⚠本笔记侧重点为覆盖Prof. Guangtao Xue (薛广涛), Dr. Yi-Chao Chen (陈奕超)班级的考试知识点，需结合Slides食用

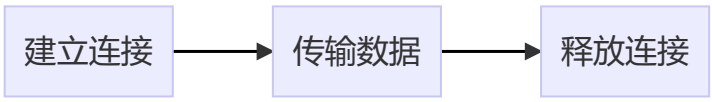
计算机网络知识点整理

💻 Peiyu Chen

✉ pietychen@sjtu.edu.cn

Ⅰ 计算机网络体系结构

- 广播式网络：所有计算机共享公共通信信道，所有计算机都收听报文，通过判断目的地址决定是否接受——**局域网**
- 点对点网络：每条物理线路连接一对计算机，没有直接连接的通过中间节点的接受、存储和转发——**广域网**
- 面向连接服务：(TCP)



- 无连接服务：(IP/UDP)尽最大努力交付
- 物理层：传输单位为比特
- 数据链路层：传输单位为数据帧，差错控制/流量控制/传输管理，点到点通信（IP地址/硬件之间）
- 网络层：传输单位为数据报，流量控制/拥塞控制/差错控制/网际互联，路由算法计算合适路径
- 传输层：传输单位为报文段（TCP）或用户数据报（UDP），端到端通信（进程之间）
- 应用层：多种协议，HTTP，FTP，SMTP
- 传输控制协议（TCP）：面向连接，传输单位为报文段，提供可靠交付
- 用户数据报协议（UDP）：无连接，传输单位为用户数据报，尽最大努力交付

Ⅱ 物理层

- 码元：用固定时长的信号波形（数字脉冲）表示一位k进制数字，如二进制中的0码元（低电平）和1码元（高电平），是离散值
- 基带传输：传输基带信号，如将0和1的数字信号直接以两种不同的电压表示
- 宽带信号：传输模拟信号，将基带信号进行调制后形成频分复用模拟信号

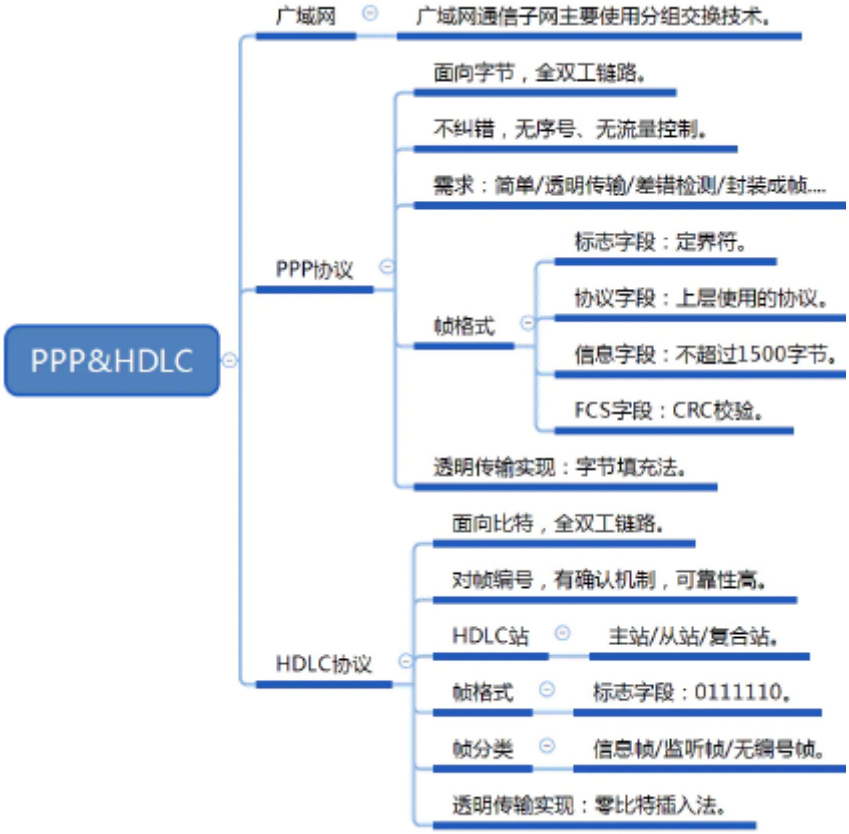
- 单工通信：单向
- 半双工通信：可以双向但无法同时
- 双工通信：有两条信道，可以同时发送和接收信息
- 信道极限容量：信道的最高码元传输速率（最高波特率）或信道的极限信息传输速率（最高比特率）
- 码元传输速率（波特率）：单位时间内数字通信系统传输的码元个数，单位Baud（波特）
- 信息传输速率（比特率）：单位时间内数字通信系统传输的**二进制**码元个数，单位是b/s
- **奈奎斯特定理：**

  理想低通信道中极限码元传输速率为$2WBaud$，则极限数据率为$2Wlog_2Vb/s$，W代表信道带宽，V代表码元离散电平数目。
- **香农定理：**

  信道极限传输速率$Wlog2(1+\frac{S}{N})$，W代表信道带宽，$\frac{S}{N}$代表信噪比。
- 数据报子网和虚电路子网的区别⚠寻址，寻径，连接，可靠性，故障，顺序
- 传输/建立连接时延，顺序，线路利用率，灵活可靠性
- 电路交换：
  - 优点：传输时延小，传输有序，不会发生冲突，实时性强控制简单
  - 缺点：建立连接时间长，灵活性差（中间断了就全断了），线路利用率低
- 报文交换：
  - 优点：无需建立连接，线路动态分配可靠性高，线路利用率高，可以做到电路交换做不到的多目标发送
  - 缺点：报文段大小不确定，需要额外缓冲区，需要存储转发，存在时延
- 分组交换：
  - 优点：所有报文交换的优点，且分组大小一定，缓冲区易于管理，数据出错概率更小，需要重传的数据量也更小
  - 缺点：存储-转发时延，每个分组需要携带额外信息，接收方需要对分组进行排序

Ⅲ 数据链路层

- PPP协议和HDLC协议的区别



- HDLC：面向bit的同步通信协议
- PPP：处理错误检测，支持多种协议，允许协商IP和身份认证
- 网段：物理层概念，所有使用同一物理层设备的能够直接相连的设备即在同一网段，网段对应冲突域
- 局域网：数据链路层概念，指二层可达网络，局域网对应广播域
- 子网：网络层概念，是一个IP划分出的几个小范围的网络
- ARQ协议：
  - 停等ARQ，窗口大小为1，发送后设置重发定时器，收不到ACK则重发，ACK1bit交替使用，重复ACK代表帧重复
  - 回退N帧ARQ，发送窗口大小最大2^n-1，若为2^n则会出现重传时接收方无法辨认是新的一组窗口还是重传。接收方只按顺序接受帧，会出现冗余重传
  - 选择重传ARQ，窗口大小最大2^{n-1}，防止重叠无法辨认新旧帧，只有选择重传有NAK，其他的都是等超时
  - 连续ARQ，窗口足够大的ARQ

MAC LAYER

多路访问协议CSMA：

- 纯ALOHA：冲突危险区2t，吞吐率$S=Ge^{-2G}$，G为帧发送的平均值
- 分隙ALOHA：冲突危险区t，吞吐率$S=Ge^{-G}$，G为帧发送的平均值
- 1-persistent CSMA：持续监听，忙则等待，不忙立即发
- 非持续CSMA：：发送则监听，忙则延续一段时间再监听
- p-persistent CSMA：忙则等到下一时隙，p概率发送，（1-p）延到下一时隙
- CSMA/CD：设信道发送时延为τ，则只要在2τ时间内没检测到冲突就不会发生冲突，故CSMA/CD规定了最小帧长为$2*\tau*v$，这也是为何要求发送方在发送完之前就要收到自己发送的数据的原因。（自发自收检测法）

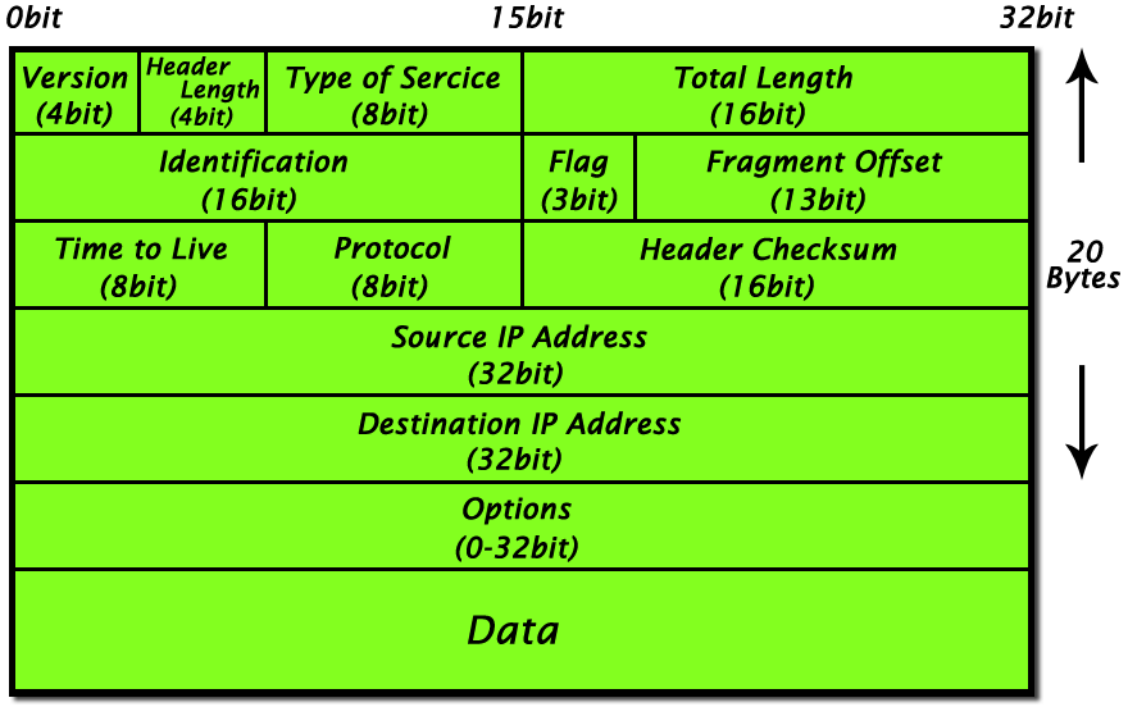
    - 还有信号电平法（基带传输，冲突时电压加倍）和过零点检测法（曼彻斯特编码，冲突时零点偏移）
- 位图协议：专门拿一个竞争周期的位来构建位图,作为竞争周期后的发送顺序，效率[$d/d+n, d/d+1$]
- 二进制倒计数法：n个发送站站号的二进制表示从大到小排列，效率$d/d+log_2 n$
- 有限竞争协议：
    - 对称竞争协议：n个站点，每个站点发送概率为p，则发送成功概率为$np(1-p)^{n-1}$，$np=1$发送效率/成功率最高
    - 适应树搜索竞争协议：每次允许参与竞争的减半
- LLC Layer:由于不同网络类型有不同的MAC层，LLC用于掩盖不同物理网络之间的差别为网络层提供服务
- 网桥：可以视作小的交换机，网桥的生成树算法解决防止环状拓扑，散列表类似流表

IV 网络层

- 路由器的两大功能：
    - 路由选择：按照分布式算法，动态改变所选择的路由
    - 分组转发：根据转发表将用户的IP数据报从合适的端口转发出去
- 路由算法：
    - 静态算法
        - Dijkstra算法
        - 扩散法
    - 自适应算法
        - DV算法：
            - 交换得到相邻路由表信息
            - 根据自己到相邻路由表的D(istance)和经过相邻路由表到其他所有节点的D(istance)更新自己的路由表
            - 问题：交换信息量大，收敛慢，坏消息传得慢，有无穷计算问题
        - LS算法
            - 发现邻接路由并得到其网络地址
            - 测量到邻接路由的开销
            - 将开销封装成组并广播
            - 每个路由都成功构建了拓扑信息，再用静态算法算最短路径
            - 优点：一致性好，坏消息传播一样快，传送数据量小占用带宽低
            - 缺点：需要较大存储量与较大计算量
    - 拓扑路由算法：
        - 分层路由
            - 最佳层数lnN，每个路由表表项$elnN$
        - 广播路由
            - 缺点：发送量大，需要知道所有地址信息；流量大，占用大量带宽
        - 多目的地路由：到分支就复制，优点为流量小带宽占用小，缺点为复制不均衡对路由器要求高
        - 生成树算法：基于链路状态算法（知道整个网络拓扑，DV算法不行，只知道局部拓扑），优点为最佳利用带宽，缺点为每个路由器都得计算生成树
        - 逆向路径算法：对邻接路由发组，邻接路由如果是第一次收到则向其邻接节点转发，若不是第一次收到则停止转发，这样逆向的构建了路由，但对大型网络不适用

        - 多址生成树与多址核心树：网络路由划分为组，核心树中每个组只有一个生成树，要发送都通过核心路由转发，优点为减少存储
    - Ad-Hoc路由算法
        - 主动路由DSDV：周期性广播路由信息并更新路由表
        - 按需路由：
            - AODV
                - 每一条链路都双向对称，需要建立路由时向邻接节点发送RREQ，邻接路由有就直接返回RREP，没有继续向邻接节点转发直到有路由的路由表记录了路径或到达目的路由
                - 沿途路径都将获得路由信息，非沿途的经timeout后丢弃之前RREQ时记录的逆向路径
                - 通过TTL递增设置限制广播范围防止大范围广播
                - 周期性广播HELLO检验邻居节点有效性，无效删除自己路由表相关信息并通知使用到该失效节点的相关邻接路由
            - DSR
                - 首先检查重复请求并相应丢弃，然后检查自己的Route Record中是否有相应路径，有则返回RREP，如果自己是目的节点则也返回RREP，如果自己是中间节点则将自己加入Route Record并继续广播
                - 如果是单项链路，目的节点再启动一次RREQ寻找源节点并将路径返回给源节点，如果是双向链路则可以直接返回路径信息
            - AODV和DSR比较：
                - AODV链路双向，DSR可以支持单向
                - AODV只有单一路径，DSR由于维护Route Record可能具有多条路径
                - 信息更新：AODV广播会让所有相关路由更新，DSR只会沿着路径向上游更新到源节点，其他路径上的路由无法完成更新
    - 拥塞控制
        - 开环控制：提前做好设计
        - 闭环控制：通过显式反馈（抑制分组）或隐式反馈（丢包率高）来调整操作并控制拥塞
        - 虚电路子网：通过准入控制拥塞，建立连接后不可能出现拥塞
        - 数据报子网：ACK上加警告位（警告位方法）/发送抑制分组/通知前一站路由减速，并不断向上游传递知道源主机（Hop by Hop）
        - 载荷脱落：RED，随机丢掉分组并显式或隐式通知
    - 服务质量
        - 流量整形：
            - 漏桶：桶满溢出导致分组丢失，不能体现通信量突发
            - 令牌桶：消耗累计令牌期间不会有速度限制，可以体现通信量突发
        - 资源预留
        - 准入控制：通过传输一组流说明参数（途径所有路由器都可以对其进行修改）到接收端，接收端判断是否准入
    - 集成服务
        - RSVP协议：基于接收端，接收端根据路径逆向的为途径路由预留资源
    - 区分服务
        - 四种优先级三种丢弃概率共十二种区分服务
    - 标签交换
        - 类似虚电路子网（的虚电路号），根据标签转发而不是根据路由表转发，可以加速
        - MPLS是标签交换的标准
- 网络互连

    - AS：自治系统，AS内由IGP（内部网关协议）处理，AS间由EGP（外部网关协议）处理
    - 分段与分组：分段报头由三个部分组成：分组号，分段的offset，是否为最后一段的标志
        - 透明分段：入口网关分段出口网关重组，可以减少包数量，降低段头带来的overhead，但需要额外重组时间和额外buffer，且分段都得走一条路，无法体现路由优越性
        - 非透明分段：由目的主机重组，有段头overhead，包多，带宽用量大
- IP header：

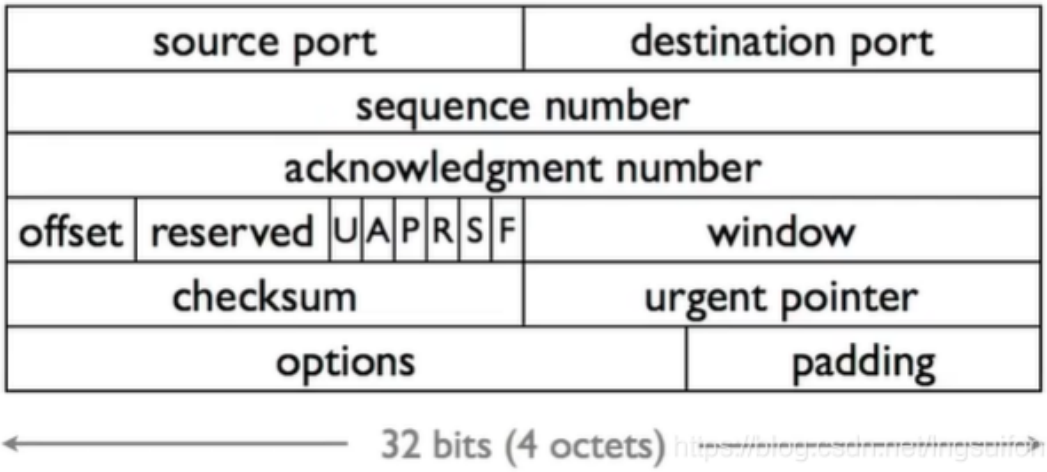


- CIDR：目的Ip与掩码长度最长的做and运算，看运算结果是否为该掩码的子网的Ip段
- NAT：内部IP和内部端口号经过Router的NAT表，根据序号获取公有IP和公有Port进行转换并把私有IP和私有Port存在相应序号处，访问结束返回Router时按相反过程存公有取私有并将数据传回主机
- IP控制协议
    - ICMP协议，把通信过程中的错误向源站点报告
        - ping
        - tracert：
            - 控制TTL递增可以知道每一跳的路径
        - 获取最小MTU：设置DF为1不允许分段，找到分组被丢弃的地方，不断减小分组长度直到可以继续发出去
    - ARP协议：IP转MAC
    - RARP协议：MAC转IP
    - BOOTP协议：和RARP类似
    - DHCP协议：
        - 新客户广播DHCP Discover
        - DHCP服务器动态（IP池）或静态（提前分配）的为请求站点分配一个IP放在DHCP OFFER中
        - 客户收到IP后选择接收哪一个服务器的IP并发回DHCP REQUEST，未收到DHCP REQUEST的服务器一段时间后归还IP地址
        - 收到DHCP REQUEST的服务器发回ACK并标记IP状态为租用状态
- 静态路由：人工配置路由表防止无限增长（特例：缺省路由）
- 动态路由：根据网络情况动态更新路由表

- RIP协议：基于DV算法
- OSPF协议：基于LS算法与分层路由
- BGP协议：基于自治区域与DV算法

VI 传输层

- 传输服务：作为资源子网和通信子网中间层，将通信子网的细节向资源子网屏蔽，为通信子网做可靠性保证
- 传输层有逻辑上的数字管道
- 数据链路层与传输层对比
    - 相同点：都可以提供可靠的数据传输
    - 不同点：
        - 数据链路层的传输通道为物理通道，传输层为一个网络
        - 数据链路层连接建立简单，传输层则需要路由，涉及到路由效率/路由算法
        - 数据链路层中间无存储转发过程，但传输层有
        - 数据链路层使用一对缓冲区做流量控制，传输层的缓冲管理复杂
- 传输层寻址：TSAP（端口号），应用层应用先连接TSAP获得端口号然后再与网络层的SAP建立连接，接收端为反过程
- TSAP如何知晓：
    - well-know，如ftp的port 21 http的port80
    - 名字服务器：类似114查号，服务器端有一个端口映射器监听TCP连接，访问者的TCP请求先发到端口映射器获取目标应用的端口号，断开与端口映射器的连接并重新根据获取到的TSAP建立新的连接
- 传输层连接建立方法：三次握手法（seq和ack不+1），第一次握手请求方发出CR，第二次握手接收方发回ACC，第三次直接传数据
- 传输层连接释放方法：
    - 非对称释放，一方直接释放，可能造成数据丢失
    - 对称释放，三次握手，第一次请求方DR，第二次接收方DR，第三次请求方ACK
    - 由于有保活计时器与重发计时器存在，一定可以释放成功
- 多路复用：
    - 向上多路复用，多个传输层使用一个网络层，提高网络层利用率
    - 向下多路复用，一个传输层使用多个网络层，提高带宽和传输速率
- 崩溃恢复：
    - 数据报子网：如果TPDU有副本，直接重发
    - 虚电路子网：建立连接，确认丢失数据，重发丢失数据，释放连接
    - 主机崩溃：一定有异步ack和写入带来的问题
- UDP协议：
    - 无连接，不做三次握手，无可靠性保证，相比IP协议只是多了端口协议，效率高
- TCP协议：
    - 面向连接，做三次握手（握手ACK+1），保证可靠性，端到端服务，全双工
    - 套接字：IP+端口，一个套接字可以有多个连接
    - TCP Header：



- TCP连接，三次握手，注意ACK+1
- 释放连接：9步
- 传输策略：剩余缓冲区大小即窗口大小，零窗口公告（持续定时器防止更新窗口公告丢失）
- 拥塞控制：动态调整拥塞窗口（慢启动（指数递增），拥塞避免（出现拥塞降到一半或降到1个MSS），快恢复（执行拥塞避免），快重传（出现多个重复ACK直接重传而不是等重发定时器结束再重发））
- 定时器管理：
  - 保活定时器：用于连接，两个相互连接的端口长时间无数据传输（时间由保活定时器控制）则发送探测保温检测是否失活，失活则断开连接
  - 重发定时器：ACK超时就重发
    - $RTT = aRTT_0 + (1-a)M_0$
    - $D = aD_0 + (1-a)|RTT_0 - M_0|$
    - 重发定时器的超时值设置为$RTT + 4D$
  - 持续定时器：防止零窗口公告后的更新窗口公告丢失，超时后发送方发送1Byte的探测报文，如果有缓冲区空出来就可以继续发了