局域网一般不大于10km 城域网的范围要比局域网大

PAN LAN MAN WAN

协议放在header里，每层数据往下传时都加一个自己的头

电路交换（交换机 独占资源 建立连接） 分组交换（**packet** 路由） 报文交换（报文：一个完整的数据单元 数据较大 **message**）

开放系统互连基本参考模型OSI：

Physical data link network transport session presentation application

（物理层 数据链路层 网络层 传输层 会话层 表示层 应用层）物链网输会示用

**物理层：**负责在**物理媒介上传输原始的比特流**。

涉及电气信号、光信号、物理连接器和硬件设备。

**数据链路层：**负责**在相邻网络设备之间传输帧**。（将比特流组织成帧，通过MAC地址实现局域网内设备的通信，**并提供错误检测功能**。）

涉及错误检测和纠正，以及流量控制。流量控制经典的滑动窗口

**网络层：**数据包的寻址和路由（在源和目的间选择路径）

**传输层：端到端的数据传输服务**，支持可靠（TCP）和不可靠（UDP）传输。它管理数据分段、流量控制和错误恢复，确保数据完整性。

**会话层：**建立、管理和终止设备间的会话，协调通信过程（断点续传，远程登录）

**表示层：**负责数据的表示、编码和转换。 **语法和语义**

确保一个系统的应用层所发送的信息可以被另一个系统的应用层读取。

**应用层：**为应用软件提供网络服务（资源访问 **文件传输 电子邮件**）

TCP/IP模型：

**网络接口层：**相当于OSI模型的物理层和数据链路层。

负责在物理媒介上**传输数据帧，以及介质访问控制。**

**网际层：**功能相当于OSI模型的网络层

**传输层：**一样，主要协议有TCP和UDP，负责提供可靠的或不可靠的端到端数据传输。

**应用层：**会话层+表示层+应用层（HTTP、FTP、SMTP、WWW等）

还有五层体系结构：application transport network datalink physical（应用层 传输层 网络层 数据链路层 物理层）

物理层：

传输方式三种：**单工（simplex） 半双工（half-duplex） 全双工（duplex**）

同步串行：

以时钟信号线对传输的信号进行比特同步

以数据块/帧/分组为单位传

异步串行：

**独立时钟**无须同步

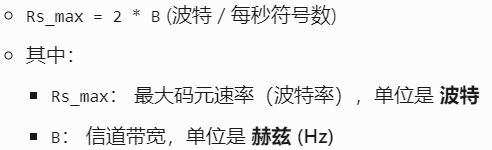
以字符为单位传

发送两个字符间隔任意 同步靠起始/停止位

奈氏准则：——理想无噪声！

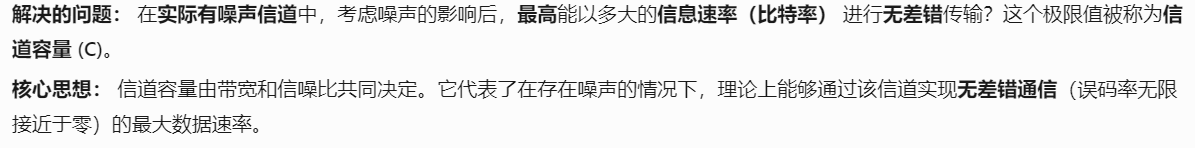


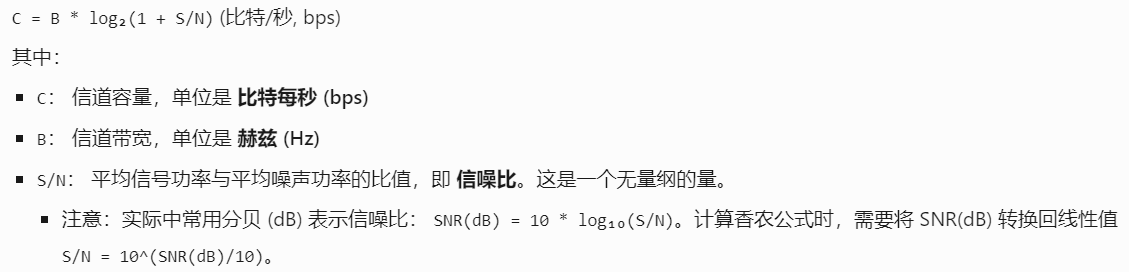
发送符号的速率不能超过信道带宽的两倍。或者说，在给定带宽 B Hz 的信道中，能够区分的独立信号状态变化（码元）的最大速率是有限的。



**也可以是：c=2Hlog2v 单位bps**

香农定理——有噪声环境





也可以是：**Hlog2（1+S/N） db=10log10S/N**

传输介质：双绞线（twisted pair） 同轴电缆（coaxial cable） 光纤电缆（fiber cable）……

低通low-pass 带通band-pass

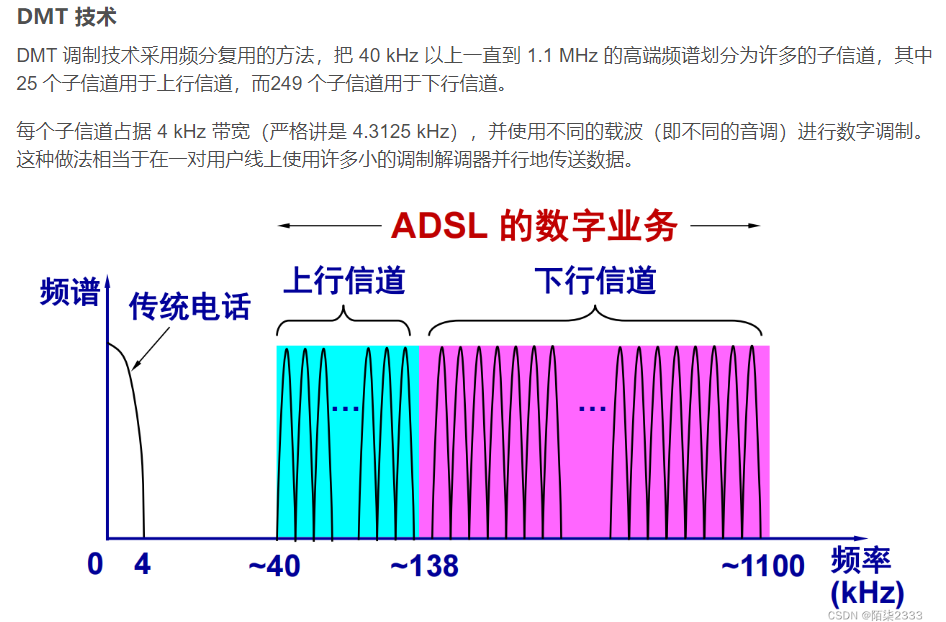
基带传输：baseband 传数字信号 宽带传输（宽带定义）：passband传analog signal

编码：

|  |  |
| --- | --- |
| ​​基带编码技术​​ |  |
| 曼彻斯特编码 | 每个比特中央电平跳变：0=低→高跳变，1=高→低跳变 |
| 差分曼彻斯特编码 | 比特起始跳变表示0，无跳变表示1；每个比特中央强制跳变（仅用于时钟同步） |
| NRZI编码 | 0=电平跳变（翻转），1=电平保持恒定 |
| ​​调制技术​​ |  |
| QPSK | 正交载波调制：用 4种相位（0°, 90°, 180°, 270°）表示 2比特 |
| QAM | 幅度与相位联合调制：用 星座图中不同点（如16-QAM有16个点）代表多比特组合 |
| ​​多路复用技术​​ |  |
| FDM | 将总带宽分割为多个频段，每路信号独占一个频段传输 |
| TDM | 将时间划分为固定时隙，多路信号轮流占用时隙传输 |
| CDMA | 每路信号用唯一正交码扩频，混合传输后接收端通过码字分离信号 |
| WDM | 在光纤中用不同波长（颜色）的光承载独立信号（光域的FDM） |

​

非对称数字用户线ADSL：



计算：**电路交换：t=s+x/b+kd**

**包交换：t=x/b+(k-1)p/b+kd （x总长！包括加的报文头）**

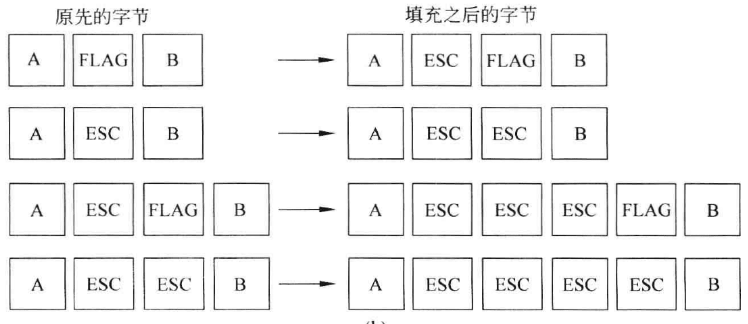
数据链路层：

提供给网络层三种服务：无确认无连接 有确认无连接 面向连接 **有连接必有确认**

ARQ：自动请求重发 **在接收端进行差错检测，并自动请求发送端重发的差错控制技术**

**经典考题——成帧的字符计数法和比特填充的标志比特法：**

byte stuffing（特殊字节FLAG作为开始和结束 出现在**数据里**面加入转义字节ESC 接收方在传给网络层时**必须删除ESC** 转义字节出现同样要加ESC） bit stuffing（特殊字符模式作为开始和结束 01111110 所以数据中遇到五个连续1添0…… 找最近的7E作为起止标志）

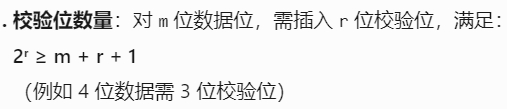


比特填充的标志比特法 (Bit Stuffing)：每个帧的开始和结束由一个特殊的比特模式，01111110 或十六进制 0x7E 标记。这种模式是一个标志字节。每当发送方的数据链路层在数据中遇到连续五个 1，它便自动在输出的比特流中填入一个比特 0。这种比特填充类似于字节填充，在数据字段的标志字节之前插入一个转义字节到出境字符流中。

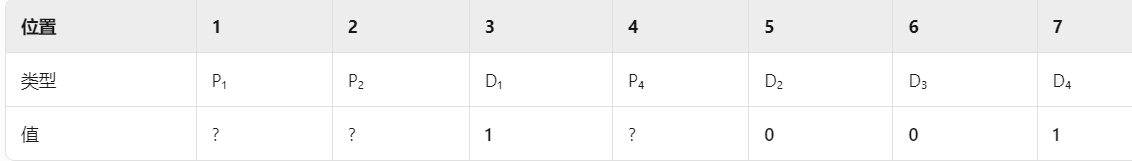
当接收方看到 5 个连续入境比特 1，并且后面紧跟一个比特 0，它就自动剔除比特 0；01111110 可能出现多次。

**差错控制的纠错码：**

海明码**（m+r+1<=2r次方）**

因为：校验位位于 2ⁿ位置​​（n=0,1,2,...）

举个例子：4 位 D = 1001 P为检验位 D为数据位

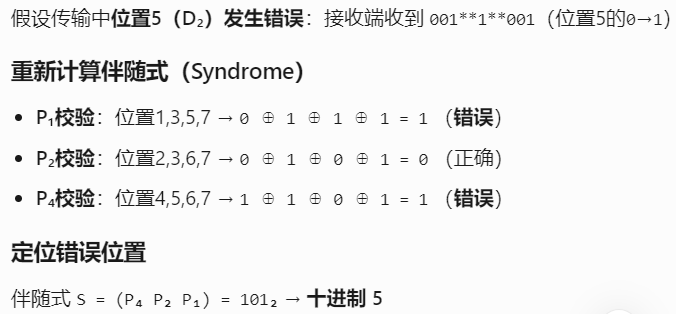


第k为检验位覆盖位置（二进制第 k 位=1），比如P2： 2 3 6 7的倒数第二位为1，应该有：

P₂ ⊕ 1 ⊕ 0 ⊕ 1 = 0 由此计算P2

海明码——》单比特纠错方案

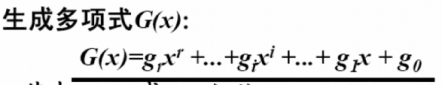
怎么纠错？伴随式： 把上面的每个检验位P和真实计算的检验位异或后，重组，若错误位置 ！=0，将该位置的比特翻转即可纠正错误。



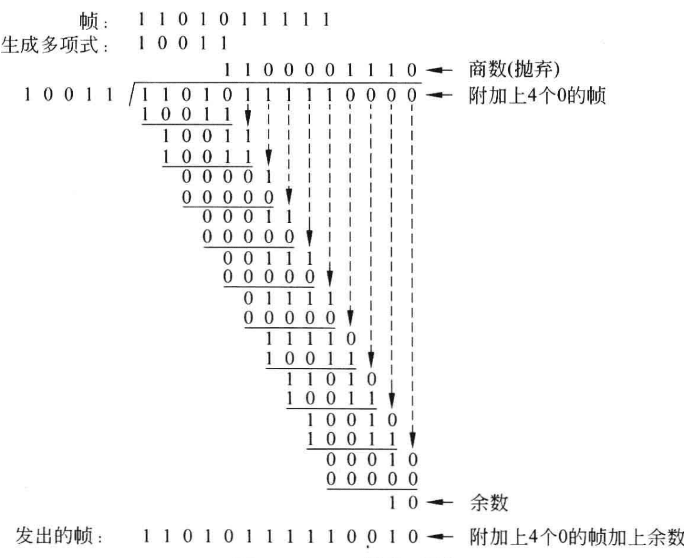
奇偶校验 CRC（多项式编码 polynomial code） 也称循环冗余码

加入**阶数个0** 模二除法（算术运算加法没进位减法无借位 全部等同于异或 最高位保持为0 舍 **remainder：余数 reminder：除数**）

除的那个数（生成多项式 由通信双方协定）



计算的例子：

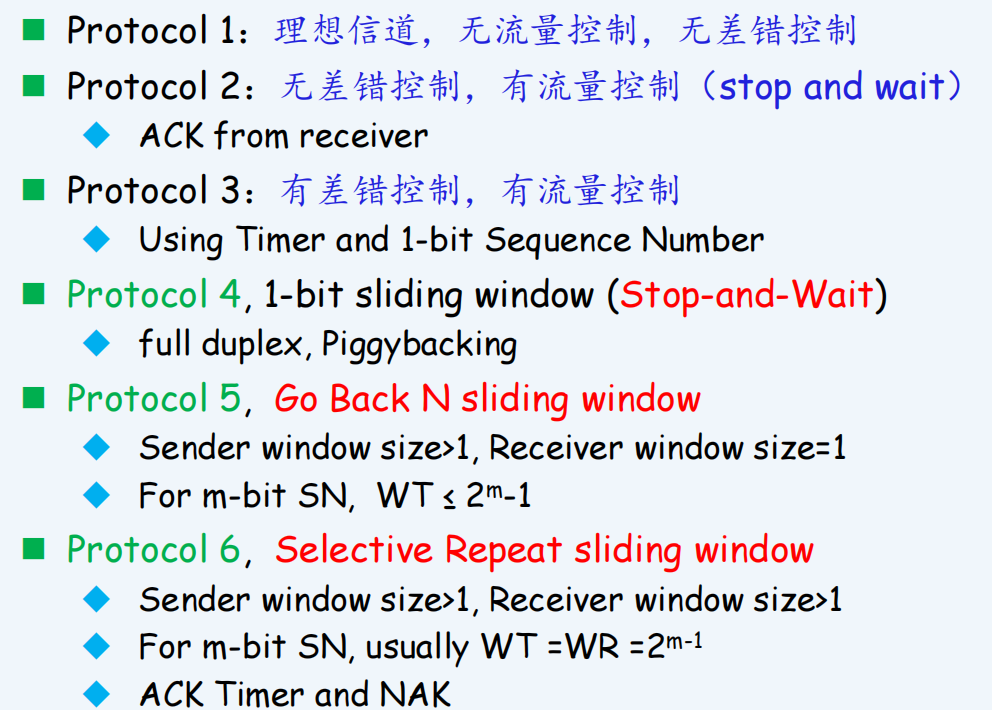


基本链路层协议：

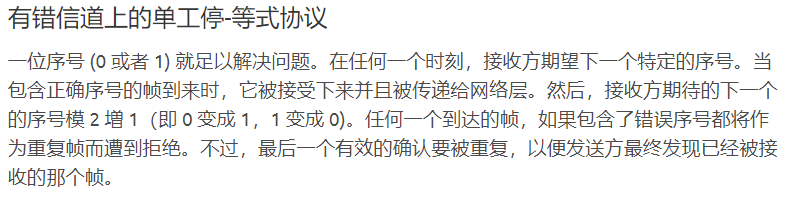
协议一：单工 无确认无连接 发送方发 接收方就收

协议二：单工停等 确认后才发 有误发NAK

协议三：帧上放序号

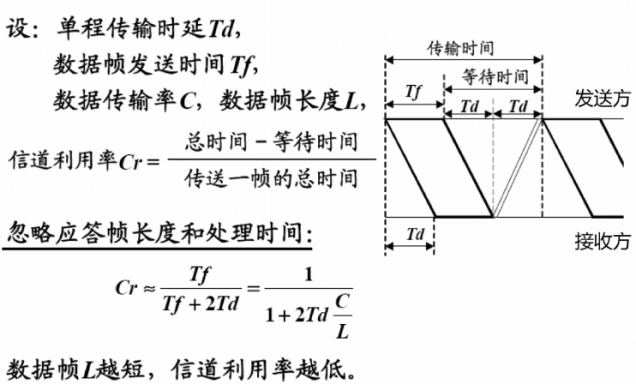


重点是后面三个 停等 GBN 选择重传

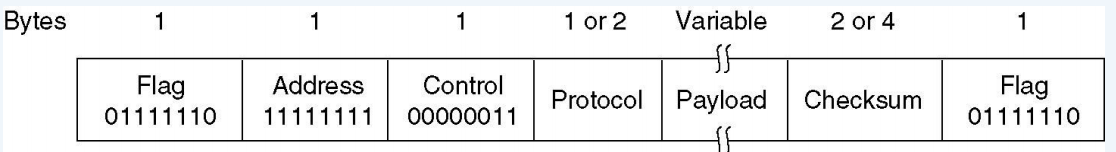


这里有自动重复请求ARQ

信道利用率计算：



链路层的基本协议有SLIP PPP等



PPP协议↑ 以字节为单位

HDLC高级数据链路控制

**HDLC——面向位的同步传输！！！（一次传输一块数据）未限制数据类型**

**介质访问子层MAC**

**——确定多路访问信道下一个使用者**

强调：MAC子层与逻辑链路控制子层（Logical Link Control, LLC）一起构成了数据链路层。

**LLC：识别网络层协议，处理数据包**

**MAC：数据帧的封装/卸装**

介质访问控制子层需要考虑信道分配问题，对于多路访问协议，有：ALOHA 载波侦听多路访问 无冲突协议 有限竞争协议 无线局域网协议

ALOHA协议：

ALOHA 系统的基本思想非常简单：当用户有数据需要发送时就传输。当然，这样做可能会产生冲突，冲突的帧将被损坏。发送方需要某种途径来发现是否发生了冲突。在 ALOHA 系统中，每个站在给中央计算机发送帧之后，该计算机把该帧重新广播给所有站。

如果帧被损坏了，则发送方要等待一段随机时间，然后再次发送该帧。

分槽ALOHA：信道时间划分成离散的时隙，其长度为一帧需要的发送时间，每个站点只能在时隙开始才允许发送（换而言之冲突只发生在时隙的开始）

载波侦听多路访问协议CSMA（在下面以太网处） 带碰撞检测的CSMA——CSMA/CD

无冲突协议：令牌环

通过传递一个称为令牌 (token) 的短消息，该令牌同样也是以预定义的顺序从一个站传到下一个站。令牌代表了发送权限。如果站有个等待传输的帧队列，当它接收到令牌就可以发送帧，然后再把令牌传递到下一站。如果它没有排队的帧要传，则它只是简单地把令牌传递下去。

帧也按令牌方向传输。这样，它们将绕着环循环，到达任何一个目标站。然而，为了阻止帧陷入无限循环（像令牌一样），一些站必须将它们从环上取下来。这个站或许是最初发送帧的原始站，在帧经历了一个完整的环游后将它取下来，或者是帧的指定接收站。

有限竞争协议：

在负载较轻的情况下，竞争方法（即纯 ALOHA 或者分槽 ALOHA) 更为理想，因为它的延迟较短（冲突很少发生）。随着负载的增加，竞争方法变得越来越缺乏吸引力，因为信道仲裁所需要的开销变得越来越大。而对于无冲突协议，则结论刚好相反。在低负载情况下，它们有相对高的延迟，但是随着负载的增加，信道的效率反而得到提高（因为开销是固定的）

所以有限竞争协议首先将所有的站划分成组（这些组不必是两两不相交的）。有 0 号组的成员才允许竞争 0 号时间槽。如果该组中的一个成员竞争成功了，则它获得信道，可以传送它的帧。如果该时间槽是空闲的，或者发生了冲突，则 1 号组的成员竞争 1 号时间槽，以此类推。

当负载很低时，每个时间槽中的站点数量就多一些；当负载很高时，每个时间槽中的站点数量就少ー些，甚至只有一个站

无线局域网协议：  
隐藏终端 暴露终端问题

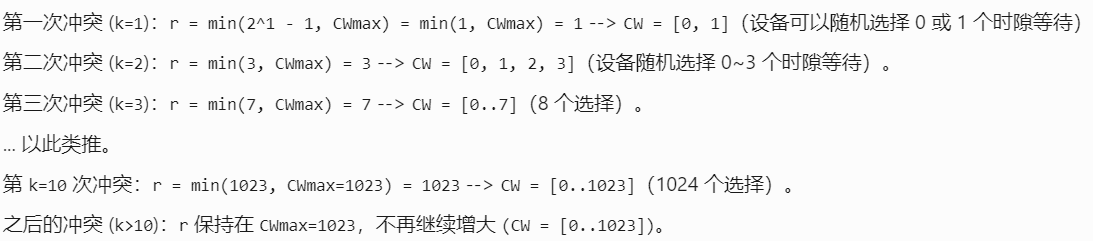
RTS请求发送帧 CTS发送许可

这一部分可再看：[计网学习笔记](https://blog.xqmmcqs.com/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BD%91%E7%BB%9C%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0%20-%20%E7%AC%AC%E5%9B%9B%E7%AB%A0/)

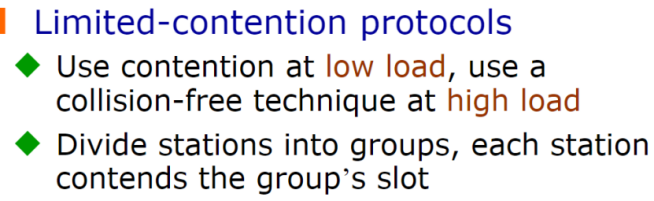
无冲突协议CSMA/CA：载波监听多点接入 / 碰撞避免协议 ——无线局域网



二进制指数退避算法：自适应的解决冲突的办法



有限竞争协议：



**MACA 冲突避免多路访问** 也是无线局域网协议

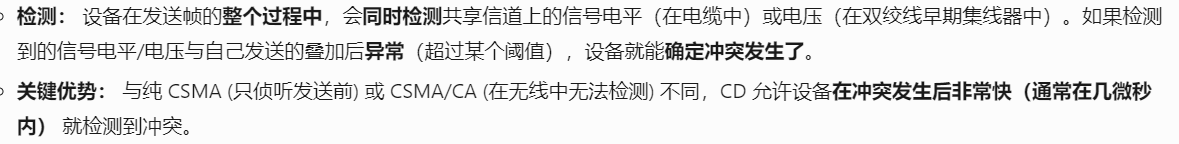
以太网：

使用：**1-坚持CSMA/CD 曼彻斯特编码！**

**CSMA：载波侦听多路访问 CD是冲突检测**

1. 坚持：“坚持”：设备会持续地侦听信道

当设备侦听到信道空闲时，会立即（以概率 1，100%）发送它的帧（不等待）



检测到冲突后，设备会立刻停发帧，发送一个短暂的干扰信号（个信号是为了确保共享信道上的所有设备（包括冲突另一端的设备）都能清晰地检测到这次冲突的存在。）

成功发送信号后，设备启动重传机制——二进制指数后退

非坚持CSMA：要发送数据时，先侦听信道。空闲，立刻发；繁忙，随机等一段时间，然后重新侦听信道

1. 坚持CSMA：

主要设计用于时分槽信道 (Slotted Channel)，时间被划分为离散的时槽。设备只能在时槽起始点开始发送。

要发送数据时，先侦听信道。繁忙，等到下一个时间槽再侦听；空闲：它以概率 P在当前时槽立即发送，并以概率 1-P延迟（推迟）一个时槽后再次侦听（重试）



**必须至少64字节长（数据为46字节 少了就padding）**

交换式以太网：

集线器hub：所有站位于同一个冲突域 必须由CSMA/CD调度

交换机switch：**每个端口都有自己的冲突域（可以同时发多个帧）**

快速以太网——802.3u 10Mbps or 100Mbps

自动协商机制（auto negotiation）双方自动协商最佳速度和双工模式（全双工和半双工）

千兆以太网（gigabit）802.3ab 1Gbps

支持自动协商

——**载波扩充和帧突发**

载波扩充（帧+填充位）定义帧的最小长度512字节，也就是说最小帧长是原来的8倍。这种方法迫使各个站为所有小于4096比特的的帧附加扩充位（填充）

帧突发——连续发送帧

万兆以太网（10 gigabit）

——**只支持全双工** 有自动协商

无线局域网：802.11 802.11a 802.11b……

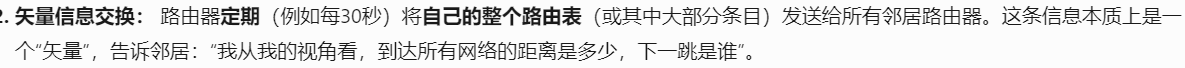
802.11MAC子层协议：**总是半双工（半双工：可以收发，但无法同时收发）**

**使用CSMA/CA 没有办法像之前有线网一样检测碰撞**

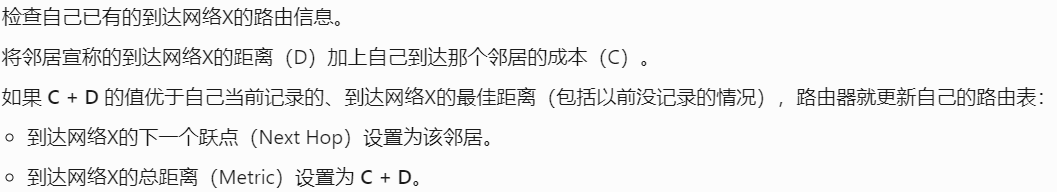
**TXOP：得到等量的通话时间 例题！**

Routing algorithm：

距离矢量算法：典型代表是RIP！

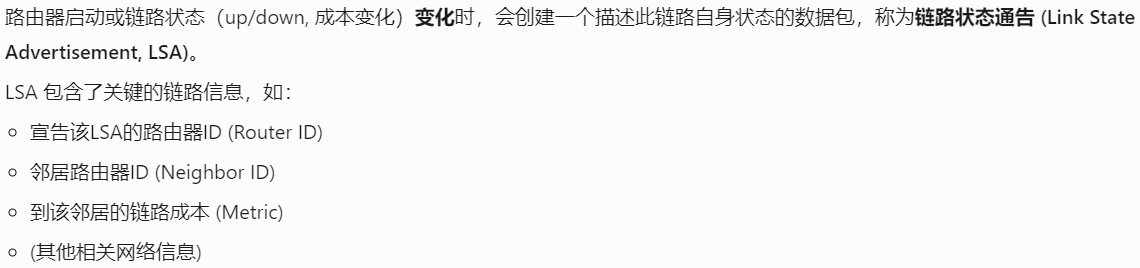


然后运行bellman-Ford



链路状态路由算法：典型代表是OSPF和IS-IS

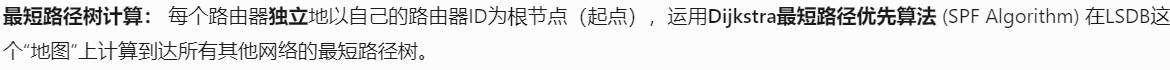
OSPF开放最短路径优先:内部网关协议（每个路由使用链路状态方法 计算到其他节点最短路径 记住最短路径集合） BGP：外部网关路由协议





通过泛洪，每个路由器不仅知道直连链路，还努力掌握整个网络区域（或一个区域）的完整拓扑信息。

然后：

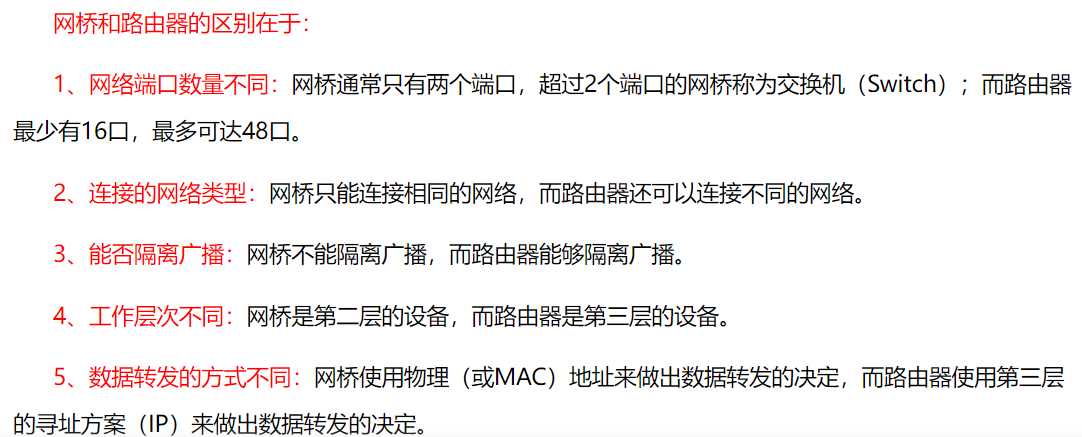


漏桶算法（leaky bucket algorithm）：发包等于注水 桶满了只能进入排队或者丢弃

令牌桶算法（token bucket）：发包等于倒水 有令牌才能发

**漏桶**一直是匀速，桶满就丢数据，并且**应对突发的能力差**，令牌桶桶满丢令牌，一直没有令牌就会丢数据，有一定应对突发能力

最大突发长度计算（length of maximum rate burst）——算时间！



**学习网桥——泛洪算法：**

举个例子理解：对于一个网桥有三个端口 初始网桥mac表为空

Host A (MAC地址 AA:AA:AA:AA:AA:AA) 连接在 Port 1 上

Host B (MAC地址 BB:BB:BB:BB:BB:BB) 连接在 Port 2 上

Host C (MAC地址 CC:CC:CC:CC:CC:CC) 连接在 Port 3 上

当A与C通信时，A数据从port1口进，C的mac未知，泛洪到B C B丢弃，C接收，网桥学习到：MAC地址 AA:AA:AA:AA:AA:AA对应是port1

网桥的直通式交换（完整接受前就开始转发 降低延迟 注意网络吞吐量没提高）和存储转发型

网络层：

虚电路（virtual circuit） 也称会话路由 事先建立连接 网络层负责可靠传输

无连接的数据报服务——尽力传 由传输层控制差错

拥塞控制之节流：  
**ECN显式拥塞通知** 路由器在它转发的任何数据包上打上印记，表明它正经历拥塞，接收方收到后，在它发送应答包时顺便告诉发送方降速

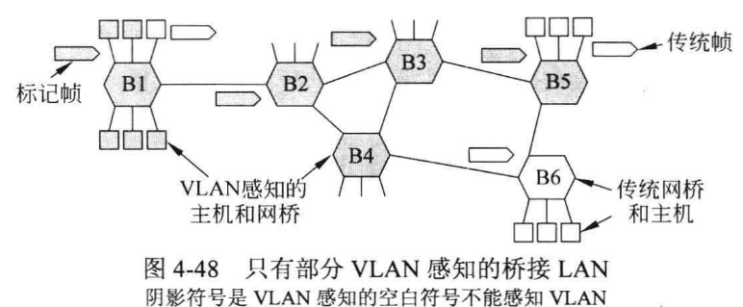
**负载脱落——随机早期检测RED：**

路由器上某条链路平均队列长度超阈值时，即将拥塞，随机丢包 传输方不给发送方回ACK，发送方发现丢包，传输协议将放慢V

ECN先告诉发送方降速而不是直接丢包，RED在buffer耗完前检测到拥塞就开始直接丢

VLAN：

为了使 VLAN 正常地运行，网桥必须建立配置表。这些配置表指明了通过哪些端口可以访问到哪些 VLAN。当一帧到来时，比如说来自灰色 VLAN，那么这帧必须被转发到所有标记为 G 的端口。这一条规则对于网桥不知道目的地位置的普通流量（即单播）以及组播和广播流量都适用。注意，一个端口可以标记为多种 VLAN 颜色。



注意上面的VLAN感知：存在一些计算机（和交换机）无法感知 VLAN，因此第一个 VLAN 感知的网桥在帧上添加一个 VLAN 字段，路径上的最后一个网桥把添加的 VLAN 字段删除。

包调度：

数据包分段（fragmentation）——**非透明分段（到一个路由就重组），透明分段（直到目的地才重组 IP方式）**

IP协议： 分组交换 **尽力而为的传输——无连接协议 结合的是TCP**

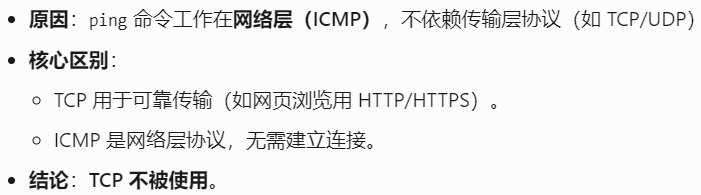
子网划分之无类域间路由CIDR （任意长度的前缀分配+后缀说明前缀位数：192.168.0.0/16）

公司局域网内ping百度，哪个协议不涉及？ARP ICMP DNS TCP选哪个

TCP不涉及 ping 命令直接使用 ICMP 协议——发送ICMP Echo Request（回显请求）包到目标地址，等待目标返回 ICMP Echo Reply（回显应答）

百度是域名，必须通过DNS解析为IP地址

由于目标IP不在本地子网，主机通过 ARP 协议解析网关路由器的 MAC 地址。（如果在同一子网，ARP 直接解析目标 IP 对应的 MAC 地址）

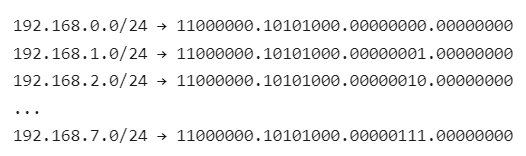


NAT：网络地址变换——解决IP短缺（内部分配私有IP，不允许出现在外网，数据包离开时通过一个NAT盒子转化为真实IP）**10 172 192（三类经典私有IP）**

ICMP协议：

网际控制报文协议 IP协议的辅助！**封装在IP内进行发送 ——主机分片 目的不可达 路由重定向（传递网络控制信息和错误报告）**

聚合路由——无需知道子网细节 目的是简化路由表



前 21 位比特相同：11000000.10101000.00000XXX 聚合为192.168.0.0/21

ARP地址解析——找MAC地址

主机发广播包到以太网络上（ARP请求分组！），对应IP地址的主机（目的主机）将源主机的IP地址和对应的物理地址添加进自身的ARP高速缓存映射表，然后，给源主机发送ARP响应报文(**注意，该报文是单播的**)告诉源主机我的物理地址

**DHCP——动态主机配置协议** discover包 offer包 request ack包 nak包……

计算机广播请求报文（discover），服务器返回空闲IP并返回（offer 主机的以太网地址由discover包带来） 离开网络收回IP



**路由表：特定路由（比如域名服务器）子网掩码全1 互联网属于默认路由 IP和掩码均全为0（IP包找不到存在的匹配路由，就转到默认路由）**

再次总结前面网络层部分：

首先是面向连接与无连接服务 虚电路与数据报

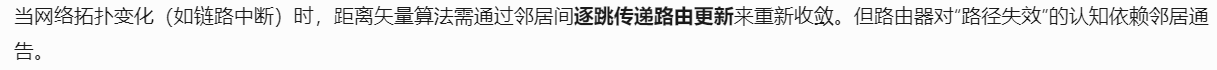
接着是路由算法——泛洪 距离矢量 链路状态路由 层次路由

距离矢量的例子：[计算机网络学习笔记 - 第五章 - xqmmcqs's blog](https://blog.xqmmcqs.com/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BD%91%E7%BB%9C%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0 - %E7%AC%AC%E4%BA%94%E7%AB%A0/)

无穷级数问题：

距离矢量的特点是好消息传得快，坏消息传的慢（坏消息靠邻居传）

无穷级数的本质：



链路状态路由：

每一个路由器必须完成以下的事情，算法才能正常工作

发现它的邻居节点，并了解其网络地址。

设置到每个邻居节点的距离或者成本度量值。

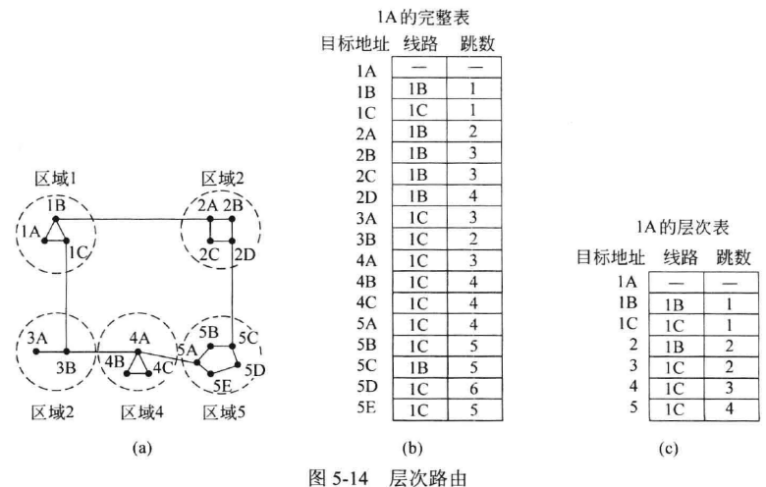
构造一个包含所有刚刚获知的链路信息包

将这个包发送给所有其他的路由器，并接收来自所有其他路由器的信息包。

计算出到每个其他路由器的最短路径实际上，算法将完整的拓扑结构分发给了每一个路由器。然后每个路由器运行 Dijkstra 算法就可以找出从本地到每一个其他路由器的最短路径

发现邻居——广播hello包 线路另一端的路由器应该返回一个应答说明自己是谁。这些名字必须是全局唯一的

层次路由：每个路由器知道如何将数据包路由到自己所在区域内的目标地址，但是对于其他区域的内部结构毫不知情 区域组成簇 簇组成区 区组成群



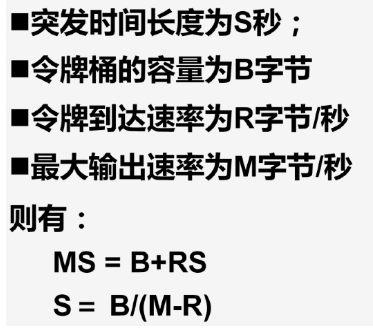
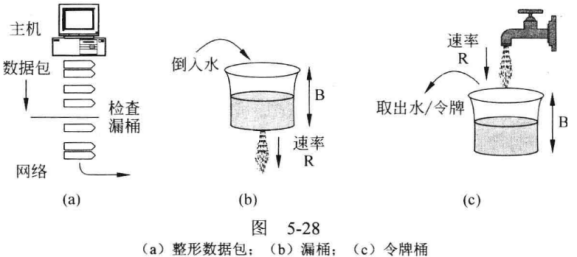
例如，考虑一个具有 720 个路由器的子网。如果没有分层，每个路由器需要 720 个路由表项；如果子网被分成 24 个区域，每个区域 30 个路由器，那么每个路由器只需要 30 个本地表项，加上 23 个远程表项，总共 53 个表项；如果采用三级层次结构，总共 8 个簇，每个簇包含 9 个区域每个区域 10 个路由器，那么，每个路由器需要 10 个表项用于记录本地路由器，8 个表项用于到同一簇内其他区域的路由，7 个表项用于远程的簇，总共 25 个表项

还有就是拥塞控制算法：包含流量感知路由，准入控制（漏桶和令牌桶），流量调节（抑制包，ECN（IP和TCP用此方法）），负载脱落（随机早期检测RED）

通知拥塞发送方的最直接方式是直接告诉发送方。在这种方法中，路由器选择一个被拥塞的数据包，给该数据包的源主机返回一个抑制包 (choke packet)。抑制包中的目标地址取自该拥塞数据包。同时，在原来的拥塞数据包上添加一个标记（设置头部中的一位），因而它在前行的路径上不会产生更多的抑制包。 抑制包的相关策略——逐跳后压

为了确定何时开始丢弃数据包，路由器要维护一个运行队列长度的平均值。当某条链路上的平均队列长度超过某个阈值时，该链路就被认为即将拥塞，因此路由器随机丢弃一小部分数据包。随机选择丢弃的数据包使得快速发送方发现丢包的可能性更大

还有就是服务质量：包含应用需求，流量整形（服务等级约定：只要顾客履行了约定中的义务，服务提供者就承诺按时送包到目的 流量监管：超出约定模式之外的数据包可能会被丢弃，或者被打上低优先级标记 漏桶（每个主机连到网络的接口处）和令牌桶），包调度



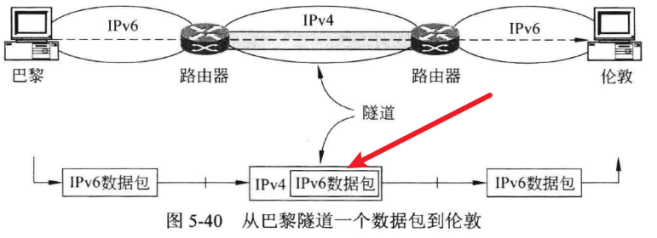
显然令牌桶允许突发

从一个源端发到一个接收方的数据包流称为一个流

服务质量QoS——带宽、延迟、抖动（数据包到达时间的变化 方差）和丢失

还有就是网络互联：

隧道技术，自治系统（AS），数据包分段（透明 不透明）



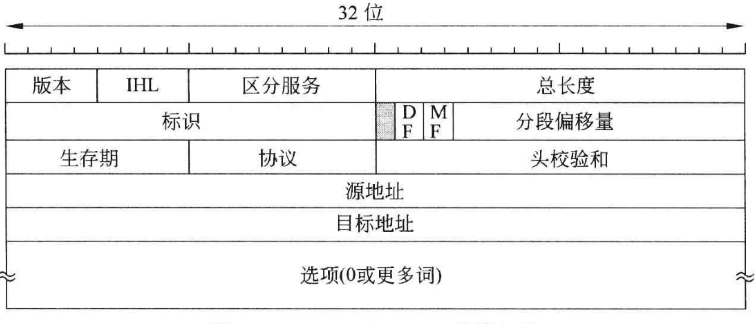
数据包长度受很多因素限制（硬件，协议，操作系统……）——路径最大传输单元 MTU

MTU发现：每个 IP 数据包发出时在它的头设置一个比特，指示不允许对该数据包实施分段操作。如果一个路由器接收的数据包太大，它就生成一个报错数据包并发送给源端，然后丢弃该包。当源端收到报错数据包，它就使用报错数据包携带的信息重新将出错数据包分段

最后是internet的网络层：包含IPV4 IPV6 控制协议（ICMP ARP DHCP） OSPF 和 BGP

IPv4协议：

头由一个 20 字节的定长部分和一个可选的变长部分组成



IHL 字段指明了头到底有多长（以 32 位字长度为单位）IHL 的最小值为 5，这表明头没有可选项。该 4 位字段的最大值为 15

区分服务：用来区分不同的服务种类。（加速服务和确保服务，后两位是显示拥塞信息）

标识：让目标主机确定一个新到达的分段属于哪一个数据报

**DF：不分段 MF：更多的段标志**（除了最后一段，其他都要设置）

分段偏移量：指明了该段在当前数据报中的位置。除了数据报的最后一个段以外，其他所有段的长度必须是 8 字节的倍数（**即单位是8字节**）

**强调：网关就是路由器，网关是路由器的旧称**！

强调：一个 IP 地址并不真正指向一台主机，而是指向一个网络接口，所以如果一台主机在两个网络上，它必须有两个 IP 地址。

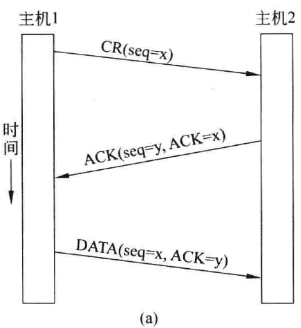
IP地址：ABCD几类 子网掩码 CIDR路由聚合 NAT

NAT 的基本思想是 ISP 为每个家庭或每个公司分配一个 IP 地址（或者，最多分配少量的 IP 地址），用这个 P 地址来传输 Internet 流量。在客户网络内部，每台计算机有唯一的 IP 地址，该地址主要用来路由内部流量 NAT盒子负责转换

ICMP：主要应用是ping和tracert

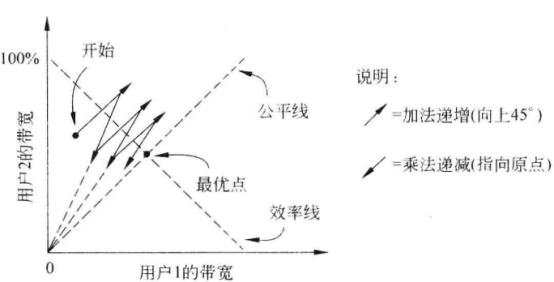
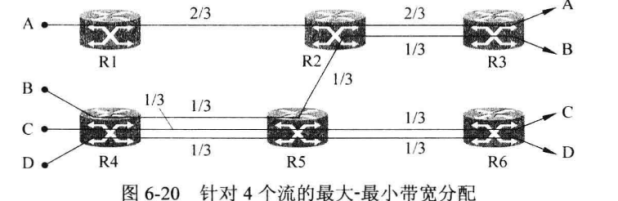
传输层：

三次握手：



非对称释放连接是电话系统的工作方式：当一方挂机后，连接就被中断了。对称释放连接是把连接看成两个独立的单向连接，要求单独释放每个单向连接。（非对称可能导致数据丢失）

拥塞控制——最大最小公平（资源分配向量的最小分量的值最大？）



一种直观的思考方式是想象所有的流从速率零开始，然后缓慢地增加速率。当任何一个流的速率遇到瓶颈，就停止该流的速率增加；所有其他的流继续加各自的速率，平等共享可用容量，直到它们也达到各自的"瓶颈"。

拥塞控制之后如何调节发送速率？——

AIMD 是 TCP 采用的拥塞控制法则 加法递增乘法递减

唯一的其他组合，即乘法递增和加法递减将会偏离最佳点

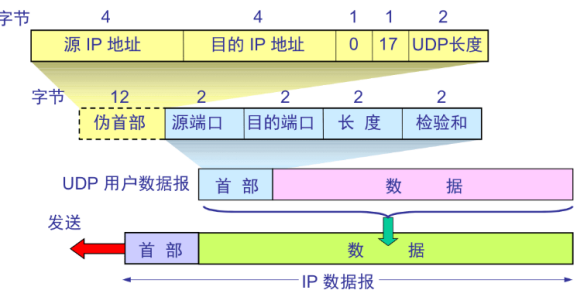
显式拥塞协议（XCP）——告诉源端以什么速度发

TCP的显示拥塞通知ECN——只告诉源端降速 不告诉降多少

UDP协议：**应用：DNS DHCP RIP**

没有流量控制 差错控制 重传坏段 支持广播和组播

使用端口（port）——为IP提供接口



（搭载IPv4的IP数据）

采用 UDP 而不是原始 IP 的最主要价值在于增加了源端口和目标端口。如果没有端口字段，传输层将无从知道如何处理每个入境数据包

“它只是提供了一个与 IP 协议的接口，并在此接口上增加通过端口号复用多个进程的功能，以及可选的端到端错误检功能。这就是 UDP 所做的一切。”

TCP：  
Socket address：IP address+port（知名端口http：80）

滑动窗口（发送窗口sw和接收窗口rw 允许连续发送多个报文段）

发送方已发Pkt1（1-1000），收到Ack=1001（期待的下一个）

接收方累积确认，提交一批

拥塞窗口是决定任何时候可以发出的字节数的因素之一。**拥塞窗口由发送方维护**，是阻止发送方和接收方之间的链路因流量过多而过载的一种手段。这不应与发送方维护的滑动窗口相混淆，滑动窗口的存在是为了防止接收方过载。拥塞窗口是通过估计链路上有多少拥塞来计算的。

拥塞——超时重传？平滑往返时间SRTT：SRTT=a\*SRTT+（1-a）\*R a：平滑因子 R：本次测量确认花费时间

拥塞控制的几种方法：

慢启动算法是在TCP连接建立时，发送方初始的数据传输速率较低，然后逐渐增加发送方的数据传输速率，直到网络出现拥塞为止。（一个往返时间加一倍 一旦有丢包被检测到，阈值降为一半，重置慢启动 超过阈值每个RTT线性+1：拥塞避免）

快重传算法是当接收方收到**重复**的数据包时，会立即发送一个重复确认（不等超时），以通知发送方有数据包丢失，从而使发送方能够更快地重传丢失的数据包。

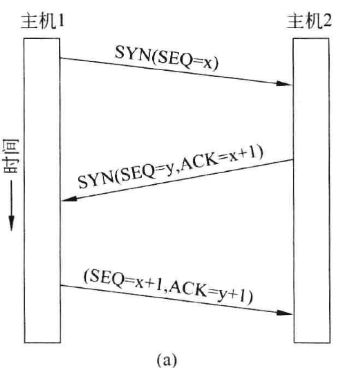
快恢复算法是**在接收到重复确认**后，发送方将拥塞窗口减半，然后继续进行拥塞避免算法，以减少网络拥塞的影响。

**TCP 发送窗口不能大于接收窗口和拥塞窗口。**

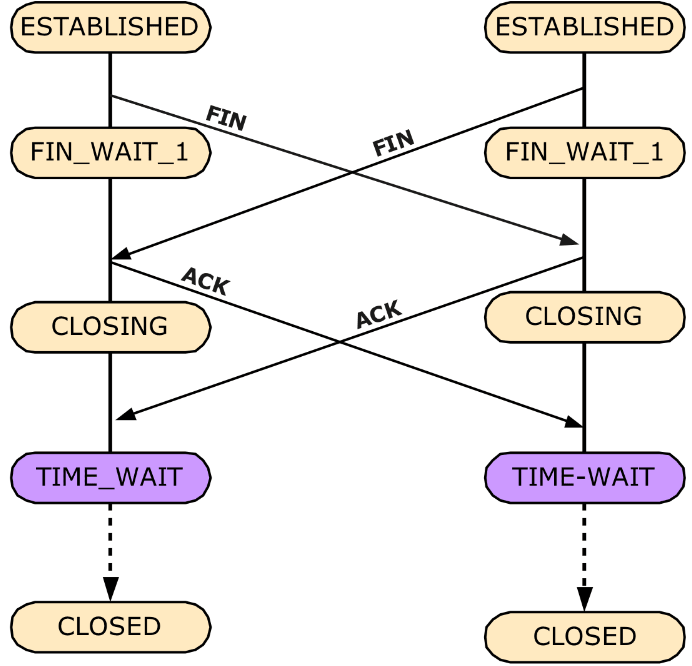
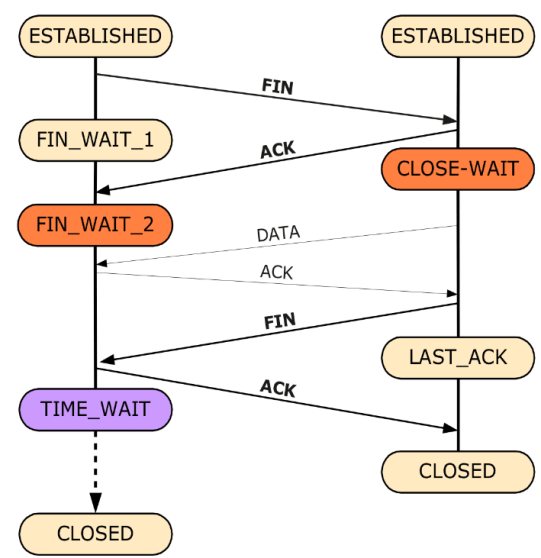
TCP不保留上层消息的边界（数据会被拆成多个段 然后重组）

UDP定位是：极简的传输层通道，仅提供基础寻址（端口）+校验和功能：将整条消息作为完整的数据报发送

TCP连接都是全双工且点到点的 TCP头：

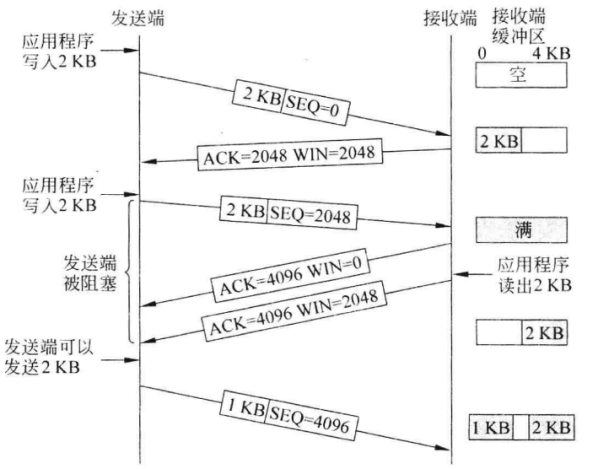


TCP的连接建立：上右图 SYN：建立连接的控制标志位（同步双方的初始序列号以建立可靠连接）



TCP的连接释放：每个单工连接的释放彼此独立 另一方还可继续发数据

为了避免两军对垒问题，需要使用计时器。如果在两倍于最大数据包生存期内，针对 FIN 的响应没有出现，那么 FIN 的发送端直接释放连接。

TCP滑动窗口：  


形象的图说明：发送方数据+起始序列号 接收方ack末尾+剩余窗口大小

当窗口变为 0 时，发送端不能如通常那样发送段了，但这里有两种意外情形。第一，紧急数据仍可以发送，比如，允许用户杀掉远程机器上运行的某一个进程。第二，发送端 可以发送一个 1 字节的段，以便强制接收端重新宣告下一个期望的字节和窗口大小。这种数据包称为窗口探测 (window probe)。TCP 标准明确地提供了这个选项，来防止窗口更新数据包丢失后发生死锁

TCP的一些问题：  
在最差的情况下，当一个字符到达发送端的 TCP 实体，TCP 创建一个 21 字节的 TCP 段，并将它交给 P 组成一个 41 字节的 P 数据报，然后发送出去：在接收端，TCP 立即发送一个 40 字节的确认 (20 字节的 TCP 头加上 20 字节的 IP 头）。以后，当远程终端读取了这个字节之后，TCP 发送一个窗口更新段，它将窗口向前移动 1 个字节。这个数据包也是 40 字节长。最后，当远程终端处理了该字符以后，它发送一个 41 字节的数据包作为该字符的回显。总共累计起来，对于每次敲入的字符，需要使用 162 字节的带宽，并发送 4 个数据段。

所以解决办法有延迟确认：短暂延迟确认（ACK）的发送，期望在此期间应用层能生成待发送数据，从而将ACK“搭载”到数据包中（即“捎带确认”），避免发送纯ACK造成的带宽浪费

Nagle算法：

数据每次发送多个小数据，发送端只是发送第一个到达的数据字节，缓存后面的包，直到发出去的包被确认，然后将所有缓冲字节放在一个TCP段中发送，然后又开始缓冲……

clark：低能窗口综合征——接收端一次接受一个字节就满了…… 强制接收端等待一段时间，等待有一定数量可用空间后才通告对方（可以处理MSS 或者一半被清空）

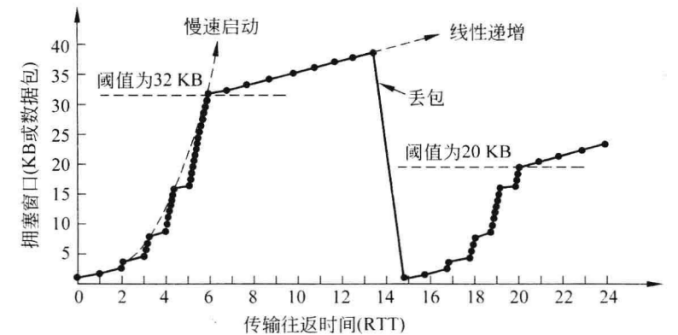
TCP的计时器管理：

 TCP 实体发出一个段时，它同时启动个重传计时器。如果在该计时器超时前该段被确认，则计时器被停止。另一方面，如果在确认到来之前计时器超时，则段被重传（并且该计时器被重新启动

动态的jacobson算法——前面的公式 ，一般α取7/8 TCP用2RTT

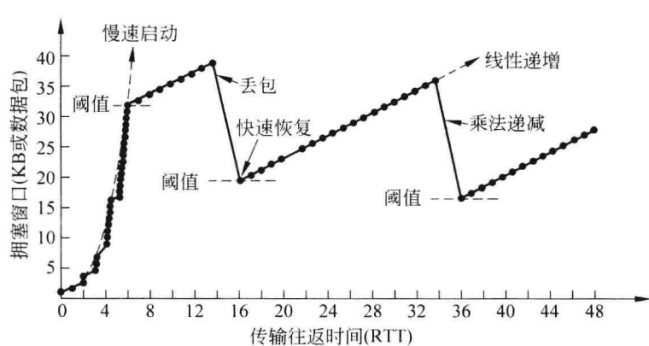
慢启动——慢启动阈值

TCP 以慢速启动方式不断增加拥塞窗口，直到发生超时，或者拥塞窗口超过该阈值（或接收端的窗口为满）



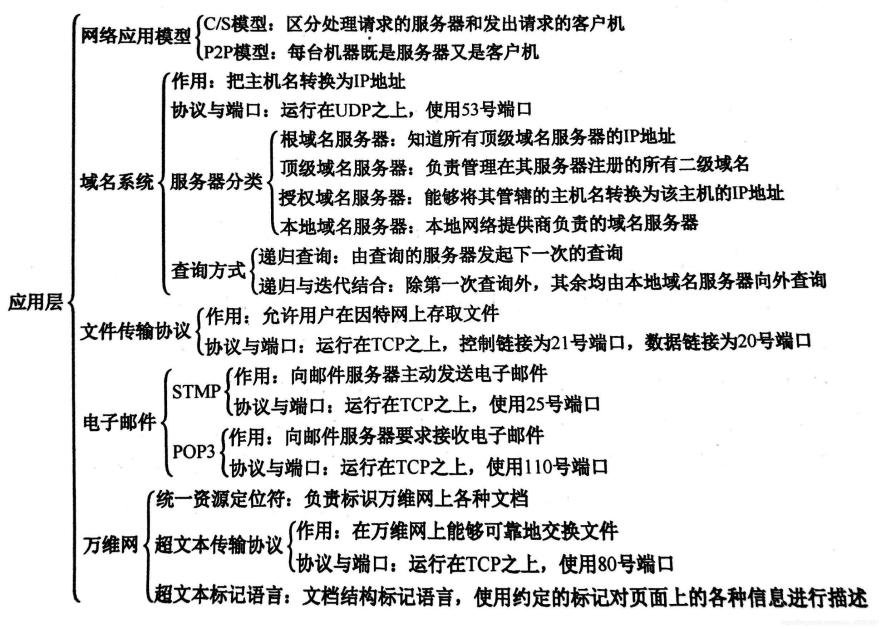
快速恢复：传统TCP中，一旦检测到丢包（超时或重复ACK），发送方会将拥塞窗口cwnd重置为1MSS（最大报文段大小），即单个丢包导致窗口骤降，吞吐量断崖式下跌

快恢复的前置条件是快重传，当发送方连续收到三个重复的ACK时，推断某个包丢失（而非网络严重拥塞），此时立即重传包（不等待超时），并进入快速恢复状态



**如果发生超时，那么慢启动（网络判断为严重拥塞），如果收到连续三个重复ACK（即丢包），发送快重传和快恢复**

应用层：



Client/server的C/S范式 和peer-peer的p2p范式

DNS：domain name system 域名和IP地址的映射

域名的解析过程：当需要把主机名解析成IP地址时，应用进程调用解析程序，并成为DNS的一个客户，把待解析的域名放在DNS请求报文中，以**UDP用户数据报**方式发给**本地域名服务器**，本地域名服务器查找域名后，把对应的IP地址放在回答报文中返回，应用进程获得目的主机的IP地址后即可进行通信（注意是以UDP数据报的形式）

 DNS使大多数名字都在本地进行解析，仅少量解析需要互联网上通信

再次强调本地域名服务器（UDP 端口53）服务于客户端当前网络环境的递归解析服务器

树状结构命名：

任何一个连接在互联网上的主机或路由器都有一个唯一的层次结构名字，即域名

域是名字空间中一个可被管理的划分，域还可以划分为子域，而子域还可继续划分

**域名的组成：由标号序列组成，各标号之间用点隔开**

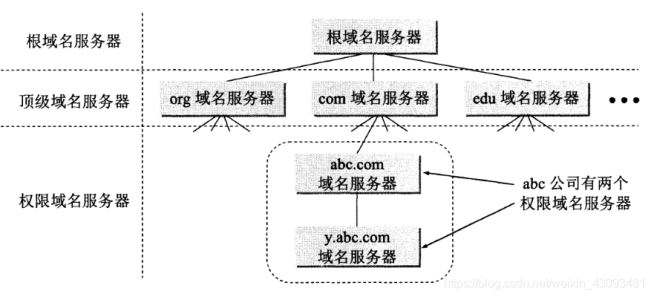
强调：**级别低的域名写在左边，级别高的写在右边**

**顶级域名——国家的（cn us） 通用顶级域名（com net org edu） 还有基础结构域名arpa用于反向域名解析**

**二级域名包括类别域名ac、com、edu、gov还有行政区域名bj、js等**

**三级，四级……**

服务器也是树状结构：



强调：所有根域名服务器都知道所有顶级域名服务器的域名和IP地址。**任何本地域名服务器只要自己无法解析，就首先求助于根域名服务器（注意是求助根！不是求助上一级）**

本地先求助根！！！（根识别域名后面的顶级域名来找要用的）

顶级域名服务器：负责管理在该顶级域名服务器注册的所有二级域名。当收到DNS查询请求时，就给出相应的回答

权限域名服务器：负责一个区的域名服务器。当权限域名服务器还不能给出回答时，就会告诉发出查询请求的DNS客户，下一步应找哪一个权限域名服务器

两种方式——递归和迭代

**递归靠别人 迭代靠自己**

**递归：根域名服务器向顶级求助，顶级向权限求助……返回也是一样**

**迭代：本地服务器如果没有，查询根，根没有，根域名告诉他顶级，本地去找顶级，本地再去找权限**

电子邮件系统：

发送邮件协议SMTP（简单邮件传输协议） 接收邮件协议POP3（只有最后用户接收会用到它）

再次强调SMTP是应用层的！使用TCP可靠数据传输服务（显然不用不可靠的UDP）



MIME：通用因特网邮件扩充 使电子邮件系统可以支持图像、视频、多种语言等（把非ASCII的内容转为ASCII）

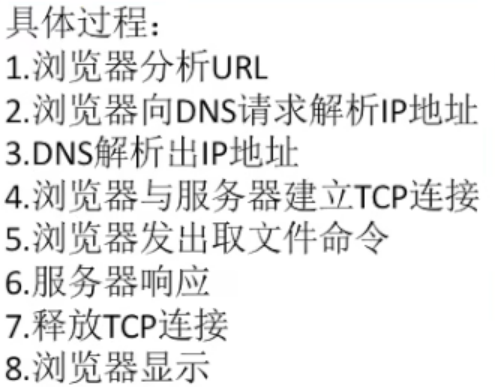
Web：万维网——它将因特网从只是很多数据网之一的地位提升为仅有的一个数据网

**http：Web的应用层协议是超文本传输协议（它是web的核心）。HTTP由两个程序实现:一个客户程序和一个服务器程序。**



HTTP同样使用TCP 且它是无状态的（第一次访问和第二次一样） 端口80

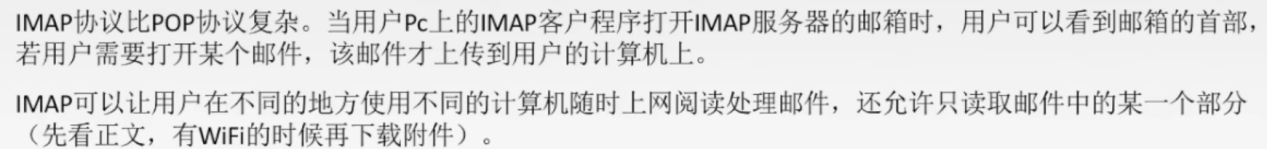
URL：统一资源定位符 域名+对象路径



SMTP一般**不使用中间邮件服务器发送邮件相连**。如果Bob的邮件服务器没有开机，该报文会保留在Alice 的邮件服务器上并等待进行新的尝试（在一个队列中），这意味着邮件并不在中间的某个邮件服务器存留 。

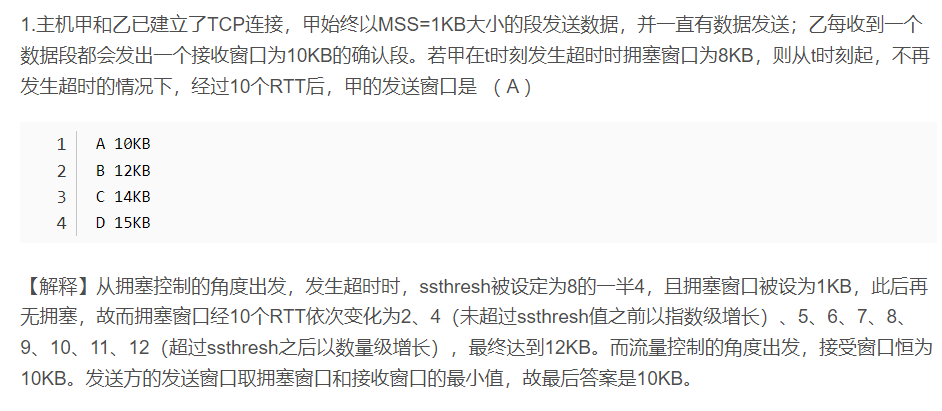


有IMAP：



其他补充：

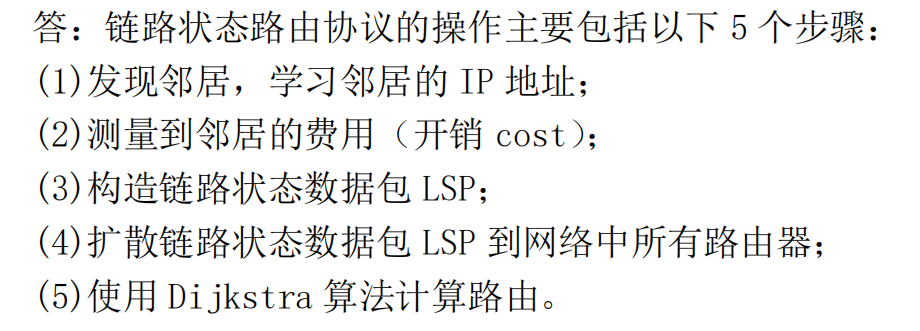
**奈氏准则（bps）c=2Hlog2v 香农定理 Hlog2（1+S/N） db=10log10S/N**



在 校 园 网 访 问 因特 网 ， 从 打 开 计 算 机 电 源 到 使用 命 令 ftp202.38.70.25 连通文件服务器的过程中，用到了 DHCP 协议分配 IP 地址、ARP 协议获得 MAC 地址、IP 协议发送和接收 IP 包，没有使用 ICMP协议

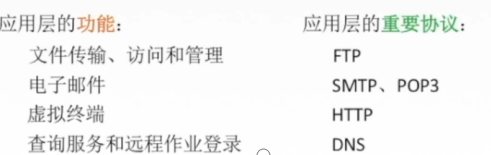
使用NAT（网络地址转换）：可以实现**私网内多台主机共享同一个因特网 IP 地址**访问因特网上的服务器的目的

强调以太网交换机工作在链路层



**协议用到了UDP协议：DHCP、DNS、RIP**

UDP没有拥塞控制，适合实时传输 且传输面向报文



**能隔离广播风暴的网络设备是 路由器 或 VLAN 交换机 网桥不能隔离**

强调：以太网——无连接不可靠服务（不事先建立连接，不确认，差错帧直接丢弃，纠错交给高层）

面向连接的服务是ATM 带确认的无连接服务是WiFi

静态划分信道（通信之前分）FDM TDM WDM CDM

动态分配信道（通信过程分）：

轮询访问介质访问控制（令牌传递协议）、随机访问介质访问控制（ALOHA CSMA CSMA/CD CSMA/CA）

可以过王道的习题