数据交换方式：

电路交换：整个数据报文直接从A-D

报文交换：整个数据报文A-B-C-D

分组交换：将数据报文分段，然后每段报文A-B-C-D

一般采用分组交换(占用带宽低，路由灵活，可靠性强)

时延：

发送时延：数据发送出去存在时延，数据帧长度/带宽

传播时延：数据传播存在时延，信道长度/光速

处理时延：路由器在收到分组时，需要时间处理

排队时延

RTT：数据传输的往返时间

五层协议体系：

应用层(应用层+表示层+会话层)：

**应用层(application-layer）的任务是通过应用进程间的交互来完成特定网络应用。**应用层协议定义的是应用进程间的通信和交互的规则。对于不同的网络应用需要不同的应用层协议(DNS，HTTP，HTTPS，SMTP协议等)。应用层交互的数据单元称为报文。

传输层：

**传输层主要任务就是负责向两台主机进程之间的通信提供通用的数据传输服务。**应用进程利用该服务传送应用层报文。

TCP(面向连接，可靠数据传输)：

TCP面向连接，点对点通信，会话之前要建立连接(3次握手)，结束会话释放连接(4次挥手)；

TCP提供可靠交付的服务。通过TCP连接传送的数据，无差错、不丢失、不重复、并且按序到达；

TCP提供全双工通信。TCP允许通信双方的应用进程在任何时候都能发送数据。TCP连接的两端都设有发送缓存和接收缓存，用来临时存放双方通信的数据；

UDP(无连接，保证数据传输的可靠性)：

面向无连接，可一对一，一对多，多对一，多对多通信

UCP不保证传输的可靠性(传输过程中可能丢失数据)

UDP没有拥塞控制，网络出现拥塞不会使源主机的发送速率降低(对实时应用很有用，如IP电话，实时视频会议等)

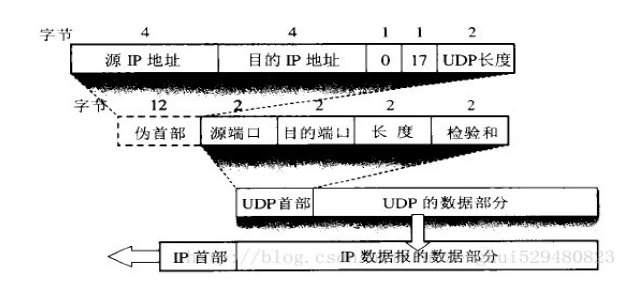
UDP首部开销小，只有8个字节，比TCP的20个字节首部短

TCP：面向连接 传输可靠 传输字节流 传输效率慢 占用资源多

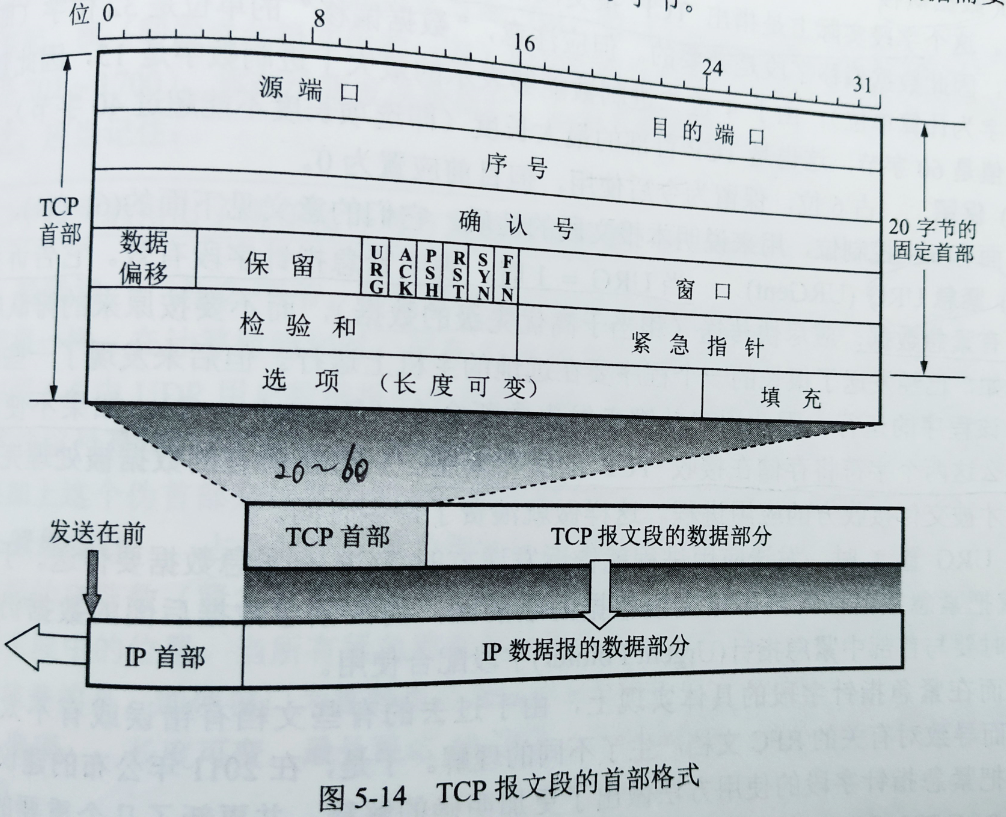
UDP：无连接 传输不可靠 传输数据报文 传输效率快 占用资源少

采用TCP还是UDP由应用层决定

UDP报文格式



TCP报文格式



TCP可靠传输原因：

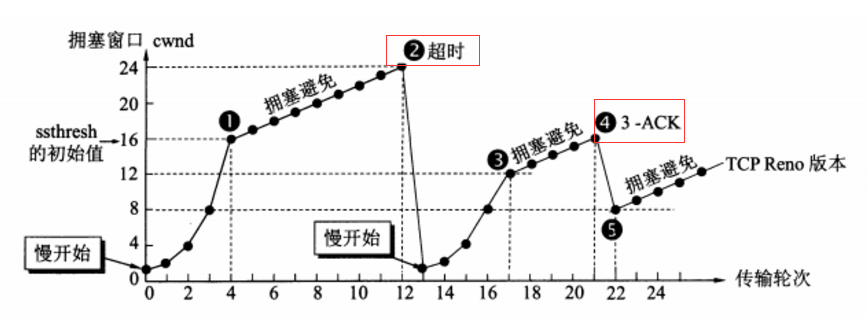
停止等待协议：TCP采用全双工通信，每发送完一个分组就停止发送，知道接收方发来确认消息；超过指定时间内未收到确认消息，发送方再次发送(超时重传)；接收方接收后，发送确认消息，但是确认消息丢失，发送方重传，接收方接收到重复的数据，丢弃重复数据，回传确认消息(确认丢失)；接收方收到消息后，发送确认消息，确认消息在网络中滞延，发送发会收到重复的确认消息，收到重复确认消息，但什么也不做(确认迟到)；

连续ARQ：

停等协议的优点就是简单，但是信道利用率低(需要收到确认消息后才发一个消息)；而连续ARQ是指一个窗口内可连续传输数据，而接收方回传的确认消息就是积累确认(TCP报文中的确认号是指接收方下一个报文希望接收的序列号，如果窗口中信息存在丢失，确认号就是丢失的序列号)；并且窗口的大小可以根据接收方返回的大小就行改变(TCP报文中的窗口)[滑动窗口协议]

TCP流量控制通过滑动窗口控制

TCP拥塞控制方法：(网络拥塞时，发送数据量就应减少，避免丢失) 慢开始，拥塞避免

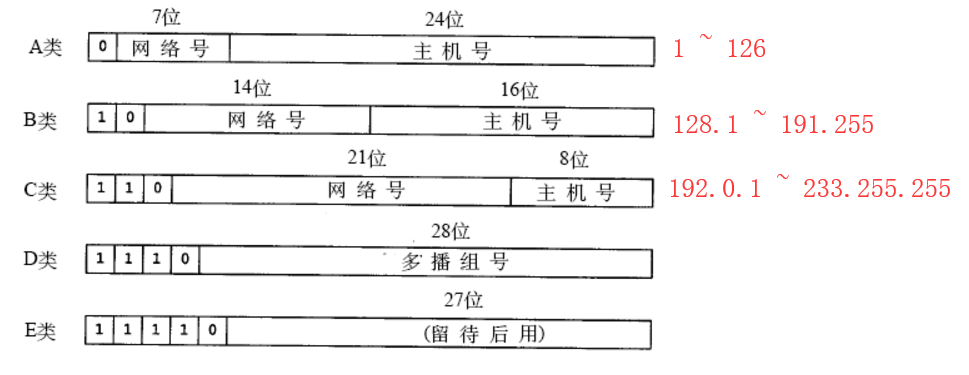


超时或3个ACK后，门限值降为原来的一半

网络层(IP层)：

**网络层(network layer)负责为分组交换网上的不同主机提供通信服务。**在发送数据时，网络层把运输层产生的报文段或用户数据报封装成分组和包进行传送。网络层的另一个任务就是选择合适的路由，使源主机运输层所传下来的分株，能通过网络层中的路由器找到目的主机。传输的IP数据报。

IP地址：



IP数据报：

格式：IP地址 首部 应用层数据

MAC地址在链路层使用，IP地址在网络层使用

ARP(地址解析协议)：

在主机ARP高速缓存中存放着本局域网内部IP地址和MAC地址对应的映射表，主机A在传输数据时，会在本地ARP中根据目的IP地址，查询到对应的MAC地址，如果ARP中没有，A会在本局域网内广播，找到IP地址对应MAC地址，并更新到ARP中，然后在链路层对MAC地址封装成MAC帧



固定部分20个字节，可变部分在0到40字节

版本(IPV4/IPV6) 区分服务(保留字段) 标识，标志，片偏移(对应用层数据分片时使用)

网络层转发过程：

判断目的IP地址是否在本地局域网内，如果在，直接在交换机上进行转发，跳转；链路层，如果不在，通过网关到达网络层，根据路由器中路由表找到下一路由器的IP地址，然后在路由器的ARP中将IP地址转换为MAC地址，跳转到数据链路层

路由器会记录到达相邻路由器的步数

路由表中为何不直接使用MAC地址，而采用IP地址：IP地址是为了隐藏底层的网络复杂性以及减少路由开销(存MAC地址在路由时消耗更多)

子网掩码和超网的作用：提高IP的利用率/减小路由表(减少路由表的变化次数)/提高IP地址的灵活性

链路层：

在两个相邻节点之间传送数据时，**数据链路层将网络层交下来的IP数据报组装程帧，在两个相邻节点间的链路上传送帧(封装成帧)。**链路层会对帧进行校验，如果发现差错，数据链路层就简单地丢弃这个出了差错的帧，以避免继续在网络中传送下去白白浪费网络资源。传输的是帧。

封装成帧：

将IP数据报组装成帧，格式 SOH 封装在帧中数据 EOT SOH帧头，EOT帧尾

透明传输：

在帧的数据中可能存在SOH或EOT，在封装成帧时，需添加转义符，SOH 转义为SOH ESC

差错检测

对接收到的帧进行检测，在传输过程中是否存在丢失或改变，采用方式CRC

帧格式常见的是PPP协议

格式：F A C 协议 IP数据报(可变长度) FCS F

字节：1 1 1 2 不超过1500字节 2 1

首尾F F作为头尾部

A C保留字段，但现在任未使用

协议：当字段为0x0021时，ppp帧的信息字段为IP数据报

FCS：CRC校验的帧检测序列

早期以太网中的计算机都连接到一条总线上的，采用的是半双工通信方式，所以有了CSMA/CD协议：

多点接入：多台计算机在一条总线上

载波监听：在发送之前/发送中，检测信道是否被占用

碰撞检测(边发送边监听)：适配器边发送数据边检测信道，检测自己在发送数据时其他站是否也在发送数据

当A发送数据时，会先检查信道是否被占用，如果在2个r(总线端到端的传播时延)内，无人占用信道，表示没有发送碰撞，A就可以发数据

当A在发送数据时，B也发送数据，AB会发送碰撞，采用截断二进制指数退避算法，计算出AB下次重传的时间(AB直接重传时间相差n个2r)，当碰撞的次数达到16次后，丢弃帧，并向高层报告。当发送碰撞后，需要发送干扰型号(32或48bit传输时间)，以便让其他机器知道。

以太网帧格式：

格式：目的地址 源地址 类型 IP数据报 FCS

字节： 6 6 2 4

传输源地址的目的是为了在以太网交换机的交换表做映射，方便下次找路

交换机不会平分带宽，每条带宽一样/集线器会平分带宽

物理层：

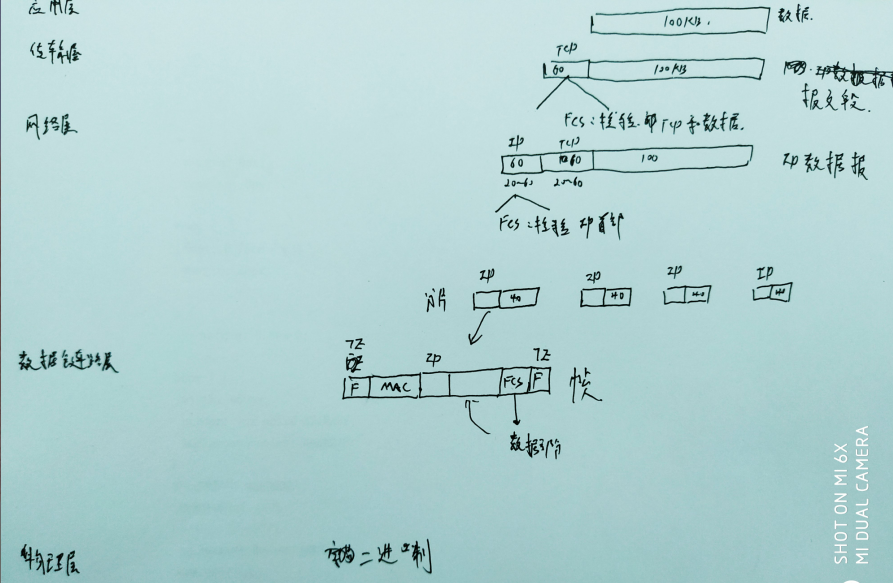
在物理层上所传送的数据单位是比特流。在光纤中传输数据。

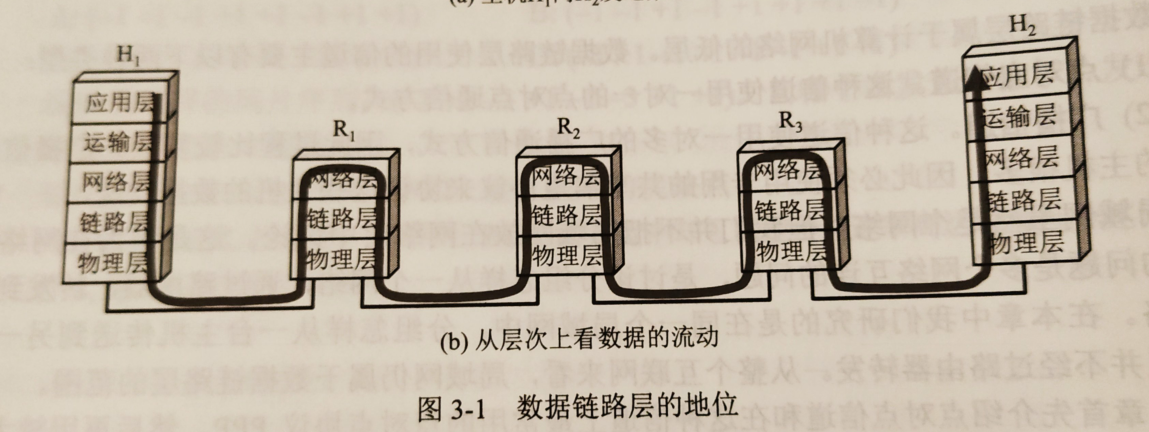
通信方式：单向通信/半双工通信(同一时间内，只能一段发送比特流)/全双工通信

信道复用方式：

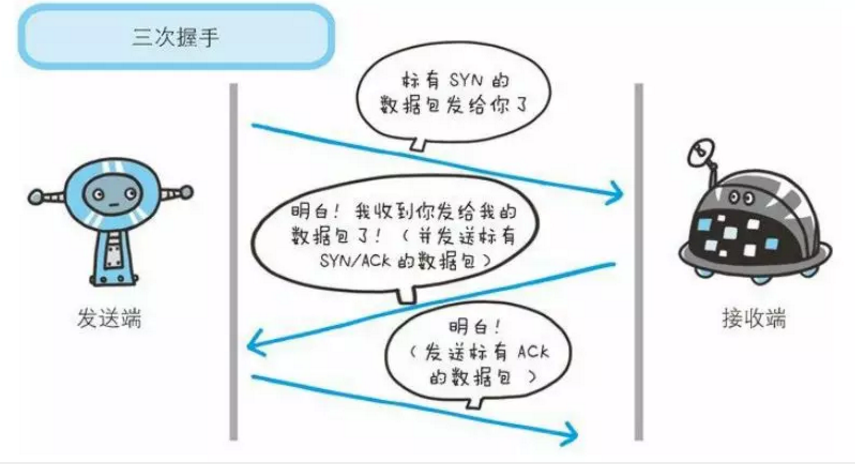
频分复用：根据传输的频率占用信道

时分复用：根据传输的时间占用信道





TCP三次握手和四次挥手：



客户端–发送带有SYN标志的数据包–一次握手–服务端

服务端–发送带有SYN/ACK标志的数据包–二次握手–客户端

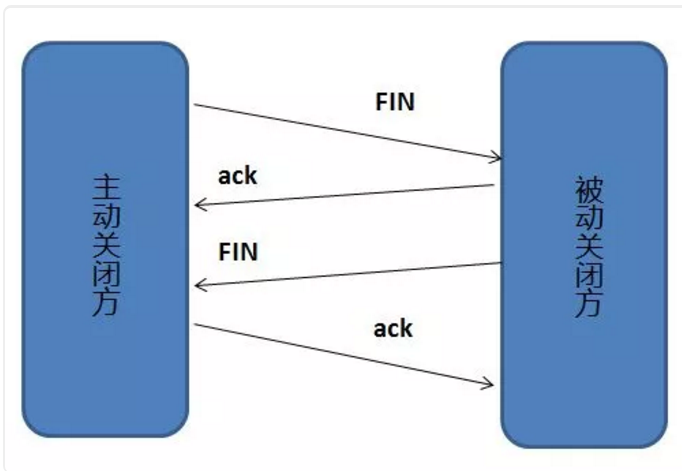
客户端–发送带有ACK标志的数据包–三次握手–服务端

回传SYN，告诉发送端，我接收到的信息确实就是你所发送的信号

ACK：接收方到发送方的通道还需要ACK信号来进行验证

**三次握手的目的是建立可靠的通信信道，三次握手最主要的目的就是双方确认自己与对方的发送与接收是正常的。**

当发送端发送第一个请求连接，但在网络中滞留，发送端再次发送请求，连接建立，连接释放。但是此时滞留的连接请求发送到接收端(第一次)，然后接收端发确认(第二次)，如果2次握手，连接建立(虽然发送端指定这是超时的请求，可以不建立连接，但是接收端在等待发送端建立连接)，导致资源消耗，如果采用3次，发送端知道这是一个超时请求，不发送第三次握手，这样接收端在一定时间内没收到消息就会不再等待连接(如果3次握手中有任何一次存在数据丢失，都是发送端再次向接收端发起请求，接收端永远不会发请求)



任何一端都可以发起关闭请求

当发送FIN(第一次)，对方确认(ack)后，处于半关闭状态，待对方发起FIN(第三次)后，关闭方再确认(ack

)即可关闭

A和B打电话，通话即将结束后，A说“我没啥要说的了”，B回答“我知道了”，但是B可能还会有要说的话，A不能要求B跟着自己的节奏结束通话，于是B可能又巴拉说了一通，最后B说“我说完了”，A回答“知道了”，这样通话才算结束。

TCP/IP协议的可靠性：

1：应用数据被分割成TCP认为最适合发送的数据块

2：超时重传

3：TCP给发送的每一个包进行编号，接收方对数据包进行排序，把有序数据传送给应用层。

4：校验和：TCP将保持它首部和数据的检验和。

5：接收端会丢弃重复的数据

6：流量控制

7：拥塞控制

8：停等协议(ARQ)：



浏览器中输入url到页面显示的过程

HTTP的长连接/短连接：

HTTP1.0默认短连接，即一次请求后，服务器与浏览器就断开

从HTTP1.1开始，默认使用长连接，在一次请求后，会保持一段时间的连接，不同的服务器可以设不同的连接保持时间，实现长连接需要客户端和服务端都支持长连接。

拥塞控制与网络的拥堵情况相关联，而流量控制与接收方的缓存状态相关联。

TCP/IP协议与HTTP协议：

TPC/IP协议是传输层协议，主要解决数据如何在网络中传输，而HTTP是应用层协议，主要解决如何包装数据。

HTTP协议作应用层协议，以封装HTTP 文本信息，然后使用TCP/IP做传输层协议将它发到网络。

RPC与HTTP：

RRPC主要是基于TCP/IP协议的(传输层)，而HTTP服务主要是基于HTTP协议的(应用层)，在传输效率上RPC的请求报文比HTTP的请求报文少，效率高；在对数据进行序列化和反序列化的时候，RPC基于二进制，HTTP基于JSON，XML，RPC效率更高；RPC自带负载均衡，而HTTP依赖第三方(Nginx)；但是RPC只支持同一门语言之前调用；RPC调用更像是直接调用本地服务；

HTTP端口80，HTTPS端口443

