

- 哈工大本部 2023 秋计算网络笔记

- 0 计算机网络概述

- 0.1 计算机网络基本概念
  - 0.1.1 计算机网络
  - 0.1.2 网络协议
- 0.2 计算机网络结构
  - 网络边缘
  - 接入网络
    - 数字用户线路 (DSL)
    - 电缆网络
    - 以太网
    - 无线接入网络
- 0.3 网络核心
  - 数据交换
- 0.4 计算机网络性能
- 0.5 计算机网络体系结构

- 1 网络层

- 1.1 网络应用体系结构
- 1.2 网络应用进程通信
- 1.3 Web 应用
  - 1.3.1 web 构成
  - 1.3.2 HTTP 协议
- 1.4 Email 应用
- 1.5 DNS
- 1.6 P2P 应用
- 1.7 Socket 编程 (略)
- 1.8 课上补充内容
  - 1.8.1 FTP (文件传输协议)
  - 1.8.2 DNS 攻击
  - 1.8.3 视频流和内容分发网

- 2 传输层

- 2.1 传输层服务
- 2.2 复用和分用
- 2.3 无连接传输协议-UDP
- 2.4 可靠数据传输基本原理
  - 2.4.1 Rdt1.0
  - 2.4.2 Rdt2.0 (ARQ)

- 2.4.3 Rdt2.1
- 2.4.4 Rdt2.2
- 2.4.5 Rdt3.0
- 2.5 滑动窗口协议
- 2.6 面向连接传输协议-TCP
  - 2.6.1 TCP 协议
  - 2.6.2 TCP 段结构
  - 2.6.3 TCP 的可靠数据传输
  - 2.6.4 TCP 的流量控制
  - 2.6.5 TCP 性能分析
  - 2.6.5 TCP 连接管理
- 2.7 拥塞控制原理
- 2.8 TCP 拥塞控制
- 2.9 补充内容
  - 2.9.1 糊涂窗口综合征
  - 2.9.2 TCP 连接控制补充
- 3 网络层
  - 3.1 网络层服务
  - 3.2 IPv4协议
  - 3.3 CIDR与路由聚集
  - 3.4 DHCP协议
  - 3.5 NAT协议
  - 3.6 ICMP协议
  - 3.7 IPv6
  - 3.8 路由算法
  - 3.9 Internet路由
- 4 数据链路层
  - 4.1 数据链路层服务
  - 4.2 差错编码
  - 4.3 多路访问协议
  - 4.4 ARP协议
  - 4.5 以太网
  - 4.6 PPP协议
- 5 物理层
  - 5.1 数据通信基础
  - 5.2 物理介质
  - 5.3 信道与信道容量
  - 5.4 基带传输基础

- 5.5 频带传输基础
- 5.6 物理层接口规程
- 5.7 802.11无线局域网
- 夹带私货

# 哈工大本部 2023 秋计算网络笔记

---

主要是 MOOC 上的 PPT+黑皮书+王道+课堂 PPT

笔记结构基本按照 MOOC 上的顺序来的，在每一部分补充王道和课堂 PPT 内容

预祝大家取得好成绩

---

## 0 计算机网络概述

---

### 0.1 计算机网络基本概念

#### 0.1.1 计算机网络

计算机网络是通信技术与计算机技术紧密结合的产物，就是一种通信网络。

计算机网络就是互连的、自治的计算机集合。

- 互连：互联互通
- 自治：无主从关系

当计算机距离远、数量大时需要通过交换网络互联主机。

计算机网络的组成

- 从组成部分上看
  - 硬件：主机（端系统）、通信链路、交换设备和通信处理机（如网卡）等组成
  - 软件：实现资源共享的软件和方便用户使用的软件
  - 协议
- 从工作方式上看
  - 边缘网络
  - 核心网络
- 从功能组成上看
  - 通信子网：传输介质、通信设备、相应的网络协议
  - 资源子网：实现资源共享功能的设备及其软件的集合

计算机网络功能

- 数据通信，最基本最重要
- 资源共享
- 分布式处理
- 提高可靠性
- 负载均衡

## 计算机网络分类

- 分布范围
  - 广域网（WAN）
  - 城域网（MAN）
  - 局域网（LAN）
  - 个人局域网（PAN）
- 传输技术
  - 广播式网络
  - 点对点网络
- 拓扑结构
  - 总线型网络
  - 星型网络
  - 环形网络
  - 网状网络
- 使用者
  - 公用网
  - 专用网
- 交换技术
  - 电路交换网络
  - 报文交换网络
  - 分组交换网络
- 传输介质
  - 有线网
  - 无线网

## 什么是 Internet:

- 组成细节角度：ISP 网络互连的“网络之网络”
  - 计算机设备的集合
  - 通信链路
  - 分组交换
- 服务角度：

- 为网络应用提供通信服务的通信基础设施
- 为网络应用提供应用编程接口

### 0.1.2 网络协议

网络协议(**network protocol**)，简称为协议，是为进行网络中的数据交换而建立的规则、标准或约定。规定了通信实体之间所交换的信息的格式、意义、顺序以及针对收到的信息或发生的事件所采取的“动作”。协议规范了网络中所有信息发送和接收过程

协议三要素：

1. 语法
2. 语义
3. 时序

## 0.2 计算机网络结构

网络边缘

- 主机（端系统）
  - 位于网络边缘
  - 可分为客户和服务器，客户/服务器应用模型(**CS**)：客户发送请求，接受服务器响应
  - 对等应用模型(**P2P**)：无（不仅依赖）专用服务器，通信在对等实体之间直接进行
- 网络应用

接入网络

将端系统物理连接到其边缘路由器的网络

数字用户线路 (**DSL**)

利用已有的电话线连接中心局的 **DSLAM**

- 数据通信通过 DSL 电话线接入 Internet
- 电话通过 DSL 电话线接入电话网
- <2.5Mbps 上行传输速率
- <24Mbps 下行传输速率

使用频分多路复用技术，在不同频带上传输不同频道

- 50kHz~1MHz 用于下行
- 4kHz~50kHz 用于上行
- 0kHz~4kHz 用于传统电话

电缆网络

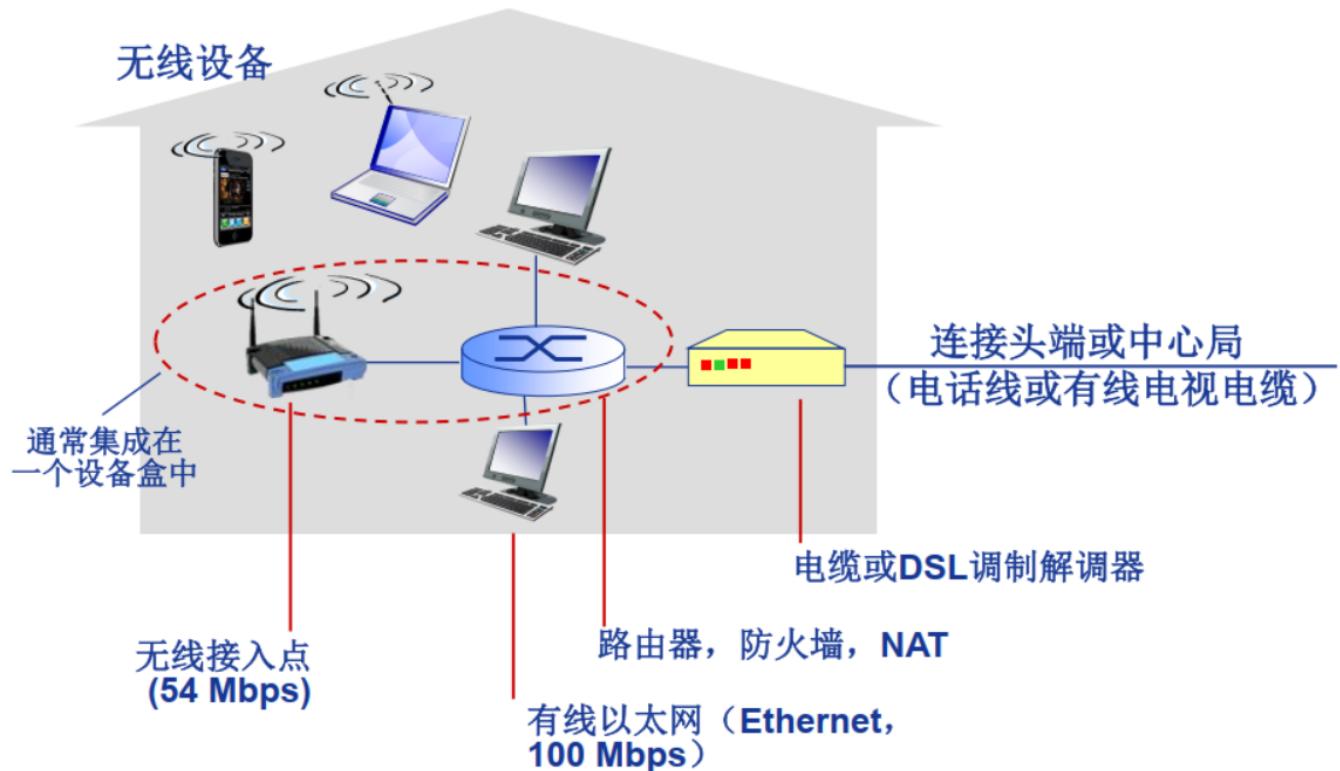
利用混合光纤同轴电缆（HFC）进行连接

- 30Mbps 下行传输速率
- 2Mbps 上行传输速率

各家庭通过电缆网络->光纤接入 ISP 路由器

- 各家庭共享家庭到电缆头端的接入网络
- DSL 独占至中心局的接入网络

典型的家庭接入网络



以太网

- 机构企业常用
- 典型传输速率： 10 Mbps, 100Mbps, 1Gbps, 10Gbps
- 目前，端系统通常直接连接以太网交换机（switch）

无线接入网络

通过共享的无线接入网络连接端系统和路由器

- 无线局域网（**LANs**）：
  - 同一建筑物内
  - 11Mbps、54Mbps 传输速率
- 广域无线接入网络：
  - 通过运营商的蜂窝网络，接入范围在几十公里
  - 1Mbps、10Mbps、100Mbps 传输速率

## 0.3 网络核心

互联的路由器网络，关键功能是：路由和转发

- 路由：确定分组从源到目的地的传输路径
- 转发：将分组从路由器的输入端口交换至正确的输出端口

### 数据交换

数据交换的类型

- 电路交换
  - 最典型的电路交换网络：电话网络
  - 三个阶段
    - 建立连接
    - 通信
    - 释放连接
  - 独占线路资源
  - 如何共享中继线-多路复用：将链路/网络资源划分为资源片
    - 将资源片分配给各路呼叫
    - 每路呼叫独占分配到的资源片进行通信
    - 资源片可能闲置
  - 典型多路复用方法
    - 频分多路复用-**FDM**（调制到不同频率）
      - 各用户占用不同的带宽资源（频率带宽）
    - 时分多路复用-**TDM**（调制到不同时间）
      - 将时间划分为等长的时分复用帧，每个用户在每个时分复用帧中占用固定序号的时隙
      - 在不同时间占用相同的频带宽度

- 波分多路复用-WDM (调制到不同波长)
  - 光的频分复用
- 码分多路复用-CDM (调制到不同编码)
  - 详见物理层
- 报文交换
  - 报文: 源发送的信息整体
- 分组交换
  - 分组: 将报文拆分成一系列相对较小的数据包
  - 分组交换需要进行报文的拆分与重组, 产生额外开销

## 报文交换和分组交换

- 报文交换和分组交换都采用存储-转发机制
  - 报文交换以报文为基本单位
  - 分组交换以分组为基本单位
- 假设
  - 报文大小为  $M \text{ bits}$
  - 链路带宽为  $R \text{ bps}$
  - 分组长度为  $L \text{ bits}$
  - 跳步数  $h$
  - 路由器数  $n$
- 报文交换的交付时间
  - $hM/R$
- 分组交换的交付时间 (参考流水线的完成时间: 流出时间+流水段数\*流水段时延)
  - $M/R + nL/R$

## 分组交换和电路交换

- 分组交换允许更多的用户同时使用网络
- 分组交换适用于突发数据传输网络, 可能产生拥塞

## 分组交换的优点

- 高效：动态分配传输带宽，对通信链路是逐段占用
- 灵活：以分组为传输单位和查找路由
- 迅速：不必先建立连接就能向其他主机发送分组
- 可靠：保证可靠性的网络协议；分布式的路由选择协议使网络有很好的生存性

分组交换的问题

- 分组在各节点存储转发需要排队，会造成一定的时延
- 分组必须携带首部也造成了一定的开销

## 0.4 计算机网络性能

速率

- 单位时间传输信息量
- 最重要的一个性能指标
- 往往指的是额定速率或标称速率

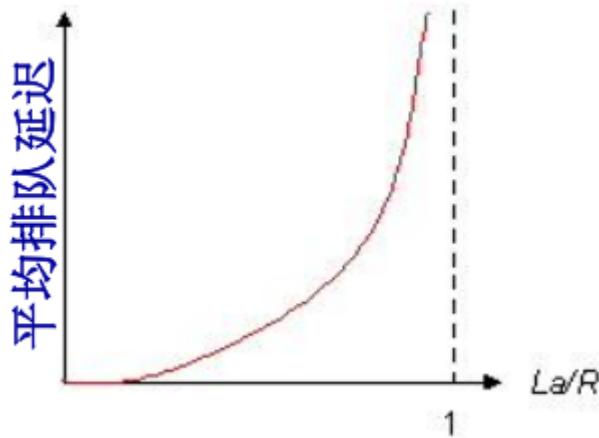
带宽

- 数字信道所能传送的最高数据率

分组时延

- 产生原因
  - 分组到达速率超出链路容量
  - 分组排队
- 分类
  - $d_{proc}$  结点处理延迟
    - 差错检测
    - 确定输出链路
  - $d_{queue}$  排队延迟
    - 等待输出链路可用
    - 取决于路由器拥塞程度
    - R: 链路带宽

- L: 分组长度
- a: 平均分组到达速率
- 流量强度= $La/R$
- 流量强度与平均排队时延的关系



- $d_{trans}$  传输延迟
  - L: 分组长度
  - R: 链路带宽
  - $d_{trans} = L/R$
- $d_{prop}$  传播延迟
  - d: 物理链路长度
  - s: 信号传播速度
  - $d_{prop} = d/s$
- $d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

## 时延带宽积

- 时延带宽积=传播时延\*带宽
- 以比特为单位的链路长度

## 分组丢失

- 队列缓存容量有限
- 分组到达已满队列将被丢弃
- 丢弃分组由前序结点或源重发（也有可能不重发）
- 丢包率=丢包数/已发送分组数

## 吞吐量

- 表示在发送端与接收端之间传输数据速率
  - 即时吞吐率：给定时间的速率
  - 平均吞吐率：一段时间的平均速率
- 瓶颈链路
  - 端到端路径上，限制端到端吞吐率的链路

信道利用率：有数据通过时间/总工作时间

## 0.5 计算机网络体系结构

- 网络体系结构是从功能上描述计算机网络结构
- 网络体系结构使分层结构
- 每层遵循某个/些网络协议完成本层功能
- 加算计网络体系结构是计算机网络的各层及其协议的集合
- 体系结构是一个计算机网络的功能层次及其关系的定义
- 体系结构是抽象的

分层网络结构的基本概念

- 实体(entity) 表示任何可发送或接收信息的硬件或软件进程。
- 协议是控制两个对等实体进行通信的规则的集合，协议是“水平的”。
- 任一层实体需要使用下层服务，遵循本层协议，实现本层功能，向上层提供服务，服务是“垂直的”。
- 下层协议的实现对上层的服务用户是透明的。
- 同系统的相邻层实体间通过接口进行交互，通过服务访问点 SAP(Service Access Point)，交换原语，指定请求的特定服务。

OSI 参考模型



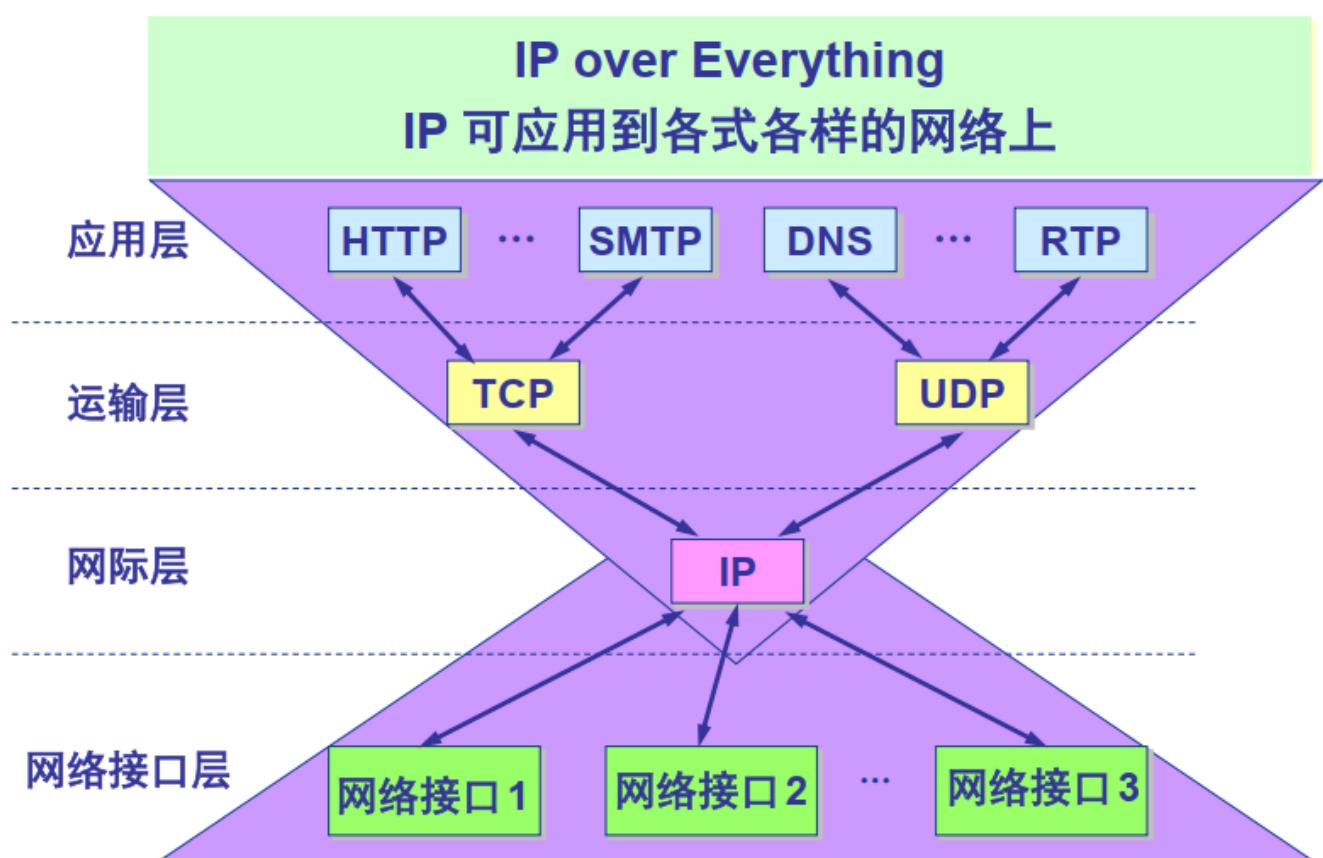
端-端层：7-4 层

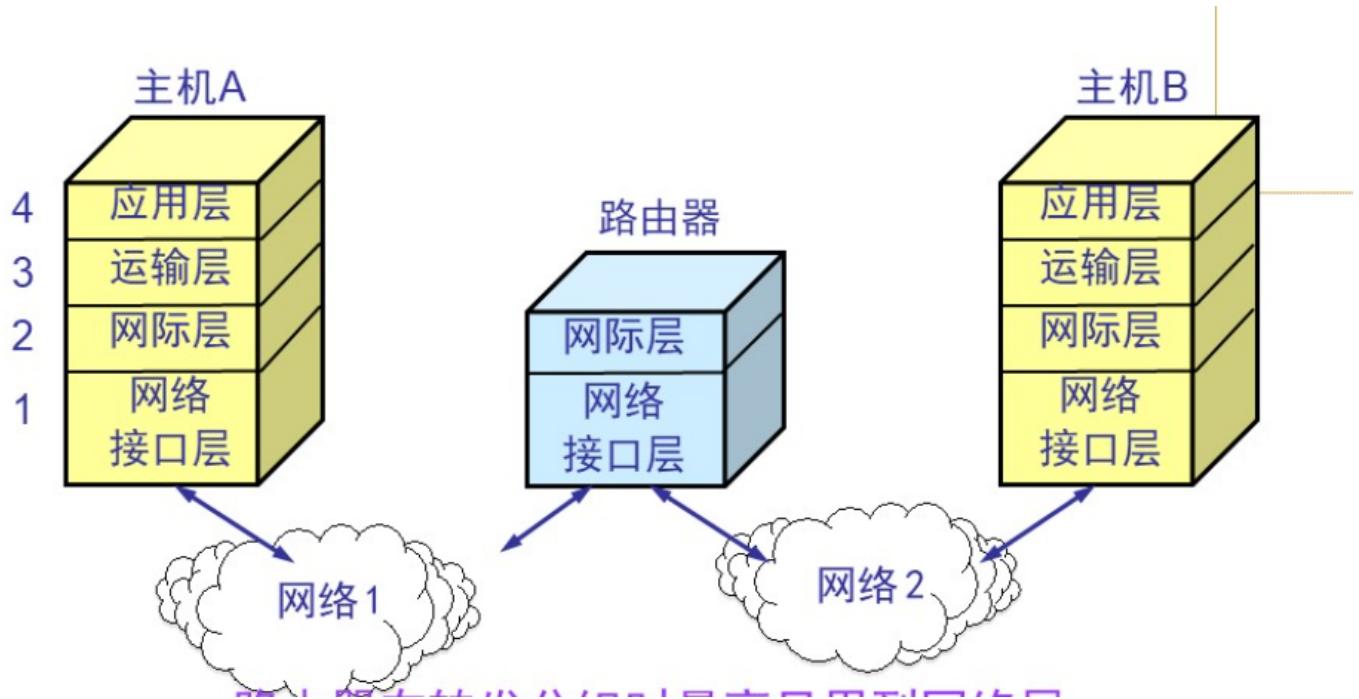
各个层次的功能

- 物理层
  - 接口特性
    - 机械特性、电器特性、功能特性、规程特性
  - 比特编码
  - 数据率
  - 比特同步
    - 时钟同步
  - 传输模式
    - 单工
    - 半双工
    - 全双工
- 数据链路层
  - 负责结点-结点数据传输
  - 组帧
  - 物理寻址
  - 流量控制
  - 差错控制：检测差错并重传、避免重复帧
  - 访问控制：决定物理介质的使用权
- 网络层
  - 负责源主机到目的主机数据分组交付

- 逻辑寻址
- 路由
- 分组转发
- 传输层
  - 负责端到端（进程间）报文传输
  - 分段与重组
  - SAP 寻址
  - 连接控制
  - 流量控制
  - 差错控制
- 会话层
  - 对话控制
  - 同步
- 表示层
  - 数据表示转化
  - 加密/解密
  - 压缩/解压缩
- 应用层
  - 支持用户通过用户代理（浏览器）或网络接口使用网络

## TCP/IP 参考模型





路由器在转发分组时最高只用到网际层而没有使用传输层和应用层

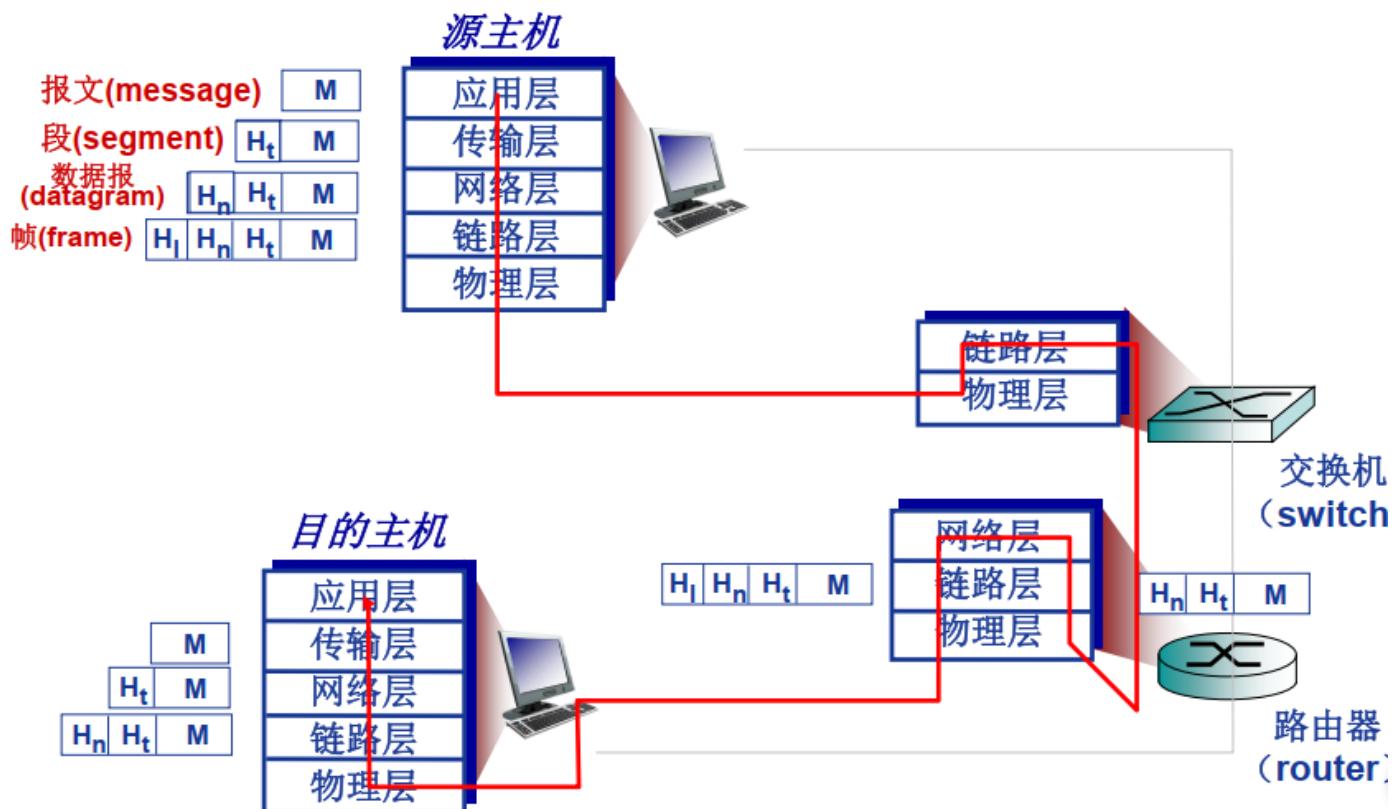
## 5 层参考模型



- 应用层：支持各种网络应用
- 传输层：进程到进程的数据传输
  - TCP/IP 的可靠性传输是在这层实现的
- 网络层：源主机到目的主机的数据分组路由转发
  - 实现跨越网络通信时的路径选择和数据传输
  - 硬件上依靠路由器实现路径选择和异构网络的协议转换
  - 软件上 IP 协议实现了同一的 IP 编址

- 链路层：相邻网络元素（主机、交换机、路由器，也称结点）的数据传输
- 物理层：比特传输
  - 面向通信
  - 无线传输
  - 数字调制以及多路复用

## 5 层参考模型的数据封装



链路层不仅加首部也加尾部

协议是水平的，服务是垂直的

- 对等层通信
  - 两个同样的层次把 PDU（数据单元加控制信息）通过水平虚线直接传递给对方
- 协议的水平
  - 各层协议实际上就是在各个对等层之间传递数据时的各项规定
- 服务的垂直
  - 在协议的控制下，两个对等实体之间的通信使得本层能够向上一层提供服务

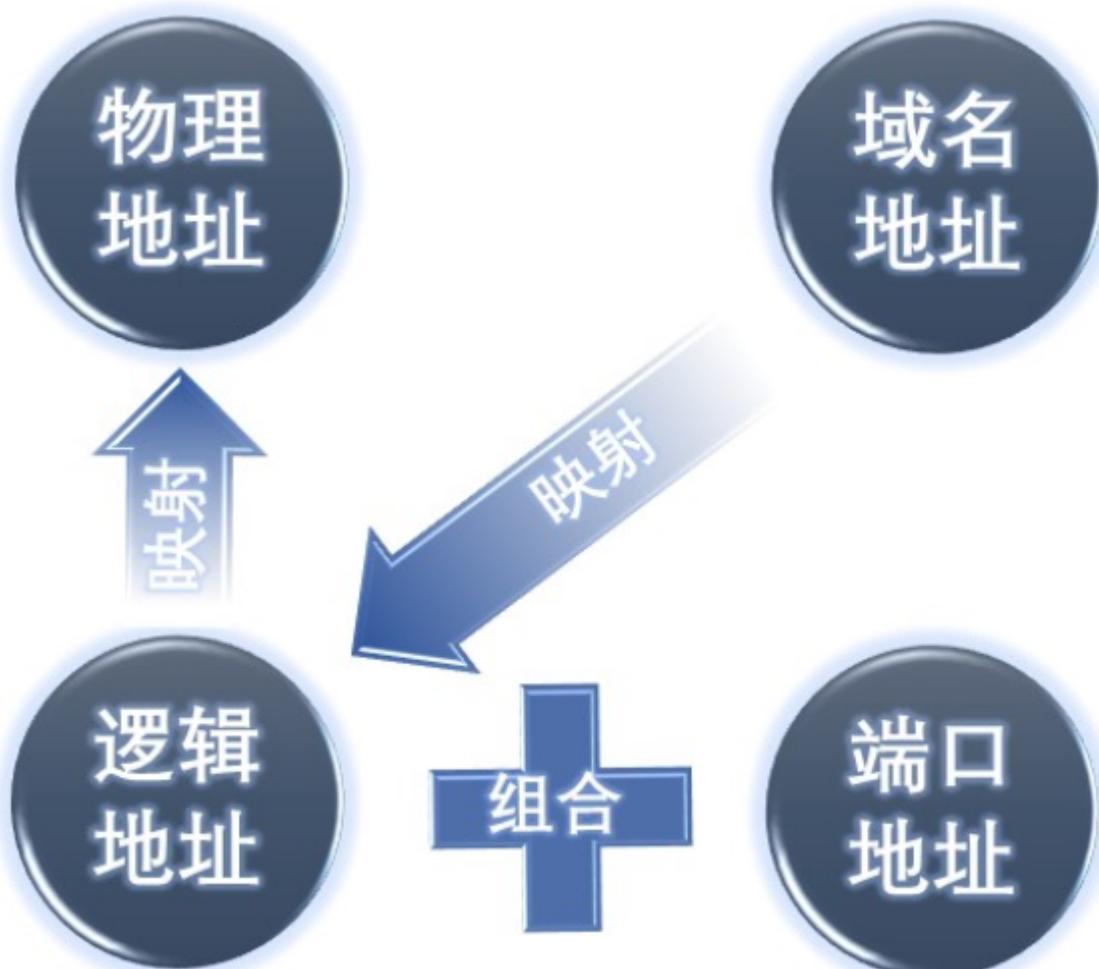
## 封装

- 应用数据沿协议栈向下流动
- 在每层都会附加相应的首部
- 首部中最重要的信息之一是地址

- 除此之外还有各种控制信息
- 每层协议使用的地址不同
- 首部只能由对等层实体拆除

## 网络中的几类地址

- 物理地址
  - 网络接口层（链路层和物理层）使用，主要指以太网使用的 MAC 地址
- 逻辑地址
  - 网络层使用，IP 地址
- 端口地址
  - 传输层使用
  - 定位主机上的进程
- 域名地址
  - 应用层使用
  - 层次化，符号化地址
  - 解决 IP 地址难记和难使用的问题，无法直接用于计算机通信
  - 使用时需要域名映射到 IP 地址（DNS）
- 关系



网络中通信双方的进程标识

- IP 地址+端口号
- 网络中双方的通信可以用（本地 IP、本地端口、远程 IP、远程端口）

# 1 网络层

---

## 1.1 网络应用体系结构

- 客户机/服务器结构（C/S）
  - 服务器
    - 7\*24 小时提供服务
    - 永久性访问地址/域名
    - 利用大量服务器实现可拓展性
  - 客户机
    - 与服务器通信，使用服务器提供的服务
    - 间歇性接入网络
    - 可能使用动态 IP 地址
    - 不会与其他客户机直接通信
  - 集中管理方便、客户机之间不直接通信、可拓展性不佳
- P2P 结构
  - 没有永远在线的服务器
  - 任意端系统/节点之间可以直接通信
  - 节点间歇性接入网络
  - 节点可能改变 IP 地址
  - 优点：高度可伸缩
  - 缺点：难以管理
- 混合结构
  - 文件传输用 P2P 结构
  - 文件搜索用 C/S 结构
    - 每个节点向中央服务器登记自己的内容
    - 每个节点向中央服务器提交查询请求

## 1.2 网络应用进程通信

进程

- 主机上运行的程序
- 客户机进程：发起通信的进程
- 服务器进程：等待通信的进程
- 进程通信
  - 同一主机：进程间通信机制，操作系统提供
  - 不同主机：消息交换
- 用 IP 地址+端口号标识，进程寻址

套接字：同一主机内应用层与传输层之间的接口，可以向网络发送报文、接收报文

- 传输协议的选择
- 参数设置

## 应用层协议

- 网络应用需要遵循的应用层协议
- 公开协议
  - 由 RFC 定义
  - 允许互操作
- 私有协议
  - P2P 文件共享应用
- 协议内容
  - 消息的类型
    - 请求消息
    - 响应消息
  - 消息的语法
    - 字段及其描述
- 字段的语义
- 规则
  - 何时发送/相应消息
  - 如何发送/相应消息

## 网络应用对传输服务的需求

- 可靠的数据传输
- 安全性
- 时间
- 吞吐量/带宽

## Internet 提供的传输服务

- TCP
  - 面向连接
  - 可靠数据传输
  - 流量控制
  - 拥塞控制
  - 不提供时间保障
  - 不提供最小带宽保障
- UDP
  - 无连接
  - 不可靠
  - 只提供差错检查和复用/分用

## 1.3 Web 应用

### 1.3.1 web 构成

网页、网页相互链接

网页包含多个对象

- 对象: **HTML** 文件、图片、视频文件、动态脚本等
- 基本 **HTML** 文件: 包含对其他对象引用的链接

对象的寻址

- **URL:** 统一资源定位器
  - 格式 Scheme://host:port/path

Web 应用遵循 **HTTP** 协议（超文本传输协议）

### 1.3.2 HTTP 协议

采用 C/S 结构

- 客户-**Browser:** 请求、接受、展示 Web 对象
- 服务器-**Web Server:** 相应客户请求、发送对象

版本

- 1.0
- 1.1

## 使用 TCP 传输服务

- 服务器在 80 端口等待客户请求
- 浏览器发起到服务器的 TCP 链接（创建套接字）
- 服务器接受来自浏览器的 TCP 链接
- 浏览器与服务器交换 HTTP 消息
- 关闭 TCP 链接

## 无状态服务

- 服务器不维护任何有关客户端过去所发请求的信息

## 连接类型

- 非持久性连接 1.0 版本
  - 每个 TCP 连接最多传输一个对象
  - 每个对象需要两个 RTT
  - 响应时间
    - 发起、建立 TCP 连接: 一个 RTT
    - 发送 HTTP 请求消息到响应消息的前几个字节到达: 一个 RTT
    - 文件传输时间
    - $\text{total}=2\text{RTT}+\text{文件发送时间}$
- 持久性连接: 每个 TCP 连接允许传输多个对象 1.1 版本
  - 不带流水线
    - 每个对象耗时 1 个 RTT
  - 带流水线（缺省模式）
    - 理想情况下，收到所有对象只需耗时约一个 RTT

## HTTP 消息格式

- 请求报文
  - 请求行: 方法、URL、版本\r\n
  - 请求头: 每行都是头部域名称、头部域值\r\n
  - 请求体: 请求数据。与请求头之间用\r\n隔开，请求行和请求头之间不用
- 响应报文
  - 相应行: 版本、状态码、原因短语
  - 请求头: 与请求报文相同
  - 响应数据

## 请求方法

- POST
- GET
- HEAD, 没有响应数据的 GET
- PUT, 上传文件, 1.1 版本才有
- DELETE, 删除文件, 1.1 版本才有

## 响应状态码

- 200 OK, 正常
- 301 Moved Permanently, 重定向
- 400 Bad Request, 请求中缺少必须信息
- 404 Not Found, 请求资源不存在
- 505 HTTP Version Not Supported

## Cookie 技术

- HTTP 无状态, 但是很多应用要求服务器掌握客户端状态
- 作用
  - 身份认证
  - 购物车
  - 推荐
- 风险
  - 信息泄露
- 组件
  - HTTP 响应消息的 cookie 头部行
  - HTTP 请求消息的 cookie 头部行
  - 保存在客户端主机上的 cookie 文件, 由浏览器管理
  - Web 服务器后台数据库

## Web 缓存/代理服务器技术

- 功能
  - 在不访问服务器的前提下满足客户端的 HTTP 请求
- 好处
  - 缩短客户请求时间
  - 减少组织流量
  - 在大范围内实现有效的内容分发

- 运行流程
  - 用户设定浏览器通过代理服务器进行 Web 访问
  - 代理服务器接收到用户请求
    - 如果请求对象在缓存中，直接返回缓存对象
    - 否则，向原始服务器发送 HTTP 请求，获取对象并返回给客户端同时缓存下来
- 代理服务器既充当客户端，也充当服务端，一般由 ISP 架设
- 缓存的更新
  - 如果缓存是最新版本，则直接返回
  - 在 HTTP 请求消息中加入 If-modified-since:<date>
  - 根据返回的状态码判断是否是最新版本：304 Not Modified

## 1.4 Email 应用

### Email 应用的构成

- 邮件客户端（用户代理）
  - 进行邮件撰写和编辑
- 邮件服务器
  - 发送、接收邮件、维护用户邮箱
- 电子邮件所需协议
  - SMTP、POP3

### 电子邮件传输协议

- SMTP 协议
  - 发送电子邮件，不涉及邮件管理
  - 一般不使用中间邮件服务器进行连接
  - 使用 TCP 进行 email 消息的可靠传输
  - 使用端口 25
  - 传输过程的三个阶段
    - 握手

- 消息传输
  - 关闭
- 命令/响应交互模式
  - 命令: ASCII 文本
  - 响应: 状态代码和语句
- Email 消息只能包含 7 位 ASCII 码
- 与 HTTP 对比

### 与HTTP对比:

- ❖ HTTP: 拉式(pull)
- ❖ SMTP: 退式(push)
- ❖ 都使用命令/响应交互模式
- ❖ 命令和状态代码都是ASCII 码
- ❖ HTTP: 每个对象封装在独立的响应消息中
- ❖ SMTP: 多个对象在由多个部分构成的消息中发送

- Email 消息格式
- 头部行
    - To
    - From
    - Subject
  - 消息体
    - 消息本身
    - 只能是 ASCII 字符
  - 针对多媒体的拓展, MIME
    - 通过在邮件头部增加额外的行声明 MIME 的内容类型

电子邮件访问协议

- POP

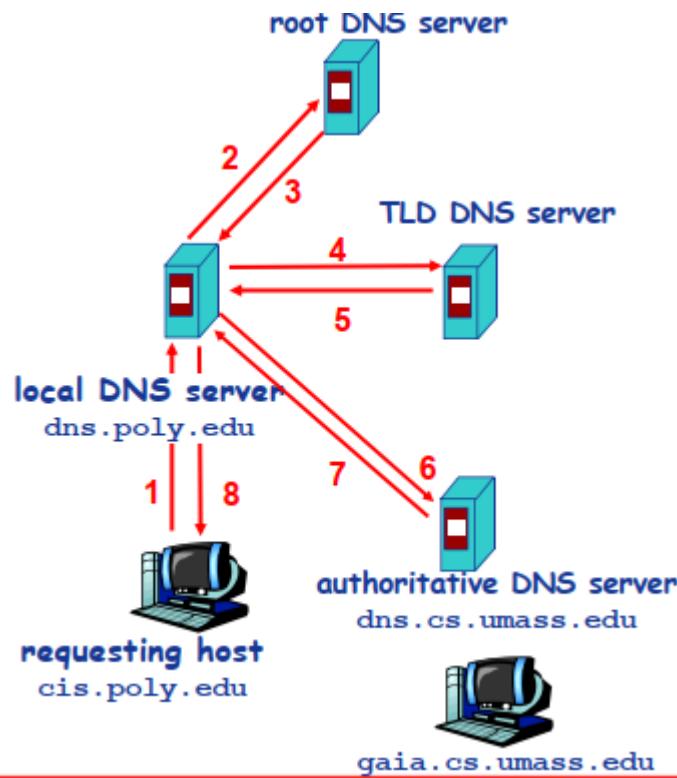
- 认证过程
  - 客户端命令
    - **User:** 用户名
    - **Pass:** 密码
  - 服务器响应
    - +OK
    - -ERR
- 事务阶段
  - **List:** 列出消息数量
  - **Retr:** 用编号获取消息
  - **Dele:** 删除消息
  - **Quit**
- 下载并删除：如果用户换了客户端软件，无法重读邮件
- 下载并保持：不同客户端都可以保留消息的拷贝
- POP3 是无状态的
- 端口号：110
- IMAP
  - 所有消息统一保存在服务器
  - 允许客户利用文件夹组织消息
  - 支持跨会话的用户状态
    - 文件夹的名字
    - 文件夹与消息 ID 之间的映射
- HTTP

## 1.5 DNS

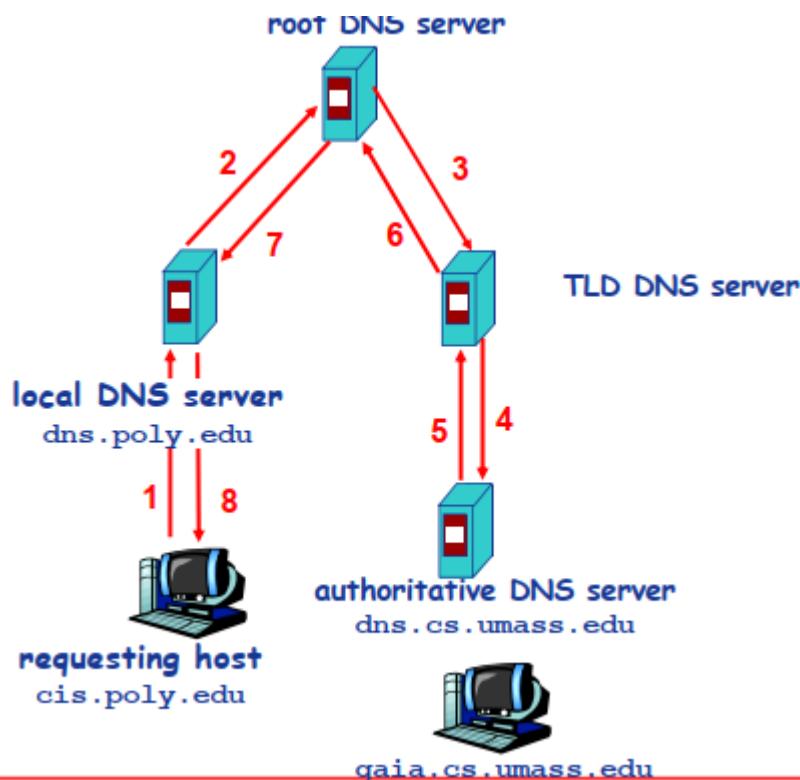
DNS (Domain Name System)

- 功能
  - 解决 Internet 上主机/路由器的识别问题
  - 域名到 IP 地址的映射
- 性质
  - 多层命名服务器构成的分布式数据库
    - 根域名服务器
    - 顶级域名服务器：负责 com, org, net, edu 等顶级域名和国家顶级域名

- 权威域名服务器：组织的域名解析服务器，提供组织内部服务器的解析服务
- 本地域名解析服务器
  - 不严格属于层次体系
  - 每个 ISP 都有一个本地域名解析服务器
  - 当主机进行 DNS 查询时，查询被发送到本地域名服务器
- 应用层协议：完成对域名的解析
- DNS 服务
  - 域名向 IP 地址的转换
  - 主机别名
  - 邮件服务器别名
  - 负载均衡
- 为什么不采用集中式 DNS
  - 单点失败问题
  - 流量问题
  - 距离问题
  - 维护性问题
- 分布式缺点
  - 多段访问，消耗时间和资源
- DNS 查询
  - 迭代式查询

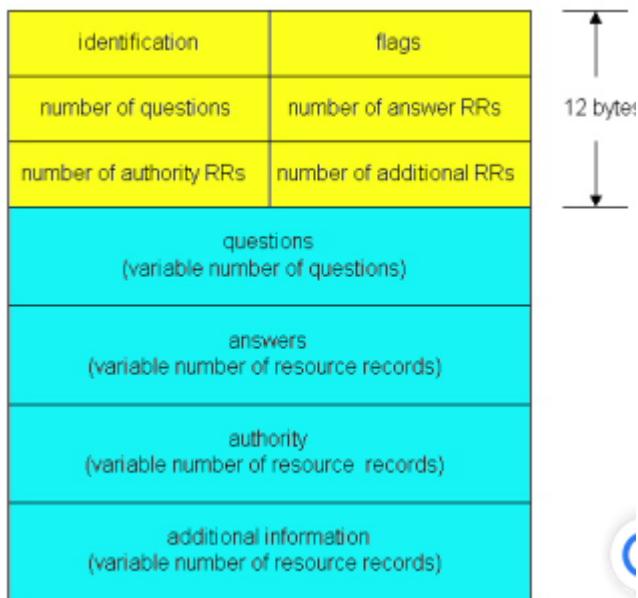


- 递归查询



- DNS 缓存-只要 DNS 服务器获得域名-IP 映射，即缓存这一映射
  - 一段时间后缓存失效
  - 本地域名服务器一般会缓存顶级域名服务器的映射，因此根域名服务器不经常被访问
- DNS 记录，资源记录 RR(name,value,type,ttl)
  - type=A

- **Name:** 主机域名
  - **Value:** IP 地址
  - **type=B**
    - **Name:** 域 (edu.cn)
    - **Value:** 该域权威域名解析服务器的主机域名
  - **type=CNAME**
    - **Name:** 某一真实域名的别名
    - **Value:** 真实域名
  - **type=MX**
    - **Value** 是别名为 name 的邮件服务器的规范主机名
  - **ttl:** 该记录的生存时间
- **DNS 协议与消息**
    - **DNS 报文:** 请求报文和响应报文相同格式



- 一个域名可以对应于多个 IP，主机有多个网卡
- 一个 IP 也可以对应于不同的域名

## 1.6 P2P 应用

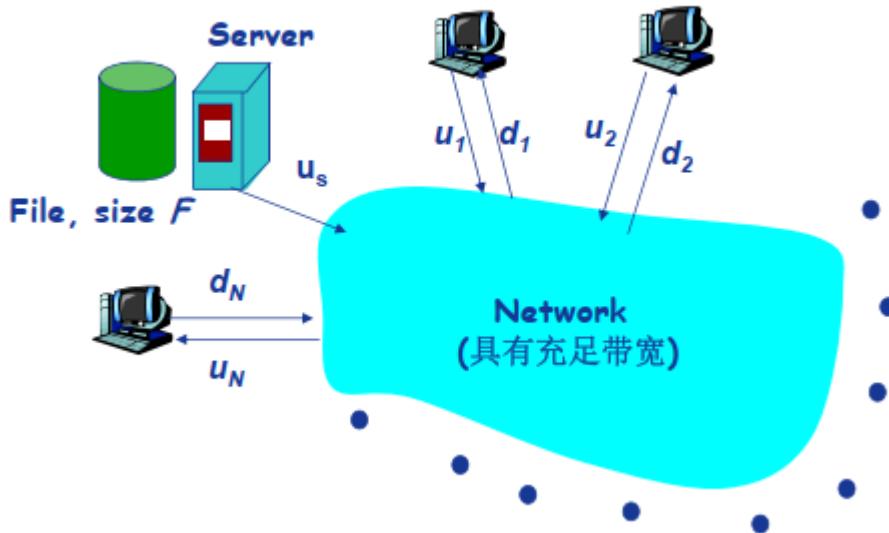
### 纯 P2P 架构

- 没有服务器
- 任意端系统之间直接通信
- 节点阶段性接入 Internet

- 节点可能更换 IP 地址

以文件分发为例

- 一个服务器向  $N$  个节点分发一个文件需要多长时间



- 假设

- $u_s$ : 服务器上传带宽
- $u_i$ : 节点*i*的上传带宽
- $d_i$ : 节点*i*的下载带宽

- C/S 结构

- 服务器串行发送  $N$  个副本,  $NF/u_s$
- 客户机*i*需要  $F/d_i$  时间下载
- 总共时间  $\max \{NF/u_s, F/\min d_i\}$

- P2P 结构

- 服务器必须发送一个副本,  $F/u_s$
- 客户机*i*需要  $F/d_i$  时间下载
- 总共需要下载  $NF$
- 最快的可能上传速率:  $u_s + \sum u_i$
- 总共时间  $\max \{F/u_s, F/\min d_i, NF/(u_s + \sum u_i)\}$

## P2P 协议-BitTorrent

- 每个文件划分为 256KB 的 chunk
- 有一个 tracker 跟踪参与 torrent 的节点

- **torrent:** 交换同一个文件的文件块的节点组
- 工作流程
  - 文件划分为 256KB 的 **chunk**
  - 节点加入 **torrent**
    - 初始时没有 **chunk**, 但是会逐渐累积
    - 向 **tracker** 注册以获取节点清单, 与某些节点建立连接
  - 节点获取 **chunk**
    - 在任一时刻, 不同节点拥有文件的不同 **chunk** 集合
    - 节点定期查询每个邻居所持有的 **chunk** 列表
    - 节点发送请求, 请求获取确实的 **chunk**-稀缺优先: 针对没有的块在邻居中决定最稀缺(副本数最少)的块来请求
  - 节点发送 **chunk**: **tit-for-tat**
    - 向正在向其发送 **chunk** 的邻居中选择速度最快的 4 个发送 **chunk**: 每十秒重新评估 **top4**
    - 每 30 秒随机选择一个其他节点向其发送 **chunk**, 这个节点可能加入 **top4**

## P2P 索引技术

- 信息到节点位置(**IP 地址+端口号**)的映射
- 集中式索引
  - 节点加入时, 通知中央服务器自身的 **IP** 地址和内容
  - 节点查找文件时, 向中央服务器发出查询, 返回拥有文件的节点的地址
  - 节点向拥有文件的节点发出请求
  - 内容和文件传输是分布式的, 但是内容定位是高度集中式的
    - 单点失效问题
    - 性能问题
    - 版权问题
- 洪泛式查询
  - 采用完全分布式架构, 每个节点只对它共享的文件进行索引
  - 查询
    - 查询信息通过已有的 **TCP** 连接发送
    - 节点转发查询信息
    - 如果查询命中则利用反向路径发回查询节点, 使用 **HTTP** 传输文件
- 层次式覆盖网络
  - 介于集中式索引和洪泛式查询之间的方法
  - 每个节点或者是一个超级节点, 或者被分配一个超级节点
    - 节点和超级节点之间维持 **TCP** 连接
    - 某些超级节点之间维持 **TCP** 连接

- 超级节点负责跟踪子节点的内容

## 1.7 Socket 编程（略）

## 1.8 课上补充内容

### 1.8.1 FTP（文件传输协议）

- 功能
  - 提供不同种类主机系统之间的文件传输能力
  - 以用户权限管理方式提供用户对远程 **FTP** 服务器上的文件管理能力
  - 以匿名 **FTP** 方式提供共用文件共享的能力
- 采用 **C/S** 的工作方式
- 使用 **TCP** 提供可靠数据传输服务
- 一个 **FTP** 服务器进程可以同时为多个客户进程提供服务。**FTP** 服务器进程由两大部分组成：一个主进程，负责接受新的请求；若干从属进程，处理单个请求
- 工作步骤
  - 服务器在 **21** 端口启动 **FTP** 服务（控制端口），等待客户端连接
  - 接受请求，启动从属进程处理客户进程发来的请求，主进程和从属进程并发执行
  - 传输完毕后由服务器关闭数据连接
- **FTP** 服务器维护状态：当前目录，身份认证
- **FTP** 在工作时使用两个并行的 **TCP** 连接：一个是控制连接（端口 **21**），一个是数据连接（端口 **20**）
  - 控制连接
    - 用来传递控制信息，控制信息以 **7** 位 **ASCII** 码传送
    - 在整个会话期间一直保持打开状态
  - 数据连接
    - 服务器的控制进程收到 **FTP** 请求后就创建数据传送进程和数据连接
    - 数据连接用来连接客户端和服务器端的数据传送进程
    - 数据连接有两种模式
      - 主动模式 **POST**：服务器使用 **20** 端口连接到用户的指定端口（客户端随机选择）
      - 被动模式 **PASV**：客户端使用端口连接到服务器的指定端口（服务器随机选择）
- 允许客户指明文件类型和格式

## 1.8.2 DNS 攻击

- DDoS 攻击
  - 用流量轰炸根服务器
    - 从未成功
      - 流量过滤
      - 本地 DNS 缓存顶级域名服务器 IP，导致绕过根服务器
    - 轰炸顶级域名服务器
      - 可能更危险
- 重定向攻击
  - man-in-middle
    - 拦截查询
  - DNS poisoning (DNS 污染)
    - 发送虚假的应答到 DNS 服务器
- 利用 DNS 实施 DDoS
  - 使用欺骗性源地址发送查询

## 1.8.3 视频流和内容分发网

### 因特网视频

- 视频：以恒定速率显示的图像序列
- 数字图像：像素陈列
  - 每位像素表示为若干比特
- 编码：在图像内部和图像之间使用冗余减少图像编码位数
  - 空间：对重复值的压缩
  - 时间：传输与上一帧的差异

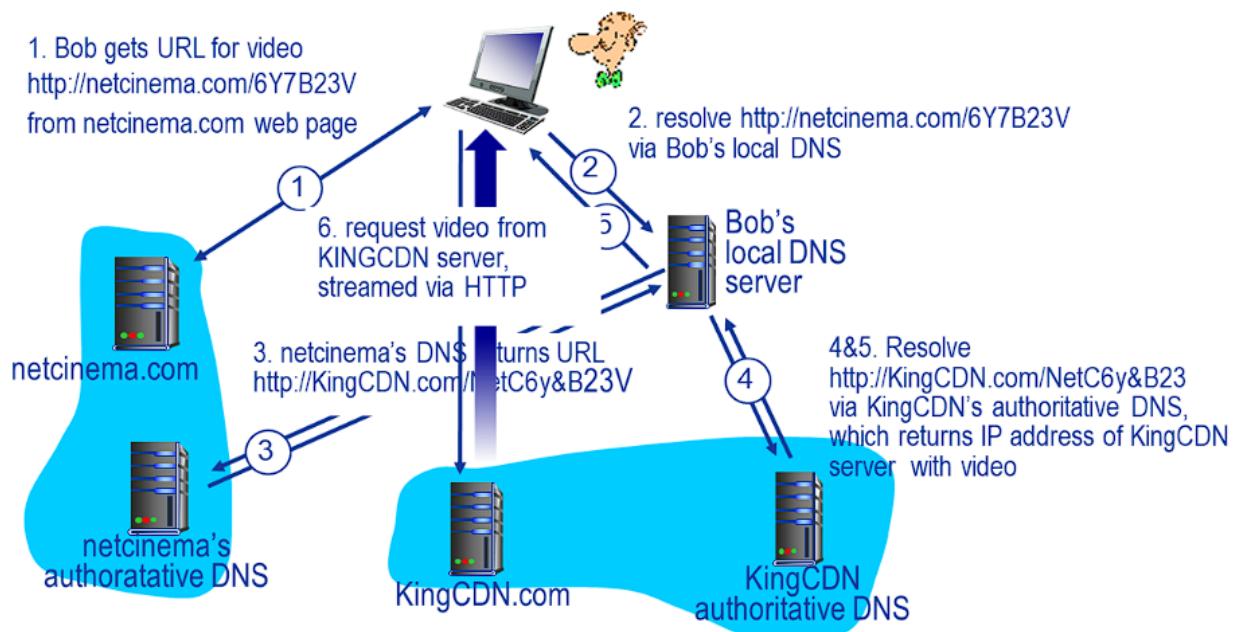
### HTTP 流和 DASH

- HTTP 流：请求视频时，客户端收到视频就播放同时缓存后面的视频
  - 缺点：所有客户收到相同编码的视频，没有考虑不同用户之间的差异
- DASH：经 HTTP 的动态适应性流
  - 服务器端
    - 将视频文件划分为块
    - 每个块以不同的速率存储、编码
    - 清单文件：为每个块提供 URL
  - 客户端
    - 定期测量服务器到客户端的带宽

- 查阅清单文件，一次请求一个块
  - 基于当前带宽、选择可持续的最大编码速率
  - 可以在不同时间点选择不同的编码速率

## 内容分发网 CDN

- 如何将内容流式传输到数十万并发客户
  - 单个巨大的服务器
    - 单点故障
    - 网络拥塞点
    - 距离问题
    - 重复发送
    - 无法拓展
  - 在多个地理位置分散的站点上提供多个视频副本 CDN
- CDN 部署原则
  - 深入：将 CDN 服务器深入到多个接入网
  - 带回家：在接入网附近的接入点部署少量较大的服务集群
- CDN 操作



- 集群选择策略，CDN 的核心

## 2 传输层

## 2.1 传输层服务

### 传输层服务和协议

- 传输层协议为运行在不同主机上的进程提供了一种逻辑通信机制
- 端系统运行传输层协议
- 传输层可以为应用提供多种协议
  - TCP
    - 可靠、按序
    - 拥塞控制
    - 流量控制
    - 连接建立
    - 不提供广播和组播
  - UDP
    - 不可靠
    - 这两种服务均不保证延迟和带宽
- 与网络层的区别
  - 逻辑通信机制
    - 网络层：主机之间
    - 传输层：应用进程之间
- 与网络层联系
  - 位于网络层之上
  - 依赖于网络层服务
  - 对网络层服务进行（可能的）增强

## 2.2 复用和分用

如果某层的一个协议对应上层的多个协议/实体，需要进行多路复用和分用

接收端进行多路分用，发送端进行多路复用

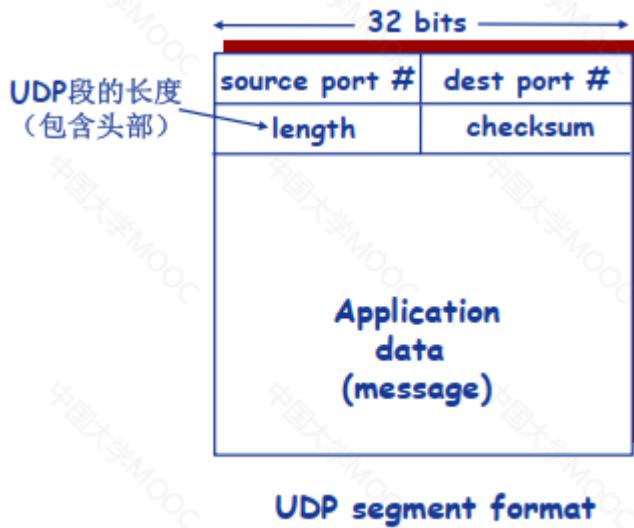
- 复用
  - 从应用层接受数据，为每个数据分装头部信息，交给网络层
- 分用
  - 从网络层接收数据，根据头部信息将数据交付正确的 **Scoket**
  - 无连接分用-UDP
    - 用二元组（目的 IP 地址，目的端口号）标识 **Scoket**
    - 不同来源的数据被导向同一个 **Scoket**

- 面向连接的分用-TCP
  - 用四元组（源 IP 地址，源端口号，目的 IP 地址，目的端口号）标识 Scoket
  - 为每一个不同来源的数据创建 Scoket
  - 多线程 Web 服务器

## 2.3 无连接传输协议-UDP

### UDP

- 基于 IP 协议
  - 复用/分用
  - 简单的错误校验
- 尽力而为服务
  - 丢失分组
  - 不是按序到达
- 无连接
- 存在原因
  - 不用建立连接（减少延迟）
  - 实现简单：无需维护连接状态
  - 头部开销少
  - 没用拥塞控制：应用可以更好的控制发送时间和速率
- 应用
  - 流媒体
  - DNS
  - SNMP
- 在 UDP 上实现可靠数据传输
  - 在应用层增加可靠机制
  - 应用特定的错误恢复机制
- 报文段格式



- UDP 校验和
  - 计算校验和时需要在 UDP 数据报前加上 12B 的伪首部
  - 高位进位需要回卷
  - 校验和全 0 代表不适用校验和，如果计算出来恰好全 0 则填充全 1

## 2.4 可靠数据传输基本原理

注：王道把这一部分和后面滑动窗口协议放在数据链路层。

可靠：不错，不丢，不乱

### 2.4.1 Rdt1.0

底层信道完全可靠

- 不会发生错误
- 不会丢弃分组



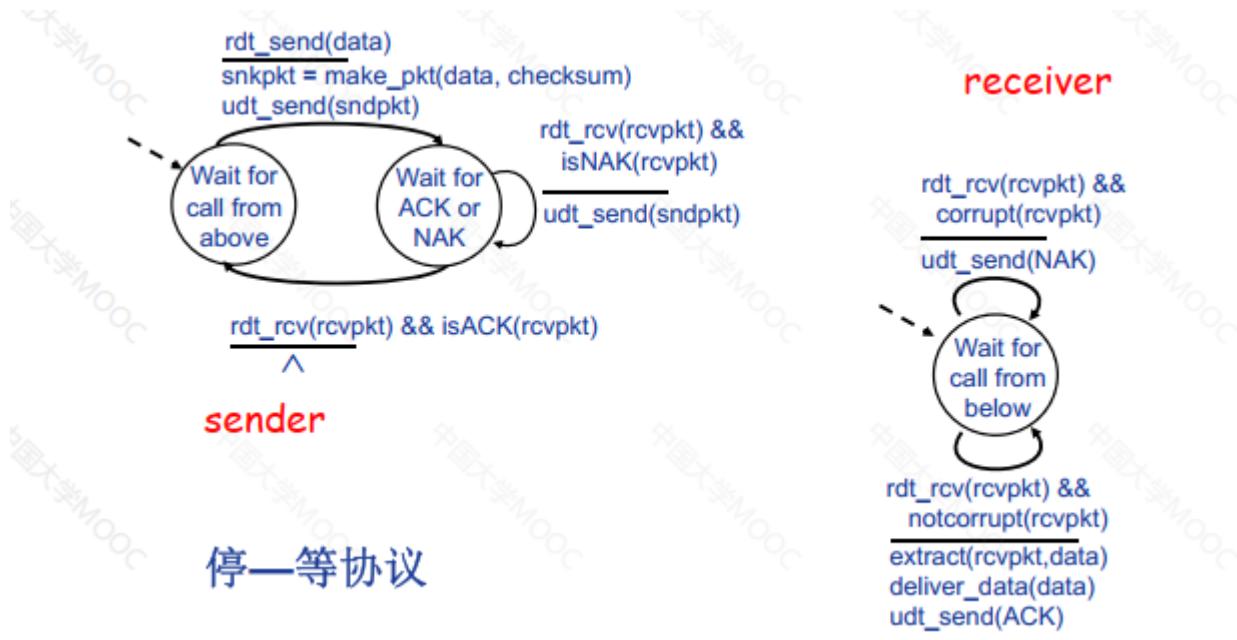
### 2.4.2 Rdt2.0 (ARQ)

底层信道可能出现位错误

- 利用校验和检测

错误恢复机制

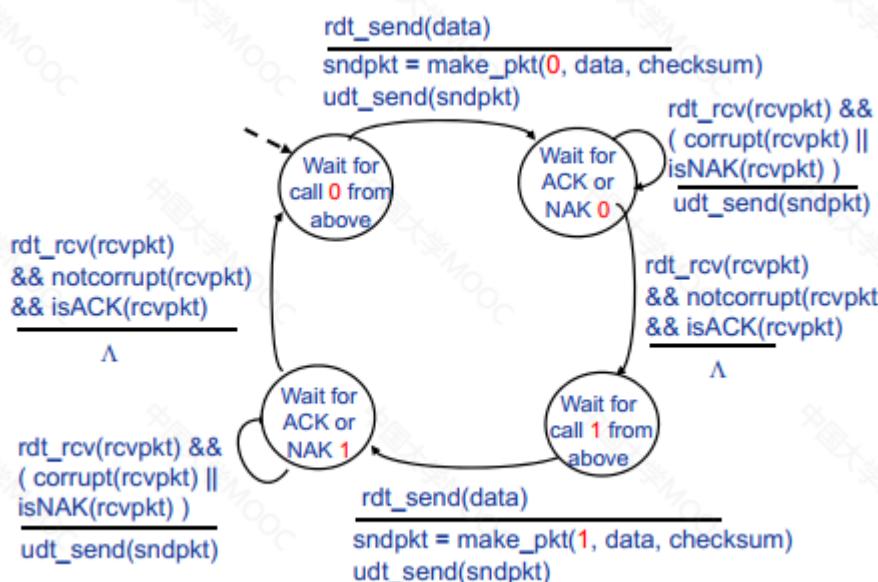
- 确认机制：接收方显式地告诉发送方分组已被正确接收
- NAK：接收方显式地告诉发送方有错误
- 发送方收到 NAK 后，重传分组



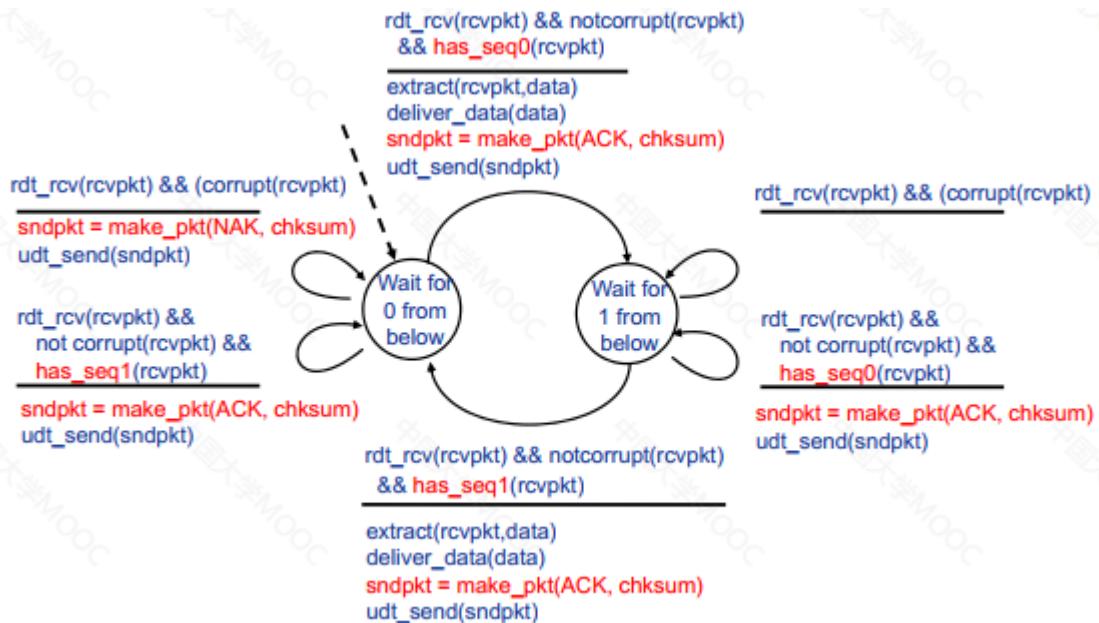
### 2.4.3 Rdt2.1

增加序列号，应对 ACK/NAK 消息被破坏导致的重复分组

发送方

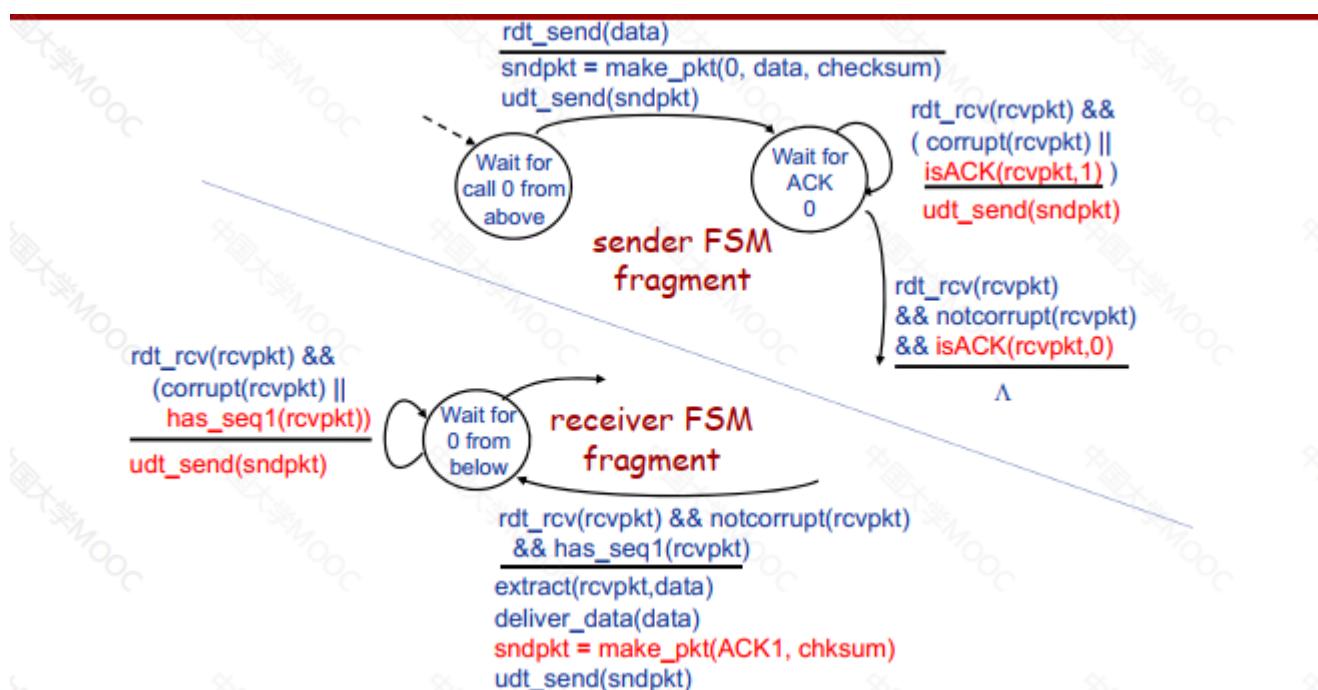


接收方：接收到错误的序列号时也要发送 ACK，但不向上层交付



## 2.4.4 Rdt2.2

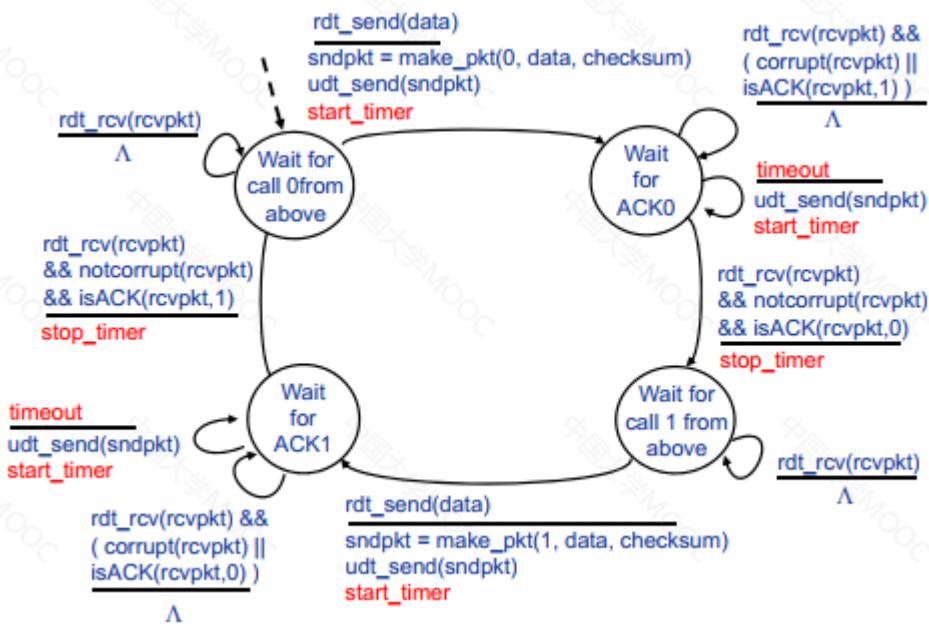
不使用 NAK 的 Rdt2.1



## 2.4.5 Rdt3.0

### 添加定时器

发送方



接收方，和 Rdt2.2 一样

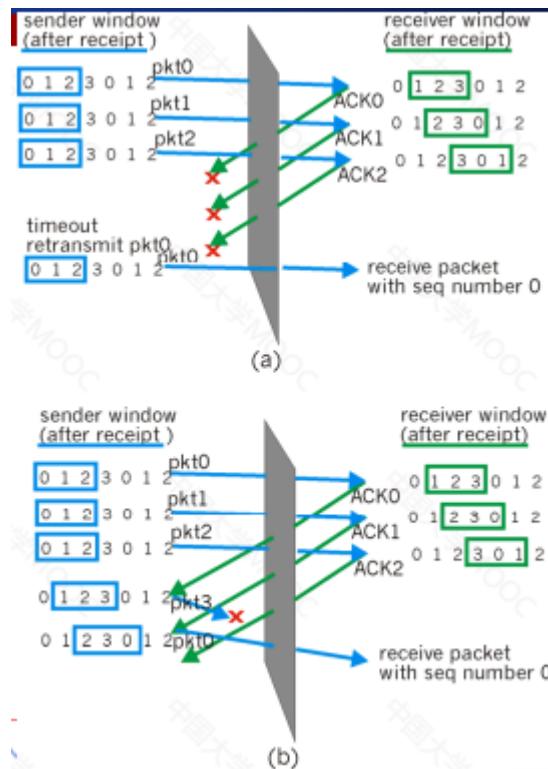
性能很差

## 2.5 滑动窗口协议

滑动窗口协议

- 窗口
  - 允许使用的序列号范围
  - 窗口尺寸为  $N$ : 最多有  $N$  个等待确认的消息
- 滑动窗口
  - 随着协议的运行，窗口在序列号空间内向前滑动
- 滑动窗口协议
  - GBN
    - 分组头添加  $k$  位序列号
    - 窗口尺寸为  $N$ , 最多允许  $N$  个分组未确认
    - 采用累计确认机制
    - 为窗口设置计时器，超时重传所有未确认分组
    - 接收方只接受当前想要的分组，乱序的直接丢弃
  - SR

- 在 GBN 的基础上，为每个发送但未确认分组设置计时器
- 分组计时器超时，重传这一分组
- 接收方也维护一个窗口，在窗口内的乱序到达分组缓存，等待前序确认之后一并交付
- 发送方针对 ACK 也采用和接收方类似的缓存机制
- 窗口大小与序列号长度的关系



窗口长度必须小于等于序列号长度的一半（这里假设发送方和接收方窗口大小一样，如果不一样就是相加小于序列号长度），否则会出现上图的困境

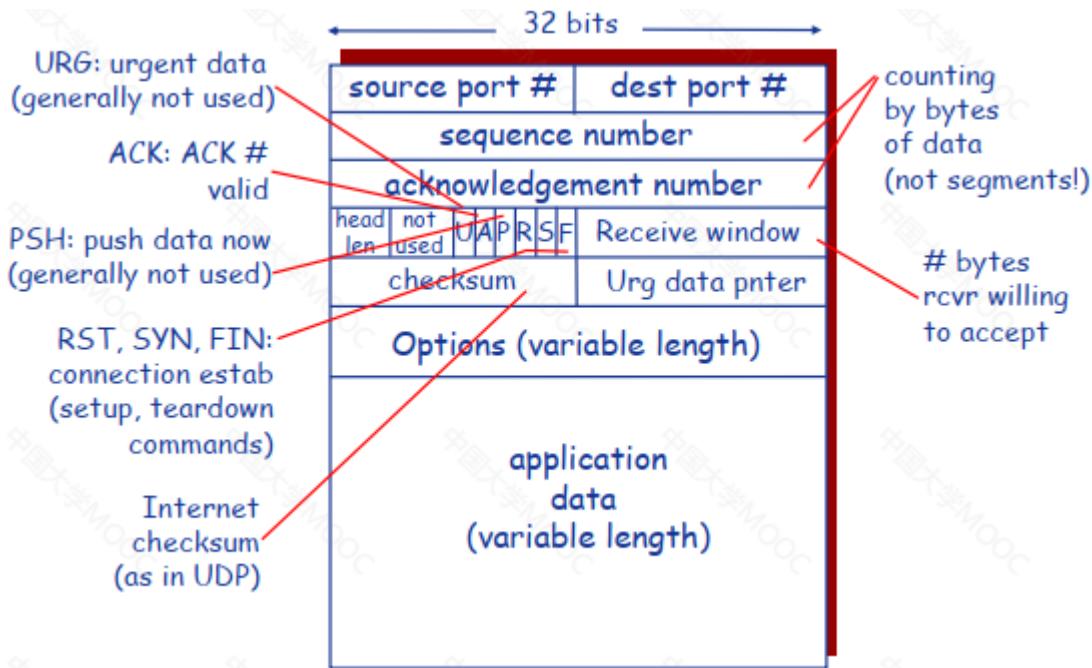
## 2.6 面向连接传输协议-TCP

### 2.6.1 TCP 协议

- 点对点协议
  - 一个发送方，一个接收方
- 可靠的按序的字节流
- 流水线机制
  - TCP 的拥塞控制和流量控制机制设置窗口尺寸（可动态调整）
- 发送方/接收方缓存，类似 SR

- 全双工
  - 同意连接能双向传递数据流
- 面向连接
  - 通信双方在发送数据之前必须建立连接
  - 连接状态旨在两端维护，中间节点不维护状态
  - TCP 连接包括：两台主机上的缓存、连接状态变量、socket 等
- 流量控制机制

## 2.6.2 TCP 段结构



- 序列号
  - 指的是 segment 中第一个字节的编号，不是 segment 的编号
  - 建立连接时，双方随机选择初始序列号
- ACKs
  - 希望收到的下一个字节的序号
  - 采用累计确认机制
- 对于乱序到达的 segment 没有做出规定

## 2.6.3 TCP 的可靠数据传输

- 流水线机制
- 累计确认
- 使用单一重传计时器
- 触发重传的事件
  - 超时
  - 重复 ACK

- 如何设置超时时间
  - 大于 RTT
  - 过短：不必要的重传
  - 过长：对丢失反应时间慢
- 如何估计 RTT
  - EstimatedRTT 估计：采样测量 SampleRTT，指数加权移动平均  

$$EstimatedRTT = (1 - \alpha) \times EstimatedRTT + \alpha \times SampleRTT$$
- 定时器超时时间的设置
  - EstimatedRTT+安全边界：EstimatedRTT 变化大意味着要更大的边界  

$$DevRTT = (1 - \beta) \times DevRTT + \beta \times |SampleRTT - EstimatedRTT|$$

$$TimeoutInterval = EstimatedRTT + 4 \times DevRTT$$
- 一般来说， $\alpha = 0.125, \beta = 0.25$
- 快速重传机制
  - 收到重复三次及以上的 ACK，假定其后的段已经丢失，进行重传
  - 在定时器超时前重传

#### 2.6.4 TCP 的流量控制

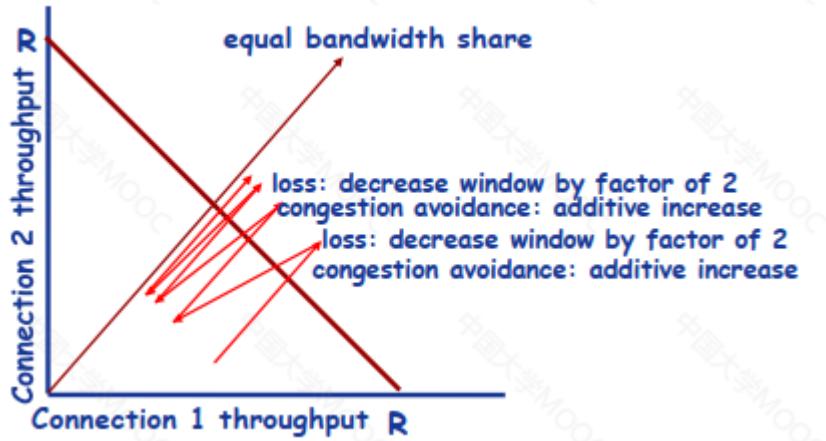
- 接收方为 TCP 连接分配 buffer，上层应用处理 buffer 数据速度可能较慢
- 流量控制旨在控制发送方发送速率使得发送方不会传输太多数据淹没接收方
- 速度匹配机制
  - 接收方在 TCP 段头部将自己 buffer 剩余空间大小告知发送方
  - 需要注意：接收方告知接收窗口的大小是在接收方当前已经接受的数据的基础上得出的，并不意味着发送方接受到一个窗口大小就可以再发送这一大小的数据。**24 王道 P231 28 题**

#### 2.6.5 TCP 性能分析

忽略慢启动

- ❖ 假定发生超时时 CongWin 的大小为 W，吞吐率是  $W/RTT$
- ❖ 超时后， $CongWin=W/2$ ，吞吐率是  $W/2RTT$
- ❖ 平均吞吐率为：  $0.75W/RTT$

公平性

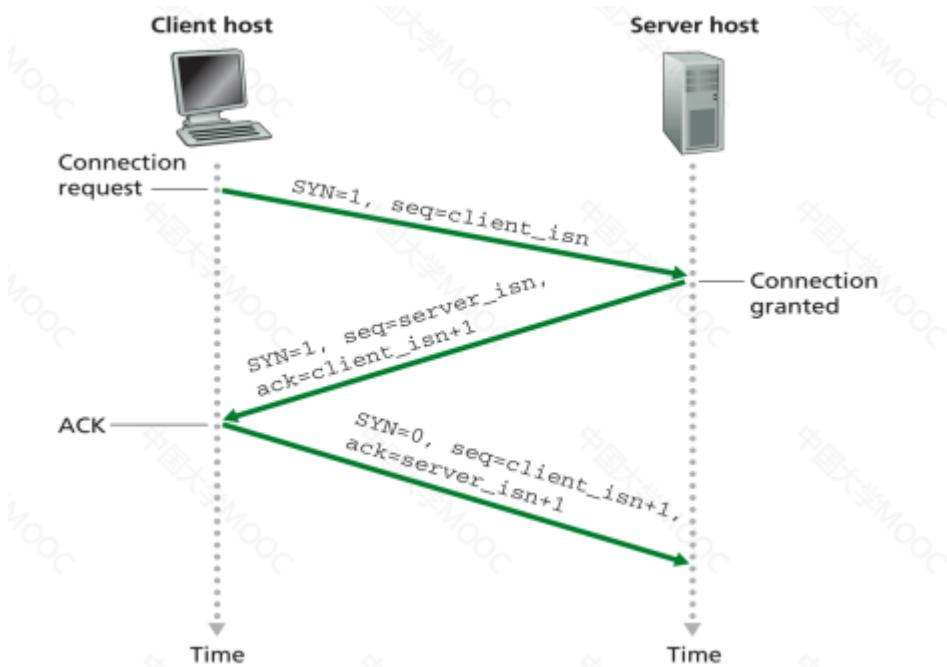


公平性是针对 TCP 连接来说的，如果一个应用同时建立多个 TCP 连接，可能会导致应用层面的不公平

## 2.6.5 TCP 连接管理

三次握手、四次挥手。重点注意序列号和 ack 的变化

建立连接



关闭连接

### Closing a connection:

client closes socket: `clientSocket.close()`;

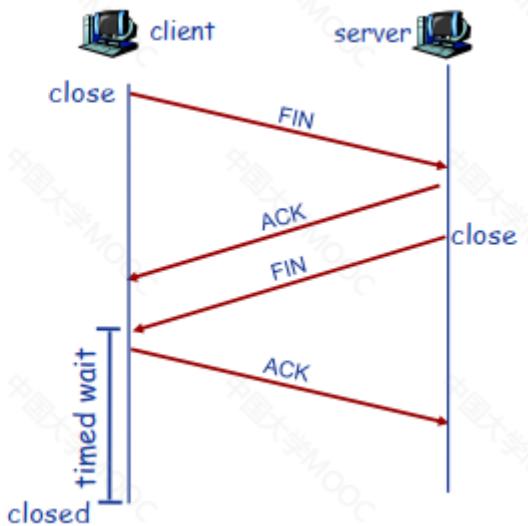
**Step 1:** client向server发送TCP FIN 控制segment

**Step 2:** server 收到FIN, 回复ACK. 关闭连接, 发送FIN.

**Step 3:** client 收到FIN, 回复ACK.

- 进入“等待” –如果收到FIN, 会重新发送ACK

**Step 4:** server收到ACK. 连接关闭.



## 2.7 拥塞控制原理

### 拥塞控制的方法

- 端到端拥塞控制
  - 网络层不需要显式的提供支持
  - 端系统通过观察 `loss`, `delay` 等网络行为判断是否发生拥塞
- 网络辅助的拥塞控制
  - 路由器显式地反馈网络拥塞信息

## 2.8 TCP 拥塞控制

### 拥塞控制算法

- 慢启动
- 拥塞避免
- 快重传
- 快恢复

超时采用慢启动和拥塞避免

- 拥塞窗口未达到阈值时, 指数增加
- 达到阈值后, 线性增加
- 超时, 新的阈值为超时时拥塞窗口大小的一半, 拥塞窗口变为 1
- 重复

冗余 ACK 采用快重传和快恢复

- 前面一样
- 检测到冗余 ACK，快速重传，阈值减小为当前窗口一半，拥塞窗口变为新阈值
- 线性增加

实际发送窗口大小是接收窗口和拥塞窗口的最小值

## 2.9 补充内容

### 2.9.1 糊涂窗口综合征

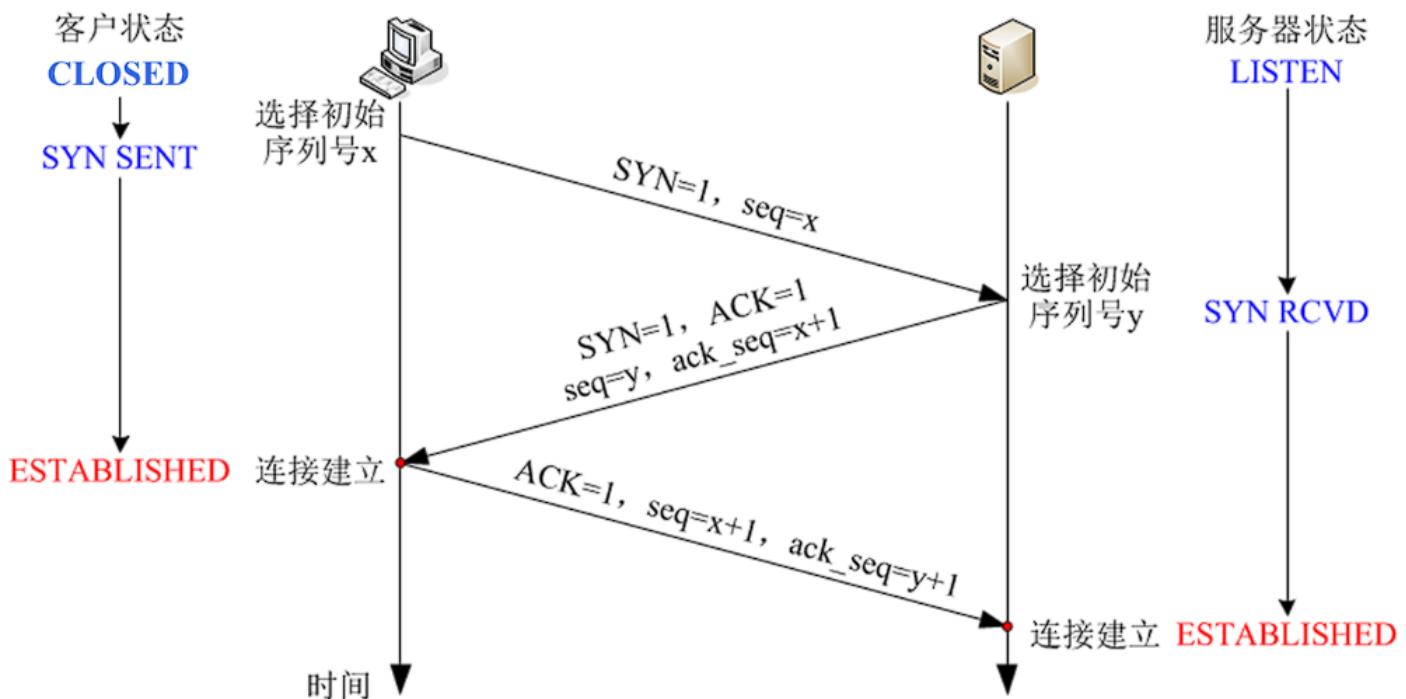
发送端发送的数据携带的实际数据很少，大部分都是头部

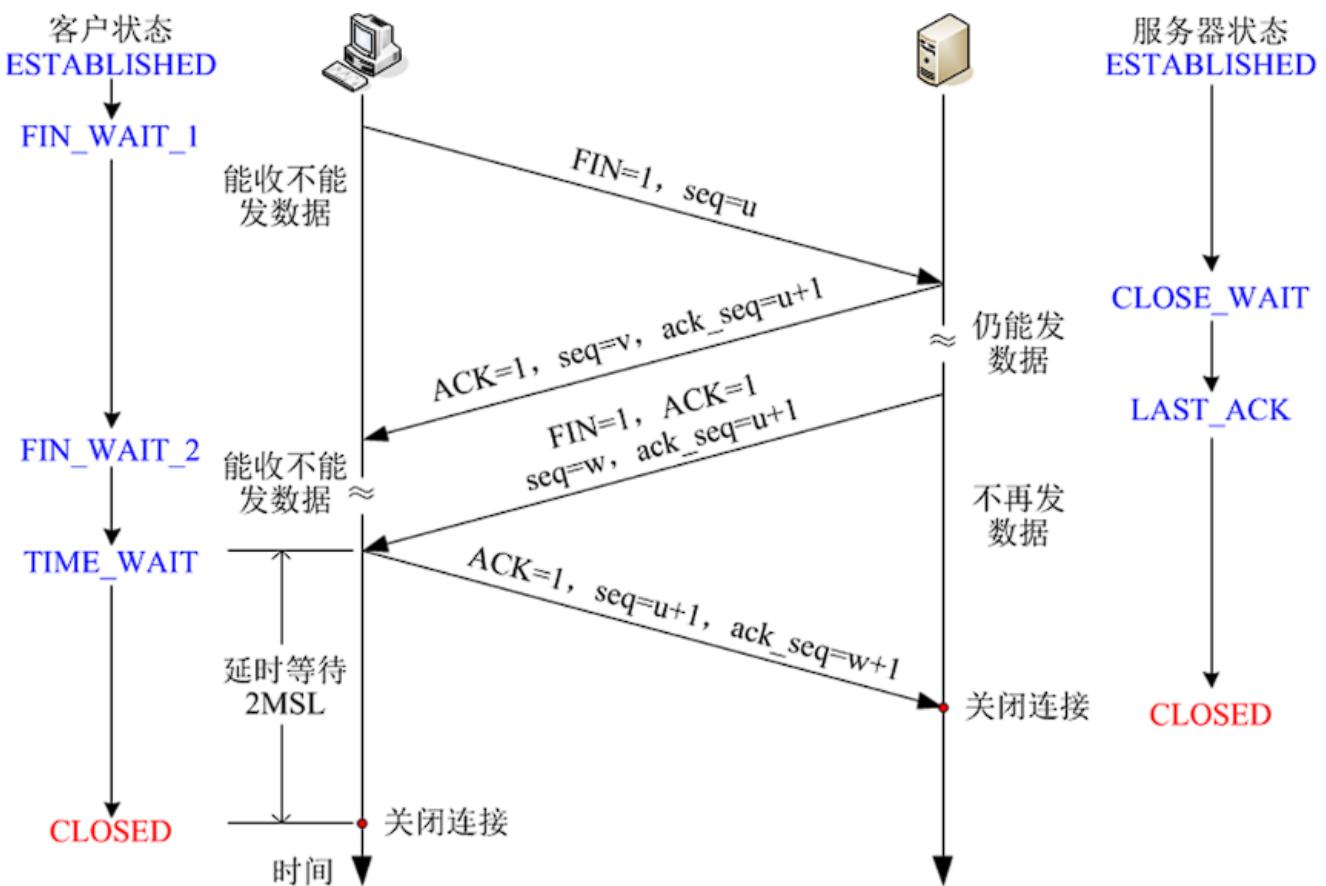
比如，接收端读取窗口中的数据很慢，接收窗口很快被填满，发送方不再发送数据。接收端应用层由于读数据很慢，每次只读一个字节，然后接收端通告大小为 1 的窗口，发送端发送一个字节，周而复始。这样虽然窗口很大，但是数据传输效率很低。

解决方法

- 接收端
  - Nagle 算法：报文一定长度再发送
- 发送端
  - Clark 方法：0 窗口确认
  - 延迟确认：阻止发送窗口滑动

### 2.9.2 TCP 连接控制补充





## 3 网络层

### 3.1 网络层服务

网络层

- 从发送主机向接受主机传送数据段
  - 发送主机：将传输层的数据段封装到数据报中
  - 接收主机：从数据报中拆出数据段交付传输层
- 每个主机和路由都运行网络层协议
- 路由器检查每个穿越它的IP数据报的头部域
- 核心功能
  - 转发和路由
    - 转发：将分组从路由器的输入端口转移到合适的输出端口
    - 路由：确定分组从源到目的的路径
    - 通过路由算法确定转发表，根据转发表确定转发端口
  - 连接建立

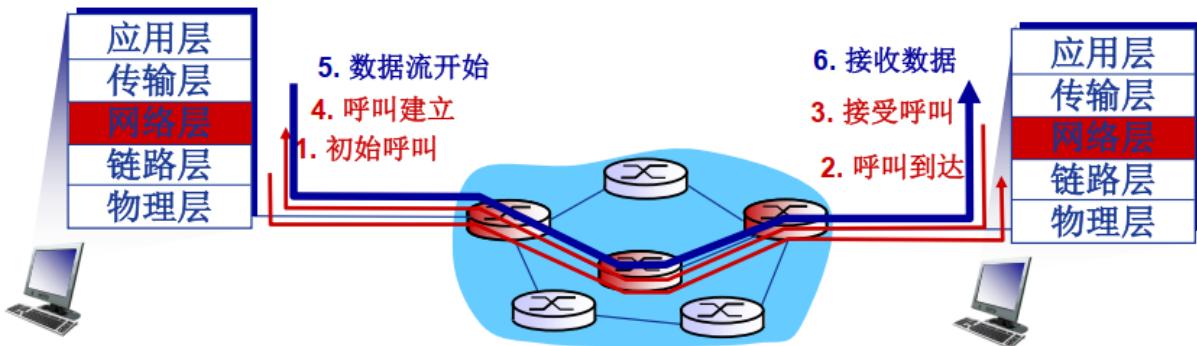
- 不是所有网络都需要连接建立
- 建立的连接是一条虚拟连接，并不是一条真正的通信链路而是使用路由器等设备中信息维护的一条路径
- 连接的作用：确定分组的传输路径
- 与传输层的不同
  - 路径每个设备都维护连接（只在端系统维护连接）
  - 是主机到主机之间的连接（应用进程之间的连接，对中间网络设备透明）

## 网络层服务模型

网络架构	服务模型	带宽	保证？			拥塞反馈
			丢失	保序	延迟	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR 恒定速率	constant rate	yes	yes	yes	no congestion
ATM	VBR 变化速率	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR 可用比特率	guaranteed minimum	no	yes	no	yes
ATM	UBR 不指名比特率	none	no	yes	no	no

- 无连接服务-数据报网络
  - 特点
    - 不事先未分组确定传输路径
    - 每个分组独立确定传输路径
    - 不同分组传输路径可能不同
    - 每个分组携带目的地址
    - 路由器根据目的地址转发分组
      - 基于路由算法构建转发表
      - 检索转发表
      - 每个分组独立选路
  - 在检索转发表时，采用最长前缀匹配原则
- 连接服务-虚电路网络
  - 特点
    - 首先为分组确定传输路径-建立连接
    - 然后沿该连接传输分组

- 每个分组传输路径相同
  - 传输结束后拆除连接
- 虚电路：一条从源主机到目的主机类似于电路的连接，逻辑连接
- 分组交换
  - 每个分组的传输利用链路的全部带宽
  - 源到目的路径经过的网络层设备共同完成虚电路功能
  - 通信过程：呼叫建立-数据传输-拆除呼叫
  - 每个分组携带虚电路标识（VCID），而不是目的主机地址
  - 虚电路经过的每个网络设备，维护每条经过它的虚电路连接状态
  - 链路、网络设备资源可以面向VC进行预分配
  - 每条虚电路包括
    - 从源主机到目的主机的一条路径
    - 虚电路号（VCID）
    - 路径上每个网络层设备，利用转发表记录经过的每条虚电路
    - 同一条虚电路在每段链路上的虚电路号通常不同，转发时通过转发表修改虚电路号
- 虚电路信令协议
- 用于虚电路的建立、维护与拆除-路径选择
  - 应用于虚电路网络



## Internet (数据报网络)

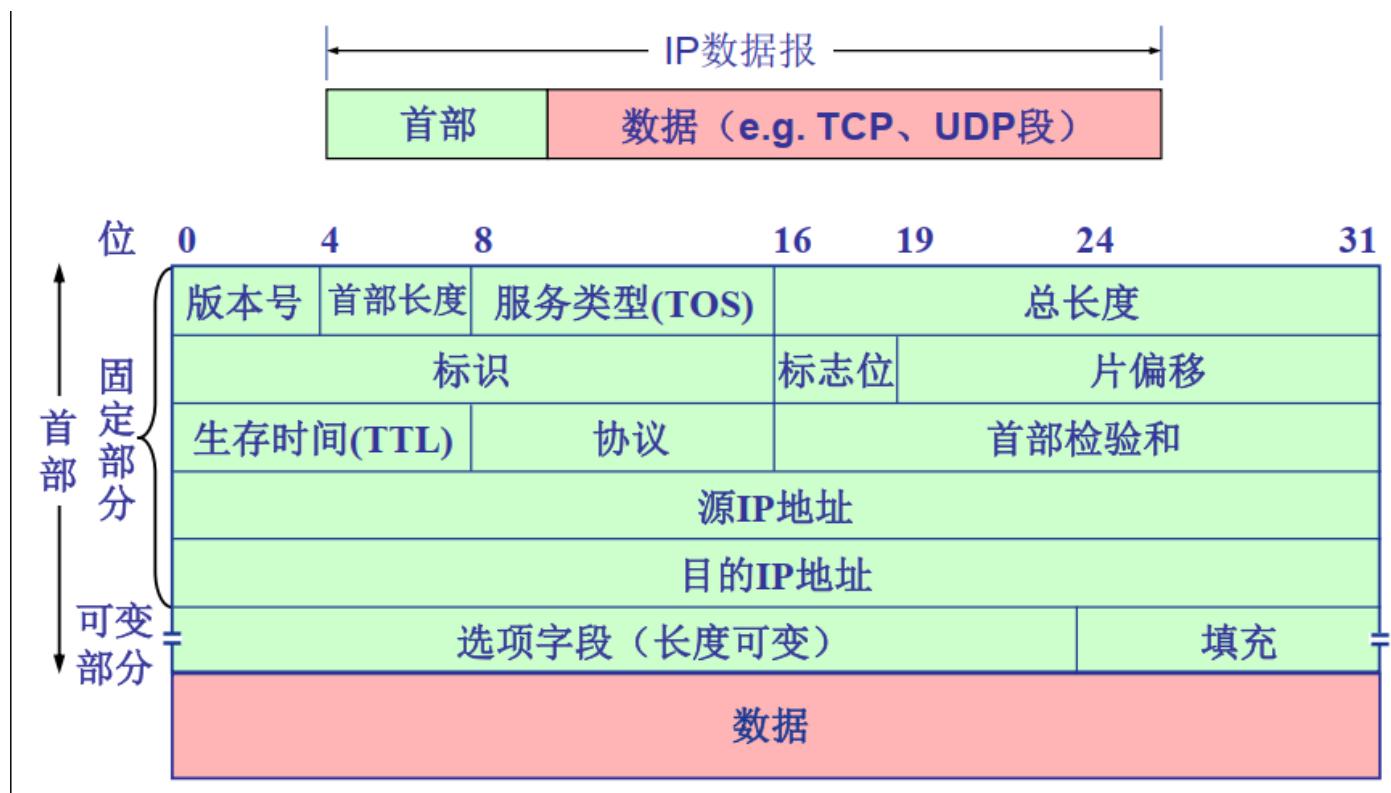
- ❖ 计算机之间的数据交换
  - “弹性”服务，没有严格时间需求
- ❖ 链路类型众多
  - 特点、性能各异
  - 统一服务困难
- ❖ “智能”端系统 (计算机)
  - 可以自适应、性能控制、差错恢复
- ❖ 简化网络，复杂“边缘”  
**复杂“边缘”**

## ATM (VC网络)

- ❖ 电话网络演化而来
- ❖ 核心业务是实时对话：
  - 严格的时间、可靠性需求
  - 需要有保障的服务
- ❖ “哑(dumb)” 端系统 (非智能)
  - 电话机
  - 传真机
- ❖ 简化“边缘”，  
**复杂网络**

## 3.2 IPv4协议

IP数据报格式



- 版本号：IP协议的版本号
- 首部长度：以四字节为单位

- 服务类型：指示期望获得那种类型的服务，一般情况下不使用
- 总长度：字节为单位，首部+数据
- 生存时间：在网络中最多的跳步数
- 协议：指示是那个协议的数据包，实现复用/分解，6为TCP，17为UDP
- 首部校验和：只对首部校验，逐跳计算、逐跳校验
- 源IP地址、目的IP地址
- 选项字段长度可变，1-40B：携带安全、源选路径、时间戳和路由记录等内容。很少被使用
- 填充：保证首部长度是4字节的倍数

## IP分片

- 当大IP分组向较小MTU链路转发时，可以被分片
- IP数据报中的标识：标识一个IP分组（不是分片）。IP协议利用一个计数器，每产生一个分组就加一
- 标志位
  - 保留位
  - DF
    - 1，禁止分片
    - 0，允许分片
  - MF
    - 1，非最后一片
    - 0，最后一片
- 片偏移：以8字节为单位
- 分片过程
  - 假设
    - 原IP分组总长度是L，待转发链路的MTU为M
  - 如果 $L > M$ 且DF=0，则可以分片
  - 分片时每个分片复制原分组的标识
  - 通常除了最后一片，其他分片均分为MTU允许的最大分片（注意首部占用20B）
  - 一个最大分片可封装的数据应该是8的倍数

## IP编址

- IPv4采用32位比特编号标识主机、路由器的接口
  - 接口：主机/路由器与物理链路的连接
    - 实现网络层功能
  - IP地址与每个接口关联

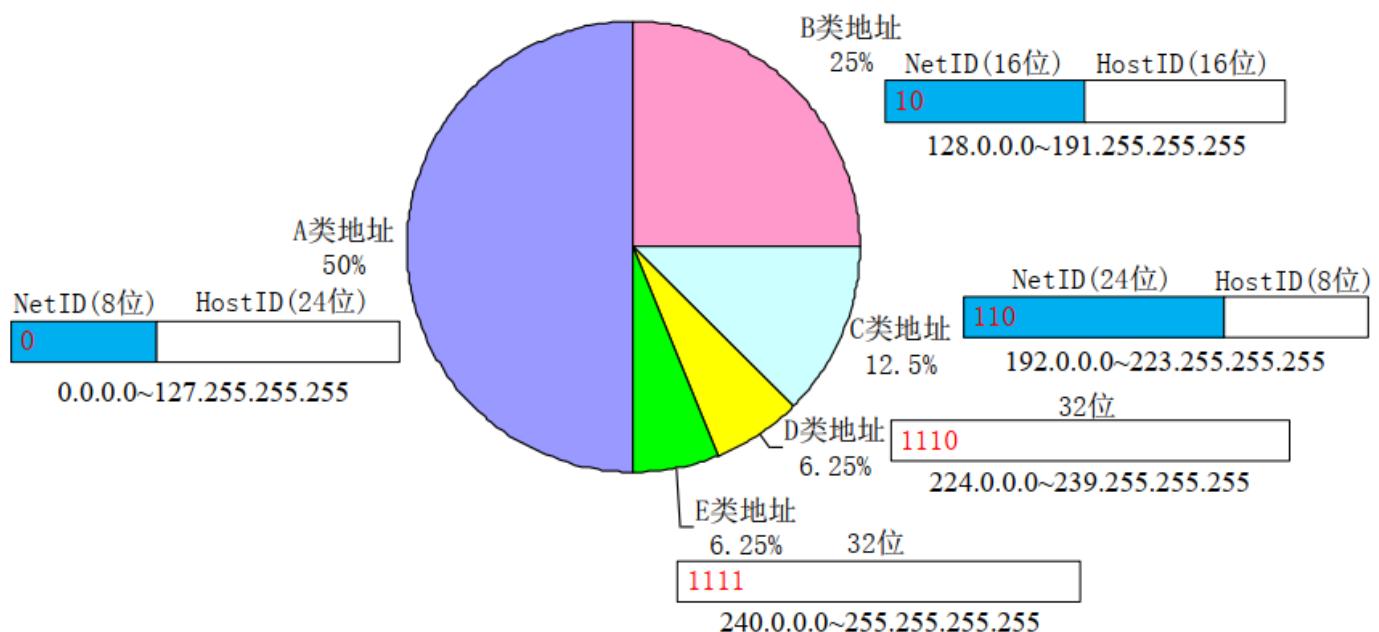
- 为了更好的分配IP地址-IP子网

- 将IP地址分为两部分
  - 网络号-高位比特
  - 主机号-低位比特

- IP子网

- 具有相同网络号设备接口
- 不跨越路由器可以彼此物理联通的接口
- 计算子网时注意路由器之间相互连接的部分也是一个子网

有类编址



特殊IP地址-计算可分配IP时注意剔除主机号全0和全1和IP地址

NetID	HostID	作为IP分组源地址	作为IP分组目的地址	用途
全0	全0	可以	不可以	在本网范围内表示本机；在路由表中用于表示默认路由(相当于表示整个Internet网络)
全0	特定值	不可以	可以	表示本网内某个特定主机
全1	全1	不可以	可以	本网广播地址(路由器不转发)
特定值	全0	不可以	不可以	网络地址，表示一个网络
特定值	全1	不可以	可以	直接广播地址，对特定网络上的所有主机进行广播
127	非全0或非全1的任何数	可以	可以	用于本地软件环回测试，称为环回地址

私有IP地址

Class	NetIDs	Blocks
A	10	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

子网划分

- 使用主机号的一部分标识一个网络中的子网
  - 网络号-高位比特
  - 子网号-原网络主机号部分比特
  - 主机号-低位比特
- 使用子网掩码确定一个子网的划分
  - 网络号和子网号全1，主机号全0
- 将目的IP地址与子网掩码按位与运算提取子网地址

## 3.3 CIDR与路由聚集

### CIDR-无类域间路由

- 消除传统有类编址地址界限
  - 网络号和子网号合在一起变成网络前缀
- 融合子网地址和子网掩码，在IP地址后标识网络前缀长度

### 路由聚集

- 如果在转发表中，经由统一接口转发的网络可用一个更大的网络描述，就可以将这些记录项用一条记录代替
- 采用最长前缀匹配原则

## 3.4 DHCP协议

### 一个主机如何获得IP地址

- 硬编码-静态配置
- 动态配置-DHCP 动态主机配置协议
  - 从服务器动态获取
    - IP地址
    - 子网掩码
    - 默认网关-第一次转发给哪个路由器
    - DNS服务器名称和IP地址
  - 即插即用
  - 允许地址重用
  - 支持在用地址续租
  - 支持移动用户加入网络
  - 实现过程
    - 主机广播DHCP discover 发现报文
    - DHCP服务器利用DHCP offer 提供报文进行响应
    - 主机请求IP地址DHCP request 请求报文

- DHCP服务器动态分配IP地址DHCP ack 确认报文
  - 包括上述提供的所有信息
- 在应用层实现
  - 请求报文封装到UDP数据报中
  - IP广播
  - 链路层广播

## 3.5 NAT协议

### NAT-网络地址转换

- 动机
  - 为了满足本地多台设备的需求，不用给每个设备都申请一个IP地址，只用申请一个IP地址
  - 在局域网范围内改变设备地址无需通告外界
  - 可以改变ISP地址，而不改变局域网内设备地址
  - 局域网内设备对外不可见
- 实现-路由器
  - 替换
    - 利用（NAT IP地址，新端口号）替换每个外出IP数据报的（源IP地址，源端口号）
    - 根据NAT转换表，利用（源IP地址，源端口号）替换每个进入内网IP数据报的（NAT IP地址，新端口号）
  - 记录
    - 将每对（NAT IP地址，新端口号）与（源IP地址，源端口号）的对应信息存储到NAT转换表中
- 主要争议
  - 路由器应该只处理第三层功能
  - 违背端到端通信原则
  - 地址短缺问题应该用IPv6解决
- NAT穿透问题-外网希望直接连接内网设备
  - 静态配置NAT
  - 利用UPnP互联网网关设备协议自动配置
    - 学习到NAT公共IP地址

- 在NAT转换表中，增删端口映射
- 中继：都连接到一个中继服务器，由中继服务器桥接

## 3.6 ICMP协议

ICMP-互联网控制报文协议

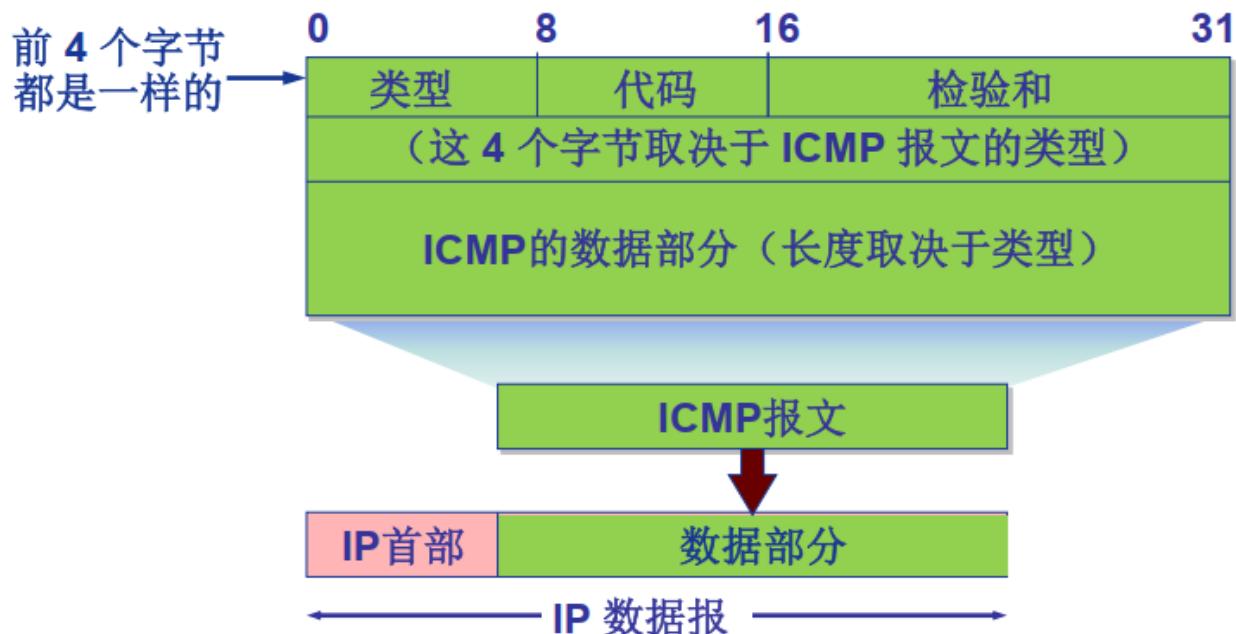
- 支持主机或路由器
  - 差错报告
    - 目的不可达
    - 源抑制：拥塞丢弃
    - 超时：TTL为0
    - 参数问题
    - 重定向
  - 网络探询报文
    - 回声请求与应答报文
    - 时间戳请求与应答报文

类型(Type)	编码(Code)	description
0	0	回声应答 (ping)
3	0	目的网络不可达
3	1	目的主机不可达
3	2	目的协议不可达
3	3	目的端口不可达
3	6	目的网络未知
3	7	目的主机未知
4	0	源抑制(拥塞控制-未用)
8	0	回声请求(ping)
9	0	路由通告
10	0	路由发现
11	0	TTL超期
12	0	IP首部错误

- 不发送ICMP差错报告报文的特殊情况

- 对ICMP差错报文不再发送差错报告报文
- 除第一个IP数据报分片外，对后续所有分片均不发送差错报文
- 对所有多播IP数据报不发送差错报文
- 对特殊地址（127.0.0.0或0.0.0.0）的数据报不发送差错报文

- 报文格式



- 应用

- PING: 测试两台主机之间连通性
- Traceroute: 追踪分组经过的路由, 发送一系列TTL递增的UDP数据报, 采用不可能使用的端口号, 当返回目的端口不可达时停止

## 3.7 IPv6

128位地址, 每16位用16进制表示, 每组之间用: 分割

动机

- IPv4地址空间使用完

- 改进首部格式

- 快速处理/转发数据报
- 支持QoS

- 数据报格式-固定长度的40B基本首部, 不允许分片

**优先级(priority):** 标识数据报的优先级

**流标签(flow Label):** 标识同一“流”中的数据报

**下一个首部(next header):** 标识下一个选项首部或上层协议首部(如TCP首部)



- 移除校验和
- 定义多个选项首部
- ICMPv6

- 附加报文格式
- 多播组管理功能

❖ 一般形式: 1080:0:FF:0:8:800:200C:417A

❖ 压缩形式: FF01:0:0:0:0:0:0:43  
压缩→FF01::43

❖ IPv4-嵌入形式: 0:0:0:0:FFFF:13.1.68.3  
或 ::FFFF:13.1.68.3

❖ 地址前缀: 2002:43c:476b::/48  
(注: IPv6不再使用掩码!)

❖ URLs: [http://\[3FFE::1:800:200C:417A\]:8000](http://[3FFE::1:800:200C:417A]:8000)

#### 基本地址类型

- 单播: 传统一对一通信
- 多播: 一对多通信
- 任意播: 一对一组之一通信

#### IPv4向IPv6过渡

- 隧道: 将IPv6数据报封装到IPv4数据报中, 穿越IPv4网络

## 3.8 路由算法

#### 分类

- 静态路由/动态路由
  - 静态路由
    - 手工配置
    - 更新慢
    - 优先级高

- 动态路由
  - 更新快
- 全局信息/分散信息
  - 全局信息-所有路由器掌握完整的网络拓扑和链路费用
    - 链路状态算法
  - 分散信息-只掌握物理相连的邻居以及链路费用
    - 距离向量算法

链路状态路由算法

Dijkstra算法

- 实现（略）
- 存在震荡的可能-费用设置为链路承载的通信量

距离向量路由算法

Bellman-Ford方程（动规）

令：

$d_x(y)$  := 从x到y最短路径的费用（距离）

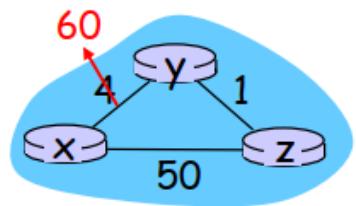
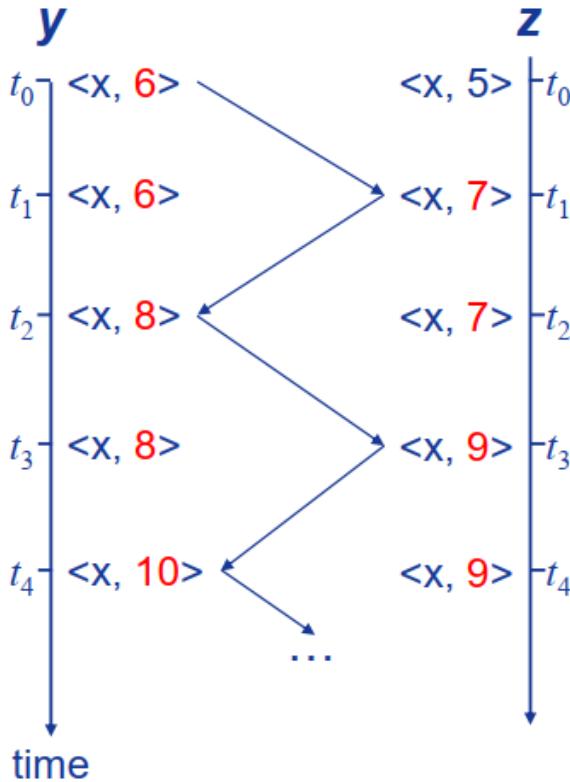
则：

$$d_x(y) = \min_v \{ c(x, v) + d_v(y) \}$$

↓                    ↓                    ↓  
 在x的所有邻居v中取最小值      x到邻居v的费用      从邻居v到达目的y的费用（距离）

- 只获得最短路径的下一跳
- 每个结点不定时将自身距离向量估计发送给其邻居
- 收到邻居发送的其自身距离向量估计时，利用BF方程更新自身向量估计
- 异步迭代
  - 引发迭代的因素

- 局部链路费用改变
- 来自邻居的距离向量更新
- 分布式
  - 每个结点只当距离向量发生变化时才通告其邻居
- 无穷计数问题



**坏消息传播慢！**  
— “无穷计数  
(count to infinity)  
问题！”

- 解决办法
  - 毒性逆转-当一个结点到目的的最小费用路径是通过某个邻居，则通告该邻居其到目的的距离为无穷大
  - 定义最大度量-定义最大跳步数，超过该跳步数代表不可达

## 3.9 Internet路由

### 层次路由

- 聚合路由器为一个区域：自治系统**AS**
- 同一**AS**内路由器运行相同的路由协议
  - 自治系统内部路由协议
  - 不同**AS**内的路由可以运行不同的**AS**内部路由协议
- 网关路由器
  - 位于**AS**边缘

- 通过链路连接其他AS的网关路由器
- 转发表由AS内部路由算法和AS间路由算法共同配置
  - AS内部路由算法设置AS内部目的网络路由入口
  - AS内部路由算法和AS间路由算法共同设置AS外部目的网络路由入口
- AS间路由任务
  - 学习哪些网络可以通过与该AS相联的AS到达
  - 将这些网络可达性信息传播给AS内路由器
- 如果一个目的网络可以通过多个AS到达，选择哪个网关进行转发
  - 热土豆算法：发送给最近的网关路由器

## AS内部路由协议-IGP

- RIP-路由信息协议
  - 基于距离向量算法
    - 距离度量-跳步数，最大15跳
    - 每30秒邻居之间交换距离向量信息，成为通告
    - 每次通告，最多25个目的子网
  - 链路失效、恢复
    - 如果180秒没有收到邻居通告-认为邻居/链路失效
    - 重新计算路由-经过该邻居的路由不可用
    - 更新后重新向邻居通告
    - 无穷级数问题
  - RIP路由表利用乘坐Route-d的应用进程进行管理
  - 通告报文通过UDP数据报发送
- OSPF-开放最短路径优先
  - 采用链路状态路由算法
    - LS分组扩散
    - 每个路由器构造完整AS网络拓扑图
    - 利用Dijkstra算法计算路由
  - 通告中每个入口对应一个邻居
  - 通告在整个AS中泛洪-直接封装到IP数据报中
  - 优点-RIP不具备
    - 安全：所有ospf报文可以被认证
    - 允许使用多条相同费用的路径
    - 对于每条链路，可以针对不同的TOS设置多个不同的费用度量
    - 集成单播路由和多播路由
    - 支持对大规模AS分层
  - 分层OSPF

- 两级分层-局部区、主干区
  - 链路状态通告只限于区内
  - 每个路由详细掌握所在区网络拓扑
  - 只知道去往其他区网络的方向
- 区边界路由器：汇总到达所在区网络距离，通告其他区边界路由器
- 主干路由器：在主干区内运行OSPF算法
- AS边界路由器：连接其他AS
- IGRP-内部网关路由协议
  - 私有协议

## AS间路由协议-BGP协议

### BGP-边界网关协议

- 事实上的标准域间路由协议
- 为每个AS提供
  - eBGP：从邻居AS获取子网可达性信息
  - iBGP：向所有AS内部路由传播子网可达性信息
- BGP会话：两个BGP路由器交换BGP报文
  - 通告去往不同目的前缀的路径
  - 报文交换基于半永久的TCP连接
  - BGP报文
    - OPEN：与其他BGP路由建立TCP连接，并认证发送方
    - UPDATE：通告新路径或撤销原路径
    - KEEPLIVE：无UPDATE时，保活连接，也用于对OPEN的确认
    - NOTIFICATION：报告先前报文的差错，也用于关闭连接
  - 当一个BGP路由向另一路由通告前缀时
    - 承诺可以将数据报转发给该子网
    - 会聚合网络前缀
  - 当路由器获得新的前缀可达性信息时，即在转发表中增加该前缀的路由项
  - 通告的前缀信息包括BGP属性
    - 两个重要属性
      - AS-PATH (AS路径)：包含前缀通告经过的AS序列
      - NEXT-HOP (下一条)：开始一个AS-PATH的路由器接口，指向下一AS
- BGP路由选择
  - 网关路由器收到通告后，利用其输入策略决定接收/拒绝该路由
    - 基于策略的路由
  - 如果有多条路由，基于以下准则先择

- 本地偏好值属性：策略决策
- 最短AS-PATH
- 最近NEXT-Hop路由器：热土豆路由

## 4 数据链路层

---

### 4.1 数据链路层服务

- 结点：主机和路由器
- 链路：连接相邻结点的通信信道
  - 有线
  - 无线
  - 局域网
- 链路层数据分组：帧，封装网络层数据报
- 数据链路层负责通过一条链路从一个节点向另一个物理链路直接相连的相邻节点传送数据报

#### 数据链路层服务

- 组帧
  - 封装数据报构成数据帧，加首部和尾部
  - 帧同步
- 链路接入
  - 如果是共享介质，需要解决信道接入
  - 帧头部中的“MAC”地址，用于标识帧的源和目的
- 相邻节点之间的可靠交付
  - 在低误码率的有线链路上很少使用（如光纤，某些双绞线等）
  - 无线链路：误码率高，需要可靠交付
- 流量控制
  - 协调相邻的结点发送和接收
- 差错检测
  - 信号衰减和噪声会引起差错
  - 接收端检测到差错：通知发送端重传或直接丢弃
- 差错纠正
  - 接收端直接纠正比特差错
- 全双工和半双工通信控制
  - 全双工：链路两端结点同时双向传输

- 半双工：链路两端结点交替双向传输

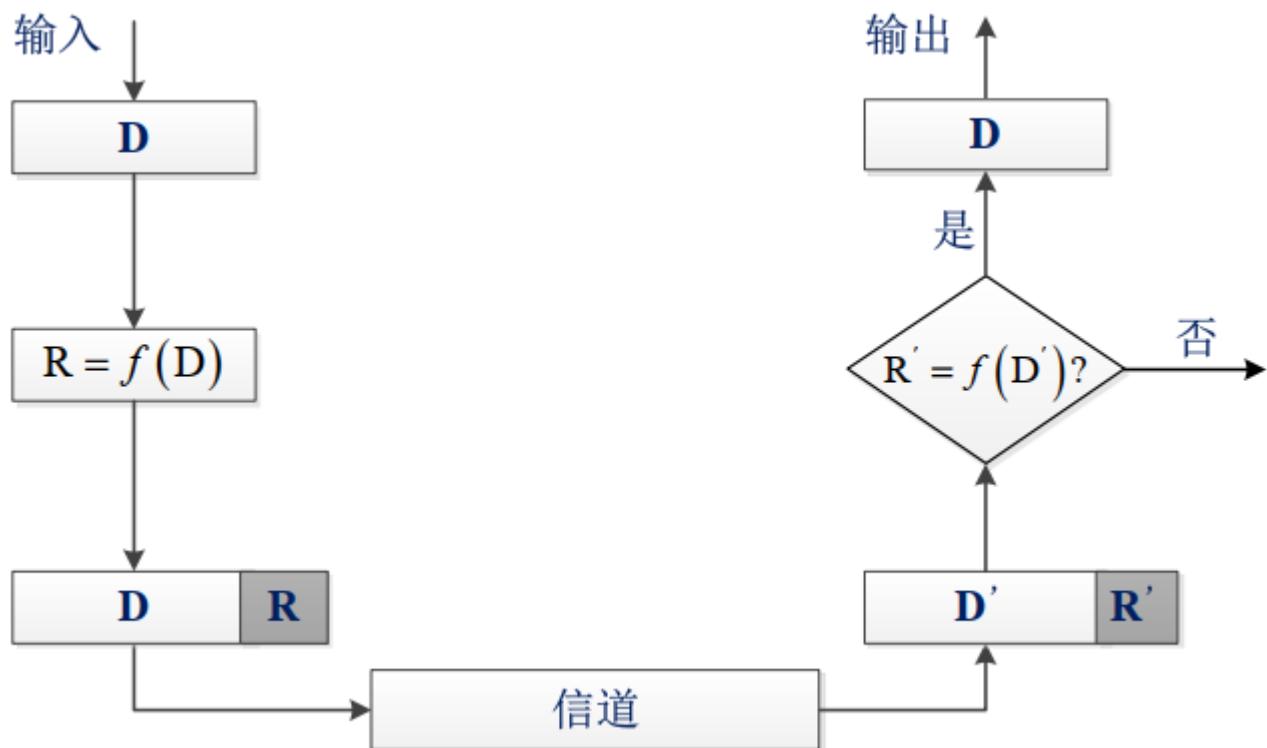
链路层的实现

- 由网卡实现
  - 链路层在适配器中实现
  - 同时实现链路层和物理层

## 4.2 差错编码

差错编码基本原理-不能保证完全可靠

- $D \rightarrow DR$ , 其中R为差错检测与纠正比特（冗余比特）



差错编码的检错能力

- 差错编码可以分为检错码和纠错码
- 对于检错码，如果编码集的汉明距离 $d_s = r + 1$ ，则可以检测r位错误
- 对于纠错码，如果编码集的汉明距离 $d_s = 2r + 1$ ，则可以纠正r位差错

奇偶校验码-填充后保证1的个数位奇/偶数

- 1比特校验位
  - 检测奇数位差错
- 二维奇偶校验
  - 检测奇数位、部分偶数位差错

- 纠正同一行/列的奇数位差错

Internet校验和-和UDP校验和类似

### 循环冗余校验码CRC

- 将数据比特 $D$ 视为一个二进制数
- 选择一个 $r + 1$ 位的比特模式， $G$
- 目标：选择 $r$ 位的CRC比特 $R$ 满足
  - $\langle D, R \rangle$ 刚好可以被 $G$ 整除
- 接收端检错
  - 用 $G$ 除 $\langle D, R \rangle$ ，整除则无差错
- 可以检测所有突发长度小于 $r + 1$ 位差错
- 计算过程
  - 生成 $R$ 
    - 利用 $G$ 除 $D \times 2^r$ （就是在 $D$ 后补 $r$ 个0），得到的余数就是 $R$
    - 注意：这里的除是模2除，即将传统除法的减法变成按位异或。~~什么年代了还在用传统除法~~

纠错编码：海明码，详见计组

## 4.3 多路访问协议

两类链路

- 点对点链路
  - 拨号接入的PPP
  - 以太网交换机与主机间的点对点链路
- 广播链路（共享介质）
  - 早期的总线以太网
  - HFC的上行链路
  - 802.11无线局域网

为什么需要MAC协议

- 两个或以上结点同时传输：干扰、冲突
- 结点同时接受到两个或多个信号：接受失败
- MAC协议采用分布式算法决定结点如何共享信道

MAC协议-多路访问协议

- 理想的MAC协议
  - 多个结点平分信道带宽、完全的分散控制、无需特定结点协调、无需时钟同步，简单
- 分类
  - 信道划分MAC协议-多路复用技术
    - TDMA-时分多路访问
    - FDMA-频分多路访问
    - WDMA-波分多路访问
    - CDMA-码分多路访问
  - 随机访问MAC协议
    - 时隙ALOHA协议
      - 假定：所有帧大小相同、时间被划分为等长的时隙、结点只能在时隙开始时刻发送帧、结点之间时钟同步、如果2个及以上结点在同一时隙发送帧，结点即检测到冲突
      - 运行
        - 结点有新的帧要发送时，在下一个时隙发送
          - 无冲突：可以在下一个时隙继续发送新的帧
          - 冲突：在下一帧以概率P重传帧，直到成功
        - 优点
          - 单个节点活动时，可以连续以信道全部带宽传输数据
          - 高度分散化：只用同步时隙
          - 简单
        - 缺点
          - 冲突浪费时隙
          - 空闲时隙
          - 结点也许能以远小于分组传输时间检测到冲突
          - 时钟同步
        - 效率：长期运行时，成功发送帧的时隙占比

**效率(efficiency):** 长期运行时，成功发送帧的时隙所占比例 (很多结点，有很多帧待发送)

- ❖ 假设: N个结点有很多帧待传输，每个结点在每个时隙均以概率p发送数据
- ❖ 对于给定的一个结点，在一个时隙将帧发送成功的概率 =  $p(1-p)^{N-1}$
- ❖ 对于任意结点成功发送帧的概率 =  $Np(1-p)^{N-1}$

- ❖ 最大效率: 求得使 $Np(1-p)^{N-1}$ 最大的 $p^*$
- ❖ 对于很多结点，求 $Np^*(1-p^*)^{N-1}$ 当N趋近无穷时的极限，可得:

$$\text{最大效率} = 1/e = 0.37$$

**最好情况:** 信道被成功利用的时间仅占37%!



- ALOHA协议-很简单，无需同步时钟，有帧立即发送
  - 效率

$$P(\text{给定结点成功发送帧}) = P(\text{该结点发送}) \cdot$$

$$P(\text{无其他结点在}[t_0-1, t_0]\text{期间发送帧}) \cdot$$

$$P(\text{无其他结点在}[t_0, t_0+1]\text{期间发送帧})$$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

... 选取最优的p，并令 $n \rightarrow \infty$

$$= 1/(2e) = 0.18$$

**比时隙ALOHA协议更差!**

- CSMA协议-载波监听多路访问协议
  - 发送之前监听信道
    - 空闲: 发送
    - 忙: 推迟发送
      - 1-坚持: 持续监听直到空闲
      - 非坚持: 放弃监听、等待随机时间重新尝试
      - P-坚持: 空闲时以概率P发送，1-P推迟下一时隙发送。
      - 忙, 持续监听。
    - 仍然可能发生冲突: 信号传播有延迟
  - CSMA/CD协议-带碰撞检测的载波监听多路访问协议 (只能半双工通信)

- 短时间内可以检测到冲突，冲突后传输终止，减少信道浪费
- 冲突检测-先听后发、边发边听、冲突停发、随机重发
  - 有线局域网易于实现，比较发送信号与接收信号强度
  - 无线局域网难以实现

网络带宽:  $R$  bps

$$L/R \geq 2d_{max}/V$$

数据帧最小长度:  $L_{min}$  (bits)

信号传播速度:  $V$  (m/s)

$$L_{min}/R = 2d_{max}/V$$

$$L_{min}/R = RTT_{max}$$

- 随机重发策略-二进制退避算法
  - 当结点传输帧后，经过了n次碰撞，结点随机从  $\{0, 1, 2, 3, \dots, 2^{n-1}\}$  中选择一个k值，用于随机选择等待时间间隔，基本时间为512比特时间，n最多到10
  - 王道：重传超过16次认为网络太拥挤，该帧永远无法正确发出，丢弃冰箱高层报告错误
  - 效率

❖  $T_{prop}$  = LAN中2个结点间的最大传播延迟

❖  $t_{trans}$  = 最长帧传输延迟

$$\text{效率} = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- CSMA/CA-见物理层
  - 轮转访问MAC协议

## 信道划分MAC协议:

- 网络负载重时，共享信道效率高，且公平
- 网络负载轻时，共享信道效率低！

## 随机访问MAC协议:

- 网络负载轻时，共享信道效率高，单个结点可以利用信道的全部带宽
- 网络负载重时，产生冲突开销

## 轮转访问MAC协议:

综合两者的优点！

- 轮询：主结点轮流邀请从属结点发送数据
  - 问题：轮询开销、等待延迟、单点故障
- 令牌传递：控制令牌（特殊帧）依次在结点中传递
  - 问题：令牌开销、等待延迟、单点故障

## 4.4 ARP协议

### MAC地址

- 48位，固化在网卡上的ROM，有时可以用软件设置
- 每8位用16进制标识中间用'-'分割
- 局域网中每块网卡都有一个唯一的MAC地址

### ARP-地址解析协议

- 在同一个LAN内，如何在已知目的接口的IP地址前提下确定其MAC地址
  - ARP表：LAN中每个IP结点维护一个表
    - 存储某些LAN结点的IP/MAC地址映射关系：<IP地址； MAC地址； TTL>
  - 如果目的接口不在ARP表中
    - 广播ARP查询分组，其中包含目的接口的IP地址
    - 目的接口接收查询分组，IP匹配，向源应答MAC地址（单播）
    - 收到应答，缓存目的的IP-MAC地址对，直到超时
  - 即插即用协议
- 不在同一局域网内
  - 发送时MAC地址是网关接收的MAC地址

- 之后每次传输时，修改源MAC地址和目的MAC地址（注意接受接口和发送接口的不同）
- IP地址一直不变

## 4.5 以太网

以太网

- 特点
  - 造价低廉
  - 应用最广泛的LAN技术
  - 简单、便宜
  - 满足网络速率需求：10Mbps-10Gbps
- 物理拓扑
  - 总线：所有结点在同一冲突域中
  - 星型：每个结点单独冲突域
- 服务类型
  - 无连接
  - 不可靠：不进行确认，差错帧直接丢弃
- 采用二进制指数退避算法的CSMA/CD协议
- 帧结构



- 前导码-8B：连续的10最后两位位11
  - 用于时钟同步
- 目的MAC地址，源MAC地址-6B
  - 目的MAC不匹配直接丢弃
- 类型-2B
  - 封装的是哪种高层协议的分组
- 数据-46-1500B

- CRC-4B
  - 循环冗余校验码
  - 丢弃差错帧

## 交换机

- 链路层设备
  - 存储-转发以太网帧
  - 校验到达帧的目的**MAC**地址，选择性向一个或多个输出链路转发
  - 利用**CSMA/CD**访问链路，发送帧
- 透明
  - 主机感知不到交换机的存在
- 即插即用
- 自学习
  - 收到帧，学习到发送帧的主机**MAC**地址
  - 将发送主机**MAC**地址/接口信息记录到交换表中
  - 转发时，如果在交换表中有匹配，直接通过对应接口转发；否则，在除接收到该帧的所有接口泛洪
- 多端口之间同时传输
  - 不同链路是不同的冲突域
  - 每段链路可以全双工

## 集线器

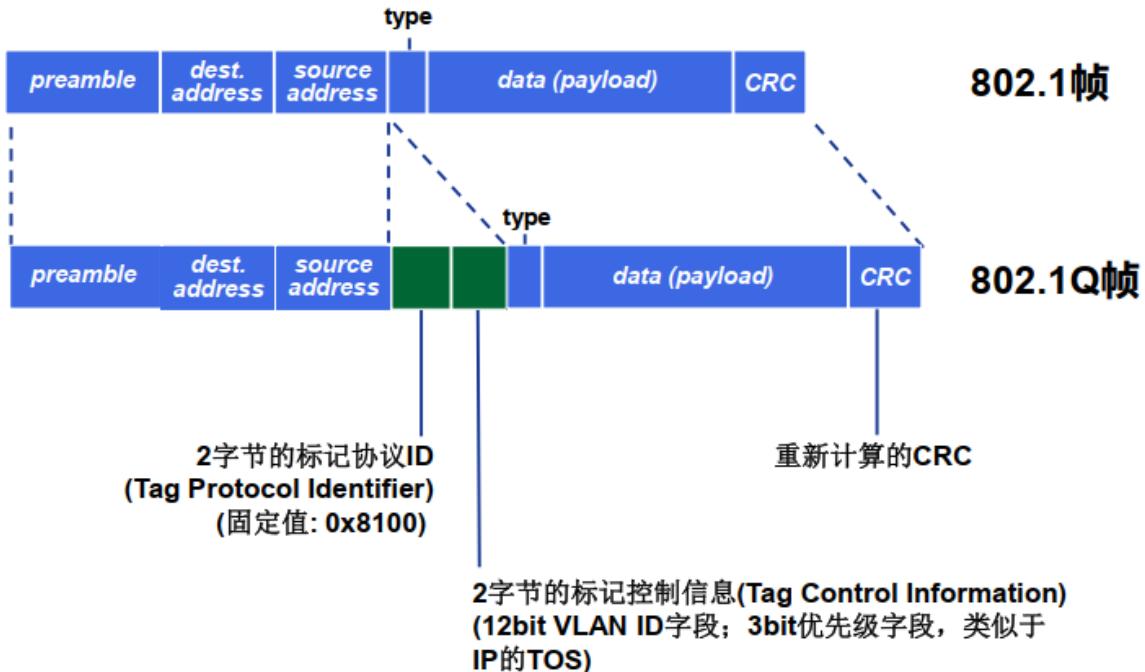
- 用于将多个设备连接到一个局域网中，以广播方式传输数据
- 无法划分冲突域

网桥：将多个以太网连接起来汇集为更大的以太网

	<u>集线器 (hub)</u>	<u>交换机 (switch)</u>	<u>网桥 (bridge)</u>	<u>路由器 (router)</u>
层次	1	2	2	3
流量(冲突域) 隔离	no	yes	yes	yes
广播域隔离	no	no	no	yes
即插即用	yes	yes	yes	no
优化路由	no	no	no	yes
直通传输 (Cut through)	yes	yes	yes	no

## VLAN-虚拟局域网

- 允许将物理网络划分为多个逻辑网络
- 基于端口的VLAN
  - 流量隔离
  - 动态成员
  - 在VLAN之间转发：通过路由器
- 跨越多交换机的VLAN
  - 多线缆连接
    - 每个线缆连接一个VLAN
  - 中继端口：在跨越多个物理交换机定义的VLAN承载帧
- 802.1Q VLAN
  - 帧格式



## 4.6 PPP协议

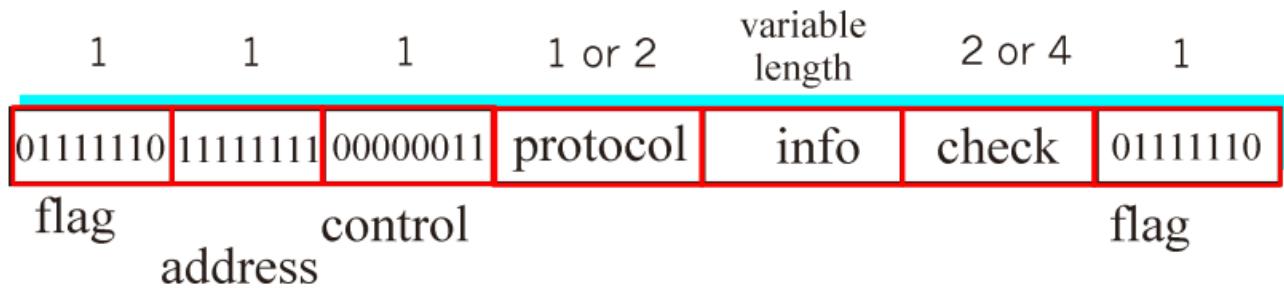
点对点数据链路控制

- 一个发送端，一个接收端，一条链路
- 常见的PPDLC协议
  - HDLC
  - PPP

PPP-点对点协议

- 设计需求
  - 组帧
    - 分装网络层数据报
    - 实现多路分用
  - 比特透明传输
  - 差错检测-不纠正
  - 连接活性-检测并通知网络层链路失效
  - 网络层地址协商-端结点可以学习/配置彼此网络地址
- 数据帧格式

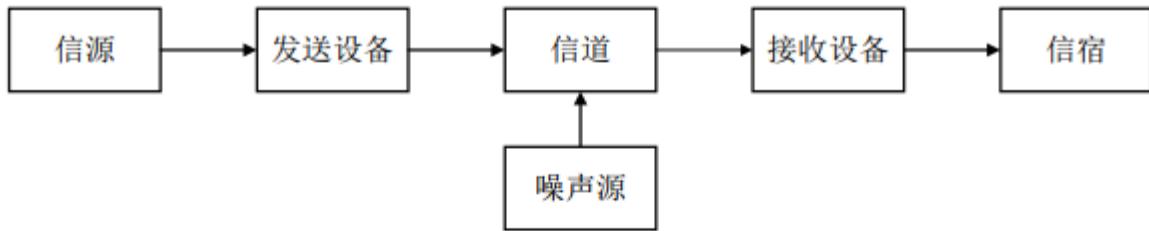
- ❖ 标志(Flag): 定界符(delimiter)
- ❖ 地址(Address): 无效(仅仅是一个选项)
- ❖ 控制(Control): 无效; 未来可能的多种控制域
- ❖ 协议(Protocol): 上层协议 (eg, PPP-LCP, IP, IPCP, etc)
- ❖ 信息(info): 上层协议分组数据
- ❖ 校验(check): CRC校验, 用于差错检测



- 字节填充-以01111110作为起始和结束标志, 如何判断是数据还是标志
  - 发送端
    - 在数据的01111110和01111101字节前添加额外字节01111101
  - 接收端
    - 单个01111101表示一个填充字节
    - 连续两个01111101丢弃第一个, 第二个作为数据接收
    - 单个01111110作为标志字节
  - 在交换网络层数据前, PPP数据链路两端必须
    - 配置PPP链路
      - 最大帧长
      - 身份认证
    - 学习/配置网络层信息
      - 对于IP协议: 交换IPCP协议报文, 完成IP地址等信息配置

## 5 物理层

### 5.1 数据通信基础

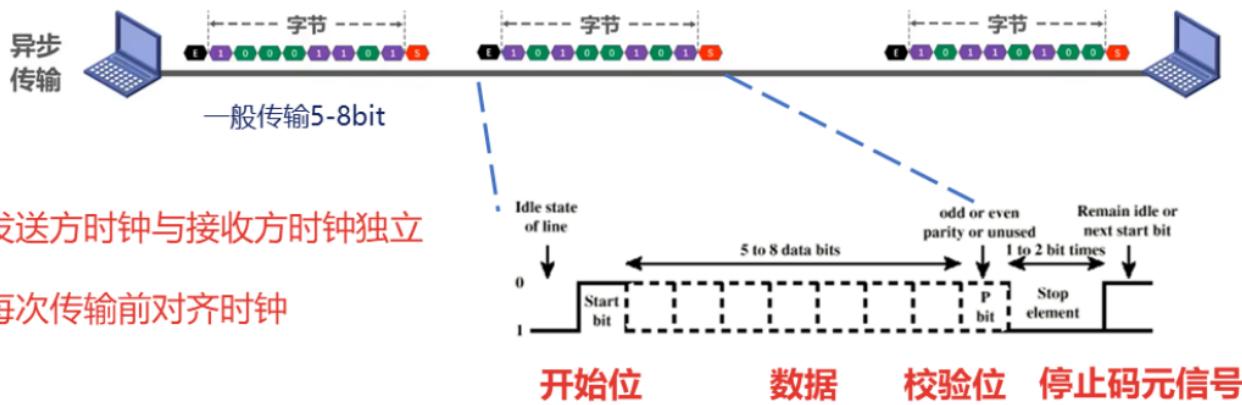


- **信源**: 将消息转换为信号的设备, 如计算机等。
- **发送设备**: 将信源产生的信号进行适当的变换装置, 使之适合于在信道中传输。主要包括编码和调制。
- **信道**: 信号传输通道, 如物理介质。
- **噪声**: 自然界和通信设备中所产生的干扰。
- **接收设备**: 完成发送设备反变换, 还原原始发送信号。
- **信宿**: 信号终点, 将信号转换为供人们能识别的消息。

## 通信术语

- **数据(data)**: 传送消息的实体。
- **信号(signal)**: 数据的电气的或电磁的表示:  $y(t) = A \sin(\omega t + \theta)$
- “模拟的” (analogous): 参数的取值是连续的
- “数字的” (digital): 参数的取值是离散的
- **码元(code)**: 信号基本波形 (信号基本单元)
- **频带(Spectrum)**: 信号频率范围
- **带宽(Bandwidth)**: 有效带宽
- **数据通信方式**: 单工、半双工、全双工
- **并行通信 vs 串行通信**

## 异步通信



同步通信



**每次传输一大块数据      时钟必须同步（使用相同的时钟信号）** 适用于短距离信号传输

另一种方式：将时钟信息编码进信号中，把同步时钟信号同时传输

信源编码

### ➤ 典型的信源编码：PCM

**PCM包括三个步骤：**采样 → 量化 → 编码

**采样：**目的就是要用一系列在时间上离散的采样值，代替时间上连续的模拟数据，即实现时间上的离散化。

**量化：**就是使采样值在取值上离散化

**编码：**就是将量化后的采样值用一定位数的二进制数码来表示。如果量化级数为N，则每个采样值就编码成 $\log_2 N$ 位二进制码

## 5.2 物理介质

导引型传输介质

- 架空明线
- 双绞线
  - 屏蔽双绞线STP
  - 非屏蔽双绞线UTP

UTP类别	带宽	典型应用
3	16 MHz	低速网络, 电话网络
4	20 MHz	10Base-T以太网
5	100 MHz	10Base-T以太网, 100Base-T快速以太网
5E (超5类)	100 MHz	100Base-T快速以太网, 1000-BaseT吉比特以太网
6	250 MHz	1000Base-T吉比特以太网, ATM网络

- 同轴电缆
- 光纤

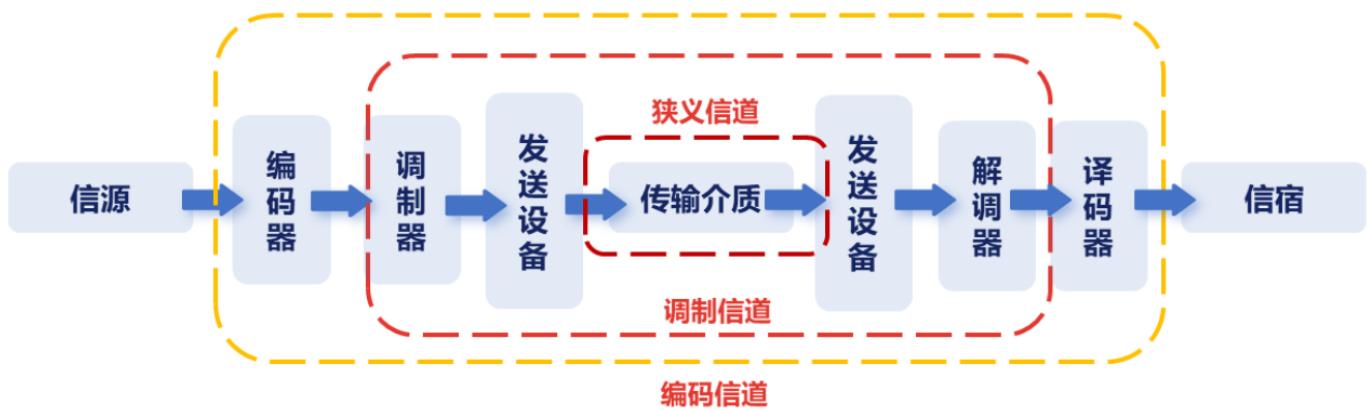
非导引型传输介质-自由空间中传输

频段	名称	典型应用
3~30 Hz	极低频 (ELF)	远程导航, 水下通信
30~300 Hz	超低频 (SLF)	水下通信
300~3000 Hz	特低频 (ULF)	远程导航
3~30 kHz	甚低频 (VLF)	远程导航, 水下通信, 声纳
30~300 kHz	低频 (LF)	导航, 水下通信, 无线电信标
300~3000 kHz	中频 (MF)	广播, 海事通信, 测向, 救险, 海岸警卫
3~30 MHz	高频 (HF)	远程广播, 电报, 电话, 传真, 搜救, 飞机与舰船通信
30~300 MHz	甚高频 (VHF)	电视, 调频广播, 陆地交通, 空中交通管制, 导航, 飞机通信
0.3~3 GHz	特高频 (UHF)	电视, 蜂窝网, 微波链路, 导航, 卫星通信, GPS, 监视雷达
3~30 GHz	超高频 (SHF)	卫星通信, 微波链路, 机载雷达, 气象雷达, 公用陆地移动通信
30~300 GHz	极高频 (EHF)	雷达着陆系统, 卫星通信, 移动通信, 铁路业务
300 GHz~3 THz	亚毫米波 (0.1~1mm)	尚为划分, 实验应用
43~430 THz	红外线 (7~0.7μm)	光通信系统
430~750 THz	可见光 (0.7~0.4μm)	光通信系统
750~3000 THz	紫外线 (0.4~0.1μm)	光通信系统

- 地波传播
- 天波传播
- 视线传播

## 5.3 信道与信道容量

信道



**广义信道：**包含信号传输介质和通信系统的变换装置，如发送装备、接收装备、天线、解调器

信道传输特性

## ➤ 恒参信道传输特性

- 各种有线信道和部分无线信道，如微波视线传播链路和卫星链路等，都属于恒参信道
- 理想的恒参信道是一个理想的无失真传输信道
- 对信号幅值产生**固定的衰减**
- 对信号输出产生**固定的时延**

## ➤ 随参信道传输特性

- 许多无线信道都是随参信道
- 信号的传输**衰减随时间随机变化**
- 信号的传输**时延随时间随机变化**
- 存在**多径传播现象**

信道容量：信道无差错传输信息的最大平均信息速率

- 奈奎斯特信道容量公式

## ➤ 奈奎斯特(Nyquist)信道容量公式

- 理想无噪声信道的信道容量:  $C=2Blog_2M$
- 其中:  $C$ 为信道容量, 单位为**b/s** (或**bps**) ;  $B$ 为信道带宽, 单位为**Hz**;  $M$ 为进制数, 即信号状态数
- 理想信道的极限容量

## ➤ Example:

- Q: 在无噪声情况下, 若某通信链路的带宽为3 kHz, 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多少?
- A: **24kbps**

- 香农信道容量公式

## ➤ 香农(Shannon)信道容量公式

- 有噪声信道的信道容量:  $C=Blog_2(1+S/N)$
- 其中:  $S/N$ 为信噪比, 即信号能量与噪声能量之比
- $S/N$ 通常以**分贝 (dB)** 为单位描述:
  - $(S/N)_{dB}=10\log_{10}(\text{Signal power}/\text{Noise power})$

## ➤ Example:

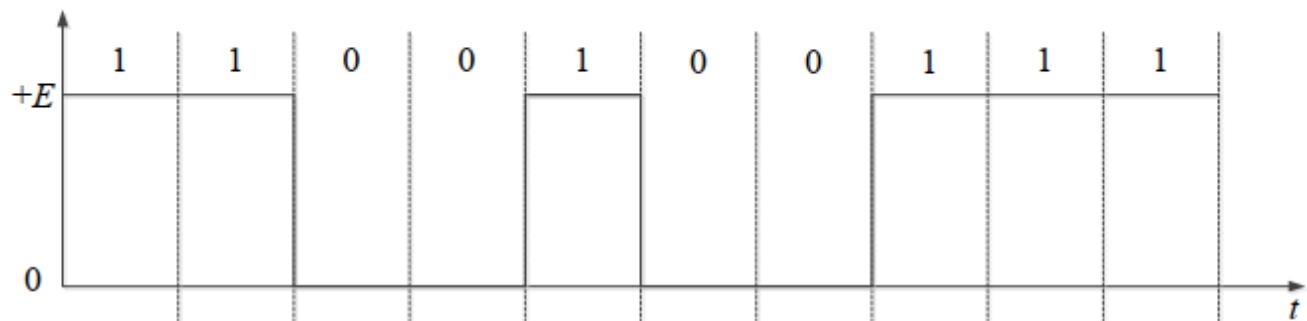
- Q: 若某通信链路的带宽为2 MHz, 信噪比为30 dB, 则该通信链路的最大数据传输速率约是多少?
- A: **20 Mbps**

## 5.4 基带传输基础

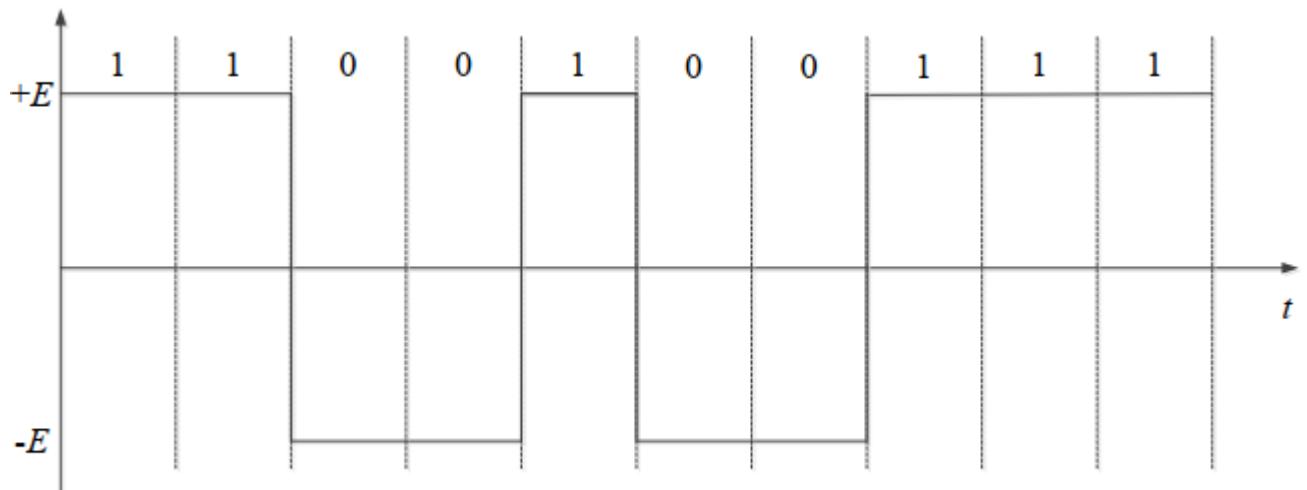
- 信源发出的原始电信号是**基带信号**
  - 模拟信源→模拟基带信号
  - 数字信源→数字基带信号
- 基带信号往往包含有较多的**低频成分**,甚至有**直流成分**
- 直接在信道中传送基带信号称为**基带传输**
- 实现基带传输的系统称为**基带传输系统**
- 在信道中直接传输数字基带信号,称为**数字基带传输**,相应的系统称为**数字基带传输系统**

信号码型

- 单极不归零码
  - 这种码型易于产生,但不适合长距离传输

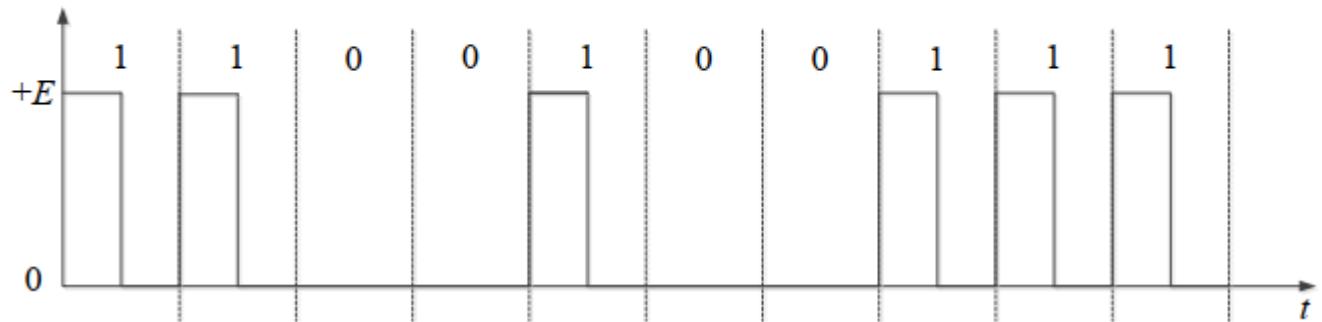


- 双极不归零码

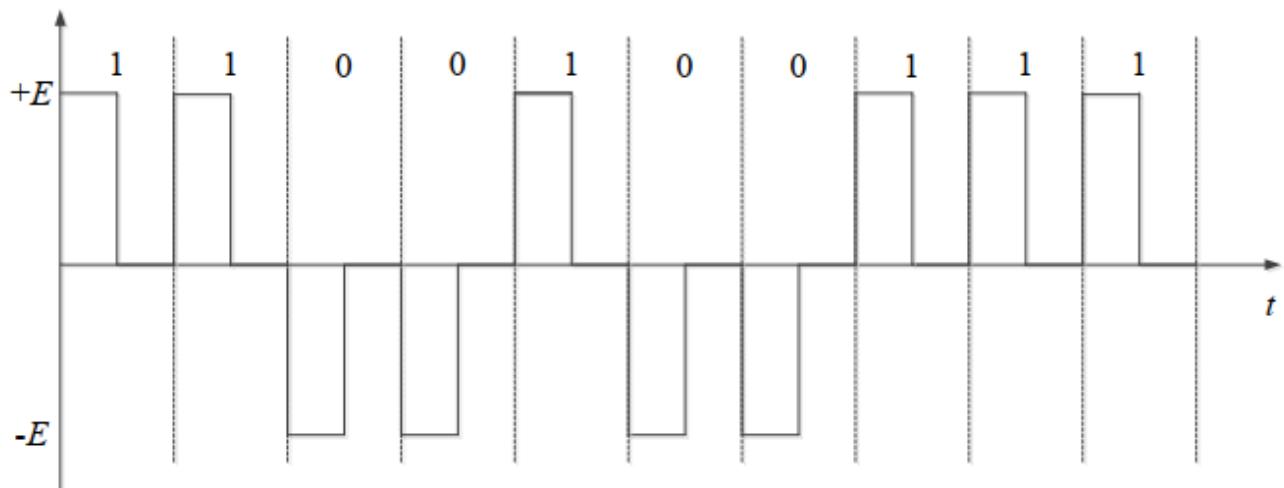


- 单极归零码

- 码元不为零的时间占一个码元周期的百分比称为**占空比**
- 若码元不为零时间为 $T_b/2$ , 码元周期为 $T_b$ , 则该单极归零码的占空比为50%

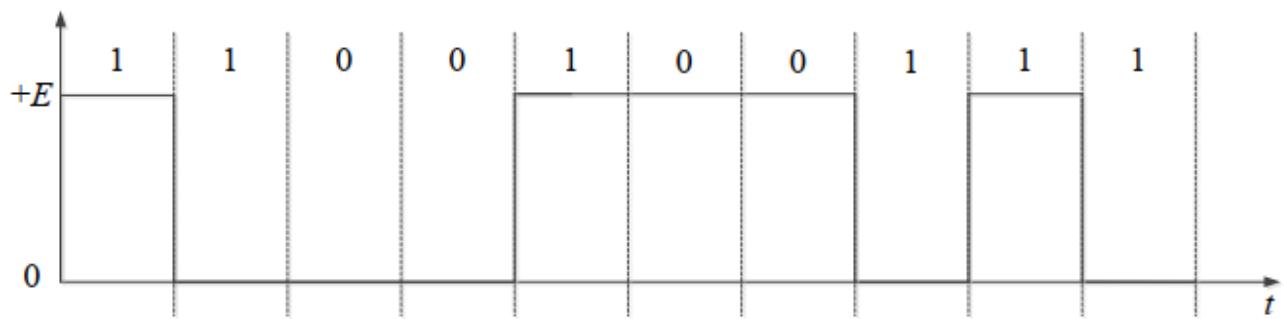


- 双极归零码



- 差分码

- 差分码又称为**相对码**
- 例如：相邻脉冲有电平跳变表示1，无跳变表示0



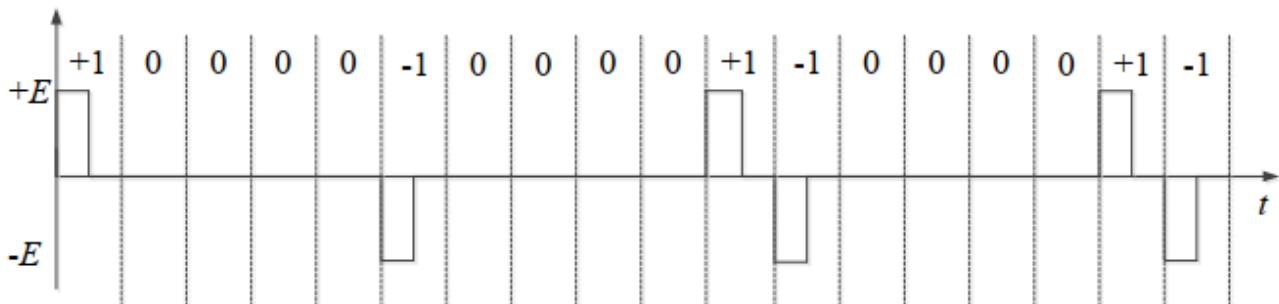
- AMI码

■ AMI (Alternative Mark Inversion) 码的全称是信号交替反转码

■ AMI码的编码规则:

- 信息码中的0编码为AMI传输码中的0 (零电平)
- 信号码中的1交替编码为AMI传输码中的+1 (正脉冲) 和-1(负脉冲)

■ 例如:



- 双相码

■ 双相码 (Biphase Code) 又称为曼彻斯特 (Manchester) 码

■ 双相码只有正、负两种电平

■ 每个比特持续时间的中间时刻要进行电平跳变

■ 正 (高) 电平跳到负 (低) 电平表示1

■ 负电平跳到正电平表示0

■ 双相码在每个比特周期中间时刻都会有电平跳变，因此便于提取定时信息

■ 双相码利用了两个脉冲编码信息码中的一个比特，相当于双极码中的两个比特

■ 10Mbps的以太网采用曼彻斯特码

- 差分双相码

■ 双相码的另一种码型是差分双相码，也称为差分曼彻斯特码

■ 差分双相码的每个比特周期的中间时刻也要进行电平跳变，但该跳变仅用于同步

■ 利用每个比特开始处是否存在电平跳变编码信息：

- 开始处有跳变表示1
- 无跳变表示0

■ IEEE802.5令牌环网采用差分曼彻斯特码

- nBmB码

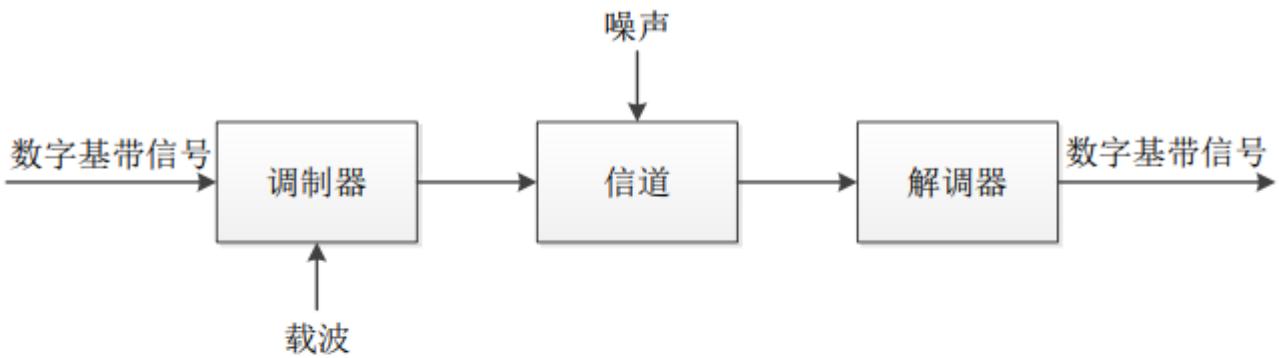
- $nBmB$ 码将 $n$ 位二进制信息码作为一组，映射成 $m$ 位二进制新码组，其中 $m > n$
- 由于 $m > n$ ，因此 $2^m$ 个码的新码组中只会用到 $2^n$ 个，多出 $(2^m - 2^n)$ 个码
- 可以从 $2^m$ 个码中优选出 $2^n$ 个码作为**有效码**，已获得良好的编码性能，其余码则作为**禁用码**，可以用于检错
- 快速以太网（100BASE-TX和100BASE-FX）传输码采用的是**4B5B编码**
  - 这样只需从 $2^5 = 32$ 个码中优化选择 $2^4 = 16$ 个码，以便保证足够的同步信息，并且可以利用剩余的16个禁用码进行差错检测

## 5.5 频带传输基础

频带传输

- 基带信号具有低通特性，可以在具有低通特性的信道中进行传输
- 许多**带通信道**（如无线信道）不具有低通特性，因此不能在这些信道中直接传输基带信号
- 只能利用基带信号去调制与对应信道传输特性相匹配的载波信号
- 通过在信道中传送经过调制的载波信号实现将基带信号所携带信息传出去
- 利用模拟基带信号调制载波，称为**模拟调制**，利用数字基带信号调制载波，称为**数字调制**
- 数字调制就是利用数字基带信号控制（或影响）载波信号的某些特征参量

数字调制系统



➤ 频带传输系统通常选择正弦波信号作为载波:

$$y(t) = a \cos(2\pi f t + \varphi)$$

➤ 二进制数字调制:

- 二进制幅移键控 (2ASK)
- 二进制频移键控 (2FSK)
- 二进制相移键控 (2PSK)

- 下面几个调制不同在于根据数字信号调整振幅、频率还是相位
- 差分相移键控: 0-相位不变、1-相位变化 $\pi$

二进制数字调制性能

**频带利用率:** 单位频带宽度上能够实现的数据传输速率 (bps/Hz)

**误码率:** 在信号传输过程中, 接收端接收到的比特中错误比特的比例

**信道特性:** 信号在传输过程中受到的影响和变化。如衰减、噪声、时延

	2ASK 幅移	2FSK 频移	2PSK 相移	2DPSK 差分相移
<b>频带利用率</b>	最高	最低	最高	最高
<b>误码率</b>	最高	高	低	最低
<b>对信道特性的敏感性</b>	敏感	不敏感	不敏感	不敏感

多进制数字调制

每个码元调制多个比特信息，即 $R_b = R_B \log_2 M$ ，M为进制数

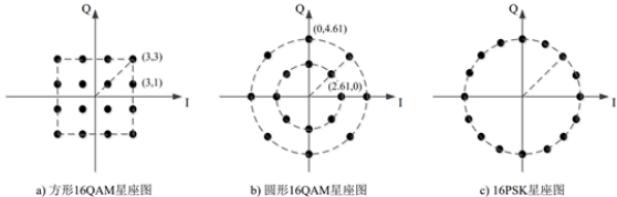
在确定带宽与频带利用率的情况下，多进制数字调制可提高数据传输速率

## 正交幅值调制QAM（幅值相位联合键控(APK)）

$$y'(t) = A_n \cos(2\pi ft) + B_n \sin(2\pi ft)$$

$$\begin{cases} A_n = s_n \cos(\varphi_n) & \text{同相信号 (I信号)} \\ B_n = -s_n \sin(\varphi_n) & \text{正交信号 (Q信号)} \end{cases}$$

- 频带利用率高
- 抗噪声能力强
- 调制解调系统简单



横轴通常表示一种振幅状态  
纵轴表示另一种振幅状态

- $R_b$  数据传输速率-比特率
- $R_B$  码元传输速率-波特率

## 5.6 物理层接口规程

### DTE (Data Terminal Equipment)

DTE是指用来处理或生成数据的设备，例如计算机、打印机、终端等。

### DCE (Data Circuit-terminating Equipment)

DCE是指数据电路终端设备，它提供了与通信线路的物理连接，并负责在通信介质上传输数据。DCE包括一些设备，例如调制解调器 (modem)、网络接口卡 (NIC)、交换机等。

DCE与DTE进行数据和控制信息的通信通过交换电路进行，需要明确的接口标准

**机械特性：**指明接口所用接线器的形状和尺寸、引线数目和排列、固定和锁定装置等等。

**电气特性：**指明在接口电缆的各条线上出现的电压的范围。

**功能特性：**指明某条线上出现的某一电平的电压表示何种意义。

**过程特性：**指明对于不同功能的各种可能事件的出现顺序。

## 5.7 802.11无线局域网

分类

## 802.11b

- ❖ 2.4-2.5GHz免费频段 (unlicensed spectrum)
- ❖ 最高速率： 11 Mbps
- ❖ 物理层采用直接序列扩频 (DSSS)技术
  - 所有主机使用相同的码片序列

## 802.11a

- 5-6 GHz频段
- 最高速率： 54 Mbps

## 802.11g

- 2.4-2.5 GHz频段
- 最高速率： 54 Mbps

## 802.11n: 多天线(MIMO)

- 2.4-2.5 GHz频段
- 最高速率： 600 Mbps

- 
- ❖ 均使用**CSMA/CA**多路访问控制协议
  - ❖ 均有基础设施(基站)网络模式和特定网(自组网)网络模式

标准	频段	数据速率	物理层	优缺点
802.11b	2.4 GHz	最高11 Mb/s	扩频	最高数据率较低，价格最低，信号传播距离最远，且不易受阻碍
802.11a	5 GHz	最高54 Mb/s	OFDM	最高数据率较高，支持更多用户同时上网，价格最高，信号传播距离较短，且易受阻
802.11g	2.4 GHz	最高54 Mb/s	OFDM	最高数据率较高，支持更多用户同时上网，信号传播距离最远，且不易受阻，价格比802.11b贵
802.11n	2.4 GHz 5 GHz	最高600 Mb/s	MIMO OFDM	使用多个发射和接收天线以允许更高的数据传输率，当使用双倍带宽(40 MHz)时速率可达600 Mb/s

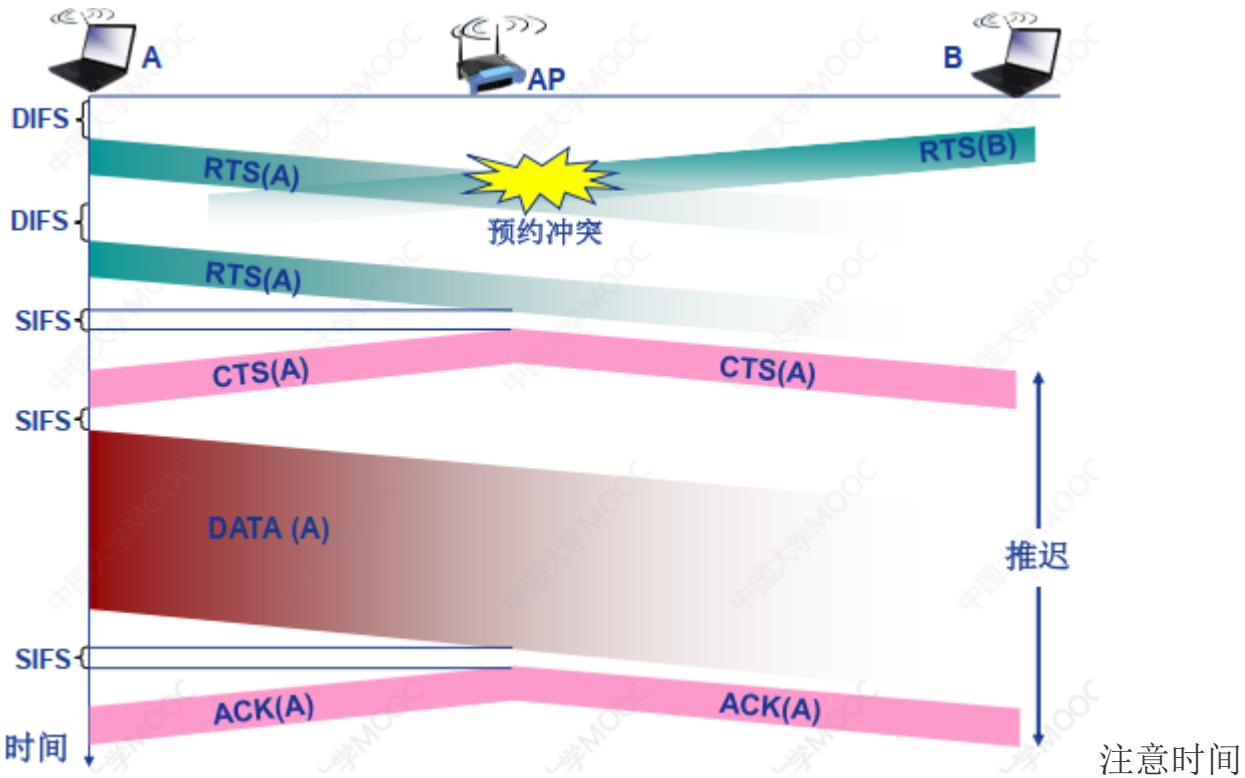
## 802.11体系结构



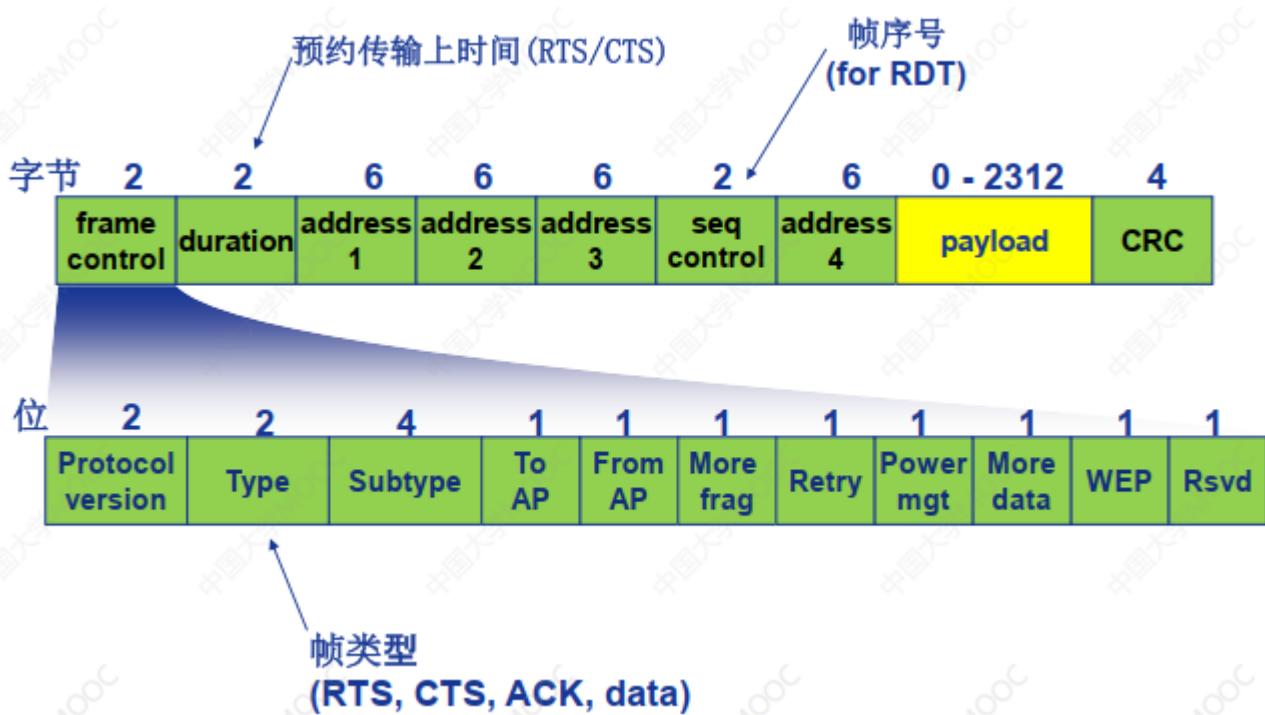
- ❖ 无线主机与基站通信
  - 基站(base station) = 访问点(access point-AP)
- ❖ 基本服务集BSS(Basic Service Set)，也称为单元(cell)
  - 基础设施网络模式:
    - 无线主机
    - AP: 基站
  - 自组网(ad hoc)模式:
    - 只有主机

### CSMA/CA协议-带有冲突避免的载波监听多路访问协议

- 发送端
  - 若监听信道空闲了DIFS时间，发送整个帧
  - 信道忙
    - 开始随机退避计时
    - 信道空闲，计时器倒计时
    - 计时器超时，发送帧
    - 如果没有收到ACK，增加随机退避间隔重复第2步
- 接收端
  - 正确接受帧-延迟SIFS时间后向发送端发送ACK
- 冲突避免-允许发送端预约信道
  - 发送端首先利用CSMA向BS发送一个很短的RTS帧-可能出现冲突
  - BS广播一个CTS帧作为回应
  - CTS帧可以被所有结点接收
    - 消除隐藏站影响
    - 发送端可以发送数据帧
    - 其他节点推迟发送



## 802.11MAC帧



- ❖ 802.11 数据帧有 4 个地址字段
- ❖ 地址 4 用于自组网络
- ❖ 地址 1~地址 3:

去往 AP (To AP)	来自 AP (From AP)	地址 1	地址 2	地址 3
0	1	目的地址	AP 地址	源地址
1	0	AP 地址	源地址	目的地址

# 夹带私货

识之律者女士万岁！

