计算机网络

引用资料链接：https://github.com/CyC2018/CS-Notes/blob/master/docs/notes

# 基础概念

**互联网**：网络的网络

**ISP**: 互联网给服务提供商

目前的的互联网是多层ISP: ISP 根据覆盖面积的大小分为第一层 ISP、区域 ISP 和接入ISP。

**IXP**：互联网交换点（允许两个 ISP 直接相连而不用经过第三个 ISP）。

**主机之间的通讯方式**：客户/服务器 （C/S）；对等（P2P）；

**电路交换与分组交换**：

电路交换就是开辟一条物理专用链路来通信，是点对点的连接。

电路交换：用于电话通信系统，两个用户要通信之前需要建立一条专用的物理链路，并且在整个通信过程中始终占用该链路。由于通信的过程中不可能一直在使用传输线路，因此电路交换对线路的利用率很低，往往不到 10%。

分组交换就是多点之间共用一条物理通信链路，允许同时传输多个分组。

分组交换：每个分组都有首部和尾部，包含了源地址和目的地址等控制信息，在同一个传输线路上同时传输多个分组互相不会影响，因此在同一条传输线路上允许同时传输多个分组，也就是说分组交换不需要占用传输线路。

**传播时延：**

排队时延：分组在路由器的输入和输出队列中排队等待的时间，取决于网络当前的通信量

处理时延：主机或者路由器接到分组时进行处理所需要的时间，如分析首部，从分组中提取数据、进行差错检验和查找适当的路由。

传输时延：主机或路由器传输数据帧所需的时间。

传输延迟 = 位长度/每秒传输位数

传播时延：电磁波在信道中传播所需要花费的时间，电磁波传播接近光速。

传播时延= 信道长度/电磁波传播速度

**计算机网路体系结构**：

五层协议：

应用层：为特定应用程序提供数据传输服务，例如http、DNS等协议。数据段位为报文。

传输层：为进程提供通用数据传输服务。运输层包括两种协议：传输控制协议TCP，提供面向连接、可靠的数据传输服务，数据单位为报文段；用户数据报协议 UDP，提供无连接、尽最大努力的数据传输服务，数据单位为用户数据报。TCP 主要提供完整性服务，UDP 主要提供及时性服务。

网络层：为主机提供数据传输服务。而传输层协议是为主机中的进程提供数据传输服务。网络层把传输层传递下来的报文段或者用户数据报封装成分组。

数据链路层：网络层针对的还是主机之间的数据传输服务，而主机之间可以有很多链路，链路层协议就是为同一链路的主机提供数据传输服务。数据链路层把网络层传下来的分组封装成帧。

物理层：考虑的是怎样在传输媒体上传输数据比特流，而不是具体的传输媒体。物理层的作用时尽可能屏蔽传输媒体和通信手段的差异，使数据链路层感觉不到这些差异。

**OSI**：

表示层：数据压缩、加密及数据描述，这使得应用程序不必关心各台主机中数据内部格式不同的问题。

会话层：建立及管理会话。

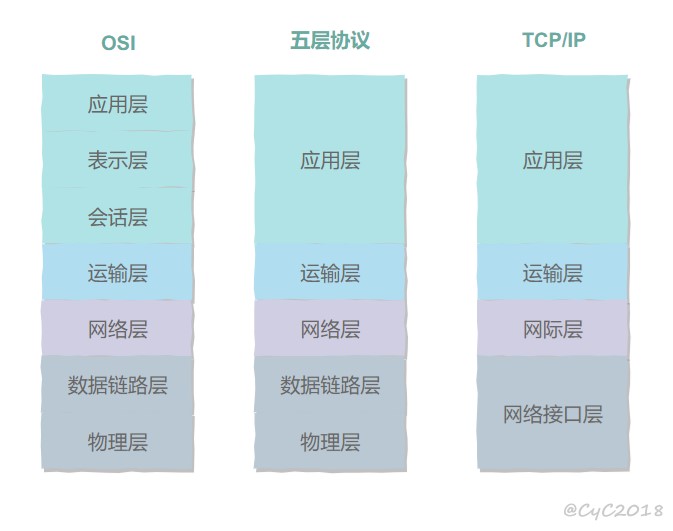
**TCP/IP：**

将五层协议中的数据链路层和物理层合并为网络接口层

**数据在各层之间的传递过程**：

在向下的过程中需要添加下层协议所需要的首部或者尾部，而向上的过程中需要不断拆开首部和尾部。

路由器只有下面三层协议，因为路由器位于网络核心中，不需要为进程或者应用程序提供服务，因此也就不需要传输层和应用。



# 物理层

**通信方式**：单工、半双工、全双工

**带通调制**：带通调制把数字信号转换为模拟信号。

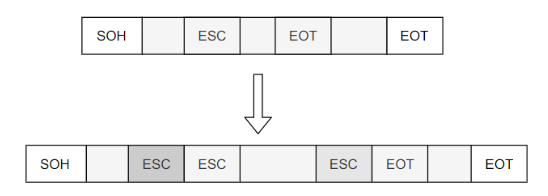
# 数据链路层

主要为同一链路上的主机提供数据传输服务。

**封装成帧**：将网络层传输下来的分组添加首部和尾部，用于标记帧的开始和结束。

**透明传输**：透明表示一个实际存在的事物看起来好像不存在。主要针对帧的数据部分包含与首尾部相同的内容，易引起帧开始结束错误判定的问题。

需要在数据部分出现首部尾部相同的内容前面插入转义字符。如果数据部分出现转义字符，那么就在转义字符前面再加个转义字符。在接收端进行处理之后可以还原出原始数据。这个过程透明传输的内容是转义字符，用户察觉不到转义字符的存在。

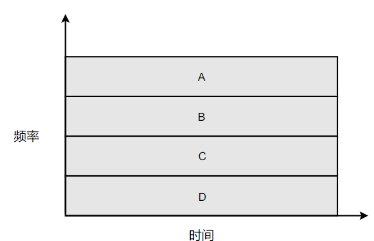
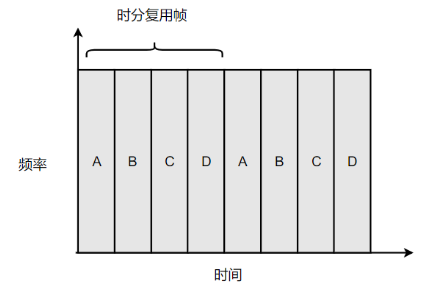


**差错检测**：目前数据链路层广泛使用了循环冗余检验(CRC)来检查比特差错。

**信道分类**：广播信道（一对多）；点对点信道（一对一）；

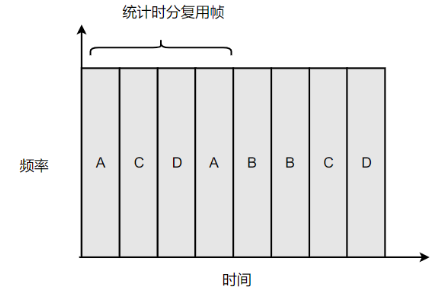
广播信道由于所有节点都在同一信道上发送数据，因此很容易发生冲突，必须转门的控制方法进行协调：信道复用技术；CSMA/CD技术

**信道复用技术**：1、频分复用；2、时分复用；3、统计时分复用；4、波分复用；码分复用；

频分复用 时分复用

统计时分复用：是对时分复用的一种改进，不固定每个用户在时分复用帧中的位置，只要有数据就集中起来组成统计时分复用帧然后发送。

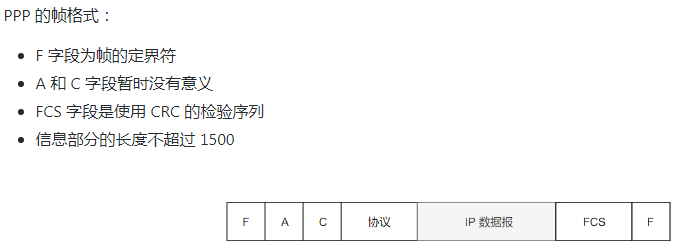


波分复用：光的频分复用。由于光的频率很高，因此习惯上用波长而不是频率来表示所使用的光载波。

码分复用：为每个用户分配 m bit 的码片，并且所有的码片正交，对于任意两个码片S和T,1/(S\*T) = 0。

**CSMA/CD协议**：载波监听多点接入/碰撞检测；多点接入：总线型网络；载波监听：每个主机必须不停地监听信道；碰撞检测：。

**PPP协议**：用户计算机和ISP进行通信时使用的数据链路层协议。



**MAC地址**：链路层地址，长度为6字节，用于唯一标识网路适配器。一台主机拥有多少个网络适配器就有多少个 MAC 地址。例如笔记本电脑普遍存在无线网络适配器和有线网络适配器，因此就有两个 MAC 地址。

**局域网**：是一种典型的广播信道。有：以太网、令牌环网、FDDI、ATM等局域网技术。

以太网：

交换机：

虚拟局域网

# 网络层

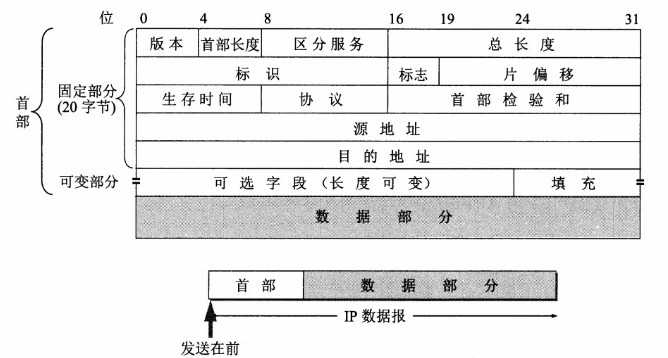
因为网络层是整个互联网的核心，因此应当让网络层尽可能简单。网络层向上只提供简单灵活的、无连接的、尽最大努力交互的数据报服务。使用 IP 协议，可以把异构的物理网络连接起来，使得在网络层看起来好像是一个统一的网络。

与 IP 协议配套使用的还有三个协议：

地址解析协议 ARP（Address Resolution Protocol）

网际控制报文协议 ICMP（Internet Control Message Protocol）

网际组管理协议 IGMP（Internet Group Management Protocol）



IP数据报的格式

**IP 地址的编址方式经历了三个历史阶段：**

分类：由两部分组成，网络号和主机号，其中不同分类具有不同的网络号长度，并且是固定的；IP:: = {<网络号>,<主机号>}

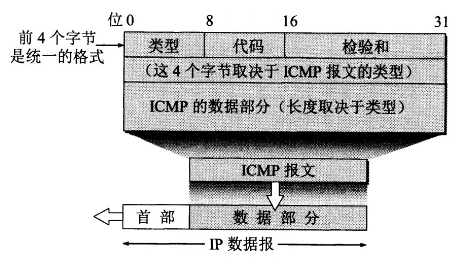
子网划分：通过在主机号字段中拿一部分作为子网号，把两级 IP 地址划分为三级 IP 地址；IP:: = {<网络号>,<子网络号>,<主机号>}

无分类：无分类编址 CIDR 消除了传统 A 类、B 类和 C 类地址以及划分子网的概念，使用网络前缀和主机号来对 IP 地址进行编码，网络前缀的长度可以根据需要变化；IP 地址 ::= {< 网络前缀号>, < 主机号 >}。

**地址解析协议ARP**：网络层实现主机之间的通信，而链路层实现具体每段链路之间的通信。因此在通信过程中，IP 数据报的源地址和目的地址始终不变，而 MAC 地址随着链路的改变而改变。

每个主机都有一个 ARP 高速缓存，里面有本局域网上的各主机和路由器的 IP 地址到 MAC 地址的映射表。

**网际控制报文协议ICMP**：ICMP 是为了更有效地转发 IP 数据报和提高交付成功的机会。它封装在 IP 数据报中，但是不属于高层协议。



网际控制报文协议

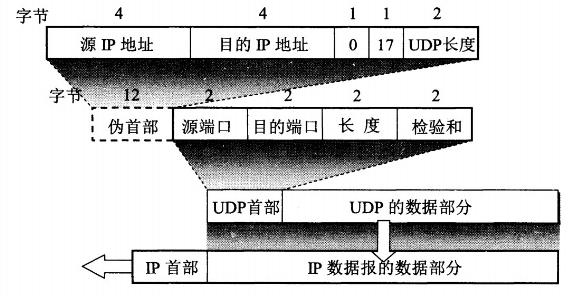
# 传输层

网络层只把分组发送到目的主机，但是真正通信的并不是主机而是主机中的进程。传输层提供了进程间的逻辑通信，传输层向高层用户屏蔽了下面网络层的核心细节，使应用程序看起来像是在两个传输层实体之间有一条端到端的逻辑通信信道。

**UDP 和 TCP 的特点**：

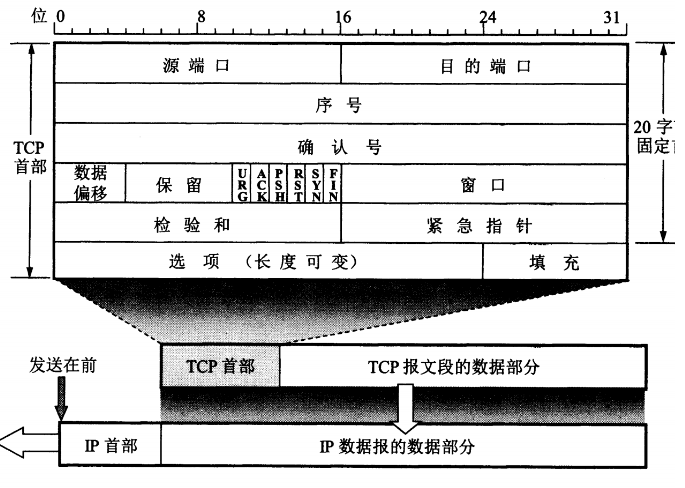
用户数据报协议 UDP（User Datagram Protocol）是无连接的，尽最大可能交付，没有拥塞控制，面向报文（对于应用程序传下来的报文不合并也不拆分，只是添加 UDP 首部），支持一对一、一对多、多对一和多对多的交互通信。

传输控制协议 TCP（Transmission Control Protocol）是面向连接的，提供可靠交付，有流量控制，拥塞控制，提供全双工通信，面向字节流（把应用层传下来的报文看成字节流，把字节流组织成大小不等的数据块），每一条 TCP 连接只能是点对点的（一对一）。



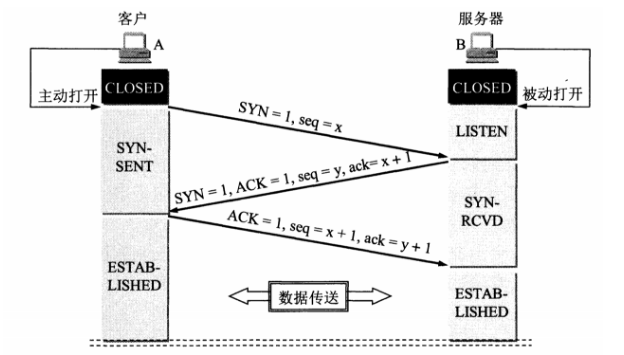
UDP用户数据报的首部和伪首部

UDP首部字段只有 8 个字节，包括源端口、目的端口、长度、检验和。12 字节的伪首部是为了计算检验和临时添加的。



TCP报文段的首部格式

**TCP的三次握手**：



用三次报文握手建立TCP连接

假设 A 为客户端，B 为服务器端。

首先 B 处于 LISTEN（监听）状态，等待客户的连接请求。

1）A 向 B 发送连接请求报文，SYN=1，ACK=0，选择一个初始的序号 x。

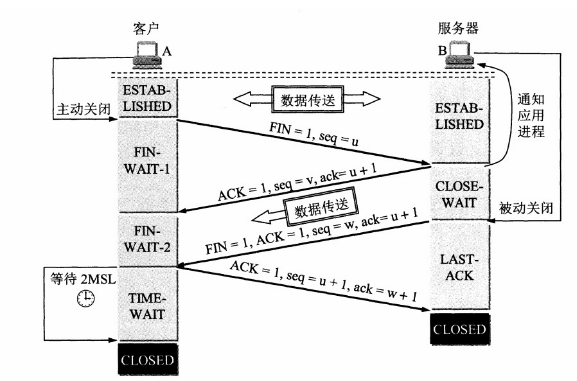
B 收到连接请求报文，如果同意建立连接，则向 A 发送连接确认报文，SYN=1，ACK=1，确认号为 x+1，同时也选择一个初始的序号 y。

2）A 收到 B 的连接确认报文后，还要向 B 发出确认，确认号为 y+1，序号为 x+1。

3）B 收到 A 的确认后，连接建立。

第三次握手是为了防止失效的连接请求到达服务器，让服务器错误打开连接。

**TCP的四次挥手**：



TCP连接释放的过程

A 发送连接释放报文，FIN=1。

B 收到之后发出确认，此时 TCP 属于半关闭状态，B 能向 A 发送数据但是 A 不能向 B 发送数据。

当 B 不再需要连接时，发送连接释放报文，FIN=1。

A 收到后发出确认，进入 TIME-WAIT 状态，等待 2 MSL（最大报文存活时间）后释放连接。

B 收到 A 的确认后释放连接。

四次挥手的原因：客户端发送了 FIN 连接释放报文之后，服务器收到了这个报文，就进入了 CLOSE-WAIT 状态。这个状态是为了让服务器端发送还未传送完毕的数据，传送完毕之后，服务器会发送 FIN 连接释放报文。

TIME\_WAIT：客户端接收到服务器端的 FIN 报文后进入此状态，此时并不是直接进入 CLOSED 状态，还需要等待一个时间计时器设置的时间 2MSL。这么做有两个理由：1）确保最后一个确认报文能够到达。如果 B 没收到 A 发送来的确认报文，那么就会重新发送连接释放请求报文，A 等待一段时间就是为了处理这种情况的发生；2）等待一段时间是为了让本连接持续时间内所产生的所有报文都从网络中消失，使得下一个新的连接不会出现旧的连接请求报文。

**TCP可靠传输**：TCP 使用超时重传来实现可靠传输：如果一个已经发送的报文段在超时时间内没有收到确认，那么就重传这个报文段。

一个报文段从发送再到接收到确认所经过的时间称为往返时间 RTT，加权平均往返时间 RTTs 计算如下：[https://camo.githubusercontent.com/02e9a4a23b05bcf814d9e5d0c1b79a6bb5ae2aee/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f525454733d28312d61292a2852545473292b612a525454](https://camo.githubusercontent.com/02e9a4a23b05bcf814d9e5d0c1b79a6bb5ae2aee/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f525454733d28312d61292a2852545473292b612a525454)，其中，0 ≤ a ＜ 1，RTTs 随着 a 的增加更容易受到 RTT 的影响。

超时时间 RTO 应该略大于 RTTs，TCP 使用的超时时间计算如下：

[https://camo.githubusercontent.com/da3da0a6cad7ef2af7d7adcd2940277780c35325/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f52544f3d525454732b342a5254545f64](https://camo.githubusercontent.com/da3da0a6cad7ef2af7d7adcd2940277780c35325/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f52544f3d525454732b342a5254545f64)，其中 RTTd 为偏差的加权平均值。

**TCP的滑动窗口**：窗口是缓存的一部分，用来暂时存放字节流。发送方和接收方各有一个窗口，接收方通过 TCP 报文段中的窗口字段告诉发送方自己的窗口大小，发送方根据这个值和其它信息设置自己的窗口大小。

**TCP的流量控制**：为了控制发送方发送速率，保证接收方来得及接收。

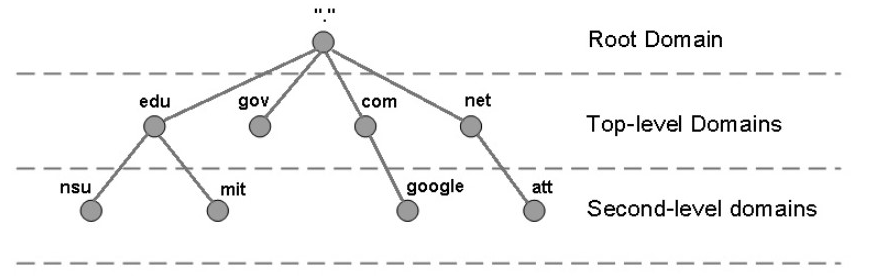
**TCP的拥塞控制**：TCP 主要通过四个算法来进行拥塞控制：慢开始、拥塞避免、快重传、快恢复。发送方需要维护一个叫做拥塞窗口（cwnd）的状态变量，注意拥塞窗口与发送方窗口的区别：拥塞窗口只是一个状态变量，实际决定发送方能发送多少数据的是发送方窗口。

# 应用层

域名系统

DNS 是一个分布式数据库，提供了主机名和 IP 地址之间相互转换的服务。这里的分布式数据库是指，每个站点只保留它自己的那部分数据。

域名具有层次结构，从上到下依次为：根域名、顶级域名、二级域名。



DNS 可以使用 UDP 或者 TCP 进行传输，使用的端口号都为 53。大多数情况下 DNS 使用 UDP 进行传输，这就要求域名解析器和域名服务器都必须自己处理超时和重传来保证可靠性。在两种情况下会使用 TCP 进行传输：

如果返回的响应超过的 512 字节（UDP 最大只支持 512 字节的数据）。

区域传送（区域传送是主域名服务器向辅助域名服务器传送变化的那部分数据）。

**文件传输协议**：

FTP 使用 TCP 进行连接，它需要两个连接来传送一个文件：控制连接；数据连接。

根据数据连接是否是服务器端主动建立，FTP 有主动和被动两种模式：主动模式；被动模式。

**动态主机配置协议**：HCP (Dynamic Host Configuration Protocol) 提供了即插即用的连网方式，用户不再需要去手动配置 IP 地址等信息。DHCP 配置的内容不仅是 IP 地址，还包括子网掩码、网关 IP 地址。