



HTTP

请求/响应

SMTP

命令/响应
POP3、IMAP

DNS(Domain Name System)

查询/回复

tcp的流量控制

- 滑动窗口
 - 机理：接收端将自己可以接收的缓冲区大小放入 TCP 首部中的“窗口大小”字段, 通过ACK端通知发送端;窗口大小字段越大, 说明网络的吞吐量越高;接收端一旦发现自己的缓冲区快满了, 就会将窗口大小设置成一个更小的值通知给发送端;发送端接受到这个窗口之后, 就会减慢自己的发送速度;如果接收端缓冲区满了, 就会将窗口置为0; 这时发送方不再发送数据, 但是需要定期发送一个窗口探测数据段, 使接收端把窗口大小告诉发送端。

tcp的拥塞控制

- 与流量控制的区别：流量控制往往是指点对点通信量的控制，是个端到端的问题。流量控制所要做的就是控制发送端发送数据的速率，以便使接收端来得及接收。
- 拥塞窗口：发送方为一个动态变化的窗口叫做拥塞窗口，拥塞窗口的大小取决于网络的拥塞程度。发送方让自己的发送窗口=拥塞窗口，但是发送窗口不是一直等于拥塞窗口的，在网络情况好的时候，拥塞窗口不断的增加，发送方的窗口自然也随着增加，但是接受方的接受能力有限，在发送方的窗口达到某个大小时就不再发生变

化了。

- **慢启动**算法的思路：主机开发发送数据报时，如果立即将大量的数据注入到网络中，可能会出现网络的拥塞。慢启动算法就是在主机刚开始发送数据报的时候先探测一下网络的状况，如果网络状况良好，发送方每发送一次文段都能正确的接受确认报文段。那么就从小到大的增加拥塞窗口的大小，即增加发送窗口的大小。
- **拥塞避免**的思路：是让cwnd缓慢的增加而不是加倍的增长，每经历过一次往返时间就使cwnd增加1，而不是加倍，这样使cwnd缓慢的增长，比慢启动要慢的多。
 - 无论是慢启动算法还是拥塞避免算法，只要判断网络出现拥塞，就要把慢启动开始门限(ssthresh)设置为设置为发送窗口的一半 (≥ 2)，cwnd(拥塞窗口)设置为1，然后在使用慢启动算法，这样做的目的能迅速的减少主机向网络中传输数据，使发生拥塞的路由器能够把队列中堆积的分组处理完毕。拥塞窗口是按照线性的规律增长，比慢启动算法拥塞窗口增长快的多。
- **快重传**算法要求首先接收方收到一个失序的报文段后就立刻发出重复确认，而不要等待自己发送数据时才进行捎带确认,如果接收方一连收到三个重复的ACK,那么发送方不必等待重传计时器到期，由于发送方尽早重传未被确认的报文段。
- **快恢复**：1. 当发送方连续接收到三个确认时，就执行乘法减小算法，把慢启动开始门限 (ssthresh) 减半，但是接下来并不执行慢开始算法。2. 此时不执行慢启动算法，而是把cwnd设置为ssthresh的一半，然后执行拥塞避免算法，使拥塞窗口缓慢增大。