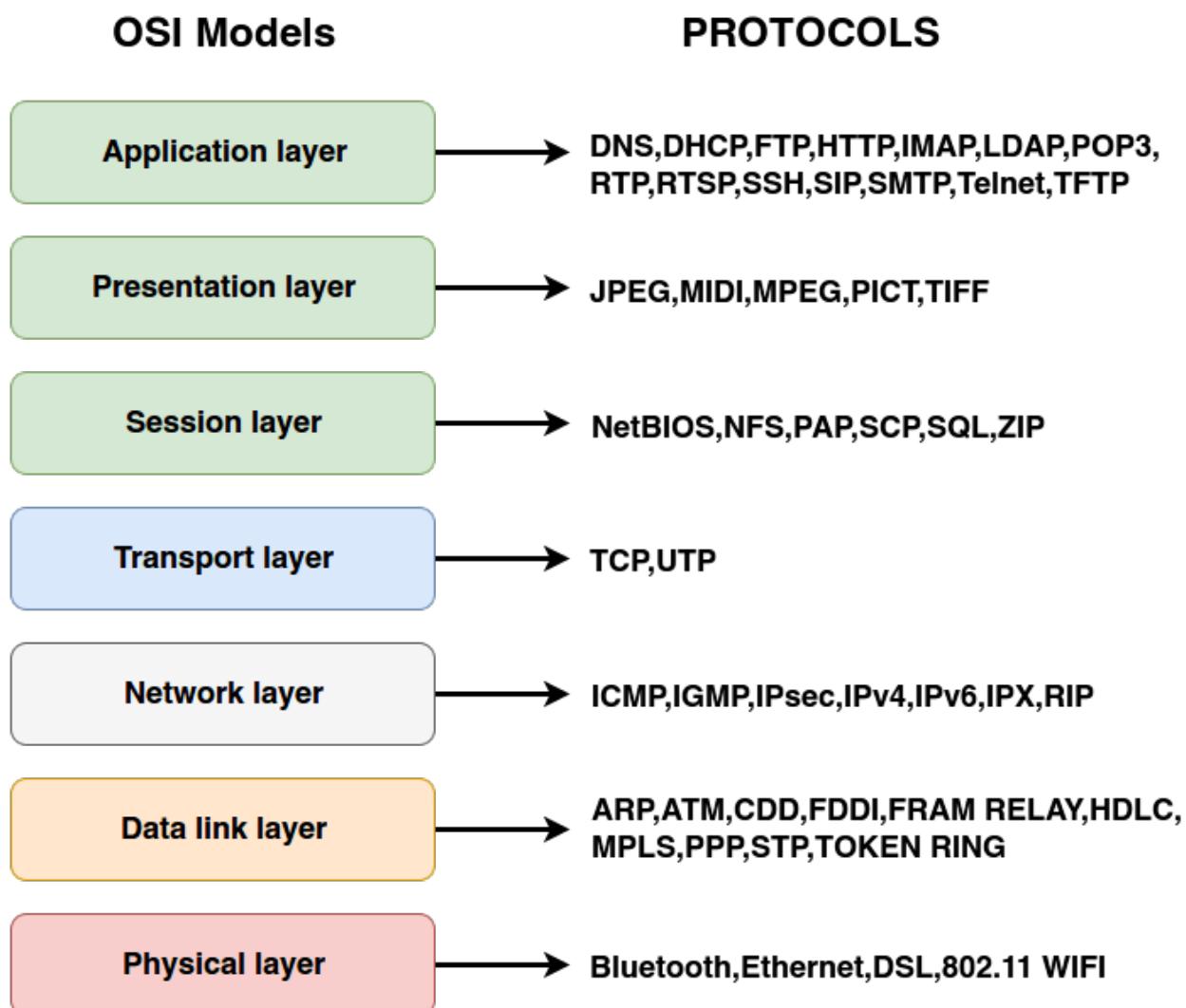


# Computer Networks Final Review

Collected by Bill Chen, 2019.12

## Chapter 1 Overview

7 - layer OSI Model



- 计算机网络：将分散的，具有独立功能的计算机系统通过通信设备与线路连接起来有完整的软件实现资源共享和信息传递
- 计算机网络是互联的，自治的

## 概念和组成

### 计算机网络的功能

- 数据通信

- 数据在信道上的传输
- 资源共享
  - 硬件、软件和数据
- 分布式处理
  - 多台计算机共同处理同一个任务
- 提高可靠性
- 负载均衡 (多台计算机可以更亲密的沟通)

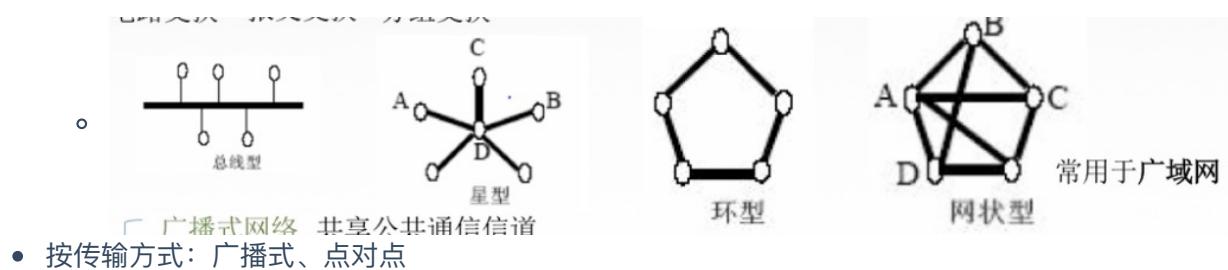
## 计算机网络的组成

- 硬件、软件、协议
- 工作方式 - 边缘部分、核心部分
- 功能组成 - 通信子网 (OSI下三层通信子网) 、资源子网 (上三层的处理)



## 分类

- 广域网WAN、城域网MAN、局域网LAN、个人区域网PAN
- 按照使用者分类：公用网、专用网（军队、政府、铁路、公安）
- 按照交换技术：电路交换、报文交换、分组交换
- 按照拓扑结构分：



## 标准化

### 标准化相关工作

- 法定标准：Eg. OSI
- 事实标准：Eg. TCP/IP

- 指定标准的流程：
  - RFC (Request For Comments): 经过四个阶段
    - Internet Draft > Proposed Standard > (Draft Standard) > Internet Standard
    - 因特尔草案 - 建议标准 - 因特尔草案

## 组织

- 国际标准化组织 ISO
- 国际电信联盟 ITU
- 国际电气电子工程师协会 IEEE - 学术标准、IEEE802相关标准
- Internet 工程任务组 IETF

## 传输指标

- 速率：数据传输率 / 比特率 b/s kb/s Mb/s Gb/s Tb/s
  - 1Byte = 8bit
- 带宽：原本：某个信号具有的频带宽度，即最高频率与最低频率之差
  - 计算机网络中：指传送数据的能力，最高数据率，**网络设备所支持的最高速度**
  - 单位：比特率 b/s kb/s Mb/s
  - 带宽变大：指的是能够注入的数据变多了，不能说是比特传输的速度变快了
- 吞吐量 (throughput): 单位时间内通过**某个网络或接口**的数据量
  - 受到网络的带宽或网络的额定速率的限制
  - 吞吐量指的是实际的数据量（不是最大承载能力）
- 时延：Latency，数据从网络一端到另一端所需要的时间
  - 发送时延（数据长度/信道带宽），传播时延（信道长度/电磁波的传播速率），排队时延（等待输出/输入链路可用的时间），处理时延（校验、寻找出口的时间）
  - 高速链路只能改变发送时延
- 时延带宽积：实验带宽积=传播时延 \* 带宽
  - 描述：某段链路现在有多少比特
- 往返时延：RTT
  - 指的是**从发送方发送数据开始，到发送发收到接收方的确认，总共经历时延**
  - 可以通过 ping 命令测试RTT
  - RTT 越大，可以发送的数据越多（在等待收到接收方的确认之前）
  - = 末端处理时间 + 传播时延 \* 2
- 利用率：分为信道利用率 & 网络利用率
  - 信道利用率：有数据通过的时间 / 有+无数据通过的时间
  - 网络利用率：信道利用率的加权平均值
  - 通常：利用率特别高的时候会增加时间，反倒会降低速度

## 分层结构

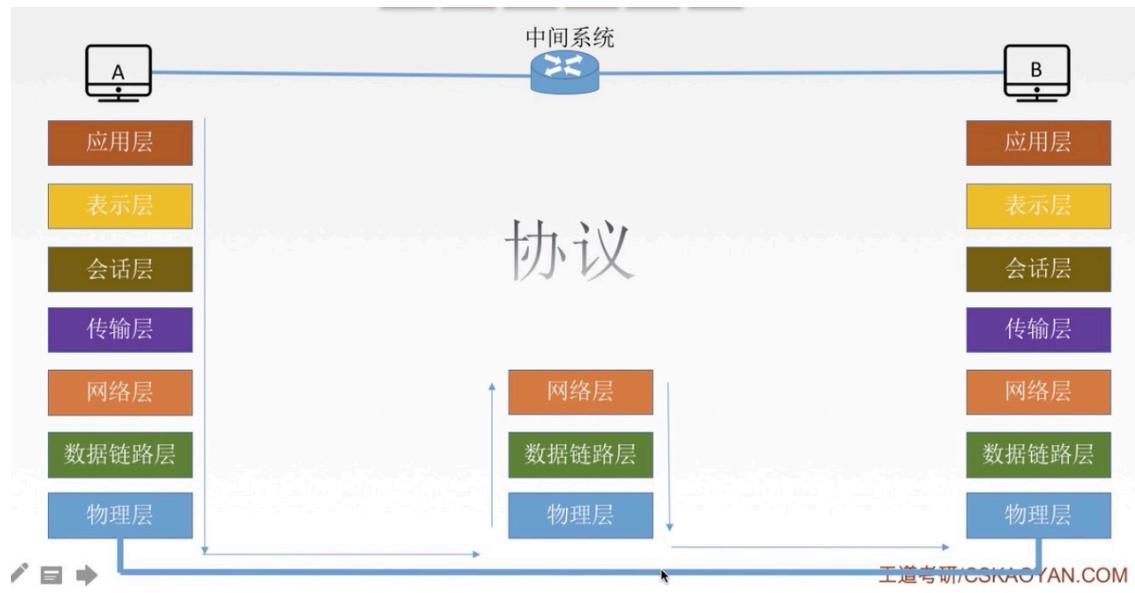
- Background：将网络通信的问题化为小问题解决
- 基本原则：每层之间相互独立
  - 每层之间界面自容清晰

- 结构上可以分隔开
- 保持下层对上层的独立性
- 促进标准化工作
- 实体第 n 层活动元素称为第 n 层实体
- 协议：为进行**对等实体**的交换而建立的交换规则，标准或约定
  - 语义、语法、同步问题的姐姐
- 接口（访问服务点SAP），**上层使用下层服务的入口**
- 服务：下层相邻上层提供的功能调用，仅仅在相邻层之间提供服务，具体的硬件软件没有规定
- **计算机网络体系结构是计算机网络的各层及其协议的集合**

## 通用分层模型

### OSI

- History: DEC公司的DNA，美国国防部的TCP/IP等之前的许多网络体系结构
  - 只能实现公司内部的网络通信，不能实现全球的互联互通
- ISO 于 1984 提出的开放系统互联（OSI）参考模型
  - ISO 提出的参考模型和互联互通参考模型
  - 理论成功，市场失败（没有办法进入市场，某些功能会在多个层次重复出现，多个层次重复出现的功能）
- 物联网淑慧试用（自下向上传的七层结构的名称）
- 通信过程

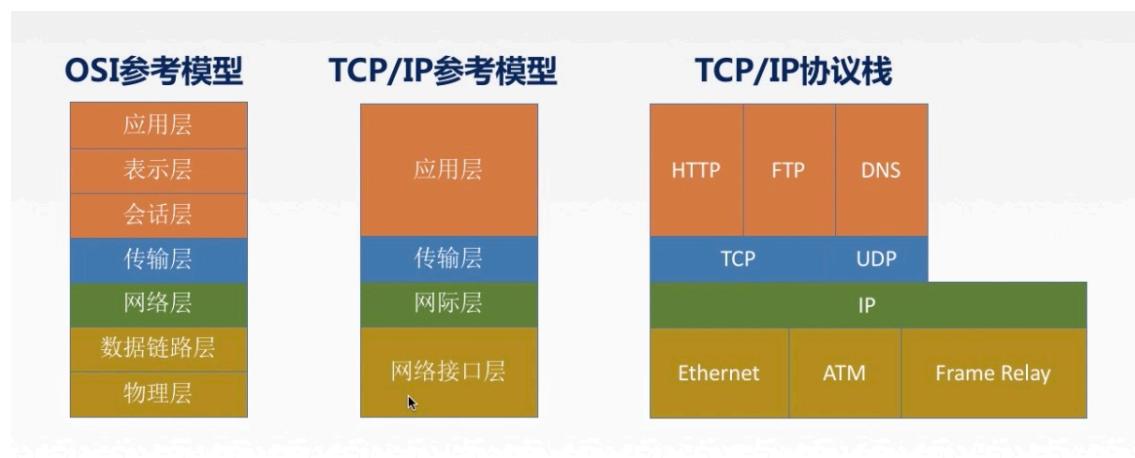


- 上四层端到端（客户端和客户端之间的通信），下三层点到点（某一个中介系统可能会传播到下一个节点）
- 各层：应用层：所有能和用户产生网络流量的程序
  - 表示层：处理两个通信系统中交换信息的方式
    - 数据格式交换
    - 数据加密解密
    - 数据压缩和恢复
  - 会话层：有序地传输数据，建立同步

- 使用校验点在通信失效的时候继续恢复通信，实现数据的同步传送
- 传输层：用处：
  - 提供可靠传输（需要确认机制）、不可靠传输（不需要确认机制）
  - 差错控制
  - 流量控制
  - 复用分用：
    - 复用：多个应用层进程可以同时使用下面传输层的服务（对于发送方）
    - 分用：传输层把收到的信息分别交付给上面应用层中的相应进程（对于接收方）
- 网络层：
  - 路由选择（最佳路径）
  - 流量控制
  - 差错控制
  - 拥塞控制（采取措施缓解所有节点都来不及接受）
- 数据链路层：
  - 把网络层传下来的数据组装成帧
  - 数据链路层/链路层的传输单位是帧
- 物理层：
  - 在物理媒体上实现的比特流的透明传输（单纯地转化成物理数据）
  - 定义接口特性、定义传输模式（单工、双工、半双工）
  - 定义传输速率
  - 比特同步、比特编码（曼切斯特编码等等）

## TCP/IP

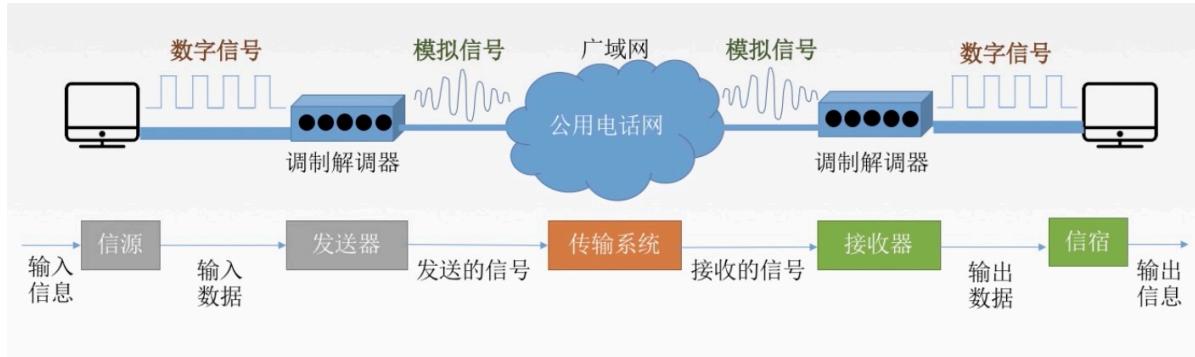
- 在 OSI 模型出现的时候 TCP/IP 已经具有了市场运用（TCP/IP 为实践）
- 应用层、传输层、网际层、网络接口层：



## Chapter 2 Physical Layer

### 通信基础

- 一个典型的数据通信模型：



## 一些概念

- 数据：传送信息的实体
- 信号：数据电气/电磁的表现（数字信号&模拟信号）
- 信源：产生和发送数据的源头
- 信宿：信号传送的终点
- **信道**：信号传输的媒介。
- 通信方式：单工、半双工、全双工

## 通信相关：

- 码元：一个固定时长的信号波形（数字脉冲），代表不同离散数值的基本波形
  - 数字通信中数字信号的计量单位
  - 1 码元可以携带多个比特的信息量
    - K进制的码元 - 码元的离散状态的个数（同样的一个波形可以对应几个比特）
- 速率：码元传输速率：1s传输多少个码元
  - 信息传输速率：别名**比特率**、**信息速率** - 1s传输多少个**比特**
    - 若一个码元携带 nbit 的信息量，M baud 的码元传输速率对应的信息传输速率 = M\*n bit/s
- 波特：**码元传输速率的单位**，每秒钟内通信线路状态改变的次数。
- 带宽：表示在单位时间内从网络的某一点到另一点所能通过的**最高数据率**（理想中的）
  - Eg.

某一数字通信系统传输的是四进制码元,4s传输了8000个码元,求系统的码元传输速率是多少?信息传输速率是多少?若另一通信系统传输的是十六进制码元,6s传输了7200个码元,求他的码元传输速率是多少?信息传输速率是多少?并指出哪个系统传输速率快?

2000Baud, 4000b/s; 1200Baud, 4800b/s; 十六进制更快

### 四进制码元系统

码元传输速率就是 $8000/4=2000$ Baud, 信息传输速率就是 $2000 \times \log_2 4 = 4000$ b/s

### 十六进制码元系统

码元传输速率就是 $7200/6=1200$ Baud, 信息传输速率就是 $1200 \times \log_2 16 = 4800$ bit/s

## ★ 奈氏准则和香农定理

### 失真

- 现实中的信道干扰对信号源造成的干扰，部分干扰可恢复，部分不可恢复
- 码间串扰：码元传输速度过快的时候让码元不清楚

## 奈氏准则（奈奎斯特定理） / Nyquist's Theorem

- 在无噪声条件下，为了避免码间串扰，极限码元传输速率为 $2W \text{ Baud}$ ，其中 $W$ 为信道带宽，Hz
  - 极限数据传输速率 =  $2W \log_2 Vb/s$ ，其中  $W$  为带宽， $V$  为码元进制数
- 定理本身为：传输上限为 $2W$ 。

## 香农定理 / Shannon's Theorem

- 考虑了噪声的影响
- 信噪比 =  $10 \log_{10}(S/N) dB$  （如果不是给定的分贝的形式一般可以直接代入计算，如果给定的分贝单位需要手动计算 $S/N$ ）
- 信道的极限传输速率 =  $W \log_2(1 + S/N)$ ， $W$  为带宽
  - 只要信息的传输速率低于信道的极限传输速率，就一定能找到方法来实现无差错的信息传送
- ! 在计算时要算两个定理取最小值

## ★ 编码与调制

- 信道：传输媒介
- 信号：分为基带信号和宽带信号
  - 基带信号：将数字 1 和 0 直接用两种不同的电压来表示，再送到数字信道上去传输（基带传输）。
  - 宽带信号：将基带信号进行调制之后形成的频分复用模拟信号，再传送到模拟信道上去传输
  - SUM：数字信道 - 基带信号；模拟信道 - 宽带信号
    - 传输距离比较近的时候，使用基带传输（信号不容易变化），传输距离较远时，使用宽带传输
- 将数据转换为：数字信号：编码；模拟信号：调制
- 数字数据编码为数字信号：
  -

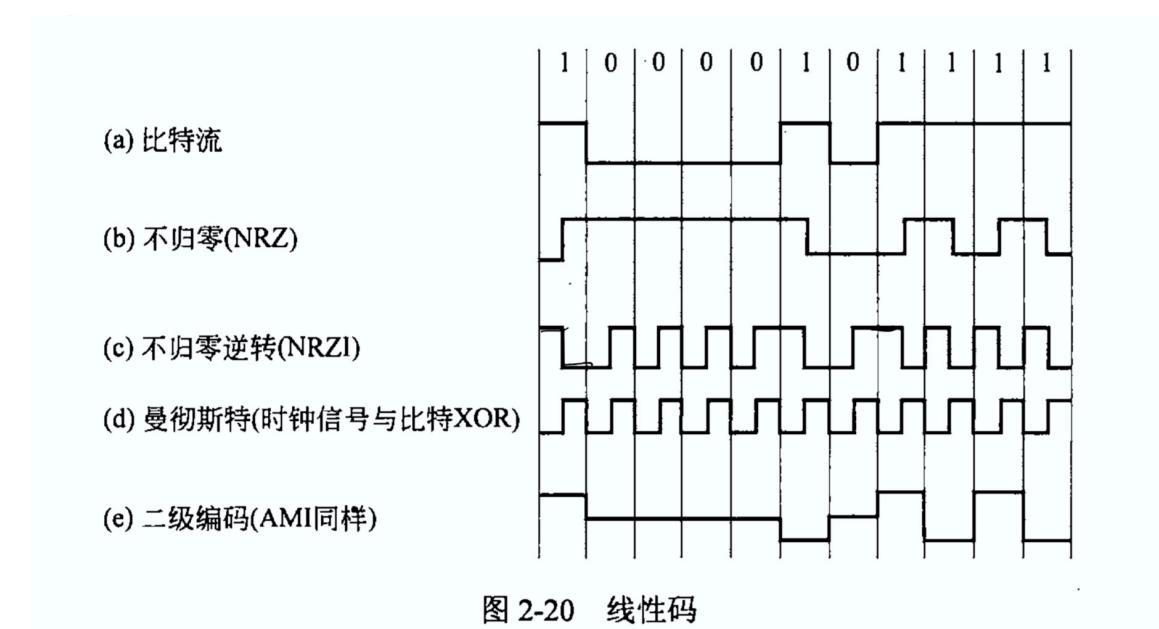
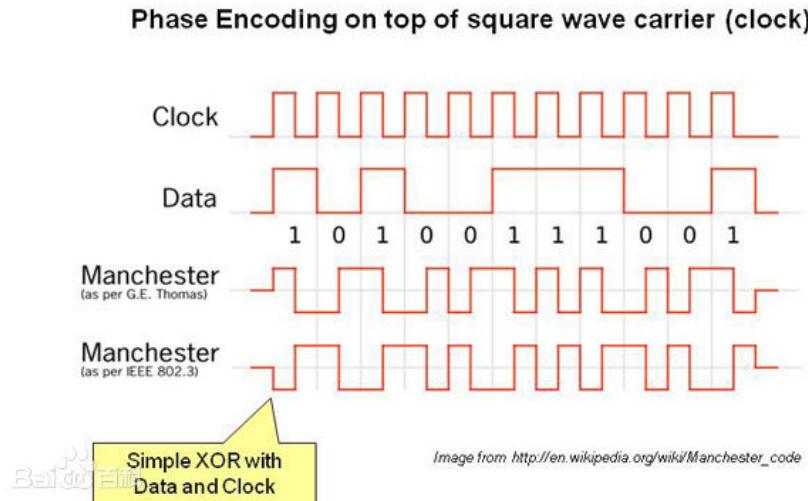


图 2-20 线性码

- NRZ - 非归零编码（编码容易实现，没有检错功能，需要建立一个信道确立时钟周期
  - 1传输完之后不归零
- 不归零逆转编码 —— 对于全 0 没有问题，对于全 1 有问题
- 曼彻斯特编码：自同步的编码方式
  - 数据传输速率只有调制速率的1/2

## Manchester Encoding



- 利用数据的上升沿或下降沿表示 0 或 1
- 差分曼彻斯特：同 1 异 0，前半个码元的电平与上一个码元
- 数字数据调制为模拟信号
  - 调幅 - 2ASK；调频 - 2FSK；调相 - 2PSK
  - 调幅 + 调相 (QAM)
    - 在第一种调制的基础上再对第二种进行调制
    - 根据不同的进制和相位计算波特率（求信号状态种数）
- 模拟数据编码为数字信号
  - PCM：能够达到最高保真水平的脉冲编码调制
    - 抽样：对模拟数据进行周期性扫描， $f_{采样频率} > 2f_{信号最高频率}$
    - 量化：把抽样的电频幅度按照标准转化为数字
    - 编码：把量化的结果转换成对应的二进制编码进行传送
- 模拟数据调制为模拟信号
  - 简单的调制与解调的过程

## 物理设备

### 传输介质

- 传输媒体 ≠ 物理层
  - 传输媒体只是单纯地传输信号
- 主要的传输介质

- 双绞线 - 绞合可以减少对相邻导线的电磁干扰
- 同轴电缆：宽带同轴电缆（数字信号） & 宽带同轴电缆（模拟信号）
  - 抗干扰性比双绞线更好
- 光纤：利用传递光脉冲进行通信，带宽超级大
  - 多模光纤（近距离） & 单模光纤（远距离）

## 物理层设备

- 中继器：对信号进行再生和还原，对衰减的信号进行放大
  - 中继器的两端连接的是网段，而不是子网；在两端使用的是同一个协议
- 集线器（多口中继器）：对信号再次进行放大转发，接着转发到其他所有处于工作状态的端口上，增加信号传输的距离，延长网络的长度
  - 注意：不具备信号的定向传送能力，共享式设备

# Chapter 3 Data Link Layer

---

## Overview

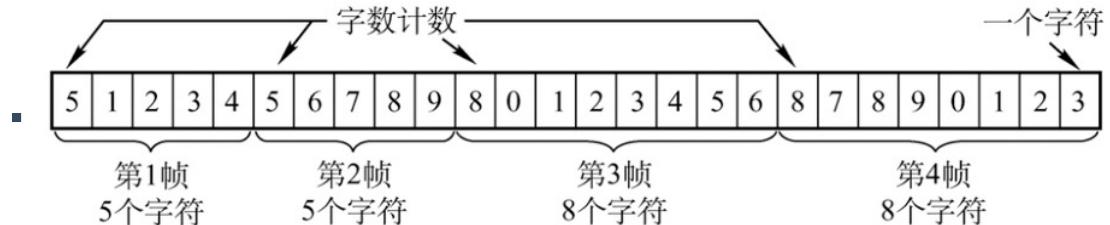
- 数据链路层的功能：在物理层的基础上为网络提供无差错的服务
  - 无确认无连接服务、有确认无连接服务、有确认面向连接服务等
  - 链路管理
  - 成帧 / Framing
  - 流量控制
  - 差错控制

## 一些概念

- 结点：主机和路由器
- 链路：两个结点之间的物理通道，链路的
- 数据链路：网络中的两个结点之间的逻辑通道，实现控制数据传输协议的硬件和软件加到链路上的有效线路

## ★ 成帧

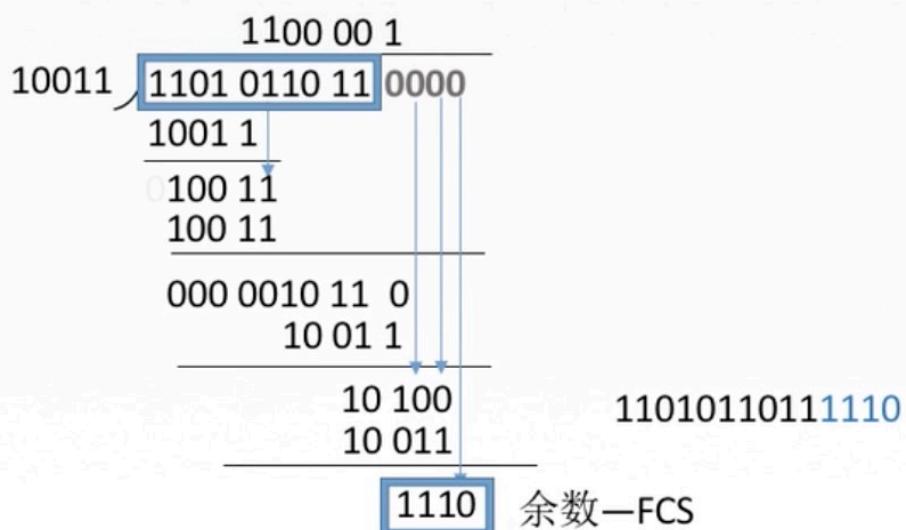
- 在数据前后添加首部和尾部
- MTU 最大数据传输单元 = 帧中的载荷大小
- 透明传输：不管传送的什么样的比特组合，都应该在数据链路层上顺利传输
- 几种方法：
  - 字符计数法：帧首部使用一个计数字段来表明字符数



- 字符填充法： SOH 和 EOT 填充为 Start of header 和 End of transmission
  - 传输的是二进制代码可能会出现结束字符一样的比特组合的时候，需要用字符填充法解决
  - 在控制信息的字符前添加 ESC 转义字符
- 零比特填充法：在发送方只要连续 5 个 1，就填入 1 个 0。
  - 在接收端收到一个帧时，先找到标志字段确定边界，再扫描，5 个 1 就去掉 1 个 0。
- 违规编码法：对于曼彻斯特编码使用“高高”、“低低”来确定帧的起始和终止

## ★ 差错控制

- 差错来源：来自传输通道本身的随机噪声、外接短暂的冲击噪声
- 差错种类：位错、帧错（丢失、重复 & 失序）
- 控制的时间：在每一段链路都进行纠正
- 检验编码
  - 奇偶校验码
    - 构造：由 n-1 位信息元 1 位校验元 构成
    - 奇校验码：使得加上校验位之后，1 的个数为奇数；偶校验码：加上校验位之后 1 的个数为偶数（x..... 中，1 的个数）
    - 对于任何的比特数，检错率为 50%
  - CRC 循环冗余码
    - 发送端将数据除以一个生成多项式，余数为 FCS 帧检测序列 / 冗余码
    - 冗余码的计算：在原数据之后加 r 个 0，使用模 2 除法，在竖式计算中使用的是异或的方式做减，最后的余数作为冗余码，将冗余码加到刚刚补的 0 后面一起传送
      - Eg.



- 接收端：求余数，为 0 则接受，不为 0 则认定由差错，直接丢弃。

- CRC 使用的是硬件级别的计算，计算速度非常快
- 纠错编码 - 海明码 Hamming Code
  - 海明码：发现双比特错，纠正单比特错
  - 海明不等式： $2^r \geq k + r + 1$ , r 为冗余信息位, k 为信息位
  - 计算方式：校验位的目标：所要校验的位异或 = 0。校验位为第  $2^n$  位。

**D=101101**

二进制	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
数据位	1	2	3	4	5	6	7	8	9	10
代码	P <sub>1</sub>	P <sub>2</sub>	D <sub>1</sub>	P <sub>3</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	P <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>
实际值			1		0	1	1		0	1

- 纠错方式：依次对校验位所校验的位进行异或计算，然后从 P<sub>4</sub>写到P<sub>1</sub>，如果不为 0，这个二进制数所代表的位数就是出错了位置，再纠正即可

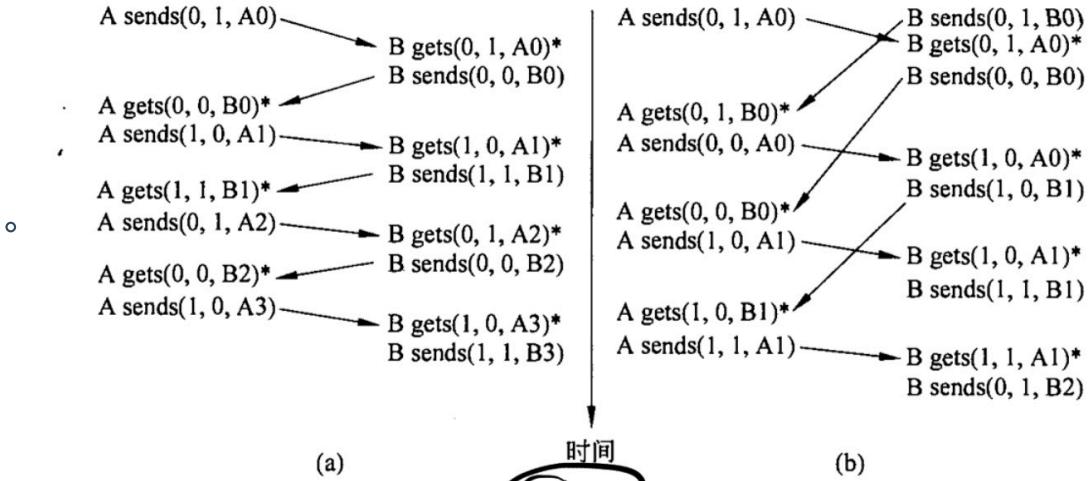
## 流量控制与可靠传输协议

### Background

- 为了控制发送方的发送速度
- 和传输层的区别：数据链路层的流量控制是点对点的，传输层的流量控制是端到端的。
  - 数据链路层的手段：接收方收不下就不回复确认
  - 传输层的流量控制手段：接收端给发送端一个窗口公告
- 主要方法：停止-等待协议、滑动窗口协议（GBN & SR）
  - 停等协议：发送窗口 = 1 接受窗口 = 1
  - 后退 N：发送窗口 > 1, 接收窗口 = 1
  - 选择重传：发送窗口 > 1, 接收窗口 > 1
- 滑动窗口主要解决的问题：流量控制和可靠传输

### 停止-等待协议 / Stop-Wait Protocol

- 是哪一层的问题？
  - 计算机网络发展前期质量不好，链路层要负责可靠传输，需要使用这些协议
  - 现在链路层可以暂时抛弃可靠传输，交给传输层解决
- 关于协议
  - 解决的问题：为了解
  - 决丢包的问题
  - 研究的前提：仅讨论单工通信
- 协议细节：



- 每次发送之后都启动一个计时器，超时计时器应当比 RTT 更长一些（自动重传）
- 发送完后必须保留副本

## 回退 N 协议 / Go Back N Protocol

- 停止等待协议信道利用率过低
- 协议中的窗口：
  - 发送窗口：发送方维持一组连续的允许发送的帧序号
  - 接收窗口：接收方维持一组连续的允许接收帧的序号
- 累计确认：GBN协议采用了累计确认的方式，表明接收方已经收到了 n 号帧和它之前的全部帧
- 超时事件：出现超时的时候，发送方会重传所有已发送但没有被确认的帧
- 协议细节：

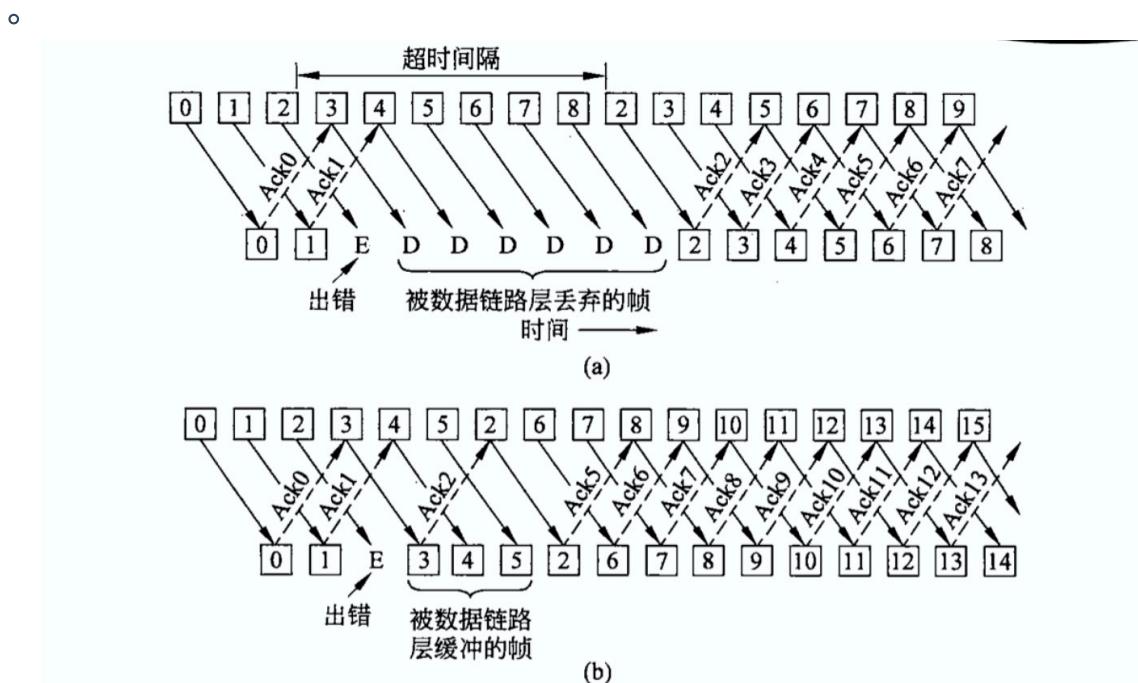


图 3-18 管道化以及差错恢复

(a) 接收方窗口为 1 时错误的影响；(b) 接收方窗口很大时错误的影响

- 对于发送方：确认窗口，开启计时器，超时回退重传
- 对于接收方：若正确，发送 ACK，并将数据交付上层
  - 若错误，则直接丢弃帧，并且按照最近接收到的帧重新发送 ACK，接收方无需缓存任何

- 失序帧，只需要维护一个 `expectedseqnum`
- 规定窗口的大小不能超过  $2^{(n-1)} - 1$ ，其中  $n$  为有多少个比特来标志了序号。

## 选择重传协议 / Selective Repeat Protocol

- 解决问题：对于 GBN 算法中可能会出现批量重传的问题（一次性损失太大）
  - 只重传出错的帧，设置 单个确认而不是累计确认，设置接收缓存，缓存乱序的帧
- 协议中的窗口：
  - 发送窗口 & 接收窗口均大于 0，如果下界未收到确认，将不会滑动
  - 接收窗口：只有窗口下界接收到才会滑动，后面为收到的处于等待接状态



- 协议细节：

◦

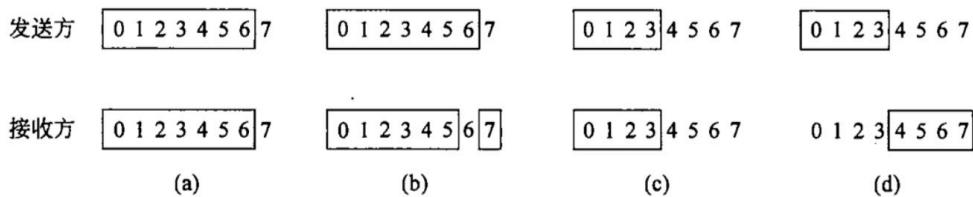


图 3-22

(a) 窗口大小为 7 的初始情形；(b) 发出 7 个帧并且接收 7 帧，但尚未确认；  
 (c) 窗口大小为 4 的初始情形；(d) 发出 4 个帧并且接收 4 帧，但尚未确认

- 调用：检查下一个帧序号，如果位于窗口中则发送，否则缓存数据或者先返回上层
- 发送方：收到 ACK，则帧将会标记为已接收，**如果是窗口下界则前进窗口**
  - 超时则重传未收到确认的帧
- 接收方：对于窗口内的帧来者不拒（不管其顺序）
  - 失序的帧将会被缓存，并且向发送方发送确认帧。
  - 如果收到了窗口之外的帧，说明 **ACK 在传输过程中丢失**，并且重新发送确认帧
- 滑动窗口的长度：
  - 发送窗口最好等于接收窗口（大了会溢出，小了没意义）
  - 规定窗口的大小不能超过  $2^{(n-1)} - 1$ ，其中  $n$  为有多少个比特来标志了序号。（为了避免出现二义性）

# Chapter 4 Media Access Control

---

## 一些术语

- DCF: Distributed Coordination Function -- ad hoc architecture
- PCF: Point Coordination Function (Optional)
- OFDM: Orthodox Frequency Division Multiplexing (正交频分多路复用)
- DSSS: Direct-Sequence Spread Spectrum(直接扩频)
- FHSS: Frequency-Hopping Spread Spectrum (跳频扩频技术)
- BSS: Basic Server Set
- ESS: Extent Server Set (服务集)
- 发展历史: 802.11b (QPSK 2.4GHz) -> 802.11a (OFDM 5.0GHz) -> 802.11g (OFDM .4GHz)

## 一些概念

- 点对点链路: PPP 协议, 用于广域网, 相邻节点通过一个链路相连
- 广播式链路: 所有主机共享通信介质
  - 典型的拓扑结构: 总线型、星型 (逻辑总线型)
- 介质访问控制: 采取一定措施, 使得两对节点之间的通信不会发生互相干扰
  - 静态划分信道、动态分配信道
- 多路复用: 用一条广播信道, 把多条信道放在一个信道上

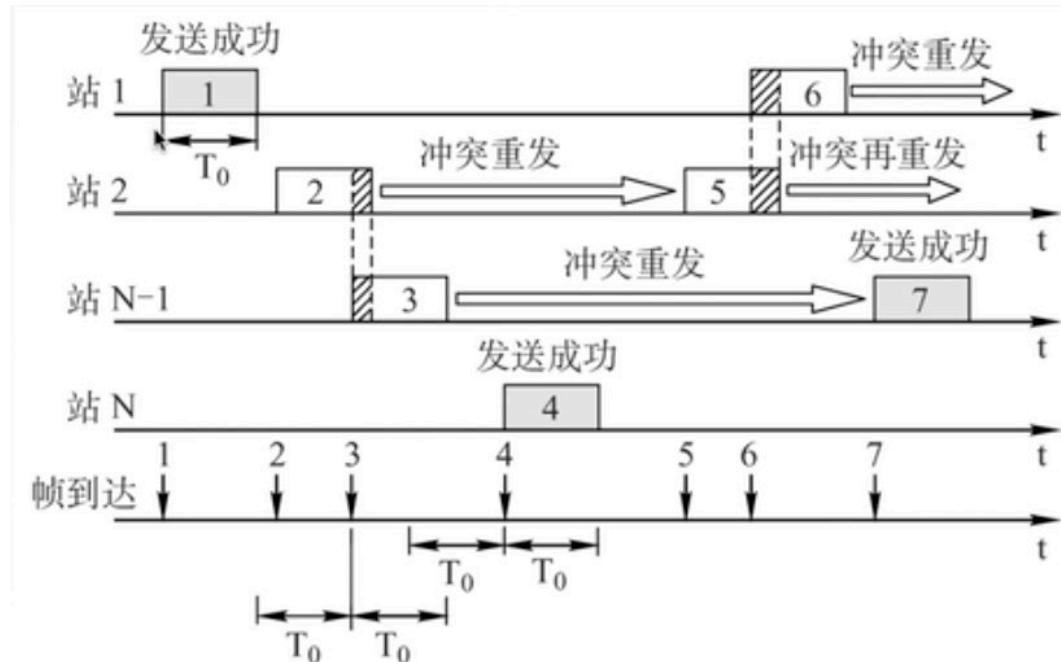
## 静态信道划分方式

- FDM 频分多路复用: 所有用户占用不同的带宽, 根据频率的不同 (类似于并行)
- TDM 时分多路复用: 把时间划分为一段段等长的时分复用帧 (类似于并发)
- STDM 统计时分复用: 显著提高时间利用率, 分配时间片
- WDM 波分多路复用 : 光的频率多路复用
- CDMA 码分多路复用: 将 1 个比特片分为多个码片 / 芯片, 每一个站点被指定一个唯一的 m 位的芯片序列, 要求各个站点的芯片序列相互正交 (通常把 0 写成 -1)
  - 发送 1 时为芯片序列, 发送 0 为反码序列
  - 合并方式: 各路信道在信道中线性相加
  - 分离方式: 合并的数据和来源的芯片序列规格化内积, 即对应位数相乘之后 / 8
    - 结果为 1 - 发送 1; 结果为 -1 - 发送 0; 结果为 0 - 未发送数据

## 动态信道划分方式

- 令牌传递协议 (轮询访问介质访问控制)
  - 令牌环网中具有多台设备
    - TCU (转发器) : 传递所有接入的帧,
  - 传递 token 获取发送权限: 令牌为一个特殊的 MAC 控制帧, 不含任何信息
    - 主机得到令牌后: 修改令牌的标志位, 在令牌之后加入数据并发送数据

- 令牌会沿着令牌环传递
- 每个节点都会在一定时间内获得发送数据的权力，并不无限持有令牌
- 主要问题：令牌开销、等待延迟、单点故障
- 主要适用环境：负载较重、通信量较大的网络
- ALOHA 协议（以下为随机访问介质访问控制）
- 纯 ALOHA 协议：随机重新发送，不按时间槽发送



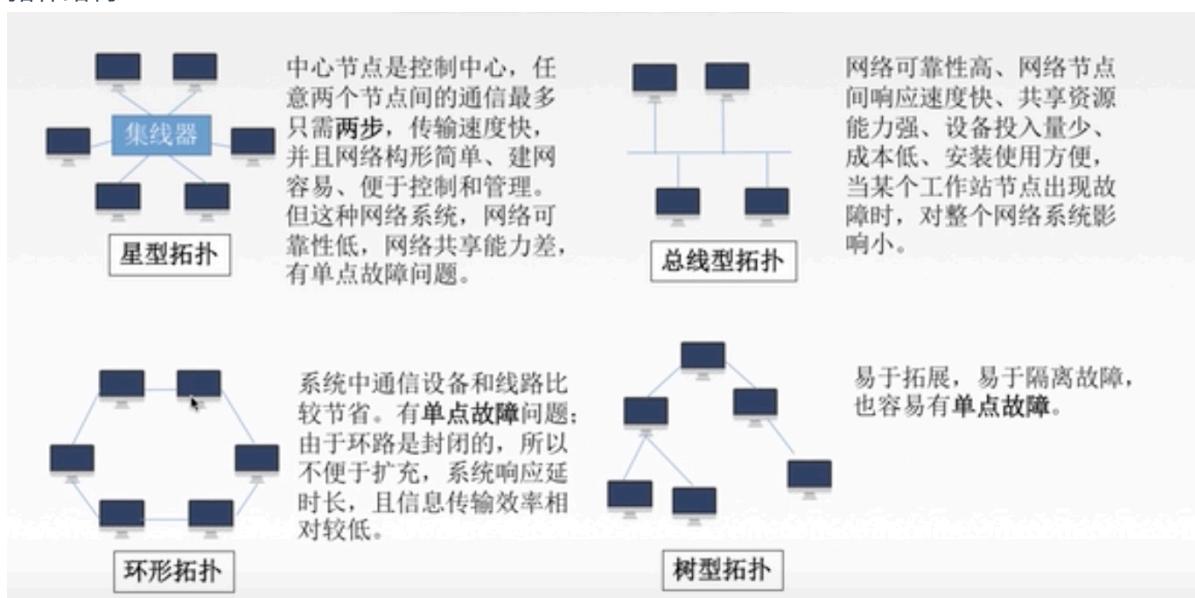
- 如果发生了冲突，则重新发送（接收方会检测是否发生了冲突，若发送方在一段时间内接受不到，则认定为冲突发生）
- 分槽 ALOHA 协议：分时间段内槽发送，所有用户在时间片开始的时候同步接入网络信道，若发生冲突，则必须等到下一个时间片开始的时候才发送
  - 相比纯 ALOHA：发生碰撞的概率更小，吞吐量和效率更高
- CSMA 协议 / Carrier Sense Multiple Access
  - CS 载波侦听：每一个站在发送数据前先检测是否有其他计算机在发送数据
  - MA 多点接入：多个计算机接入同一根总线
  - 坚持 CSMA / persistent CSMA：
    - 空闲则直接传输，无需等待；忙则一直监听，一空闲马上传输
  - 非坚持 CSMA / non-persistent CSMA：
    - 空闲则直接传输，无需等待；忙则等待一个随机的时间之后再监听
  - p-坚持 CSMA / p-persistent CSMA：
    - 空闲则以  $p$  概率直接传输，概率  $1-p$  等到下一个时间槽开始的时候传输；
    - 忙则等待一个随机的时间之后再监听
    - 既能减少冲突，又可以减少空闲等待时间
- CSMA/CD / Carrier Sense Multiple Access / Collision Detection
  - 传播时延对载波侦听的影响：当出现数据碰撞的时候停止发送
  - 最迟检测碰撞的时间： $2t$ ，其中  $t$  为单程传播时延 ( $2t = \text{碰撞窗口}/\text{冲突窗口}/\text{争用期}$ )
  - 确认重传时机的细节：

- 确认退避时间  $(2t) > \text{定义参数 } k = \min\{\text{重传次数}, 10\}$
- 从离散的整数集合  $\{0, 1, 2, \dots, 2^k - 1\}$  选择一个数  $r$ , 退避时间即为  $r$  倍的基本退避时间
- 当重传了 16 次仍然不成功, 说明网络太拥挤, 抛弃。
- 最小帧长问题:
  - 为了能使协议检测到碰撞的时候数据还没有传输完成
  - 帧的传输时延至少要两倍于信号在总线中的传输时延
    - **最小帧长 = 总线传播时延 \* 数据传输速率 \* 2 =  $2t * \text{数据传输速率}$**
    - 以太网规定的最短帧长为 64B, 长度更小的会被认定为冲突帧。所以在发送前需要进行字节填充。
- CSMA/CA / Carrier Sense Multiple Access / Collision Avoidance
  - 主要用于无线局域网, 无法做到全面检测, 解决**隐蔽站**问题
    - 让周围的所有节点都能够知道某个信道要发送数据
  - 协议细节
    - 发送数据前, 先检测信道是否空闲。发送 **RTS** 帧 (Request to send), 等待接收端发送 **CTS** (Clear to send)
    - 发送端 **CTS** 后会预约信道, 告诉其他信道自己要传送多少数据
    - 接收端收到数据后用 CRC 检验数据是否正确, 正确则响应 **ACK**
    - 发送方收到 **ACK**之后可以进行下一个帧的发送 (二进制退避算法计算推迟时间)
- 和 CSMA/CD 相比:
  - CSMA/CD 用于以太网, CSMA/CA 用于无线局域网
  - 载波检测不同

## 具体应用

### 局域网

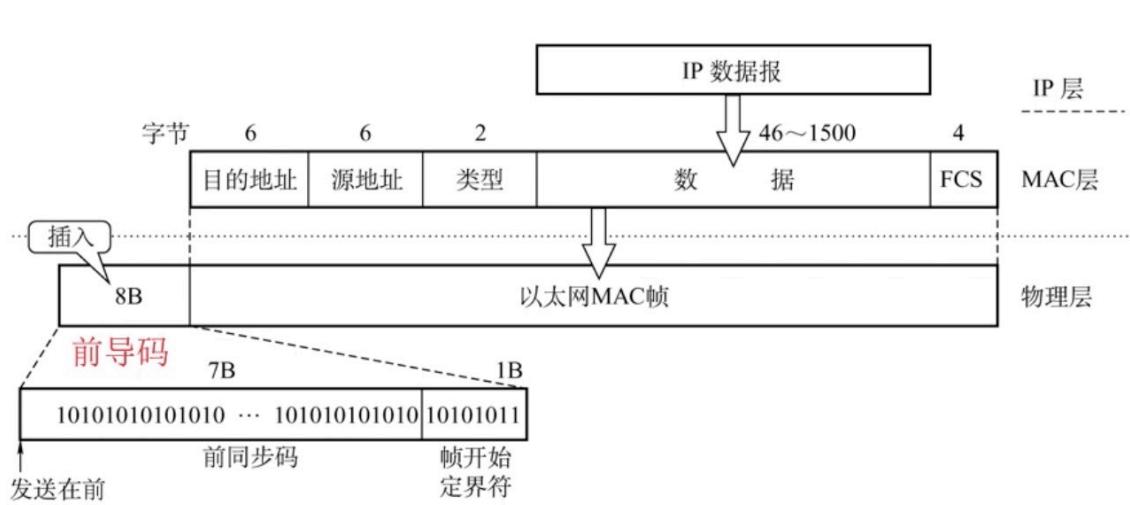
- LAN Local Area Network
- 特点: 覆盖地理范围小, 使用专门的传输介质, 通信延迟时间短, 各站为平等关系, 共享传输信道, 各站平等关系, 广播信道
- 拓扑结构:



- 局域网介质访问控制的方法
  - 使用 CSMA/CD：适用于**总线型局域网**
  - 令牌总线：适用与**总线型局域网**，将总线型网络中的各个工作站按照顺序排列好发送
  - 令牌环：适用于**环型局域网**
- 局域网的分类：
  - 以太网、令牌环网、FDDI、ATM、WLAN
- 使用的标准：IEEE 802 标准，具有一系列的标准
  - IEEE 802.3 以太网
  - IEEE 802.5 令牌环网
  - IEEE 802.8 光纤技术咨询组，提供有关光纤联网的技术咨询
  - **IEEE 802.11 WLAN**
- **MAC 子层和 LLC 子层**：LLC负责识别网络协议，MAC子层负责数据帧的封装

## ★ 以太网

- Ethernet，使用**CSMA/CD** 技术，在各种技术中占统治性地位
  - 网络速率快，造价低廉
  - 提供**无连接、不可靠**的传输服务（尽最大努力交付，只实现无差错接受）
- 拓补结构：逻辑上总线型，物理上星型（由于集线器的引入）
- 适配器与 MAC 地址：
  - 计算机与外界局域网的连接通过**通信适配器**实现，每个适配器含有**全球唯一的 MAC 地址**
- **以太网 MAC 帧**：(V2格式)
  -



- 数据段含义：
  - 目的地址：如果是全 f 则为广播地址
  - 类型：数据的协议类型
  - 由于至少需要 64 字节，数据部分的长度在 46~1500 字节之间
  - FCS：循环冗余校验码

## 无线局域网

- 802.11 的 MAC 帧头格式：

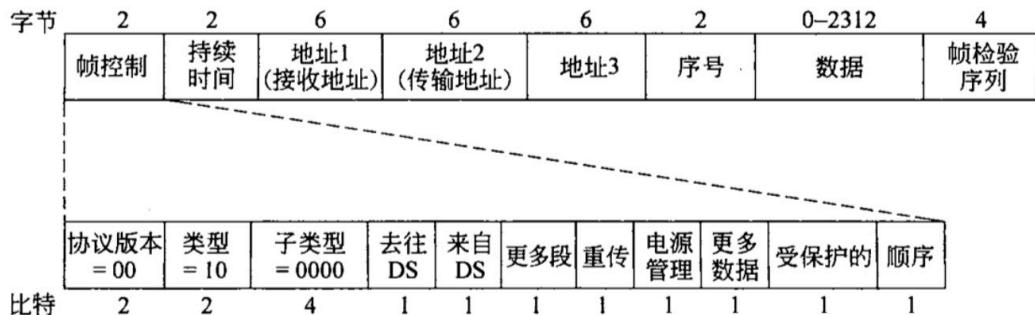


图 4-29 802.11 数据帧的格式

- 基础服务集 BSS, 拓展服务集 ESS (有限和无限服务集)
- 服务器标识符: WIFI名称

## 链路层设备

- 网桥: 根据 MAC 地址转发, 确定将帧转发到哪一个端口
  - 透明网桥: 不知道经过了哪个网桥, 自学习
  - 源路由网桥: 确定最佳路由放到帧的首部
- 交换机 (多接口网桥)
  - 直通式交换机 (可靠性低) / 存储转发式交换机 (可以检查正确)

## Chapter 5 Network Layer

- 主要功能
  - 路由转发与分组转发、异构网络互连、拥塞控制
    - 拥塞控制: 开环控制 (静态) & 闭环控制 (动态)

## 数据报文交换

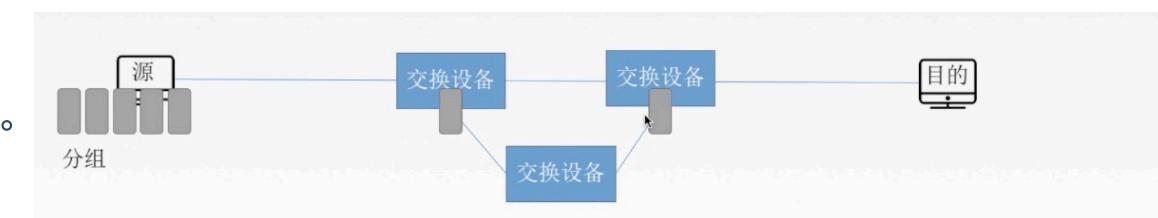
- ### 电路交换
- 在打电话的时候使用的交换方式
  - 建立连接 -- 通信 -- 释放连接
  - 特点: 使用多路复用划分电路, 支持所有人能够同时使用资源
    - 注意电路交换中每一条线路是独占的形式
    - 优点: 数据直接传输, 会按照顺序发送, 有序传输, 且不会发生冲突, 具有很强的实时性
  - 缺点: 建立时间差、线路使用效率低、灵活性差、没有差错控制能力

### 报文交换

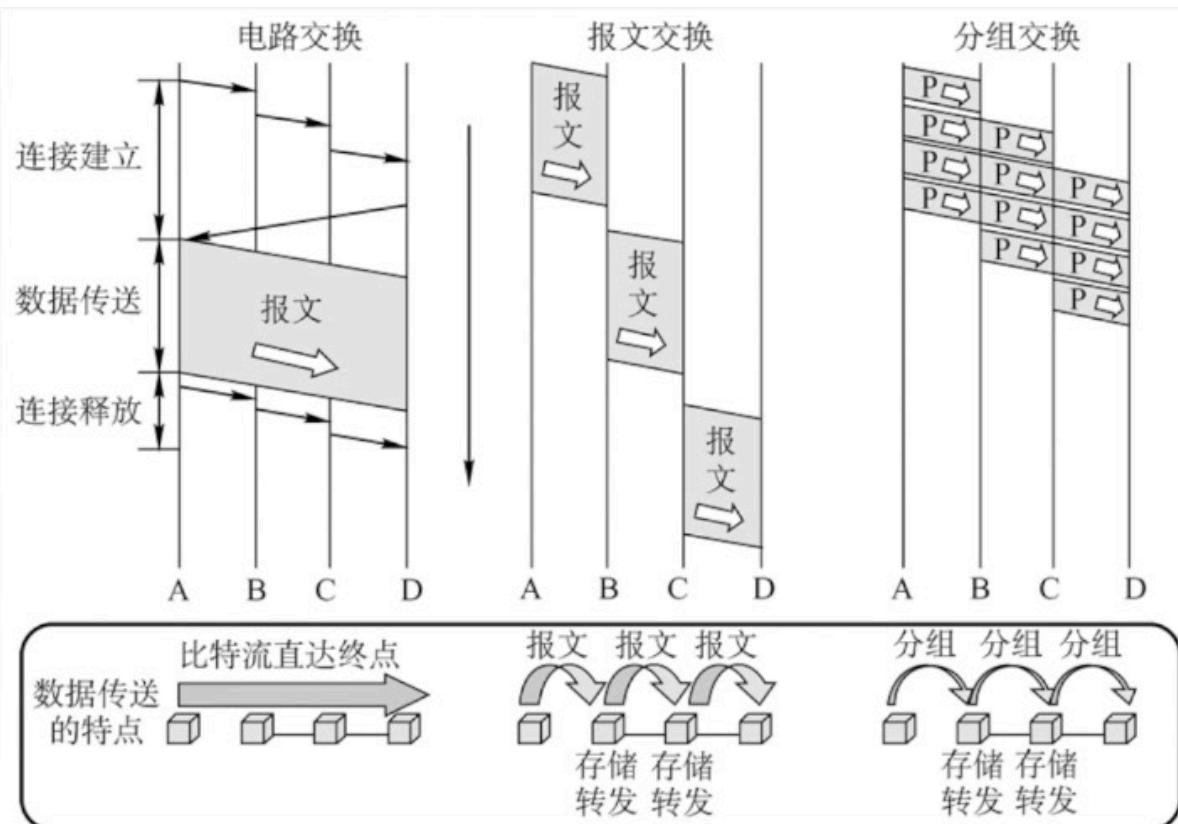
- 报文发送到交换设备中 (通常为交换机), 经过交换机之间的沟通交付给目的主机
- 有点: 无需提前建立连接, 动态分配线路, 线路可靠性高, 利用率高, 多目标服务
- 缺点: 有存储转发时延、报文大小不定

### 分组交换

- 分组：把大的数据块分割为小的数据块
- 源主机报文切割为等大的分隔片段，依次发送到交换设备中



- 相对于报文交换，存储管理更容易，线路的可靠性也有所提高（报文变短）
- 缺点：乱序到达，需要增大排序开销
- 上述三种方式的比较：



## 数据报（因特网使用）

- 无连接服务
- 不事先为分组的传输确定路径，每个分组独立确定传输路径，不用的路由器传输路径不同
- 每个分组携带源地址和目的地址
  - 基于路由协议/算法建立转发表，检索转发表并选择线路

## 虚电路方式

- 连接服务
- 结合数据包和电路交换方式结合
- 协议细节：
  - 记录了一条源主机到目的主机的类似于电路的路径（逻辑连接），路径上的所有节点都要维持这条虚电路的建立，维持一张虚电路表

- 数据分组传送，全双工通信（以分组的方式转发）

- 和数据报的对比：

	数据报服务	虚电路服务
连接的建立	不要	必须有
目的地址	每个分组都有完整的目的地址	仅在建立连接阶段使用，之后每个分组使用长度较短的虚电路号
路由选择	每个分组独立地进行 路由选择和转发	属于同一条虚电路的分组按照同一路由转发
分组顺序	不保证分组的有序到达	保证分组的有序到达
可靠性	不保证可靠通信，可靠性由用户主机来保证	可靠性由网络保证
对网络故障的适应性	出故障的结点丢失分组，其他分组路径选择发生变化，可正常传输	所有经过故障结点的虚电路均不能正常工作
差错处理和流量控制	由用户主机进行流量控制，不保证数据报的可靠性	可由分组交换网负责，也可由用户主机负责

## ★ 路由算法

- 分类
  - 静态路由算法：管理员手动配置路由信息
  - 动态路由算法：彼此交换信息，按照路由算法算出表项
- 分层次的路由算法：使用自治系统处理（AS），由内部行政单位来管辖
  - 每个单位内部使用网关协议（如RIP、OSPF）
  - 外部使用网关协议（BGP-4）进行交换

### RIP 距离矢量路由

- 和相邻路由器交换信息（直接交换路由表）
  - 每隔 30s 都会刷新一次
- 一个示例：

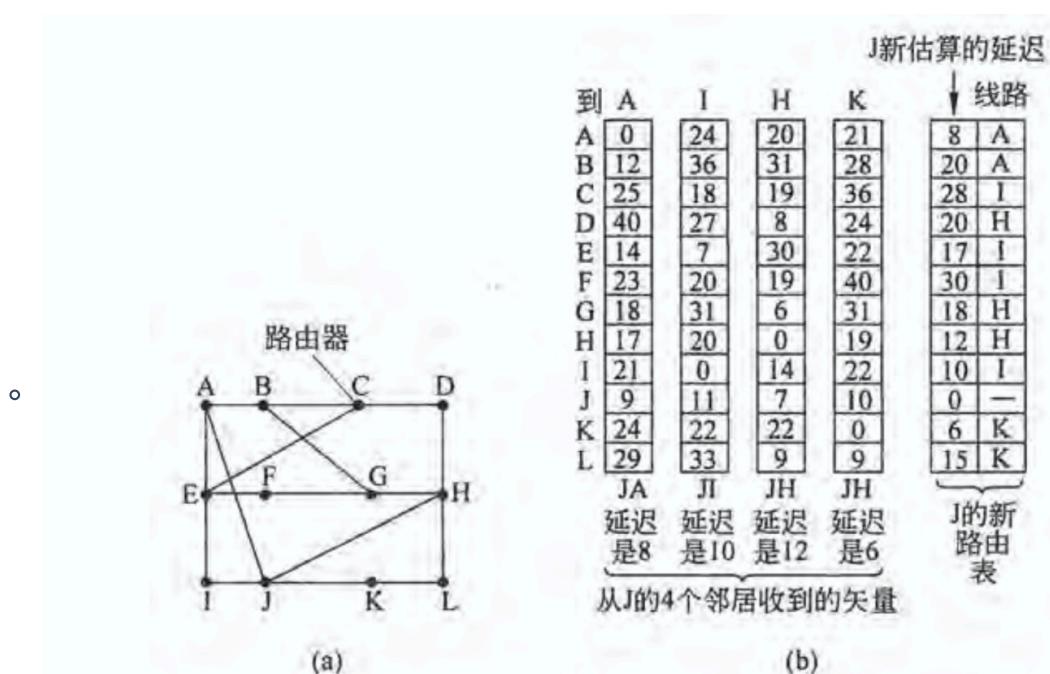


图 5-9  
(a) 一个网络示例；(b) 来自 A、I、H、K 的输入，以及 J 的新路由表

## 链路状态路由

- OSPF: 开放最短路径优先协议
  - 每隔 30min 刷新一次链路状态
  - 公开发表, 不受某一个厂商控制
  - 和所有路由器发送信息, 只发送临近的路由器的信息 (对比 RIP, RIP 发送来自所有路由器的信息)
  - 使用广播的方式, 和周围的路由器发送路由器的状态 (和相邻路由器交换链路状态)
  - 只有当链路状态发生变化的时候泛洪发送信息
- 路由算法细节
  - 发现他的邻居节点并了解邻居节点的网络地址
  - 设置到每个邻居的成本 metric
  - 构造数据库描述分组, 向邻站发送
  - 如果都有则不作处理, 如果没有或者更新的则发送 LSR 请求更新信息
  - 收到 LSR 分组后发送 LSU 进行更新, 并返回 LSAck 链路状态确认分组进行确认

## 逆向路径转发

- 所有路由器都能建立一个链路状态路由表

## 生成树算法

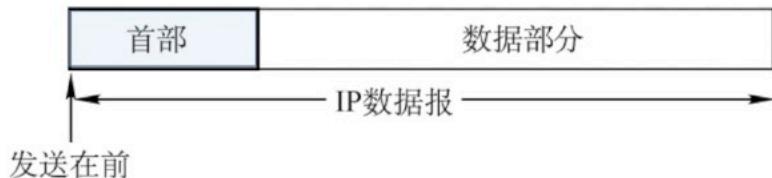
- 使用 sink tree 汇集树广播路由

## 边界网关 BGP

- 支持 CIDR, 路由表包括网络前缀
- 用于和其他自治系统 AS 的邻站 BGP 发言人交换信息
  - 交换网络可达性的信息, 即要到达某个网络需要经过的一系列 AS (用于层次路由)

## IP / Internet Protocol

### 报文格式



- 报文中的单位： 总长度单位 1B， 片偏移长度 8B， 首部长度单位 4B
- 总长度：首部 + 数据 (单位为 1B)
- 可选字段的范围：0-40字节，用来支持未知特性
- 填充：补 0，保证首部长度为 4 的整数倍
- 分组相关：
  - MTU 最大传送单元 (以太网的 MTU 为 1500 字节)
  - 每一个片都使用同一个标识
  - 标志位： DF Don't Fragment; MF More Fragment
  - 片偏移：指出片在原分组中的相对位置，以 8 字节为单位 (除了最后一个分片，每个分片都是 8 字节的整数)
    - 该值为每一个数据分片下限和元数据报 0 字节位的偏移
- Eg.



## IPv4 地址

- 给每一个主机一个标识符，=网络号，主机号

- 分类的 IP 地址:

- (现在已几乎不使用这种方式) 主要使用子网划分和 CIDR

- 



- 一些特殊 IP 地址:

- 全为 0, 表示默认路由
- 全为 1: 本网广播地址
- 网络号为特定值, 主机号全 0, 表示一个网络
- 网络号为特定值, 主机号全 0, 表示一个直接广播地址, 可以对特定主机进行广播
- **主机号为 127: 本地软件环回测试, 环回地址**

- 私有 IP 地址:

地址类别	地址范围	网段个数
A类	10.0.0.0~10.255.255.255	1
B类	172.16.0.0~172.31.255.255	16
C类	192.168.0.0~192.168.255.255	256

- 各类地址可以划分的网络个数:

网络类别	最大可用网络数	第一个可用的网络号	最后一个可用的网络号	每个网络中的最大主机数
A	$2^7-2$	1	126	$2^{24}-2$
B	$2^{14}-1$	128.1	191.255	$2^{16}-2$
C	$2^{21}-1$	192.0.1	223.255.255	$2^8-2$

- 减 2: 不能为全 0 (本机地址), 不能为全 1 (广播地址)

- 网络地址转换 NAT

- 在专用网和因特网上安装 NAT 软件, 路由器至少应具有一个外部全球 IP 地址
- 同一个专用网中的所有主机通过 NAT 路由器进行地址转换, 转换成外部 IP 地址
- 路由器存储有 NAT 转换表 (WAN 到 LAN)

- 子网划分和子网掩码

- 分类地址对 IP 地址浪费过多

- 原理：将主机号分为子网号和主机号（子网号和主机号不能全 0），使用三级地址
- 子网掩码和源地址作与运算为子网
- 无分类编址 CIDR
  - 同一个网络下可以使用不同的子网掩码
- 原理：使用二级地址：使用网络前缀+主机号
  - 记法：IP 地址后加上 /
- 融合了子网地址与子网掩码，方便子网划分
  - 构成超网：如果前缀相同，则可以将前缀缩短减少路由表的冗余
    - 在路由表中使用最长前缀匹配，即有多个网络号符合要求则匹配更精确的一个
    - 在 CIDR 技术中，子网号和前缀号都可以全 1 和全 0

## 其他协议

- ARP (Address Resolution Protocol) 地址解析协议，使用 mac 地址查找 IP 地址
- DHCP 动态主机配置协议：获取动态配置地址，应用层
- ICMP (Internet Control Message Protocol) Internet 控制报文协议，在 IP 主机，路由器之间传递控制消息

## 网络层设备

- 路由器：转发，分组，具有多个输入口多个输出口的专用计算机
  - 包含有路由表与转发表，转发表由路由表得来，可以用软件实现
- 和其他层次的区别：
  - 路由器（可以互联两个不同网络层协议的网段），网桥（可以互联两个物理层和链路层的不同网段），集线器（不能互联两个物层的不同网段）

# Chapter 6 Transport Layer

---

## UDP 协议

- User Datagram Protocol
- 不建立连接
- 减少了开销和时延
- 使用最大努力交付，不保证可靠交付（由应用层保证可靠）
- 面型报文的，适合一次性传输少量数据的网络应用
  - 应用层给了多长的报文就发送多长的报文
  - 首部开销少，仅 8 字节
- 简单的协议头：



- 伪首部：仅用于计算校验和



- 校验和的计算：加上伪首部，全0填充校验和字段，全0填充数据部分，求和，把反码填入校验和字段
- 校验和的验证：填上伪首部，伪首部+首部+数据采用二进制反码求和，如果结果全为1则无差错，否则出错

## TCP 协议特点

- Transmission Control Protocol
- 面向连接（点对点，虚连接）
- 每个TCP是只能点对点的（不能用于多端口）
- 提供可靠的交付服务，无差错，不丢失
- 提供全双工通信（可以同时发送和接受数据）
  - 具有发送缓存和接受缓存（不可以过早删除因为可能会被删除）
- 面向字节流 - 会把数据看做字节流

## TCP 报文格式



- TCP首部必须为 4字节的整数倍 (填充位的作用)
- 序号字段：第一个字节使用的编号是多少 (TCP会把每个字节都编号)
- 确认号：期望收到对方下一个报文的字节数
- 加了长度字段之后会有一个字段固定首部多长
- 窗口：反映了发送方自己可以容纳的最多字节流
  - 体现了自己能容纳的最大大小
- 数据偏移：TCP报文段的数据距离起始部分有多远， (会  $\times 4$ )
  - 用于规定首部有多长
- 校验和：检验 首部+TCP数据，对于头部使用 ip伪头部 + TCP头
  - 把伪首部、TCP报头、TCP数据分为16位的字，如果总长度为奇数个字节，则在最后增添一个位都为0的字节。
  - 把TCP报头中的校验和字段置为0。
  - 用反码相加法累加所有的16位字 (进位也要累加)。
  - 对计算结果取反，作为TCP的校验和。
  - 伪首部共有12字节，包含如下信息：源IP地址、目的IP地址、保留字节(置0)、传输层协议号(TCP是6)、TCP报文长度(报头+数据)。 (和 UDP 相同)
- 紧急指针：之处紧急数据的字节数
- 6个控制位
  - URG 紧急位：高优先级 (移到发送方缓存的最前方)
  - ACK 确认为：在连接建立后所有位设置为 1
  - PSH 推送为：接收方应该及时交付给上层 (及时从接收方的缓存移走)
  - RST 复位：表明出现严重差错，必须释放连接
  - SYN 同步位：表示是一个请求连接或接受连接的报文
  - 终止位 FIN：表明报文发送完毕，需要终止

- 选项：MSS、SACK、扩大等

## ★ TCP 建立连接

- 三次握手建立连接

◦

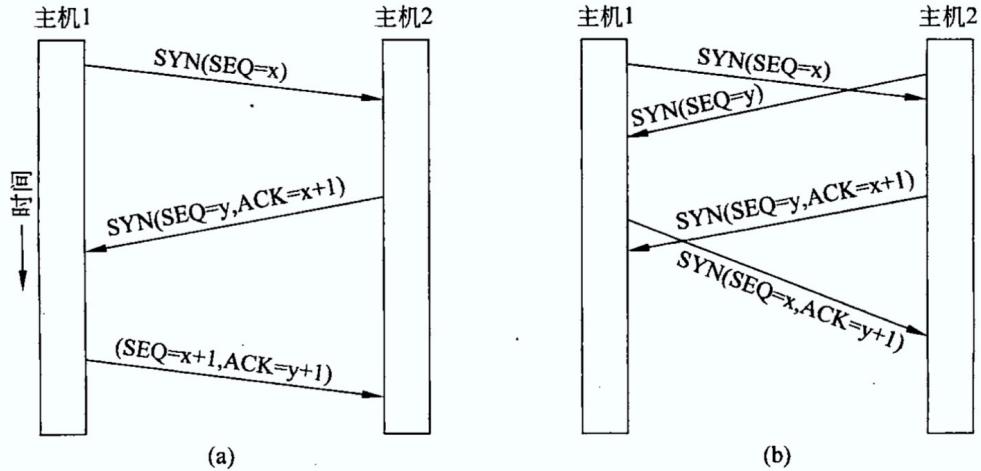


图 6-37

(a) 正常情况下的 TCP 连接建立；(b) 两端同时建立连接的情形

- $\text{SYN} = 1, \text{seq} = x$  (报文段是序号  $x$ )
- $\text{SYN} = 1, \text{ACK} = 1, \text{seq} = y, \text{ack} = x + 1$  (服务器收到的确认就是客户端的  $x + 1$ )
  - 服务器分配缓存和变量，并向客户端返回确认报文
- $\text{SYN} = 0, \text{ACK} = 1, \text{seq} = x + 1, \text{ack} = y + 1$ 
  - 客户端分配缓存和变量
- SYN 洪泛攻击

黑客发送大量请求字段，使客户端无法完成连接的建立(Solution. SYN cookie)

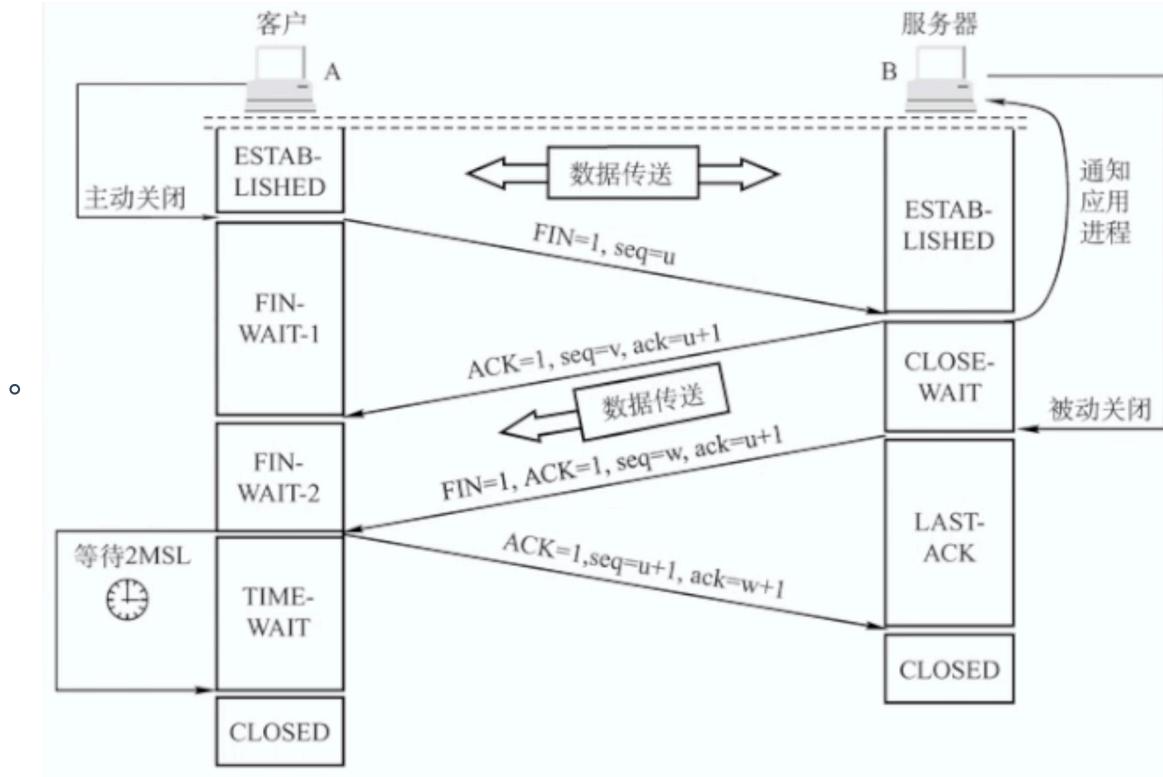
- 服务原语

状态	描述
CLOSED	没有活跃的连接或者挂起
LISTEN	服务器等待入境呼叫
SYN RCVD	到达一个连接请求；等待ACK
SYN SENT	应用已经启动了打开一个连接
ESTABLISHED	正常的数据传送状态
FIN WAIT1	应用没有数据要发了
FIN WAIT2	另一端同意释放连接
TIME WAIT	等待所有数据包寿终正寝
CLOSING	两端同时试图关闭连接
CLOSE WAIT	另一端已经发起关闭连接
LAST ACK	等待所有数据包寿终正寝

图 6-38 TCP 连接管理有限状态机使用的状态

## TCP 连接释放

- 释放连接的两种方式
  - 对称释放：两方作为单独的连接控制断开与连接（TCP使用的方式）
  - 非对称释放：一旦一方断开连接两侧都断开
- 四次挥手



- 客户端： $FIN = 1, seq = u$
- 服务器： $ACK = 1, seq = v, ack = u + 1$  （服务器端关闭，已经处于 CLOSE-WAIT 状态）
- 服务器（发完数据）： $FIN = 1, ACK = 1, seq = w, ack = u + 1$  （这段时间客户端不可能发送任何数据所以确认位相同）
- 客户端：回复一个确认报文段，再等待计时器设置的 2MSL 后彻底关闭连接
  - MSL: Maximum Survival Time 数据包的最长生存时间

## TCP 差错控制

- 确认机制：发送方把缓存发送给客户端之后，等待接收方的确认
  - 在收到接收方的确认之后从缓存中删除
  - 接收方使用累计确认的方式把已经收到的数据发送回发送方
  - 之后接收方可以从缓存中删除
  - 如果少了位，接收方 ack 将不会前进
- 重传机制：发送条件复杂，不可简单设置超时重传（会增大网络负荷）
  - RTTs: Round Trip Time (smooth): 加权的平均往返
    - 超时重传（可能会等待时间过长）
  - 冗余 ACK: 每当比期望小大的时候发送一个冗余 ACK，表明期待的下一个字节的序号
    - 比如连续收到多个未来的包，会发送多个 ACK，而接收方确认了多个冗余的 ACK 之后，则认定该包丢失，会重传（快重传）

## TCP 流量控制

- 用于解决点到点的问题（区别于拥塞控制）
- 滑动窗口机制：动态调整接受缓存机制 - Sliding Window
  - 发送窗口取决于接收窗口 rwnd 和拥塞窗口 cwnd 的最小值
  - 发送方的发送窗口可以变化，根据接收方发送回来的报文段改变
  - 发送也可以指定每一个报文的大小
  - 当窗口满的时候发送方不再发送新的消息，等待超时重传
  -

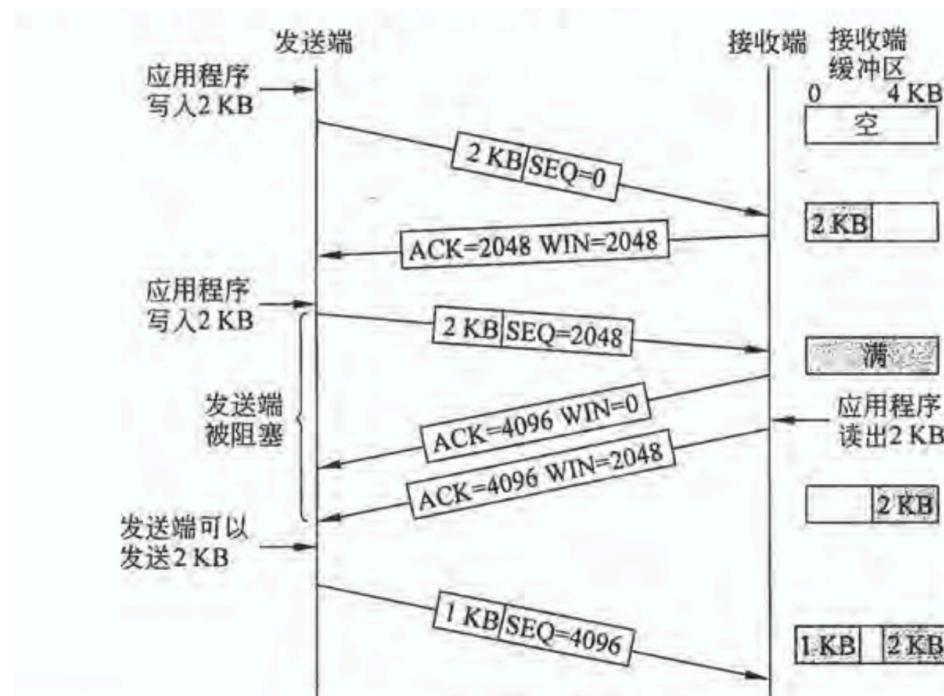


图 6-40 TCP 的窗口管理

- 死锁解决（例如客户端发送回来的发送包因为传输原因消失
  - TCP设置了计时器，如果超时会发送探测报文段
  - 如果窗口仍然为0，则发送方重新设置持续计时器（超时后重新发送报文段）

## TCP 拥塞控制

- 解决问题：网络中的资源不够用（解决客户端之间的问题）
- XCP - eXplicit Congestion Protocol 显示拥塞协议
- ECN - Explicit Congestion Notification 显示拥塞通知
  - TCP 协议使用，使用比特位告知发送者是否拥塞但不控制具体拥塞的成都和量
- AIMD - Additive Increase Multiplicative Decrease 加法递增乘法递减法则
  - 达到有效和公平（收敛原则）

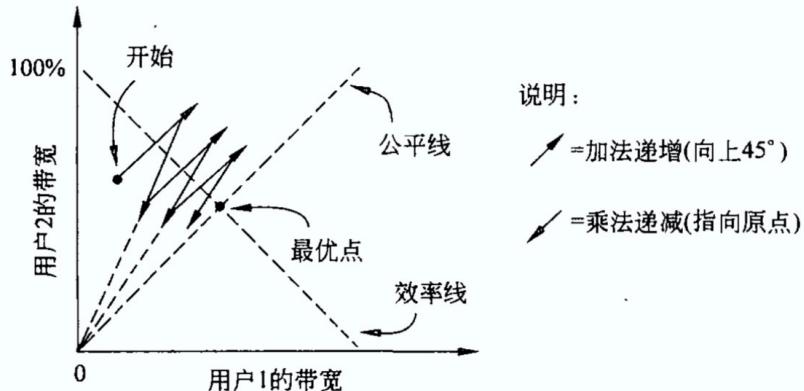
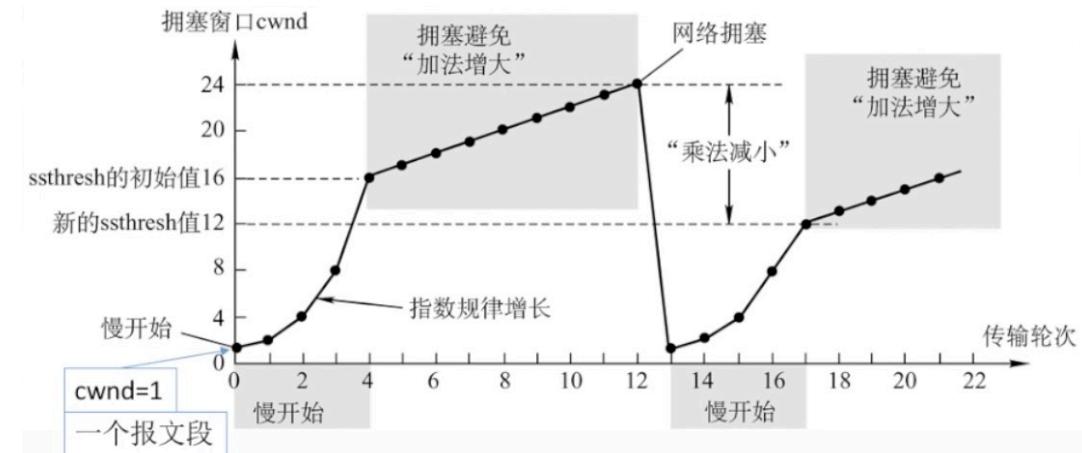


图 6-25 加法递增乘法递减 (AIMD) 控制法则

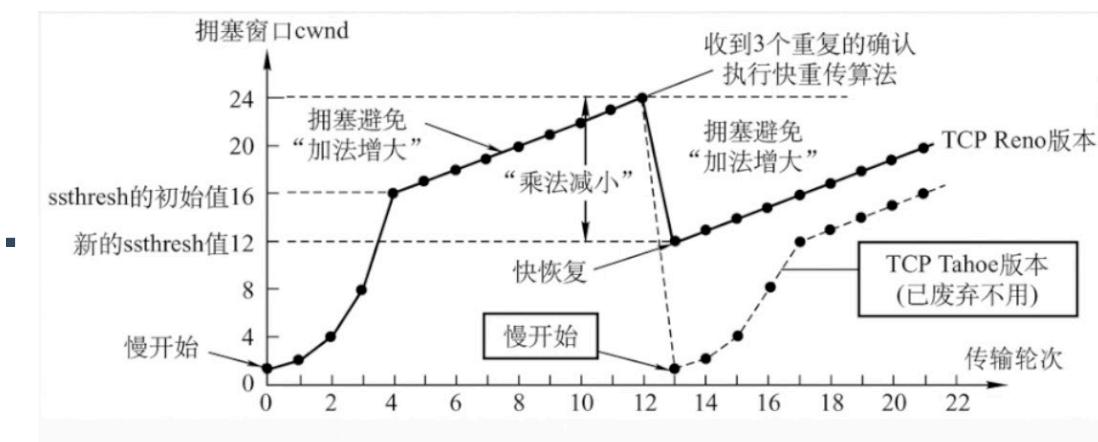
- 四种算法

- 慢开始和拥塞避免：



- 一个传输轮次：发送了一批报文段并收到确认的时间 - RTT
- ssthresh：慢开始阈值，之后则线性增长 （新的 ssthresh 值应当为出现拥塞的大小的  $1/2$ ）
- 发现拥塞之后瞬间缩小到慢开始的情况
- 增倍时间：一旦收到消息确认，立即翻倍

- 快重传和快恢复：



- 收到冗余 ACK 的时候执行快重传算法，在重传之后 快恢复
  - 恢复不用降低到1，只用降低到新的 ssthresh 值，线性加法增大。
- TCP Reno 已被弃用