计网复习

第一章 概述

- 1.1 互联网发展情况
- 1.2 互联网发展史
  - 1.2.1 互联网起源
  - 1.2.2 Internet 的形成
  - 1.2.3 互联网普及
- 1.3 基本概念解析
  - 1.3.1 网络
  - 1.3.2 拓扑 (Topology)
  - 1.3.3 协议
  - 1.3.4 数字带宽 (Bandwidth)
  - 1.3.5 吞吐量
  - 1.3.6 点到点与端到端
  - 1.3.7 计算机网络的分类
  - 1.3.8 课程总结
- 1.4 参考模型
  - 1.4.1 参考模型的必要性
  - 1.4.2 OSI 七层模型
  - 1.4.3 TCP/IP 四层参考模型
  - 1.4.4 OSI与TCP/IP参考模型比较
  - 1.4.5 课程总结
- 1.5 参考模型中相关概念解析
  - 1.5.1 发送方与接收方之间的通信
  - 1.5.2 对等实体与虚拟通信
  - 1.5.3 课程总结

第二章 物理层

- 2.1 物理层概述
  - 2.1.1 物理层功能、特性以及物理带宽
  - 2.1.2 奈奎斯特定理
  - 2.1.3 香农定理
  - 2.1.4 课程总结
- 2.2 引导性传输介质
  - 2.2.1 传输介质分类
  - 2.2.2 同轴电缆 (铜线)
  - 2.2.3 双绞线 (铜线)
  - 2.2.4 电力线 (铜线)
  - 2.2.5 光纤
  - 2.2.6 如何选择传输介质
  - 2.2.7 课程总结
- 2.3 复用技术
  - 2.3.1 复用技术概念
  - 2.3.2 频分多路复用 (FDM)
  - 2.3.3 波分多路复用 (WDM)
  - 2.3.4 时分多路复用 (TDM)
  - 2.3.5 码分多路复用 (CDMA)
  - 2.3.6 课程总结

2.4 调制技术
2.4.1 基带传输
2.4.2 通带传输
2.4.3 课程总结
2.5 公共交换电话网络 (PSTN)
2.5.1 PSTN 概述
2.5.2 本地回路
2.5.3 干线
2.5.4 交换局
2.5.5 课程总结
2.6 物理层设备
2.6.1 物理层部件 / 设备分类
2.6.2 收发器 (Transceiver)
2.6.3 中继器 (Reapter)
2.6.4 集线器 (Hub)
2.6.5 课程总结
第三章 数据链路层
3.1 数据链路层概述
3.1.1 数据链路层功能与帧
3.1.2 字符计数法 (不用)
3.1.3 字节填充的标志字节法 (常用)
3.1.4 比特填充的标志比特法 (常用)
3.1.5 物理层编码违例法
3.1.6 课程总结
3.2 差错处理概述
3.2.1 差错处理概述
3.2.2 海明距离
3.2.3 课程总结
3.3 纠 1 位错的海明码
3.3.1 数据位与冗余位
3.3.2 海明纠错码
3.3.3 课程总结
3.4 检错码
3.4.1 奇偶校验码
3.4.2 互联网校验和
3.4.3 循环冗余校验码 (CRC)
3.4.4 课程总结
3.5 基本数据链路协议 (1-3)
3.5.1 前提假设
3.5.2 无限制的单工协议 (协议 1)
3.5.3 单工停-等协议 (协议 2)
3.5.4 有噪声信道的单工协议 (协议 3)
3.5.5 课程总结
3.6 滑动窗口协议 (协议4)
3.6.1 滑动窗口基本概念
3.6.2 窗口大小为1的滑动窗口协议
3.6.3 协议 4 的信道利用率
3.6.4 课程总结
3.7 回退n帧与选择性重传(协议5、6)

3.7.1 批量发送面临的问题
3.7.2 回退n帧 (协议5)
3.7.3 选择性重传 (协议6)
3.7.4 三个滑动窗口协议窗口大小
3.7.5 课程总结
第四章 介质访问控制子层
4.1 MAC子层概述
4.1.1 数据链路层的组成
4.1.2 数据通信的三种方式
4.1.3 介质多路访问控制 / 介质访问控制MAC
4.1.4 课程总结
4.2 ALOHA 协议
4.2.1 纯ALOHA协议
4.2.2 分隙(分槽)ALOHA协议
4.2.3 课程总结
4.3 CSMA 协议
4.3.1 CSMA 协议概述
4.3.2 非持续式 CSMA 协议
4.3.3 持续式 CSMA 协议
4.3.4 CSMA/CD
4.3.5 课程总结
4.4 以太网概述
4.4.1 以太网基础知识
4.4.2 经典以太网
4.4.3 交换式以太网
课程总结
4.5 以太网帧格式
4.5.1 IEEE 802.3与以太网 MAC 子层协议
4.5.2 MAC地址
4.5.3 课程总结
4.6 二层交换的基本原理
4.6.1 二层交换基本概念
4.6.2 网桥
4.6.3 网桥和中继器对比
4.6.4 课程总结
4.7 生成树协议
4.7.1 冗余结构带来的问题
4.7.2 生成树协议 (STP)
4.7.3 课程总结
4.8 虚拟局域网
4.9 二层设备
4.9.1 网卡 (NIC)
4.9.2 网桥
4.9.3 交换机 (Switch)
4.9.4 课程总结
第五章 网络层 (上)
5.1 网络层引言
5.1.1 网络层概述
5.1.2 数据报网络与虚电路网络

5.1.3 课程总结
5.2 IP地址
5.2.1 IP概述
5.2.2 IP地址表示形式
5.2.3 IP地址的分类
5.2.3 课程总结
5.3 子网规划
5.3.1 子网概念
5.3.2 子网掩码
5.3.3 子网规划
5.3.4 课程总结
5.4 IP寻址
5.4.1 IP寻址与MAC寻址概念
5.4.2 IP寻址与MAC寻址的比较
5.4.3 IP寻址具体过程
5.4.4 课程总结
5.5 IP分组
5.5.1 IP分组格式
5.5.2 课程总结
5.6 IPv6的发展
5.6.1 IPv4的弊端
5.6.2 IPv6的诞生
5.6.3 课程总结
5.7 IPv6地址
5.7.1 IPv6地址表示
5.7.2 IPv6地址分类
5.7.3 IPv6子网规划
5.8 IPv6分组
5.8.1 IPv6分组格式
5.8.2 IPv6拓展头
5.8.3 课程总结
5.9 IPv6过渡技术
5.9.1 IPv6过渡时代
5.9.2 双协议栈 (Dual Stack)
5.9.3 隧道 (Tunnel)
5.9.4 地址转换技术
5.10 路由表概述
5.10.1 路由器功能
5.10.2 直连路由
5.10.3 静态路由
5.10.4 动态路由
5.10.5 课堂总结
5.11 距离矢量路由选择协议
5.11.1 DV概述
5.11.2 DV的具体过程
5.11.3 课堂总结
5.12 RIP
5.12.1 RIP概述
5.12.2 RIP主要缺陷

5.12.3 课堂总结
第五章 网络层 (下)
5.13 链路状态路由选择
5.13.1 链路状态路由概述
5.13.2 发现邻居节点
5.13.3 设置链路成本
5.13.4 链路状态分组/公告 (LSP)
5.13.5 发送LSP
5.13.6 计算新的路由路径
5.13.7 课程总结
5.14 单区域OSPF
5.14.1 OSPF概述
5.14.2 OSPF分组类型
5.14.3 OSPF运行过程
5.14.4 课程总结
5.15 无类域间路由 CIDR
5.15.1 CIDR概述
5.15.2 CIDR具体原理
5.15.3 路由聚合
5.15.4 课程总结
5.16 网络地址翻译 NAT
5.16.1 NAT概述
5.16.2 NAT 具体工作过程
5.16.3 课程总结
5.17 互联网控制协议 ICMP
5.17.1 ICMP 概述
5.17.2 Ping
5.17.3 Tracert
5.17.4 PMTU
5.17.5 课程总结
5.18 地址解析协议 ARP
5.18.1 ARP概述
5.18.2 ARP优化
5.18.3 远程主机MAC地址解析
5.18.4 ARP表
5.18.5 ARP欺骗
5.18.6 课程总结
5.19 拥塞控制
5.19.1 拥塞概述
5.19.2 拥塞控制
5.19.3 开环系统中的拥塞控制
5.19.4 警告位
5.19.5 抑制分组
5.19.6 逐跳抑制分组
5.19.7 负载丢弃/载荷脱落
5.19.8 课程总结
5.20 流量整形
5.20.1 流量整形概述
5.20.2 漏桶算法

5.20.3 令牌桶算法
5.20.4 课程总结
第六章 传输层
6.1 传输层概述
6.1.1 传输层概述
6.1.2 传输层的必要性
6.1.3 数据链路层与传输层的比较
6.1.4 传输服务原语
6.1.5 数据段
6.1.6 课程总结
6.2 用户数据报协议 UDP
6.2.1 UDP 概述
6.2.2 UDP 数据段头
6.2.3 UDP 校验和
6.2.4 课程总结
6.3 通信模型
6.4 TCP 数据段
6.4.1 TCP 概述
6.4.2 TCP 数据段格式 (TPDU)
6.4.3 课程总结
6.5 TCP 三次握手建立连接
6.5.1 TCP 连接的建立
6.5.2 DoS 攻击
6.5.3 课程总结
6.6 TCP 连接释放
6.6.1 TCP 连接释放
6.6.2 半开放连接
6.6.3 三次握手正常释放连接
6.6.4 课程总结
6.7 TCP 传输策略
6.7.1 Window Size 流量控制
6.7.2 优化发送端
6.7.3 优化接收端
6.7.4 课程总结
6.8 TCP 拥塞控制
6.8.1 TCP拥塞概述
6.8.2 慢启动算法 (Slow Start)
6.8.3 阈值
6.8.4 课程总结
6.9 TCP 定时器
6.9.1 重传定时器
6.9.2 持续定时器
6.9.3 保活定时器
6.9.4 课程总结
第七章 应用层
7.1 应用层概述
7.2 域名系统 DNS 概述
7.2.1 域名概述
7.2.2 域的资源记录

7.2.3 课程总结
7.3 DNS域名解析
7.4 电子邮件
7.4.1 电子邮件系统概述
7.4.2 MIME
7.4.3 SMTP
7.4.5 最后传递过程
7.4.6 课程总结
7.5 万维网 (World Wide Web)
7.5.1 万维网概述
7.5.2 统一资源定位符
7.5.3 客户端
7.5.4 服务端
7.5.5 课程总结
7.6 其它应用
7.6.1 文件传输 (FTP)
7.6.2 远程登录 (Telnet)
7.6.3 多媒体应用
7.6.4 课程总结

计网复习

- **P32 OSI 参考模型**

1. 什么是协议 —— 通信双方为了通讯制定的协议集合
2. 端口问题 —— MAC地址、IP地址
3. 交换单元的名称 —— 各层交换的单位、TPDU各字母代表含义
4. 中间系统 —— 交换机、路由器
5. 最底下有实体介质用于两方通讯

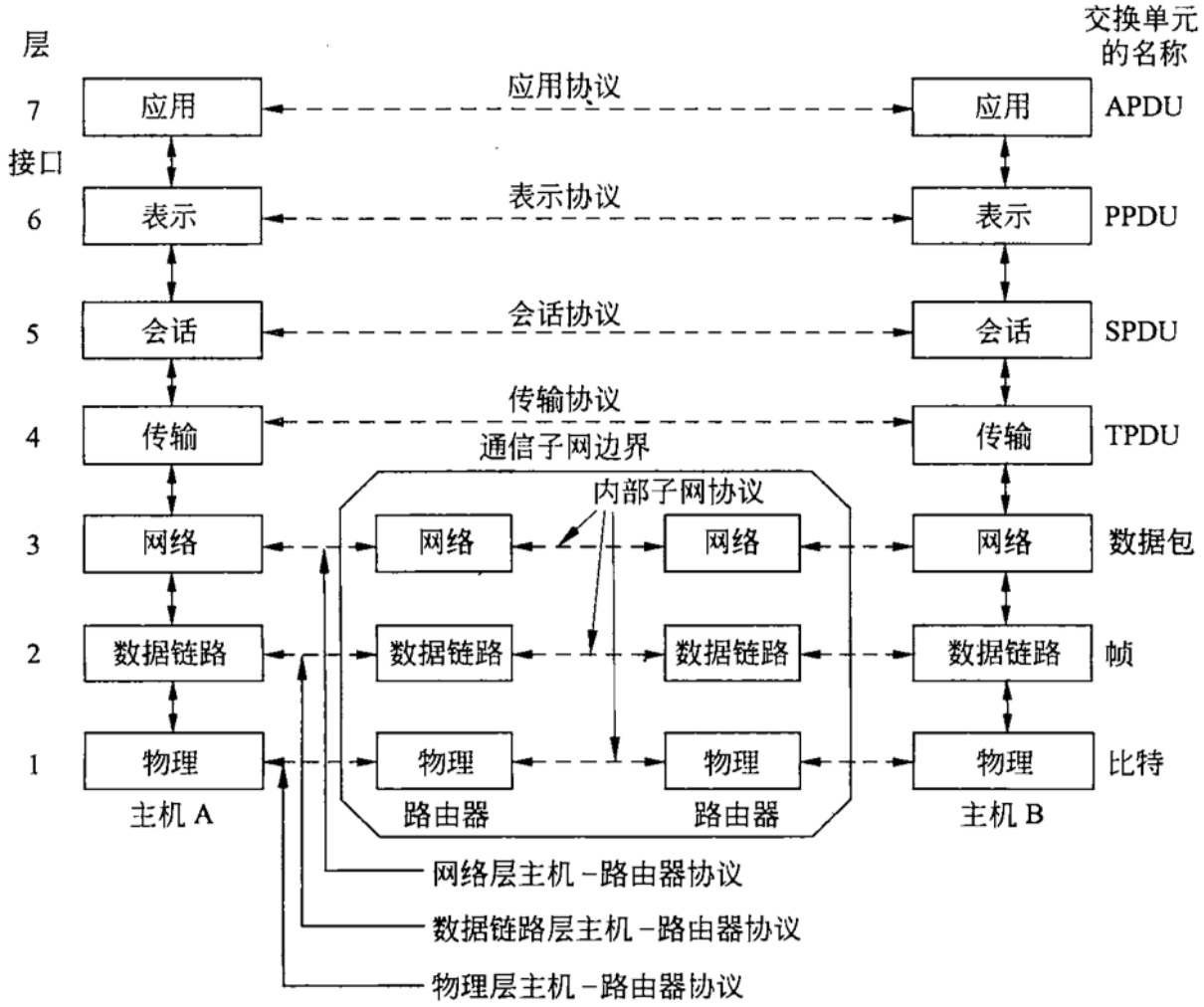


图 1-20　OSI 参考模型

- **P25 虚拟通信的信息流**

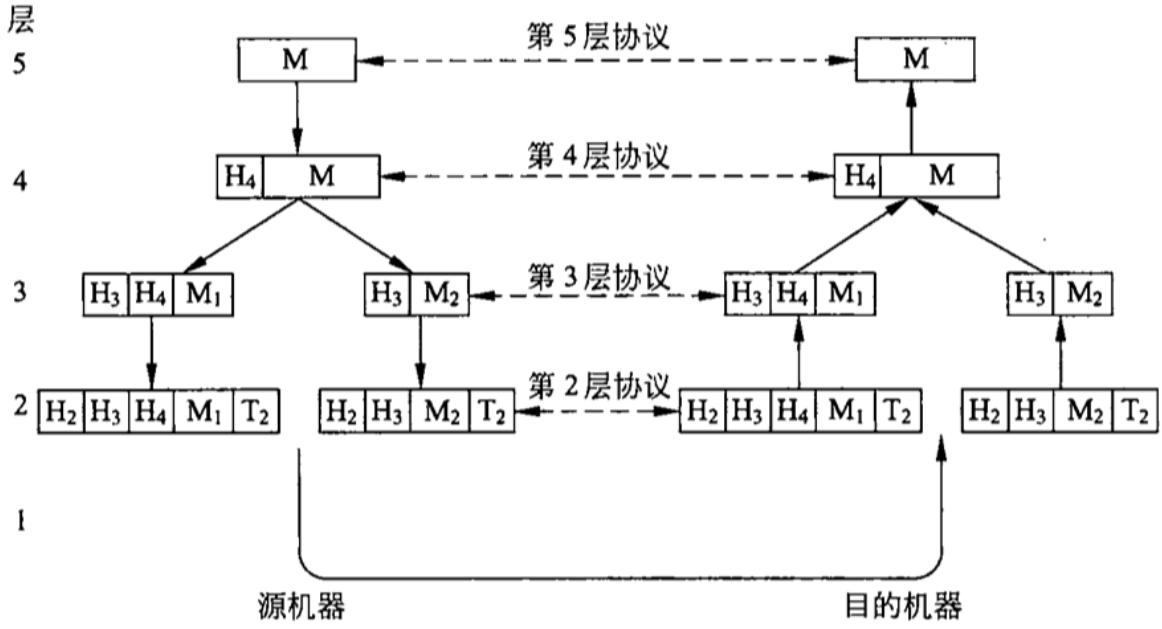


图 1-15　支持第 5 层虚拟通信的信息流

1. 在实际生活中，TCP/IP系统为什么经常把帧切分为 1460个 Byte 进行传输?

1500-20-20 = 1460，IP头 20 Byte，TCP协议头 20 Byte。

- **P37 本书所用的模型层次**

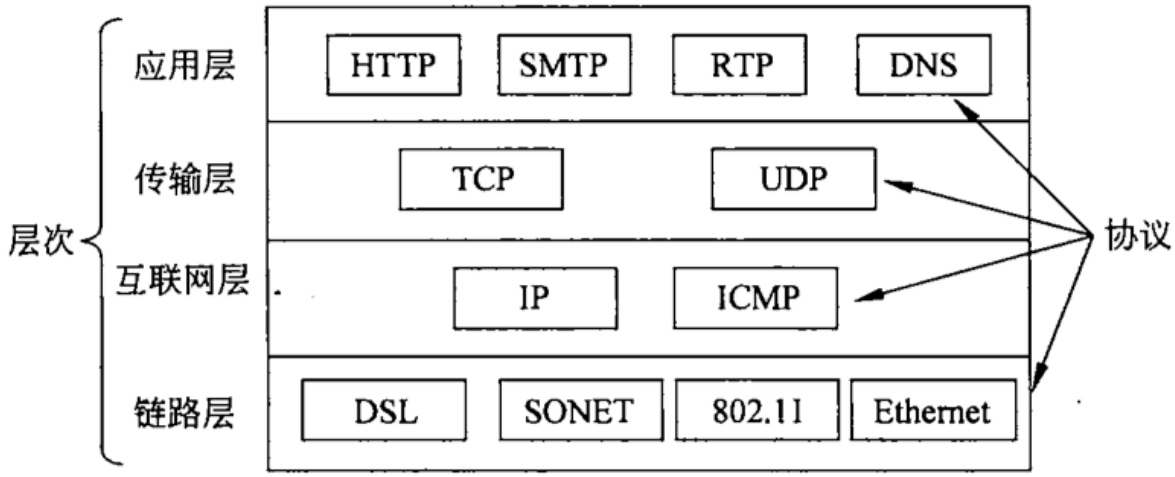


图 1-22　TCP/IP 模型及一些将要学习的协议

- **第一章**

考：第三节和第四节

- **第二章**

考：2.1 —— 尼奎斯特定理、香农定理 [知道公式]

2.2 —— 磁介质、双绞线、同轴电缆、电力线、光纤 [了解这些线]

2.5 —— 基带传输 [曼彻斯特、差分曼彻斯特]、通带传输、数字变成模拟信号传、模拟信号变成数字传、频分复用、时分复用

2.6 —— ADSL、欧洲(英国) E-One、T-one (1.544Mbps)、SONET (2-39与2-40)、波分多路复用、包交换

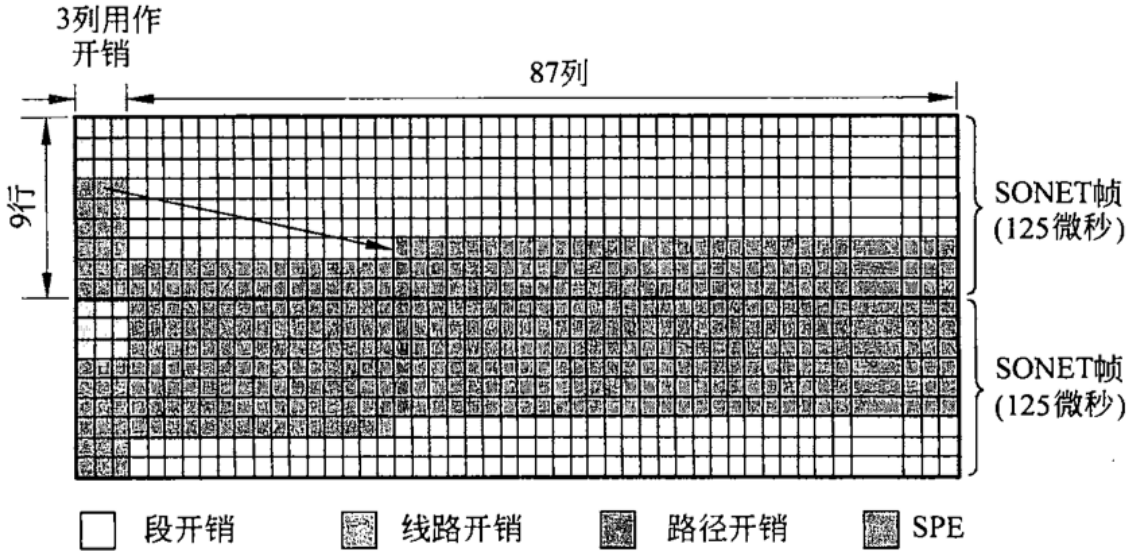


图 2-39　两个连续的 SONET 帧

SONET		SDH	数据率(Mbps)		
电子	光	光	总速率	SPE	用户速率
STS-1	OC-1		51.84	50.112	49.536
STS-3	OC-3	STM-1	155.52	150.336	148.608
STS-12	OC-12	STM-46	22.08	601.344	594.432
STS-48	OC-48	STM-16	2488.32	2405.376	2377.728
STS-192	OC-192	STM-64	9953.28	9621.504	9510.912
STS-768	OC-768	STM-256	39813.12	38486.016	38043.648

图 2-40　SONET 和 SDH 多路复用率

不考：2.3、2.4、码分复用、2.7 (掌握点概念)、2.8 (掌握点概念)

- 第三章

考：3.1-3.2 (CRC校验)

3.3-3.5 (六大协议)、Tripel-P协议

- 第四章

考：4.1-4.2(CSMA、隐藏栈、暴露栈)、4.3 (整个都很重要，冲突、二进制指数后退、交换)、4.4 (很重要，802.11)、4.8 (重要、网桥、设备)

不考：4.1-4.2 (看故事)、4.3的万兆以太网、4.5 (知道标准是什么，802.16)、4.6 (不考)、4.7 (不考)

- 第五章

考：5.2 (重要，距离矢量协议、链路状态协议)、5.5 (如何互联、隧道、包的分片分段)

CIDR、NAT (重要)、IPv6 (掌握概念)、MPLS (了解)

不考：5.1 (了解思想、面向连接/不连接的服务)、5.2 (泛洪掌握思想、层次路由、广播路由、组播路由[看懂])、选播路由、移动主机路由 (知道概念)、5.3-5.4 (知道概念)、BGP、组播、移动IP

- 第六章

考：TCP (重要，三次握手等)、拥塞控制 (重要)、P445、P446

不考：6.6、6.7、6.8

- 第七章

考：了解概念即可

- 题型

名词解释 (2分一个，协议、有一个名词在第七章中)

单项选择 (10分)

填空 (10分，每个题一分，一个空 四个字两个字或者一个公式)

计算题 (尼奎斯特定理、传输速度、IP地址划分、和数字有关的题目)

论述题 (两题)

综合题 (一题，非常综合的考察)

名词解释：

- 第一章
  - CERNET、CNGI-CERNET 2、ARPA、ARPANET、网络、计算机网络
  - 互联网络、万维网、拓扑、协议、带宽、吞吐量、点到点、端到端、个域网、局域网、城域网、广域网、互联网、OSI参考模型、TCP/IP参考模型、数据流、数据段、包、帧、比特流、PDU、实体、对等实体、虚拟通信、服务
- 第二章
  - 物理层、模拟信号、数字信号、码元、物理带宽、奈奎斯特定理、香农定理、引导性传输介质、同轴电缆、双绞线、非屏蔽双绞线、电力线、光纤、单模光纤、多模光纤

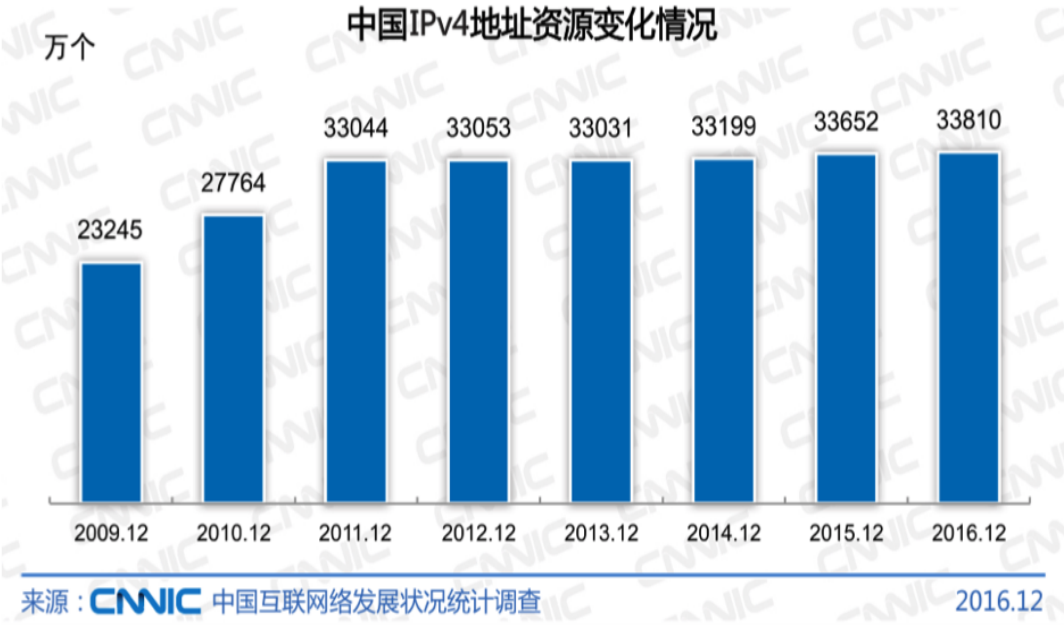
第一章 概述

1.1 互联网发展情况

- 国内网络发展现状
  - 截止到 2016.12 ，中国出口带宽总量为 6,640,291 M。
    -

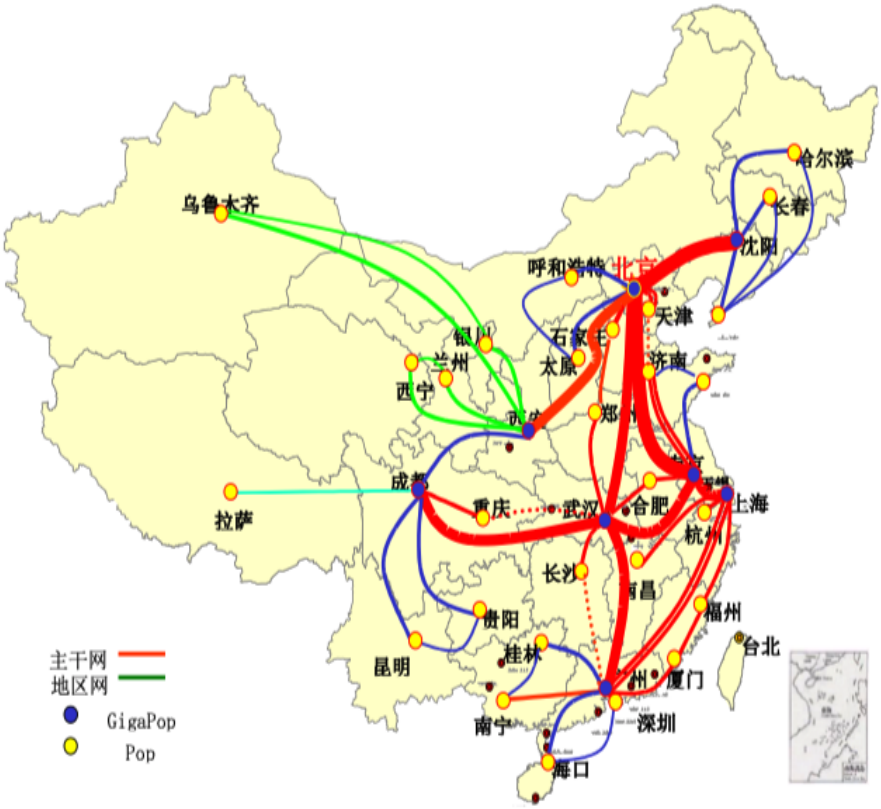
	国际出口带宽数（Mbps）
中国电信	3,886,527
中国联通	1,700,446
中国移动	959,108
中国教育和科研计算机网	40,960
中国科技网	53,248
中国国际经济贸易互联网	2
合计	6,640,291

  - 截止到 2016.12，中国 IPv4 地址数维持在 3.38 亿，已经达到饱和。

- 

  - 截止到 2016.6，中国 IPv6 /32 地址块数已达21188个。[仍在持续增长中]
-

  - 中国 ".cn" 域名总数为2061万，年增长 25.9%，占中国域名总数的 48.7%。".com" 域名数为1435万，占比33.9%。
  - 中国网站 (在中国境内注册) 数量为 482 万个，年增长 14.1%。
  - 中国网页数量为 2360 亿个，年增长 11.2%。
- 中国教育科研网 (CERNET)
  - CERNET

- 

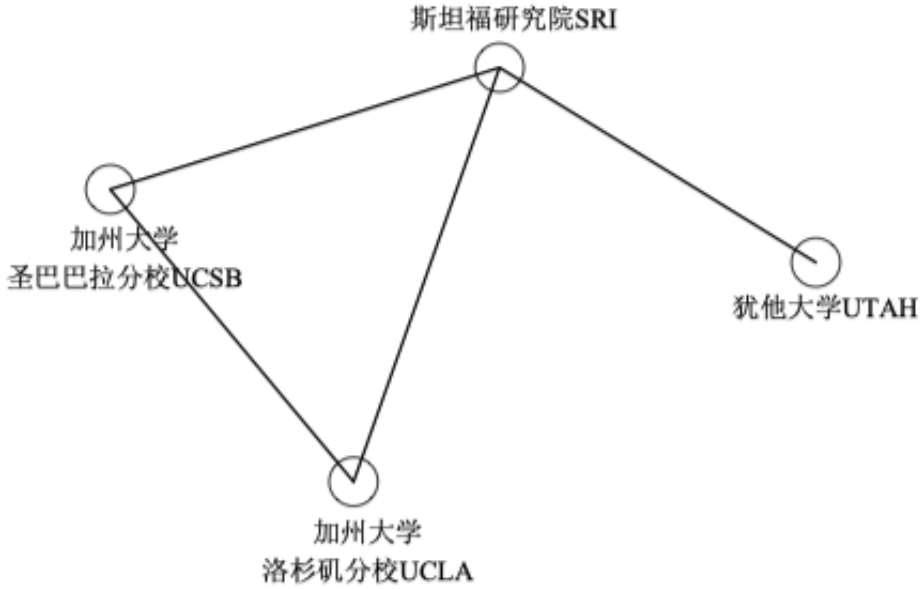
  - CNGI-CERNET 2
    - 世界上最大的纯 IPv6 网络。第二代中国教育和科研计算机网。
    -

1.2 互联网发展史

1.2.1 互联网起源

- 时间：1955-1970s初期

- 起源
  - 1957.10 前苏联发布第一颗人造地球卫星，直接促使美国国防部在 1958.2 成立了ARPA (高级研究计划署 - Advanced Research Projects Agency)，即之后的 DAPRA (国防部高级研究计划署)。
  - ARPA的使命
    - 为保卫国家安全的技术突破而做关键性的投资，不以满足军方的现实需求为目标，而是感知军方的未来潜在的需求，探索国防科技。
    - 曾经推动的重要项目：BMP、波音 X-37、GPS、ARPANET
  - 1969年，ARPA 推出了 ARPANET，即 Internet 的前身。
    - 一开始的 ARPANET 包括四个主要节点。



    - ARPANET 的主要技术特点 —— 采用了包交换 (分组交换技术)，完全不同于当时最主要的通信系统 (电话网络) 所采用的电路交换技术。
  - 分组交换技术
    - 让 ARPANET 具备抗毁性。即传输节点在失效后，独立寻径的那些包可以找到其他的路径到达对方。
    - 执行分组交换的设备 —— IMP (接口报文处理器 - Interface Message Processor)

1.2.2 Internet 的形成

- TCP/IP协议的诞生
  - 瑟夫、凯恩、柏兰登等人一起发明了互联网的骨干协议 TCP/IP。
  - 当今互联网的主要特点 —— TCP/IP 协议簇的形成和应用。
- 互联网正式诞生
  - 1981年，CSnet (计算机科学网络) 成立，第一次实现了两网互连，标志着互联网正式诞生。

1.2.3 互联网普及

- 技术储备

    - 个人电脑的出现 (鼠标的使用)
    - 视窗操作系统的出现
    - 浏览器的诞生
        - 1989年，Tim Berners Lee 成功开发出世界上第一个 Web 服务器和第一个 Web 客户机。World Wide Web 就此诞生。
- 中国的互联网络历史
    - 1987.9.20，从北京向德国卡鲁厄尔大学发送了第一封邮件，标志着中国人使用 Internet 的起点。
        - 邮件历时一周，邮件内容 "Across the Great Wall we can reach every corner in the world !"。
    - 1994.4.20，中国实现了与 Internet 的全功能连接。至此中国实现与互联网的完全对接。

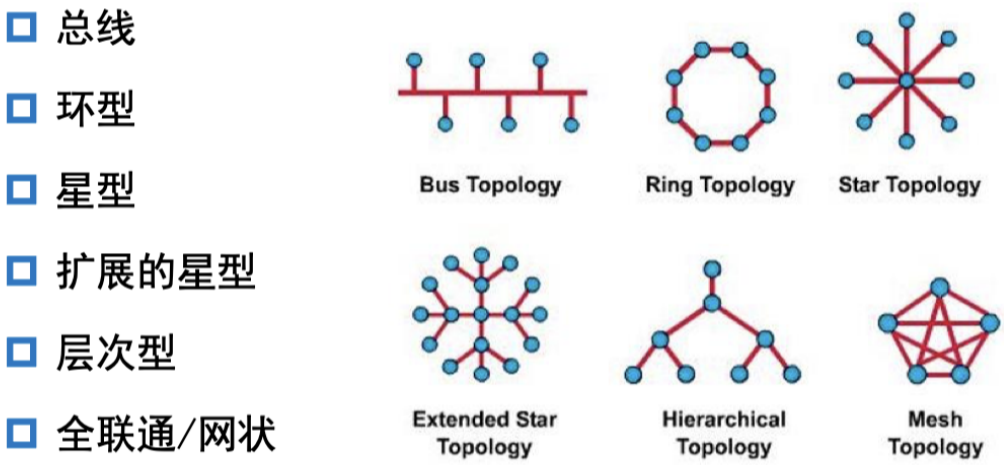
1.3 基本概念解析

1.3.1 网络

- 广义网络
    - 广义上网络，是指类似的事物连接在一起，以提供某些功能。如交通网络、自来水网络、邮政网络、电话网络等。
- 计算机网络
    - 计算机网络是网络的一种，是使用单一技术相互连接的自主计算机的互联集合。
    - 单台计算机具有独立自主性，它的开与关是独立自主的。
    - 计算机的连接介质可以是光纤、铜线，也可以是微波、红外和卫星。网络可以呈现出不同的大小、形状和形式。
- 互联网络
    - 互联网络是计算机网络的互相连接，是网络的网络。互联网络就是 Internet，是覆盖全球的最大的一张互联网络。
- 万维网 (World Wide Web)
    - 不是互联网网络，而是构建在互联网络上的一个分布式系统。
    - 是信息资源的网络，资源、资源标识和传输协议三部分支撑 www 的运作。

1.3.2 拓扑 (Topology)

- 信道 (信号的通道) 的分布方式。
- 常见的拓扑结构：总线型、星型、环型、树型和网状。

- 

- 最常见的拓扑方式是总线拓扑和星型拓扑。(以太网的两种拓扑方式)
  - 总线拓扑：主机挂接在总线上，相互直接通达。
    - 从主机上发出的信号在总线上双向同时传输。
    - 所有的主机都可收到这个信号。
  - 星型拓扑：主机都挂接在一个中心节点上。
    - 早期中心节点由集线器充当，现在的中心节点主要由交换机充当。
    - 缺点：单点故障

1.3.3 协议

- 协议：一系列规则和约定的规范性描述，它控制网络中的设备之间如何进行信息交换。

1.3.4 数字带宽 (Bandwidth)

- 数字带宽：单位时间内流经的信息总量
- 带宽越大，里面的信息越大。
- 单位
  - 数字带宽的基本单位 - 比特每秒，即 bps
  - $1\ \text{Gbps} = 10^3\ \text{Mbps} = 10^6\ \text{Kbps} = 10^9\ \text{bps}$
  - 更大的单位：T、P、E、Z、Y，倍数均为 10^3
- 带宽延迟积
  - $\text{带宽} * \text{传播延迟时间} = \text{任何特定时间该线路上的最大数据量}(\text{已发送但尚未确认的数据})$

1.3.5 吞吐量

- 吞吐量：指实际的、可测到的带宽
  - 数字带宽是理想的带宽，吞吐量才是用户真实体验到的带宽。
  - 吞吐量是用户评价网络性能的一个非常重要的指标。
- 吞吐量影响因素

- 网络设备、传输数据类型、网络拓扑、用户数量、服务器等

- 信息量、带宽和传输时间的关系
  - $T = \frac{S}{BW} = \frac{S}{P}$
  - S为信息量，BW为带宽，P为吞吐量，T为传输时间
  - 如果ISDN的带宽为 128kbps，OC-48的带宽为 2.488 Gbps，如果用ISDN传输一张装满数据的1.44M软盘，用OC-48传输装满10G的硬盘数据，问哪一种传输所用的时间更少？

    解：按照理想的传输状况来计算，即根据： $T = \frac{S}{BW}$

    有： $T_{fd} = \frac{1.44M}{128kbps} = \frac{1.44 \times 10^3 \times 8}{128} = 90s$

    $T_{hd} = \frac{10G \times 8}{2.488Gbps} = 32.152s$

    答：传输10G的硬盘数据所化的时间更少。

1.3.6 点到点与端到端

- 点到点
  - 点到点基于MAC地址或IP地址，指一个设备发数据给与其直接连接的其他设备，这台设备又在合适的时候将数据传递给与它相连的下一个设备，通过一台一台直接相连的设备把数据传递到接收端。
  - 主要用于数据链路层和网络层。
  - 优点
    - 发送端发出数据即完成任务，不需要参与整个传输过程。
    - 即使接收端设备关机或故障，点到点传输也可以采用存储转发技术进行缓冲。
  - 缺点
    - 发送端发出数据后，不知道接收端能否收到或何时收到。
- 端到端
  - 端到端指在数据传输之前，先为数据的传输开辟一条通道，然后再进行传输。从发送端发出数据到接收端接收完毕时结束。
  - 主要用于传输层。
  - 优点
    - 链路建立之后，发送端知道接收端一定能收到，并且经过中间交换设备时不需要进行存储转发，因此传输延迟小。
  - 缺点
    - 在接收端收到数据之前，发送端的设备一直要参与传输。如果整个传输的延迟很长，会对发送端设备造成很大浪费。
    - 如果接收设备关机或故障，则端到端传输不可能实现。

1.3.7 计算机网络的分类

- 按传输介质分
  - 有线网络和无线网络
- 按网络的大小和规模来划分
  - 个域网 (PAN - Personal Area Networks)
    - 覆盖范围 1 米左右
    - 电脑、键盘、鼠标构成了一个非常小的 PAN
  - 局域网 (LAN - Local Area Networks)
    - 覆盖范围 1 公里左右
    - 一个校园、一个单位或一个建筑
  - 城域网 (MAN - Metropolitan Area Networks)
    - 覆盖范围 10 公里左右
    - 一个城市
  - 广域网 (WAN - Wide Area Networks)
    - 覆盖范围 100/1000 公里左右
    - 一个州或一个国家
  - 互联网 (Internet)
    - 覆盖范围 10^4 公里左右
    - 全球
  - 分辨网络类别不能只看覆盖范围，关键要看是否具有该网络类别下的技术特征。

1.3.8 课程总结

- 什么是计算机网络?
- 什么是互联网络和万维网？两者之间有什么关系?
- 什么是协议?
- 什么是网络的物理拓扑?
- 什么是数字带宽和吞吐量？两者之间有什么关系?
- 什么是点到点和端到端?
- 按照规模划分，计算机网络分成哪些类?

1.4 参考模型

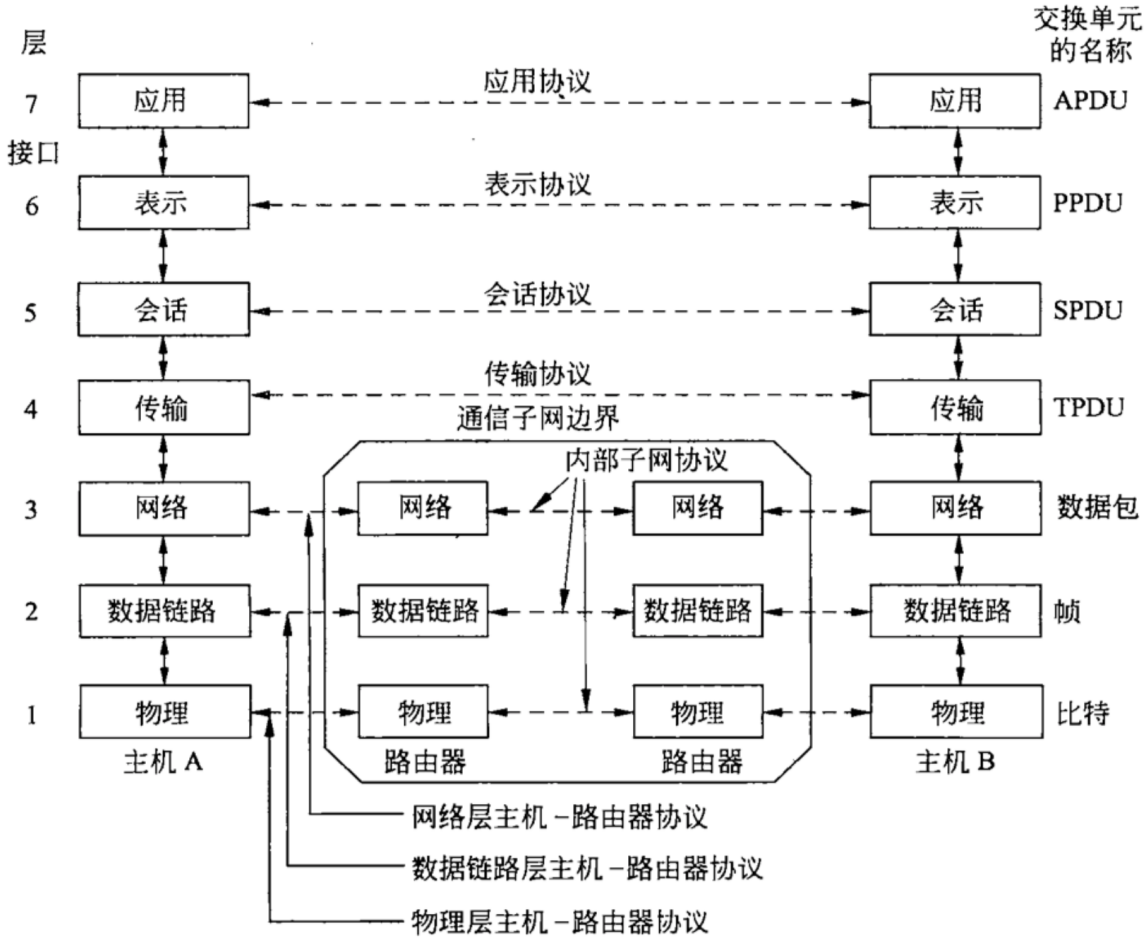
1.4.1 参考模型的必要性

- 分层的好处
  - 各层独立工作，层之间通过接口联系，降低协议工作的复杂程度。
  - 灵活性好，任何一层的改变不影响其它层。
  - 每层的技术实现不同，减少了实现的复杂度。
  - 易于维护，每一层可以单独进行调试。

    - 便于标准化。
- 分层原则
    - 信宿机第 n 层收到的对象应与信源机第 n 层发出的对象完全一致。

1.4.2 OSI 七层模型

- OSI (Open System Interconnection)
    - 由 ISO (International Standards Organization) 在 1983 年提出。
-

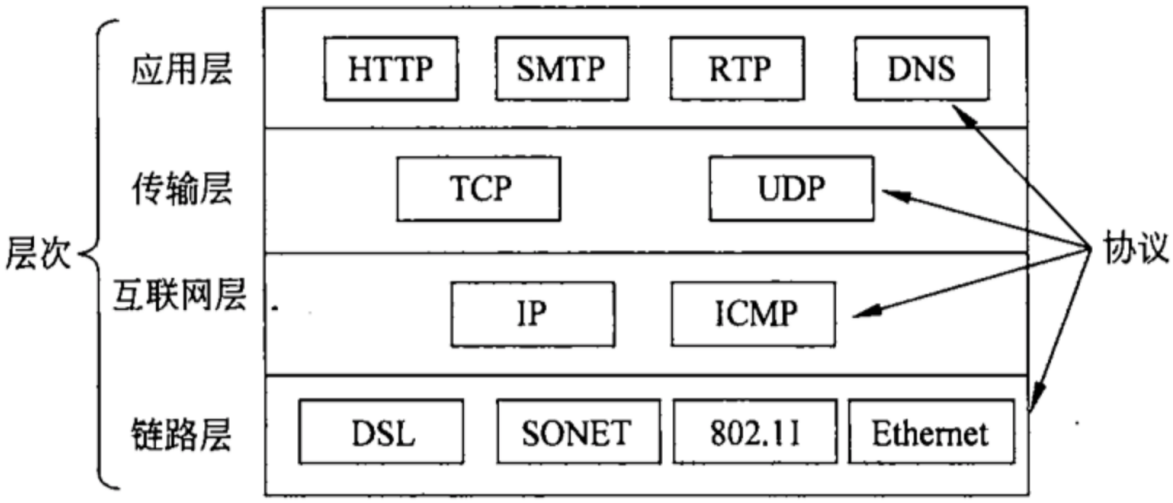


- 七层模型
    - 7-应用层
        - 主要为各种各样的网络应用提供网络服务，如Email、FTP、微信等。
    - 6-表示层
        - 将信息表示为一定形式和格式的数据流，完成压缩解压缩、加密解密等任务。
    - 5-会话层
        - 负责通信主机间会话的建立、管理和拆除，协调通信双方的会话。
    - 4-传输层
        - 参考模型上的核心层之一，负责通信主机间端到端的数据传递。(核心功能)
        - 对于 TCP，该层还负责提供可靠传输、差错恢复、拥塞控制等额外的功能。
    - 3-网络层
        - 核心层之一，两大主要功能：地址、最优路径。负责把每一个分组从源机一路送达到目

的机。
        - **地址**，即为通信主机提供标识，在网络层上用 IP 地址进行标识。
        - **最优路径**，在路由和寻径时，每一个中间设备都会为到达的分组找到一个最优的路径并送出。
    - 2-数据链路层
        - 该层主要提供介质访问服务。
        - 通过物理地址来标识通信主机，提供可靠的帧传递并做差错控制和流量控制。
    - 1-物理层
        - 该层提供透明的比特流传输，可以是光、电、无线信号。
        - 该层只关心比特流的传输，而忽略比特流里面的具体内容。
- 特点
    - 每一层都利用它下层的服务，为它的上层提供服务。除了第 1 层和第 7 层。

1.4.3 TCP/IP 四层参考模型

- 四层模型 (DOD模型)
    - 4-应用层
    - 3-传输层
    - 2-互联网层
    - 1-链路层
-



1.4.4 OSI与TCP/IP参考模型比较

- TCP/IP参考模型是当今互联网所采用的模型，而 OSI 仅存在于教科书中。
- 两个模型分层方式不一样，但都采用分组交换技术进行信息传输。
- OSI参考模型失败原因?
    - 时机糟糕 —— TCP/IP 在 OSI 出现之前已经非常流行。
    - 技术糟糕 —— 掺杂了太多政策的考量
    - 实现困难 —— 庞大而笨拙

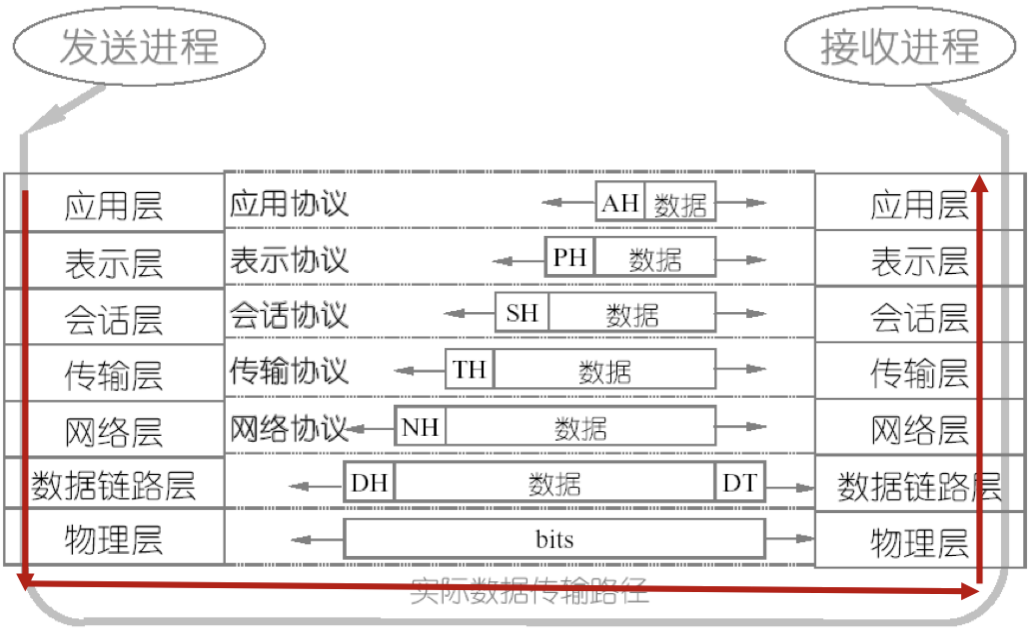
1.4.5 课程总结

- 为什么需要参考模型?
- ISO-OSI参考模型上各层名称与各层主要功能?
- TCP/IP参考模型的各层名称和功能?
- 比较ISO-OSI参考模型和TCP/IP参考模型。
- 为什么ISO-OSI参考模型最后失败了?

1.5 参考模型中相关概念解析

1.5.1 发送方与接收方之间的通信

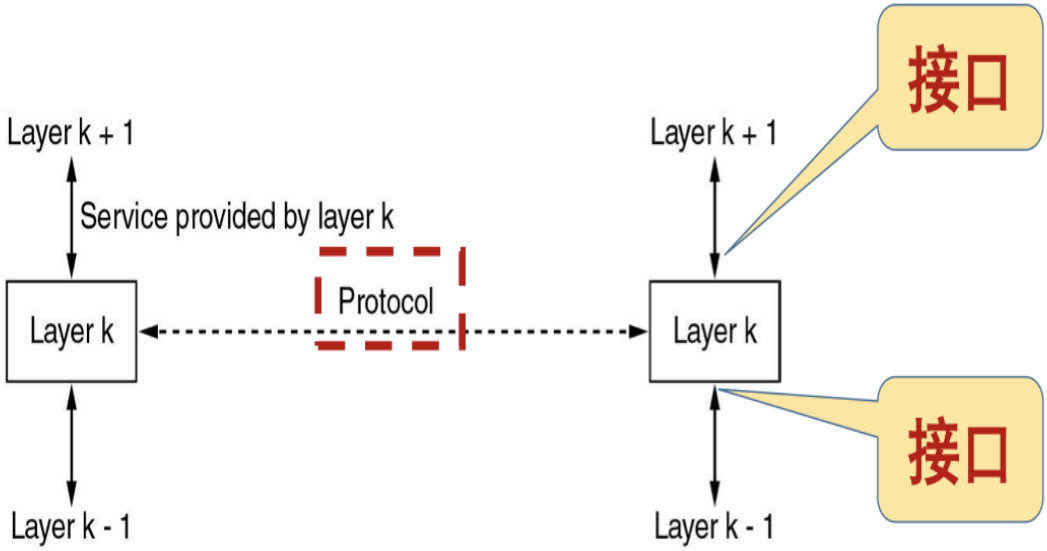
- 发送方
  - 封装/打包
    - 将信息打包，从最高层 (应用层) 开始逐渐下行到最底层 (物理层)。
    - 在每一层上，数据都被加上头部信息，用于传递信息。
  - 具体过程
    - ① 在 OSI **参考模型的上三层**，信息被表示为一定格式和形式的数据流 (DataStream)。
    - ② 数据流被传到**传输层**，将其切割为适合传输的数据段 (Segment)，并加上段头，段头中包含定位应用进程的端口号等信息。
    - ③ 数据段传到**网络层**，添加分组头部形成分组 (包-Packet)。分组头部中包含寻址主机的IP地址和其它一些传输需要用到的信息 (如生存周期、长度等)。
    - ④ 分组再传到**数据链路层**，添加帧头形成帧 (Frame)。帧头中包含寻址主机所需的物理地址、校验等信息。
    - ⑤ 帧到了**物理层**，被转成了可供传输的比特流 (Bits)。
  - 数据在各层的形式
    - PDU (协议数据单元 - Protocol Data Unit)，指数据在各层的形式 (即各层处理的数据对象)，下述数据形式统称为PDU。PDU 包含"控制信息" 和"用户数据"两个部分。
    - 应用层 —— 信息 (Information)
    - 上三层 —— 数据流 (Data Stream)
    - 传输层 —— 数据段 (Segment)
    - 网络层 —— 分组/包/报文 (Packet)
    - 数据链路层 —— 帧 (Frame)
    - 物理层 —— 比特流 (Bits)
- 接收方
  - 解封装/解包
    - 将收到的比特流解包，从最低层 (物理层) 开始逐渐上行到最高层 (应用层) 提取出信息。
    - 解封装的过程是封装的逆向过程，在每层去掉头部信息，最终还原出应用层的输出信息。
  -



- 任何一次通信，总是以发方的封装开始，以接收方的解封装结束。

1.5.2 对等实体与虚拟通信

- 实体：每层中活动的元素
  - 在每一层都有一个实体，负责完成封装、解封装等基本功能。
  - 实体可能是硬件，也可能是软件。
- 对等实体 (Peer Entity)：收发双方对应层上的实体互称为对等实体。
- 虚拟通信 (对等通信)
  - 收方双方对应层之间 (对等实体之间) 有一根直接的通道 (虚通道)。沿着这根通道，PDU从发方到达收方。
- 服务、协议和接口之间的关系
  - 参考模型上的每一层都为它的上层提供服务，以接口的形式供上层调用。
  - 协议作用于对应层之间。
  - 层与层之间有接口，用于服务的调用。



1.5.3 课程总结

- 数据传输三要素是什么？(信源-传输介质-信宿)
- 数据传输时，起点-源要做什么？
- 数据传到目的地时，目的机要做什么？
- 封装的过程是怎样的？
- 解封装的过程是怎样的？
- 什么是实体 (Entity)？
- 什么是对等通信 / 虚拟通信？
- 服务、协议和接口之间的关系是怎样的？

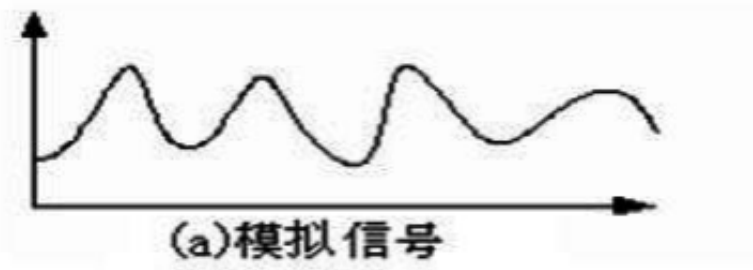
第二章 物理层

2.1 物理层概述

2.1.1 物理层功能、特性以及物理带宽

- **物理层主要功能**：提供透明的比特流传输
  - 封装好的数据以 "0"、"1" 比特流的形式进行传递，从一个地方搬到另一个地方。
  - 物理层上的传输，从不关心比特流里面携带的信息，只关心比特流的正确搬运。
- **物理层的四大特性**
  - 机械特性
    - 指明接口所有接线器的形状、尺寸、引脚数和排列等，如RJ45水晶头。
  - 电器特性
    - 指明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。
    - 规程特性 (过程特性)
        - 指明对于不同功能的各种可能事件的出现顺序。规程的概念类似于协议。
- 物理层上数据的传输
    - 数据传输以信号形式进行。信号 —— 数据的电气或电磁表现，分为模拟信号和数字信号两类。
    - 模拟信号
        - 对应时域的信号取值是连续的。



    - 数字信号
        - 对应时域的信号取值是离散的。

        - 码元 —— 代表不同离散值的基本波形。
    - 信号在信道/传输介质上传输
        - 信号在传输的过程中，可以看成由很多不同频率的分量的传输。高频分量的不等量衰减，接收方收到的信号衰减和变形 (失真) 的。
        - 在0~fc这一频段，振幅在传输过程中不会明显衰减，fc就称为截止频率 (物理带宽)。(单位：Hz)
- 物理带宽
    - 传输过程中振幅不会明显衰减的频率范围。
    - 物理带宽是一种物理特性，通常取决于介质材料、厚度、长度等。因此材质的物理带宽是可以确定的。

2.1.2 奈奎斯特定理

- 特点：描述理想信道的最大传输速率 (数字带宽)。
- 奈奎斯特定理
    - 在无噪声信道中，当物理带宽为 B Hz，信号离散等级为 V 级 ($V = 2^{比特数}$)，则

$\text{最大传输速率} = 2Blog_2V\ (bps)$

- 任意一个信号通过一个物理带宽为 B 的低通滤波器，只要进行每秒 2B 次的采样就可以完全重构出被滤掉的信号。任何高于 2B 次的采样都毫无意义。每次采样是 log_2V 个比特。

  有一条4-kHz的无噪声信道，每秒采样8000次，如果每个采样是16比特，则信道的最大传输速率是 _____kbps

  解：直接套用乃奎斯特定理

  - $$\text{最大传输速率} = 2\mathrm{B}\log_2 V(bps)$$

    得到 8k×16bit=128kbps

    注意 每个采样16比特，相当于信号级别是2^{16}。

2.1.3 香农定理

- 特点：描述有噪声信道中的最大传输速率 (数字带宽)。
- 香农定理
  - 在噪声信道中，信噪比为 S/N ，物理带宽为 B Hz，则 $\text{最大传输速率} = Blog_2(1+\frac{S}{N})\ (bps)$
  - 信噪比
    - 信号功率和噪声功率的比值，通常将该比率表示成对数形式，对数取值单位为分贝 (db)。
    - 10的信噪比为10db，100的信噪比为20db，1000的信噪比为30db。
    - 信噪比无单位，但通常用对数形式来度量信噪比，对数形式有单位，单位为db。
  - $\text{分贝值} = 10 * log_{10}\frac{S}{N}(db)$ ，在物理带宽之后，只有通过提高信噪比才能提高传输速率。
  -

    如果一个二进制信号通过一条4-kHz的噪声信道，噪声是30分贝，则最大传输速率是_____ kpbs

    解：根据 30dB=10×$\log_{10}$S/N

    得到：S/N=1000

    再根据香农定理：最大传输速率 $= \mathrm{B}\log_2(1+\frac{S}{N})(bps)$

    得到最大传输率是： $4\mathrm{k}\times\log_2 1000 \approx 40\mathrm{kbps}$

  -

7 单选 (2分) 为了在1 MHz带宽的信道上得到 4 Mbps 的比特率，信噪比需要达到多少？ 得分/总分

- A. 15dB ✕0.00/2.00
- B. 11.76
- C. 1.18dB
- D. 15

正确答案：D 你错选为A

解析： C、Bitrate = B log2(1+S/N)
=> 4 * 10**6 = 10**6 log2(1+S/N)
=> log2(1+S/N) = 4
=> 1+S/N = 16
=> S/N = 15
香农极限：R = B log2(1+S/N)，R为比特率，B是信道带宽，S/N是信噪比
转换公式：S/N (in dB) = 10*log10(S/N)=11.8db

15 单选 (2分) 若连接R2和R3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多少？ 得分/总分

- A. 20kbps ✕0.00/2.00
- B. 80kbps
- C. 40kbps
- D. 8kbps

正确答案：C 你错选为A

解析： B、解析：香农定理给出了贷款受限且有高斯白噪声干扰的信道的极限数据传输速率，香农定理定义为：信道的极限数据传输速率=Blog2(1+S/N)，单位b/s。其中，S/N为信噪比，即信号的平均功率和噪声的平均功率之比，信噪比=10log10(S/N)，单位dB，当S/N=1000时，信噪比为30dB。则该链路的实际传输速率约为50%*Blog2(1+S/N)=50%*8k*log2(1+1000)=40kbps

-

2.1.4 课程总结

- 物理层的主要功能是什么？
- 物理层的4大特性分别是什么？
- 什么是物理带宽？它的单位是什么？
- 数字带宽 (bps) 与物理带宽 (hz) 之间有什么关系？

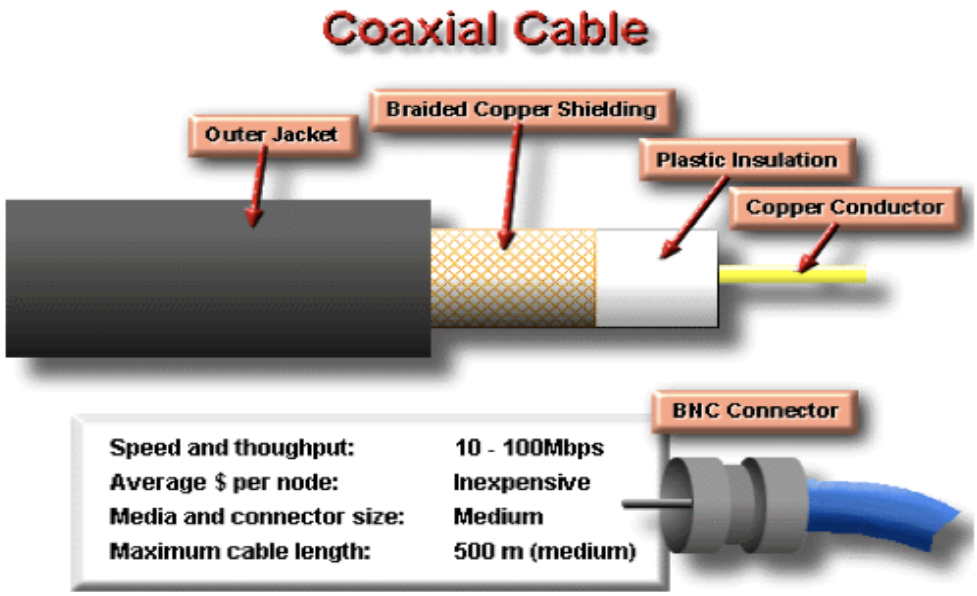
2.2 引导性传输介质

2.2.1 传输介质分类

- 按照是否有形进行分类

- 引导性 (有线) —— 铜线、光缆
- 非引导性 (无线) —— 无线电、卫星

2.2.2 同轴电缆 (铜线)

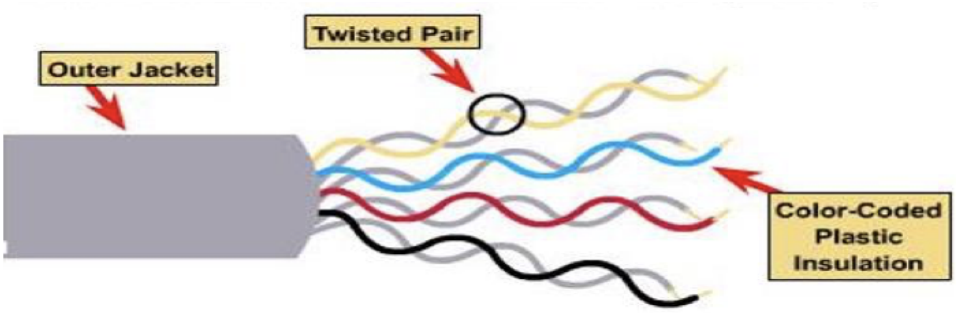
- 同轴电缆的组成
  - 

  - "中心导体 - 绝缘材料层 - 网状导体 - 外部绝缘层" 四层组成。
- 同轴电缆的分类
  - 按照功能分类
    - 基带同轴电缆：阻抗 = 50 Ω，用于数字传输 (屏蔽层为铜)
    - 宽带同轴电缆：阻抗 = 75 Ω，用于模拟传输 (屏蔽层为铝)
  - 按照粗细分类
    - 粗缆：最大传输距离到达 500 米，早期用来做 10Base5 的总线。
    - 细缆：最大传输距离为 185 米，早期用来做 10Base2 的总线。

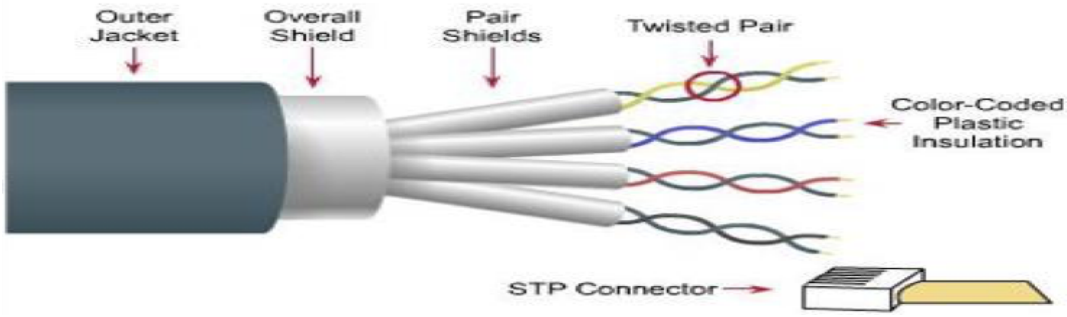
2.2.3 双绞线 (铜线)

- 双绞线的组成
  - 由两根具有绝缘层的铜导线按一定密度，逆时针方向绞合而成。
  - 绞合目的：消除近端串扰，让电感和电容相互抵消，减低自身线对之间产生的干扰，也可以抵御一部分的电磁波干扰。绞距越紧 (小)，越均匀，则抵消效果越好，传输性能越好。
- 双绞线的分类
  - 非屏蔽双绞线 (UTP - Unshielded Twisted Pair)
    - 即五类双绞线，常用于 RJ45 水晶头中。
    - 提供10M、100M的数字带宽，使用了其中的两对线，分别用于收、发。在1000M的以太网中，用到了全部的 4 对线。

        - 最大传输距离为 100 米，广泛用于局域网中。用户桌面的线缆中大量使用UTP (水平电缆)。



        - 优点：成本低、尺寸小、易于安装。
        - 缺点：易受干扰 (无屏蔽层)，传输距离和性能都受到绞距的影响。
        - 局域网中，使用最多的是 UTP。
        - 直通线
            - 线两头的线序一致，连接两个不同设备的时候会使用直通线。(比如交换机和PC)
        - 交叉线
            - 两段线序不一致 (左边12对右边36)，连接两个相同的设备。(比如两个路由器)
    - 屏蔽双绞线 (STP - Shielded Twisted Pair)
        - 相比 UTP，STP加了两层屏蔽层，分别位于每对线之外和全部4对线之外。



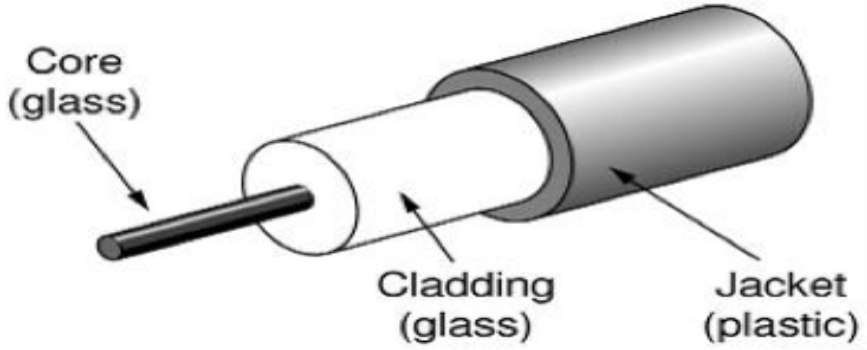
        - 优点：抗 EMI/RFI 干扰
        - 缺点：成本高、安装不易 (尺寸变大)
    - 网屏式双绞线 (ScTP - Screened Twisted Pair)
        - 在成本和抗干扰之间做了折中。只有一个屏蔽层，将STP中每一个线对上的屏蔽层去掉了。

2.2.4 电力线 (铜线)

- 电力线的赫兹固定为50Hz。其电气特性等并不适合传输数据，现在用电力线传输数据并未普及。

2.2.5 光纤

- 组成
    - 光纤，即光导纤维。由极细的玻璃纤维构成，把光封闭在其中并沿轴向进行传播。
    - "极细的玻璃芯-玻璃覆盖层-塑料封套" 三部分组成



- 优缺点
  - 优点：重量轻、损耗低、不受电磁辐射干扰 (最大优点)、传输频带宽和通信容量大，干线上大量使用光纤 (垂直电缆)。
  - 缺点：昂贵、易断裂
- 分类
  - 单模光纤
    - 单一模式传输，激光产生的单束光。纤心细、高带宽、长距离，运行波长为850nm或1300nm。
    - 规格 —— 8.3/125 um
  - 多模光纤
    - 以多个模式同时传输，LED产生的多束光。
    - 规格 —— 62.5/125 um
- 光纤的三种连接方式
  - 光纤连接器 (光损失 10~20%)
  - 机械拼接，特殊的套管夹紧 (光损失 10%)
  - 光纤熔接机 - 熔合 (几乎无损失)
- 与铜线对比
  - 光纤 —— 带宽高、距离远、损耗低、重量轻、无电磁干扰和射频干扰 (防窃听)、价格高

2.2.6 如何选择传输介质

- 选择传输介质需要考虑的因素
  - 传输速率、成本、周围环境、介质间的互操作性、相容性、最优的性价比
  - 原则 —— 满足需求的前提下选择性价比最高的

2.2.7 课程总结

- 什么是引导性传输介质?
- 铜传输介质主要包括哪些? (同轴电缆、双绞线、电力线)
- 非屏蔽双绞线有什么特点? 主要用在哪里?
- 非屏蔽双绞线、屏蔽双绞线各自的优缺点是什么?
- 光纤有什么特点? 主要用在哪里?
- 单模光纤和多模光纤有什么区别?

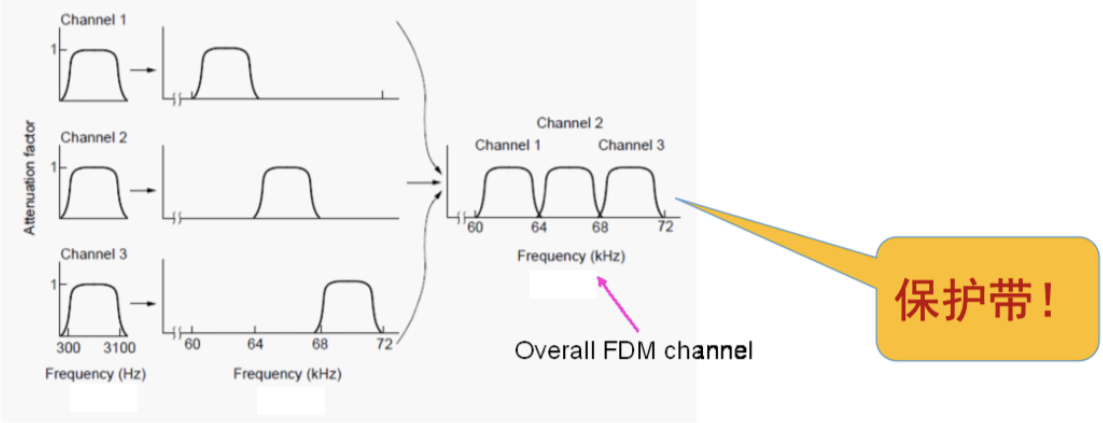
2.3 复用技术

2.3.1 复用技术概念

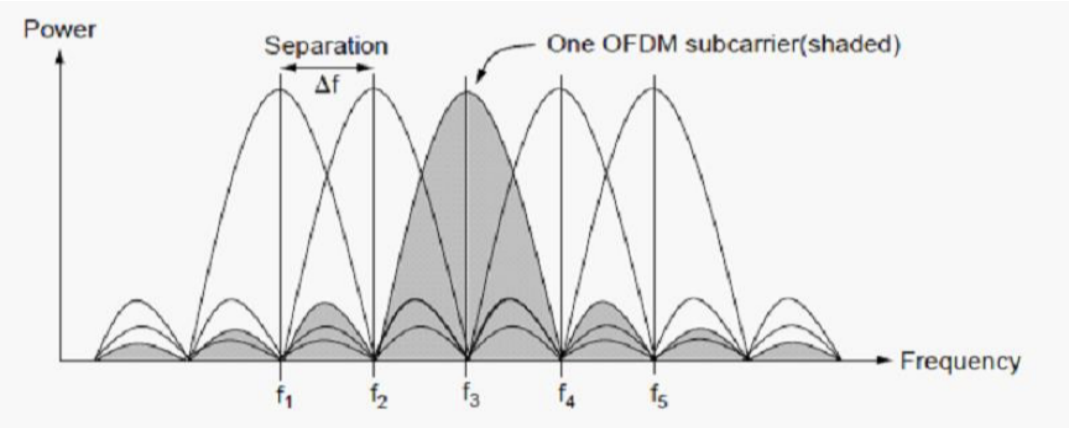
- 复用技术：让多个用户共享同一个信道。
  - 在干线上的技术。要解决的问题是干线起点如何共用，干线终点如何分离的问题。

2.3.2 频分多路复用 (FDM)

- 原理
  - 在干线起点，信道的频谱被分成若干段 (子带)，每个用户占据一段来传输自己的信号。到了干线的终点，每个子带的信号被单独分离出来给各个用户。
  - 相邻用户使用的频段 (子带) 之间通常留有一定的带宽，避免混淆。这个频段称为保护带。

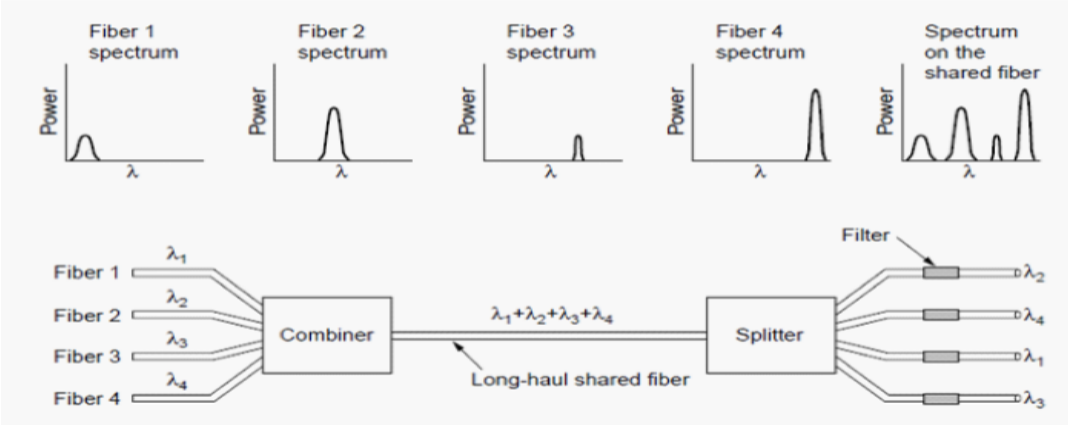


- 其他类别
  - 正交频分多路复用 (OFDM - Orthogonal FDM)
    - 没有保护带，且子带之间相互重叠。同样的干线可以承载更多的用户。被广泛地使用于802.11、有线电视网络等。



2.3.3 波分多路复用 (WDM)

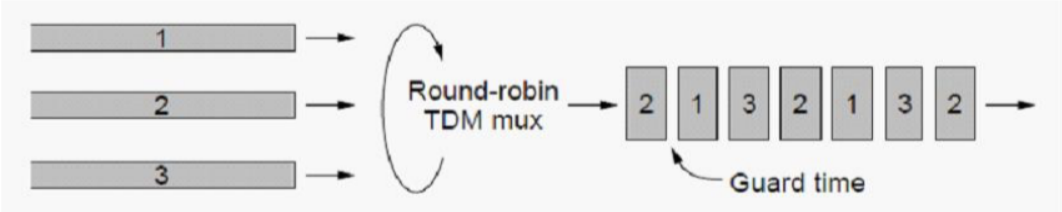
- 原理
  - 本质上和FDM一样，在光纤上复用信号。按照不同的波长，将干线分成了若干份，承载了不同用户的光信号。到了终点，分离器分离出不同波长的光信号。



- 类别
  - 密集波分多路复用 (DWDM - Dense)
    - 当相邻波长间隔非常非常接近时，子信道的数目就会非常非常大，此时 WDM 就变成了 DWDM，即密集波分多路复用。这是一个当下非常热门的技术。

2.3.4 时分多路复用 (TDM)

- 原理
  - 在时间上共享信道，将时间划分为非常短的时间片。
  - 每个用户周期性的在属于自己的那个时间片内使用整个带宽。TDM 广泛用于电话系统和蜂窝系统，要求时间上必须同步，为了适应时钟的微小变化，可能要求增加保护时间间隔。

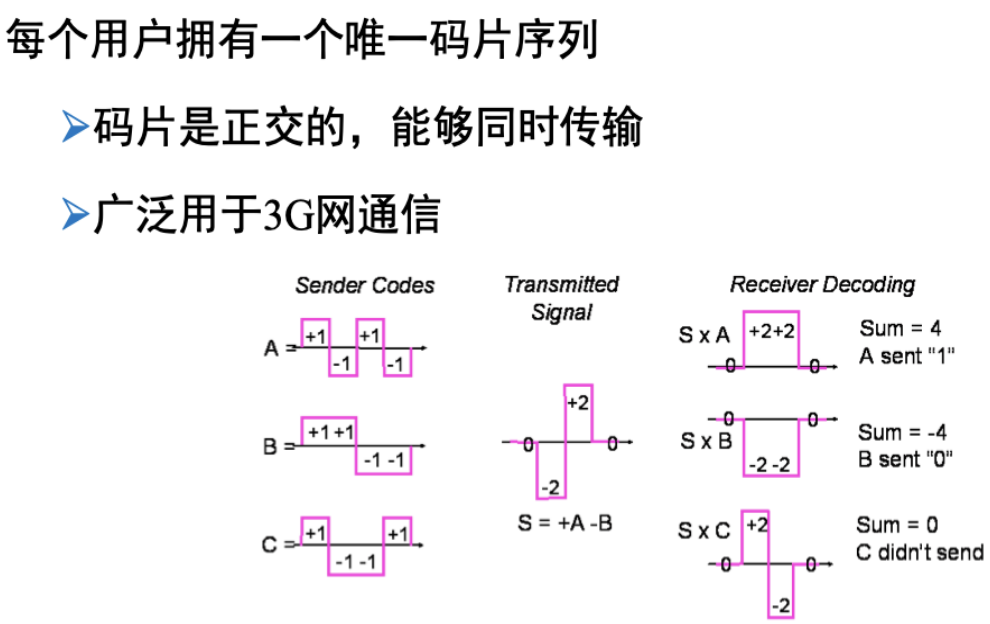


  - 缺点
    - 如果各用户需要的带宽的不均衡，而TDM用户时间片的使用却是一样的，就会造成信道的浪费，不高效。
- 类别
  - 统计时分多路复用技术 (STDM)
    - 动态分配信道。不适用信道的用户不分配，分给有需要的用户使用，利用率可提高2~4倍 (按需分配)。
    - 实现技术较为复杂，通常只在高速远程通信中使用，如 ATM。

2.3.5 码分多路复用 (CDMA)

- 原理

- 扩展频谱技术。
- CDMA允许每个站利用整个频段发送信号，而且没有任何时间限制。
- 类比 —— 大厅中人们两两交谈
  - TDM：许多人按顺序交谈
  - FDM：不同人按不同语调同时进行交谈。
  - CDMA：每对交谈者使用不同的语言。
- CDMA的关键在于能够提取出需要的信号，同时拒绝所有其它的信号，并把这些信号当作噪声。
- 在CDMA中，每个比特时间被细分成 m 个更短的时间间隔，这更短的时间间隔被称为码片。通常每个比特被分为64或128。
-



- 例题

10 单选 (2分) 假设一个CDMA系统有3个工作站，码片序列由4位构成，三个工作站A、B和 C的码片序列分别是： (1,1,1,1),(1,-1,1,-1) and (1,1,-1,-1)。如果工作站 C 连续收到了一个复用信号： (2,0,2,0,0,-2,0,-2,0,2,0,2)，请用计算结果回答：工作站C从工作站A收到了什么信号？ 得分/总分

- A. 100
- B. 010
- C. 001 ✕0.00/2.00
- D. 101

正确答案：D 你错选为C

解析： C、(2.0,2,0)*A /4= 1
(0,-2,0,-2)*A/4 = -1;
(0,2,0,2)*A/4 = 1;
所以，工作站C 从A收到了比特流“ 101”

2.3.6 课程总结

- 为什么要复用？在哪里复用? (干线)
- FDM技术的工作原理是什么？
- TDM技术的工作原理是什么？
- WDM技术的工作原理是什么？
- CDMA技术的工作原理是什么？
- 习题
  - 10 个用户使用 TDM 或 FDM 共享8 M bps 链路，使用 TDM的每个用户都要以一个固定的顺序轮流完全占据连接 1 ms（毫秒）；当用户传输一个3000 字节的消息时，哪个方法（TDM还是FDM）具有最低的可能延迟，该延迟时间是多少？

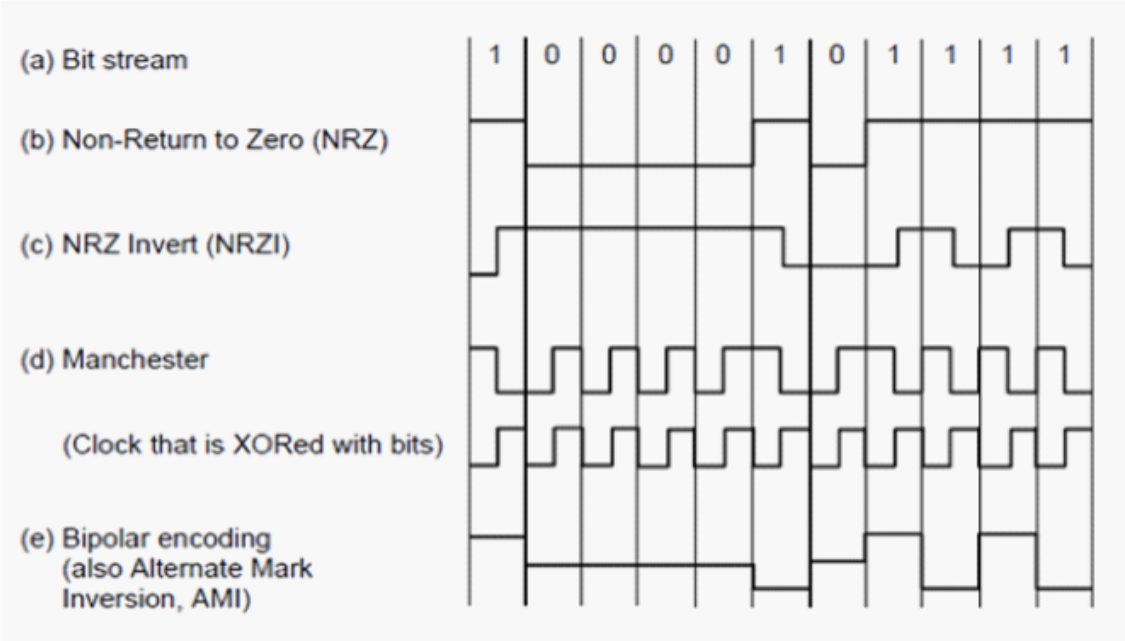
    A. TDM， 21 ms　　C. FDM，18 ms

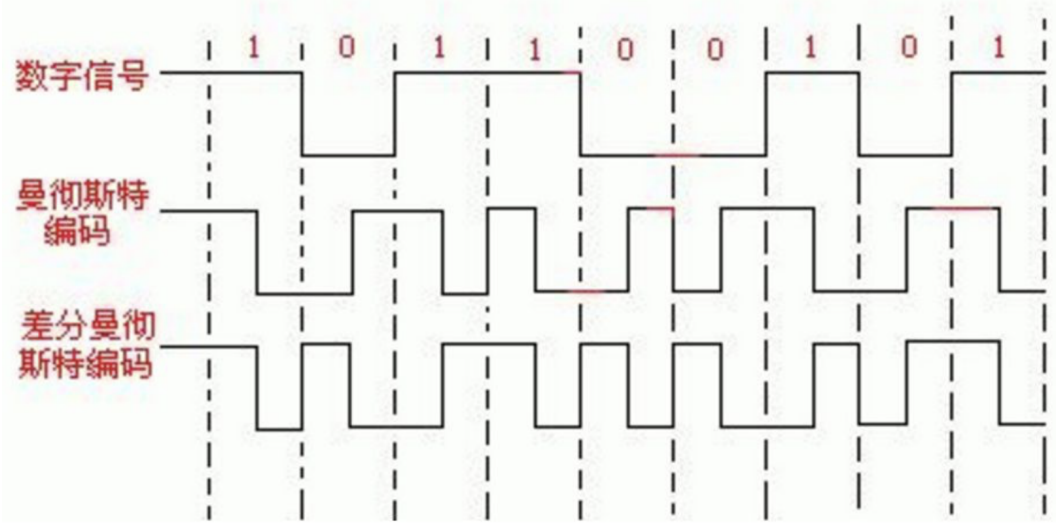
    B. 两种方法有相同的延迟　　D. TDM，27 ms
  - FDM：每个用户分得带宽 $\frac{8M}{10} = 800\ kbps$，所以传输 3000 字节需要时间约 $\frac{3000*8}{800kbps} = 0.03s = 30ms$。
  - TDM：每个用户轮发数据量为 $8M * 1ms = 8000b$，发送 $3000B\ (24000bit)$ 需要轮 3 次。那么需要等待时间为 $(3-1)*10 = 20ms$，发送剩下的 8000b 需要时间 1ms，共需 21ms。

2.4 调制技术

2.4.1 基带传输

- 特点
  - 数字调制，即比特与代表它们的信号之间的转换过程。
  - 信号的传输占据传输介质从零到最大值之间的全部频率。(有线传输介质普遍采用的方法，如以太网)。
- 线路编码
  -

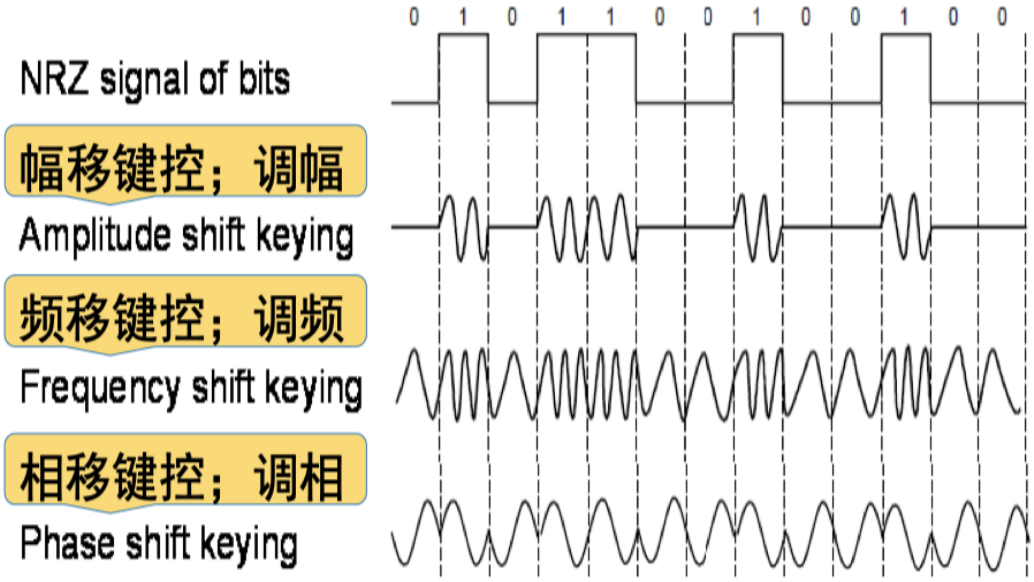


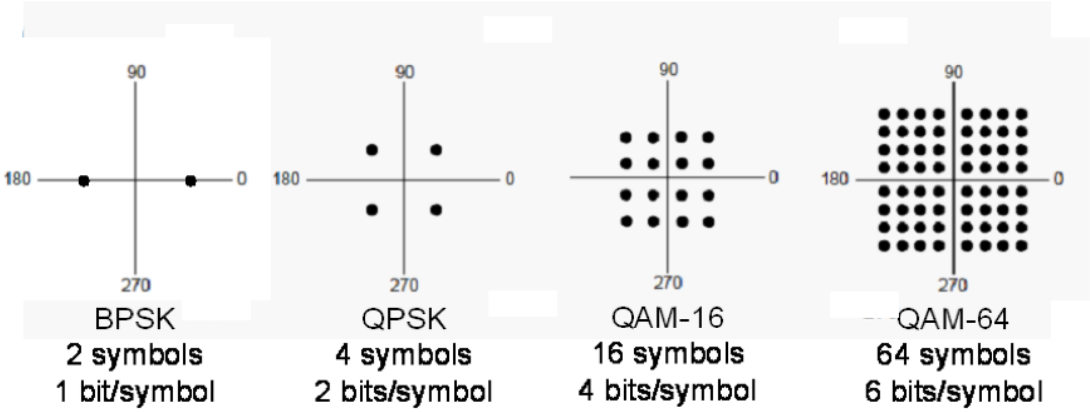
- 不归零 (NRZ)
  - 高电平表示 "1"，低电平表示 "0"。
  - 缺点：如果出现连续的"1"或者"0"，随着时间漂移的累计，接收方无法分辨出是几个"1"或"0"。
- 不归零逆转 (NRZI)
  - 在比特时间中间做电压的跳变，表示"1"。无跳变则表示"0"。
  - 连续"1"问题得到了解决，但连续"0"问题仍然存在。主要应用于USB中。
- 曼彻斯特编码 (Manchester)
  - 在比特时间中间，电压从高跳变到低，表示"1"；从低跳变到高，表示"0"。
  - 解决了连续"1"和"0"的问题，应用于 10Base 以太网中。
  - 缺点：由于在比特时间中跳变，编码效率只有50%。
- 差分曼彻斯特编码
  - 

  - 特点：该编码在每个时钟周期的中间都有一次电平跳变，这个跳变做同步之用。在每个时钟周期的起始处，跳变则说明该比特是0，不跳变则说明该比特是 1 。
  - 优点：收发双方可以根据编码自带的时钟信号来保持同步，无需专门传递同步信号的线路，因此成本低。
  - 缺点：实现技术复杂。
- 双级编码 (交替标记逆转AMI)

        - 两级电压的交替出现表示"1"，不出现则表示"0"。实现了信号的平衡。
    - 4B/5B 编码
        - 将4个比特数据映射为1个5比特的模式，抛开了连续0的组合，解决了连续0的问题。
        - 相比曼彻斯特编码，该方法将效率提升到了80%。

2.4.2 通带传输

- 特点
    - 通过调节信号的振幅、相位或频率来传输比特。
    - 信号占据了以载波信号频率为中心的一段频带。
- 编码方法
    - 在一个信道上，发送信息所使用的频率范围并不是从0开始的。而是在某个频段上，通过调节信号的振幅、相位或频率来传输比特。
    - 

- 信号星座图
    - 

    - QAM-16
        - QAM：正交振幅调制，16：调制方式中有16种不同振幅和相位的信号组合模式。一个符号可以传输4个比特。
    - QPSK

        - QPSK：正交相移键控。使用了4个相位角度，每次采样 (码元、样本) 可表达的级别有4个。每次码元可表示2比特。
    - QAM-64
        - 允许64个不同的信号组合，即64个信号级别。每个码元可以传输6个比特。
- 码元
    - 定义：承载信息量的基本信号单位。
    - 数字通信中常用时间间隔相同的符号来表示一个二进制数字。这样时间间隔内的信号就称为二进制码元。
    - 使用时域的波形表示数字信号时，不同离散值的基本波形就是码元。一秒钟能够发送的码元的个数就称为波特率 (码率)。代表每秒钟信号变化的次数。
- 波特率与比特率
    - 波特率 (符号率、采样率)：每秒钟信号变化的次数，为 2 倍的物理带宽，即 2 B。
    - 比特率 (位传输率、数字带宽)：数据传输速率
    - $比特率 = 波特率 * log_2 n$，$C = B * log_2 n$，n为信号呈现的个数，为2的整数倍。
- 格子架编码调制TCM
    - 为了降低高速调制错误，在每个样本中采用一些额外的位用作纠错，剩下的位才用来传输数据。
    - V.32 调制标准
        - 波特率 —— 2400，采用 QAM-32，每个码元传输 5 个比特。但其中的 1 个比特用来做奇偶校验，只有4个比特用来传输数据。$数据传输速率 = 2400 * 4 = 9600\ bps$。

2.4.3 课程总结

- 什么时候需要调制?
- 调幅、调相、调频的工作原理分别是什么?
- 信号星座是什么?
- 什么是波特率/采样率/符号率?
- 例题

例 一个采用了QAM-64调制的系统，如果波特率是2400，试问，（1）系统能够提供的数字带宽是多少？（2）如果每个采样中的1位用于检错，这种情况下，系统能够提供的数字带宽是多少？

- 解：（1）根据公式 $C = B \times \log_2 n$

  B=2400；$n=\log_2 64=6$（每个采样传输6比特）

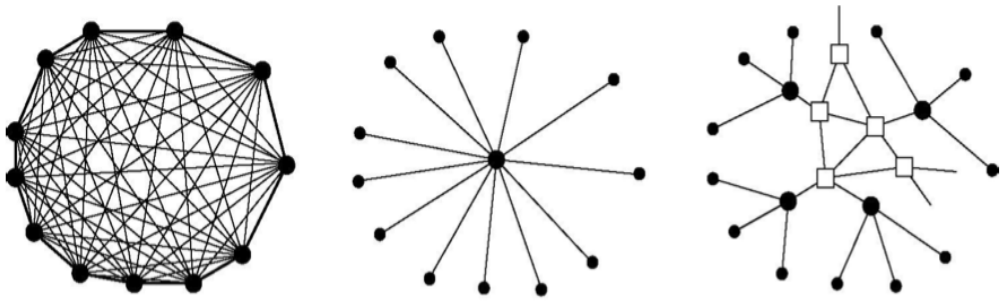
  所以：数字带宽（比特率）=2400×6=14400（bps）

  （2）每个采样传输的不再是6比特，而只有5比特，所以，

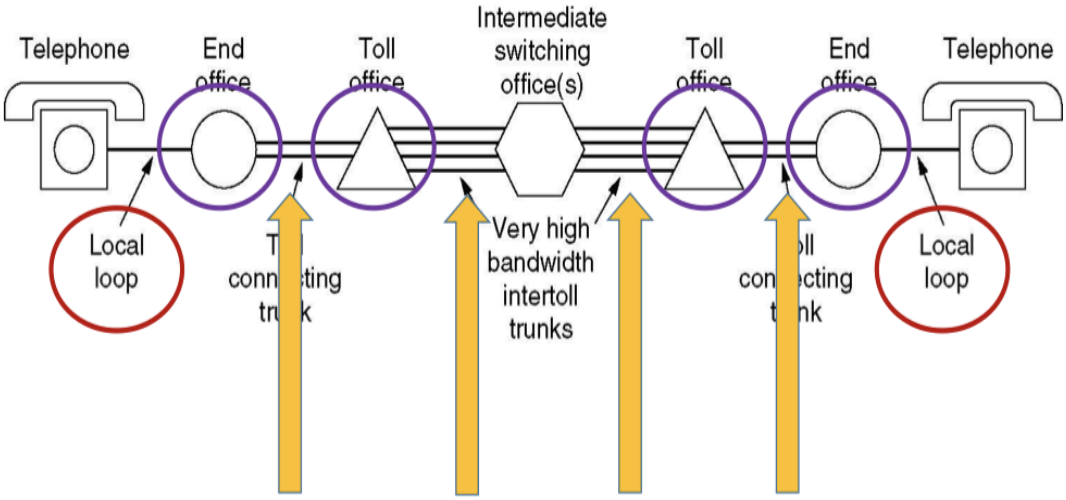
  数字带宽（比特率）=2400×5=12000（bps）

2.5 公共交换电话网络 (PSTN)

2.5.1 PSTN 概述

- PSTN (Public Switched Telephone Network) 与计算机网络
  - PSTN的任务：传递人类的语音
  - 计算机网络的目标：传输数据
- PSTN 发展历史
  - "全联通网 -> 中心交换网 -> 层次交换网"，"人工交换 -> 全自动交换" 的演化过程。
  - 

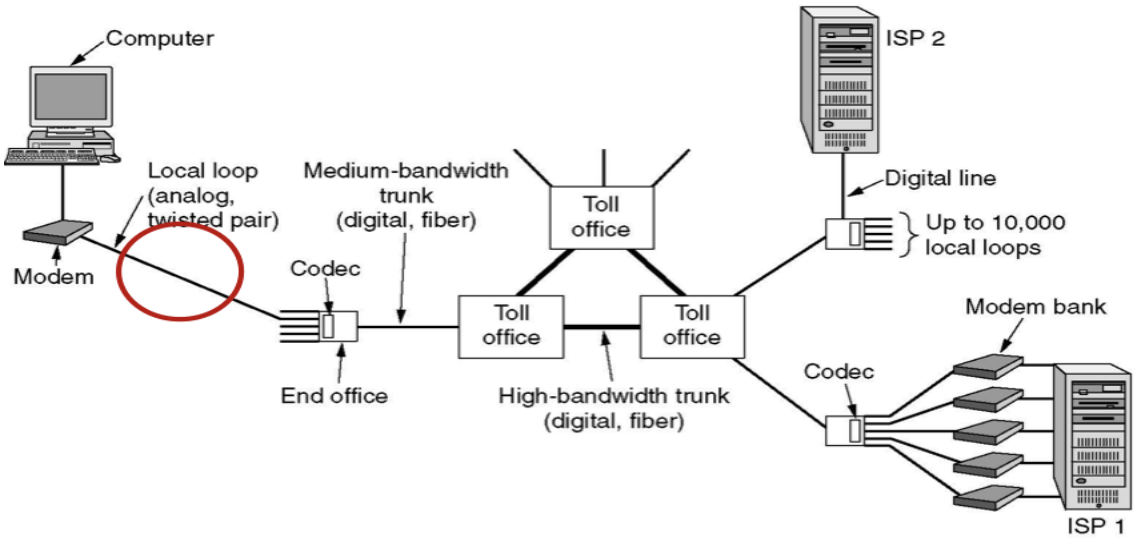
- PSTN 的结构

- 

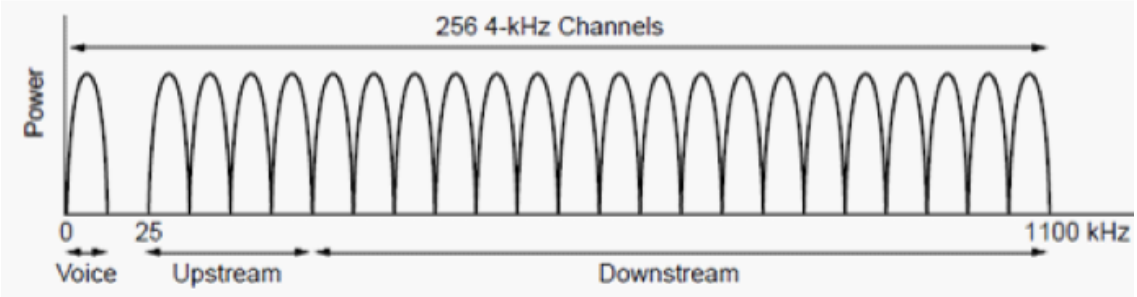
  - 呼叫方的话音 -(本地回路)-> 端局 -(干线)-> 上级交换局 -(一级..一级)-> 对方端局 -> 被叫方
  - 通话双方之间搭建一根实际的点到点的物理通路。
- 构成 PSTN 的三大部分
  - 本地回路 (Local loops)
    - 模拟线路，通常用三类双绞线实现。连接端局和千家万户或业务部门。
  - 干线 (Trunks)
    - 用数字光纤实现，用来连接交换局与端局。
  - 交换局 (Switching offices，包括端局)
    - 语音接驳干线的场所。

2.5.2 本地回路

- 传输模拟信号。
- 模拟信号如何搭载计算机产生的数字信号?
  - 使用调制解调器，对数字信号与模拟信号进行转化传输。



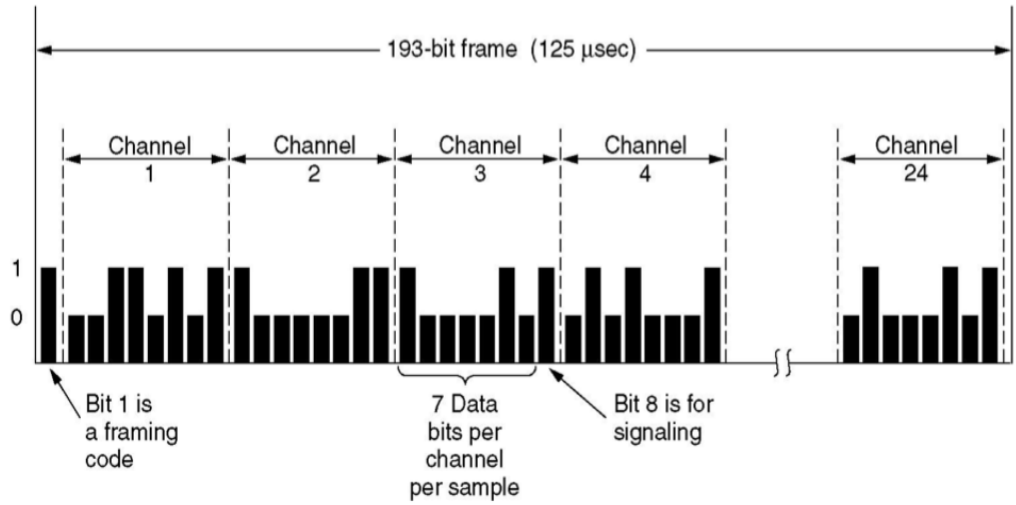
  - 调制解调器 (猫-Modem)，带宽为56k。
  - 56k的调制解调器
    - 采用V.90标准，电话线路的频率约为 4000 Hz (300~3400Hz)

        - $_{采样率} = 2 * 4000 = 8000\ sample/s$
        - 每个码元传输 8 个比特，其中的 1 个比特用于控制错误，所以 $_{传输数据速率} = 8000 * 7 = 56,000\ bps$ 。(毛速率为64 kbps)
    - 连接调制解调器的带宽为限制在了4K之内，因为电话的带宽就在4K。
- xDSL
    - 取消电话系统中的滤波器之后，xDSL可以使用本地回路的全部物理带宽，即全部的 1.1M 带宽。
    - 

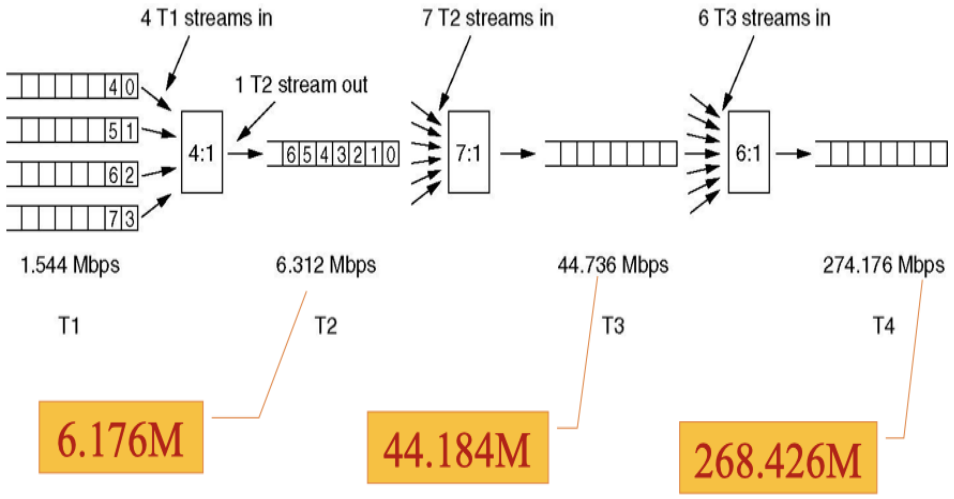
    - ADSL (非对称数字用户线)
        - 在1999年获得批准，称为G.dmt，也称为宽带服务，允许8M的上行带宽，1M的下行带宽。之后不断演进，下行带宽达到了24M。非对称指上行和下行带宽不同。
        - 将 1.1M 频宽分成了256根信道，每根约 4K。第一根 4k 信道仍然用于做语音通话。信道1~5空闲，防止语音信号和数据信号相互干扰。
        - 在剩下的250条信道中，一条用于上行流控制，另一条用于下行流控制，其它的信道全部用于用户数据。
    - 光纤到户 (FTTH - Fiber To The Home)
        - 起因：本地回路通常使用3类UTP来承载信号，但是3类UTP的物体特性限制了带宽的增长上限。因此将本地回路替换为容量更大的光纤，来增加数据传输速率。
        - "光进铜退 - 最后一英里革命"：更大的带宽，更高的安全性和可靠性。

2.5.3 干线

- 干线：多路复用 (时分多路复用)
    - 用光纤承载信号，连接交换局 (包括端局)。
- 编解码器 (codec)
    - 端局中的一个设备。
    - 作用：模拟信号数字化、数字信号模拟化。
    - 技术：脉码调制 (PCM)
        - 将模拟信号数字化的技术。构成了现代 PSTN 的核心。
        - 基于 PCM 的 TDM 在干线 (中继线) 上传输多路电话语音，每 125 us 发送一个语音样本。
        - 用于北美和日本的 T1 载波，可以处理 24 路信号的复用。一个 TDM 复用帧有193比特。

- 

  - 每一路话音有8个比特，一共24路话音，再加上一个用于控制的比特，一共193个比特。$24 * 8 = 192\ bits + 1\ bit\ for\ framing = 193\ bits/frame$
  - 话音信道的采样率是 8000次/s，传递 TDM 复用帧的时间间隔为$1/8000s = 125\ us$，$T1_{线路的传输速率} = 193\ bits/0.000125\ s = 1.544Mbps$。
  - TDM允许更高级别的复用。



- SONET / SDH (同步光网络)
  - 起因：在光纤的早期阶段，每个电话公司都有各自的光纤TDM系统。因此随着光纤的发展和普及，标准化需求非常迫切。
  - SONET是美国ANSIS制定的在光介质上进行同步数据传输的标准。
  - SDH是国际标准组织 (ITU) 制定的在光介质上进行同步数据传输的标准，与SONET几乎无差异。
  - SONET的 4 个设计目标
    - 不同的承运商可协同工作
    - 统一美国、欧洲和日本的数字系统
    - 提供一种复用多数字信道的方法
    - 提供操作、维护和管理。

2.5.4 交换局

- PSTN中一共使用了两种交换技术。
  - 电路交换 (Circuit Switching)
    - 应用于传统的电话系统中。
    - 要求在正式通话之前，必须建立一条端到端的通路，而且是物理的实际的通路。
    - 该通路称为连接。当连接建立好之后，话音在沿着通道按顺序送达对方。
    - 数据传输完之后，拆除连接。
    - 建立连接 - 传数据 - 拆除连接
  - 包交换 (分组交换 - Packet Switching)
    - 应用于IP电话业务中。
    - 限制 包 / 分组 大小。 -> 允许 包 / 分组 存储在交换局的内存中。
    - 每个包都携带着目的地址、信息。 -> 每个包要独立寻径。
    - 乱序到达。 -> 先发的可能后到。
  - 两种交换方式的比较
    - 带宽的分配方式不同：包交换按需分配，电路交换是提前分配 (分配之后即被独占)。
    - 容错能力的不同：分组交换容错能力更强 (独立寻径，不会因为某一节点失效而瘫痪)
    - 交换顺序的不同：包交换乱序到达，电路交换按序到达。
    - 收费方法不同：包交换按流量收费，电路交换按照时间来收费。
    - 呼叫建立、专用物理路径、每个包遵循相同路由、包按序到达、交换机崩溃是否指明、可用带宽、可能拥塞时间、潜在浪费带宽、存储-转发传输、收费。

Item	Circuit-switched	Packet-switched
Call setup	Required	Not needed
Dedicated physical path	Yes	No
Each packet follows the same route	Yes	No
Packets arrive in order	Yes	No
Is a switch crash fatal	Yes	No
Bandwidth available	Fixed	Dynamic
When can congestion occur	At setup time	On every packet
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Transparency	Yes	No
Charging	Per minute	Per packet

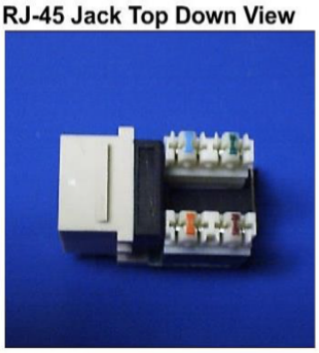
2.5.5 课程总结

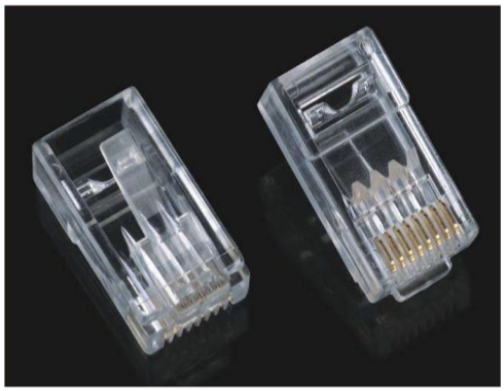
- 什么是公共电话网PSTN?
- PSTN有哪三部分构成?
- 调制解调器在哪里？完成什么功能?
- 调制解调器的 64K 是怎样计算出来的?
- Codec在哪里？完成什么功能?
- T1 基本复用帧中有多少位开销?
- T1线路的总带宽是多少？怎样计算出来的?

2.6 物理层设备

2.6.1 物理层部件 / 设备分类

- 被动 (无源) 部件/设备
  - 插座、插头、电源、电缆等。
  - 被动设备中最重要的是RJ45的插座和水晶头。
  -




- 主动 (有源) 部件/设备
  - 转发器、中继器和集线器等。

2.6.2 收发器 (Transceiver)

- 名称由来
  - Transceiver = Transmitter+Receiver
  - 也称为 MAU：Media Attachment Unit
- 功能：将一种形式的信号转变为另一种信号
- 早期是一个外设，现在是网卡上的部件，是计算机和传输介质的连接部件，负责收发信号。

2.6.3 中继器 (Reapter)

- 功能：再生信号 (去除噪声、放大信号)
  - 让线缆可以延伸得更远，突破 UTP 100米的传输距离限制。
- 注意：中继器不能过滤流量。
  - 过滤 (Filter)：是指设备以一定的特征来屏蔽网络流量，并根据标准确定将流量转发或丢弃。

2.6.4 集线器 (Hub)

- 定义：多端口的中继器
- 功能：再生信号，即去噪和放大。
    - 集线器上的多端口，允许很多设备连接上来。(工作站、主机)
    - 早期以太网中，集线器作为星型拓扑的中心。
    - 依然不能过滤流量。
    - 集线器收到信号之后，会直接将信号进行广播。
        - 广播 (泛洪)：从除了来的那个端口外的所有其它端口转发出去。
    -
- 冲突：信号的碰撞
    - 当使用物理层设备时，更多的用户争抢共享资源，容易导致冲突，即两个或多个信号碰撞在一起。
    - 冲突的电气表现：电压异常、数据被破坏。冲突了的数据需要重传。
    - 冲突域
        - 定义：数据包产生和冲突的网络区域，即指共享介质的区域。
        - 冲突域越大，产生冲突的可能性越大，网络性能下降。
        - 使用了中继器和集线器之后，冲突域会扩大，网络性能会下降。因为中继器使得更远更多的用户可以来一起争抢资源。

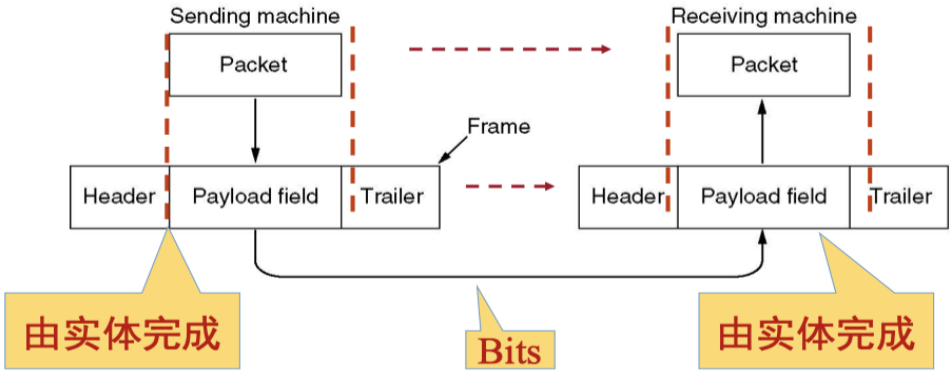
2.6.5 课程总结

- 物理层 (第一层) 的设备都是傻瓜设备，不具备过滤流量的智能功能。
- 物理层设备的使用，增大了冲突域，降低了网络的性能。
- 现今，已经很少再使用中继器 (光中继器除外) 和集线器了。
- 问题
    - 中继器的作用是什么?
    - 集线器的作用是什么?
    - 作为曾经的星型拓扑中心的集线器会消失吗?
    - 什么是冲突?
    - 什么是冲突域?

第三章 数据链路层

3.1 数据链路层概述

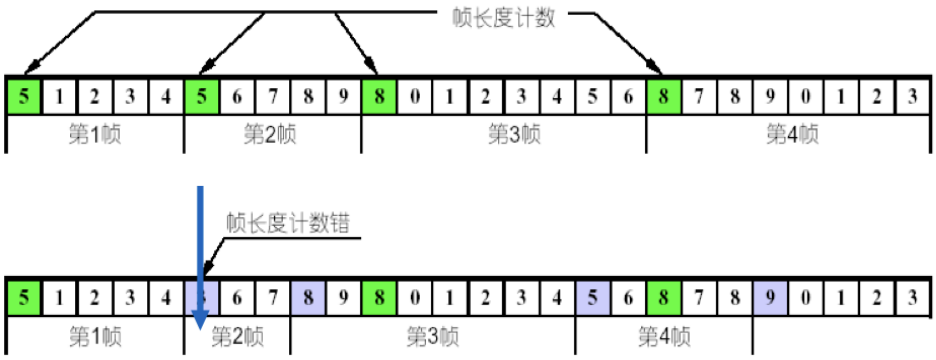
3.1.1 数据链路层功能与帧

- 数据链路层功能：保证数据传输的有效、可靠
  - 差错检测和控制
  - 流量控制
    - 基于速率
    - 基于反馈 (常采用)：接收方告诉发送方自己的处理能力水平，发送方再根据接收方的反馈发出对应的流量。
- 数据链路层的 PDU —— 帧
  - 帧 = 帧头 + 载荷 + 帧尾
  - 

  - 帧头：定位所需的物理地址信息。
  - 帧尾：校验和作为帧的校验。
- 成帧：将物理层的位流分散到离散的帧中。
  - 四个成帧方法 —— 字符计数法、字节填充的标志字节法、比特填充的标志比特法、物理层编码违例法

3.1.2 字符计数法 (不用)

- 发送方：在每个帧头部的第一个字段，标识该帧的长度，即一共有多少字符数。
- 接收方：根据帧头部第一个字段，得知该帧在何时结束。
- 优点
  - 简单，易实现。
- 缺点
  - 一旦出错，无法恢复，因此极少使用。



3.1.3 字节填充的标志字节法 (常用)

- 考虑了重新同步问题，每一帧采用一个特殊的字节作帧界，即帧的开始和结束。
- 特殊字节即为标记字节 (Flag Byte)，常用于 PPP (点对点) 协议中。
- 问题 1：传输的数据中有标记字节。
- 解决 1：转义符，在特殊数据前加转义符。(扫描到转义符时，会删除转义符并取后一个字符)

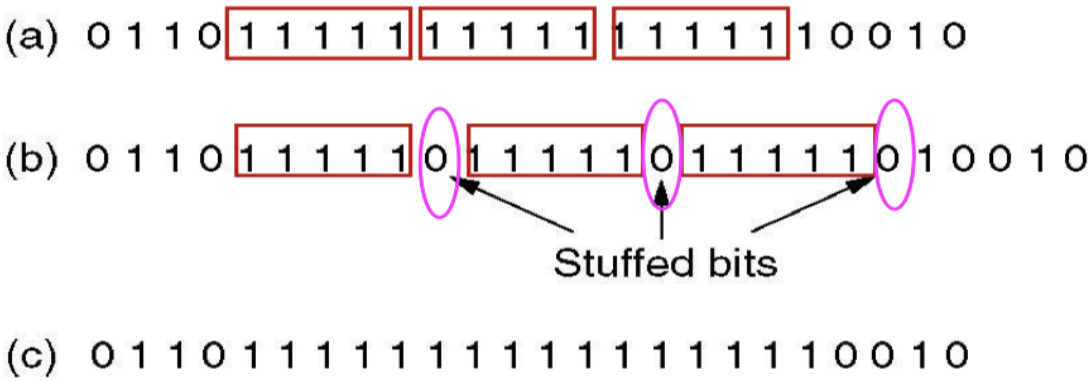
- 问题 2：任意比特数的帧不适用，必须是 8 位整数倍。
- 解决 2：比特标记法。(比特填充的标志比特法)

3.1.4 比特填充的标志比特法 (常用)

- 方法
  - 帧标记
    - 将所有需要传输的数据以比特位一字排开，并以特殊的位模式 01111110 作为帧标志，即一个帧的开始 (同时标志前一个帧的结束)
  - 数据中出现帧标记
    - 当帧内容中出现一个与帧标志相同的位串 01111110 时，则在 5 个 1 后插入一个 0，即变成 01111101，接收方将自动删除第 5 个 1 后的 0。这种方法为位填充法 (零比特填充法)，也称为透明传输。
  - 数据恢复
    - 如果由于干扰，一个帧没有正确接收，则可扫描接收串，一旦扫描到 01111110，即新

的一帧从此开始，可以重新进行同步。

- 例子



- 优点
  - 传输任意比特数的帧
  - 传输效率更高

3.1.5 物理层编码违例法

- 特点：将冗余信号作为帧界。
- 例子
  - 在4B/5B编码中，4B被映射为5B传输，32个模式中，只用到16个，剩下的可以用帧界。
  - 在曼彻斯特编码中，编码过程只用到了高电平到低电平的跳变，以及低电平到高电平的跳变。而低到低与高到高并没有使用，因此我们可以将这两种跳变作为帧界。
- 优点：用冗余信号作为帧界，避免混淆，也不需要填充，传输效率高。

3.1.6 课程总结

- 数据链路层在哪里？(物理层之上，网络层之下)
- 数据链路层的主要功能是什么？(差错检测和控制、流量控制)
- 字符计数法成帧的主要缺点是什么？(一旦出错，无法恢复)
- 字节填充的标记字节法的基本原理是什么？
- 比特填充的标记比特法的基本原理是什么？

3.2 差错处理概述

3.2.1 差错处理概述

- 差错类型
  - 单个错误：分散在各块数据中
  - 突发错误：集中于一个数据块，整个块都是错误

- 差错处理
  - 纠错
    - 纠错码 (前向纠错技术)：发现错误，从错误中恢复出正确的来。有线网络中极少使用，主要用于无线网络中。
  - 检错
    - 检错码：只能发现错误，不能从错误中恢复，但可采用重传恢复。局域网中主要采用检错码。
- 奇偶校验
  - 核心思想：在数据位之后加一位校验位
  - 奇校验：校验位的添加，确保 数据位+校验位 一共有奇数个"1"。
  - 偶校验：校验位的添加，确保 数据位+校验位 一共有偶数个"1"。

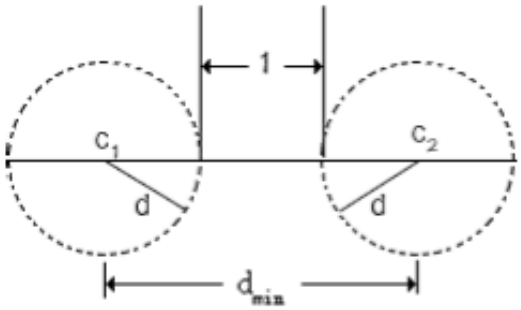
3.2.2 海明距离

- 码字：包含数据位和校验位的 n 位单元 (模式)
- 定义
  - 两个码字的海明距离：两个码字之间不同位的数目。
  - 全部码字的海明距离：全部码字中任意两个码字之间海明距离的最小值。
- 海明距离意义：如果海明距离为d，则一个码字需要发生d个1位错误才能变成另外一个码字。
- **海明距离与检错的关系**
  - 海明距离为d+1的编码能检测出d位差错
  - 例如

例：偶校验

- **一个系统要传输的原始码字：**
  - 00、01、10、11
- **编码后的偶校验码字（合法/正确）是：　（海明距离2）**
  - 000、011、101、110
- **如果接收方收到一个码字001，这是一个非法码字，所以，出错了。**
- **如果系统收到000，这是一个合法码字，正确。不排除这是110跳变2后的错误码字。**

- **海明距离与纠错的关系**
  - 海明距离为 2d+1 的编码，能纠正 d 位差错。因为此时任何一个码字有d位发生差错，它仍然距离原来的码字距离最近，可以直接恢复为该码。

- 

- 海明距离越大，纠错能力越强，检错能力也越强。但是一个系统中海明距离增加，合法码字就减少，传输效率降低。

3.2.3 课程总结

- 什么是码字?
- 什么是两个码字的海明距离？什么是全部码字的海明距离?
- 差错的类型有哪两种?
- 海明距离跟检错有什么关系?
- 海明距离跟纠错有什么关系?

3.3 纠 1 位错的海明码

3.3.1 数据位与冗余位

- 码字长度 = n，数据位 = m，冗余位 = r，要实现纠正 1 位错，即海明距离为 3，则有下述公式。
- $(n+1)2^m \leq 2^n, (m+r+1) \leq 2^r$
- 公式推导
  - $n = m + r$ ，合法码字个数为 2^m ，全部码字个数为 2^n 。
  - 要实现每个合法码字发生一位错误之后都是一个新的码字，则 $(n+1)2^m \leq 2^n$ 。

3.3.2 海明纠错码

- 海明纠错码特点
  - 每一个码字，从左到右编号，1~n。
  - 校验位：编号为 2 的幂的位，如1、2、4、8、16等。
    - 校验位可以自己设置。依据为包括自身在内的一些位的集合的奇偶值。
  - 数据位：3、5、7、9、10、11等，数据位直接填入我们要传输的数据。
- 发送方发送海明纠错码

- 校验位的值为该校验位对应数据集合的偶校验的值，对应数据集合为所有二进制拆分下包含该数字的数据位。
  - 例如，m = 7，r = 4。
  -

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
$1=2^0$	√		√		√		√		√		√
$2=2^1$		√	√			√	√			√	√
$4=2^2$				√	√	√	√				
$8=2^3$								√	√	√	√

  -

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
信息码	-	-	1	-	0	0	1	-	0	0	0
检验位	0	0	-	1	-	-	-	0	-	-	-
海明码	0	0	1	1	0	0	1	0	0	0	0

使用偶校验，一个校验集合里的1的个数是偶数

- 接收方纠错
  - 令差错计数器 $counter = 0$ 。依次检查每一位校验位，如果某一位校验位的值出错，则 $counter = counter + \text{该校验位的编号}$ 。
  - 最后，如果 $counter = 0$，则无差错。若 $counter \neq 0$，则出错，出错位为 $counter$ 值所对应的位，因为某一位发生错误会影响到该位拆成二进制形式下的每一个 "1" 的位置。

3.3.3 课程总结

- 16 位数据位，需要纠 1 位错，冗余位应该为多少?
- 纠 1 位错的海明码，发送方怎样确定校验位?
- 纠 1 位错的海明码，接收方怎样判定收到的码字是否正确？如果出错了，怎么纠正成正确的码字?

3.4 检错码

3.4.1 奇偶校验码

- 偶校验，保证码字中 "1" 的个数为偶数。
- 奇校验，保证码字中 "1" 的个数为奇数。
- 如果偶数个位同时发生错误，则奇偶校验会失效，因此奇偶校验正确判断的概率为 50%。

3.4.2 互联网校验和

- 按照 N 位码字，进行模 2 加运算。
- 发送方将运算结果追加到报文的尾部。
- 例如，互联网 16 位补码校验和。
  - **发送方**：将待校验的相邻字节成对组成16比特的整数一行，按行从低位开始计算其模 2 和；并将结果按位取反码，作为校验和取值。
  - **接收方**：检查校验和时，将所有字节，包括校验和，进行相加并求二进制反码。若结果为全1，则无误，否则传输错误。
  - **注意**：若某列的模 2 和有溢出，则向高位进位，如果高位产生进位，循环向低位进位。
-

7 单选 (2分) 假设使用Internet校验和（4位字）来发送一个消息1001110010100011.试问校验和的值是什么？ 得分/总分

A. 0011

B. 0100

C. 1011 ✔2.00/2.00

D. 0010

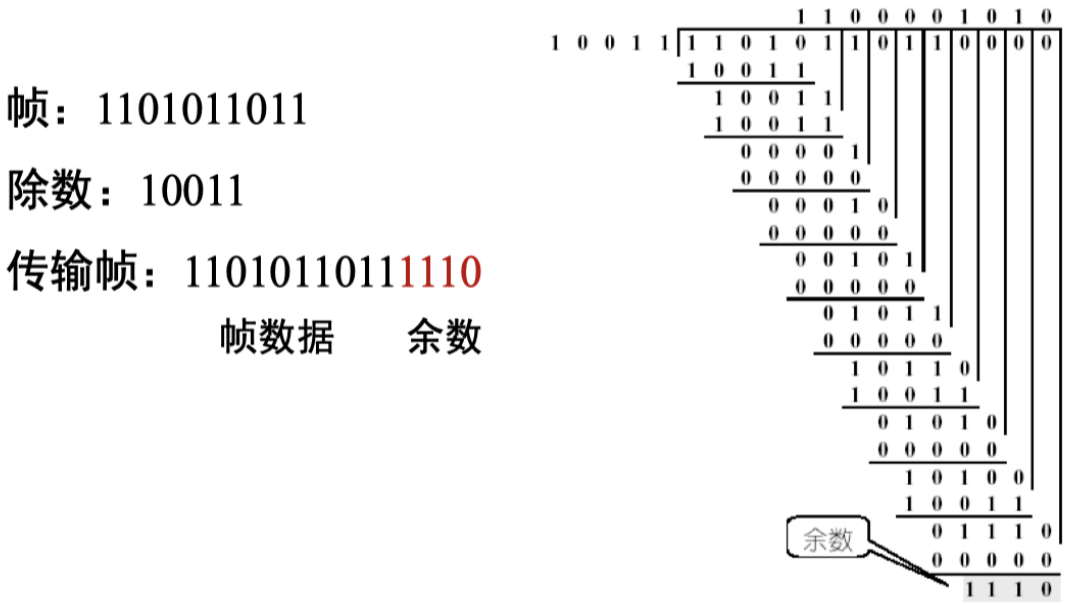
正确答案：C 你选对了

解析： C、按照4位1行进行排列，按列相加，低位开始相加（模2加）；产生进位，向高位进，最高位产生进位，向最低位循环进位。所得结果再取反码，作为校验和。

3.4.3 循环冗余校验码 (CRC)

- 工作原理
  - k 位的帧 $= k-1$ 次的多项式
    - 例如，1011001，可以看成是 $x^6+x^4+x^3+x^0$ 。
  - 双方约定一个生成多项式 $G(x)$，$G(x)$ 为 r 阶，r 就是冗余位。
  - m 位帧的多项式，m > r，M(x) > G(x)。
    - $\frac{x^r M(x)}{G(x)} = Q(x) + R(x)$，$R(x)$ 为余数，$Q(x)$ 为商。
    - $\frac{x^r M(x) - R(x)}{G(x)} = Q(x)$，消除余数。
    - $x^r M(x) - R(x)$ 即为 CRC 校验码。
    - 注意：此处的除法以及加减法过程均为异或运算，与实际中的运算不同。
  - 接收方检验
    - $\frac{CRC_{校验码}}{G(x)}$ 的余数为0，则传输无误，否则传输错误。
- 例如

- 帧：$1101011011\ (m=10)$
- $M(x)=x^9+x^8+x^6+x^4+x^3+x+1$
- 令 $G(x)=x^4+x+1\ (r=4阶)$，即 10011。
- $T(x)=x^4M(x)=x^{13}+x^{12}+x^{10}+x^8+x^7+x^5+x^4$ (相当于在原码字后加 r 个 0)
-



- 计算 $\frac{11010110110000}{10011}$ 得到余数 1110，$11010110110000-1110=11010110111110$，因此编码后 CRC 码 $=11010110111110$ 。
  - 当码字到达接收方时。如果CRC码在接收端能被 10011 整除则说明接受正确，否则即可检测到出错。

- 生成多项式 $G(x)$ 国际标准
  - CRC-12：$x^{12}+x^{11}+x^3+x^2+x^1+1$
  - CRC-16：$x^{16}+x^15+x^2+1$
  - CRC-CCITT：$x^16+x^12+x^5+1$，用于长度字符为 8 位。
  - CRC-32，用于以太网计算循环冗余校验。
    - $G(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^3+x+1$
- 练习
  - $G(x)=x^3+x^2+1$，待传送的原始码字分别为 1111 和 1100，求 CRC 编码之后的码字分别是什么？
    - $G(x)$ 对应 1101，$\frac{1111_{左移3位}=1111000}{1101}=1011+111_{(余)}$ ，CRC

$$
\begin{array}{r}
1011 \\
1101\overline{)\,1111000} \\
\underline{1101} \\
1000 \\
\underline{1101} \\
1010 \\
\underline{1101} \\
111
\end{array}
$$

$= 1111000 - 111 = 1111111$

- $G(x)$ 对应 1101，$\frac{1100_{\text{左移}3\text{位}}=1100000}{1101} = 1001 + 101(_{\text{余}})$，CRC

$$
\begin{array}{r}
1001 \\
1101\overline{)\,1100000} \\
\underline{1101} \\
1000 \\
\underline{1101} \\
101
\end{array}
$$

$= 1100000 - 101 = 1100101$

3.4.4 课程总结

- 什么是检错码?
- 采用循环冗余校验码，发送方怎么做?
- 采用循环冗余检验码，接收方怎么做?
- 循环冗余校验码，能够检查出多少位错误?

3.5 基本数据链路协议 (1-3)

3.5.1 前提假设

- 三个协议都是单工协议，即在某一时刻信道是单向的。
- 前提假设
  - 物理层、数据链路层和网络层各自是独立的处理进程。
  - 机器 A 希望向 B 发送的是一个可靠的、面向连接的长数据流。
  - 假设机器不会崩溃，即使崩溃我们也不会处理因崩溃而产生的错误。
  - 从网络层拿到的数据是纯的数据，即这个数据是作为纯数据装到帧的载荷之中。
- 三个协议的共同头文件

```
#define MAX_PKT 1024                    /* 定义了包的字节大小 */
```

```c
#define inc(k) if(k < MAX_SEQ) k = k + 1; else k = 0;

typedef enum {false, true} boolean;       /* boolean类型 */
typedef enum {frmae_arrival, cksum_err, timeout} event_type;
/* 事件类型(帧到达、校验和错误、定时器超时) */
typedef unsigned int seq_nr;              /* sequence or ack numbers */
typedef struct {
  unsigned char data[MAX_PKT];
} packet;                                     /* packet的定义 */
typedef enum {data, ack, nak} frame_kind;   /* frame_kind的定义 */

typedef struct {
  frame_kind kind;              /* 帧的类型 */
  seq_nr seq;                   /* sequence number (帧的序列号) */
  seq_nr ack;                   /* acknowledgement number (帧的确认号) */
  packet info;                  /* the network layer packet (帧的载荷) */
} frame;

void from_network_layer(packet *p);       /* 从网络层拿包 */
void to_physical_layer(frame *r);         /* 将帧传到物理层 */

void from_physical_layer(frame *r);       /* 从物理层拿帧 */
void to_network_layer(packet *p);         /* 将包传到网络层 */

void wait_for_event(event_type *event);   /* 等待某个事件发生 */

void start_timer(seq_nr k);                 /* 重传定时器开始 */
void stop_timer(seq_nr k);                  /* 重传定时器停止 */

void start_ack_timer(void);                 /* 稍带确认定时器开始 */
void stop_ack_timer(void);                  /* 捎带确认定时器停止 */
```

3.5.2 无限制的单工协议 (协议 1)

- 前提假设 (乌托邦环境)
  - 数据单向传送
  - 收发双方的网络层都处于就绪状态 (随时待命)
  - 对帧的处理时间忽略不计 (瞬间完成)
  - 可用的缓存空间无穷大 (无限空间)
  - 数据链路层之间的信道不会损坏并且不丢帧 (完美通道)
- 发送方

```c
void sender1(void){
  frame s;
  packet buffer;
  while(true){
    from_network_layer(&buffer);    /* 从网络层拿包,拿到之后直接放入buffer中 */
    s.info = buffer;                 /* 组成了一个帧 */
    to_physical_layer(&s);          /* 把帧传给物理层 */
  }
}
```

- 接收方

```c
void receiver1(void){               /* 只要帧到达，就开始收帧(解封装) */
  frame r;
  event_type event;
  while(true){
    wait_for_event(&event);         /* 等待一个事件的到来(帧到达) */
    from_physical_layer(&r);        /* 从物理层拿比特流 */
    to_network_layer(&r.info);      /* 将帧中的包传给网络层 */
  }
}
```

3.5.3 单工停-等协议 (协议 2)

- 改进
  - 取消了协议 1 中接收方允许无限量接受的假设。因此接收方有可能被大量涌入的数据所淹没。
- 解决方案
  - 接收方收到一个数据，如果它有处理能力的话，它会回发一个哑帧给对方。
  - 只要发送方收到了哑帧，它就知道接收方已经做好了接收数据的准备，因此可以继续发下一帧的数据。
- 半双工协议
  - 此时数据传输已经是双向了，严格的说这应该是一个半双工协议。
- 发送方

```c
void sender2(void){
  frame s;
  packet buffer;
  event_type event;                /* 变量取值为帧到达 */
  while(true){
    from_network_layer(&buffer);  /* 从网络层拿包,拿到之后直接放入buffer中 */
    s.info = buffer;               /* 组成了一个帧 */
    to_physical_layer(&s);        /* 把帧传给物理层 */
    wait_for_event(&event);      /* 等待哑帧 */
  }
}
```

- 接收方

```c
void receiver2(void){               /* 只要帧到达，就开始收帧(解封装) */
  frame r;
  event_type event;
  while(true){
    wait_for_event(&event);         /* 等待一个事件的到来(帧到达) */
    from_physical_layer(&r);        /* 从物理层拿比特流 */
    to_network_layer(&r.info);      /* 将帧中的包传给网络层 */
    to_physical_layer(&s);         /* 向发送方发送一个哑帧 */
  }
}
```

3.5.4 有噪声信道的单工协议 (协议 3)

- 改进
  - 在协议 2 基础上取消了帧传输不会发生错误的假设，即允许信道中存在噪声。
  - 因此会出现如下问题：
    - 接收方收到帧之后，发现该帧无法通过校验，即发生了错误。
- 解决方
  - 增加了接收方对正确帧的确认。
    - 任何时候接收方收到的帧通过了校验，接收方才会向发送方发一个确认帧。
    - 发送方只有当收到了确认之后，才会去发下一帧。
  - 发送方增加计时器。
    - 目的：避免数据帧或确认帧丢失后，发送方持续等待的情况。
    - 发送方发出数据帧的同时启动定时器，如果定时器超时了仍未收到确认帧，发送方会重新发送数据帧，直到接收方回复确认帧。
  - 给每一个帧一个独一无二的序号。
    - 目的：数据帧未到但定时器超时，接收方同时收到了数据帧以及重传帧，帧序号用于区别重传帧。
    - 帧序号还能用于对帧的重排与重组。

- 肯定确认重传机制 (PAR)
  - 名称
    - 肯定确认重传机制-PAR (Positive Acknowledgement with Retransmission)
    - 也称自动重传请求 - ARQ (Automatic Repeat ReQuest)
  - 具体过程
    - 发送方每发出一个帧，它都启动一个重传定时器，期望在超时之前收到对方的确认。如果超时之前收到了就拆除定时器，否则发方重传数据帧并重新设置定时器。
  - 发送方

```c
void sender3(void){
  seq_nr next_frame_to_send;
  frame s;
  packet buffer;
  event_type event;
  next_frame_to_send = 0;
  from_network_layer(&buffer);
  while(true){
    s.info = buffer;
    s.seq = next_frame_to_send;
    to_physical_layer(&s);
    start_timer(s.seq);                    /* 发出帧之后，启动定时器 */
    wait_for_event(&event);
    if(event == frame_arrival){
      from_physical_layer(&s);
      if(s.ack == next_frame_to_send){
        stop_timer(s.ack);                 /* 收到确认后，拆除定时器 */
        from_network_layer(&buffer);
        inc(next_frame_to_send);          /* 模2计数 */
      }
    }
  }
}
```

  - 接收方

```c
void receiver3(void){
  seq_nr frame_expected;
  frame r,s;
  event_type event;
  frame_expected = 0;
  while(true){
    wait_for_event(&event);         /* 等待一个事件的到来(帧到达) */
    if(event == frame_arrival){
      from_physical_layer(&r);
      if(r.seq == frame_expected){
        to_network_layer(&r.info);
```

```
        inc(frame_expected);          /* 模2计数 */
      }
      s.ack = 1 - frame_expected;    /* 对收到的帧做一个确认 */
      to_physical_layer(&s);         /* 将确认帧发送给对方 */
    }
  }
}
```

- 提高传输效率
  - 全双工：每个用户既是发方又是收方，两方可以互相来发数据。
  - 捎带确认：接收方将确认帧放到一个将要外发的数据帧中，搭便车。
    - 捎带确认要启动一个捎带确认的定时器，避免等待时间过长，发送方重传。
  - 批发数据 (管道化的技术)
    - 等待第一个确认帧的时候继续发后续的数据帧。等后续数据帧发出去之后，第一个确认帧就到达了。

3.5.5 课程总结

- 协议 1： 乌托邦的单工协议
- 协议 2： 做了简单的流量控制，防止接收方被数据所淹没
- 协议 3： 解决噪声信道带来的错误，引出肯定确认技术。
- 问题
  - 如何解决接收方被大量数据所淹没？(哑帧)
  - 肯定确认重传技术是怎样工作的？
  - 重传定时器什么时候启动，什么时候拆除？
  - 接收方为什么会收到重复帧？
  - 如果发送的帧丢失了，意味着发方永远收不到这帧的确认，发方会死等确认帧 (永远不可能到来) 的到来吗？
  - 什么是捎带确认？

3.6 滑动窗口协议 (协议4)

3.6.1 滑动窗口基本概念

- 每个待发送帧被赋予一个序列号seq
  - seq的取值范围是 0~2^n-1 (n 位字段)
- 建立缓冲区
  - 发送窗口：缓存已发送、待确认的帧
    - 顺序接收来自网络层的分组，成帧，赋予序列号
    - 最多保存 W 个已经发送、等待确认的帧

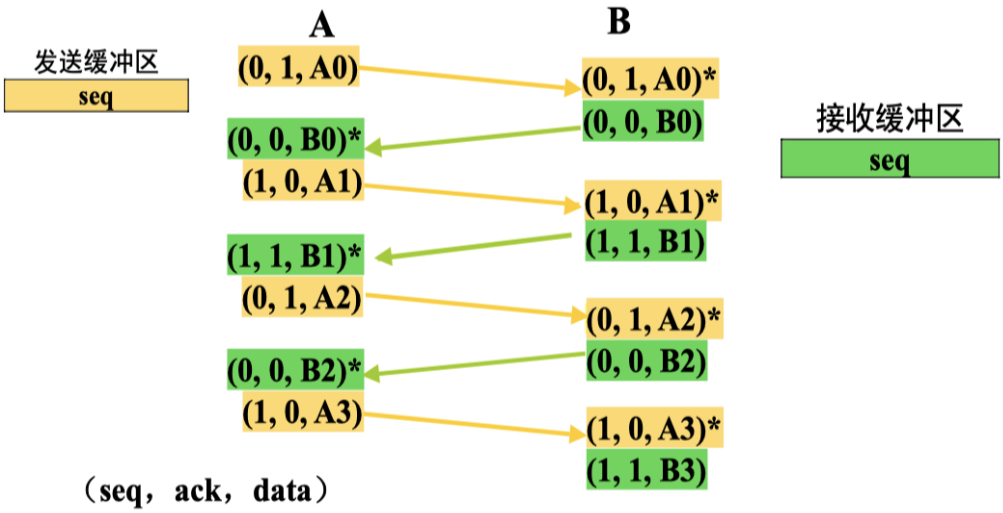
        - 窗口达到最大值 W 时强制关闭网络层
    - 接收窗口：缓存期待接收的帧 (序号)
        - 对进入窗口的帧顺序提交网络层，产生确认
        - 落在窗口外的帧被丢弃

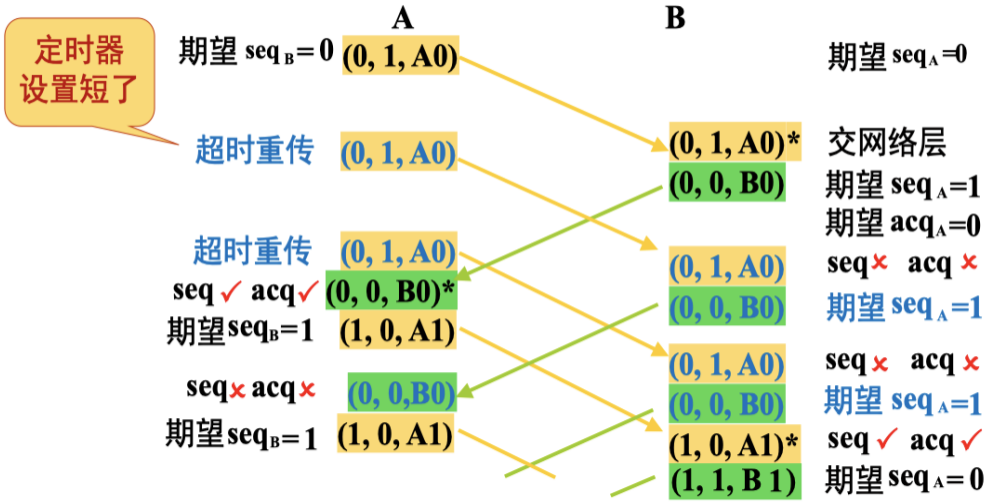
3.6.2 窗口大小为1的滑动窗口协议

- 接收窗口滑动的条件
    - 接收方收到帧之后，核对收到的帧的序列号是否是 frame_expected。如果是就接受这一帧。接受之后就向下滑动窗口，即 $\text{frame_expected} = \text{frame_expected} + 1$。
- 发送窗口滑动的条件
    - 发送方收到接收方的捎带确认之后，核对这个确认帧号是否是 next_frame_to_send。如果是，则发送方移动窗口，即 $\text{next_frame_to_send} = \text{next_frame_to_send} + 1$。
- 全双工代码实现

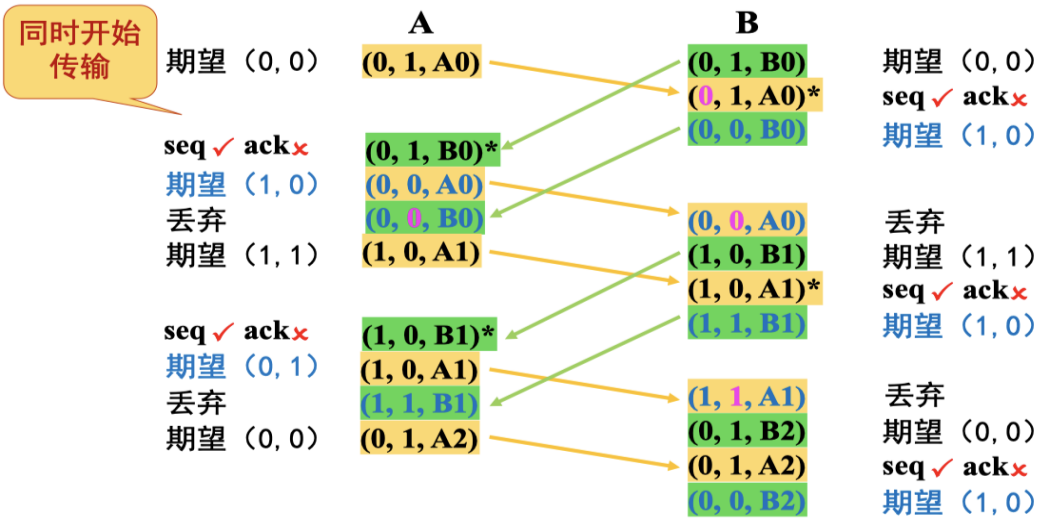
```c
void protocol4(void){
  seq_nr next_frame_to_send;          /* 即将或已经发送但还未被确认的帧序号 */
  seq_nr frame_expected;              /* 期望收到的帧序号 */
  frame r, s;
  packet buffer;
  event_type event;
  next_frame_to_send = 0;              /* 从0号帧开始发送 */
  frame_expected = 0;                  /* 接收方开始期待的是对方发来的0号帧 */
  from_network_layer(&buffer);
  s.info = buffer;
  s.seq = next_frame_to_send;         /* 初始为0，发送0号帧 */
  s.ack = 1 - frame_expected;         /* 初始为1，表示收到了1号帧 */
  to_physical_layer(&s);
  start_timer(s.seq);                  /* 启动计时器 */

  while(true){
    wait_for_event(&event);
    if(event == frame_arrival){
      from_physical_layer(&r);
      if(r.seq == frame_expected){         /* 接收方收到了期待帧 */
        to_network_layer(&r.info);
        inc(frame_expected);               /* 移动接收窗口 */
      }
      if(r.ack == next_frame_to_send){    /* 发送方收到了接收方的捎带确认 */
        stop_timer(r.ack);                 /* 停止计时器 */
        from_network_layer(&buffer);
        inc(next_frame_to_send);          /* 移动发送窗口 */
      }
    }
    s.info = buffer;
    s.seq = next_frame_to_send;           /* 发送下一帧 */
    s.ack = 1 - frame_expected;           /* ack表示收到的帧 */
```

```
    to_physical_layer(&s);
    start_timer(s.seq);                    /* 重新开始计时 */
  }
}
```

- 窗口大小为 1 的滑动窗口协议的基本工作原理
  - 特点
    - 序列号seq和确认值ack "0"、"1"交替。
    - 滑动窗口长度 $W=1$，收到确认才移动窗口。
    - 接收方保证按顺序将接收到的正确帧一次性地提交到网络层。
  - 正常工作情况
    - 

  - 异常情况一：对重复帧的差错控制
    - 即使不断发生超时重传，协议仍能正常工作，不过信道利用率会大幅下降。
    - 

  - 异常情况二：同步开始发送过程的差错控制
    - A、B同时开始发送帧，也会导致大量重复帧的出现，但协议仍能正常工作，不过大量重复帧浪费带宽。



3.6.3 协议 4 的信道利用率

- 假设：忽略接收方处理到达帧的时间
- 事实上，在低速信道上，来回时间 (RTT-the round-trip time)可能非常大，发送方在这段时间处于阻塞状态。
- 举例：卫星信道
  - 信道传输速率 $b = 50\ kbps$，传输延迟 (来回时间) $R = 500\ ms$，每帧的大小 $k = 1000\ bit$
  - 计算方式 (一)
    - 在源端发送数据帧所需时间 $T_f = k/b = 20\ ms$。
    - 从发送完毕到确认帧返回所需时间 (双程延迟) $R = 500\ ms$。
    - 从开始发送到确认返回一共所需时间 $(T_f + R) = 20 + 500 = 520\ ms$。
    - 线路利用率 $\frac{T_f}{T_f+R} = \frac{20}{520} = 3.85\%$ 。
  - 计算方式 (二)
    - 线路利用率 $\frac{k}{k+bR} = 3.85\%$
- 提高信道利用率 (窗口大小为 W)
  - $\text{信道利用率} = \frac{W*T_f}{T_f+R} = \frac{W*k}{k+bR} = 100\%$
  - 在上述卫星信道例子中，$W = 26$ 时利用率能够达到 100%。即在发送第一个帧出去，等待确认帧的过程中，继续发送26个帧。刚好发完26个帧后，第一个确认帧到达。
- 如何找到一个合适的 W 值?
  - 信道容量：一帧从发出到目的期间，信道上能够容纳的帧数量。
  - 延迟：从帧出发到它到达目的所需时间。
  - 带宽-延迟积：即 $\text{带宽延迟积} = \text{带宽} * \text{延迟} = B * D$，$W = \frac{2*BD}{k} + 1$。通常难以达到 100%，因此$W \leq \frac{2*BD}{k} + 1$。
- 例题

主机甲和主机乙之间使用后退N帧协议（GBN）传输数据，甲的发送窗口尺寸为1000，数据帧长为1000字节，信道为100Mbps，乙每收到一个数据帧立即利用一个短帧（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传输速率约为：

A. 10Mnps　　B. 20Mbps

C. 80Mbps　　D. 100Mbps

设可达到的最大传输率为 x，于是

- 1000f*1000Bpf*8=xbps*2*50ms/1000ms
- 8000000=8000+2*0.05x
- x=80Mbps
- 或者：w=1251，现在只有 1000，于是
- X=100M*（1000/1251）=80M

3　单选 (2分) 在一个有 100 ms （单程） 延迟（用D表示）的 1 Mbps 卫星链路上的发送 10,000 位的消息ARQ（自动重传请求）的最高吞吐量是多少?　　得分/总分

A. 91 kbps

B. 48 kbps

C. 50 kbps　　✕0.00/2.00

D. 45 kbps

正确答案：B　你错选为C

解析： B、ARQ中， 每 RTT（Round Trip Time，来回时间）发送最多一个数据包，，发送完成之后，需要等待对方确认到达，才能发送下一个。RTT是发送一条消息后，接收到一个简短的答复的最短时间。
题目中数据包中包含10000位的消息，发送它需要的时间（传输时间）是：10000/1*10**6s=10ms。发出后，再经过RTT的传播时间，确认回来，RTT时间是：2*D=100*2=200ms。这样，发送一个消息到收到确认（意味着可发下一个消息）的总耗时是200+10=210ms，即每 210 毫秒发送 10000 位，计算器传输速率应该为： 10000b/0.21s = 48 kbps。

也可以这样计算：信道的利用率是10ms/210ms，吞吐量应是：10/210*1Mbps=48kbps

- 2 单选 (2分) 使用滑动窗口，每个窗口10 个数据包， RTT 为100 ms ，假设有 1250 字节数据包，最大吞吐量是多少？（注意 1250 字节数据包是 10000 位。要找出最大的吞吐量，假设网络容量不是一个限制因素，忽略数据包丢失) 得分/总分

  A. 1 MBps (每秒兆字节数)

  B. 2 Mbps （兆比特每秒） ✕0.00/2.00

  C. 1 Mbps （兆比特每秒）

  D. 125 kbps (千比特每秒)

  正确答案：C 你错选为B

  解析： C、10 个包 = 100000 位，需要在每 0.1 秒完成传输，意味着 100000/0.1 = 1 Mbps

3.6.4 课程总结

- 为什么提出滑窗技术?
- 什么是发送窗口?
- 发送窗口什么时候滑动?
- 什么是接收窗口?
- 接收窗口什么时候滑动?
- 什么是带宽延迟积?
- 窗口值怎么确定？跟哪些因素有关?

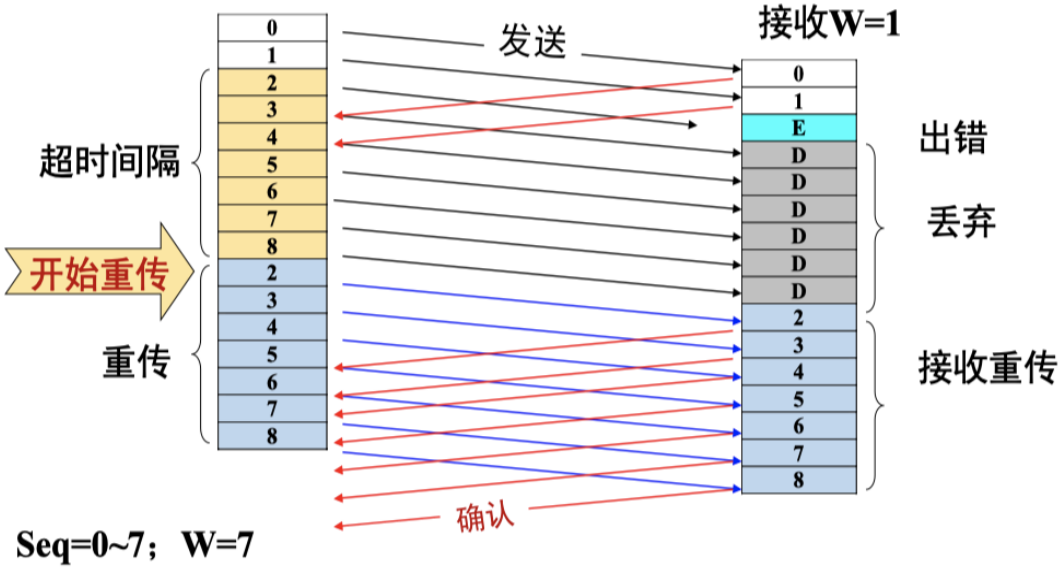
3.7 回退n帧与选择性重传(协议5、6)

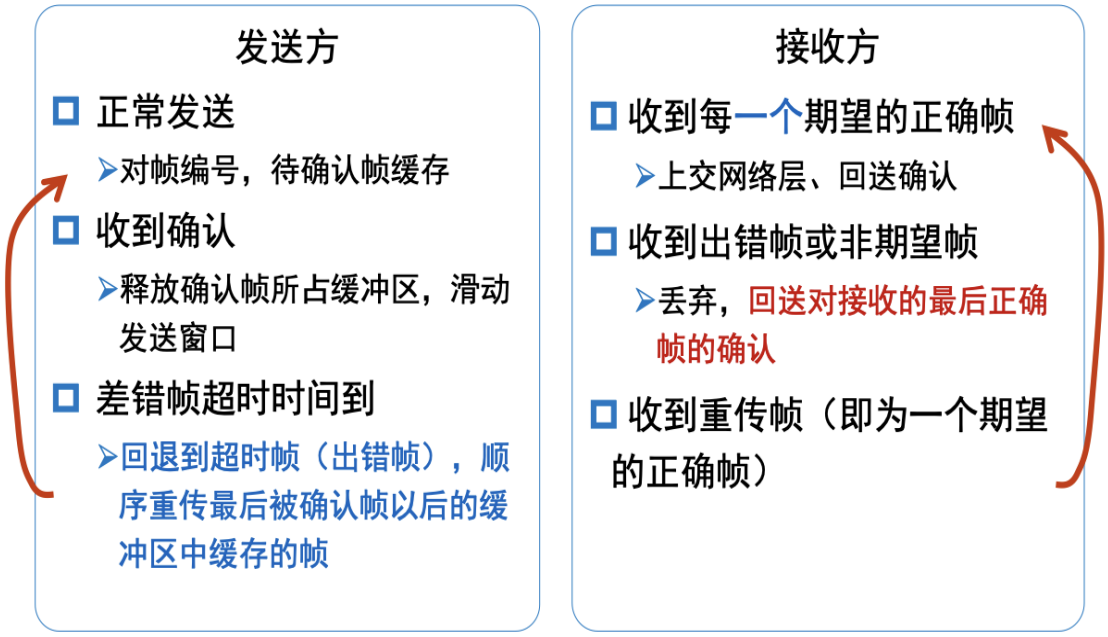
3.7.1 批量发送面临的问题

- 出错情况
  - 连续发送W个数据帧，其中有一帧出错，但其后续帧被成功发送，此时应该如何处理?
- 接收方接收策略选择
  - 丢弃错帧及后续帧，其后续帧因不是期望接受帧也被丢弃
  - 丢弃错帧，缓存后续正确接收帧
- 对应发送方的重传策略选择
  - 缓存在发送窗口中的出错帧以及其后续帧全部重发 —— 协议5：回退n帧
  - 只重发出错帧 —— 协议6：选择性重传

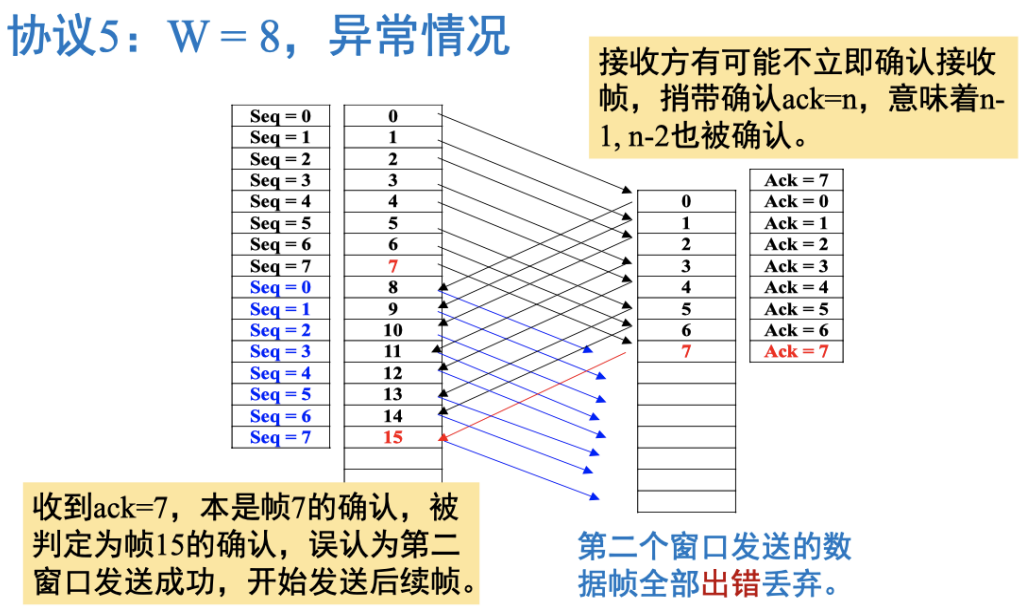
3.7.2 回退n帧 (协议5)

- 接收方与发送方策略
  - 接收方 —— 丢弃错帧及后续帧，其后续帧因不是期望接受帧也被丢弃

        - 接受窗口 $W = 1$
    - 发送方 —— 缓存在发送窗口的出错帧以及其后续帧全部重发
- 回退n帧协议的基本概念
    - 定义序列号 seq 的取值范围和滑动窗口 W
    - 发送方连续发送至发送窗口满
    - 接受窗口为 1，对出错帧不确认 (引发超时)
    - 发送方超时重传，从未被确认帧开始
- 回退n帧工作原理
    - 

    - 

- 累计确认
    - 回退n帧采用了累计确认，发送方会开辟一个较大的空间保存着所有未被确认的帧。
    - n号帧的确认到达时，暗含：n-1、n-2...等 n 号帧之前的帧也被确认。
    - 采用累计确认之后，收到了 n 帧的确认，n帧之前的帧都可从缓存中删掉。
- 滑动窗口大小 W 的选择
    - 发送窗口：$W \leq \mathrm{MAX_SEQ}$
    - 比如，序列号由 3 位表示，3位表示的最大数为7，则窗口最大为7。

- 如果 $W = 8$，则会出错。
  - 第一次发送0-7帧，接收方回复ack=7表示都收到了。
  - 第二次继续发送0-7帧，但这次帧在传输过程中出错，接收方仍回复ack=7，由此导致错误。
  - 

- 总结
  - 回退n帧，需要发送方付出更多的缓存代价。
    - 适合出错率较少的高速信道。
  - 回退n帧的窗口数
    - 发送窗口：$W \leq \text{MAX_SEQ}$
    - 接受窗口：$W = 1$

3.7.3 选择性重传 (协议6)

- 接收方与发送方策略
  - 接收方 —— 丢弃错帧，缓存后续正确接收帧
  - 发送方 —— 只重发出错帧
- 工作原理
  - 发送方
    - 正常发送：对帧编号，待确认帧缓存
    - 收到确认：释放确认帧所占缓冲区，滑动发送窗口
    - 重传的差错帧超时：重传缓存的最后被确认帧以后的那一帧
  - 接收方
    - 正常接收：上交网络层、回送确认，滑动接收窗口
    - 收到非期望的正确帧：缓存，回送对接收的最后正确帧的确认
    - 收到重传帧：将缓存帧排序上交，回送确认，滑动接收窗口
  - 三大关键步骤
    - 接收方收到非期望的正确帧 —— 缓存
    - 发送方选择帧seq2重传

        - 接收方收到重传帧seq2 —— 与之前的帧排序上交网络层
- 否定确认 NAK
    - 当我们收到第二号出错帧时，之前是直接丢弃等发送方超时重传。
    - 现在可以使用NAK，收到第二号出错帧之后，向发送方发送 NAK=2，明确告诉发送方第二号帧出错。
    - 加快重传节奏。
- 协议5与协议6的比较
    - 回退n帧
        - 发送方需要较大的缓冲区，以便重传
        - 重传帧数多，适用于信道出错率较小的情况
    - 选择重传
        - 接收方需要较大的缓冲区，以便按正确顺序将分组提交网络层
        - 重传帧数少，适于信道质量不好的情况
- 滑动窗口 W 大小的选择
    - 接收窗口：$W = \frac{\text{MAX_SEQ}+1}{2}$，保证接收方新老窗口不相交。发送窗口通常小于接收窗口。
    - 比如，序列号为3位，则 $\text{MAX_SEQ} = 7$，$W = 4$。
    - 例如W=7，即会出错。
        - 一开始接收窗口为 $0\ 1\ 2\ 3\ 4\ 5\ 6$，发送方发送 0-6，接收方收到之后，接收窗口变为 $7\ 0\ 1\ 2\ 3\ 4\ 5$，并向发送方发送 0-6 号帧的确认帧。
        - 但是0-6号帧的确认帧在传输过程中全部丢失，发送方的0号帧超时重发。
        - 接收方收到0号帧后，发现接收窗口中有0号帧，误以为是新帧便将其缓存。但该0号帧只是一个重传帧，由此导致了协议错误。
        - 因此使得协议不发生错误，只能保证接收方新老窗口不相交，即 $W = \frac{\text{MAX_SEQ}+1}{2}$。
- 例题

4 单选 (2分) 若数据链路层的发送窗口尺寸W=4，在发送3号帧、并收到2号帧的确认帧后，发送方还可以连续发送多少个帧？（采用累计确认） 得分/总分

A. 2

B. 1

C. 3 ✔2.00/2.00

D. 4

正确答案：C 你选对了

解析： C、ack=2意味着序列号=2及之前的帧均被正确接收，可以清空他们占用的窗口，现在又发了一帧（序列号=3）等待确认，占用了发送窗口，窗口总大小是4，顶多能再发4-1=3帧。

    -

3.7.4 三个滑动窗口协议窗口大小

-

	协议4：滑动窗口	协议5：回退 n 帧	协议6：选择性重传
发送窗口 (SWnd)	$0 < \mathrm{SWnd} \leq 1$	$0 < \mathrm{SWnd} \leq \mathrm{MAX_SEQ}$	$0 \leq \mathrm{SWnd} \leq \mathrm{RWnd}$
接收窗口 (RWnd)	$\mathrm{RWnd} = 1$	$\mathrm{RWnd} = 1$	$\mathrm{RWnd} = \frac{\mathrm{MAX_SEQ}+1}{2}$

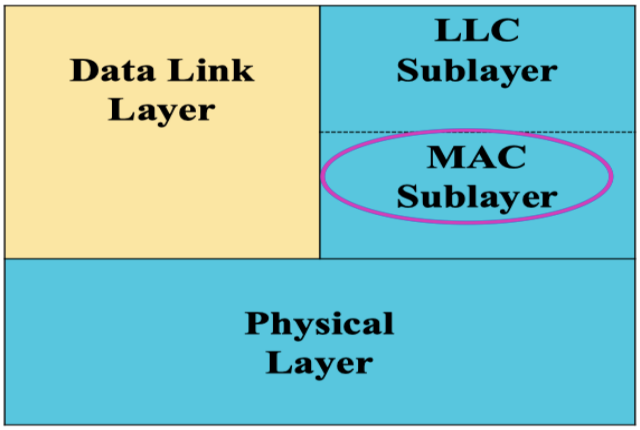
3.7.5 课程总结

- 回退 n 帧的工作原理是怎样的?
- 什么是累计确认?
- 如何确定回退 n 帧协议的窗口数?
- 什么是否定确认重传？有什么作用?
- 选择性重传的工作原理是怎样的?
- 如何确定选择性重传协议的窗口数?

第四章 介质访问控制子层

4.1 MAC子层概述

4.1.1 数据链路层的组成

- 数据链路层分成两个子层
  - 逻辑链路控制 (LLC - Logical Link Control)
  - 介质访问控制 (MAC - Madia Access Control)
  - 

4.1.2 数据通信的三种方式

- 单播 (unicast)：一对一的通信
- 广播 (broadcast)：一对所有的通信，发出一个信息要送达到所有人
  - 局域网主要采用共享信道的方式来进行通信，主要采取的方式是广播。
- 组播 (multicast)：一对一个组的通信，一个信息发送给一个特定的组的用户

4.1.3 介质多路访问控制 / 介质访问控制MAC

- 广播的问题
  - 共享信道 (多路访问信道 / 广播信道)
  - 可能会有很多用户同时请求使用这根共享信道，需要确定广播信道上的下一个使用者，尽量避免冲突。
- 分配信道的两种方式 (介质访问控制)
  - 静态分配：按照某种方式，预先分配给各用户，不同用户使用情况不同
    - PSTN上干线的多路复用，信道被按照频率或时间进行分配
  - 动态分配
    - 信道不需要提前分配，信道是开放的
    - 信道按照需求动态的分配给用户
- 静态分配
  - 静态分配的问题
    - 资源分配不合理，不同用户对资源需求不同
    - 有资源浪费，效率低
    - 信道是平均分配的，延迟时间增大 N 倍。
  - 静态分配适用环境
    - 适用于用户数量少且用户数目固定的情况。
    - 适用于通信量大且流量稳定的情况。
    - 不适用于突发性业务的情况。
- 动态分配
  - 多路访问协议 - 动态分配信道资源 - 提高信道利用率
- 多路访问协议
  - 随机访问协议
    - 特点：站点或用户来争用信道，可能会出现冲突
    - 典型随机访问协议：ALOHA协议、CSMA协议、CSMA/CD协议 (以太网采用)
  - 受控访问协议
    - 特点：站点被明确地分配，不会出现冲突。

4.1.4 课程总结

- 什么是广播?
- 局域网中主要采用什么数据通信方式?
- 介质访问控制用来做什么?
- 介质访问控制协议分成哪两类?

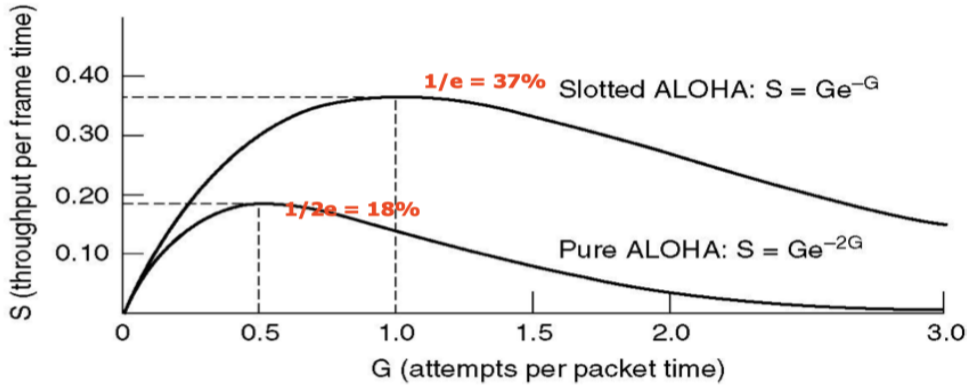
4.2 ALOHA 协议

4.2.1 纯ALOHA协议

- 工作原理
  - 任何一个工作站都可以在帧生成之后立即发送，并通过信号的反馈，检测信道判断是否发送成功。
  - 如果发出的帧发生冲突，则经过随机等待的时间之后重发数据帧。
- 特点
  - 每个站点可在任意时间发送数据 (不关心信道是否已被占用)，非常任性，想发就发
  - 两个以上站点同时发送数据时就会产生冲突。
- 性能分析
  - 吞吐率 S：一个帧时 (T) 内发送成功的平均帧数，$0 < S < 1$，$S = 1$ 则利用率达到100%。
  - 运载负载 / 网络负载 G：在一个帧时 (T) 内，所有通信站一共发送的帧的平均值 (原发+重发)，$S \leq G$
    - $G = S$，无冲突
    - $G > S$，有冲突
    - $G > 1$，冲突频繁
  - P_0：一帧发送成功的概率，即发送成功的帧数在全部已发送帧的总数中所占的比例。
    - $S = G * P_0$
  - 冲突危险期：可能发生冲突的时间长度 $= 2T$，发送之前的 T 加上发送之后的 T
    - 生成帧均值：$2G$
    - 生成 k 帧的概率服从泊松分布 (假设)：$Pr[k] = \frac{G^k e^{-G}}{k!}$ ，$Pr[0] = e^{-G}$
    - 在 $2T$ 内发送成功的概率，即为 $2T$ 内都不产生新帧，即 $P_0 = Pr[0] * Pr[0] = e^{-2G}$
    - $S = G * e^{-2G}$，求 S 极大值，$S' = e^{-2G} - 2G * e^{-2G} = 0$。当 $G = 0.5$ 时，$S_{max} = 0.184$。因此纯 ALOHA 协议信道利用率最高为 18.4%。

4.2.2 分隙(分槽)ALOHA协议

- 工作原理
  - 将时间分成时隙 (时间片)，时隙=帧时 (T)。
  - 发送帧必须在时隙的起点，不允许随机发送，因此冲突只发生在时隙的起点。冲突危险期缩短为 T，因为只会与发送之后出现的帧冲突，不会与发送之前出现的帧冲突。
    - 一旦某个站占用某个时隙并发送成功，则在该时隙内不会出现冲突。
- 性能分析
  - $P_0 = Pr[0] = e^{-G}$
  - $S = G * e^{-G}$，$G = 1$ 时，$S_{max} = 0.368$，即分隙 ALOHA 协议信道利用率最高为 36.8%。
- 与纯ALOHA协议对比

- 

4.2.3 课程总结

- 纯ALOHA协议的工作原理是什么？
- 纯ALOHA协议的冲突危险期是多少，信道利用率怎么样？
- 分隙ALOHA协议的工作原理是什么？
- 分隙ALOHA协议的冲突危险期是多少，信道利用率怎么样？

4.3 CSMA 协议

4.3.1 CSMA 协议概述

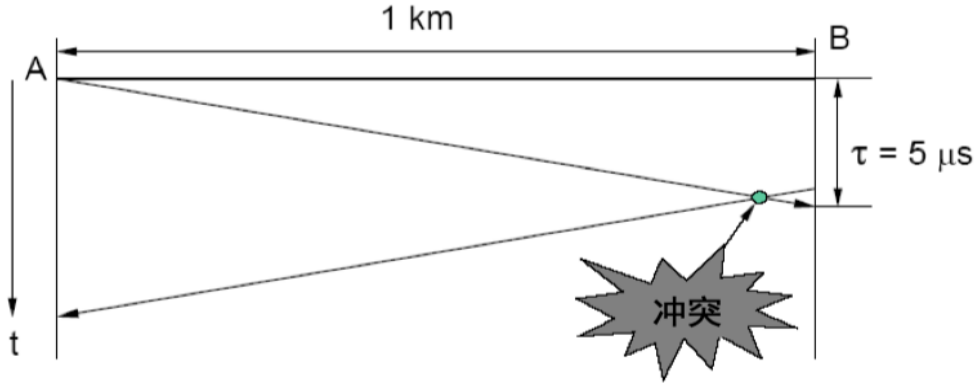
- 定义：载波侦听多路访问协议 (CSMA - Carrier Sense Multiple Access)
  - 改进的 ALOHA 协议 —— "先听后发"
- 分类
  - 非持续式 CSMA 协议
  - 持续式 CSMA 协议
    - 1-持续 CSMA
    - P-持续 CSMA

4.3.2 非持续式 CSMA 协议

- 工作过程
  - ① 发送数据帧时侦听信道，如果信道空闲，则发送自己的帧。
  - ② 如果侦听时，信道被占用，则等待一个随机分布的时间，然后再重复 ①。
- 缺点：随机等待的时间内信道上没有数据传送，这段时间是浪费的。

4.3.3 持续式 CSMA 协议

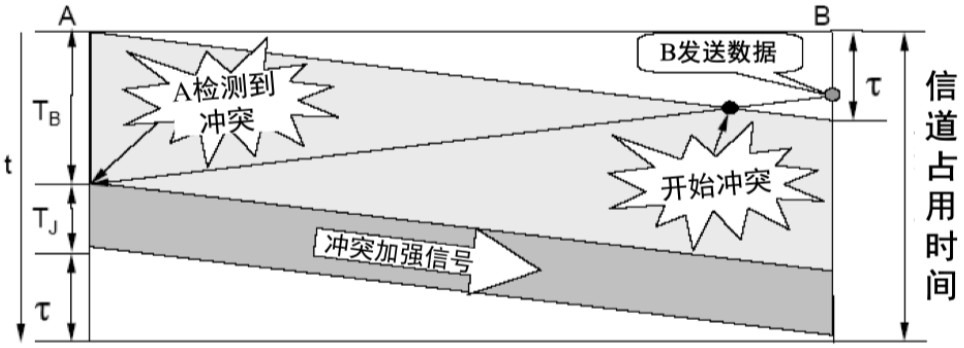
- 工作过程

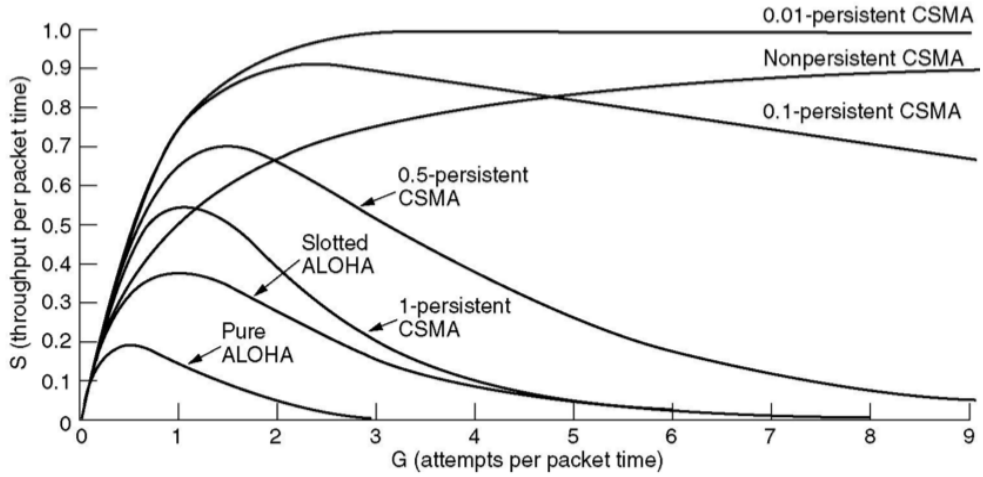
    - ① 发送数据帧时侦听信道，如果信道空闲，则发送自己的帧。
    - ② 如果侦听时，信道被占用，则持续侦听。一旦空闲立即发送数据帧。
    - ③ 如果发生冲突，则等待一个随机分布的时间，再重复 ①。
- 缺点：如果两个以上的站同时在侦听信道，一旦介质空闲，两个站同时发送数据帧，则会发生冲突。
- P-持续 CSMA
    - 工作过程
        - ① 发送数据帧时侦听信道，如果信道空闲，则以 P 的概率发送数据帧，(1-P) 的概率延迟一个时间发送。
        - ② 如果侦听时，信道被占用，则持续侦听。一旦空闲则重复 ①。
        - ③ 发送推迟一个时间单元后，继续重复 ①。
- 传播延迟对载波侦听的影响
    - 信道传输 200 m/us。A、B距离 1000 m，A 侦听信道空闲后向 B发送一个数据帧。
    - 4us 过去后，数据帧还未到达 B。此时 B 要发送一个数据帧，因此侦听信道。
    - 由于 A 发的数据帧还未到达 B，因此 B 以为信道空闲，则发送数据帧，两个数据帧发生冲突。
    - 

- 冲突窗口
    - 定义：一个工作站能够检测到冲突的时间的最大值，数值上等于最远两站传播时间的两倍，即 RTT (Round Trip Time)。
    - 介质传播速度：$v = 200\ m/us$，网卡延迟：t_{PHY}，最远两个工作站距离：S
        - 两站传播时间：$t = \frac{S}{v}$
        - 冲突窗口：$2 * (t + t_{PHY} + N * t_{中继器})$，N为经过中继器个数，$t_{中继器}$ 为到达中继器到离开中继器的一段时间。

4.3.4 CSMA/CD

- 定义：CSMA with Collision Detection (带冲突检测的载波侦听多路访问协议)
    - 本质是 1-持续 CSMA
    - 半双工以太网中使用
- 工作原理
    - ① 经侦听，如介质空闲，则发送。
    - ② 如介质忙，则持续侦听，一旦空闲立即发送。
    - ③ 发送成功后，边发边听，侦听发送出去的帧有无和其它帧发生碰撞。如果没有发生冲突，

则表明发送成功。
    - ④ 如果发生冲突，等待一个随机分布的时间再重复 ①。
    - 总结：先听后发、边发边听
- 堵塞信号 (强化冲突信号)
    - 所有工作站在发送的同时也接收自己的信号 (广播)，监测发送的情况，一旦收到的信号与发出的不一致，就说明发生了冲突。
    - 发送站感知冲突后立即停止帧的发送，并且发一个简短的堵塞信号 (称强化冲突信号，Jamming signal)，通知各站点该处发生冲突，本站及网上所有站都等待一段随机分布的时间，然后再按CSMA/CD方式重发该帧。
    - 

- 性能比较
    - P-持续 CSMA中，P越小发生冲突可能性越低，信道利用率越高，但是等待的时间越长。
    - 

- 冲突检测和处理
    - 冲突检测
        - 检测发出的信号与收到的信号，能量与脉冲宽度的变化，如果发生了变化则发生了冲突。
    - 冲突检测的要求
        - $\text{时隙宽度} = \text{最大冲突检测时间}(\text{冲突窗口})$，保证在一个时隙内能够检测到最远距离的冲突。
        - $\text{最大冲突检测时间}(\text{冲突窗口}) \leq \text{发送有效帧的时间}$，防止在发生冲突时，已完成短帧发送而无法知道实际上已经发送失败。

4.3.5 课程总结

- CSMA系列协议的本质是什么?
- CSMA/CD 的基本工作原理是什么?
- 什么是冲突危险期?
- 先听后发仍然会发生冲突，原因是什么？(两个原因：同时发送与传播延迟)
- 冲突检测的原理是什么?

4.4 以太网概述

4.4.1 以太网基础知识

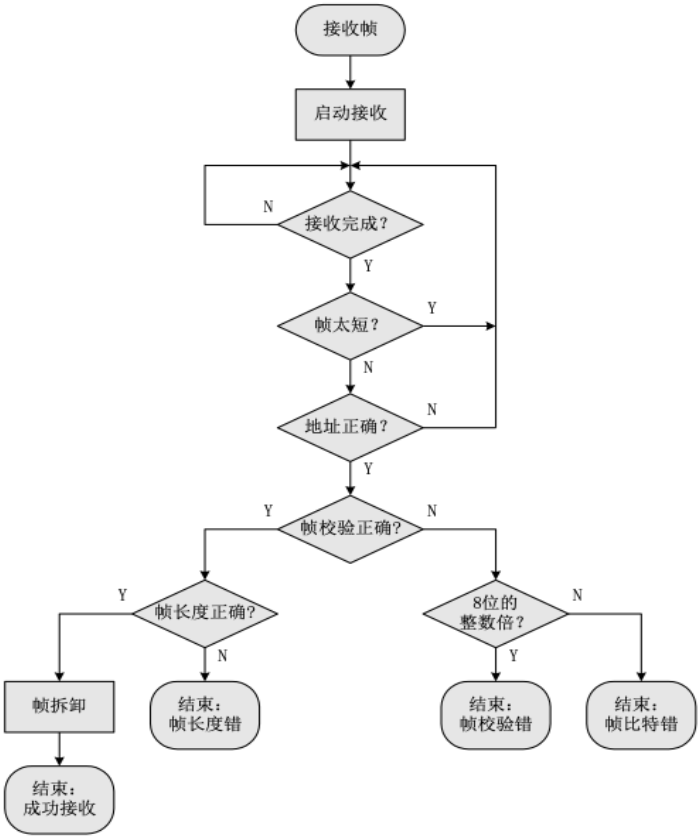
- 以太网发展史
  - 1973年，Robert Metcalf 及其同事设计了以太网雏形 (施乐公司)
  - 1980年，DIX 发布了最早的以太网标准，开放标准
  - 1985年，IEEE 802.3 对以太网作了小的修改，基本一致，兼容以太网
  - 1995年，IEEE 宣布了 100Mbps 以太网标准
  - 1998-1999年，通过了千兆以太网标准 (吉比特以太网 - 1000M)
  - 2002年，IEEE 通过了10Gbps 以太网标准
  - 2016年，IEEE 通过了25Gbps 以太网标准
- 2 种以太网
  - 经典以太网
    - 3M~10Mbps，不再使用
    - 10Base2、10Base5、10BaseT
  - 交换式以太网
    - 10M、100M、1G，广泛使用

4.4.2 经典以太网

- IEEE 以太网命名规则
  - 10Base2 (IEEE 802.3a)
    - 10：传输带宽 (Mbps)
    - Base：基带传输
    - -2 (或5)：支持的分段长度 (100m为单位，四舍五入)，此处为 200m 传输距离
  - 10Base-TX (IEEE 802.3X)
    - T：铜制非屏蔽双绞线
    - F：表示光缆
- 以太网线缆
  -

名称	电缆	最大区间长度	节点数/段	优点	接口
10Base5	粗缆	500m	100	用于主干	AUI
10Base2	细缆	200m(185)	30	廉价	BNC
10Base-T	双绞线	100m	1024	易于维护	RJ-45
10Base-F	光纤	2km	1024	用于楼间	ST

    - 其中 10Base5 与 10Base2 采用了总线拓扑的方式。
    - 10BaseT
        - 物理连接是星型/扩展星型结构，中心是集线器 (Hub)
        - 逻辑上是总线结构 (站点争用总线)
        - 采用双绞线，与同轴电缆相比的优点
            - 安装成本大大降低
            - 即插即用，组网灵活
            - 星型结构，故障隔离
            - 适于大批量制造
            - 具有开放式标准的互操作能力
        - 问题：多用户共享一条 10M 速率信道。
- 以太网编码
    - 以太网采用曼彻斯特编码，$_{\text{比特率}} = 10\ Mbps$，$_{\text{波特率}} = 20\ MHz$ 。
        - 1：高电压 -> 低电压
        - 0：低电压 -> 高电压
    - 802.5 采用差分曼彻斯特编码
- IEEE 802.3 / 以太网 MAC 子层协议
    - IEEE 802.3 协议描述了运行在各种介质上 1 Mb/s ~ 10Mb/s 的 1-持续 CSMA/CD 协议的局域网标准。
    - IEEE802.3 与 以太网有一定差别 (比如帧格式)。
- 以太网工作站接收数据流程

- 

- 二进制指数后退算法
  - 起源：以太网采用 CSMA/CD，发送方在检测到冲突后，双方 (或多方) 都将延时一段时间，那么这段时间到底是多长?
  - 具体算法
    - 冲突检测到后，时间被分成离散的时隙
    - 时隙的长度等于信号在介质上来回传输时间 (51.2 us)
    - 一般地，i 次冲突后，等待的时隙数将从 (0~2^i-1)*51.2 us 中随机选择。
    - $0<i\le 10$ 时，取 (0~2^i-1)*51.2 us。
    - $10<i<16$ 时，取 (0~1023)*51.2 us。
    - $i>16$ 时，放弃发送。
  - 二进制指数后退算法的优化
    - 上述主要讨论的是发送方怎样避免冲突，或冲突后怎样再次成功发送。
    - 但是，一旦成功发送后，接收方如需发确认帧，其中又有争用信道的问题。如把一次成功发送后的第一个时隙留给接收方，则可及时收到确认帧。
- 经典以太网提高负载的方法
  - 提速到 100 M
  - 采用全双工
  - 采用交换式网络，将集线器 (Hub) 换成 交换机 (Swicth)

4.4.3 交换式以太网

- 100Mbps以太网 —— 802.3u
  - 要求：兼容性、成本低、标准化

- 基本思想：保留原有帧格式、接口和过程规则，将比特时间 (位时间-发送 1 bit 所需时间) 降为10 ns (原100 ns)，电缆最大长度降到10分之一 (原2500米)。
- 传输介质与传输距离
  -

名称	传输介质	最大距离
100Base-T4	3 类双绞线	100 米
100Base-TX	5 类双绞线	100 米
100Base-FX	光纤	2000 米

- 采用了 4B/5B 编码方式
  - 将 4 bits 数据编码成 5 bits 信号，效率更高、易于实现、电压平衡
  -

4B/5B编码表

十进制	4位二进制数	4B/5B 码	十进制	4位二进制数	4B/5B 码
0	0000	11110	8	1000	10010
1	0001	01001	9	1001	10011
2	0010	10100	10	1010	10110
3	0011	10101	11	1011	10111
4	0100	01010	12	1100	11010
5	0101	01011	13	1101	11011
6	0110	01110	14	1110	11100
7	0111	01111	15	1111	11101

- 千兆位以太网 (吉比特以太网，GE) —— 802.3z
  - 淘汰了 FDDI(100M)、ATM(155/622M)
  - GE 与现有 10M/100M 以太网技术兼容
    - FDDI和ATM与10M/100M以太网不兼容，需要改变帧格式
  - 主要工作在全双工模式，如果在半双工工作，需要使用 CSMA/CD 技术，可能带来传输距离过短的问题。
    - 可使用帧串、帧扩充等方法解决。
  - 速度提高带来的冲突检测问题以及解决办法
    - 只有在半双工的模式下才会遇到。
- 以太网强大生命力原因
  - 简单性和灵活性
    - 可以很方便的将机器挂接到以太网上
  - 易于维护
  - 支持TCP/IP协议，互联容易
  - 使用4B/5B等编码，擅于借鉴其他协议(FDDI)。
- 以太网建立无连接不可靠的服务

- 5 单选 (2分) 以太网的MAC协议提供的是什么服务? 得分/总分

  A. 无连接可靠的服务

  B. 面向链接的可靠的服务 ✕0.00/2.00

  C. 面向链接的无可靠的服务

  D. 无连接不可靠的服务

  正确答案：D 你错选为B

  解析： B、经典以太网采用的MAC控制是CSMA/CD，发帧之前，不需要或对方建立连接（无连接）；也不需要对方发送确认，可能丢帧，可能碰撞/冲突（不可靠）。所以，选择无连接不可靠的服务。 D、经典以太网采用的MAC控制是CSMA/CD，发帧之前，不需要或对方建立连接（无连接）；也不需要对方发送确认，可能丢帧，可能碰撞/冲突（不可靠）。所以，选择无连接不可靠的服务。

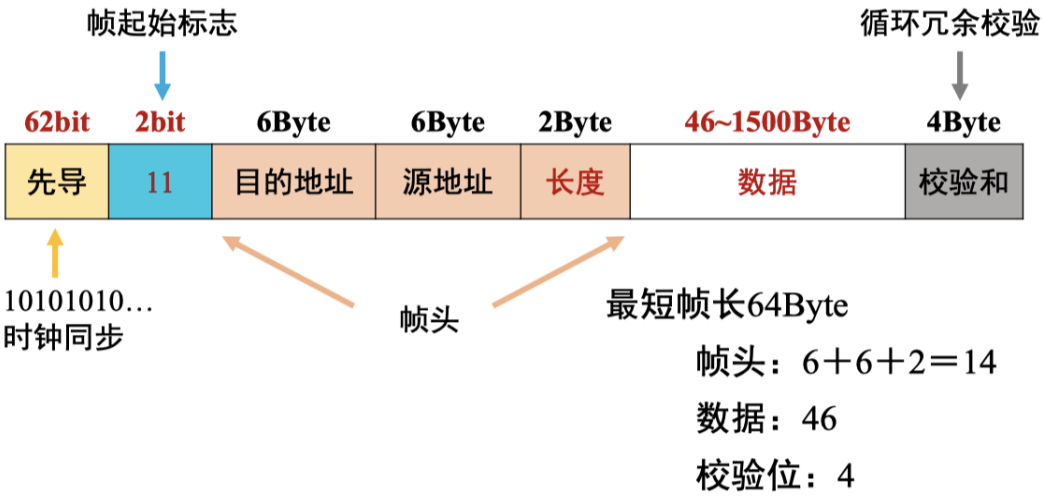
课程总结

- 以太网是多路访问协议的实现：CSMA/CD
- 二进制指数回退算法，降低了再次冲突的可能
- 问题
  - 二进制指数回退算法是怎样做的?
  - 经典以太网的特点是什么？(半双工、使用集线器、曼彻斯特编码)
  - 交换式以太网的特点是什么？(全双工、交换式网络、4B/5B)
  - 以太网为什么具有强大的生命力?
  - 以太网采用的编码方法是什么?

4.5 以太网帧格式

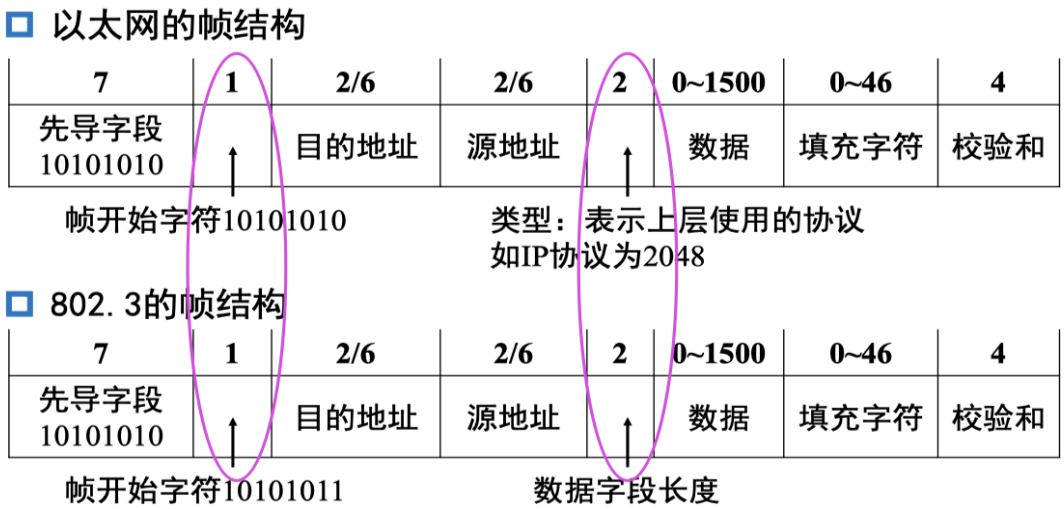
4.5.1 IEEE 802.3与以太网 MAC 子层协议

- 以太网覆盖物体层与数据链路层，而802.3在OSI模型中仅覆盖一层半 (即物理层+数据链路层的MAC子层)
- IEEE 802.3帧格式
  -



- [第一个字段] 先导+帧起始标记 (前导码)：表明一个帧的开始。帧开始字符为10101011。
- [第二、三个字段] 目的地址与源地址：均为物理地址，即MAC地址。
  - MAC地址为48位，一共有7万亿个。
  - 目的地址：接收方的MAC地址
  - 源地址：发送方的MAC地址
- [第四个字段] 长度字段：表明了这个帧的长度
  - 802.3帧长度，最短64字节，最长不能超过1518字节。(不包括前导码)
  - 最长长度限制：早期内存昂贵，接收方存储帧需要内存，因此有最长长度限制
  - 最短长度限制：CSMA/CD的要求，要求 $\text{最短帧的发送时间} \geq \text{争用时隙}\ 2t(\text{冲突窗口})$
    - 802.3 规定，在10Mbps局域网中，$\text{时隙}\ 2t = 51.2\ us$
    - $\text{最短帧长度} = \frac{10\ Mbps * 2t}{8} = 64Byte$
- [第五个字段] 数据字段：46~1500字节
  - 如果数据字段不足46字节，则必须进行填充，保证长度到达帧的最小长度，即64字节。
- [第六个字段] 校验和字段
  - 采用CRC循环冗余校验
  - 校验范围：目的/源地址、长度、LLC数据等字段 (除了前导码的所有帧数据)

- IEEE 802.3 与以太网帧 (DIX) 的区别
  -



- ① 帧开始字符
  - 以太网帧：10101010
  - 802.3 帧：10101011
- ② 802.3帧的长度字段、DIX以太网帧的类型字段
  - 类型字段：指明网络层所使用的协议。(以太网覆盖了下两层，需要直接与网络层打交道，以太网帧是实际中使用的帧)
  - 长度字段：指明帧的长度。(不需指明类型，因为802.3帧不需要与网络层打交道，802.3在OSI模型中只覆盖了一层半)
- 如何鉴别以太帧与802.3帧 —— 查看第四个字段
  - $_{\text{数值}} < 1536(0x600)$，则为长度，为 802.3 帧。
  - $_{\text{数值}} > 1536(0x600)$，则为类型，为 以太帧。

4.5.2 MAC地址

- MAC地址为6字节，一共48位，可以表示7万亿个地址。
- MAC地址组成
  - MAC地址分为前24位与后24位，前24位表示一个公司或一个机构，后24位由该公司或机构自行划分，前24位称为OUI。
    - 如3Com的前24位地址：00-60-8C-XX-XX-XX
    - 典型的Ethernet (以太网) 地址：00-60-8C-01-28-12
- 工作站的源地址具有一个特性：全球唯一性
  - 每个工作站的MAC地址由IEEE分配，保证世界上没有两个工作站具有的MAC地址是相同的。
    - IEEE 要求厂商遵循两条规定
      - 必须使用该供应商分配的OUI作为前3个字节
      - OUI相同的所有MAC地址的最后3个字节必须分配唯一的值
  - MAC地址存储在网卡的ROM中，当计算机启动时，MAC地址从ROM拷贝到RAM中。
- MAC地址的3种表示方法
  - 使用破折号：00-60-2F-3A-07-BC
  - 使用冒号：00:60:2F:3A:07:BC
  - 使用句点：00.60.2F.3A.07.BC
  - 必须使用16进制来表示MAC地址。

4.5.3 课程总结

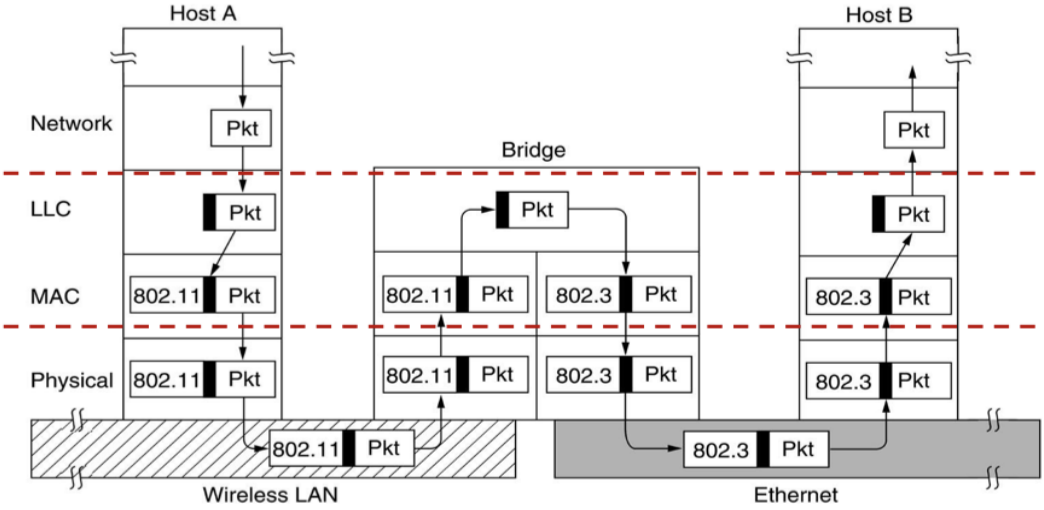
- 以太帧的各字段分别是什么?
- 如何识别类型/长度字段?
- MAC地址具有哪些特征?
- 为什么以太帧最短至少需要64字节?

4.6 二层交换的基本原理

4.6.1 二层交换基本概念

- 二层交换
  - 有很多 LAN(局域网)，需要二层交换将它们连接起来。
  - 可以用 网桥(bridge) 将它们连接起来。
    - 网桥工作在 DLL层(数据链路层)，通过检查MAC地址做出转发帧的决策。
    - 因此网桥不会检查网络层，即IPv4、IPv6、AppleTalk、ATM、IPX与OSI分组均可穿越网桥
    - 交换机是网桥的现代名称。

4.6.2 网桥

- 网桥连接802.11 (无线) 与 802.3，使其正常工作
  - 

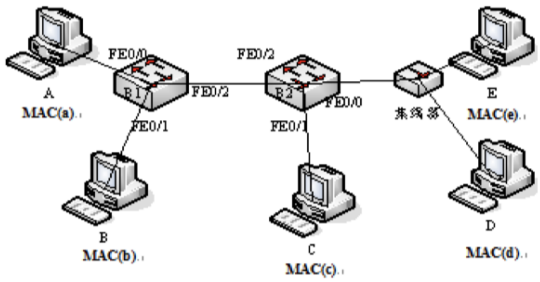
- 网桥连接 802.X 到 802.Y 的操作
  - 不同的帧格式，需要转换 —— 重新封装
  - 不同的数据传输速率 —— Buffering，缓存
  - 不同的802LAN有不同的最大帧长度 (802.3的1526字节，802.11的2346字节)
  - 安全：802.11和802.16支持数据链路层的加密，但802.3不支持
  - 服务质量：802.11和802.16提供了服务质量，但802.3没有
- 透明的网桥 (transparent bridges)
  - 透明的网桥 —— 将LAN插上网桥之后，一切自动工作，不需要任何其他的设置。
  - 通过透明网桥将多个LAN连接起来，硬件和软件不需要做任何的变化
  - 透明网桥工作在混杂模式 (promiscuous mode)，它接收所有跟它相联的LAN的帧
  - 当一个帧到达网桥时，它必须作出丢弃 (discard) 还是转发 (forward) 的决策。
    - 如果是转发，必须知道向哪个LAN转发。
  - 决策是通过网桥内部的一张地址表 (Hash Table) 中查找目的MAC地址而作出的。
- 网桥如何维护内部转发表
  - 初始时，转发表为空。

- 泛洪算法 (扩散算法 - flooding algorithm)
  - MAC地址表为空时，大量采用广播
  - 当网桥不知道目的地址时 (表中查不到)，网桥会将这帧从除来的LAN外的所有LAN转发出去。
- 后向学习法 (backward learning)
  - 信息对 - 源地址与其到达的端口
  - 网桥从到达帧的源地址认识到源地址对应的那台机是在帧来的那个LAN上，所以将它写入MAC地址表。
- 网桥如何适应网络拓扑的变化 (即LAN的连接与断开)
  - 无论何时，凡往表中加入记录，必须同时打下时戳。
  - 到达帧的源地址在表中已有记录，则将时戳更新为当前时间。
  - 网桥周期性地扫描表，将那些超时的记录从表中删除。
  - 地址表始终记录着最新且活跃的那些工作站的信息！

- 网桥工作原理
  - 当一帧达到时，网桥启动如下算法：[过滤]
    - 网桥打开帧，查看目的与源MAC地址，在地址表中找到对应的LAN
    - 如果源LAN和目的LAN相同，则丢弃该帧。
    - 如果源LAN和目的LAN不同，则向目的LAN转发该帧。
    - 如果目的LAN未知，则广播该帧。(表中未查到目的MAC地址对应的LAN)
  - 每当一帧达到，执行上述算法。
  - 使用专用的VLSI芯片，可以在几微秒内完成查找和更新表项的动作。
- 网桥/交换机分割了冲突域，每个LAN中设备的通信不会影响，因此提高了性能。
- 例题

  请写出下列数据帧正常收发之后，两个交换机内部的MAC地址表。

  - **A向D传送一个帧**
  - **E向A传送一个帧**
  - **D向E发送一个帧**



B1的MAC地址表（自学习表）	
MAC地址	端口
Mac(a)	FE0/0
Mac(e)	FE0/2

B2的MAC地址表（自学习表）	
MAC地址	端口
Mac(a)	FE0/2
Mac(e)	FE0/0
Mac(d)	FE0/0

4.6.3 网桥和中继器对比

- 网桥与中继器对比
  -

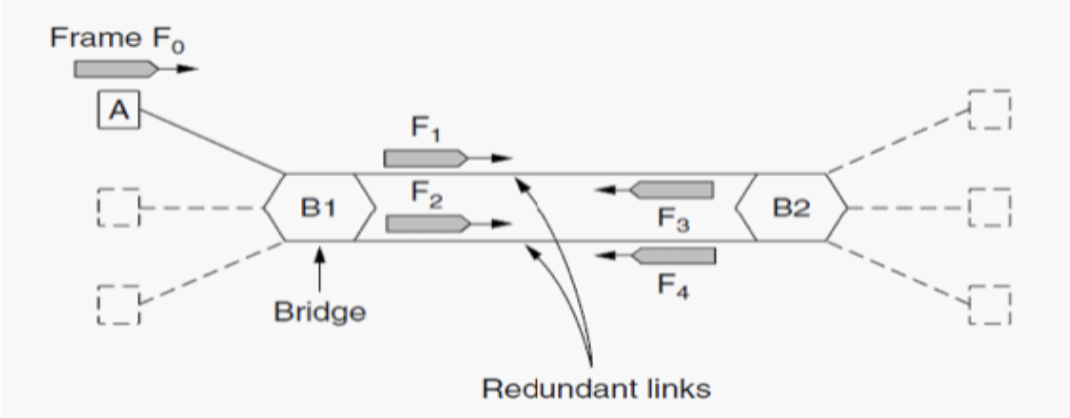
功能	网桥	中继器
再生信号	Yes	Yes
连接采用不同MAC协议的网段	Yes	No
隔离冲突域	Yes	No
根据帧头的物理地址转发帧	Yes	No
丢弃损坏帧	Yes	No

4.6.4 课程总结

- 为什么需要二层交换？(不同LAN协议不同，802.3、802.11...)
- 二层交换的基本原理是什么？
- 什么是扩散 (泛洪)？
- 为什么网桥又叫透明网桥？
- 为什么采用了交换机而不是集线器作为星形拓扑的中心，性能会得到提升？

4.7 生成树协议

4.7.1 冗余结构带来的问题

- 冗余结构
  - 为了网络的稳定，在网络设施中经常会有多根介质连接相同的机器
  - 有时也会对交换机做冗余处理，对单个交换机进行拆分，每个交换机连接不同的LAN
  - 

- 冗余结构带来的问题
  - 多帧传送

        - 多个帧通过不同路径进行传送和转发
    - 广播风暴
        - 如ARP请求，两个在环路上的交换机不断对请求进行广播，导致大量重复广播帧的产生
        - 严重时会导致网络崩溃
    - MAC 地址库不稳定
        - 每个源机器有多条路径可达，导致MAC地址库不断更新变化
        - 会使得交换机在转发帧时，转发行为不确定

4.7.2 生成树协议 (STP)

- 定义
    - STP - spanning tree protocol，由 Radia perlman 提出。
    - IEEE 802.1 D 标准
    - 目的：维护一个无环路的网络拓扑
- 工作原理
    - 每个网络都一个根网桥
    - 每个网桥都有一个根端口
    - 每个网段都有一个指定端口 (即每个LAN只能通过某个特定端口传输)
    - 非指定端口不被使用
- 注意
    - 生成树算法，生成在逻辑上无回路的树，即生成树。
    - 生成树算法能在有物理回路的网络中，生成一颗没有逻辑回路的生成树，但并不能保证其中的路径是最优的。
    - 非指定端口虽然不参与数据帧的转发，但它会侦听树的工作报文。
        - 当树上的某些端口失效了之后，非指定端口将重新启用，形成新的生成树。
- 小诗
    - 我想我永远也不会看到

      像一颗树那么优美的图画

      树那至关紧要的特性

      是无回路的连通

      树需要无限的扩展

      包才能到达每一个LAN

      首先，需要选好树根

      指定ID即可选定

      从树根开始，计算最小代价的路径

      这些路径，就是这棵树的枝条

      网络出自我等愚人之手

      而桥则发现了一棵生成树
- 例题

- 5 单选 (2分) 以下关于生成树协议作用的描述中，错误的是: 得分/总分

  - A. 生成树协议是一种路由选择协议 ✔2.00/2.00
  - B. 当局域网拓扑发生变化时，能够重新计算并形成新的无环路的结构
  - C. 能够自动控制局域网系统的拓扑，形成一个无环路的逻辑结构
  - D. 使得任意两个网桥或交换机之间只有一条有效帧传输路径

  正确答案：A 你选对了

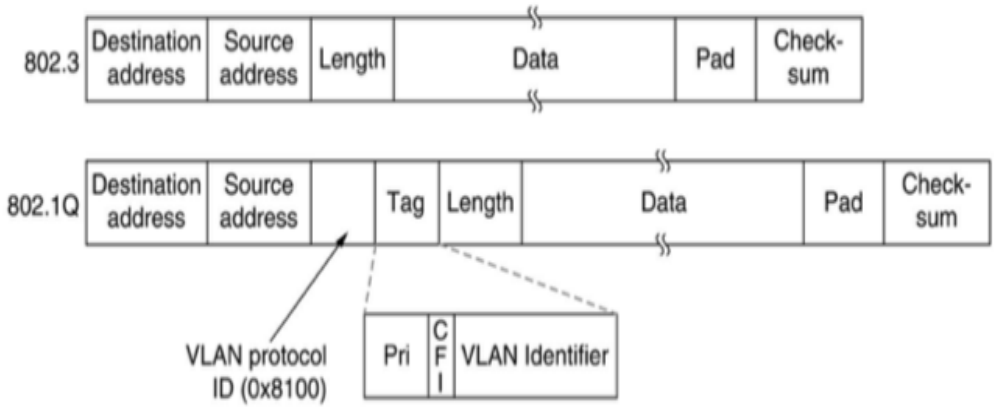
  解析： A、生成树协议是一种链路管理协议，而不是路由选择 协议。因此这项是错误的。 B、当局域网拓扑发生变化时，生成树协议能够重新计算并形成新的无环路的结构。这项是正确的。 C、生成树协议使得任意两个网桥或交换机之间、任意两个局域网之间只有一条有效的帧传输路径。这项是正确的。 D、生成树协议能够自动控制局域网系统的拓扑，形成一个无环路（loop-free）的逻辑结构，这项是正确的。

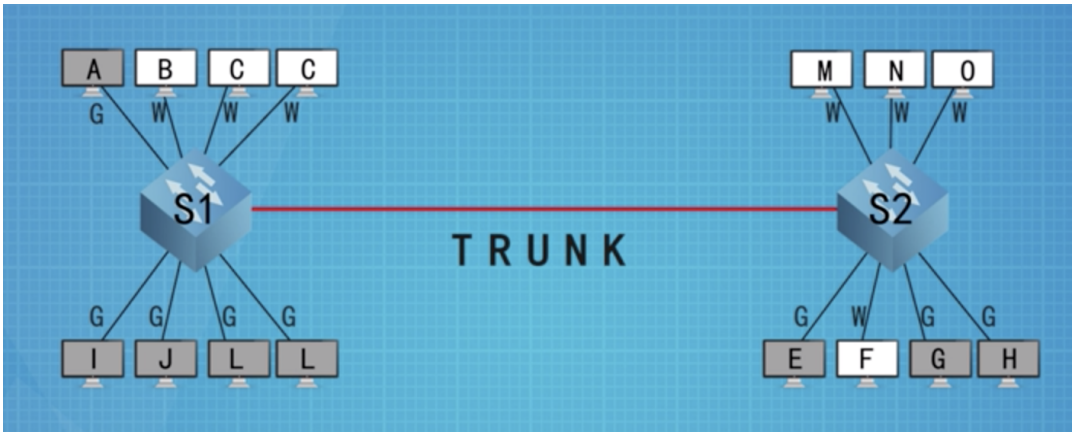
4.7.3 课程总结

- 为了可靠，采用冗余结构
- 冗余造成环路，会产生多帧传送、广播风暴和MAC地址库不稳定等问题
- 这些问题的根源就是有环路，STP就是打断环路，维持逻辑上的无环路
- STP可能产生非最优路径 (付出的代价)
- 当逻辑STP树上的某点出了故障，非指定端口将会被重新启用。
- 问题
  - 为什么需要冗余拓扑?
  - 冗余拓扑会带来什么问题?
  - 怎么解决这些问题?
  - STP是否消除了冗余拓扑带来的可靠的好处?

4.8 虚拟局域网

- 经典LAN
  - 物理LAN，地理位置靠近的工作站或楼层构成的LAN
- VLAN (虚拟局域网 - Virtual)
  - 一组逻辑上的设备或用户，不考虑地理位置而考虑组织架构。
- VLAN的三种实现方式
  - 基于MAC地址
  - 基于三层协议
  - 基于端口 (最常用)
    - 交换机内部有一张 VLAN 成员的配置表，知道哪些端口属于哪些VLAN。
- IEEE 802.1Q (帧标记方法)

- 解决问题：如何处理一个VLAN用户跨越不同的交换机？
- 每个帧在干线起点打上标签，干线终点拆除这个标签。[帧涂上其所在VLAN的颜色]
- 帧结构中多加了一个字段，表明这个帧所属的VLAN ID
  - 

- 举例
  - 

    - S1、S2为交换机，所有灰色机器为VLAN1中工作站，所有白色机器为VLAN2中工作站。
    - 现在 A 向所有 VLAN1 中的工作站发送一个帧。
    - ① 一开始帧没有 VLAN ID 标记，帧到达 S1 之后，S1 将帧广播给了直接与 S1 相连的VLAN1中的工作站，即 I、J、L、L。
    - ② 之后 S1 将该帧的 VLAN ID 打上了标记 (干线起点)，并通过干线将该帧转发给了 S2。
    - ③ 帧到达 S2 之后 (干线终点)，S2 将该帧的 VLAN ID 去掉，然后转发给了与 S2 相连的所有 VLAN1 中的工作站。
- 小结
  - VLAN 是一组逻辑上的用户。
  - 一个 VLAN 特性等同于一个物理 VLAN。
  - 一个 VLAN 就是一个广播域。
  - 有了 VLAN，可使用二层交换机实现广播域的分割。
  - 帧标记法可以让帧穿过交换机的干线。
  - 问题
    - 什么是 VLAN?
    - VLAN 成员可以跨越交换机吗?
    - 帧标记法有什么用?
    - 最常用的帧标记法是哪个?
    - 采用帧标记法的帧是如何通过干线的?

4.9 二层设备

4.9.1 网卡 (NIC)

- 网卡特点
  - 网卡 (NIC - Network Interface Card)，第一层和第二层的设备
  - 为主机提供介质的访问
  - MAC地址烧录在网卡的ROM中
- 网卡功能
  - 第一层提供的功能
    - Signaling：再生信号，创建信号和与介质的接口
    - 内建转发器
  - 第二层提供的功能
    - Logical Link control (LLC)：提供和上层的通信
    - Naming：提供一个独特的 MAC 地址标识符
    - Framing：成帧过程，封装过程的一部分，为传输比特流打包
    - Media Access Control (MAC)：为访问共享介质提供访问策略
- 网卡分类
  - 按计算机类型分类
    - 普通PC网卡、笔记本网卡
  - 按网络类型分类
    - 以太网卡 (最常用)、TOKEN Ring NIC、FDDI NIC
  - 按介质类型分类
    - Twisted pair、coax、fiber
  - 按系统总线类型分类
    - PCI、ISA

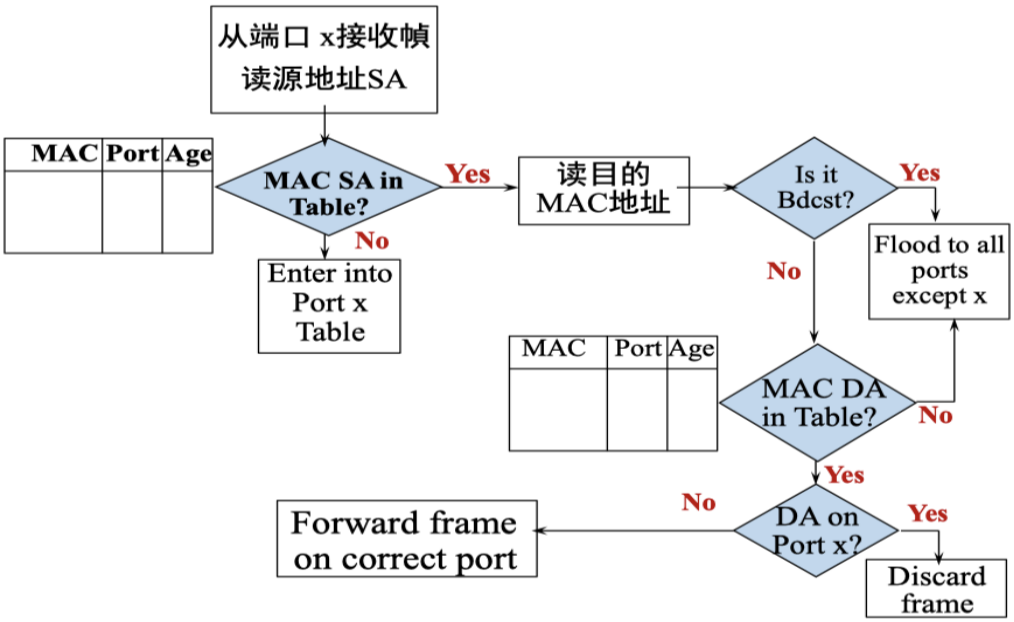
4.9.2 网桥

- 功能
  - 连接不同的LAN网段
  - 通过过滤部分交通流量，减少冲突的机会，改善网络性能。
  - 以网段分流交通，基于 MAC 地址过滤流量。

4.9.3 交换机 (Switch)

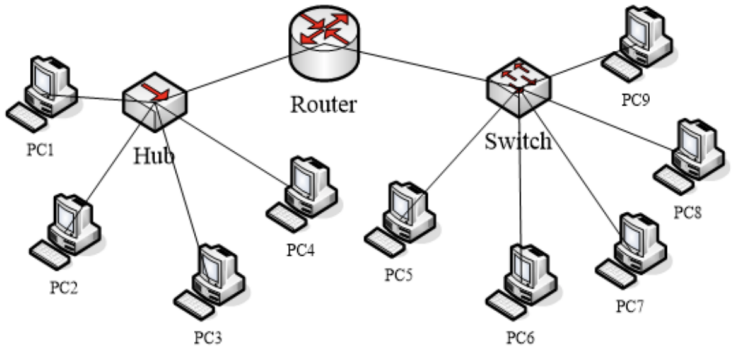
- 特点
  - LAN 交换机是多端口网桥，基本工作原理与网桥一致。

- 连接 LAN 网段，并使用一张 MAC 地址表，来决定每一帧转发的端口。
- 交换机常被用来替换 集线器 (Hub)，以改善现有网络性能。
  - 将交换机作为以太网星型拓扑的中心。
- 交换机具有比网桥更高的交换速度。
- 支持新的功能，如 VLAN。

- LAN 交换机中地址表的维护
  - 动态更新
    - 直接读取数据包的源地址信息，存入 CAM，如在 CAM 中没有找到所需的地址，则添加到 CAM 中。
  - 删除过时的地址记录 (时间标记 - 时戳)
    - 每增加一条记录，为它打上时戳
    - 每引用或找到某条记录，为它打上新的时戳
    - 周期性扫描 CAM，当某条地址记录超过一定时间没被引用，则删除该条记录。
- 交换机工作原理
  - 三选一操作
    - flooding (泛洪/广播)：当目的地址未知或为广播地址时，桥发送帧到除源端口外的每个端口。
    - forwarding (转发)：对于已学到的目的地址，桥直接发送帧到对应的目的设备所在端口。
    - filtering (过滤)：如果目的地址和源地址在同一端口，桥将丢掉该帧。
  - 每一帧都进行的操作
    - learning (后向学习)：通过读取每个帧的源地址和对应源端口来学习连在网段上的每个设备的地址。
  - 交换机处理到达帧的过程
    - 

- 交换机作用
  - 使网络段处于无冲突环境
    - 微分段

        - LAN被交换机分隔开的网段，在一个大的冲突域中产生无冲突域。交换机的每一个端口，只接一个工作站，而不是通过 Hub 再接大量工作站。
    - 虚拟线路
        - 在交换机内部把段连接成一个虚拟网络的电路，只在需要时才使用。
  - 广播域与冲突域
    - Hub所有端口都在同一个广播域、冲突域内，所以Hub不能分割冲突域和广播域。
    - Switch所有端口都在同一个广播域内，而每一个端口就是一个冲突域，所以交换机能分割冲突域，但分割不了广播域。Vlan技术可以隔离广播域。
    - Router的每个端口属于不同的广播域、冲突域
    - 10 多选 (2分) 下图中，有多少个广播域？有多少个冲突域？ 得分/总分

      A. 7 ✔1.00/2.00
      B. 2 ✔1.00/2.00
      C. 5
      D. 3

- 交换机的三种交换模式
  - **存储转发**
    - 交换机把整个帧接收下来，计算它的校验和，检查这个帧没有错误了，再按照MAC地址表的指示把它转发出去或做对应的处理。
    - 特点：延迟大、出错率小
  - **直通交换** (贯穿交换、虫孔交换)
    - 交换机直接读取到帧的目的地址，根据MAC地址表直接将帧进行转发。
    - 特点：延迟小 (帧在交换机内部停留时间非常短)、出错率大。
  - **无分片交换** (无碎片交换)
    - 直到读取到帧上的64个字节之后，才进行转发，保证不转发碎片帧。
    - 特点：在延迟和出错率上做了一个折中。
- 交换机和集线器的比较

-

功能	网桥	中继器
再生信号	Yes	Yes
连接采用不同MAC协议的网段	Yes	No
隔离冲突域	Yes	No
根据帧头的物理地址转发帧	Yes	No
丢弃损坏帧	Yes	No

4.9.4 课程总结

- 网卡的主要功能是什么?
- 交换机的基本工作原理是怎样的?
- 交换机内部的 MAC 地址表示怎样建立和维护的?
- 交换机能创建的最小冲突域多大?
- 三种交换方式的优缺点是什么?

第五章 网络层 (上)

5.1 网络层引言

5.1.1 网络层概述

- 网络层主要功能
  - 源端数据包 (分组)，一路送达到目的机
- 如何实现功能
  - 封装源数据
  - 找到一根好的路径 (路由) [路由选择协议、路由器]
  - 识别目的机
- 可能会遭遇的问题
  - 地址不够用 (CIDR、IPv6、NAT)
  - 丢包 (ICMP)
  - 拥塞 (拥塞控制)

5.1.2 数据报网络与虚电路网络

- 源和目的之间的网络的类别
  - 数据报网络 (提供无连接的服务)
  - 虚电路网络 (提供面向连接的服务)

- 数据报网络与虚电路网路的比较
  -

比较项目	数据报子网 (无连接服务)	虚电路子网 (面向连接服务 - virtual circuit)
建立电路	不需要建立	一定要建立
地址信息	每个分组包含完整的源地址和目的地址，每个分组独立寻径	每个 VC 包含一个很短的 VC 号码
状态信息	路由器不保留任何连接状态和信息	每个 VC 都要求路由器建立表项
路由	每个分组独立选择路由	每个分组沿建立 VC 时确定的路由
路由器失效影响	没有影响，只有系统崩溃时丢失分组	所有经过失效路由器的 VC 都终止
服务质量与拥塞控制	很难实现	总资源 (带宽、缓存) 足够的情况下，采用提前给每个 VC 分配资源的方法，很容易实现。

5.1.3 课程总结

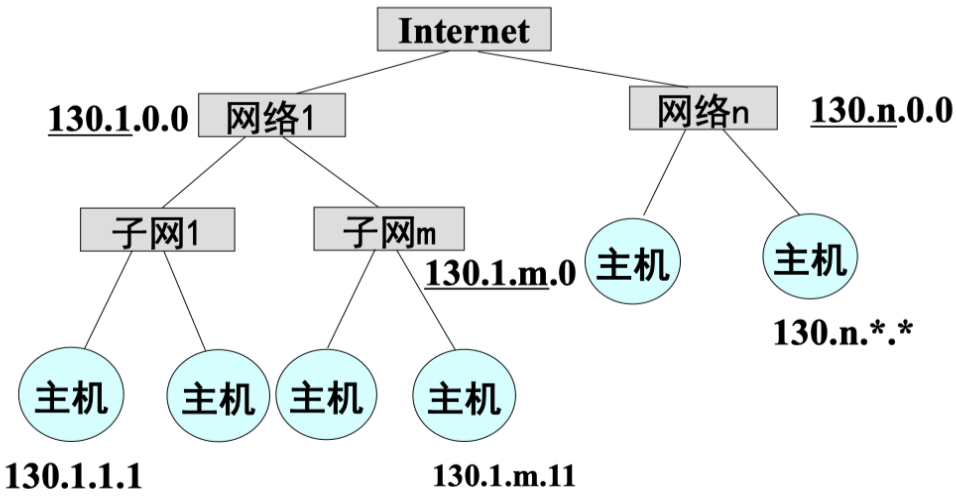
- 网络层的目标：把数据分组一路送到接收机。
- 网络层利用下层 (数据链路层) 提供的服务为它的上层 (传输层) 提供服务。
- 两种类型的网络分别提供两种类型的服务
  - 数据报网络：无连接的服务
  - 虚电路网络：面向连接的服务
- 问题
  - 网络层的主要目标是什么?
  - 虚电路网络上中，是否不需要进行路径选择?
  - 数据报网络中，路由表从何而来?
  - 数据时怎样穿过数据报网络的?
  - 数据时怎样穿过虚电路网络的?

5.2 IP地址

5.2.1 IP概述

- IP (Internet Protocol)
  - 特点
    - IP是互联网协议，是协议栈的核心之一
    - IP是胶水，将整个互联网互联起来。
    - IP为路由提供所需要的信息，因此也被称为被路由的协议
  - IP 的任务
    - 提供一种尽力而为 (best-effort) 地把数据从源端传输到接收方的方法。
- IP包括两个内容
  - IP地址 (用来标识收发主机)
  - IP分组 (信息分装的格式)

5.2.2 IP地址表示形式

- 特点
  - IP地址由32位二进制表示，提供大约 $2^32 = 43$ 亿个地址。
- IP地址表示形式
  - 点分十进制
    - 将32位分成4个8位组，相邻 8 位组之间用 '.' 隔开
    - 每个8位组转换成十进制数，从 0~255
    - 举例
      - IP地址：10000011 01101100 01111010 11001100
      - 点分十进制：131 . 108 . 122 . 204
- IP地址的层次结构
  - 

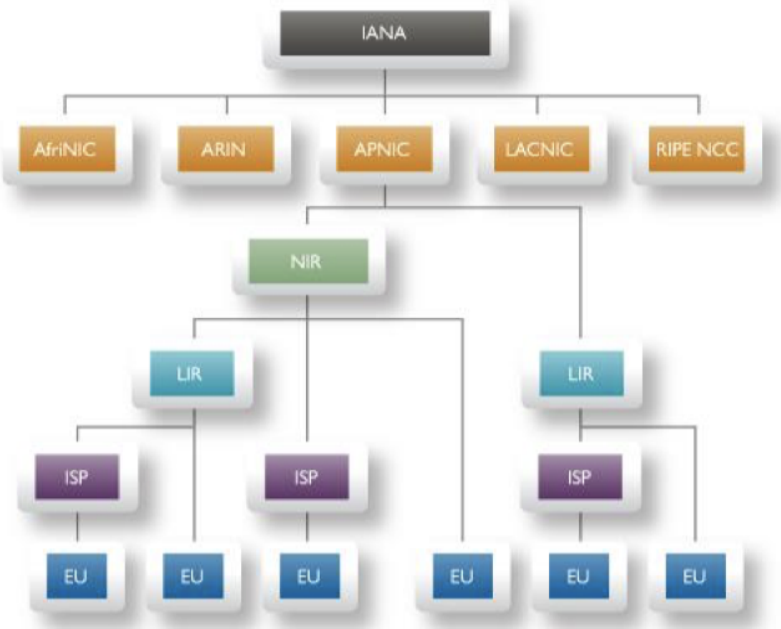
5.2.3 IP地址的分类

- IP地址由两部分组成：网络地址、主机地址 (前部分为网络地址，后部分为主机地址)
- A类地址
  - 0xxxxxxx . xxxxxxxx . xxxxxxxx . xxxxxxxx

    - 前 1 字节标识网络地址部分，后 3 字节标识主机地址部分。
    - 第 1 个最高位固定为0，第 1 字节变化范围为 0~127，因此 A 类地址一共只有 128 个。
    - 每个网络最多可容纳 $2^{24}-2$ 台主机。
- B类地址
    - 10xxxxxx . xxxxxxxx . xxxxxxxx . xxxxxxxx
    - 前 2 字节标识网络地址部分，后 2 字节标识主机地址部分。
    - 前 2 位固定为 '10'，第 1 字节变化范围为 128~191，以此辨识 B 类地址。
    - 每个网络最多可容纳 $2^{16}-2$ 台主机。
    - 具有 B 类地址特征的网络总数为 2^{14} 个。
- C类地址
    - 110xxxxx . xxxxxxxx . xxxxxxxx . xxxxxxxx
    - 前 3 字节标识网络地址部分，后 1 字节标识主机地址部分。
    - 前 3 位固定为 '110'，第 1 字节变化范围为 192~223，以此辨识 C 类地址。
    - 每个网络最多可容纳 $2^{8}-2=254$ 台主机。
    - 具有 C 类地址特征的网络总数为 2^{21} 个。
- 三类 IP 地址的比较
    - 通常根据第一个字节的数值来判断 IP 地址的类别。
    -

类别	网络数	主机数/网络	最高字节取值范围	网络规模
A 类	128	1600 万	0 - 127	大型
B 类	1.6 万	6.5 万	128 - 191	中型
C 类	200 万	254	192 - 223	小型

- 保留的 IP 地址
    - D类地址
        - 1110xxxx . xxxxxxxx . xxxxxxxx . xxxxxxxx
        - 主要用于组播，(224.0.0.0 ~ 239.0.0.0)
    - E类地址
        - 11110xxx . xxxxxxxx . xxxxxxxx . xxxxxxxx
        - 保留用于将来科研使用，(240.0.0.0 ~ 254.0.0.0)
    - 网络地址
        - 主机部分全部为 "0" 的IP地址，上述 5 类网络中均有对应的网络地址
    - 广播地址
        - 主机部分全部为 "1" 的IP地址，上述 5 类网络中均有对应的网络地址
    - 0.0.0.0
        - 32位全为 0 的地址，标识这个主机、这个网络。也是一些路由器指定的默认路由。
    - 255.255.255.255

        - 32位全为 1 的地址，为泛洪广播地址。
        - 但是由于对整个互联网进行泛洪过于可怕，该地址退化成了本地广播地址，即 Local Broadcast。
    - 127 开头的地址
        - 第一个字节为127的地址，表示环回地址 (Lookback Network)
        - 如 127.0.0.1，即为本机地址，localhost
    - 169 . 254 . x . x
        - 表示非正常地址，不能和外界进行正常通信。
- 举例：分析一个IP地址，172 . 16 . 20 . 200
    - 根据第一个字节，可以判断这是一个B类地址
    - 因此网络部分：172.16
    - 主机部分：20.200
    - 网络地址：172.16.0.0
    - 广播地址：172.16.255.255
- IP地址的分配
    - 由ICANN域名和地址分配机构负责对IP进行层级分配
    - 

    - 当今世界上，美国共3亿人口，有16亿个IP地址。而中国有7亿网民，只有3亿多个IP地址。

5.2.3 课程总结

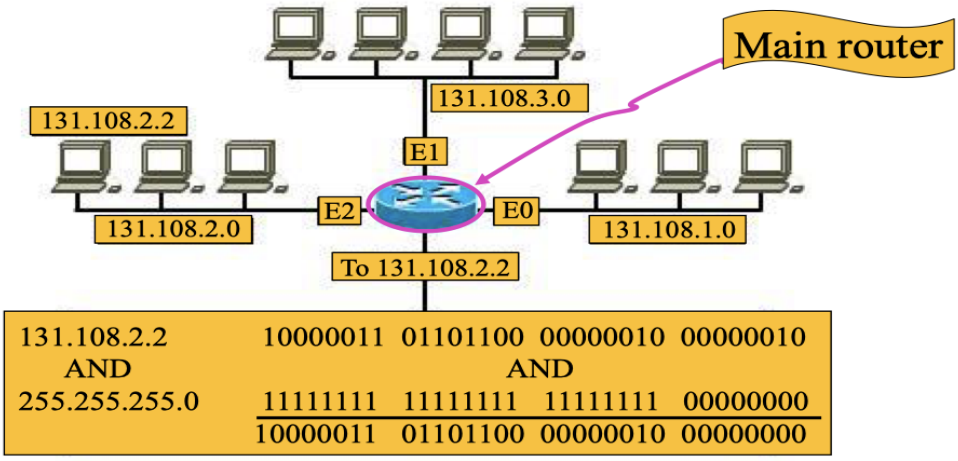
- IP地址共有5类。
- 保留地址有全0、全1、广播地址、网络地址、环回地址等。
- 问题
    - IP地址为什么要按类别分配?
    - 什么是保留地址?
    - 保留地址有哪些?

5.3 子网规划

5.3.1 子网概念

- 现状：局域网不断增长，越来越难于管理，必须将它分割成子网
- 一个网络被分割成几个部分 (子网)，但是在外界看来，该网络仍被看成一个整体 (体现在路由表上，就是外部的路由器只对应一条路由)
- 这也允许不同的子网在一个组织内部连接起来。

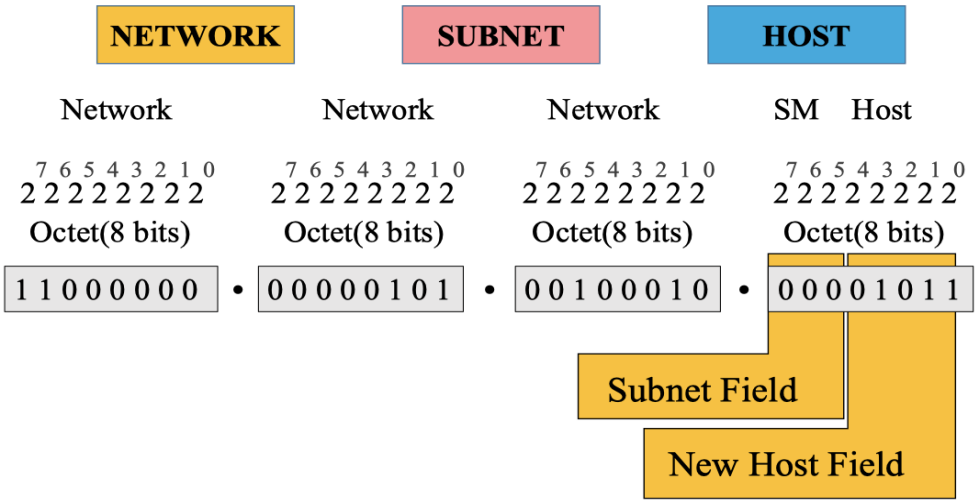
5.3.2 子网掩码

- 子网掩码作用
  - 路由器使用子网掩码决定分组往哪个子网转发
- 子网掩码的两种表示方法
  - 点分十进制表示 (1表示网络位，0表示主机位)
    - 如 255.255.255.224
  - "/网络位数+子网位数"表示
    - 如 202.10.23.102/27，27即为子网掩码
- 如何得到目的网络地址
  - 路由器采用 "AND" 操作 (目的IP和子网掩码)，得到目的网络地址
  - 优点：路由器不必记录全部主机的IP地址，缩减了路由器的规模
  - 

- 子网掩码举例
  - A类网络：255.0.0.0 或 /8
  - B类网络：255.255.0.0 或 /16
  - C类网络：255.255.255.0 或 /24
  - 其它网络地址举例
    - 网络地址：166.111.0.0 (主机位全0为网络地址)
    - 子网掩码：166.111/16
    - 广播地址：166.111.255.255 (主机位全1为广播地址)

        - 主机地址：166.111.0.1~166.111.255.254

5.3.3 子网规划

- 子网划分
    - 划分子网实际上建立了一个由网络、子网和主机构成的三级层次结构
    - 根据用户需求，划分出合适数量和规模的子网
    - 规划要求基于每个子网的规模、每个子网的主机数量和主机地址的分配方式进行决策
- 子网位
    - 增加子网位来表示子网
    - 

    - 子网位如何而来 —— 借位
        - 构建子网是通过从网络地址的主机部分借位来进行
        - 广播地址：主机部分为全 "1" 的 IP 地址
        - 子网络地址：主机部分为全 "0" 的 IP 地址
- 子网规划原则
    - 子网规划：将大网络分割成小网络
    - 借位原则
        - 从主机域的高位开始借位
        - 主机域至少保留 2 位 (网络位+广播占两位，因此主机位至少 4 位)
    - 子网规划考虑两个因素
        - 所需子网数量
        - 所需主机地址的数量
    - 可用主机数量为 $2^n - 2$
        - n为剩余的主机位个数
        - -2，子网中不能使用子网ID和广播地址
- 子网规划举例
    - 原网络地址：192.168.1.0/24
    - 现从主机位中取前两位作为子网位，由此得到4个子网地址
        - 192.168.1.00 000000 = 192.168.1.0/26
        - 192.168.1.01 000000 = 192.168.1.64/26

        - 192.168.1.10 000000 = 192.168.1.128/26
        - 192.168.1.11 000000 = 192.169.1.192/26
    - 主机位从开始的8位变成了6位，主机数一共有 $2^6 - 2 = 62$ 个。

5.3.4 课程总结

- 子网让两层IP地址结构变成了3层
    - 网络、子网和主机
- 子网掩码两种表示形式：点分十进制、/网络位+子网位
- 路由器执行按位 "与"，来确定目的网络地址
    - 路由表的查找入口
- A/B/C 三类地址的缺省子网掩码
    - A：255.0.0.0/8
    - B：255.255.0.0/16
    - C：255.255.255.0/24
- 通过向主机位借位来创建子网
- 借位原则
    - 高位开始借
    - 至少保留2位主机位

5.4 IP寻址

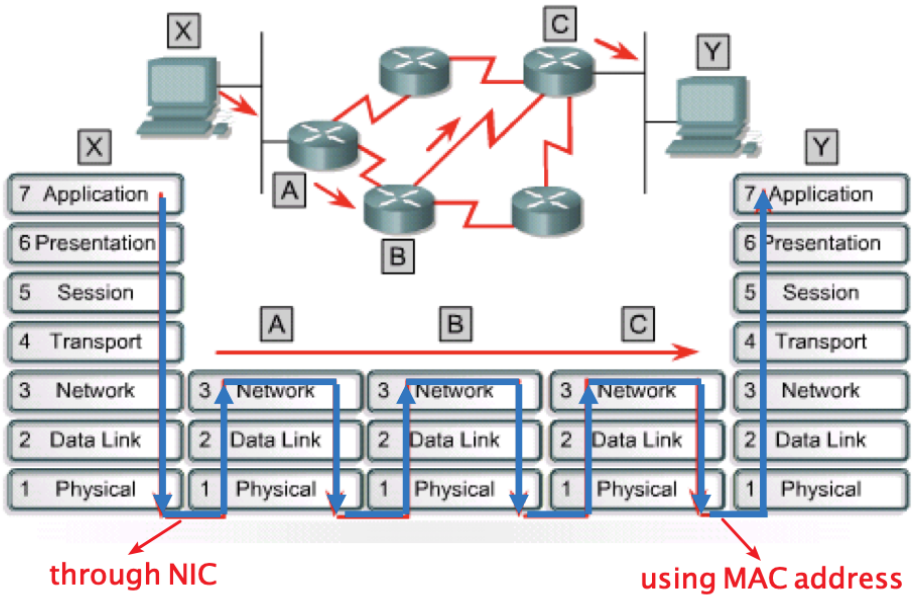
5.4.1 IP寻址与MAC寻址概念

- IP寻址
    - 定义：根据目的IP地址，找到目的网络的过程
- 路由器
    - 执行IP寻址的主要设备是路由器。
    - 一个路由器称为一跳。(Hop)
    - 当数据到达最后一跳时，即最后一个路由器。路由器将分组拆开查看目的IP地址，然后将分组发往其对应的网络。
        - 到达对应网络之后，进行MAC寻址。
- MAC寻址
    - 定义：根据目的MAC地址，找到目的机的过程
    - 执行MAC寻址的主要设备：交换机

5.4.2 IP寻址与MAC寻址的比较

- 适用的网络范围不同
  - MAC寻址：小型网络，在局域网中对应目标
  - IP寻址：互联网，依托路由器逐跳达到目的网络
- 所依赖的地址结构不同
  - MAC地址：平面地址 (全球唯一)
  - IP地址：结构化地址
- 所处的OSI参考模型层数不同
  - MAC寻址：数据链路层
  - IP寻址：网络层
- 地址数目的限制
  - MAC地址：无耗尽危险 (7万亿个)
  - IP地址：正在耗尽
- 两种地址的格式不一样
  - MAC地址：十六进制
  - IP地址：点分十进制

5.4.3 IP寻址具体过程

- 路由器进行IP寻址的具体过程
  - 打开分组 (Packet) —— 只打开到第三层 (网络层)，提取目的IP地址
  - 定网络，查路由
    - 通过目的IP地址和子网掩码，进行按位与操作，得到目的网络。
    - 在路由表中查找目的网络。
  - 将分组重新封装，转发 (Next Hop)
- 举例
  - 

  - A、B、C路由器只会解封装到第三层，获取目的IP地址。
  - 所有在 C 子网上的机器，都会收到这个分组的比特流，但是由于MAC地址的存在，最后只有 Y 会接收这个比特流。

- 路由表
  - 定义：路由器转发数据的依据
  - 每个路由器中都有一张路由表，路由表包括目的网络地址、接口、代价、子网掩码、网关等信息
    -

目的网络	路径代价	下一跳	转出接口
202.112.18.0/24	5	192.168.1.1	Fe0/0
……	……	……	……

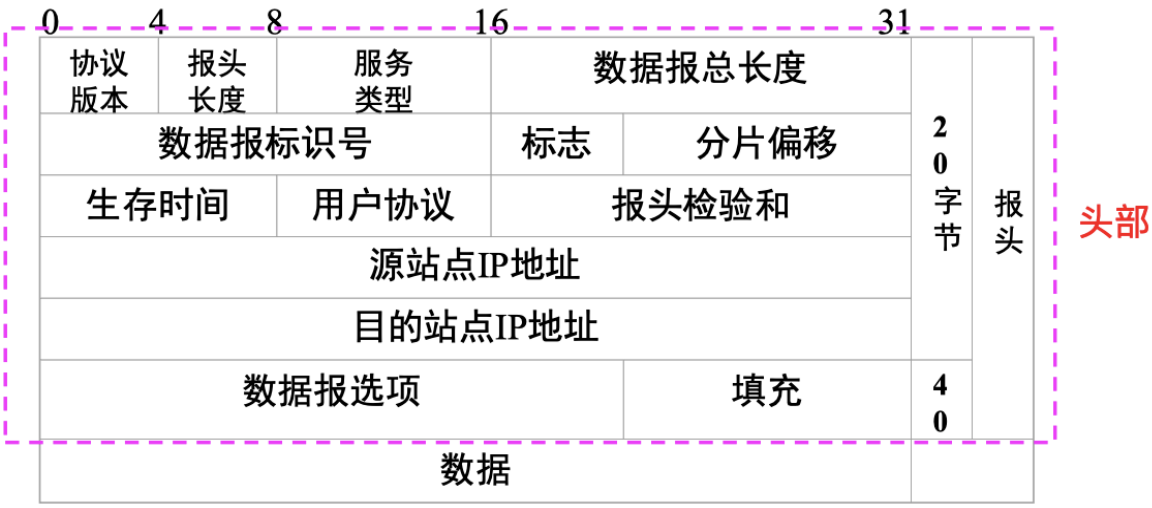
  - 除了路由表之外，路由器内部还有一张 ARP 表
    - ARP 表记录所有子网设备的IP-MAC映射 (包括网关)
  - 路由表可能因厂家的不同而不同

5.4.4 课程总结

- 两种寻址方式
  - IP寻址：找到目的机所在的网络
  - MAC寻址：通过MAC地址定位到目的机
  - 通常先通过IP寻址找到目的网络，再通过MAC寻址找到目的机
- 路由器收到分组的处理过程 (三步)
  - 解封装
  - 查表，确定目的网络
  - 封装后转发
- 路由器决策的依据：路由表
- 问题
  - 什么是IP寻址?
  - 什么是MAC寻址?
  - IP寻址和MAC寻址有什么不同?
  - 只用IP寻址，或只用MAC寻址是否可以完成找到目的机的任务?
  - 路由器处理分组的过程是怎样的?
  - 路由表是怎样的表？包括哪些主要字段?

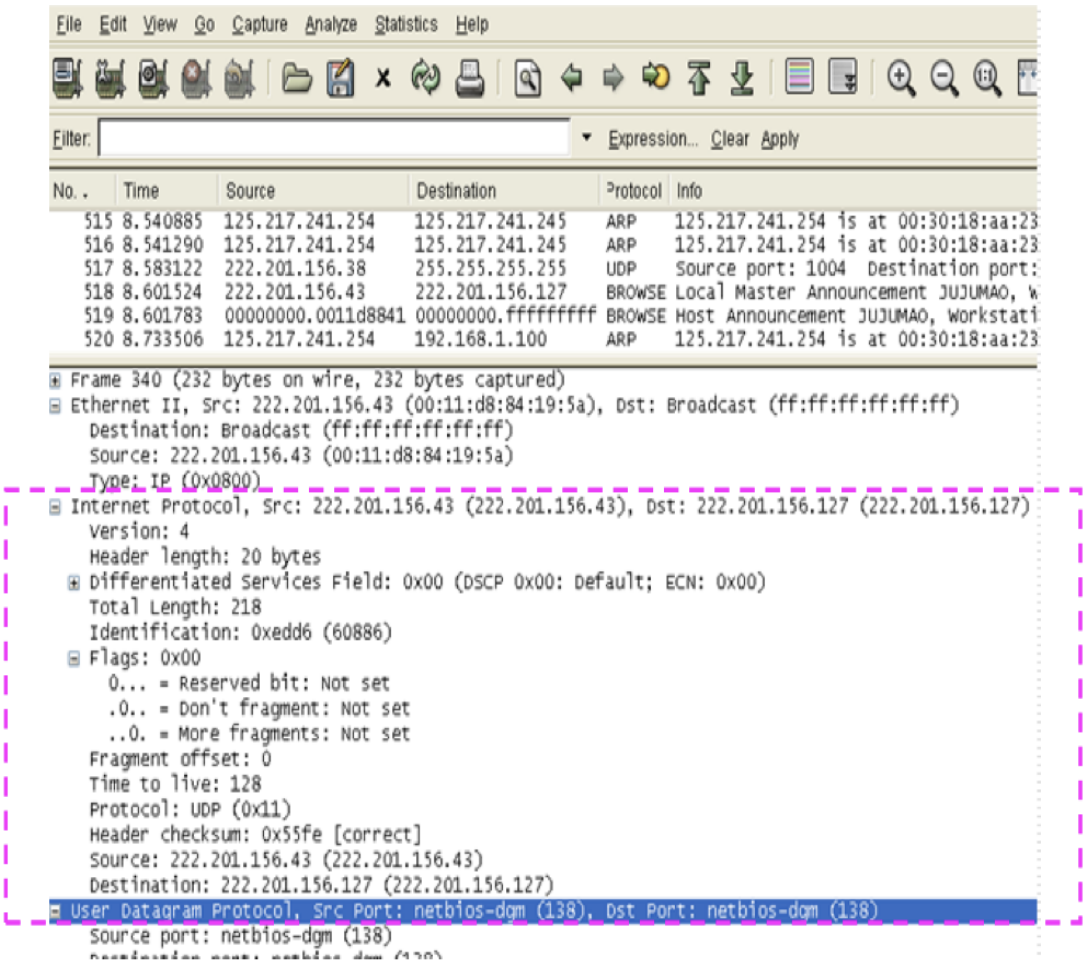
5.5 IP分组

5.5.1 IP分组格式

- IP分组由头部和数据构成
- 

- 不含选项的字段有12个，一共20字节。按照4个字节，32位为一行进行排列，则一共有5行。
- 第一个字段：协议版本 (4 bits)
  - 标明IP协议的版本号
    - IPv4：0100
    - IPv6：0110
- 第二个字段：报头长度 (4 bits)
  - 标明IP分组头部的长度。
  - 单位：32bit (4字节)，指向数据开始处。
  - 范围：0101~1111，即头部长度 20~60 字节
- 第三个字段：区分服务 (8 bits)
  - DS：Differentiated Service
  - 在互联网上基本没怎么使用，表示分组的优先级、丢弃级，用来保证服务质量、拥塞控制等。
- 第四个字段：数据报总长度 (16 bits)
  - 标明整个数据分组的总长度，包括数据头和数据。
  - 单位：字节
  - 范围：$\mathrm{MAX} = 65535$ 字节
- 第五个字段：数据报标识号 (16 bits)
  - 标明当前数据报的序列号
  - 由发送者分配，以便接收方可以依据这个标识号对包进行重组。
- 第六个字段：标志 (3 bits)
  - 标明分组是否分片
  - 三位，第一位是一个不使用的位，第二位是DF (Don't Fragment)，第三位是MF (More Fragment)
    - DF = 1，表示不允许路由器分割该数据报。
    - MF = 1，表示还有更多的段。
- 第七个字段：分片偏移 (13 bits)
  - 帮助接收方对包进行重组。

- 第八个字段：生存时间 (8 bits) —— TTL (Time-to-Live)
  - TTL 维护一个计数器 (单位：一跳)，每次经过一个路由器，TTL-1
  - 当TTL递减为0时，数据报就被丢弃，同时路由器向源发送一个超时消息，表明这个分组还未找到目的，但由于超时被丢弃了。
  - TTL目的：防止分组在网络中无限循环。
- 第九个字段：用户协议 (8 bits)
  - 标明传输层协议。常见数值为 17(UDP) 与 6(TCP)。
  - 17：上层采用UDP协议，搭载UDP数据段
  - 6：上层采用TCP传输层协议，搭载TCP数据段
- 第十个字段：报头校验和 (16 bits)
  - 采用互联网校验和，针对分组头部计算校验和，用于验证分组头部的正确性
- 第十一个字段：源IP地址 (32 bits)
  - 标明分组来源的IP地址
- 第十二个字段：目的IP地址 (32 bits)
  - 标明分组发向的IP地址
- 第十三个字段：数据报选项 (长度不等)
  - 可变长字段。
  - 该字段允许主机支持不同的选项，比如安全、源路由、时戳等。但是很少被使用。
- 第十四个字段：填充 (长度不等)
  - 可变长字段，如果选项不是 32 bits 的整数倍，就需要增加一个填充字段，使得 填充字段+选项字段 = 32 bits 的整数倍。
- 举例

- 

5.5.2 课程总结

- IP分组的重要字段
  - 头部长度、总长度
  - TTL
  - 用户协议
  - 目的IP
- 问题
  - IP分组中包括哪些主要字段?
  - 如何得知IP分组的净载荷中封装的是什么数据段?
  - 当路由器发现一个分组的 $\mathrm{TTL}-1=0$ 时，它会如何对待该分组?
  - 如何得知一个分组是否含有选项?

5.6 IPv6的发展

5.6.1 IPv4的弊端

- IPv4先天不足

    - 地址危机
        - 约43亿地址，在分类中浪费了大量的IP地址。
        - 2011.2.3 IANA地址池枯竭。
    - 路由表的膨胀
        - 路由表中维护的数据成千上万条，过度膨胀
    - 安全问题
        - IPv4的诞生从未考虑过安全问题
    - 配置复杂
    - QoS和性能问题
    - 端到端业务模式
    - 移动性支持不够

5.6.2 IPv6的诞生

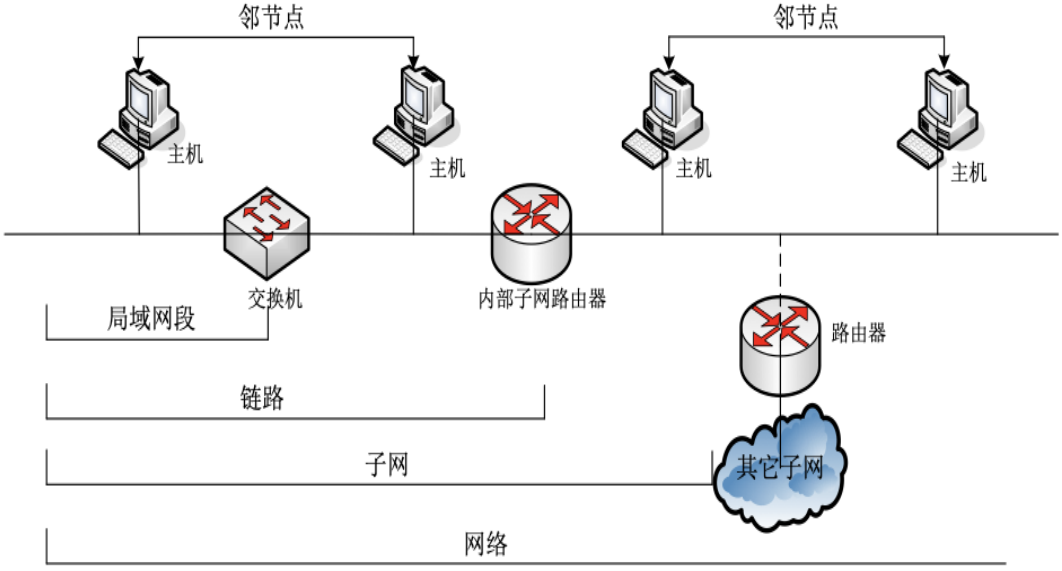
- IPv6的目标
    - 支持几十亿台主机
    - 缩减路由表的规模
    - 简化协议，让路由器可以更快地处理分组
    - 提供比IPv4好的安全性
    - 更加关注服务类型 (尤其是实时数据)
    - 允许通过指定范围来支持组播传输
    - 允许主机在不改变地址的情况下能够漫游
    - 允许协议具有扩展性
    - 允许新老协议共存多年
- IPv6标准组织 —— IETF
    - 制订IPv6规范和标准
    - 提供关于IPv6的指导
- 一些命名问题
    - IPv1、IPv2、IPv3、IPv5 在实验中都被使用过了。

5.6.3 课程总结

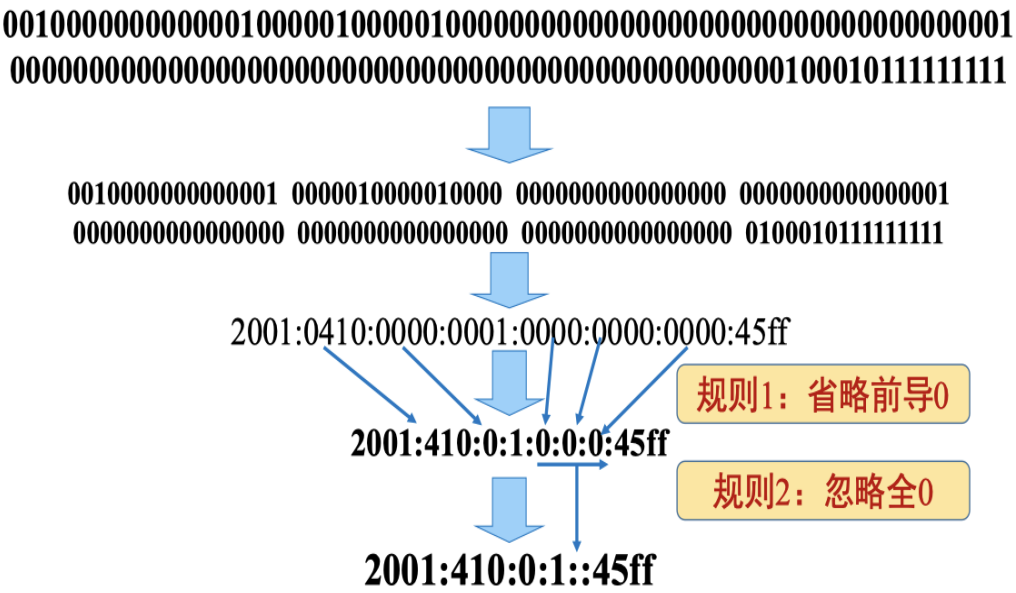
- IPv4有哪些问题?
- 什么是IPv6?
- IPv6的相关标准主要由哪个标准组织推出?

5.7 IPv6地址

5.7.1 IPv6地址表示

- IPv6基本术语
  - 基本术语 —— 局域网段、链路、子网、网络、邻节点 (链路上相邻节点)
  - 

- IPv6地址空间
  - IPv6一共128位，$2^{128} = 340 * 10^{36}$ 。
  - 全球人均每人 $5 * 10^{28}$ 个，每平方厘米 $6.7 * 10^{19}$ 个地址。
- IPv6地址表示 —— 冒分十六进制
  - 每16位一组，每组用4个16进制数表示，一共8组
  - 规则
    - 省略前导0 —— 忽略每组中的前导0
    - 忽略全0 —— 中间都是0，则用双冒号进行表示 (双冒号至多使用一次)
    - 地址前缀长度用 "/xx" 表示，如 1::1/64
  - 同一个地址有不同的表示方法
    - 0001:0123:0000:0000:0000:ABCD:0000:0001/96
    - 1:123:0:0:0:ABCD::1/96
    - 1:123::ABCD:0:1/96
    - 地址中至多出现一次双冒号，否则无法确定组数。
  - 举例

- 

5.7.2 IPv6地址分类

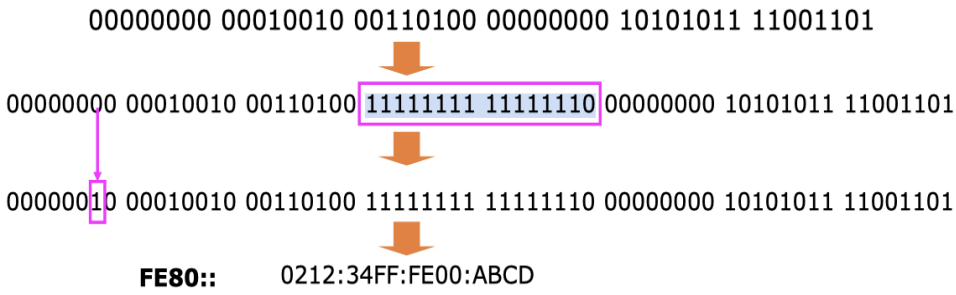
- 单播地址 (Unicast Address)
  - 链路本地地址
    - 只用在单一的链路上。
    - 如果分组中目的地址或源地址含有链路本地地址，则该数据包不会转发到其它链路上，只在本链路上通信。
- 组播地址 (Multicast Address)
- 任播地址 (Anycast Address)
- 特殊地址
  -

地址类型	二进制前缀	IPv6标识
未指定	00...0 (128 bits)	::/128
环回地址	00...1 (128 bits)	::1/128
组播	11111111	FF00::/8
链路本地地址	1111111010	FE80::/10
网点本地地址	1111111011	FEC0::/10
全局单播	（其他）	

- 链路本地地址的生成
  - 前64位：FE80:0:0:0
  - 后64位：EUI-64地址 (由MAC地址生成)
    - MAC地址一共48位，在中间分开，然后加上 11111111 11111110 16位构成64位
    - 再将第7位改为1，即可构成 EUI-64 地址

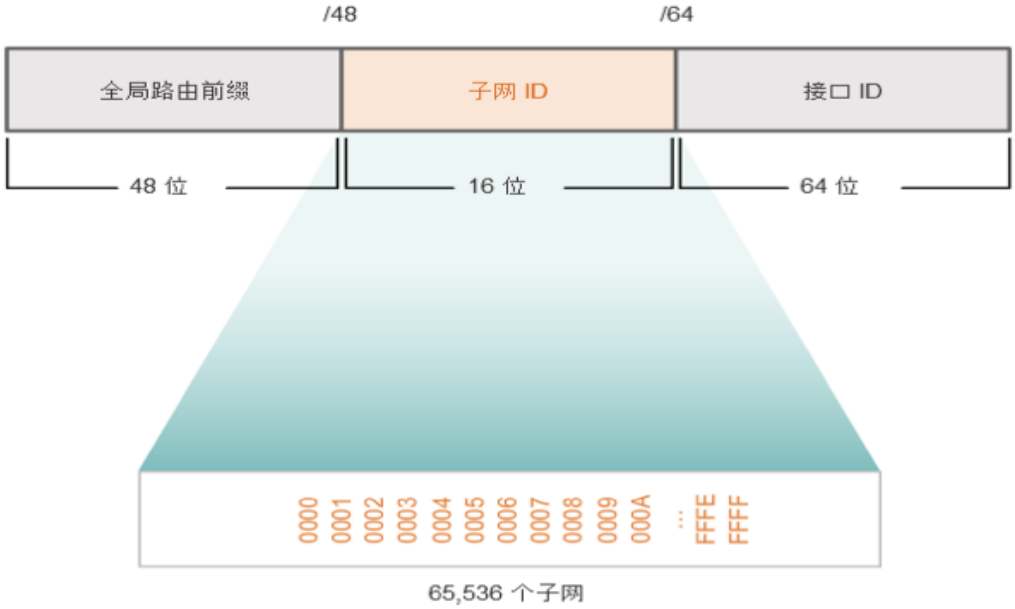
一台主机的MAC地址是：0012:3400:ABCD，试求其生成的链路本地地址。

解：首先将MAC地址写成二进制形式：

- 

5.7.3 IPv6子网规划

- IPv6子网划分根据路由器数量及它们所支持的网络来构建寻址分层结构
- 子网部分一共16位，支持65536个子网划分。

- 

5.8 IPv6分组

5.8.1 IPv6分组格式

-

<table>
<tr><td>版本(4)</td><td>业务等级(8)</td><td colspan="3">流标记(20)</td></tr>
<tr><td colspan="2">净荷长度(16)</td><td colspan="2">下一个头(8)</td><td>跳数限制(8)</td></tr>
<tr><td colspan="5">信源地址(128)</td></tr>
<tr><td colspan="5">信宿地址(128)</td></tr>
</table>

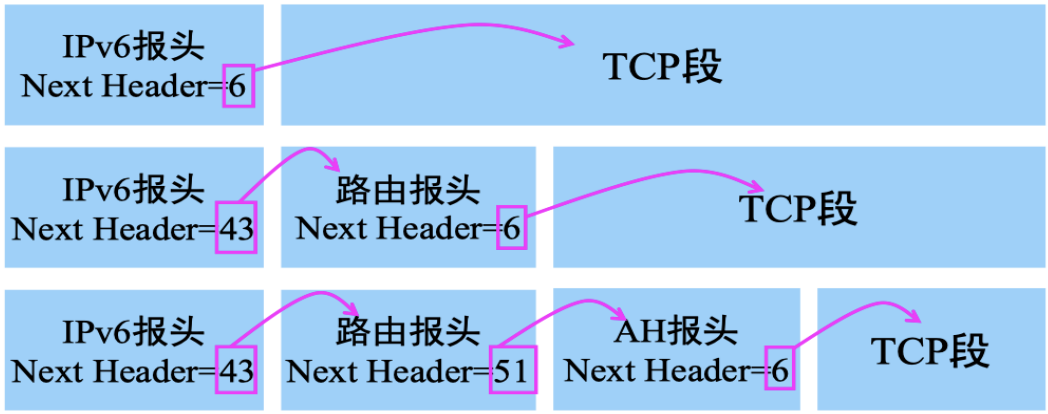
- IPv6头部一共40个字节。
- 第一个字段：版本 (4 bits)
  - 标明版本号，0100为IPv4，0110为IPv6
- 第二个字段：区分服务 (8 bits)
  - 表示数据分组的服务类型，表示该分组的重要程度、优先级
- 第三个字段：流标签 (20 bits)
  - 为源端和接收端提供了一种建立伪连接的方式，即源端和接收方把一组具有同样需求并希望得到网络同等对待的数据分组打上同样的标签。
- 第四个字段：净荷长度 (16 bits)
  - 指明了搭载的数据的长度 (不含报头)，单位：字节
  - 总长度：$2^{16}-1=65535$ 字节
- 第五个字段：下一个头 (8 bits)
  - 指明了紧跟的扩展头类型。
  - 如果当前头是最后一个扩展头或无扩展头，则退化为 "协议"，如TCP(6)、UDP(17)
- 第六个字段：跳数限制 (8 bits)
  - 与 IPv4 中的 TTL 一致，防止分组无限循环。
- 第七、八个字段：源、目的 IPv6 地址。
- IPv6去除了校验和与分段，每次用 PMTU 算法计算出链路上的最小MTU。

5.8.2 IPv6拓展头

- 拓展头分类
  - 一共有 6 种拓展头。
  -

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

- 扩展头使用规定
  - 扩展头是可选的，可以有多个扩展头，但是必须按一定的顺序排列。
  - 扩展头有固定的格式，其他扩展头包含可变数目的可变长度域。
  - 每个可变项都被编码成 (Type, Length, Value) 三元组。

- 举例
  - 

5.8.3 课程总结

- IPv6分组由基本头部、扩展头部和载荷三部分构成，下一个头 (Next Header) 字段描绘了三者间的关系。
- 相比 IPv4分组，IPv6的字段精简了，但是头部总长度并没有减少。
- 问题
  - IPv6分组基本头部包括了哪些字段?
  - 相比IPv4分组，IPv6分组做了哪些删改?
  - 相比IPv4分组，IPv6分组做了增加吗？增加了什么字段?

5.9 IPv6过渡技术

5.9.1 IPv6过渡时代

- IPv6过渡的三个阶段
  - IPv6孤岛
  - IPv4和IPv6均大量存在
  - IPv4孤岛
- 三类基本过渡技术
  - 双协议栈
  - 隧道
  - 地址转换技术
- 检测是否使用了IPv6的网络
  - http://ipv6.ustb.edu.cn/，如果看到了跑动的猎豹，则为IPv6的网络。
  - http://www.kame.net，看到游动的乌龟，也为IPv6网络。

5.9.2 双协议栈 (Dual Stack)

- 网络设备，服务器等必须支持双栈
- 主机端设备：可仅IPv6，也可仅IPv4
- 主机端设备如果支持双栈
    - 优先IPv6访问
    - 不通，再切换IPv4访问

5.9.3 隧道 (Tunnel)

- IPv4分组搭载IPv6分组 (IPv6分组作为纯数据)，作为普通IPv4分组穿过IPv4网络。
- 反过来，IPv6分组搭载IPv4分组 (IPv4分组作为纯数据)，作为普通IPv6分组穿过IPv6网络。
- 适用于第一、三阶段的过渡时期。

5.9.4 地址转换技术

- 从IPv4转换到IPv6，或反过来。不仅发生在网络层，还有传输层和应用层。
- 当双栈和隧道都无法使用的时候才使用，适用于纯IPv4节点和纯IPv6节点间的通信。

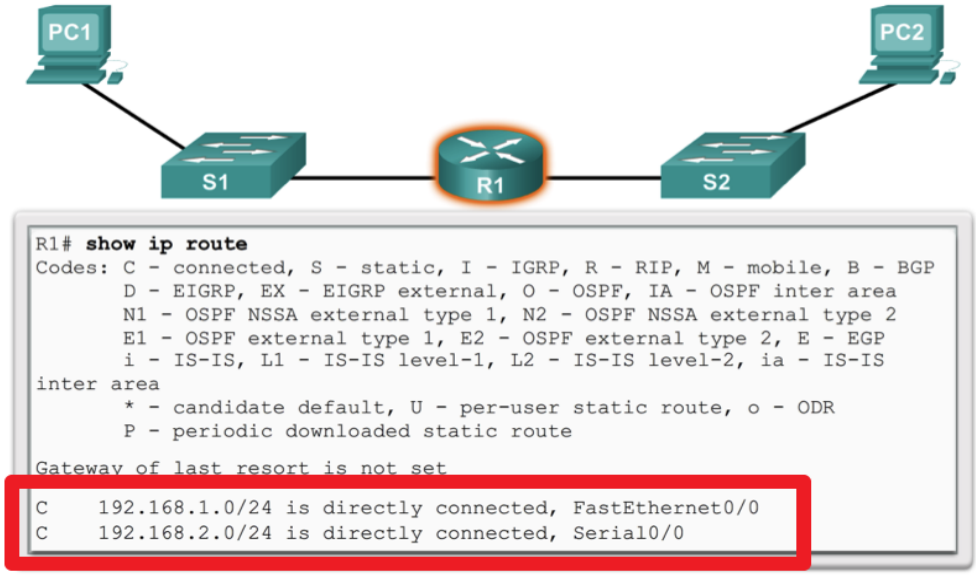
5.10 路由表概述

5.10.1 路由器功能

- 路由器转发分组的依据：路由表
- 路由表来源
    - 直连路由：路由器学习感知
    - 静态路由：人工配置
    - 动态路由：路由选择协议生成、更新和维护
- 路由器收到一个分组之后的操作：
    - 打开分组至第三层，提取出目的IP地址
    - 确定目标网络，查找路由表，进行按位"AND"操作
    - 重新封装，转发
        - TTL-1，置换源和目的MAC地址，计算校验和，可选分片等

5.10.2 直连路由

- 直连路由
    - 开启路由器接口之后，路由器自动将子网与接口进行匹配存储。
-



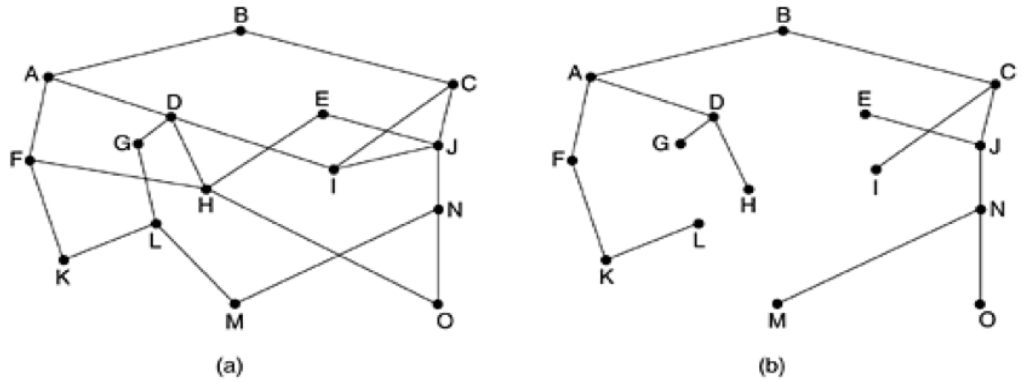
Cisco IOS command line interface (CLI) can be used to view the routing table.

5.10.3 静态路由

- 静态路由：管理员手工配置的路由
- 一种常用的静态路由：默认/缺省路由
  - 是默认的路径，找不到路的时候可以从这里通过
    - 避免错误丢包
    - 缩减路由表的规模
    - 减少路由器的运行负担
- 静态路由的优缺点
  - 优点
    - 在小型网络中容易实施
    - 非常安全，不会发送通告
    - 总是通过同一路径到达同一目的网络
    - 不需要CPU、RAM、带宽等开销
  - 缺点
    - 仅适用于简单网络或特殊用途 (比如默认路由)
    - 随着网络增大，配置复杂性显著增长
    - 需要人工更新和维护

5.10.4 动态路由

- 动态路由：由路由选择协议动态地建立、更新和维护的路由
  - 适合大型的、经常变动的网络，需要维护开销
  - 减少了网络管理员的负担
- 动态路由的获取 —— 路由选择算法
  - 距离矢量路由选择 (DV)

        - RIP：路由信息协议
        - 过程：计算最优路径、装载最优路径到路由表、引导IP分组转发
    - 链路状态路由选择 (LS)
        - OSPF：开放的最短路径优先
- 如何量度路径的优劣?
    - 路径长度
        - 由网络管理员定义每条网络链路的代价 (cost)，从源到宿的代价总和为路径长度，hop (跳数)
    - 带宽
        - 链路的最大传输能力以及网络流量
        - 在代价模型中，带宽越大代价越小，因此描述优劣时，带宽在分母上
    - 可靠性：链路数据传输的可靠性 (误码率)
    - 延迟：数据包从源到宿需要花费的传输时间
    - 负载：网络资源 (例如路由器的CPU) 使用率
    - 通信代价：占用通信线路的费用
- 最优化原理
    - 如果一个路由器 J 处在路由器 I 到路由器 K 的最优路径上，那么从路由器 J 到路由器 K 的最优路径也在同样的这条路径上。
- 汇集树 (sink tree)
    - 从所有的源到一个给定的目的的最优路径形成的一棵树，树根是目的。
    - 汇集树不必是唯一的。
    - 路由算法的目的：为所有的路由器发现和使用汇集树。
    - 

- 动态路由的优缺点
    -

优点	缺点
适用于有多个路由器的网络	可能会使实施更加复杂
通常不受网络规模的限制	有安全隐患 有通告在路由器之间不断穿梭
自动更新和维护路由信息	需要CPU、RAM、带宽等开销来维护

5.10.5 课堂总结

- 什么是静态路由?
- 静态路由有什么优点、缺点?
- 默认的静态路由有什么用处?
- 什么是动态路由?
- 为什么需要动态路由? (优点)
- 按照原理划分，动态路由分为几类?
- 路由算法的目的是什么? (为所有路由器发现和使用汇集树)
- 路径的常用量度有哪些?

5.11 距离矢量路由选择协议

5.11.1 DV概述

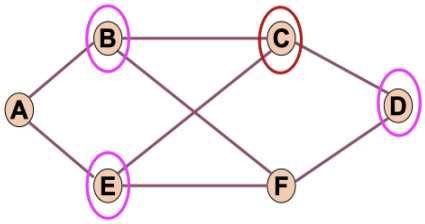
- 距离矢量路由选择 (DV - Distance Vector)
- D-V是分布式 Bellman-Ford 路由算法
  - 常被用于小型网络
  - RIP是一个典型的DV算法，在早期互联网中被广泛使用

5.11.2 DV的具体过程

- 工作原理：
  - 每个路由器 (节点) 维护两个向量，D_i 和 S_i，分别表示从该路由器到所有其它路由器的距离以及相应的下一跳 (next hop)
  - 在邻居路由器之间交换路由信息 (矢量)
  - 每个路由器 (节点) 根据收到的矢量信息，更新自己的路由表。
  - 核心三要素 —— 维护、交换、更新
- 维护两个向量
  - d_{i1}：从节点 i 到节点 1 的度量 (代价)
  - S_{i1}：沿着从节点 i 到节点 1 的最优路径上的下一跳
  - n：网络中的节点数，节点从1、2、...、n

- $$D_i = \begin{bmatrix} d_{i1} \\ d_{i2} \\ d_{i3} \\ \cdots \\ d_{in} \end{bmatrix} \qquad S_i = \begin{bmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \\ \cdots \\ s_{in} \end{bmatrix}$$

- 更新路由表
  - 当邻居间交换了矢量信息之后：
    - 更新距离：$d_{ij} = \mathrm{Min}[d_{ix} + d_{xj}], x \in A$
      - A：节点 i 的邻居集合
      - d_{ij}：从节点 i 到节点 j 的最短距离
      - d_{ix}：从节点 i 到节点 x 的最短距离
      - d_{xj}：从节点 x 到节点 j 的最短距离
    - 更新下一跳：$S_{ij} = x$
- D-V算法特点
  - 优点：简单
  - 缺点
    - 交换的信息太大了
    - 路由传播慢，可能导致路径信息不一致
    - 收敛 (寻找最优路径的过程) 慢，度量计数到无穷
    - 不适合大型的网络
- 例题
  - 一个网络拓扑如下，某个时刻，路由器C接收到三个邻居发过来的矢量如下：

	A	B	C	D	E	F
From router B：	（5，	0，	8，	12，	6，	2）
From router D:	（16，	12，	6，	0，	9，	10）
From router E:	（7，	6，	3，	9，	0，	4）

    - 现在，路由器C到 B、D 和 E 的代价分别是 6、3 和 5，试回答路由器C更新后的路由表

分析：通过路由器 B、D 和 E进行转发的话,路由器C的矢量分别是：

Via router B：（11，6，14，18，12，8）

Via router D:（19，15，9，3，12，13）

- Via router E:（12，11，8，14，5，9）

所以，路由器C更新后的路由表如下：

（11，6，0，3，5，8）

（B，B，-，D，E，B）

5.11.3 课堂总结

- 什么是距离矢量路由选择协议?
- DV的工作原理是怎样的?
- 运行DV的好处是什么?

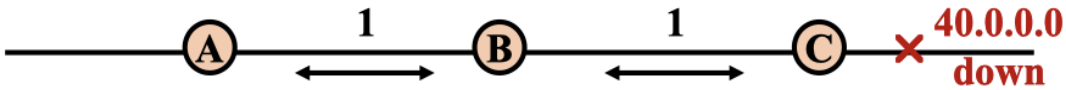
5.12 RIP

5.12.1 RIP概述

- RIP定义
  - 路由选择信息协议 (Routing information protocol)
    - 1988年，RFC1058
    - RIPv1：有类的路由选择协议
    - RIPv2：无类的路由选择协议，支持CIDR、VLSM等
    - RIPng：支持IPv6
  - 在现代网络中已经极少使用RIP协议。
- 主要特点
  - RIP是一种典型的D-V路由选择协议
  - RIP采用了跳数 (hop) 作为量度
  - 当量度超过 15 跳时，目的被认为不可达
  - 默认地，每 30s 交换一次矢量/向量信息 (全部路由表)

5.12.2 RIP主要缺陷

- 不能到达量度超过15跳的目标网络

- RIP的度量 (代价) 是跳数，即沿途经过的路由器的个数，有时候并不合理，不能真正反映网络的状况
- 无穷计数
  - 好消息跑得快，坏消息跑得慢
    - 当网络中有一条比较短的路径时，这个信息会被快速传播
    - 但是当网络中有一个节点出现故障时，这个信息会缓慢的在周围进行传播，因为其他路由器只能该节点距离越来越大，但无法判断哪里出了故障。
- 路由环
  - 错误的消息在C与B之间不断复制和修改，形成路径传播的环路。
  - 

到达信宿40.0.0.0的路由变化

时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C，1+1=2
第 1 步	2	3	2	C→B，2+1=3
第 2 步	4	3	4	B→C，3+1=4 B→A，3+1=4
…				

  - 解决方法：定义路径度量 (代价) 的最大值
    - 当代价超过最大值即可认为该路由器不可达
    - 其余方法：水平分割、毒性逆转、抑制定时器、触发更新
- 收敛慢
- 路由面临的复杂情况
  - 站得高才能看得远，确定全局最佳路径，但是站的高需要付出代价 (带宽、延迟)
  - 途径线路、站点以及目的网络都是动态变化的，最佳路径也要随之发生变化，需要及时获取状态变化信息。
  - 在站的不够高、跟的不够紧的情况下，只能直接获取近邻信息，远处信息通过逐站信息传播而间接获取，有可能传播、学习到错误的、过时的信息。
  - 最坏情况，全网传播和学习过时的信息，永远无法达到稳定状态：算法不收敛。
- 例题

1 单选 (2分) 某自治系统采用 RIP 协议，若该自治系统内的路由器 R1 收到其邻居路由器 R2 的距离矢量中包含的信 息<net1，16>，则可能得出的结论是？ 得分/总分

- A. R1 不能通过R2到达net1
- B. R2 可以到达 net1，代价是 16
- C. R1 可以通过R2到达 net1，代价是17 ✕0.00/2.00
- D. R2 可以通过R1到达net1 ,代价是17

正确答案：A 你错选为C

解析： A、对于RIP来说，一条代价超过了15跳的路径都被认为不可达。

5.12.3 课堂总结

- DV、RIP的主要问题是计算到无穷，产生路由环
- 问题
  - 什么是RIP?
  - RIP的优点是什么?
  - RIP的缺点是什么?
  - RIP的基本工作原理是怎样的?
  - 如何配置RIP?
  - 定义一个路径代价的最大数产生了什么后果?
  - RIP会彻底消失吗?

第五章 网络层 (下)

5.13 链路状态路由选择

5.13.1 链路状态路由概述

- 定义
  - 链路状态路由 (LS - Link State)
- 主要思想
  - 发现：发现它的邻居节点们，了解它们的网络地址
  - 设置：设置到它的每个邻居的成本度量
  - 构造：构造一个分组LSP，包含它所了解到的上述所有信息
  - 发送：将这个分组发送给所有其他的路由器
  - 计算：基于完整的网络图，计算到每个目标的最短路径

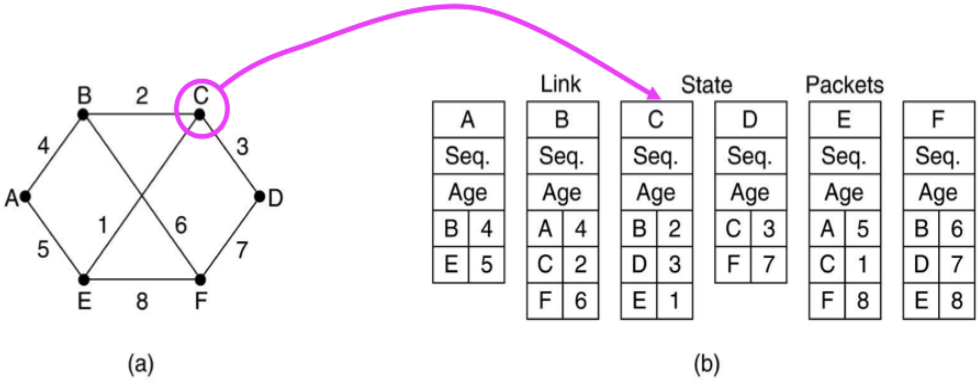
5.13.2 发现邻居节点

- 当一个路由器启动的时候，在每个点到点的线路上发送一个特别的HELLO分组
- 收到HELLO分组的路由器应该回送一个应答，应答中有它自己的名字 (采用一个全球唯一的名字 globally unique name)
- 这样路由器就可以发现它的所有邻居节点了。

5.13.3 设置链路成本

- 链路成本 (开销 / 量度 / 代价)
  - 通常为自动发现设置或人工配置
    - 常见量度：链路带宽 (反比)
- 延迟也可作为量度
  - 路由器发送一个特别的ECHO分组，另一端立刻回送一个应答
  - 通过测量往返时间 (RTT)，发送方路由器可以获得一个合理的延迟估计值
    - 为了得到更准确的结果，可多次测量，取平均值

5.13.4 链路状态分组/公告 (LSP)

- 链路状态分组/公告 (LSP - Link State Packet / Advertisement)
- LSP组成
  - 发送方的标识 (ID of the sender)
  - 序列号 (sequence number)
  - 年龄 (age)
  - 邻居列表 (list of neighbors)
  - 到邻居的成本 / 量度 (delay to each neighbor)
- 构造LSP的时间
  - 周期性地构造和发送
  - 如果有特别的事件发生，也会构造LSP。(某条线路或邻居down了、某个接口上来了等)
- 举例
  - 

(a) A subnet. (b) The link state packets for this subnet.

- 4 单选 (2分) 下列哪个答案最能说明由节点A泛洪发送的链路状态数据包（LSA/LSP）所代表的信息。 得分/总分

  A. 所有从A到其相邻路由器的链路成本。

  B. 所有网络中的路由器的标识以及估计的到达这些路由器的最小开销 ✕0.00/2.00

  C. 所有与A直接相连的邻居路由的标识以及到他们之间的链路成本

  D. 所有与A相连的邻居路由的标识。

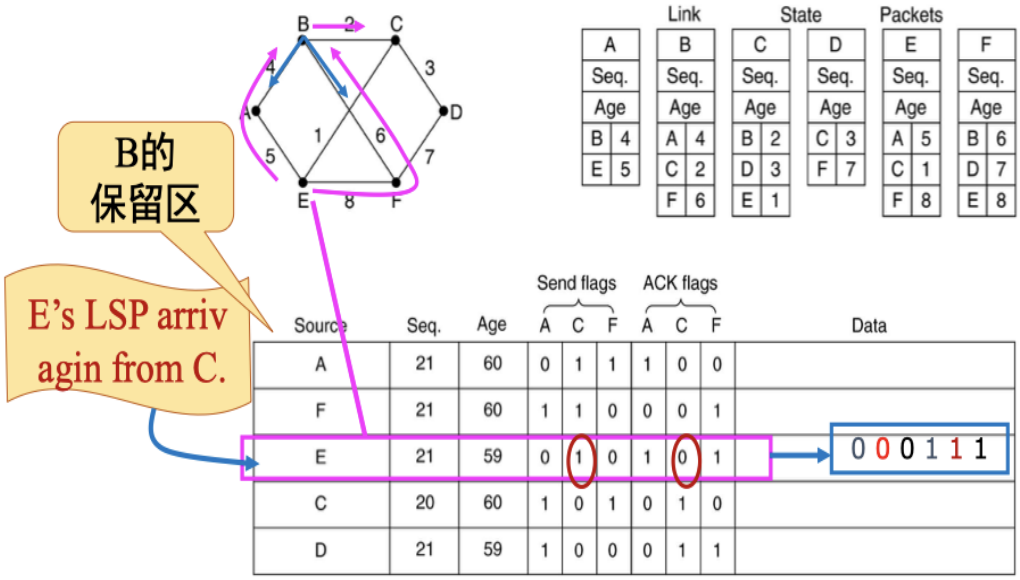
  正确答案：C 你错选为B

  解析： C、LSA中包括的信息主要就是周边的邻居及到邻居的成本，即小地图信息。

5.13.5 发送LSP

- 算法过程
  - 每个分组都包含一个序列号，序列号随着新分组产生而递增
  - 路由器记录下它看见的所有 (源路由器，序列号) 对
  - 当一个新的分组到达时，路由器根据它的记录进行判断：
    - 如果该分组是新的，就被从除了来线路外的所有其他线路转发出去 (flooding，泛洪)
    - 如果是重复分组，即被丢弃 (喜新厌旧)
    - 如果该分组的序列号比对应的源路由器发送的到过此地的分组的最大序列号还小，则该分组被当作过时的信息而被拒绝。(拒绝过时分组)
- 算法遇到的问题
  - 序列号回转，引起新老分组识别混淆 (如 000 和 111 哪个是新，哪个是旧)
    - 解决方法：使用 32 位的序列号，即使每秒产生一个分组，也需要 137 年才发生号码回转。
  - 如果一台路由器崩溃，那么它将丢失自己的序列号记录，如果他再从 0 开始，新分组会被当做旧分组而被拒绝。
    - 解决方法：age
  - 如果一个序列号被破坏了，比如发送方的序列号是4，但是由于产生了1位错误，序列号被看作65540。那么，序列号为5-65540的分组都被当作过时分组而被拒绝。
    - 解决办法：age
- age
  - 解决上述的路由器崩溃和序列号损坏的方法：
    - 每个分组序列号之后是年龄 (age)，并且每秒钟年龄减 1。
    - 当年龄为0时，来自该路由器的信息被丢弃
    - 通常，每隔一段时间，如10秒钟，一个新分组就会到来。所以，只有路由器down机才可能导致超时 (或者，连续6个间隔因为丢失，没有收到新的分组)。
- 其他改进方法使算法更健壮
  - 保留区

        - 当一个链路状态路由分组到达某个路由器时，它首先被放到一个保留区中等待一段时间。
        - 如果来自相同路由器的另一个分组到达了，这两个分组的序列号会被比较：
            - 如果相等，是重复分组，丢弃
            - 如果不相等，旧的那个被丢弃
    - 为了防止路由器到路由器的线路发生错误，所有的链路状态分组都要被确认
    - 当一条线路空闲的时候，路由器扫描保留区，以便选择一个分组或确认，并将其发送出去



        - 其中E此时的状态为通过A、F到达，要转发到C。发送标记为010，确认标记位101，向A、F发送确认，向C转发LSP。

5.13.6 计算新的路由路径

- 一旦一个路由器获得了全部的链路状态分组就可以构造出全网络图来了
- 可以使用最短路径算法 (Dijkstra) 来计算路由器之间的最短路径
    - 计算结果是一棵树 (汇集树)，会形成相应的路由，安装在路由表中，引导数据分组的转发。

5.13.7 课程总结

- LS路由算法的特点
    - 优点
        - 每个路由器的认识一致
        - 收敛快
        - 适合在大型网络里使用
    - 缺点
        - 每个路由器需要较大的存储空间
        - 计算负担很大
- LS路由算法基本原理

    - 发现邻居、设置成本、构造LSP、分发LSP、计算
- 问题
    - 相比DV，LS具有哪些优点
    - 相比DV，LS具有哪些缺点
    - LS的基本工作原理是怎样的？哪一步最关键？(分发LSP)
- 例题
    - 8 多选 (2分) 动态路由路由选择协议的两个主要任务是什么？(选二个) 得分/总分

      A. 发现主机 ✕该题无法得分/2.00

      B. 更新和维护路由表 ✓该题无法得分/2.00

      C. 分配IP地址

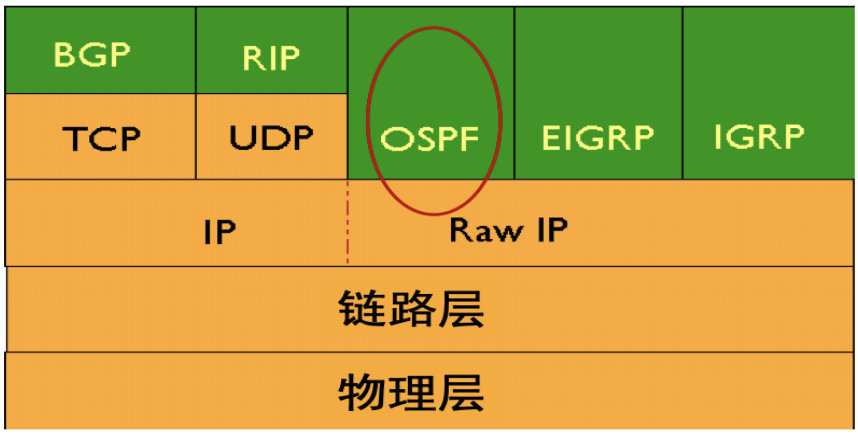
      D. 网络发现

      正确答案：B、D 你错选为A、B

      解析： B、不言而喻，路由器转发依据是路由表 D、网络发现，路由器学习远程网络，转载到路由表中

5.14 单区域OSPF

5.14.1 OSPF概述

- 定义
    - 开放的最短路径优先 (OSPF - Open Shortest Path First)
- OSPF在OSI参考模型中的地位
    - 

- 特点
    - 使用图来表述真实网络
        - 每个路由器/LAN都是一个节点
        - 测量代价/量度 (metric)

- OSPF是一种基于开放标准的链路状态路由协议，是目前IGP中应用最广、性能最优的一个协议。
  - OSPF可以在大型网络中使用，并且无路由自环。
  - OSPF支持VLSM、CIDR等。
  - 使用带宽作为量度值 ($\frac{10^8}{BW}$)
  - 收敛速度快
  - 通过分区实现高效的网络管理
- OSPF必须划分区域
  - 存在一个Area0 (区域0)，作为骨干区域
  - 所有其它区域都必须连接到区域0上。

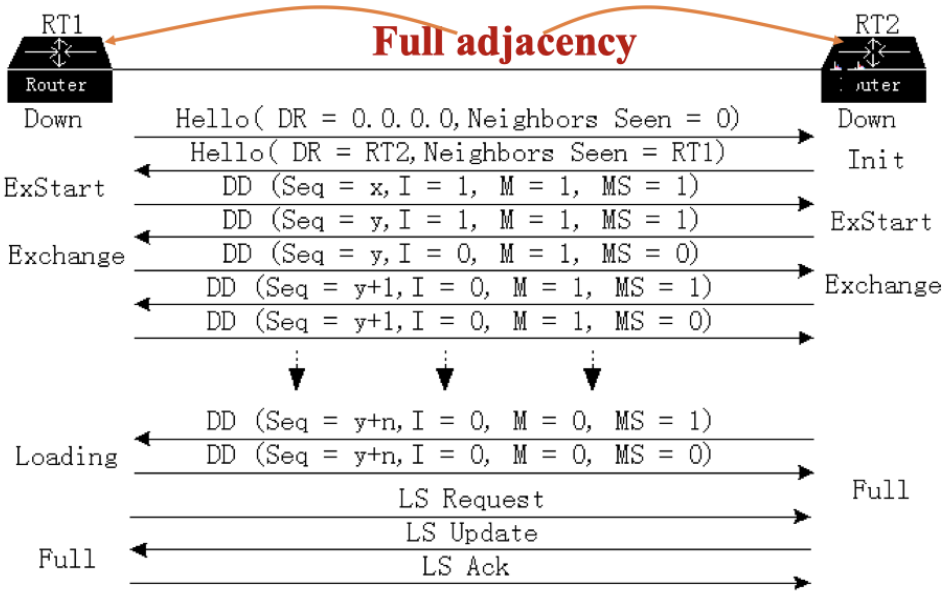
- 重要术语
  - RouterID：一个32位的无符号整数，是一台路由器的唯一标识，在整个自治系统内唯一。
  - 协议号：IP头中代表OSPF报文的协议号是89
    - | IP Header
（Protocol # 89） | OSPF Packet |
  - TTL = 1：通常OSPF报文不转发，只被传递一条，即在IP报头的TTL值被设为 1，但虚连接除外。
  - 代价
    - 使用带宽作为代价的量度，计算时用 $\text{代价} = \frac{10^8}{\text{带宽}}$
    - 例如，10M 链路的代价为 $\frac{10^8}{10*10^6} = 10$

5.14.2 OSPF分组类型

- OSPF 五大分组 (packet) 类型
  -

OSPF数据包类型	描述
Type1 — Hello	与邻居建立和维护毗邻 (keep alive) 关系
Type2 — 数据库描述包 (DD)	描述一个OSPF路由器的链路状态数据库内容。 (即所有LSP信息的摘要信息，占总大小的30%)
Type3 — 链路状态请求 (LSR)	请求相邻路由器发送其链路状态数据库中的具体条目。 (查看摘要DD之后，再确定要交换哪些信息)
Type4 — 链路状态更新 (LSU)	向邻居路由器发送链路状态通告。 (新网络事件发生，或对于LSR的应答)
Type5 — 链路状态确认 (LSA)	确认收到了邻居路由器的LSU。 (收到LSU之后，发送应答。)

5.14.3 OSPF运行过程

- 建立路由器毗邻关系
  - 建立路由器毗邻关系的过程
    - 

  - OSPF状态
    - Down、Init (初始)、Two-way (双向)、ExStart (准启动)
    - Exchange (交换)、Loading (加载)、Full adjacency (全毗邻)
- 选举DR和BDR
  - 选举DR和BDR的原因
    - 建立毗邻关系的过程称为同步。网络自由进行同步的话，同步次数为 $\frac{n*(n-1)}{2}$ 。但是如

果指定一台路由器作为DR (Designated Router)，则同步次数降为 $n-1$ 次，即每台路由器都只与 DR 进行同步。
    - DR选举过程 (类似于村长选举)
        - 本网段内的OSPF路由器
        - 本网段内的 priority > 0 的OSPF路由器
        - 所有的 priority > 0 的OSPF路由器都认为自己是DR
        - 选priority值最大的，若priority值相等，选RouterID最大的作为DR
    - DR中的核心思想
        - 选举制：DR是路由器选出来的，而非人工指定，可以自动运行。
        - 终身制：DR一旦当选，除非路由器故障，否则不会更换。
        - 世袭制：DR选出的同时，也选出BDR，DR故障后，由BDR接替DR称为新的DR。
    - DR带来的变化
        - 同步的次数减少了 [O(n)]，减少了带宽的利用
        - 路由器的角色：DR、BDR、DROther
        - 路由器间的关系：Unknown、Neighbor、Adjacent (全毗邻关系)
- 发现路由
- 选择最佳路由
    - 每台路由器分别以自己为根节点计算最小生成树
- 维护路由信息
    - 触发更新
        - 如果发生了一些网络事件，则触发自动更新机制，发送LSU通告
    - Hello分组发送的时间间隔：缺省10s
    - Hello分组的失效间隔：缺省40s
        - 路由器40s没有收到邻居的Hello分组，那么对方down了。
    - 即使没有拓扑变化，LSA在条目过期 (缺省30分钟)后，会发送LSU，通告链路存活。

5.14.4 课程总结

- OSPF克服了路由自环
    - 每一条LSP都标记了生成者，(用生成该LSP的路由器的RouterID标记)，其他路由器只负责传输，这样不会在传输的过程中发生对该信息的改变和错误理解。
    - 路由计算的算法是SPF，计算的结果是一棵树，路由是树上的叶子节点，从根节点到叶子节点是单向不可往返路径。
    - 区域之间的环路通过规定骨干区域避免。
- 问题
    - OSPF的量度 (代价、开销) 采用了什么？一个100M的快速以太网链路，其量度是多少?
    - OSPF克服了路由环吗？为什么?
    - 怎样建立全毗邻关系?
    - 为什么要选举DR?

5.15 无类域间路由 CIDR

5.15.1 CIDR概述

- IP地址问题
  - IP分类造成了数百万个地址浪费。超过一半的B类网络拥有的主机数不超过50台。
  - 路由表膨胀
- 无类别域间路由 (CIDR - Classless InterDomain Routing)
  - 缓解了地址枯竭的趋势，控制甚至缩减了路由表的开销
  - CIDR基本思想
    - 分配IP地址时不再按类别来分，而是按照可变长的地址块来分配。
- 举例，CIDR分配
  - **一块地址从194.24.0.0开始，可用地址数为8192（2^13），即194.24.0.0/19**
    - **剑桥申请2048个地址：194.24.0.0~194.24.7.255**
    - **牛津申请4096个地址：194.24.16.0~194.24.31.255**
    - **爱丁堡申请1024个地址：194.24.8.0~194.24.11.255**

University	First address	Last address	How many	Written as
Cambridge	194.24.0.0	194.24.7.	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

  - 上述牛津申请的4096没有直接从8或者12开始，原因是子网网络号的最高位必须与子网中任意一个网络地址一致，而8和12无法做到。

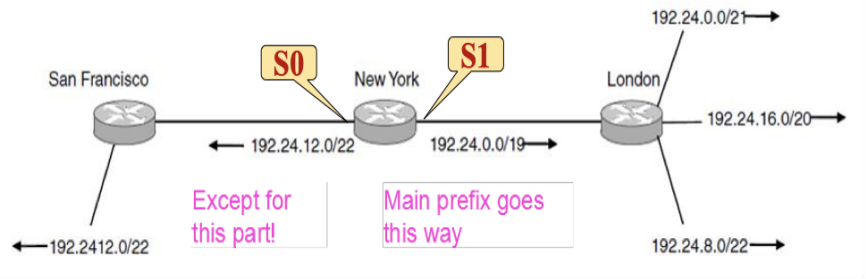
5.15.2 CIDR具体原理

- 路由表扩展
  - 增加一个32位的子网掩码
  - 路由表中维护三元组：(目的IP地址、子网掩码、输出线路)，(IP address、subnet mask、outgoing line)
- 路由表工作过程 (一个分组到来时)
  - 分组中的目的IP地址被检查
  - 目标IP和子网掩码进行与操作，获得目标网络地址，以查找路由表
  - 如果路由表中有多个表项匹配 (这些表项有不同的子网掩码)，使用子网掩码最长的那个表项
    - 子网掩码越长，意味着网络越小，最靠近目的网络。
  - 举例

目标IP 192.24.12.4:11000000. 00011000. 00001100. 00000100

路由表中有两个表项匹配

- 192.24.12.0/24（11000000. 00011000. 00001100. 00000000）
- 192.24.0.0/19 （11000000. 00011000. 00001100. 00000000）



5.15.3 路由聚合

- 路由聚合前提条件
  - 子网构成的地址空间是连续的
  - 下一跳相同
- 路由聚合方法
  - 找到IP地址中最长的不变的位数，再令该地址为该路由网络位即可。
  - 200.199.48.0/24
    200.199.49.0/24
    200.199.50.0/24
    200.199.51.0/24

    聚合

    200.199.48.0/22

    00110000
    00110001
    00110010
    00110011

    不变的位数：8+8+6=22
    （网络位数或掩码位）
- 路由聚合结果
  - 形成超网 (SuperNet)
- 功能
  - 缩减路由表规模
    - CIDR功能，只需记录子网网络号
  - 隔离路由翻动
    - 某路由器下的其他路由的Down/Up状态，不会影响到汇总后的路由状态。

5.15.4 课程总结

- CIDR的功能是什么？
- 采用了CIDR技术，怎样完成路由聚合？
- 怎样计算一个无类网络的可用/合法IP地址？

5.16 网络地址翻译 NAT

5.16.1 NAT概述

- 定义
  - NAT (net address translate)
    - 私有IP地址和公有IP地址之间的转换
  - PAT (port address translate) [超载]
    - 多个私有IP地址影射到同一个IP地址的不同端口
- 三类私有地址
  - 私有地址 (Private IP address)
    - 不可路由的地址、也不可用于广域网链路上
  -

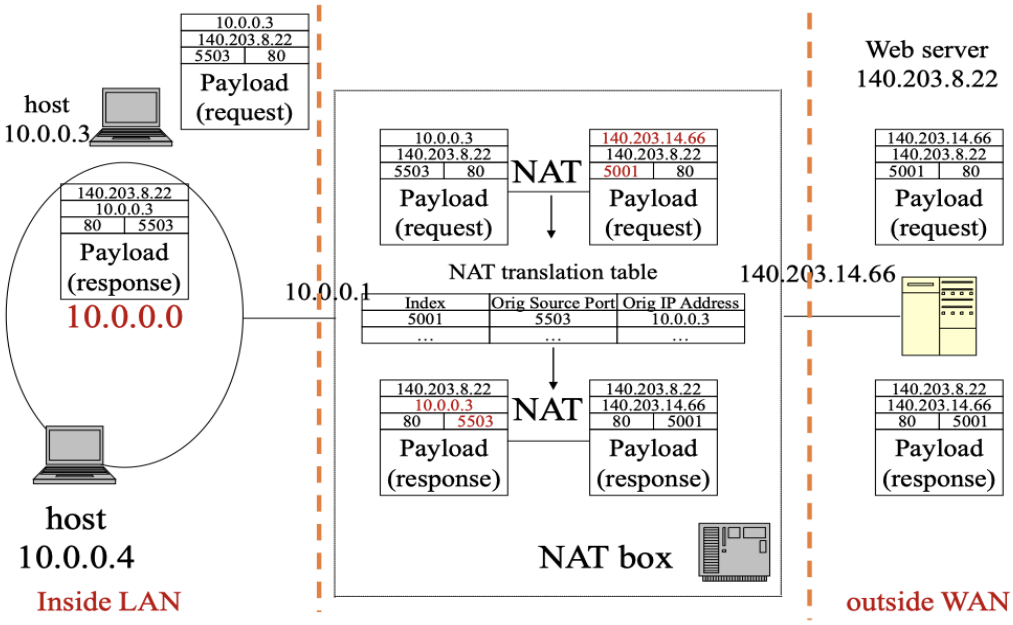
Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 – 192.168.255.255	192.168.0.0/16

不具备唯一性

- NAT转换器 (NAT Box) 的位置和功能
  - 由NAT转换器完成私人地址和公有地址的转换，并且维护一个地址转换表。
  - 当有外网的分组到达时，NAT转换器查找地址转换表，转换分组目标地址之后，转发该分组到内网。
  - NAT转换器可以是一个专用的服务器，也可以运行在内网的边界路由器上，也可以在家用路由器 (AP) 上。

5.16.2 NAT 具体工作过程

- 具体工作过程
  - 私有网络中 10.0.0.3:5503 与 外网中 140.203.8.22:80 进行通信。
  -



- 第一步
  - 私有网络 10.0.0.3 发出一个分组。分组中信息从上到下分别是源IP地址、目的IP地址、源端口 (5503)、目的端口 (80)。
- 第二步
  - NAT收到这个分组之后，将源IP地址转换为共有IP地址，即140.203.14.66，将源端口转换为5001。
  - 并将这个转换信息记录在NAT转换表中，其中Index为转换后的端口。
  - 将这个分组发向外网
- 第三步
  - 外部工作站收到分组之后，将信息封装重新发回原网络中，其中源IP地址、目的IP地址与源端口、目的端口互换。
  - NAT收到这个分组之后，根据NAT转换表中的信息，将目的IP地址和目的端口进行转换，重新发往内网中。到此即完成通信。

- NAT带来的问题
  - 违背了IP的结构模型
    - 每个IP地址唯一地标识了一台机器
  - 将互联网从"无连接"变成"面向连接"的网络
    - NAT转换器维护着连接的状态，一旦它崩溃，连接就没有了。
  - 违背了最基本的协议分层原则
    - 不但置换了IP地址，还置换了端口号，跨层工作。
  - 如果传输层不是采用TCP或UDP，而是采用了其它的协议，NAT将不再工作
  - 有些应用会在payload中插入IP地址，然后接收方会提取出该IP地址并使用，但是NAT转换器对此一无所知，导致该类应用不再有效
  - NAT让一个IP地址可以承载 61440 (65536-4096) 个私人地址 (超载，PAT)
- 无NAT服务的网络中使用私有地址
  - 无NAT却使用私有地址，因此仅支持私网内部通信
  - 私网不能与其它外网和私网通信，可以避免被外网攻击，有一定安全性。
- 例题

- 3 单选 (2分) 下列关于NAT（网络地址翻译转换）的语句中，哪个是错误的? 得分/总分

  A. 可能单一的内部 IP, Port对，有多个 NAT 表项。

  B. 可能单一的内部 IP地址，有多个 NAT 表项。

  C. 不同的 NAT 表项可以包含不同的内部 IP 地址。

  D. 不同的 NAT 表项可以包含不同的外部 IP 地址。 ✕0.00/2.00

  正确答案：A 你错选为D

  解析： A、错误的，一个单一的IP/port对，只能有一个表项，如果有多个表项，这个源自内网的包送给谁呢？ B、正确，一个私人地址所在的主机可以开启多个进程，都经过NAT转换器转换 D、正确，可以有多个全球唯一的IP地址来承载内部网络

5.16.3 课程总结

- NAT优点
  - 节省了共有IP地址
  - 提供了内部私网访问外网的灵活性
  - 有一定的保密性。
- NAT缺点
  - 影响了部分协议和应用的通信
  - 增加了网络延时
  - NAT转换设备可能成为网络的瓶颈
  - 影响了路由追踪工具的使用
- 问题
  - 为什么需要 NAT/PAT?
  - 私人地址空间有哪些?
  - NAT转换器完成什么功能?
  - NAT有什么优点?
  - NAT有什么缺点?

5.17 互联网控制协议 ICMP

5.17.1 ICMP 概述

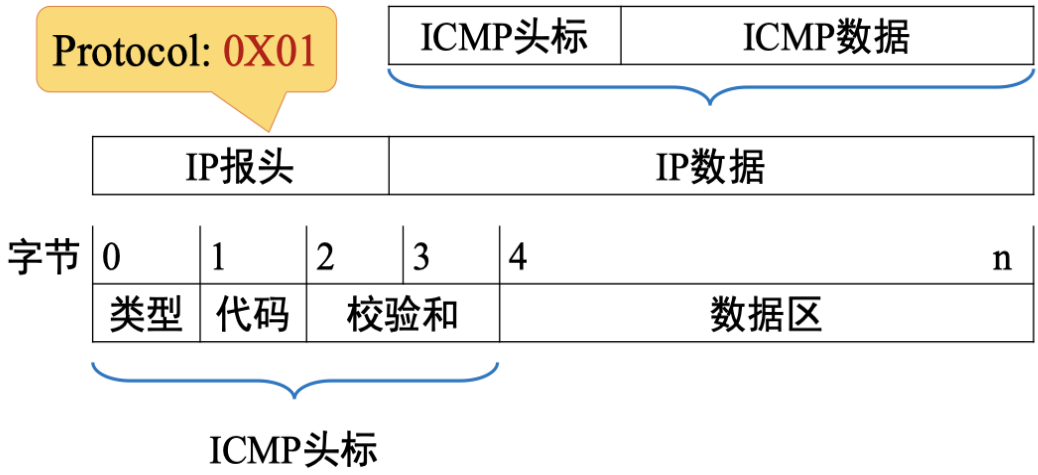
- 互联网控制消息协议 (ICMP - Internet Control Message Protocol)
  - 作用：用来报告意外的事件或测试互联网
- ICMP类型

-

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

TTL=0

- ICMP消息格式
  -



  - IP分组用户协议中为 1 表示是ICMP消息。
  - ICMP消息将IP分组数据部分分为 ICMP头标+ICMP数据
    - ICMP头标中包含三部分：类型、代码、校验和
    - 类型和代码用来表示是哪一种ICMP消息
    -

```
    Destination: 192.168.1.1
⊟ Internet Control Message P
    Type: 3 (Destination unr
    Code: 10 (Host administr
    Checksum: 0xe342 [correc
```

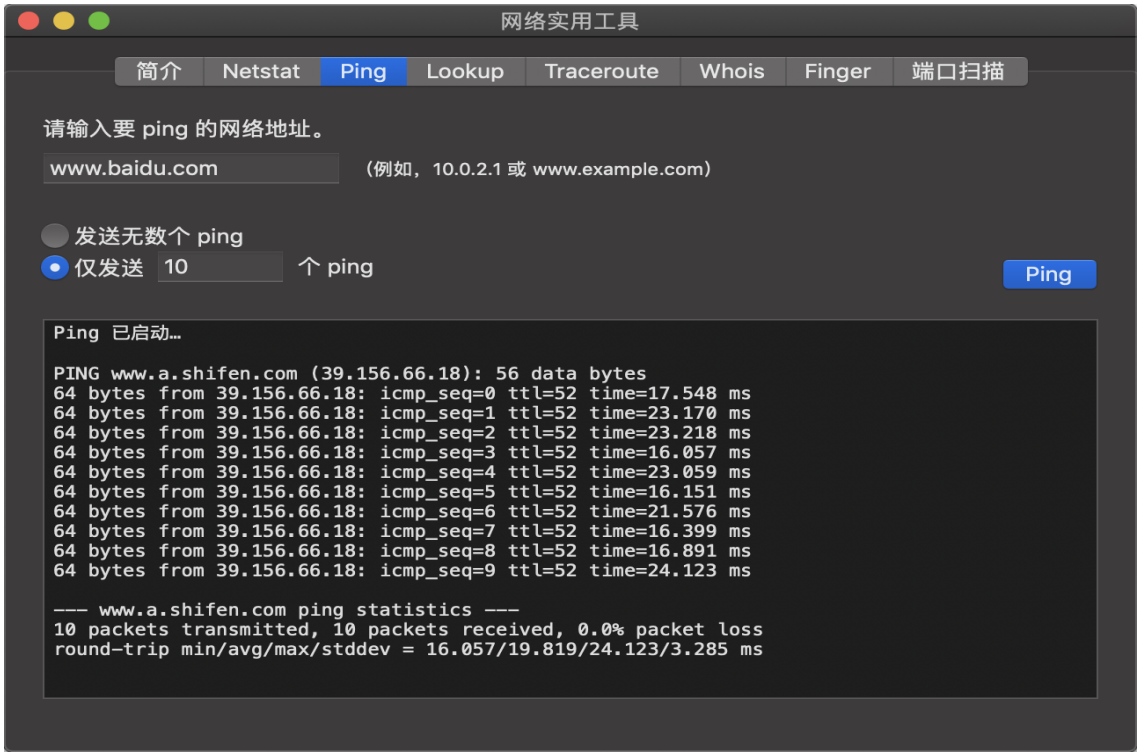
主机不可达

```
    Protocol: ICMP (1)
⊞   Header checksum: 0x4c1f [correct]
    Source: 113.67.24.203 (113.67.24.203)
    Destination: 192.168.1.101 (192.168.1.101)
⊟ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x7795 [correct]
⊟   Internet Protocol Version 4, Src: 192.168.1.
```

端口不可达

  - ICMP三大应用
    - Ping
    - Tracert
    - PMTU

5.17.2 Ping

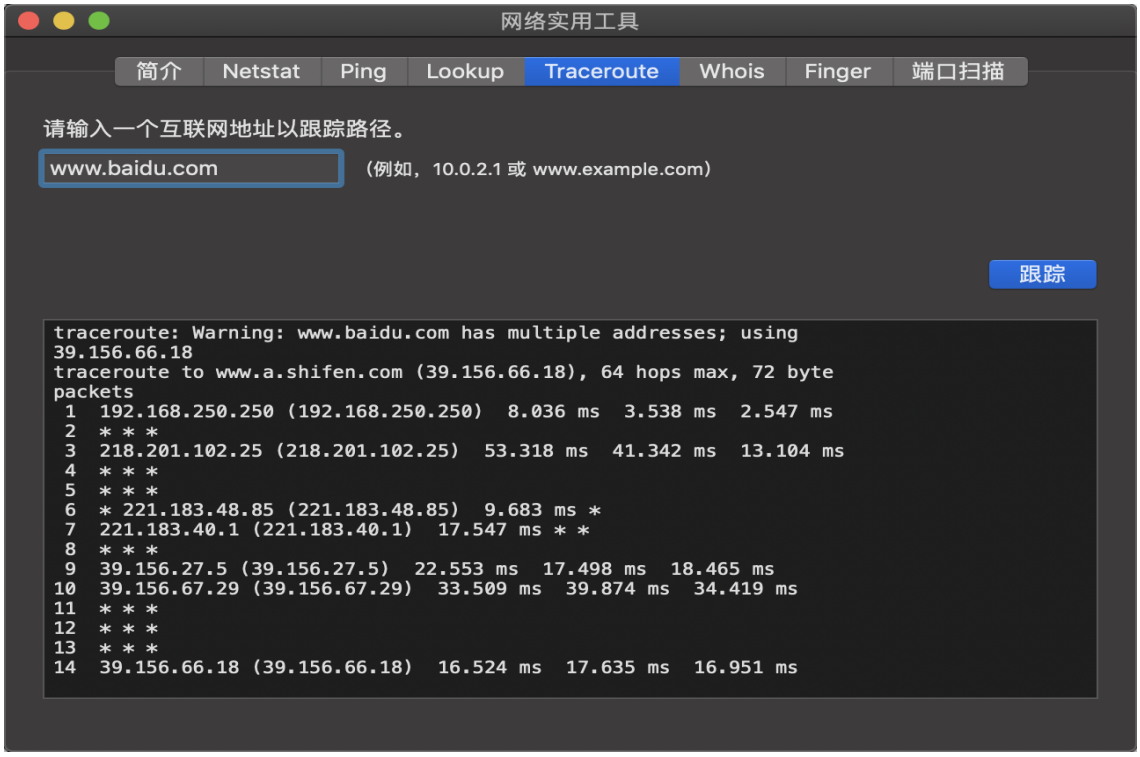
- 具体过程
  - 使用ping命令时，将向目的站点发送一个ICMP回声请求报文 (包括一些任选的数据)
  - 如目的站点接收到该报文，必须向源站点发回一个ICMP回声应答报文，源站点收到应答报文 (且其中的任选数据与所发送的相同)，则认为目的站点是可达的，否则为不可达。
- Ping作用
  - 检测TCP/IP是否正常工作：ping 127.0.0.1
  - 网络设备是否正确：ping 本机IP地址
  - 检查对外连接的路由器：ping 默认网关IP
  - 检查与某台设备的畅通情况：ping IP
  - 检查DNS设置：ping www.baidu.com
  - 执行DNS反向查询：ping -a IP地址
- 举例
  - 

5.17.3 Tracert

- 具体过程
  - Tracert过程是通过ICMP数据报超时报文来得到一张途径的路由器列表
  - 源主机向目的主机发一个IP报文，并置 TTL = 1，到达第一个路由器时，TTL-1=0，则该路由器回发一个ICMP数据报超时报文，源主机取出路由器的IP地址即为途径的第一个路由端口地址。
  - 接着源主机再向目的主机发第二个IP报文，并置 TTL = 2，然后再发第三个、第四个IP数据报，...直至达到目的主机。
  - 但互联网的运行环境状态是动态的，每次路径的选择有可能不一致，所以，只有在相对较稳定 (相对变化缓慢) 的网络中，Tracert才有意义。

- Tracert 最大为30跳，超过30跳就会终止这个过程。

- 举例

  - 

5.17.4 PMTU

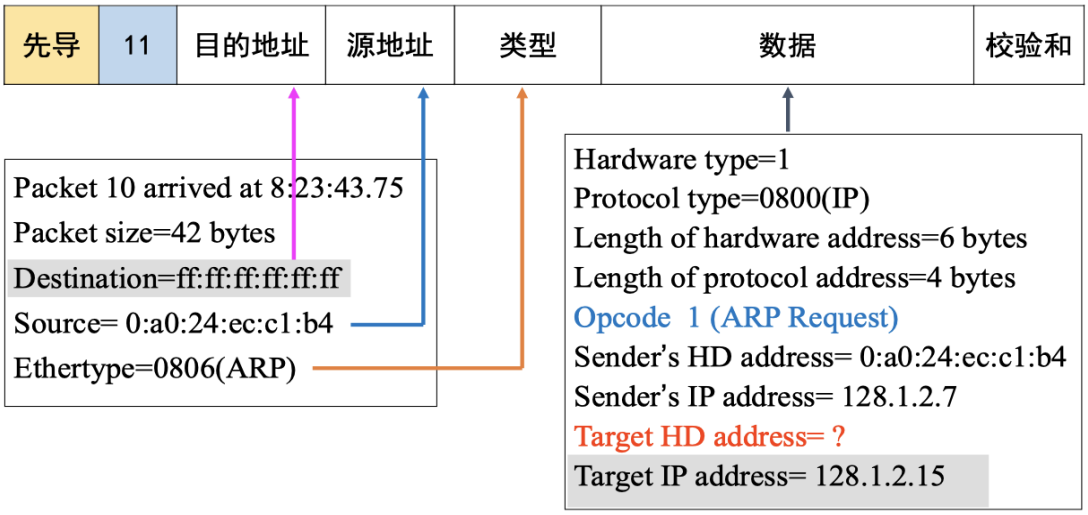
- 路径MTU：指能经过路径而无需分段的最大数据包
  - 通过在发出的数据包中设置DF位，发送方知道这个数据包要要么完整地到达目的地，要么有个报错信息反馈回来。
  - 标志位中 DF = 1，表示数据包不能分段。
- PMTU：动态发现互联网上任意一条路径的最大传输单元的技术
  - 发数据包，分段标记DF=1，尝试1400、1200、900，直到到达目的机。

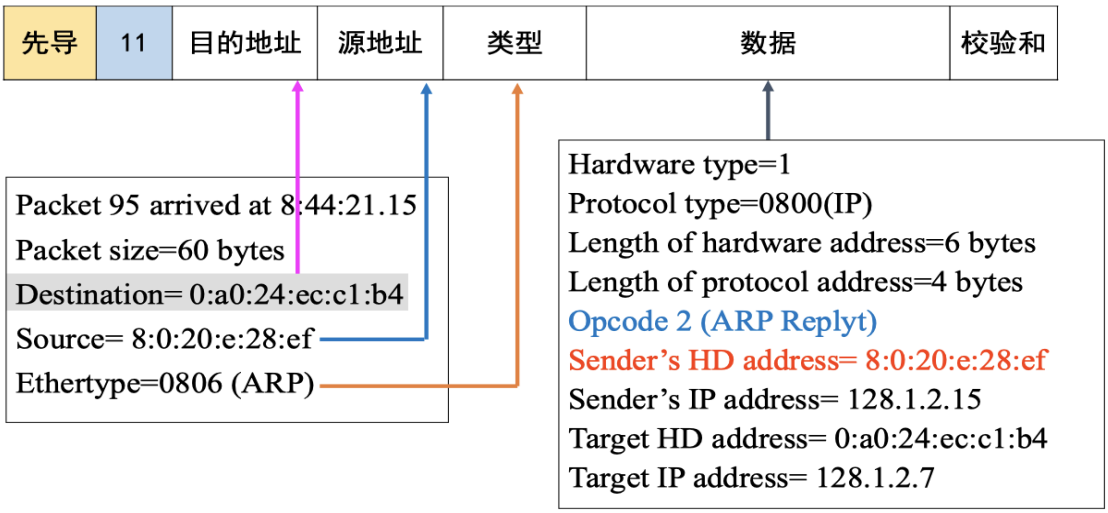
5.17.5 课程总结

- 注意
  - 一般来说，ICMP消息仅送给源机。
  - ICMP数据传输方式和其他数据传输方式一样，也可能遇到同样的错误，规定：ICMP消息不生成自己的差错报告。(否则消息可能会大量激增)
- ICMP可用来报告网络事件和测试网络
- ICMP消息封装在IP分组中
- 问题
  - 为什么需要ICMP?
  - ICMP消息的封装格式是怎样的?
  - ICMP应用ping是利用的什么类型的消息?
  - ICMP应用Tracert是利用的什么类型的消息?

5.18 地址解析协议 ARP

5.18.1 ARP概述

- 地址解析协议 (ARP - Address Resolution Protocol)
  - 目的：找到一个给定IP地址所对应的MAC地址
  - 是网络层上的协议（涉及到了IP），用广播帧来实现。
- ARP请求 (Request)
  - 

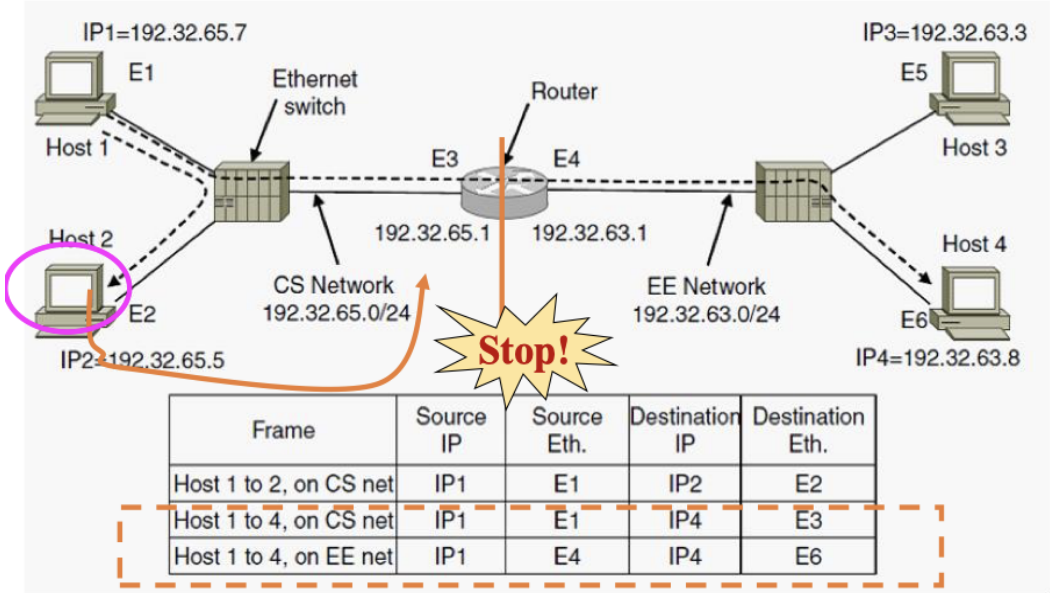
- ARP回答 (Reply)
  - 

5.18.2 ARP优化

- ARP优化措施
  - 缓存ARP结果
  - 在ARP请求中包括源机的 IP-to-MAC 地址的映射
  - 每台机器在启动的时候，广播它的 IP-MAC 地址对

- 免费ARP
  - 当一台主机启动时，会发送一个免费ARP (如果意外收到一个应答，则IP地址发生冲突)
  - 当一个接口的配置发生了改变，会发送一个免费ARP
  - 免费ARP帧中，目的IP地址 = 源IP地址

5.18.3 远程主机MAC地址解析

- 缺省网关 (默认网关、代理ARP)
  - 当源设备需要目的地址与自己不在同一个网络时，如果源机不知道目的MAC地址，它必须使用路由器的服务使它的数据达到目的，当路由器在这种方式下使用时，称为缺省网关。
  - 缺省网关是与源设备所处的网段相连的路由器接口上的IP地址。
- 工作过程
  - E1要与E6通信，需要发送ARP请求获取E6 MAC地址
  - 

Frame	Source IP	Source Eth.	Destination IP	Destination Eth.
Host 1 to 2, on CS net	IP1	E1	IP2	E2
Host 1 to 4, on CS net	IP1	E1	IP4	E3
Host 1 to 4, on EE net	IP1	E4	IP4	E6

  - ① E1 发现 E6 不在同一个LAN中
  - ② E1 将目的MAC地址改为默认网关MAC地址发向默认网关。
    - 默认网关：和主机直接相连的路由器，承接所有发向外网的数据，数值 = 直接相连路由器的那个端口上的IP地址。
    - 如果主机不知道默认网关的MAC地址，会先发送一个ARP请求获得默认网关MAC地址。
  - ③ 默认网关拿到帧之后，用arp请求获取E6的MAC地址
    - 将源MAC地址改为默认网关MAC地址，并发送arp请求获取目的机器的MAC地址
  - 全过程中，源IP地址和目的IP地址都不会发生变化。但是源MAC地址和目的MAC地址在重新封装时都会被置换。

5.18.4 ARP表

- ARP Table
  - IP地址到MAC地址的映射表。
  - 为了减少ARP请求的次数，每个设备拥有自己的ARP表，包括路由器。

    - 储存在存储器 (RAM) 中，自动维护。(掉电消失)
- 自动维护ARP表
    - 通过广播ARP请求中的源设备信息添加更新表
    - 利用自己的ARP请求的应答信息来添加、更新表
    - 删除超过一定时限的信息

5.18.5 ARP欺骗

- ARP欺骗
    -

- 如何避免ARP欺骗
    - 静态ARP，取消动态更新维护
    - 不马上写ARP缓存
    - 设置ARP服务器
    - 硬件屏蔽 (如路由器采用静态ARP且作全权代理)

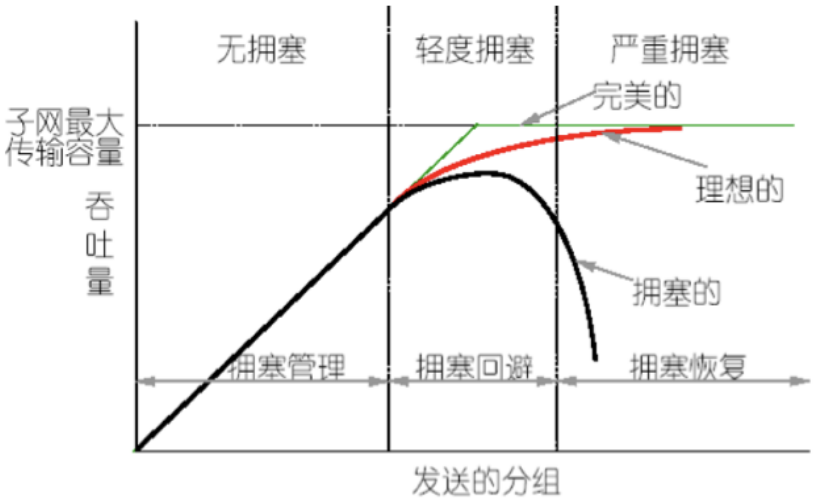
5.18.6 课程总结

- ARP工作原理是怎样的?
- 远程主机的MAC地址如何解析?
- 如何维护ARP表?
- 什么是默认网关/缺省网关?
- ARP病毒是怎么产生的?

5.19 拥塞控制

5.19.1 拥塞概述

- 拥塞：当一个子网或子网的一部分出现太多分组时，网络的性能急剧下降，这就是拥塞 (Congestion)

- 

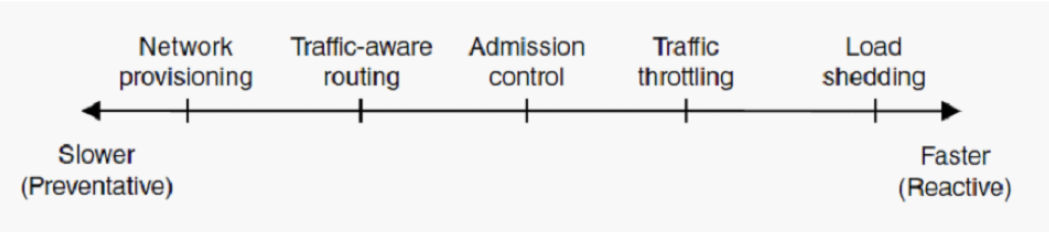
- 导致的拥塞的因素
  - 本质 —— 负载 > 资源
    - 输入流量大于输出线路的容量
    - 慢速的处理器也可能引起拥塞，如线路容量充足，但处理器来不及处理
    - 线路容量和处理器能力需要平衡

5.19.2 拥塞控制

- 拥塞控制 VS 流量控制
  - 拥塞控制
    - 确保子网能够承载所到达的流量
    - 这是个全局的问题，涉及到主机、路由器，存储转发的过程等方方面面的问题。
  - 流控
    - 只与特定的发送方和接收方之间的点到点流量有关
    - 确保一个快速的发送方不会持续地以超过接收方接受能力的速率传输数据
- 拥塞控制的两个原则
  - 开环 (Open Loop)
    - 试图用良好的设计来解决问题，本质是从一开始就保证问题不会发生。
    - 开环决策指定不考虑网络的当前状态
    - 缺点：网络情况更新太快，难以确定
  - 闭环 (Closed Loop)
    - 建立在反馈环路的概念上，分三个步骤
      - 监视系统，检测何时何地发生了拥塞
      - 把这些信息传递到能够采取行动的地方
      - 调整系统的运行，以改正问题
- 如何知道发生了拥塞 (拥塞量度) —— 数值越大则拥塞程度越重
  - 因为缺乏缓存空间而丢弃的分组百分比
  - 平均队列长度
  - 超时和重传的分组数
  - 平均分组延迟

    - 分组延迟的标准方差
- 拥塞信息传播
    - 检测到拥塞的路由器发送一个警告分组给流量源
    - 每个分组保留一位或一个域，当拥塞量度超过阈值的时候，路由器填充该位或域，以此警告它的邻居。(警告位)
    - 主机或路由器周期性地向外发送探寻分组，显式地询问有关拥塞的情况，然后，在有问题的区域利用回收的信息来路由流量
- 如何解决拥塞问题 (负载 > 资源)
    - 增加资源
        - 在某些点之间使用更多的通道增加带宽
        - 把流量分散到多条路径
        - 启用空闲或备份的路由器
    - 降低负载
        - 拒绝为某些用户提供服务
        - 给某些用户的服务降低等级
        - 让用户更有预见性地安排他们的需求

5.19.3 开环系统中的拥塞控制

- 开环系统
    - 一开始就将拥塞发生的可能性降到最低，而不是随它发生了之后再去反应
    - 从左到右，方法见效速度越来越快。
    - 网络供给、流量感知路由、准入控制、流量调整、负载脱落
    - 

- 数据报子网中流量限制
    - 每台路由器可以监视它的输出线路和其它资源的使用情况
    - 每条线路和一个实变量 u 关联在一起，其值位于 (0.0-1.0) 之间
    - 无论何时，只要 u 超过了阈值，对应的线路就进入到警告 "warning" 状态
    - 每个新到达的分组都将被检查，看它的输出线路是否处于 "警告状态"
- 处于警告状态后的措施
    - 警告位、抑制分组、逐跳抑制分组、负载脱落

5.19.4 警告位

- 分组头部中的一个特殊的位
- 该位被复制到下一个确认分组中，被传回源
- 源机会监视设置了警告位的分组的比例，相应地调整它的发送速度。

5.19.5 抑制分组

- 进入警告状态之后，路由器给源机发回一个抑制分组，并在抑制分组中指明原分组的目标地址
- 当源机收到抑制分组后，它会以某种百分比减少它发向该目标的流量
- 在一段固定时间内，源机会忽略同样一个目标的抑制分组
- 过了这段时间，会继续侦听有无新的抑制分组，如果没有，源机将增加流量。

5.19.6 逐跳抑制分组

- 逐跳抑制分组 (Hop-by-Hop Choke Packets)
  - 当网络速度很高或者路由器距源主机的距离很远的时候，给源主机发送抑制分组并不能很好地起作用，因为反应太慢。
- 改进方法
  - 让抑制分组路径的每个路由器都采取相应的措施
- 逐跳抑制的效果
  - 拥塞点上的拥塞很快得到了缓解，但是付出的代价就是，上游路由器需要更多的缓存空间。

5.19.7 负载丢弃/载荷脱落

- 负载丢弃/载荷脱落
  - 处理拥塞的最极端方法，也是最有效的方法。
- 当路由器收到的分组超载了，一些分组会被丢掉。
- 丢分组的策略
  - 随机丢弃
  - 丢弃新到达的 (葡萄酒策略，适合文件传输类)
  - 丢弃早到达的分组 (牛奶策略，适合多媒体类)
  - 丢弃不太重要的分组
    - 需要发送方在它们的分组中标明优先级
- 随机早期检测RED (防患于未然)
  - 当情况变得恶化无可救药之前就开始丢弃分组
  - 为了确定什么时候开始丢弃分组，路由器维护着最近的队列平均长度
  - 当某条线上的队列平均长度超过了某阈值时，该线路被认定是拥塞的，可以采取相应的措施。

5.19.8 课程总结

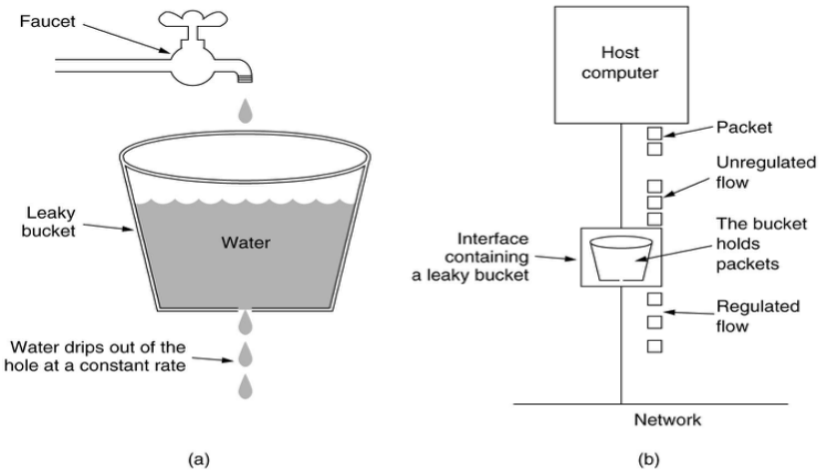
- 拥塞的量度有哪些?
- 怎么控制拥塞?
- 抑制分组和逐跳抑制分组是怎么控制拥塞的?
- 载荷脱落有哪些策略？分别适合什么样的应用?

5.20 流量整形

5.20.1 流量整形概述

- 流量整形
  - 目的：调节数据传输的平均速率 (和突发数据流)，减少拥塞
    - 原因：用户发送的数据流量忽大忽小，需要调节
- 流量整形算法
  - 漏桶 (leaky bucket)
  - 令牌桶 (token bucket)
  - 其它：资源预留、准入控制、分组调度

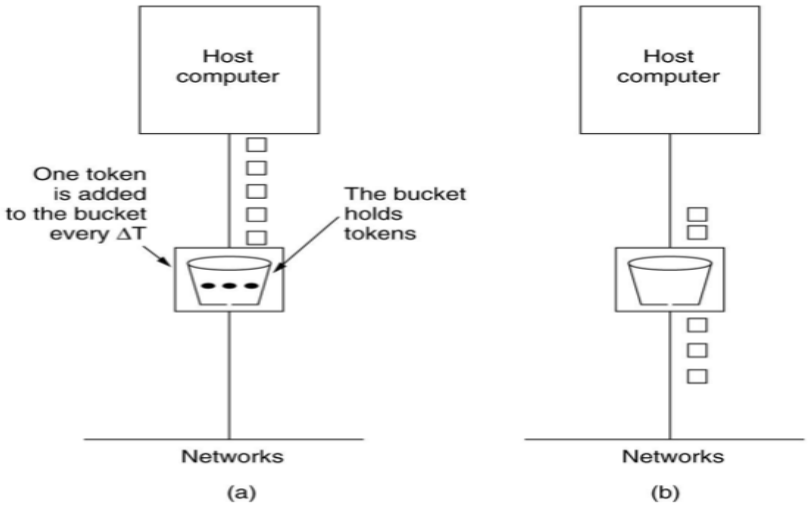
5.20.2 漏桶算法

- 具体算法描述
  - 每个主机连接到网络的接口中都有一个漏桶，即一个优先长度的内部队列。
  - 当桶中有分组的时候，输出速率是恒定的，当桶空的时候，输出速率是0。
  - 当一个分组到达满的桶的时候，分组将被丢弃 (满则溢)
  - 每个时钟滴答 (tick)，仅允许一个分组或固定数量的分组发送出去
- 算法效果
  - 主机内用户进程产生的分组流往往是一个不稳定的流，漏桶可以让它输出到网络时变成一个稳定流，抹平了突发尖峰，极大地减少了发生拥塞的机会。
- 举例
  - 水龙头速度很快，但是漏桶速度不变。
  - 

  - 数据产生速率：25MBps，路由器工作速率：2MBps，数据突发时间持续：40ms，漏桶输出速率：2MBps，漏桶容量：1MB
  - 则漏桶需要 500ms 才能处理完这些突发数据。
- 漏桶的缺点

- 当漏桶满了之后，数据将被丢弃
- 不能大量地突发数据

5.20.3 令牌桶算法

- 具体算法描述
  - 当大量数据突发的时候，令牌桶算法允许输出加快到某种程度
  - 令牌桶拥有令牌 (tokens)，且以每 ΔT 秒产生一个令牌的速度往桶中输入令牌。
  - 一个分组要发送的时候，它必须从桶中取出和获取到一个令牌。
  - 令牌桶算法允许累计令牌，但最多可以累计 n (令牌桶的容量) 个令牌。
- 与漏桶算法相比
  - 令牌桶允许突发，但是最大突发受制于令牌桶容量的限制
  - 当桶满的时候，令牌桶算法丢掉的是令牌 (不是分组)
- 举例
  - 每个分组必须拿到一个令牌才能被发送。
  - 

- 算法计算最大突发时间
  - 突发时间：S秒，令牌桶容量：B字节，令牌到达的速率：R 字节/秒，最大输出速率：M 字节/秒
  - $B+RS=MS, S=\frac{B}{M-R}$ ，即求出了能够承受的最大突发时间。
    - 最大突发时间与 B 成正比，与 (M-R) 成反比。
  - 练习题
    - 一个10Mb/s的网上有一台由令牌捅控制的主机，令牌桶以2Mb/s的速率填充。假定令牌桶有20Mb的容积。

      （1）问主机能以10Mb/s全速发送数据的最大可能的时间是多少?

      （2）主机能以10Mb/s全速发送数据的最大数据量是多少？

- 已知：

  令牌桶容量：B ＝ 20Mb

  令牌产生速率：R ＝ 2Mb/s

  最大突发速率： M＝ 10 Mb/s

  设突发时间为 s，则：

  s ＝ B / (M －R) ＝20 / (10－2) ＝2.5 (s)

- 设最大突发数据量是 Vmax，则：

  Vmax＝s× M＝2.5(s)X10Mb/s＝25M（bits）

5.20.4 课程总结

- 漏桶算法可让一个不稳定的流整为一个稳定的流
  - 无法突发
  - 桶满时丢掉的是分组
- 令牌桶可以累计突发数据的令牌，允许有上限的突发，有时候在令牌桶之后再串接一个漏桶
  - 桶满时丢掉的是令牌
- 通过整形算法，可以减少拥塞可能，提高服务质量 QoS
- 问题
  - 什么是流量整形?
  - 漏桶的工作原理是怎样的?
  - 令牌桶的工作原理是怎样的?
  - 比较漏桶和令牌桶两种整形算法?

第六章 传输层

6.1 传输层概述

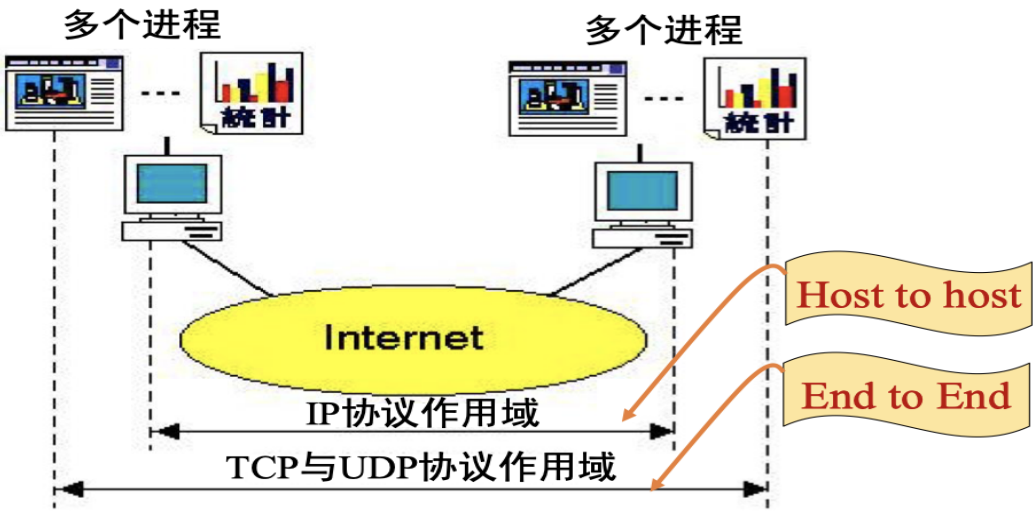
6.1.1 传输层概述

- 传输层：整个协议栈 (TCP/IP) 的核心
  - 任务：提供可靠的、高效的的数据传输
- 传输实体：完成传输层任务的硬件或软件
  - 可能位于：
    - 操作系统内核

        - 独立的用户进程
        - 绑定在网络应用中的链接库
        - 网络接口卡
- 传输层服务
    - 面向连接的服务 (与网络层提供的服务相似)
    - 无连接的服务
- 传输层提供的功能
    - 端点标识
    - 传输服务
        - 面向连接：端到端的连接服务、流控制、差错控制
        - 无连接
- 两大传输层协议
    - UDP (User datagram protocol) —— 用户数据报协议
    - TCP (Transport control protocol) —— 传输控制协议

6.1.2 传输层的必要性

- 为什么需要两个独立的不同的层?
    - 网络层运行在由承运商操作的路由器上，因此用户无法真正控制到网络层。
    - 把另一层放在网络层之上，可以让用户能够控制到服务质量。
    - 传输层服务原语独立于网络层原语，而网络层原语会因为网络的不同而不同。
- 网络层与传输层作用范围比较?
    - 网络层负责把数据从源机送达到目的机
    - 传输层负责把数据送达到具体的应用进程



    - 传输层：end to end，此处 end 为端点 (end point) 或套接字 (socket)

6.1.3 数据链路层与传输层的比较

- 功能相似

    - 流控制、差错控制、数据 (报文/帧) 排序
- 传输环境不同
    - 通信子网
    - 点对点的物理信道 (数据链路层)

6.1.4 传输服务原语

- 传输服务原语让应用程序可以有途径访问到传输服务
- 传输服务和网络服务的两个主要差别是：
    - 网络服务试图按照实际网络提供的服务来建模 (不可靠的)，而面向连接的传输服务是可靠的。
    - 网络服务仅被传输实体所使用，而传输服务直接被应用程序所使用，必须方便易用。
- 简单的传输服务原语
    -

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

6.1.5 数据段

- TPDU (Transport Protocol Data Unit)
    - 从传输实体发到对端传输实体的信息。
- TPDUs被封装在分组 (Packet) 中，由网络层交换
    - 分组被封装在帧 (frames) 中，由数据链路层交换

6.1.6 课程总结

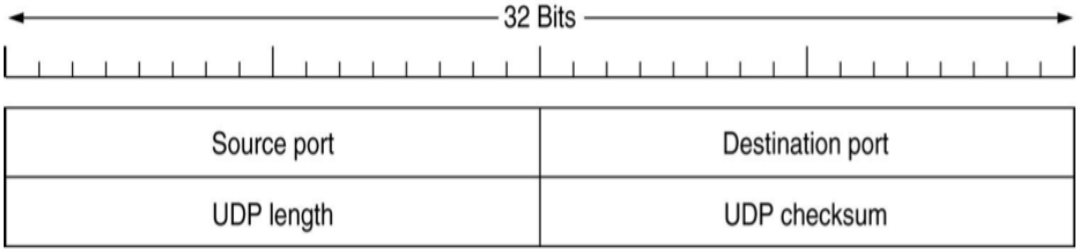
- 传输层提供什么样的服务?
- 为什么需要一个独立于网络层的传输层?
- 传输层的协议数据单元 (PDU) 是什么?
- 传输层的协议有哪些?

6.2 用户数据报协议 UDP

6.2.1 UDP 概述

- UDP 特点
  - UDP是一个无连接的传输层协议
  - UDP传输数据段，无须建立连接，不提供数据段的可靠传输
  - 如DNS通过UDP发送一个请求，然后对方进行应答

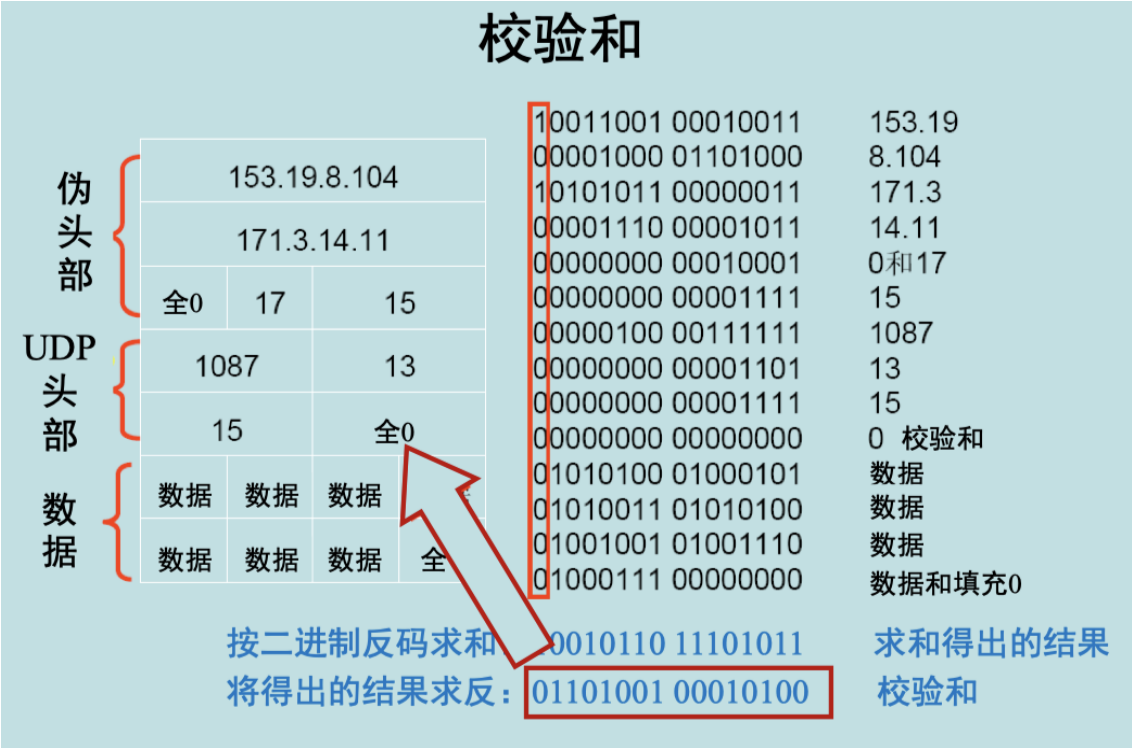
6.2.2 UDP 数据段头

- UDP数据段包括头部和数据两个部分，其中头部有4部分，每部分2字节，共 8 字节
  - 

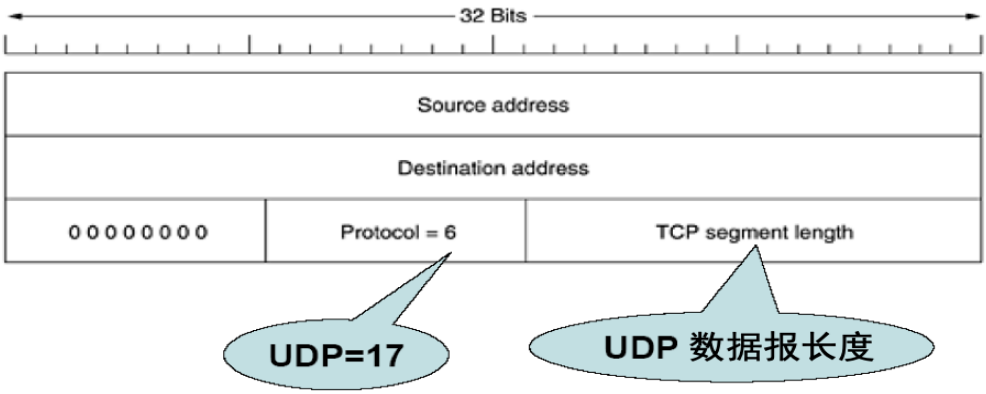
  - 第一部分和第二部分分别为源端口和目的端口
  - 第三部分的长度包括头部和数据总共的长度
  - 校验和可选，如果不计算校验和，则该域置为 0。
  - UDP比IP好的地方在于它可以使用源端口和目的端口。
- 端口
  - 端口共16位，共有 2^{16} 个端口
  - 端口范围：0~65535
  - 端口分为三部分
    - ① ≤ 1023，用于公共应用 (保留，全局分配，用于标准服务器)，由 IANA 分配
    - ② 1024~49151，非特权用户端口，注册端口
    - ③ ≥ 49152，动态端口，私人端口
  - 自由端口
    - 本地分配、动态的随机端口
    - 访问网站时，网站的端口通常是一些知名端口 (比如80)，而本地主机端口是系统自动分配的一个自由端口，通常 ≥ 49152。

6.2.3 UDP 校验和

- 校验和
  - 将IP伪头部、UDP段头、数据三个部分，按照16位一行，然后按列进行补码相加求和，再将相加结果拿来取反码，作为最后的校验和。
  -



- 如果收方的校验和为全 1，则传输无错
- 二进制反码求和
  - 从低位到高位逐列计算
  - 0和0相加是0，0和1相加是1，1和1相加是0，但产生进位
  - 最高位相加产生进位，该位为 1
- 检错能力较弱，但简单快速
- 使用协议地址，破坏了分层原则

- TCP/UDP 伪头部
  - 

  - 第一、二字段为源、目的IP地址
  - 第四字段为用户协议，6为TCP，17位UDP
  - 第五字段为UDP头部中的长度字段 (UDP头部与数据长度之和)

6.2.4 课程总结

- UDP 提供端点标识，端到端的数据传输
- UDP 不提供差错检测和可靠传输，但简洁高效

- UDP 是传输层的一个轻量级协议，提供高效的端到端的数据段传输
- UDP 数据段包括头部和载荷两部分，头部有8个字节，共4个字段。
- UDP 不提供数据传输的可靠保证。
- 问题
  - 为什么需要 UDP?
  - UDP 的段格式是怎样吗?
  - UDP 提供的服务是可靠的吗?
  - UDP 提供的服务可以用什么样的词来描述?

6.3 通信模型

- 通信 5 元组
  - 源端点：源IP、源端口
  - 目的端点：目的IP、目的端口
  - 协议：UDP、TCP
- 知名端口
  - The list of well-known ports is given at http://www.iana.org

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial file transfer protocol
79	Finger	Lookup information about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

不同于交换机上的端口，此端口非彼端口！

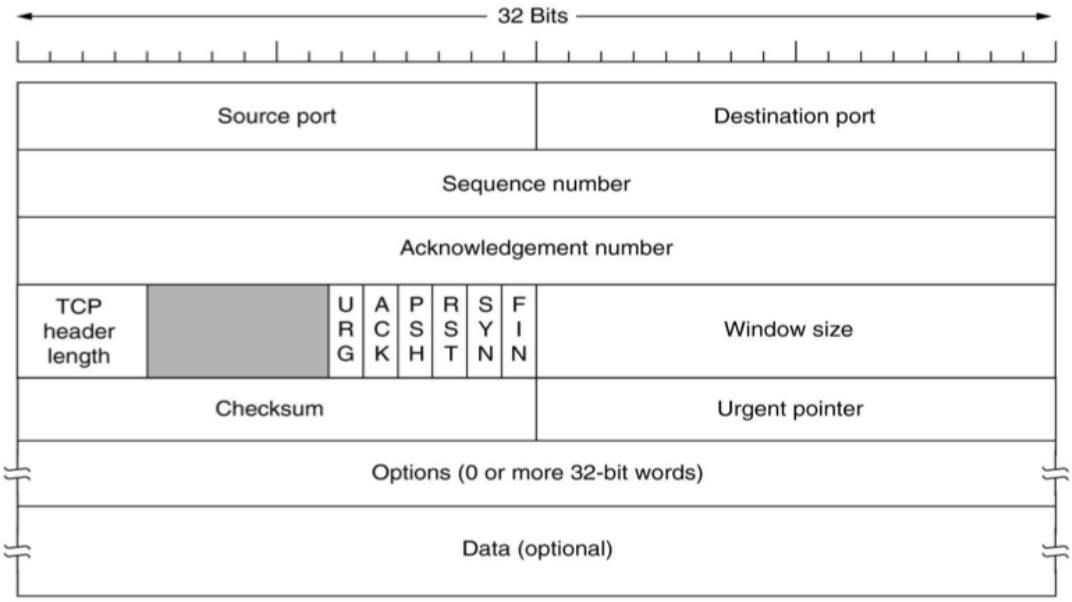
6.4 TCP 数据段

6.4.1 TCP 概述

- 传输控制协议 (TCP - Transmission Control Protocol)
  - TCP 是专门为了在不可靠的互联网络上提供可靠的端到端字节流而设计的。

    - TCP 必须动态地适应不同的拓扑、带宽、延迟、分组大小和其它的参数，并且当有错误的时候，能够足够健壮。
- 支持 TCP 的机器都有一个 TCP 实体，或者是用户进程或者是操作系统内核，都可以管理 TCP 流和跟 IP层的接口。
    - 发：封装
        - TCP实体接收本地进程的用户数据流，将其分割成不超过64KB的分片 (实践中，通常分割成1460字节，以通过以太网传输)
    - 收：解封装
        - 当包含TCP数据段的报文到达某台机器的时候，被提交给传输实体，传输实体将其重构出原始的字节流。
- TCP协议
    - TCP连接上的每个字节都有它自己独有的32位序列号
    - 收发双方的TCP实体以数据段的形式交换数据
    - 一个数据段包括20字节的头部 (不包括可选项) 和数据域 (0或更多字节)
    - TCP软件决定数据段的大小，有两个因素限制了数据段的长度
        - TCP数据段必须适合IP的 $65515 = 65535 - 20$ 的载荷限制
        - 每个TCP数据段必须适合于下层网络的MTU (如，1500字节 - 以太网载荷大小)
    - TCP使用的基本协议具有动态窗口大小的滑动窗口协议

6.4.2 TCP 数据段格式 (TPDU)

- TCP 数据段格式 (TPDU)
    - 20字节的头部 (不包括可选项)、变长的数据字段/域 (可以是0个或更多字节)、不含选项的头部有14个字段/域
    - 

- TCP 数据段头
    - 第一、二个字段：源端口、目的端口
        - 标明了一个连接的两个端点

    - 用来跟踪同一时间内通过网络的不同会话。一般每个端口对应一个应用程序。
  - 第三个字段：序列号
    - 字节号 (32位) [当前所携带的数据的第一个字节的顺序号]
    - 初始序列号 ISNs (initial sequence numbers)：随机产生
    - SYN (连接请求数据段)：携带了ISNs和SYN控制位的数据段
  - 第四个字段：确认号
    - 期望接收的字节号 (32位)
    - TCP的可靠传输保证，采用了肯定确认机制。
    - 也采用了累计确认，如果确认号为500，表明收到了序列号小于500的所有数据段。
  - 第五个字段：TCP 段头长度 (4 bits)
    - 单位：4字节
  - 第六个字段：保留域/字段
    - 可以用来拥塞控制
  - 接下来是6个控制比特位
  - 第六个字段：URG
    - 当紧急指针使用的时候，URG被置为 1。紧急指针是一个对于当前序列号的字节偏移量，标明紧急数据从哪里开始。
    - 当 URG = 1 时，表明有紧急数据，必须首先处理。
    - 与紧急指针配合使用。
    - 收方收到这样的数据后，马上处理，处理完后恢复正常操作。
    - 即使 win = 0 (告诉对方不可以发送数据)，也可以发送这样的紧急数据段。(比如 ctrl+c 中断命令)
  - 第七个字段：ACK
    - ACK = 1，表明确认号有效，启用了捎带确认
    - ACK = 0，表明确认号失效
  - 第八个字段：PSH
    - PSH = 1，表明这是带有PUSH标志的数据
    - 接收方收到PUSH标记的数据，应该立刻送到上层，而不需要缓存它。
  - 第九个字段：RST
    - 重置一个已经混乱的连接
    - 如果在连接建立阶段，就直接拒绝建立连接
  - 第十个字段：SYN
    - SYN用在连接建立的过程
    - SYN = 1，ACK = 0，表明请求连接
    - SYN = 1，ACK = 1，表明连接接受
  - 第十一个字段：FIN
    - 用来释放连接，表明发送方已经没有数据要传输了，但是可以继续接收数据。
  - 第十二个字段：Window Size
    - TCP中的流控使用一个可变长的滑动窗口来完成的。
    - Window Size，告诉对方可以发送的数据字节数 (从确认字节号开始 - 由接收方决定)
  - 第十三个字段：Checksum
    - 提供额外的可靠性

        - 检验的范围包括头部、数据和伪头部 (伪头部中用户协议取值为6)
    - 第十四个字段：紧急指针
        - 和URG控制位配合使用，指明了紧急数据
    - 第十五个字段：选项域
        - 提供了一种增加基本头没有包含内容的方法
- 选项字段
    - 允许每台主机指定他愿意接收的最大TCP净荷长度
    - 窗口尺度 (Window scale) 选项允许收发双方协商一个窗口尺度因子，这个因子允许双方把窗口尺寸域向左移动至14位
    - 选择性重传选项
- 例题
    - 9 多选 (2分) 下列哪个信息不是TCP端点发送的数据段头部中的一部分？ 得分/总分

      A. 数据段大小 ✔1.00/2.00

      B. 端点接收到的数据段中的目标端口

      C. 在 TCP 流中接收数据的端点中的可用缓冲空间

      D. 目的地 IP 地址 ✔1.00/2.00

      正确答案：A、D 你选对了

      解析： A、不对，段头部中只有段头长度字段 B、目的端口 C、是WindowSize D、不对，这是分组头部内容

      10 判断 (2分) TCP提供了面向连接的数据传输服务，保证了每一个字节的绝对可靠。 得分/总分

      A. ✕ ✔2.00/2.00

      B. ✔

      正确答案：A 你选对了

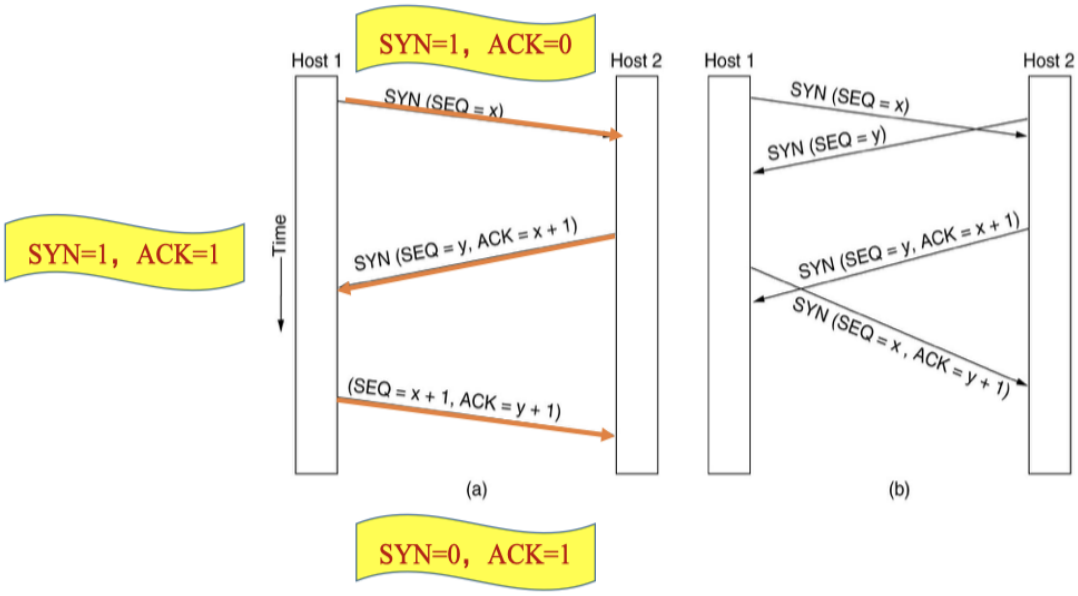
      解析：大部分情况是可以可靠传输的，但是也有可能出现一些特例，比如校验和验不出来的错误。

6.4.3 课程总结

- TCP数据段中的序列号是段的编号吗?
- TCP数据段中的初始序列号是怎样来的?
- TCP数据段中的确认号有什么用?
- TCP数据段中的窗口尺寸有何作用?

6.5 TCP 三次握手建立连接

6.5.1 TCP 连接的建立

- 采用三次握手建立连接
  - 一方 (server) 被动地等待一个进来的连接请求
  - 另一方 (the client) 通过发送连接请求，设置一些参数
  - 服务器方回发确认应答
  - 应答到达请求方，请求方最后确认，连接建立
- TCP 连接建立图示
  - 

  - 第一次 Host1 向 Host2 发连接请求
    - 连接请求数据段：SYN = 1， ACK = 0，SEQ = x，SEQ为系统随机产生
  - 第二次 Host2 向 Host1 发连接应答
    - 连接应答数据段：SYN = 1，ACK = 1，SEQ = y，ACK = x+1，SEQ为系统随机产生
  - 第三次 Host1 再向 Host2 发确认数据段
    - 确认数据段：SYN = 0，ACK = 1，SEQ = x+1，ACK = y+1，三次握手结束，连接建立
- 三次握手建立TCP连接，也称为同步。这个过程中，双方交换了一个最重要的参数 —— 初始序列号，这个可以用来跟踪后续交换的每一个字节。即后续的字节的编号就是以这个初始序列号作为基础的。建立TCP连接的双方没有主从之分，它们可以相互收发数据，因此TCP数据段的传输是全双工的。
- 为什么是三次握手?
  - 为了实现可靠数据传输， TCP 协议的通信双方， 都必须维护一个序列号， 以标识发送出去的数据包中， 哪些是已经被对方收到的。 三次握手的过程即是通信双方相互告知序列号起始值， 并确认对方已经收到了序列号起始值的必经步骤。
  - 如果只是两次握手， 至多只有连接发起方的起始序列号能被确认， 另一方选择的序列号则得不到确认。

6.5.2 DoS 攻击

- SYN泛洪导致DoS攻击 (伪造源IP)

    - 大量 Agent 向被攻击方发送TCP请求连接数据段，但源IP伪造成一个不存在的IP
    - 由于不会有主机回发确认数据段，被攻击方会挂起大量进程，最终资源耗竭而崩溃。
- 数据传输开始后可能有两个原因导致阻塞
    - 快的机器向慢的机器发送数据
    - 多台机器同时向一台机器发送数据

6.5.3 课程总结

- TCP连接是怎样建立起来的?
- 使用2次握手来建立TCP连接可以吗?
- 为什么TCP连接过程又叫同步?
- SYN泛洪攻击是怎样产生的?

6.6 TCP 连接释放

6.6.1 TCP 连接释放

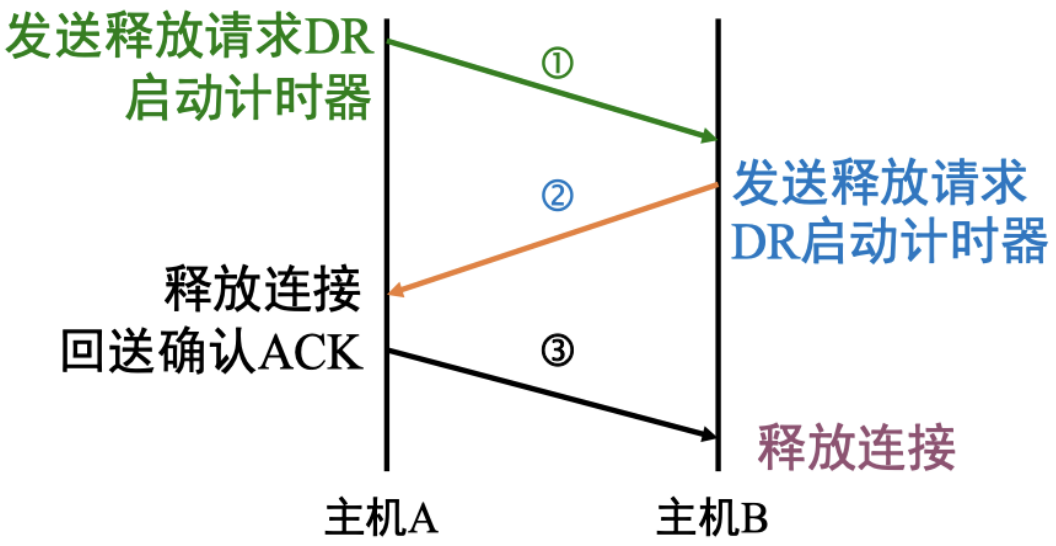
- TCP 连接释放
    - 任何一方在没有数据要传送的时候，都可以发送一个 FIN = 1 的 TCP 数据段
    - 当 FIN 被确认之后，该方向的连接关闭
    - 当双向连接都关闭了的时候，连接释放
- 连接释放的问题
    - 连接释放取决于另一方，这导致连接释放要求连接的两端都释放，但是决定什么时候两边释放非常困难
    - 最后信息的发送者，永远无法知道这个信息是否到达
- 解决方案 1
    - 将释放连接的决定权交给请求者独立裁定，而不是对方。
        - 比如，一方发送连接释放请求DR (Disconnect Request)，且期待对方的确认ACK
        - 当DR到达接收端，它回发ACK，并且也发送一个DR
        - ACK到达发送端的时候，连接释放；同时，它回发确认ACK，当这个ACK到达接收端，反方向的连接也释放了。
- 解决方案 2：使用定时器
    - 如果一方发送了FIN数据段出去，却在一个设定的时间没有收到应答，释放连接
    - 另一方最终会注意到连接的对方已经不在了，超时后连接释放。

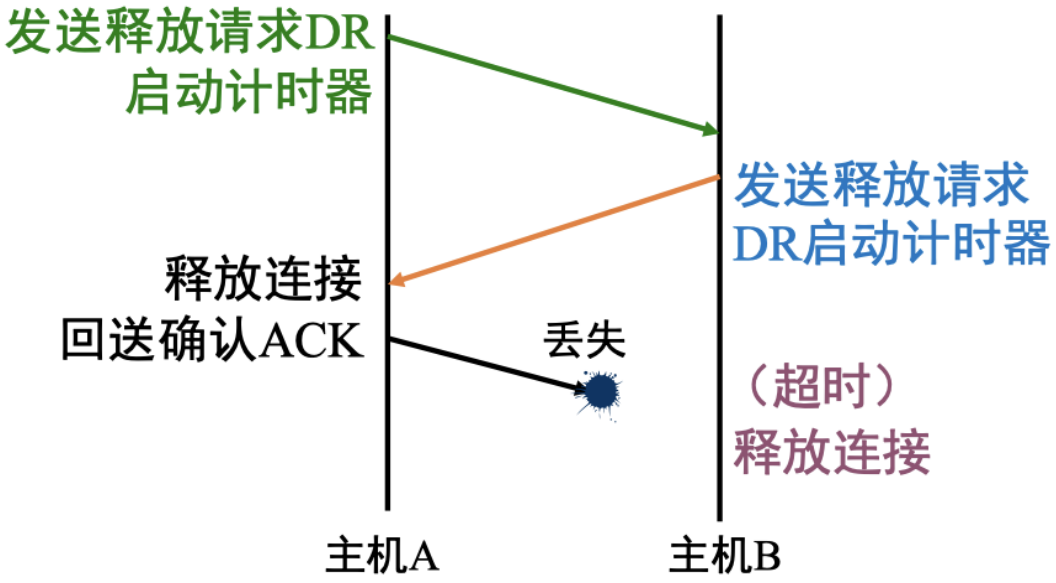
6.6.2 半开放连接

- 定义
    - 理论上讲，如果初始DR和重传都丢了，发送者将因超时放弃发送且释放连接。但是，另外一端却不知道这些情况，仍然处于活跃的状态。

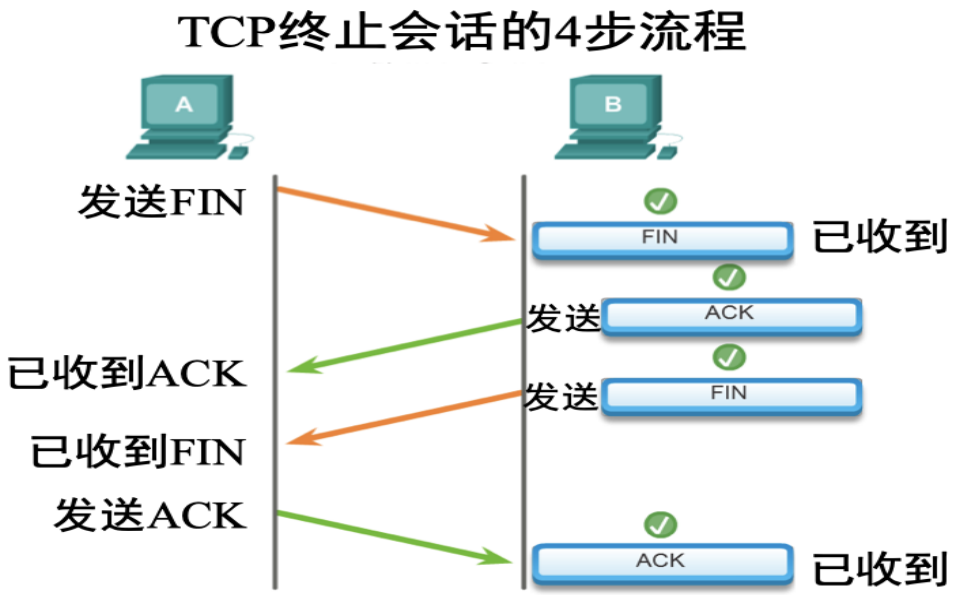
    - 这种情况导致半开放连接。(half-open)
- 杀死半开放连接的方式
    - 如果在一定的时间内，没有TPDUs到达的话，连接自动释放
    - 因此传输实体在发送一个TPDU的时候必须启动定时器，定时器超期，将发送一个哑TPDU，以免连接断开。

6.6.3 三次握手正常释放连接

- 正常释放连接过程
    - 

- 如果确认ACK丢失
    - 

- TCP 4步终止会话

- 

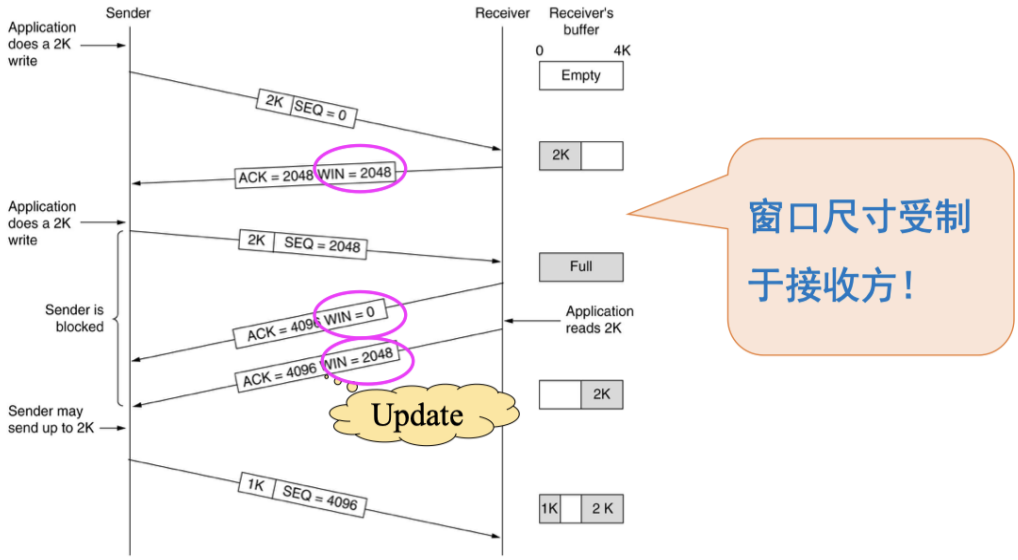
6.6.4 课程总结

- TCP是全双工的，连接的释放必须是双向的
  - FIN = 1
- 半开半闭的连接必须杀掉
- 超时会挂掉连接
- 问题
  - TCP连接是怎样释放的?
  - 半开半闭的连接怎么办?

6.7 TCP 传输策略

6.7.1 Window Size 流量控制

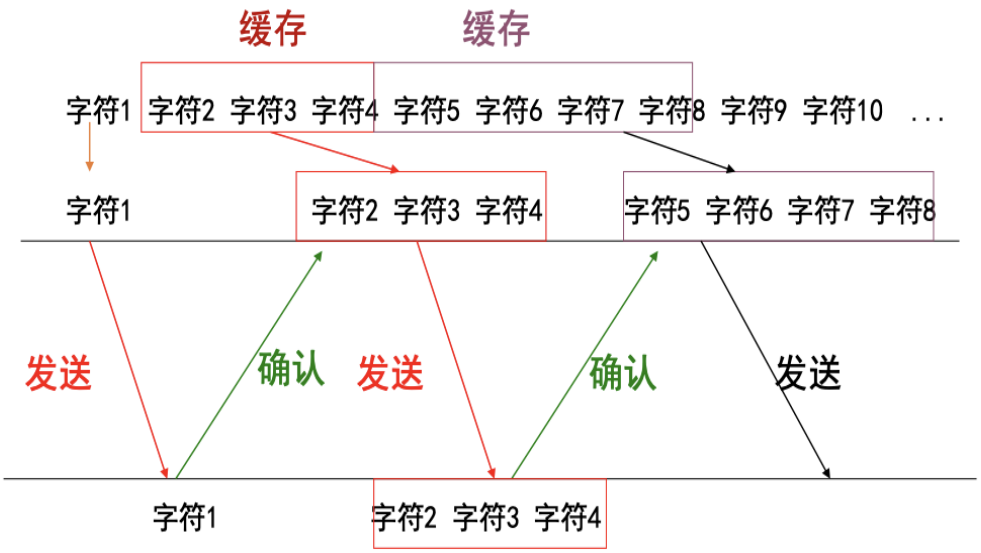
- Window Size 流量控制具体过程

- 

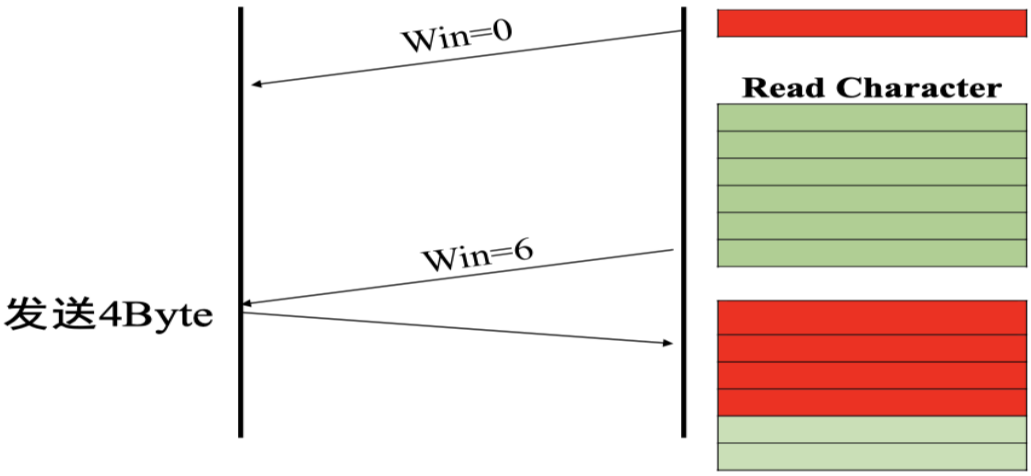
  - 窗口尺寸由接收方决定，当WIN = 0时，发送方不能再发送数据。当接收方有流量之后，会像发送方重新发送TCP数据段，更新WIN。
- WIN = 0
  - 当 WIN = 0 时，发送者不能正常发送数据段，除非
    - Urgent数据。例如，用户想杀掉远端机器上的进程时，可以发送数据。
    - 发送者可以发送一个字节的数据段，以便让接受者再次发送期待的字节号和窗口数 (避免死锁)

6.7.2 优化发送端

- Nagle's algorithm
  - 当数据以一次一字节的速度到达的时候，只发送第一个字节，然后将后续的字节缓存起来，直到发出的字节得到确认。
  - 将缓存起来的字节在一个数据段中发出，再继续缓存，直到发出的数据得到确认。
  - Nagle算法在很多TCP上实现，但是有些时候最好禁用。
    - 比如，当一个X-Window应用在互联网运行的时候，鼠标的移动事件必须发送给远程计算机，把这些移动事件收集起来一批一批发送出去，使得鼠标的移动极不连贯。
- Nagle 算法图示

- 

6.7.3 优化接收端

- Clark 算法
  - 阻止接收方发送只有 1 个字节的窗口更新，相反，它必须等待一段时间，当有了一定数量的空间之后再告诉发送方。
  - 接收方可以维护一个内部缓冲，且阻塞上层应用的 READ 请求，直到它有大块的数据提供。
- Clark 算法图示
  - 

6.7.4 课程总结

- 发送方
  - 尽量不发送数据含量小的数据段
  - 缓存应用层的数据，达到一定量再发送
- 接收方
  - 不请求对方发送短数据段 (window size)
  - 延迟窗口变更信息，使接收缓冲区足够大
- 问题

- TCP如何进行流控?
- 怎样提高TCP传输的效率?
- TCP的发方怎样优化?
- TCP的收方怎样优化?

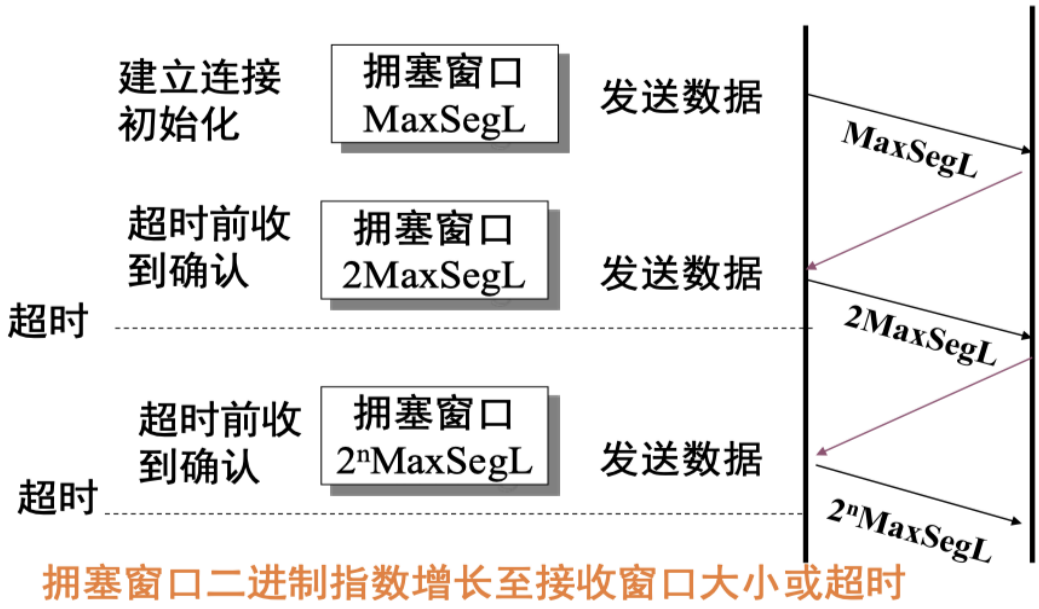
6.8 TCP 拥塞控制

6.8.1 TCP拥塞概述

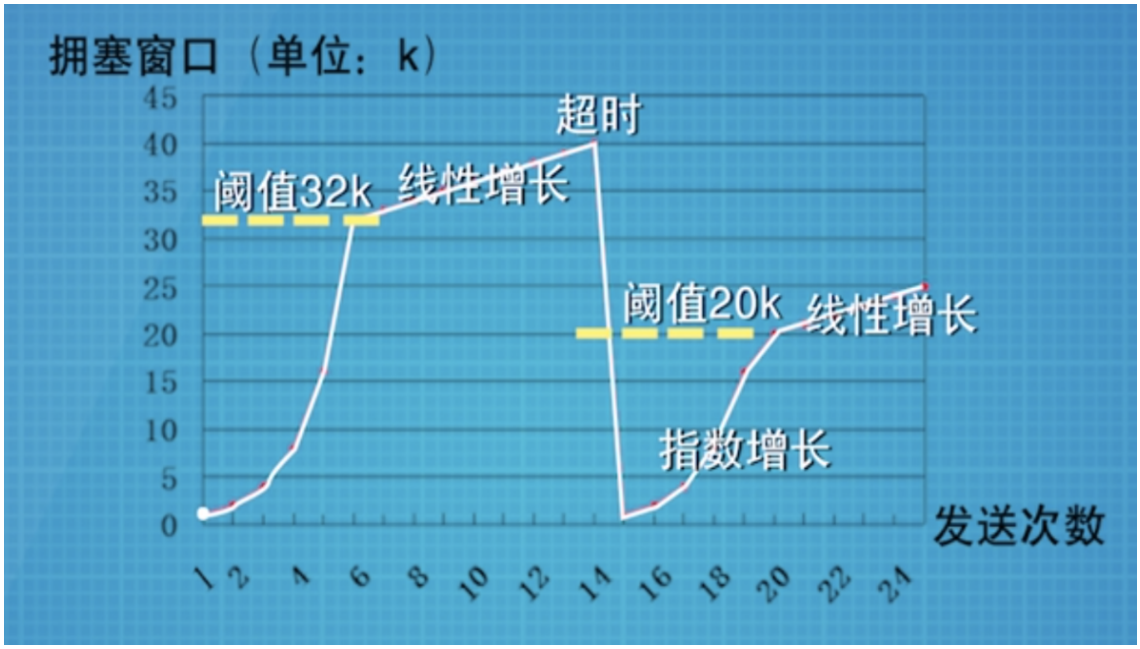
- 分组守恒：当有一个老的分组离开之后才允许新的分组注入网络
  - TCP 希望通过动态维护窗口大小来实现这个目标
- 拥塞检测
  - 所有的互联网TCP算法都假定超时是由拥塞引起的，并且通过监视超时的情况来判断是否出现问题。
- 拥塞控制
  - 接受者容量 (维护接收者窗口 —— 容易控制)
    - 当一个连接建立的时候，双方选择一个合适的窗口大小，接收方根据自己的缓冲区大小来指定窗口的大小。
    - 如果发送者遵守此窗口大小的限制，则接收端不会出现缓冲区溢出的问题，但可能由于网络内部的拥塞而发生问题。
  - 网络容量 (维护拥塞窗口 —— 难于控制)

6.8.2 慢启动算法 (Slow Start)

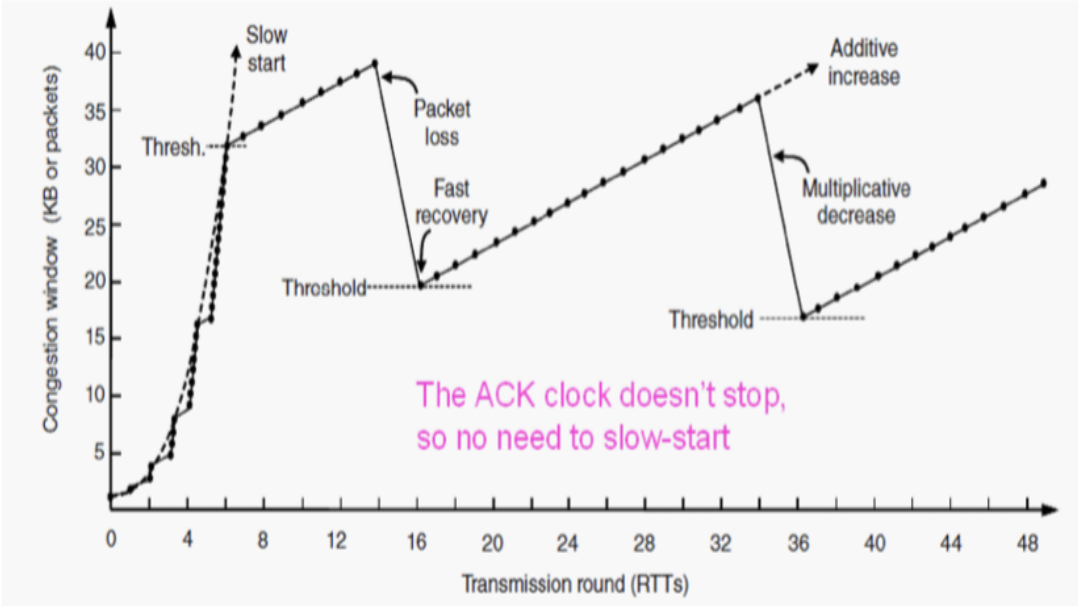
- 慢启动算法过程
  - 当连接建立的时候，发送者用当前使用的最大数据段长度初始化用塞窗口，然后发送一个最大的数据段。
  - 如果在定时器超期之前收到确认，则将拥塞窗口翻倍，然后发送两个数据段.....直至超时 (或达到接收方窗口的大小)
  - 确定出拥塞窗口的大小
    - 例如，一开始发4096字节没有问题，下一次就发8192字节，如果发生超时，则拥塞窗口设为4096字节。
- 慢启动算法图示

- 

6.8.3 阈值

- 除了使用接受者窗口和拥塞窗口，TCP拥塞控制还是用了第三个参数，阈值 (threshold)，初始化为 64K。
  - 当一个超时发生的时候，阈值降为当前拥塞窗口的一半，同时将拥塞窗口设为一个最大数据段的长度。
  - 使用慢启动算法来决定网络的容量，拥塞窗口增长到阈值时停止指数增长。从这个点开始，每次成功的传输都会让拥塞窗口线性增长 (每次仅增长一个最大的数据段长度)
- 慢启动算法+阈值图示
  - 

  - 一开始2倍倍增增长，到达阈值后变为线性增长。
  - 线性增长到超时之后，阈值降为一半，拥塞窗口回到初始值。重复步骤。
- 线性增长，可以将越来越粗放的窗口尝试粒度变小，以获得更准确的拥塞窗口值。
- TCP慢启动算法就是这样不断超时、不断重启，尝试出的拥塞窗口值也随着网络状况的变化而发生变化，达到拥塞控制的目的。

- 快速恢复
  - 重新慢启动时，拥塞窗口值不用重置为一个数据段大小，而是可以设置为阈值大小，从这里直接开始线性增长，即快速恢复。此时，反复的慢启动过程呈现出一个锯齿形状。
  - 

6.8.4 课程总结

- TCP拥塞控制的原则是什么?
- 引起TCP拥塞的两种因素是什么?
- TCP怎么进行拥塞控制?
- 拥塞窗口怎么获得?
- 阈值有什么作用?
- 多次慢启动尝试，CWND是否会达到一个不变的常数?

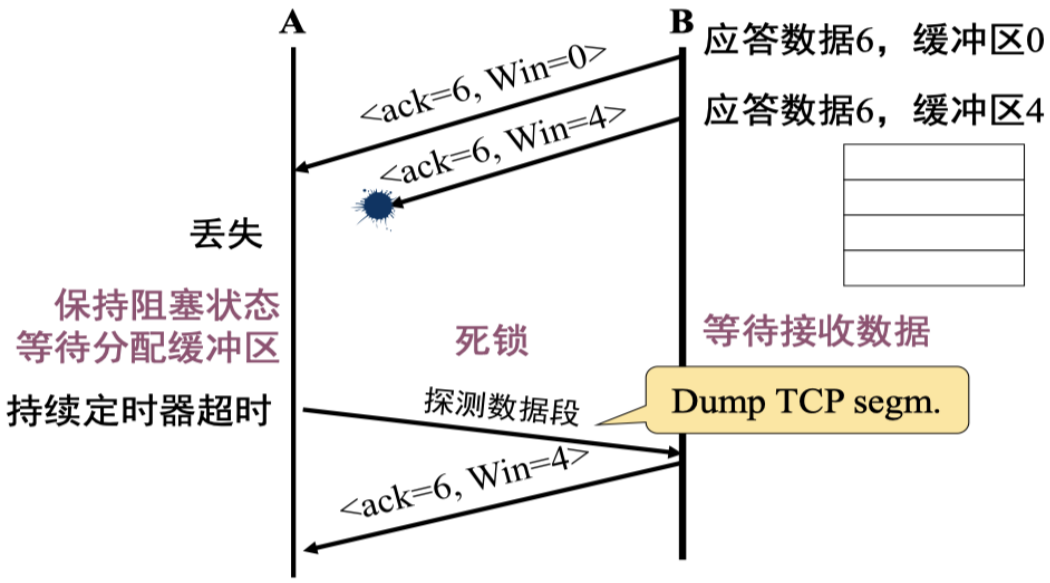
6.9 TCP 定时器

6.9.1 重传定时器

- 为了解决数据段丢失的问题，每发一个数据段都会启动一个定时器 (重传定时器)。
- 该定时器的时间长短设置需要仔细的考量。

6.9.2 持续定时器

- 避免如下的死锁发生：
  - 接收方发送了一个窗口数为0的确认 (窗口更新)，告诉发送方等待。
  - 稍后，接收方空出了缓存，发送更新窗口的数据段，但是该分组丢失了。
  - 现在，收发双方都在等待对方发送数据段过来，但大家都等不到，由此死锁产生！

- 解决死锁的方法：
  - 发送方在收到 win = 0时，启动一个持续定时器，如果定时器超期没有收到更新窗口，则发送一个探测数据段，引发对方重新发出更新窗口！
  - 

6.9.3 保活定时器

- 保活定时器 (keep-alive timer)
  - 用来检查连接是否存活，当一个连接空闲的时间超过保活定时器的时间，该连接将被杀掉。
  - 在关闭时刻处于timed wait状态中使用的定时器：运行两倍的最大分组生存时间，以确保连接关闭之后，该连接上的所有分组都完全消失。

6.9.4 课程总结

- TCP与UDP的比较
  - 可靠性：TCP可靠，UDP不可靠
  - 传播延迟：TCP不确定，UDP为网络延迟
  - 拥塞控制：TCP有拥塞控制，UDP没有拥塞控制
- 选择TCP的情况
  - 需要可靠传输方式
  - 需要让应用程序简单化，程序员不必进行错误检查、修正等工作
- 选择UDP的情况
  - 为了降低对计算机资源的需求 (DNS)
  - 应用程序本身已提供数据完整性的检查机制，勿须依赖传输层的而协议来保证。
  - 应用程序传输的并非关键性的数据。
  - 一对多方式，必须使用UDP (视频传播) (TCP仅限于一对一的传送)
- 问题
  - TCP中的重传定时器如何工作？
  - TCP的持续定时器用来做什么？如何做？

- 比较UDP和TCP?
- 如何选择传输层的协议?

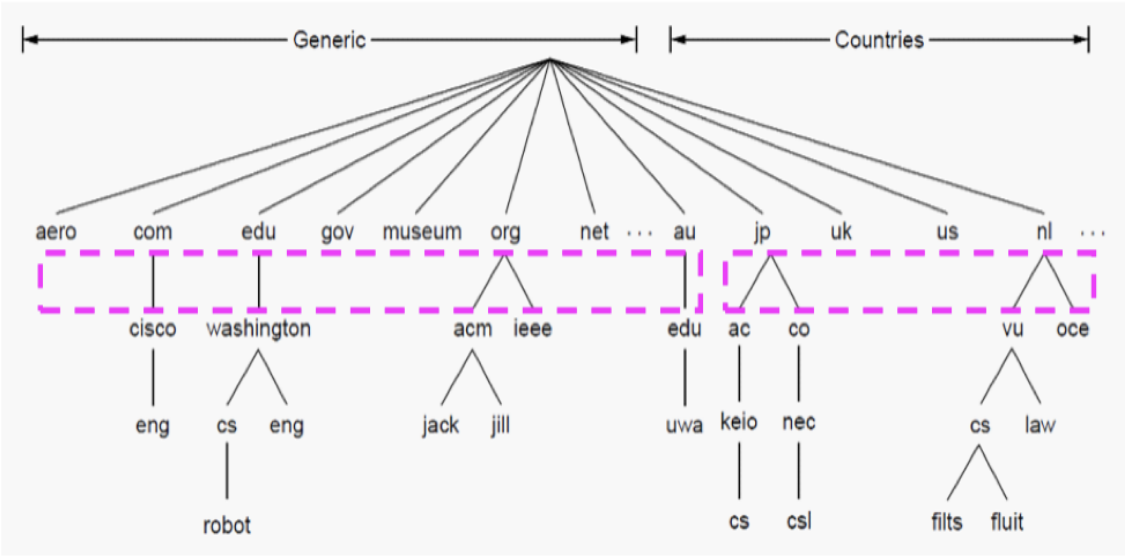
第七章 应用层

7.1 应用层概述

- 应用层直接为用户服务
- 应用层特点
  - 直接网络应用程序
    - 浏览器、电子邮件、文件传输、远程登录
  - 间接网络应用程序
    - Word、资源管理器
    - 依靠重定向器实现网络功能。
    - 重定向器是置于应用中的小软件，它是透明的。

7.2 域名系统 DNS 概述

7.2.1 域名概述

- 域名系统 (DNS - Domain Name System)
  - 域名系统是分层次的，并使用分布式数据库实现。
  - 两类顶级域名：通用域名与国家域名
  - 

The computer *robot.cs.washington.edu*

- 域名的表示方式
  - 每个域的名字：从它向上到根的路径，各个部分间用圆点隔开
  - scut.edu.cn —— 三级域名 . 二级域名 . 顶级域名
- 域名的组成
  - 域名无关大小写。
  - 各组成部分的名字最多有63个字符长，整个路径不超过255个字符。
  - 没有规则限制同时在两个或多个顶级域名下的注册。
- 域的权限
  - 每个域自行控制其下面的子域的划分
    - 日本的 ac.jp 和 co.jp 分别对应于 edu 和 com

7.2.2 域的资源记录

- DNS的主要功能：将域名映射到资源记录上
- 域的资源记录：域名、生存时间、类别、类型、值
  - 域名：指出这条记录适用于哪个域
    - 域名字段是匹配查询条件的主要关键字。
    - 每个域有多条记录，数据库则保存多个域的信息。
    - 资源记录在数据库中的顺序无关紧要。
  - 生存时间：表明这条记录的稳定程度。
    - 极稳定的资源记录会被分配一个很大的值。
    - 非常不稳定的资源记录会被分配一个很小的值。
  - 类别：只要一个类别 (IN)
  - 类型：指出这是什么类型的记录。
    - A：IPv4地址
    - AAAA：IPv6地址
  - 值：类型对应的值，可以是数字，也可以是ASCII字符串等。
- 资源记录存储
  - 存储在域名服务器中。
  - 互联网需要多台域名服务器。将互联网划分多个区域，每个区域有一个主域名服务器，还有一个次域名服务器。
    - 主域名服务器和次域名服务器之间同步数据。
  - 根域名服务器
    - 存储所有顶级域名的名字和对应IP地址等信息。
    - 全球一共有13个根域名服务器，有100+镜像个域名服务器。

7.2.3 课程总结

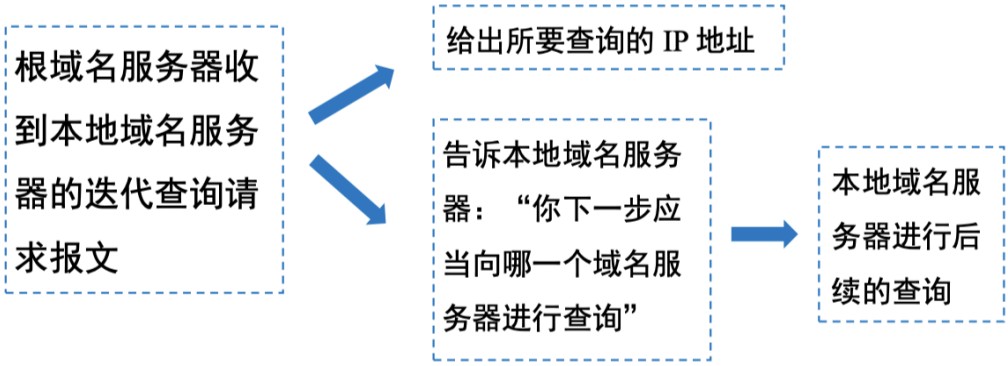
- IP地址难以记忆，且IP地址经常发生变动，便出现了域名。
- DNS是一个层次化的分布式数据库系统，提供域名解析服务。
- 域名解析的结果主要是资源记录。

- 任何域名均对应一条或多条资源记录。
- 资源记录包括权威资源记录。
- 资源记录均存储在域名服务器中。

7.3 DNS域名解析

- 域名解析过程
  -

  - DNS的使用方法：
    - 为了将一个名字映射为IP地址，应用程序调用一个叫解析器（resolver）的库过程，把名字作为参数传递给这个过程（如：gethostbyname()就是一个解析器）
    - 解析器发送一个UDP分组给本地DNS服务器，它会负责查找该名字，然后将对应的IP地址返回给解析器
    - 解析器返回结果给应用程序，然后应用程序即可开始工作了（封装，发送……）
- 域名解析的两个类别
  - 递归查询
    - 向本地DNS服务器查询资源记录
    - 主机所询问的本地域名服务器不知道被查询域名的 IP 地址 → 本地域名服务器以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文
  - 迭代查询

- 

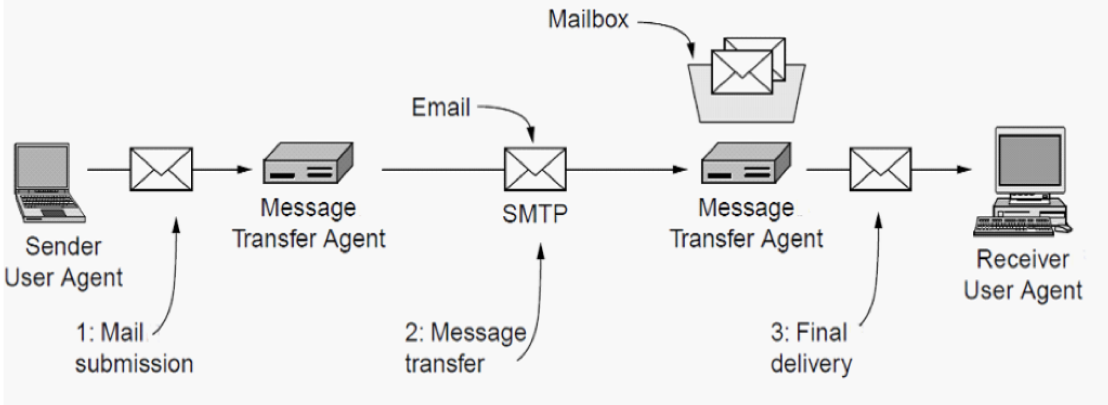
-

- 优化方法
  - 高速缓存 —— 减少查询环节，提高效率
  - 缺点：缓存中的内容不具有权威性

7.4 电子邮件

7.4.1 电子邮件系统概述

- 电子邮件系统由两部分组成
  - 用户代理 (UA)：让用户能够阅读和发送邮件
    - 本地程序，提供命令行或图形界面，让用户和电子邮件系统交互
    - 例如Gmail、Outlook等
    - 功能：邮件显示、邮件归档、邮件处置、自动响应、签名块、邮件列表
  - 消息传输代理 (MTA)：将消息从源端送到目标端
    - 通常是系统守护进程，即运行在后台的进程，在系统中传递电子邮件
- 传输过程图示

- 

- 电子邮件消息格式
  - ASCII电子邮件采用RFC 822
    - 消息由一个信封 (RFC 821)、一些头域、一个空行和消息体组成。
    - 每个头域由一行ASCII文本组成，包括域名、一个冒号，对于大多数头域来说，还包括一个值
  - 用户可以发明新的消息头供自己私人使用，只要这些消息头以 X- 开头
  -

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

RFC 5322

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	E-mail address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display

7.4.2 MIME

- 多用途互联网邮件扩展 (MIME - the Multipurpose Internet Mail Extensions)
  - 解决问题：传输带有重音符的语言、非拉丁字母、不带字母的语言、完全不包含文本的消息
- MIME基本思想
  - 继续使用 RFC 822 格式，但是在消息体中增加了结构，为非 ASCII 消息定义了编码规则
  - MIME可以使用现有的程序和协议来发送，所有必须要改变的是接收和发送的程序

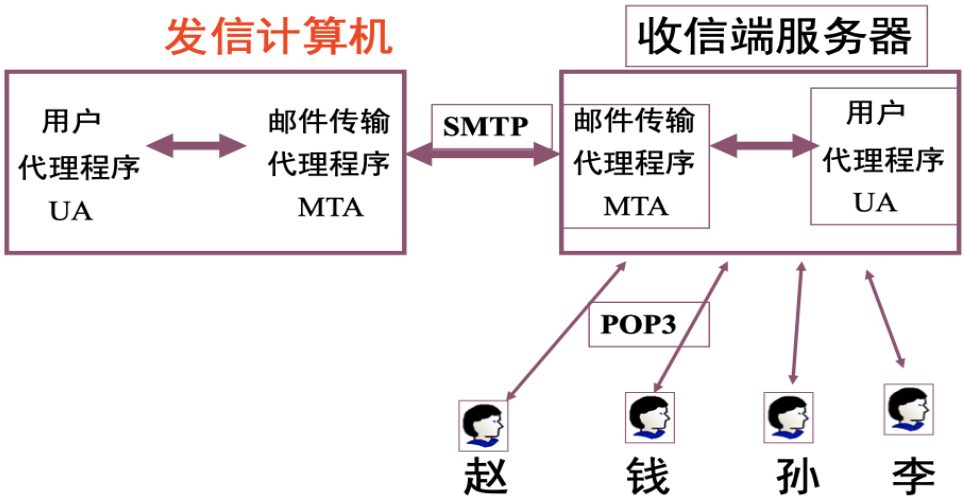
7.4.3 SMTP

- SMTP (Simple Mail Transfer Protocol)

    - 源机与目标机 (SMTP守护进程在监听) 的 25 端口建立TCP连接
    - 如果消息不能被投递，则向消息的发送方返回一个错误报告 (包含了不能投递消息的第一部分)
    - 这是一个简单的 ASCII 协议
- SMTP 工作过程
    - 建立连接：在25端口，用TCP协议建立连接
    - 数据交换
        - 客户机等待服务器首先开始通话
        - 服务器首先发送一行文本，给出它自己的标识，并且告诉客户机是否已准备好接收邮件
        - 如果服务器没有准备好，则客户机释放连接，以后再重试
        - 如果服务器愿意接收电子邮件，则客户机申明发信人和收信人
        - 如果服务器确实存在这样的收信人，则服务器指示客户可以发送邮件
        - 客户发送消息，服务器回发确认
    - 连接释放
- 与MIME的工作图示
    -

7.4.5 最后传递过程

- 问题：收件方不可能一天 24h 在线，需要一个邮件收发代理
- POP3
    - 当用户启动邮件阅读器的时候，POP3开始工作
    - 用户呼叫ISP (除非已有一个连接)，然后与MTA在110端口建立TCP连接
    - 一旦连接建立，POP3协议按顺序经历下述三种状态
        - 授权：处理用户登录的过程
        - 事务：用户收取电子邮件，并将邮件标记为删除
        - 更新：将标为删除的电子邮件删除
    -



- IMAP (Internet Message Access Protocol)
  - IMAP 假设所有的电子邮件都永久地保存在服务器上的多个邮箱中
  - IMAP 提供了阅读消息或阅读部分消息的机制
  - IMAP 服务器在143端口监听
  - IMAP 也可以接收外发的邮件
  - IMAP 有更多的命令，更复杂
- WebMail
  - 能上网即可用，无需配置

7.4.6 课程总结

- 电子邮件系统主要包括MTA和UA两部分
  - UA主要负责读写邮件
  - MTA主要负责收发邮件
- 邮件传输协议：SMTP
- 非纯文本邮件：MIME (多用途互联网邮件扩展)
- 最后投递
  - IMAP (早期为POP3)

7.5 万维网 (World Wide Web)

7.5.1 万维网概述

- 万维网 (WWW - World Wide Web)
  - Web是web网页的集合
  - 每个页面包含了指向其他页面的连接 (超级连接)
  - 浏览器 —— 显示阅读web页面的程序
- WWW的组成部分

- 资源
  - web页面，html
- 统一资源定位器：URL
  - 找到资源的地址
- 通信协议HTTP
  - 将远端资源和本地联系起来

- Web的体系结构
  -

7.5.2 统一资源定位符

- 统一资源定位符 (URL - Uniform Resource Locators)
  - Web页面由URL标识
- URL组成
  - 协议：http
  - 页面所在机器的DNS域名 (或IP地址)：www.abcd.com
  - 包含web页面的文件的名字：products.html
  - 例子
    -

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

7.5.3 客户端

- 当用户单击一个超级链接 (URL) 时：
  - 浏览器检查URL (读取浏览器的输入)
  - 浏览器向DNS服务器询问域名的IP地址
  - DNS 返回对应的IP地址
  - 浏览器和Web服务器建立TCP连接 (80端口)
  - 浏览器发送请求，要求获取文件 products.html
  - Web服务器返回被请求的文件
  - TCP连接被释放
  - 浏览器解释显示下载到本地的文件
- 浏览器扩展
  - 目的：解析各类文件，如PDF文件、GIF图标、MPEG视频等
  - 两种方式
    - 插件：代码模块，运行在浏览器的内部。过多插件会让浏览器过度膨胀。
    - 外挂助手程序：独立的程序，浏览器只是把参数传入，不会影响到浏览器本身。

7.5.4 服务端

- 典型Web服务器的操作
  - 接收来自客户的TCP连接
  - 获取所需文件的名字
  - 从本地磁盘上获取文件 (静态文件)
  - 将文件返回给客户
  - 释放TCP连接
- 改进
  - 在内存中维护一个缓存，保存最近用过的n个文件。
  - 多线程服务器
    - TCP移交：TCP端点被传递给处理节点，应答可以直接向客户端发送 (可以越过多线程服务器的前端)
- 高速代理服务器 (Web cache)
  - 万维网高速缓存把最近的一些请求和相应暂存在本地磁盘中。
  - 当与暂时存放的请求相同的新请求到达时，万维网高速缓存就把暂存的响应发送出去，而不需要按URL的地址再去因特网访问该资源。
- Cookie
  - 目的：不同用户登录同一个web服务器，想获得不同的资源，即个性化服务。
  - 特点
    - 一个小于 4 kB的命名串
    - 当客户请求时，web服务器除了应答外，附送一个cookie，存储在客户机磁盘中
    - 客户再访问同一个web服务器时，同时发送cookie
    - 服务器辨识出用户，并得到它关心的一些信息

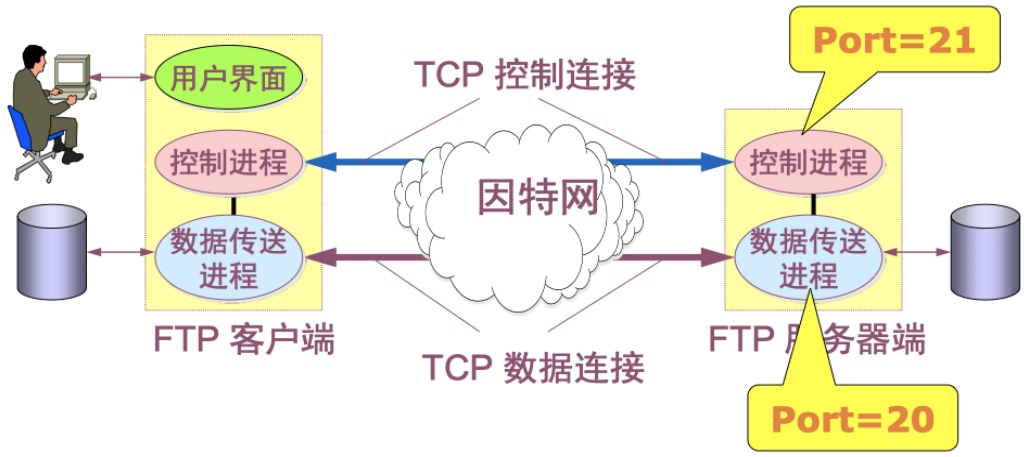
- 缺点：侵犯用户隐私，有安全隐患

7.5.5 课程总结

- 什么是万维网?
- 万维网的主要组成部分有哪些?
- URL是怎样构成的?
- 为什么需要TCP移交？它解决了什么问题?
- 热门网站如何应对众多客户请求?
- 什么是COOKIE?

7.6 其它应用

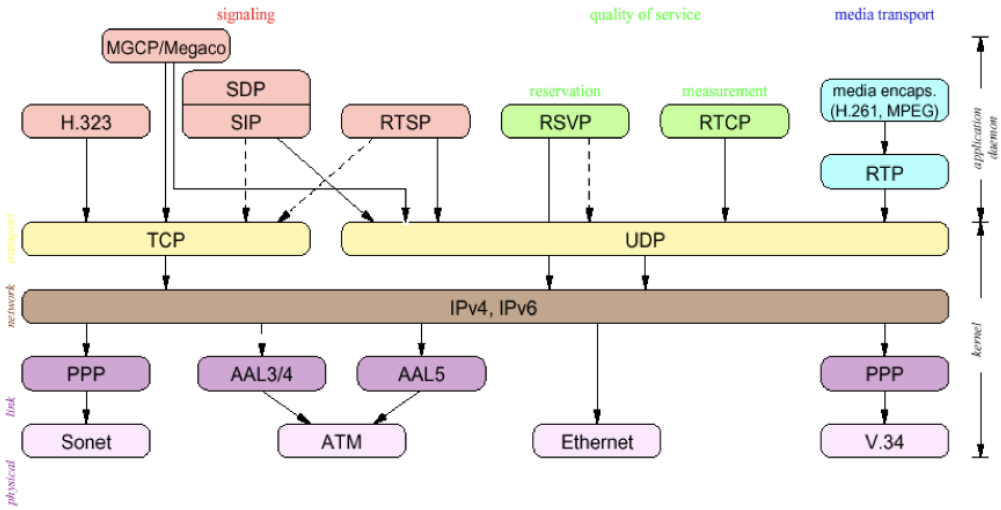
7.6.1 文件传输 (FTP)

- FTP
  - 一种可靠的面向连接的服务，采用TCP在支持FTP的系统间传输文件，支持双向二进制文件和ASCII文件传输。
- TFTP
  - 一种无连接的不可靠的服务，采用UDP在支持TFTP的系统间传输文件。
  - 如：路由器备份下载文件。
- 两个术语
  - 上载：将文件从自己的计算机中拷贝到远程计算机上。(upload)
  - 下载：将文件从远程计算机上拷贝到自己的计算机上。(download)
- 工作原理
  - 双TCP连接，实现简单，提供终止传输、断点续传等服务。
  - 

7.6.2 远程登录 (Telnet)

- 远程登录
  - 不要求远地系统创建众多的服务器，只需为每个远程登录用户建立一个进程，这个进程再通过创建子进程为远程登录用户提供各种允许的服务。
  - 优点：提供与本地登录几乎完全相同的用户界面。
- SSH (Secure Shell)
  - 工作在22端口，为远程登录这样的明文传输协议提供安全的传输壳。

7.6.3 多媒体应用

- 多媒体应用分类
  - 实时应用
  - 非实时应用
- 多媒体应用相关的协议
  - 

7.6.4 课程总结

- FTP有两根TCP连接
  - 数据连接 (20)
  - 控制连接 (21)
- 远程登录TELNET工作在23端口
  - SSH工作在22端口
- FTP和TELNET的传输层都用的是TCP
- 多媒体应用分类：实时、非实时
- 例题

8 多选 (2分) 这个问题探讨了你对不同应用程序要求的服务质量的理解。请选择所有正确的语句。（多选） 得分/总分

- [ ] A. 视频点播 （如 Netflix 流） 比远程登录要求更低分组延迟。
- [x] B. IP 语音比 web 访问要求更低的抖动。 ✔1.00/2.00
- [x] C. 文件共享 （如 BitTorrent 下载） 比电子邮件要求更高的服务质量。 ✔1.00/2.00
- [ ] D. 电话服务 （如 Skype 的音频呼叫） 相比文件共享需要相同的服务质量。

正确答案：B、C 你选对了

解析： A、错误。当视频流开始时，它对延迟不敏感，而远程登录对延迟敏感，因为用户期望进行交互式响应。 B、正确。低抖动 （或延迟的变化） 是在防止缺口上对于VoIP是重要的，而 web 访问对于抖动不是特别敏感。 C、正确。文件共享，相比电子邮件 （它将运行在几乎任何网络）需要更高的带宽。 D、错误。文件共享希望高带宽和延迟不敏感。电话想要低延迟/抖动，需要小的带宽。