# 网络分层

1 发送数据MTU( 最大发送单元)

a 在建立tcp连接，3次握手的时候，获取对方的MSS, MSS加上TCP头（20个字节）和IP头（20个字节），就是MTU(最大传输单元)

b 取自己的MTU和对方的MTU取最小值。大多数MTU是1500个字节（即MSS为1460），有些网络启用了巨帧（Jumbo Frame）,达到9000个字节（即MSS为8960）

c 根据MTU对数据进行切分

# TCP的连接启蒙

TCP和UDP的区别：

TCP需要先建立连接；UDP不需要建立连接。

TCP提供有序的传输，切分的数据段都会有序号(seq 消息的偏移量)，接收方接收到乱序的包后，根据序号重新排序。

## Tcp的几个参数

1 Seq=数值 数据段的序号，即数据段的偏移量。

2 len=数值 数据段的长度，不包括tcp头。

3 ack=数值 确认号。收到对方 在此 之前的所有数据。

## TCP头中的标志位

1 SYN ： 请求建立连接

2 FIN： 请求断开连接

3 RST： 拒绝一个无效的请求

## TCP建连的3次握手

1 [ SYC]

2 返回的ack值是对方的seq+1

## TCP断开连接的4次挥手

1 [FIN,ACK]

2 返回的ack值是对方的seq+1

# TCP窗口

1 什么是TCP发送窗口？

客户端和服务端数据交互的时候，发送方能给接收方发送多少数据量，然后等待对方的确认，再继续发送。

2 TCP发送窗口与MSS之间的关系

MSS 是最大数据段大小，取客户端和服务端的最小值。即单个包的最大字节数。（不包含TCP头），决定了要分多少个包来发送。

TCP窗口：接收方最大缓存的数据量（TCP头不计入在内），需要确认一次。

3 以上的定义是剔除了网络因素的影响。实际上，发送窗口除了受接收窗口的影响，还收到网络因素的影响，

4 接收窗口是在变化中的。随着有没有确认而告知对方剩余多少。

5 TCP Window Scale

# 重传的讲究

1 拥塞点： 网络因素，数据被拥塞，从而发生丢包。

我们的目的是在不断变化的网络状况下，避免拥塞，避免重传。

2 虚拟拥塞窗口策略:

算法

3 如果遇到拥塞，怎么办？

拥塞-》迟迟不能收到确认-》认为包丢失-》重传-》调整阻塞窗口，重新进入慢启动

RTO:Retransmission TimeOut 重传超时时间。

4 快速重传

当发生其他原因影响的单个丢包，（如校验码不对），而不是拥塞导致的丢包。

当发送方收到接收方3个或以上重复确认（Dup Ack）时，会知道相应的包丢失了，会进行重传。

5 结论：

a 没有拥塞时，tcp发送窗口越大，性能越好。

B 当发生拥塞时，超时重传对性能影响太大，要限制tcp发送窗口（通过限制接收窗口来达到减少发送窗口）

C 快速重传对性能影响较小。

D 丢包对小文件的影响比大文件要大。

# 延迟确认与Nagle算法

即延迟确认和延迟发送，不直接提高性能，而是把包尽可能的一起发出，减轻网络负担。

# UDP

1 设计简单，传输层 拿到应用层数据后，数据+8个字节的头（端口，包长度，校验码）直接交给下一个层。如果数据过大，网络层再分片（加重网络层的负担）。

2 不需要建立连接，没有seq和ack号等概念。丢包没有重传机制。

而是由应用层来处理丢包，重新发送全部的数据。

# FTP协议