

# Computer Networking Notes

- [1. OVERVIEW](#)

- [1.1. 因特网概述](#)
- [1.2. 电路交换、分组交换和报文交换](#)
- [1.3. 计算机网络的定义和分类](#)
- [1.4. 计算机网络的性能指标](#)
- [1.5. 计算机网络的体系结构](#)
- [1.6. 题目](#)

- [2. Physical layer](#)

- [2.1. 物理层概述](#)
- [2.2. 物理层下面的传输媒体](#)
- [2.3. 传输方式](#)
- [2.4. 编码与调制](#)
- [2.5. 信道的极限容量](#)
- [2.6. 信道复用技术](#)
- [2.7. 题目](#)

- [3. data link layer](#)

- [3.1. 数据链路层概述](#)
- [3.2. 数据链路层的三个重要问题](#)
- [3.3. 点对点协议](#)
- [3.4. 共享式以太网](#)
- [3.5. 交换式以太网](#)
- [3.6. 以太网的MAC帧格式](#)
- [3.7. 虚拟局域网](#)
- [3.8. 以太网的发展](#)
- [3.9. 无线局域网](#)
- [3.10. 题目](#)

- [4. Network layer](#)

- [4.1. 网络层概述](#)
- [4.2. 网际协议IP](#)
- [4.3. 静态路由配置](#)
- [4.4. 因特网的路由选择协议](#)
- [4.5. 网际控制协议ICMP（封装在IP）](#)
- [4.6. 虚拟专用网VPN和网络地址转换NAT](#)
- [4.7. IP多播技术](#)
- [4.8. 移动IP技术概述](#)
- [4.9. IPv6](#)
- [4.10. 软件定义网络SDN](#)
- [4.11. 题目](#)

- [5. Transport layer](#)

- [5.1. 运输层概述](#)
- [5.2. UDP和TCP的对比](#)
- [5.3. 传输控制协议](#)
- [5.4. 题目](#)

- [6. Application layer](#)
  - [6.1. 应用层概述](#)
  - [6.2. 客户/服务器方式和对等方式](#)
  - [6.3. 动态主机配置协议DHCP](#)
  - [6.4. 域名系统DNS\(Domain Name System\)](#)
  - [6.5. 文件传送协议FTP\(File Transfer Protocol\)](#)
  - [6.6. 电子邮件](#)
  - [6.7. 万维网WWW](#)
  - [6.8. 题目](#)

## 1. OVERVIEW

### 1.1. 因特网概述

#### 1.1.1. 网络、互联网、因特网

- 网络 (Network) : 结点 (Node) + 链路 (Link)
- 互联网 (互连网) : 网络的网络 (Network of Networks)，可以通过路由器互连起来
- 因特网 (Internet) : 当今世界上最大的互联网

internet	对比	Internet
通用名词		专用名词
互连网（互联网）		因特网
任意通信协议		TCP/IP协议族

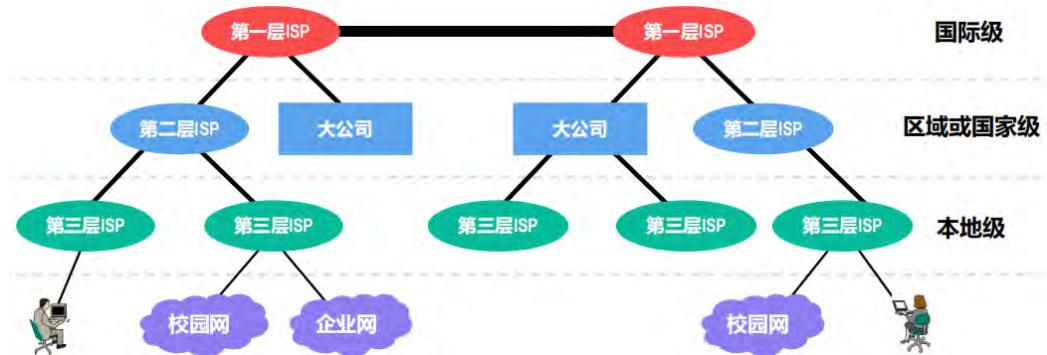
#### 1.1.2. 因特网发展的三个阶段



- 第一个分组交换网: ARPANET, 1969
- 因特网诞生时间: 1983年

#### 1.1.3. 因特网服务提供者 (Internet Service Provider, ISP)

#### 1.1.4. 因特网已发展成为基于ISP的多层次结构的互连网络



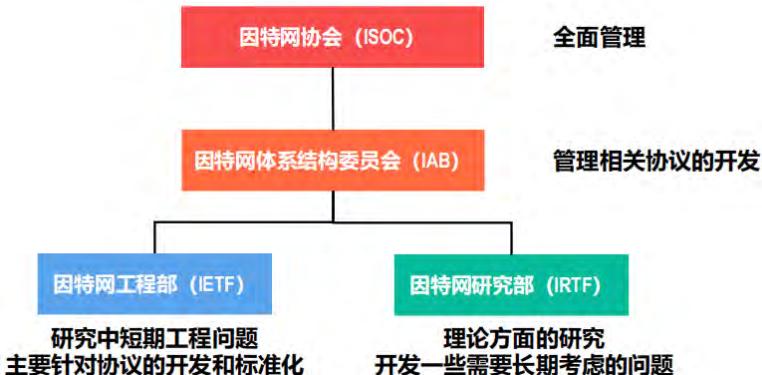
### 1.1.5. 因特网的标准化工作

- 特点：面向公众，其任何一个建议标准在成为因特网标准之前都以 RFC 技术文档的形式在因特网上发表
  - RFC (Request For Comments) 的意思是“请求评论”。任何人都可以从因特网上免费下载RFC文档 (<http://www.ietf.org/rfc.html>)，并随时对某个 RFC 文档发表意见和建议。

### 1.1.6. 制定因特网正式标准的四个阶段

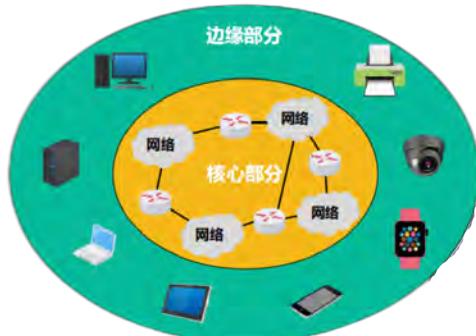


### 1.1.7. 因特网的管理机构



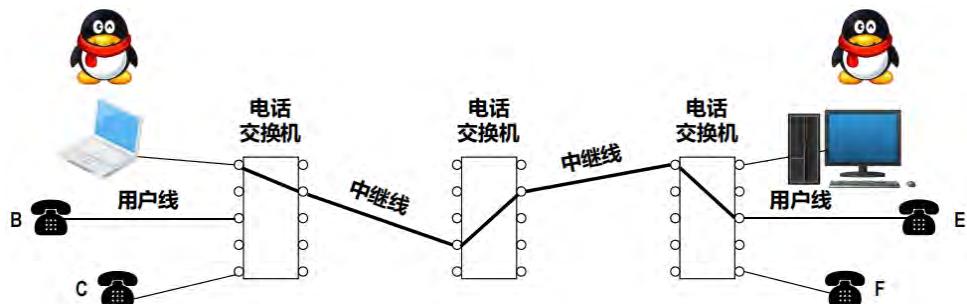
### 1.1.8. 因特网的组成：边缘部分 + 核心部分

- 边缘：主机，用户直接使用，用于通信和资源共享
- 核心：大量网络 + 路由器，为边缘部分提供服务



## 1.2. 电路交换、分组交换和报文交换

### 1.2.1. 电路交换



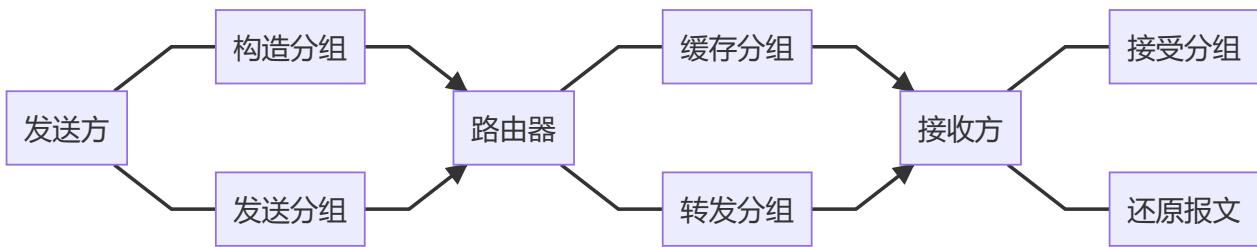
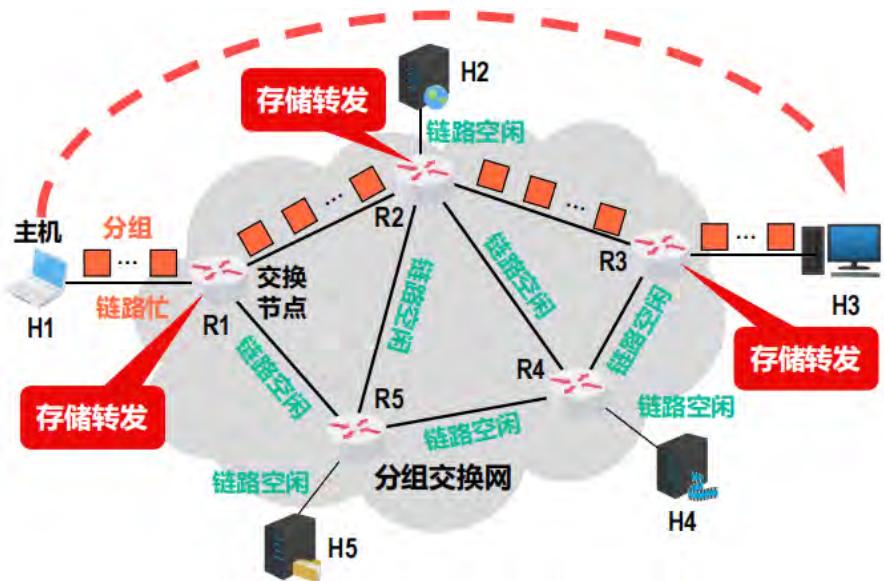
- 方式：电话交换机接通电话线

- step

- 建立连接（分配通信资源）
- 通话（一直占用通信资源）
- 释放连接（归还通信资源）

- 线路的传输效率低

### 1.2.2. 分组交换



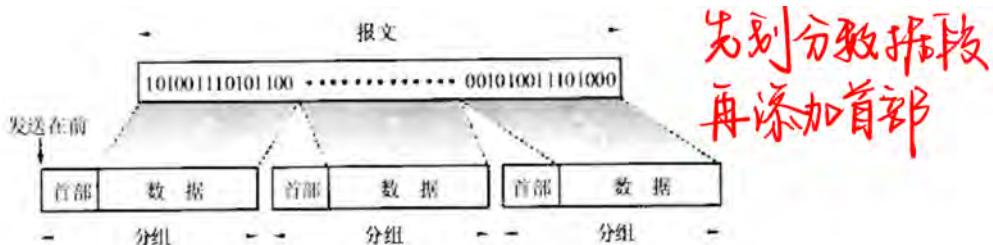


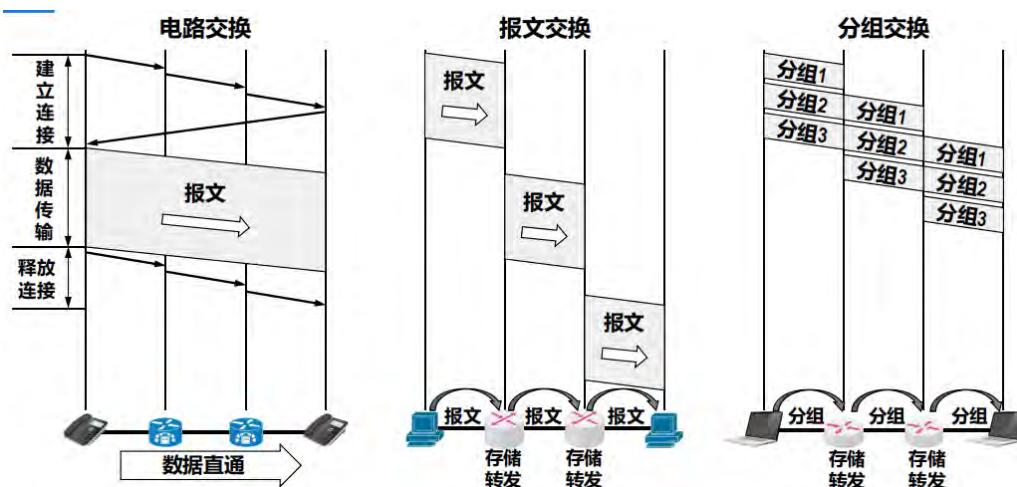
图 1-9 以分组为基本单位在网络中传送

优点	缺点
没有建立连接和释放连接的过程。	分组首部带来了额外的传输开销。
分组传输过程中逐段占用通信链路，有较高的通信线路利用率。	交换节点存储转发分组会造成一定的时延。
交换节点可以为每一个分组独立选择转发路由，使得网络有很好的生存性。	无法确保通信时端到端通信资源全部可用，在通信量较大时可能造成网络拥塞。
有利于差错控制，具有更好的灵活性	分组可能会出现失序和丢失等问题

### 1.2.3. 报文交换

- 报文交换是分组交换的前身
- 方式：报文被整个地发送、整体接收完成后才能转发
- 缺点：引起转发时延、需要较大存储缓存空间

### 1.2.4. 三种交换方式的对比



- 报文交换和分组交换都不需要建立连接，传送突发数据时可以提高通信线路利用率

## 1.3. 计算机网络的定义和分类

### 1.3.1. 计算机网络的定义

- 没有精确和统一的定义

#### 1.3.1.1. 早期的简单定义

互联、自治的计算机集合

### 1.3.1.2. 现阶段的一个较好定义

计算机网络主要是由一些通用的、可编程的硬件互连而成的

- 这些硬件
  - 并非专门用来实现某一特定目的（例如，传送数据或视频信号）
  - 而是能够用来传送多种不同类型的数据，并能支持广泛的和日益增长的应用。

### 1.3.2. 计算机网络的分类

#### 1.3.2.1. 交换方式

电路交换、报文交换、分组交换

#### 1.3.2.2. 使用者

公用网（因特网）、专用网（军队、铁路、电力、银行）

#### 1.3.2.3. 传输介质

有线网络、无线网络

#### 1.3.2.4. 覆盖范围

WAN、MAN、LAN、PAN

#### 1.3.2.5. 拓扑结构

总线形、星形、环形、网状形

## 1.4. 计算机网络的性能指标

### 1.4.1. 速率

数据的传输速率，也称数据率/比特率

数据量的单位	换算关系
比特 (b)	基本单位
字节 (B)	$1B = 8bit$
千字节 (KB)	$KB = 2^{10}B$
兆字节 (MB)	$MB = K \cdot KB = 2^{20}B$
吉字节 (GB)	$GB = K \cdot MB = 2^{30}B$
太字节 (TB)	$TB = K \cdot GB = 2^{40}B$

速率的单位	换算关系
比特/秒 (b/s)	基本单位
干比特/秒 (kb/s)	$kb/s = 10^3 b/s$
兆比特/秒 (Mb/s)	$Mb/s = k \cdot kb/s = 10^6 b/s$
吉比特/秒 (Gb/s)	$Gb/s = k \cdot Mb/s = 10^9 b/s$
太比特/秒 (Tb/s)	$Tb/s = k \cdot Gb/s = 10^{12} b/s$

### 1.4.2. 带宽

#### 带宽在模拟信号系统中的意义

- 某个信号所包含的各种不同频率成分所占据的频率范围。
- 单位：Hz (kHz, MHz, GHz)。

#### 带宽在计算机网络中的意义

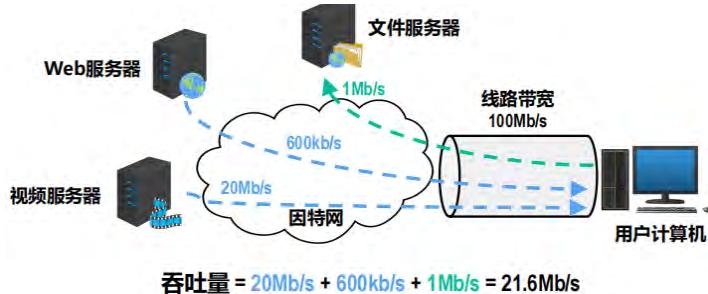
- 用来表示网络的通信线路所能传送数据的能力，即在单位时间内从网络中的某一点到另一点所能通过的最高数据率。
- 单位：b/s (kb/s, Mb/s, Gb/s, Tb/s)。

数据传送速率 =  $\min$  [主机接口速率, 线路带宽, 交换机或路由器的接口速率]

### 1.4.3. 吞吐量

单位时间内通过某个网络或接口的实际数据量

- 常被用于对实际网络的测量，以便获知到底有多少数据量通过了网络。
- 受网络带宽的限制



### 1.4.4. 时延

指数据从网络的一端传送到另一端所耗费的时间，也称为延迟或迟延。

- 数据构成：一个或多个分组、甚至是一个比特

#### 1.4.4.1. 发送时延

$$\text{发送时延} = \frac{\text{分组长度}(b)}{\text{发送速率}(b/s)}$$

发送速率 =  $\min[\text{主机接口速率}, \text{线路带宽}, \text{交换机或路由器的接口速率}]$

#### 1.4.4.2. 传播时延

$$\text{传播时延} = \frac{\text{信道长度}(m)}{\text{信号传播速率}(m/s)}$$

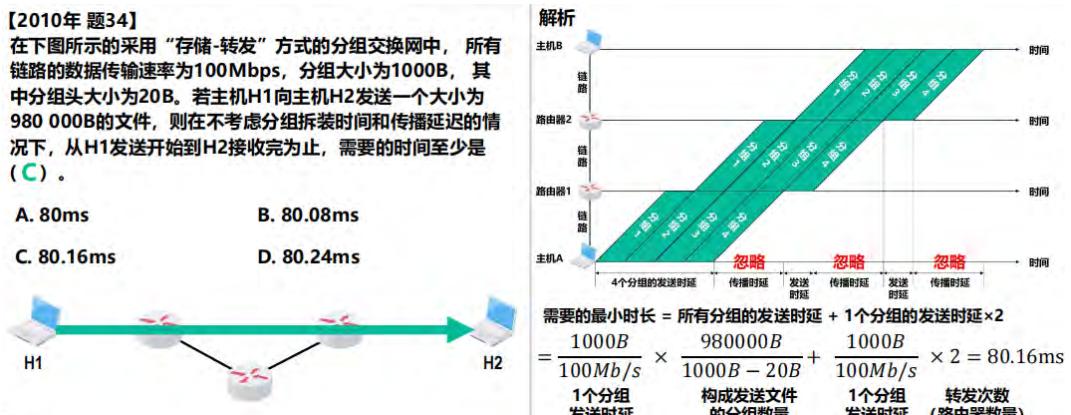
介质	速率
自由空间	$3.0 \times 10^8 \text{ m/s}$
铜线	$2.3 \times 10^8 \text{ m/s}$
光纤	$2.0 \times 10^8 \text{ m/s}$

#### 1.4.4.3. 排队时延

不方便计算、不考虑

#### 1.4.4.4. 处理时延

不方便计算、不考虑

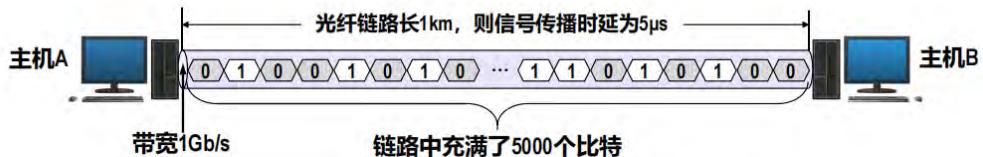


#### 1.4.5. 时延带宽积

传播时延和带宽的乘积，也称为以比特为单位的链路长度

**【举例】** 主机A和B之间采用光纤链路，链路长1km，链路带宽为1Gb/s，请计算该链路的时延带宽积。

$$\text{时延带宽积} = \frac{1\text{km}}{2 \times 10^8 \text{m/s}} \times 1\text{Gb/s} = 5000\text{b}$$



#### 1.4.6. 往返时间 (RTT)

指从发送端发送数据分组开始，到发送端收到接收端发来的相应确认分组为止，总共耗费的时间。

#### 1.4.7. 利用率

##### 1.4.7.1. 链路利用率

指某条链路有百分之几的时间是被利用的（即有数据通过）。

- 当某链路的利用率增大时，该链路引起的时延就会迅速增加。

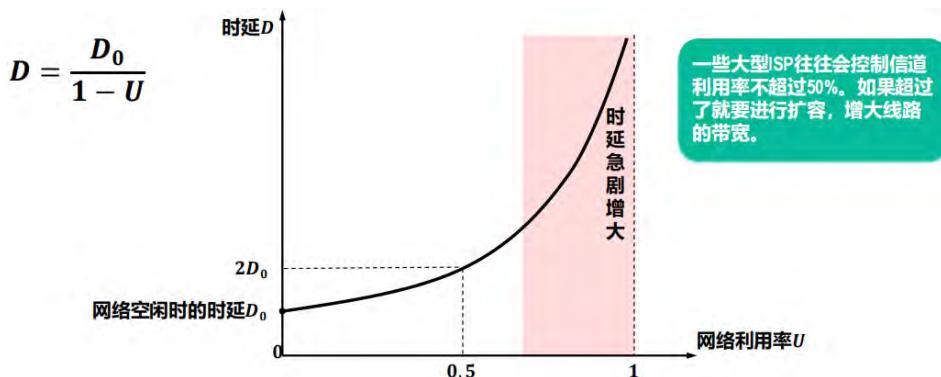
##### 1.4.7.2. 网络利用率

指网络中所有链路的链路利用率的加权平均。

- 在网络通信量不断增大时，分组在交换节点（路由器或交换机）中的排队时延会随之增大，因此网络引起的时延就会增大。

令 $D_0$ 表示网络空闲时的时延， $D$ 表示网络当前的时延，那么在理想的假定条件下，可用下式来表示 $D$ 、 $D_0$ 和网络利用率 $U$ 之间的关系。

$$D = \frac{D_0}{1 - U}$$



#### 1.4.8. 丢包率

是指在一定的时间范围内，传输过程中丢失的分组数量与总分组数量的比率。

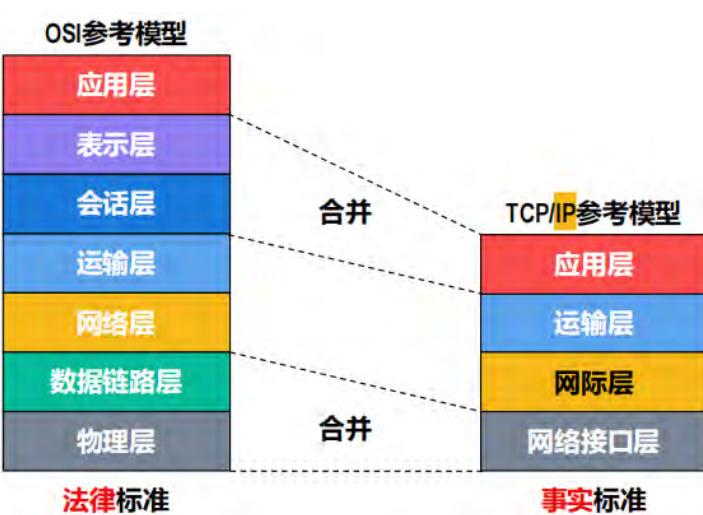
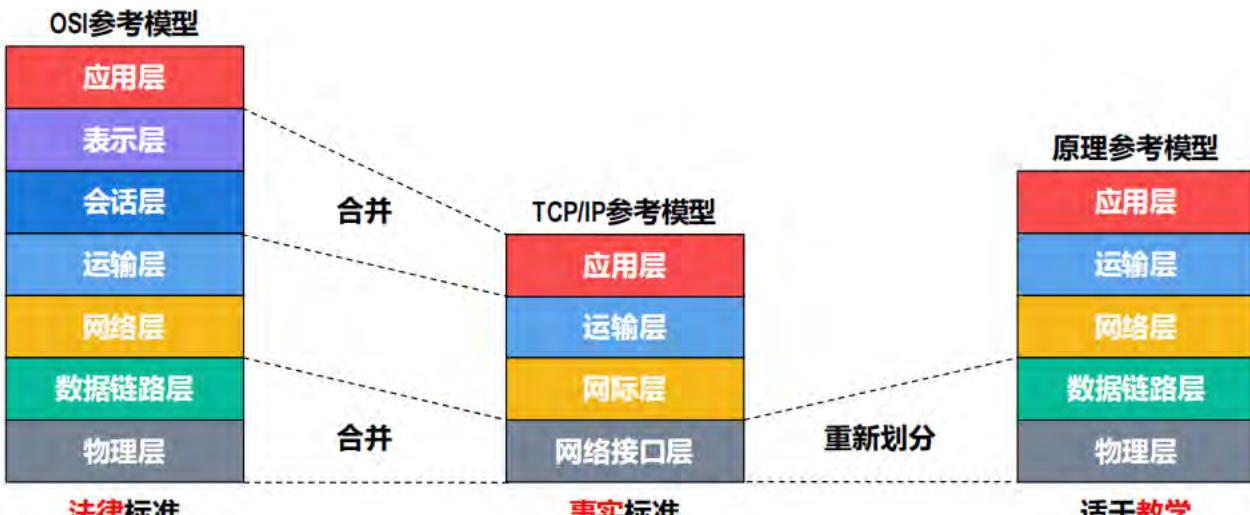
- 分类：接口丢包率、结点丢包率、链路丢包率、路径丢包率、网络丢包率
- 分组丢失主要有以下两种情况：
  - 分组在传输过程中出现误码，被传输路径中的节点交换机（例如路由器）或目的主机检测出误码而丢弃。

- 节点交换机根据丢弃策略主动丢弃分组。

## 1.5. 计算机网络的体系结构

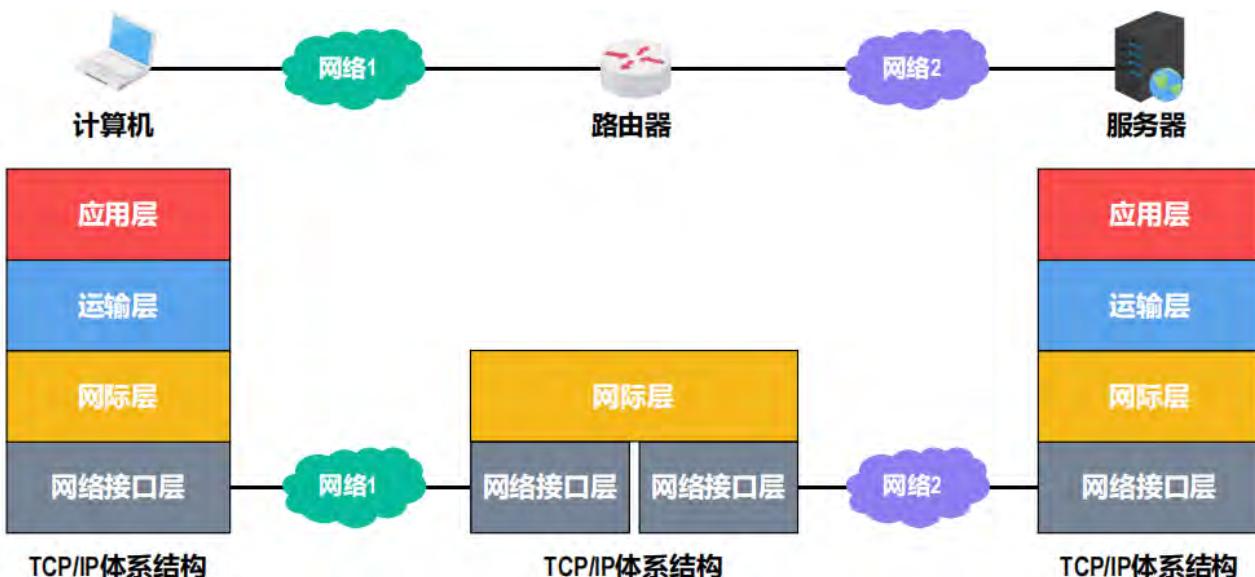
### 1.5.1. 常见的三种计算机网络体系结构

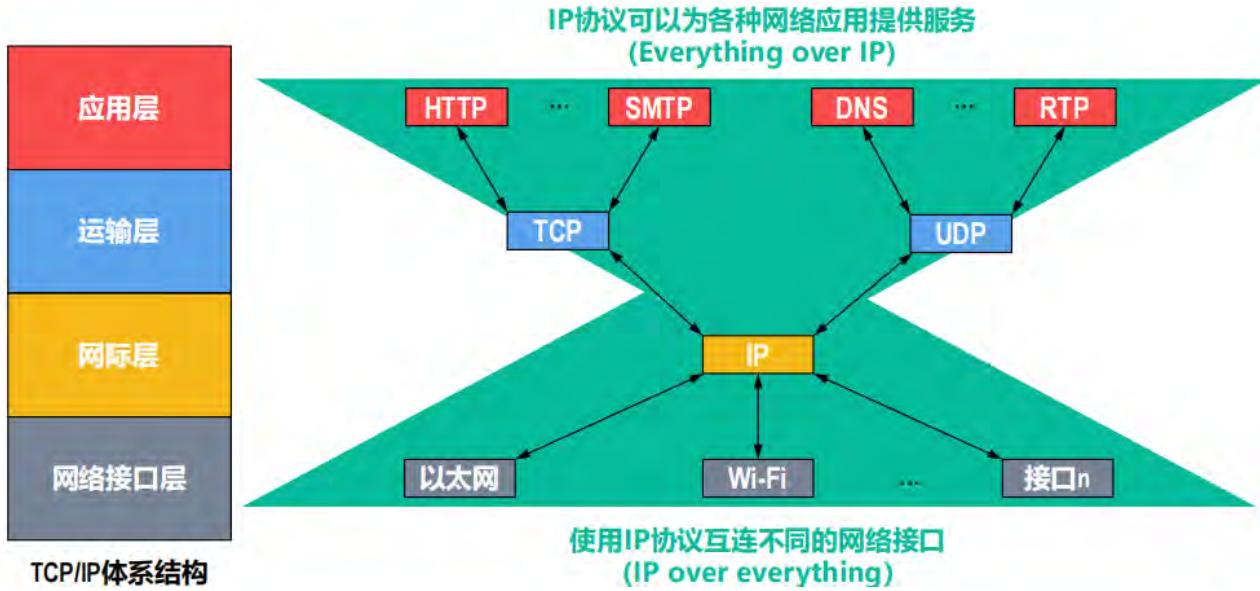
OSI参考模型、TCP/IP参考模型、原理参考模型



### OSI标准失败的原因

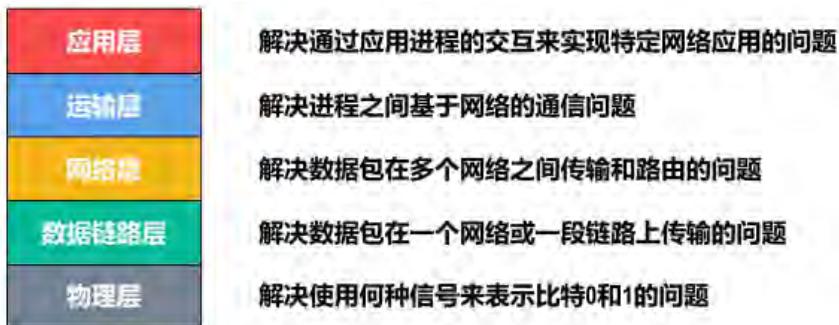
- 01 专家没有实际经验  
完成标准时没有商业驱动力
- 02 协议实现过分复杂  
运行效率很低
- 03 标准的制定周期太长  
产品无法及时进入市场
- 04 层次划分不太合理  
有些功能在多个层次中重复出现





### 1.5.2. 分层的必要性

- 计算机网络是个非常复杂的系统
- “分层”可将庞大复杂的问题转化为若干较小的局部问题



#### 1.5.2.1. 物理层

- 解决使用何种信号来表示比特0和1的问题
  - 采用什么传输媒体（介质）
  - 采用什么物理接口
  - 采用什么信号表示比特0和1

#### 1.5.2.2. 数据链路层

- 解决数据包在一个网络或一段链路上传输的问题
  - 标识网络中各主机（主机编址，例如MAC地址）
  - 从比特流中区分出地址和数据（数据封装格式）
  - 协调各主机争用总线（媒体接入控制）
  - 以太网交换机的实现（自学习和转发帧）
  - 检测数据是否误码（差错检测）
  - 出现传输差错如何处理（可靠传输和不可靠传输）
  - 接收方控制发送方注入网络的数据量（流量控制）

### 1.5.2.3. 网络层

- 解决数据包在多个网络之间传输和路由的问题
  - 标识网络和网络中的各主机（网络和主机共同编址，例如IP地址）
  - 路由器转发分组（路由选择协议、路由表和转发表）

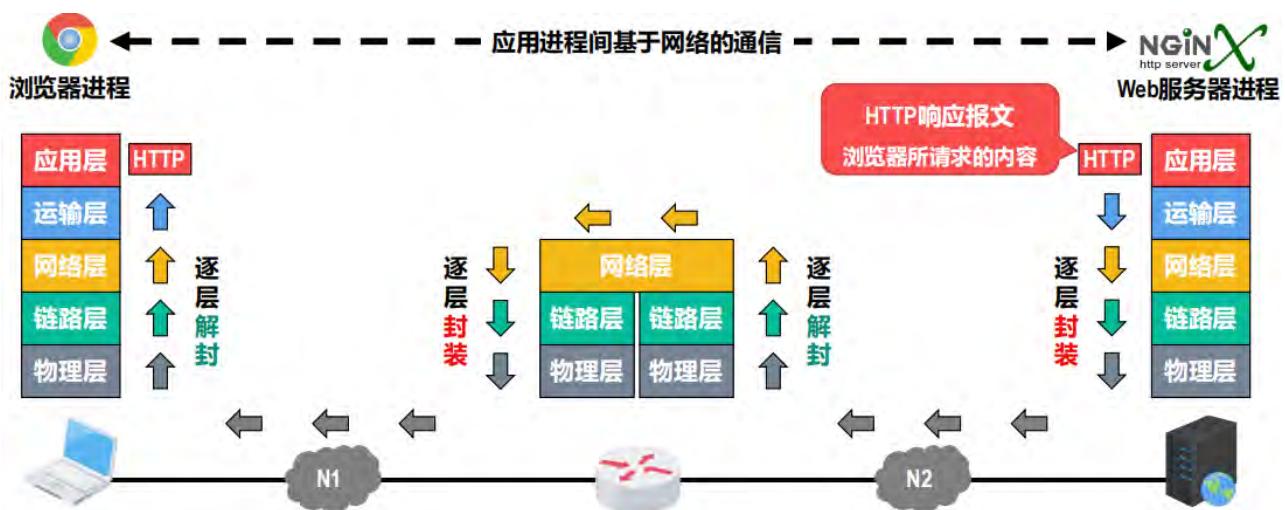
### 1.5.2.4. 运输层

- 解决进程之间基于网络的通信问题
  - 进程之间基于网络的通信（进程的标识，例如端口号）
  - 出现传输差错如何处理（可靠传输和不可靠传输）

### 1.5.2.5. 应用层

- 解决通过应用进程的交互来实现特定网络应用的问题
  - 通过应用进程间的交互来完成特定的网络应用
  - 进行会话管理和数据表示

### 1.5.3. 分层思想举例



### 1.5.4. 专用术语

#### 1.5.4.1. 实体

指任何可发送或接收信息的硬件或软件进程

##### 1.5.4.1.1. 对等实体

指通信双方相同层次中的实体。



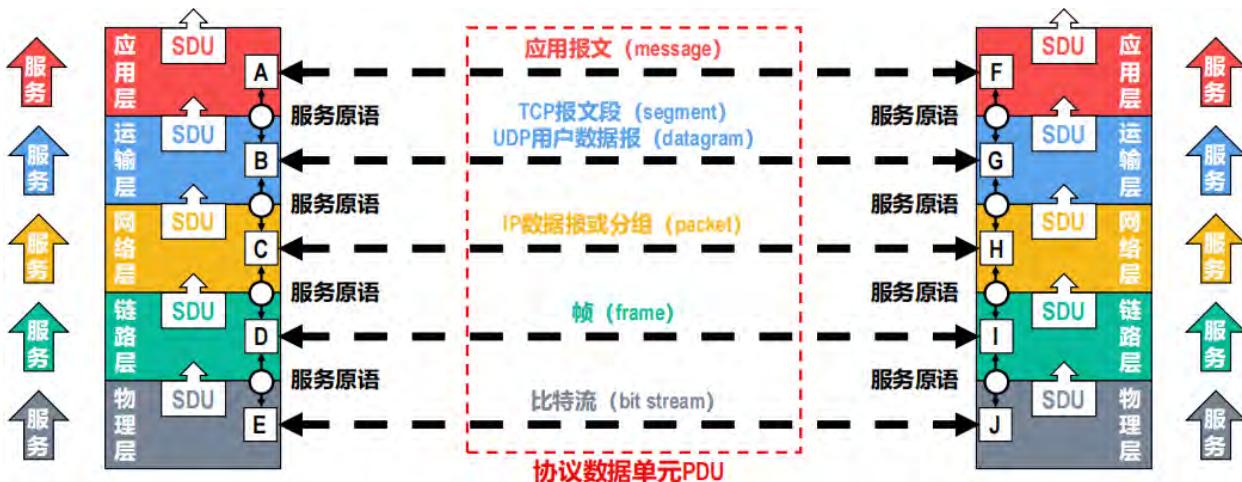
#### 1.5.4.2. 协议

是控制两个对等实体在“水平方向”进行“逻辑通信”的规则的集合。

- 三要素
  - 语法：定义所交换信息的格式
  - 语义：定义通信双方所要完成的操作
  - 同步：定义通信双方的时序关系

#### 1.5.4.3. 服务

- 在协议的控制下，两个对等实体在水平方向的逻辑通信使得本层能够向上一层提供服务。
- 要实现本层协议，还需要使用下面一层所提供的服务
- 协议是“水平”的，而服务是“垂直”的。
- 实体看得见下层提供的服务，但并不知道实现该服务的具体协议。
- 下层的协议对上层的实体是“透明”的。
- 服务访问点SAP
  - 在同一系统中相邻两层的实体交换信息的逻辑接口，用于区分不同的服务类型。
  - 帧的“类型”字段、IP数据报的“协议”字段，TCP报文段或UDP用户数据报的“端口号”字段都是SAP。
- 服务原语
  - 上层要使用下层所提供的服务时通过与下层交换的一些命令
- 协议数据单元 (Protocol Data Unit, PDU)
  - 对等层次之间传送的数据包
- 服务数据单元 (Service Data Unit, SDU)
  - 同一系统内层与层之间交换的数据包



## 1.6. 题目

### 1.6.1. 体系结构相关

1.6.1.1. 【2009 33】补充：OSI表示层与会话层的功能、数据链路层、网络层、运输层作用范围

【2009年题33】在OSI参考模型中，自下而上第一个提供端到端服务的层次是 **B**

- A. 数据链路层      B. 传输层      C. 会话层      D. 应用层

⑦ 应用层	解决通过应用进程之间的交互来实现特定网络应用的问题
⑥ 表示层	解决通信双方交换信息的表示问题
⑤ 会话层	解决进程之间进行会话问题
④ 运输层	解决进程之间基于网络的通信问题
③ 网络层	解决分组在多个网络之间传输（路由）的问题
② 数据链路层	解决分组在一个网络（或一段链路）上传输的问题
① 物理层	解决使用何种信号来传输比特0和1的问题



1.6.1.2. 【2010 33】网络体系结构描述的内容

【2010年题33】下列选项中，不属于网络体系结构所描述的内容是 C

- A. 网络的层次 ✗
- B. 每一层使用的协议 ✗
- C. 协议的内部实现细节 ✓
- D. 每一层必须完成的功能 ✗



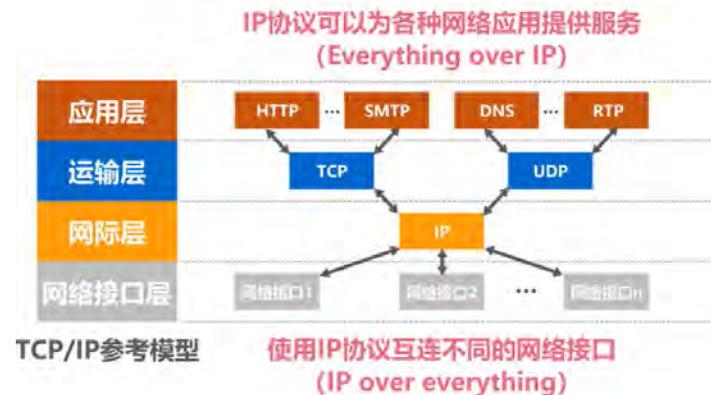
【解析】

计算机网络的体系结构就是计算机网络及其构件所应完成的功能的精确定义。需要强调的是：这些功能的实现细节（例如采用何种硬件或软件），则是遵守这种体系结构的具体实现问题，并不属于体系结构本身所描述的内容。

#### 1.6.1.3. 【2011 33】TCP/IP参考模型的网路层提供无连接不可靠的数据报服务

【2011年题33】TCP/IP参考模型的网络层提供的是 A

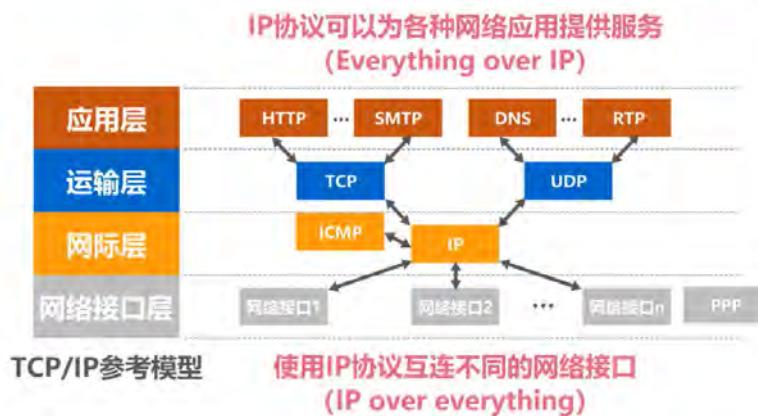
- A. 无连接不可靠的数据报服务
- B. 无连接可靠的报文交换服务
- C. 有连接不可靠的虚电路服务
- D. 有连接可靠的虚电路服务



#### 1.6.1.4. 【2012 33】TCP/IP体系结构中 IP直接为ICMP提供服务

【2012年题33】在TCP/IP体系结构中，直接为ICMP提供服务的协议是 B

- A. PPP
- B. IP
- C. UDP
- D. TCP



### 1.6.1.5. 【2013 33】各层的功能

【2013年 题33】在OSI参考模型中，下列功能需由应用层的相邻层实现的是 B

- A. 对话管理 ⑤ 会话层      B. 数据格式转换      C. 路由选择 ① 网络层      D. 可靠数据传输



【解析】

OSI参考模型应用层的相邻层是表示层。表示层的任务是实现与数据表示相关的功能，主要包括数据字符集的转换、数据格式化、文本压缩、数据加密以及解密等工作。

⑦ 应用层	解决通过应用进程之间的交互来实现特定网络应用的问题
⑥ 表示层	解决通信双方交换信息的表示问题
⑤ 会话层	解决进程之间进行会话问题
④ 运输层	解决进程之间基于网络的通信问题
③ 网络层	解决分组在多个网络之间传输（路由）的问题
② 数据链路层	解决分组在一个网络（或一段链路）上传输的问题
① 物理层	解决使用何种信号来传输比特0和1的问题

OSI参考模型

### 1.6.1.6. 【2014 33】传输层为会话层提供服务

【2014年 题33】在OSI参考模型中，直接为会话层提供服务的是 C

- A. 应用层      B. 表示层      C. 传输层      D. 网络层

【解析】

网络体系结构中的某层为其相邻上层直接提供服务。在OSI参考模型中，传输层为其相邻上层会话层直接提供服务。

⑦ 应用层	解决通过应用进程之间的交互来实现特定网络应用的问题
⑥ 表示层	解决通信双方交换信息的表示问题
⑤ 会话层	解决进程之间进行会话问题
④ 运输层	解决进程之间基于网络的通信问题
③ 网络层	解决分组在多个网络之间传输（路由）的问题
② 数据链路层	解决分组在一个网络（或一段链路）上传输的问题
① 物理层	解决使用何种信号来传输比特0和1的问题

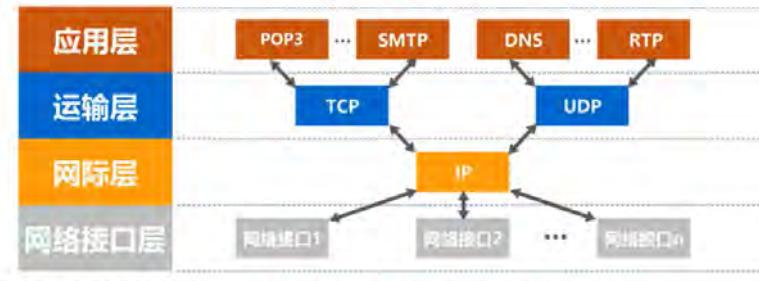
OSI参考模型

### 1.6.1.7. 【2015 33】POP3协议

【2015年 题33】通过POP3协议接收邮件时，使用的传输层服务类型是 D

- |                  |                 |
|------------------|-----------------|
| A. 无连接不可靠的数据传输服务 | B. 无连接可靠的数据传输服务 |
| C. 有连接不可靠的数据传输服务 | D. 有连接可靠的数据传输服务 |

IP协议可以为各种网络应用提供服务  
(Everything over IP)



TCP/IP参考模型

使用IP协议互连不同的网络接口  
(IP over everything)

### 1.6.1.8. 【2016 33】 R1、Switch、Hub

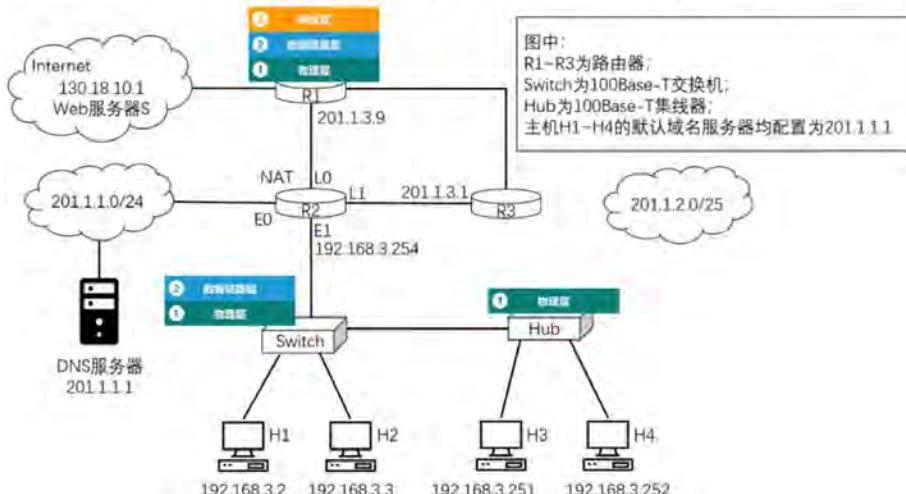
【2016年题33】在OSI参考模型中，R1、Switch、Hub实现的最高功能层分别是 C

A. 2、2、1

B. 2、2、2

C. 3、2、1

D. 3、2、2



### 1.6.1.9. 【2017 33】传输效率

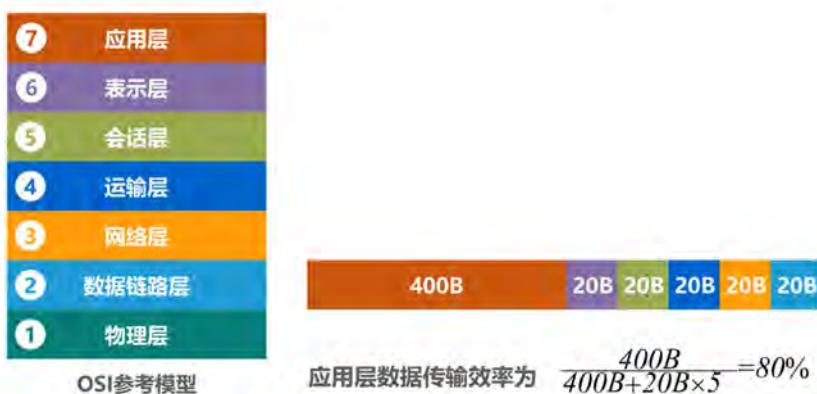
【2017年题33】假设OSI参考模型的应用层欲发送400B的数据（无拆分），除物理层和应用层之外，其他各层在封装PDU时均引入20B的额外开销，则应用层数据传输效率约为 A

A. 80%

B. 83%

C. 87%

D. 91%



### 1.6.1.10. 【2018 33】DNS可以使用传输层无连接服务

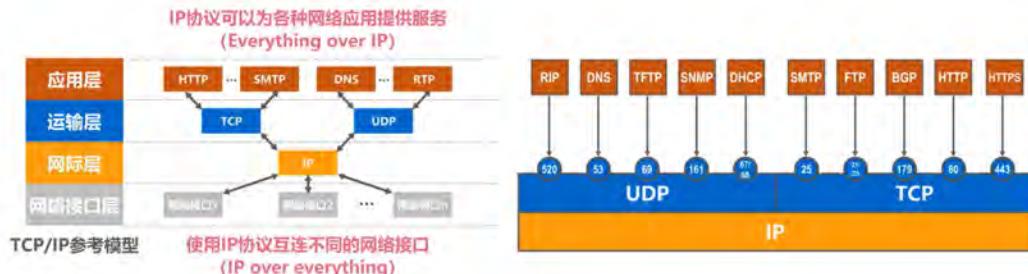
【2018年题33】下列TCP/IP应用层协议中，可以使用传输层无连接服务的是 B

A. FTP

B. DNS

C. SMTP

D. HTTP



### 1.6.1.11. 练习

【练习 1】在OSI参考模型中，提供分组在一个网络（或一段链路）上传输服务的层次是 **B**

- A. 应用层      B. 数据链路层      C. 运输层      D. 网络层

【练习 2】TCP/IP体系结构的网络接口层对应OSI体系结构的 **A**

- |          |         |           |          |
|----------|---------|-----------|----------|
| I. 数据链路层 | II. 物理层 | III. 网络层  | IV. 运输层  |
| A. I、II  | B. I、IV | C. II、III | D. II、IV |

【练习 3】TCP/IP协议族的核心协议是 **C**

- A. TCP      B. UDP      C. IP      D. PPP

【练习 4】在OSI参考模型中，直接为网络层提供服务的是 **D**

- A. 应用层      B. 物理层      C. 运输层      D. 数据链路层

【练习 5】假设OSI参考模型的应用层欲发送600B的数据（无拆分），除应用层之外，其他各层在封装PDU时均引入20B的额外开销，则应用层数据传输效率约为 **C**

- A. 68%      B. 76.8%      C. 83.3%      D. 96%

## 1.6.2. 时延相关

### 1.6.2.1. 【1】

【练习1】有一个待发送的数据块，大小为100MB，网卡的发送速率为100Mbps，则网卡发送完该数据块需要多长时间？

**解析**

**不能直接约分**

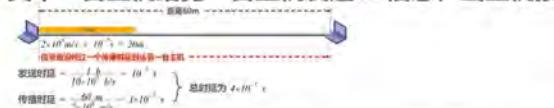
$$\frac{100\text{MB}}{100\text{Mb/s}} = \frac{\text{MB}}{\text{Mb/s}} = \frac{2^{20}B}{10^6b/s} = \frac{2^{20} \times 8b}{10^6b/s} = 8.388608s$$

**估算时  
直接约分**  $\approx \frac{B}{b/s} = \frac{8b}{b/s} = 8s$

### 1.6.2.2. 【2】

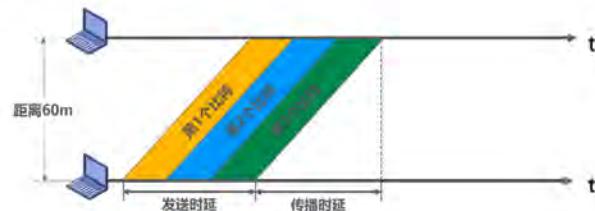
【习题1】两主机间的链路长度为60m，链路带宽为10Mb/s，信号的传播速率为  $2.0 \times 10^8 \text{ m/s}$ ，其中一台主机给另一台主机发送1b信息，当主机接收完该信息时共耗费多长时间？

**【解析】**



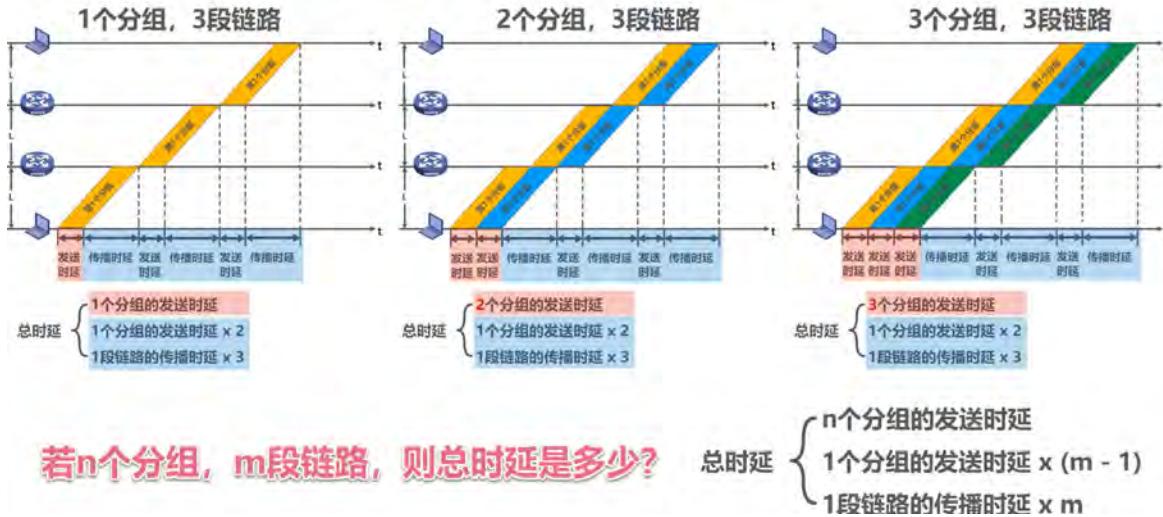
若其中一台主机给另一台主机连续发送n比特信息，当主机接收完该信息时共耗费多长时间？

发送时延 =  $\frac{1\text{b}}{10^6 \text{ b/s}} = 10^{-6} \text{ s}$   
总时延为  $4 \times 10^{-6} \text{ s}$   
传播时延 =  $\frac{n \times 60\text{m}}{2.0 \times 10^8 \text{ m/s}} = 3 \times 10^{-9} \text{ s}$   
接收完n比特的总时延 =  $(4 + 10^{-6}) \times n$



### 1.6.2.2.1. 一般结论

假设：分组等长，各链路长度相同、带宽也相同，忽略路由器的处理时延



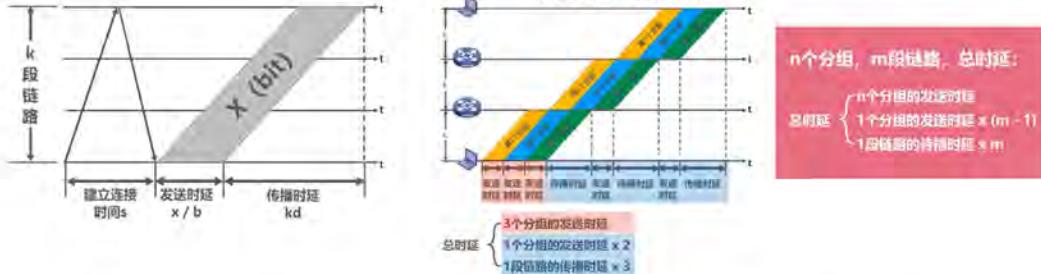
### 1.6.2.3. 【3】解不等式

【习题2】试在下列条件下比较电路交换和分组交换。

要传送的报文共 $x$ (bit)。从源点到终点共经过 $k$ 段链路，每段链路的传播时延为 $d(s)$ ，带宽为 $b$ (bit/s)。在电路交换时，电路的建立时间为 $s(s)$ 。在分组交换时，报文可被划分成若干个长度为 $p$ (bit)的数据段，添加首部后即可构成分组，假设分组首部的长度以及分组在各结点的排队等待时间忽略不计。

问在怎样的条件下，电路交换的时延比分组交换的要大？

【解析】



$$\text{电路交换的时延} = s + \frac{x}{b} + kd$$

$$\text{令电路交换的时延} > \text{分组交换的时延, 解得: } s > (k-1)\frac{p}{b}$$

### 1.6.2.4. 【4】

【习题3】在习题2的分组交换网中，设报文长度和分组长度分别为 $x$ 和 $(p+h)$ (bit)，其中 $p$ 为分组的数据部分的长度，而 $h$ 为每个分组的首部，其长度固定，与 $p$ 的大小无关。通信的两端共经过 $k$ 段链路。链路带宽为 $b$ (bit/s)，但传播时延和结点的排队时间均忽略不计。若打算使总的时延为最小，问分组的数据部分长度 $p$ 应取多大？

【解析】写出总时延D的表达式：

$$D = \frac{p+h}{b} \frac{x}{p} + \frac{p+h}{b} (k-1)$$

一个分组的发送时延      一个分组的发送时延  
分组数据                  转发次数

为了计算D的极值，求D对p的导数，令其为0，如下所示：

$$\frac{k-1}{b} - \frac{xh}{b} \frac{1}{p^2} = 0$$

解出：

$$p = \sqrt{\frac{xh}{k-1}}$$

## 1.6.2.5. 【5】 【2013 35】

例2：【考研 2013年35题】主机甲通过1个路由器（存储转发方式）与主机乙互联，两段链路的数据传输速率为10 Mbps，主机甲分别采用报文交换和分组大小为10 kb的分组交换向主机乙发送1个大小为8 Mb ( $1M=10^6$ ) 的报文。若忽略链路传播延迟、分组头开销和分组拆装时间，则两种交换方式完成该报文传输所需的总时间分别为

A. 800ms, 1600ms

C. 1600ms, 800ms

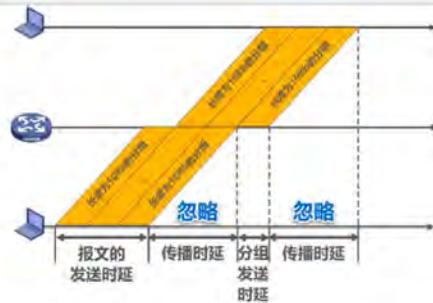
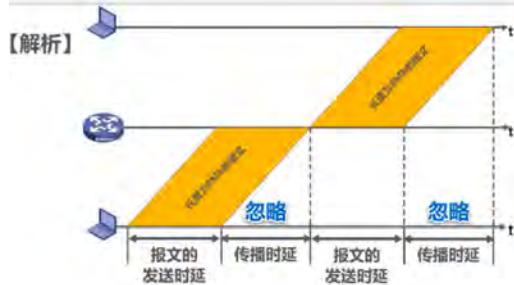
B. 801ms, 1600ms

D. 1600ms, 801ms

$$\text{不分组} \left\{ \begin{array}{l} \text{发送时延: } \frac{8Mb}{10Mbps} = 0.8s = 800ms \\ \text{接收时延: } 800ms \end{array} \right. \rightarrow 1600ms.$$

$$\text{分组后} \left\{ \begin{array}{l} \text{发送: } \frac{10kb}{10Mbps} = 1ms \\ \text{发送第二个时: 第一个开始接收} \\ \text{报文拆装: } 1ms \\ \text{与第1分组: } 801ms \end{array} \right.$$

$$\begin{aligned} & \frac{8Mb}{10kb} = 800 \\ & = \frac{8 \cdot 10^3 kb}{10kb} = 800 \end{aligned}$$



$$\begin{aligned} \text{报文交换总时间} &= \text{报文的发送时延} \times 2 \\ &= \frac{8Mb}{10Mbps} \times 2 = 1.6s = 1600ms \end{aligned}$$

$$\begin{aligned} \text{分组交换总时间} &= \text{报文的发送时延} + 1\text{个分组的发送时延} \\ &= \frac{8Mb}{10Mbps} + \frac{10kb}{10Mbps} = 0.801s = 801ms \end{aligned}$$

## 1.6.2.6. 【6】 【2010 34】

## 【2010年题34】

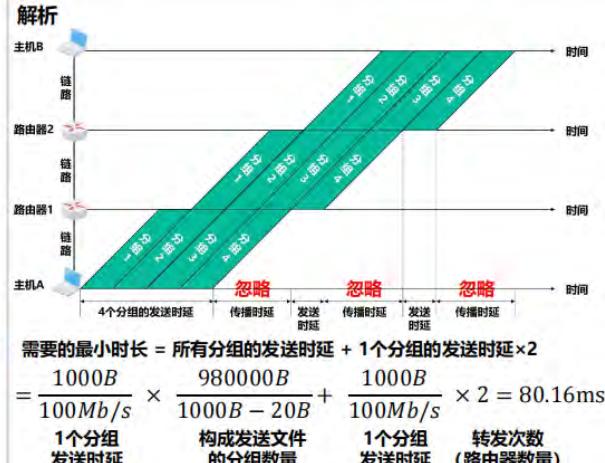
在下图所示的采用“存储-转发”方式的分组交换网中，所有链路的数据传输速率为100Mbps，分组大小为1000B，其中分组头大小为20B。若主机H1向主机H2发送一个大小为980 000B的文件，则在不考虑分组拆装时间和传播延迟的情况下，从H1发送开始到H2接收完为止，需要的时间至少是(C)。

A. 80ms

C. 80.16ms

B. 80.08ms

D. 80.24ms



## 2. Physical layer

## 2.1. 物理层概述

## 2.1.1. 物理层要实现的功能

- “透明”传输比特流
- 数据链路层只管“享受”物理层提供的比特流传输服务，不需知道其具体传输方法

## 2.1.2. 物理层的接口特性

## 2.1.2.1. 机械特性

- 形状和尺寸
- 引脚数目和排列
- 固定和锁定装置

### 2.1.2.2. 电气特性

- 信号电压的范围
- 阻抗匹配的情况
- 传输速率
- 距离限制

### 2.1.2.3. 功能特性

- 规定接口电缆的各条信号线的作用

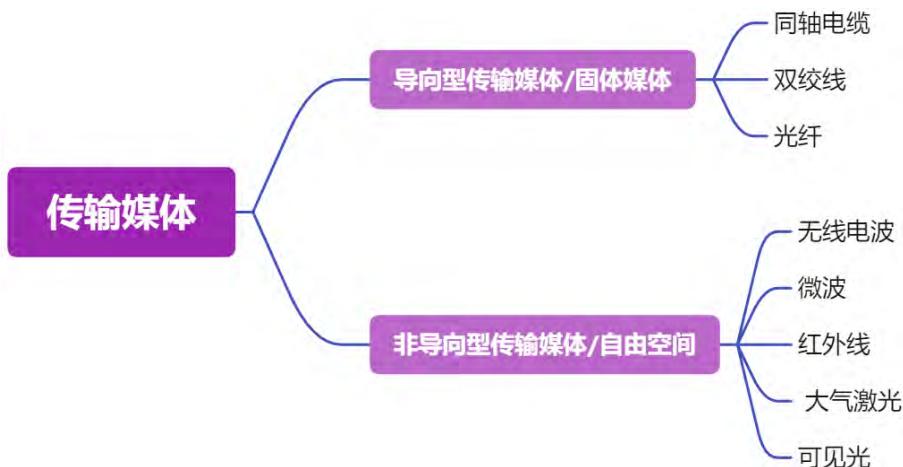
### 2.1.2.4. 过程特性

- 规定在信号线上传输比特流的一组操作过程，包括各信号间的时序关系

## 2.2. 物理层下面的传输媒体

### 2.2.1. 传输媒体的分类

- 传输媒体是计算机网络设备之间的物理通路，也称为传输介质或传输媒介
- 传输媒体并不包含在计算机网络体系结构中



### 2.2.2. 导向型传输媒体

#### 2.2.2.1. 同轴电缆

- 价格较贵且布线不够灵活和方便

##### 2.2.2.1.1. 基带同轴电缆 ( $50\Omega$ )

- 用于数字传输，在早期局域网中广泛使用

##### 2.2.2.1.2. 宽带同轴电缆 ( $75\Omega$ )

- 用于模拟传输，目前主要用于有线电视的入户线。

#### 2.2.2.2. 双绞线

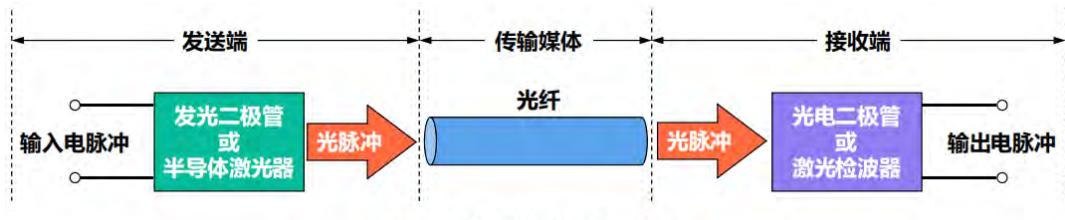
- 局域网领域基本上都采用双绞线作为传输媒体



- 绞合的作用
  - 减少相邻导线间的电磁干扰
  - 抵御部分来自外界的电磁干扰

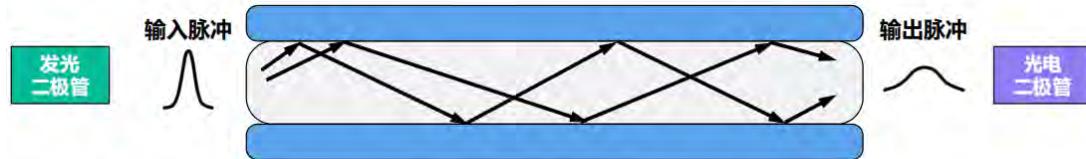
### 2.2.2.3. 光纤

- 光纤通信系统的传输带宽远大于目前其他各种传输媒体的带宽
- 优点
  - 通信容量非常大
  - 抗雷电和电磁干扰性能好
  - 传输损耗小，中继距离长
  - 无串音干扰，保密性好
  - 体积小，重量轻
- 缺点
  - 切割光纤需要较贵的专用设备
  - 目前光电接口还比较昂贵



典型光纤通信系统结构示意图

#### 2.2.2.3.1. 多模光纤



- 多条光波在多模光纤中不断地全反射
- 只适合于建筑物内的近距离传输

#### 2.2.2.3.2. 单模光纤



- 光在单模光纤中一直向前传播
- 适合长距离传输且衰减更小

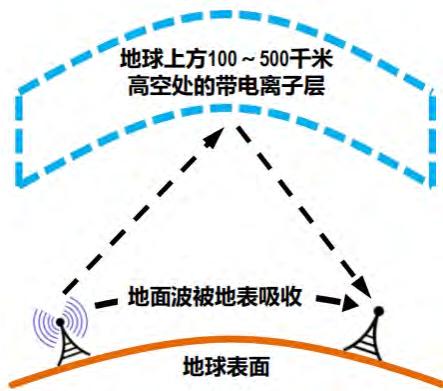
### 2.2.3. 非导向型传输媒体

#### 2.2.3.1. 无线电波

##### 2.2.3.1.1. 中低频



### 2.2.3.1.2. 高频甚高频



### 2.2.3.2. 微波

#### 2.2.3.2.1. 地面微波接力通信



#### 2.2.3.2.2. 卫星通信

- 三颗同步卫星
- 中低轨道同步卫星

### 2.2.3.3. 红外线

- 点对点无线传输
- 直线传输，中间不能有障碍物，传输距离短
- 传输速率低 (4Mb/s ~ 16Mb/s)

### 2.2.3.4. 激光

#### 2.2.3.4.1. 大气激光通信

- 优点
  - 通信容量大
  - 保密性强
  - 结构轻便
  - 设备经济
- 缺点
  - 通信距离受限于视距
  - 易受气候影响
  - 瞄准困难

#### 2.2.3.4.2. 光纤通信

### 2.2.3.5. 可见光

- LIFI可见光通信

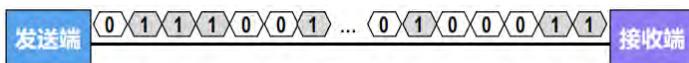
## 2.3. 传输方式

### 2.3.1. 串行传输和并行传输

- 网络同时具有串行传输和并行传输（并串转换、串并转换）

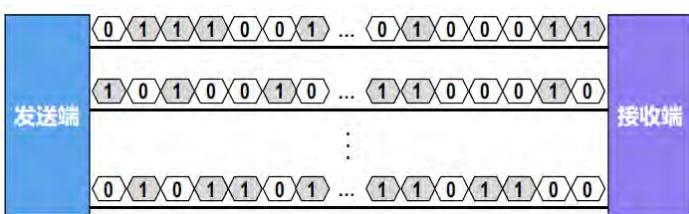
#### 2.3.1.1. 串行传输

- 用于远距离传输



#### 2.3.1.2. 并行传输

- 用于短距离传输



### 2.3.2. 同步传输和异步传输

#### 2.3.2.1. 同步传输

- 收发双方时钟同步的方法
  - 外同步：在收发双方之间增加一条时钟信号线。
  - 内同步：发送端将时钟信号编码到发送数据中一起发送（例如曼彻斯特编码）。

#### 2.3.2.2. 异步传输

- 字节之间异步，即字节之间的时间间隔不固定。
- 字节中的每个比特仍然要同步，即各比特的持续时间是相同的。

### 2.3.3. 单向通信、双向交替通信和双向同时通信

- 单向通信（单工）：无线电广播
- 双向交替通信（半双工）：不能同时、如对讲机
- 双向同时通信（全双工）：可同时、如手机

## 2.4. 编码与调制

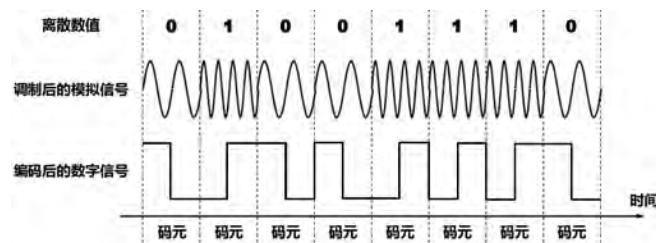
### 2.4.1. 编码与调制的基本概念

#### 2.4.1.1. 调制

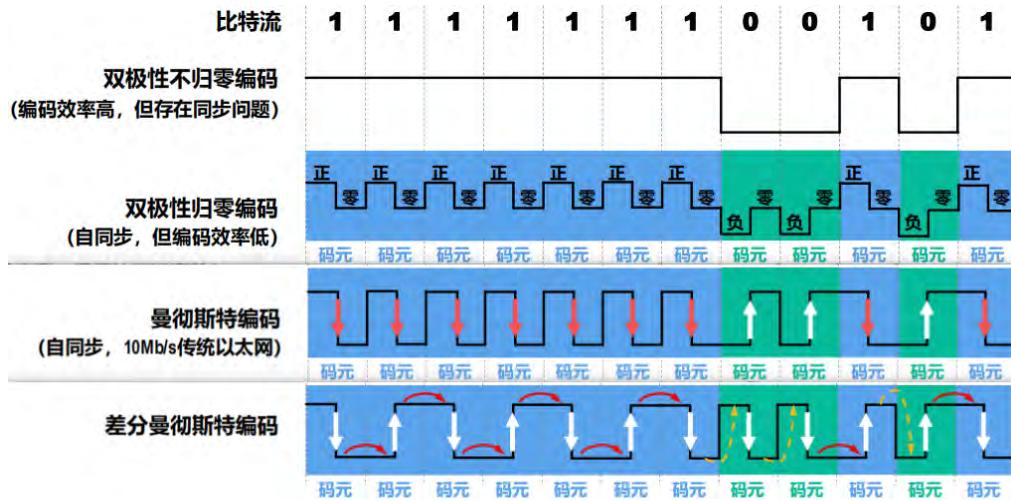
- 基带调制（编码）：数字信道
  - 以太网采用的曼彻斯特编码、4B/5B、8B/10B
- 带通调制：模拟信道
  - Wi-Fi采用的CCK/DSSS/OFDM调制

### 2.4.1.2. 码元

- 在使用时间域的波形表示信号时，代表不同离散数值的基本波形称为码元



### 2.4.2. 常用编码方式



### 2.4.2.1. 双极性不归零编码

- 编码效率高，但存在同步问题

### 2.4.2.2. 双极性归零编码

- 自同步，但编码效率低
- 码元中间时刻的电平跳变既表示时钟信号，也表示数据。
- 正跳变表示1还是0，负跳变表示0还是1，可以自行定义。

### 2.4.2.3. 曼彻斯特编码

- 自同步，10Mb/s传统以太网

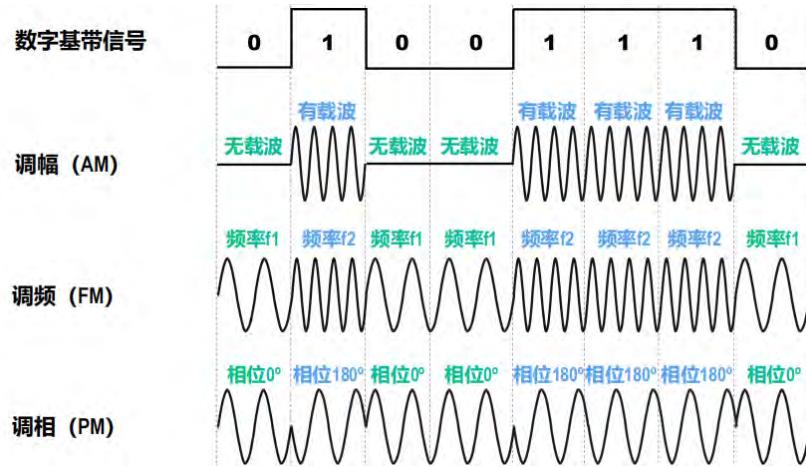
### 2.4.2.4. 差分曼彻斯特编码

- 码元中间时刻的电平跳变仅表示时钟信号，而不表示数据。
- 数据的表示在于每一个码元开始处是否有电平跳变：无跳变表示1，有跳变表示0。

## 2.4.3. 基本的带通调制方法和混合调制方法

### 2.4.3.1. 基本的带通调制方法

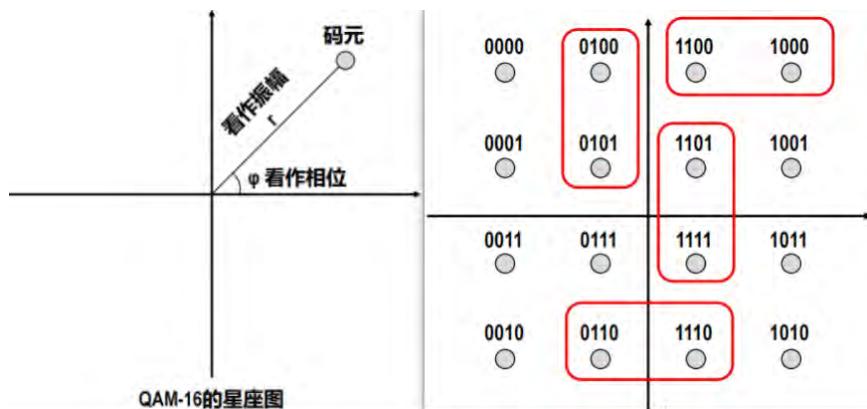
- 调幅 (AM)、调频 (FM)、调相 (PM)



#### 2.4.3.2. 混合调制方法

- 频率和相位不能进行混合调制（频率是相位随时间的变化率）
- 通常情况下，载波的相位和振幅可以结合起来一起调制，例如正交振幅调制QAM。

##### 2.4.3.2.1. 正交振幅调制QAM-16



- 12种相位
- 每种相位有1或2种振幅可选
- 可以调制出16种码元（波形），每种码元可以对应表示4个比特 ( $\log_2 16 = 4$ )
- 每个码元与4个比特的对应关系采用格雷码，即任意两个相邻码元只有1个比特不同

## 2.5. 信道的极限容量

### 2.5.1. 造成信号失真的主要因素

- 信道上传输的数字信号，可以看做是多个频率的模拟信号进行多次叠加后形成的方波。
- 码间串扰
  - 接收端接收到的信号波形失去了码元之间的清晰界限
  - 信道的频带越宽，则能够通过的信号的高频分量就越多，那么码元的传输速率就可以更高，而不会导致码间串扰。
- 信道的频率带宽是有上限的  $\rightarrow$  码元的传输速率也有上限

#### 2.5.1.1. 码元的传输速率

- 越高越严重

### 2.5.1.2. 信号的传输距离

- 越远越严重

### 2.5.1.3. 噪声干扰

- 越大越严重

### 2.5.1.4. 传输媒体的质量

- 越差越严重

## 2.5.2. 奈氏准则

理想低通信道的最高码元传输速率 =  $2W$  Baud

**W**: 信道的频率带宽 (单位为Hz)

**Baud**: 波特, 即码元/秒 =  $2 W$  码元/秒

- 一个实际的信道所能传输的最高码元传输速率, 要明显低于奈氏准则给出的上限值
- 码元传输速率又称为波特率、调制速率、波形速率或符号速率

### 2.5.3. 香农公式

$$C = W \log_2 \left( 1 + \frac{S}{N} \right)$$

- 带宽受限且有高斯白噪声干扰的信道的极限信息传输速率

**C**: 信道的极限信息传输速率 (单位为b/s)

**W**: 信道的频率带宽 (单位为Hz)

**S**: 信道内所传信号的平均功率

**N**: 信道内的高斯噪声功率

$\frac{S}{N}$ : 信噪比, 使用分贝 (dB) 作为度量单位

$$\text{信噪比}(db) = 10 \log_{10} \left( \frac{S}{N} \right) (db)$$

## 2.6. 信道复用技术

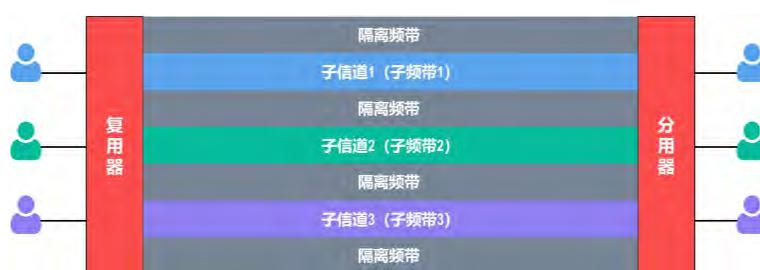
### 2.6.1. 信道复用技术的基本原理

- 复用 (Multiplexing) 就是在一条传输媒体上同时传输多路用户的信号。
- 当一条传输媒体的传输容量大于多条信道传输的总容量时, 就可以通过复用技术, 在这条传输媒体上建立多条通信信道, 以便充分利用传输媒体的带宽。

### 2.6.2. 常见的信道复用技术

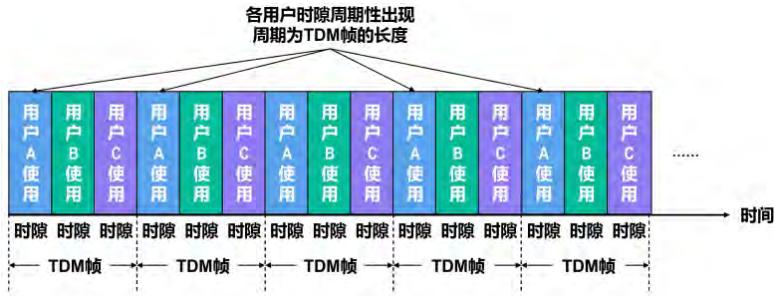
#### 2.6.2.1. 频分复用FDM

- 频分复用的所有用户同时占用不同的频带资源并行通信



### 2.6.2.2. 时分复用TDM

- 时分复用的所有用户在不同的时间占用同样的频带



### 2.6.2.3. 波分复用WDM

- 根据频分复用的设计思想，可在一根光纤上同时传输多个频率（波长）相近的光载波信号，实现基于光纤的频分复用技术。
- 目前可以在一根光纤上复用80路或更多路的光载波信号。因此，这种复用技术也称为密集波分复用DWDM。

### 2.6.2.4. 码分复用CDM

- 码分复用 (Code Division Multiplexing, CDM) 常称为码分多址 (Code Division Multiple Access, CDMA)，它是在扩频通信技术的基础上发展起来的一种无线通信技术。
- 与FDM和TDM不同，CDMA的每个用户可以在相同的时间使用相同的频带进行通信。
- 码片 (Chip)
  - CDMA将每个比特时间划分为m个更短的时间片
  - m的取值通常为64或128
- m比特码片序列 (Chip Sequence)
  - 某个站要发送比特1，则发送它自己的m比特码片序列
  - 某个站要发送比特0，则发送它自己的m比特码片序列的反码
- 码片向量
  - 将码片序列中的比特0记为-1，而比特1记为+1
- 码片序列规则
  - 每个站的码片序列必须各不相同，实际常采用伪随机码序列。
  - 每个站的码片序列必须相互正交，即各码片序列相应的码片向量之间的规格化内积为0。

令向量A表示站A的码片向量，向量B表示站B的码片向量，则

$$A \cdot B = \frac{1}{m} \sum_{i=1}^m A_i B_i = 0$$

各手机用自己的码片向量与收到的叠加后的码片向量，做规格化内积运算：

$$(A + \bar{B}) \cdot A = A \cdot A + A \cdot \bar{B} = 1 + 0 = 1$$

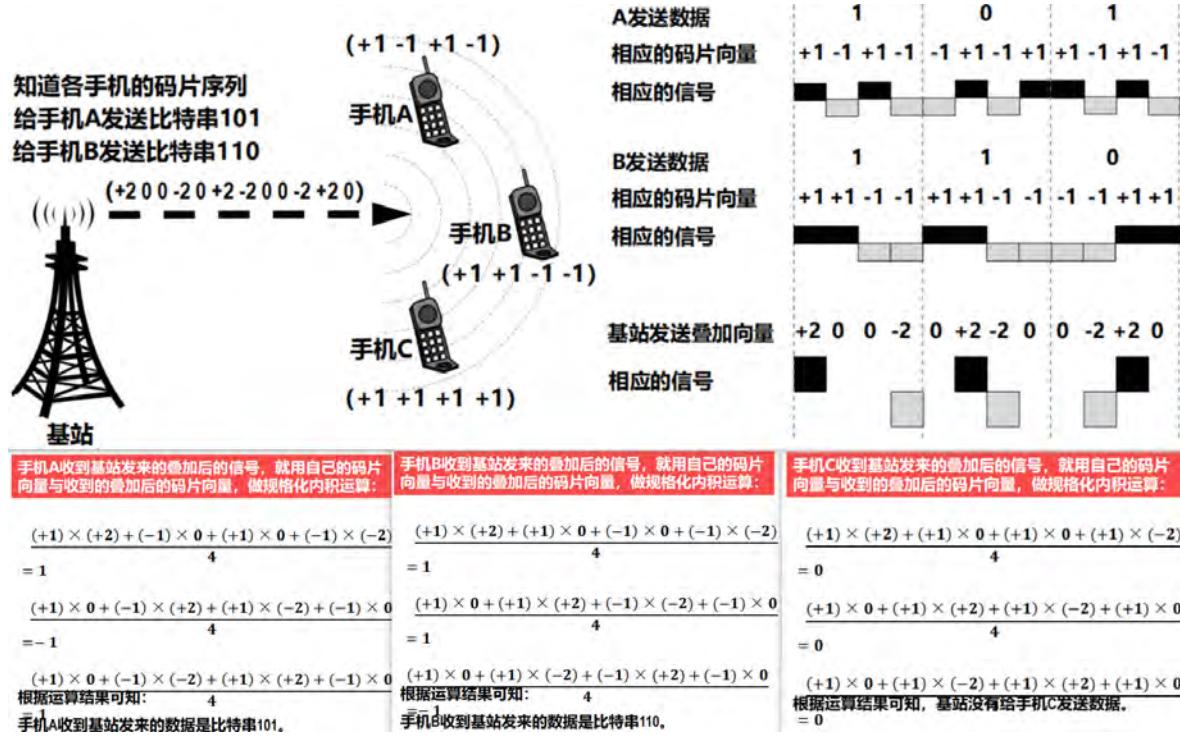
运算结果为1，表明收到的是比特1

$$(A + \bar{B}) \cdot B = A \cdot B + \bar{B} \cdot B = 0 + (-1) = -1$$

运算结果为-1，表明收到的是比特0

$$(A + \bar{B}) \cdot C = A \cdot C + \bar{B} \cdot C = 0 + 0 = 0$$

运算结果为0，表明没有收到信息



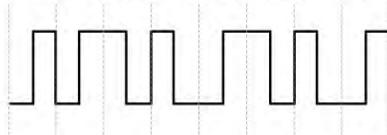
## 2.7. 题目

【2018年题34】下列选项中，不属于物理层接口规范定义范畴的是 (C)。

- A. 接口形状      物理层接口 机械特性
- B. 引脚功能      物理层接口 功能特性
- C. 物理地址      硬件地址或MAC地址
- D. 信号电平      物理层接口 电气特性

### 数据链路层

【2013年题34】若下图为10BaseT网卡接收到的信号波形，则该网卡收到的比特串是 (A)。



A. 0011 0110

B. 1010 1101

C. 0101 0010

D. 1100 0101

解析

1. 10BaseT以太网使用的是曼彻斯特编码。
2. 每个码元的中间时刻电平发生跳变：正跳变表示1还是0，负跳变表示0还是1，可以自行定义。

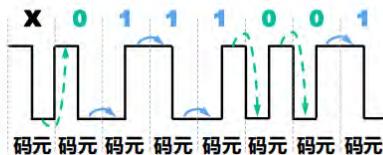


【2021年题34】若下图为一段差分曼彻斯特编码信号波形，则其编码的二进制位串是（A）。

- A. 1011 1001      B. 1101 0001      C. 0010 1110      D. 1011 0110

解析

1. 码元中间时刻的电平跳变仅表示时钟信号，而不表示数据。
2. 数据的表示在于每一个码元开始处是否有电平跳变：无跳变表示1，有跳变表示0。



奈氏准则

理想低通信道的最高码元传输速率为  
2W 码元/秒

香农公式

带宽受限且有高斯白噪声干扰的信道的  
极限信息传输速率  $C = W \log_2 \left( 1 + \frac{S}{N} \right) (b/s)$

【2009年题34】在无噪声情况下，若某通信链路的带宽为3kHz，采用4个相位，每个相位具有4种振幅的QAM调制技术，则该通信链路的最大数据传输速率是（B）。

- A. 12kbps      B. 24kbps      C. 48kbps      D. 96kbps

解析

1. 根据奈氏准则，该通信链路的最高码元传输速率 =  $2 \times 3k = 6k$  (码元/秒)
2. 采用4个相位，每个相位4种振幅的QAM调制技术，可以调制出  $4 \times 4 = 16$  个不同的基本波形（码元）。  
采用二进制对这16个不同的码元进行编码，需要使用4个比特 ( $\log_2 16 = 4$ )。  
即每个码元可以携带的信息量为4个比特。

综合1和2可知，该通信链路的最大数据传输速率 =  $6k$  (码元/秒)  $\times 4$  (比特/码元) =  $24k$  (比特/秒) = 24kbps

【2011年题34】若某通信链路的数据传输速率为2400bps，采用4相位调制，则该链路的波特率是（B）。

- A. 600波特      B. 1200波特      C. 4800波特      D. 9600波特

解析

1. 采用4相位调制，可以调制出4个相位不同的基本波形（码元）。  
采用二进制对这4个不同的码元进行编码，需要使用2个比特 ( $\log_2 4 = 2$ )。  
即每个码元可以携带的信息量为2个比特。
2. 数据的传输速率 = 波特率（码元传输速率） $\times$ 每个码元所携带的信息量  
 $2400$  (比特/秒) = 波特率  $\times 2$  (比特/码元)  
波特率 =  $1200$  (码元/秒) = 1200波特

【2014年题35】下列因素中，不会影响信道数据传输速率的是（D）。

解析

A. 信噪比

从香农公式可知  
信噪比和频率带宽都会影响  
信道数据传输速率

B. 频率带宽

C. 调制速度

从奈氏准则可知  
调制速度（码元传输速度）  
会影响信道数据传输速率

D. 信号传播速度

不影响  
信道数据传输速率  
自由空间:  $3.0 \times 10^8$  m/s  
铜线:  $2.3 \times 10^8$  m/s  
光纤:  $2.0 \times 10^8$  m/s

【2016年题34】若某链路的频率带宽为8kHz，信噪比为30dB，该链路实际数据传输速率约为理论最大数据传输速率的50%，则该链路的实际数据传输速率约是（C）。

- A. 8kbps      B. 20kbps      C. 40kbps      D. 80kbps

解析

$$\text{根据香农公式可计算出理论最大数据传输速率 } C = 8k \cdot \log_2 \left( 1 + \frac{S}{N} \right)$$

$$30(\text{dB}) = 10 \cdot \log_{10} \left( \frac{S}{N} \right) (\text{dB}) \quad \text{解得 } \frac{S}{N} = 1000 \quad \text{代入上式}$$

$$C = 8k \cdot \log_2 (1 + 1000) \approx 8k \cdot \log_2 (2^{10}) = 80 \text{ kbps}$$

该链路的实际数据传输速率约为  $C \times 50\% = 80 \text{ kbps} \times 50\% = 40 \text{ kbps}$

【2017年题34】若信道在无噪声情况下的极限数据传输速率不小于信噪比为30dB条件下的极限数据传输速率，则信号的状态数至少是（D）。

- A. 4      B. 8      C. 16      D. 32

解析

设信号状态数（可调制出的不同基本波形或码元数量）为X

则每个码元可携带的比特数量为 $\log_2 X$

信道在无噪声情况下的极限数据传输速率（用奈氏准则计算）=  $2W$ （码元/秒）=  $2W \log_2 X$ （比特/秒）

30dB信噪比条件下的极限数据传输速率（用香农公式计算）=  $W \log_2 (1+1000)$ （比特/秒）

根据题意列出不等式  $2W \log_2 X \geq W \log_2 (1+1000)$  解得  $X \geq 32$

【2014年题37】站点A、B、C通过CDMA共享链路，A、B、C的码片序列分别是(1,1,1,1)、(1,-1,1,-1)和(1,1,-1,-1)。若C从链路上收到的序列是(2,0,2,0,0,-2,0,-2,0,2,0,2)，则C收到A发送的数据是（B）。

- A. 000      B. 101      C. 110      D. 111

解析

由于题目所给各站的码片序列为4比特，因此将站点C收到的序列分成三部分，每部分也由4比特组成：

$$(2, 0, 2, 0, 0, -2, 0, 2, 0, 2) \longrightarrow (2, 0, 2, 0) \quad (0, -2, 0, -2) \quad (0, 2, 0, 2)$$

将站点A的码片序列（1,1,1,1）分别与上述三个部分进行规格化内积运算，根据结果可判断A发送的数据

$$(1, 1, 1, 1) \cdot (2, 0, 2, 0) = (1 \times 2 + 1 \times 0 + 1 \times 2 + 1 \times 0) \div 4 = 1 \quad \text{发送的是比特1}$$

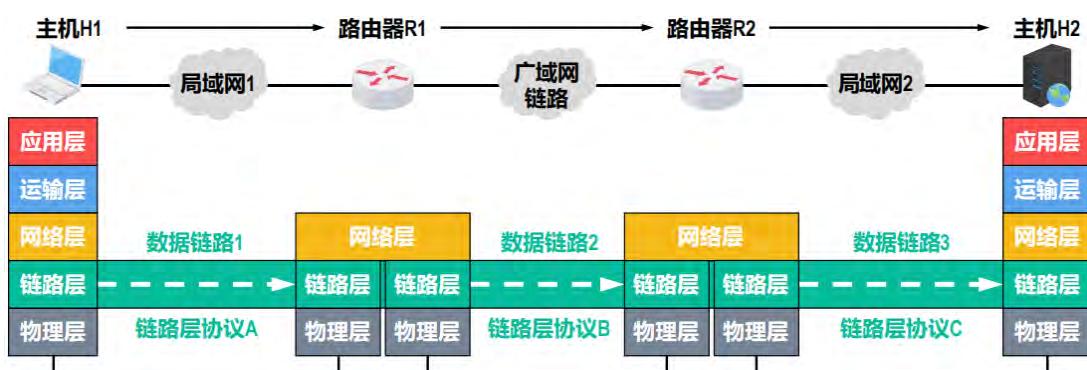
$$(1, 1, 1, 1) \cdot (0, -2, 0, -2) = (1 \times 0 + 1 \times (-2) + 1 \times 0 + 1 \times (-2)) \div 4 = -1 \quad \text{发送的是比特0}$$

$$(1, 1, 1, 1) \cdot (0, 2, 0, 2) = (1 \times 0 + 1 \times 2 + 1 \times 0 + 1 \times 2) \div 4 = 1 \quad \text{发送的是比特1}$$

## 3. data link layer

### 3.1. 数据链路层概述

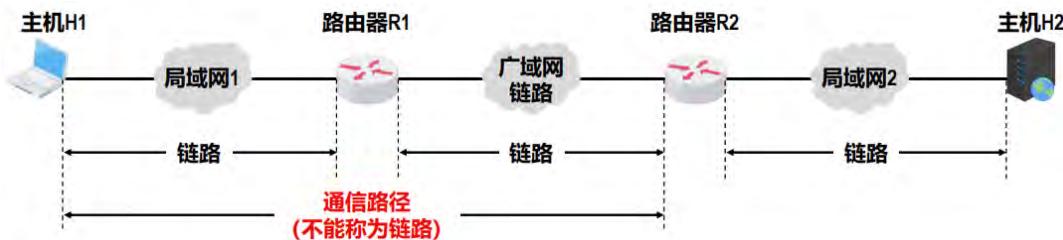
#### 3.1.1. 数据链路层在网络体系结构中所处的地位



#### 3.1.2. 链路、数据链路和帧

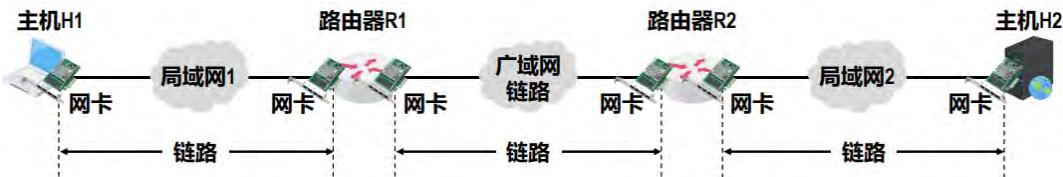
### 3.1.2.1. 链路 (Link)

- 指从一个节点到相邻节点的一段物理线路（有线或无线），而中间没有任何其他的交换节点。



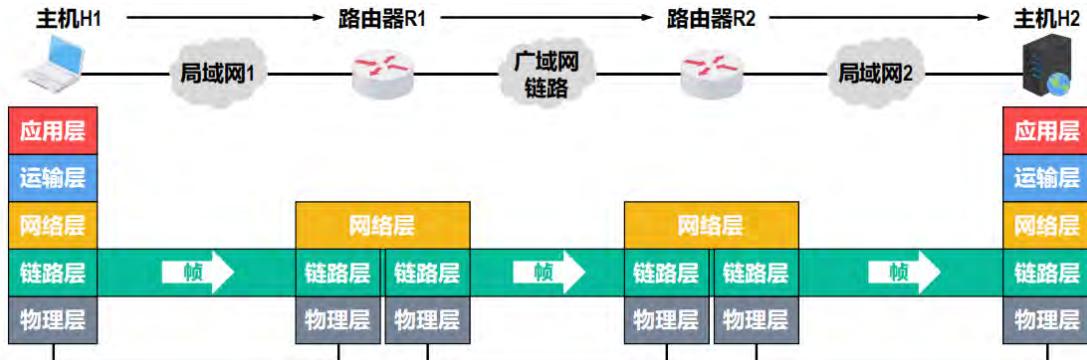
### 3.1.2.2. 数据链路 (Data Link)

- 数据链路是基于链路的。
- 当在一条链路上传送数据时，除需要链路本身，还需要一些必要的通信协议来控制这些数据的传输，把实现这些协议的硬件和软件加到链路上，就构成了数据链路。
- 计算机中的网络适配器（俗称网卡）和其相应的软件驱动程序就实现了这些协议。
- 一般的网络适配器都包含了物理层和数据链路层这两层的功能。



### 3.1.2.3. 帧 (Frame)

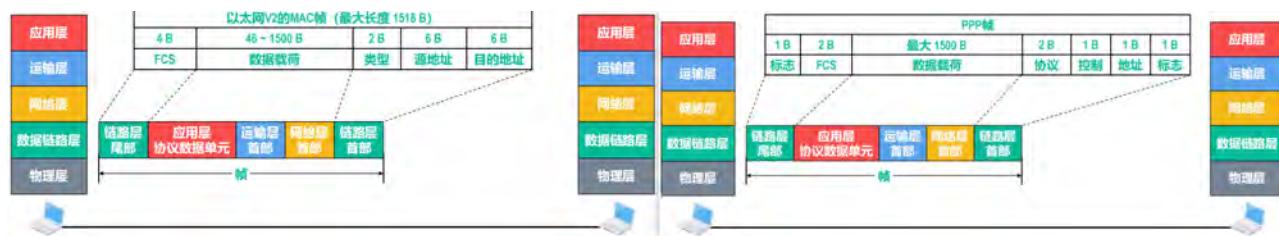
- 是数据链路层对等实体之间在水平方向进行逻辑通信的协议数据单元PDU。



## 3.2. 数据链路层的三个重要问题

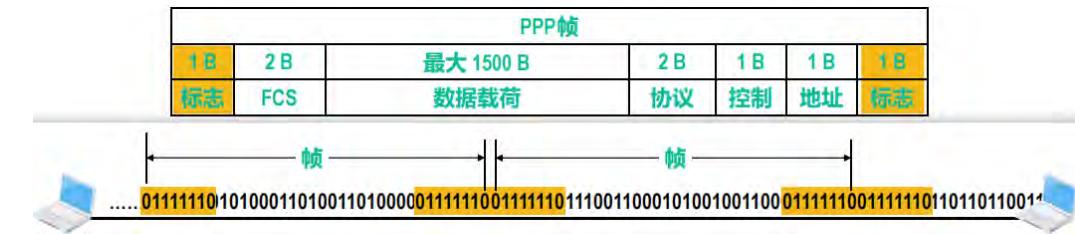
### 3.2.1. 封装成帧和透明传输

#### 3.2.1.1. 封装成帧

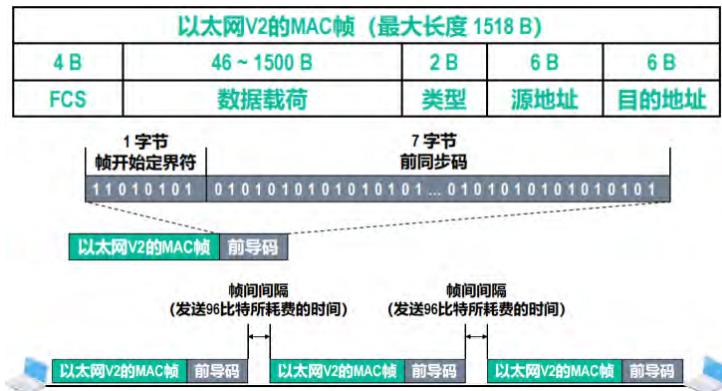


- 是指数据链路层给上层交付下来的协议数据单元PDU添加一个首部和一个尾部，使之成为帧。
  - 帧的首部和尾部中包含有一些重要的控制信息。
  - 帧首部和尾部的作用之一就是帧定界。

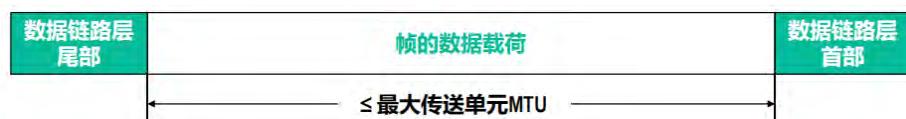
- 帧定界：接收端根据帧首部和帧尾部的标志字段，就可以从收到的比特流中识别出帧的开始和结束



- 并不是每一种数据链路层协议的帧都包含有帧定界标志。



- 为了提高数据链路层传输帧的效率，应当使帧的数据载荷的长度尽可能地大于首部和尾部的长度。
  - 考虑到对缓存空间的需求以及差错控制等诸多因素，每一种数据链路层协议都规定了帧的数据载荷的长度上限，即最大传送单元（Maximum Transfer Unit, MTU）。



- 例如，以太网的MTU为1500个字节。

### 3.2.1.2. 透明传输



- 透明传输：数据链路层对上层交付下来的协议数据单元PDU没有任何限制，就好像数据链路层不存在一样。

### 3.2.1.2.1. 字节填充

- 插入转义字符、帧定界符



### 3.2.1.2.2. 比特填充

- 如每遇到5个连续的比特1，就再其后面插入一个比特0（HDLC协议）

### 3.2.2. 差错检测

#### 3.2.3. 误码的相关概念

- 比特差错
  - 比特在传输过程中：比特1可能变成比特0；比特0可能变成比特1。
- 误码率（Bit Error Rate, BER）：传输错误的比特数量占所传输比特总数的比率
  - 提高链路的信噪比，可以降低误码率。
  - 在实际的通信链路上，不可能使误码率下降为零。



- 使用差错检测技术来检测数据在传输过程中是否产生了比特差错，是数据链路层所要解决的重要问题之一。
  - 帧在传输的过程中可能出现误码。
  - 接收方根据发送方添加在帧尾部中的检错码，可以检测出帧是否出现了误码。（采用与发送方相同的检错技术）
  - 帧检验序列（FCS）：帧尾部中用来存放检错码的字段

#### 3.2.4. 奇偶校验

- 奇校验是在待发送的数据后面添加1个校验位，使得添加该校验位后的整个数据中比特1的个数为奇数。



- 偶校验是在待发送的数据后面添加1个校验位，使得添加该校验位后的整个数据中比特1的个数为偶数。



- 奇数误码可检出，偶数误码会漏检。
- 在实际使用时，奇偶校验又可分为垂直奇偶校验、水平奇偶校验以及水平垂直奇偶校验。

#### 3.2.5. 循环冗余校验

- 数据链路层广泛使用漏检率极低的循环冗余校验（Cyclic Redundancy Check, CRC）检错技术。

### 3.2.5.1. CPC基本思想

- 收发双方约定好一个生成多项式G(X)。
- 发送方基于待发送的数据和生成多项式G(X)，计算出差错检测码（冗余码），将冗余码添加到待发送数据的后面一起传输。
- 接收方收到数据和冗余码后，通过生成多项式G(X)来计算收到的数据和冗余码是否产生了误码。

### 3.2.5.2. 发送方CRC操作



### 3.2.5.3. 接收方CRC操作



### 3.2.5.4. 生成多项式

- 举例

#### 【生成多项式举例】

$$\begin{aligned} G(X) &= X^4 + X^2 + X + 1 \\ &= \boxed{1} \cdot X^4 + \boxed{0} \cdot X^3 + \boxed{1} \cdot X^2 + \boxed{1} \cdot X^1 + \boxed{1} \cdot X^0 \end{aligned}$$

生成多项式各项系数构成的比特串: 10111 (计算冗余码时作为除数)

- 常用的生成多项式

#### 【常用的生成多项式】

算法要求生成多项式必须包含最低次项

$$CRC - 16 = x^{16} + x^{15} + x^2 + \boxed{1}$$

$$CRC - CCITT = x^{16} + x^{12} + x^5 + \boxed{1}$$

$$CRC - 32 = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + \boxed{1}$$

### 3.2.5.5. CRC举例

- 发送方

【CRC举例】待发送的数据为101001，生成多项式为 $G(X) = X^3 + X^2 + 1$ ，计算冗余码。



- 接收方

【CRC举例】接收到的信息为101101001，生成多项式为 $G(X) = X^3 + X^2 + 1$ ，判断传输是否误码？



余数不为0  
可认为传输过程产生了误码！

### 3.2.5.6. 注意

- 奇偶校验、循环冗余校验等差错检测技术，只能检测出传输过程中出现了差错，但并不能定位错误，因此无法纠正错误。
- 要想纠正传输中的差错，可以使用冗余信息更多的纠错码（例如海明码）进行前向纠错。但纠错码的开销比较大，在计算机网络中较少使用。
- 在计算机网络中，通常采用我们后续课程中将要介绍的检错重传方式来纠正传输中的差错，或者仅仅丢弃检测到差错的帧，这取决于数据链路层向其上层提供的是可靠传输服务还是不可靠传输服务。
- 循环冗余校验CRC具有很好的检错能力（漏检率极低），虽然计算比较复杂，但非常易于用硬件实现，因此被广泛应用于数据链路层。

### 3.2.6. 可靠传输

#### 3.2.6.1. 可靠传输的相关基本概念

##### 3.2.6.1.1. 不可靠传输

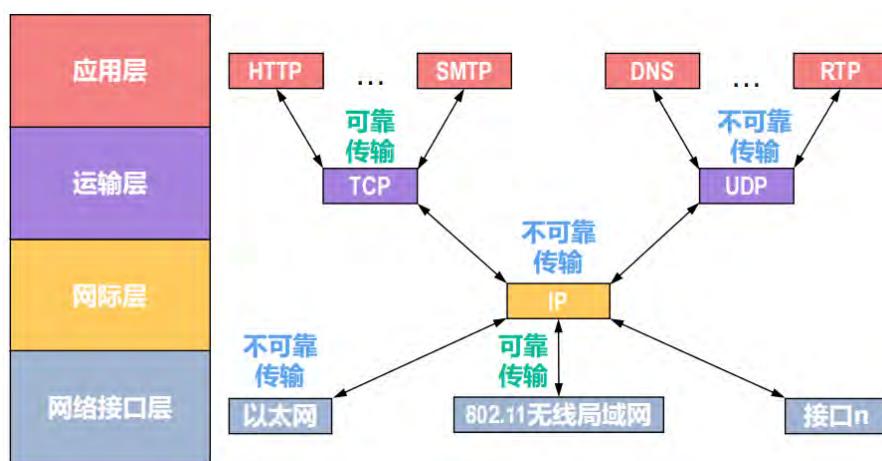
- 直接丢弃有误码的帧，其他什么也不做。

##### 3.2.6.1.2. 可靠传输

- 通过某种机制实现，实现发送方发送什么，接收方最终都能正确收到。
- 有线链路的误码率比较低，并不要求数据链路层向其上层提供可靠传输服务。
- 无线链路易受干扰，误码率比较高，因此要求数据链路层必须向其上层提供可靠传输服务。

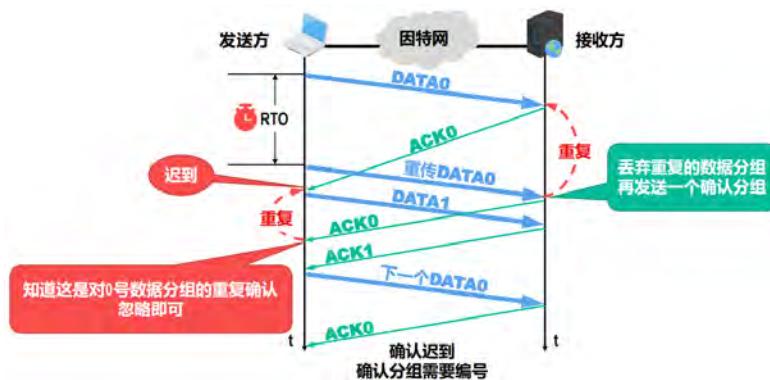
### 3.2.6.1.3. 传输差错

- 数据链路层及其下层
  - 误码（比特差错）
- 数据链路层的上层
  - 分组丢失
  - 分组失序
  - 分组重复
- 可靠传输服务并不局限于数据链路层，其他各层均可选择实现可靠传输。
- 可靠传输的实现比较复杂，开销比较大，是否使用可靠传输取决于应用需求。



### 3.2.6.2. 停止-等待协议 (Stop-and-Wait, SW)

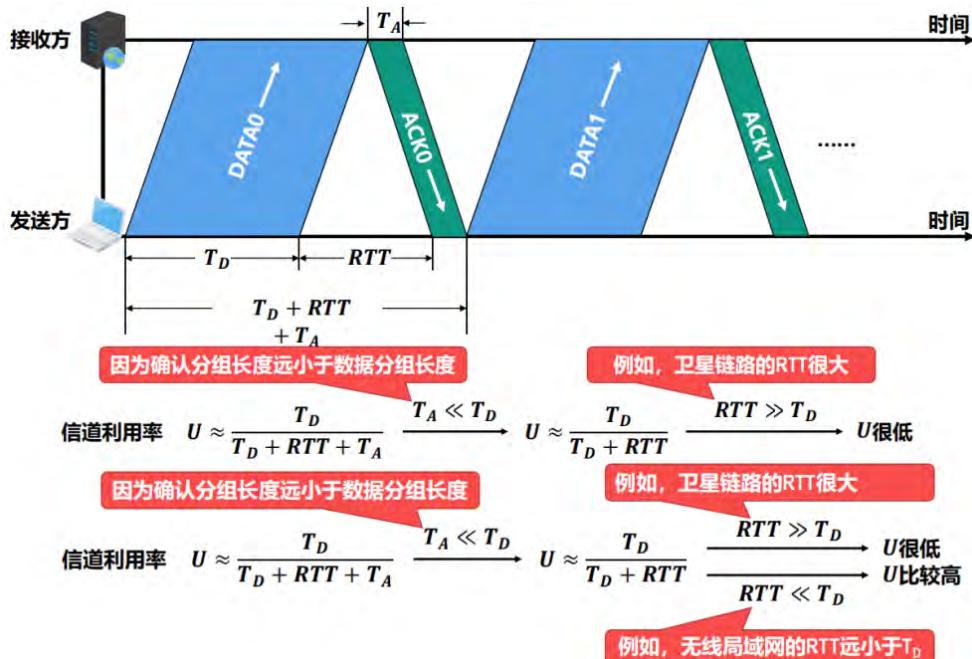
#### 3.2.6.2.1. 实现原理



- 确认、否认和重传
  - 使用超时重传机制后，就可以不使用否认机制了，这样可使协议实现起来更加简单。但是，如果点对点链路的误码率较高，使用否认机制可以使发送方在超时计时器超时前就尽快重传。
  - 超时重传时间 (Retransmission Time-Out, RTO)：一般将RTO设置为略大于收发双方的平均往返时间RTT
    - 在数据链路层，点对点的往返时间RTT比较固定，RTO就比较好设定。
    - 在运输层，由于端到端往返时间非常不确定，设置合适的超时重传时间RTO有时不容易。
- 分组编号 (数据分组+确认分组)

- 只需1个比特编序号即可，即序号0和序号1。
- 数据链路层一般不会出现确认分组迟到的情况，因此在数据链路层实现停止-等待协议可以不用给确认分组编号。
- 停止-等待协议属于自动请求重传（Automatic Repeat reQuest，ARQ）协议。即重传的请求是发送方自动进行的，而不是接收方请求发送方重传某个误码的数据分组。

### 3.2.6.2.2. 信道利用率



- 若出现超时重传，对于传送有用的数据信息来说，信道利用率还要降低。
- 在往返时间RTT相对较大的情况下，为了提高信道利用率，收发双方不适合采用停止-等待协议，而可以选择使用回退N帧（GBN）协议或选择重传（SR）协议。

### 3.2.6.3. 回退N帧协议（Go-back-N，GBN）

发送方	接收方
<ul style="list-style-type: none"> <li>■ 发送窗口<math>W_T</math>的取值范围是<math>1 &lt; W_T \leq (2^n - 1)</math>，其中，<math>n</math>是构成分组序号的比特数量。           <ul style="list-style-type: none"> <li>□ 如果<math>W_T = 1</math> 变成了停止-等待协议</li> <li>□ 如果<math>W_T &gt; (2^n - 1)</math> 接收方无法分辨新旧数据分组</li> </ul> </li> <li>■ 可在未收到接收方确认分组的情况下，将序号落入发送窗口内的多个数据分组全部发送出去。</li> <li>■ 只有收到对已发送数据分组的确认分组时，发送窗口才能向前滑动到相应位置。</li> <li>■ 收到多个重复确认时，可在重传计时器超时前尽早开始重传，由具体实现决定。</li> <li>■ 发送窗口内某个已发送的数据分组产生超时重传时，发送窗口内该数据分组的后续已发送的数据分组也必须全部重传，这就是回退N帧（Go-back-N，GBN）协议名称的由来。</li> </ul>	<ul style="list-style-type: none"> <li>■ 接收窗口<math>W_R = 1</math>的，因此只能按序接收数据分组。</li> <li>■ 只接收序号落入接收窗口内且无误码的数据分组，并且将接收窗口向前滑动一个位置，与此同时给发送方发送相应的确认分组。</li> <li>■ 为了减少开销，接收方不必每收到一个按序到达且无误码的数据分组就给发送方发送一个相应的确认分组。           <ul style="list-style-type: none"> <li>□ 可以在连续收到多个按序到达且无误码的数据分组后（数量由具体实现决定），才针对最后一个数据分组发送确认分组，这称为累积确认。</li> <li>□ 或者可以在自己有数据分组要发送时才对之前按序接收且无误码的数据分组进行捎带确认。</li> </ul> </li> <li>■ 接收方收到未按序到达的数据分组后，除丢弃外，还可对之前最后一个按序到达的数据分组进行重复确认，以便发送方尽快重传。</li> </ul>

回退N帧协议在流水线传输的基础上，利用发送窗口来限制发送方连续发送数据分组的数量，是一种连续ARQ协议。  
 在回退N帧协议的工作过程中，发送窗口和接收窗口不断向前滑动，因此这类协议又称为滑动窗口协议。  
 在信道质量较差（容易出现误码）的情况下，回退N帧协议的信道利用率并不比停止-等待协议的信道利用率高。

### 3.2.6.4. 选择重传协议（Selective Repeat, SR）

用 $n(n > 1)$ 个比特给分组编号，发送窗口 $W_T$ 与接收窗口 $W_R$ 的关系如下：

$W_R$ 超过 $W_T$ 没有意义 确保接收窗口向前滑动后，落入接收窗口内的新序号与之前的旧序号没有重叠，避免无法分辨新旧数据分组。	$\left. \begin{array}{l} 1 < W_R \leq W_T \\ W_T + W_R \leq 2^n \end{array} \right\} 1 < W_R \leq 2^{(n-1)}$	当 $W_R$ 取最大值 $2^{(n-1)}$ 时， $W_T$ 能取到的最大值也为 $2^{(n-1)}$ 。
---	--	--

发送方	接收方
<ul style="list-style-type: none"> <li>■ 可在未收到接收方确认分组的情况下，将序号落入发送窗口内的多个数据分组全部发送出去。</li> <li>■ 只有按序收到对已发送数据分组的确认分组时，发送窗口才能向前滑动到相应位置。</li> <li>■ 如果收到未按序到达的确认分组，应对其进行记录，以防止其相应数据分组的超时重发，但发送窗口不能向前滑动。</li> </ul>	<ul style="list-style-type: none"> <li>■ 可接收未按序到达但没有误码并且序号落入接收窗口内的数据分组。</li> <li>■ 为了使发送方仅重传出现差错的分组，接收方不再采用累积确认，而需要对每一个正确接收到的数据分组进行逐一确认。</li> <li>■ 只有在按序接收数据分组后，接收窗口才能向前滑动到相应位置。</li> </ul>

### 3.3. 点对点协议

#### 3.3.1. 点对点协议PPP概述

- 点对点协议 (Point-to-Point Protocol, PPP) 是目前使用最广泛的点对点数据链路层协议。
- 点对点协议PPP是因特网工程任务组 (Internet Engineering Task Force, IETF) 于1992年制定的。经过多次修订，目前PPP已成为因特网的正式标准[RFC1661, RFC1662]。
- 应用
  - 用户接入因特网，用户计算机与ISP通信。
  - 广泛应用于广域网路由器之间的专用线路。
- 从网络体系结构的角度看点对点协议PPP的组成



#### 3.3.2. PPP的帧格式



**标志 (Flag)** 字段：PPP帧的定界符，取值为0x7E。

**地址 (Address)** 字段：取值为0xFF，预留（目前没有什么作用）。

**控制 (Control)** 字段：取值为0x03，预留（目前没有什么作用）。

**协议 (Protocol)** 字段：其值用来指明帧的数据载荷应向上交付给哪个协议处理。

7E	FF	03	0021	IP数据报	FCS	7E
7E	FF	03	C021	LCP分组	FCS	7E
7E	FF	03	8021	NCP分组	FCS	7E

**帧检验序列 (Frame Check Sequence, FCS)** 字段：其值是使用循环冗余校验CRC计算出的检错码。

### 3.3.3. PPP帧的透明传输

- 面向字节的异步链路使用字节填充来实现透明传输[RFC1662]



#### 发送方的处理：

- (1) 将数据载荷中出现的每一个 $0x7E$ 减去 $0x20$ （相当于异或 $0x20$ ），然后在其前面插入转义字符 $0x7D$ 。  
(2) 若数据载荷中原来就含有 $0x7D$ ，则把每一个 $0x7D$ 减去 $0x20$ ，然后在其前面插入转义字符 $0x7D$ 。  
(3) 将数据载荷中出现的每一个ASCII码控制字符（即数值小于 $0x20$ 的字符），加上 $0x20$ （相当于异或 $0x20$ ，将其转换成非控制字符），然后在其前面插入转义字符 $0x7D$ 。

### **接收方的处理：**

进行与发送方相反的变换，就可以正确地恢复出未经过字节填充的原始数据载荷。

- 面向比特的同步链路使用零比特填充来实现透明传输



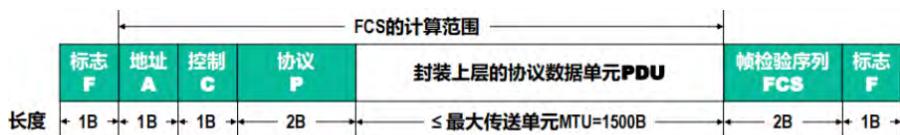
### 发送方的处理：

对帧的数据载荷进行扫描（一般由硬件完成），每出现5个连续的比特1，则在其后填充一个比特0。

### 接收方的处理：

对帧的数据载荷进行扫描，每出现5个连续的比特1时，就把其后的一个比特0删除

### 3.3.4 PPP帧的差错检测



**帧检验序列FCS字段：**其值是使用循环冗余校验CRC计算出的检错码。

CRC采用的生成多项式为  $CRC - CCITT = X^{16} + X^{12} + X^5 + 1$

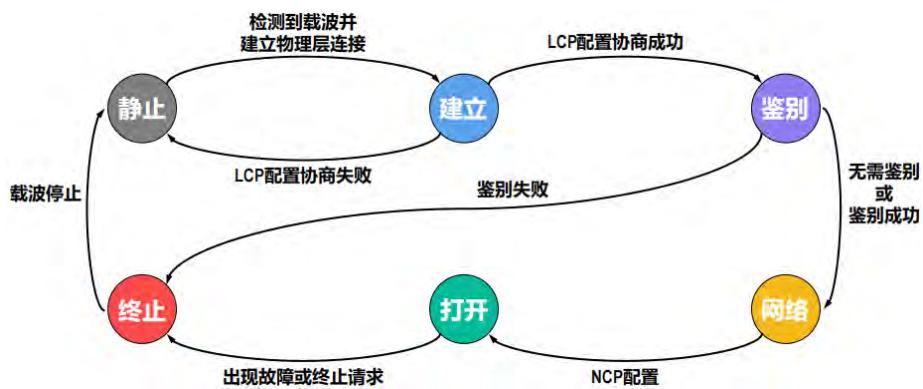
[RFC1662]文档的附录部分给出了FCS的计算方法的C语言实现（查表法）。

接收方每收到一个PPP帧，就进行CRC检验。若CRC检验正确，就收下这个帧；否则，就丢弃这个帧。

使用PPP的数据链路层，向上提供的是不可靠数据传输服务。

### 3.3.5 PPP的工作状态

- 以用户主机拨号接入因特网服务提供者ISP的拨号服务器的过程为例



## 3.4. 共享式以太网

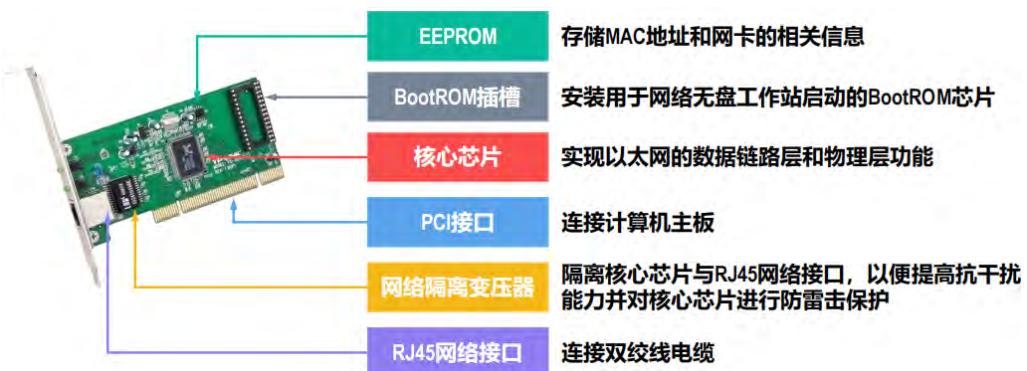
### 3.4.1. 概述

- 以太网（Ethernet）以曾经被假想的电磁波传播介质——以太（Ether）来命名。
- 以太网最初采用无源电缆（不包含电源线）作为共享总线来传输帧，属于基带总线局域网，传输速率为2.94Mb/s。
- 发展
  - 1975 以太网诞生
  - 1976 以太网里程碑论文
  - 1979 3Com公司成立
  - 1980 以太网标准V1
  - 1982 以太网标准V2
  - 1983 IEEE以太网标准
- 以太网目前已经从传统的共享式以太网发展到交换式以太网，传输速率已经从10Mb/s提高到100Mb/s、1Gb/s甚至10Gb/s。

### 3.4.2. 网络适配器和MAC地址

#### 3.4.2.1. 网络适配器

- 网络适配器（Adapter）（一般简称为“网卡”）：将计算机连接到以太网。

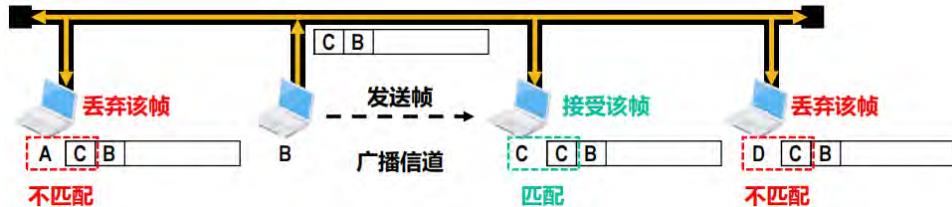


- 在计算机内部，网卡与CPU之间的通信，一般是通过计算机主板上的I/O总线以并行传输方式进行
- 网卡与外部以太网（局域网）之间的通信，一般是通过传输媒体（同轴电缆、双绞线电缆、光纤）以串行方式进行的。
- 网卡除要实现物理层和数据链路层功能，其另外一个重要功能就是要进行并行传输和串行传输的转换。由于网络的传输速率和计算机内部总线上的传输速率并不相同，因此在网卡的核心芯片中都会包含用于缓存数据的存储器。
- 在确保网卡硬件正确的情况下，为了使网卡正常工作，还必须要在计算机的操作系统中为网卡安装相应的设备驱动程序。驱动程序负责驱动网卡发送和接收帧。

### 3.4.2.2. MAC地址

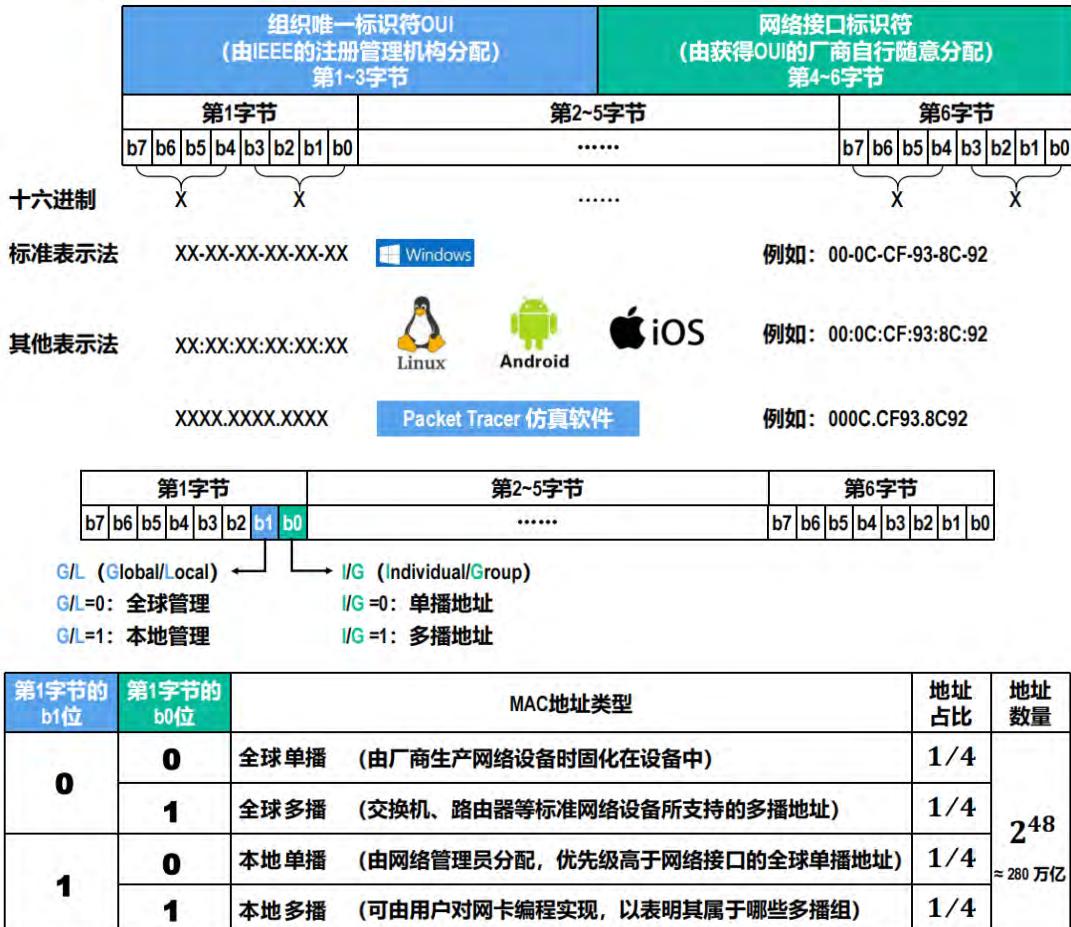
#### 3.4.2.2.1. 概念

- 当多个主机连接在同一个广播信道上，要想实现两个主机之间的通信，则每个主机都必须有一个唯一的标识，即一个数据链路层地址。
- MAC地址：用于媒体接入控制（Medium Access Control，MAC）；每个主机发送的帧的首部中，携带有发送主机（源主机）和接收主机（目的主机）的数据链路层地址。



- MAC地址一般被固化在网卡的电可擦可编程只读存储器EEPROM中，因此MAC地址也被称为硬件地址。
- MAC地址有时也被称为物理地址。
- 普通用户计算机中往往会包含两块网卡
  - 用于接入有线局域网的以太网卡
  - 用于接入无线局域网的Wi-Fi网卡
- 每块网卡都有一个全球唯一的MAC地址。
- MAC地址是对网络上各接口的唯一标识，而不是对网络上各设备的唯一标识。

#### 3.4.2.2. IEEE 802局域网的MAC地址格式



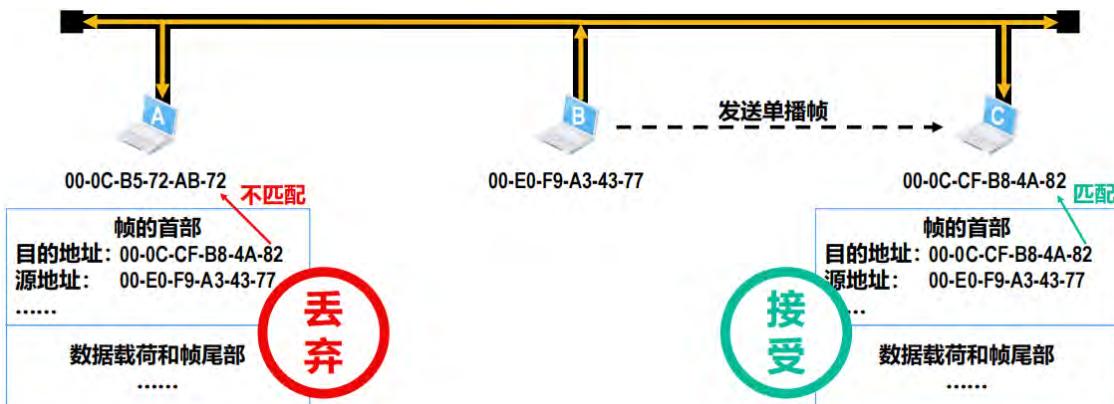
### 3.4.2.2.3. IEEE 802局域网的MAC地址发送顺序

组织唯一标识符OUI (由IEEE的注册管理机构分配) 第1~3字节						网络接口标识符 (由获得OUI的厂商自行随意分配) 第4~6字节											
第1字节		第2~5字节					第6字节										
b7	b6	b5	b4	b3	b2	b1	b0	.....		b7	b6	b5	b4	b3	b2	b1	b0

字节发送顺序：第1字节 → 第6字节

字节内的比特发送顺序：b0 → b7

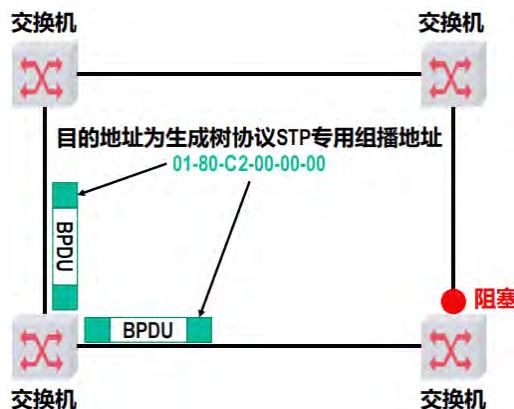
### 3.4.2.2.4. 单播MAC地址举例



### 3.4.2.2.5. 广播MAC地址举例



### 3.4.2.2.6. 多播MAC地址举例



- 网卡从网络上每收到一个帧，就检查帧首部中的目的MAC地址，按以下情况处理：
  - 如果目的MAC地址是广播地址（FF-FF-FF-FF-FF-FF），则接受该帧。

- (2) 如果目的MAC地址与网卡上固化的全球单播MAC地址相同，则接受该帧。
- (3) 如果目的MAC地址是网卡支持的多播地址，则接受该帧。
- (4) 除上述(1)、(2)和(3)情况外，丢弃该帧。
- 网卡还可被设置为一种特殊的工作方式：混杂模式（Promiscuous Mode）。工作在混杂模式的网卡，只要收到共享媒体上传来的帧就会收下，而不管帧的目的MAC地址是什么。
  - 对于网络维护和管理人员，这种方式可以监视和分析局域网上的流量，以便找出提高网络性能的具体措施。
  - 嗅探器（Sniffer）就是一种工作在混杂模式的网卡，再配合相应的工具软件（Wireshark），就可以作为一种非常有用的网络工具来学习和分析网络。
  - 混杂模式就像一把“双刃剑”，黑客常利用这种方式非法获取网络用户的口令。

全球单播MAC地址就如同身份证上的身份证号码，具有唯一性，它往往与用户个人信息绑定在一起。因此，用户应尽量确保自己拥有的全球单播MAC地址不被泄露。

为了避免用户设备连接Wi-Fi热点时MAC地址泄露的安全问题，目前大多数移动设备都已经采用了随机MAC地址技术。

### 3.4.3. CSMA/CD协议

#### 3.4.3.1. CSMA/CD协议的基本原理

- 为了解决各站点争用总线的问题，共享总线以太网使用了一种专用协议CSMA/CD，它是载波监听多址接入/碰撞检测（Carrier Sense Multiple Access Collision Detection）的英文缩写词。

**共享总线以太网的一个重要问题：如何协调总线上的各站点争用总线。**



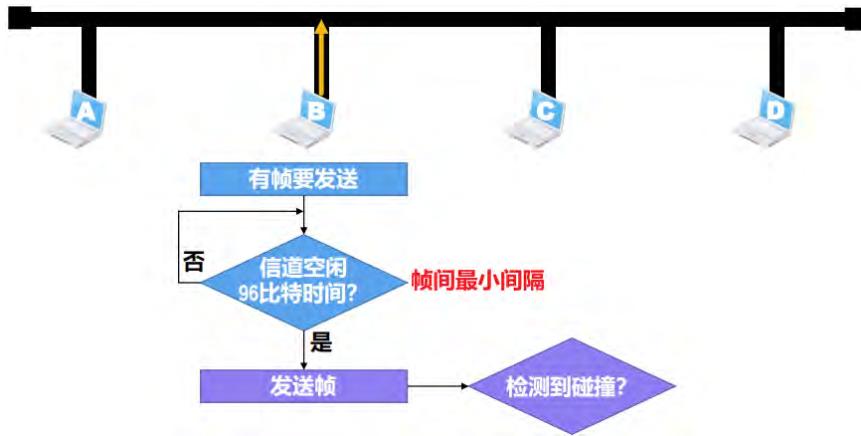
- 载波监听检测到总线空闲，但总线并不一定空闲。
- 使用CSMA/CD协议的共享总线以太网上的各站点，只是尽量避免碰撞并在出现碰撞时做出退避后重发的处理，但不能完全避免碰撞。
- 在使用CSMA/CD协议时，由于正在发送帧的站点必须“边发送帧边检测碰撞”，因此站点不可能同时进行发送和接收，也就是不可能进行全双工通信，而只能进行半双工通信（双向交替通信）。

##### 3.4.3.1.1. 多址接入 MA

- 多个站点连接在一条总线上，竞争使用总线。

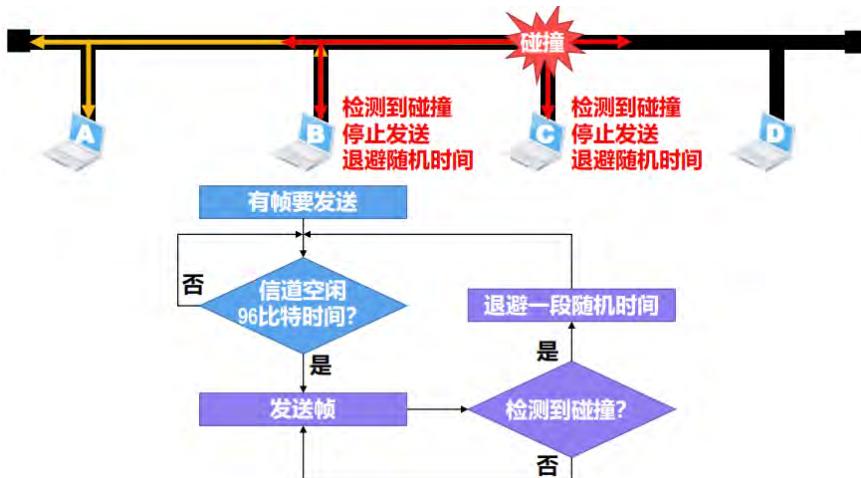
##### 3.4.3.1.2. 载波监听 CS

- 每个站点在发送帧之前，先要检测一下总线上是否有其他站点在发送帧（“先听后说”）



### 3.4.3.1.3. 碰撞检测CD

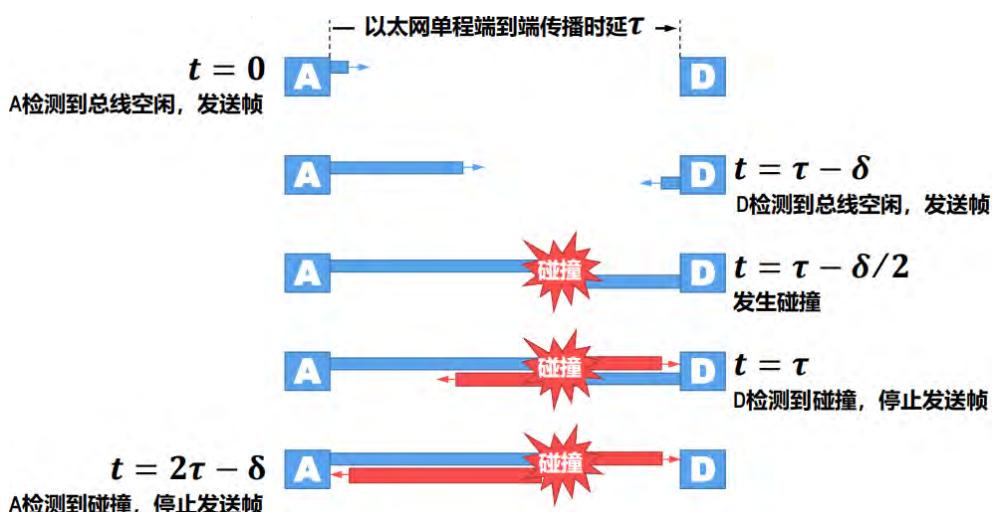
- 每个正在发送帧的站点边发送边检测碰撞（“边说边听”）
  - 一旦冲突，立即停说，等待时机，重新再说



- 强化碰撞
  - 发送帧的站点一旦检测到碰撞，除了立即停止发送帧外，还要再继续发送32比特或48比特的人为干扰信号（Jamming Signal），以便有足够的碰撞信号使所有站点都能检测出碰撞。

### 3.4.3.2. 共享式以太网的争用期

- 共享总线以太网的端到端往返时间 $2\tau$ 被称为争用期（Contention Period）或碰撞窗口（Collision Window）
- 站点从发送帧开始，最多经过时长 $2\tau$ （即 $\delta \rightarrow 0$ ）就可检测出所发送的帧是否遭遇了碰撞。



- 从争用期的概念可以看出，共享总线以太网上的每一个站点从发送帧开始，到之后的一小段时间内，都有可能遭遇碰撞，而这一小段时间的长短是不确定的，它取决于另一个发送帧的站点与本站点的距离，但不会超过总线的端到端往返传播时延，即一个争用期 $2\tau$ 。
  - 总线的长度越长（单程端到端传播时延越大），网络中站点数量越多，发生碰撞的概率就越大。
  - 共享以太网的总线长度不能太长，接入的站点数量也不能太多。

### 3.4.3.3. 共享式以太网的最小帧长和最大帧长

- 最小帧长 = 数据传输速率 \* 争用期  $2\tau$ 
  - 为了确保共享总线以太网上的每一个站点在发送完一个完整的帧之前，能够检测出是否产生了碰撞，帧的发送时延就不能少于共享总线以太网端到端的往返时间，即一个争用期 $2\tau$ 。
  - 对于10mb/s的共享总线以太网，其争用期 $2\tau$ 的值规定51.2μs，因此其最小帧长为512b，即64B。

**■ 10Mb/s共享总线以太网（传统以太网）规定：争用期 $2\tau$  的值为512比特的发送时间，即51.2μs。**

$$\text{争用期 } 2\tau = \frac{512 \text{ b}}{10 \text{ Mb/s}} = \frac{512 \text{ b}}{10 \times 10^6 \text{ b/s}} = 51.2 \mu\text{s}$$

$$\text{单程端到端传播时延 } \tau = \frac{51.2 \mu\text{s}}{2} = 25.6 \mu\text{s}$$

除考虑了信号传播时延外，还考虑到网络中可能存在转发器所带来的时延以及产生碰撞时继续发送32比特或48比特人为干扰信号所持续的时间等。

假设信号的传播速率为 $2 \times 10^8 \text{ m/s}$

则总线长度为 $2 \times 10^8 \text{ m/s} \times 25.6 \mu\text{s} = 5120 \text{ m}$

共享总线以太网规定：  
总线长度不能超过2500m。

$$10 \text{ Mb/s} \times 51.2 \mu\text{s} = 512 \text{ b} = 64 \text{ B}$$

- 当某个站点在发送帧时，如果帧的前64B没有遭遇碰撞，那么帧的后续部分也就不会遭遇碰撞。也就是说，如果遭遇碰撞，就一定是在帧的前64B之内。
- 由于发送帧的站点边发送帧边检测碰撞，一旦检测到碰撞就立即中止帧的发送，此时已发送的数据量一定小于64B。因此，接收站点收到长度小于64B的帧，就可判定这是一个遭遇了碰撞而异常中止的无效帧，将其丢弃即可。

### • 最大帧长

- 一般来说，帧的数据载荷的长度应远大于帧首部和尾部的总长度，这样可以提高帧的传输效率。

首部	帧的数据载荷	尾部
----	--------	----

- 然而，如果不限制数据载荷的长度上限，就可能使得帧的长度太长，这会带来一些问题。



以太网V2的MAC帧 (最大长度1518B)				
目的地址	源地址	类型	数据载荷	FCS
6B	6B	2B	46B ~ 1500B	4B

满足最小帧长为64B的要求  
( 6B + 6B + 2B + 46B + 4B = 64B )

### 3.4.3.4. 共享式以太网的退避算法

- 在使用CSMA/CD协议的共享总线以太网中，正在发送帧的站点一边发送帧一边检测碰撞，当检测到碰撞时就立即停止发送，**退避一段随机时间后再重新发送**。

#### 3.4.3.4.1. 截断二进制指数退避

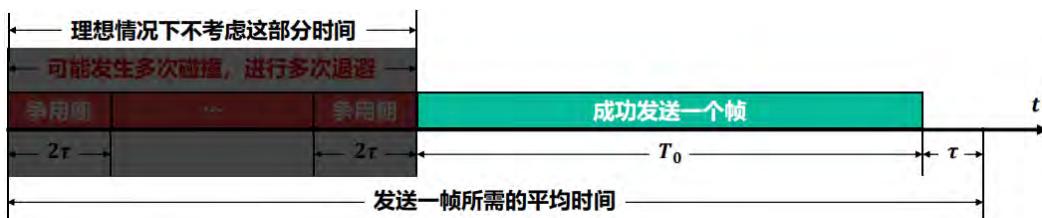
- 共享总线以太网中的各站点采用截断二进制指数退避 (Truncated Binary Exponential Backoff) 算法来选择退避的随机时间。



重传次数	$k$	离散的整数集合 $\{0, 1, \dots, (2^k - 1)\}$	可能的退避时间
1	1	{0, 1}	$0 \times 2\tau, 1 \times 2\tau$
2	2	{0, 1, 2, 3}	$0 \times 2\tau, 1 \times 2\tau, 2 \times 2\tau, 3 \times 2\tau$
12	10	{0, 1, 2, 3, 4, 5, ..., 1023}	$0 \times 2\tau, 1 \times 2\tau, 2 \times 2\tau, \dots, 1023 \times 2\tau$

- 如果连续多次发送碰撞，就表明可能有较多的站点参与竞争信道。但使用上述退避算法可使重传需要推迟的平均时间随重传次数而增大（即动态退避），因而减小产生碰撞的概率。
- 当重传达16次仍不能成功时，就表明同时打算发送帧的站点太多，以至于连续产生碰撞，此时应放弃重传并向高层报告。

### 3.4.3.5. 共享式以太网的信道利用率



■ 考虑以下这种理想情况：

- 总线一旦空闲就有某个站点立即发送帧
- 各站点发送帧都不会产生碰撞
- 发送一帧占用总线的时间为  $T_0 + \tau$ ，而帧本身的发送时间是  $T_0$

$$\text{极限信道利用率 } S_{max} = \frac{T_0}{T_0 + \tau} = \frac{1}{1 + \frac{\tau}{T_0}} = \frac{1}{1 + a}$$

$$\text{极限信道利用率 } S_{max} = \frac{T_0}{T_0 + \tau} = \frac{1}{1 + \frac{\tau}{T_0}} = \frac{1}{1 + a}$$

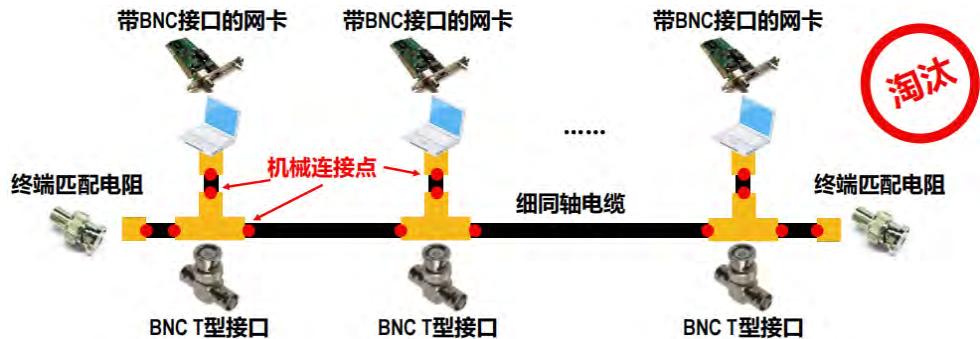
参数  $a$  的值应尽量小，以提高信道利用率

$$a = \frac{\tau}{T_0}$$

共享总线以太网端到端的距离不应太长  
 帧的长度应尽量大

### 3.4.4. 使用集线器的共享式以太网

### 3.4.4.1. 粗 (->细) 同轴电缆的共享总线以太网



- 若总线上的某个机械连接点接触不良或断开，则整个网络通信就不稳定或彻底断网。

### 3.4.4.2. 集线器

- 集线器 (Hub) :

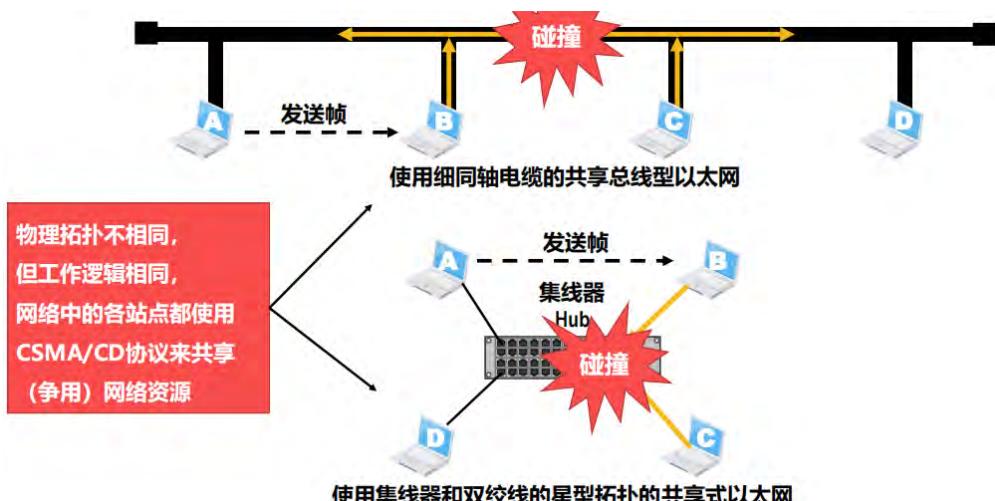
- 使用大规模集成电路来替代总线、并且可靠性非常高的设备。
- 站点连接到集线器的传输媒体也转而使用更便宜、更灵活的双绞线电缆。



- 特点

- 物理拓扑是星型的，但在逻辑上仍然是一个总线网。总线上的各站点共享总线资源，使用的还是CSMA/CD协议。
- 只工作在物理层，仅简单地转发比特，并不进行碰撞检测。碰撞检测的任务由各站点中的网卡负责。
  - 例如，若网络中某个站点的网卡出现了故障而不停地发送帧，集线器可以检测到这个问题，在内部断开与出故障网卡的连线，使整个以太网能正常工作。

### 3.4.4.3. 对比



#### 3.4.4.4. BASE-T星型以太网

- IEEE于1990年制定了10BASE-T星型以太网的标准802.3i，这种以太网是局域网发展史上的一座非常重要的里程碑，它为以太网在局域网中的统治地位奠定了牢固的基础。
- 10BASE-T以太网的通信距离较短，每个站点到集线器的距离不能超过100m。
- IEEE 802.3以太网还可使用光纤作为传输媒体，相应的标准为10BASE-F，“F”表示光纤。光纤主要用作集线器之间的远程连接。



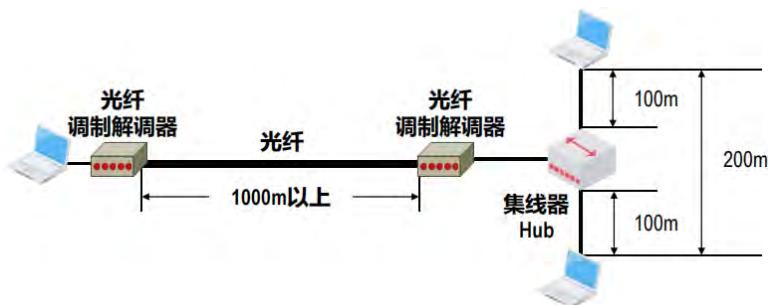
#### 3.4.5. 在物理层扩展以太网

##### 3.4.5.1. 扩展站点与集线器之间的距离

- 中两站点之间的距离太远会使传输的信号就会衰减到使CSMA/CD协议无法正常工作。
- 在早期广泛使用粗同轴电缆或细同轴电缆共享总线以太网时，为了提高网络的地理覆盖范围，常用的是工作在物理层的转发器。
- IEEE 802.3标准规定，两个网段可用一个转发器连接起来，任意两个站点之间最多可以经过三个网段。



- 在10BASE-T星型以太网中，可使用光纤和一对光纤调制解调器来扩展站点与集线器之间的距离。
  - 这种扩展方法比较简单，所需付出的代价是：为站点和集线器各增加一个用于电信号和光信号转换的光纤调制解调器，以及它们之间的一对通信光纤。
- 信号在光纤中的衰减和失真很小，因此使用这种方法可以很简单地将站点与集线器之间的距离扩展到1000以上。



- 在物理层扩展的共享式以太网仍然是一个碰撞域，不能连接太多的站点，否则可能会出现大量的碰撞，导致平均吞吐量太低。

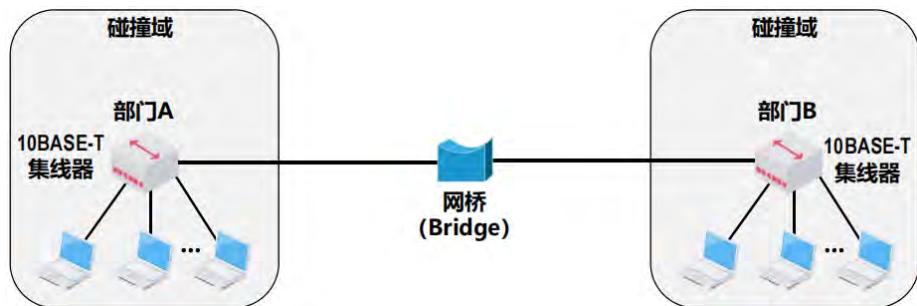


### 3.4.5.2. 扩展共享式以太网的覆盖范围和站点数量

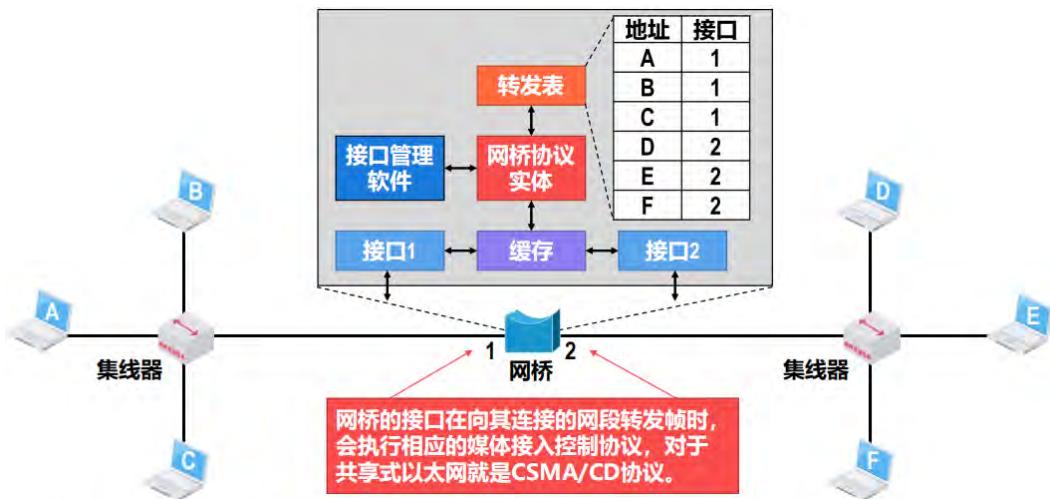
### 3.4.6. 在数据链路层扩展以太网

#### 3.4.6.1. 使用网桥在数据链路层扩展以太网

- 网桥（bridge）工作在数据链路层（包含其下的物理层），因此网桥具备属于数据链路层范畴的相关能力。
  - 网桥可以识别帧的结构。
  - 网桥可以根据帧首部中的目的MAC地址和网桥自身的帧转发表来转发或丢弃所收到的帧。



#### 3.4.6.2. 网桥的主要结构和基本工作原理



#### 3.4.6.3. 透明网桥的自学习和转发帧的流程

- 透明网桥（Transparent Bridge）通过自学习算法建立转发表。
  - “透明”，是指以太网中的各站点并不知道自己所发送的帧将会经过哪些网桥的转发，最终到达目的站点。也就是说，以太网中的各网桥对于各站点而言是看不见的。
  - 透明网桥的标准是IEEE 802.1D，它通过一种自学习算法基于以太网中各站点间的相互通信逐步建立起自己的转发表。

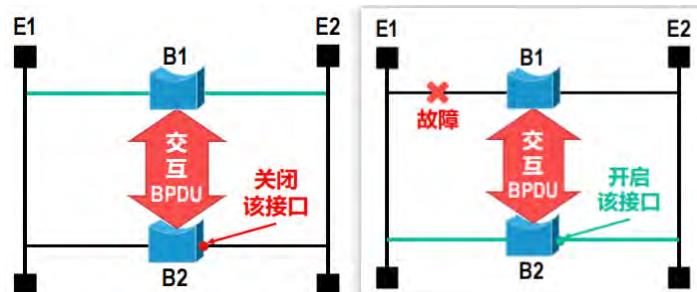
- ① 网桥收到帧后进行登记（即自学习），登记的内容为帧的源MAC地址和进入网桥的接口号。
- ② 网桥根据帧的目的MAC地址和网桥的转发表对帧进行转发，包含以下三种情况：
  - **明确转发**：网桥知道应当从哪个接口转发帧。
  - **盲目转发**：网桥不知道应当从哪个接口转发帧，只能将其通过除进入网桥的接口外的其他所有接口转发。
  - **丢弃**：网桥知道不应该转发该帧，将其丢弃。

请同学们注意：

- (1) 如果网桥收到有误码的帧则直接丢弃。
- (2) 如果网桥收到一个无误码的广播帧，则不用进行查表，而是直接从除接收该广播帧的接口的其他接口转发该广播帧。
- (3) 转发表中的每条记录都有其有效时间，到期自动删除！这是因为各站点的MAC地址与网桥接口的对应关系并不是永久性的，例如某个站点更换了网卡，其MAC地址就会改变。

#### 3.4.6.4. 透明网桥的生成树协议STP

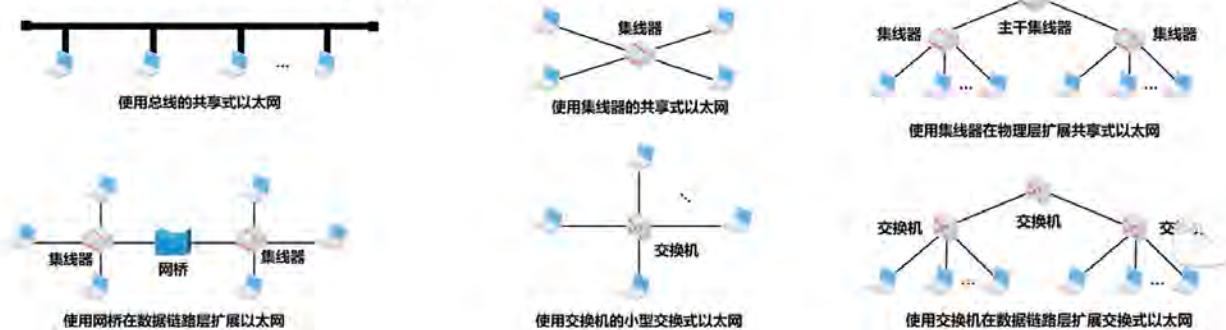
- 为了提高以太网的可靠性，有时需要在两个以太网之间使用多个透明网桥来提供冗余链路。
  - 在增加冗余链路提高以太网可靠性的同时，却给网络引入了环路。
  - 网络中的广播帧将在环路中永久兜圈，造成广播帧充斥整个网络，网络资源被白白浪费，而网络中的主机之间无法正常通信！
- 生成树协议（Spanning Tree Protocol, STP）
  - 避免广播帧在环路中永久兜圈；可以在增加冗余链路提高网络可靠性的同时，又避免环路带来的问题。
  - 不管网桥之间连接成了怎样复杂的带环拓扑，网桥之间通过交互网桥协议单元（Bridge Protocol DataUnit, BPDU），找出原网络拓扑的一个连通子集（即生成树），在这个子集里整个连通的网络中不存在环路。
  - 当首次连接网桥或网络拓扑发生变化时（人为改变或出现故障），网桥都会重新构造生成树，以确保网络的连通。



### 3.5. 交换式以太网

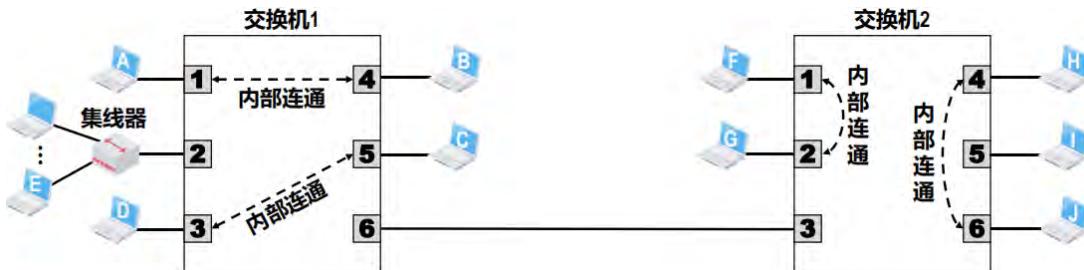
#### 3.5.1. 交换式以太网

- 网桥的接口数量很少，通常只有2~4个，一般只用来连接不同的网段。
- 1990年面世的交换式集线器（Switching Hub），实质上是具有多个接口的网桥，常称为以太网交换机（Switch）或二层交换机。
  - “二层”是指以太网交换机工作在数据链路层（包括物理层）。
  - 与网桥相同，交换机内部的转发表也是通过自学习算法，基于网络中各主机间的通信，自动地逐步建立起来的。
  - 另外，交换机也使用生成树协议STP，来产生能够连通全网但不产生环路的通信路径。
- 仅使用交换机（而不使用集线器）的以太网就是交换式以太网。

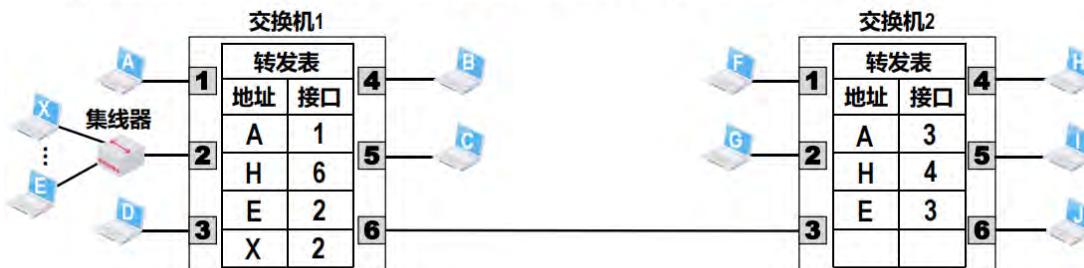


### 3.5.2. 以太网交换机

- 以太网交换机
  - 本质上就是一个多接口的网桥
  - 交换机自学习和转发帧的流程与网桥是相同的
  - 交换机也使用生成树协议STP，来产生能够连通全网但不产生环路的通信路径。
- 交换机的每个接口可以连接计算机，也可以连接集线器或另一个交换机。
  - 当交换机的接口与计算机或交换机连接时，可以工作在全双工方式，并能在自身内部同时连通多对接口，使每一对相互通信的计算机都能像独占传输媒体那样，无碰撞地传输数据，这样就不需要使用CSMA/CD协议了。
  - 当交换机的接口连接的是集线器时，该接口就只能使用CSMA/CD协议并只能工作在半双工方式。
  - 现在的交换机和计算机中的网卡都能自动识别上述两种情况，并自动切换到相应的工作方式。
  - 交换机一般都具有多种速率的接口，例如10Mb/s、100Mb/s、1Gb/s甚至10Gb/s的接口，大部分接口支持多速率自适应。



【练习题】交换机自学习和转发帧的流程（参考网桥自学习和转发帧的流程）



#### 主机间的通信

- A→B
- H→A
- E→X
- X→E

#### 交换机1的操作

- 登记 转发（盲目）
- 登记 转发（明确）
- 登记 转发（盲目）
- 登记 丢弃

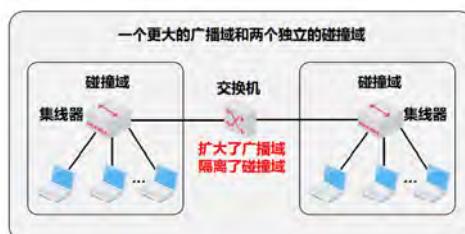
#### 交换机2的操作

- 登记 转发（盲目）
- 登记 转发（明确）
- 登记 转发（盲目）
- 收不到

### 3.5.3. 共享式以太网与交换式以太网的对比



- 集线器
  - 扩大了广播域
  - 扩大了碰撞域

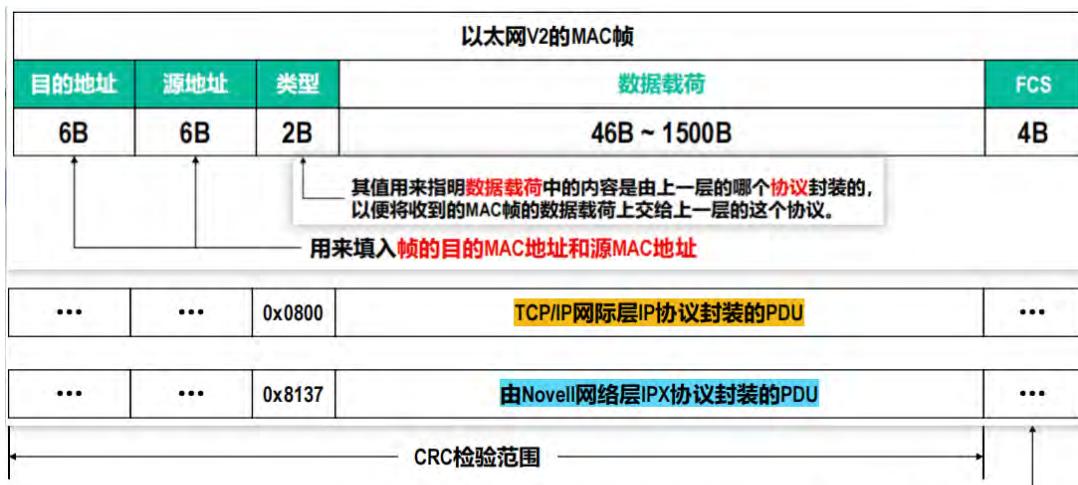


- 交换机
  - 扩大了广播域
  - 隔离了碰撞域
- 交换式以太网的网络性能远高于共享式以太网，集线器早已被交换机取代。

## 3.6. 以太网的MAC帧格式



### 3.6.1. 以太网V2的MAC帧



## 数据链路层

## 以太网V2的MAC帧

## 物理层

## 以太网V2的MAC帧

## 前导码



- 接收方可能收到的无效MAC帧包括以下几种：
  - MAC帧的长度不是整数个字节
  - 通过MAC帧的FCS字段的值检测出帧有误码
  - MAC帧的长度不在64~1518字节之间
- 接收方收到无效的MAC帧时，就简单将其丢弃，以太网的数据链路层没有重传机制。

### 3.7. 虚拟局域网

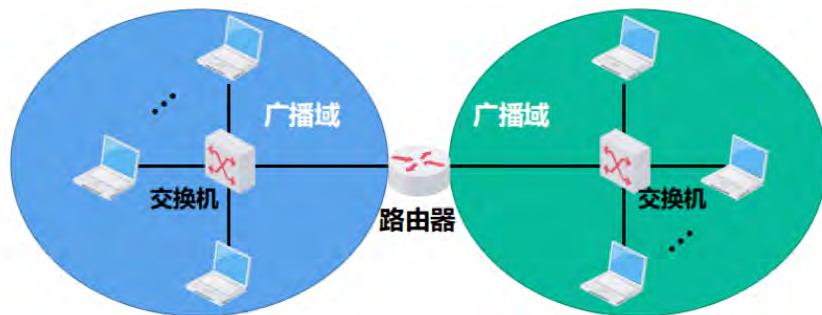
#### 3.7.1. 虚拟局域网VLAN的诞生背景

- 将多个站点通过一个或多个以太网交换机连接起来就构建出了交换式以太网。
- 交换式以太网中的所有站点都属于同一个广播域。
- 随着交换式以太网规模的扩大，广播域也相应扩大。

#### 3.7.1.1. 巨大的广播域会带来一系列问题

- 广播风暴（广播风暴会浪费网络资源和各主机的CPU资源）
- 难以管理和维护，带来潜在的安全问题。

#### 3.7.1.2. 分割广播域的方法



- 使用路由器可以隔离广播域（成本较高）
- 虚拟局域网技术应运而生

#### 3.7.2. 虚拟局域网VLAN概述

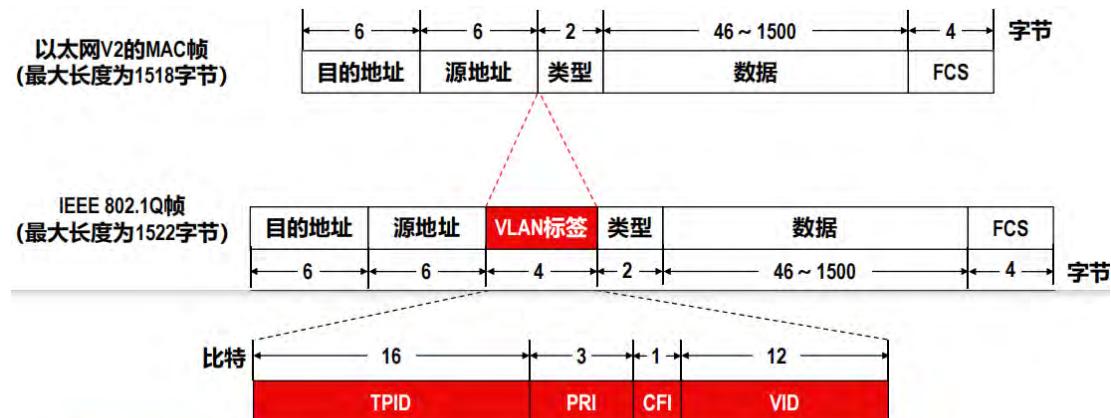
- 虚拟局域网（Virtual Local Area Network, VLAN）是一种将局域网内的站点划分成与物理位置无关的逻辑组的技术，一个逻辑组就是一个VLAN，VLAN中的各站点具有某些共同的应用需求。
- 属于同一VLAN的站点之间可以直接进行通信，而不同VLAN中的站点之间不能直接通信。

- 网络管理员可对局域网中的各交换机进行配置来建立多个逻辑上独立的VLAN
  - 连接在同一交换机上的多个站点可以属于不同的VLAN，而属于同一VLAN的多个站点可以连接在不同的交换机上。
- 虚拟局域网VLAN并不是一种新型网络，它只是局域网能够提供给用户的一种服务。

### 3.7.3. 虚拟局域网VLAN的实现机制 —— IEEE 802.1Q帧

#### 3.7.3.1. IEEE 802.1Q帧

- 虚拟局域网VLAN有多种实现技术。
- 最常见的就是基于以太网交换机的接口来实现VLAN。
- 这就需要以太网交换机能够实现以下两个功能：
  - 能够处理带有VLAN标记的帧，也就是IEEE 802.1Q帧。
  - 交换机的各接口可以支持不同的接口类型，不同接口类型的接口对帧的处理方式有所不同。
- IEEE 802.1Q帧也称为Dot One Q帧，它对以太网V2的MAC帧格式进行了扩展：
  - 在源地址字段和类型字段之间插入了4字节的VLAN标签（tag）字段。
  - 最大长度为1522字节



**标签协议标识符TPID：**长度为16比特，其值固定为0x8100，表示该帧是IEEE 802.1Q帧。

**优先级PRI：**长度为3比特，取值范围是0~7，值越大优先级越高。当网络阻塞时，设备优先发送优先级高的802.1Q帧。

**规范格式指示符CFI：**长度为1比特，取值为0表示MAC地址以规范格式封装，取值为1表示MAC地址以非规范格式封装。对于以太网，CFI的取值为0。

**虚拟局域网标识符VID：**长度为12比特，取值范围是0~4095，其中0和4095保留不使用。

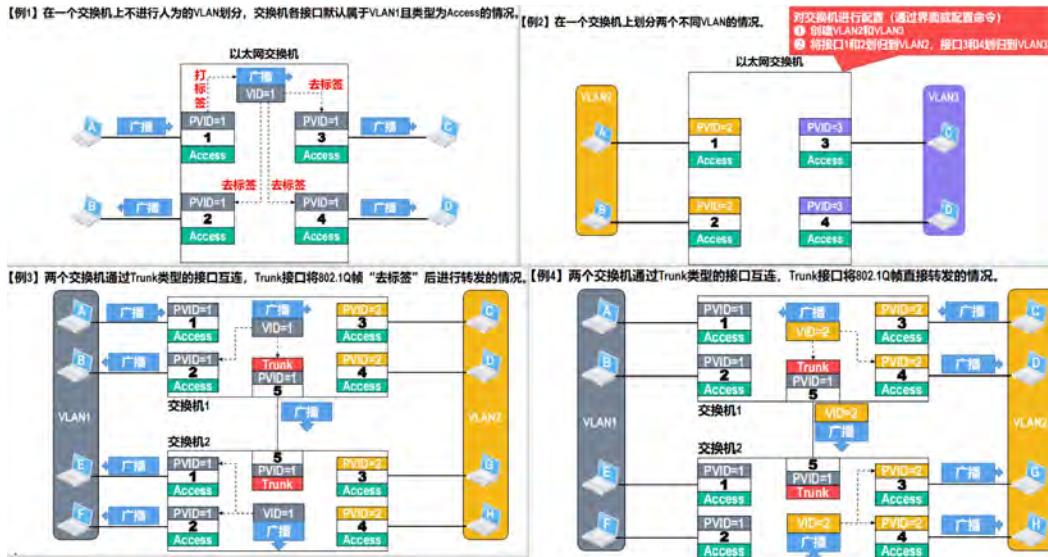
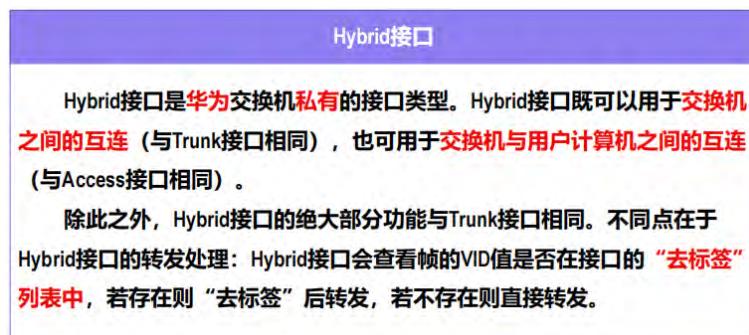
VID是802.1Q帧所属VLAN的编号，设备利用VID来识别帧所属的VLAN。

广播帧只在同一VLAN内转发，这样就将广播域限制在了一个VLAN内。

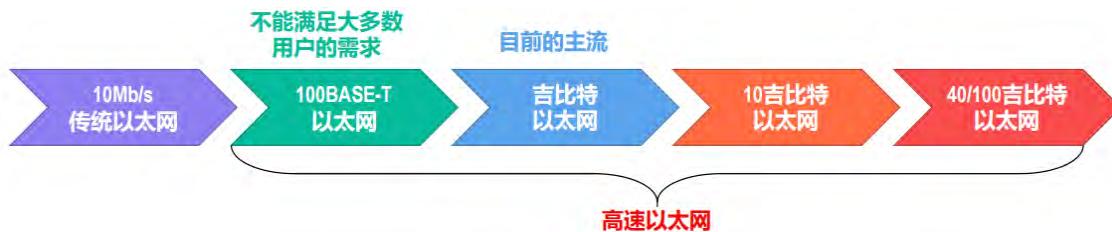
- 802.1Q帧一般不由用户主机处理，而是由以太网交换机来处理：
  - “打标签”：
    - 当交换机收到普通的以太网MAC帧时，会给其插入4字节的VLAN标签使之成为802.1Q帧。
  - “去标签”
    - 当交换机转发802.1Q帧时，可能会删除其4字节的VLAN标签使之成为普通的以太网MAC帧。
    - 交换机转发802.1Q帧时也有可能不进行“去标签”处理，是否进行“去标签”处理取决于交换机的接口类型。

### 3.7.3.2. 以太网交换机的接口类型

- 根据接口在接收帧和发送帧时对帧的处理方式的不同，以及接口连接对象的不同，以太网交换机的接口类型一般分为Access和Trunk两种。
- 当以太网交换机上电启动后，若之前未对其各接口进行过VLAN的相关设置，则各接口的接口类型默认为Access，并且各接口的缺省VLAN ID为1，即各接口默认属于VLAN1。
  - 对于思科交换机，接口的缺省VLAN ID称为本征VLAN (Native VLAN)。
  - 对于华为交换机，接口的缺省VLAN ID称为端口VLAN ID (Port VLAN ID)，简记为PVID。
- 交换机的每个接口有且仅有一个PVID



### 3.8. 以太网的发展



#### 3.8.1. BASE-T以太网

- 100BASE-T以太网是指在双绞线上传输基带信号的速率为100Mb/s的以太网，也称为快速以太网（Fast Ethernet）。
 

**100BASE-T**

速率 100Mb/s  
传输媒体 双绞线  
基带信号
- 100BASE-T以太网与10Mb/s标准以太网（传统以太网）一样，仍然使用IEEE 802.3的帧格式和CSMA/CD协议。
- 100BASE-T以太网为了与10Mb/s标准以太网保持兼容，需要以太网最小帧长保持不变，即仍为64字节。
- 网段的最大电缆长度从1000m减小到100m
- 争用期缩短为 $5.12\mu s$
- 帧间最小间隔缩短为 $0.96\mu s$
- 100BASE-T以太网还可以使用以太网交换机来提供比集线器更好的服务质量，即在全双工方式下无碰撞工作。因此，使用交换机的100BASE-T以太网，工作在全双工方式下，并不使用CSMA/CD协议。
- 1995年，IEEE的802委员会正式批准100BASE-T以太网的标准为802.3u。实际上，IEEE 802.3u只是对原有IEEE 802.3标准的补充。
- 除100BASE-T以太网外，百兆以太网有多种不同的物理层标准：

名称	传输介质	网段最大长度	说 明
100BASE-TX	铜缆	100m	两对UTP5类线或屏蔽双绞线STP
100BASE-T4	铜缆	100m	4对UTP3类线或5类线
100BASE-FX	光缆	2000m	两根光纤，发送和接收各用一条

#### 3.8.2. 吉比特以太网

##### 3.8.2.1. 简介

- 吉比特以太网也称为千兆以太网（Gigabit Ethernet）。1998年，千兆以太网的标准802.3z成为正式标准。
- 近几年来，千兆以太网已迅速占领市场，成为了以太网的主流产品。
- IEEE 802.3z千兆以太网的主要特点有：
  - 速率为1000Mb/s (1Gb/s)
  - 使用IEEE 802.3的帧格式（与10Mb/s和100Mb/s以太网相同）
  - 支持半双工方式（使用CSMA/CD协议）和全双工方式（不使用CSMA/CD协议）
  - 兼容10BASE-T和100BASE-T技术

### 3.8.2.2. 载波延伸

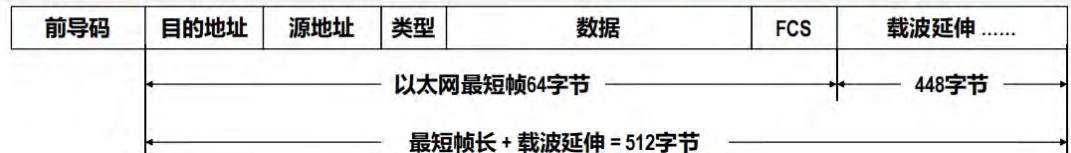
■ 当千兆以太网工作在半双工方式时，需要使用CSMA/CD协议。由于速率已经提高到了1000Mb/s，因此只有减小网段最大长度或增大最小帧长，才能使3.4.2节中介绍的以太网的参数 $a(\tau/T_0)$ 保持为较小的数值。

- 若将网段最大长度减小到10m，则网络基本失去了应用价值。
- 若将最小帧长增大到640字节，则当上层交付的待封装的协议数据单元PDU很短时，开销就会太大。

千兆以太网的网段最大长度仍保持为100m，最小帧长仍保持为64字节（与10BASE-T和100BASE-T兼容）。

■ 这就需要使用**载波延伸**（Carrier Extension）的办法，将**争用期增大为512字节的发送时间而保持最小帧长仍为64字节**。

- 只要发送的MAC帧的长度不足512字节时，就在MAC帧尾部填充一些特殊字符，使MAC帧的长度增大到512字节。



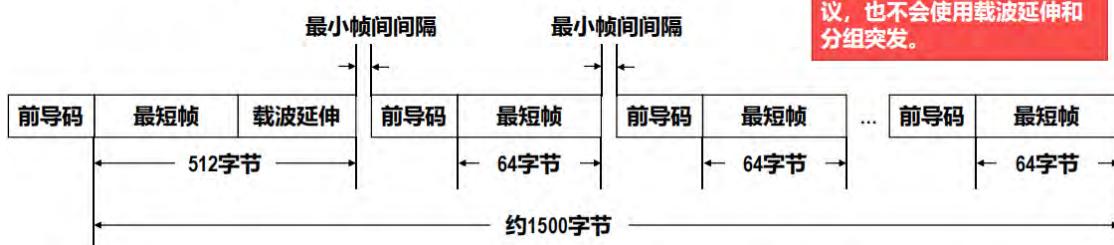
### 3.8.2.3. 分组突发

■ 在使用载波延伸的机制下，如果原本**发送的是大量的64字节长的短帧**，则**每一个短帧都会被填充448字节的特殊字符**，这样会**造成很大的开销**。

■ 因此，千兆以太网还使用了**分组突发**（Packet Bursting）功能。也就是当**有很多短帧要连续发送时，只将第一个短帧用载波延伸的方法进行填充**，而其后面的一系列短帧不用填充就可一个接一个地发送，它们之间只需空开必要的帧间最小间隔即可。

- 这样就形成了一连串分组的突发，当累积发送1500字节或稍多一些为止。

当千兆以太网工作在全双工方式时，不使用CSMA/CD协议，也不会使用载波延伸和分组突发。



### 3.8.2.4. 物理层标准

■ 千兆以太网有多种不同的物理层标准：

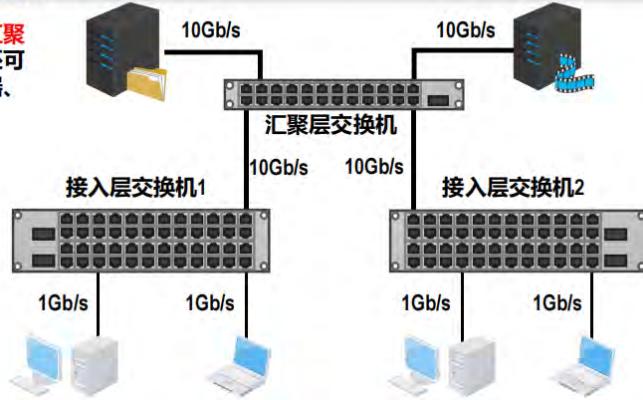
名称	传输介质	网段最大长度	说 明
1000BASE-SE	光缆	550m	多模光纤 (50和62.5 $\mu m$ )
1000BASE-LX	光缆	5000m或550m	单模光纤 (10 $\mu m$ ) 或多模光纤 (50和62.5 $\mu m$ )
1000BASE-CX	铜缆	25m	使用2对屏蔽双绞线电缆STP
1000BASE-T	铜缆	100m	使用4对UTP5类线

### 3.8.3. 吉比特以太网

#### 3.8.3.1. 简介

- 2002年6月，IEEE 802.3ae委员会通过**10吉比特以太网（10GE）**的正式标准，10GE也称为**万兆以太网**。
- 万兆以太网并不是将千兆以太网的速率简单地提高了10倍。万兆以太网的目标是将以太网从**局域网范围**（校园网或企业网）**扩展到城域网与广域网**，成为城域网和广域网的主干网的主流技术之一。
- IEEE 802.3ae万兆以太网的主要特点有：
  - 速率为10Gb/s
  - 使用IEEE 802.3标准的帧格式（与10Mb/s、100Mb/s和1Gb/s以太网相同）
  - 保留IEEE 802.3标准对以太网最小帧长和最大帧长的规定。这是为了用户升级以太网时，仍能和较低速率的以太网方便地通信。
  - 只工作在全双工方式而不存在争用媒体的问题，因此不需要使用CSMA/CD协议，这样传输距离就不再受碰撞检测的限制。
  - 增加了支持城域网和广域网的物理层标准

- 万兆以太网交换机常作为千兆以太网的**汇聚层交换机**，与千兆以太网交换机相连，还可以连接对传输速率要求极高的视频服务器、文件服务器等设备。



### 3.8.3.2. 物理层标准

- 万兆以太网有多种不同的物理层标准：

名称	传输介质	网段最大长度	说 明
10GBASE-SR	光缆	300m	多模光纤 (850nm)
10GBASE-LR	光缆	10km	单模光纤 (1300nm)
10GBASE-ER	光缆	40km	单模光纤 (1500nm)
10GBASE-CX4	铜缆	15m	使用4对双芯同轴电缆 (twinax)
10GBASE-T	铜缆	100m	使用4对6A类UTP双绞线

### 3.8.4. 吉比特/100吉比特以太网

#### 3.8.4.1. 简介

- 2010年，IEEE发布了**40吉比特/100吉比特以太网**（40GE/100GE）的IEEE 802.3ba标准，40GE/100GE也称为**四万兆/十万兆以太网**。
- 为了使**以太网能够更高效、更经济地满足局域网、城域网和广域网的不同应用需求**，IEEE 802.3ba标准定义了**两种速率类型**：
  - 40Gb/s主要用于**计算应用**
  - 100Gb/s主要用于**汇聚应用**
- IEEE 802.3ba标准**只工作在全双工方式（不使用CSMA/CD协议）**，但仍**使用IEEE 802.3标准的帧格式并遵守最小帧长和最大帧长的规定**。
- IEEE 802.3ba标准的**两种速率各有4种不同的传输媒体**

物理层	40GE	100GE
在背板上传输至少超过1m	40GBASE-KR4	
在铜缆上传输至少超过7m	40GBASE-CR4	100GBASE-CR10
在多模光纤上传输至少100m	40GBASE-SR4	100GBASE-SR10
在单模光纤上传输至少10km	40GBASE-LR4	100GBASE-LR4
在单模光纤上传输至少40km		100GBASE-ER4

## 3.9. 无线局域网

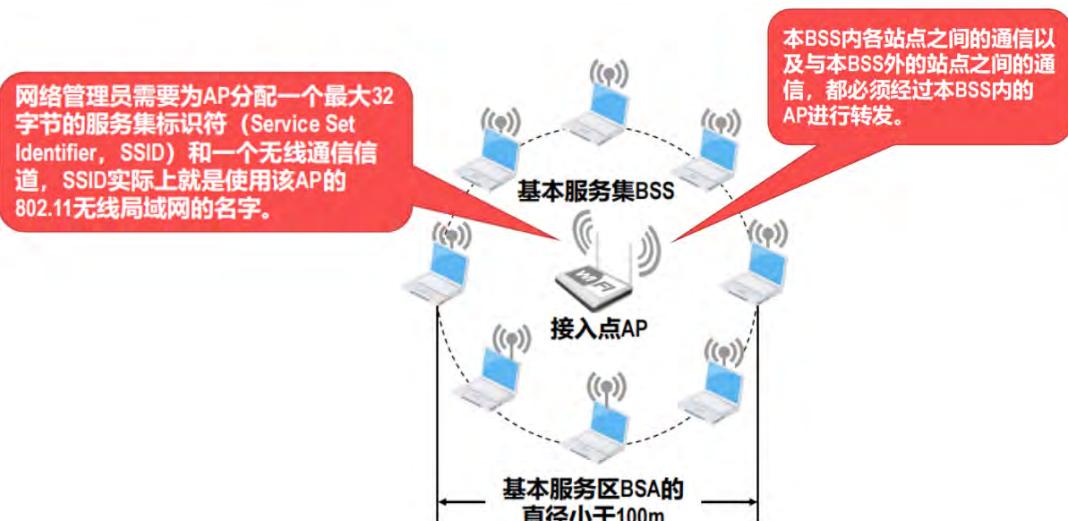
### 3.9.1. 无线局域网的组成

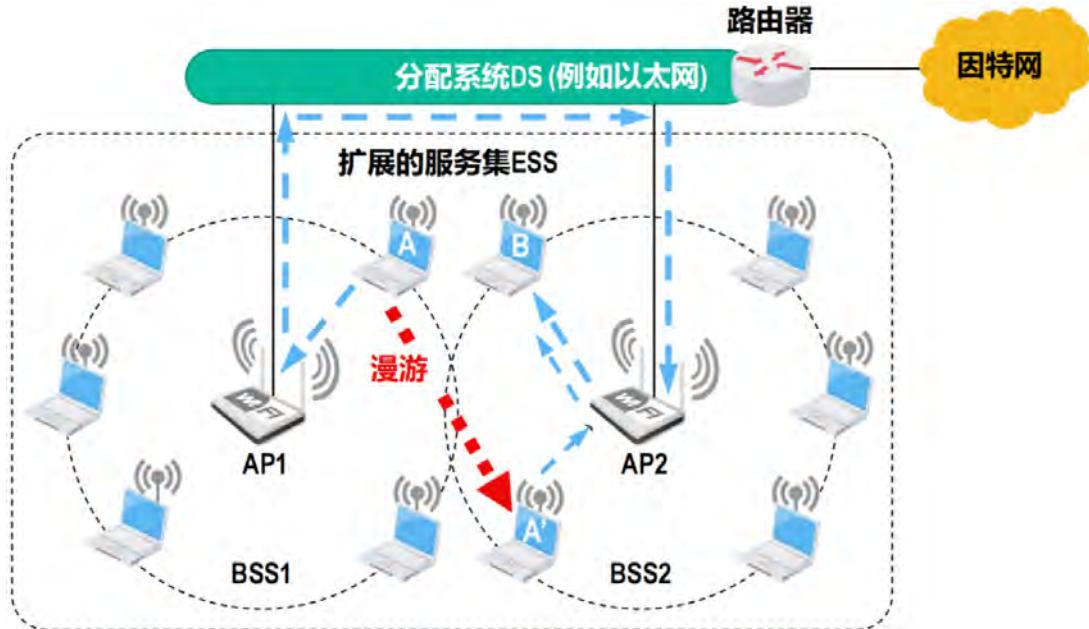
- 随着移动通信技术的发展，**无线局域网**（Wireless Local Area Network，WLAN）自20世纪80年代末以来逐步进入市场。
- IEEE于1997年制定了**无线局域网**的协议标准802.11，**802.11无线局域网**是目前应用最广泛的**无线局域网**之一，人们更多地将其简称为**Wi-Fi**（Wireless Fidelity，无线保真度）。
- 802.11无线局域网使用最多的是它的**固定基础设施的组网方式**。

#### 3.9.1.1. 无线局域网分类

##### 3.9.1.1.1. 有固定基础设施的

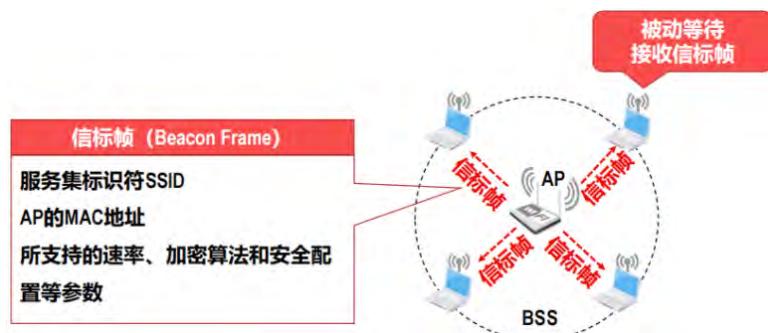
- 固定基础设施
  - 预先建立的
  - 能够覆盖一定地理范围的
  - 多个固定的通信基站





- 802.11标准并没有定义实现漫游的具体方法，仅定义了以下一些基本服务

- 关联 (Association) 服务
  - 移动站与接入点AP建立关联的方法有以下两种：
    - 被动扫描
    - 主动扫描



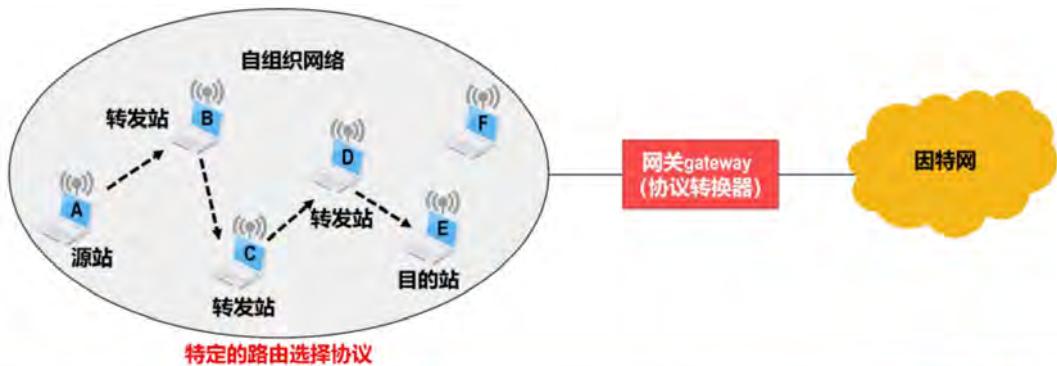
- 主动扫描



- 重建关联 (Reassociation) 服务和分离 (Dissociation) 服务
  - 如果一个移动站要把与某个接入点AP的关联转移到另一个AP，就可以使用重建关联服务；若要终止关联服务，就应使用分离服务。

### 3.9.1.1.2. 无固定基础设施的

- 自组织网络 (ad hoc Network)
  - 无基站或接入点AP
  - 是由一些对等的移动站点构成的临时网络
  - 数据在网络中被多跳存储转发
  - 转发站需要路由功能



自组织网络组网方便，不需要基站，并且具有非常好的生存性，这使得自组织网络在军用和民用领域都有很好的应用前景。

802.11无线局域网的ad hoc模式允许网络中的各站点在其通信范围内直接通信，也就是支持站点间的单跳通信，而标准中并没有包括多跳路由功能。因此，802.11无线局域网的ad hoc模式应用较少。

### 3.9.2. 无线局域网的物理层

#### 3.9.2.1. 概述

- 802.11无线局域网的物理层非常复杂，依据工作频段、调制方式、传输速率等，可将其分为多种物理层标准

标准	频段	调制方式	最高速率	特 点	时间
802.11b	2.4GHz	DSSS	11Mb/s	信号传播距离远且不易受阻碍，最高速率较低	1999年
802.11a	5GHz	OFDM	54Mb/s	信号传播距离较短且易受阻碍，最高速率较高，支持更多用户同时上网	1999年
802.11g	2.4GHz	OFDM	54Mb/s	信号传播距离远且不易受阻碍，最高速率较高，支持更多用户同时上网	2003年
802.11n	2.4GHz 5GHz	MIMO OFDM	600Mb/s	使用多个发射和接收天线达到更高的最高速率，使用双倍带宽 (40MHz) 时最高速率可达600Mb/s	2009年

- 802.11无线网卡一般会被做成多模的，以便能适应多种不同的物理层标准，例如支持802.11b/g/n。
- 无线局域网最初还使用红外技术 (infrared, IR) 和跳频扩频 (Frequency Hopping Spread Spectrum, FHSS) 技术，但目前已经很少使用了。
- 跳频技术的发明人，是好莱坞黄金时代的著名女星海蒂·拉玛，跳频技术为CDMA和Wi-Fi等无线通信技术奠定了基础。因此，海蒂·拉玛被誉为“Wi-Fi之母”。

#### 3.9.2.2. 近年新的物理层标准

- 最近几年，802.11无线局域网又有一些新的物理层标准陆续推出



### 3.9.3. 无线局域网使用CSMA/CA协议的原因

#### 3.9.3.1. 区别

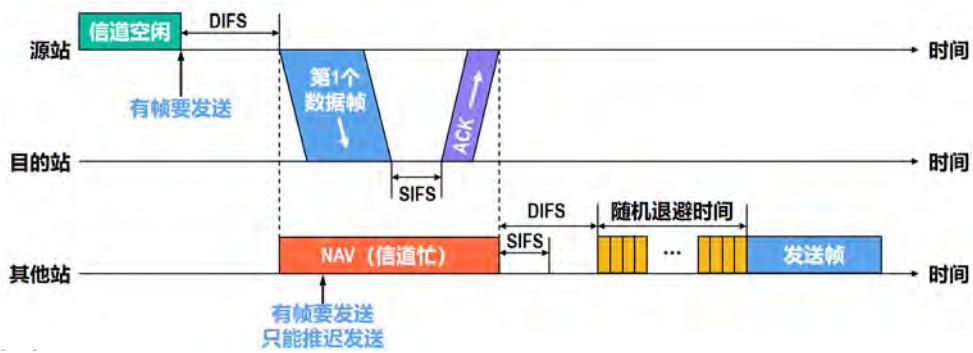
- 对于802.11无线局域网，其使用无线信道传输数据，这与共享总线以太网使用有线传输介质不同。因此，802.11无线局域网不能简单照搬共享总线以太网使用的CSMA/CD协议。
- 802.11无线局域网采用了另一种称为CSMA/CA的协议，也就是载波监听多址接入/碰撞避免（Carrier Sense Multiple Access/Collision Avoidance, CSMA/CA）。
- CSMA/CA协议仍然采用CSMA/CD协议中的CSMA，以“先听后说”的方式来减少碰撞的发生，但是将“碰撞检测CD”改为了“碰撞避免CA”。
  - 尽管CA表示碰撞避免，但并不能避免所有的碰撞，而是尽量减少碰撞发生的概率。

#### 3.9.3.2. 原因

- 由于无线信道的传输环境复杂且信号强度的动态范围非常大，在802.11无线网卡上接收到的信号强度一般都远远小于发送信号的强度，信号强度甚至相差百万倍。因此，如果要在802.11无线网卡上实现碰撞检测，对硬件的要求非常高。
- 即使能够在硬件上实现碰撞检测功能，但由于无线电波传播的特殊性（存在隐蔽站问题），还会出现无法检测到碰撞的情况，因此实现碰撞检测并没有意义。
  - 无线局域网的隐蔽站问题（Hidden Station Problem）



### 3.9.4. CSMA/CA协议的基本工作原理



#### 3.9.4.1. DIFS

- DCF帧间间隔DIFS的长度为 $128\mu s$ ，在DCF方式中，DIFS用来发送数据帧和管理帧。DCF是分布式协调功能（Distributed Coordination Function，DCF）的英文缩写词。在DCF方式下，没有中心控制站点，每个站点使用CSMA/CA协议通过争用信道来获取发送权。DCF方式是802.11定义的默认方式（必须实现）。
- 等待DIFS间隔是考虑到可能有其他的站有高优先级的帧要发送。

#### 3.9.4.2. 虚拟载波监听（Virtual Carrier Sense）机制

- 帧首部中的“持续时间”字段的值指出了源站要占用信道的时间（包括目的站发回确认帧所需的时间）
- 当某个站检测到正在信道中传送的帧首部中的“持续时间”字段时，就调整自己的网络分配向量（Network Allocation Vector，NAV）。NAV指出了完成这次帧的传送且信道转入空闲状态所需的时间。

#### 3.9.4.3. SIFS

- 短帧间间隔（Short Interframe Space，SIFS）的长度为 $28\mu s$ ，它是最短的帧间间隔，用来分隔开属于一次对话的各帧。一个站点应当能够在这段时间内从发送方式切换到接收方式。使用SIFS的帧类型有ACK帧、CTS帧等。
- 由于无线信道的误码率较高，CSMA/CA协议还需要使用停止——等待的确认机制来实现可靠传输，这与使用CSMA/CD协议的共享式以太网不同。
- 当某个站在发送帧时，很可能有多个站都在监听信道并等待发送帧，一旦信道空闲，这些站几乎同时发送帧而产生碰撞。

#### 3.9.4.4. 退避算法

##### 3.9.4.4.1. 使用情况

- 为了避免上述情况，所有要发送帧的站检测到信道从忙转为空闲后，都要执行退避算法。这样不仅可以减少发生碰撞的概率，还可避免某个站长时间占用无线信道。
- 当某个站要发送数据帧时，仅在这种情况下才不使用退避算法：检测到信道空闲，并且该数据帧不是成功发送完上一个数据帧之后立即连续发送的数据帧。除此之外的以下情况，都必须使用退避算法：
  - 在发送帧之前检测到信道处于忙态
  - 在每一次重传一个帧时
  - 在每一次成功发送帧后要连续发送下一个帧时

##### 3.9.4.4.2. 算法内容



- 在执行退避算法时，站点为退避计时器设置一个随机的退避时间：
  - 当退避计时器的时间减小到零时，就开始发送数据；
  - 当退避计时器的时间还未减小到零时而信道又转变为忙状态，这时就冻结退避计时器的数值，重新等待信道变为空闲，再经过帧间间隔DIFS后，继续启动退避计时器。
- 在进行第*i*次退避时，退避时间在时隙编号{0, 1, ..., 2^{i-1}-1}中随机选择一个，然后乘以基本退避时间（也就是一个时隙的长度）就可以得到随机的退避时间。当时隙编号达到255时（对应于第6次退避）就不再增加了。

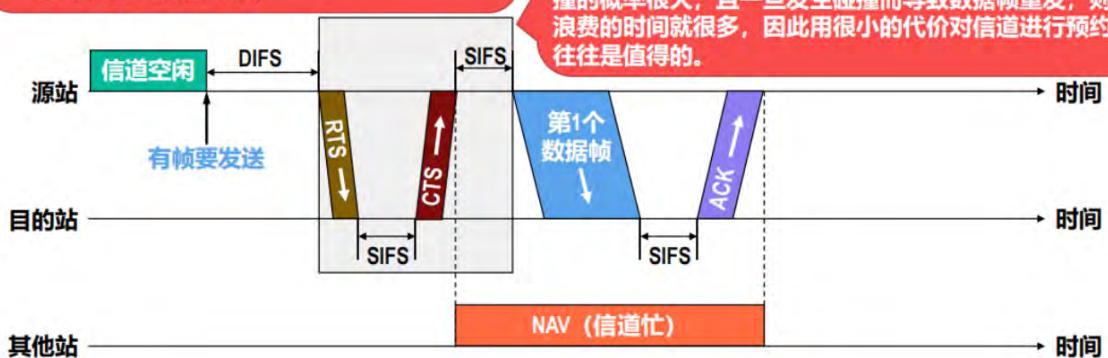
#### 3.9.4.4.3. 对信道进行预约

- 为了进一步降低发生碰撞的概率，802.11无线局域网允许源站对信道进行预约。
- RTS (Request To Send) 帧是短的控制帧
  - 包括源地址
  - 目的地址
  - 本次通信（包括目的站发回确认帧所需的时间）所需的持续时间。
- CTS (Clear To Send) 帧是短的响应控制帧
  - 包括本次通信所需的持续时间（从RTS帧中将此持续时间复制到CTS帧中）。
- 除源站和目的站的其他各站，在收到CTS帧或数据帧后就推迟访问信道。这样就确保了源站和目的站之间的通信不会受到其他站的干扰
- 若RTS帧发生碰撞，源站就不可能收到CTS帧，源站会执行退避算法重传RTS帧。

尽管如此，802.11无线局域网仍为用户提供了以下三种选择：

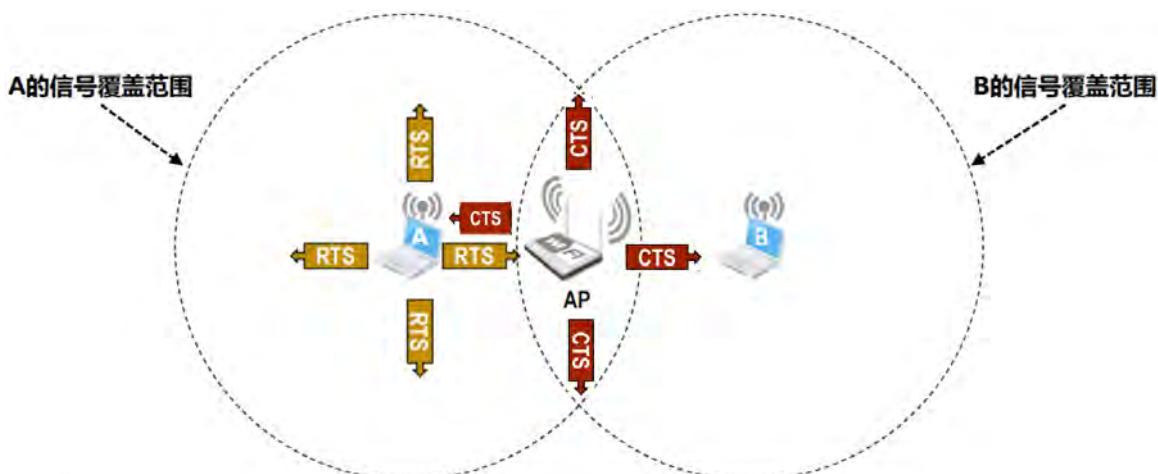
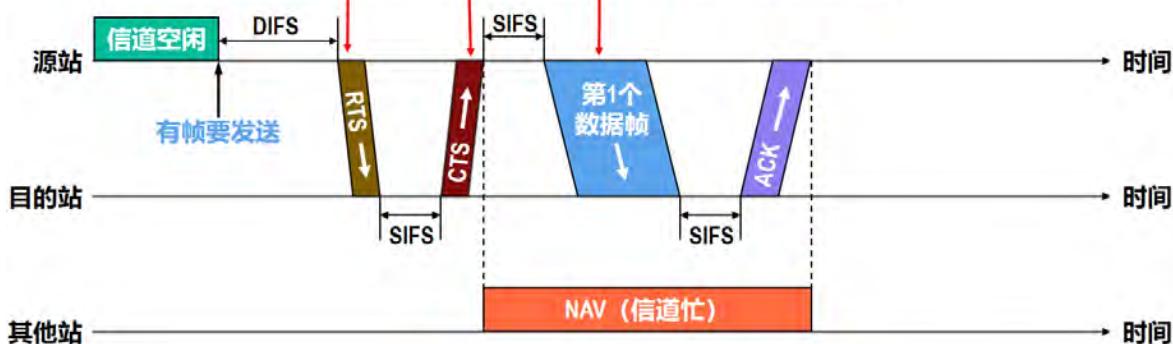
- ① 使用RTS帧和CTS帧。
- ② 只有当数据帧的长度超过某个数值时才使用RTS帧和CTS帧。
- ③ 不使用RTS帧和CTS帧。

使用RTS帧和CTS帧进行信道预约会带来额外的开销。但由于RTS帧和CTS帧都很短，发生碰撞的概率、碰撞产生的开销以及本身的开销都很小。对于一般的数据帧，其发送时延往往远大于传播时延（因为是局域网），碰撞的概率很大，且一旦发生碰撞而导致数据帧重发，则浪费的时间就很多，因此用很小的代价对信道进行预约往往是值得的。



由于RTS帧和CTS帧都会携带通信需要持续的时间，这与之前介绍过的数据帧可以携带通信所需持续时间的虚拟载波监听机制是一样的，因此使用RTS帧和CTS帧进行信道预约，也属于虚拟载波监听机制。

利用虚拟载波监听机制，站点只要监听到数据帧、RTS帧或CTS帧中的任何一个，就能知道信道将被占用的持续时间，而不需要真正监听到信道上的信号，因此虚拟载波监听机制能减少隐蔽站带来的碰撞问题。



### 3.9.5. 无线局域网的MAC帧

数据帧	控制帧	管理帧
<ul style="list-style-type: none"> <li>● 用于在站点间传输数据</li> </ul>	<ul style="list-style-type: none"> <li>● 通常与数据帧搭配使用</li> <li>● 负责区域的清空、虚拟载波监听的维护以及信道的接入，并于收到数据帧时予以确认。</li> <li>● ACK帧、RTS帧以及CTS帧等都属于控制帧。</li> </ul>	<ul style="list-style-type: none"> <li>● 用于加入或退出无线网络，以及处理AP之间连接的转移事宜。</li> <li>● 信标帧、关联请求帧以及身份认证帧等都属于管理帧。</li> </ul>



协议版本	类型	子类型	去往DS	来自DS	更多分片	重试	功率管理	更多数据	WEP	顺序
2比特	2比特	4比特	1比特	1比特	1比特	1比特	1比特	1比特	1比特	1比特



用来实现802.11的可靠传输，对数据帧进行编号。

用于实现CSMA/CA的虚拟载波监听和信道预约机制。在数据帧、RTS帧和CTS帧中用该字段指出将要持续占用信道的时长。



取决于帧控制字段中的“去往DS”（到分配系统）和“来自DS”（分配系统）这两个字段的值。



802.11无线局域网数据帧地址字段的4种使用情况

去往DS	来自DS	地址1	地址2	地址3	地址4
0	0	目的地址	源地址	BSSID	未被使用
0	1	目的地址	发送AP地址	源地址	未被使用
1	0	接收AP地址	源地址	目的地址	未被使用
1	1	接收AP地址	发送AP地址	目的地址	源地址

### 3.10. 题目

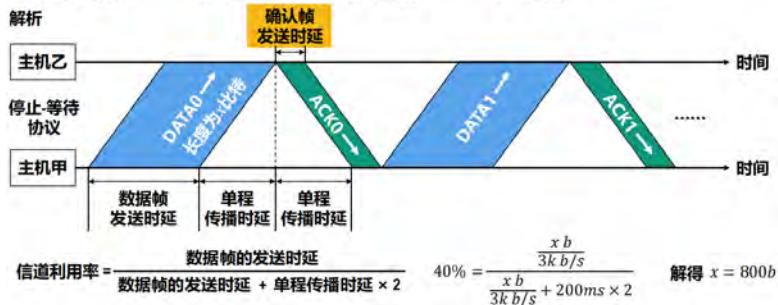
从滑动窗口的角度对比停止-等待协议、回退N帧协议和选择重传协议

停止-等待协议	回退N帧协议	选择重传协议
仅用1个比特给分组编号 $W_T = 1$ $W_R = 1$	用 $n(n > 1)$ 个比特给分组编号 $1 < W_T \leq (2^n - 1)$ $W_R = 1$	用 $n(n > 1)$ 个比特给分组编号 $W_R$ 超过上限 无法分辨新旧数据分组 $1 < W_R \leq W_T$ $W_T + W_R \leq 2^n$
		确保接收窗口向前滑动后，落入接收窗口内的新序号与之前的旧序号没有重叠，避免无法分辨新旧数据分组。 当 $W_R$ 取最大值 $2^{(n-1)}$ 时， $W_T$ 能取到的最大值也为 $2^{(n-1)}$ 。

停止等待协议

【2018年题36】主机甲采用停-等协议向主机乙发送数据，数据传输速率是3kbps，单向传播延时是200ms，忽略确认帧的传输延时。当信道利用率为40%时，数据帧的长度为 (D)。

- A. 240比特    B. 400比特    C. 480比特    D. 800比特

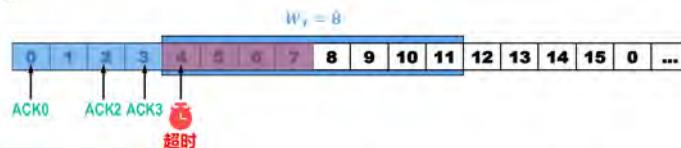


## 回退N帧协议

【2009年题35】数据链路层使用后退N帧(GBN)协议，发送方已经发送了编号0~7的帧。当计时器超时时，若发送方只收到了0、2、3号帧的确认，则发送方需要重发的帧数是 (C)。

- A. 2    B. 3    C. 4    D. 5

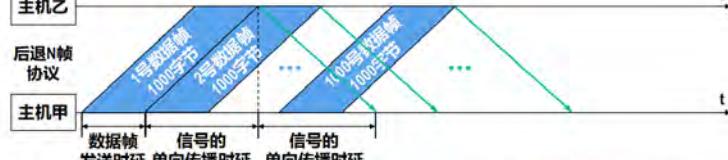
解析



【2014年题36】主机甲与主机乙之间使用后退N帧(GBN)协议传输数据，甲的发送窗口尺寸为1000，数据帧长为1000字节，信道带宽为100Mbps，乙每收到一个数据帧就立即利用一个短帧(忽略其传输延迟)进行确认，若甲乙之间的单向传播延迟是50ms，则甲可以达到的最大平均数据传输速率约为 (C)。

- A. 10Mbps    B. 20Mbps    C. 80Mbps    D. 100Mbps

解析



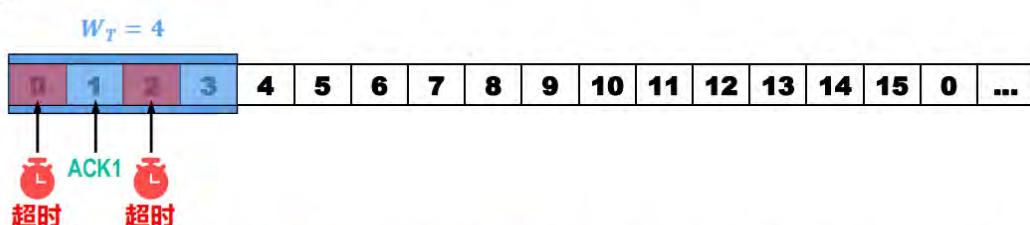
$$\begin{aligned} \text{甲可以达到的最大平均数据传输速率} &= \frac{\text{可发送的数据量}}{\text{(数据帧的发送时延} + \text{信号的单向传播时延} \times 2)} \\ &= ((1000 \times 8) \times 1000) \div ((1000 \times 8) \div (100 \times 10^6) + (50 \times 10^{-3}) \times 2) \\ &\approx 80Mbps \end{aligned}$$

## 选择重传协议

【2011年题35】数据链路层采用选择重传协议(SR)传输数据，发送方已发送了0~3号数据帧，现已收到1号帧的确认，而0、2号帧依次超时，则此时需要重传的帧数是 (B)。

- A. 1    B. 2    C. 3    D. 4

解析



与回退N帧协议不同，选择重传协议不采用累积确认，接收方需要对每一个正确接收的数据分组进行逐一确认。

发送方仅重传未收到确认而超时的数据帧，因此重传0号和2号这两个数据帧。

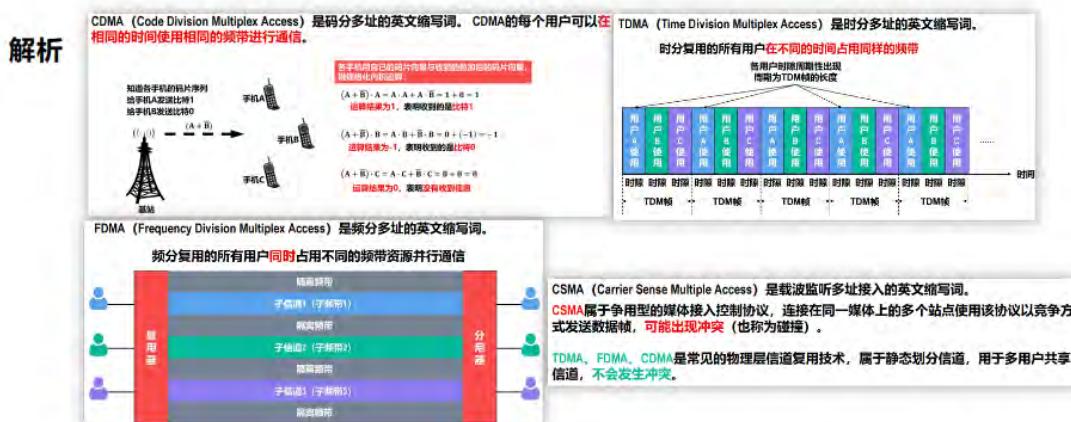
【2018年题34】下列选项中，不属于物理层接口规范定义范畴的是（C）。

- |         |                                 |
|---------|---------------------------------|
| A. 接口形状 | 物理层接口的机械特性                      |
| B. 引脚功能 | 物理层接口的功能特性                      |
| C. 物理地址 | <b>数据链路层使用的地址 又称为硬件地址或MAC地址</b> |
| D. 信号电平 | 物理层接口的电气特性                      |

- CDMA

【2013年题36】下列介质访问控制方法中，可能发生冲突的是（B）。

- A. CDMA      B. CSMA      C. TDMA      D. FDMA

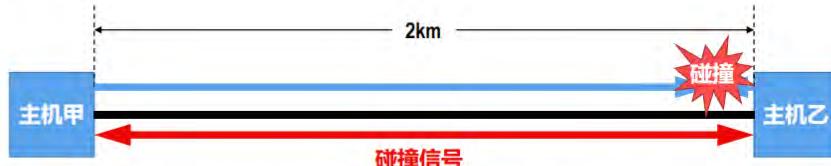


- 争用期

【2010年题47】某局域网采用CSMA/CD协议实现介质访问控制，数据传输速率为10Mbps，主机甲和主机乙之间的距离为2km，信号传播速度是200 000km/s。请回答下列问题，要求说明理由或写出计算过程。

- (1) 若主机甲和主机乙发送数据时发送冲突，则从开始发送数据时刻起，到两台主机均检测到冲突时刻止，最短需经过多长时间？最长需经过多长时间（假设主机甲和主机乙发送数据过程中，其他主机不发送数据）？

**解析**



最短需要经过的时长：两主机之间信号的单程传播时延，记为 $\tau$ 。

最长需要经过的时长：两主机之间信号的往返传播时延，记为 $2\tau$ 。

$$\tau = \frac{2\text{km}}{200000\text{km/s}} = 10^{-5} \text{s}$$

- 最小帧长

**【2009年题37】**在一个采用CSMA/CD协议的网络中，传输介质是一根完整的电缆，传输速率为1Gbps，电缆中的信号传播速度是200 000km/s。若最小数据帧长度减少800比特，则最远的两个站点之间的距离至少需要（D）。

- A. 增加160m      B. 增加80m      C. 减少160m      D. 减少80m
- 排除
- 排除

#### 解析

$$\text{最小帧长} = \text{数据传输速率} \times \text{争用期}2\tau$$

设最远两个站点之间的距离为 $d(m)$ ，最小帧长为 $l(b)$ ，

与题目给定相关已知量一起代入上式：

$$l = \left( \frac{d}{200000 \times 10^3} \times 2 \right) \times 10^9 \longrightarrow d = \frac{l}{10}$$

显然，若最小帧长减少800比特，最远的两个站点之间的距离至少需要减少80m。

- 共享式以太网的信道利用率

**【2015年题36】**下列关于CSMA/CD协议的叙述中，错误的是（B）。

- A. 边发送数据帧，边检测是否发生冲突
- B. 适用于无线网络，以实现无线链路共享
- C. 需要根据网络跨距和数据传输速率限定最小帧长
- D. 当信号传播延迟趋近于0时，信道利用率趋近100%

#### 解析

选项A的描述正确，其所描述的是CSMA/CD协议中的“碰撞检测（冲突检测）CD”。

选项B的描述错误，因为CSMA/CD协议不适用于无线网络。对于802.11无线局域网，可以使用CSMA/CA协议。

选项C的描述正确，因为“最小帧长 = 数据传输速率 × 争用期 $2\tau$ ”。也就是说，最小帧长取决于数据传输速率和争用期 $2\tau$ 。选项C中给出的“网络跨距”相当于给出了“端到端单程传播时延 $\tau$ ”，进而可得出“争用期 $2\tau$ ”。

选项D的描述正确，因为选项D中给出“信号传播延迟趋近于0”，这相当于信号瞬间到达整个网络，网络中各站点瞬间就知道总线被占用，因此不会出现碰撞，进而使信道利用率趋近100%。

- 10BASE-T

**【2019年题34】**100BaseT快速以太网使用的导向传输介质是（A）。

- A. 双绞线
- B. 单模光纤
- C. 多模光纤
- D. 同轴电缆

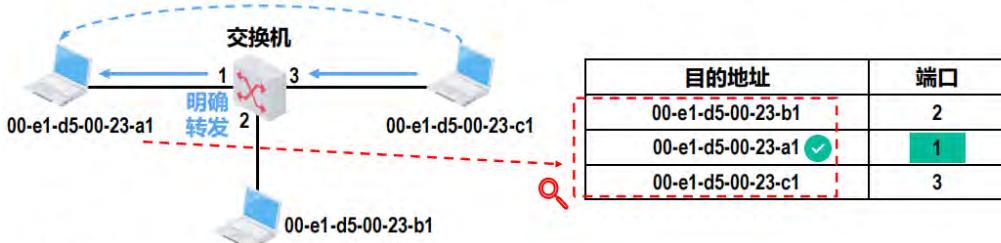
#### 解析



- 以太网交换机

**[2014年 题34]** 某以太网拓扑及交换机当前转发表如下图所示，主机00-e1-d5-00-23-a1向主机00-e1-d5-00-23-c1发送1个数据帧，主机00-e1-d5-00-23-c1收到该帧后，向主机00-e1-d5-00-23-a1发送1个确认帧，交换机对这两个帧的转发端口分别是 (B)。

- A. {3}和{1}      B. {2, 3}和{1}      C. {2, 3}和{1, 2}      D. {1, 2, 3}和{1}



#### 解析

交换机转发数据帧的端口为{2, 3}

交换机转发确认帧的端口为{1}

**[2009年 题36]** 以太网交换机进行转发决策时使用的PDU地址是 (A)。

- A. 目的物理地址      B. 目的IP地址      C. 源物理地址      D. 源IP地址

#### 解析

PDU (Protocol Data Unit) 的意思是**协议数据单元**，它是计算机网络体系结构中对等实体间逻辑通信的对象。

以太网交换机工作在数据链路层（包括物理层），它接收并转发的PDU通常称为**帧**。以太网交换机收到帧后，在转发表中查找帧的**目的MAC地址**所对应的接口号，然后通过该接口转发帧。

**MAC地址又称为硬件地址或物理地址**。请注意：不要被“物理”二字误导认为物理地址属于物理层范畴，物理地址属于数据链路层范畴。

**[2013年 题38]** 对于100Mbps的以太网交换机，当输出端口无排队，以直通交换 (cut-through switching) 方式转发一个以太网帧 (不包括前导码) 时，引入的转发延迟至少是 (B)。

- A. 0μs      B. 0.48μs      C. 5.12μs      D. 121.44μs

#### 解析

以太网帧格式	目的MAC地址 6B	源MAC地址 6B	类型 2B	数据载荷 46B~1500B	FCS 4B
--------	---------------	--------------	----------	-------------------	-----------

物理层在发送以太网帧之前还要在其前面添加8B的前导码。

题目给定：输出端口无排队    直通交换    不包括前导码

可以推出：只要接收完以太网帧的目的MAC地址就可以将帧直接转发到目的端口，而不缓存帧也不检验帧。

引入的**最小转发延迟**就是接收完目的MAC地址所耗费的时间。

$$\frac{6 \times 8}{100 \times 10^6} = 0.48(\mu\text{s})$$

- 共享式以太网与交换式以太网的对比

**[2015年 题37]** 下列关于交换机的叙述中，正确的是 (A)。

- A. 以太网交换机本质上是一种多端口网桥      叙述正确  
 B. 通过交换机互连的一组工作站构成一个冲突域  
 C. 交换机每个端口所连网络构成一个独立的广播域  
 D. 以太网交换机可实现采用不同网络层协议的网络互联

#### 解析

网桥 (bridge) 工作在数据链路层 (包含其下的物理层)，因此**网桥具备属于数据链路层范畴的相关能力**。

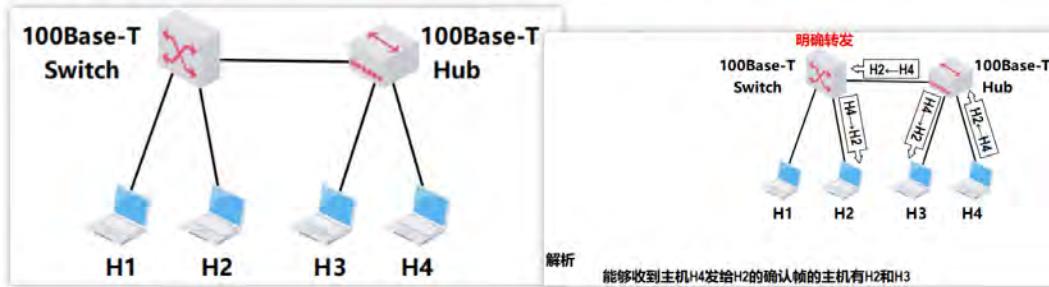
网桥的**接口数量很少**，通常只有2~4个，一般只用来**连接不同的网段**。

1990年面世的**交换式集线器** (Switching Hub)，实质上是**具有多个接口的网桥**，常称为**以太网交换机** (Switch) 或**二层交换机**。



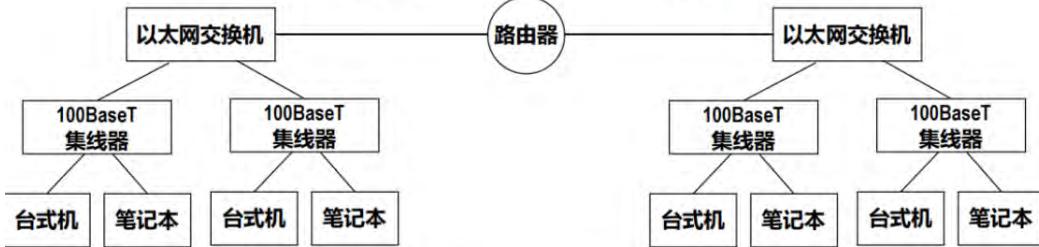
【2016年题35】若下图中的主机H2向主机H4发送1个数据帧，主机H4向主机H2立即发送一个确认帧，则除H4外，从物理层上能够收到该确认帧的主机还有（D）。

- A. 仅H2    B. 仅H3    C. 仅H1、H2    D. 仅H2、H3



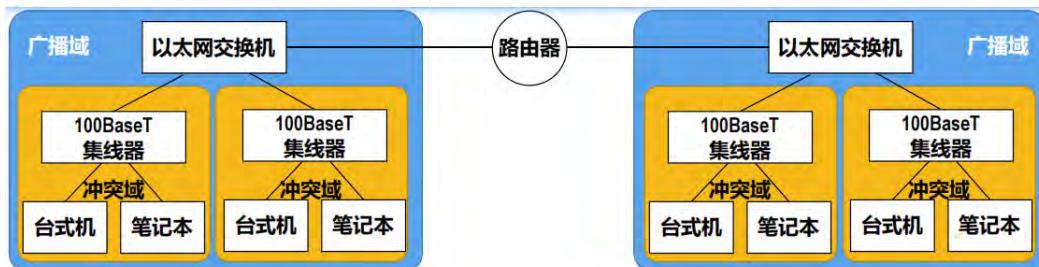
【2020年题35】在下图所示的网络中，冲突域和广播域的个数分别是（C）。

- A. 2, 2    B. 2, 4    C. 4, 2    D. 4, 4



解析

	隔离冲突域	隔离广播域
集线器	✗	✗
交换机	✓	✗
路由器	✓	✓



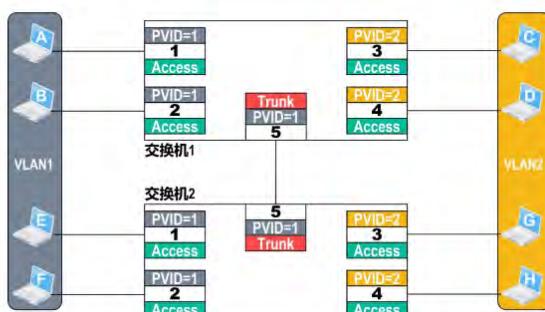
- 虚拟局域网VLAN的实现机制

【练习1】以下关于VLAN的描述中，错误的是（D）。

- A. 从数据链路层的角度看，不同VLAN中的站点之间不能直接通信。
- B. 属于同一个VLAN中的两个站点可能连接在不同的交换机上。
- C. 虚拟局域网只是局域网给用户提供的一种服务，而不是一种新型局域网。
- D. VLAN使用的802.1Q帧的最大长度为1518字节。

【练习2】如右图所示，在交换机1和2上进行了VLAN划分，PVID是交换机端口的本征VLAN，Access和Trunk是交换机的接口类型，以下说法正确的是（C）。

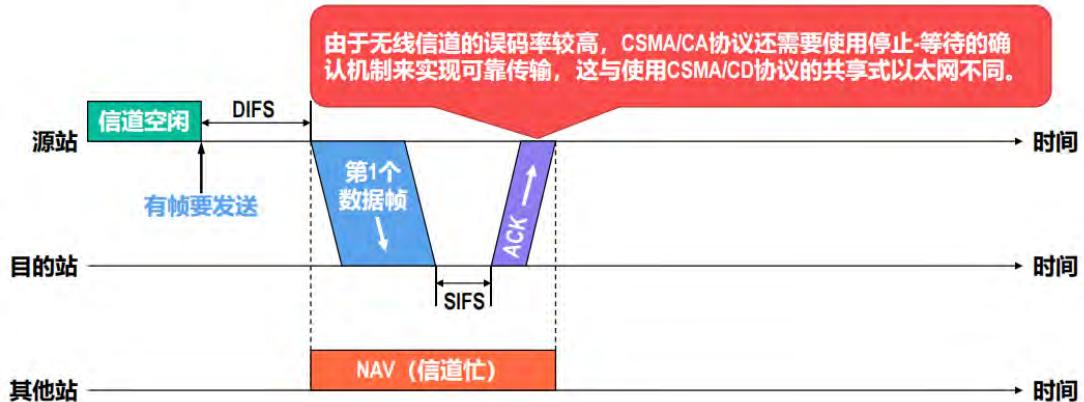
- A. B可以收到H发给B的单播帧
- B. E可以收到D发送的广播帧
- C. 能收到C发送的广播帧的有D、G和H
- D. 能收到E发送的广播帧的有A、B、F、G和H



- CSMA/CA协议的基本工作原理

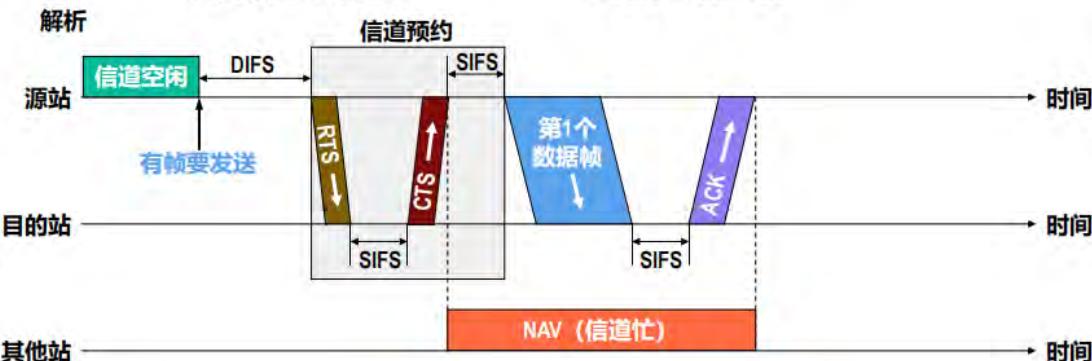
【2011年 题36】下列选项中，对正确接收到的数据帧进行确认的MAC协议是（D）。

- A. CSMA      B. CDMA      C. CSMA/CD      D. CSMA/CA



【2018年 题35】IEEE 802.11无线局域网的MAC协议CSMA/CA进行信道预约的方法是（D）。

- A. 发送确认帧      B. 采用二进制指数退避  
C. 使用多个MAC地址      D. 交换RTS和CTS帧



【2020年 题37】某IEEE 802.11无线局域网中主机H与AP之间发送或接收CSMA/CA帧的过程如下图所示，在H或AP发送帧前所等待的帧间间隔时间（IFS）中最长的是（A）。

- A. IFS1      B. IFS2      C. IFS3      D. IFS4



- 802.11无线局域网的MAC帧

【2017年 题35】在下图所示的网络中，若主机H发送一个封装访问Internet的IP分组的IEEE 802.11数据帧F，则帧F的地址1、地址2和地址3分别是（B）。

- A. 00-12-34-56-78-9a, 00-12-34-56-78-9b, 00-12-34-56-78-9c      B. 00-12-34-56-78-9b, 00-12-34-56-78-9a, 00-12-34-56-78-9i  
 C. 00-12-34-56-78-9b, 00-12-34-56-78-9c, 00-12-34-56-78-9a      D. 00-12-34-56-78-9a, 00-12-34-56-78-9c, 00-12-34-56-78-9i



解析

802.11无线局域网数据帧地址字段的4种使用情况

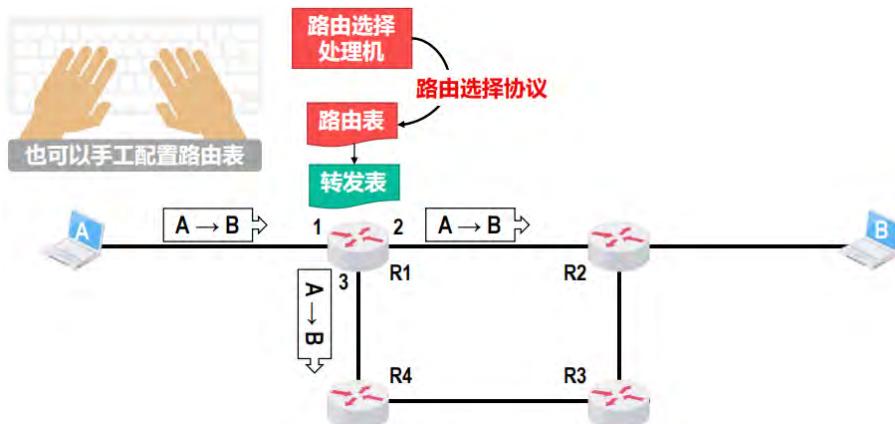
去往DS	来自DS	地址1	地址2	地址3	地址4
0	0	目的地址	源地址	BSSID	未被使用
0	1	目的地址	发送AP地址	源地址	未被使用
1	0	接收AP地址	源地址	目的地址	未被使用
1	1	接收AP地址	发送AP地址	目的地址	源地址

## 4. Network layer

### 4.1. 网络层概述

#### 4.1.1. 分组转发和路由选择

- 网络层的主要任务就是将分组从源主机经过多个网络和多段链路传输到目的主机，可以将该任务划分为组转发和路由选择两种重要的功能。

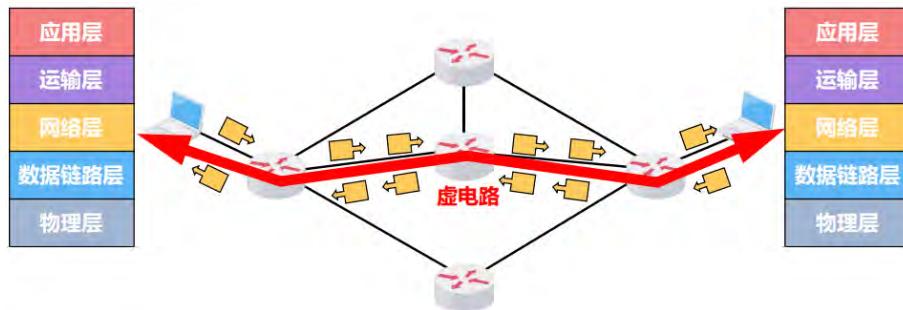


#### 4.1.2. 网络层向其上层提供的两种服务

对比方面	虚电路服务	数据报服务
核心思想	可靠通信应当由网络自身来保证	可靠通信应当由用户主机来保证
连接	必须建立网络层连接	不需要建立网络层连接
目的地址	仅在连接建立阶段使用，之后每个分组使用短的虚电路号	每个分组都必须携带完整的目的地址
分组转发	属于同一条虚电路的分组均按同一路由进行转发	每个分组可走不同的路由
节点故障	所有通过出故障的节点的虚电路均不能工作	出故障的节点可能会丢失分组，一些路由可能会发生变化
分组顺序	总是按发送顺序到达目的主机	到达目的主机时不一定按发送顺序
服务质量	可以将通信资源提前分配给每一个虚电路，因此容易实现	很难实现

#### 4.1.2.1. 面向连接的虚电路服务

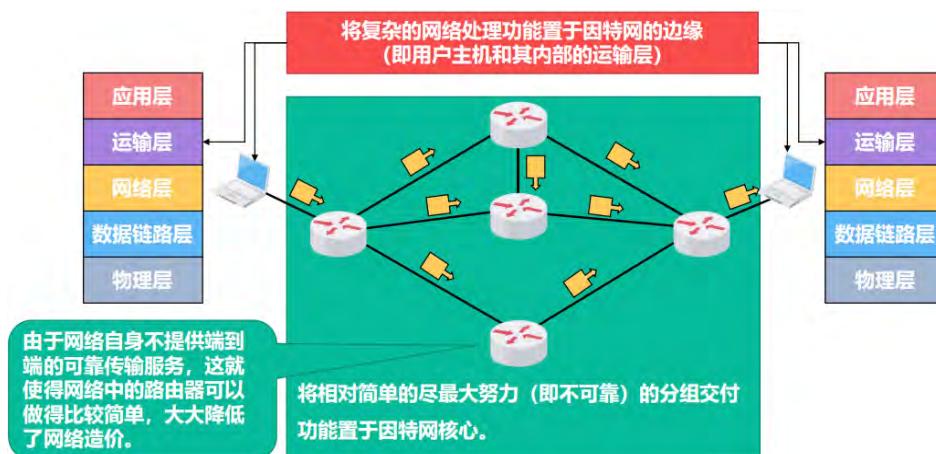
- 核心思想是“可靠通信应由网络自身来保证”。
- 必须首先建立网络层连接——虚电路（Virtual Circuit, VC），以保证通信双方所需的一切网络资源。
- 通信双方沿着已建立的虚电路发送分组。
- 通信结束后，需要释放之前所建立的虚电路。



- 这种通信方式如果再使用可靠传输的网络协议，就可使所发送的分组最终正确（无差错按序到达、不丢失、不重复）到达接收方。
- 很多广域分组交换网都使用面向连接的虚电路服务。例如，曾经的X.25和逐渐过时的帧中继（Frame Relay, FR）、异步传输模式（Asynchronous Transfer Mode, ATM）。
- 然而，因特网的先驱者并没有采用这种设计思想，而是采用了无连接的数据报服务。

#### 4.1.2.2. 无连接的数据报服务

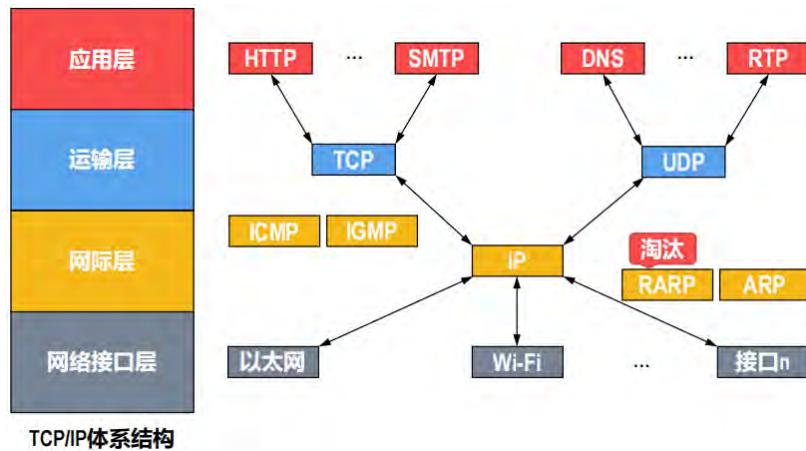
- 核心思想是“可靠通信应由用户主机来保证”。
- 不需要建立网络层连接。
- 每个分组可走不同的路径。因此，每个分组的首部都必须携带目的主机的完整地址。
- 通信结束后，没有需要释放的连接。



- 这种通信方式所传送的分组可能误码、丢失、重复和失序。

## 4.2. 网际协议IP

- 网际协议（Internet Protocol, IP）是TCP/IP体系结构网际层中的核心协议。

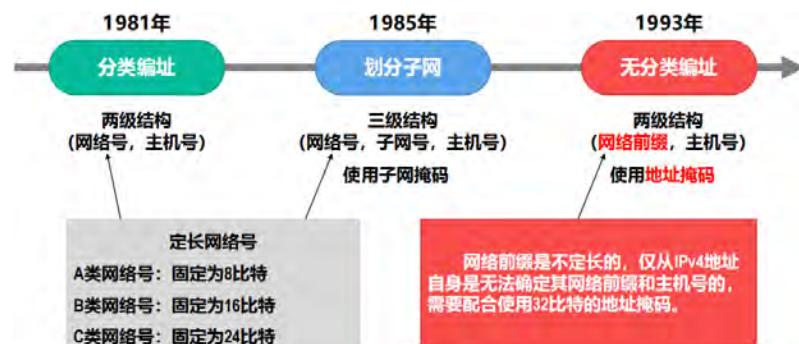


#### 4.2.1. 异构网络互连



- 这些网络的拓扑、性能以及所使用的网络协议都不尽相同，这是由用户需求的多样性造成的，没有一种单一的网络能够适应所有用户的需求。
- 要将众多的异构型网络都互连起来，并且能够互相通信，则会面临许多需要解决的问题。
  - 不同的网络接入机制
  - 不同的差错恢复方法
  - 不同的路由选择技术
  - 不同的寻址方案
  - 不同的最大分组长度
  - 不同的服务（面向连接服务和无连接服务）

#### 4.2.2. IPv4地址及其编址方法

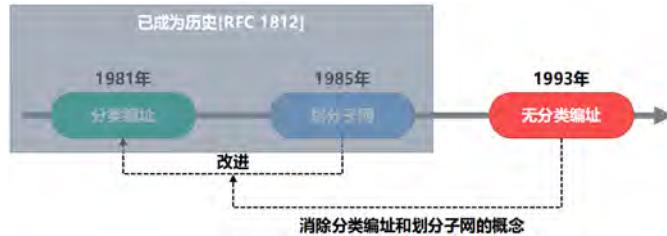


### 4.2.2.1. 概述

- IPv4地址是给因特网（Internet）上的每一个主机（或路由器）的每一个接口分配的一个在全世界范围内唯一的32比特的标识符。

- IPv4地址由因特网名字和数字分配机构（Internet Corporation for Assigned Names and Numbers, ICANN）进行分配。
  - 我国用户可向亚太网络信息中心（Asia Pacific Network Information Center, APNIC）申请IP地址，需要缴纳相应的费用，一般不接受个人申请。
  - 2011年2月3日，因特网号码分配管理局（Internet Assigned Numbers Authority, IANA）（由ICANN行使职能）宣布，IPv4地址已经分配完毕。
  - 我国在2014至2015年也逐步停止了向新用户和应用分配IPv4地址，同时全面开展商用部署IPv6。

- IPv4地址的编址方法经历了三个历史阶段



### 4.2.2.2. 表示方法：点分十进制

- 举例



- 练习

【练习】请将以下IPv4地址转换为点分十进制形式。

- (1) 

0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

10	.	254	.	15	.	240
----	---	-----	---	----	---	-----
- (2) 

1	0	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

172	.	16	.	191	.	247
-----	---	----	---	-----	---	-----
- (3) 

1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

192	.	168	.	165	.	7
-----	---	-----	---	-----	---	---

- 二进制转十进制

$$(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2 = (b_7 \times 2^7 + b_6 \times 2^6 + b_5 \times 2^5 + b_4 \times 2^4 + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0)_{10}$$

8位二进制数的每个位的权值: 128 64 32 16 8 4 2 1

【举例】

$$\begin{aligned} (10101010)_2 &= (1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0)_{10} \\ &= (1 \times 128 + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1)_{10} \\ &= (170)_{10} \end{aligned}$$

$$(11111100)_2 = (255 - 2 - 1)_{10} = (252)_{10}$$

比特0在整个8位无符号二进制整数中数量不多

$$(11110000)_2 = (255 - 8 - 4 - 2 - 1)_{10} = (240)_{10}$$

$$(10000001)_2 = (128 + 1)_{10} = (129)_{10}$$

比特0在整个8位无符号二进制整数中数量较多

- 十进制转二进制

除2取余法（逆序输出）		凑值法（必须记住8位二进制数各位的权值 128 64 32 16 8 4 2 1）	
【举例】		【举例】	
$(130)_{10} = (10000010)_2$		$(171)_{10} = (10101011)_2$	
130 ÷ 2 = 65	余0	$= (1 \times 128 + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1)_{10}$	
65 ÷ 2 = 32	余1	$\downarrow b_7 \quad \downarrow b_6 \quad \downarrow b_5 \quad \downarrow b_4 \quad \downarrow b_3 \quad \downarrow b_2 \quad \downarrow b_1 \quad \downarrow b_0$	
32 ÷ 2 = 16	余0	$(b_7 \times 2^7 + b_6 \times 2^6 + b_5 \times 2^5 + b_4 \times 2^4 + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0)_{10}$	
16 ÷ 2 = 8	余0	$128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$	
8 ÷ 2 = 4	余0		
4 ÷ 2 = 2	余0		
2 ÷ 2 = 1	余0		
1 ÷ 2 = 0	余1		

#### 4.2.2.3. 分类编制方法

- IPv4分类编址方法不够灵活、容易造成大量IPv4地址资源浪费。

32比特的IPv4地址	
网络号	主机号
● 标志主机（或路由器）的接口所连接到的网络	● 标志主机（或路由器）的接口
● 同一个网络中，不同主机（或路由器）的接口的IPv4地址的网络号必须相同，表示它们属于同一个网络。	● 同一个网络中，不同主机（或路由器）的接口的IPv4地址的主机号必须各不相同，以便区分各主机（或路由器）的接口。

■ A类、B类和C类地址都是单播地址，只有单播地址可以分配给网络中的主机（或路由器）的各接口。

■ 主机号为“全0”的地址是网络地址，不能分配给主机（或路由器）的各接口。

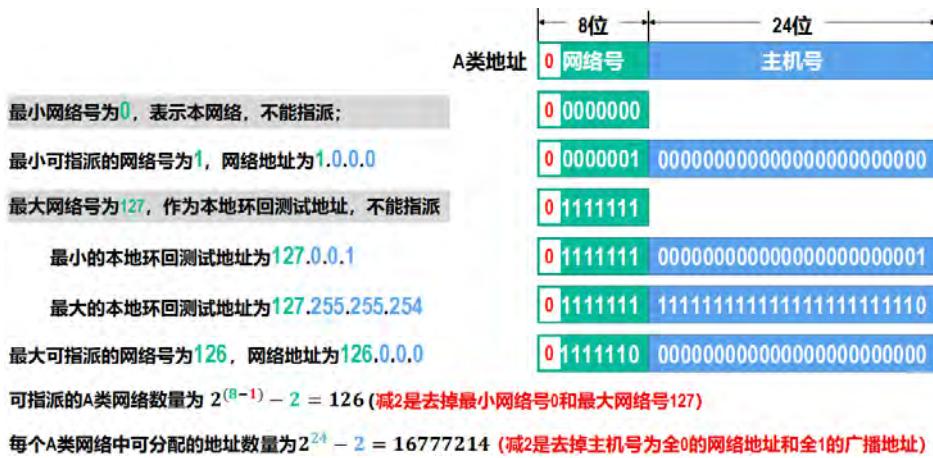
■ 主机号为“全1”的地址是广播地址，不能分配给主机（或路由器）的各接口。



网络类别	最小可指派网络号	最大可指派网络号	可指派网络数量	每个网络中最大可分配地址数量	不能指派的网络号	占总地址空间
A	1	126	$(2^{8-1} - 2)$	$16777214$ $(2^{24} - 2)$	0和127	$50\%$ $(2^{32-1}/2^{32})$
B	128.0	191.255	$(2^{16-2})$	$65534$ $(2^{16} - 2)$	无	$25\%$ $(2^{32-2}/2^{32})$
C	192.0.0	223.255.255	$(2^{24-3})$	$254$ $(2^8 - 2)$	无	$12.5\%$ $(2^{32-3}/2^{32})$

网络类别	作用	第一个地址	最后一个地址	地址数量	占总地址空间
D	多播地址	224.0.0.0	239.255.255.255	$268435456$ $(2^{28})$	$6.25\%$ $(2^{32-4}/2^{32})$
E	保留	240.0.0.0	255.255.255.255	$268435456$ $(2^{28})$	$6.25\%$ $(2^{32-4}/2^{32})$

- A类细节



- B类细节



- C类细节



- 一般不使用的Ipv4地址

网络号	主机号	IP地址	作为源地址	作为目的地址	表示的意思
0	0	0.0.0.0	可以	不可以	在本网络上的本主机 (例如, DHCP协议)
0	host-id	0.host-id	可以	不可以	在本网络上的某台主机host-id
全1	全1	255.255.255.255	不可以	可以	只在本网络上进行广播 (各路由器均不转发)
net-id	全1	A类: net-id.255.255.255 B类: net-id.255.255 C类: net-id.255	不可以	可以	对网络net-id上的所有主机进行广播
127	非全0或全1的任何数	127.0.0.1~127.255.255.254	可以	可以	用于本地软件环回测试

- 练习

**【练习】请填写以下两个表格的内容。**

**解析**

- (1) 根据地址左起第一个十进制数的值，可以判断出地址类别：  
小于127为A类；  
128~191为B类；  
192~223为C类。

- (2) 根据地址类别，可以找出地址中的网络号部分和主机号部分：  
A类：网络号为左起第一个字节  
B类：网络号为左起前两个字节  
C类：网络号为左起前三个字节
- (3) 以下三种情况的地址不能分配给主机或路由器的接口：  
A类网络号0和127；  
主机号为全0，这是网络地址；  
主机号为全1，这是广播地址。

IPv4地址	类别	是否可以分配给主机或路由器接口
0.1.2.3	A	不能分配，网络号0是保留的网络号
1.2.3.4	A	可以分配，网络号为1，主机号为2.3.4
126.255.255.255	A	不能分配，网络号为126，主机号为255.255.255，广播地址
127.0.0.1	A	不能分配，网络号为127，主机号为0.0.1，本地环回测试地址
128.0.255.255	B	不能分配，网络号为128.0，主机号为255.255，广播地址
166.16.18.255	B	可以分配，网络号为166.16，主机号为18.255
172.18.255.255	B	不能分配，网络号为172.18，主机号为255.255，广播地址
191.255.255.252	B	可以分配，网络号为191.255，主机号为255.252
192.0.0.255	C	不能分配，网络号为192.0.0，主机号为255，广播地址
196.2.3.8	C	可以分配，网络号为196.2.3，主机号为8
218.75.230.30	C	可以分配，网络号为218.75.230，主机号为30
223.255.255.252	C	可以分配，网络号为223.255.255，主机号为252

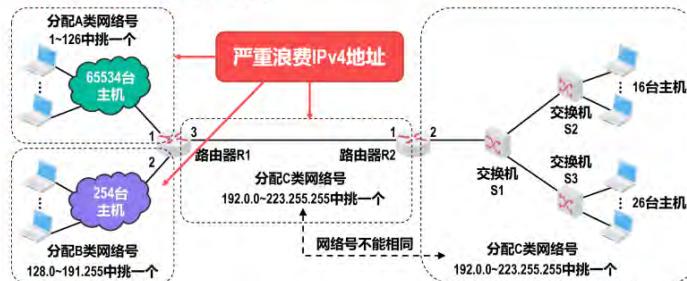
**【练习】请给出下图各网络的IPv4地址分配方案，要求尽量节约IP地址。**

**解析** 1. 找出图中有哪些网络；

2. 根据各网络中主机和路由器的接口总数量给各网络分配相应类别的网络号；

3. 给各网络中的各主机和路由器的各接口分配IP地址。

**同一个网络中，不同主机和路由器的各接口的IPv4地址的主机号必须各不相同，并且不能为“全0”（网络地址）和“全1”（广播地址）。**



#### 4.2.2.4. 划分子网的编址方法

- 在主机号中借用几位比特作为子网号。
- 子网掩码：表明分类IPv4地址的主机号部分被借用了几个比特作为子网号。
  - 由32比特构成。
  - 用左起多个连续的比特1对应IPv4地址中的网络号和子网号。
  - 之后的多个连续的比特0对应IPv4地址中的主机号。
- 将划分子网的IPv4地址与相应的子网掩码进行逐比特的逻辑与运算，就可得到该IPv4地址所在子网的网络地址。

32比特的划分子网的IPv4地址

网络号 子网号 主机号

32比特的子网掩码

1111...1111 0000000...00000000

**逻辑与运算**

IPv4地址所在子网的网络地址

网络号和子网号被保留

主机号被清零

- 默认子网掩码

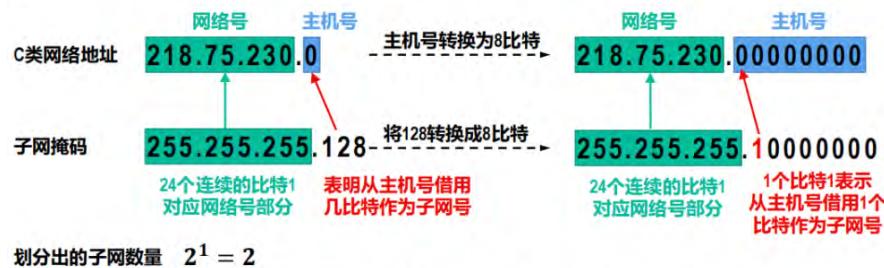
- 指在未划分子网的情况下使用的子网掩码
- ABC类主机号全置为0

点分十进制形式			
A类地址	8比特网络号	24比特主机号	
A类地址的默认子网掩码	11111111	00000000 00000000 00000000	255.0.0.0
B类地址	16比特网络号	16比特主机号	
B类地址的默认子网掩码	11111111 11111111	00000000 00000000	255.255.0.0
C类地址	24比特网络号	8比特主机号	
C类地址的默认子网掩码	11111111 11111111 11111111	00000000	255.255.255.0

- 子网划分细节——练习

【举例】已知某个网络的地址为218.75.230.0，使用子网掩码255.255.255.128对其进行子网划分，请给出划分细节。

解析



每个子网可分配的地址数量  $2^{(8-1)} - 2 = 126$  (减2是去掉主机号为全0的网络地址和全1的广播地址)

网络号	子网号	主机号	子网0的网络地址	218.75.230.0
218.75.230.0	0 0 0 0 0 0 0 0	子网0的最小地址 218.75.230.1	子网0的最大地址 218.75.230.126	126个
218.75.230.0	0 0 0 0 0 0 0 1	子网0的广播地址 218.75.230.127		128个
⋮				
218.75.230.0	0 1 1 1 1 1 1 0	子网1的最小地址 218.75.230.128	子网1的最大地址 218.75.230.254	126个
218.75.230.0	0 1 1 1 1 1 1 1	子网1的广播地址 218.75.230.255		128个
⋮				
218.75.230.1	0 0 0 0 0 0 0 0	子网0的最小地址 218.75.230.129	子网0的最大地址 218.75.230.254	126个
218.75.230.1	0 0 0 0 0 0 0 1	子网0的广播地址 218.75.230.255		128个
⋮				
218.75.230.1	1 1 1 1 1 1 1 0	子网1的最小地址 218.75.230.255	子网1的最大地址 218.75.230.255	126个
218.75.230.1	1 1 1 1 1 1 1 1	子网1的广播地址 218.75.230.255		128个

【练习】已知某个网络的地址为145.13.0.0，使用子网掩码255.255.192.0对其进行子网划分，请给出划分细节。

解析

- 根据所给网络地址可知其为B类网络地址，网络号和主机号各占2字节；
- 根据所给子网掩码可知，从2字节主机号中借用2比特作为子网号。

B类网145.13.0.0 包含的全部地址 (共256个)

划分出的子网数量  $2^2 = 4$

每个子网可分配的地址数量  $2^{(16-2)} - 2$

网络号	子网号	主机号	子网0的网络地址	145.13.0.0
145.13.0.0	0 0 0 0 0 0 0 0	子网0的最小地址 145.13.0.1	子网0的最大地址 145.13.63.254	可分配最小地址 145.13.0.1
145.13.0.0	0 0 0 0 0 0 0 1	子网0的广播地址 145.13.63.255		可分配最大地址 145.13.63.254
⋮				
145.13.0.0	0 1 0 0 0 0 0 0	子网1的网络地址 145.13.64.0	子网1的广播地址 145.13.64.1	可分配最小地址 145.13.64.1
145.13.0.0	0 1 0 0 0 0 0 1	子网1的广播地址 145.13.64.1		可分配最大地址 145.13.64.1
⋮				
145.13.0.0	1 0 1 1 1 1 1 0	子网2的网络地址 145.13.127.254	子网2的广播地址 145.13.127.255	可分配最大地址 145.13.127.254
145.13.0.0	1 0 1 1 1 1 1 1	子网2的广播地址 145.13.127.255		可分配最小地址 145.13.127.255
⋮				
145.13.0.0	1 1 0 0 0 0 0 0	子网3的网络地址 145.13.192.0	子网3的广播地址 145.13.192.1	可分配最大地址 145.13.192.0
145.13.0.0	1 1 0 0 0 0 0 1	子网3的广播地址 145.13.192.1		可分配最小地址 145.13.192.1
⋮				
145.13.0.0	1 1 1 1 1 1 1 0	子网4的网络地址 145.13.255.254	子网4的广播地址 145.13.255.255	可分配最大地址 145.13.255.254
145.13.0.0	1 1 1 1 1 1 1 1	子网4的广播地址 145.13.255.255		可分配最小地址 145.13.255.255

#### 4.2.2.5. 无分类编址方法

- 无分类编址方法使用的地址掩码与划分子网使用的子网掩码类似，由32比特构成。

- 用左起多个连续的比特1对应IPv4地址中的网络前缀。
- 之后的多个连续的比特0对应IPv4地址中的主机号。

- 用斜线记法标记网络前缀比特数。

- /30 应用在分配只有两个路由器接口的点对点链路。【2019 47(2)】

128.14.35.7 / 20
 
 网络前缀: 20比特  
 主机号: 12比特 (32-20)

- 无分类域间路由选择 (Classless Inter-Domain Routing, CIDR)
- CIDR消除了传统A类、B类和C类地址以及划分子网的概念。
- CIDR可以更加有效地分配IPv4地址资源，并且可以在IPv6使用之前允许因特网的规模继续增长。
- CIDR地址块：由将网络前缀都相同的、连续的多个无分类IPv4地址组成
  - 地址块中的最小地址
  - 地址块中的最大地址
  - 地址块中的地址数
  - 地址块中聚合某类网络（A类、B类、C类）的数量
  - 地址掩码
- CIDR细节练习

**【例1】**给定的无分类编址的IPv4地址为128.14.35.7/20，请给出该地址所在CIDR地址块的全部细节。

解析

	将左起第3、4个十进制数 转换成二进制形式	← 20比特网络前缀 → ← 12比特主机号 →
128.14.35.7/20		128.14.00100011.00000111
最小地址 128.14.32.0		128.14.00100000.00000000
最大地址 128.14.47.255		128.14.00101111.11111111

地址数量  $2^{32-20}$

聚合C类网的数量  $2^{32-20} \div 2^8$

	地址掩码 255.255.240.0	← 对应20比特网络前缀 → ← 对应12比特主机号 →
		1111111.1111111.11110000.00000000

**【练习】**给定的无分类编址的IPv4地址为206.0.64.8/18，请给出该地址所在CIDR地址块的全部细节。

解析

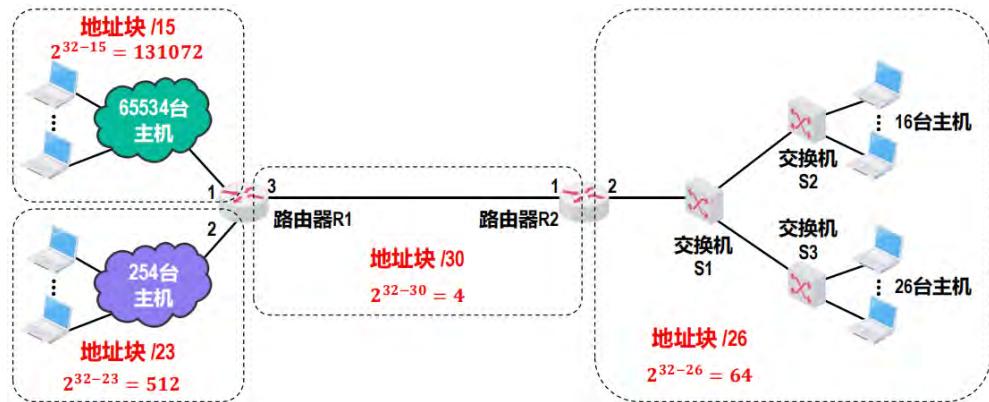
	将左起第3、4个十进制数 转换成二进制形式	← 18比特网络前缀 → ← 14比特主机号 →
206.0.64.8/18		206.0.0100000.00001000
最小地址 206.0.64.0		206.0.0100000.00000000
最大地址 206.0.127.255		206.0.011111.11111111

地址数量  $2^{32-18}$

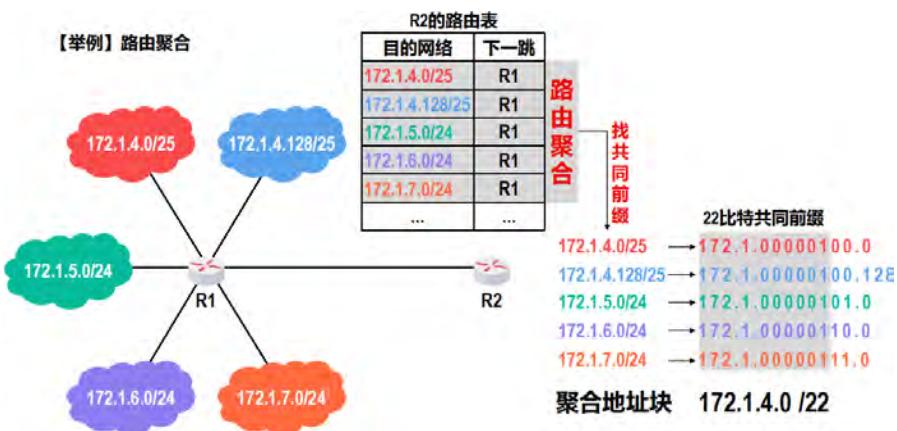
聚合C类网的数量  $2^{32-18} \div 2^8$

	地址掩码 255.255.192.0	← 对应18比特网络前缀 → ← 对应14比特主机号 →
		1111111.1111111.11000000.00000000

- 使用无分类编址方法，可以根据客户的需要分配适当大小的CIDR地址块，因此可以更加有效地分配IPv4的地址空间。



- 使用无分类编址方法的另一个好处是路由聚合（也称为构造超网）。
  - 网络前缀越长，地址块越小，路由越具体。
  - 若路由器查表转发分组时发现有多条路由条目匹配，则选择网络前缀最长的那条路由条目，这称为最长前缀匹配，因为这样的路由更具体。



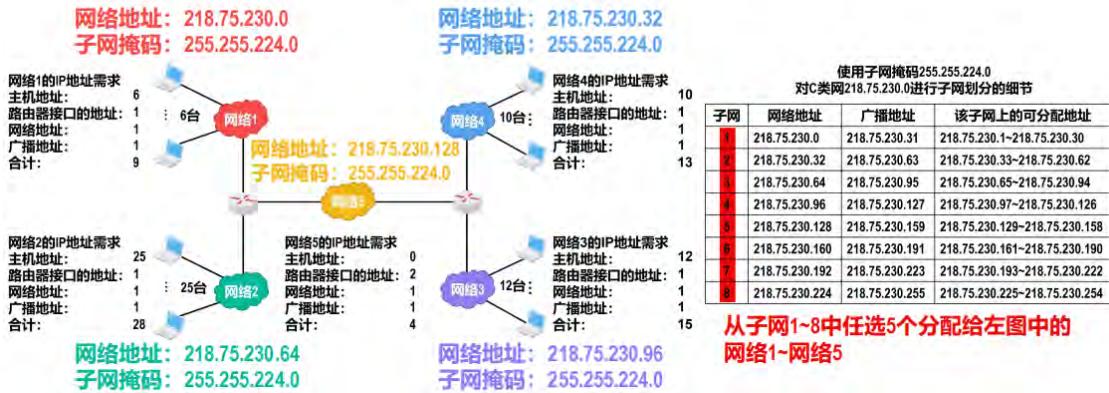
#### 4.2.3. IPv4地址的应用规划

■ IPv4地址的应用规划是指将给定的IPv4地址块（或分类网络）划分成若干个更小的地址块（或子网），并将这些地址块（或子网）分配给互联网中的不同网络，进而可以给各网络中的主机和路由器的接口分配IPv4地址。

定长的子网掩码 (Fixed Length Subnet Mask, FLSM)	变长的子网掩码 (Variable Length Subnet Mask, VLSM)
<ul style="list-style-type: none"> <li>所划分出的每一个子网都使用同一个子网掩码。</li> <li>每个子网所分配的IP地址数量相同，容易造成地址资源的浪费。</li> </ul>	<ul style="list-style-type: none"> <li>所划分出的每一个子网可以使用不同的子网掩码。</li> <li>每个子网所分配的IP地址数量可以不同，尽可能减少对地址资源的浪费。</li> </ul>

##### 4.2.3.1. 定长子网掩码

**【举例】假设申请到的C类网络为218.75.230.0，使用定长的子网掩码给下图所示的小型互联网中的各设备分配IPv4地址。**



#### 4.2.3.2. 变长子网掩码

**【举例】假设申请到的地址块为218.75.230.0/24，使用变长的子网掩码给下图所示的小型互联网中的各设备分配IPv4地址。**

**应用需求：**

从地址块218.75.230.0/24中取出5个地址块：  
/30, /28, /28, /28, /27，按需分配给图中的  
的5个网络。



218.75.230.0/24地址块所包含的全部地址如下：

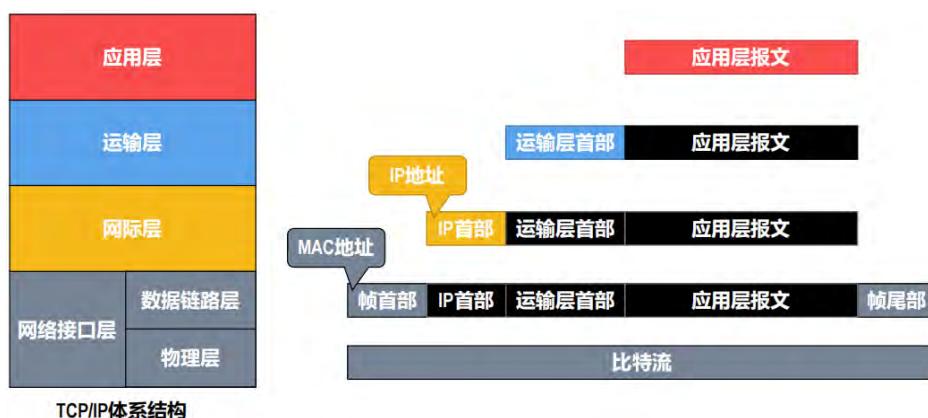
218.75.230.0	网络2的网络地址
218.75.230.31	网络2的可分配地址
218.75.230.32	网络1的网络地址
218.75.230.47	网络1的可分配地址
218.75.230.48	网络1的广播地址
218.75.230.63	网络3的网络地址
218.75.230.64	网络3的可分配地址
218.75.230.79	网络3的广播地址
218.75.230.80	网络4的网络地址
218.75.230.83	网络4的可分配地址
218.75.230.84	网络4的广播地址
218.75.230.255	剩余待分配

- 在地址块中选取子块的原则

- 每个子块的起点位置不能随便选取，只能选取主机号部分是块大小整数倍的地址作为起点。
- 建议先为大的子块选取。

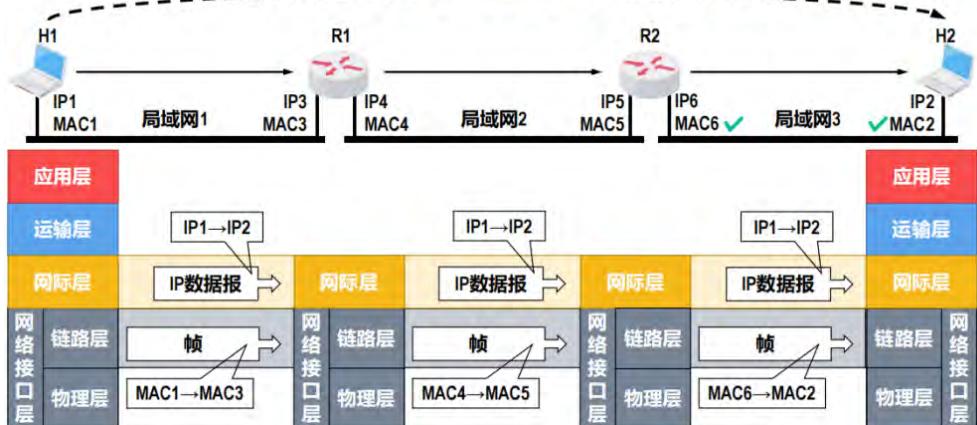
#### 4.2.4. IPv4地址与MAC地址

##### 4.2.4.1. IPv4地址与MAC地址的封装位置



#### 4.2.4.2. 数据包传送过程中IPv4地址与MAC地址的变化情况

- 在数据包的传送过程中，数据包的源IP地址和目的IP地址保持不变；
- 在数据包的传送过程中，数据包的源MAC地址和目的MAC地址逐链路（或逐网络）改变。



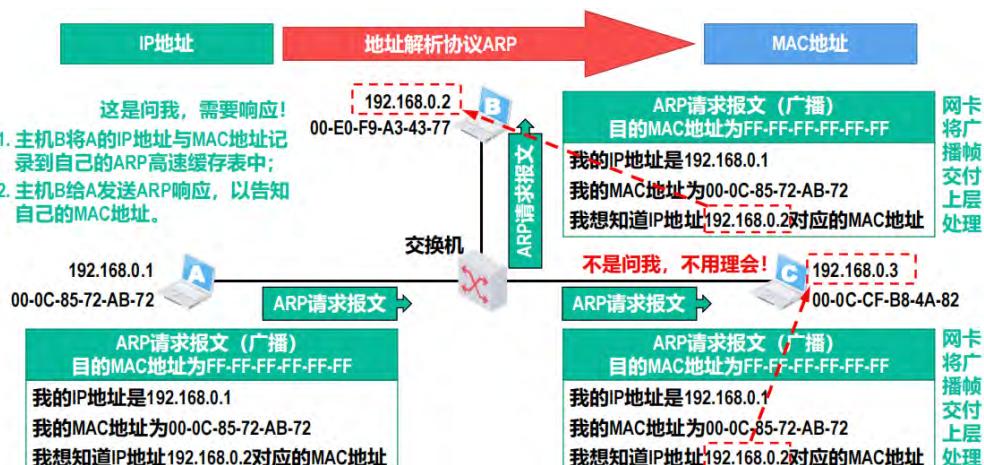
#### 4.2.4.3. IPv4地址与MAC地址的关系

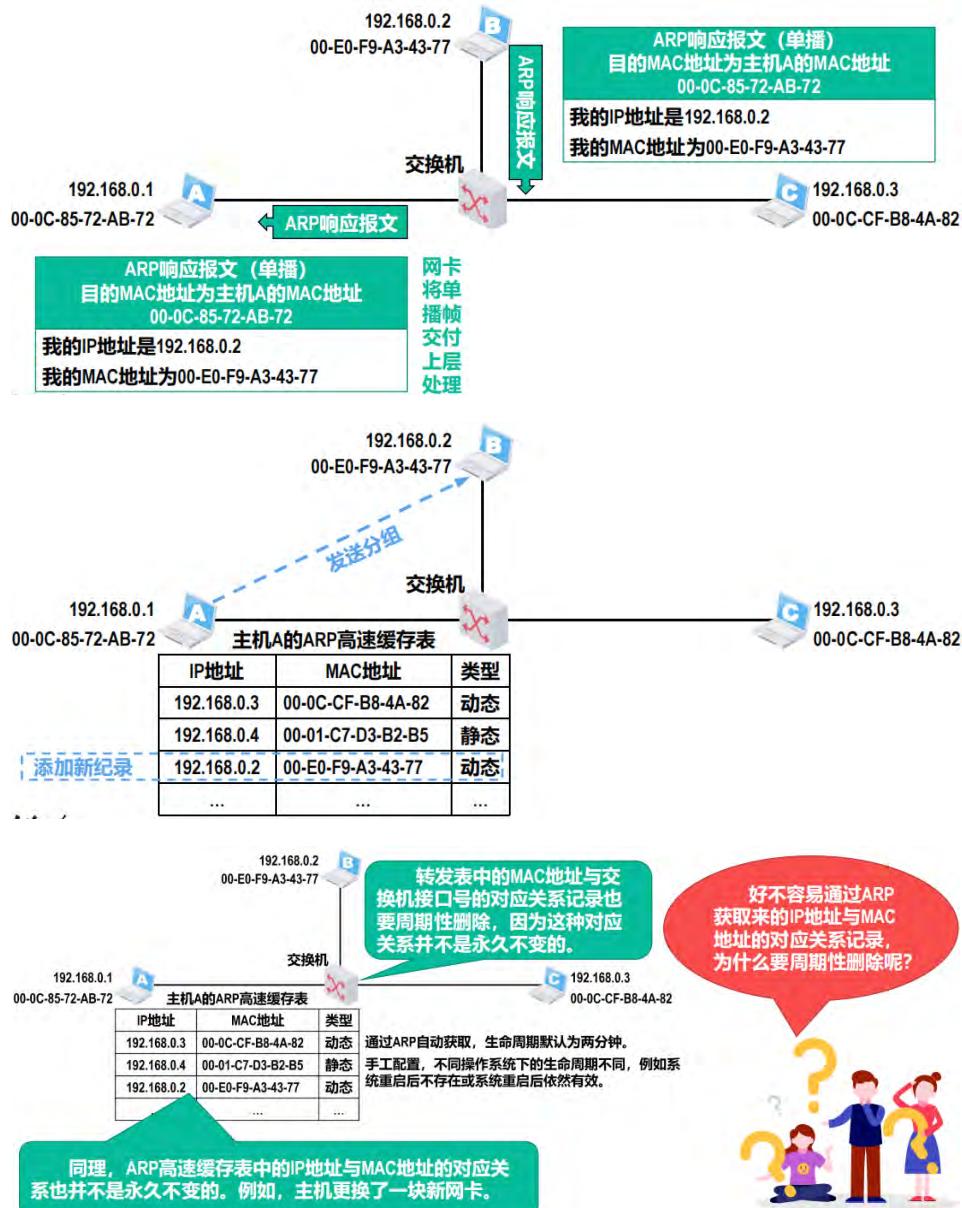
- 如果仅使用MAC地址进行通信，则会出现以下主要问题：
  - 因特网中的每台路由器的路由表中就必须记录因特网上所有主机和路由器各接口的MAC地址。
  - 手工给各路由器配置路由表几乎是不可能完成的任务，即使使用路由协议让路由器通过相互交换路由信息来自动构建路由表，也会因为路由信息需要包含海量的MAC地址信息而严重占用通信资源。
  - 包含海量MAC地址的路由信息需要路由器具备极大的存储空间，并且会给分组的查表转发带来非常大的时延。
- 因特网的网际层使用IP地址进行寻址，就可使因特网中各路由器的路由表中的路由记录的数量大大减少，因为只需记录部分网络的网络地址，而不是记录每个网络中各通信设备的各接口的MAC地址。
  - 路由器收到IP数据报后，根据其首部中的目的IP地址的网络号部分，基于自己的路由表进行查表转发。

查表转发的结果可以指明IP数据报的下一跳路由器的IP地址，但无法指明该IP地址所对应的MAC地址。因此，在数据链路层封装该IP数据报成为帧时，帧首部中的目的MAC地址字段就无法填写，该问题需要使用网际层中的地址解析协议ARP来解决。

#### 4.2.5. 地址解析协议ARP

- Address Resolution Protocol





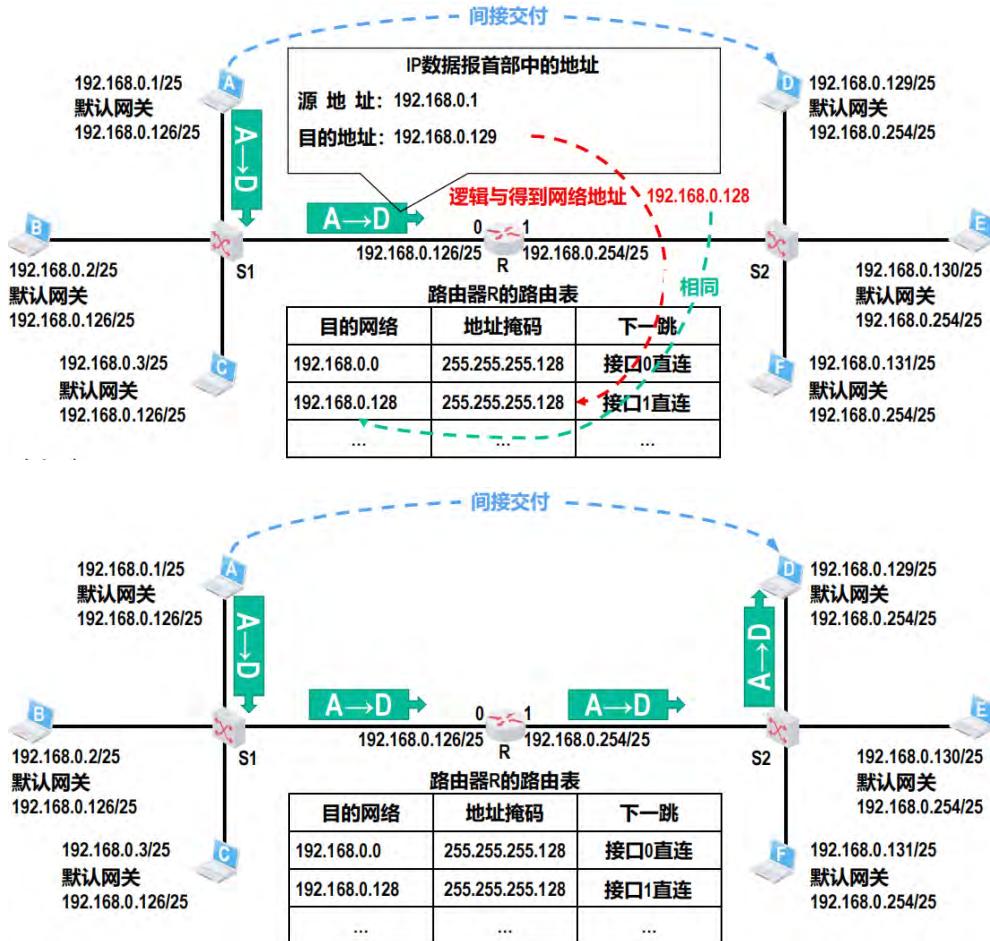
#### ■ ARP协议的相关注意事项:

- 由于ARP协议的主要用途是从网际层使用的IP地址解析出在数据链路层使用的MAC地址。因此，有的教材将ARP协议划归在网际层，而有的教材将ARP协议划归在数据链路层。这两种做法都是可以的。
- 除了本节课介绍的ARP请求报文和响应报文，**ARP协议还有其他类型的报文**，例如用于检查IP地址冲突的“无故ARP”（Gratuitous ARP）。
- 由于ARP协议很早就制定出来了（1982年11月），当时并没有考虑网络安全问题。因此，ARP协议没有安全验证机制，存在**ARP欺骗和攻击**等问题。

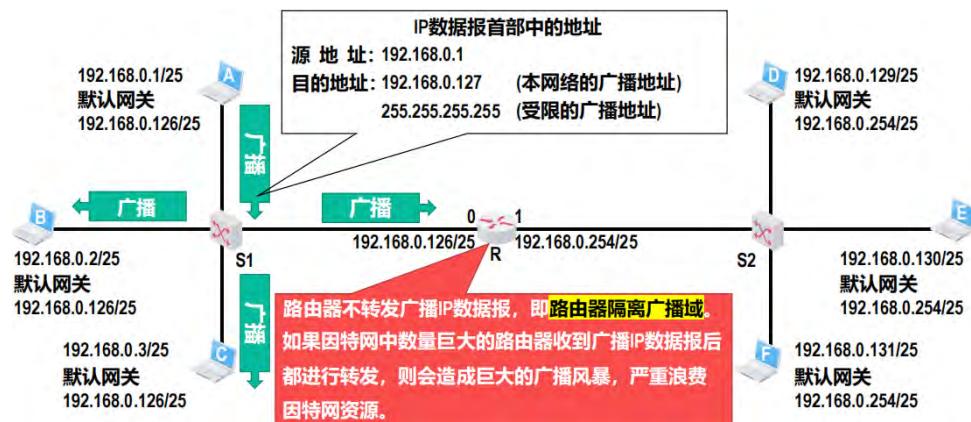


#### 4.2.6. IP数据报的发送和转发流程

- 单播
  - 默认网关：路由器的IP地址
  - 将IP数据报中的目的地址与路由表中的地址掩码进行逻辑与运算找到下一跳



- 广播



#### 4.2.6.1. 主机发送IP数据报

- 判断目的主机是否与自己在同一个网络
  - 若在同一个网络，则属于直接交付
  - 若不在同一个网络，则属于间接交付。发送给主机所在网络的默认网关（路由器），由默认网关帮忙转发。

#### 4.2.6.2. 路由器转发IP数据报

- ① 检查收到的P数据报是否正确（生存时间是否结束；首部是否误码）
  - 若出错，则丢弃该P数据报并给发送该P数据报的源主机发送差错报告；
  - 若正确，则进行查表转发。
- ② 基于P数据报首部中的目的P地址在路由表中进行查找

- 若找到匹配的路由条目，则按该路由条目的指示进行转发；
- 若找不到匹配的路由条目，则丢弃该P数据报，并向发送该P数据报的源主机发送差错报告。
- ③ 查找步骤

- 先根据路由表中目的地址的网络前缀，得出各目的地址的子网掩码。
- 将子网掩码与所收到的IP数据包中的目的地址相与。
- 若相与结果和路由表中对应的目的地址匹配，则选取网络前缀最长的那个目的地址进行转发。
- 若相与结果不匹配任一目的地址，则选择默认路由0.0.0.0/0进行转发。

#### 4.2.7. IPv4数据报的首部格式

- IPv4数据报的首部格式及其内容是实现IPv4协议各种功能的基础。
- 在TCP/IP标准中，各种数据格式常常以32比特（即4字节）为单位来描述。



- 固定部分是指每个IPv4数据报都必须要包含的部分。
- 某些IPv4数据报的首部，除了包含20字节的固定部分，还包含一些可选的字段来增加IPv4数据报的功能。
- IPv4数据报首部中的各字段或某些字段的组合，用来表达IPv4协议的相关功能。

##### 4.2.7.1. 版本

- 长度为4个比特，用来表示IP协议的版本。
- 通信双方使用的IP协议的版本必须一致。目前广泛使用的IP协议的版本号为4（即IPv4）。

##### 4.2.7.2. 首部长度

- 长度为4个比特，该字段的取值以4字节为单位，用来表示IPv4数据报的首部长度。
  - 最小取值为二进制的0101，即十进制的5，再乘以4字节单位，表示IPv4数据报首部只有20字节固定部分。
  - 最大取值为二进制的1111，即十进制的15，再乘以4字节单位，表示IPv4数据报首部包含20字节固定部分和最大40字节可变部分。

##### 4.2.7.3. 可选字段

- 长度从1字节到40字节不等，用来支持排错、测量以及安全措施等功能。
  - 虽然可选字段增加了IPv4数据报的功能，但这同时也使得IPv4数据报的首部长度成为可变的，这就增加了因特网中每一个路由器处理IPv4数据报的开销。
  - 实际上，可选字段很少被使用。

##### 4.2.7.4. 填充

- 用来确保IPv4数据报的首部长度是4字节的整数倍，使用全0进行填充。

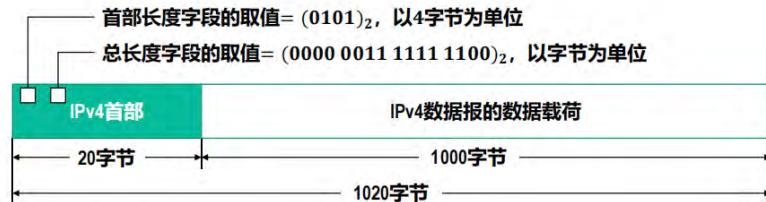
#### 4.2.7.5. 区分服务

- 长度为8个比特，用来获得更好的服务。

#### 4.2.7.6. 总长度

- 长度为16个比特，该字段的取值以字节为单位，用来表示IPv4数据报的长度（首部长度+数据载荷长度）。
- 最大取值为二进制的16个比特1，即十进制的65535（很少传输这么长的IPv4数据报）。
- 计算举例

【举例】IPv4数据报首部中的首部长度字段和总长度字段。



$$\text{首部长度} = (0101)_2 \times 4 = 5 \times 4 = 20 \text{ (字节)}$$

$$\text{总 长 度} = (0000\ 0011\ 1111\ 1100)_2 = 1020 \text{ (字节)}$$

$$\text{数据载荷长度} = \text{总长度} - \text{首部长度} = 1020 - 20 = 1000 \text{ (字节)}$$

#### 4.2.7.7. IP数据包分片



##### 4.2.7.7.1. 标识

- 长度为16个比特，属于同一个IPv4数据报的各分片数据报应该具有相同的标识。
- IP软件会维持一个计数器，每产生一个IPv4数据报，计数器值就加1，并将此值赋给标识字段。

##### 4.2.7.7.2. 标志

- 最低位 (More Fragment, MF)
  - MF=1表示本分片后面还有分片
  - MF=0表示本分片后面没有分片
- 中间位 (Don't Fragment, DF)
  - DF=1表示不允许分片
  - DF=0表示允许分片
- 最高位为保留位，必须设置为0

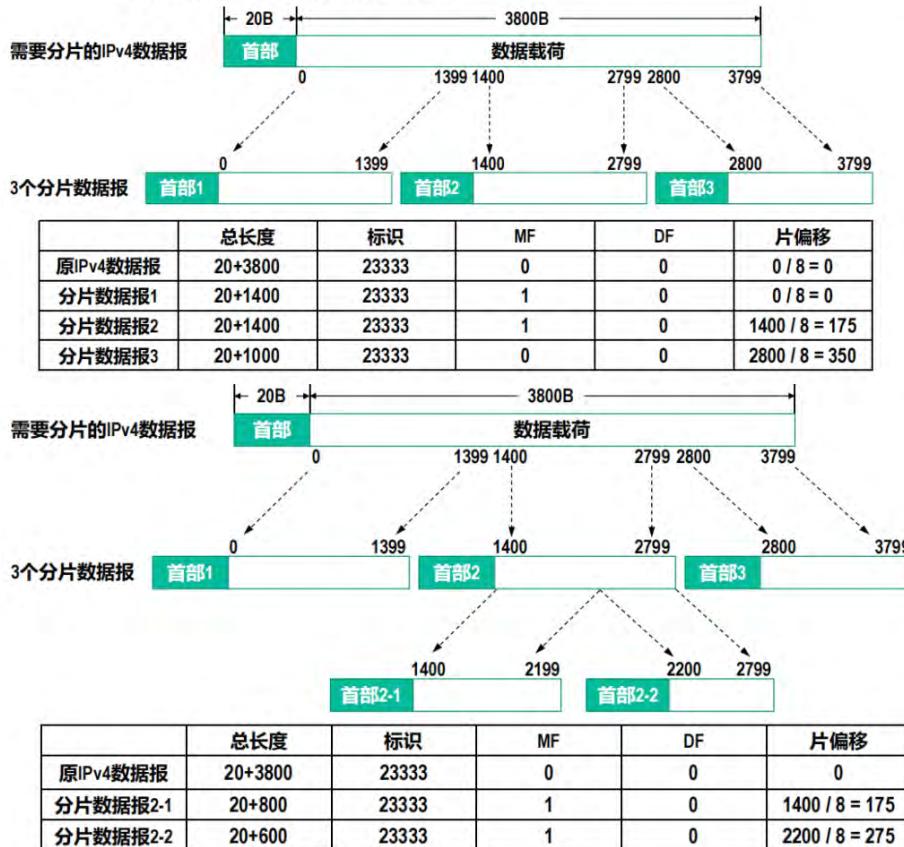
##### 4.2.7.7.3. 片偏移

- 长度为13个比特，该字段的取值以8字节为单位。
- 作用：指出分片IPv4数据报的数据载荷偏移其在原IPv4数据报的位置有多远。
- 计算
  - 分片第一个字节的序号/8

- 片偏移必须是整数，并且以8字节为单位，若算出来小数，则需要调整分片长度为8的倍数。【2021 36】

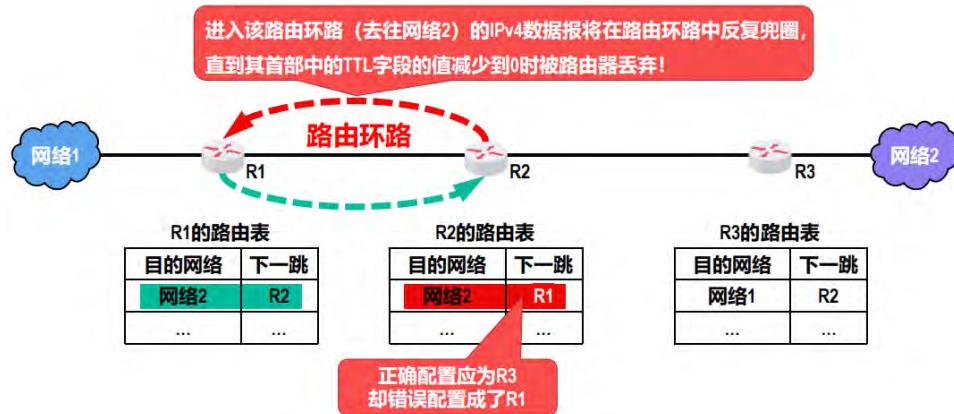
### 计算举例

【举例】某个IPv4数据报总长度为3820字节，采用20字节固定首部，根据数据链路层要求，需要将该IPv4数据报分片为长度不超过1420字节的数据报片。



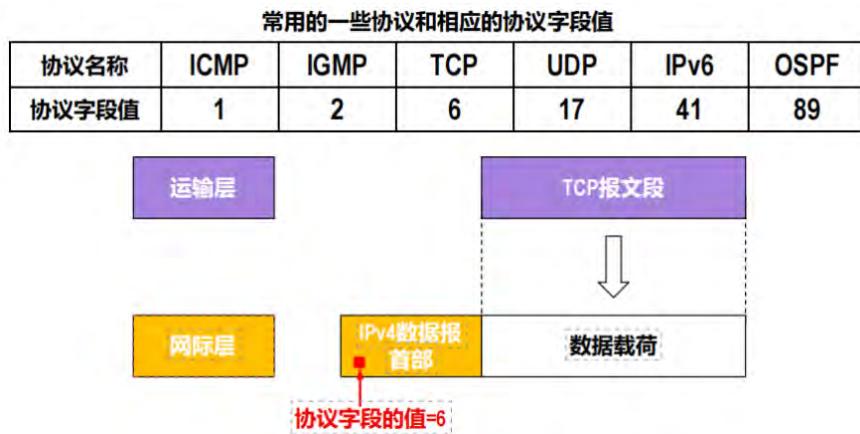
#### 4.2.7.8. 生存时间 (Time To Live, TTL)

- 长度为8个比特，最大取值为二进制的11111111，即十进制的255。
- 该字段的取值最初以秒为单位。
  - 因此，IPv4数据报的最大生存时间为255秒。
  - 路由器转发IPv4数据报时，将其首部中该字段的值减去该数据报在路由器上所耗费的时间，若结果不为0就转发，否则就丢弃。
- 生存时间字段后来改为以“跳数”为单位
  - 路由器收到待转发的IPv4数据报时，将其首部中的该字段的值减1，若结果不为0就转发，否则就丢弃。
- 作用：防止被错误路由的IPv4数据报无限制地在因特网中兜圈。



#### 4.2.7.9. 协议

- 长度为8个比特
- 作用：指明IPv4数据报的数据载荷是何种协议数据单元PDU。



#### 4.2.7.10. 首部检验和

- 长度为16个比特
- 作用：用于检测IPv4数据报在传输过程中其首部是否出现了差错。
- IPv4数据报每经过一个路由器，其首部中的某些字段的值（例如生存时间TTL、标志以及片偏移等）都可能发生变化，因此路由器都要重新计算一下首部检验和。

##### 4.2.7.10.1. 首部检验和的计算方法



- 由于网际层并不向其高层提供可靠传输的服务，并且计算首部检验和是一项耗时的操作，因此在IPv6中，路由器不再计算首部检验和，从而更快转发IP数据报。

#### 4.2.7.11. 源IP地址

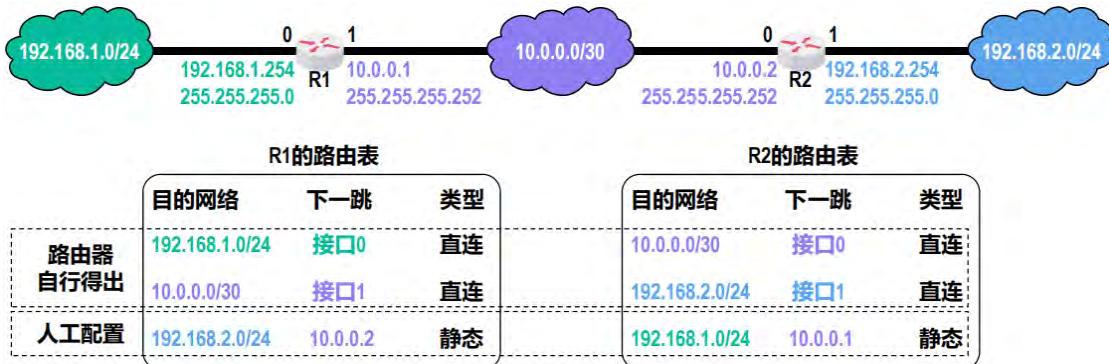
- 长度为32个比特
- 作用：用来填写发送IPv4数据报的源主机的IPv4地址。

#### 4.2.7.12. 目的IP地址

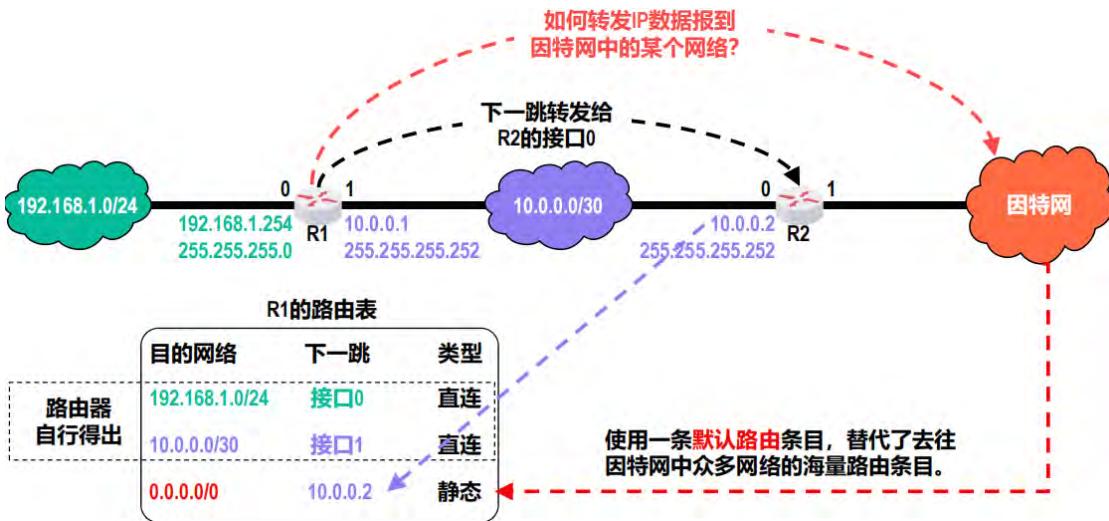
- 长度为32个比特
- 用来填写接收IPv4数据报的目的主机的IPv4地址。

### 4.3. 静态路由配置

#### 4.3.1. 人工配置静态路由



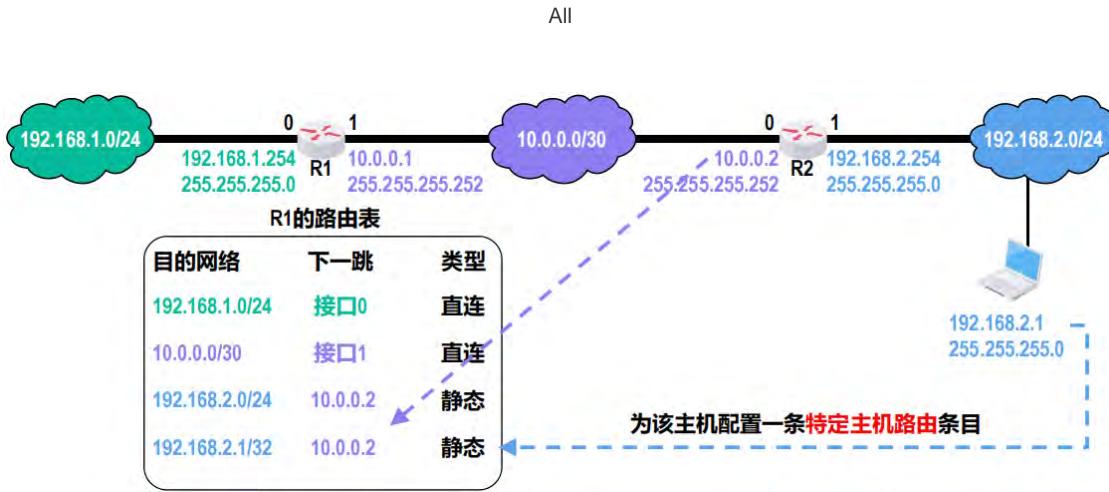
#### 4.3.2. 默认路由0.0.0.0/0



- 默认路由条目中的目的网络0.0.0.0/0，其中0.0.0.0表示任意网络，而网络前缀“/0”（相应的地址掩码为0.0.0.0）是最短的网络前缀。
- 路由器在查找转发表转发IP数据报时，遵循“最长前缀匹配”的原则，因此默认路由条目的匹配优先级最低。
- 默认路由可以减少路由表所占用的存储空间和搜索路由表所耗费的时间。

#### 4.3.3. 特定主机路由/32

- 特定主机路由条目的匹配优先级最高。



- 特定主机路由条目中的目的网络192.168.2.1/32，其中192.168.2.1是特定主机的IP地址，而网络前缀“/32”（相应地址掩码为255.255.255.255）是最长的网络前缀。
- 图中，在查表转发去往192.168.2.1这台特定主机的IP数据报时，192.168.2.0/24和192.168.2.1/32两个路由条目都可以匹配，遵循“最长前缀匹配”的原则，按照匹配优先级最高的特定主机路由条目进行转发。

#### 4.3.4. 需注意的问题

- 进行静态路由配置需要认真考虑和谨慎操作，否则可能出现以下问题：
  - 路由条目配置错误，甚至导致出现路环路。
  - 聚合路由条目时可能引入不存在的网络。

### 4.4. 因特网的路由选择协议

#### 4.4.1. 路由选择分类

##### 4.4.1.1. 静态路由选择

- 采用人工配置的方式给路由器添加网络路由、默认路由和特定主机路由等路由条目。
- 静态路由选择简单、开销小，但不能及时适应网络状态（流量、拓扑等）的变化。
- 静态路由选择一般只在小规模网络中采用。

##### 4.4.1.2. 动态路由选择

- 路由器通过路由选择协议自动获取路由信息。
- 动态路由选择比较复杂、开销比较大，但能较好地适应网络状态的变化。
- 动态路由选择适用于大规模网络。

#### 4.4.2. 因特网采用分层次的路由选择协议

##### 4.4.2.1. 特点

**自适应**

因特网采用**动态路由选择**，能较好地适应网络状态的变化。

**分布式**

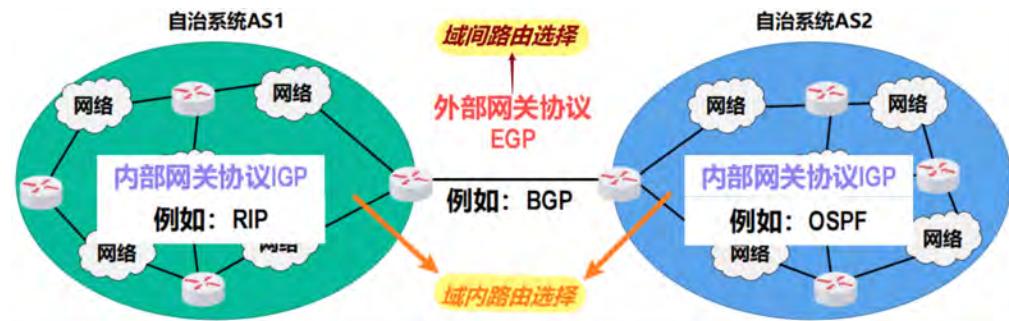
因特网中的**各路由器**通过相互间的信息交互，**共同完成路由信息的获取和更新**。

**分层次**

将整个因特网划分为许多较小的**自治系统 (Autonomous System, AS)**。

在自治系统内部和外部采用不同类别的路由选择协议，分别进行路由选择。

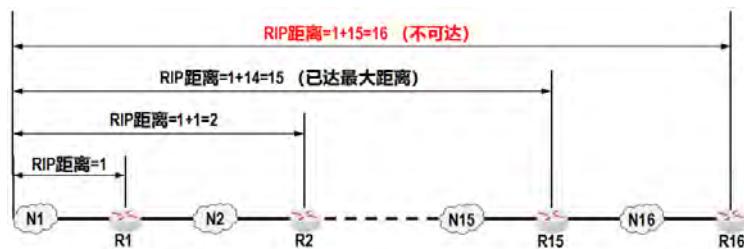
##### 4.4.2.2. 举例



#### 4.4.3. 路由信息协议RIP（封装在UDP）

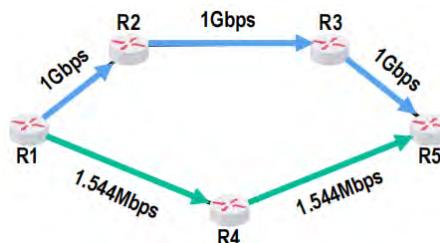
##### 4.4.3.1. 基本概念

- 路由信息协议 (Routing Information Protocol, RIP) 是内部网关协议中最先得到广泛使用的协议之一，其相关标准文档为[RFC 1058]。
- 距离向量 (Distance-Vector, D-V) : RIP要求自治系统AS内的每一个路由器，都要维护从它自己到AS内其他每一个网络的距离记录。这组距离称为距离向量。
- 跳数 (Hop Count) : 度量 (Metric) 来衡量到达目的网络的距离
  - 将路由器到直连网络的距离定义为1。
  - 将路由器到非直连网络的距离定义为所经过的路由器数加1。
  - 允许一条路径最多只能包含15个路由器，距离等于16时相当于不可达。因此RIP只适用于小型互联网。



- 好的路由：距离短，即所通过路由器数量最少的路由。

– 如下图，RIP认为R1到R5的好路由是：R1→R4→R5



- 等价负载均衡：当到达同一目的网络有多条RIP距离相等的路由时，可以进行等价负载均衡，也就是将通信量均匀地分布到多条等价的路径上。

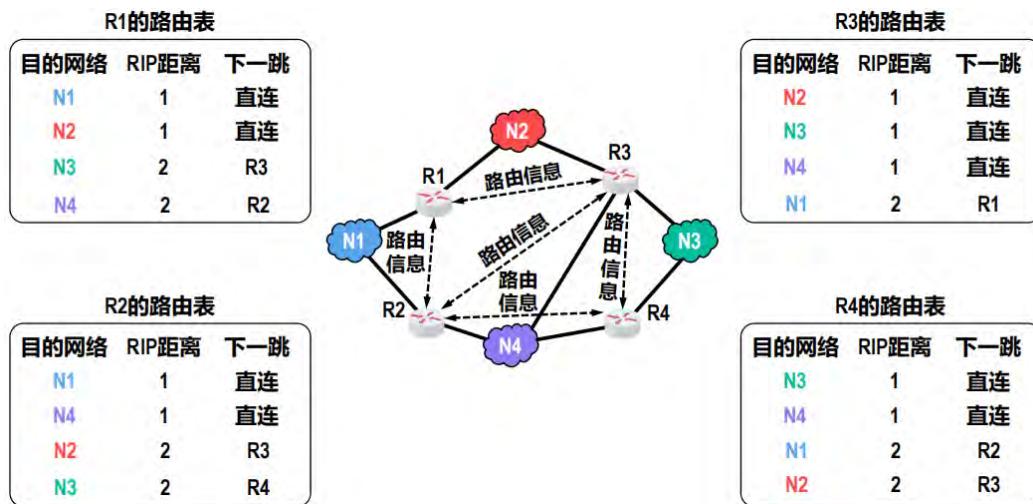


- 特点

– 和谁交换信息

- 仅和相邻路由器交换信息。
- 交换什么信息
  - 路由器自己的路由表。
  - 即本路由器到所在自治系统AS中各网络的最短RIP距离，以及到各网络应经过的下一跳路由器。
- 何时交换信息
  - 周期性交换（例如，每个约30秒）。
  - 为了加快RIP的收敛速度，当网络拓扑发生变化时，路由器要及时向相邻路由器通告拓扑变化后的路由信息，这称为触发更新。

#### 4.4.3.2. 基本工作过程



- 路由器刚开始工作时，只知道自己到直连网络的RIP距离为1。
- 每个路由器仅和相邻路由器周期性地交换并更新路由信息。
- 收敛：若干次交换和更新后，每个路由器都知道到达本自治系统AS内各网络的最短距离和下一跳路由器。

#### 4.4.3.3. 距离向量算法



- D不需要关心C中下一跳的内容，收到后将C中的下一条均改为C。
- 从D修改后路由器C的路由表的目的网络往下看
  - N2：到达目的网络，相同的下一跳，表示信息为最新，要更新D中N2的RIP距离
  - N3：发现新的网络，向D的路由表中添加。
  - N6：到达目的网络，不同的下一条（F、C），新路由RIP更短（有优势），要更新D中N6的RIP距离
  - N8：到达目的网络，不同的下一条，RIP距离相等，可以等价负载均衡，向D的路由表中添加。

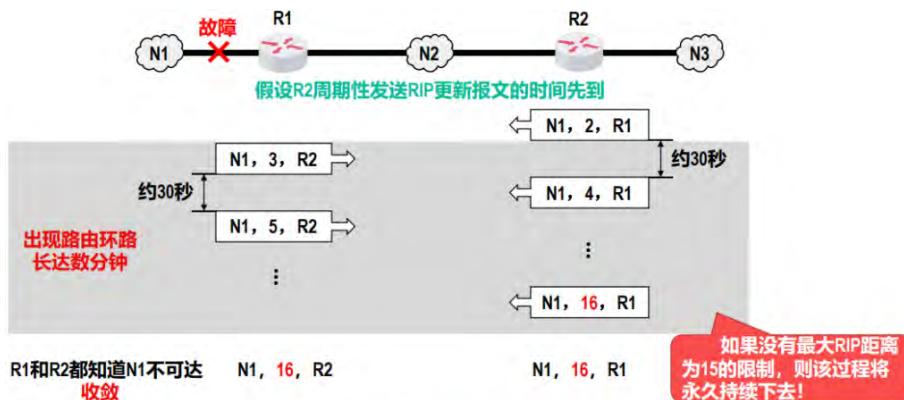
- N9: 到达目的网络, 不同的下一条 (F、C), 新路由RIP更长 (处于劣势), 不更新更新D中N9的RIP距离
- 更新后D的路由表如下

目的网络	RIP距离	下一跳
N1	7	A
N2	5	C
N3	9	C
N6	5	C
N8	4	C
N8	4	E
N9	4	F

- 除了上述RIP路由条目更新规则, 在RIP的距离向量算法中还包含以下一些时间参数:

- 路由器每隔大约30秒向其所有相邻路由器发送路由更新报文。
- 若180秒 (默认) 没有收到某条路由条目的更新报文, 则把该路由条目标记为无效 (即把RIP距离设置为16, 表示不可达)。
- 若再过一段时间 (如120秒), 还没有收到该路由条目的更新报文, 则将该路由条目从路由表中删除。

#### 4.4.3.4. 存在的问题——“坏消息传播得慢”



■ “坏消息传播得慢”的问题又被称为路由环路或RIP距离无穷计数问题。这是距离向量算法的一个固有问题。可以采取以下多种措施减少出现该问题的概率或减小该问题带来的危害:

- 限制最大RIP距离为15 (16表示不可达)。
- 当路由表发生变化时就立即发送路由更新报文 (即“触发更新”), 而不仅是周期性发送。
- 让路由器记录收到某个特定路由信息的接口, 而不让同一路由信息再通过此接口向反方向传送 (即“水平分割”)。

请同学们注意:

使用上述措施仍无法彻底解决问题。因为在距离向量算法中, 每个路由器都缺少到目的网络整个路径的完整信息, 无法判断所选的路由是否出现了环路。

#### 4.4.3.5. 版本和相关报文的封装

■ 现在较新的RIP版本是1998年11月公布的RIP2[RFC 2453], 已经成为因特网标准协议。与RIP1相比, RIP2可以支持变长子网掩码和CIDR。另外, RIP2还提供简单的鉴别过程并支持多播。

■ RIP相关报文使用运输层的用户数据报协议UDP进行封装, 使用的UDP端口号为520。

- 从RIP报文封装的角度看, RIP属于TCP/IP体系结构的应用层。
- 但RIP的核心功能是路由选择, 这属于TCP/IP体系结构的网际层。

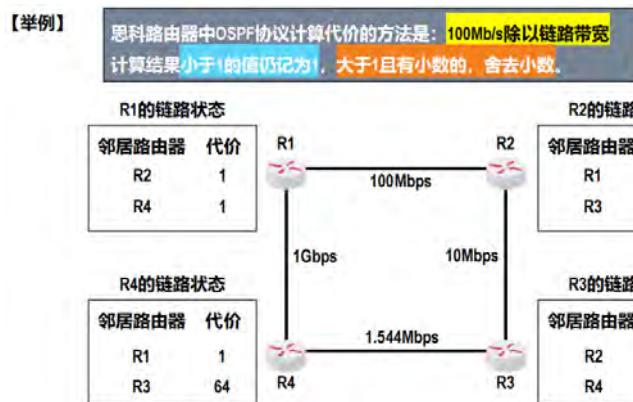
#### 4.4.3.6. 优缺点

优点	缺点
<ul style="list-style-type: none"> <li>● 实现简单，路由器开销小。</li> <li>● 如果一个路由器发现了RIP距离更短的路由，那么这种更新信息就传播得很快，即“<b>好消息传播得快</b>”。</li> </ul>	<ul style="list-style-type: none"> <li>● RIP限制了<b>最大RIP距离为15</b>，这就限制了使用RIP的自治系统AS的规模。</li> <li>● 相邻路由器之间交换的路由信息是路由器中的<b>完整路由表</b>，因而随着网络规模的扩大，开销也随之增大。</li> <li>● “<b>坏消息传播得慢</b>”，使更新过程的收敛时间过长。因此，对于规模较大的自治系统AS，应当使用OSPF协议。</li> </ul>

#### 4.4.4. 开放最短路径优先OSPF（封装在IP）

##### 4.4.4.1. 基本概念

- 开放最短路径优先 (Open Shortest Path First, OSPF) 协议是为了克服路由信息协议RIP的缺点在1989年开发出来的。
  - “开放”表明OSPF协议不是受某一厂商控制，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** (Shortest Path First, SPF)。
- “开放最短路径优先”只是一个路由选择协议的名称，但这并不表示其他的路由选择协议不是“最短路径优先”。实际上，用于自治系统AS内部的各种路由选择协议（例如RIP），都要寻找一条“最短”的路径。
- OSPF是基于链路状态的，而不像RIP是基于距离向量的。
  - 链路状态 (Link State, LS) 是指本路由器都和哪些路由器相邻，以及相应链路的“代价 (cost)”。
    - “代价”用来表示费用、距离、时延和带宽等，这些都由网络管理人员来决定。



- 优点
  - OSPF基于链路状态并采用最短路径算法计算路由，从算法上保证了不会产生路由环路。
  - OSPF不限制网络规模，更新效率高，收敛速度快。
- OSPF路由器邻居关系的建立和维护
  - OSPF相邻路由器之间通过交互问候 (Hello) 分组来建立和维护邻居关系。

- 问候 (Hello) 分组封装在IP数据报中，发往组播地址224.0.0.5。IP数据报首部中的协议号字段的取值为89，表明IP数据报的数据载荷为OSPF分组。

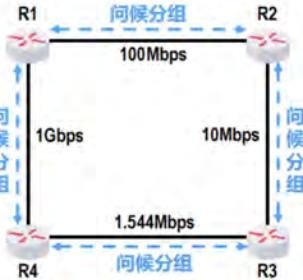


- 问候 (Hello) 分组的发送周期为10秒。

- 若40秒未收到来自邻居路由器的问候 (Hello) 分组，则认为邻居路由器不可达。

- 每个路由器都会建立一张邻居表。

R1的邻居表		
邻居ID	接口	“存活”倒计时
R2	1	36秒
R4	0	18秒

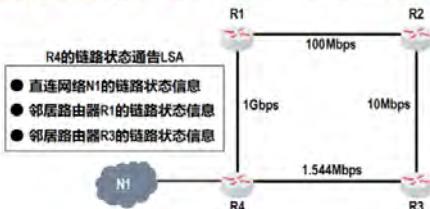


- 链路状态通告

### ■ 使用OSPF的每个路由器都会产生链路状态通告 (Link State Advertisement, LSA)。

#### ■ LSA中包含以下两类链路状态信息：

- 直连网络的链路状态信息
- 邻居路由器的链路状态信息



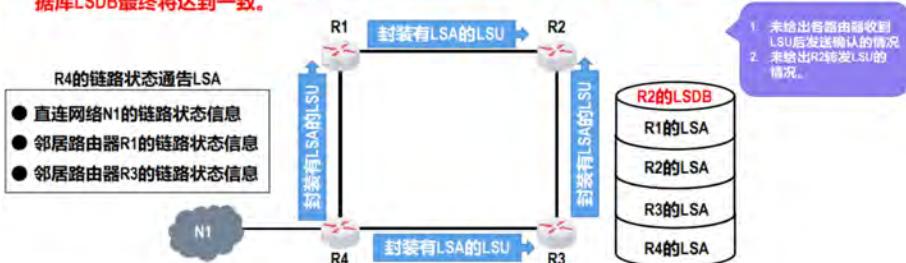
- 链路状态更新分组
- 链路状态数据库

### ■ 链路状态通告LSA被封装在链路状态更新 (Link State Update, LSU) 分组中，采用可靠的洪泛法 (Flooding) 进行发送。

- 洪泛法的要点是路由器向自己所有的邻居路由器发送链路状态更新分组，收到该分组的各路由器又将该分组转发给自己所有的邻居路由器（但其上游路由器除外），以此类推。
- 可靠是指收到链路状态更新分组后要发送确认，收到重复的更新分组无需再次转发，但要发送一次确认。

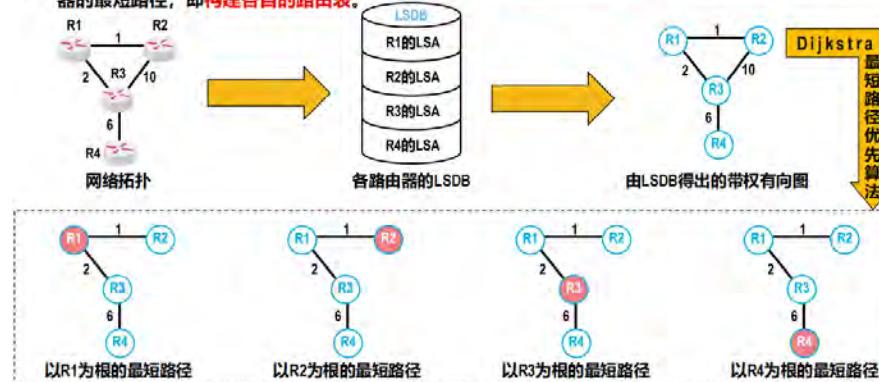
### ■ 使用OSPF的每一个路由器都有一个链路状态数据库 (Link State Database, LSDB)，用于存储链路状态通告LSA。

### ■ 通过各路由器洪泛发送封装有各自链路状态通告LSA的链路状态更新分组LSU，各路由器的链路状态数据库LSDB最终将达到一致。



- 基于链路状态数据库进行最短路径优先计算

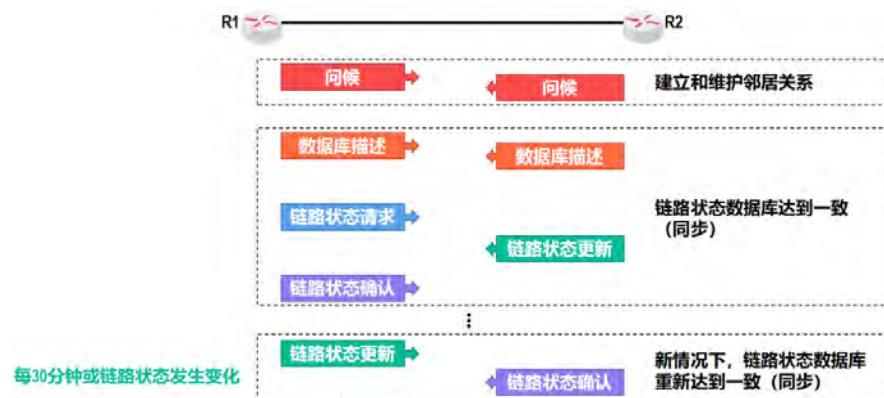
### ■ 使用OSPF的各路由器，基于链路状态数据库LSDB进行最短路径优先计算，构建出各自到达其他各路由器的最短路径，即构建各自的路由表。



#### 4.4.4.2. 五种分组类型

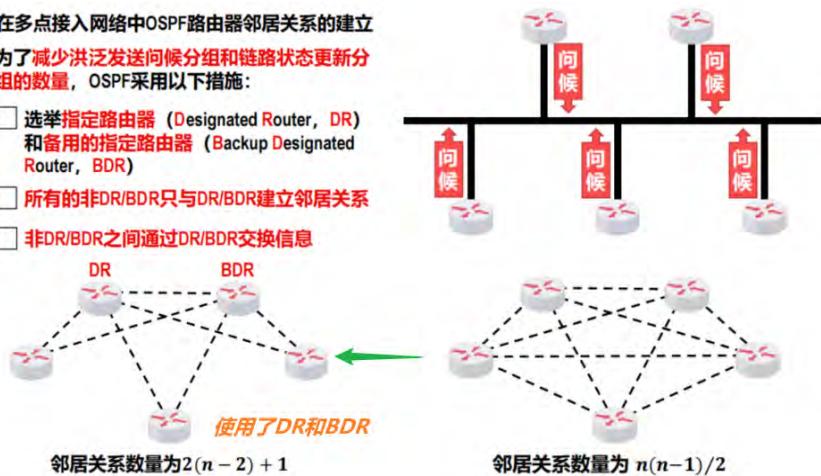
问候 (Hello)	用来发现和维护邻居路由器的可达性。
数据库描述 (Database Description)	用来向邻居路由器给出自己的链路状态数据库中的所有链路状态项目的摘要信息。
链路状态请求 (Link State Request)	用来向邻居路由器请求发送某些链路状态项目的详细信息。
链路状态更新 (Link State Update)	路由器使用链路状态更新分组将其链路状态信息进行洪泛发送，即用洪泛法对整个系统更新链路状态。
链路状态确认 (Link State Acknowledgement)	对链路状态更新分组的确认分组。

#### 4.4.4.3. 基本工作过程



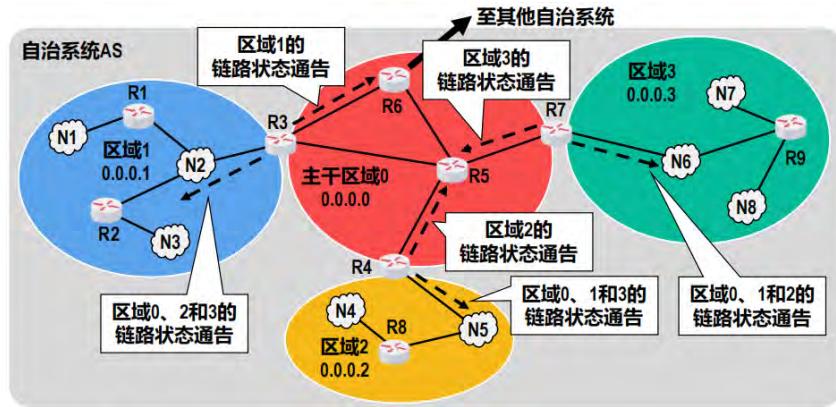
#### 4.4.4.4. 多点接入网络中得OSPF路由器

- 在多点接入网络中OSPF路由器邻居关系的建立
- 为了减少洪泛发送问候分组和链路状态更新分组的数量，OSPF采用以下措施：
  - 选举指定路由器 (Designated Router, DR) 和备用的指定路由器 (Backup Designated Router, BDR)
  - 所有的非DR/BDR只与DR/BDR建立邻居关系
  - 非DR/BDR之间通过DR/BDR交换信息



#### 4.4.4.5. OSPF划分区域

- 为了使OSPF协议能够用于规模很大的网络，OSPF把一个自治系统AS再划分为若干个更小的范围，称为区域 (area) 。
  - 每个区域的规模不应太大，一般所包含的路由器不应超过200个。
  - 划分区域的好处就是把利用洪泛法交换链路状态信息的范围局限于每一个区域，而不是整个自治系统 AS，这样就减少了整个网络上的通信量。

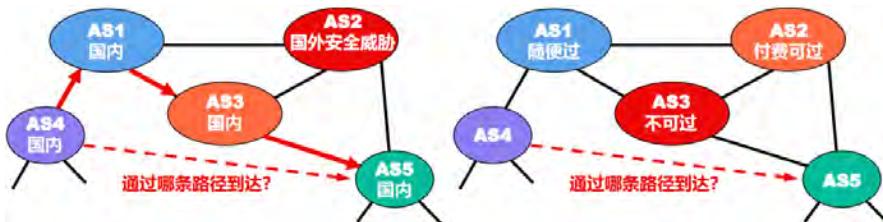


- 自治系统边界路由器 (AS Border Router, ASBR) : R6
- 主干路由器 (Backbone Router, BBR) : R3、R4、R5、R6和R7
- 区域内路由器 (Internal Router, IR) : 区域1内的R1和R2, 区域2内的R8, 区域3内的R9
- 区域边界路由器 (Area Border Router, ABR) : R3、R4和R7

#### 4.4.5. 边界网关协议BGP (封装在TCP)

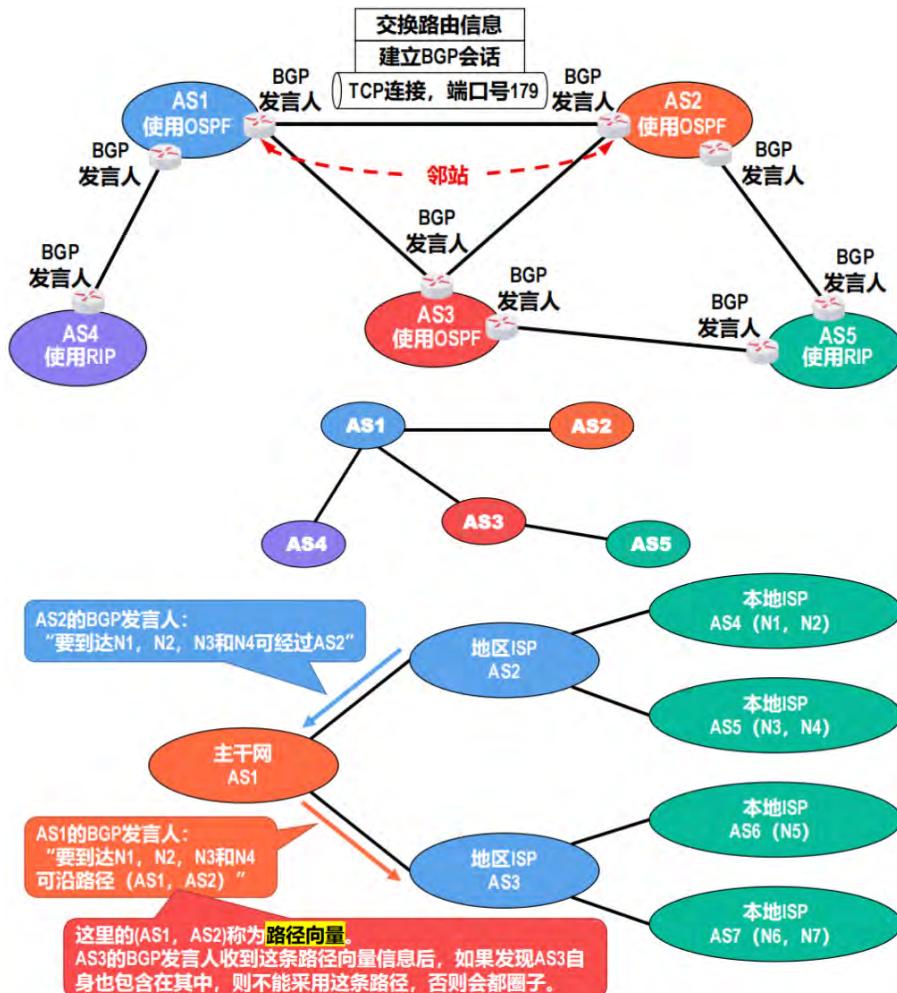
##### 4.4.5.1. 基本概念

- 边界网关协议 (Border Gateway Protocol, BGP) 属于外部网关协议EGP这个类别，用于自治系统AS之间的路由选择协议。
  - 由于在不同AS内度量路由的“代价”（距离、带宽、费用等）可能不同，因此对于AS之间的路由选择，使用统一的“代价”作为度量来寻找最佳路由是不行的。
  - AS之间的路由选择还必须考虑相关策略（政治、经济、安全等）。
  - BGP只能是力求寻找一条能够到达目的网络且比较好的路由（即不能兜圈子），而并非要寻找一条最佳路由。



##### • BGP发言人

- 在配置BGP时，每个AS的管理员要选择至少一个路由器作为该AS的“BGP发言人”。
- 一般来说，两个BGP发言人都是通过一个共享网络连接在一起的，而BGP发言人往往就是BGP边界路由器。
- 使用TCP连接交换路由信息的两个BGP发言人，彼此称为对方的邻站 (neighbor) 或对等站 (peer)。
- BGP发言人除了运行BGP协议外，还必须运行自己所在AS所使用的内部网关协议IGP，例如RIP或OSPF。
- BGP发言人交换网络可达性的信息，也就是要到达某个网络所要经过的一系列自治系统。
- 当BGP发言人相互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自治系统的较好的路由，也就是构造出树形结构且不存在环路的自治系统连通图。
- BGP适用于多级结构的因特网。



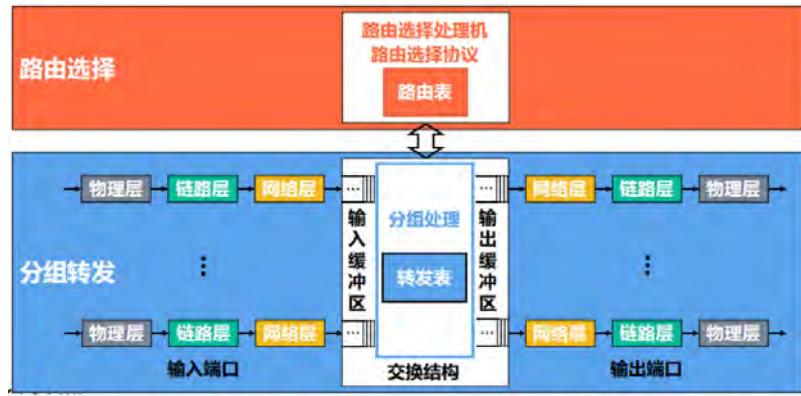
- BGP交换路由信息，要先建立TCP连接，在此基础上交换TCP报文。

#### 4.4.5.2. BGP-4的四种报文

打开 OPEN	用来与相邻的另一个BGP发言人建立关系，使通信初始化。
保活 KEEPALIVE	用来周期性地证实邻站的连通性。
更新 UPDATE	用来通告某一条路由的信息，以及列出要撤销的多条路由。
通知 NOTIFICATION	用来发送检测到的差错。

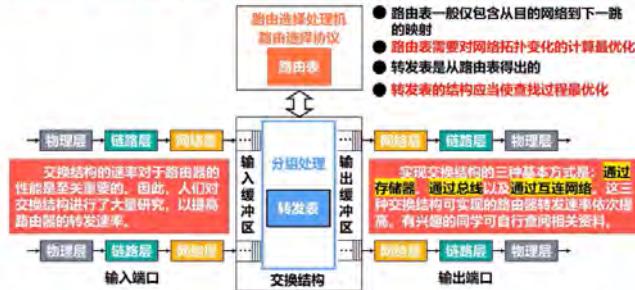
#### 4.4.6. 路由器的基本工作原理

- 路由器是一种具有多个输入端口和输出端口的专用计算机，其任务是转发分组。



□ 路由选择部分：核心构件是**路由选择处理机**，其任务是根据所使用的**路由选择协议**，周期性地与其他路由器进行路由信息的交换，以便构建和更新**路由表**。

□ 分组转发部分：由一组输入端口、**交换结构**以及一组输出端口。

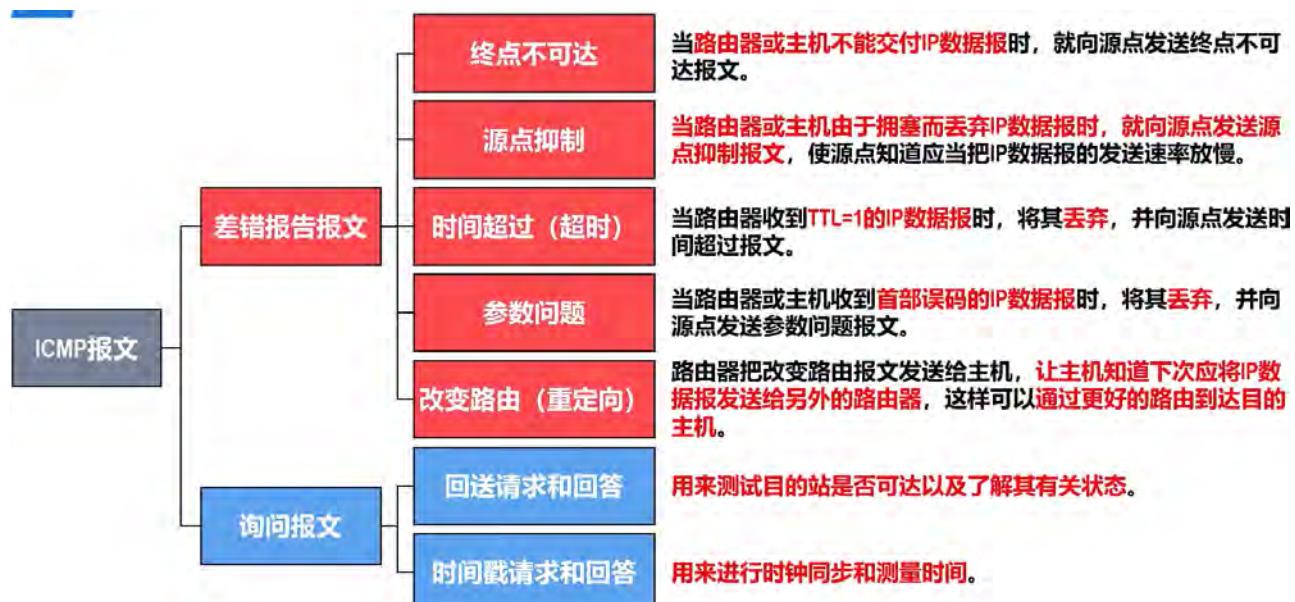


## 4.5. 网际控制协议ICMP (封装在IP)

### 4.5.1. 概述

- 为了更有效地转发IP数据报以及提高IP数据报交付成功的机会，TCP/IP体系结构的网际层使用了**网际控制报文协议** (Internet Control Message Protocol, ICMP) [RFC 792]。
- 主机或路由器使用ICMP来发送差错报告报文和询问报文。
- ICMP报文被封装在IP数据报中发送。

### 4.5.2. ICMP报文种类



#### 4.5.2.1. 差错报告报文

用来向主机或路由器报告差错情况。

- 终点不可达
- 源点抑制
- 时间超过（超时）
- 参数问题
- 改变路由（重定向）

```

1 以下情况不应发送ICMP差错报告报文:
2   对ICMP差错报告报文不再发送ICMP差错报告报文。
3   对第一个分片的IP数据报片的所有后续数据报片都不发送ICMP差错报告报文。
4   对具有多播地址的IP数据报都不发送ICMP差错报告报文。
5   对具有特殊地址（例如127.0.0.0或0.0.0.0）的IP数据报不发送ICMP差错报告报文。

```

#### 4.5.2.2. 询问报文

用来向主机或路由器询问情况。

- 回送请求和回答
  - 由主机或路由器向一个特定的目的主机或路由器发出。
  - 收到此报文的主机或路由器必须给发送该报文的源主机或路由器发送ICMP回送回答报文。
  - 这种询问报文用来测试目的站是否可达以及了解其有关状态。
- 时间戳请求和回答
  - 用来请求某个主机或路由器回答当前的日期和时间。
  - 在ICMP时间戳回答报文中有一个32比特的字段，其中写入的整数代表从1900年1月1日起到当前时刻一共有多少秒。
  - 这种询问报文用来进行时钟同步和测量时间。

#### 4.5.3. ICMP的典型应用

##### 4.5.3.1. 分组网间探测（Packet InterNet Groper, PING）

- 分组网间探测PING用来测试主机或路由器之间的连通性。
  - PING是TCP/IP体系结构的应用层直接使用网际层ICMP的一个例子，它并不使用运输层的TCP或UDP。
  - PING应用所使用的ICMP报文类型为回送请求和回答。

```

1 C:\Users\ASUS>ping www.bilibili.com
2 正在 Ping a.w.bilicdn1.com [2409:8c3c:4:2::75] 具有 32 字节的数据:
3 来自 2409:8c3c:4:2::75 的回复: 时间=31ms
4 来自 2409:8c3c:4:2::75 的回复: 时间=26ms
5 来自 2409:8c3c:4:2::75 的回复: 时间=25ms
6 来自 2409:8c3c:4:2::75 的回复: 时间=29ms
7
8 2409:8c3c:4:2::75 的 Ping 统计信息:
9     数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
10    往返行程的估计时间(以毫秒为单位):
11        最短 = 25ms, 最长 = 31ms, 平均 = 27ms
12

```

- 某些主机或服务器为了防止恶意攻击，并会不理睬外界发来的ICMP回送请求报文。

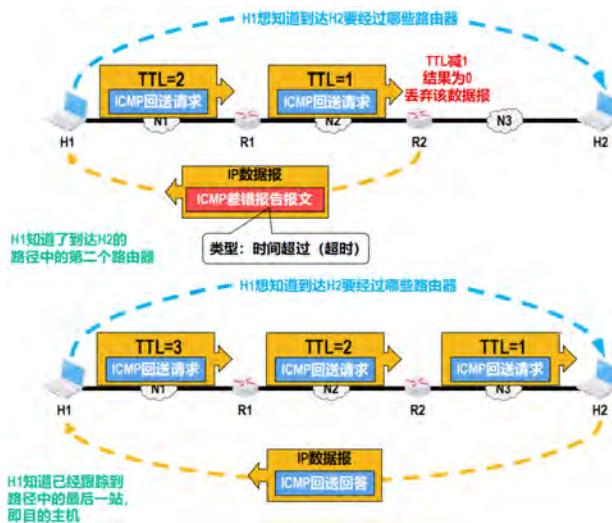
#### 4.5.3.2. 跟踪路由 (traceroute)

- 跟踪路由应用traceroute，用于探测IP数据报从源主机到达目的主机要经过哪些路由器。
  - 在UNIX版本中，具体命令为“traceroute”，其在运输层使用UDP协议，在网络层使用ICMP报文类型只有差错报告报文。
  - 在Windows版本中，具体命令为“tracert”，其应用层直接使用网际层的ICMP协议，所使用的ICMP报文类型有回送请求和回答报文以及差错报告报文。

```

1 C:\Users\ASUS>tracert -d www.bilibili.com
2
3 通过最多 30 个跃点跟踪
4 到 a.w.bilicdn1.com [240e:bf:b800:4300:1::15] 的路由:
5
6 1 6 ms * 33 ms 2001:da8:20c:a053::1
7 2 2 ms 1 ms 1 ms 2001:da8:20c:f0e8::1
8 3 3 ms 3 ms 3 ms 2001:250:215::1
9 4 * * * 请求超时。
10 5 4 ms 3 ms 6 ms 2001:da8:2:122::1
11 6 15 ms 5 ms 7 ms 2001:da8:2:4::1
12 7 4 ms 3 ms 4 ms 2001:da8:2:701:110:108:14:2
13 8 4 ms 3 ms 5 ms 240e::c:1:6200:302
14 9 * 23 ms 23 ms 240e::1:11:51:6303
15 10 23 ms 25 ms 24 ms 240e:f:b800:14e::3
16 11 36 ms 24 ms 24 ms 240e:f:b800:400::3
17 12 26 ms 26 ms 26 ms 240e:f:b800:c85::3
18 13 * * * 请求超时。
19 14 21 ms 23 ms 23 ms 240e:bf:b800:4300:1::15
20
21 跟踪完成。

```

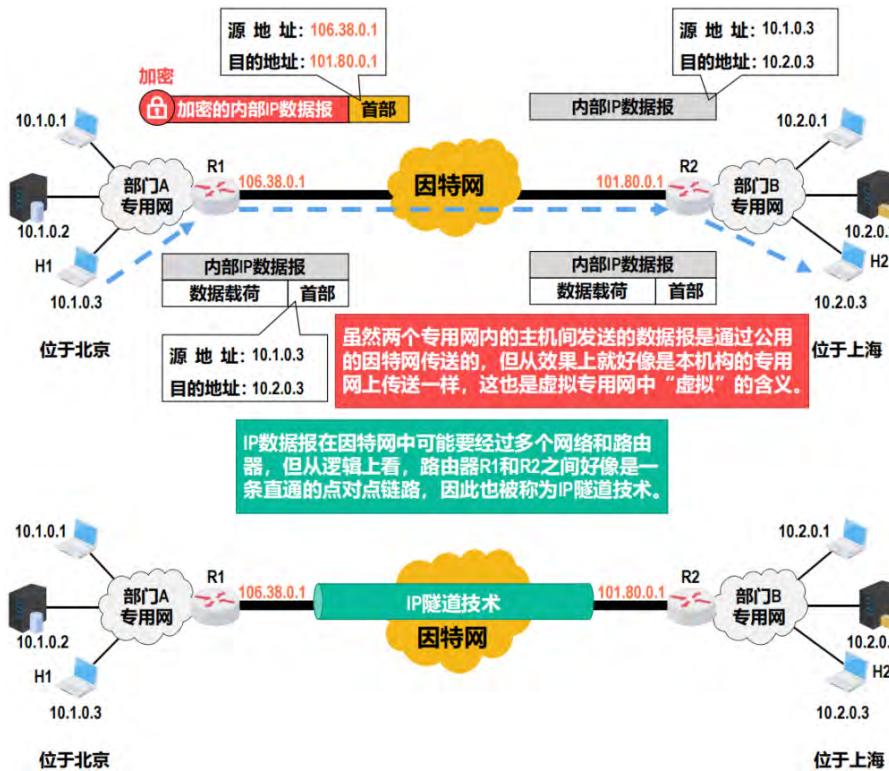


## 4.6. 虚拟专用网VPN和网络地址转换NAT

### 4.6.1. 虚拟专用网VPN

- 虚拟专用网（Virtual Private Network, VPN）利用公用的因特网作为本机构各专用网之间的通信载体，这样形成的网络又称为虚拟专用网。
- 给专用网内各主机配置的IP地址应该是该专用网所在机构可以自行分配的IP地址，这类IP地址仅在机构内部有效，称为专用地址（Private Address），不需要向因特网的管理机构申请。
- [RFC 1918]规定了以下三个CIDR地址块中的地址作为专用地址：
  - 10.0.0.0~10.255.255.255 (CIDR地址块10/8)
  - 172.16.0.0~172.31.255.255 (CIDR地址块172.16/12)
  - 192.168.0.0~192.168.255.255 (CIDR地址块192.168/16)
- 很显然，全世界可能有很多不同机构的专用网具有相同的专用IP地址，但这并不会引起麻烦，因为这些专用地址仅在机构内部使用。

- 在因特网中的所有路由器，对目的地址是专用地址的IP数据报一律不进行转发，这需要由因特网服务提供者ISP对其拥有的因特网路由器进行设置来实现。



- 本例所示的是同一机构内不同部门的内部网络所构成的VPN，又称为内联网VPN。

#### 4.6.2. 网络地址转换NAT

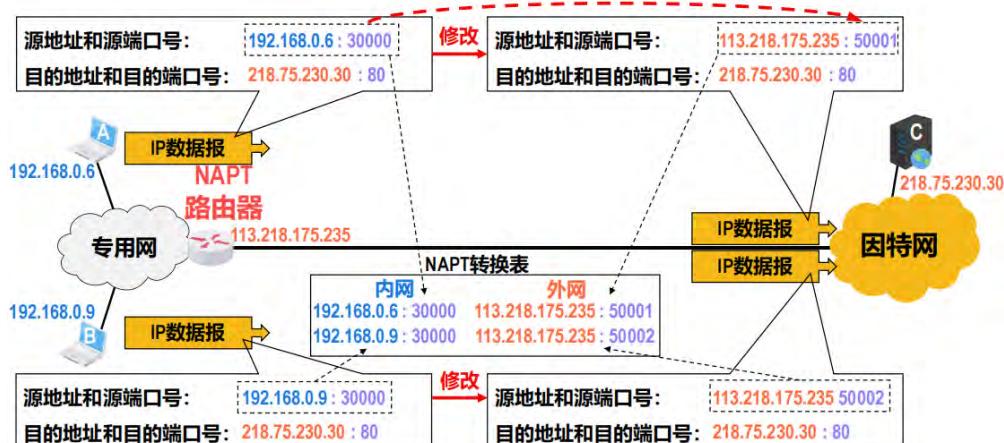
- 网络地址转换 (Network Address Translation, NAT) 技术于1994年被提出，用来缓解IPv4地址空间即将耗尽的问题。
  - NAT能使大量使用内部专用地址的专用网络用户共享少量外部全球地址来访问因特网上的主机和资源。
  - 这种方法需要在专用网络连接到因特网的路由器上安装NAT软件。
  - 装有NAT软件的路由器称为NAT路由器，它至少要有一个有效的外部全球地址 $IP_G$ 。
  - 这样，所有使用内部专用地址的主机在和外部因特网通信时，都要在NAT路由器上将其内部专用地址转换成 $IP_G$ 。

##### 4.6.2.1. 最基本的NET方法

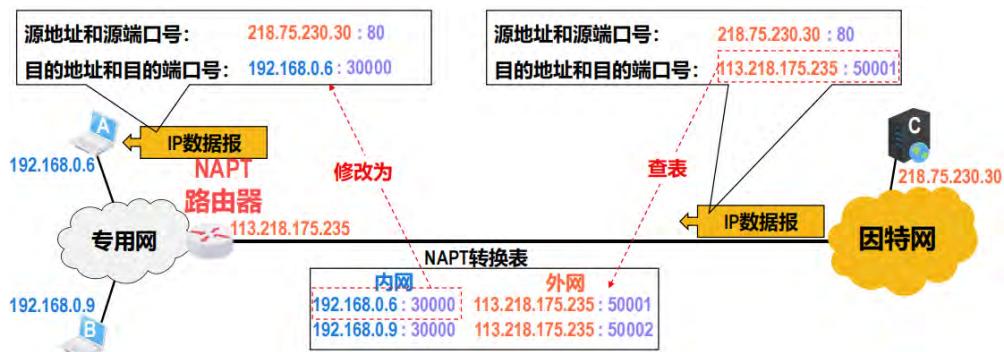


#### 4.6.2.2. 网络地址与端口号转换方法 (NAPT)

- 由于目前绝大多数基于TCP/IP协议栈的网络应用，都使用运输层的传输控制协议TCP或用户数据报协议UDP，为了更加有效地利用NAT路由器中的全球IP地址，现在常将NAT转换和运输层端口号结合使用。
  - 这样就可以使内部专用网中使用专用地址的大量主机，共用NAT路由器上的1个全球IP地址，因而可以同时与因特网中的不同主机进行通信。
- 将NAT和运输层端口号结合使用，称为网络地址与端口号转换（Network Address and Port Translation，NAPT）。
  - 现在很多家用路由器将家中各种智能设备（手机、平板、笔记本电脑、台式电脑、物联网设备等）接入因特网，这种路由器实际上就是一个NAPT路由器，但往往并不运行路由选择协议。
- 主机向因特网发送
  - 与主机A选择的源端口号相同，这纯属巧合（端口号仅在本主机中才有意义）。特意这样举例，就是为了能更好地说明NAPT路由器还会对源端口号重新动态分配。



- 因特网向主机发回



- 尽管NAT（和NAPT）的出现在很大程度上缓解了IPv4地址资源紧张的局面，但NAT（和NAPT）对网络应用并不完全透明，会对某些网络应用产生影响。
- NAT（和NAPT）的一个重要特点就是通信必须由专用网内部发起，因此拥有内部专用地址的主机不能直接充当因特网中的服务器。

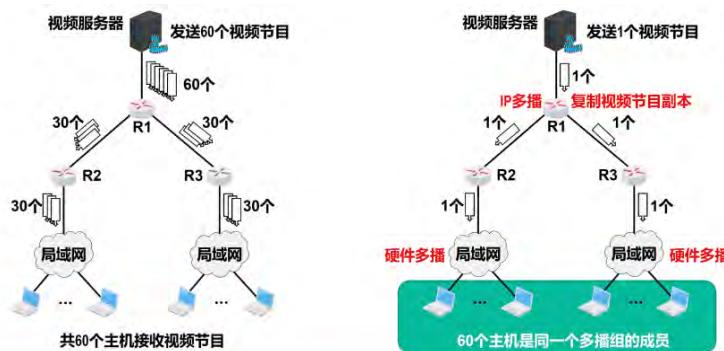


- 对于目前P2P这类需要外网主机主动与内网主机进行通信的网络应用，在通过NAT时会遇到问题，需要网络应用自身使用一些特殊的NAT穿透技术来解决。

## 4.7. IP多播技术

### 4.7.1. 相关基本概念

- 多播（Multicast，也称为组播）是一种实现“一对多”通信的技术，与传统单播“一对一”通信相比，多播可以极大地节省网络资源。
  - 在因特网上进行的多播，称为IP多播。



- 实现IP多播，则因特网中的路由器需要解决的问题
  - IP多播数据报的寻址问题
  - 多播路由选择问题

### 4.7.2. IP多播地址和多播组

- 在IPv4中，D类地址被作为多播地址。



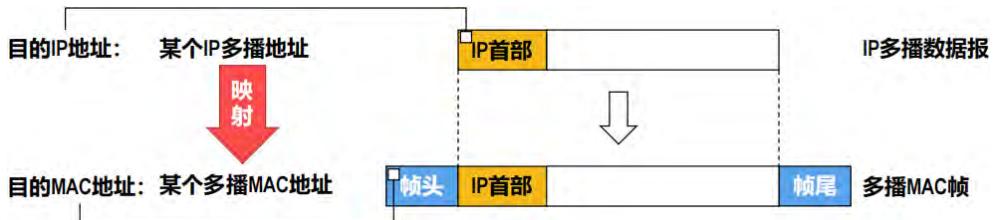
- 多播地址只能用作目的地址，而不能用作源地址。
- 用每一个D类地址来标识一个多播组，使用同一个IP多播地址接收IP多播数据报的所有主机就构成了一个多播组。
  - 每个多播组的成员是可以随时变动的，一台主机可以随时加入或离开多播组。
  - 多播组成员的数量和所在的地理位置也不受限制，一台主机可以属于几个多播组。
  - 非多播组成员也可以向多播组发送IP多播数据报
- 与IP数据报相同，IP多播数据报也是“尽最大努力交付”，不保证一定能够交付给多播组内的所有成员。
- IPv4多播地址又可分为预留的多播地址（永久多播地址）、全球范围可用的多播地址以及本地管理的多播地址[RFC 3330]。

224.0.0.0	基地址（保留）
224.0.0.1	仅在本子网上的所有参加多播的主机和路由器
224.0.0.2	仅在本子网上的所有参加多播的路由器
224.0.0.3	未指派
224.0.0.4	DVMRP路由器
224.0.0.5	OSPF路由器
.....	
	永久多播地址
224.0.1.0	全球范围内都可使用的多播地址
.....	
238.255.255.255	
239.0.0.0	本地管理的多播地址，仅在特定的本地范围内有效
.....	
239.255.255.255	

- IP多播可以分为以下两种
  - 只在本局域网上进行的硬件多播。
  - 在因特网上进行的多播。
- 目前大部分主机都是通过局域网接入因特网的。因此，在因特网上进行多播的最后阶段，还是要把IP多播数据报在局域网上用硬件多播交付给多播组的所有成员。

#### 4.7.3. 在局域网上进行硬件多播

- 由于MAC地址（也称为硬件地址）有多播MAC地址这种类型，因此只要把IPv4多播地址映射成多播MAC地址，即可将IP多播数据报封装在局域网的MAC帧中，而MAC帧首部中的目的MAC地址字段的值，就设置为由IPv4多播地址映射成的多播MAC地址。这样，可以很方便地利用硬件多播来实现局域网内的IP多播。
- 当给某个多播组的成员主机配置其所属多播组的IP多播地址时，系统就会根据映射规则从该IP多播地址生成相应的局域网多播MAC地址。



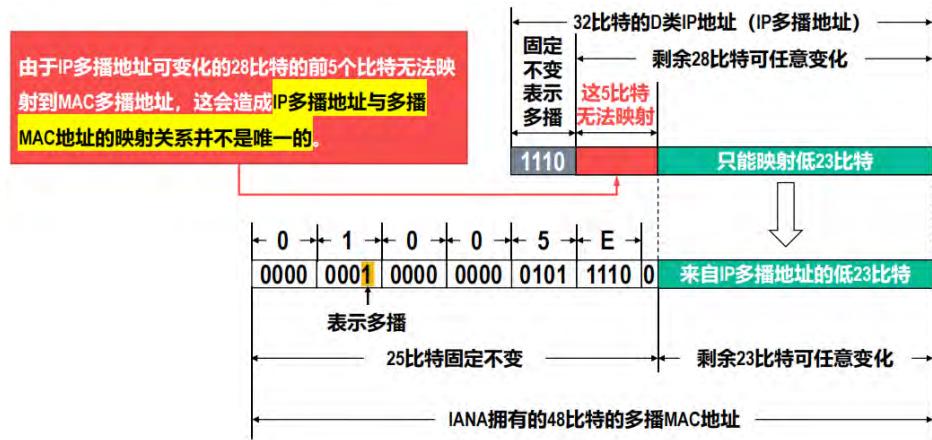
- 因特网号码指派管理局IANA，将自己从IEEE注册管理机构申请到的以太网MAC地址块中从01-00-5E-00-00-00到01-00-5E-7F-FF-FF的多播MAC地址，用于映射IPv4多播地址。
  - 这些多播MAC地址的左起前25个比特都是相同的，剩余23个比特可以任意变化，因此共有 $2^{23}$ 。



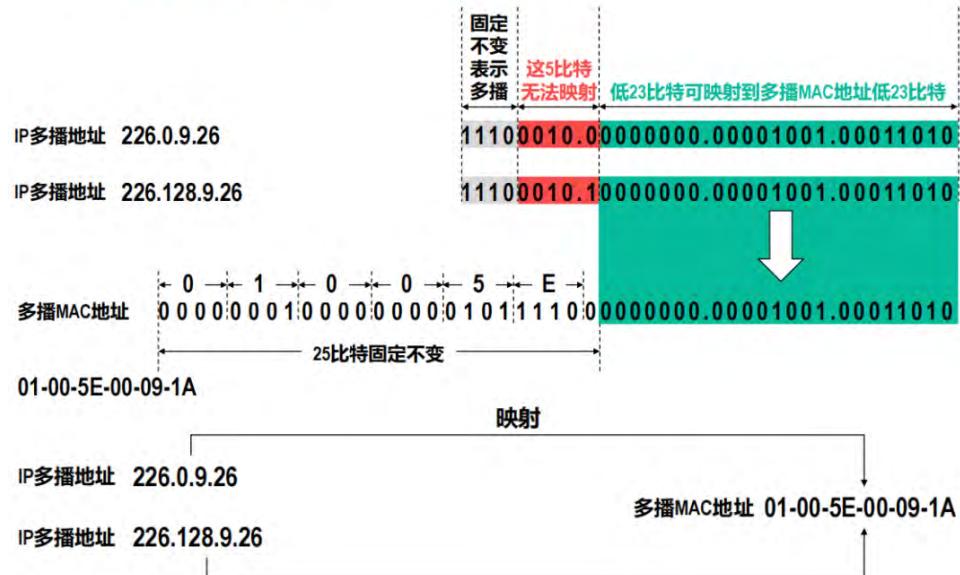
**最小多播MAC地址**  
01-00-5E-00-00-00      0000 | 0001 | 0000 | 0000 | 0101 | 1110 | 0      剩余23比特为“全0”

**最大多播MAC地址**  
01-00-5E-7F-FF-FF      0000 | 0001 | 0000 | 0000 | 0101 | 1110 | 0      剩余23比特为“全1”

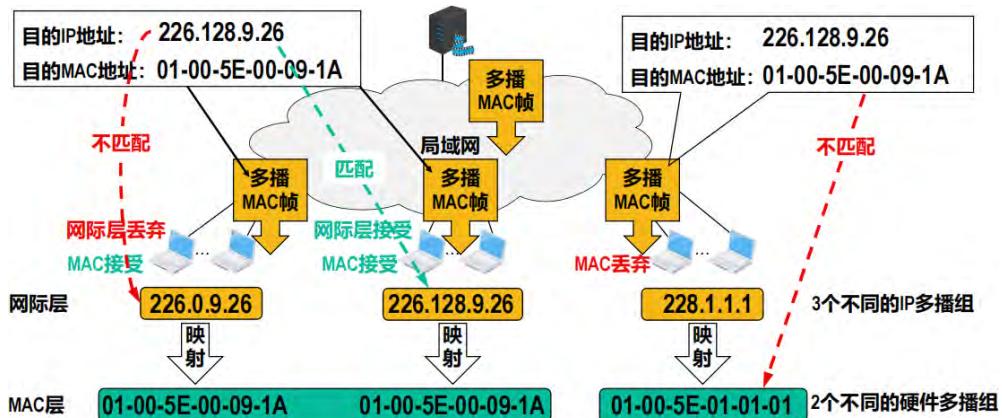
- IP多播地址与多播MAC地址的映射关系并不是唯一的



【举例】IP多播地址与多播MAC地址的映射关系并不是唯一的。

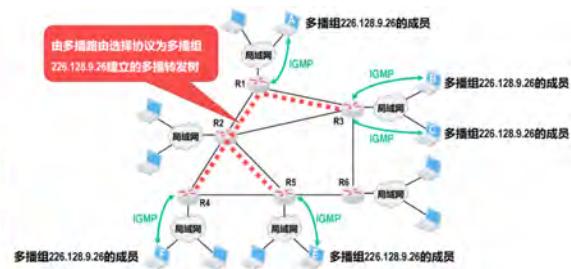
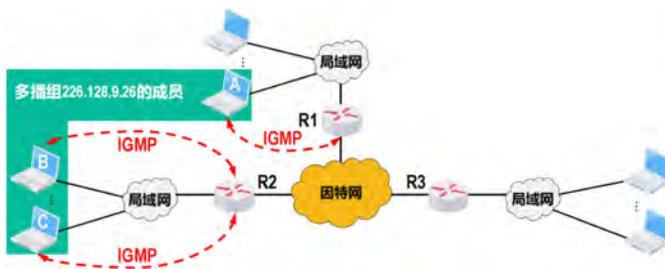


- 收到IP多播数据报的主机还要在网际层利用软件进行过滤，把不是主机要接收的IP多播数据报丢弃



#### 4.7.4. 在因特网上进行IP多播需要的两种协议

网际组管理协议IGMP	多播路由选择协议
<p>■ 网际组管理协议 (Internet Group Management Protocol, IGMP) 是TCP/IP体系结构网际层中的协议，其作用是让连接在本地局域网上的多播路由器知道本局域网上是否有主机（实际上是主机中的某个进程）加入或退出了某个多播组。</p> <p>■ IGMP仅在本网络有效，使用IGMP并不能知道多播组所包含的成员数量，也不能知道多播组的成员都分布在哪些网络中。</p> <p>■ 仅使用IGMP并不能在因特网上进行IP多播。连接在局域网上的多播路由器还必须和因特网上的其他多播路由器协同工作，以便把IP多播数据报用最小的代价传送给所有的多播组成员，这就需要使用多播路由选择协议。</p>	<p>■ 多播路由选择协议的主要任务是：在多播路由器之间为每个多播组建立一个多播转发树。</p> <ul style="list-style-type: none"> <li>□ 多播转发树连接多播源和所有拥有该多播组成员的路由器。</li> <li>□ IP多播数据报只要沿着多播转发树进行洪泛，就能被传送到所有拥有该多播组成员的多播路由器。</li> <li>□ 之后，在多播路由器所直连的局域网内，多播路由器通过硬件多播，将IP多播数据报发送给该多播组的所有成员。</li> </ul> <p>■ 针对不同的多播组需要维护不同的多播转发树，而且必须动态地适应多播组成员的变化，但此时网络拓扑并不一定发生变化，因此多播路由选择协议要比单播路由选择协议（例如RIP、OSPF等）复杂得多。</p> <p>■ 即使某个主机不是任何多播组的成员，它也可以向任一多播组发送多播数据报。</p> <p>■ 为了覆盖多播组的所有成员，多播转发树可能要经过一些没有多播组成员的路由器。</p>



#### 4.7.5. 网际组管理协议IGMP（封装在IP）

- 网际组管理协议IGMP目前的最新版本是2002年10月公布的IGMPv3[RFC 3376]。
- IGMP报文被封装在IP数据报中传送



##### 4.7.5.1. 三种报文类型

- 成员报文报文
- 成员查询报文
- 离开组报文

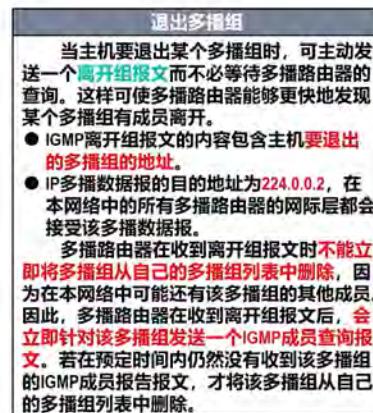
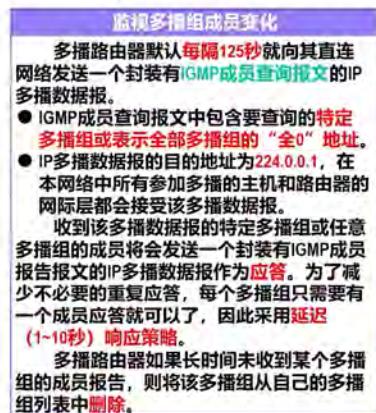
##### 4.7.5.2. 基本工作原理

## 网际组管理协议IGMP

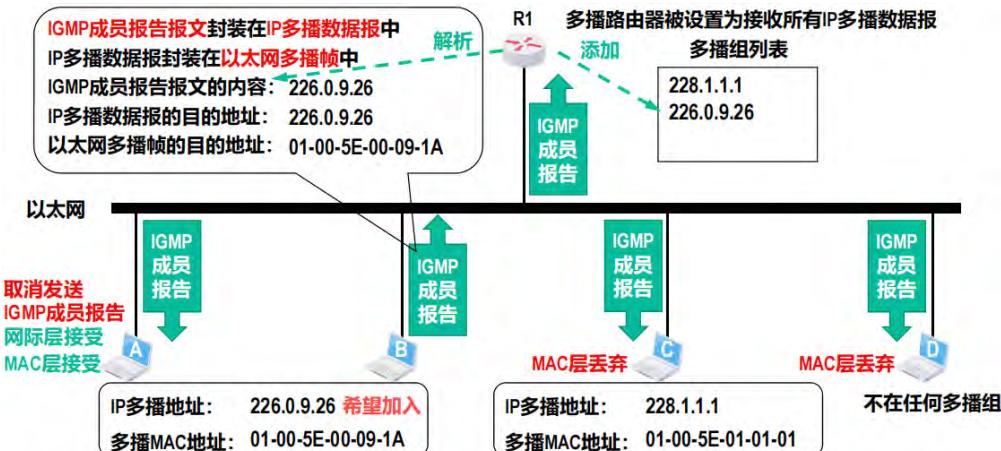
- IGMP有三种报文类型：

成员报告报文 离开组报文 成员查询报文

协议字段的值 = 2, 表示数据载荷部分是IGMP报文。  
目的地址字段的值根据其所封装IGMP报文类型各有不同，但都属于IP多播地址。  
生成时间TTL字段的值 = 1, 避免封装IGMP报文的IP多播数据报被路由器转发到其他网络。

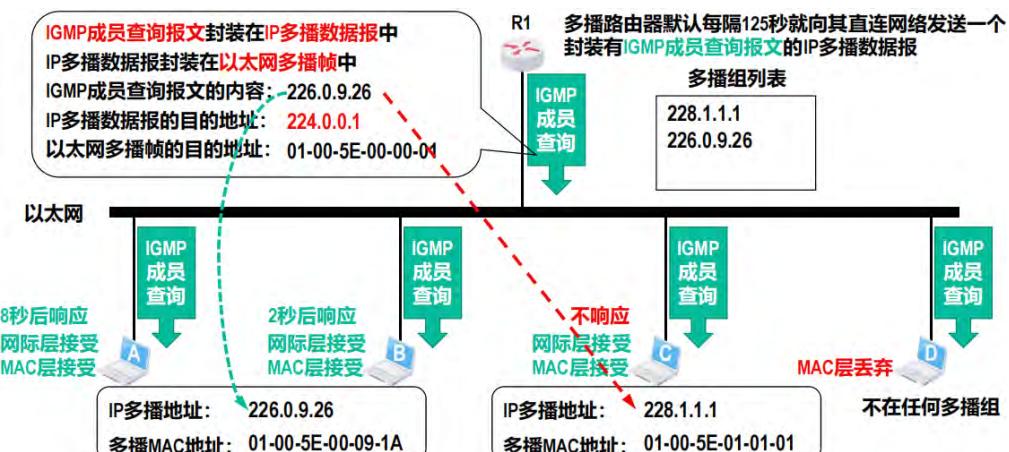


### 4.7.5.2.1. 加入多播组



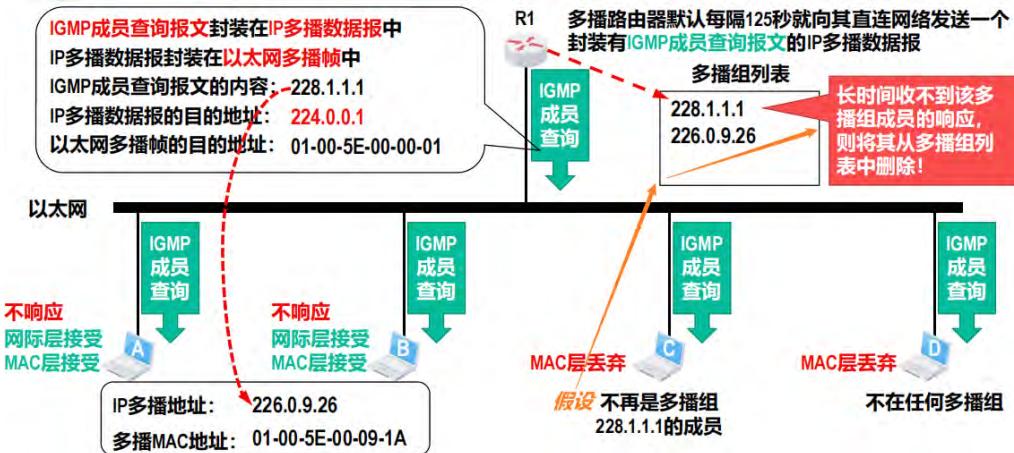
### 4.7.5.2.2. 监视多播组的成员变化

- 多播路由器默认每隔125秒就向其直连网络发送一个封装有IGMP成员查询报文的IP多播数据报。
- 成员查询报文中，224.0.0.1为特殊的IP多播地址，在本网络中所有参加多播的主机和路由器的网际层都会接受该多播数据报。



- 收到IGMP成员查询报文的被查询多播组的任何成员，都会发送IGMP成员报告文作为应答。
- 为了减少不必要的重复应答，每个多播组只需要一个成员应答就行了。

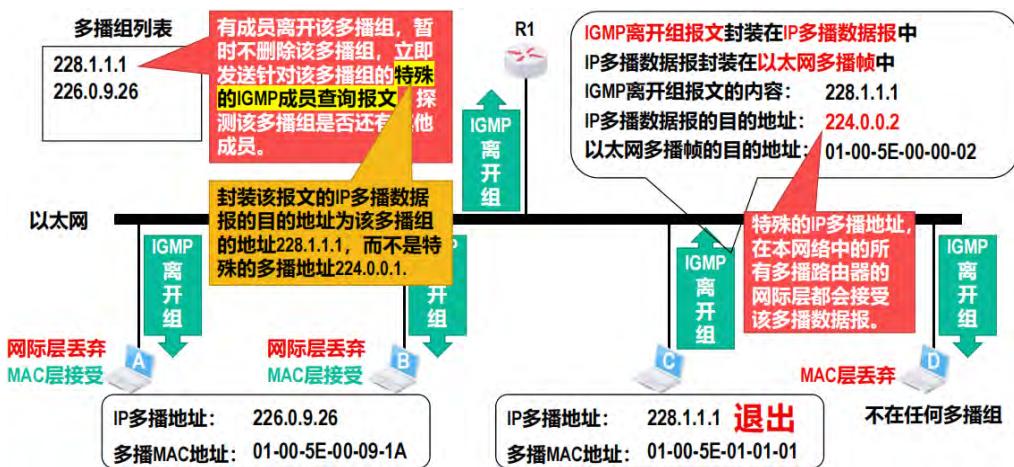
- 应答采用延迟响应的策略，主机收到查询报文后再1-10秒内等待一段随机时间后进行响应（不是立即响应）。
- 在这段时间内如果收到了同组成员发送的成员报告报文，则取消响应。
  - 本例中B先发送成员报告报文，则A收到B的成员报告文后取消响应。
- 路由器R1收到B的成员报告报文后对其进行解析并更新多播组列表。



- 同一网络中的多播路由器可能不止一个，但没有必要每个多播路由器都周期性地发送IGMP成员查询报文。
- 只要在这些多播路由器中选择一个作为查询路由器，由查询路由器发送IGMP成员查询报文，而其他的多播路由器仅被动接收响应并更新自己的多播组列表即可。
- 选择查询路由器的方法：
  - 每个多播路由器若监听到源IP地址比自己的IP地址小的IGMP成员查询报文则退出选举。
  - 最后，网络中只有IP地址最小的多播路由器成为查询路由器。

#### 4.7.5.2.3. 退出多播组

- IGMPv2在IGMPv1的基础上增加了一个可选项：当主机要退出某个组时，可主动发送一个离开组报文而不必等待多播路由器的查询。这样可使多播路由器能够更快地发现某个组有成员离开。



#### 4.7.6. 多播路由选择协议（封装在）

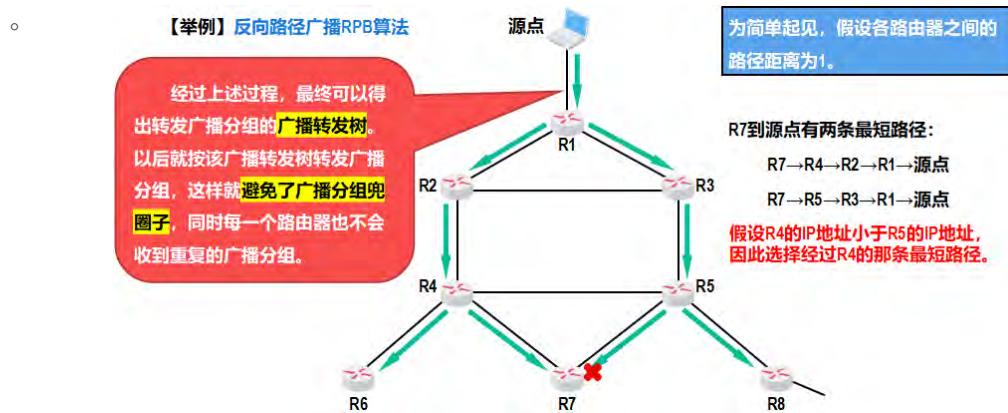
##### 4.7.6.1. 多播路由选择协议

- 多播路由选择协议的主要任务是：在多播路由器之间为每个多播组建立一个多播转发树。
  - 多播转发树连接多播源和所拥有该多播组成员的路由器。
- 建多播转发树的方法

- 基于源树 (Source-Base Tree) 多播路由选择
- 组共享树 (Group-Shared Tree) 多播路由选择

#### 4.7.6.1.1. 基于源树多播路由选择

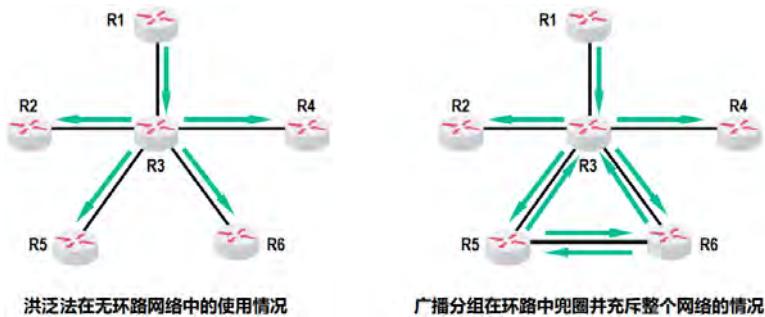
- 反向路径多播算法 (Reverse Path Multicasting, RPM)
  - 利用反向路径广播 (Reverse Path Broadcasting, RPB) 算法建立一个广播转发树。
    - 每一台路由器在收到一个广播分组时，先检查该广播分组是否是从源点经最短路径传来的。
      - 若是，本路由器就从自己除刚才接收该广播分组的接口的所有其他接口转发该广播分组。
      - 否则，丢弃该广播分组。
      - 如果本路由器有好几个邻居路由器都处在到源点的最短路径上，也就是存在好几条同样长度的最短路径，那么只能选取一条最短路径。选取的规则是这几条最短路径中的邻居路由器的IP地址最小的那条最短路径。
    - RPB中“反向路径”的意思是：在计算最短路径时把源点当作终点。



- 利用剪枝 (Pruning) 算法，剪除广播转发树中的下游非成员路由器，获得一个多播转发树。

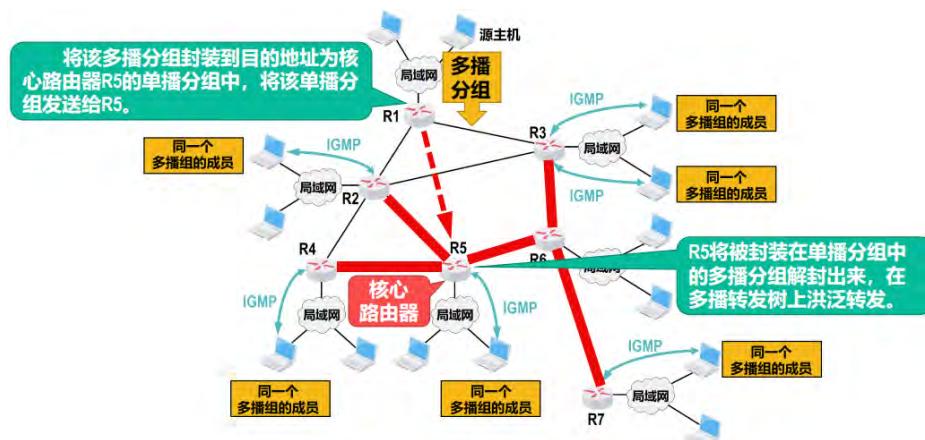


- 若被剪枝的路由器通过IGMP又发现了新的多播组成员，则会向上游路由器发送一个嫁接报文，并重新加入到多播转发树中。
- 尽管R2没有多播组成员，但也要保留R2以确保多播转发树的连通性。
- 要建立广播转发树，可以使用洪泛 (Flooding) 法。
  - 利用反向路径广播RPB算法生成的广播转发树，不会存在环路，因此可以避免广播分组在环路中兜圈。



#### 4.7.6.1.2. 组共享树多播路由选择

- 组共享树多播路由选择采用基于核心的分布式生成树算法来建立共享树。
  - 该方法在每个多播组中指定一个核心（core）路由器，以该路由器为根，建立一棵连接多播组的所有成员路由器的生成树，作为多播转发树。
- 每个多播组中除了核心路由器，其他所有成员路由器都会向自己多播组中的核心路由器单播加入报文。
  - 加入报文通过单播朝着核心路由器转发，直到它到达已经属于该多播生成树的某个节点或者直接到达该核心路由器。
  - 加入报文所经过的路径，就确定了一条从单播该报文的边缘节点到核心路由器之间的分支，而这个新分支就被嫁接到现有的多播转发树上。



#### 4.7.6.2. 因特网的多播路由选择协议

■ 目前还没有在整个因特网范围使用的多播路由选择协议。下面是一些建议使用的多播路由选择协议：

- 距离向量多播路由选择协议 (Distance Vector Multicast Routing Protocol, DVMRP) [RFC 1075]。
- 开放最短路径优先的多播扩展 (Multicast Extensions to OSPF, MOSPF) [RFC 1585]。
- 协议无关多播-稀疏方式 (Protocol Independent Multicast-Sparse Mode, PIM-SM) [RFC 2362]。
- 协议无关多播-密集方式 (Protocol Independent Multicast-Dense Mode, PIM-DM) [RFC 3973]。
- 基于核心的转发树 (Core Based Tree, CBT) [RFC 2189, RFC 2201]。

■ 尽管因特网工程任务组 IETF 努力推动着因特网上的全球多播主干网 (Multicast Backbone On the Internet, MBONE) 的建设，但至今在因特网上的 IP 多播还没有得到大规模的应用。

- 主要原因是：改变一个已成功运行且广泛部署的网络层协议是一件及其困难的事情。
- 目前 IP 多播主要应用在一些局部的园区网络、专用网络或者虚拟专用网中。

另外，P2P 技术的广泛应用推动了应用层多播技术的发展，许多视频流公司和内容分发公司，通过构建自己的应用层多播覆盖网络来分发它们的内容。但上述多播路由选择协议的算法思想在应用层多播中依然适用。

## 4.8. 移动IP技术概述

### 4.8.1. 移动性对因特网应用的影响

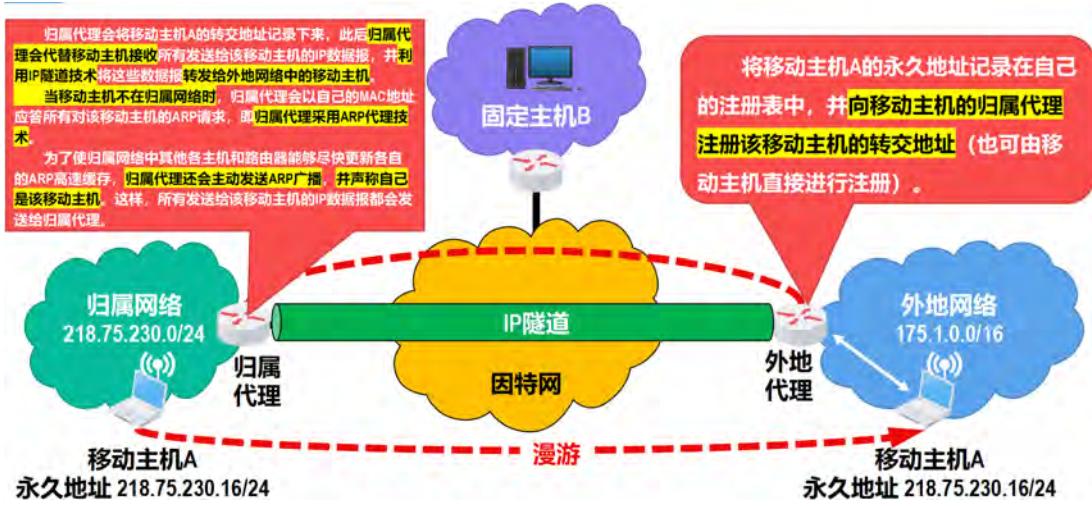


### 4.8.2. 移动IP技术的相关基本概念

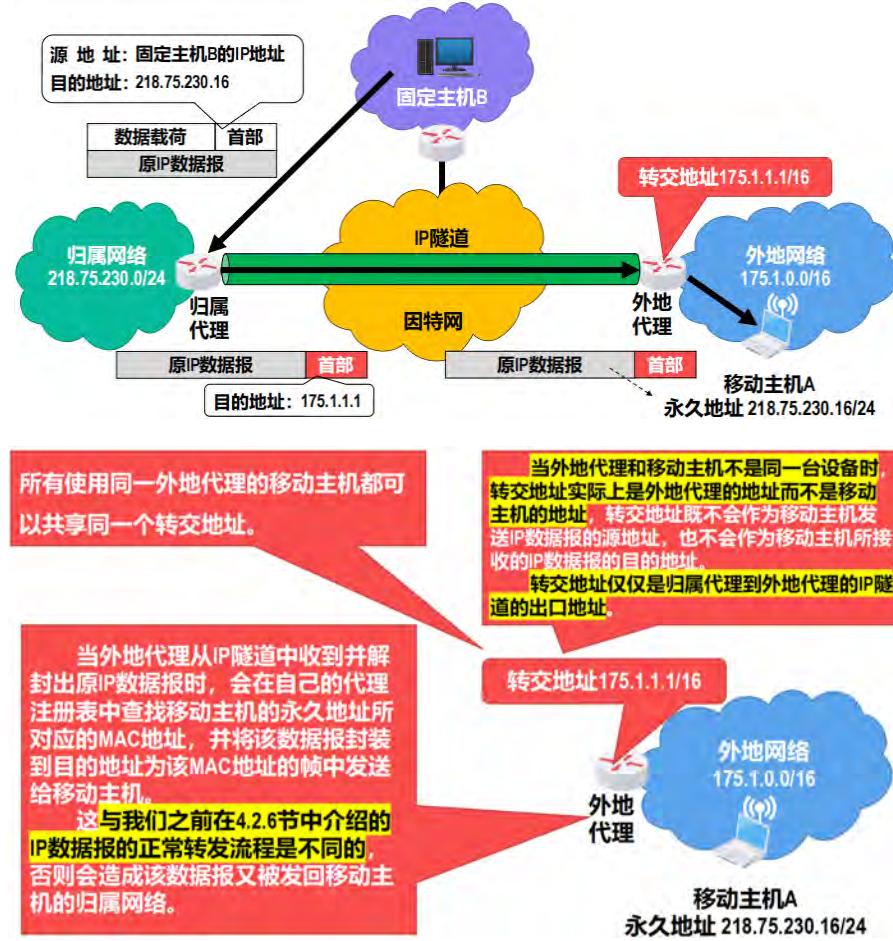
- 移动IP (Mobile IP) 是因特网工程任务组IETF开发的一种技术[RFC 3344]，该技术使得移动主机在各网络之间漫游时，仍然能够保持其原来的IP地址不变。
- 移动IP技术还为因特网中的非移动主机提供了相应机制，使得它们能够将IP数据报正确发送到移动主机。
- 基本概念
  - 归属网络 (Home Network)
    - 每个移动主机都有一个默认连接的网络或初始申请接入的网络。
  - 永久地址 (Permanent Address) 或归属地址 (Home Address)。
    - 移动主机在归属网络中的IP地址 (在其整个移动通信过程中是始终不变的)
  - 归属代理 (Home Agent)
    - 在归属网络中，代表移动主机执行移动管理功能的实体。
    - 归属代理通常就是连接在归属网络上的路由器，然而它作为代理的特定功能则是在网络层完成的。
  - 外地网络 (Foreign Network) 或被访网络 (VisitedNetwork)
    - 移动主机当前漫游所在的网络。
  - 外地代理 (ForeignAgent)
    - 在外地网络中，帮助移动主机执行移动管理功能的实体。
    - 外地代理通常就是连接在外地网络上的路由器。
  - 转交地址 (Care-of Address)
    - 外地代理会为移动主机提供一个临时使用的属于外地网络的转交地址。

### 4.8.3. 移动IP技术的基本工作原理

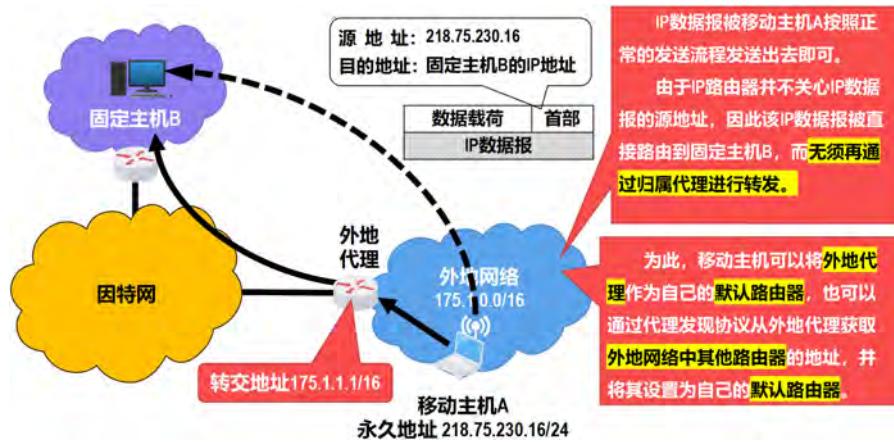
#### 4.8.3.1. 代理发现与注册



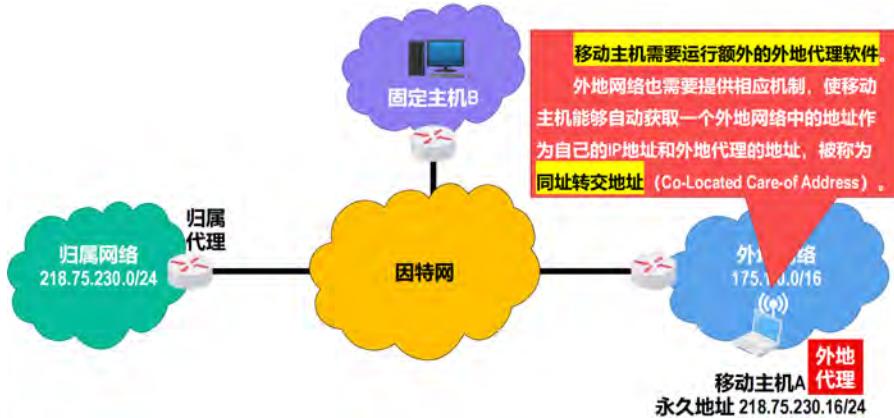
#### 4.8.3.2. 固定主机向移动主机发送IP数据报



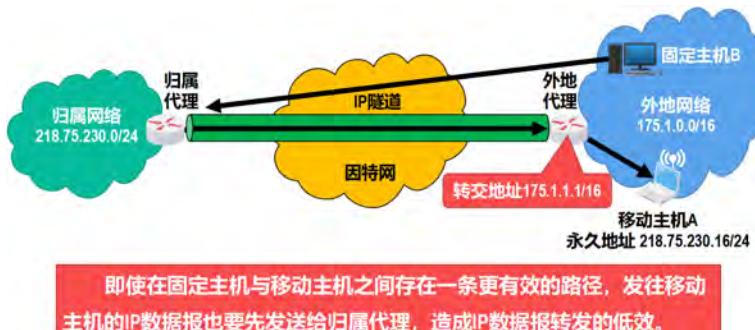
#### 4.8.3.3. 移动主机向固定主机发送IP数据报



#### 4.8.3.4. 同址转交地址方式



#### 4.8.3.5. 三角形路由问题



- 解决三角形路由问题的一种方法

- 给固定主机配置一个**通信代理**，固定主机发送给移动主机的IP数据报，都要通过该通信代理转发。
- 通信代理先从归属代理获取移动主机的转交地址，之后所有发送给移动主机的IP数据报，都利用转交地址直接通过IP隧道发送给移动主机的外地代理，而无须再通过移动主机的归属代理进行转发。
- 这种方法以增加复杂性为代价，并要求固定主机也要配置通信代理，也就是对固定主机不再透明。

## 4.9. IPv6

### 4.9.1. IPv6的诞生背景

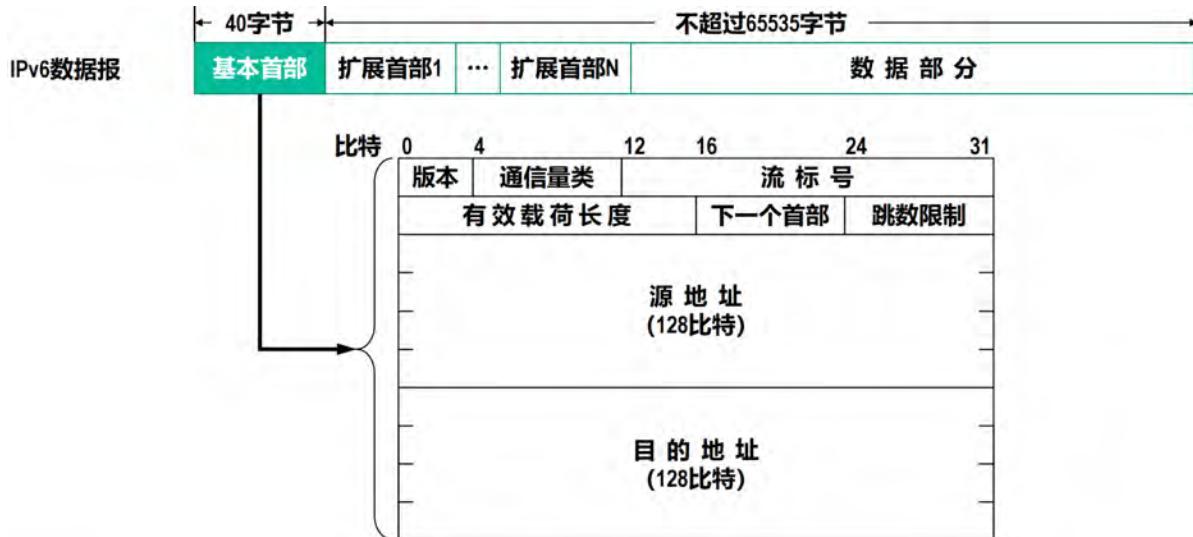
- IPv4地址存在缺陷
  - IPv4地址的长度仅为32比特
  - 早期的编址方法不够合理，造成IPv4地址资源的浪费。
- 2011年2月3日，因特网号码分配管理局IANA宣布IPv4地址已经分配完毕
- 如果没有网络地址转换NAT技术的广泛应用，IPv4早已停止发展。

- 解决IPv4地址耗尽的根本措施就是采用具有更大地址空间（IP地址的长度为128比特）的新版本IP，即IPv6。
- 到目前为止，IPv6还只是草案标准阶段
- 尽早开始过渡到IPv6的好处
  - 有更多时间来平滑过渡
  - 有更多时间来培养IPv6的专门人才
  - 及早提供IPv6服务比较便宜

#### 4.9.2. IPv6引进的主要变化

<b>更大的地址空间</b>	IPv6将IPv4的 <b>32比特</b> 地址空间增大到了 <b>128比特</b> ，在采用合理编址方法的情况下，在可预见的未来是不会用完的。
<b>扩展的地址层次结构</b>	可划分为更多的层次，这样可以更好地反映出因特网的拓扑结构，使得 <b>对寻址和路由层次的设计更具有灵活性</b> 。
<b>灵活的头部格式</b>	与IPv4头部并不兼容。IPv6定义了许多 <b>可选的的扩展头部</b> ，不仅可提供比IPv4更多的功能，而且还可以 <b>提高路由器的处理效率</b> ，因为路由器对逐跳扩展头部外的其他扩展头部都不进行处理。
<b>改进的选项</b>	IPv6允许分组 <b>包含有选项的控制信息</b> ，因而可以包含一些新的选项。然而IPv4规定的选项却是 <b>固定不变的</b> 。
<b>允许协议继续扩充</b>	<b>这一点很重要，因为技术总是在不断地发展，而新的应用也会层出不穷。然而IPv4的功能却是固定不变的。</b>
<b>支持即插即用 (即自动配置)</b>	IPv6支持主机或路由器自动配置IPv6地址及其他网络配置参数。因此 <b>IPv6不需要使用DHCP</b> 。
<b>支持资源的预分配</b>	IPv6能为实时音视频等要求保证一定带宽和时延的应用，提供 <b>更好的服务质量保证</b> 。

#### 4.9.3. IPv6数据包的基本首部



#### 4.9.4. IPv6数据包的扩展首部

- IPv4数据报如果在其首部中使用了选项字段，则在数据报的整个传送路径中的全部路由器，都要对选项字段进行检查，这就降低了路由器处理数据报的速度。
  - 实际上，在路径中的路由器对很多选项是不需要检查的。因此，为了提高路由器对数据包的处理效率，IPv6把原来IPv4首部中的选项字段都放在了扩展首部中，由路径两端的源点和终点的主机来处理，而数据报传送路径中的所有路由器都不处理这些扩展首部（除逐跳选项扩展首部）。
  - 在[RFC 2460]中定义了以下六种扩展首部：

■ 在[RFC 2460]中定义了以下六种扩展首部：

- (1) 逐跳选项
  - (2) 路由选择
  - (3) 分片
  - (4) 鉴别
  - (5) 封装安全有效载荷
  - (6) 目的站选项

- 每一个扩展首部都由若干个字段组成，它们的长度也各不相同。
  - 所有扩展首部中的第一个字段都是8比特的下一个首部字段。该字段的值指出在该扩展首部后面是何种扩展首部。
  - 当使用多个扩展首部时，应按以上的先后顺序出现。

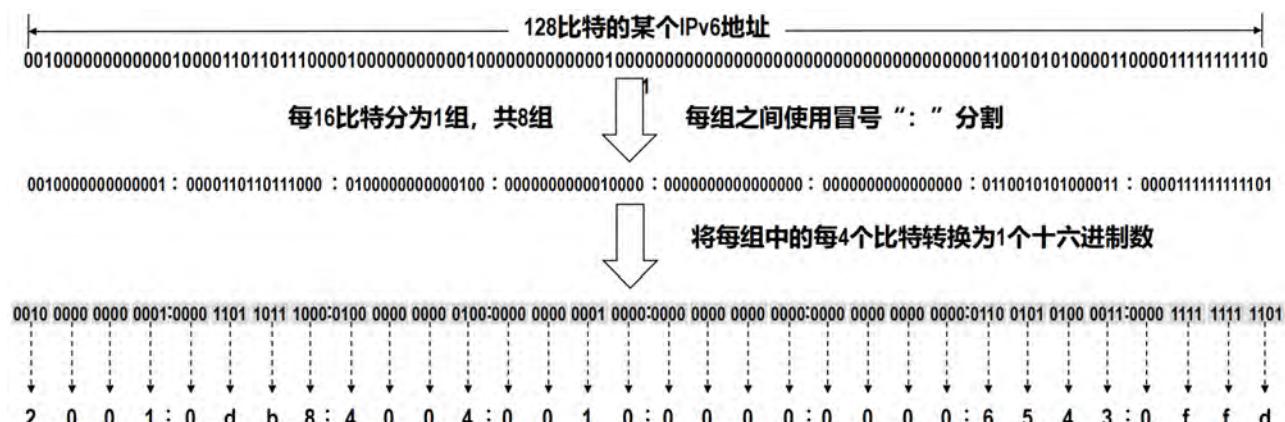
#### 4.9.5. IPv6地址

#### 4.9.5.1. IPv6地址空间大小

- 在IPv6中，每个地址占128个比特。

IPv6地址空间大小为  $2^{128}$  (大于  $3.4 \times 10^{38}$ )

#### 4.9.5.2. IPv6地址的表示方法



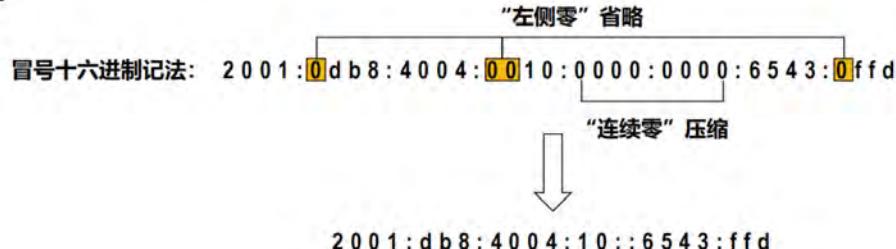
冒号十六进制记法： 2001:0db8:4004:0010:0000:0000:6543:0ffd

**注意：不区分大小写**

■ 在IPv6地址的冒号十六进制记法的基础上，再使用“左侧零”省略和“连续零”压缩，可使IPv6地址的表示更加简洁。

- “左侧零”省略是指两个冒号间的十六进制数中最前面的一串0可以省略不写。
- “连续零”压缩是指一连串连续的0可以用一对冒号取代。

【举例】



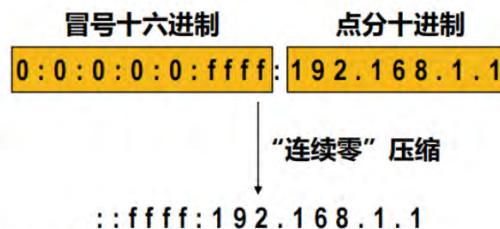
■ 在一个IPv6地址中只能使用一次“连续零”压缩，否则会导致歧义。

【举例】

2001:0000:0000:abcd:0000:0000:0000:1234	}	对每个地址进行多次“连续零”压缩 得到同一个有歧义的地址	2001::abcd::1234
2001:0000:0000:0000:abcd:0000:0000:1234			
2001:0000:abcd:0000:0000:0000:0000:1234			
2001:0000:0000:0000:0000:abcd:0000:1234			
2001:0000:0000:abcd:0000:0000:0000:1234		只使用一次“连续零”压缩，并使用“左侧零”省略	2001:0:abcd::1234
2001:0000:0000:0000:abcd:0000:0000:1234			2001::abcd:0:0:1234
2001:0000:abcd:0000:0000:0000:0000:1234			2001:0:abcd::1234
2001:0000:0000:0000:0000:abcd:0000:1234			2001::abcd:0:1234

■ 冒号十六进制记法还可结合点分十进制的后缀。这在IPv4向IPv6过渡阶段非常有用。

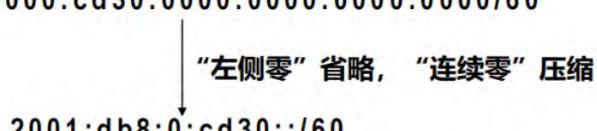
【举例】



■ CIDR的斜线表示法在IPv6中仍然可用。

【举例】

2001:0db8:0000:cd30:0000:0000:0000/60



#### 4.9.5.3. IPv6地址的分类

■ IPv6数据报的**目的地址**有三种基本类型：

单播 (unicast)	传统的点对点通信
多播 (multicast)	一点对多点的通信。数据报发送到一组计算机中的每一个。IPv6没有采用广播的术语，而将广播看作多播的一个特例。
任播 (anycast)	这是IPv6新增的一种类型。任播的 <b>终点是一组计算机，但数据报只交付其中的一个</b> ，通常是按照路由算法得出的距离最近的一个。
未指明地址	128个比特为“全0”的地址，可缩写为两个冒号“::”。 该地址不能用作目的地址，只能用于还没有配置到一个标准IPv6地址的主机用作源地址。 未指明地址只有一个。
环回地址	最低比特为1，其余127个比特为“全0”，即0:0:0:0:0:0:1，可缩写为::1。 该地址的作用与IPv4的环回地址相同。 IPv6的环回地址只有一个。
多播地址	最高8比特为“全1”的地址，可记为FF00::/8。 IPv6多播地址的功能与IPv4多播地址相同。 这类地址占IPv6地址空间的1/256。
本地链路单播地址	最高10比特为1111111010的地址，可记为FE80::/10。即使用户网络没有连接到因特网，但仍然可以使用TCP/IP协议。连接在这种网络上的主机都可以使用本地链路单播地址进行通信，但不能和因特网上的其他主机通信。这类地址占IPv6地址空间的1/1024。
全球单播地址	全球单播地址是使用得最多的一类地址。 IPv6全球单播地址采用三级结构，这是为了使路由器可以更快地查找路由。

The diagram illustrates the structure of an IPv6 global unicast address. It consists of three main fields: a 48-bit Global Routing Prefix, a 16-bit Subnet Identifier, and a 64-bit Interface Identifier. Below the diagram, three callout boxes explain the functions of each field:

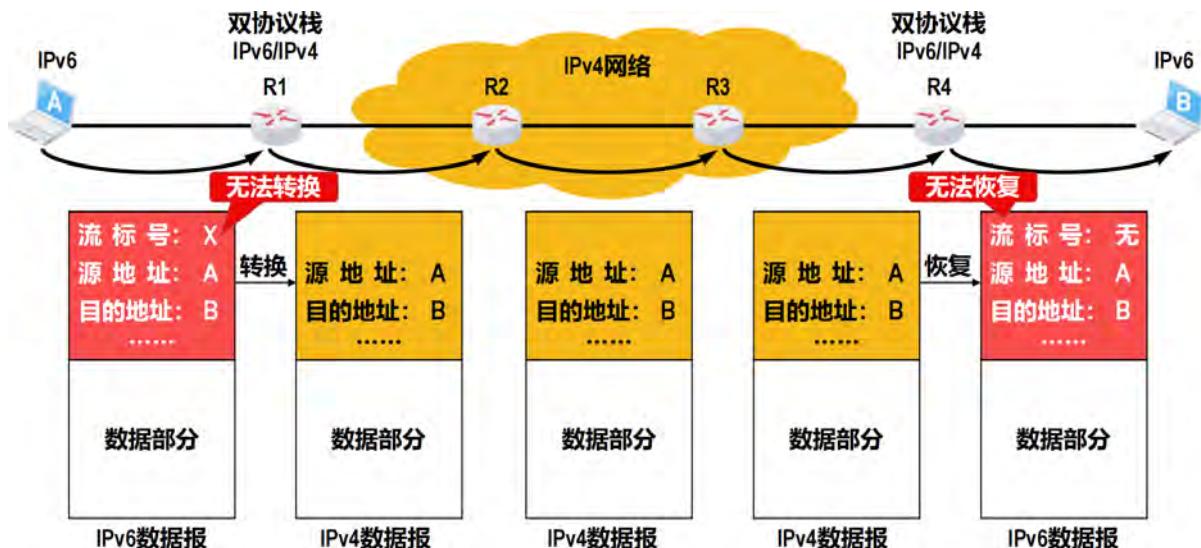
- 分配给公司和机构，用于因特网中路由器的路由选择，相当于IPv4分类地址中的网络号。
- 用于各公司和机构构建自己的子网。
- 用于指明主机或路由器的单个网络接口，相当于IPv4分类地址中的主机号。有64个比特，足以将各种接口的硬件地址直接进行编码，这样就不需要使用ARP。

#### 4.9.6. 从IPv4向IPv6过渡

- 因特网上使用IPv4的路由器的数量太大，要让所有路由器都改用IPv6并不能一蹴而就。因此，从IPv4转变到IPv6只能采用**逐步演进**的办法。
- 另外，新部署的IPv6系统必须能够向后兼容，也就是IPv6系统必须能够接收和转发IPv4数据报，并且能够为IPv4数据报选择路由。

##### 4.9.6.1. 使用双协议栈

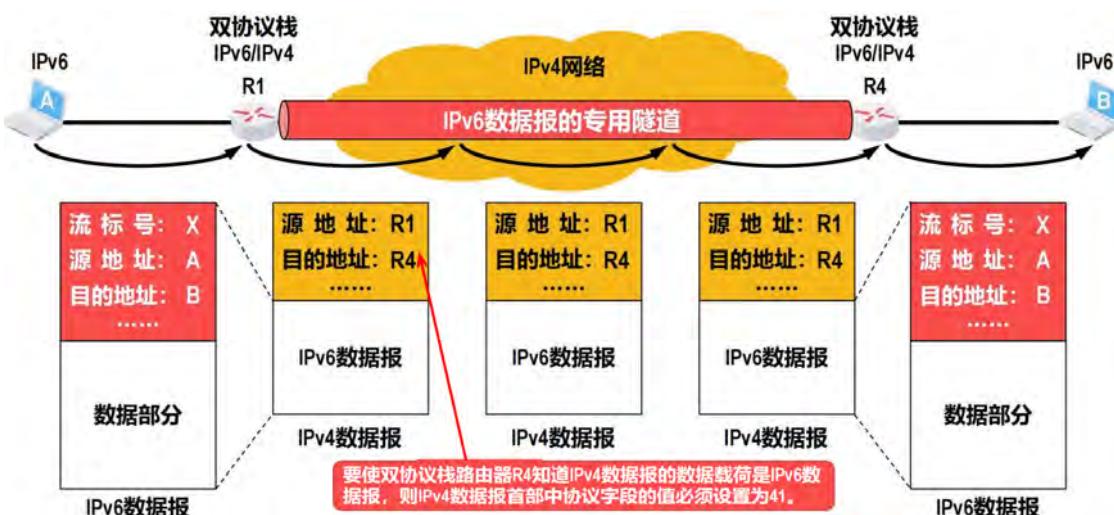
- 双协议栈（Dual Stack）是指在完全过渡到IPv6之前，使一部分主机或路由器装有IPv4和IPv6两套协议栈。
- 双协议栈主机或路由器既可以和IPv6系统通信，又可以和IPv4系统通信。
- 双协议栈主机或路由器记为IPv6/IPv4，表明它具有一个IPv6地址和一个IPv4地址。
  - 双协议栈主机在与IPv6主机通信时采用IPv6地址，而与IPv4主机通信时采用IPv4地址
  - 双协议栈主机通过域名系统DNS查询目的主机采用的IP地址：
    - 若DNS返回的是IPv4地址，则双协议栈的源主机就使用IPv4地址
    - 若DNS返回的是IPv6地址，则双协议栈的源主机就使用IPv6地址



#### 4.9.6.2. 使用隧道技术

- 隧道技术 (Tunneling) 的核心思想是：

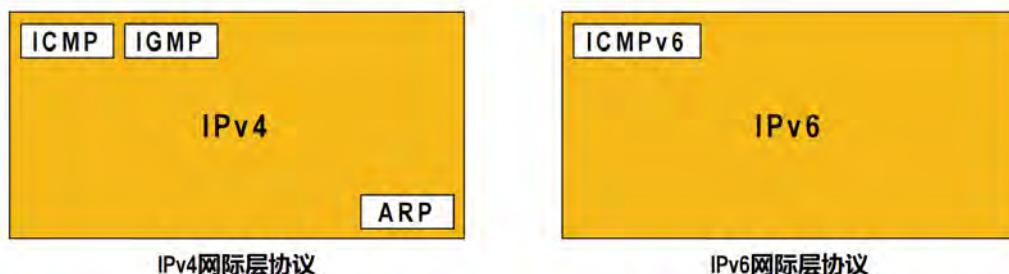
- 当IPv6数据报要进入IPv4网络时，将IPv6数据报重新封装成IPv4数据报，即整个IPv6数据报成为IPv4数据报的数据载荷。
- 封装有IPv6数据报的IPv4数据报在IPv4网络中传输。
- 当IPv4数据报要离开IPv4网络时，再将其数据载荷（即原来的IPv6数据报）取出并转发到IPv6网络。



#### 4.9.7. 网际控制报文协议ICMPv6

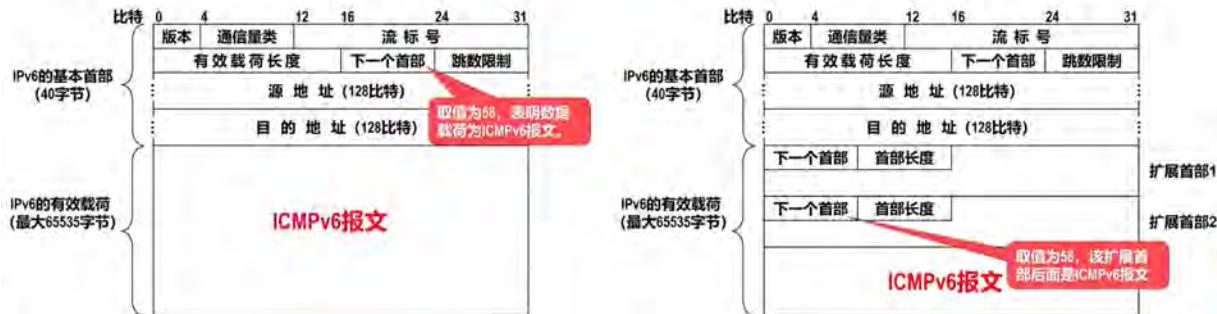
##### 4.9.7.1. 概述

- 由于IPv6与IPv4一样，都不确保数据报的可靠交付，因此IPv6也需要使用网际控制报文协议ICMP来向发送IPv6数据报的源主机反馈一些差错信息，相应的ICMP版本为ICMPv6。
- ICMPv6比ICMPv4要复杂得多，它合并了原来的地址解析协议ARP和网际组管理协议IGMP的功能。因此与IPv6配套使用的网际层协议就只有ICMPv6这一个协议。



### 4.9.7.2. ICMPv6报文的封装

- ICMPv6报文需要封装成IPv6数据报进行发送。



### 4.9.7.3. ICMPv6报文的分类

- ICMPv6报文可被用来报告差错、获取信息、探测邻站或管理多播通信。
- 在对ICMPv6报文进行分类时，不同的RFC文档使用了不同的策略：
  - 在[RFC 2463]中定义了六种类型的ICMPv6报文
  - 在[RFC 2461]中定义了五种类型的ICMPv6报文
  - 在[RFC 2710]中定义了三种类型的ICMPv6报文

常用的几种ICMPv6报文		
ICMP报文种类	类型的值	ICMP报文的类型
差错报告报文	1	目的站不可达
	2	分组太长
	3	时间超过
	4	参数问题
回送请求与回答报文	128	回送请求
	129	回送回答
多播听众发现报文	130	多播听众查询
	131	多播听众报告
	132	多播听众完成
邻站发现报文	133	路由器询问
	134	路由器通告
	135	邻站询问
	136	邻站通告
	137	改变路由

替代原来的IGMP协议      替代原来的ARP协议

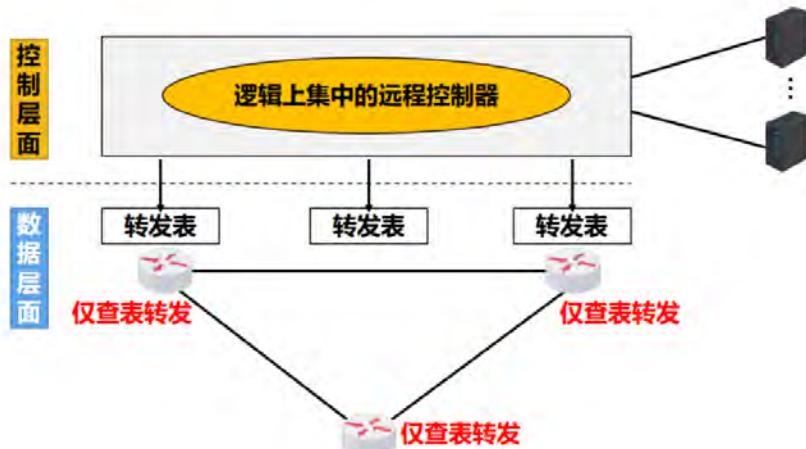
## 4.10. 软件定义网络SDN

### 4.10.1. 概述

- 软件定义网络 (Software Defined Network, SDN) 的概念最早由斯坦福大学的Nick McKeown教授于2009年提出。
- SDN最初只是学术界讨论的一种新型网络体系结构。
- SDN成功案例：谷歌于2010~2012年间建立的数据中心网络B4。
- SDN是当前网络领域最热门和最具发展前途的技术之一，成为近年来的研究热点。

### 4.10.2. 网络层的数据层面和控制层面

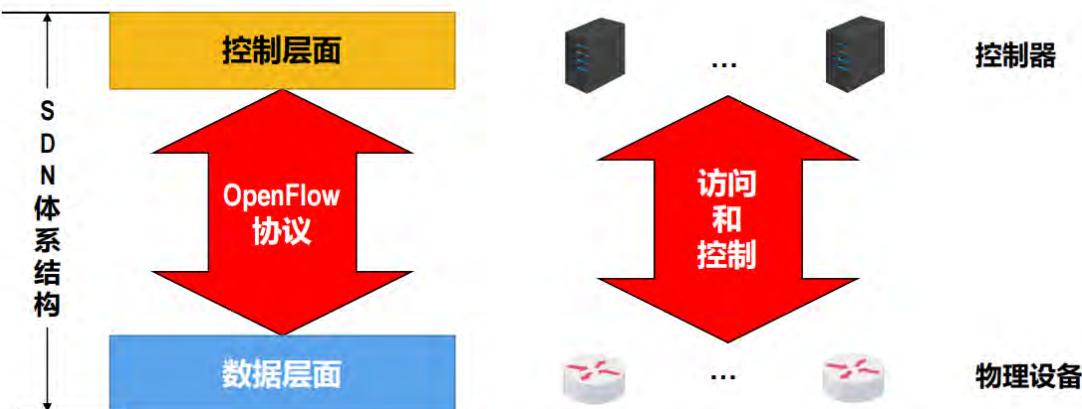
- 在SDN体系结构中，路由器中的路由软件都不存在了。因此，路由器之间不再交换路由信息。
- 在控制层面中，有一个在逻辑上集中的远程控制器。
- 逻辑上集中的远程控制器在物理上可由不同地点的多个服务器组成。
- 远程控制器掌握各主机和整个网络的状态。
- 远程控制器能够为每一个分组计算出最佳的路由。
- 远程控制器为每一个路由器生成其正确的转发表。
- SDN这种新型网络体系结构的核心思想：把网络的控制层面和数据层面分离，而让控制层面利用软件来控制数据层面中的许多设备。



#### 4.10.3. OpenFlow协议

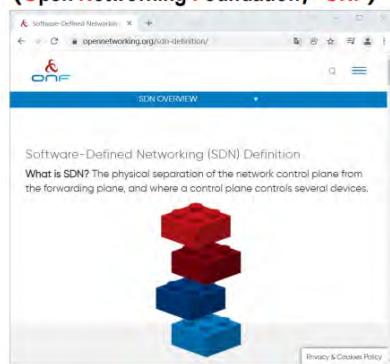
##### 4.10.3.1. 概述

- OpenFlow协议是一个得到高度认可的标准，在讨论SDN时往往与OpenFlow一起讨论。
- OpenFlow协议可被看成是SDN体系结构中控制层面与数据层面之间的通信接口。
- OpenFlow协议使得控制层面的控制器可以对数据层面中的物理设备进行直接访问和控制。



- OpenFlow协议的技术规范由非营利性的产业联盟开放网络基金会 (Open Networking Foundation, ONF) 负责制定。

- ONF的任务是致力于SDN的发展和标准化。
- SDN并未规定必须使用OpenFlow，只不过大部分SDN产品采用了OpenFlow作为其控制层面与数据层面的通信接口。
- OpenFlow从2009年底发表的1.0版开始，每年都被更新，历经12次更新，到2015年3月发布了1.5.1版，目前较为成熟的是1.3版本。



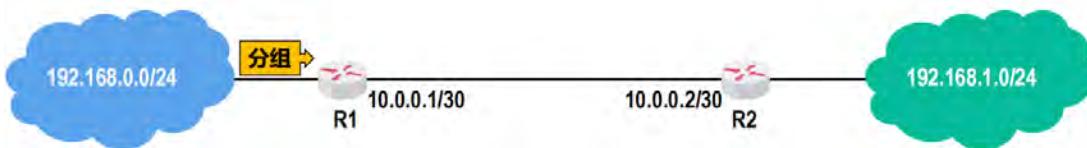
#### 4.10.3.2. 传统意义上的数据层面的任务

- 传统意义上的数据层面的任务：根据转发表转发分组

- 转发分组分为以下两个步骤：

- ① 进行“匹配”：查找转发表中的网络前缀，进行最长前缀匹配。
- ② 执行“动作”：把分组从匹配结果指明的接口转发出去。

R1的路由表		
目的网络	地址掩码	下一跳
192.168.1.0	255.255.255.0	10.0.0.2
...	...	...



#### 4.10.3.3. SDN中的广义转发

- SDN的广义转发分为以下两个步骤：

- ① 进行“匹配”：能够对网络体系结构中各层（数据链路层、网络层、运输层）首部中的字段进行匹配。
- ② 执行“动作”：不仅转发分组，还可以负载均衡、重写IP首部（类似NAT路由器中的地址转换）、人为地阻挡或丢弃一些分组（类似防火墙一样）。

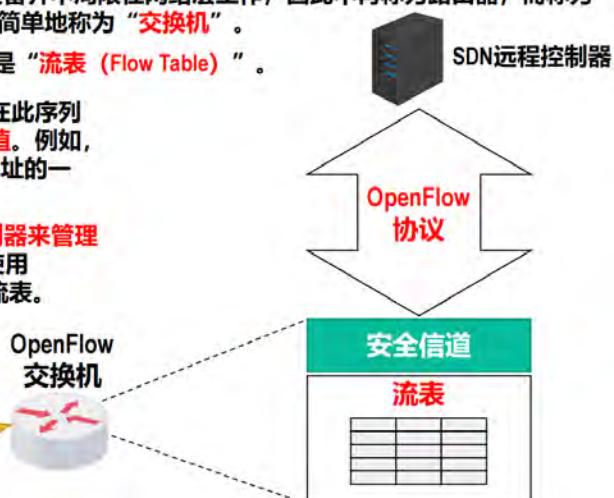
#### 4.10.3.4. OpenFlow交换机和流表

- 在SDN的广义转发中，完成“匹配+动作”的设备并不局限在网络层工作，因此不再称为路由器，而称为“OpenFlow交换机”或“分组交换机”，或更简单地称为“交换机”。

- 相应的，在SDN中取代传统路由器中转发表的是“流表（Flow Table）”。

- 一个流就是穿过网络的一种分组序列，而在此序列中的每个分组都共享分组首部某些字段的值。例如，某个流可以是具有相同源IP地址和目的IP地址的一连串分组。
- OpenFlow交换机中的流表是由SDN远程控制器来管理的。SDN远程控制器通过一个安全信道，使用OpenFlow协议来管理OpenFlow交换机中的流表。

1. 网络设备可以由不同厂商来生产，可以使用在不同类型的网络中。
2. 从SDN远程控制器看到的，是统一的逻辑交换功能。



- 每个OpenFlow交换机必须有一个或多个流表。
- 每一个流表可以包含多个流表项。
- 每个流表项包含三个字段：首部字段值（或称匹配字段）、计数器、动作。

OpenFlow1.0版本的流表		
首部字段值	计数器	动作
首部字段值	计数器	动作
...	...	...
首部字段值	计数器	动作

- 首部字段值字段包含有一组字段，用来使入分组（Incoming Packet）的对应首部与之匹配，因此又称为匹配字段。匹配不上的分组就被丢弃，或被发送到SDN远程控制器做更多的处理。
- 在OpenFlow交换机中，既可以处理数据链路层的帧，也可以处理网际层的IP数据报，还可以处理运输层的TCP或UDP报文。

OpenFlow1.0版本的流表		
首部字段值	计数器	动作
首部字段值	计数器	动作
...	...	...
首部字段值	计数器	动作

首部字段值字段包含11个项目涉及三个层次的首部

入端口	源MAC地址	目的MAC地址	类型	ID	优先级	源IP地址	目的IP地址	协议	服务类型	源端口	目的端口
	以太网			VLAN		IP			TCP/UDP		
	数据链路层					网际层			运输层		

■ 计数器字段是一组计数器：

- 记录已经与该流表项匹配的分组数量的计数器；
- 记录该流表项上次更新到现在经历时间的计数器。

■ 动作字段是一组动作，当分组匹配某个流表项时，执行该流表项中动作字段指明的以下某个或多个动作：

- 把分组转发到指明的端口
- 丢弃分组
- 把分组进行复制后再从多个端口转发出去
- 重写分组的首部字段（包括数据链路层、网际层以及运输层的首部）

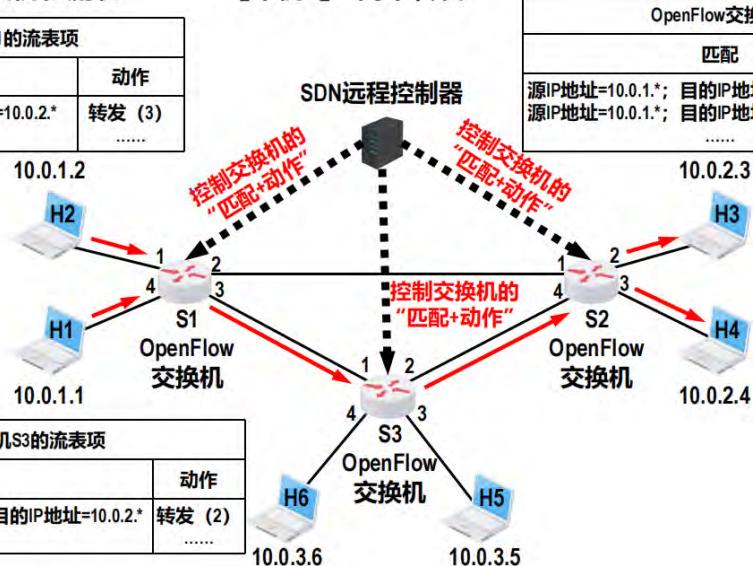
- 简单转发

#### 04 OpenFlow交换机和流表

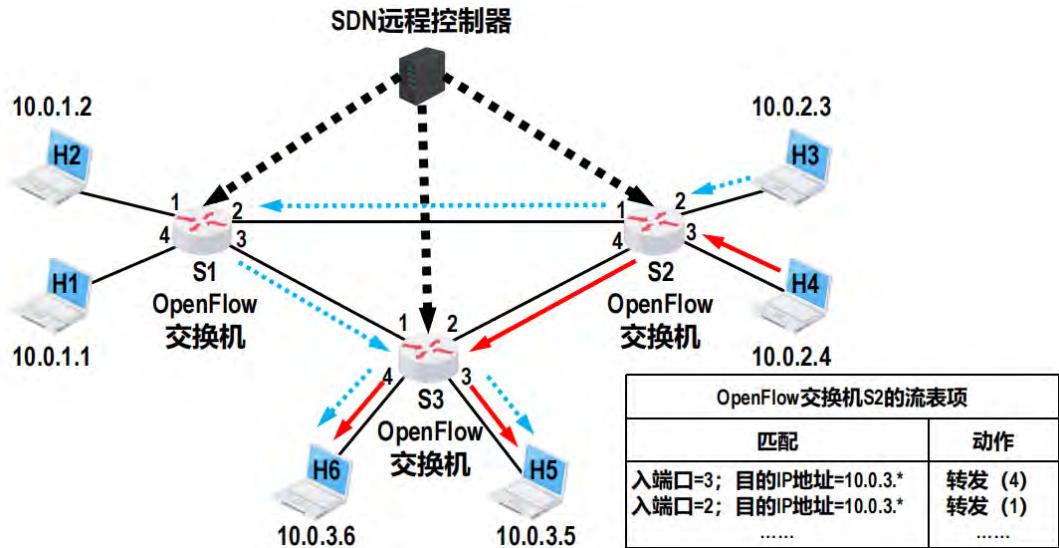
#### 【举例1】简单转发

OpenFlow交换机S1的流表项	
匹配	动作
源IP地址=10.0.1.*； 目的IP地址=10.0.2.*	转发 (3)
.....	.....

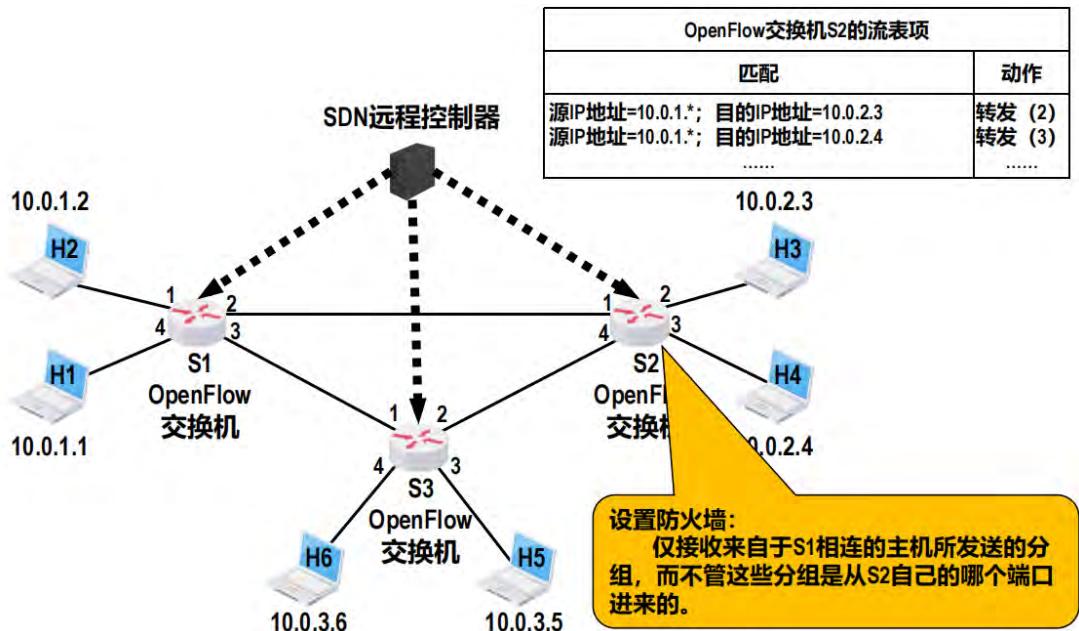
OpenFlow交换机S2的流表项	
匹配	动作
源IP地址=10.0.1.*； 目的IP地址=10.0.2.3	转发 (2)
源IP地址=10.0.1.*； 目的IP地址=10.0.2.4	转发 (3)
.....	.....



- 负载均衡



- 防火墙

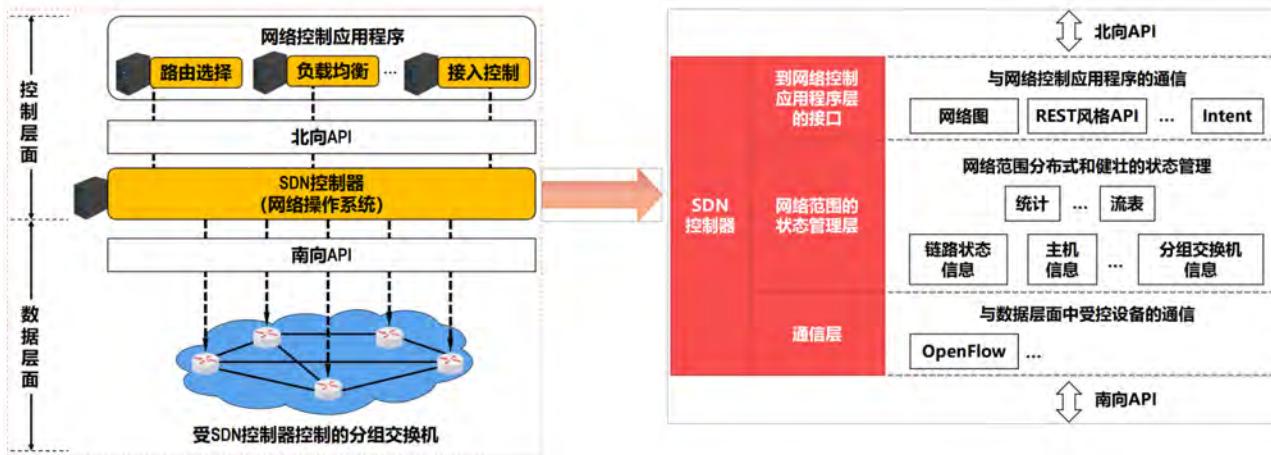


#### 4.10.4. SDN体系结构

##### 4.10.4.1. SDN体系结构及其四个关键特征

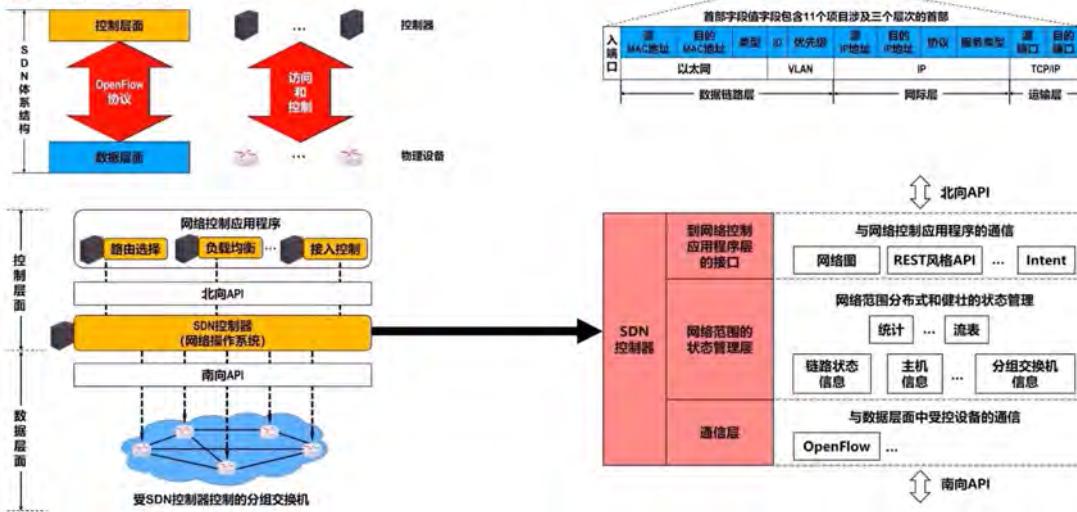
- 基于流的转发
- 数据层面与控制层面分离
- 位于数据层面分组交换机之外的网络控制功能
- 可编程的网络

##### 4.10.4.2. SDN控制器



#### 4.10.5. 总结

- SDN这种新型网络体系结构的核心思想：把网络的控制层面和数据层面分离，而让控制层面利用软件来控制数据层面中的许多设备。
- OpenFlow协议可被看成是SDN体系结构中控制层面与数据层面之间的通信接口。
- 在SDN中取代传统路由器中转发表的是“流表（Flow Table）”。在OpenFlow交换机中，既可以处理数据链路层的帧，也可以处理网际层的IP数据报，还可以处理运输层的TCP或UDP报文。



## 4.11. 题目

### 4.11.1. IPv4分类编址方法

#### 4.11.1.1. 【2017年36题】

【2017年题36】下列IP地址中，只能作为IP分组的源IP地址但不能作为目的IP地址的是（A）。

- A. 0.0.0.0      B. 127.0.0.1      C. 20.10.10.3      D. 255.255.255.255

解析

普通的A类地址  
既可以作为源地址，  
也可以作为目的地址

一般不使用的特殊IPv4地址

	网络号	主机号	IP地址	作为源地址	作为目的地址	表示的意思
选项A	0	0	0.0.0.0	可以	不可以	在本网络上的本主机（例如，DHCP协议）
	0	host-id	0.host-id	可以	不可以	在本网络上的某台主机host-id
选项D	全1	全1	255.255.255.255	不可以	可以	只在本网络上进行广播（各路由器均不转发）
	net-id	全1	A类：net-id.255.255.255 B类：net-id.255.255 C类：net-id.255	不可以	可以	对网络net-id上的所有主机进行广播
选项B	127	非全0或全1的任何数	127.0.0.1~127.255.255.254	可以	可以	用于本地软件环回测试

## 4.11.2. Ipv4划分子网编址方法

### 4.11.2.1. 【2012 39】

【2012年题39】某主机的IP地址为180.80.77.55，子网掩码255.255.252.0，若该主机向其所在子网发送广播分组，则目的地址可以是 (D)。

- A. 180.80.76.0    B. 180.80.76.255    C. 180.80.77.255    D. 180.80.79.255

解析

	网络号	子网号	主机号	
B类网地址	180.80.	01001101.00110111		
主机所在子网的网络地址	180.80.	01001100.00000000		180.80.76.0

找广播地址!

主机所在子网的广播地址	180.80.01001111.11111111	180.80.79.255
-------------	--------------------------	---------------

## 4.11.3. 无分类编址方法

### 4.11.3.1. 【2011 38】

【2011年题38】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 (C)。

- A. 0    B. 1    C. 2    D. 4

解析

192.168.4.0/30		将左起第4个十进制数 转换成二进制形式	30比特网络前缀	2比特主机号
最小地址	作为网络地址	192.168.4.0	192.168.4.0 0 0 0 0 0 0 0 0	0 0
可分配的最小地址		192.168.4.1	192.168.4.0 0 0 0 0 0 0 0 1	0 1
可分配的最大地址		192.168.4.2	192.168.4.0 0 0 0 0 0 0 1 0	1 0
最大地址	作为广播地址	192.168.4.3	192.168.4.0 0 0 0 0 0 0 1 1	1 1

### 4.11.3.2. 【2018 38】

【2018年题38】某路由表有转发接口相同的4条路由表项，其目的网络地址分别为35.230.32.0/21、35.230.40.0/21、35.230.48.0/21和35.230.56.0/21，将这4条路由聚合后的目的网络地址为 (C)。

- A. 35.230.0.0/19    B. 35.230.0.0/20    C. 35.230.32.0/19    D. 35.230.32.0/20

解析

路由聚合的方法：找共同前缀

35.230.32.0/21	将左起第3个十进制数 转换成二进制形式	19比特共同前缀
35.230.40.0/21		35.230.00100000.0
35.230.48.0/21		35.230.00101000.0
35.230.56.0/21		35.230.00110000.0
		35.230.00111000.0

路由聚合后的目的网络地址 35.230.32.0 /19

### 4.11.3.3. 【2021 35】

【2021年题35】现将一个IP网络划分为3个子网，若其中一个子网是192.168.9.128/26，则下列网络中，不可能是另外两个子网之一的是(B)。

- A. 192.168.9.0/25 ✘ B. 192.168.9.0/26 C. 192.168.9.192/26 ✘ D. 192.168.9.192/27 ✘

**解析** ①聚合后可能引入其他地址块

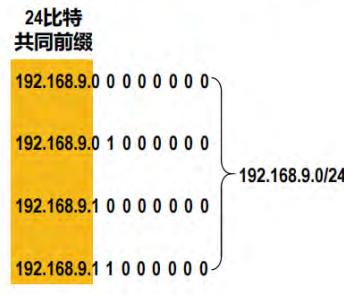
②看选项地址块与题干地址块是否相邻，不相邻则会划分出4个子网而不是三个

选项B的地址块 192.168.9.0

第三个地址块 192.168.9.6

题干给定的地址块 192.168.9.1

第四个地址块 192.168.9.1



#### 4.11.4. IPv4地址的应用规划

#### 4.11.4.1. [2019 37]

**【2019年 题37】若将101.200.16.0/20划分为5个子网，则可能的最小子网的可分配IP地址数量 (B)。**

- A. 126      B. 254      C. 510      D. 1022

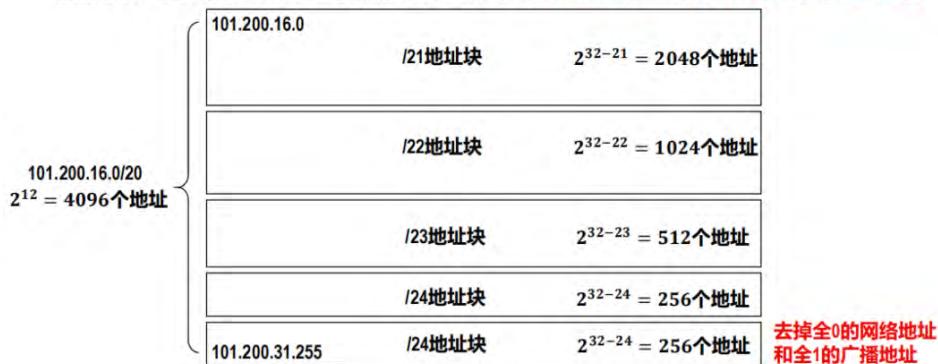
解析

需要使用变长子网划分的方法

**101.200.16.0/20** { 网络前缀: 20比特  
                          主机号: 12比特

$$\text{地址数量 } 2^{12} = 4096$$

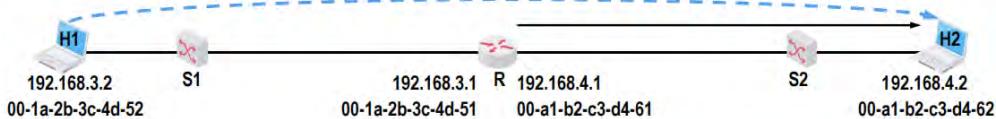
根据题意，需要将这4096个地址划分成5个地址块，其中4个地址块都尽量大，则剩余1个地址块就最小。



#### 4.11.5. 数据报传送过程中IPv4地址与MAC地址的变化情况

【2018年 题37】路由器R通过以太网交换机S1和S2连接两个网络，R的接口、主机H1和H2的IP地址与MAC地址如下图所示。若H1向H2发送一个IP分组P，则H1发出的封装P的以太网帧的目的MAC地址、H2收到的封装P的以太网帧的源MAC地址分别是（D）。

- A. 00-a1-b2-c3-d4-62    00-1a-2b-3c-4d-52    B. 00-a1-b2-c3-d4-62    00-1a-2b-3c-4d-61  
C. 00-1a-2b-3c-4d-51    00-1a-2b-3c-4d-52    D. 00-1a-2b-3c-4d-51    00-a1-b2-c3-d4-61



解析

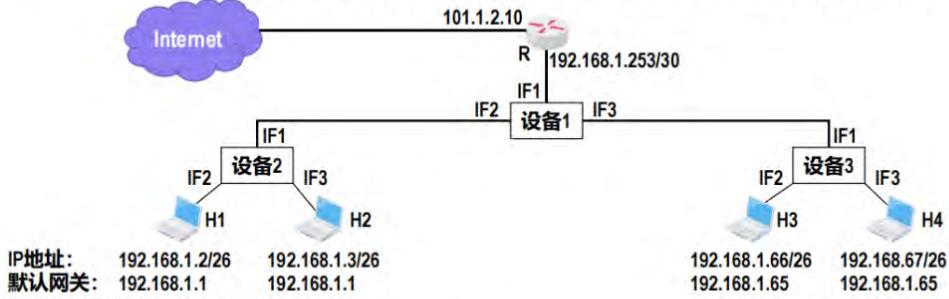
在数据包的传送过程中，源IP地址和目的IP地址保持不变，而源MAC地址和目的MAC地址逐链路（或逐网络）改变。

数据包传输区间	在网络层写入IP数据报首部的IP地址		在数据链路层写入帧首部的MAC地址	
	源IP地址	目的IP地址	源MAC地址	目的MAC地址
H1→R	192.168.3.2	192.168.4.2	00-1a-2b-3c-4d-52	00-1a-2b-3c-4d-51
R→H2	192.168.3.2	192.168.4.2	00-a1-b2-c3-d4-61	00-a1-b2-c3-d4-62

## 4.11.6. IP数据报的发送和转发过程

### 4.11.6.1. 【2019 47】

【2019年 题47】某网络拓扑如下图所示，其中R为路由器，主机H1-H4的IP地址配置以及R的各接口IP地址配置如图中所示。现有若干台以太网交换机（无VLAN功能）和路由器两类网络互连设备可供选择。



请回答以下问题：

- (1) 设备1、设备2和设备3分别应选择什么类型网络设备？
- (2) 设备1、设备2和设备3中，哪几个设备的接口需要配置IP地址？并为对应的接口配置正确的IP地址。
- (3) 若主机H3发送一个目的地址为192.168.1.127的IP数据报，网络中哪几个主机会收到该数据报？

(1) 设备1为路由器，设备2、设备3为交换机。  
(2) 设备1的IF2的ip地址为H1、H2的默认网关192.168.1.1  
设备1的IF3的ip地址为H3、H4的默认网关192.168.1.65  
设备1的IF1的ip地址根据R的接口ip的CIDR形式推出，  
/30用来分配给只有两个路由器接口的点对点链路，故IF1: 192.168.1.253  
(3) 容易发现该目的地址为广播地址，故H4会接收改数据报。

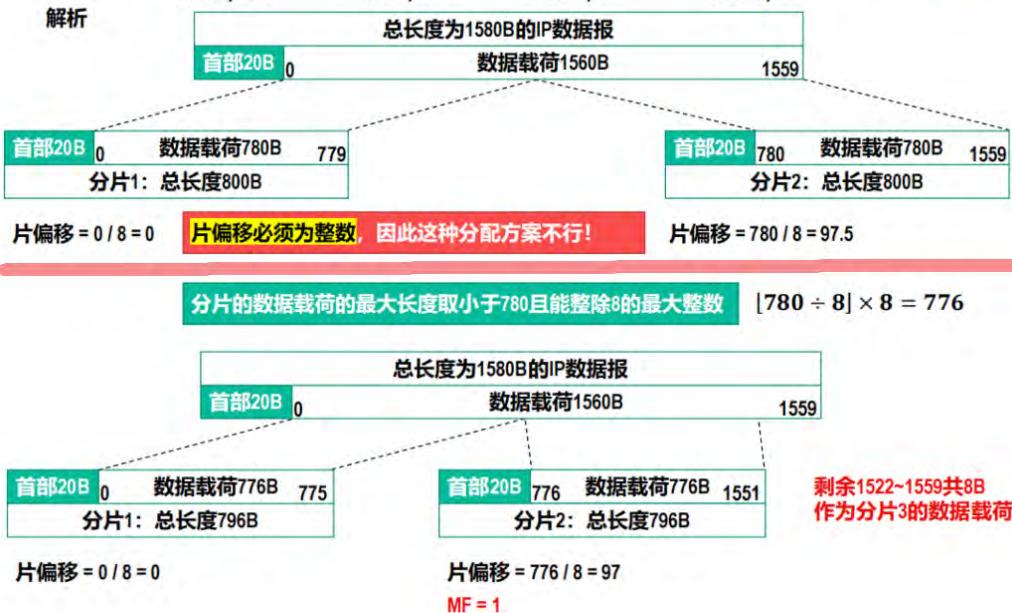
## 4.11.7. 片偏移

### 4.11.7.1. 【2020 36】

【2021年 题36】若路由器向MTU=800B的链路转发一个总长度为1580B的IP数据报（首部长度为20B）时，进行了分片，且每个分片尽可能大，则第2个分片的总长度字段和MF标志位的值分别是（ ）。

- A. 796, 0      B. 796, 1      C. 800, 0      D. 800, 1

解析



## 4.11.8. RIP

### 4.11.8.1. 【2010 35】

【2010年题35】某自治系统内采用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，距离矢量中包含信息<net1,16>，则能得出的结论是 (D)。

- 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

#### 解析

在RIP协议中，**距离16表明目的网络不可达**。

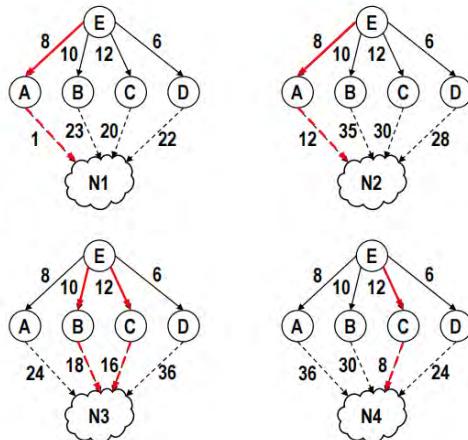
因此，R2无法到达net1，R1也无法通过R2到达net1。

#### 4.11.8.2. 【2021 37】

【2021年题37】某网络中的所有路由器均采用距离向量路由算法计算路由。若路由器E与邻居路由器A、B、C和D之间的直接链路距离分别是8, 10, 12和6，且E收到邻居路由器的距离向量如下表所示，则路由器E更新后的到达目的网络Net1~Net4的距离分别是 (D)。

目的 网络	A的 距离向量	B的 距离向量	C的 距离向量	D的 距离向量
Net1	1	23	20	22
Net2	12	35	30	28
Net3	24	18	16	36
Net4	36	30	8	24

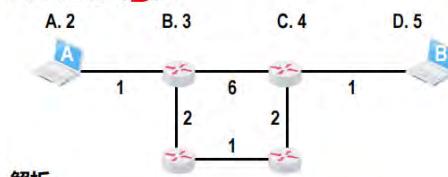
- A. 9, 10, 12, 6
- B. 9, 10, 28, 20
- C. 9, 20, 12, 20
- D. 9, 20, 28, 20



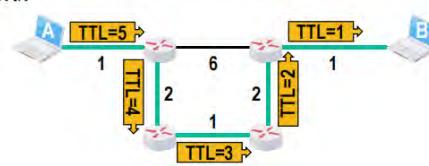
#### 4.11.9. OSPF

##### 4.11.9.1. 【2014 43改】

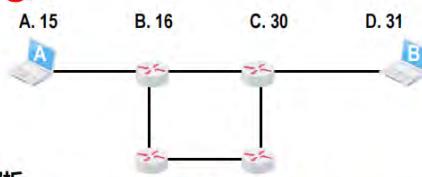
【改自2014年题43】网络拓扑如下图所示，假设各路由器使用OSPF协议进行路由选择且已收敛，各链路的度量已标注在其旁边，主机A给B发送一个IP数据报，为了让IP数据报能够到达主机B，其首部中的TTL字段的取值至少应设置为 (D)。



解析



【举一反三】网络拓扑如下图所示，假设各路由器使用RIP协议进行路由选择且已收敛，主机A给B发送一个IP数据报，其首部中的TTL字段的值设置为32，则当主机B正确接收到该IP数据报时，其首部中的TTL字段的值为 (C)。



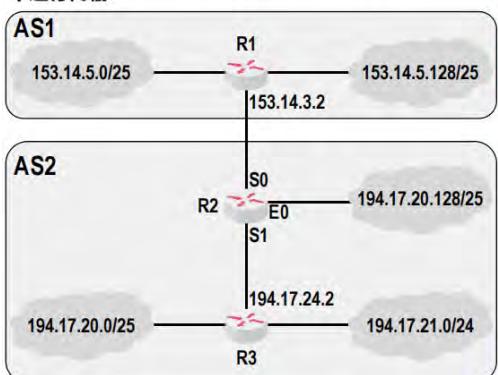
解析



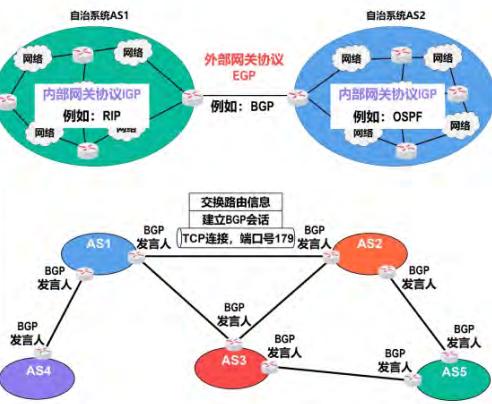
#### 4.11.10. BGP

##### 4.11.10.1. 【2013 46(3)】

【2013年题47（3）】R1与R2之间利用哪个路由协议交换路由信息？该路由协议的报文被封装到哪个协议的分组中进行传输？



解析

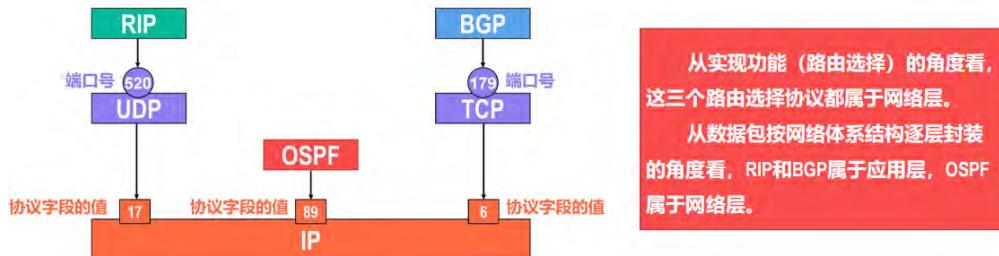


#### 4.11.10.2. 【2017 37】

【2017年题37】直接封装RIP、OSPF、BGP报文的协议分别是（D）。

- A. TCP、UDP、IP      B. TCP、IP、UDP      C. UDP、TCP、IP      D. UDP、IP、CTP

解析



从实现功能（路由选择）的角度看，这三个路由选择协议都属于网络层。

从数据包按网络体系结构逐层封装的角度看，RIP和BGP属于应用层，OSPF属于网络层。

#### 4.11.11. ICMP

##### 4.11.11.1. 【2010 36】

【2010年题36】若路由器R因为拥塞丢弃IP分组，则此时R可向发出该IP分组的源主机发送的ICMP报文类型是（C）。

- A. 路由重定向      B. 目的不可达      C. 源点抑制      D. 超时

#### 4.11.12. SDN题目

【习题1】下列有关SDN的描述中，正确的是（ ）。

- A. SDN是近年来出现的一种新型物理网络  
C. SDN将网络的控制层面和数据层面分开
- B. SDN等同于OpenFlow  
D. OpenFlow交换机就是IP路由器

【习题2】下列有关SDN的描述中，错误的是（ ）。

- A. SDN是近年来出现的一种新型网络体系结构  
C. SDN远程控制器位于OpenFlow交换机中
- B. OpenFlow可被看作是SDN的控制层面与数据层面的通信接口  
D. OpenFlow交换机基于“流表”转发分组

【习题3】下列各种首部中的字段，不能在OpenFlow1.0中匹配的是（ ）。

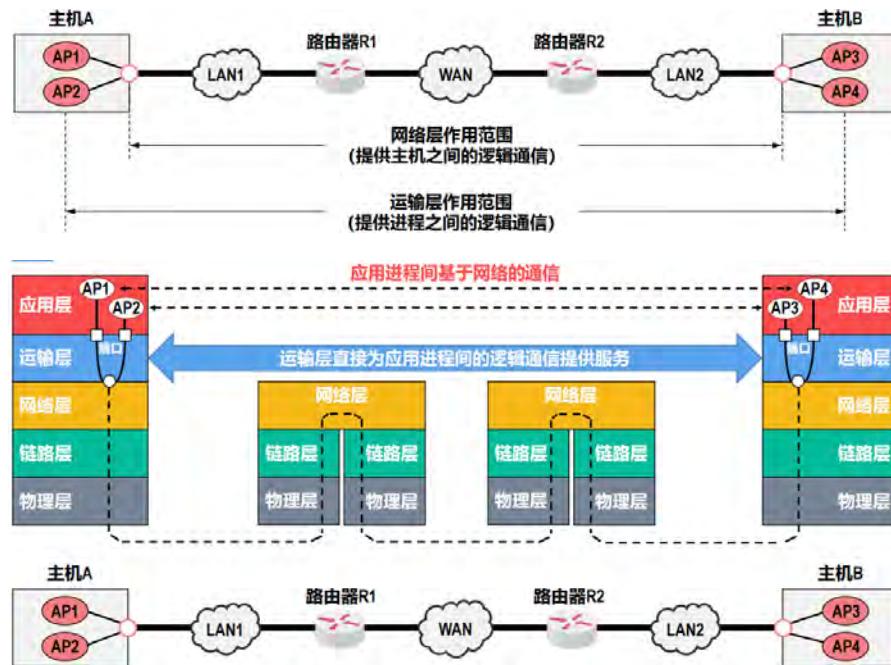
- A. 目的MAC地址  
C. 源IP地址  
D. 窗口

## 5. Transport layer

## 5.1. 运输层概述

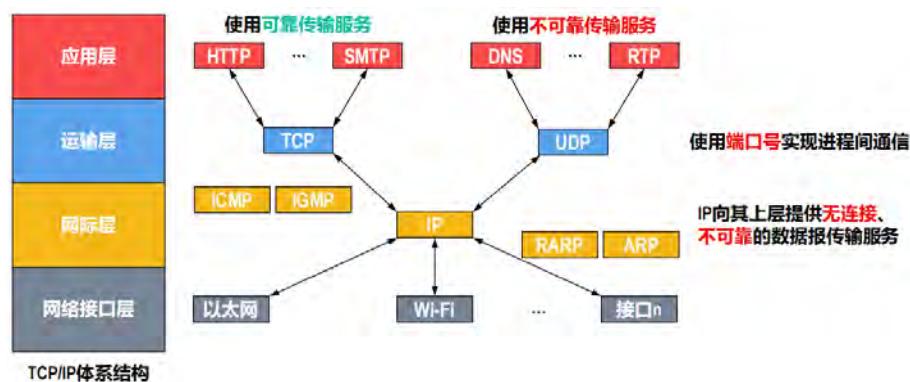
### 5.1.1. 进程间基于网络的通信

- 计算机网络体系结构中的物理层、数据链路层和网络层，它们共同解决了将主机通过异构网络互联起来所面临的问题，实现了主机到主机的通信。
- 计算机网络中实际进行通信的真正实体，是位于通信两端主机中的进程。
- 运输层的主要任务：为运行在不同主机上的应用进程提供直接的逻辑通信服务
- 运输层协议又称为端到端协议。



- 运输层向应用层实体屏蔽了下面网络核心的细节（例如网络拓扑、所采用的路由选择协议等），它使应用进程看见的好像是在两个运输层实体之间有一条端到端的逻辑通信信道。
- 根据应用需求的不同，因特网的运输层为应用层提供了两种不同的运输层协议，即面向连接的TCP和无连接的UDP，这两种协议就是本章要讨论的主要内容。

### 5.1.2. TCP/IP运输层中的两个重要协议



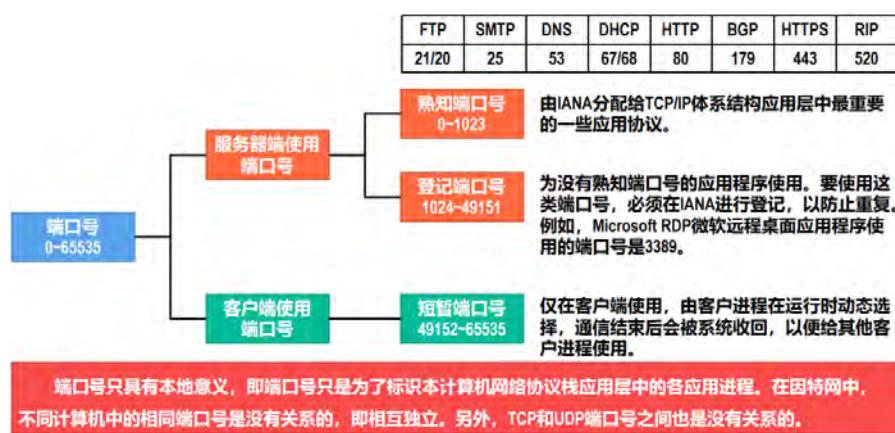
TCP	UDP
<ul style="list-style-type: none"> <li>● 传输控制协议 (Transmission Control Protocol, TCP) 为其上层提供的是面向连接的可靠的数据传输服务。</li> <li>● 使用TCP通信的双方，在传送数据之前必须首先建立TCP连接（逻辑连接，而非物理连接）。数据传输结束后必须要释放TCP连接。</li> <li>● TCP为了实现可靠传输，就必须使用很多措施，例如TCP连接管理、确认机制、超时重传、流量控制以及拥塞控制等。</li> <li>● TCP的实现复杂，TCP报文段的首部比较大，占用处理机资源比较多。</li> </ul>	<ul style="list-style-type: none"> <li>● 用户数据报协议 (User Datagram Protocol, UDP) 为其上层提供的是无连接的不可靠的数据传输服务。</li> <li>● 使用UDP通信的双方，在传送数据之前不需要建立连接。</li> <li>● UDP不需要实现可靠传输，因此不需要使用实现可靠传输的各种机制。</li> <li>● UDP的实现简单，UDP用户数据报的首部比较小。</li> </ul>

因特网中的一些典型应用所使用的TCP/IP应用层协议和相应的运输层协议

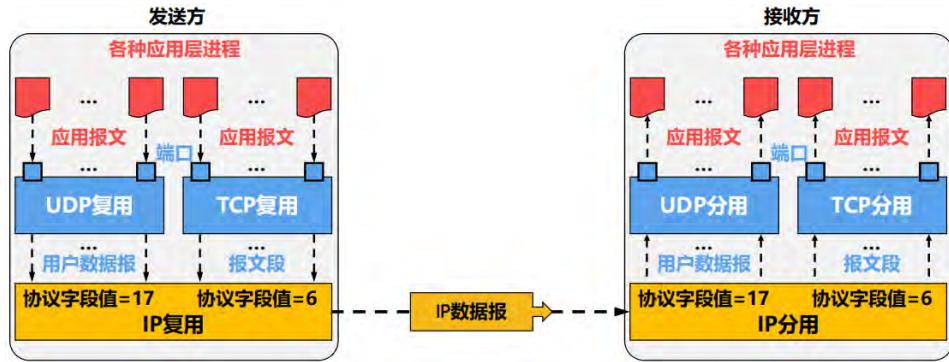
因特网应用	TCP/IP应用层协议	TCP/IP运输层协议
域名解析	域名系统DNS	UDP
文件传送	简单文件传送协议TFTP	UDP
路由选择	路由信息协议RIP	UDP
网络参数配置	动态主机配置协议DHCP	UDP
网络管理	简单网络管理协议SNMP	UDP
远程文件服务器	网络文件系统NFS	UDP
IP电话	专用协议	UDP
流媒体通信	专用协议	UDP
IP多播	网际组管理协议IGMP	UDP
电子邮件	简单邮件传送协议SMTP	TCP
远程终端接入	电传机网络TELNET	TCP
万维网	超文本传送协议HTTP	TCP
文件传送	文件传送协议FTP	TCP

### 5.1.3. 运输层端口号

- 运行在计算机上的进程是使用进程标识符 (Process Identification, PID) 来标识的。
  - 不同操作系统 (Windows、Linux、MacOS) 又使用不同格式的进程标识符。
  - 为了使运行不同操作系统的计算机的应用进程之间能够基于网络进行通信，就必须使用统一的方法对TCP/IP体系的应用进程进行标识。
- TCP/IP体系结构的运输层使用端口号来标识和区分应用层的不同应用进程。
- 端口号的长度为16比特，取值范围是0~65535。

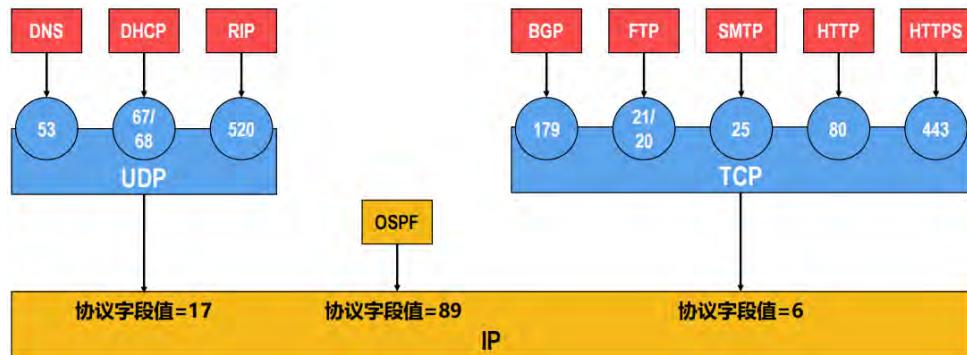


### 5.1.4. 发送方的复用和接收方的分用

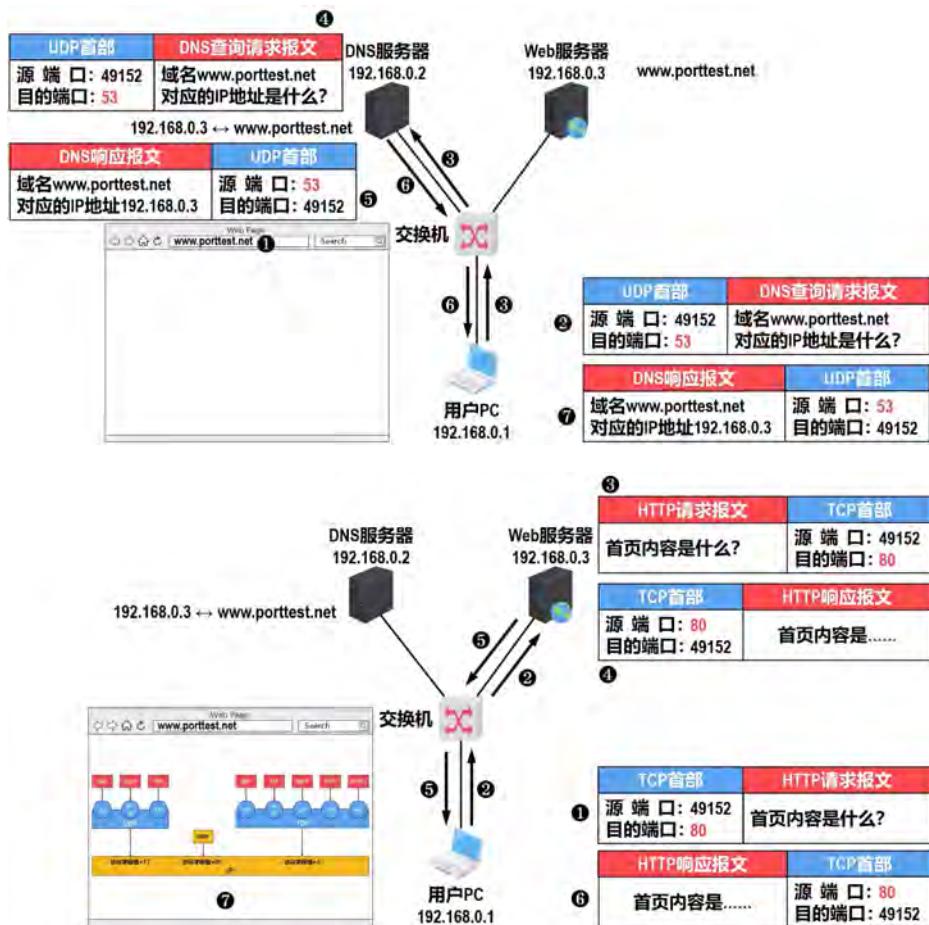


#### 5.1.4.1. TCP/IP体系结构应用层常用协议所使用的运输层协议和熟知端口号

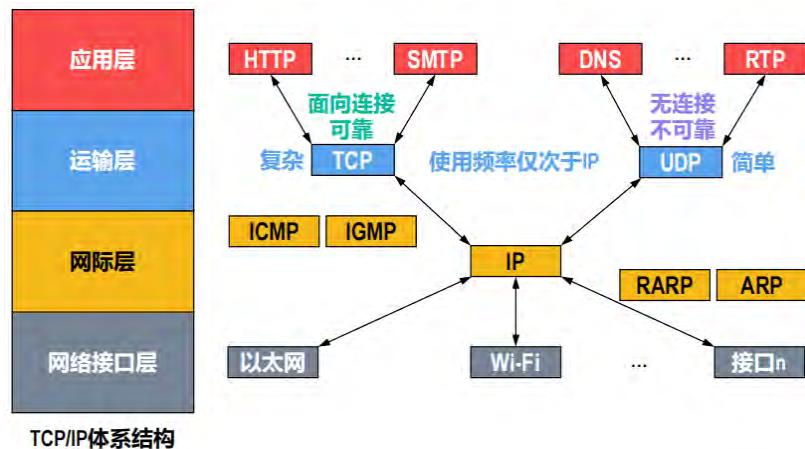
- OSPF报文并不使用运输层的UDP或TCP进行封装，而是直接使用网际层的IP进行封装。



### 5.1.5. 运输层端口号应用举例



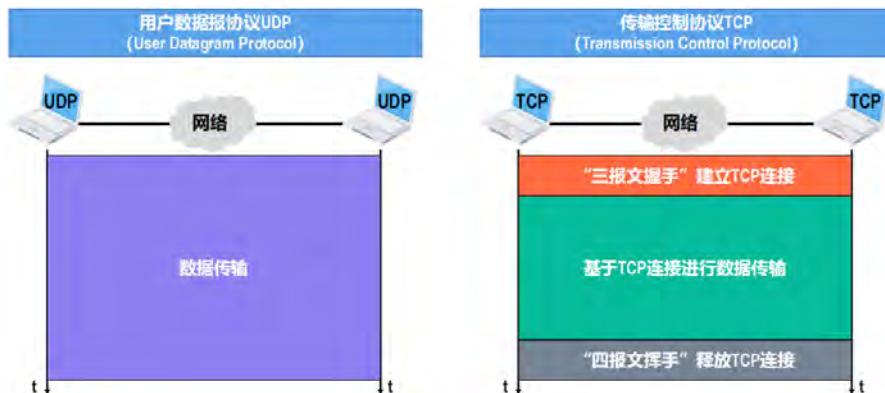
## 5.2. UDP和TCP的对比



UDP	TCP
无连接	面向连接
支持“一对一”、“一对多”、“多对一”和“多对多”交互通信。	每一条TCP连接只能有两个端点EP，只能是一对一通信。
面向应用报文	面向字节流
尽最大努力交付，即不可靠；不使用流量控制和拥塞控制。	可靠传输，使用流量控制和拥塞控制。
首部开销小，仅8字节。	首部最小20字节，最大60字节。

### 5.2.1. 无连接的UDP和面向连接的TCP

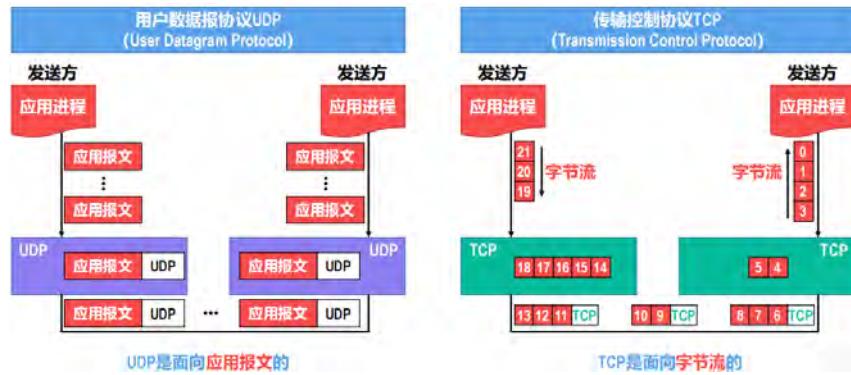
- 这里的“连接”指的是逻辑链接关系，不是物理连接。



### 5.2.2. 对单播、多播和广播的支持情况



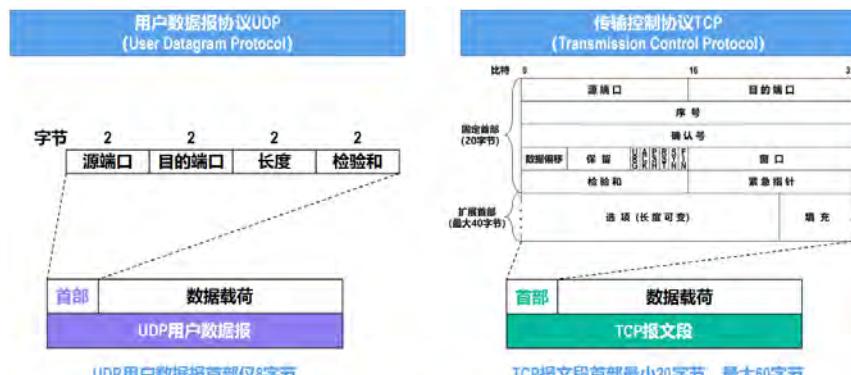
### 5.2.3. 对应用层报文的处理



### 5.2.4. 对数据传输可靠性的支持情况



### 5.2.5. 首部对比



## 5.3. 传输控制协议

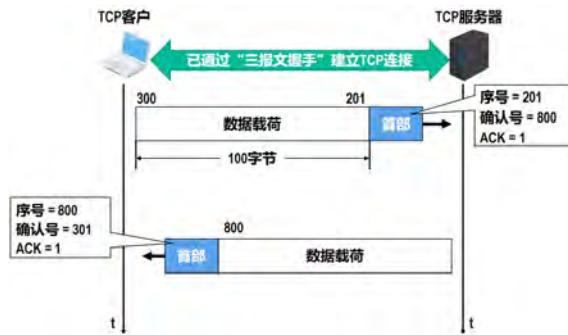
### 5.3.1. TCP报文段的首部格式



### 5.3.1.1. 序号、确认号、确认标志位ACK

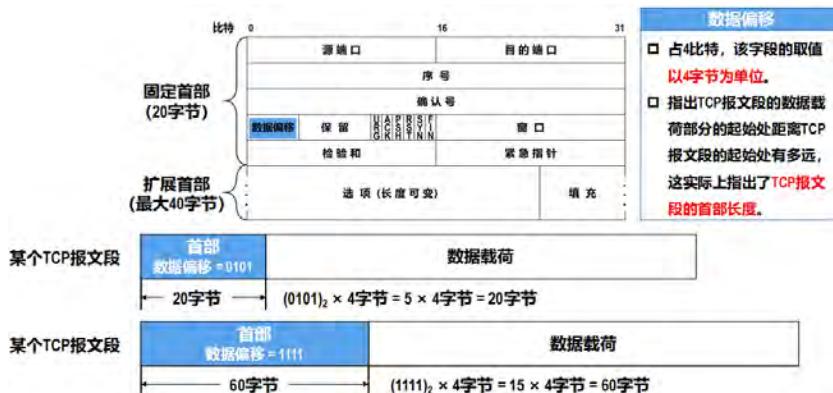
序号	占32比特，取值范围0~ $2^{32}-1$ 。当序号增加到最后一个时，下一个序号又回到0。用来指出本TCP报文段数据载荷的第一个字节的序号。
确认号	占32比特，取值范围0~ $2^{32}-1$ 。当确认号增加到最后一个时，下一个确认号又回到0。用来指出期望收到对方下一个TCP报文段的数据载荷的第一个字节的序号，同时也是对之前收到的所有数据的确认。
确认标志位ACK	只有当ACK取值为1时，确认号字段才有效。ACK取值为0时，确认号字段无效。 TCP规定：在TCP连接建立后，所有传送的TCP报文段都必须把ACK置1。

- 举例



- 确认号应该是已接收且按序到达的最后一次的数据载荷的第一个字节序号。（连续发送中间有丢失情况）

### 5.3.1.2. 数据偏移

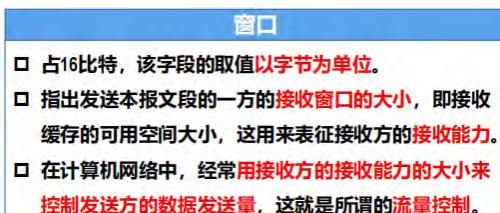


### 5.3.1.3. 保留

- 占6比特
- 保留为今后使用
- 目前应置为0

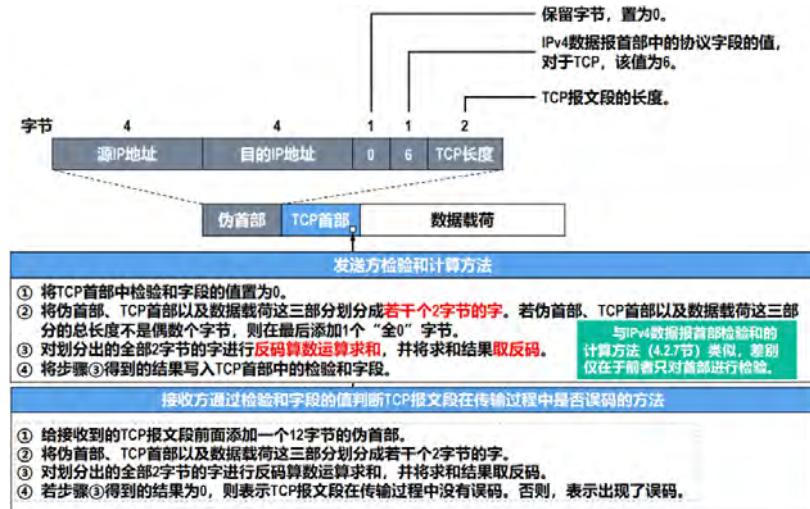
### 5.3.1.4. 窗口

- 取值范围 $[0, 2^{16} - 1]$



### 5.3.1.5. 检验和

- 占16比特
- 用来检查整个TCP报文段在传输过程中是否出现了误码。
- 与UDP类似，在计算检验和时，要在TCP报文段前面加上12字节的伪首部



- 对比UDP



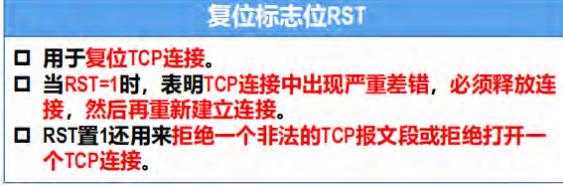
### 5.3.1.6. 同步标志位SYN



### 5.3.1.7. 终止标志位FIN



### 5.3.1.8. 复位标志位RST



### 5.3.1.9. 推送标志位PSH

推送标志位PSH
<ul style="list-style-type: none"> <li><input type="checkbox"/> 出于效率的考虑，TCP的发送方可能会延迟发送数据，而TCP的接收方可能会延迟向应用进程交付数据。这样可以一次处理更多的数据。</li> <li><input type="checkbox"/> 但是当两个应用进程进行交互式通信时，有时在一端的应用进程希望在插入一个命令后立即就能收到对方的响应。在这种情况下，应用进程可以通知TCP使用推送（PUSH）操作。</li> </ul>

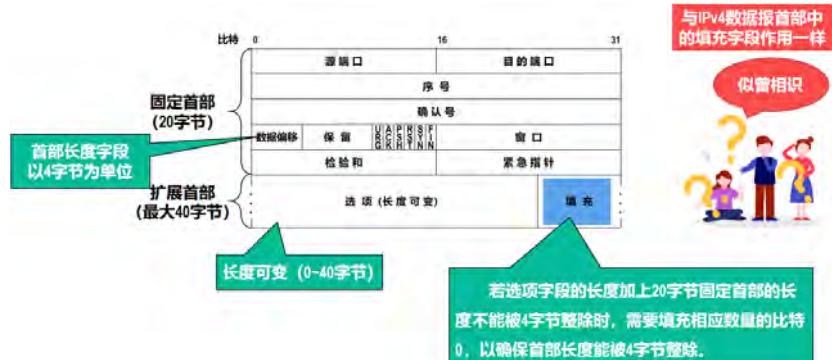
### 5.3.1.10. 紧急标志位URG、紧急指针

紧急标志位URG	紧急指针
<ul style="list-style-type: none"> <li><input type="checkbox"/> 当URG=1时，紧急指针字段有效。</li> <li><input type="checkbox"/> 当URG=0时，紧急指针字段无效。</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> 占16比特，以字节为单位，用来指明紧急数据的长度。</li> <li><input type="checkbox"/> 当发送方有紧急数据时，可将紧急数据“捕获”到发送缓存的最前面，并立刻封装到一个TCP报文段中进行发送。紧急指针会指出本报文段数据载荷部分包含了多长的紧急数据，紧急数据之后是普通数据。</li> <li><input type="checkbox"/> 接收方收到紧急标志位为1的TCP报文段，会按照紧急指针字段的值从报文段数据载荷中取出紧急数据并直接上交应用进程，而不必在接收缓存中排队。</li> </ul>

### 5.3.1.11. 选项（长度可变，最大40字节）

选项（长度可变，最大40字节）
<ul style="list-style-type: none"> <li><input type="checkbox"/> 最大报文段长度MSS选项：指出TCP报文段数据载荷部分的最大长度，而不是整个TCP报文段的长度。</li> <li><input type="checkbox"/> 窗口扩大选项：用来扩大窗口，提高吞吐率。</li> <li><input type="checkbox"/> 时间戳选项： <ul style="list-style-type: none"> <li>• 用于计算往返时间RTT</li> <li>• 用于处理序号超范围的情况，又称为防止序号绕回PAWS。</li> </ul> </li> <li><input type="checkbox"/> 选择确认选项：用来实现选择确认功能。</li> </ul>

### 5.3.1.12. 填充



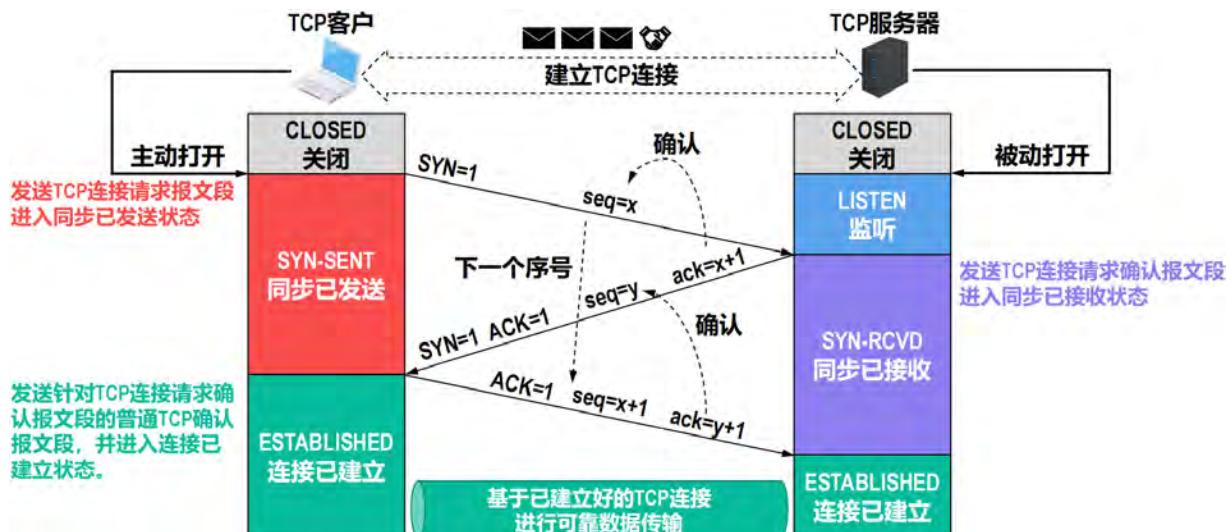
### 5.3.2. TCP的运输层连接管理

- **TCP是面向连接的协议，它基于运输连接来传送TCP报文段。**
- TCP运输连接的建立和释放，是每一次面向连接的通信中必不可少的过程。
- TCP运输连接有以下三个阶段：
  - ① 通过“三报文握手”来建立TCP连接。
  - ② 基于已建立的TCP连接进行可靠的数据传输。
  - ③ 在数据传输结束后，还要通过“四报文挥手”来释放TCP连接。

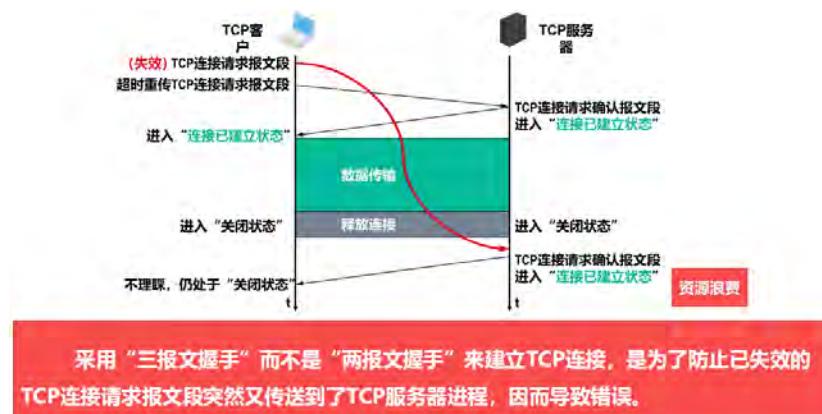


### 5.3.2.1. “三报文握手” 建立连接

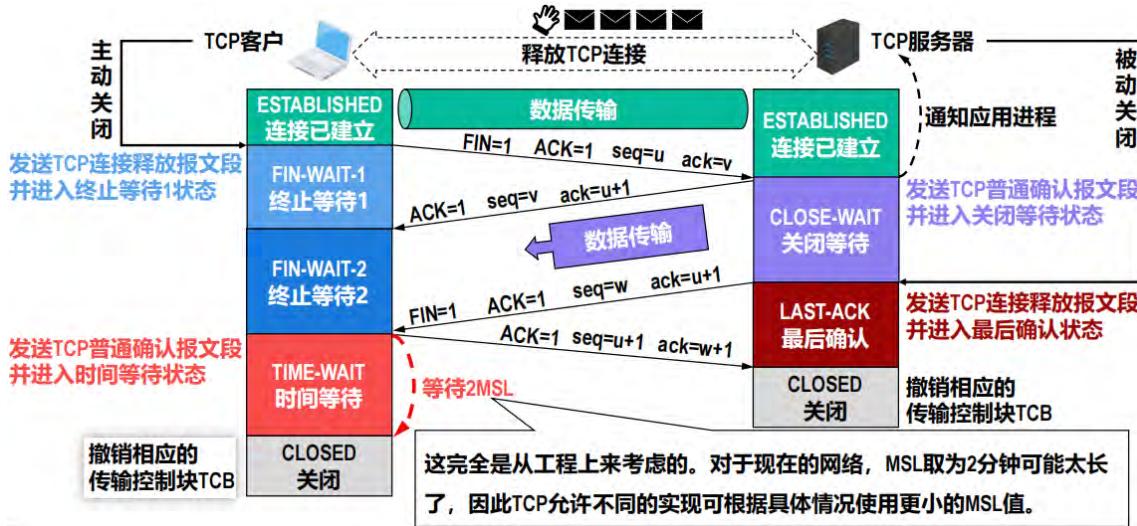
- 三报文握手”建立TCP连接的目的
  - 使TCP双方能够确知对方的存在。
  - 使TCP双方能够协商一些参数
    - 例如最大报文段长度、最大窗口大小、时间戳选项等。
    - 使TCP双方能够对运输实体资源进行分配和初始化。
      - 运输实体资源包括缓存大小、各状态变量、连接表中的项目等。



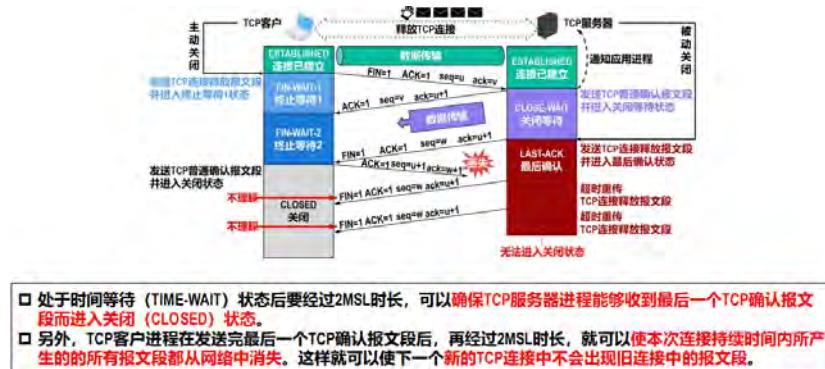
- “三报文”而不是“两报文”



### 5.3.2.2. “四报文挥手” 释放连接



- 等待2MSL原因

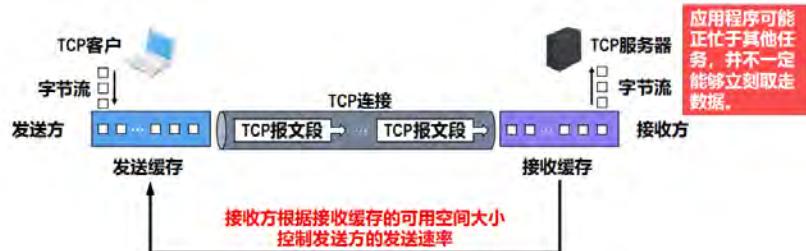


- TCP保活计时器的作用



### 5.3.3. TCP的流量控制

#### 5.3.3.1. 基本概念



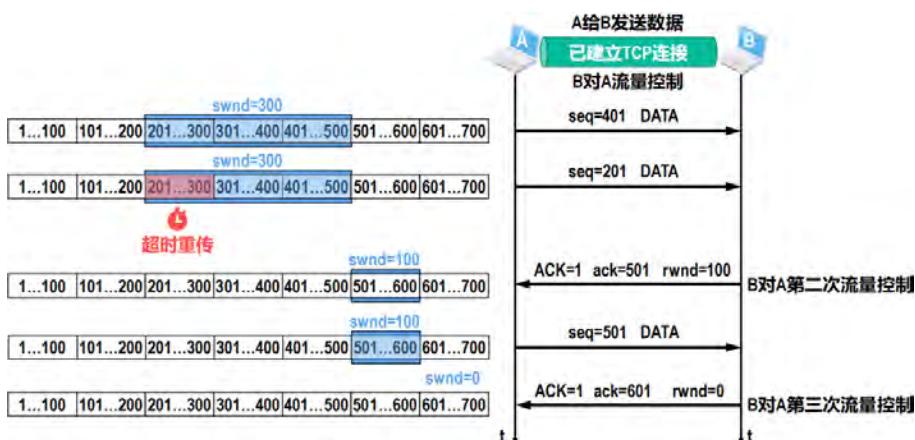
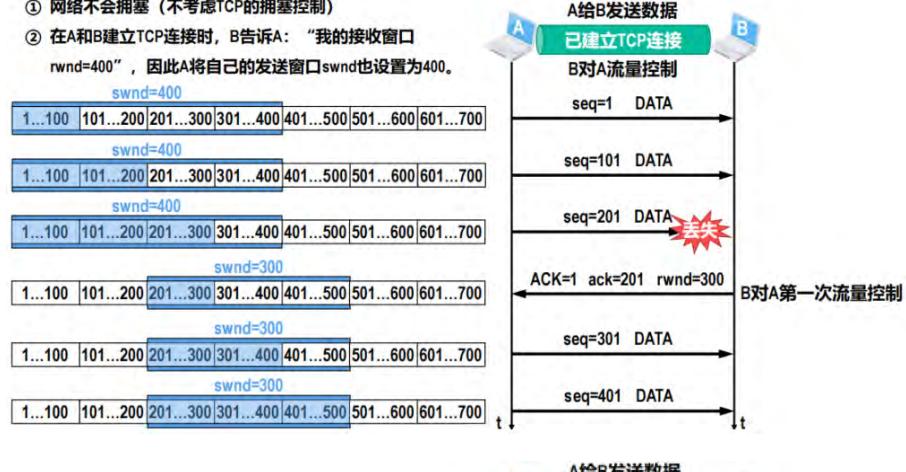
■ TCP为应用程序提供了流量控制(Flow Control)机制,以解决因发送方发送数据太快而导致接收方来不及接收,造成接收方的接收缓存溢出的问题。

**流量控制的基本方法：**接收方根据自己的接收能力（接收缓存的可用空间大小）控制发送方的发送速率。

### 5.3.3.2. 控制方法

假设：

- ① 网络不会拥塞（不考虑TCP的拥塞控制）
  - ② 在A和B建立TCP连接时，B告诉A：“我的接收窗口 rwnd=400”，因此A将自己的发送窗口swnd也设置为400。



为了打破由于非零窗口通知报文段丢失而引起的双方互相等待的死锁局面，TCP为每一个连接都设有一个持

- 口 只要TCP连接的一方收到对方的零窗口通知，就启动持续计时器。
  - 口 当持续计时器超时时，就发送一个零窗口探测报文段，仅携带1字节的数据。
  - 口 对方在确认这个零窗口探测报文段时，给出自己现在的接收窗口值。
  - 口 如果接收窗口值仍然是0，那么收到这个报文段的一方就重新启动持续计时器。
  - 口 如果接收窗口值不是0，那么死锁的局面就可以被打破了。

A一直等待B发送的  
非零窗口通知

**如果不采取措施，这种互相等待而形成的死锁局面将一直持续下去！**

B一直等待A发送的数据

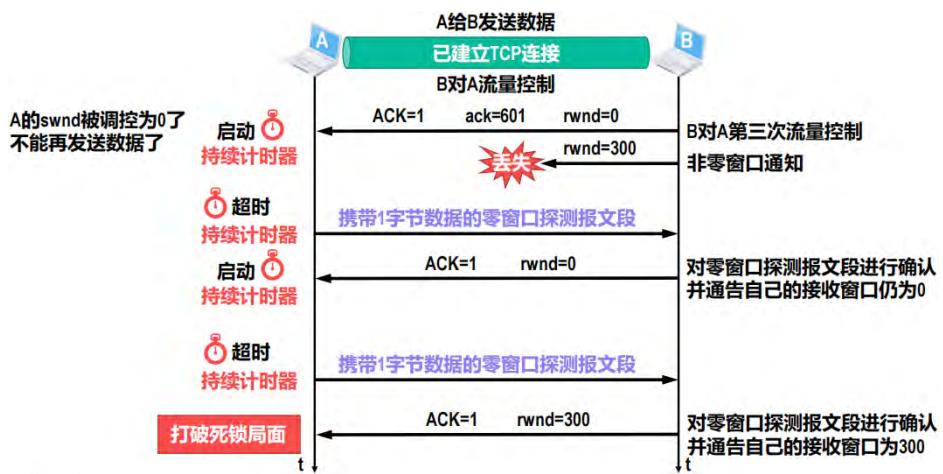
R—直等待A发

B一直等待A发送的数据报文段

100

相等待而形成的死锁局面  
将一直持续下去！

2000-01

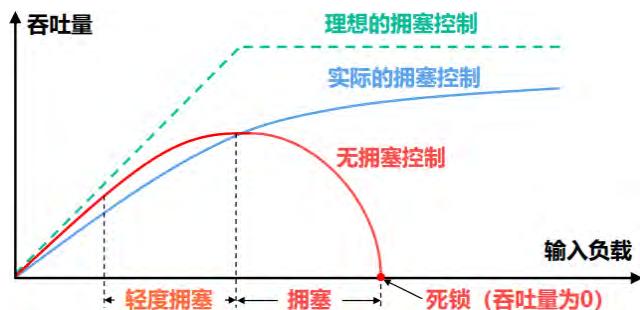


- A发送的零窗口探测报文段到达B时，如果B此时的接收窗口值仍然为0，那么B根本就无法接受该报文段，又怎么会针对该报文段给A发回确认呢？
  - 实际上TCP规定：即使接收窗口值为0，也必须接受零窗口探测报文段、确认报文段以及携带有紧急数据的报文段。
- 如果零窗口探测报文段丢失了，还会打破死锁的局面吗？
  - 回答是肯定的。因为零窗口探测报文段也有重传计时器，当重传计时器超时后，零窗口探测报文段会被重传。

#### 5.3.4. TCP的拥塞控制

##### 5.3.4.1. 基本概念

- 拥塞 (congestion)
  - 在某段时间，若对网络中某一资源的需求超过了该资源所能提供的可用部分，网络性能就要变坏的情况。
- 计算机网络中的链路容量（带宽）、交换节点中的缓存和处理机等都是网络的资源。
- 若出现拥塞而不进行控制，整个网络的吞吐量将随输入负载的增大而下降。

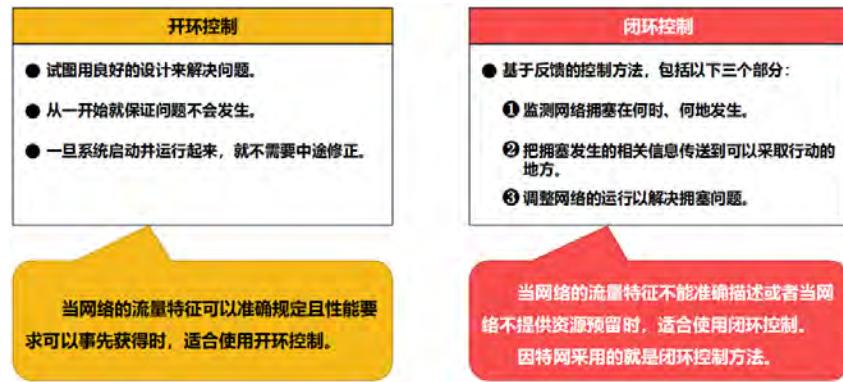


##### 5.3.4.2. 基本方法

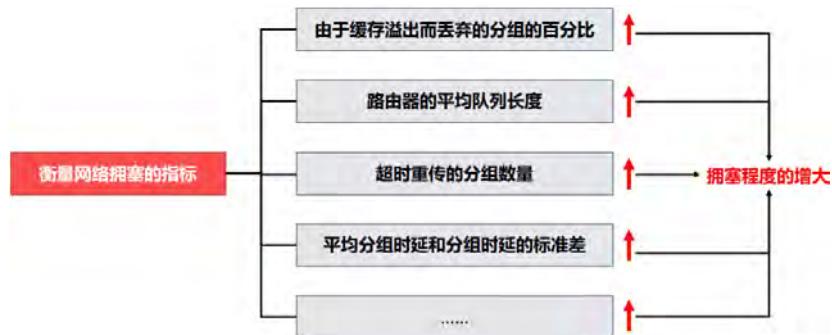
###### 5.3.4.2.1. 对比流量控制



### 5.3.4.2.2. 开环控制与闭环控制



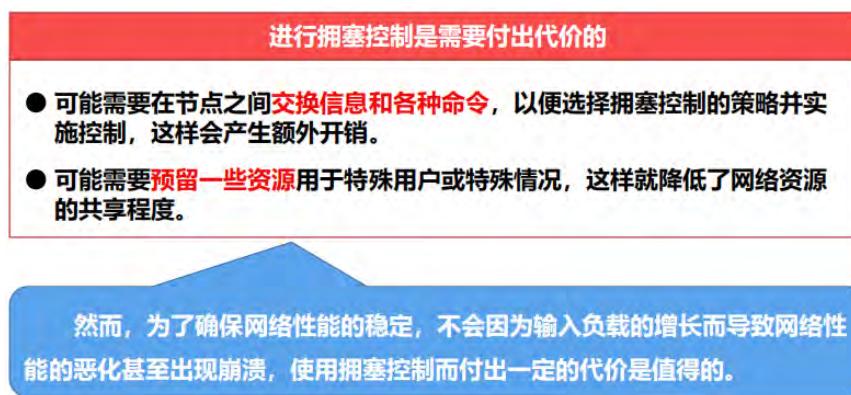
### 5.3.4.2.3. 衡量网络拥塞的指标



### 5.3.4.2.4. 闭环拥塞控制算法



### 5.3.4.2.5. 代价

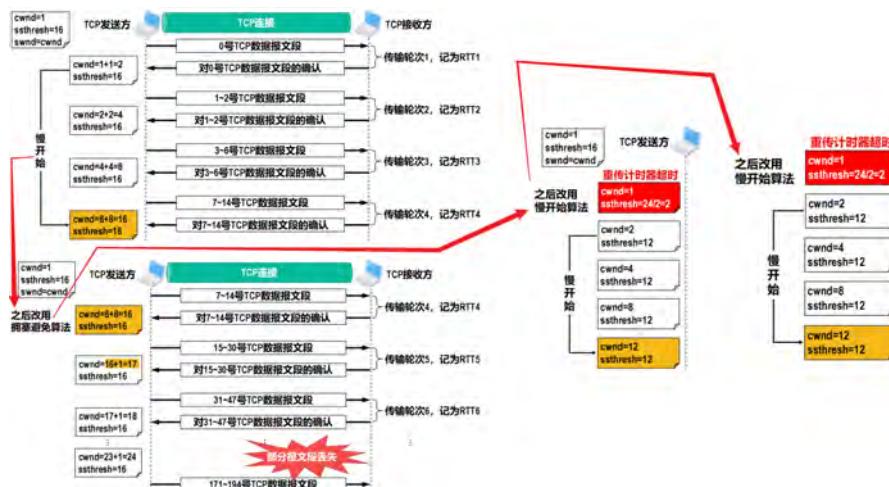
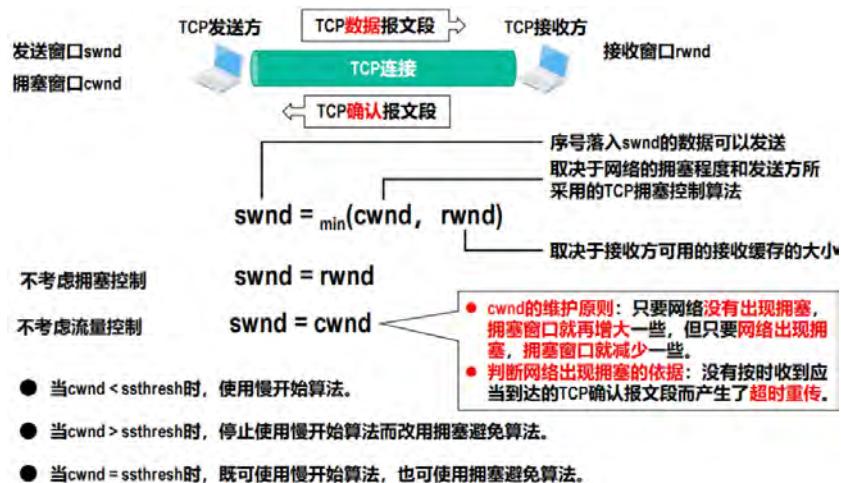


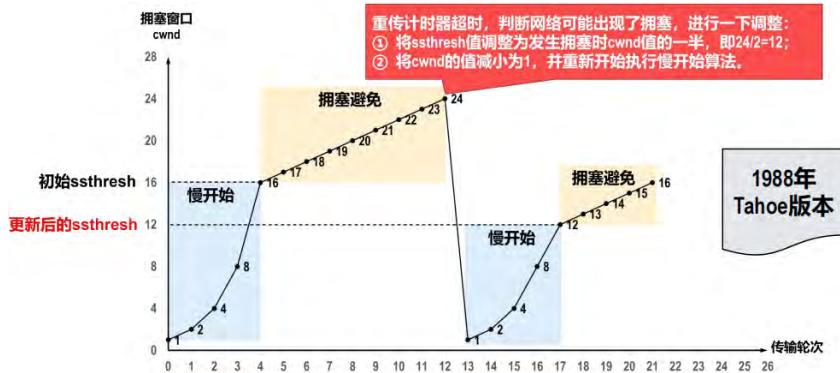
#### 5.3.4.3. TCP的四种拥塞控制方法



#### 5.3.4.3.1. 慢开始算法和拥塞避免算法

- “慢开始”是指一开始向网络注入的报文段少，而并不是指拥塞窗口cwnd的值增长速度慢。
  - “拥塞避免”也并非指完全能够避免拥塞，而是指在拥塞避免阶段将cwnd值控制为按线性规律增长，使网络比较不容易出现拥塞。

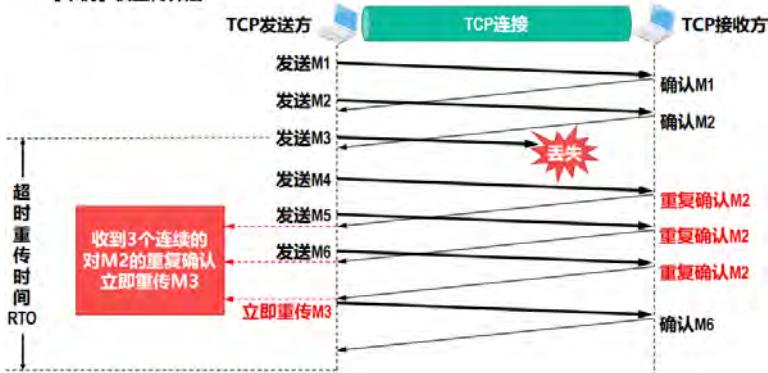




### 5.3.4.3.2. 快重传算法和快恢复算法（改进TCP性能，1990年Reno版本）

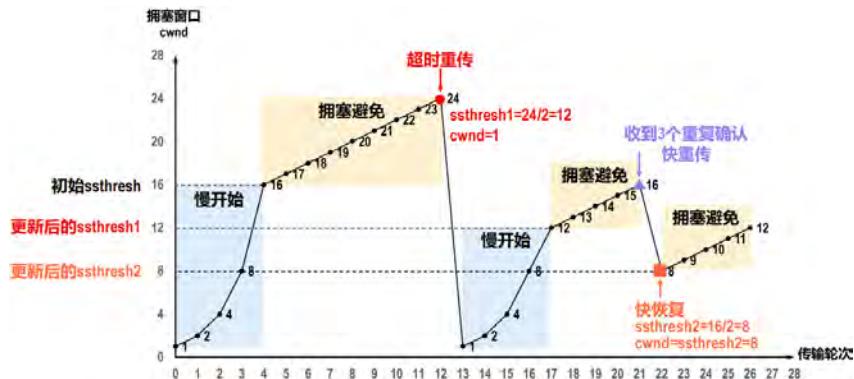
- 改进：IP数据包误码被丢弃，导致重传计时器超时，使发送方误认为网络出现了拥塞，重新执行慢开始算法，降低了传输效率。
- “快重传”是指使发送方尽快（尽早）进行重传，而不是等重传计时器超时再重传。
  - 这就要求接收方不要等待自己发送数据时才进行捎带确认，而是要立即发送确认，即使收到了失序的报文段也要立即发出对已收到的报文段的重复确认。
  - 发送方一旦收到3个连续的重复确认，就将相应的报文段立即重传，而不是等该报文段的重传计时器超时再重传。
- 快重传：收到不是按序到达的报文段就发送上一个报文段的重复确认。
  - 采用快重传算法可以让发送方尽早知道发生了个别TCP报文段的丢失。

#### 【举例】快重传算法

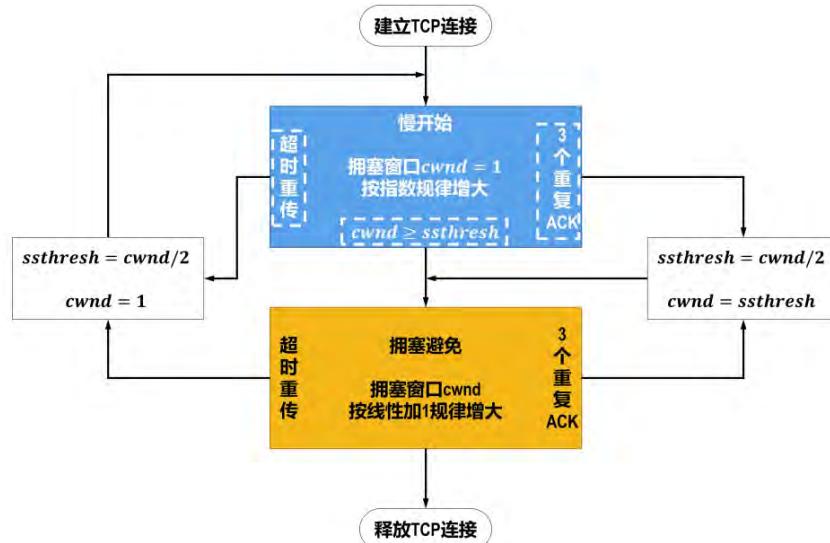


- 快恢复：发送方一旦收到3个重复确认，就知道现在只是丢失了个别的报文段，于是不启动慢开始算法，而是执行快恢复算法。
  - 发送方将慢开始门限ssthresh的值和拥塞窗口cwnd的值都调整为当前cwnd值的一半，并开始执行拥塞避免算法。
  - 也有的快恢复实现是把快恢复开始时的cwnd值再增大一些，即 $cwnd = \text{newssthresh} + 3$ 。
    - 既然发送方收到了3个重复的确认，就表明有3个数据报文段已经离开了网络。
    - 这3个报文段不再消耗网络资源而是停留在接收方的接收缓存中。
    - 可见现在网络中不是堆积了报文段而是减少了3个报文段，因此可以适当把cwnd值增大一些。

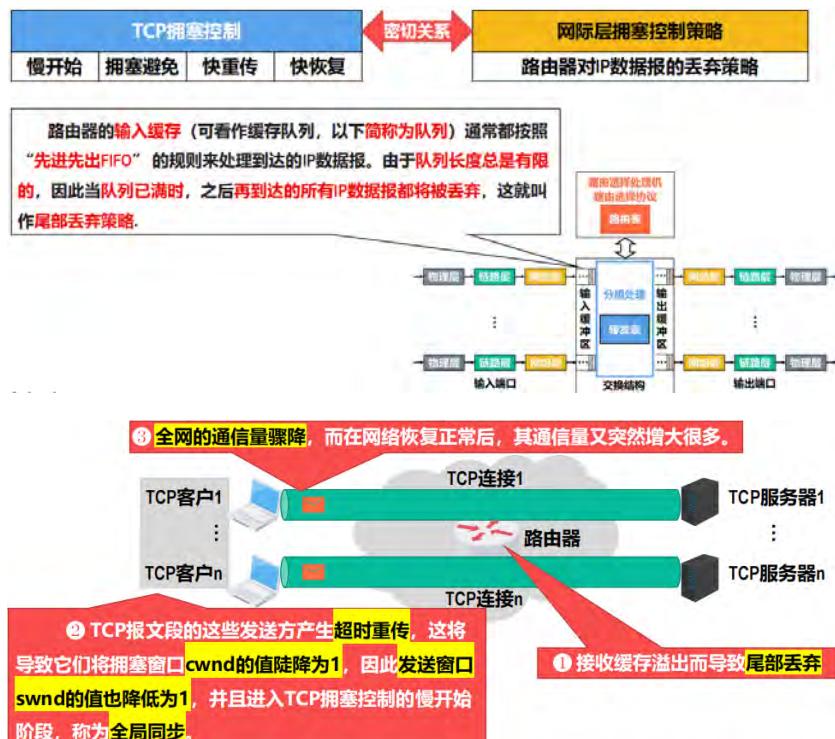
### 5.3.4.3.3. 四种控制方法拥塞窗口变化图



#### 5.3.4.3.4. TCP拥塞控制流程图



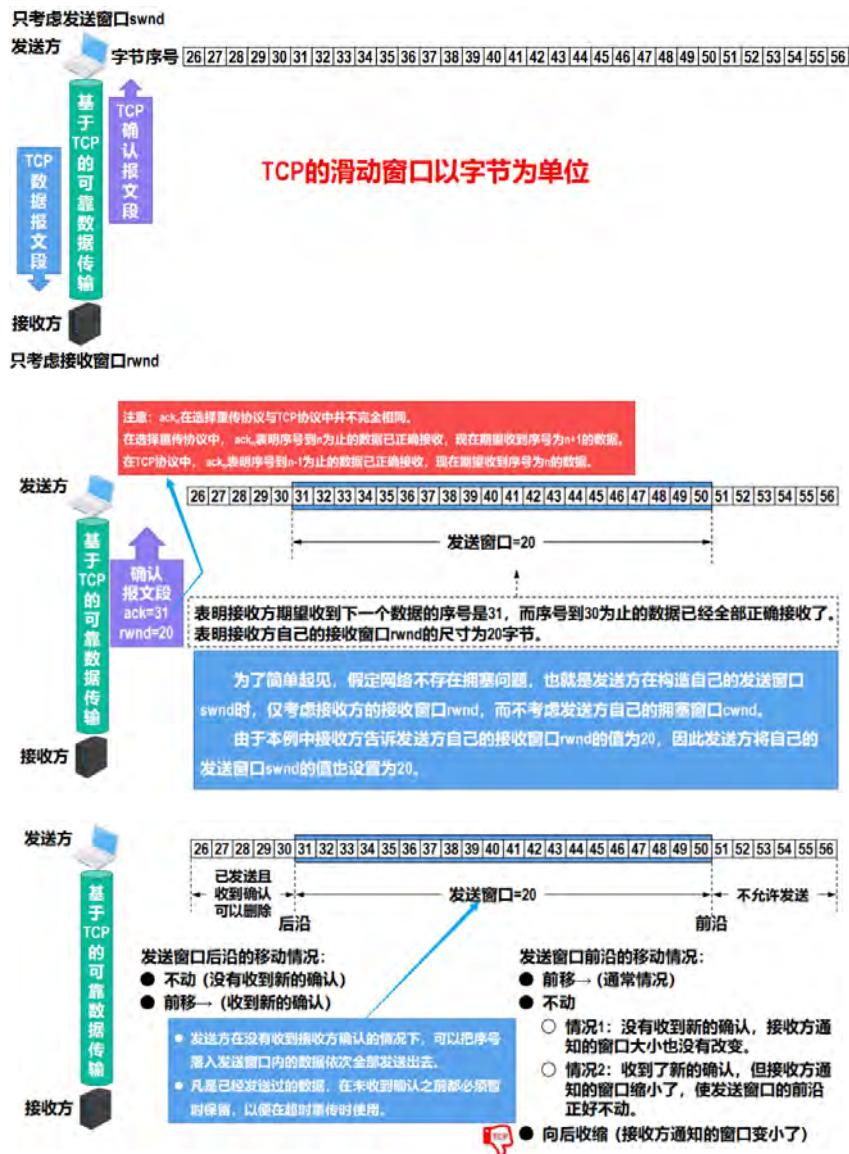
#### 5.3.4.4. TCP拥塞控制与网际层拥塞控制的关系

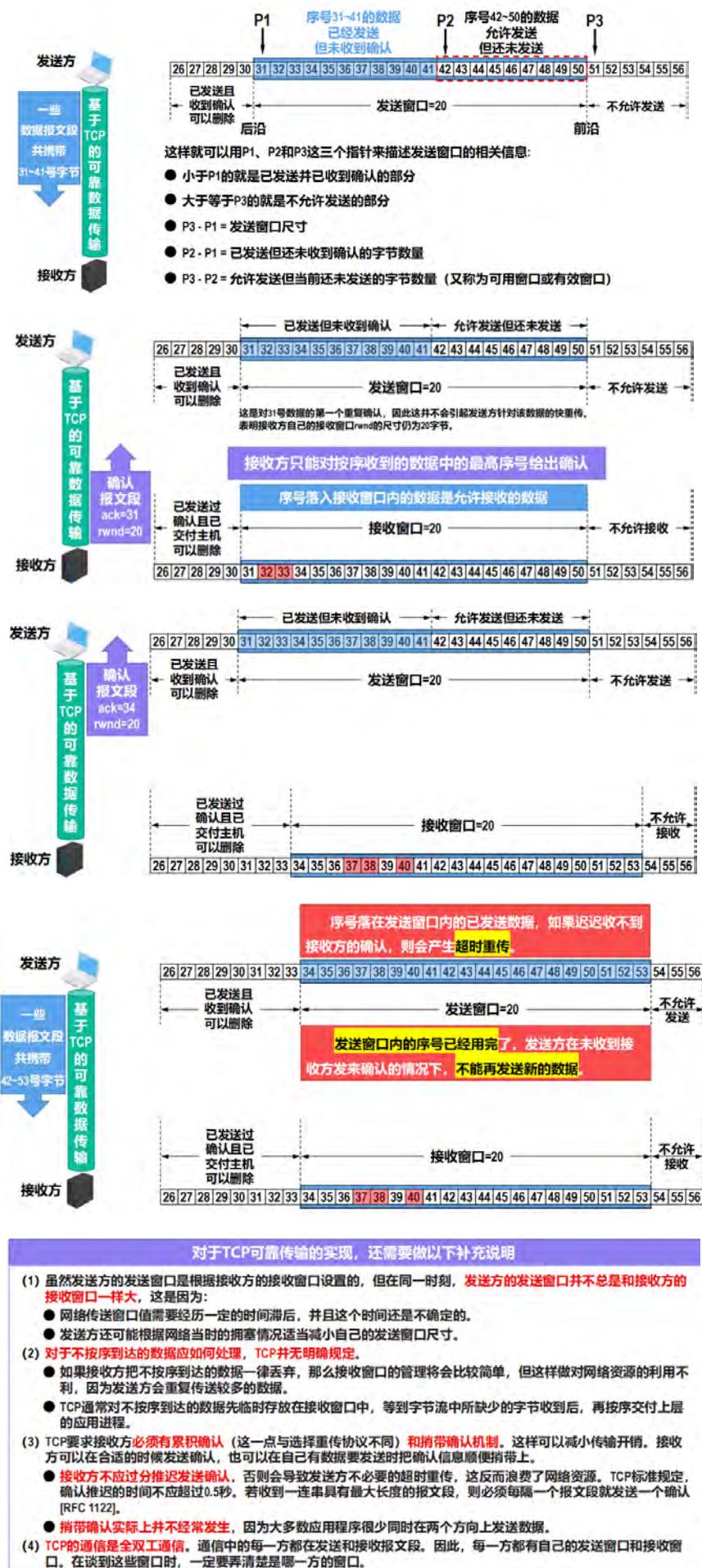


- 为了避免网络中出现全局同步问题，在1998年提出了**主动队列管理 (Active Queue Management, AQM)**。
- 所谓“主动”，就是在**路由器的队列长度达到某个阈值但还未满时就主动丢弃IP数据报，而不是要等到路由器的队列已满时才不得不丢弃后面到达的IP数据报**，这样就太被动了。
  - 应当在**路由器队列长度达到某个值得警惕的数值时，也就是网络出现了某些拥塞征兆时，就主动丢弃到达的IP数据报来造成发送方的超时重传，进而降低发送方的发送速率，因而有可能减轻网络的拥塞程度，甚至不出现网络拥塞**。
- 主动队列管理AQM可以有不同的实现方法，其中曾流行多年的就是**随机早期检测 (Random Early Detection, RED)**，也称为**随机早期丢弃 (Random Early Drop, RED 或 Random Early Discard, RED)**。
- 路由器需要维护两个参数来实现RED：**队列长度最小门限和最大门限**。当每一个IP数据报到达路由器时，RED就按照规定的算法计算出当前的平均队列长度。
    - 若**平均队列长度小于最小门限**，则把新到达的IP数据报**存入队列进行排队**。
    - 若**平均队列长度大于最大门限**，则把新到达的IP数据报**丢弃**。
    - 若**平均队列长度在最小门限和最大门限之间**，则按照某**一丢弃概率p**把新到达的IP数据报**丢弃**（这体现了丢弃IP数据报的随机性）。

因特网工程任务组IETF曾经推荐在因特网中的路由器使用RED机制[RFC 2309]，但多年的实践证明，RED的使用效果并不理想。因此，在2015年公布的RFC 7567已经把[RFC 2309]列为“陈旧的”，并且不再推荐使用RED。然而，对路由器进行主动队列管理AQM仍然是必要的。现在已经有几种不同的算法来代替旧的RED，但都还在实验阶段。目前还没有一种算法能够成为IETF的标准，有兴趣的同学可以注意这方面的进展。

### 5.3.5. TCP可靠传输的实现





### 5.3.6. TCP超时重传时间的选择



- RTTS计算

- 不能直接使用略大于某次测量得到的往返时间RTT样本的值作为超时重传时间RTO。
- 但是，可以利用每次测量得到的RTT样本计算加权平均往返时间RTTs，这样可以得到比较平滑的往返时间。

$$RTT_{S1} = RTT_1$$

$$\text{新的}RTT_s = (1 - \alpha) \times \text{旧的}RTT_s + \alpha \times \text{新的RTT样本}$$

在上式中， $0 \leq \alpha < 1$

若 $\alpha$ 很接近于0，则新RTT样本对RTTs的影响不大；

若 $\alpha$ 很接近于1，则新RTT样本对RTTs的影响较大；

已成为建议标准的[RFC 6298]推荐的 $\alpha$ 值为 $1/8$ ，即0.125。

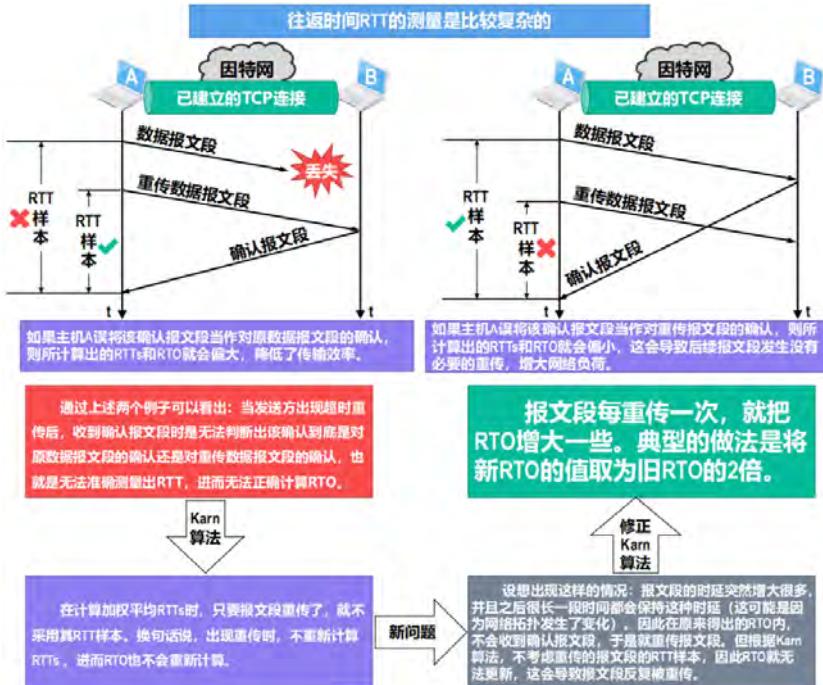
■ 显然，超时重传时间RTO的值应略大于加权平均往返时间RTTs的值（而不是某个RTT样本的值）。

■ [RFC 6298]建议使用下式来计算超时重传时间RTO：

$$RTO = RTT_s + 4 \times RTT_D$$

加权平均往返时间RTT <sub>s</sub>
$RTT_{S1} = RTT_1$
$\text{新的}RTT_s = (1 - \alpha) \times \text{旧的}RTT_s + \alpha \times \text{新的RTT样本}$
在上式中， $0 \leq \alpha < 1$ ，已成为建议标准的[RFC 6298]推荐的 $\alpha$ 值为 $1/8$ ，即0.125。

RTT偏差的加权平均RTT <sub>D</sub>
$RTT_{D1} = RTT_1 / 2$
$\text{新的}RTT_D = (1 - \beta) \times \text{旧的}RTT_D + \beta \times  \text{RTT}_s - \text{新的}RTT\text{样本} $
在上式中， $0 \leq \beta < 1$ ，已成为建议标准的[RFC 6298]推荐的 $\beta$ 值为 $1/4$ ，即0.25。



### • 总结

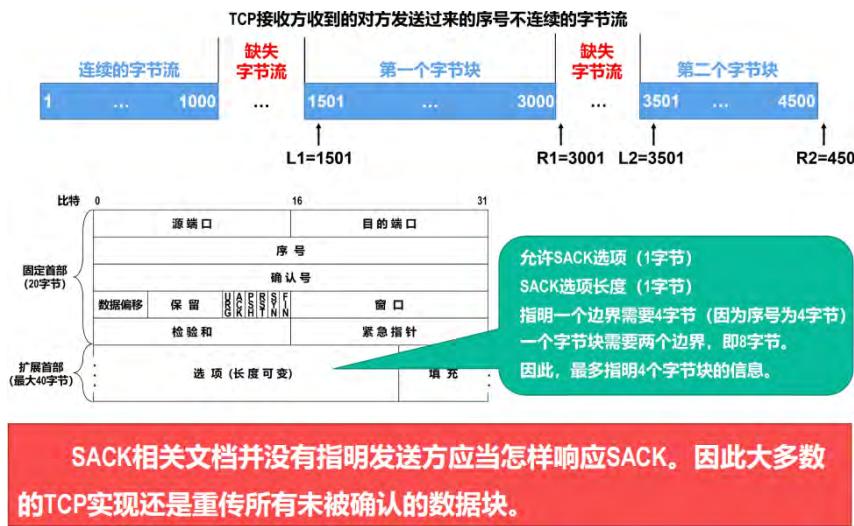
$$RTO = RTT_s + 4 \times RTT_d$$

加权平均往返时间RTT <sub>s</sub>	
RTT <sub>s1</sub> = RTT <sub>1</sub>	
新的RTT <sub>s</sub> = (1 - α) × 旧的RTT <sub>s</sub> + α × 新的RTT样本	
在上式中，0 ≤ α < 1，已成为建议标准的[RFC 6298]推荐的α值为1/8，即0.125。	
RTT偏差的加权平均RTT <sub>d</sub>	
RTT <sub>d1</sub> = RTT <sub>1</sub> ÷ 2	
新的RTT <sub>d</sub> = (1 - β) × 旧的RTT <sub>d</sub> + β ×  RTT <sub>s</sub> - 新的RTT样本	
在上式中，0 ≤ β < 1，已成为建议标准的[RFC 6298]推荐的β值为1/4，即0.25。	
报文段发生重传，就不采用RTT样本计算RTO，而是把RTO增大一些。典型的做法是将新RTO的值取为旧RTO的2倍。	

### 5.3.7. TCP的选择确认

■ 在之前介绍TCP的快重传和可靠传输时，TCP接收方只能对按序收到的数据中的最高序号给出确认。当发送方超时重传时，接收方之前已收到的未按序到达的数据也会被重传。





## 5.4. 题目

### 5.4.1. TCP序号、确认号

#### 5.4.1.1. 【2009 38】

【2009年题38】主机甲与主机乙之间已建立一个TCP连接，主机甲向主机乙发送了两个连续的TCP段，分别包含300字节和500字节的有效载荷，第一个段的序列号为200，主机乙正确接收到两个段后，发给主机甲的确认序列号是（D）。

- A. 500      B. 700      C. 800      D. 1000

解析

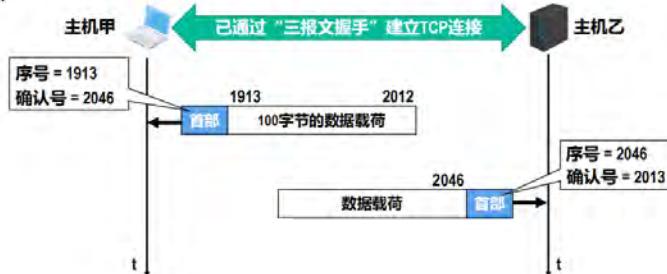


#### 5.4.1.2. 【2013 39】

【2013年题39】主机甲与主机乙之间已建立一个TCP连接，双方持续有数据传输，且数据无差错与丢失。若甲收到1个来自自己的TCP段，该段的序号是1913、确认序号为2046、有效载荷为100字节，则甲立即发送给乙的TCP段的序号和确认序号分别是（B）。

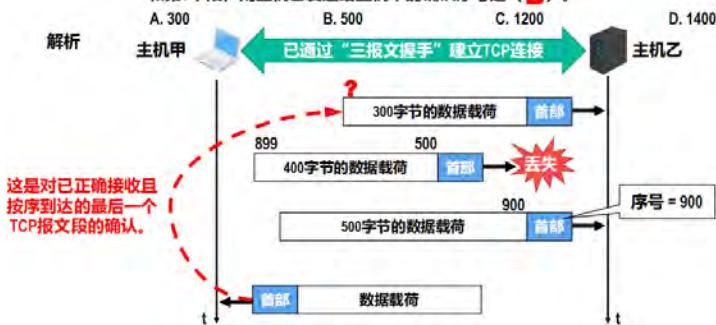
- A. 2046、2012      B. 2046、2013      C. 2047、2012      D. 2047、2013

解析



#### 5.4.1.3. 【2011 40】

【2011年题40】主机甲与主机乙之间已建立一个TCP连接，主机甲向主机乙发送了3个连续的TCP段，分别包含300字节、400字节和500字节的有效载荷，第3个段的序号为900。若主机乙仅正确接收到第1和第3个段，则主机乙发送给主机甲的确认序号是（**B**）。

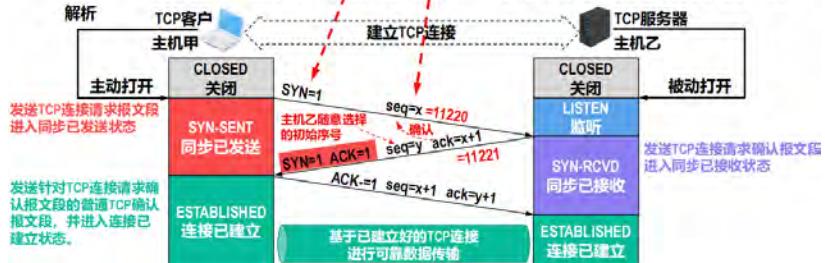


## 5.4.2. 建立TCP连接

### 5.4.2.1. 【2011 39】

【2011年题39】主机甲向主机乙发送一个（SYN=1, seq=11220）的TCP段，期望与主机乙建立TCP连接，若主机乙接受该连接请求，则主机乙向主机甲发送的正确的TCP段可能是（**C**）。

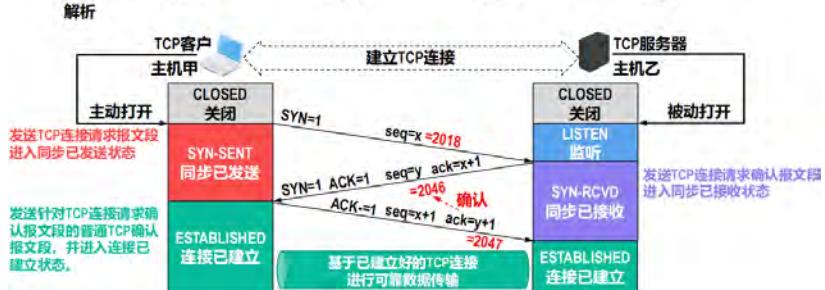
- A. (SYN=0, ACK=0, seq=11221, ack=11220)
- B. (SYN=1, ACK=1, seq=11220, ack=11220)
- C. (SYN=1, ACK=1, seq=11221, ack=11221)
- D. (SYN=0, ACK=0, seq=11220, ack=11220)



### 5.4.2.2. 【2019 39】

【2019年题39】若主机甲主动发起一个与主机乙的TCP连接，甲、乙选择的初始序列号分别为2018和2046，则第三次握手TCP段的确认序列号是（**D**）。

- A. 2018
- B. 2019
- C. 2046
- D. 2047



## 5.4.3. 释放TCP连接

### 5.4.3.1. 【2020 39】

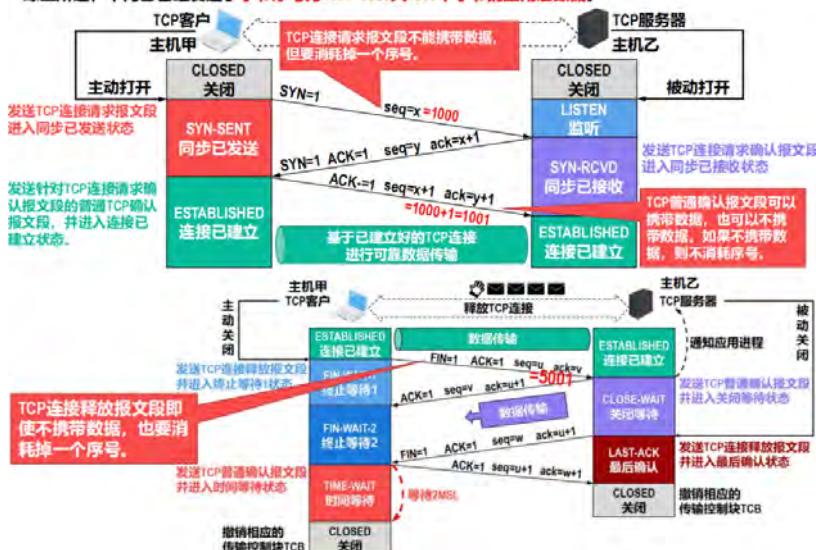
【2020年题39】若主机甲与主机乙建立TCP连接时发送的SYN段中的序号为1000，在断开连接时，甲发送给乙的FIN段中的序号为5001，则在无任何重传的情况下，甲向乙已经发送的应用层数据的字节数为 (C)。

- A. 4002      B. 4001      C. 4000      D. 3999

解析 甲给乙发送的第一个应用层数据字节的TCP序号为1001，因为应用层数据作为数据载荷被封装在TCP报文段中。

甲在发送FIN段之前，给乙发送的最后一个应用层数据字节的TCP序号为5000。

综上所述，甲向乙已经发送了字节序号为1001~5000共4000个字节的应用层数据。



#### 5.4.4. TCP流量控制

##### 5.4.4.1. 【2010 39】

【2010年题39】主机甲和主机乙之间建立了一个TCP连接，TCP最大报文段长度为1000字节。若主机甲的当前拥塞窗口为4000字节，在主机甲向主机乙连续发送两个最大报文段后，成功收到主机乙发送的第一个报文段的确认段，确认段中通告的接收窗口大小为2000字节，则此时主机甲还可以向主机乙发送的最大字节数是 (A)。

- A. 1000      B. 2000      C. 3000      D. 4000

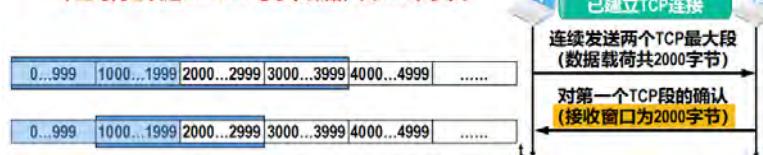
解析

TCP发送方的发送窗口值 =  $\min(\text{TCP发送方的拥塞窗口值}, \text{TCP接收方的接收窗口值})$

题目未给出TCP发送方的发送窗口值以及TCP接收方的接收窗口值，则取拥塞窗口值作为发送窗口值。

TCP最大报文段长度MSS，并不是指整个TCP报文段的长度，而是指TCP报文段的数据载荷的长度。

甲还可向乙发送2000~2999字节数据，共1000个字节。



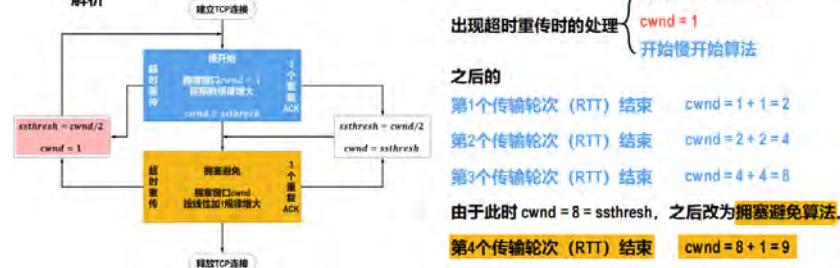
#### 5.4.5. TCP拥塞控制

##### 5.4.5.1. 【2009 39】

【2009年题39】一个TCP连接总是以1KB的最大报文段发送TCP段，发送方有足够的数据要发送。当拥塞窗口为16KB时发生了超时，如果接下来的4个RTT（往返时间）时间内的TCP段的传输都是成功的，那么当第4个RTT时间内发送的所有TCP段都得到肯定应答时，拥塞窗口大小是 (C)。

- A. 7KB      B. 8KB      C. 9KB      D. 16KB

解析



### 5.4.5.2. 【2014 38】

【2014年题38】主机甲和主机乙已建立了TCP连接，甲始终以MSS=1KB大小的段发送数据，并一直有数据发送；乙每收到一个数据段都会发出一个接收窗口为10KB的确认段。若甲在t时刻发生超时并拥塞窗口为8KB，则从t时刻起，不再发生超时的情况下，经过10个RTT后，甲的发送窗口是(A)。

- A. 10KB      B. 12KB 排除      C. 14KB 排除      D. 15KB 排除

解析

本题不知是出题人在给出的选项上有小小的失误，还是有意而为之，使得我们可以“秒杀”。

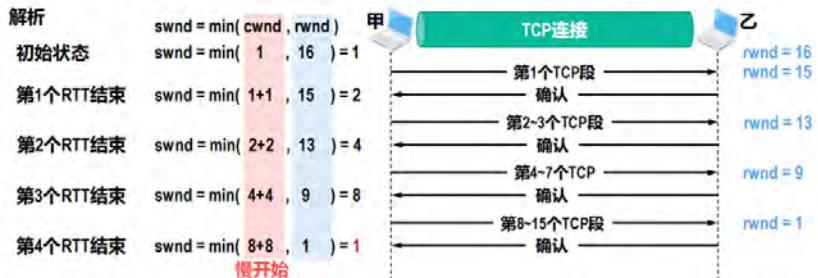


### 5.4.5.3. 【2015 39】

【2015年题39】主机甲和主机乙新建一个TCP连接，甲的拥塞控制初始阈值为32KB，甲向乙始终以MSS=1KB大小的段发送数据，并一直有数据发送；乙为该连接分配16KB接收缓存，并对每个数据段进行确认，忽略段传输延迟。若乙收到的数据全部存入缓存，不被取走，则甲从连接建立成功时刻起，未发生超时的情况下，经过4个RTT后，甲的发送窗口是(A)。

- A. 1KB      B. 8KB      C. 16KB      D. 32KB

解析



### 5.4.5.4. 【2017 39】

【2017年题39】若甲向乙发起一个TCP连接，最大段长MSS=1KB，RTT=5ms，乙开辟的接收缓存为64KB，则甲从连接建立成功至发送窗口达到32KB，需经过的时间至少是(A)。

- A. 25ms      B. 30ms      C. 160ms      D. 165ms

解析



### 5.4.5.5. 【2019 38】

【2019年题38】若客户通过一个TCP连接向服务器发送数据的部分过程如下图所示。客户在t0时刻第一次收到确认序列号ack\_seq=100的段，并发送序列号seq=100的段，但发生丢失。若TCP支持快速重传，则客户重新发送seq=100段的时刻是 (C)。



#### 5.4.5.6. 【2020 38】

【2020年题38】若主机甲与主机乙已建立一条TCP连接，最大段长MSS为1KB，往返时间RTT为2ms，则在不出现拥塞的前提下，拥塞窗口从8KB增长到20KB所需的最长时间是 (C)。

- A. 4ms    B. 8ms    C. 24ms    D. 48ms

解析

若拥塞窗口从8KB增长到20KB所需的时间最长，则该过程应处于拥塞避免算法的执行过程中。

在上述过程中：

第1个传输轮次 (RTT) 结束 cwnd = 8 + 1 = 9

第2个传输轮次 (RTT) 结束 cwnd = 9 + 1 = 10

⋮

第12个传输轮次 (RTT) 结束 cwnd = 19 + 1 = 20

共经历了12个传输轮次 (RTT)，总耗时为  $2\text{ms} \times 12 = 24\text{ms}$

#### 5.4.6. TCP超时重传的时间选择

【练习】假设要进行五次TCP往返时间RTT的测量，从第一次开始，依次测得RTT为30ms, 26ms, 32ms, 24ms, 而第五次测量RTT时出现了超时重传。设 $\alpha = \beta = 0.1$ ，请计算每次测量后所计算出的超时重传时间RTO。

解析

根据题意可知， $RTT_1=30\text{ms}$ ,  $RTT_2=26\text{ms}$ ,  $RTT_3=32\text{ms}$ ,  $RTT_4=24\text{ms}$ , 测量 $RTT_5$ 时出现超时重传。

$$RTT_{S1} = RTT_1 = 30\text{ms}$$

$$RTT_{D1} = RTT_1 / 2 = 15\text{ms}$$

$$\begin{aligned} RTT_{S2} &= (1 - \alpha) \times RTT_{S1} + \alpha \times RTT_2 \\ &= (1 - 0.1) \times RTT_{S1} + 0.1 \times RTT_2 \\ &= 0.9 \times 30\text{ms} + 0.1 \times 26\text{ms} = 29.6\text{ms} \end{aligned}$$

$$\begin{aligned} RTT_{D2} &= (1 - \beta) \times RTT_{D1} + \beta \times |RTT_{S2} - RTT_2| \\ &= (1 - 0.1) \times RTT_{D1} + 0.1 \times |RTT_{S2} - RTT_2| \\ &= 0.9 \times 15\text{ms} + 0.1 \times |29.6\text{ms} - 26\text{ms}| = 13.86\text{ms} \end{aligned}$$

$$RTO = RTT_{S1} + 4 \times RTT_{D1}$$

$$RTO_1 = RTT_{S1} + 4 \times RTT_{D1} = 30\text{ms} + 4 \times 15\text{ms} = 90\text{ms}$$

$$\begin{aligned} RTT_{S3} &= (1 - 0.1) \times RTT_{S2} + 0.1 \times RTT_3 \\ &= 0.9 \times 29.6\text{ms} + 0.1 \times 32\text{ms} = 29.84\text{ms} \end{aligned}$$

$$\begin{aligned} RTT_{D3} &= (1 - \alpha) \times RTT_{D2} + 0.1 \times |RTT_{S3} - RTT_3| \\ &= 0.9 \times 13.86\text{ms} + 0.1 \times |29.84\text{ms} - 32\text{ms}| = 12.89\text{ms} \end{aligned}$$

$$RTO_2 = RTT_{S2} + 4 \times RTT_{D2} = 29.6\text{ms} + 4 \times 13.86\text{ms} = 85.04\text{ms}$$

$$\begin{aligned} RTT_{S4} &= (1 - 0.1) \times RTT_{S3} + 0.1 \times RTT_4 \\ &= 0.9 \times 29.84\text{ms} + 0.1 \times 24\text{ms} = 29.256\text{ms} \end{aligned}$$

$$\begin{aligned} RTT_{D4} &= (1 - \beta) \times RTT_{D3} + 0.1 \times |RTT_{S4} - RTT_4| \\ &= 0.9 \times 12.89\text{ms} + 0.1 \times |29.256\text{ms} - 24\text{ms}| = 11.946\text{ms} \end{aligned}$$

$$RTO_3 = RTT_{S3} + 4 \times RTT_{D3} = 29.84\text{ms} + 4 \times 12.89\text{ms} = 80.6\text{ms}$$

由于测量 $RTT_5$ 时出现超时重传，因此不用计算 $RTT_{S5}$ 。

由于测量 $RTT_5$ 时出现超时重传，因此不用计算 $RTT_{D5}$ 。

$$RTO_4 = RTT_{S4} + 4 \times RTT_{D4} = 29.256\text{ms} + 4 \times 11.946\text{ms} = 77.0424\text{ms}$$

$$RTO_5 = 2 \times RTO_4 = 2 \times 77.0424\text{ms} = 154.0848\text{ms}$$

## 6. Application layer

### 6.1. 应用层概述

■ 应用层是计算机网络体系结构的最顶层，是设计和建立计算机网络的最终目的，也是计算机网络中发展最快的部分。

- 早期基于文本的应用（电子邮件、远程登录、文件传输、新闻组）
- 20世纪90年代将因特网带入千家万户的万维网WWW
- 当今流行的即时通信、P2P文件共享及各种音视频应用
- 计算设备的小型化和“无处不在”，宽带住宅接入和无线接入的日益普及和迅速发展，为未来更多的新型应用提供了广阔的舞台

■ 在本章中，我们以一些经典的网络应用为例来学习有关网络应用的原理、协议和实现方面的知识。



## 6.2. 客户/服务器方式和对等方式

### 6.2.1. C/S方式

#### ■ 客户/服务器 (Client/Server, C/S) 方式

- 客户和服务器是指通信中所涉及的两个应用进程。
- 客户/服务器方式所描述的是进程之间服务和被服务的关系。
- 客户是服务请求方，服务器是服务提供方。
- 服务器总是处于运行状态，并等待客户的服务请求。

服务器具有固定端口号（例如HTTP服务器的默认端口号为80），而运行服务器的主机也具有固定的IP地址。

■ C/S方式是因特网上传统的、同时也是最成熟的方式，很多我们熟悉的网络应用采用的都是C/S方式。包括万维网WWW、电子邮件、文件传输FTP等。

■ 基于C/S方式的应用服务通常是服务集中型的，即应用服务集中在网络中比客户计算机少得多的服务器计算机上。

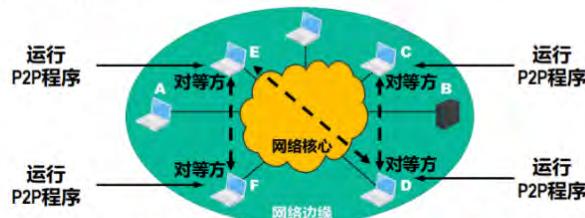
- 由于一台服务器计算机要为多个客户机提供服务，在C/S应用中，常会出现服务器计算机跟不上众多客户机请求的情况。
- 为此，在C/S应用中，常用计算机群集（或服务器场）构建一个强大的虚拟服务器。



### 6.2.2. P2P方式

### ■ 对等 (Peer-to-Peer, P2P) 方式

- 在P2P方式中，**没有固定的服务请求者和服务提供者**，分布在网络边缘各端系统中的应用进程是对等的，被称为**对等方**。对等方相互之间直接通信，每个对等方既是服务的请求者，又是服务的提供者。
- 目前，在因特网上流行的P2P应用主要包括P2P文件共享、即时通信、P2P流媒体、分布式存储等。
- 基于P2P的应用是**服务分散型**的，因为服务不是集中在少数几个服务器计算机中，而是分散在大量对等计算机中，这些计算机并不为服务提供商所有，而是为个人控制的桌面计算机和笔记本电脑，它们通常位于住宅、校园和办公室中。

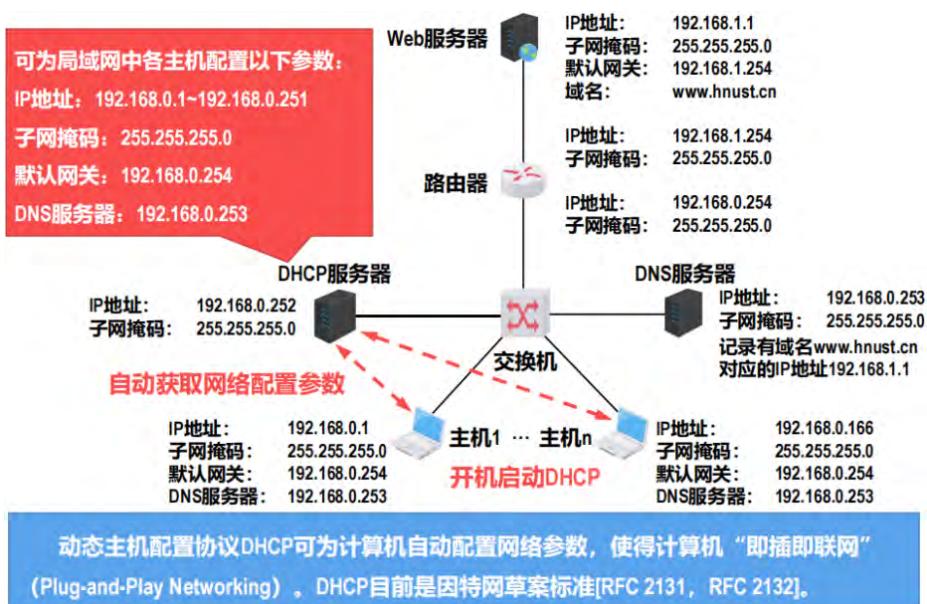


- P2P方式的最突出特性之一就是它的**可扩展性**。因为系统每增加一个对等方，不仅增加的是服务的请求者，同时也增加了服务的提供者，**系统性能不会因规模的增大而降低**。
- P2P方式**具有成本上的优势**，因为它通常不需要庞大的服务器设置和服务器带宽。为了降低成本，服务提供商对于将P2P方式用于应用的兴趣越来越大。

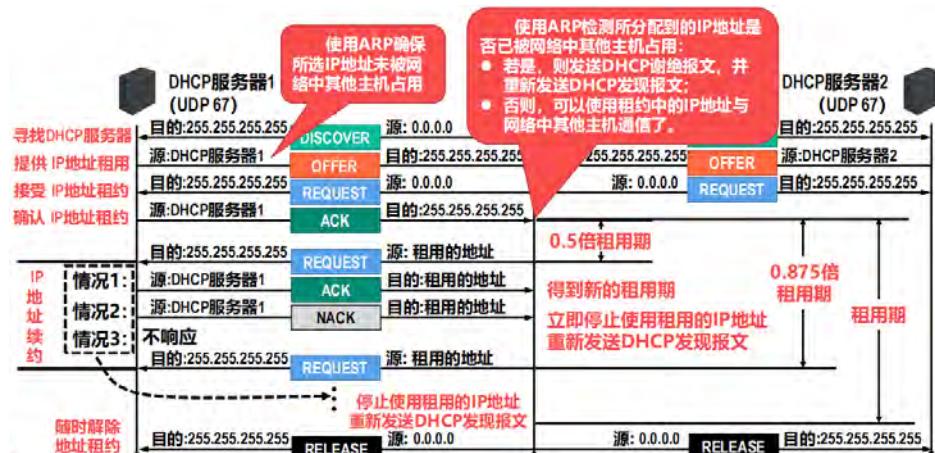
## 6.3. 动态主机配置协议DHCP

### 6.3.1. 动态主机配置协议DHCP的作用

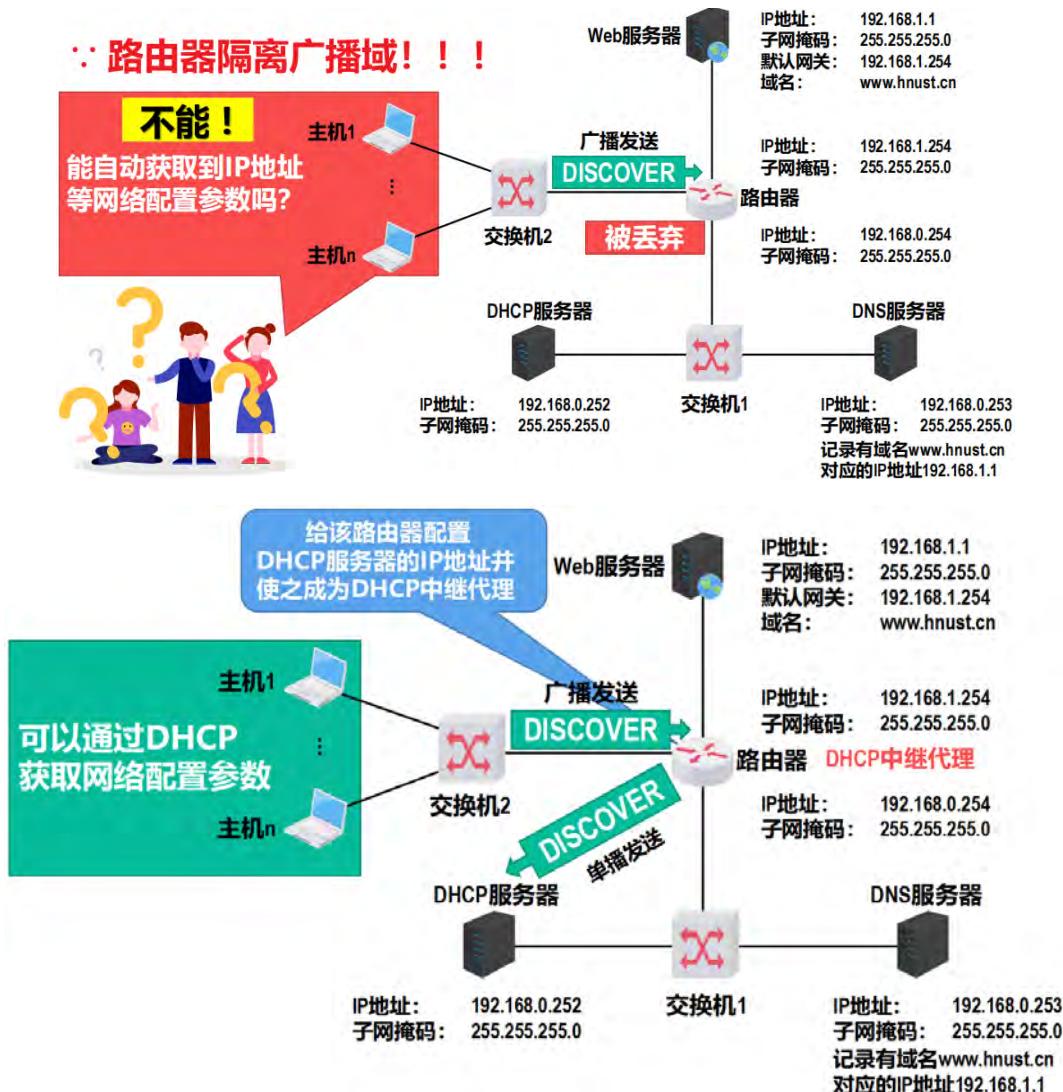
- 手动配置工作量大且容易出错



### 6.3.2. 动态主机配置协议DHCP的基本工作过程



### 6.3.3. DHCP 中继代理



### 6.3.4. 总结

- 动态主机配置协议DHCP(Dynamic Host Configuration Protocol)提供了一种机制，称为即插即用连网。这种机制**允许一台计算机加入新网络时可自动获取IP地址等网络配置信息而不用手工参与。**
- DHCP主要使用以下报文来实现其功能：
  - DHCP DISCOVER: DHCP发现报文
  - DHCP REQUEST: DHCP请求报文
  - DHCP NACK: DHCP否认报文
  - DHCP OFFER: DHCP提供报文
  - DHCP ACK: DHCP确认报文
  - DHCP RELEASE: DHCP释放报文
- DHCP报文**在运输层使用UDP协议封装**
  - DHCP客户使用的UDP端口号为68
  - DHCP服务器使用的UDP端口号为67
- DHCP客户在**未获取到IP地址时使用地址0.0.0.0**
- 在每一个网络上都设置一个DHCP服务器会使DHCP服务器的数量太多。因此现在是使每一个网络至少有一个**DHCP中继代理**（通常是一台路由器），它配置了DHCP服务器的IP地址信息，作为各网络中计算机与DHCP服务器的桥梁。

## 6.4. 域名系统DNS(Domain Name System)

### 6.4.1. 域名系统的作用

- 因特网是否可以只使用一台DNS服务器？
  - 理论可行但不可取。因为因特网的规模很大，这样的域名服务器肯定会因为超负荷而无法正常工作，而且一旦域名服务器出现故障，整个因特网就会瘫痪。

- 早在1983年，因特网就开始采用层次结构的命名树作为主机的名字（即域名），并使用分布式的域名系统DNS。
- DNS的作用
  - 系统效率高：DNS使大多数域名都在本地解析，仅少量解析需要在因特网上通信。
  - 于DNS是分布式系统，即使单个计算机出了故障，也不会妨碍整个系统的正常运行。

#### 6.4.2. 因特网的域名结构（层次树状结构）

- DNS报文使用运输层的UDP协议进行封装，运输层端口号为53。

##### 6.4.2.1. 概述

■ 因特网采用**层次树状结构的域名结构**。

■ 域名的结构由若干个分量组成，各分量之间用“点”隔开，分别代表不同级别的域名。

### ... . 三级域名.二级域名.顶级域名

每一级的域名都由英文字母和数字组成，不超过63个字符，不区分大小写字母。

级别最低的域名写在最左边，而级别最高的顶级域名写在最右边。

完整的域名不超过255个字符。

■ 域名系统既不规定一个域名需要包含多少个下级域名，也不规定每一级的域名代表什么意思。

■ 各级域名由其上一级的域名管理机构管理，而最高的顶级域名则由因特网名称与数字地址分配机构ICANN进行管理。

【举例】湖南科技大学网络信息中心的域名

n i c .	h n u s t .	e d u .	c n
四级域名	三级域名	二级域名	顶级域名

##### 6.4.2.2. 分类

■ **顶级域名 (Top Level Domain, TLD)** 分为以下三类：

**国家顶级域名nTLD** 采用ISO 3166的规定。如cn表示中国，us表示美国，uk表示英国、等等。

**通用顶级域名gTLD** 最常见的通用顶级域名有七个，即：com (公司企业)、net (网络服务机构)、org (非营利性组织)、int (国际组织)、edu (美国教育机构)、gov (美国政府部门)、mil (美国军事部门)。

**反向域名arpa** 用于反向域名解析，即IP地址反向解析为域名。

■ 在**国家顶级域名下注册的二级域名均由该国家自行确定**。例如，顶级域名为jp的日本，将其教育和企业的二级域名定为ac和co，而不用edu和com。

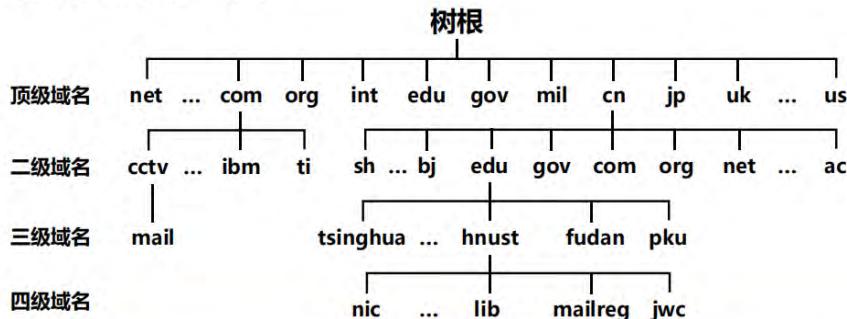
■ **我国则将二级域名划分为以下两类：**

**类别域名** 共七个：ac (科研机构)、com (工、商、金融等企业)、edu (教育机构)、gov (政府部门)、net (提供网络服务的机构)、mil (军事机构) 和org (非营利性组织)。

**行政区域名** 共34个，适用于我国的各省、自治区、直辖市。例如：bj为北京市、sh为上海市、js为江苏省，等等。

##### 6.4.2.3. 举例

【举例】因特网的域名空间



这种按等级管理的命名方法便于维护名字的唯一性，并且也容易设计出一种高效的域名查询机制。需要注意的是，域名只是个逻辑概念，并不代表计算机所在的物理地点。

### 6.4.3. 因特网上的域名服务器

- 域名和IP地址的映射关系必须保存在域名服务器中，供所有其他应用查询。显然不能将所有信息都储存在一台域名服务器中。
- DNS使用分布在各地的域名服务器来实现域名到IP地址的转换。

#### 6.4.3.1. 根域名服务器



- 根域名服务器是最高层次的域名服务器。**
- 每个根域名服务器都知道所有的顶级域名服务器的域名及其IP地址。**
- 尽管我们将这**13个**根域名服务器中的每一个都视为单个的服务器，但“每台服务器”实际上是由许多分布在世界各地的计算机构成的**服务器群集**。
  - 当本地域名服务器向根域名服务器发出查询请求时，路由器就把查询请求报文转发到离这个DNS客户最近的一个根域名服务器。
  - 这就加快了DNS的查询过程，同时也更合理地利用了因特网的资源。**根域名服务器通常并不直接对域名进行解析，而是返回该域名所属顶级域名的顶级域名服务器的IP地址。**

#### 6.4.3.2. 顶级域名服务器



- 顶级域名服务器负责**管理**在该顶级域名服务器注册的所有二级域名。**
- 当收到DNS查询请求时就给出相应的回答，可能是最后的结果，也可能是下一级权限域名服务器的IP地址。

#### 6.4.3.3. 权限域名服务器



- 权限域名服务器负责**管理**某个区的域名。**
- 每一个主机的域名都必须在某个权限域名服务器处注册登记。**因此权限域名服务器知道其管辖的域名与IP地址的映射关系。
- 另外，权限域名服务器还知道其下级域名服务器的地址。

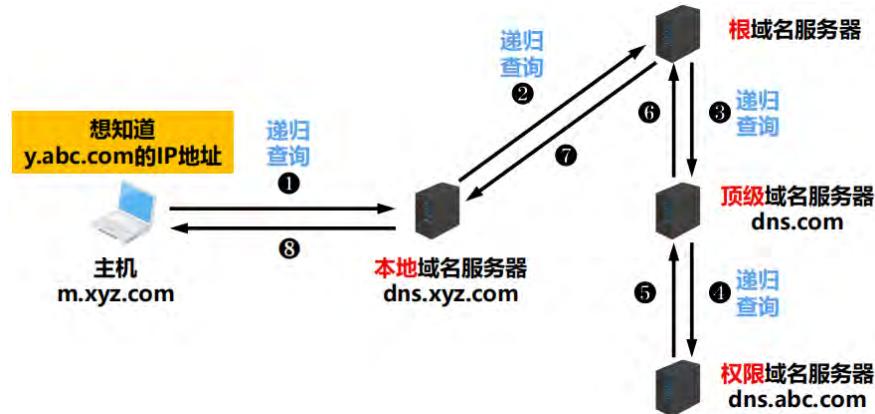
#### 6.4.3.4. 本地域名服务器



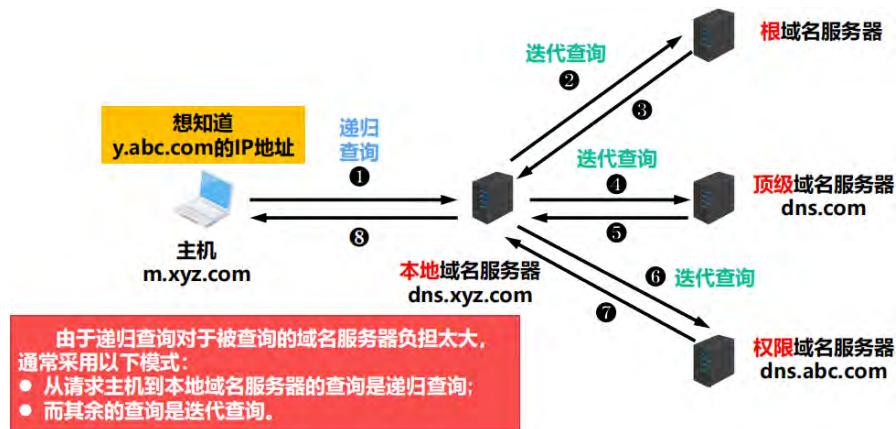
- 本地域名服务器不属于上述的域名服务器的等级结构。**
- 当一个主机发出DNS请求报文时，这个报文就首先被送往该主机的本地域名服务器。**
- 本地域名服务器起着代理的作用，会将该报文转发到上述的域名服务器的等级结构中。**
- 每一个因特网服务提供者ISP，一个大学，甚至一个大学里的学院，都可以拥有一个本地域名服务器，它有时也称为**默认域名服务器**。**
- 本地域名服务器离用户较近，一般不超过几个路由器的距离，也有可能就在同一个局域网中。本地域名服务器的IP地址需要直接配置在需要域名解析的主机中。**

### 6.4.4. 因特网的域名解析过程

#### 6.4.4.1. 递归查询

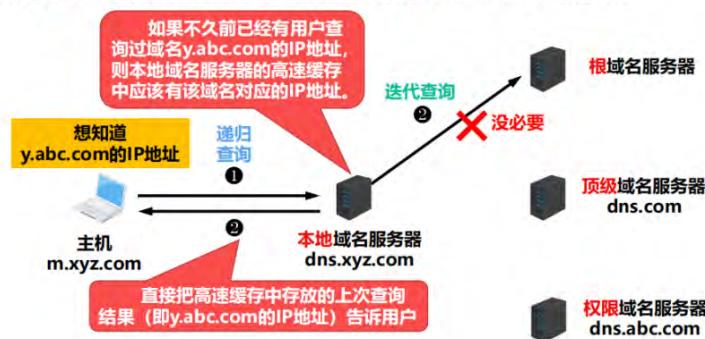


#### 6.4.4.2. 迭代查询



#### 6.4.4.3. 高速缓存

- 为了提高DNS的查询效率，并减轻根域名服务器的负担和减少因特网上的DNS查询报文数量，在域名服务器中广泛地使用了**高速缓存**。高速缓存用来存放最近查询过的域名以及从何处获得域名映射信息的记录。
- 由于域名到IP地址的映射关系并不是永久不变，为保持高速缓存中的内容正确，域名服务器**应为每项内容设置计时器并删除超过合理时间的项**（例如，每个项目只存放两天）。
- 不但在本地域名服务器中需要高速缓存，在用户**主机**中也很需要。许多用户主机在启动时从本地域名服务器下载域名和IP地址的全部数据库，维护存放自己最近使用的域名的高速缓存，并且只在从缓存中找不到域名时才向域名服务器查询。同理，主机也需要保持**高速缓存**中内容的正确性。



## 6.5. 文件传送协议FTP(File Transfer Protocol)

### 6.5.1. 文件传送协议FTP的作用

- 将某台计算机中的文件通过网络传送到可能相距很远的另一台计算机中，是一项基本的网络应用，即**文件传送**。
- **文件传送协议** (File Transfer Protocol, **FTP**) 是因特网上使用得最广泛的文件传送协议。
- FTP提供交互式的访问**，允许客户**指明文件的类型与格式**（如指明是否使用ASCII码），并允许**文件具有存取权限**（如访问文件的用户必须经过授权，并输入有效的口令）。
- FTP屏蔽了各计算机系统的细节**，因而适合于在异构网络中任意计算机之间传送文件。
- 在因特网发展的早期阶段，用FTP传送文件约占整个因特网的通信量的三分之一，而由电子邮件和域名系统所产生的通信量还要小于FTP所产生的通信量。只是到了1995年，万维网WWW的通信量才首次超过了FTP。

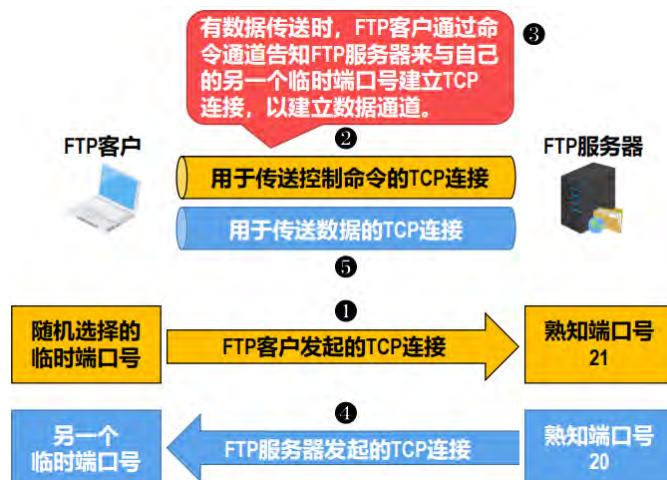
- FTP的常见用途是在计算机之间传输文件，尤其是用于批量传输文件。
- FTP的另一个常见用途是让网站设计者将构成网站内容的大量文件批量上传到他们的Web服务器。
- FTP客户可向FTP服务器上传或下载文件。
- 根据应用需求的不同，FTP服务器可能
  - 需要一台高性能和高可靠性的服务器计算机，
  - 也可能只需要一台普通的个人计算机即可。
- 在windows系统中添加了一个FTP站点（FTP服务器）的方法。
  - <https://www.jianshu.com/p/ece21421e246>
  - [https://blog.51cto.com/u\\_15351682/3729949](https://blog.51cto.com/u_15351682/3729949)

### 6.5.2. 文件传送协议FTP的基本工作原理

- FTP客户和服务器之间要建立以下两个并行的TCP连接
  - 控制连接
    - 在整个会话期间一直保持打开，用于传送FTP相关控制命令。
  - 数据连接
    - 用于文件传输，在每次文件传输时才建立，传输结束就关闭。
- 默认情况下，FTP使用
  - TCP 21端口进行控制连接
  - TCP 20端口进行数据连接。
- 但是，是否使用TCP 20端口建立数据连接与传输模式有关
  - 主动方式使用TCP 20端口
  - 被动方式由服务器和客户端自行协商决定。

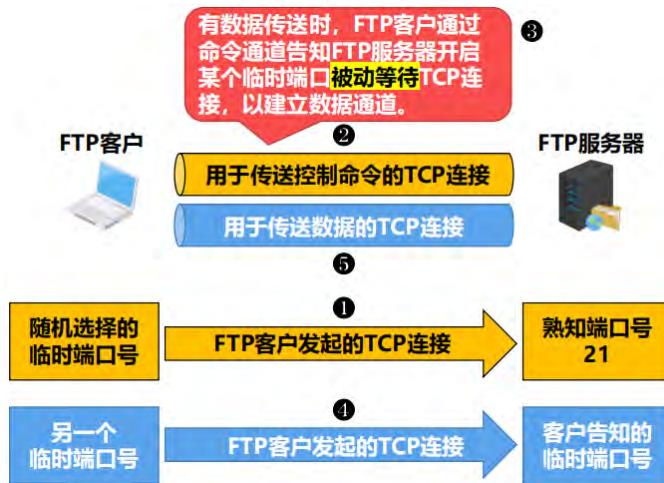
#### 6.5.2.1. 主动模式

- 建立数据通道时，FTP服务器主动连接FTP客户。



#### 6.5.2.2. 被动模式

- 建立数据通道时，FTP服务器被动等待FTP客户的连接。



## 6.6. 电子邮件

### 6.6.1. 电子邮件的作用

- 电子邮件E-mail是因特网上**最早流行的一种应用**, 并且仍然是当今因特网上**最重要、最实用的应用之一**。
- 传统的电话通信属于实时通信, 存在以下两个缺点:
  - 电话通信的主叫和被叫双方必须同时在场;
  - 一些不是十分紧迫的电话也常常不必要地打断人们的工作或休息。
- 而电子邮件与邮政系统的寄信相似。
  - ① 发件人将邮件发送到自己使用的**邮件服务器**;
  - ② 发件人的邮件服务器将收到的邮件按其目的地址转发到收件人邮件服务器中的收件人邮箱;
  - ③ 收件人在方便的时候访问收件人邮件服务器中自己的邮箱, 获取收到的电子邮件。
- 电子邮件使用方便、传递迅速而且费用低廉。它不仅可以传送文字信息, 而且还可附上声音和图像。
- 由于电子邮件的广泛使用, 现在许多国家已经正式取消了电报业务。在我国, **电信局的电报业务也因电子邮件的普及而濒临消失**。

### 6.6.2. 电子邮件系统的组成

- 电子邮件系统采用客户/服务器方式。
- 电子邮件系统的三个主要组成: 构件用户代理, 邮件服务器, 以及电子邮件所需的协议

#### 6.6.2.1. 用户代理

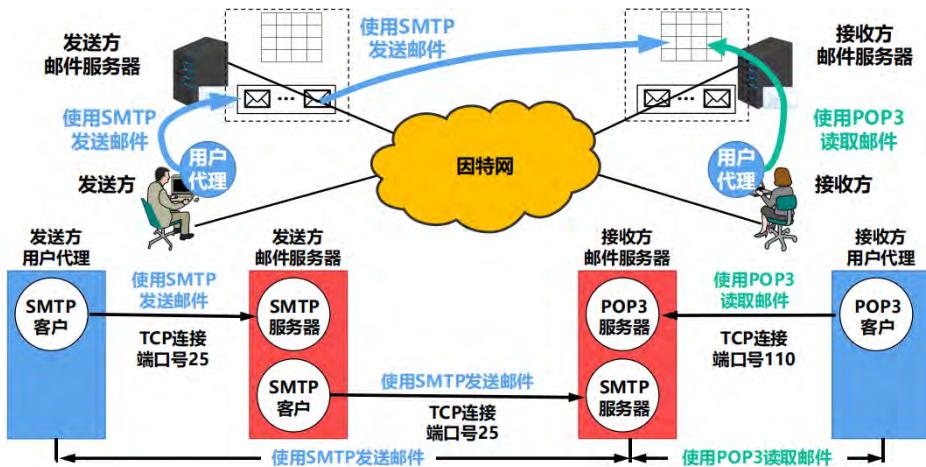
- 用户与电子邮件系统的接口, 又称为电子邮件客户端软件。

#### 6.6.2.2. 邮件服务器

- 电子邮件系统的基础设施。
- 因特网上所有的因特网服务提供者ISP都有邮件服务器, 其功能是发送和接收邮件, 同时还要负责维护用户的邮箱。

#### 6.6.2.3. 协议

- 发送协议 (例如SMTP)
- 读取协议 (例如POP3, IMAP)



### 6.6.3. 简单邮件传送协议SMTP的基本工作过程

- Simple Mail Transfer Protocol, SMTP

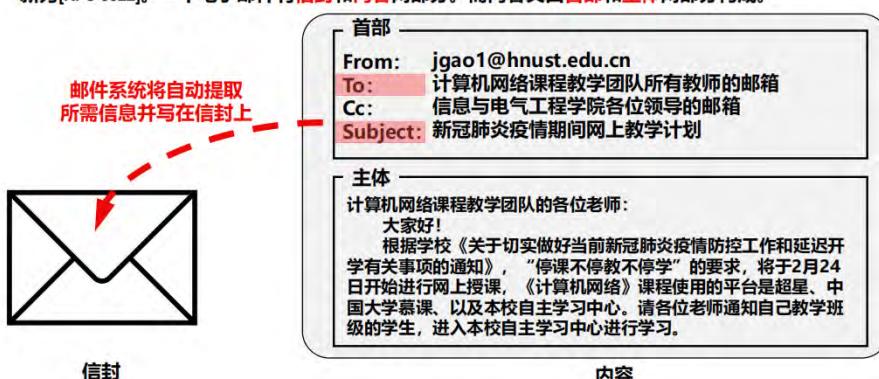
- 基于TCP连接，端口号为25。
- 只能传送ASCII码文本。
- 用于用户代理向邮件服务器发送邮件以及邮件服务器之间的邮件发送。



### 6.6.4. 电子邮件的信息格式

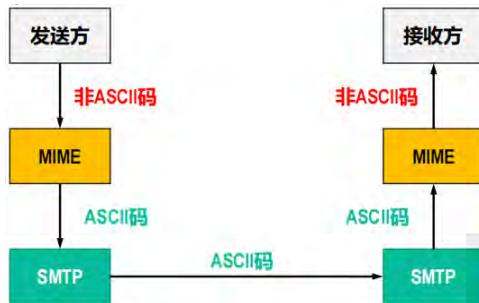
- 信封
- 内容
  - 首部
  - 主体

■ 电子邮件的信息格式并不是由SMTP定义的，而是在[RFC 822]中单独定义的。这个RFC文档已在2008年更新为[RFC 5322]。一个电子邮件有信封和内容两部分。而内容又由首部和主体两部分构成。



### 6.6.5. 多用途因特网邮件扩展MIME

- 为解决SMTP传送非ASCII码文本的问题，提出MIME。
- 多用途因特网邮件扩展 (Multipurpose Internet Mail Extensions, MIME)
  - 增加了5个新的邮件首部字段，这些字段提供了有关邮件主体的信息。
  - 定义了许多邮件内容的格式，对多媒体电子邮件的表示方法进行了标准化。定义了传送编码，可对任何内容格式进行转换，而不会被邮件系统改变
  - 定义了传送编码，可对任何内容格式进行转换，而不会被邮件系统改变。
- MIME不仅仅用于SMTP，也用于后来的同样面向ASCII字符的HTTP。

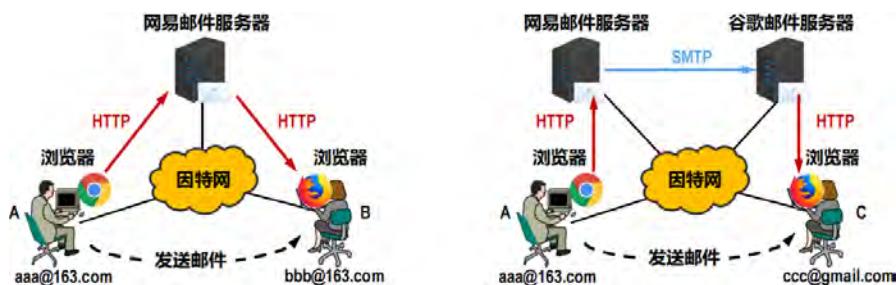


### 6.6.6. 常用的邮件读取协议 (POP、IMAP)

邮局协议 (Post Office Protocol, POP)	因特网邮件访问协议 (Internet Message Access Protocol, IMAP)
<ul style="list-style-type: none"> <li>● POP3是其第三个版本，是因特网正式标准。</li> <li>● 非常简单、功能有限的邮件读取协议。</li> <li>● 用户只能以<b>下载并删除方式或下载并保留方式</b>从邮件服务器下载邮件到用户方计算机。</li> <li>● <b>不允许用户在邮件服务器上管理自己的邮件。</b> (例如创建文件夹，对邮件进行分类管理等)。</li> </ul>	<ul style="list-style-type: none"> <li>● IMAP4是其第四个版本，是因特网建议标准。</li> <li>● 功能比POP3强大的邮件读取协议。</li> <li>● <b>用户在自己的计算机上就可以操控邮件服务器中的邮箱，就像在本地操控一样，因此IMAP是一个联机协议。</b></li> </ul>
端口号为110	POP3和IMAP4都采用基于TCP连接的客户/服务器方式
	端口号为143

### 6.6.7. 基于万维网的电子邮件

- 通过浏览器登录（提供用户名和口令）邮件服务器万维网网站就可以撰写、收发、阅读和管理电子邮件。
- 这种工作模式与IMAP很类似，不同的是用户计算机无需安装专门的用户代理程序，只需要使用万维网浏览器。
  - 用户浏览器与邮件服务器网站之间使用HTTP协议
  - 邮件服务器之间使用SMTP协议。
- 邮件服务器网站通常都提供非常强大和方便的邮件管理功能，用户可以在邮件服务器网站上管理和处理自己的邮件，而不需要将邮件下载到本地进行管理。



## 6.7. 万维网WWW

### 6.7.1. 万维网概述

- 万维网（World Wide Web, WWW）并非某种特殊的计算机网络。它是一个大规模的、联机式的信息储藏所，是运行在因特网上的一个分布式应用。
- 万维网利用网页之间的超链接将不同网站的网页链接成一张逻辑上的信息网。
- 浏览器最重要的部分是渲染引擎，也就是浏览器内核。负责对网页内容进行解析和显示。
  - 不同的浏览器内核对网页内容的解析也有不同，因此同一网页在不同内核的浏览器里的显示效果可能不同；
  - 网页编写者需要在不同内核的浏览器中测试网页显示效果。



### 6.7.2. 统一资源定位符URL

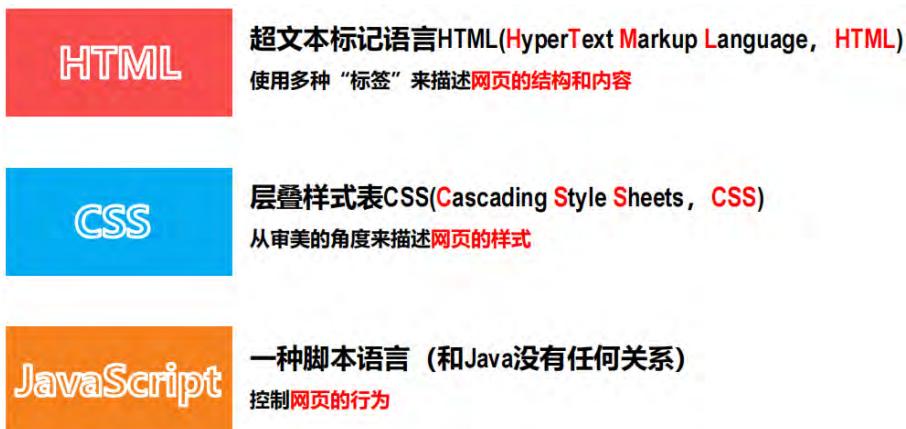
- Uniform Resource Locator
  - 指明因特网上任何种类“资源”的位置。
  - 可以方便地访问在世界范围的文档。

**■ URL的一般形式由以下四个部分组成：**

< 协议 > : / / < 主机 > : < 端口 > / < 路径 >

**http://www.hnust.cn:80/ggtz/119945.htm**

### 6.7.3. 万维网文档



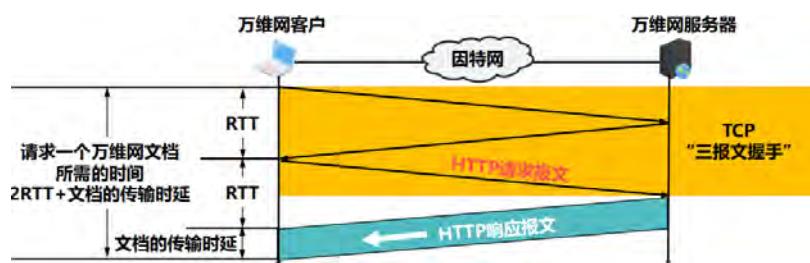
### 6.7.4. 超文本传输协议HTTP

- HyperText Transfer Protocol
  - HTTP定义了浏览器（即万维网客户进程）怎样向万维网服务器请求万维网文档，以及万维网服务器怎样把万维网文档传送给浏览器。



#### 6.7.4.1. HTTP/1.0

- 非持续连接：每次浏览器要请求一个文件都要与服务器建立TCP连接，当收到响应后就立即关闭连接。
  - 每请求一个文档就要有两倍的RTT的开销。若一个网页上有很多引用对象（例如图片等），那么请求每一个对象都需要花费2RTT的时间。
  - 为了减小时延，浏览器通常会建立多个并行的TCP连接同时请求多个对象。但是，这会大量占用万维网服务器的资源，特别是万维网服务器往往要同时服务于大量客户的请求，这会使负担很重。

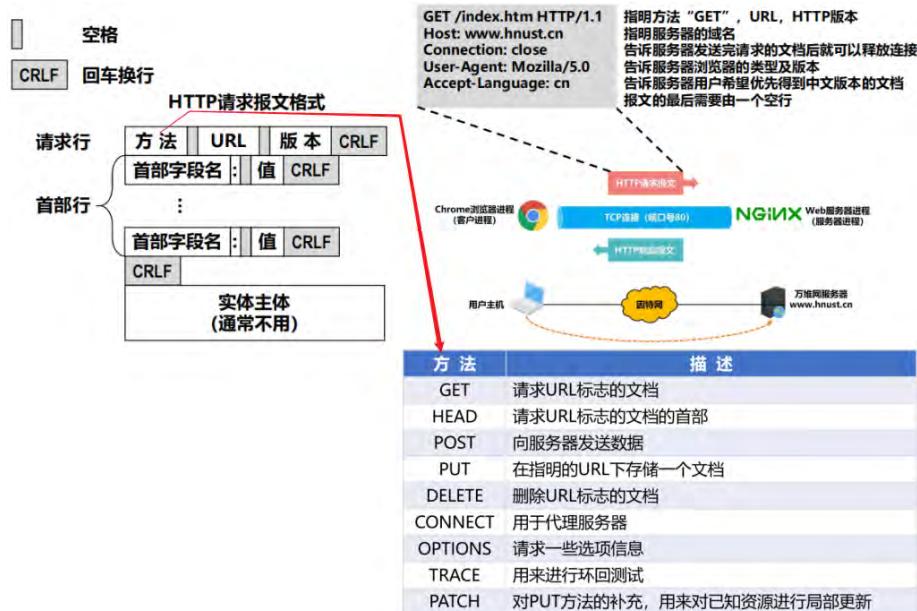


#### 6.7.4.2. HTTP/1.1

- 持续连接：万维网服务器在发送响应后仍然保持这条连接，使同一个客户（浏览器）和该服务器可以继续在这条连接上传送后续的HTTP请求报文和响应报文。这并不局限于传送同一个页面上引用的对象，而是只要这些文档都在同一个服务器上就行。
  - 为了进一步提高效率，HTTP/1.1的持续连接还可以使用流水线方式工作，即浏览器在收到HTTP的响应报文之前就能够连续发送多个请求报文。这样的一个接一个的请求报文到达服务器后，服务器就发回一个接一个的响应报文。这样就节省了很多个RTT时间，使TCP连接中的空闲时间减少，提高了下载文档的效率。

#### 6.7.4.3. HTTP报文格式

- HTTP是面向文本的，其报文中的每一个字段都是一些ASCII码串，并且每个字段的长度都是不确定的。
- 请求报文格式
  - 判断是否为持续连接要看报文中Connection的值，close为非持续连接，keep-alive为持续连接；而不能只是看HTTP版本号。

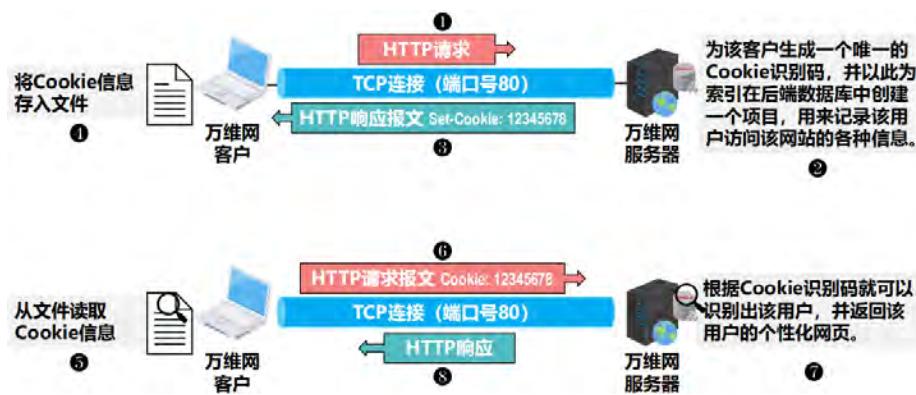


#### • 响应报文格式



#### 6.7.5. 使用Cookie在服务器上记录信息

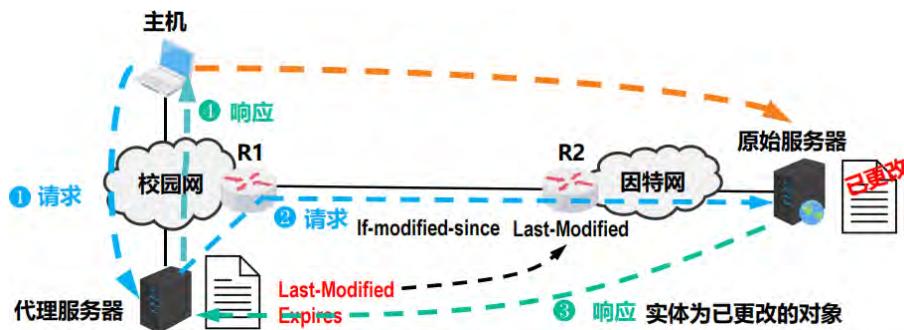
- 早期用户在万维网上仅查看存放在不同服务器上的各种静态文档。
- HTTP被设计为一种无状态的协议。
- Cookie是一种对无状态的HTTP进行状态化的技术，提供了一种机制使万维网服务器能够“记住”用户。



#### 6.7.6. 万维网缓存与代理服务器

- 万维网中可以使用缓存机制以提高万维网的效率。
- 万维网缓存又称为Web缓存 (Web Cache)，可位于客户机，也可位于中间系统上，位于中间系统上的Web缓存又称为代理服务器 (Proxy Server)。

- Web缓存把最近的一些请求和响应暂存在本地磁盘中。当新请求到达时，若发现这个请求与暂时存放的请求相同，就返回暂存的响应，而不需要按URL的地址再次去因特网访问该资源。
  - 若Web缓存的命中率比较高，则大大减少了该链路上的通信量因而减少了访问因特网的时延。
  - 代理服务器会给每一个响应对象设定一个修改时间字段（Last-Modified）和有效时间字段（Expires）
    - 若有效时间过期，代理服务器会给原始服务器发送包含if-modified-since的请求，原始服务器通过文档修改日期的对比就可判断出代理服务器的文档是否与当前文档一致。
    - 一致，发回304 Not Modified响应，代理服务器更新文档有效日期，封装在响应报文中发回主机。
    - 不一致，发回封装新文档的响应报文，代理服务器再将该文档封装再响应报文中发回主机。



## 6.8. 题目

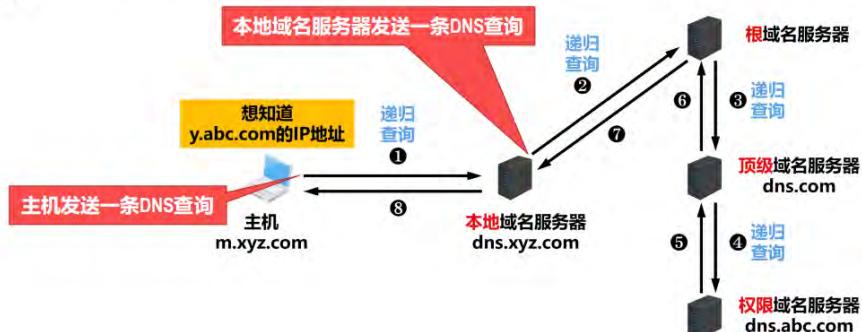
### 6.8.1. 域名解析

#### 6.8.1.1. 【2010 40】

【2010年题40】如果本地域名服务器无缓存，当采用递归方法解析另一网络某主机域名时，用户主机、本地域名服务器发送的域名请求消息数分别为（A）。

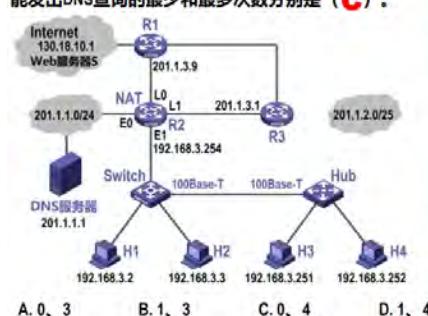
- A. 一条、一条      B. 一条、多条      C. 多条、一条      D. 多条、多条

解析



#### 6.8.1.2. 【2016 40】

【2016年题40】假设所有域名服务器均采用迭代查询方式进行域名解析，当H4访问规范命名为www.abc.xyz.com的网站时，域名服务器201.1.1.1在完成该域名解析过程中，可发出DNS查询的最少和最次数分别是（C）。



解析

- 若主机中已有缓存，则无需向本地域名服务器查询。
- 若主机中没有缓存，则需要向本地域名服务器递归查询。
- 若本地域名服务器中已有缓存，则无需向其他域名服务器查询（共1次）。
- 若本地域名服务器中没有缓存，则需要进行一系列迭代查询（共4次）。



## 6.8.2. FTP

### 6.8.2.1. 【2009 40】

【2009年 题40】FTP客户和服务器间传递FTP命令时，使用的连接是（A）。

- A. 建立在TCP之上的控制连接
- B. 建立在TCP之上的数据连接
- C. 建立在UDP之上的控制连接
- D. 建立在UDP之上的数据连接

#### 解析

FTP客户和服务器之间要建立以下两个并行的TCP连接：

一个是控制连接，在整个会话期间一直保持打开，用于传送FTP相关控制命令。

另一个是数据连接，用于文件传输，在每次文件传输时才建立，传输结束就关闭。

### 6.8.2.2. 【2017 40】

【2017年 题40】下列关于FTP协议的叙述中，错误的是（C）。

- A. 数据连接在每次数据传输完毕后就关闭
- B. 控制连接在整个会话期间保持打开状态
- C. 服务器与客户端的TCP 20端口建立数据连接
- D. 客户端与服务器的TCP 21端口建立控制连接

#### 解析

FTP客户和服务器之间要建立“控制连接”和“数据连接”这两个并行的TCP连接。控制连接在整个会话期间都保持打开状态，而数据连接在每次文件传输时才建立，传输结束就关闭。

默认情况下，FTP使用TCP 21端口进行控制连接，TCP 20端口进行数据连接。

但是，是否使用TCP 20端口建立数据连接与传输模式有关，主动方式使用TCP 20端口，被动方式由服务器和客户端自行协商决定。

## 6.8.3. 电子邮件

### 6.8.3.1. 【2012 40】

【2012年 题40】若用户1与用户2之间发送和接收电子邮件的过程如下图所示，则图中①、②、③阶段分别使用的应用层协议可以是（D）。

- A. SMTP、SMTP、SMTP
- B. POP3、SMTP、POP3
- C. POP3、SMTP、SMTP
- D. SMTP、SMTP、POP3



#### 解析

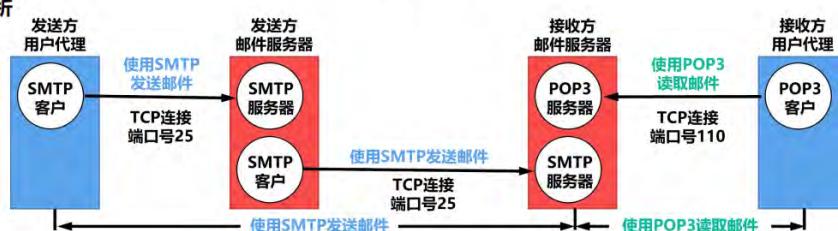


### 6.8.3.2. 【2013 40】

【2013年 题40】下列关于SMTP协议的叙述中，正确的是（A）。

- I. 只支持传输7比特ASCII码内容 ✓
  - II. 支持在邮件服务器之间发送邮件 ✓
  - III. 支持从用户代理向邮件服务器发送邮件 ✓
  - IV. 支持从邮件服务器向用户代理发送邮件 ✗
- A. 仅I、II和III
  - B. 仅I、II和IV
  - C. 仅I、III和IV
  - D. 仅II、III和IV

#### 解析

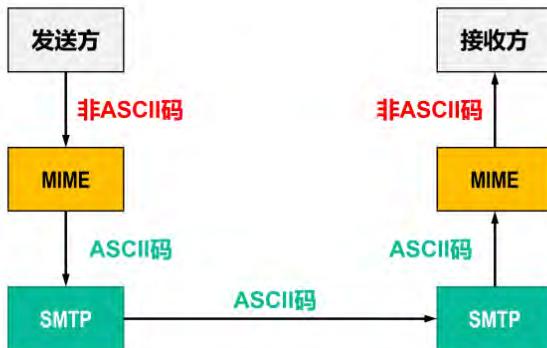


## 6.8.3.3. 【2018 40】

【2018年 题40】无需转换即可由SMTP协议直接传输的内容是 ( D )。

- A. JPEG图形    B. MPEG视频    C. EXE文件    D. ASCII文本

解析



## 6.8.4. 万维网

## 6.8.4.1. 【2015 40】

【2015年 题40】某浏览器发出的HTTP请求报文如下

```
GET /index.html HTTP/1.1
Host: www.test.edu.cn
Connection: Close
Cookie: 123456
```

- 下列叙述中，错误的是 (C)。
- 该浏览器请求浏览index.html
  - index.html存放在www.test.edu.cn上
  - 该浏览器请求使用持续连接
  - 该浏览器曾经浏览过www.test.edu.cn

解析

请求行：指明方法GET，URL，HTTP版本  
首部行的开始：指明服务器的域名

告诉服务器发送完请求的文档后就可释放连接，即非持续连接；  
若是持续连接方式，取值应为keep-alive，而不是Close

这是服务器为浏览器生成的Cookie识别码，表明该浏览器曾经访问过www.test.edu.cn

## 6.8.4.2. 【2011 47(3)改】

【修改自 2011年 题47 第 (3) 问】假设HTTP1.1协议以持续的非流水线方式工作，一次请求-响应的时间为RTT，rfc.html页面引用了2个JPEG小图像，则浏览器从开始建立TCP连接到收到全部内容为止，需要多少个RTT？

