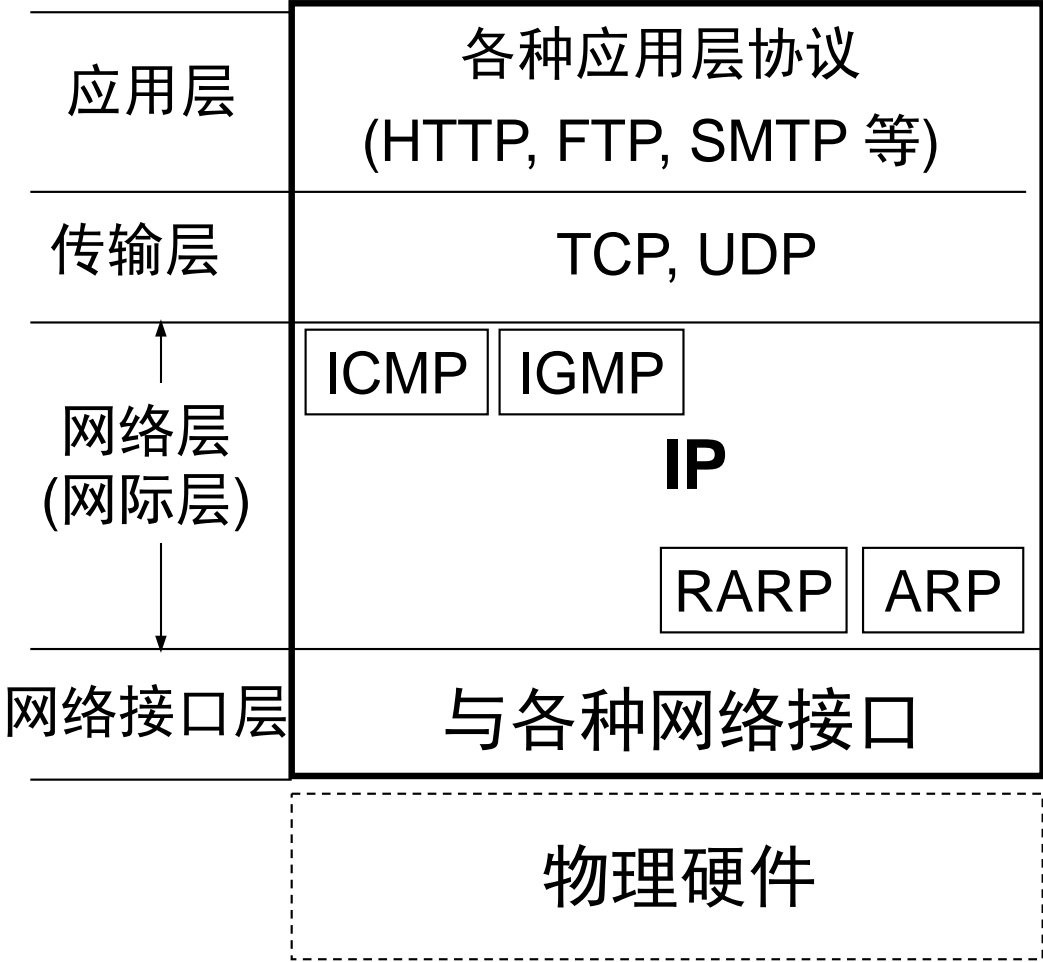
## 4.2 网际协议 IP



# 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)



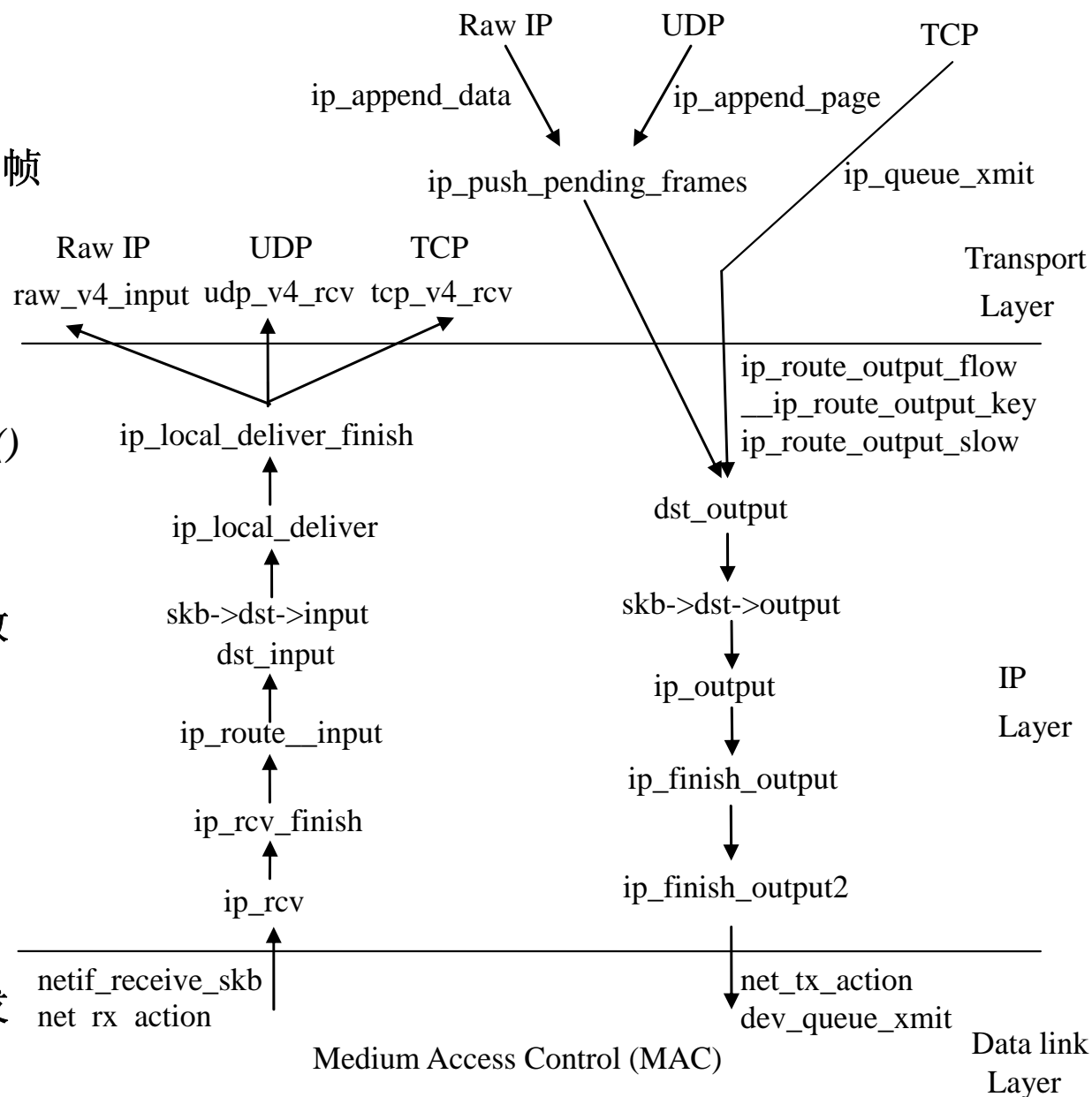
# Linux中接收/发送包的调用图

## • 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

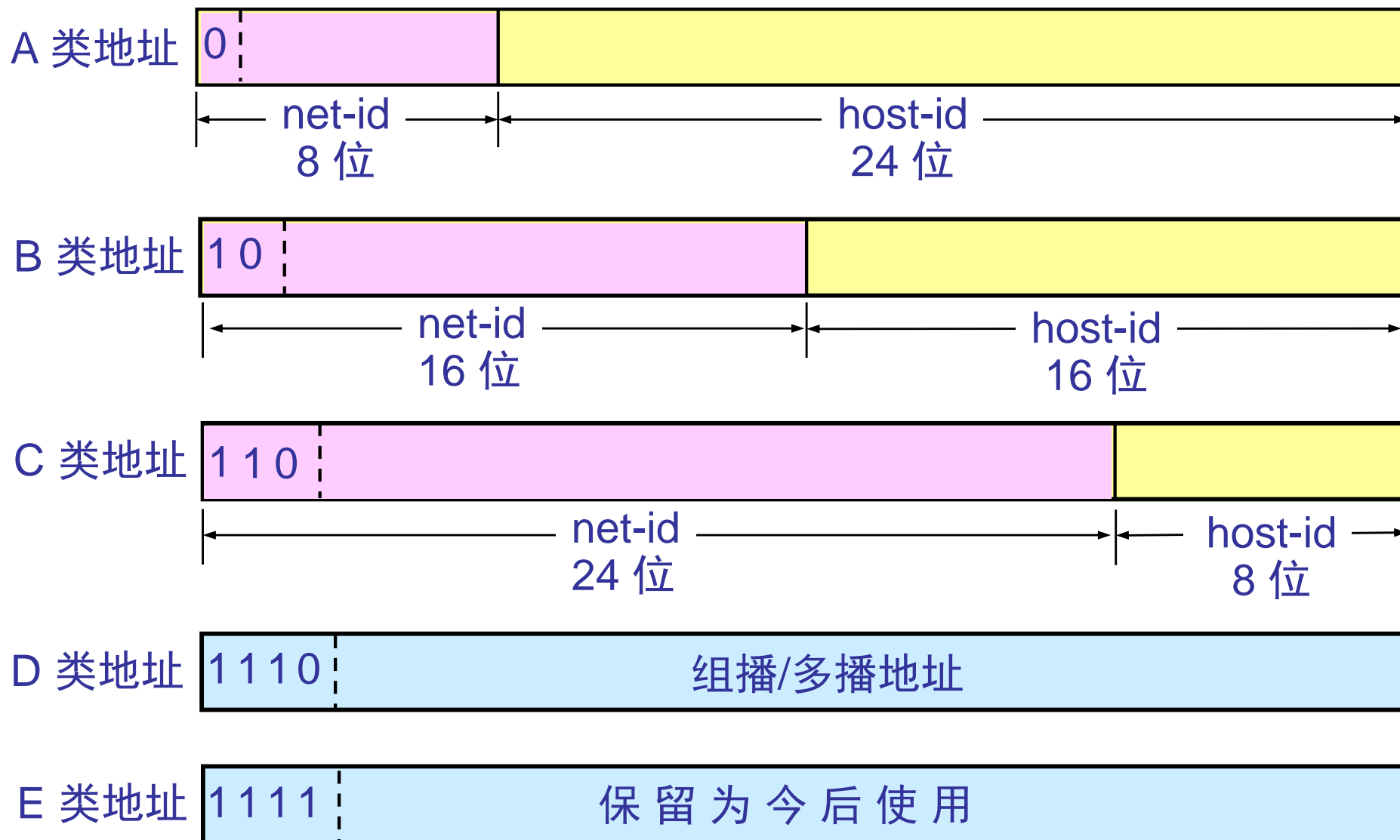
---

### 二、分类的IP 地址

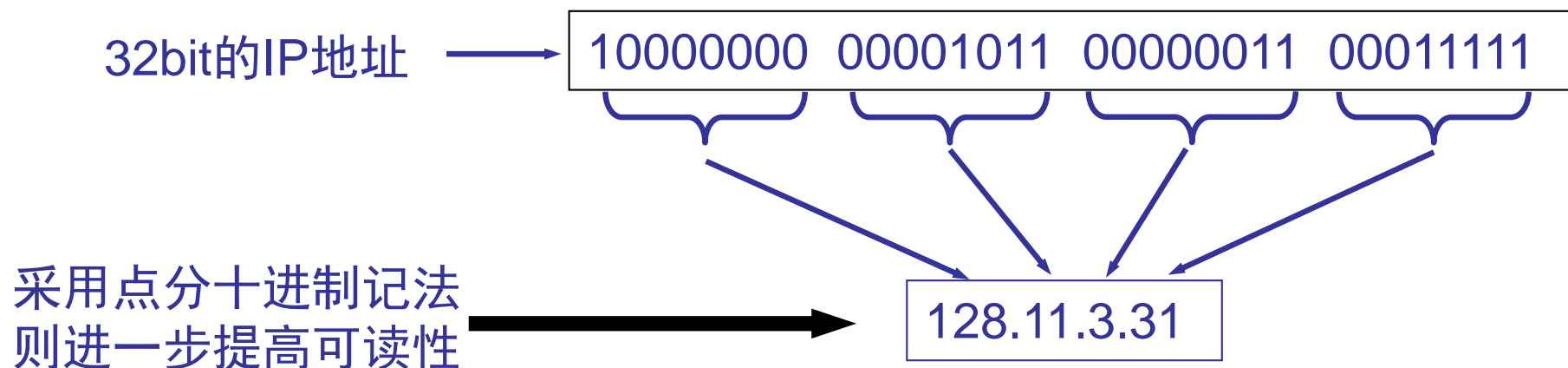
- IP 地址
    - 分配给主机或路由器的标识符，目前使用的IPv4为32位IP地址
    - IP 地址的分配由ICANN (Internet Corporation for Assigned Names and Numbers)负责
  - IP地址的编址方法经历了三个阶段：
    - 分类的 IP 地址：最基本的编址方法，1981 年通过标准
    - 子网的划分：最基本编址方法的改进，1985 年成为标准[RFC 950]
    - 构成超网：比较新的无分类编址方法，1993 年提出
- } 4.3节  
介绍
- 分类的IP地址
    - IP地址被分为A, B, C, 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**

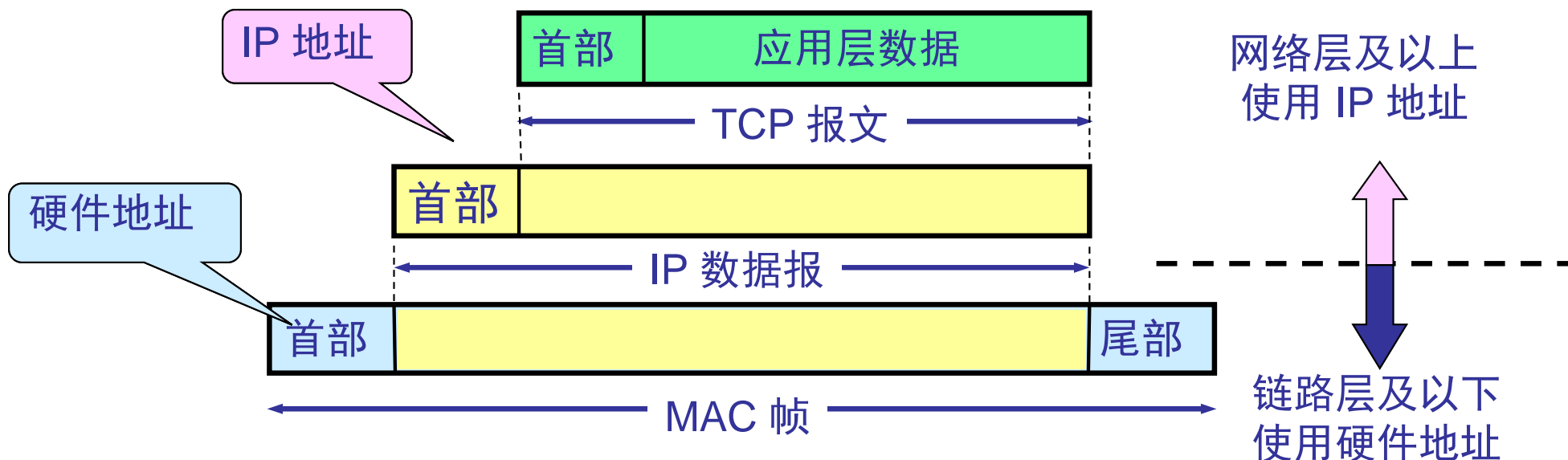
- 全0表示本网络或本主机
- 全1表示广播地址

|   |                                |
|---|--------------------------------|
| 0 | This host                      |
| 0 0      ...      0 0      Host                                 | A host on this network         |
| 1 | Broadcast on the local network |
| Network      1 1 1 1      ...      1 1 1 1                      | Broadcast on a distant network |
| 127      (Anything)   | Loopback                       |

## 4.2 网际协议IP

### 三、IP 地址与硬件地址

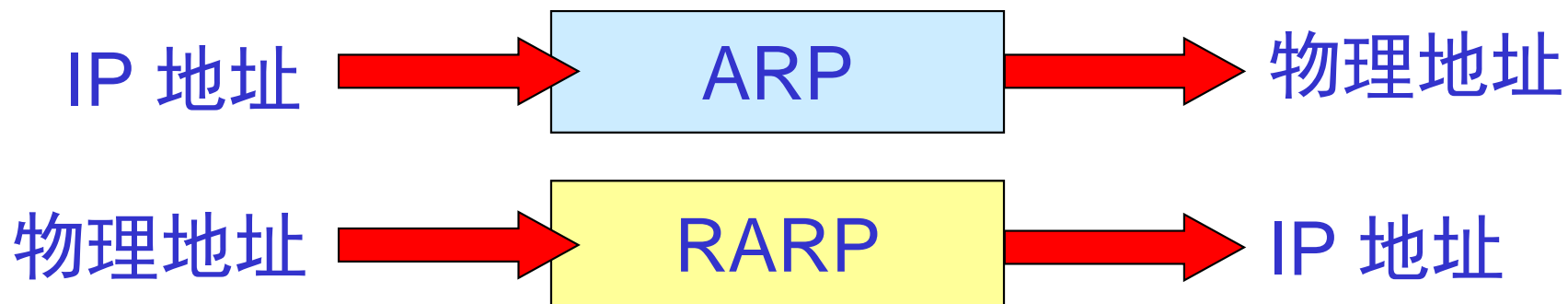
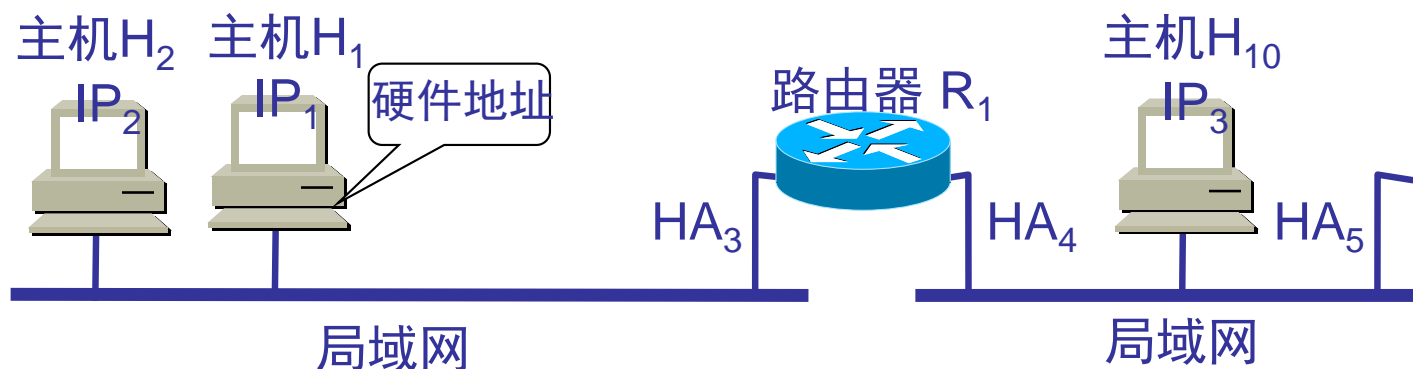
- IP地址
  - 网络层及以上各层使用的地址，是一种逻辑地址
  - 存放在IP包头部
- 物理地址
  - 数据链路层及物理层使用的地址
  - 存放在数据链路层的帧中
    - 问题：帧中有无IP地址？



## 4.2 网际协议IP

### 四、ARP与RARP协议

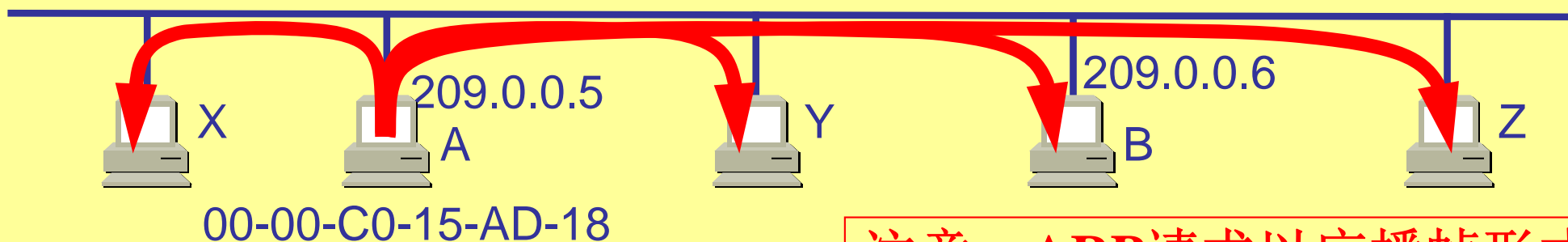
- IP 地址与物理地址的相互转换问题
  - 例：如下图，主机 $H_{10}$ 向主机 $H_1$ 发送了IP包，路由器 $R_1$ 要想在局域网中将IP包发送给主机 $H_1$ ，需知道 $H_1$ 的物理地址
- RFC 826: An Ethernet Address Resolution Protocol



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

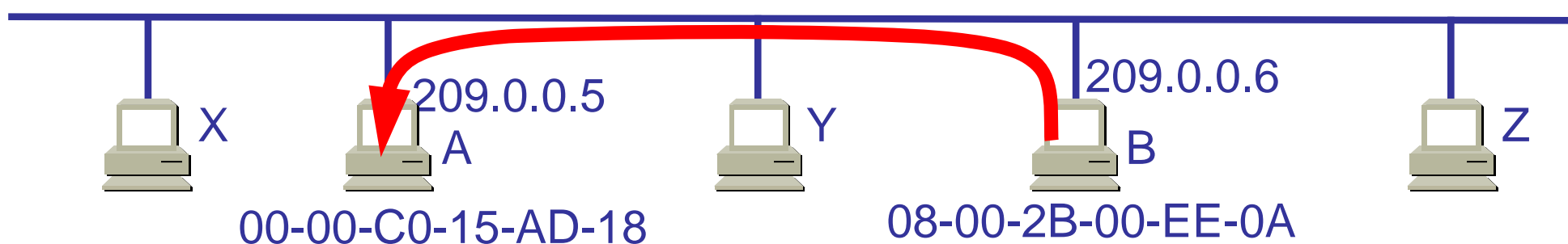


注意：ARP请求以广播帧形式发送

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

