

计算机网络

qhy

2018 年 1 月 14 日

目录

1 介绍	3
2 参考模型	3
2.1 分层模型	3
2.2 ISO-OSI模型	4
2.3 TCP/IP参考模型	7
2.4 混合参考模型	8
3 物理层	9
3.1 傅里叶分析	9
3.2 传输介质	11
3.3 数字调制和复用	13
3.3.1 基带传输	13
3.3.2 通带传输	13
3.3.3 复用技术	14
3.4 物理层设备与冲突域	15
3.5 公共电话交换网络 PSTN	15
4 数据链路层	17
4.1 成帧	18
4.2 差错控制	18
4.2.1 纠错	19
4.2.2 检错	21
4.3 6个基本链路协议	22
4.3.1 无限制的单工协议	22
4.3.2 单工停-等协议	23
4.3.3 有噪声信道的单工协议	23
4.3.4 滑动窗口协议	23
4.3.5 协议4	23
4.3.6 协议5:回退n帧	25
4.3.7 协议6:选择重传协议	26
4.4 点到点协议 PPP协议	28

5 介质访问子层	29
5.1 ALOHA协议	30
5.1.1 纯ALOHA协议	30
5.1.2 分槽ALOHA协议	30
5.2 CSMA协议载波侦听多路访问协议	31
5.2.1 CSMA协议	31
5.2.2 CSMA/CD	31
5.3 其他多路访问协议	32
5.4 位图协议	32
6 IEEE800系列标准与以太网	32
6.1 快速以太网(100M)	36
7 交换机	36
7.1 千兆以太网(1000M)	36
8 网络层	46
9 123	57

1 介绍

WWW:world wide web,是信息资源的网络,资源、 资源标识 、 传输协议 三部分支撑www的运转做。

拓扑:信道的分布方式 ,常见的拓扑结构有: 总线型, 星型 ,环型 ,树型 ,网状

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

数字带宽: 指在单位时间内流经的信息总量,单位:bps , kbps , Mbps , Gbps(一般取 10^3 计算)

吞吐量:指实际的、可观测到的带宽

传输时间: $T = \frac{S}{BW}$ $T = \frac{S}{P}$

单位问题: $1B = 8b$, $1M = 1MB$ (省略了的是B)

点到点:两个机器的直接物理连接

端到端:信源机与信宿机之间的直接通信 ,好像拥有一条直接的线路

网络分类:

- 按传输技术:广播式网络, 点到点网络
- 按传输距离:个域网(PAN,1m) , 局域网(LAN,10m-1km) , 城域网(MAN,10km) , 广域网(WAN, 10-1000km) , 互联网(Internet,10000km)
- 按传输介质:有线网 , 无线网
- 按拓扑结构: 总线 , 环型 , 网状 , 星型

网关:连接异构网络, 提供必要的转换

互连网络:网络的集合 , 主要形式:被WAN连接起来

子网、网络、互联网:子网完成基础转发 , 子网和主机组成网络

2 参考模型

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

网络协议的三个要素:

- 语法: 数据与控制信息的结构或格式
- 语义: 控制信息, 指出完成的动作及响应
- 同步: 事件执行顺序的详细说明

2.1 分层模型

协议分层的优点:

- 各层工作独立, 层间通过接口联系 ,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他层
- 每层的实现技术可以不同 ,减少了实现的复杂度
- 易于维护 ,每层可以单独进行调试

- 便于标准化

分层的意义:

- 网络互连的自然需求
- 分而治之, 简化网络操作
- 提供即插即用的兼容性和不同厂商之间集成的标准
- 使工程师们可以专注于某一功能模块的设计和优化
- 防止不同区域网络之间的相互影响

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

典型分层模型:OSI七层模型 , TCP/IP(DoD)四层模型

每一层的功能:为它的上一层服务

实体Entity:每层中活动的元素

对等实体peer

层与层之间的关系:第n层是服务提供者, 第n+1层是服务独享, 即服务的消费者(下方的层是低层)

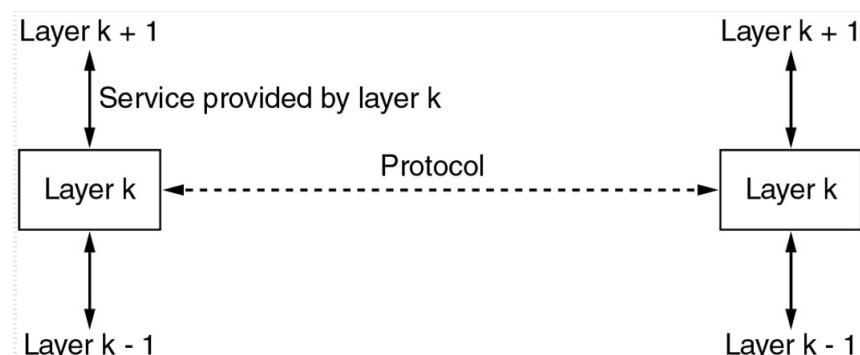


图 1: 层和协议的关系

协议数据单元:PDU , protocol data unit

2.2 ISO-OSI模型

ISO-OSI模型:

- 协议很少再使用, 但模型却很流行
- 每层都定义了标准
- 本身不是网络架构, 因为它本身并没有规定每层确切的服务和协议



图 2: 7层OSI模型

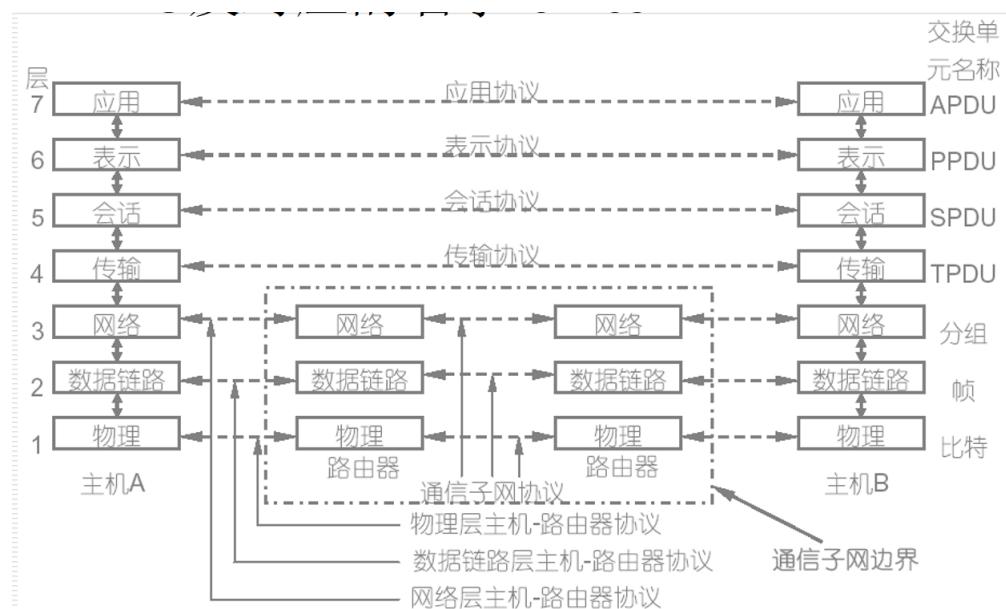


图 3: PDU及对应的名字

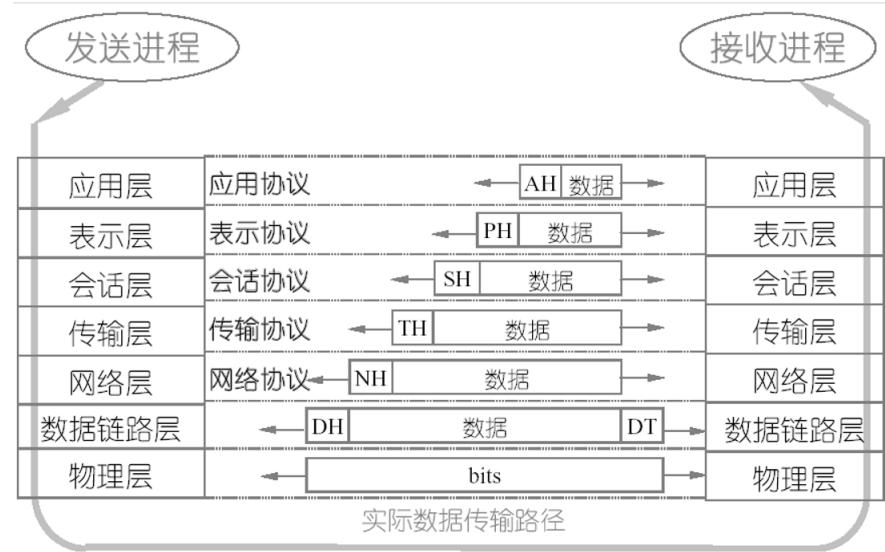


图 4: OSI参考模型上的数据流

Peer-to-Peer Communications

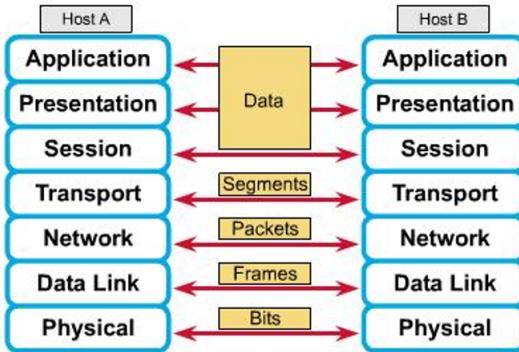


图 5: 对等通信、虚拟通信

为什么OSI参考模型没有占据主流?糟糕的时机, 糟糕的技术 ,糟糕的实现 ,糟糕的政策

- 糟糕的时机

当OSI协议出现的时候,与之竞争的TCP/IP协议已经被广泛地应用于大学和科研机构,没有厂家愿意支持第二个协议栈

- 糟糕的技术

会话层和表示层几乎为空, 数据链路层和网络层包含太多的内容

OSI模型以及相应的服务定义和协议都极其复杂,难以实现,操作起来也和低效

- 糟糕的实现

OSI模型和协议过于复杂,最初的实现不仅庞大而且笨拙,效率也很慢

- 糟糕的政策

政府官僚试图把技术不足的标准强加给那些实际开发计算机网络的可怜的研究人员和程序员

2.3 TCP/IP参考模型

TCP/IP模型:

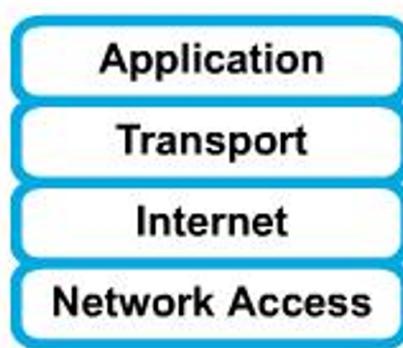


图 6: TCP/IP模型

- 互联网层

与OSI中的网络层对应

该层定义了正式的分组格式和协议，即 IP协议，每个IP包的路由问题是通过互联网每个IP包自己寻径，到达顺序可能不相同

- 传输层

与OSI中的传输层对应

使源端和目的端主机的对等实体进行对话

制定了两个端到端的协议:TCP传输控制协议，用户数据报协议UDP

- 应用层

与OSI中的上三层相对应

该层包括:telnet标准终端仿真协议，ftp文件传输协议，smtp简单邮件传输协议，域名解析服务DNS

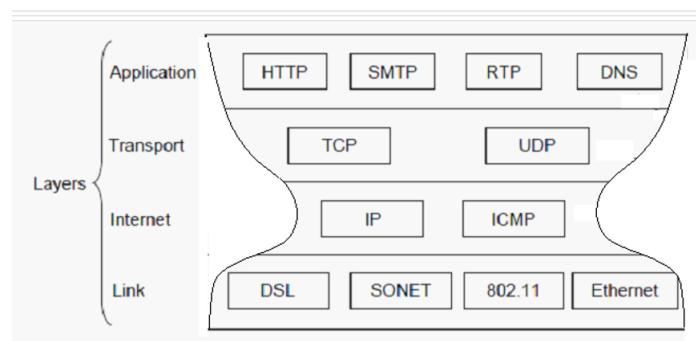


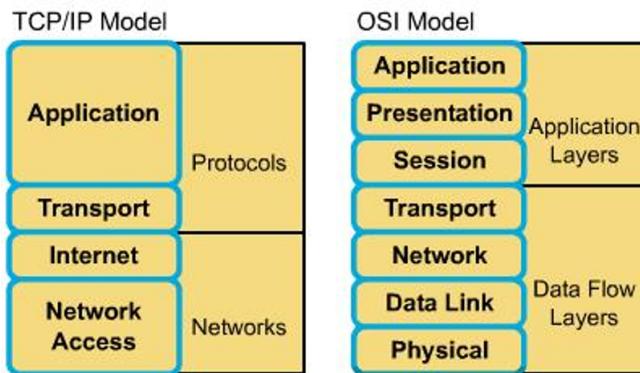
图 7: TCP/IP参考模型中的协议

TCP/IP参考模型和协议的缺点:

- 没有区分服务、接口和协议的概念
- 不是通用的模型

- 主机至网络层不是常规意义上的层
- 没有区分物理层和数据链路层
- 有些协议的实现比较草率

TCP/IP参考模型和OSI模型的对比:



相同点:

- 都分层
- 都有应用层,尽管他们提供的服务不同
- 使用的分组交换而不是电路交换技术

不同点:

- TCP/IP将表示层和会话层包含到了应用层
- TCP/IP将OSI的数据链路层和物理层包括到了一层中
- TCP/IP更简洁, 但OSI更易开发和排除故障
- TCP/IP在实践中产生

2.4 混合参考模型

计算机网络书中的参考模型:

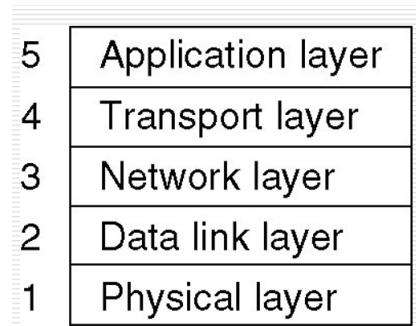


图 8: 混合参考模型

3 物理层

物理层上信号的传输: 信号

有两类: 模拟信号;数字信号

物理层的功能 :在两个网络设备之间提供透明的比特流传输

物理层的4个特性:机械特性 ,电气特性, 功能特性,规程特性

3.1 傅里叶分析

傅里叶级数: 任何正常周期为T的函数 $g(t)$, 都可由(无限个)正弦和余弦函数合成
任何信号的传输都可理解为以傅里叶级数的形式传递

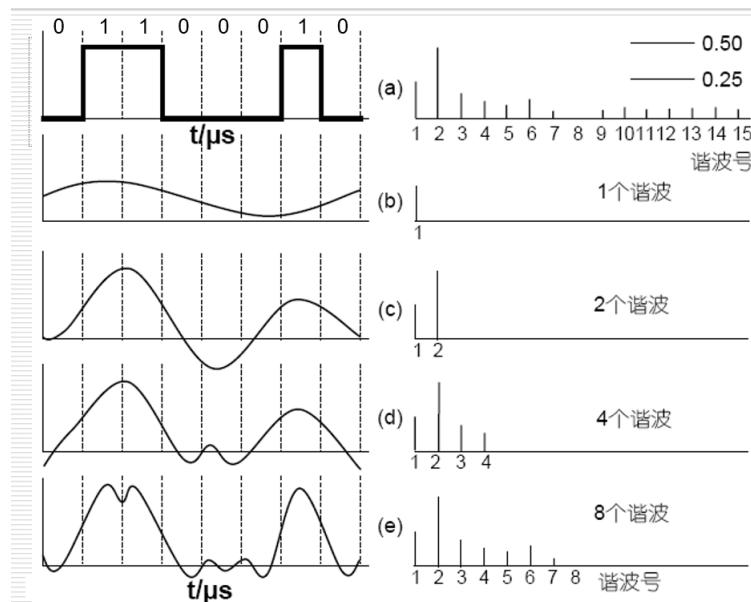


图 9: 谐波数越高, 传输质量越好

如果每个傅里叶级数的信号分量被等量衰减, 则合成分量, 振幅有所衰减, 基本形状不变

但, 所有的传输设备对不同傅里叶分量的衰减并不相同 ,因此会导致信号变形(失真)

信号在传输的过程中, 接收方接受到的信号可能是衰减和变形的 (失真)

一般来说, 从 $0 - f_c$ 这一频段, 振幅在传输过程不会明显衰减 ,则 f_c 称为 截止频率

(物理)带宽: 传输工程中振幅不会明显瞬间的频率范围, 是一种 物理特性, 通常取决于介质的材料构成、厚度、长度等^{1 2}

物理带宽和数字带宽的关系是?????????????????

如果比特率是 b bps, 传输8bit需要的时间
 T 是 $8/b$ 秒, 则第一次谐波的频率 (基本频率) 是 $b/8$ Hz。

图 10: 频率是时间的倒数

¹数字带宽: 指在单位时间内流经的信息总量, 单位:bps , kbps , Mbps , Gbps(一般取 10^3 计算)

²吞吐量:指实际的、可观测到的带宽

信道的最大数据传输速率:奈奎斯特定理(理想信道, 无噪声信道) , 香农定理(有噪声信道)

- 奈奎斯特 Nquist 定理

在无噪声信道中, 当带宽为 $B\text{Hz}$, 信号电平为 V 级, 则:

最大传输速率为 $2B \log_2 V \text{bps}$

其中, V 为信号的电平级数 , 在二进制中 , 一位表示 0 或 1 , V 级可以表示 $\log_2 V$ 位, 而对于带宽 B 最多有 $2B$ 次采样³⁴

- 香农 Shannon 定理

在噪声信道中, 带宽为 $B\text{Hz}$, 信噪比为 $\frac{S}{N}$, 则

最大传输速率为: $B \log_2(1 + \frac{S}{N}) \text{bps}$

很多时候, 噪声使用分贝 dB 表示: $10 \log_{10} \frac{S}{N} \text{db}$ ⁵

设某条信道的带宽为 **3000hz**, 信噪比为
30db, 求其最大数据传输速率

■ 解: 由信噪比 **30db** 知: $S/N = 1000$

再根据香农定理: 最大传输速率 = $3000 * \log_2 (1 + 1000) = 30000 \text{bps}$

□ 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

□ 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
_____ kbps

128kbps, about 40kbps

³奈奎斯特定理: 关于 $2B$ 次的一个理解: 假设一个正弦波, 采样点的值为正表示 1 , 为负表示 2 , 那么一个周期内最多能表示 2 个值 , 如果我们在一个周期内采样 3 次, 那么必定有一次是无用的, 即一个周期最多采样两次, 而已知频率为 B , 那么采样的频率最大只能为 $2B$, 否则解析信号结果和想要发送的结果将不一致

⁴ $2B$ 表示波特率

⁵香农定理: 噪声为 1, 信号为 $\frac{S}{N}$

如果一条信道的带宽在 3MHz 和 4MHz 之间，且信噪比是 24 分贝，问：

- (1) 信道的传输能力（最大传输速度）如何？
- (2) 为了达到这个传输能力，信号级别需要多少级？

$$24\text{dB} = 10 \times \log_{10} S/N$$

$$S/N \approx 251$$

(1) 最大传输速度是多少？

- $B = 4M - 3M = 1\text{MHz}$
- $S/N = 251$
- $C = 10^6 \times \log_2(1+251) \approx 8\text{Mbps}$

(2) 信号级别

- $C = 2B \log_2 V$
- $V = 16$

3.2 传输介质

- 磁介质

千万别低估一辆满载磁带的高速飞驰的货车！

- 如果一个标准磁带携带 200G 的数据
- 一个 $60 \times 60 \times 60 \text{ cm}$ 的盒子可以携带 1000 个这样的磁带，数据量达 200TB，或 1600Tb.
- 联邦快递可在 24 小时内，将盒子送达全美各地，传输速率可达 19Gbps；如果送达到 1 小时的目标，速率可达 400Gbps
- 运送成本：海运需 \$5000，约 3 美分 1GB。

所以，为什么我们不选择磁带呢？

图 11: 有趣的例子

- 双绞线：非屏蔽双绞线(UTP)，屏蔽双绞线(STP)

两个具有绝缘层的铜线按一定密度，逆时针方向绞合而成，一般绞距越小(紧)，则抵消效果越好，传输性能越好

非屏蔽：成本低；尺寸小；易于安装但易受干扰；传输距离性能受绞距影响；受绞距限制，功能有限，最大传输距离 100m(短)；10 100Mbps

STP 屏蔽双绞线：4 对线，每对线都有屏蔽层，最外面还有一层屏蔽层，成本高，安装不容易；最大距离 100m，10 1000Mbps

网屏式双绞线，保留最外的屏蔽层，除了屏蔽性能，其他是一样的

双绞线的线序：568B，568A（直通线，两根头的线序相同，交叉线，线序相反）

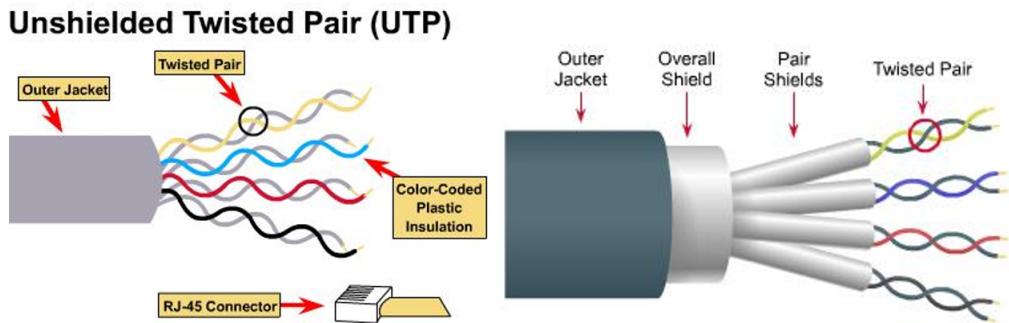


图 12: 双绞线

- 同轴电缆:由中心导体;绝缘层;网状导体;外部绝缘层组成

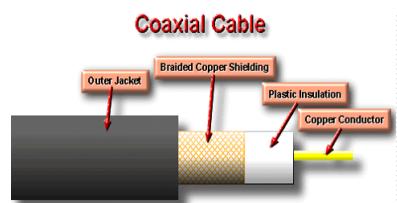
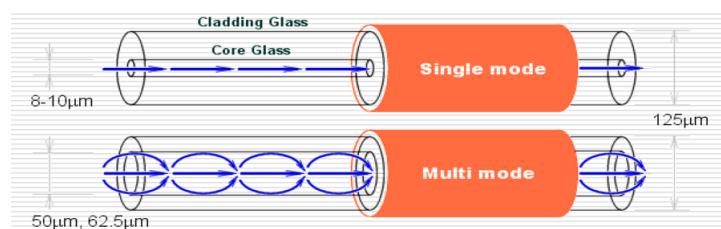


图 13: 同轴电缆

粗缆:最大传输距离500m,两端安装终接器,以保证电缆屏蔽层接地

细缆:最大传输距离185m,两头安装BNC头,接在T型连接器两端

- 电力线
- 光纤(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
3层, 两层玻璃纤维, 最外层为防护层
原理: 全反射, 因而损耗低
优点:重量轻;损耗低;不收电磁辐射干扰;传输频带,通信容量大
缺点:昂贵;易断裂
单模光纤:以单一模式传输,激光产生单束光(8-10 um), 纤芯细, 高带宽,长距离,运行波长为850nm或1300nm
多模光纤:以多个模式(多个入射角, 只要大于临界角度)同时传输,LED产生的多束光, 纤芯粗(50-62.5um), 低带宽, 短距离,运行波长为1310或1550nm



光纤连接: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%);熔合(几乎无损失)

光纤比铜线的特性:带宽高;距离远;耗损低;重量轻;无电磁干扰和射频干扰;防窃听;端口设备价格高

3.3 数字调制和复用

基带传输:直接将数据比特转换成信号,使用高低调位为标准描述0/1两个信号
 通带传输:通过调节信号的振幅;相位;频率来传输比特,使用振幅,频率,相位等为标准描述

3.3.1 基带传输

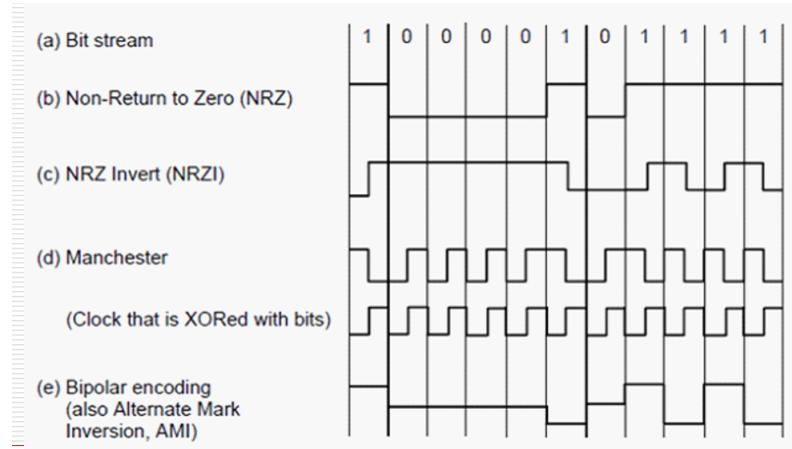


图 14: 基带传输

- (b) NRZ:高电平表示1,低电平表示0
- (c) 不归0逆转;NRZ Invert;NRZI:跳变表示1 , 不变表示0
- (d) Mncheter:高跳变到低表示1 ,低跳变到高表示0
- (e) 双极性编码:0始终用一个中间电压表示, 1使用高电压表示之后使用低电压表示,中间表示0 ,两边表示1 , 连续1 跳变

波特率(符号率;采样率):每秒信号变化的次数。

比特率(位传输率;数据传输率)与波特率的关系:

$$C = B \times \log_2 n$$

其中,C为比特率, B为波特率, n为电平数或线路的状态数

3.3.2 通带传输

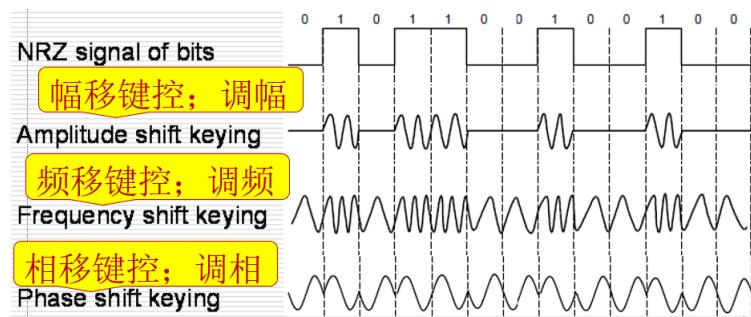


图 15: 通带传输

- 调幅, 使用不同振幅表示信号, 振幅为1表示1, 没有振幅表示0
- 调频, 使用不同频率载波信号表示振幅
- 调相: 使用不同相位表示0和1

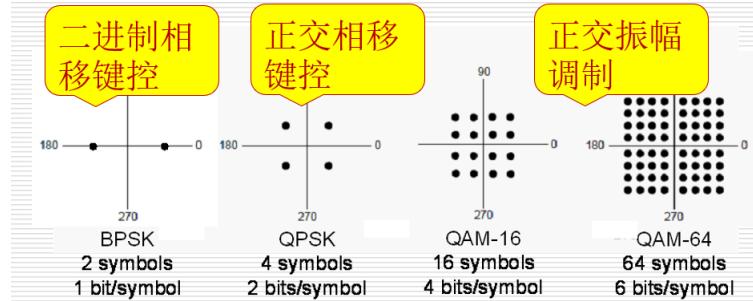


图 16: 信号星座

通常传输3种情况的综合应用:使用不同相位、波长表示不用的信号,这样接受一个符号,这个符号能表示的范围就变大,从而减少传输的时间。

比如每次接受一个波形,能可能表示4个情况,那么这个波形就能表示两个位(00,01,10,11)的一种

3.3.3 复用技术

复用技术:多个用户使用同一个通道。

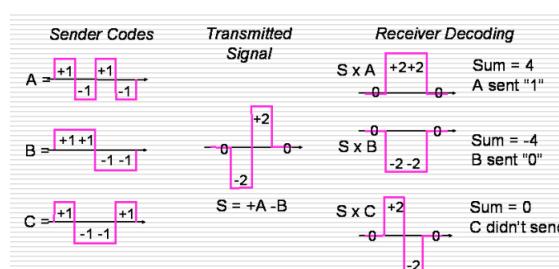
前面相位、波长用来表示数字,而频率和其他冗余信息则主要用于复用技术方面。

复用技术主要有以下几种类型:

- 频分多路复用(FDM): 不同频率直接叠加,最终经过滤波器分开
- 正交FDM OFDM: 普通的FDM每个频率段之间是不交的,而正交FDM有相交部分,但是也能区分开来
- 波分多路复用 WDN: 和FDM一样
- 时分多路复用: 时间上共享,一个接一个使用
- 统计时分多路复用 STDM: 时分多路复用的话,是每个用户一次分配到使用时间,不管有没有用到,而统计时分多路复用则是没有用到就不分配时间
- 码分多路复用 CDMA:

码元: 承载信息量的基本信号单位, 常使用时间间隔相同的符号来表示二进制数字每个用户拥有一个唯一的码片,每个码片相互正交(主要用于3G网络)

比如,传送4位信号,里面能包含3个用户发送的信息情况等。



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？

FDM: 每个用户分得带宽 $8M/10 = 800kbps$

■ 所以传输3000字节需要时间约： $(3000*8)/800kbps=0.03s=30ms$

TDM: 每个用户轮发数据量为
 $8M*1ms=8000b$ ；发送3000B (24000bit) 需
要轮3次，那么需要等待的时间为 $(3-1)*10=20ms$ ，发送剩下的8000b需要时间1ms，
共需21ms。

图 17: 也就是TDM不用考虑发送完之后那个周期的剩余时间(9ms)

3.4 物理层设备与冲突域

- 被动设备：接线板，插头，插座，电缆
- 主动设备：转发器（网卡的部分），中继器（再生信号：去噪和放大信号），集线器
- 冲突：同时发送数据,引起冲突

怎么防止冲突?减小冲突域

中继器和集线器会扩大冲突域

3.5 公共电话交换网络 PSTN

PSTN:public switched telephone network

主要构成及技术:本地回路(调制技术)，干线及复用技术，交换

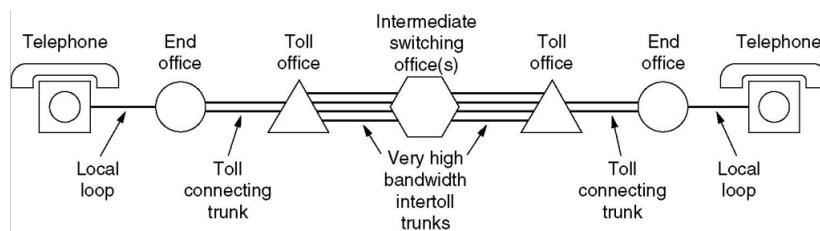


图 18: 一个中距离电话的典型路径

- 本地回路:模拟线路,调制技术,用户到交换局
- 干线:数字光纤,连接交换局,采用复用技术
- 交换局:话音接驳干线的场所

调制解调器:数字信号→模拟信号

56k的调制解调器P114

为什么使用**56 kbps** 的调制解调器? (采用V.90 标准)

- 电话线路的频率约是 **4000 Hz (300 ~ 3400 Hz)**.
- 采样率 = $2 \times 4000 = 8000$ sample/sec
- 每个码元传输 **8比特**, 其中的**1比特**用来控制错误, 传输数据速率是 $8000 \times 7 = 56,000$ bit/sec.

xDSL: modem带宽低, xDSL本地回路使用全部的1.1Mhz, 宽带能达到8M

TCM: 每次采样中,有一位用于纠错

SONET/SDH

SONET 帧结构: (Synchronous Transport Signal-1)

- 9(行) x 90(列) = **810字节**
- 头**3列** 用于系统管理信息
- 头**9行** 包括各种传输开销: 跨越不同链接, 指定语音信道, 连接帧等的开销。
- 其余的**87列** 包括用户数据, 即同步载荷封包 **SPE** (Synchronous Payload Envelope), 其中的**第1列** 又用于路径开销。
- **STS-1: $8000 \times 810 \times 8 = 51.85Mbps$**
- STS-N 帧是由N个STS-1基本帧构成的**

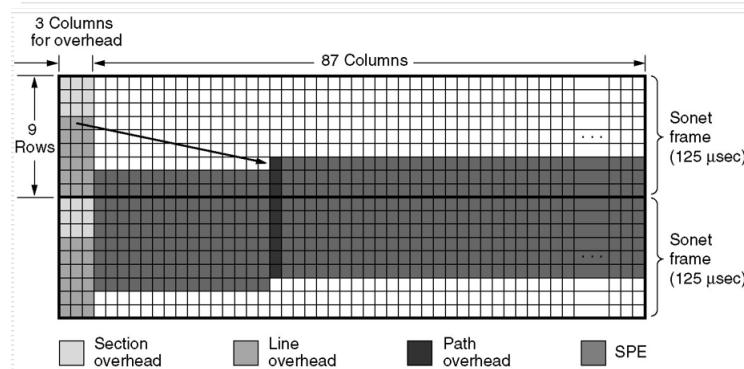


图 19: 干路复用技术 SONET

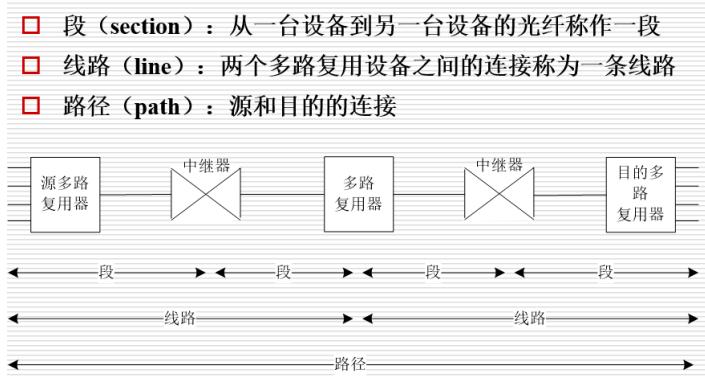


图 20: 段、线路、路径

除了北美和日本,其他国家使用E1系列线路:

- E1可以处理 **32** 条语音的复用: $32 \times 8 = 256\text{bits/frame}$
- 予以按采样率是每秒 **8000** 次

例如: **OC-1**

$$\text{总传输速率: } 8 \times (9 \times 90) \times 8000 = 51.84\text{M b/s}$$

$$\text{SPE: } 8 \times (9 \times 87) \times 8000 = 50.112\text{M b/s}$$

$$\text{用户数据: } 8 \times (9 \times 86) \times 8000 = 49.536\text{M b/s}$$

图 21: 8000是采样率;OC-12 指的是 Optimal Carrier , 结果是OC-1的数值×12

交换:

- 电路交换: 双方打通通道
- 报文交换: 同分组交换
- 分组交换: 每个分组独立寻径, 直接投放数据(分组交换的分组有大小限制,而报文没有)

4 数据链路层

数据链路层的作用:

- 为网络层提供服务,良好的服务接口
- 保证数据传输的有效、可靠:处理传输错误(差错检测和控制);流量控制(基于速率,基于反馈)

确认:接受方收到数据帧后,必须给发送方发回一个确认

面向连接:发送方和接收方在传输数据之前必须建立逻辑连接,传输结束后必须释放连接

服务种类:无确认无连接服务;有确认无连接服务;有确认的面向连接服务

无确认并非不可靠,可靠性由上层协议负责

4.1 成帧

数据链路层使用物理层提供的服务,物理层处理的是 位流 ,数据链路层处理的是 帧
将原始的位流分散到离散的帧中, 叫成帧, 成帧的方法有:

- 字符计数法

帧头为字节长度, 一旦出错, 后续完全错误, 无法恢复, 很少使用

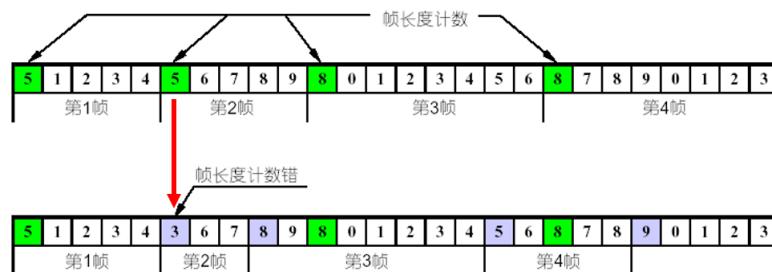


图 22: 字符计数法

- 带字节/字符填充的标志字节法

使用特殊的字节(flag byte)作为字节作为帧的开始, 需要定义转义字节来在数据中使用这个特殊的字节

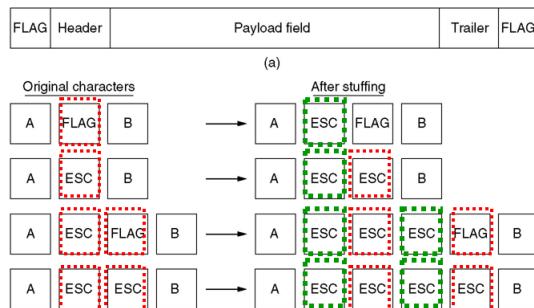


图 23: 字节填充的标志字节法

缺点:依赖于8位字符

- 比特填充的比特标志法 (零比特传输, 透明传输)

以特殊的位模式作为帧的开头, 比如01111110 (0+6个1+0), 如果要数据和帧头一样, 则在第5个1后面加个0, 这样就能保证数据一定不会出现6个1, 当出现5个1的时候, 自动把后面的0去掉

- 物理层编码违例法

在曼彻斯特编码张, 只有电压跳变是数据, 那么可以使用高高, 或者低低这样的电平作为帧的边界

使用跳变作为数据, 使用连续高/低电平作为帧的边界

4.2 差错控制

差错的类型 :单个错误;突发错误

通常利用处理单个错误的方法来应对突发错误

差错处理：纠错(主要用于无线网络中), 检错

- 纠错码：发现错误，从错误中回复正确的数据，主要用于无线网络
- 检错码：只能发现错误，不能从错误中恢复，但可采用重传

4.2.1 纠错

码字：包含数据位和校验位的n位单元

两个码字海明距离：两个码字不同位的数目

全部码字的海明距离：最小的海明距离

海明距离：如果海明距离为d，则一个码字需要发生 $d+1$ 位错误才能变成另一个码字

- 海明距离为 $d+1$ 的编码能检测出d位错误
- 海明距离为 $2d+1$ 的编码,能纠正d位错误(恢复成最近的海明吗)

纠一位错的海明码

纠一位错需要多少位冗余位？

冗余为r位,数据为m位, 纠一位错误:数据有 2^m 种可能,每条数据都有各自的 $m + r$ 个距离为1的错误的码,算上它自己,算上所有的数据可能,那么总共有 $(m + r + 1)2^m$,而总的数据有 2^{m+r} 种可能,那么要求 $(m + r + 1)2^m \leq 2^{m+r}$

得: $m + r + 1 \leq 2^r$

由此不等式,我们可以求出,在给定m的条件下,至少需要多少(r位)校验位

纠一位错的海明码：编号位2的幂的位都是数据位, 每一位表示包括自己在内的一些位集合奇偶值(编号从1开始)

那么这些位集合怎么取? 把编号按2的幂求和展开, 这一编号所代表的位, 就要参与这些2的幂对应的校验位的奇偶计算

检错: 校验位不对的就是了, 逐个检查每个校验位的奇偶性

纠错: 一个位出错, 会影响到所有它所在的集合对应校验位, 因此只要对出错的校验位的编号进行累加, 得到的编号, 就是出错的位的编号

优化：连续k个码字按行排列成矩阵, 按列发送, 注意, 列没有海明编码, 就是按列直接发送

当出现突发错误的时候, 发送的一列都出错了, 对于一行来说, 只有一位发生错误, 也能根据行的海明码进行纠错

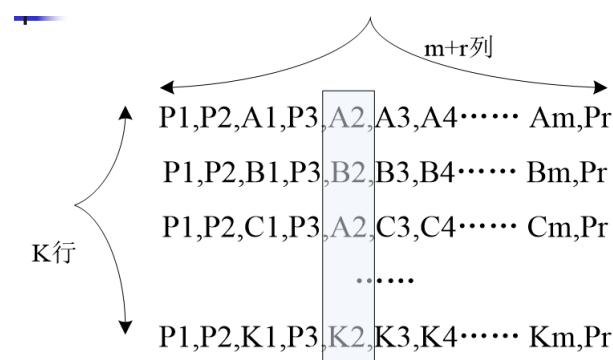


图 24: 优化

	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

图 25: 海明码例子1:采用偶校验

	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	-	1	0	0	-	0	0	1
检验位	1	0	-	1	-	-	-	1	-	-	-
海明码	1	0	1	1	1	0	0	1	0	0	1

图 26: 海明码例子2:采用偶校验

例如接收到码字为**00111000100**，校验各校验位：

- 第一位：**00111000100**，校验集合有3个1，错
- 第二位：**00111000100**，校验集合有1个1，错
- 第四位：**00111000100**，校验集合有2个1，对
- 第八位：**00111000100**，校验集合有1个1，错

累加出错位编号： **1+2+8=11**

可计算得其第**11**位出错，将该位由**0**改为**1**，即纠正得到正确结果：**00111000101**

图 27: 海明码纠错例子

原码字为：10101111，采用偶校验海明纠

1位错编码，请问编码后的码字是什么？ 编码后码字是：**101001001111**

采用上面这道题一样的编码，假设接收方收到

一个码字：**100110001100**

(**m=8,r=4**)，请问这个码字对还是错？如果错，正确的码字应该是什么？

解答：计数器累加：

1+2=3，所以，第**3**位出错，正确码字应该为：

101110001100

4.2.2 检错

检错码仅包含接收方检查出是否有错误的足够信息,如果物理层的错误率(铜线或光纤)足够低,采用检错和重传机制更加有效

奇偶校验：保证码字的1的个数是奇/偶数个，但只能检测出奇数个错误(海明距离为2,检1位错)

校验和：校验和通常是按N位码字来模拟模2和运算,发送将运算结果附加在数据报文尾部,作为校验位

- 比奇偶校验更好的检错性能
- 能检出高至N位的突发错
- 检错随机率 $1 - 2^{-N}$
- 易受系统错误干扰,比如,增加的“0”

RFC1071: computing the internet checksum

- (1) 待校验的相邻字节成对组成16比特的整数一行，按列从低位开始计算其模2和；并按位取反码，作为校验和取值。
- (2) 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。接收方：如果结果为全1，无错误
- 注意：如果某列的模2和有溢出，向高位进位，如果高位产生进位，循环向低位进位。

图 28: 互联网校验和计算文档

CRC循环冗余检错码: 任何一位k位的帧，可以看成是 **k-1** 位的多项式，比如101表示 $x^2 + x^0$ 设定一个生成多项式G(x)，G(x)是r阶，假设数据为M(x)，则发送 $x^r M(x) + R(x)$, R(x)为 $x^r M(x) / G(x)$ 的余数，从编码来看就是 $T(x) = [M(x), R(x)]$ 接收方收到的时候， $T(x) / G(x)$ ，能整除，则说明没有出错

注：上面提到的加减运算都是模二加减（异或），不进位也不借位

如一帧为**1101011011 (m=10)**

$$\text{即 } M(x) = x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$$

$$G(x) = x^4 + x + 1 \quad (r=4\text{阶})$$

$$T(x) = x^4 M(x) \quad (\text{相当于在原码字后加r个0})$$

$$= x^4(x^9 + x^8 + x^6 + x^4 + x^3 + x + 1)$$

$$= x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4$$

1	0	0	1	1											
1	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0
1	0	0	1	1			0	1	1	0	0	0	0	0	0
1	0	0	1	1			0	0	1	0	0	0	0	0	0
1	0	0	1	1			0	0	0	0	0	0	0	0	0
0	0	0	0	1			0	0	1	0	0	0	0	0	0
0	0	0	0	0			0	0	0	1	0	0	0	0	0
0	0	0	0	0			0	0	1	0	0	0	0	0	0
0	1	0	1	1			0	0	0	0	0	0	0	0	0
0	0	0	0	0			1	0	1	1	0	0	0	0	0
1	0	0	1	1			1	0	0	1	1	0	0	0	0
1	0	0	1	1			0	1	0	1	0	0	0	0	0
1	0	0	1	1			0	0	0	1	1	0	0	0	0
0	1	1	1	0			0	0	0	0	0	0	0	0	0
1	1	1	0	0			1	1	1	0	0	0	0	0	0

图 29: CRC码计算距离

例子: 多项式1101 , 待发送1111和1100, 编码之后是多少? 1111111和1100101

4.3 6个基本链路协议

几个假设:

- 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希望向机器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
- 从网络层拿到的数据是纯数据

4.3.1 无限制的单工协议

协议1的假设:

- 1. 数据单向传输
- 2. 收发双方的网络层都处于就绪状态, (随时待命)
- 3. 粗粝时间忽略不计 (瞬间完成)
- 4. 可用的缓存空间无限大 (无限空间)
- 5. 假设DLL之间的信道永远不会损坏或者丢失数据帧 (完美通道)

协议1:发送方不断发，收方不断收

4.3.2 单工停-等协议

协议2：解决如何避免收方被涌入的数据淹没,即取消“接收方允许无限量接受”的假设
考虑接收方的处理能力有限（缓存有限）

实际上是半双工协议，而不是单工协议

发送方发送之后等待接收方回复的哑帧才继续发下一个数据帧

4.3.3 有噪声信道的单工协议

协议3：信道不是完美的，可能丢失帧也可能接收到错误帧

- 1. 对正确帧的确认（确认帧丢失，发送方超时重发，接收方拒收（根据编号判断），但是重发确认帧）
- 2. 超时重传（可以防止死锁的产生，就是A发送B，等待B确认，结果帧丢失，A依旧等待B发送，变成双方互相等待）
- 3. 对帧进行编号（当确认帧丢失以后，发送方重发，而接收方接收之后还需要判断是重发的还是新的帧，因此需要编号）

小结：超时可能是数据帧丢失也可能是确认帧丢失

4.3.4 滑动窗口协议

如果发送端可连续发送一批数据帧,必须考虑接收端是否来得及接纳和处理这么多帧,这里提出了网络流量控制问题

流量控制机制：在接收方缓冲区达到一定一定量时,应及时通知发送方,暂停发送,等候通知

滑窗协议（协议4-6）

原理:当接收方收到帧之后,先核对是否是预期帧号,是则移动接收窗口,并返回确认帧;发送方收到应答帧,核对响应帧号,是则移动发送窗口

- 协议4: $n = 1$ 引出滑动窗口的技术
- 协议5-6: 滑动窗口对出错帧的两种应对措施
 - 全双工:双方可以互相发送提高效率
 - 捎带确认:确认帧放在接收方下次发送的帧,如果没有才发送哑帧
- 批发数据:多个数据帧同时发送(滑动窗口)

4.3.5 协议4

双方互相发送 + 捎带确认 + 窗口大小为1, $n = 1$

发送窗口：对应已经发送但还未被确认的帧的序号，当接收到确认之后窗口滑动，发送下一帧

接收窗口：对应期望接收的帧的序号，收到期望的帧，滑动窗口

异常：

- 1. 定时器设短了，能正常工作，但是会发送大量的重复帧，

- 2. 同时发送，能正常但重复（数据接收没问题，但是确认帧号不对，因此接收到的会重发一次数据）

信道利用率计算：信道传输速率 b bps, 帧大小 k bit, 来回时间 R s

则信道利用率： $\frac{k}{k+Rb}$ (实际发送数据 / 最大可发送数据)

发送过程：发送 k , 然后等待 R_s 确认

已知：

- 信道容量 **$b = 50 \text{ kbps}$**
- 传输延迟 **$R = 500 \text{ ms}$** (双程)
- 数据帧的长度 **$k = 1000 \text{ bit}$**
- 设接收方收到数据帧后马上回送确认短帧，没有延时

求：信道利用率

- 在源端发送数据帧过程需要的时间
 $T_f = k/b = 20 \text{ ms}$
- 从发送完毕到确认帧返回需要的时间 (双程延迟)
 $R = 500 \text{ ms}$
- 从开始发送到确认返回总共需要的时间
 $(T_f + R) = 20 + 500 = 520 \text{ ms}$
- 线路的利用率

$$T_f/(T_f+R) = 20/520 = 3.85\%$$

Or: $k / (k+bR) = 1000 / (1000 + 50 \text{ kbps} * 500 \text{ ms}) = 3.85\%$

提高信道利用率：管道化技术

滑动窗口长度增加到 W , 即一次发送 W 个数据帧

信道利用率为： $\frac{W*k}{k+RbW}$ (是利用发送一帧的空闲部分发送剩余 $W-1$ 帧, 因此分母是 k)

令上式=100%，可以得到信道利用率为100%下的 W

怎么寻找一合适的 W 的值？宽带延迟积： bd , 信道上的容量：一帧从发送方传输到接收期间可容纳的帧数量 $bd = \text{bandwidth} * \text{delay}$

窗口值： $w = 2*bd + 1$

2表示往返两个方向的时间，+1表示呆在网卡上的还没发送出去的一帧

36. 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. **10Mbps**
- B. **20Mbps**
- C. **80Mbps**
- D. **100Mbps**

设可达到的最大传输速率为 **x**，于是

$$1000f * 1000Bpf * 8 = xbps * 2 * 50ms / 1000ms$$

$$\rightarrow 8000000 = 8000 + 2 * 0.05x$$

$$\rightarrow x = \mathbf{80Mbps}$$

图 30: 2014考研题:最大发送1000个窗口，时间为50*2，求速度

那么，如果帧出错了，怎么处理？连续发送w个数据帧，其中一帧出错，但其后帧被成功发送
接收方策略：

- 丢弃错误的帧以及后续非期望接收的帧(协议5)
- 丢弃错帧,缓存后续正确接收帧(协议6)

对应发送方的重传策略：

- 缓存在发送窗口中的出错数据以及其后续的帧全部重发(协议5)
- 只重发错帧(协议6)

4.3.6 协议5:回退n帧

接收方策略:丢弃错误的帧以及后续非期望接收的帧(协议5)

对应发送方的重传策略:缓存在发送窗口中的出错数据以及其后续的帧全部重发(协议5)

- 接收窗口大小为1，再大也没用
- 对错误帧不确认
- 定义序列号seq的取值范围和滑动窗口长度W
- 发送方超时重传，从未被确认帧开始



接收窗口大小 : $W = MAX_SEQ$

异常: $W > MAX_SEQ$ 时 , 若 $MAX_SEQ = 7$, 对第0帧的确认 , 可能会被误判为对第8帧(编号也为0)的确认

- 发送方需要较大的缓冲区 , 以便重传
- 重传帧比较多 ,适合信号出错率较少的情况

4.3.7 协议6:选择重传协议

接收方策略:丢弃错帧,缓存后续正确接收帧(协议6)

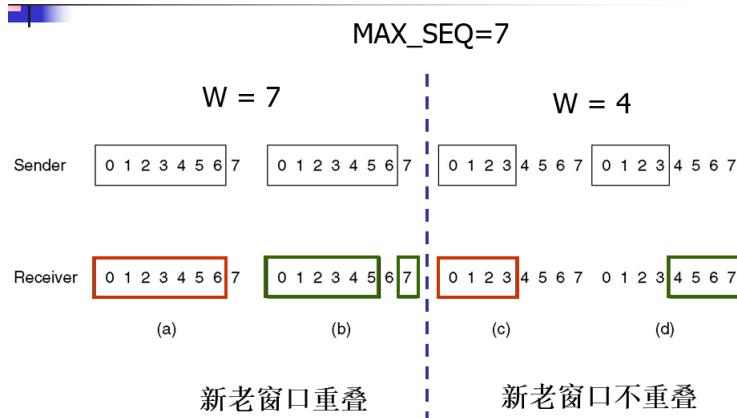
对应发送方的重传策略:只重发错帧(协议6)

- 接收窗口只存储差错帧后继的所有正确帧
- 发送方只重传差错帧
- 接收方接收重传帧 , 按正确顺序将分组提交网络
- 每接收到一个非期望帧 , 发送一NAK , 加速出错帧的重传 , 使发送方不再等待超时重传



图 31: 选择重传协议的工作原理分析

接收窗口大小: $W = (MAX_SEQ + 1)/2$, 序列号从 $0 \rightarrow MAX_SEQ$, 发送窗口和接收窗口同大
异常 : $W > (MAX_SEQ + 1)/2$ 时 , 重传的出发有两种可能,一个是接收到NAK , 一个超时 , 触发超时可能是因为最后一帧的确认帧的丢失 , 而此时接收方已经把帧提交给上一层 , 此时等待接受新的帧, 而对于发送方来说,它要重发旧帧, 由于旧帧和待接受的新帧编号一致 , 就导致了错误。即错误的原因是由于新的窗口的编号和旧的窗口的编号有重叠, 主要保证编号不重叠就不会发生这个问题 ,因此滑动窗口要取 $W > (MAX_SEQ + 1)/2$



- 接收方需要较大的缓冲区，以便按正确顺序将分组提交给网络层
- 重传帧数少，适于信道质量不好的情况

采用累计确认，当帧好n的确认到底时，其之前的帧也成功接收（这个是对发送方而言，如果接收到n的确认，n-1的确认丢失了也没关系）（同时会发接收到的最后一帧的确认，这句感觉是废话啊，不是收到就确认了吗？其实不是，丢弃一个就确认一次，因此不是接收完就确认，而是接收到下一帧错误的时候才确认，那么正常情况下hi怎么确认的呀？正常是接收到最大的序列号就确认）

- **One-Bit sliding window (协议4):**
 - $0 \leq \text{size of Sending window} \leq 1$
 - $\text{size of receiving window} = 1$
- **Go-back-N (协议5):**
 - $0 \leq \text{size of Sending window} \leq \text{MAX_SEQ}$
 - $\text{size of receiving window} = 1$
- **Selective Repeat (协议6):**
 - $0 \leq \text{size of Sending window} \leq (\text{MAX_SEQ}+1)/2$
 - $\text{size of receiving window} = (\text{MAX_SEQ}+1)/2$

图 32: 3个协议窗口大小小结

1 数据链路层最重要的作用就是：通过一些（）协议，在不太可靠的物理链路上实现（）数据传输。

2 在数据链路层，数据的传送单位是（）。

3 在计算机网络通信中，常采用（）方式进行差错控制。

4 所谓（）就是不管所传数据是什么样的比特组合，都应当能够在链路上传送。

5 物理层要解决（）同步的问题；数据链路层要解决（）同步的问题。

6 所谓（）就是从收到的比特流中正确无误地判断出一个帧从哪个比特开始以及到哪个比特结束。

1 数据链路层、可靠的

2 帧

3 检错重发

4 透明传输

5 比特、帧

6 帧同步

在停-等待协议中，应答帧为什么不需要序号？

由停止等待协议的工作原理可知：收方每收到一个正确的数据后，都立即向发方发送一个应答帧，发方只有收到上一个数据的确认帧后，才能继续发送下一帧。所以，在停止等待协议中，无须对应答帧进行编号。

解释零比特填充法。

在**HDLC**的帧结构中，若在两个标志字段之间的比特串中，碰巧出现了和标志字段**7E**（为**6个连续1**加上两边各一个**0**）一样的比特组合，那么就会误认为是帧的边界。为了避免出现这种情况，**HDLC**采用零比特填充法使一帧中不会出现**6个连续1**。

数据链路(逻辑链路)与链路(物理链路)有何区别？

物理链路：就是一条无源的点到点的物理线路段，中间没有任何其他的交换结点。在进行数据通信时，两个计算机之间的通路往往是由许多的链路串接而成的。

逻辑链路：在物理线路之外，加上一些必要的规程来控制这些数据的传输。实现这些规程的硬件和软件加到链路上，就构成了逻辑链路。

4.4 点到点协议 PPP协议

前6个都是模拟协议，PPP协议，点到点协议 Point to Point Protocol
PPP是一种在链路上传输分组的常用方法：

- 采用字节填充的帧界法(0x7E)
- 无序号帧(无确认无连接)用于承载IP分组
- 采用校验和检错

使用Internet校验和（4位字），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

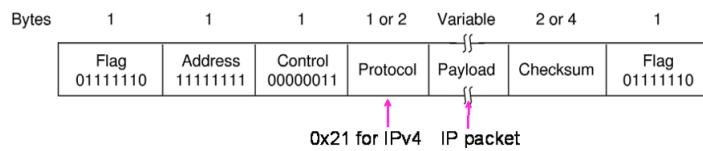


图 33: PPP帧格式

传输之前需要认证，两种认证方法：

- PAP：2次握手，传送账号密码，接收方检查后接收或拒绝
不足：明文传送，可以不断尝试密码，造成DOS攻击，拒绝服务攻击
- CHAP：3次握手，由中心节点发送一个随机数，再等待对方回复，根据对方的回复决定是否接收或拒绝对方

发送：0x7E，先使用0x7D填充，再发送 0x7E XOR 0x20 即0x5E

接收：扫到0x7D就丢弃，后面的数字xor 0x20（这也是为什么0x7D是 0x 7d 0x 5d）

地址固定为11111，因为点到点传输，对方的地址是非常明确的，因此不用设置地址。控制0011表示无确认无传输

protocol: BIAOSHI表示上层协议

5 介质访问子层

数据链路层分为两个子层:LLC Logical Link Control,MAC Media Access Control

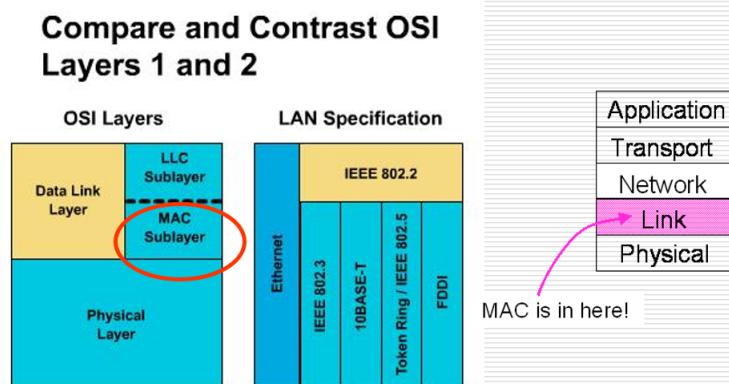


图 34: 介质访问控制子层的位置

广播需要解决的问题: 可能存在两个或更多的站点同时请求占用信道

解决办法:介质的多路访问控制，在多路访问信道上确定下一个使用者

怎么分配信道?(介质访问控制)

- 静态分配：只有一个站 / 用户使用信道，不用就浪费了；频分多路复用 FDM；时分多路复用 TDM；
问题:资源分配不合理，不满足用户对资源占用的不同需求；有资源浪费，效率低；延迟时间增大N倍

适用情况：适于用户数量少且用户数量稳定的情况；适用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：信道是开放的，没有预分配的

基本思想：通过多路访问协议动态分配信道资源，提高信道利用率

5个关键假设：流量独立、单信道、冲突可观察、载波侦听与否、帧的发送方式

多路访问协议：

- 随机访问协议：站点争用信道，可能出现站点之间的冲突；
典型协议：**ALOHA协议**、**CSMA协议**、**CSMA/CD协议**
- 受控访问协议：站点被分配占用信道，无冲突

5.1 ALOHA协议

ALOHA协议：想什么时候发送数据帧就什么时候发送

5.1.1 纯ALOHA协议

纯ALOHA协议：想发就发，当检测到冲突（发送失败），就延迟重发

帧时：发送一个标准长的帧所需时间

统计规律：每个时间帧发送k个数据的帧数满足泊松分布 $P_r[k] = \frac{G^k e^{-G}}{k!}$ 其中，G表示帧时T内，信道内的帧数（包括重发的）

吞吐率S：在发送时间T内成功发送的平均帧数（ $0 < S < 1$ ）

信道利用率：T为单位时间的时候的吞吐率

运载负载：G，发送时间T内，总共发送所有的帧数平均值（包括原发和重发）

根据定义有： $S = GP_0$ ，其中 P_0 表示一帧发送成功的概率。那么问题是， P_0 怎么求？

对于ALOHA协议，当一个发送端想要发送数据的时候，它想要发送成功，就要求发送的时间段内没有其他数据帧发送。而发送端所占的时间，至少会有两个帧时是处于危险期（可能冲突），因此需要两个帧时都没有数据帧，根据前面的分布规律

$$P_0 = (Pr[0])^2 = e^{-2G}$$

求吞吐率S的极大值： $S' = e^{-2G} - 2Ge^{-2G} = 0$ 得 $G = 0.5$ 时， $S = 0.184$

即纯ALOHA协议的信道利用率最高为18.4%

5.1.2 分槽ALOHA协议

分槽ALOHA协议：只有时间片开头才发送帧，冲突发生会浪费一个时间槽（把冲突时间降低为1个t，而不是纯ALOHA的2个t）

- 时间分槽，只有在时间槽开始的时候才能发送数据帧
- 一个时间槽只有一个帧，那么这一帧一定能成功发送
- 多个帧发送，那么均发送失败，这个时间槽作废
- 时间槽一般取帧时

而对于分槽的ALOHA协议，只要没有其他人抢占帧时，那么就一定能发送成功，因而只需要考虑一个帧时内没有数据帧即可，因此

$$P_0 = Pr[0] = e^{-G}$$

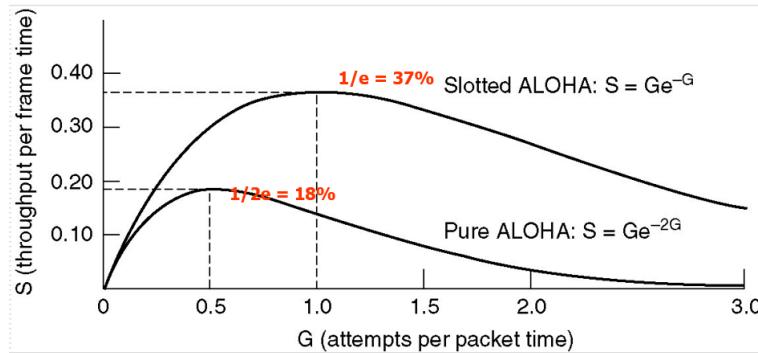


图 35: ALOHA吞吐率和G之间的关系

ALOHA协议的应用:电缆传送数据，基站之间发送数据，多个RFID与RFID读写器

5.2 CSMA协议载波侦听多路访问协议

5.2.1 CSMA协议

CSMA协议载波侦听多路访问协议:先听后写(改进的ALOHA协议)

当一个站有数据要发送的时候,首先看线路上是否有其他线路发送数据

- 非坚持
先监听,介质空闲,开始发送,如果忙就等待随机时间,然后重复 (随机浪费资源)
- 坚持
先监听,介质空闲就发送,如果忙就持续监听,一旦有空就发送 (多人等待就会冲突)
- p-坚持
前两种的这种,当信道空闲的时候,有p的概率发送,有1-p的概率继续等待随机时间(为了避免抢占信道时候的冲突)

这种协议,在信道忙的时候不会冲突,而当信道闲的时候,还是可能发生冲突

- 第一种情况是,多个帧等待发送
- 第二种情况是,由于延迟问题,一个发送端抢占信道之后,另一端未能检测到继续发送而导致冲突

5.2.2 CSMA/CD

冲突窗口:至多需要发送来回的时间就能检测到冲突,发生冲突时间的上限,数值上等于最远两站传播时间的两倍

CSMA/CD(以太网使用):先听(是否空闲)后发,边听(是否冲突)边发,一旦冲突,等待随机时间,并且同时发送强化冲突信号通知其他工作站也等待随机时间

这样从结果上来看,CSMA/CD从时间上划分出了竞争期和非竞争期,竞争期(最大冲突检测时间)大家随便发直到检测冲突,然后检测到不冲突的时候,就继续发送(非竞争期) (因为是边听边发送,所以只要渡过了竞争期,其他线路都能确定监听到发送的信号,所以后面就能安稳发送完所有数据帧)

收发冲突的标记:发送和接受到的信号不一致

要求:

- 时隙宽度 = 最大冲突检测时间：保证一个时隙能检测最远冲突
- 发送帧的有效时间 \geq 最大冲突检测时间：防止发送冲突时已经完成短帧发送而造成异常情况

5.3 其他多路访问协议

5.4 位图协议

位图协议：也叫预留协议，如有N个站点共享信道，编号为0~N-1，其竞争周期将分为N个时隙，每个站点占有一个时隙，如某站准备发送，则可在属于它的时隙内填入1，一个竞争周期后，则将按顺序发送，不会产生冲突

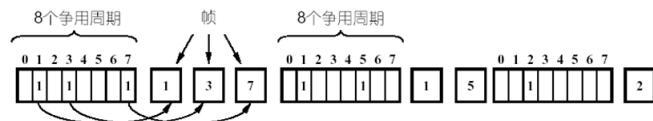


图 36: 位图协议图示

- 假设系统有N个用户，需要N个时隙
- 在低负荷条件下，如每帧的数量为d bit

6 IEEE800系列标准与以太网

以太网的两种类型：

- 经典以太网
总线拓扑，使用集线器扩充网络
使用集线器的星型拓扑结构在逻辑上相当于总线
- 交换式以太网
星型拓扑，以交换机为中心

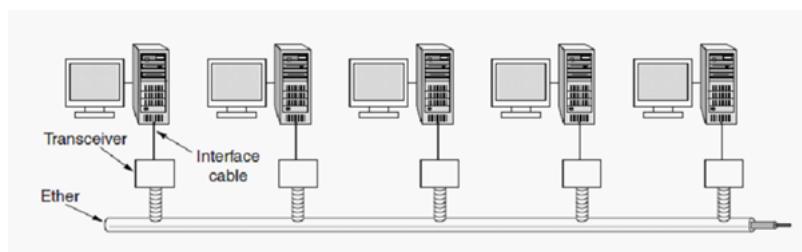


图 37: 经典以太网

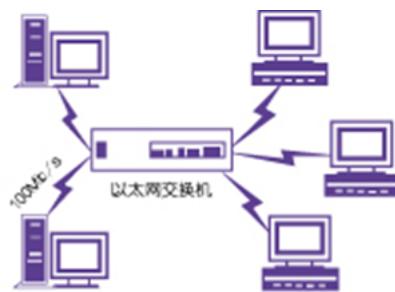


图4 交换式以太网

图 38: 交换式以太网

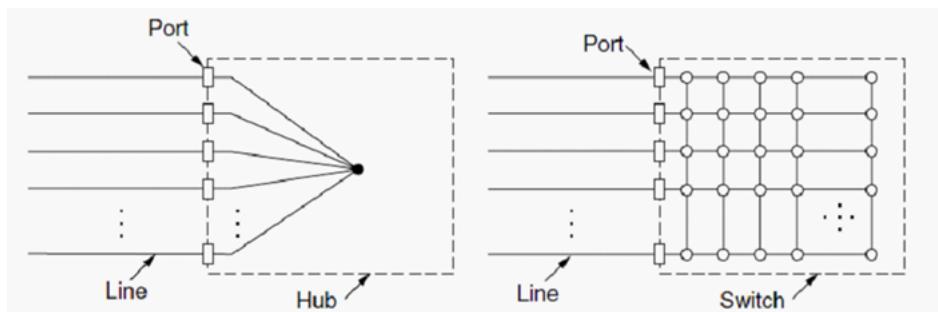


图 39: 集线器与交换机的区别

以太网的命名规则:

- 以太网的帧结构

7	1	2/6	2/6	2	0~1500	0~46	4
先导字段		目的地址	源地址		数据	填充字符	校验和
10101010	10101011			2	0~1500	0~46	4

帧开始字符10101011 类型: 表示上层使用的协议
如IP协议为2048

- 802.3的帧结构

7	1	2/6	2/6	2	0~1500	0~46	4
先导字段		目的地址	源地址		数据	填充字符	校验和
10101010	10101011			2	0~1500	0~46	4

帧开始字符10101011 数据字段长度

图 40: 以太网的命名规则,分段长度中,2表示200米的意思,基本单位是100米

以太网的编码方式:采用曼彻斯特编码对于802.5采用差分曼彻斯特协议

IEEE802.3与以太网帧组成的区别:

- 以太网的帧结构

7	1	2/6	2/6	2	0~1500	0~46	4
先导字段 10101010		目的地址	源地址		数据	填充字符	校验和

帧开始字符10101011 类型: 表示上层使用的协议
如IP协议为2048

- 802.3的帧结构

7	1	2/6	2/6	2	0~1500	0~46	4
先导字段 10101010		目的地址	源地址		数据	填充字符	校验和

帧开始字符10101011 数据字段长度

图 41: IEEE802.3与以太网帧组成的区别

802.3的帧组成:

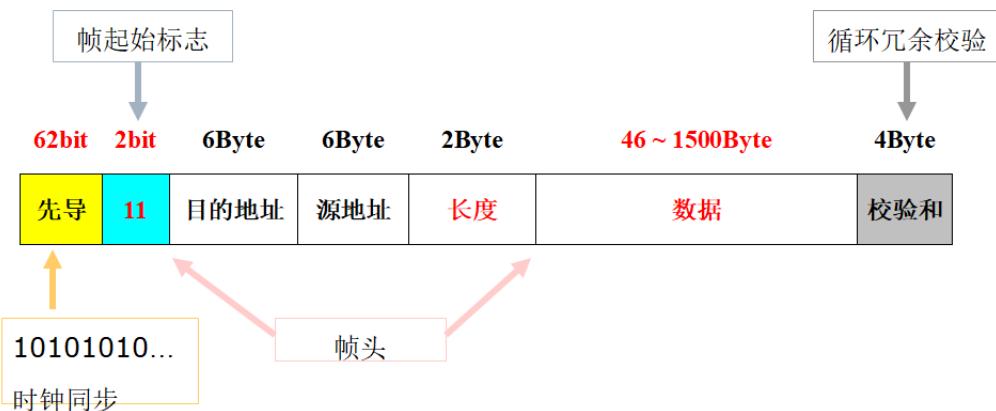


图 42: 802.3的帧组成

最短帧长:64B

- 帧头
 - 6B目的地址,6B源地址,2B长度
- 数据位
 - 最少46B
- 校验位
 - 4B

貌似没有计算先导字段

MAC地址:物理地址,xx:xx:xx:xx:xx:xx

位	47	46	45.....41	40	39.....24	23.....0
	制造商标识	全局/局部地址标志	制造商标识	组播标志位	制造商标识	系列号

图 43: MAC地址组成:前24位表示公司

长度字段:帧最短长度为64B,最大为1518B

为什么是64B?为什么是1518B?为什么要有最短长度?P227,第一问在后面有解答

数据字段:最少为64B,不够则填充

帧校验字段:采用32b,CRC校验

怎么区分到底代表类型还是长度呢 检查这个字段的数值: 如果小于等于 1536(0x600), 则是长度 (802.3) 字段, 如果大于 1536, 则表示类型 (以太帧)

为什么要大于64B?

为什么有效帧长度≥ 64 Byte? P219

- CSMA/CD的要求
 - 最短帧的发送时间 ≥ 争用时隙 2τ
- 以太网 (802.3) 规定, 在10Mbps局域网中
 - 时隙: $2\tau = 51.2$ 微秒
 - 最短帧长度: $10Mbps \times 2\tau/8 = 64$ Byte
 - 或者: $(51200/100ns)/8=64Byte$

图 44: 为什么要大于64B?

为什么要51.2微秒的时隙?

二进制指数后退法:有冲突检测的载波侦听策略当冲突的时候会随机等待一段时间,那么这个时间的范围什么确定?

■ i次冲突后时间片为:

- $0 < i \leq 10$ 时, 取 $(0 \sim 2^i - 1) \times 2\tau$
- $10 < i < 16$ 时, 取 $(0 \sim 1023) \times 2\tau$
- $i > 16$ 时, 放弃发送

图 45: 第i次冲突等待时间

进一步优化:一般确认接收之后,则下一个时隙留给接收方返回确认消息,而不是进入新一轮竞争

6.1 快速以太网(100M)

以太网提高负载的方法:

- 提速到100M
- 全双工
- 使用交换机代替集线器

100M以太网:保留原来的帧格式、接口和偶成UI则,只是将比特时间从100ns降低到10ns(物理上表现为电缆长度为原来的 $\frac{1}{10}$)

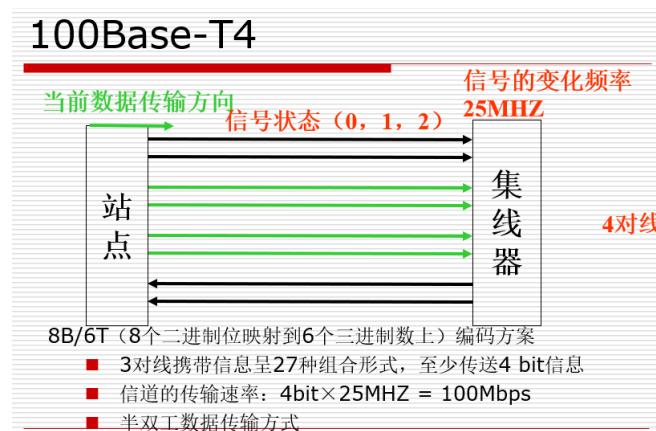


图 46: 100BaseT4

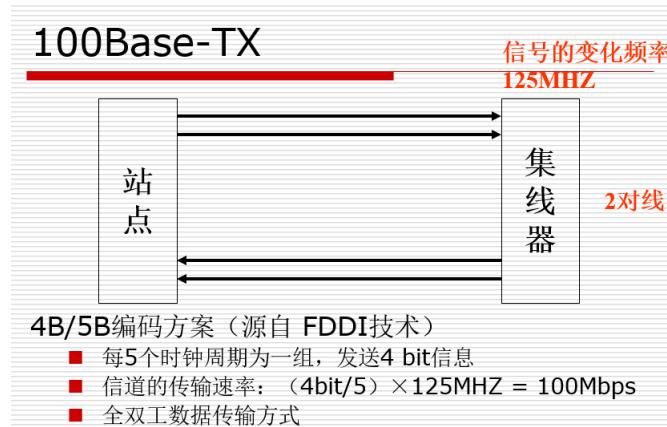


图 47: 100BaseTX

7 交换机

网桥与交换机的关系:就像中继器与集线器的关系

7.1 千兆以太网(1000M)

出现的问题:1000M以太网速度太快导致传输时间太短,进而传输的距离也短

- 载荷扩充 (carrier extension)
 - 方法
 - 在发送方硬件加入/接收方硬件删除，将帧长扩展到 512Byte(8倍)
 - 目的
 - 保证网络半径为合理长度 (200米=25*8)
 - 保证兼容10M/100M的最短帧64字节特性
 - 缺点：线路利用率低下
- 帧串 (frame bursting)
 - 方法
 - 连续发送多个帧，只有当帧串小于512Byte时填充
 - 目的：提高信道利用率

图 48: 千兆以太网解决办法

注:计算机网络中一般取 $1M = 10^6$, $1K = 10^3$,存储单位采取2的指数
以太网:

OSI Layers

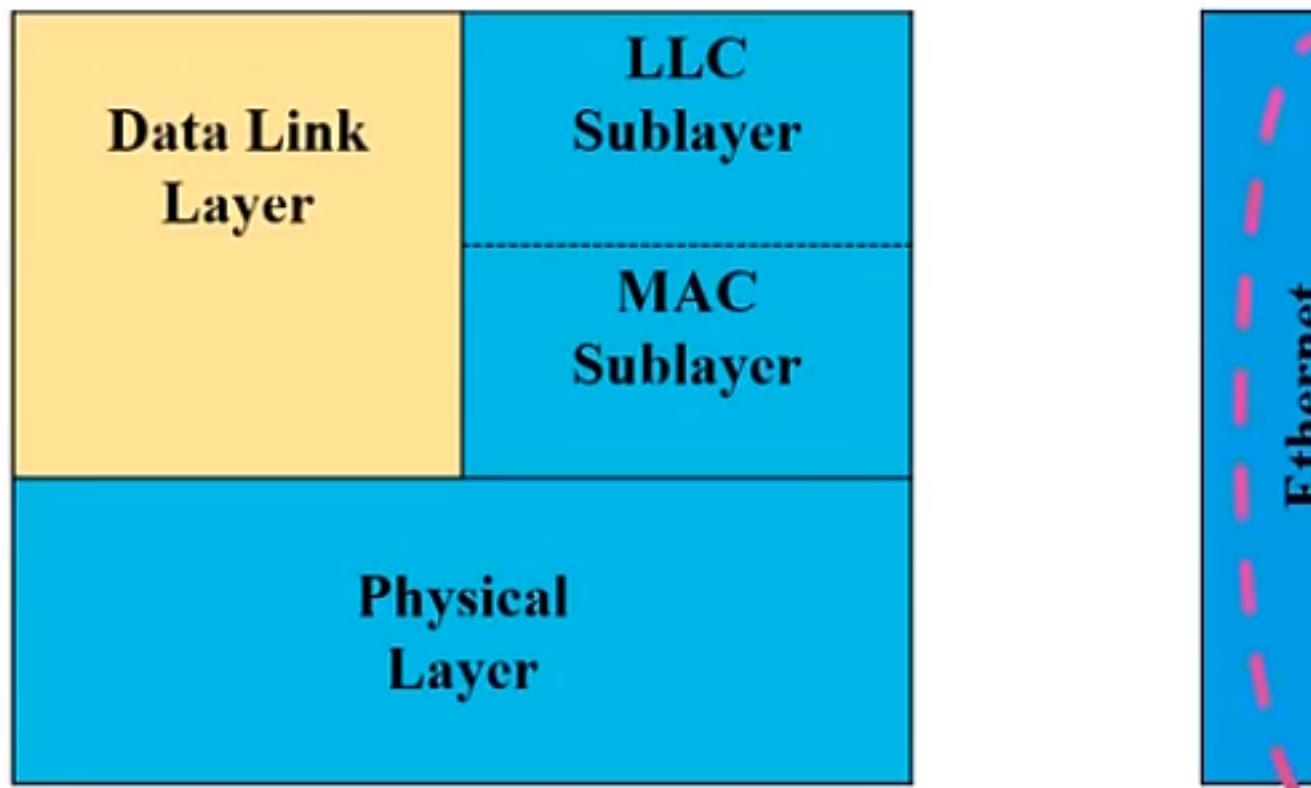


图 49: 以太网的位置

- 经典以太网(3-10M)
- 交换式以太网(10M,100M,1G)

以太网的命名规则:

□ 10Base2 (IEEE 802.3a)

- 10: 传输带宽 (单位Mbps)
- Base: 基带传输
- 2 (或5) : 支持的分段长度 (100米为单位, 四

□ 10Base-TX (IEEE 802.3X)

- T: 铜制非屏蔽双绞线
- F: 表示光缆

图 50: 以太网的命名规则

以太网采用曼彻斯特编码

怎么确定CSMA/CD的等待时间?使用二进制指数后退算法

二进制指数后退算阿飞:检测到冲突后,等待的时间以2的指数增长,等待时间设置一个上界(第10 16次),对于以太网,一个时隙是(51.2 us),16次冲突后放弃发送

□ i次冲突后时间片为:

- $0 < i \leq 10$ 时, 取($0 \sim 2^i - 1$) $\times 2\tau$
- $10 < i < 16$ 时, 取 ($0 \sim 1023$) $\times 2\tau$
- $i > 16$ 时, 放弃发送

图 51: 以太网的冲突等待时间

100M以太网:保留原来10M的帧格式,只是传输距离改为原来的1/10(原来为2500m),并且采用4B/5B编码

以太网的帧格式:



图 52: 以太网的冲突等待时间

IEEE802.3帧格式

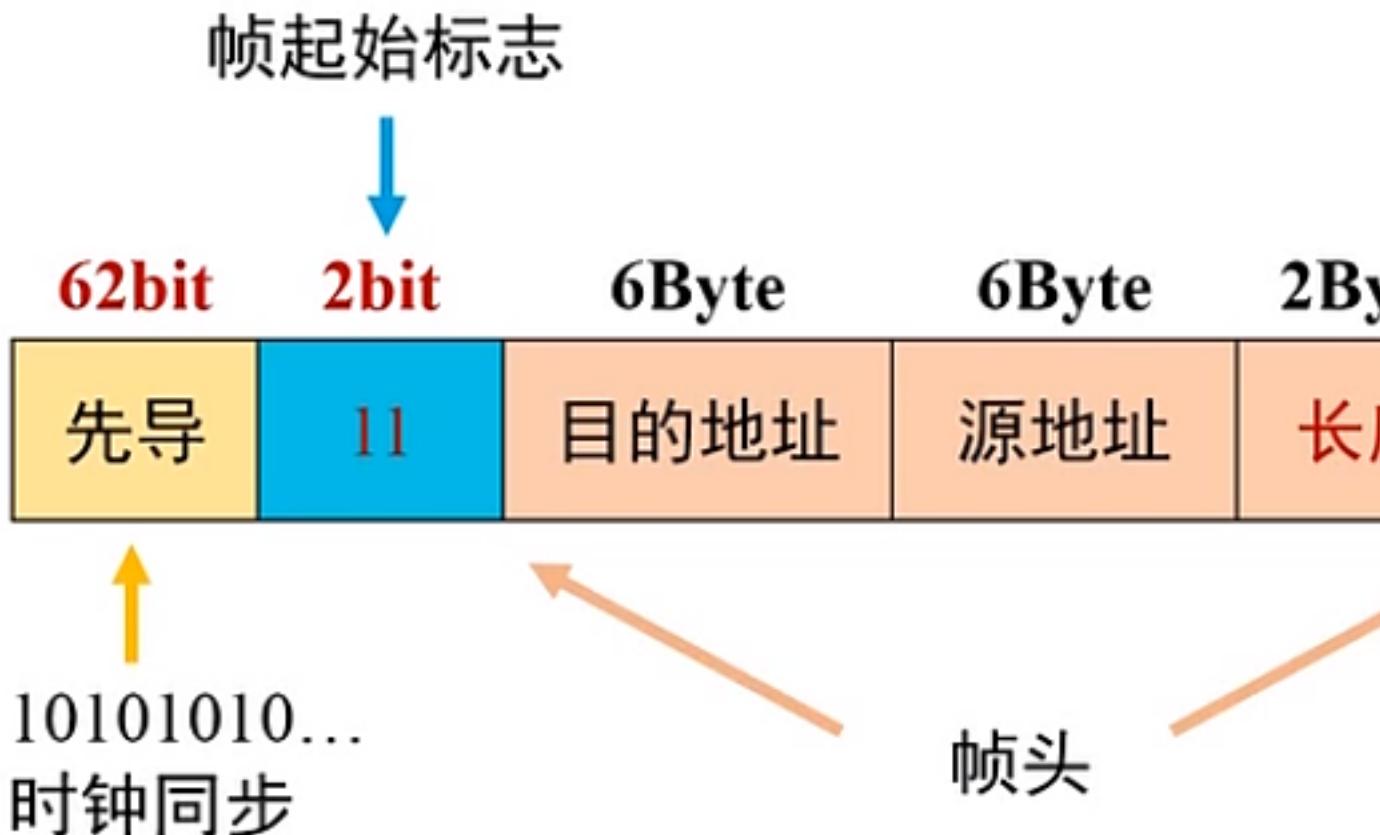


图 53: 以太网的冲突等待时间

前导码:定为一个新的帧其实

目的地址/源地址:48位,MAC地址,全球唯一

长度字段:长度范围:64字节 1518字节,不包括前导码

数据字段:长度最小为46个字节,少于46个就填充

校验字段:32位的CRC校验,校验范围:目的/源地址,长度,LLC数据等字段

怎么区分是类别还是长度呢? 数值小于1536(0x600)的就是长度(802.3)字段,否则表示类型(以太帧)

最少64字节怎么计算出来的?

□ CSMA/CD的要求

➤最短帧的发送时间 \geq 争用时隙 2τ

□ 以太网（802.3）规定，在10Mbps局域网中

➤时隙： $2\tau = \underbrace{51.2}_{\text{微秒}}$

➤最短帧长度： $10\text{Mbps} \times 2\tau/8 - \underbrace{64}_{\text{Byte}}$

或者： $(51200/100\text{ns}) / 8 = 64\text{Byte}$

图 54: 以太网的最小长度计算

二层交换:不同协议需要进行重新包装

网桥如何维护内部转发表?

- 泛洪算法

对于未知的(表中无记录),转发到其他所有端口

- 反向学习

记录帧来源的MAC地址加入到转发表中

但是网络的拓扑结构是变化的,网桥怎没适应这些变化? 定期删除超时的表项,每次来一帧,就更新表项的时间戳为当前时间。

怎么利用转发表? 源LAN和目的LAN相同,则丢弃,否则查表,有的转发,无则广播

为保证安全,会使用冗余的拓扑结构(多条线),但是冗余的拓扑结构可能带来以下问题

- 广播风暴

广播的时候,如果拓扑结构有回路,则广播信息会不断在回路循环

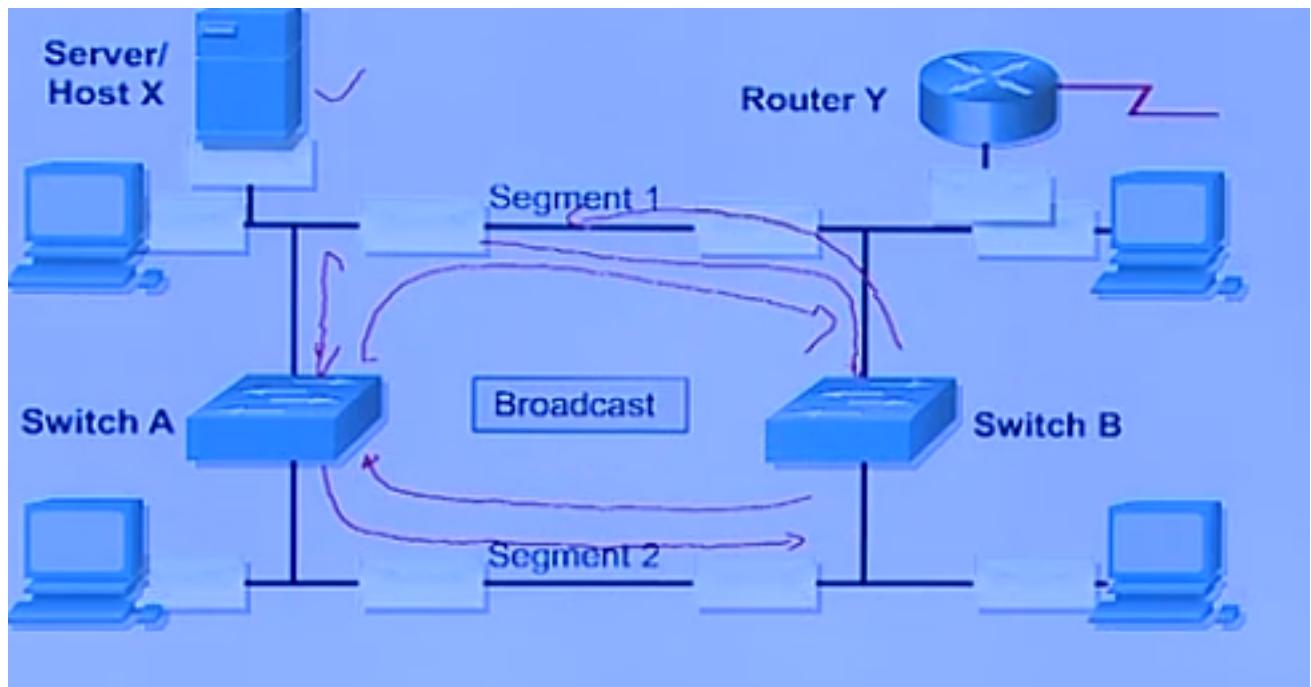


图 55: 广播风暴

- 多帧传送

一个接收点,会接受到多个相同的帧

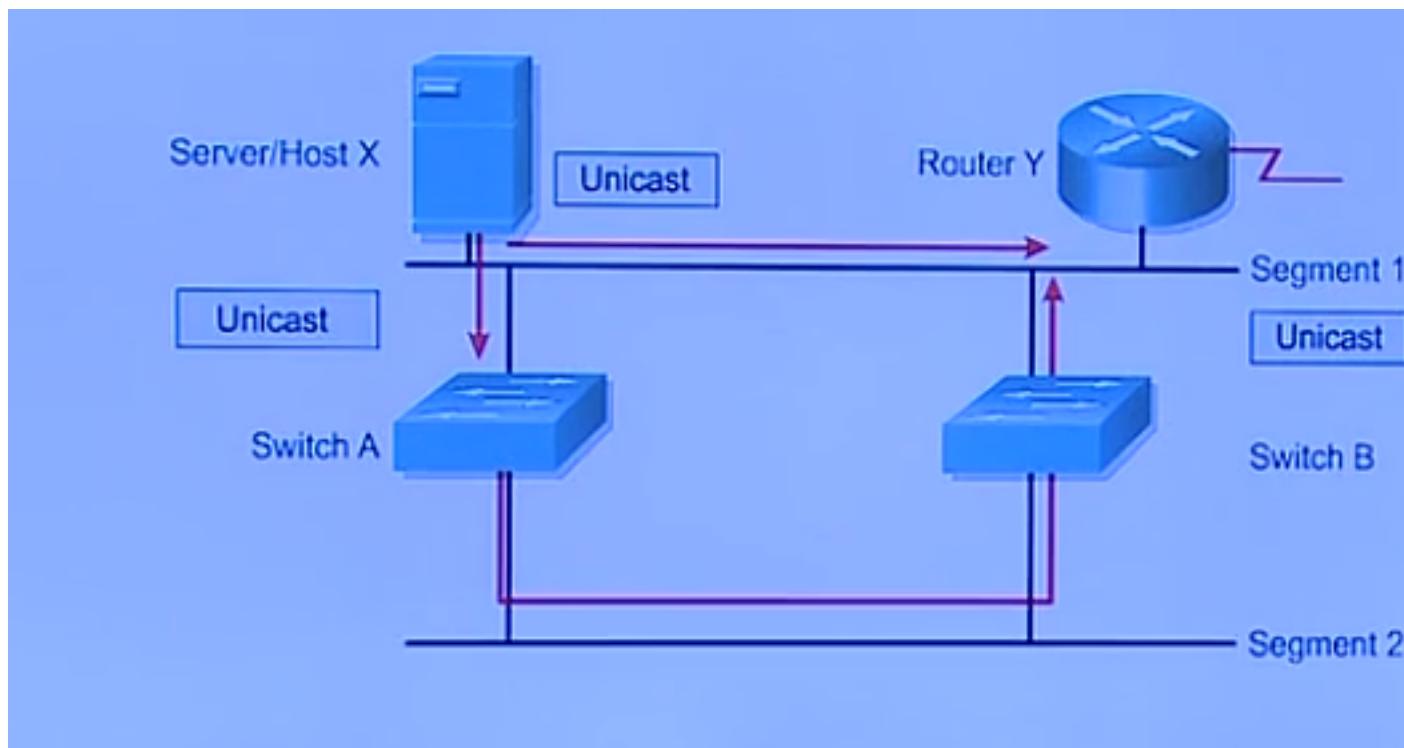


图 56: 多帧传送

- MAC地址库不稳定
因为多条路径,源帧的MAC地址库不稳定,导致转发的行为不稳定

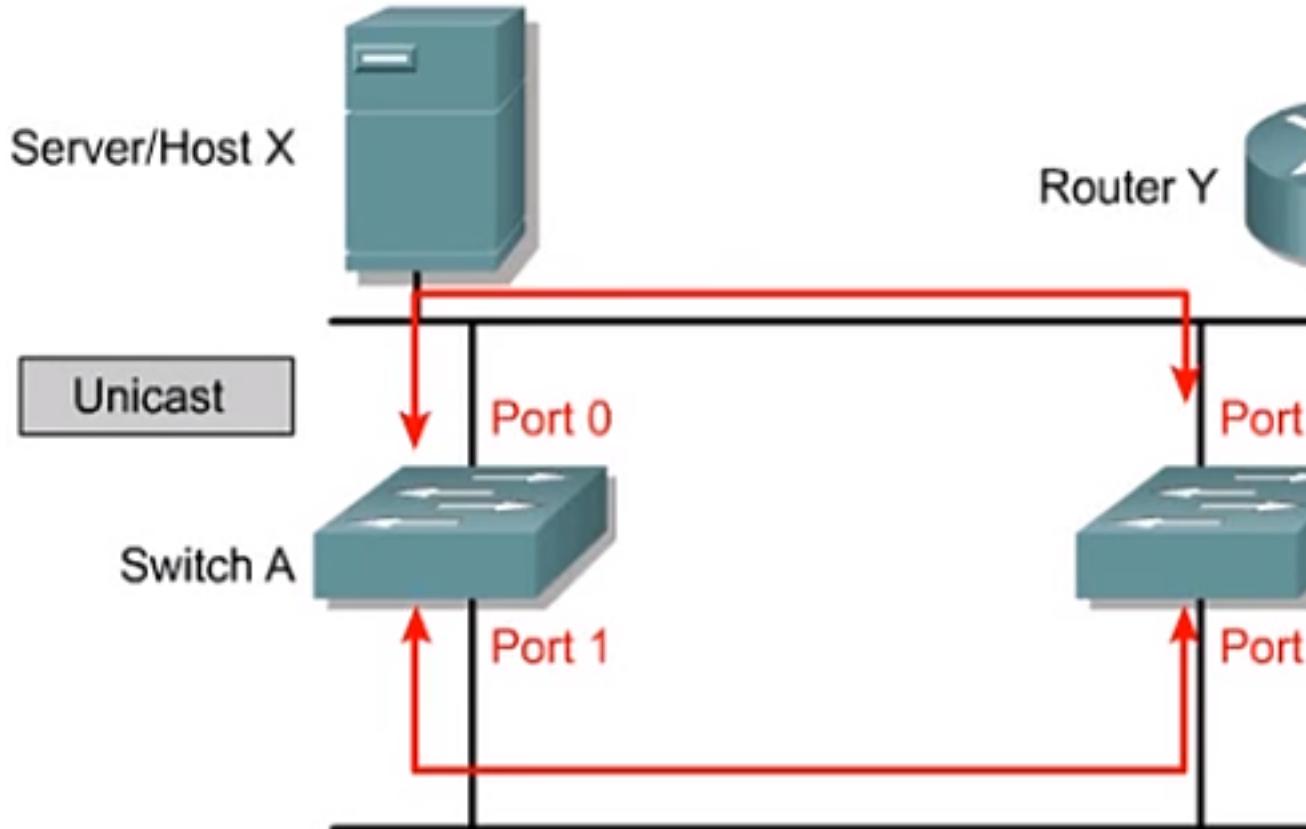


图 57: MAC地址不稳定

解决办法:生成树协议,使用生成树打断环路,生成逻辑上无回路的树,但是不能保证路径最优

- 每个网络选择一个根网桥
- 每个网桥有一个根端口
- 每个网段一个指定端口
- 非指定端口不被使用

逻辑上禁用端口,虽然这个端口虽然不使用,但是还会参与侦测,当出现短路,马上启用新的端口

虚拟局域网:逻辑上的设备等同于物理上的LAN

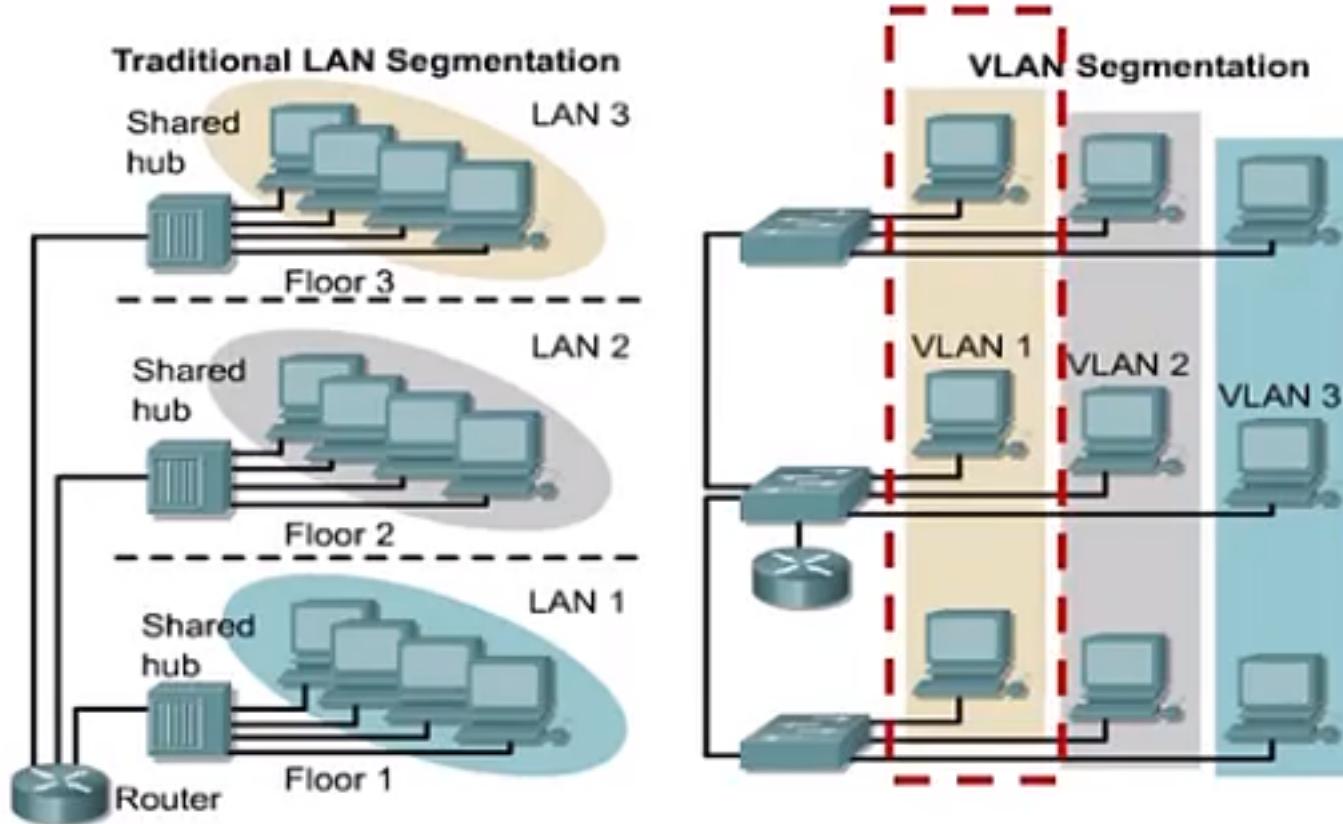


图 58: VLAN

VLAN的实现:

- 基于端口
交换机上,把某个端口划分给某个VLAN
帧标记法来区分是哪个VLAN的成员(用于帧穿越(TRUNK)干线(交换机之间)使用),进入干线打上标记,离开干线去除标记
- 基于MAC地址
- 基于三层协议

交换机交换方式:

- 存储转发
先存下来,校验完再转发,不转发错误帧
- 直通交换
一旦读取目的地址就立刻转发
- 无分片交换
读取到64字节才转发(过滤小于64字节的碎片帧)

8 网络层

目标:把源短数据包(分组)一路送到接收方
有3个方面的问题:

- 封装源数据(IP分组)
- 识别目的机(IP地址)
- 路由(寻路径,路由选择协议,路由器)

传输过程中可能遇到的问题:

- IP地址不够用
NAT技术
- 丢包
重传或者恢复
- 拥塞
拥塞控制

IP协议:

- IP地址
- IP分组

路由选择协议:

- DV协议(早期路由)
- LS协议

网络层协议的技术:

- ARP协议
不知道MAC地址的协议
- ICMP
- CIDR
- NAT

源和目的机之间的网络:

- 无连接网络,数据报网络
随机到达
需要包含目的地址
每个分组独立选择路由

- 面向连接网络,虚电路网络

每个路由保存虚电路的连接方向,有连接编号以区别不同的连接

先发先到

只需要包含下一个路由的地址

一次路由

IP:Internet protocol(互联网协议)

IP地址:32bit

IP地址的分类:

IP地址分为A、B、C、D、E类					
	0	8	16	31	
A类	0	前缀		后缀	
B类	1 0	前缀		后缀	
C类	1 1 0	前缀		后缀	
D类	1 1 1 0	多址传送地址			
E类	1 1 1 1	保留将来使用			

图 59: IP地址的分类

- A类地址

前8位为网络部分,剩下为主机部分

第一位为0,剩下7位为某个定值(即第一个数字为0-127)的地址

- B类地址

前16位为网络部分,剩余部分为主机部分

以10开头

即第一个数字为(128-191)的地址

- C类地址

前24位为网络部分,剩下为主机部分

以110开头,及第一个数字为(192-223)

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

图 60: IP地址的分类-2

- 保留地址(D类和E类)

保留地址:不能分配给某台主机某个设备的地址,包括以下几个情况

D类:224.0.0.0-239.0.0.0

E类:240.0.0.0-254.0.0.0

网络地址:主机部分全部为0的地址

广播地址:主机部分全为1的地址,可以出现在分组的目的地址,表示在这个网络内广播

缺省地址:0.0.0.0,表示这个主机,这个网络

泛洪广播地址:255.255.255.255(实际上是本地网络的广播,而不是整个网)

以127开头的地址,127.0.0.0:表示环路地址(其中127.0.0.1,代表本机)

169.254.x.x,非正常地址,未从DHCP获得的地址

为什么泛洪广播 = 本地广播?

因为路由器会把泛洪广播丢弃

定向广播地址:主机部分全部为1,比如192.168.1.255,路由器会把帧转发到192.168.1.0这个网络

ipv4地址在11年已经分配完啦,撒花

数据如何穿越路由器的?

寻址方式:

- IP寻址

根据目的IP地址,找到目的网络

- MAC寻址

根据目的MAC地址,找到目标机

路由器接收到一个分组后怎么处理?

- 解封装到网络层
得到目的IP
- 确定目标网络
目的IP与子网掩码进行与操作确定目的网络
- 重新封装,转发

A R Q: 接收到确认之后才继续发送

RTT是客户到服务器往返所花时间 (round-trip time, 简称RTT)

网络层: 网络层分类:

- 数据报网络
提供无连接服务,根据路由表进行寻址转发,一旦某个点故障可以绕过某个点达到目的机
- 虚电路网络
提供面向连接的网络,路径上保存来向和去向

路由表: 直连路由:当直连网络接入的时候路由器可以直接得到

静态路由:由管理人员直接配置的网络默认路由(缺省路由):当分组找不到路的时候(查不到其他表项的时候),往默认路由转发,默认路由可以避免错误的丢包,缩减路由表规模,减少路由器的运行负担

动态路由:由路由选择协议动态地建立、更新和维护的路由

路由选择协议(算法):

- 距离矢量路由选择(DV)
例子:RIP,路由信息协议
- 链路状态路由选择(LS)
例子:OSPF:开放的最短路径优先

路径的度量、代价、开销、成本的评判标准?

- 路径长度(跳数 , hop)
- 可靠性(误码率)
- 延迟
- 带宽

距离矢量路由选择协议(DV协议):每个路由器维护一张表 ,表中列出了当前一直的到每个目标的最佳距离,以及为了到达这个目标,应该从哪个接口转发。

DV协议常用小型网络,RIP是典型的DV , 是早期互联网中广为使用的路由选择协议

工作原理:

- 每个路由器维护两个向量, D_i 和 S_i , 分别表示从该路由器到其他所有路由器的距离即相应的下一跳。
- 在邻居路由器之间交换路由信息(矢量)
- 每个路由器根据收到的矢量信息更新自己的路由表

$D_{i1} : h:i01 S_{i1} : @i01\Gamma , \Omega ff\Gamma$

更新路由表:

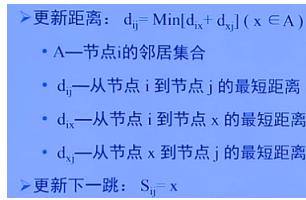


图 61: 更新路由表

优点:简单

缺点:

- 交换信息太大
- 路由信息传播慢,可能导致路由信息不一致
- 收敛慢
- 可能会出现计数到无穷大的问题
- 完全相信邻居,站得不高,看得不远

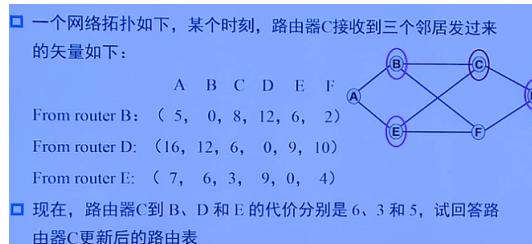


图 62: 更新路由表

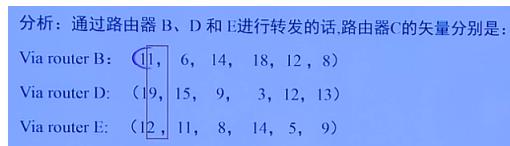


图 63: 更新路由表

RIP:路由信息协议,Routing information protocol,主要运行80年代,是DV的一种

- 采用跳数hop作为度量(metric)
- 当度量超过15跳,目的被认为不可达
- 默认地,每30秒交换一次矢量信息

配置RIP协议: router RIP network 192.168.1.0 network 192.168.2.0

为什么rip会衰败?

- 路由环路 routing loop
- 计数到无穷 count to infinite
- 收敛慢 slow convergence

链路状态路由选择:

发现邻居节点:当路由启动的时候会发送一个hello分组,收到hello分许的路由会会送一个应答,应答自己的名字(全球唯一)

决定线路的开销:路由器发送一个特别的ECHO分组,另一个边立刻会送一个应答

构造链路状态分组LSP / LSA:

- 发送方的标识
- 序列号
- 年龄
- 邻居列表
- 到邻居的成本开销

什么时候构造分组?周期性地构造和分发,或者有特别的事件发生时构造,比如某条小路down掉了

基本算法:当一个新的分组到达时:如果是新分组(序列号大),就执行泛洪操作,否则(重复,过时)丢弃

序列号回转问题:序列号重新回来解决办法:使用32bit表示序列号,当即使每秒产生一个分组,也需要137年才能回转

如果一台路由崩溃,它将丢失自己的序列号记录,如果从0开始,新分组将会被当作旧分组而被拒绝如果序列号被破坏,产生错误变成一个超大的值,就会导致后面的包都被当作旧包解决办法:设置age字段,每秒减一,过期丢弃

为防止线路发生错误,所有的链路状态分组都要被确认(线路空闲的时候确认)

优点:

- 每个路由器认识一致
- 收敛快

缺点:

- 每个路由器需要较大的存储空间
- 计算负担大

OSPF:一种基于开放标准的链路状态协议、是目前世IGP中应用最广、性能最优的协议

- 使用带宽作为度量
由于度量是越小越好,因此把带宽作为分母,具体为 $\frac{10^8}{BW}$
- 适合在大型网络中使用
- 收敛速度快
- 通过分区实现高效的网络管理

单区域OSPF:

- ROUTER ID 32位无符号整数,路由器的唯一表示(在整个自治系统内唯一)
- 协议号:89
- TTL = 1(跳不出这个区域?虚连接除外)

OSPF分组类型:

- DD报文
链路状态数据的摘要信息,用来确定所需要的状态数据以节省带宽
- LSR
请求DD报文确定之后所需要的报文
- LSU
返回LSR的请求,或者链路状态发生变化的时候发送
- LSA
收到LSU回送LSA

OSPF数据包类型	描述
Type 1—Hello	与邻居建立和维护毗邻关系。
Type 2—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。
Type 3—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
Type 4—链路状态更新 (LSU)	向邻居路由器发送链路状态通告
Type 5—链路状态确认 (LSA)	确认收到了邻居路由器的LSU

图 64: OSPF分组类型

ospF运行步骤:

- 建立路由器的毗邻关系
- 选举DR和BDR
- 发现路由
- 选择最佳路由
- 维护路由信息

建立路由器的毗邻关系:

- 广播hello报文
 - 回送hello报文
- 此时hello报文内,DR为回送的路由,Neighbors Senn为广播的路由,用以确定双方关系
此时状态称为2-way状态

- 互发DD报文

DD报文有4个参数

第一个报文的seq初始随机生成

(initial)I=1表示是第一个报文

M = 1表示后面还有DD报文

MS(master) = 1,表示我是master(由每个路由的路由id决定)

谁是master,则后续的DD报文的seq就由谁决定对seq做递增操作

- loading

根据对方的路由状态,发送LSR请求,直到双方状态一致

- full

选举DR和BDR:

为什么?对于全连接网络,如果两两之间同步,可能同步时间很长,但是如果只和某个路由器(DR)同步,就可以加快同步的速度

选举DR的过程:

- 登记本网段所有的ospf路由器
- 登记本网段所有优先级 > 0 的ospf路由器
- 选举优先级最大的,若优先级相同,则选路由id最大的

DR一旦当选,除非故障,否则不会更换

DR选举的过程,也会选取次优的(BDR , B = Backup),当DR挂了,BDR上位
(疑问:当DR重新上线了,BDR是否要下位?)

DR存在的问题:如果不是直接相连的路由,无法和DR通信(因为TTL = 1)

解决办法:由管理员配置成PTMP,不选举DR,两两进行同步

选择最佳路由:同步完之后,怎么最优分发分组

每个路由器都有一个路径树(各自不一定相同),记录自己到各个路由的最短距离

维护路由信息:

- 触发更新,收到LSU
- hello分组发送间隔:10秒
- hello分组的失效间隔:缺省40秒(即4次没接收到hello分组就判断为down了)
- LSA在条目过期(30分钟)后,发送LSU,通告链路存货

OSPF与大型网络:

- LSDB非常庞大,占用存储空间
- 计算负担重
- 一点变化引发重新计算,网络经常处于动荡状态

解决办法:分区域

存在问题: 区域之间的路由通过和边界路由打听确定路由路径(类似DV),因而可能遇到环路的问题,这里通过强制定义一个0号区域(骨干网络),使得其他所有区域都和这个区域挂接

但是对于一些很远的区域,可能无法直接相连,就定义虚连接来解决遥远路由器的问题

BGP:边界网关协议:自治系统之间的协议(运行在TCP之上),关注是安全问题

BGP:是DV协议, 但不存在环路问题,因为它还从邻居处得到全路径信息

无类间路由,CIDR:

动机:分类导致地址浪费, 路由表膨胀问题

不再按类别分配地址,而是按用户需求分配地址

路由表需要扩展,增加一个32bit的子网掩码,以表明网络的规模

1. 按需分配,减少浪费 2. 路由汇聚(只向上传递一个网络地址+掩码),减小路由规模 3. 隔离路由翻动,路由翻动只在一个小网络内翻动

怎么聚合?只有连续的网络才能聚合

NAT技术:网络地址翻译

动机:ipv4地址枯竭,无法继续ipv6地址

做法:从ABC类地址中各划分出一段作为私人地址:

- 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:192.168.0.0-192.16+.255.255,192.168.0.0/16

私人地址:不可路由地址,不具备唯一性

怎么通信?使用NAT技术,私有地址和共有地址进行转换

其中一种称为PAT(Port address translate , 超载),把多个私有IP地址映射到同一个公网ip地址的不同端口

NAT违背了ip的结构模型(每个ip地址唯一地表示了一台机器),违背了最基本的协议分层原则,网络编程了面向连接的网络,NAT维护着连接状态,一旦崩溃,连接也消失了

NAT有一定的安全性,如果不开启NAT转换,内网机器无法和外部连接

互联网控制协议:ICMP

为什么需要?

- IP分组不可靠,可能遭遇丢包,拥塞,产生很大延迟、抖动等问题
- ICMP用来向源(通常)报告这些问题或状况
- ICMP也常用来测试网络

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

图 65: ICMP消息类型

超时消息:当TTL等于0的时候,向源IP发送超时消息回声应答:收到就应答

应用:

PING 命令(回声应答)

tracert 命令

PMTU:探测一个合适的MTU值

ARP:地址解析协议,ARP表保存ip对应的MAC地址

ARP请求:



图 66: ARP请求

优化:减少ARP请求

- 缓存ARP结果
- 收到广播的ARP请求,偷偷保存源IP和对应的MAC地址
- 启动时广播IP-MAC地址对

免费ARP请求:targetIP是自己,用来判断自己的IP是否冲突

如果远程主机不在同一子网怎么办? 使用缺省网关,目的MAC地址设置成网关,网关置換源MAC地址为自己

怎么避免被ARP欺骗?

- 静态ARP
- 不马上写ARP缓存
- 设置ARP服务器
- 硬件屏蔽

DHCP:动态主机配置协议

- 初始化状态
- 选择状态
- 请求状态
- 绑定状态

拥塞控制:确保子网能够承购所到达的流量

流控:只与指定的发送方和接受方的点到点流量有关

拥塞:当一个子网或子网部分出现太多分组,网络性能急剧下降

导致拥塞的原因:输入流量大于输出线路的容量,(负载 ; 资源) 可以通过增加内存缓存,但是延迟可能导致超时重发,恶性循环,不能有效解决

拥塞控制:

- 开环

采用良好的设计,估算容量的峰值,从一开始就保证问题不会发生
但是网络变化十分快,因此开环不合适

- 闭环

建立在反馈环路上的概念,分为3个步骤:

- 见识系统
- 反馈信息到能采取行动的地方
- 调整系统,改正问题

怎么知道发生了拥塞?

- 丢弃分组的百分比
- 平均队列长度
- 超时和重传的分组数
- 平均分组延迟
- 分组延迟的标准差-

增加资源:

- 加大节点之间的带宽
- 把流量分散到多条路径
- 启用空闲或备份的路由器

数据报子网的拥塞控制: 每个路由器监视它的输出线路和其他资源的使用情况,每个线路使用一个变量 u 关联,当 u 超过阈值的时候,对应线路就进入警告状态,每个新到达的分组都将被检查它的输出线路是否处于警告状态

抑制分组机制: 向目标源发送抑制分组,源头减少发送,但是生效时间慢

改进:逐条抑制,沿途的路由都减少分组发送,但是每条中间的路由器需要开辟缓存来保存延迟发送的流量,同时也可能导致分组超时

载荷脱落:丢弃分组,最简单有效的方法

- 随机丢弃
- 葡萄酒策略,丢弃新到达的分组(适合文件传输类)
- 牛奶策略,丢弃旧的分组(适合多媒体类)
- 丢弃不太重要的分组,但是难以衡量分组的重要性

随机早起检测:当情况恶化之前就开始丢弃分组,为确定什么时候开始丢弃分组,路由器维护者最近队列的平均长度,当超过某个阈值的时候,就被认定为拥塞,可以采取相应的措施

流量整形:调节数据传输的平均速率,因为突发的流量导致线路拥塞

算法:

- 漏桶算法

每台主机连接到网络的接口都连接有一个漏桶,分组不断放入桶中,桶取出的速率是恒定的,当桶满的时候,分组将被丢弃,每个时钟滴答只允许一个或固定数量的分组发送出去
把用户进程产生的不稳定流编程稳定的流,使用时间来换取平衡性,不能突发地数据

- 令牌算法

允许数据突发到某个程度,桶里面不再装分组,而是装令牌,一个分组只有拿到令牌才能分发出去

例子计算最大突发时间:

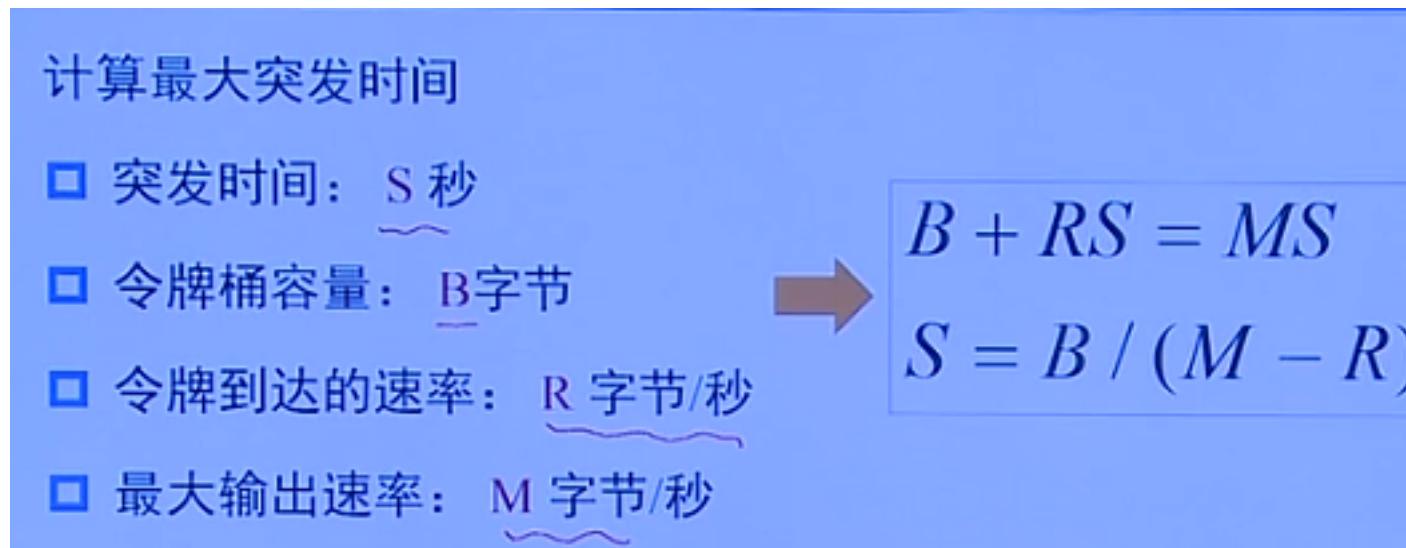


图 67: 计算最大突发时间

9 123

传输层

既然有了网络层,为什么还需要传输层? 1. 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 2. 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 3. 传输层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

传输原语: listen connect send receive disconnect

TPDU,数据段,传输协议数据单元

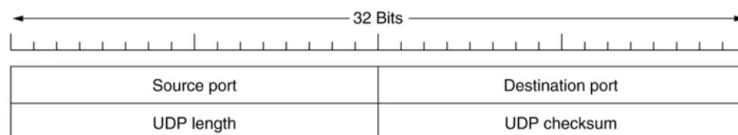
传输协议: UDP, TCP

UDP:无连接的传输协议

UDP是不可靠的服务,为什么需要UDP?

UDP 数据段头

- UDP 数据段包括8字节（8-Byte）的头部和数据两个部分
- 其中的长度域表示的长度包括头部和数据**总共的长度**
- 校验和（checksum）是**可选的**，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 (port) 定义

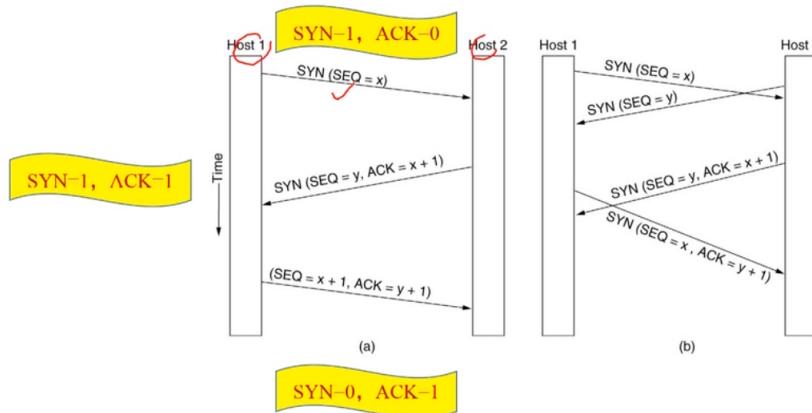
- 16 位，共有 2^{16} 个端口

➤ 端口范围：0~65535

<1023	用于公共应用（保留，全局分配， 用于标准服务器），IANA分配
1024~49151	用户端口，注册端口
>49152	动态端口，私人端口

RFC 6335

TCP连接的建立-3次握手



TCP连接的释放

TCP拥塞控制：

慢速启动:每次数据大小为原来的两倍

阈值:到达阈值时,线性递增,超时,阈值变成原来的一般。当慢速启动超过阈值时,值直接设置成阈值,然后线性增长

线性递增:超过阈值时线性递增

注:超过阈值之后,是线性递增,还是重新慢速启动,根据题意决定

TCP定时器管理:超时询问,保活定时器,关闭定时器(2倍时间后强制关闭)

TCP和UDP的对比:PPT74 最本质的区别:有无连接