**考试：**

* 生成树的相关考点
* RIP
* RPF（汇集树的求解）
* IP地址判断
* CRC校验
* 流量控制
* TCP滑动窗口
* 邮件使用协议（传输：SMTP，接收：POP3）
* 输入浏览器地址栏后的一系列操作

**Chapter 01：Introduction**

**-----------------------------------------------------------**

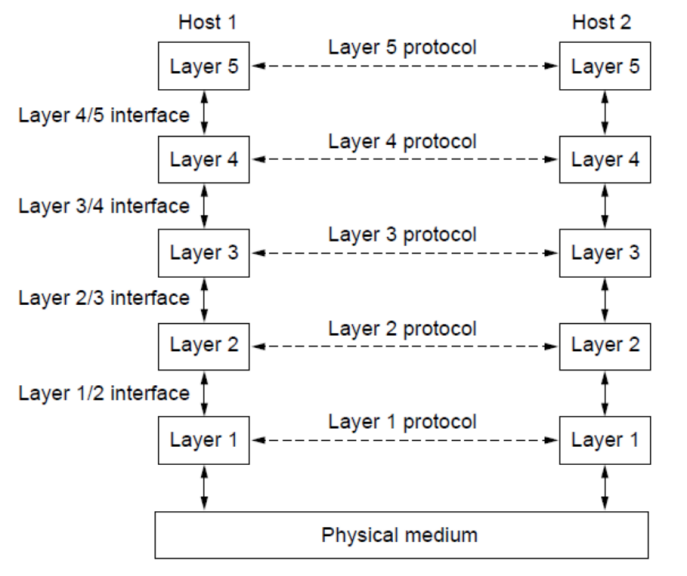
**计算机网络应用：**

* 商业应用：分客户端和服务端
* 家庭应用：资源共享，不分角色
  + B2C：线上购物
  + B2B：制造商向原料供应商订货
  + G2C：纳税系统
  + C2C：跳蚤市场
  + P2P：线上分享内容
* 移动用户：使用移动设备参与
  + 有线、非移动：办公室中的台式机
  + 有线、移动：酒店房间配备的笔记本电脑
  + 无线、非移动：未铺设线路建筑中的网络
  + 无线、移动：掌上电脑
* 社会应用：在社会事务中应当承担的角色
  + 网络中立性
  + 版权问题
  + 分析用户隐私问题
  + 网络诈骗

**计算机网络硬件：**

* 规模（从小到大）：
  + PAN：个人局域网（蓝牙、热点）
  + LAN：局域网（学校机房）
  + MAN：城域网（有线电视）
  + WAN：宽域网（跨大洲、大国网络）
  + Internet：全球网
* 具体部分：
  + User（以PC为例）：网卡->无线/有线
  + ISP（服务供应商，如三大运营商）：信道、交换机、路由器
  + ICP（内容供应商，如游戏公司）：服务器、app

**计算机网络软件：**

* 通信模型：
  + Tx：源端
  + Rx：目的端
  + PHY：信道，又称中间介质。会受到干扰，称为噪声
* 通信协议：
  + 基本结构：
  + 协议：实体对等层的约定或规范的集合。其规定不受其他层次协议的影响，又称协议分层的独立性
  + 层次接口：低层次为高层次服务，而高层次的服务会调用低层次的服务。由于协调上下层次的服务，需要有明确界定
  + 结构优点：降低耦合性，在工程上，协议的升级或替换不需要更改整个系统
  + 信息处理：底层到高层称为解包（unpacket），高层到底层称为包装（packet）
* 通信特征：
  + 分割运输（解决传输带宽不足）
  + 给每个分割部分贴标签（解决信息分割后无法重新有序排列）
  + 协议规定标签的大小（解决标签占位）
  + 协议效率：包中数据大小和包总大小的比值（1B标签，4B数据，效率为80%），可用来评估协议
  + packet方法：
    - 若分包，由于分层独立性，上层给的标签在分解后的包中只能存在一份
    - 在上一点操作完后，给每一个包添加本层标签
    - 若下一层是PHY层，增加尾部信息
    - 传给下一层
  + unpacket方法：
    - 去除所有包的本层标签，若从PHY层传来，去掉尾部信息
    - 若多个包有一个上层标签，则将多个包合成为一个包（信息拼接，上层标签保留一个）
    - 传给上一层
  + 尾部信息：根据包内数据和特定数学规则生成，用作校验
* 通信服务
  + 面向连接服务
    - 可靠报文流：页序列（报纸的电传）
    - 可靠字节流：下载电影
    - 不可靠连接：打IP电话
  + 无连接服务
    - 不可靠数据包：垃圾邮件
    - 认可的数据包：发短信
    - 请求回应：数据库查询（银行余额柜员机查询）
* 通信过程：六个原语
  + LISTEN：等待连接（一般为服务器的状态）
  + CONNECT：建立连接
  + ACCEPT：接受对方的链接
  + RECEIVE：等待对方报文
  + SEND：给对方发报文
  + DISCONNECT：断开连接
* 通信方式
  + 双工：双方都可收发
  + 半双工：同一时间内只能收/发
  + 单工：总是只能收/发

**计算机网络软件参考模型：OSI(Open System Internet)**

* 性质：学院派产品，没有物理实现
* 特点：层级优化（跨越边界的信息流应尽量简化）
* 层级：
  + 底层由**运营商**构建，又称**通信子网**
  + 从传输层开始，目的地由**用户**给出
  + 每一层的交换单元（层处理数据的最小单位不同），网络层往上的**PDU：协议数据单位（Protocol Data Unit）**
  + **物理层----Bit----：定义了传输的物理实现**
  + **数据链路层----Frame(KB)----：点对点传输时，不对数据具体传输做要求，只做抽象的要求**
  + **网络层----Packet(Group of frames)----：实现点对点的传输**
  + **传输层----TPDU----：确保数据正确地传输到另一端（目的地不变）**
  + **会话层----SPDU----：服务端和客户端相同但需要不同数据流（微信聊天窗口）时使用**
  + **表示层----PPDU----：不同计算机数据的内部表示不同，为了能进行通信，需要通过一个抽象定义的数据结构（标准）进行数据表示**
  + **应用层----APDU----：用户协议（HTTP）**

**计算机网络软件参考模型：TCP/IP**

* 性质：正在成熟使用的一种实现
* 特点：简化层级，无物理层、会话层和表示层（提出了host-networks、对数据链路层进行了NCP/LCP等适配......）
* 层级：
  + **链路层**
  + **互联网层**
  + **传输层**
  + **应用层**

**Chapter 02：The Physical Layer**

**-----------------------------------------------------------**

**数据通信的理论基础：**

* 傅里叶分析：将周期信号分解为三角函数
* 带宽有限的信号：
  + 波特率：每秒钟传送波形的数量
  + 比特率：每秒钟传送bit的数量
  + 设波形种类为N，一个波形传送的bit量：。因此，**当波形只有两种（高点平、低电平）时，b=1，波特率=比特率**
* 信道的最大数据速率：
  + 尼奎斯特定理：带宽为B，只要每秒2B次采样即可完全重构信号
  + 信噪比：信号功率（S）与噪声功率（N）的比值
  + 香农公式：带宽为B(Hz)，信噪比为S/N的信道，最大数据速率（容量）为

|  |  |  |  |
| --- | --- | --- | --- |
| bps | T(ms) | 第一个频率f | 叠加谐波数 |
| 300 | 2/75 | 37.5 | 80 |
| 600 | 1/75 | 75 | 40 |
| 1200 | 1/150 | 150 | 20 |
| 2400 | 1/300 | 300 | 10 |
| 4800 | 1/600 | 600 | 5 |
| 9600 | 1/1200 | 1200 | 2 |
| 19200 | 1/2400 | 2400 | 1 |
| 38400 | 1/4800 | 4800 | 0 |

* 从数据率与频率关系可知，截止频率一定，数据率越高反而导致叠加谐波数减少甚至到0，会导致信息丢失。所以数据传输速率实际上是一定的

**引导性传输介质：**

* 磁介质：光盘、磁带
* 双绞线
* 同轴电缆
* 电力线
* 光纤（多根构成光缆）

**无线传输：**

* 电磁频谱
  + WiFi：2.5G/5G
  + BT：2.5G
* ISM频段（Industry, Science, Medical）：工业科学医学频段，其中2.4GHz频段免费使用

**通信卫星：**

|  |  |  |  |
| --- | --- | --- | --- |
| 高度(km) | 类型 | 延迟(ms) | 全覆盖数量 |
| 35000 | GEO | 270 | 3 |
| 20000-35000 | 空区域 | | |
| 15000-20000 | 上范艾伦辐射带 | | |
| 5000-15000 | MEO | 35-85 | 10 |
| LOW-5000 | 下范艾伦辐射带 | | |
| LOW | LEO | 1-7 | 50 |

* 范艾伦辐射带：带电粒子层，卫星在此处运行会很快毁坏
* round-trip：往返，在这样一个过程中总延迟是上表各延迟的两倍

**数字调制与多路复用：**

* 数字调制：比特与代表他们的信号之间的转换
  + 基带传输：信号占有传输介质上从0到最大值之间的全部频段
  + 通带传输：信号占据以载波信号频率为中心的一段频带
    - 幅移键控（ASK）：1振幅不为0
    - 频移键控（FSK）：1频率较高
    - 相移键控（PSK）：1波形偏移180°
* 多路复用：信道被多个信号共享
  + 频分复用（FDM）：多个信道分一个较大的频带，各自有一个较小的频带进行传输（之间有保护带（guard band），使信道之间完全隔离）
  + 时分复用（TDM）：多个信道分一段较长的时间，各自有一个较短的时间进行传输（之间有保护时间（guard time），为适应时钟的微小变化）
  + 码分复用（CDM）：

**公共电话交换网络：**

* 电话系统结构：
  + 全连通：两两连接
  + 中心交换：一个中心
  + 两层体系：两级中心
  + 基本贝尔系统：电话->端局->长途局->中间交换局->~（越靠近中心线路越“粗”）
* 本地回路：
  + 调制解调器（猫、modem）：
    - 调制：数字信号->模拟信号
    - 解调：模拟信号->数字信号
  + 编码解码器：
  + 数字用户线：
    - 线距离增大，带宽明显减小
    - ADSL：256个4kHz带宽的信道，保留一个语音信道，小部分低频率为上行流，大部分高频率为下行流（因为用户下载多上传少）

**移动电话系统：**

* 1G：模拟语音（将物理区域划分为蜂窝，相邻蜂窝不能重用频率。为了增加用户可以将蜂窝划小）
* 2G：数字语音
* 3G：数字语音和数据（使用基站，实现软切换）

**有线电视：**

* 共用天线电视：一个天线，一根同轴电缆，接头引出多个分支电缆分到各户
* 线缆上的Internet：使用光纤接到光纤节点，引出同轴电缆到若干户

**Chapter 03：The Data Link Layer**

**-----------------------------------------------------------**

**该层解决问题：物理连接点的逻辑抽象**

**数据链路层的设计问题：**

* 提供给网络层的服务：
  + 接收网络层传来的比特，并逻辑上沿虚拟数据路径完成传输（实际上会经过物理层）
  + 种类：
    - 无确认、无连接：不需要连接，开销很小，可靠性低。适合错误率很低或者实时通信的场景
    - 有确认、无连接：确认每一帧是否正确到达
    - 有确认、有连接：每一帧被编号，确保每一帧都按正确顺序收到一次
    - **无确认、有连接：不存在，可能是连接已经实现了确认**
* **成帧：让接收方能在比特流中识别帧**
  + 字节计数法：第一个字节说明随后的ｎ个字节在同一帧中（如果有一个字节出错，后续全部错误）
  + 字节填充的标志字节法：用特殊字符（标志字节）标志帧的开始和结束（但图片、音频等数据也可能出现特殊字符，因此在真正的特殊字符前加入转义字符ESC解决）
  + 比特填充的标志比特法：连续５个１后插入０，对应地，接收方在接受５个１后舍去后面的那个０（填充后数据很大程度取决于原始数据，若连续的１多填充就多）
  + 物理层编码违禁法：走物理层捷径
* 差错控制：检错、纠错
* 流量控制：

**差错检测和纠正：**

* 纠错码：让接收方可以推断被发送的数据是什么
* 检错码：让接收方知道是否发生了错误，却不知道哪里发生了错误

**基本数据链路层协议：**

P169的头文件定义（阅读）

* 帧种类
  + 信息帧（I）
  + 管理数据帧（M）
  + 控制帧（C）
* 乌托邦式单工协议
  + 设定：
    - 单向传输
    - 信噪比->∞
    - 处理时间->0
    - 缓存空间->∞
    - 发送方一直在向网络中发送信息
* 无错信道上的单工停-等式协议**（其实应具备双向传输能力，至少是半双工）**：
  + 设定：
    - 单向传输
    - 信噪比->无穷
  + 改进：
    - 在接收方传数据到网络层后，回传一个哑帧给发送方
    - 发送方在发送一帧数据后，只有在接到接收方回传的哑帧时，才继续发下一帧
* 自动重复请求协议
  + 设定：
    - 发送方把一帧数据的发送放在循环中，若回传的状态正确：
      * 序列号正确，则序列号自增并继续发下一帧
      * 序列号错误，则接收方可能把这边重传过去帧的确认信息发回了，不做操作，重新发送该帧
      * 如果状态错误则不做操作，重新发送该帧
    - 接收方把一帧数据的发送放在循环中，若回传的状态正确：
      * 序列号正确，则自增预期的下一帧序列号，并把帧传到网络层
      * 序列号错误，发送方可能存在重传现象，预期序列号不变
      * 无论如何，之后会把当前帧已收到的信息回传给发送方
  + 原理：发送方的数据和接收方的回应都有可能因为噪声而丢失，因此需要在重传的基础上检查序列号

**滑动窗口协议：**

* 捎带确认：暂时延缓确认，把确认信息搭载在下一个发给发送方的数据帧上
* 1位协议：（类似双方对等的自动重复请求协议）
* 回退N协议（管道化传输时错误处理）：
  + 链路利用问题：
    - 问题：实际上，帧发送到接收方以及确认帧的返回需要时间，一旦发送方窗口较小，就会在填满信道之前阻塞，降低链路的利用率
    - 解决：
      * 首先放宽限制，设发送方阻塞之前可以发w个帧
      * 设某个信道的单向传输延迟为，发送方发一帧耗时，**接收方发送确认帧不需要时间，且确认帧长度为0**。要在第一个确认帧回到接收方之前，接收方尽可能多发送w个帧，则
      * 在带宽为B，单向传输延迟为D，帧大小为F的一个信道中，显然，代入得
      * 由于理想状态，实际的w应该更大。对于一个实际发送方的W，**当它较小且产生阻塞时**，链路使用率p满足：
  + 回退n（适用接收窗口为1的情况）：
    - 接收方：简单丢弃所有到达的后续帧，且不返回确认，等到下一次预期帧来到的时候再接收并发送确认信息
    - 发送方：下次滑动窗口后，从未确认的第一帧开始发送
  + 选择重传（适用接收窗口大于1的情况）：
    - 接收方：只丢弃坏帧，继续接收后续的正常帧（存入缓冲区），除了在后续第一个正常帧发送否定确认坏帧（NAK(x)）信息外，其余确认信息都不发送。直到重发的帧到达，将缓冲区原有的帧与重发帧排好序后交给网络层，并直接发送最后一帧的确认信息，一次性确认了被缓冲的所有块的接收
    - 发送方：超时的时候，接收到NAK(x)，重发帧x。重发之后继续按原顺序发送

**Chapter 04：Medium Access Control Sublayer**

**-----------------------------------------------------------**

**信道分配问题：**

* 静态信道分配
* **动态信道分配假设：**
  + **流量独立：**由独立的若干个站（如计算机、电话）组成
  + **单信道：**所有通信都只用这一个信道
  + **冲突可观察：**两个帧的传输在时间上重叠，发生冲突。所有的站都能检测到冲突
  + **时间连续或分槽：**
    - 时间是连续的，任何时刻都能发送帧
    - 时间分成离散的间隔，帧的传输只能从某个槽的起始点开始
  + **载波侦听或不听：**
    - 一个站在使用信道前就侦听，判断信道是否在使用
    - 盲目传输，之后判断本次传输是否成功

**多路访问协议：**

* **ALOHA：**
  + **纯ALOHA：**时间连续，相互独立发送
  + **分槽ALOHA：**时间分割为槽，每槽发送一帧。由一个特殊站在每个间隔开始发送脉冲信号
* **载波侦听多路访问协议：**
  + **坚持CSMA：**侦听信道忙时，继续等直到信道空闲才发送
    - **1-坚持CSMA：**发现信道空闲时，立刻发送帧（多个站同时等待时会有冲突）
    - **p-坚持CSMA：**分槽信道适用。发现信道空闲时，p概率发送，1-p概率推迟到下个时间槽
  + **非坚持CSMA：**侦听信道忙时，不坚持侦听，而是随机等待一段时间再次侦听
  + **带冲突检测的CSMA：**
    - 思想：每个站发送时将自己正在发送这个信号公之于众，其他站发送时检测到冲突时立即中止数据传输。
    - 原理：设两个站信号传输最久需要，某个站只有在发送后还没有发现冲突，就可以认为自己“抓住”了信道，继续传输；否则内若发现冲突也要停止

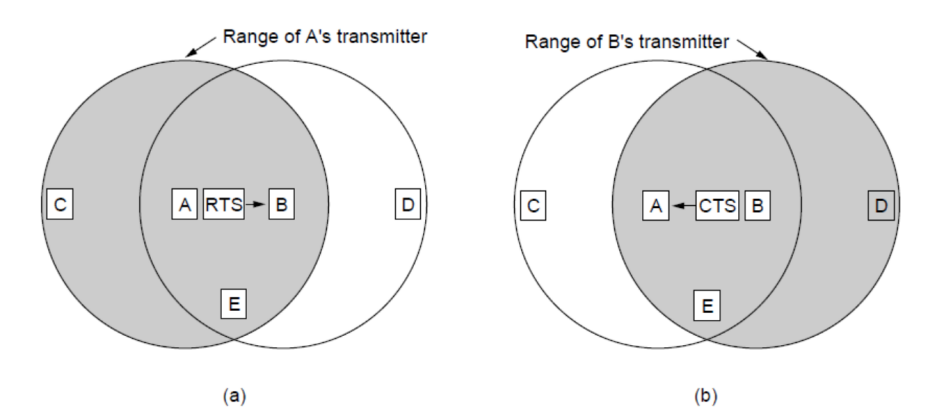
**无冲突协议：**

* **位图协议：**位图由N位组成，每次发送前若站x需要发送帧，则将1发送到位图的第x位。遍历所有站后，位图遍历每一位，是1的就给对应的站发送帧。
* **令牌传递：**站按照环形连接，一个令牌沿着某个方向在环上传输。令牌是发送权限，对于拿到令牌的站，若有信息要发则直接发送；若无信息则不做操作。之后将令牌沿着环方向传给下一个站。
* **二进制倒计数：**每个站按照固定长度的二进制数表示，每个位时间内进行一轮竞争。在第N轮内，首先将所有参与竞争站号的第N位进行或运算，结果为0则全部进入第N+1轮；结果为1则第N位为0的站放弃竞争，为1的进入第N+1轮。直到剩下一个站，该站进行传输（总结：号最高的站进行传输）

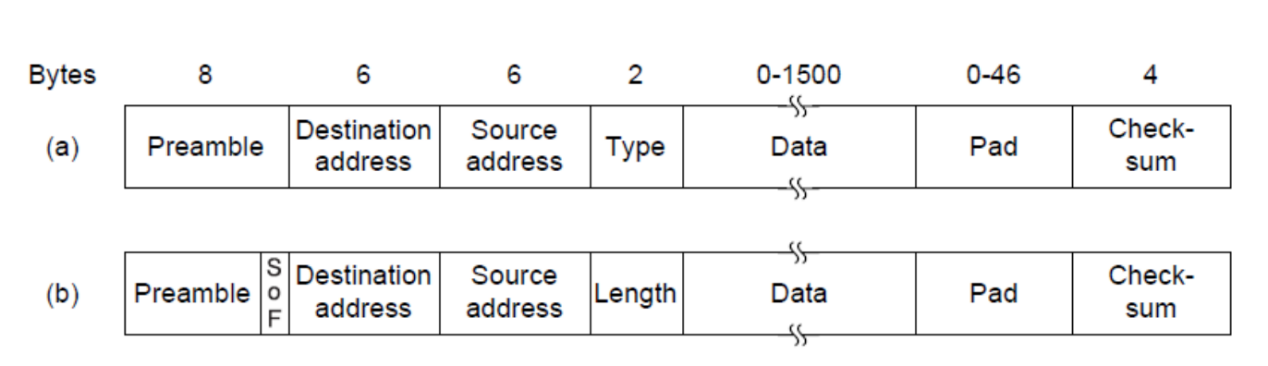
**有限竞争协议：**

* 自适应树遍历协议：
  + 灵感：二战中美军测试士兵是否感染梅毒算法，N个士兵的血样混合检测，若呈阳性，则平分为两份，再次检测。直到找出感染士兵为止
  + 实现：0号竞争槽中，若N个站中产生冲突，先给左子树的站分配，左子树的传完后的第1个槽分配给右子树，直到所有站都发送了消息

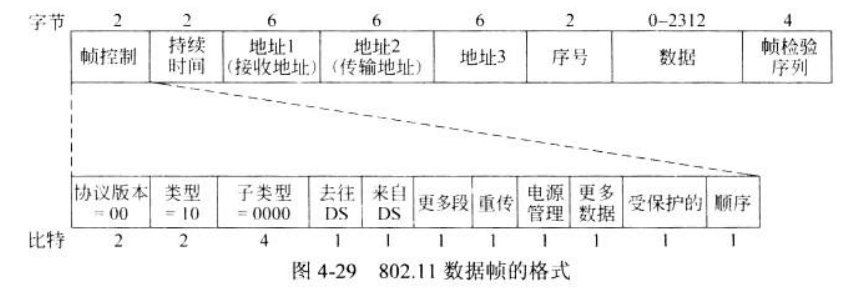
**无线局域网协议：**

* 复杂性：无线传输信号衰减的很厉害（只能使用事后确认）、无线传输范围有限
* **无线CSMA局限性（A-B-C-D）：**
  + **A、C同时发给B：**A已经在发送给B了，C不应当发，但由于无法侦听到A的发送信号，因此C仍然认为信道空闲，会发送信息**（无法发现潜在的竞争者——隐藏终端问题）**
  + **B、C同时发给A、D：**两者发送并不冲突，但由于B、C互相能侦听到对方的发送信号，因此都认为信道忙，不会发送信息**（因为暴露在其他站的发送信号范围内而误认为信道忙——暴露终端问题）**
  + **总结：**没有考虑无线传输的空间特性，这是有线传输所不考虑的问题
* **解决：MACA协议（多路访问冲突避免协议）**
  + 
  + 当A要发送信息给B时，发送一个RTS(Request To Send)给B
    - RTS内容：某时某刻，给B发送数据
    - 效果：此时C、E都知道A要发信息了，不做操作
  + 当B收到A的RTS后，回传一个CTS(Clear To Send)给A
    - CTS内容：重复RTS的内容
    - 效果：此时D、E都知道B要收信息了，不做操作
  + 一来一回，在A、B信号范围内，只有A、B可以进行收发，其他站会“回避”

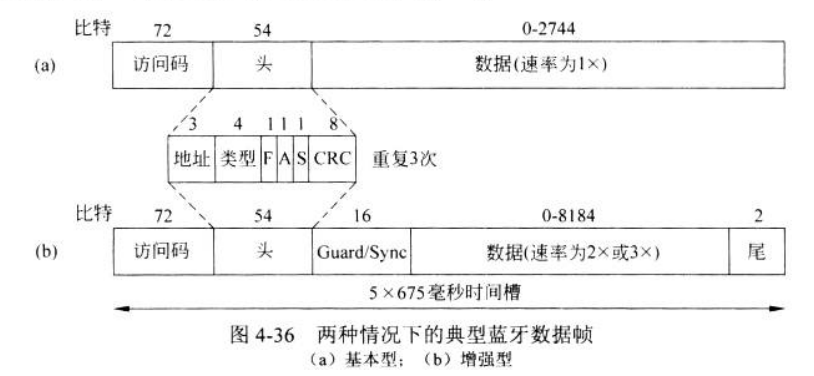
**以太网：**

* **经典以太网：10Mbps**
  + MAC子层协议（帧格式：(a)为DIX，(b)为IEEE 802.3）：
    - 
      * **SoF：**帧起始定界符
      * **Type：**告诉接收方帧内容是什么，接收方根据其值调用不同的网络层协议处理数据包
      * **Length：**数据帧长度。由于DIX中Type值都大于1500，而0x600=1536，因此小于等于0x600时可解读为Length，大于0x600可解读为Type，巧妙地让两种协议共存了
      * **Data：**长度为1500，理由如上
      * **Pad：**填充
      * **Checksum：**校验和
    - 最小帧长限制：
      * **发送时间：**从开始发送到完整发送一帧的用时
      * **传播延迟：**数据从一个站到另一个站的用时
      * 最小帧长原因：最远两个站传播延迟为，则冲突检测需要。至少得在发送后确认没有冲突，才是真正的成功。若帧太短，内帧就发送完毕，发送方会误以为帧发送成功，无法检测到冲突（这也是填充位的由来）
* **交换式以太网：**
  + 集线器：将所有线缆焊在一起（逻辑上等同于单根电缆的经典以太网，无法增加容量）
  + 交换机：有开关矩阵，通过CPU控制开关，把进入交换机的帧传到想输出的端口
* **快速以太网：双绞线、光纤**
* 千兆以太网：交换机、集线器
* 万兆以太网

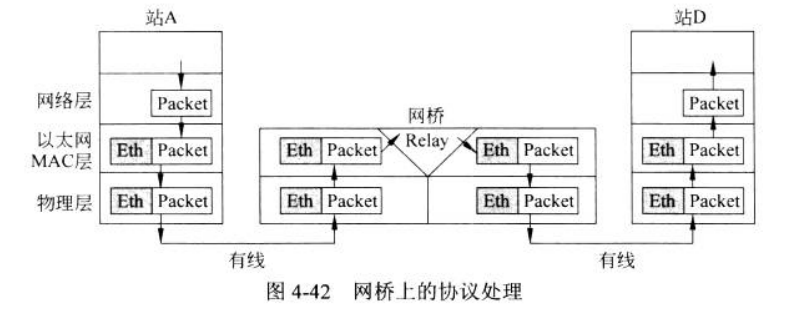
**无线局域网：**

* 802.11体系结构：
  + 有架构模式：计算机逻辑上分布在多个集合中，集合间通过路由器进行通信
  + 自组织模式：各个计算机可互相访问
* 802.11 MAC子层协议：
  + 使用带有CSMA/CA的机制发送帧
  + 需要有接收方的确认（与经典以太网/802.3的主要区别）
  + 有后退，即各个冲突方强制等待一段时间，有助于避免冲突（与经典以太网/802.3的主要区别）
  + RTS/CTS机制使用NAV（网络分配向量）：帧携带NAV字段，指出帧所属一系列数据需要传输多长时间，其余侦听到NAV的站明白该段时间内信道一定是繁忙的，便在这段时间内沉默不语。这可以防止隐藏终端也同时发送数据
* 802.11帧结构：
  + 

**蓝牙：**

* 配对：设备彼此发现并连接
* 体系结构：
  + 微网：1个主节点和≤7个从节点
  + 桥接从节点：可以加入多个微网，微网间通过其通信
  + 散网：由多个微网桥接而成
* 数据帧：
  + 
  + CRC在数据前面：若一些较为重要的帧，接收方发现

**数据链路层交换：**

* 学习网桥：
  + 交换机有一张端口到地址的映射表，但一开始是空的，交换机并不知道端口到地址的映射
  + 不过数据在交换机上转发时，交换机知道包的目的地，只是不懂要从哪个端口发出去，因此它会在所有端口上“喊一嗓子”，询问各端口的地址，收到回复后，将包从正确的端口转发出去
  + 经过学习后，映射表得到更新，若以后继续转发时，目的地在映射表中则直接转发，目的地不在映射表中，则进行一次学习更新映射表
  + 协议处理：
    - 
    - 说明交换机是数据链路层设备
* 生成树网桥：
  + 引入：网桥间有两条平行的链路，一旦一个结点要发送数据到一个不存在的地址，由于交换机无法判断平行链路上的两个帧是同一个帧，会当作两个不同帧处理，那么交换机的学习会无休止进行下去
  + 解决：使用生成树连接各个网桥，各个网桥关闭不属于生成树那些链路的端口（保证没有环路），并一致认定MAC地址最小的交换机所在网桥为根桥（保证正确性）
* 设备层次：
  + 应用层：应用网关
  + 传输层：传输网关
  + 网络层：路由器
  + 数据链路层：网桥、交换机
  + 物理层：中继器、集线器
* 虚拟局域网（VLAN）：
  + 网桥的每个端口被标记上颜色（一个端口可以是多种颜色），在来自某个颜色的报文到达后，只能被转发到颜色相同的端口
  + IEEE 802.1Q：
    - 新增了VLAN协议字段和标签字段
    - 在802.3的Length/Type位置为0x8100（表示VLAN协议），由于大于1500，它会被解析为Type，不会转发给传统网卡
    - 新增标签字段：
      * 优先级（3位）：让交换设备识别是硬实时流量还是软实时流量，以提供更好的网络服务
      * CFI（1位）
      * VLAN标识符（12位）：标识报文来自于哪种“颜色”

**Chapter 05：Network Layer**

**-----------------------------------------------------------**

**网络层的设计问题：**

* 提供给传输层的服务：
  + 服务与路由器技术无关
  + 向传输层屏蔽路由器数量、类型和拓扑关系
  + 传输层可用的网络地址有统一编址方案，跨越LAN和WAN（此时两层有耦合，但应用广泛，因此依然被接受了）
* 无连接服务的实现：
  + 路由表记录目的路由器和下一站转发目的路由器的映射，会随着网络拓扑变化或链路状态变化而实时更新
* 面向连接服务的实现：
  + 在终端建立连接后，约定该连接有一个标识符（如数字1、2），叫连接号
  + 对每一个连接，在经过的每个路由器上都有入境和出境的信息
  + 路由表会记录对于每个连接号入境端和出境端的映射
  + 作为网关的路由器，可能入境的多个端都对应到同一个连接号，因为该路由器很容易识别出信息从哪个终端发送**（但非网关路由器无法识别，因此入境中各端对应的连接号必须不同！）**
* 两种服务比较：

|  |  |  |
| --- | --- | --- |
| **问题** | **数据报网络（无连接）** | **虚电路网络（有连接）** |
| 电路建立 | 不需要 | 需要 |
| 寻址 | 包有全部的源和目的地址 | 包只有简短的VC号 |
| 状态信息 | 不保存连接状态 | 每条VC需要存储状态 |
| 路由方式 | 每个数据包单独路由 | 建立VC时选择路由，后续不变 |
| 路由器失效影响 | 无影响（崩溃期间直接丢失的包除外） | 穿过故障路由器的VC全部中断 |
| 服务质量控制 | 困难（因为还关系到网络设备） | 容易（如果在建立VC时有足够资源可分配） |
| 拥塞控制 | 困难（因为没有资源的预先分配） |

**路由算法：**

* 职责：确定一个入境数据包应该被发送到哪一条输出线路上
* 解决问题：
  + 公平性：在一个链路通信时，其余与该链路有重合的链路不应当通信，不干扰该链路的通信
  + 有效性：为了网络总体效率，那些“干扰别人多”的链路应当少通信
  + 这两个性质是冲突的
* 特性：正确性、简单性、鲁棒性、稳定性、公平性、有效性
* 类型：
  + 非自适应算法（静态路由）：不会根据当前测量或估计的流量和拓扑结构来调整路由决策
  + 自适应算法（动态路由）：改变路由决策以反映拓扑结构或流量的变化情况
* **实例：**
  + **最优化原则（总论）：**若J在I和K的最优路径上，则J到K的最优路径正好在I到K的最优路径上
  + **最短路径算法：**Dijkstra算法，但权重有可能是包括距离、带宽、平均流量、通信成本、平均延迟等因素在内的一个值
  + **距离矢量算法（RIP）：**
    - 概念：每个路由器维护一张表，存储当前已知的到每个目标的最佳距离以及下一跳的路由器。该路由器通过与其邻居不断进行信息交换而刷新表
    - 算法实现：
      * 每个路由器给它的每个邻居发送列表（所有其他路由器到它的最短时间），这种情况下，它也能接收到自己每一个邻居发来的列表
      * 对于某个路由器，它接收了邻居的表（其中表示i到X的最短时间），且知道到邻居的延迟，显然就明白了从出境，经过可以到达，因此遍历所有的就可以知道，从哪一个邻居出境时可以最快到达
      * 显然，在某个停机时，会马上感知到并设置对应
      * 更新自己的路由表，到的那一项中，出境路由器更新为，用时为
    - 算法评估：
      * 网络收敛：整个网络最佳路径的寻找过程
      * 该算法由于无穷计算问题，收敛不会结束。核心原因：X告诉Y说有一条到Z的路径，X无法得知自己在不在这条路径上，而如果在就会形成闭环，发生无穷计算问题
  + **链路状态路由（OSPF）：**
    - 算法划分：
      * 发现邻居、认识邻居
      * 设置到各个邻居的距离或成本
      * 把你知道的邻居信息打包
      * 将包发给所有邻居（同时接收邻居发来的包）
      * 计算最短路径
    - 算法实现：
      * 发现邻居：从每个端口发送HELLO数据包，另一端需返回说自己是谁（若有多个路由器在一个广播LAN中，LAN抽象为一个连接到各路由器的节点）
      * 设置链路成本：是多种因素影响的综合指标（网络运营商会提供度量算法）
      * 构建链路状态包：
        + 本路由器到所有邻居的代价
        + 序号：表示信息的新鲜程度，供接收方更新
        + 年龄：表示数据包的存在时间

背景：路由器可能在崩溃后重启，从0开始发，由于序号太小被误以为是过时信息而被拒收；数据包在信道中受到干扰错位，i号变为j号且j > i，后续发送的i+1~j-1号包由于序号太小被误以为是过时信息而被拒收）

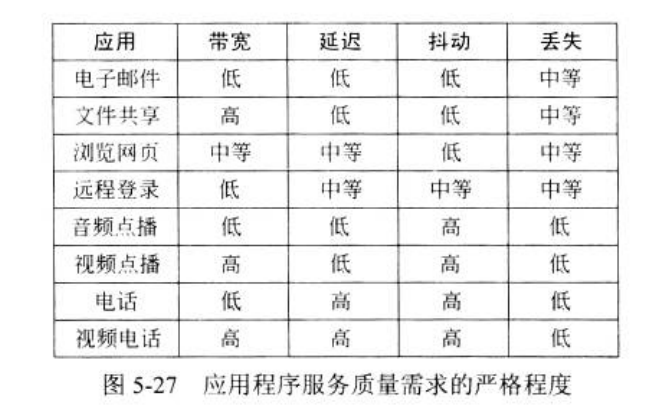
解决：由计算机网络思想，可以不立即解决，只要过一段时间后网络状态回到正常就行。因此设置年龄，设定一定条件使其持续减少，到0时失效。这样那些“错误”的包一定时间后会失效，后续包就可以顺利接收了

* + - 分发链路状态包：
      * 方法：泛洪法
      * 缓冲区：
        + 包信息外
        + 发送标志：要向哪个邻居转发
        + 确认标志：要向哪个邻居确认
        + 特点：对于直接邻居D，向D以外的其余邻居转发，向D确认；对于间接邻居U，若它转发时经过多个直接邻居D**x**，则向D**x**以外的其余邻居转发，向这些D**x**确认（不难发现，发送标志和确认标志对于同一个D，一个是1，另一个就一定是0）
    - 计算新路由：可以参考之前的RIP
  + 层次路由：分成若干个区域，每个路由器不必知道其他区域的拓扑结构，只需知道如何去到自己所在区域的其他节点以及从哪里去到其他区域
  + 广播路由
    - 根据网络各边权值构建最小生成树
    - 从最小生成树的每一个点广播出去，可以得到一棵汇集树
  + 组播路由
  + 选播路由
  + 移动主机路由
  + 自组织网络路由（类似BFS）

**拥塞控制算法：**

* 途径：各种解决方案，要么预先避免拥塞（预防性），要么一旦发生拥塞随之作出反应（反应性），反应速度由慢到快：
  + 网络供给：提升带宽
  + 流量感知路由：感知到某条路情况不好，换一条路
  + 准入控制：拒绝新连接的建立
  + 流量限制：限制某条线路的流量
  + 负载脱落：暂时停止某条忙线

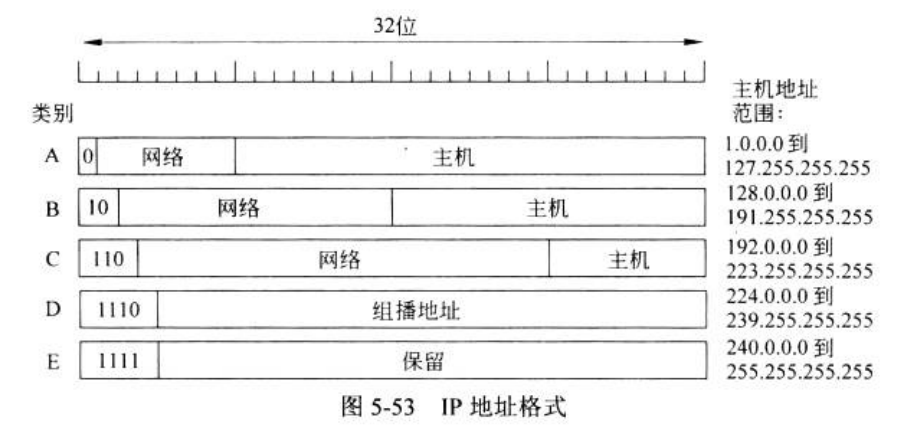
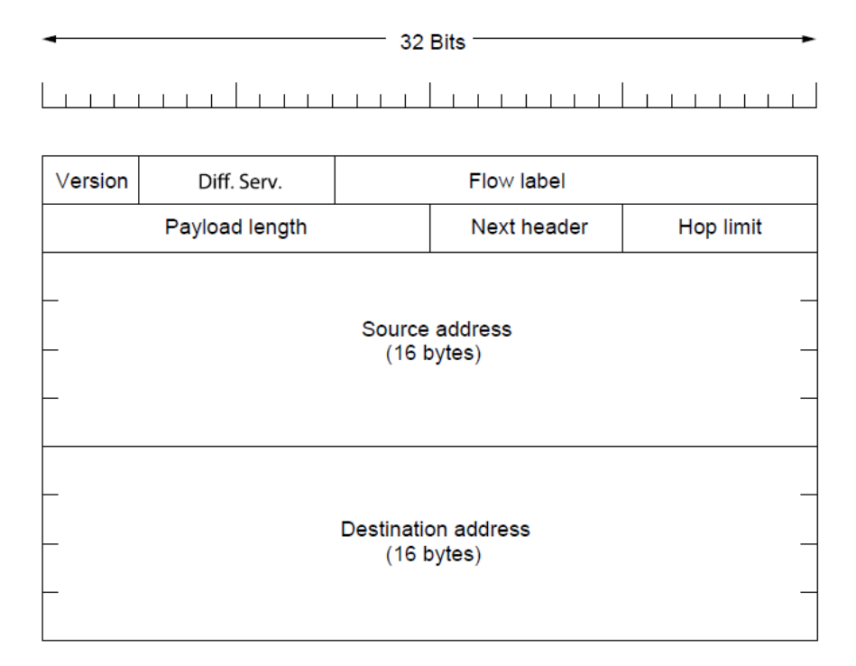
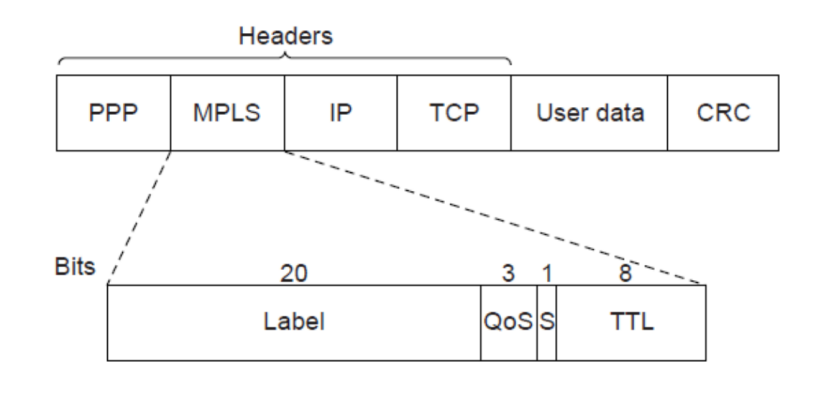
**服务质量：**

* 应用需求：源端到接收端的一些参数，决定服务质量
  + 带宽：传输的性能
  + 延迟：即传输延迟
  + 抖动：数据到达速率的浮动
  + 丢失：丢包情况
  + 
  + 服务质量类型：
    - 恒定比特：电话
    - 实时可变比特率：压缩视频会议
    - 非实时可变比特率：点播电影
    - 可用比特率：文件传输
* 流量整形：
  + 含义：允许客户端发送任意流量
  + 方式：
    - 漏桶：桶以固定速率漏水，一旦满了则水会沿桶壁流失
    - 令牌桶：将水另称为令牌，要往桶里加水，得先取出一些令牌让桶有空间
* 包调度：
  + 节省资源：带宽、缓冲区、CPU周期
  + 算法：
    - 先来先服务
    - 轮询：遍历各个输入线路，轮流给线路发一个包
    - 加权公平

**网络互联：**

* 含义：实际上不同协议的网络会连在一起，在提供服务、寻址、广播等方面处理不同
* 
* 网络连接：设计一种IP格式，任何网络都许可，在经过网络边界路由器时，更换报头后转发
* 隧道：源端和目的端所在网络完全相同，但中间可能经过不同网络。解决方案是直接将源端的包完全当作数据封装到中间网络的包中直接传输，不用与中间网络打交道而直接到达目的网络
* 数据包分段：
  + 透明分段：经过一个网络时分段，在进入下一网络前先整合，然后再次分段通过（重组很多次，开销很大）
  + 非透明分段：第一次分段后，每一段都当成原始数据包，不再整合，直到目的地（网络传输开销大、一些报头是无用的、丢包概率增加）
  + MTU发现：先不允许分段，尝试直接发送，在路由器发现该包太大时，返回一个错误数据包，它指示了该路由器能接收的大小，然后丢弃该包。源端收到报错信息后再分段重新发送，收到报错就再次分段，直到发送成功。这样能保证段分得最少

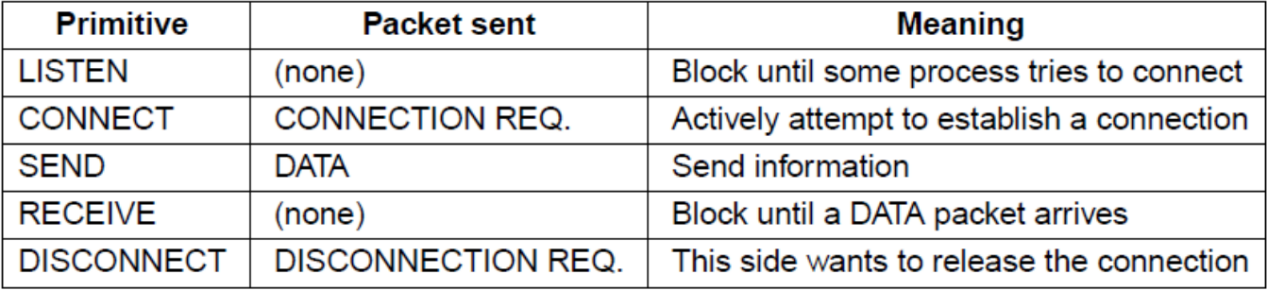
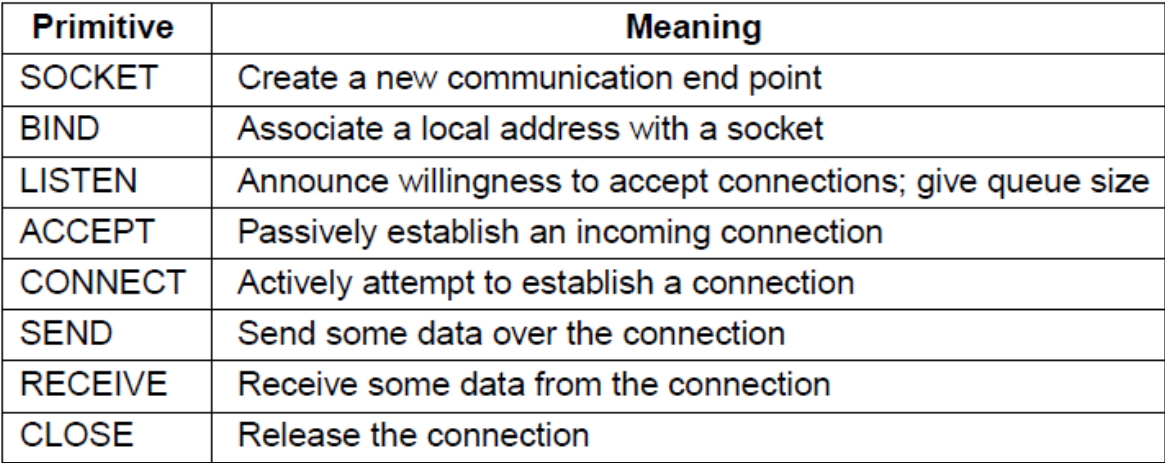
**Internet的网络层：**

* IPv4协议：（看IP报文实验报告）
* IP地址：
  + 结构：
    - 网络地址：即网段，网段内的主机可互联
    - 主机地址：标识唯一的一个终端
    - 子网掩码：标出网络地址和主机地址的分界，高位全为1，低位全为0。对于一个IP地址一般配有掩码，为1的位对应的位是网段地址，为0的位对应的位是主机地址
  + 类型：
    - 
    - A类私有：10.X.X.X
    - B类私有：172.16.X.X-172.31.X.X
    - C类私有：192.168.X.X
    - 私有地址不要在公网出现，同时尽量不要在主机地址上设为全0或全1，以应对不同厂商对这种特殊地址的处理
  + CIDR（解决以类别划分子网的利用率不高的问题）：
    - 路由聚合：小路由前缀（X.X.X.X/x）可以聚合成大路由前缀（Y.Y.Y.Y/y），其中y<x
    - 路由分解：上述的逆过程
  + NAT地址转换（实验报告辅助学习）：
    - 通过防火墙/盒子进行NAT服务提供和地址转换
    - 用端口号进行NAT池的地址复用（虽然使用了传输层的概念解决网络层的问题，破坏了协议的分层独立性，但由于该工程问题没有更好的办法，所以还是推广使用了）
* IPv6：
  + 目的：
    - 解决IPv4地址有限的问题
    - 减少路由表空间
    - 简化协议
    - ...
  + 报文格式（固定部分）：
    - 
    - 流标签：希望得到“特殊待遇”的包会在流标签上修改
    - 缺陷：一种流标签有专门的一套转发策略，渐渐有了面向连接的特性，与网络层根据状态转发的策略不一致了
    - Next header：以链表形式构建一系列包，指定下一个包信息，比IPv4简洁
    - Hop limit：与IPv4的TTL类似
* Internet控制协议：
  + ICMP：ping
  + IGMP：互联网广播分组协议
  + DHCP：广播DHCP服务信息，主机若需要则查看该DHCP提供哪些地址，找一个来配置自己的IP信息
  + ARP：根据IP地址获取目标MAC地址
  + RARP：在无IP的情况下根据MAC地址申请IP地址（适合无盘工作站）
* MPLS：
  + 
  + 先打上label，然后其余路由器根据不同label决定转发路径（有IPv6的思想）
* OSPF：**在内网使用**，直接走**IP**协议
* RIP：**在内网使用**，使用**UDP**传输层协议
* BGP：**跨内外网使用**，使用**TCP**传输层协议

**Chapter 06：Transport Layer**

**-----------------------------------------------------------**

**传输服务：**

* 目的：给用户提供高效、可靠、成本低的数据传输服务（即并不相连的端到端传输，且接收端是透明的）
* 给应用层提供的服务：与目的相同
  + 在传输层内完成工作的单位称为**传输实体**
  + 传输实体与应用层可能有多个服务访问点（通过端口号划分），可能是主机的多个进程都需要进行传输
* 传输服务原语
  + 
  + DDos攻击：大量无用的连接请求发到服务器上，挤占资源导致真正有需求的连接无法完成
* Berkeley套接字
  + 
  + BIND将底层信息绑定到套接字上，如分配硬件资源、针对大小端存储的操作等

**传输协议要素：**

* 寻址：
  + TSAP：传输服务访问点（端口号）
  + NSAP：网络服务访问点（IP地址）
  + 使用五元组表示传输层连接：
  + 服务器有“前台进程”，专门处理传输层访问，给客户端明确应当去访问哪个端口号
* 连接建立：
  + 前提：主机发送数据的序列号产生使用了时间作为种子，这样不会被别的主机“猜到”要发送的序列号从而伪造数据发起攻击
  + 三次握手：
    - 正常流：
      * 客户端发送连接请求（seq=x）
      * 服务器发送确认（seq=y, ack=x）
      * 客户端发送再确认（seq=x, ack=y）
      * 连接建立
    - 异常流（三次握手可以解决的问题）：
      * 旧的/无用的连接请求到达服务器（seq=x）
      * 服务器给客户端发送确认（seq=y, ack=x）
      * 客户端发送拒绝信息（ack=y）
      * 连接没有建立
    - 然而该协议不是完全可靠**（参考两军对垒问题：穿过不确定区的数据包总是不确定的）**
* 连接释放
  + 四次挥手：
    - 正常流还是三次握手：
      * ①客户端请求拆链
      * ②服务器确认拆链
      * ③客户端再次确认
    - 异常流及处理：
      * ③数据包未能到达服务器（服务器定时器超时后也释放连接）
      * ②数据包未能到达客户端（客户端定时器超时后重新发送①数据包请求）
      * ②数据包总是未能到达客户端（客户端N次失败后释放连接，服务器定时器超时后也释放连接**（后者如果不做会维护半开连接，严重挤占资源）**）
* 差错控制和流量控制
* 多路复用：一个IP有多个端口号，一个端口号可以走多个IP进行传输
* 崩溃恢复：针对使用某一发送策略的发送方，接收方使用多种接收策略都无法应对所有情况，因此只能交给应用层解决

**拥塞控制：**

* 理想带宽分配：
  + 理想情况下，实际吞吐量接近于网络使用的容量
  + 理想情况下，延迟一定
  + 最大-最小公平性：
    - 假设各路由器间带宽相等
    - 在链路确定时，若同时竞争则平分，一旦某一段有结果，则整个链路的流量就确定了
    - 在一个有竞争的链路上，若其它流的流量已确定，剩下的全部平分给剩下不确定的流
* 调整发送速率：
  + 流量控制：接收方缓冲区不足
  + 拥塞控制：网络容量不足
  + 加法递增乘法递减：接收方逐步按加法（一次增加固定带宽）发送，“试探”接收方的处理能力。一旦感受到拥塞，减少为原来的几分之一

**UDP协议：**

* 概述：
  + 提供无连接服务，不需要建立连接
  + 报文仅有源端口、目的端口、UDP长度、校验和（伪首部仅用于校验，不参与传输）
* 远程过程调用**（应用层）**
  + 对于终端，向主机发送消息并获得应答，人们希望抽象成函数的调用。在进行编程时，隐藏网络的一些细节
  + 问题：指针传递（指针空间完全不同）、无法推断矢量大小、无法推断参数类型、全局变量
* 实时传输协议**（应用层）**
  + RTP运行在UDP之上，通过套接字与UDP层连接
  + 数据包到达一般不平均（各包的传输延迟不同），此时对于实时传输，需要在接收方创建缓冲区，缓冲后就可以在一定时间内平滑处理数据
  + 数据抖动：衡量数据包传输延迟的差异
  + 在抖动的情况下，通过RTP协议传输

**TCP协议：**

* 概述：提供可靠的端到端字节流传输服务
* 服务模型：
  + 套接字有套接字编号，由IP地址和端口号组成（0-1023端口号保留，用户不能使用）
  + 为了提高传输效率，TCP使用粘包机制，等到待发送数据缓冲到一定大小再发送（但这也降低了实时性，故数据实时性不如UDP）
* TCP报文格式（看实验PDF）
* 连接建立：三次握手（置SYN位为真）
* 连接释放：四次挥手（有的FIN位为真、有的ACK位为真）
* 滑动窗口：双方进行协商，接收方的ACK报文中会包含剩余缓冲区大小，供发送方进行判断
  + 在发送短数据时效率低下
  + 可以通过Nagel算法（即“粘包”机制）提高传输效率，但会降低实时性
  + Clark解决方案是，禁止接收方发送剩余缓冲区为1的ACK消息，相反，到缓冲区剩余空间达到一定大小时才告知发送方
* 计时管理：RTT基本符合正态分布，且越尖说明抖动越小
* 拥塞控制：加法递增乘法递减

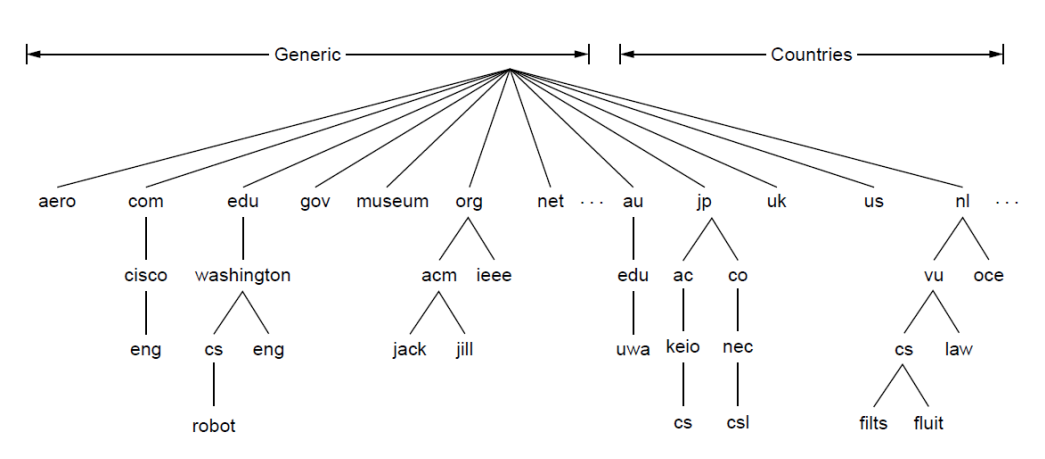
**性能问题：**

* 计算机网络中的性能问题：UDP解决广播风暴问题
* 网络性能测量：
  + 样本要足够且有代表性
  + 缓存的处理方法会破坏测量结果（缓冲区满时策略：牛奶策略尽可能将新数据包转发；酒策略先将陈旧数据包转发）
  + 小心使用粗粒度时钟
* 对快速网络的主机设计：
  + 提高主机性能更重要
  + 减少包发送数量
  + 避免拥塞比崩溃恢复更好

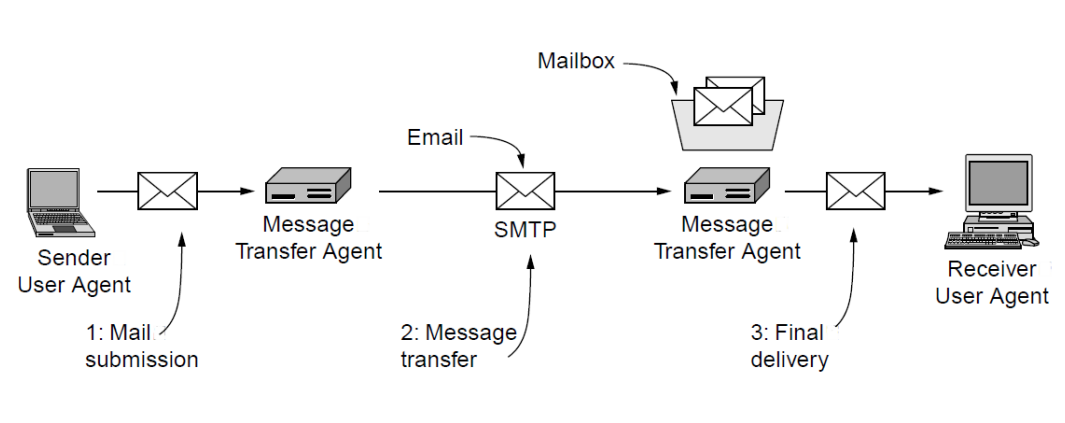
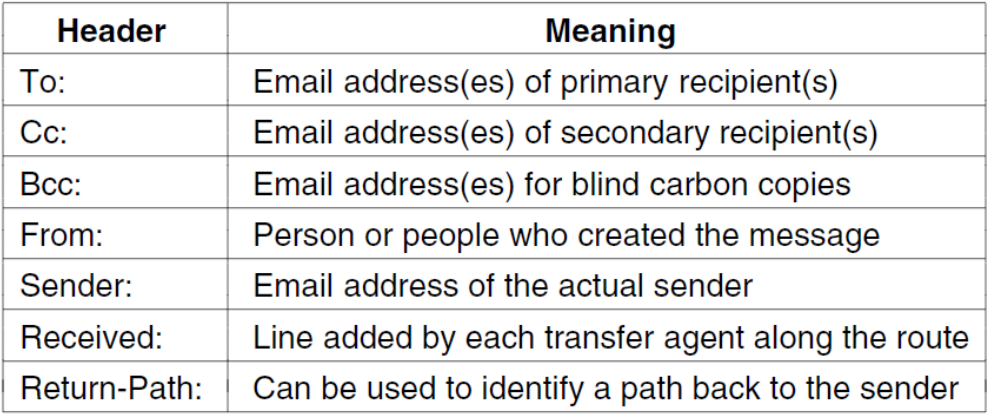
**Chapter 07：Application Layer**

**-----------------------------------------------------------**

**DNS：**

* 命名空间：分有通用域名和国家域名
  + 
* 域名会递归查找，一个DNS服务器没找到则去上一层DNS服务器查找

**电子邮件：**

* 使用协议：SMTP
* 体系结构与服务：双方都要通过传送代理（邮件服务器）进行邮件操作
  + 
* 邮件格式：
  + 
  + Cc：抄送，且其他人可以看到你还发给了哪些人
  + Bcc：密件抄送，即抄送但其他人不可以看到你还发给了哪些人

**万维网：**

* 体系结构：浏览器给不同服务器发送HTTP请求