

http三次握手和四次挥手

TCP(Transmission Control Protocol) 传输控制协议

TCP是主机对主机层的传输控制协议，提供可靠的连接服务，采用三次握手确认建立一个连接:

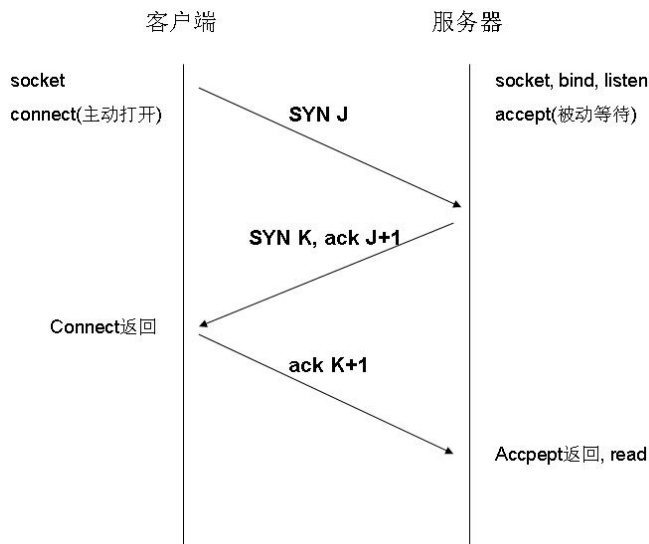
位码即tcp标志位,有6种标示:SYN(synchronous建立联机) ACK(acknowledgement 确认) PSH(push传送) FIN(finish结束)

RST(reset重置) URG(urgent紧急)

Sequence number(顺序号码) Acknowledge number(确认号码)



TCP/IP基础——TCP三次握手



TCP/IP基础——TCP三次握手

第一次握手: 主机A发送位码为 $\text{syn}=1$,随机产生 $\text{seq number}=1234567$ 的数据包到服务器, 主机B由 $\text{SYN}=1$ 知道, A要求建立联机;

第二次握手: 主机B收到请求后要确认联机信息, 向A发送 $\text{ack number}=(\text{主机A的seq}+1), \text{syn}=1, \text{ack}=1$,随机产生 $\text{seq}=7654321$ 的包

第三次握手: 主机A收到后检查 ack number 是否正确, 即第一次发送的 $\text{seq number}+1$,以及位码 ack 是否为1, 若正确, 主机A会再发送 $\text{ack number}=(\text{主机B的seq}+1), \text{ack}=1$, 主机B收到后确认 seq 值与 $\text{ack}=1$ 则连接建立成功。

完成三次握手, 主机A与主机B开始传送数据。

在TCP/IP协议中, TCP协议提供可靠的连接服务, 采用三次握手建立一个连接。

第一次握手: 建立连接时, 客户端发送 syn 包($\text{syn}=j$)到服务器, 并进入 SYN_SEND 状态, 等待服务器确认;

第二次握手: 服务器收到 syn 包, 必须确认客户的 $\text{SYN}(\text{ack}=j+1)$, 同时自己也发送一个 SYN 包($\text{syn}=k$), 即

$\text{SYN}+\text{ACK}$ 包, 此时服务器进入 SYN_RCV 状态; 第三次握手: 客户端收到服务器的 $\text{SYN}+\text{ACK}$ 包, 向服务器发送确认包 $\text{ACK}(\text{ack}=k+1)$, 此包发送完毕, 客户端和服务器进入 ESTABLISHED 状态, 完成三次握手。完成三次握手, 客户端与服务器开始传送数据。

实例:

IP 192.168.1.116.3337 > 192.168.1.123.7788: S 3626544836:3626544836

IP 192.168.1.123.7788 > 192.168.1.116.3337: S 1739326486:1739326486 ack 3626544837

IP 192.168.1.116.3337 > 192.168.1.123.7788: ack 1739326487,ack 1

第一次握手: 192.168.1.116发送位码 $\text{syn}=1$,随机产生 $\text{seq number}=3626544836$ 的数据包到

192.168.1.123,192.168.1.123由 $\text{SYN}=1$ 知道192.168.1.116要求建立联机;

第二次握手: 192.168.1.123收到请求后要确认联机信息, 向192.168.1.116发送 $\text{ack number}=3626544837, \text{syn}=1, \text{ack}=1$,

随机产生seq=1739326486的包;

第三次握手: 192.168.1.116收到后检查ack number是否正确, 即第一次发送的seq number+1, 以及位码ack是否为1, 若正确, 192.168.1.116会再发送ack number=1739326487, ack=1, 192.168.1.123收到后确认seq=seq+1, ack=1则连接建立成功。

扩展: 四次挥手

1. 客户机打算关闭连接, 就向其TCP发送一个**连接释放报文**, 并停止再发送数据, 主动关闭TCP连接, 该报文的**FIN**标志位被置**1**, **seq=u**, 它等于前面已经传送过的数据的最后一个字节的序号加1 (FIN报文即使不携带数据, 也要消耗掉一个序号)
2. 服务器接收连接释放报文后即发出确认, 确认号是**ack=u+1**, 这个报文自己的序号是**v**, 等于它前面已传送过的数据的最后一个自己的序号加1. 此时, 从客户机到服务器这个方向的连接就释放了, TCP连接处于半关闭状态. 但服务器若发送数据, 客户机仍要接收, 即从服务器到客户机的连接仍未关闭.
3. 若服务器已经没有了要向客户机发送的数据, 就**通知TCP释放连接**, 此时其发出**FIN=1**的连接释放报文
4. 客户机收到连接释放报文后, 必须发出确认. 在确认报文中, **ACK**字段被置为**1**, 确认号**ack=w+1**, 序号**seq=u+1**. 此时, TCP连接还没有释放掉, 必须经过等待计时器设置的时间2MSL后, A才进入到连接关闭状态.