## 每一条连接唯一地被通信两端的两个端点 (socket pair) 所确定

socket=(IP地址:端口号)

建立连接,三次握手

面向连接协议

保证客户端发送的最后一个ACK报文段能够到达服务器

防止"已失效连接请求报文段"出现在本连接中

客户端要有个TIME WAIT状态,设置为2个MSL

释放连接,四次挥手

滑动窗口协议

发送窗口是根据就收窗口设置的,但在同一时刻发送窗口并不总是和接收窗口一样大

对于未按序到达的数据TCP标准没有规定怎么处理,一般是先临时存下,等缺失字节流到达后,按序一并交付上层

接收方有累计确认功能,但不应过分推迟发送确认

新的RTT(s)=(1- $\alpha$ )\*(旧的RTT(s)) +  $\alpha$ \*(新的RTT样本)  $\alpha$ =0.125

可靠传输

RTO=RTT(s) + 4\*RTT(d)

RTT(d)初始值为当时RTT样本值的一半  $\beta$ =0.25 新的RTT(d)=(1- $\beta$ )\*(旧的RTT(d)) +  $\beta$ \*(RTT(s)-新的RTT样本)

Karn算法: 若报文段重传了,则不采用本次往返时间样本。Karn算法修改:报文段每重传一次,就把超时重传时间RTO设置为旧的重传时间的2倍

SACK选项对不连续的字节快发送确认(不常用)

全双工通信

**TCP** 

面向字节流, '流' 指流入到进程或从进程流出的字节序列

发送发窗口不能超过接收方给出的接收窗口的数值

TCP每个连接都设有持续计时器,打破因报文段丢失造成的死锁局面

流量控制

Nagle算法: 当到达缓存的数据已达到发送窗口的一半或MSS时立即发送一个报文段

当接收方的缓存有一个MSS或一半的空闲空间时发送确认报文

基于窗口的拥塞控制,四种算法:慢开始,拥塞避免,快重传,快恢复

初始cwnd的设置规则:

若SMSS>2190,cwnd=2\*SMSS 若1095<SMSS<=2190,cwnd=3\*SMSS 若 SMSS<=1095,cwnd=4\*SMSS

cwnd每次增加量=min(N,SMSS)

拥塞控制

若cwnd < ssthresh时执行慢开始算法;若cwnd > = ssthresh时执行拥塞避免算法。若超时则调整ssthresh = ssthresh/2,cwnd = 1;若连续收到3个重复ACK调整cwnd = ssthresh/2,ssthresh = ssthresh/2

发送窗口的上限值=min(rwnd,cwnd)

最小长度20字节, 其选项字段可变为4n字节

源端口,目标端口,序号,确认号,数据偏移(首部长度),保留,紧急URG,确认ACK,推送PSH,复位RST,同步SYN,终止FIN,窗口,检验和,紧急指针,选项

首部格式

