1.2 TCP协议简介

TCP将应用层发送的数据分段，形成多个报文，并为数据的每个字节分配序列号，通过序列号机制实现报文确认、丢失和错误报文重传、乱序报文恢复顺序等功能，为应用层提供面向连接的可靠数据传输功能；通过应答报文中的窗口大小字段，TCP控制发送方的最大发送数据量，实现简单的流量控制功能；TCP通过IP地址和端口号形成Socket，用一个Socket对表示一个具体连接，使得系统中的多个进程能够同时使用TCP通信，实现多路并传功能；通过报文超时和重复确认，TCP检测链路的拥塞情况，在拥塞发生时减慢发送数据的速度，提供拥塞控制功能。本节将简单介绍TCP协议，分析TCP提供上述功能的原理。

1.2.1 TCP头部

TCP位于传输层，其报文被封装在IP报文中，如图1.1所示。



图1.1 TCP报文封装

TCP头部格式如图1.2所示，在没有选项的情况下，其头部大小为20字节。



图1.2 TCP头部格式

TCP头部中的源端口号和目的端口号用于标识数据发送进程和数据接收进程，IP地址加端口号可标识出一个Socket，则每条TCP连接可由源IP地址、目的IP地址、源端口号、目的端口号唯一地标识出。

32位序列号为报文中所含数据的第一个字节在发送数据字节流中被分配的序列号；32位确认号为接收方期待接收的下一段数据中第一个字节的序列号，确认号只有在ACK标志被置1时才有效。

在TCP连接建立时，SYN标志被置1，在连接拆除时，FIN标志被置1。

16位窗口大小表示接收方能接收的最大数据长度，被用于控制发送方的最大数据发送量，实现TCP的流量控制功能。

16位校验和包含了TCP头部和负载数据，由发送方计算，接收方验证，用于保证数据的正确性。

1.2.2 TCP连接的建立和拆除

TCP是面向连接的协议，在发送数据之前需要先建立连接。TCP连接建立的过程被称为三次握手，如图1.3所示，其细节为：

1. 客户端向服务器端发送SYN报文，其中包含服务器的目的端口号和客户端初始序列号；

2. 服务器端回应SYN报文，其中包含服务器端的初始序列号，确认号为客户端初始序列号加1；

3. 客户端向服务器端发送确认报文，确认号为服务器端初始序列号加1。

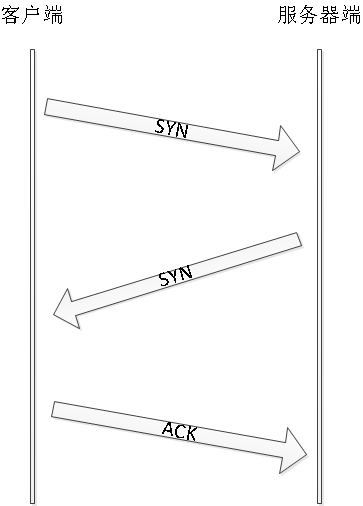


图1.3 TCP连接的建立

建立TCP连接时客户端和服务器端需要交换三个报文，而连接拆除时需要交换四个报文，如图1.4所示。ACK报文中的确认号为接收到FIN报文中的序列号加1。

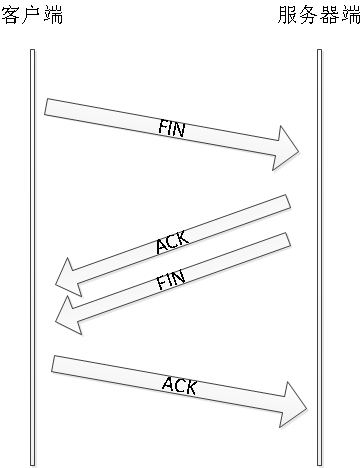


图1.4 TCP连接的拆除

上述过程为最理想的情形，实际情况远比此复杂，TCP为连接设立了多个状态，在连接建立和拆除时的状态转换图如图1.5所示。



图1.5 TCP状态转换

1.2.3 TCP选项

TCP头部可以包含选项，常见的选项格式如图1.6所示。每种选项的开头都有一个字节的类型字段，表明该选项的操作类型；类型为0或1的选项无其他字段，其他类型的选项有一个字节的长度字段，表示该选项的总长度。



图1.6 TCP选项格式

MSS（Maximum Segment Size）声明接收方所能接收最大报文的大小，进而控制发送方发送报文大小，避免报文在传输过程中被分段。连接建立时，SYN报文中包含MSS字段，连接建立后另一端将按照SYN报文中的MSS值发送大小合适的数据报文。

窗口缩放选项使得接收方声明的窗口大小突破16比特的限制，实际的窗口大小为左移偏移数量后得到的值。

时间戳选项让TCP可为每个报文加时间戳，接收方在确认报文中回复该时间戳，使得数据发送方可以更好地计算RTT（Round Trip Time）。TCP连接建立时，主动发起连接的一方在SYN报文中添加时间戳选项，若收到另一端的SYN报文中也包含时间戳选项，则表明另一端也支持时间戳选项，在此后的数据报文中将添加时间戳选项，否则不添加。

1.2.4 TCP滑动窗口机制

TCP提供大小由数据接收方声明的滑动窗口机制，控制发送方发送报文的数量，实现简单的流量控制。如图1.7所示，接收方声明的窗口被分为两部分：一部分为已发送但未确认的字节，另一部分为可用窗口，包含能够发送的字节。



图1.7 TCP滑动窗口机制

如图1.8所示，随着数据的发送和接收方确认数据，滑动窗口向右移动，窗口左端和右端的移动使得窗口大小不断变化：

1. 当已发送的数据被确认时，窗口左端向右移动，这被称为窗口关闭；

2. 当接收方读取数据后，将释放接收数据缓存，声明更大的窗口，使得发送方窗口右端向右移动，更多数据能被发送，这被称为窗口打开。



图1.8 TCP窗口的关闭和打开

1.2.5 RTT和数据重传

为实现可靠的数据传输服务，TCP要求数据接收方向发送方发送ACK报文，但ACK报文可能会丢失，为处理这种情况，TCP在发送数据后设定重传计时器，若在重传计时器超时后仍未收到ACK报文，则TCP重传数据。

设定重传计时器的基础在于连接上RTT的计算，由于网络流量的变化，连接上数据传输的路径会发生变化，进而引起RTT的不断变化，所以TCP需要实时的计算RTT值，不断修改重传计时器。用表示测量所得RTT值，表示TCP计算所得RTT值，其计算公式为(1-1)：

 (1-1)

其中为平滑因子，通常取0.9。根据RTT计算超时时间RTO（Retransmission Timeout）的公式为(1-2):

 (1-2)

其中通常取2。

1.2.6 慢启动和拥塞避免

TCP通过慢启动和拥塞避免算法实现简单的拥塞控制，算法在每条连接上增加两个变量：拥塞窗口大小cwnd和慢启动阈值ssthresh，如图1.9所示，算法的核心思想为：

1. 初始化每条连接的cwnd为一个MSS，初始化ssthresh为65536字节；

2. 发送方不发送超过cwnd和ssthresh中最小值的数据；

3. 重传计时器超时或收到重复ACK报文表示拥塞发生，此时将cwnd和ssthresh最小值的一半赋值给ssthresh，若重传计时器超时，将cwnd设定为一个MSS；

4. 收到连接另一端发送的ACK报文时，增加cwnd。若cwnd小于等于ssthresh，执行慢启动算法，给cwnd增加一个MSS，这使得cwnd指数级的增长；当cwnd大于ssthresh时，执行拥塞避免算法，在一个RTT时间内无论收到多少ACK报文，cwnd都只增加一个MSS。

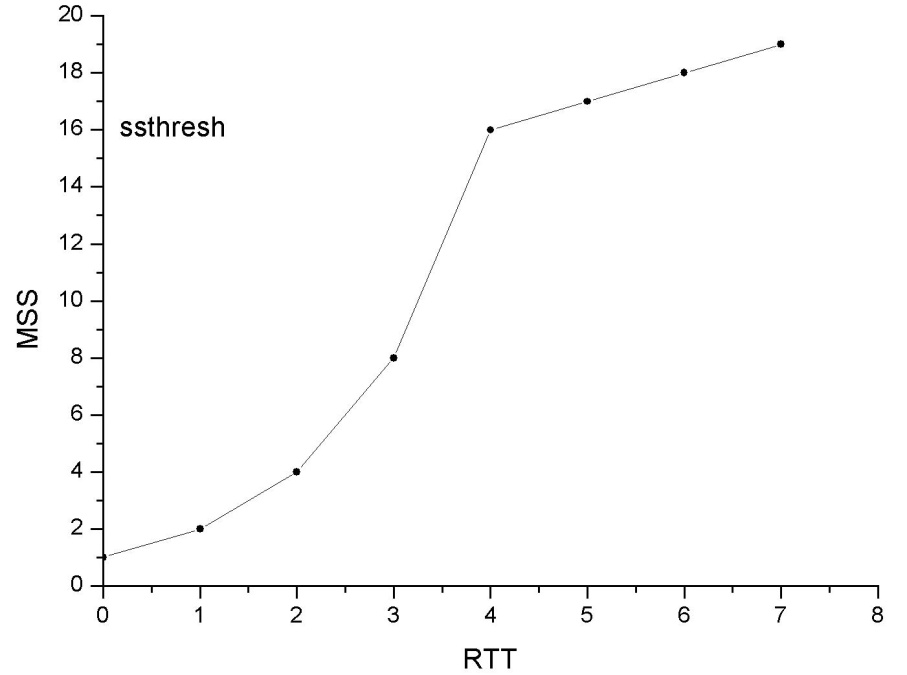


图1.9 慢启动和拥塞避免