TCP/IP协议 三次握手与四次挥手

星期六, 六月 11, 2016　　6:07 下午

https请求

https协议和http协议的区别就在于https在建立tcp连接之后，有一段相互认证的过程，

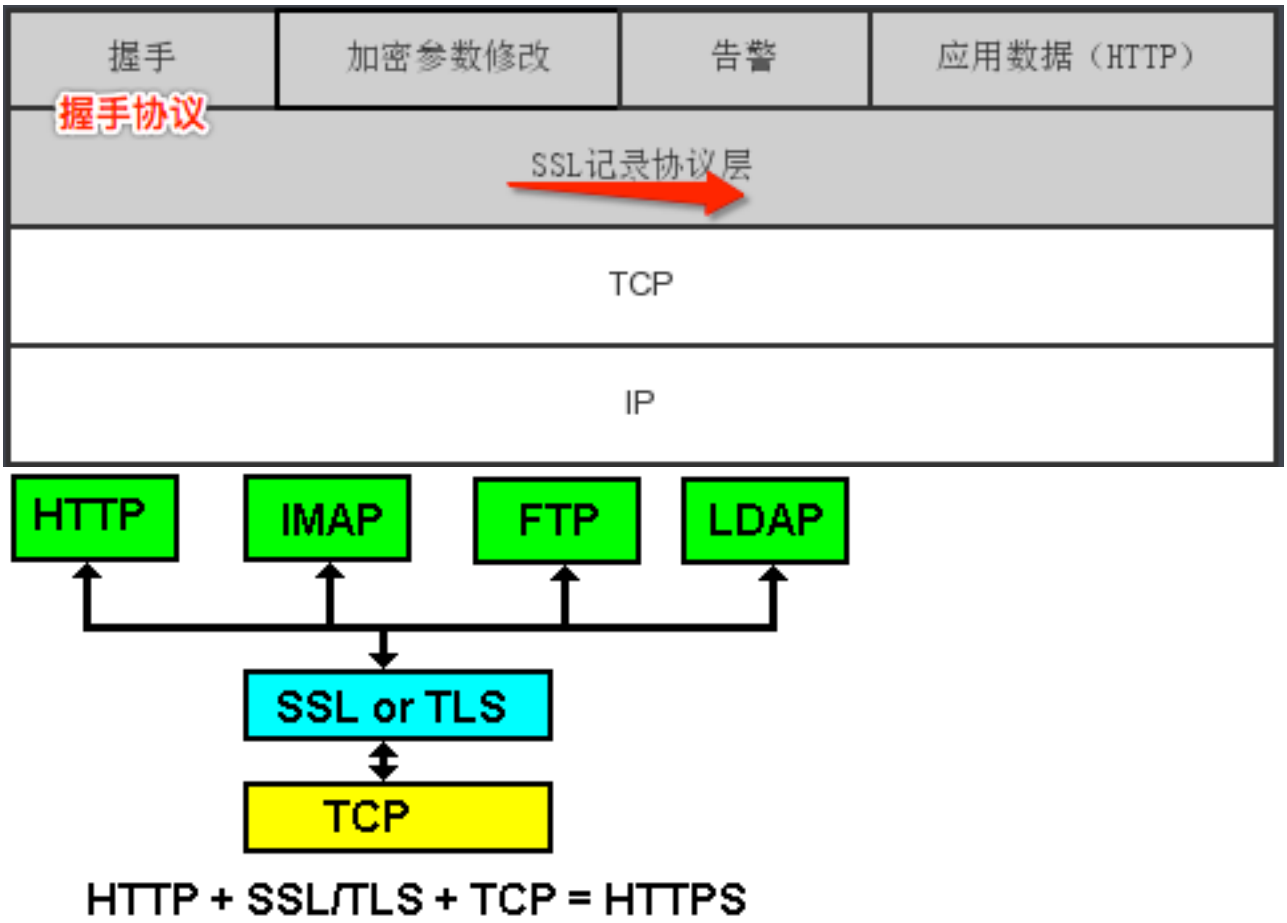
HTTPS：当客户端第一次发送请求的时候，服务器会返回一个包含公钥的受保护空间（也成为证

送请求的时候，公钥会将请求加密再发送给服务器，服务器接到请求之后，用自带的私钥进行解

返回数据。这就是 HTTPS 的安全性所在

什么是SSL/TLS？跟HTTP和HTTPS有什么关系

SSL（Secure Sockets Layer）、TLS（Transport Layer Security）



证书），当我们发

解密，如果正确再

ATS（App Transport Security）

1. 所有信息都是加密传播，第三方无法窃听。
2. 具有校验机制，一旦被篡改，通信双方会立刻发现。
3. 配备身份证书，防止身份被冒充。

一、TCP报文格式

TCP/IP协议的详细信息参看《TCP/IP协议详解》三卷本。下面是TCP报文格式图：

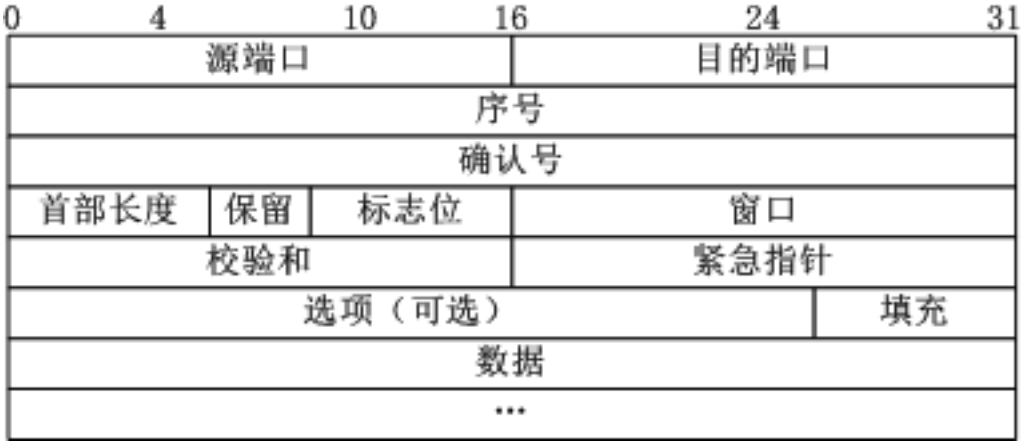


图1 TCP报文格式

上图中有几个字段需要重点介绍下：

（1）序号：Seq序号，占32位，用来标识从TCP源端向目的端发送的字节流，发起方发送

记。

（2）确认序号：Ack序号，占32位，只有ACK标志位为1时，确认序号字段才有效，Ack=S

（3）标志位：共6个，即URG、ACK、PSH、RST、SYN、FIN等，具体含义如下：

（A）URG：紧急指针（urgent pointer）有效。

（B）ACK：确认序号有效。

（C）PSH：接收方应该尽快将这个报文交给应用层。

（D）RST：重置连接。

送数据时对此进行标

Seq+1。

（E）SYN：发起一个新连接。

（F）FIN：释放一个连接。

需要注意的是：

（A）不要将确认序号Ack与标志位中的ACK搞混了。

（B）确认方Ack=发起方Req+1，两端配对。

二、三次握手

所谓三次握手（Three-Way Handshake）即建立TCP连接，就是指建立一个TCP连接时，需
总共发送3个包以确认连接的建立。在socket编程中，这一过程由客户端执行connect来触发，整
示：

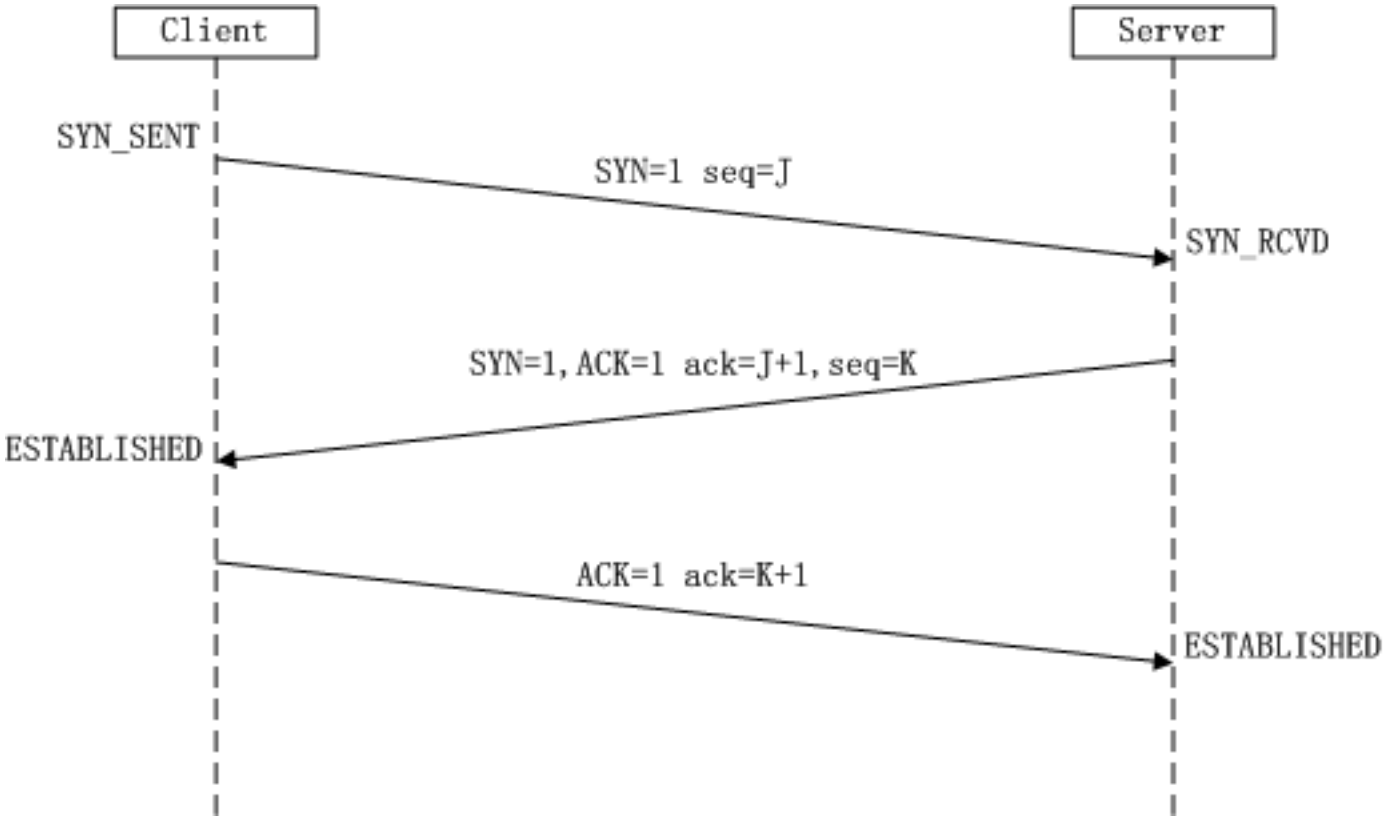


图2 TCP三次握手

（1）第一次握手：Client将标志位SYN置为1，随机产生一个值seq=J，并将该数据包发
Client进入SYN_SENT状态，等待Server确认。

（2）第二次握手：Server收到数据包后由标志位SYN=1知道Client请求建立连接，Server将
都置为1，ack=J+1，随机产生一个值seq=K，并将该数据包发送给Client以确认连接请求，Server进
态。

要客户端和服务端
个流程如下图所

发送给Server，

将标志位SYN和ACK
进入SYN_RCVD状

态。

（3）第三次握手：Client收到确认后，检查ack是否为J+1，ACK是否为1，如果正确则将标ack=K+1，并将该数据包发送给Server，Server检查ack是否为K+1，ACK是否为1，如果正确则连接和Server进入ESTABLISHED状态，完成三次握手，随后Client与Server之间可以开始传输数据了。

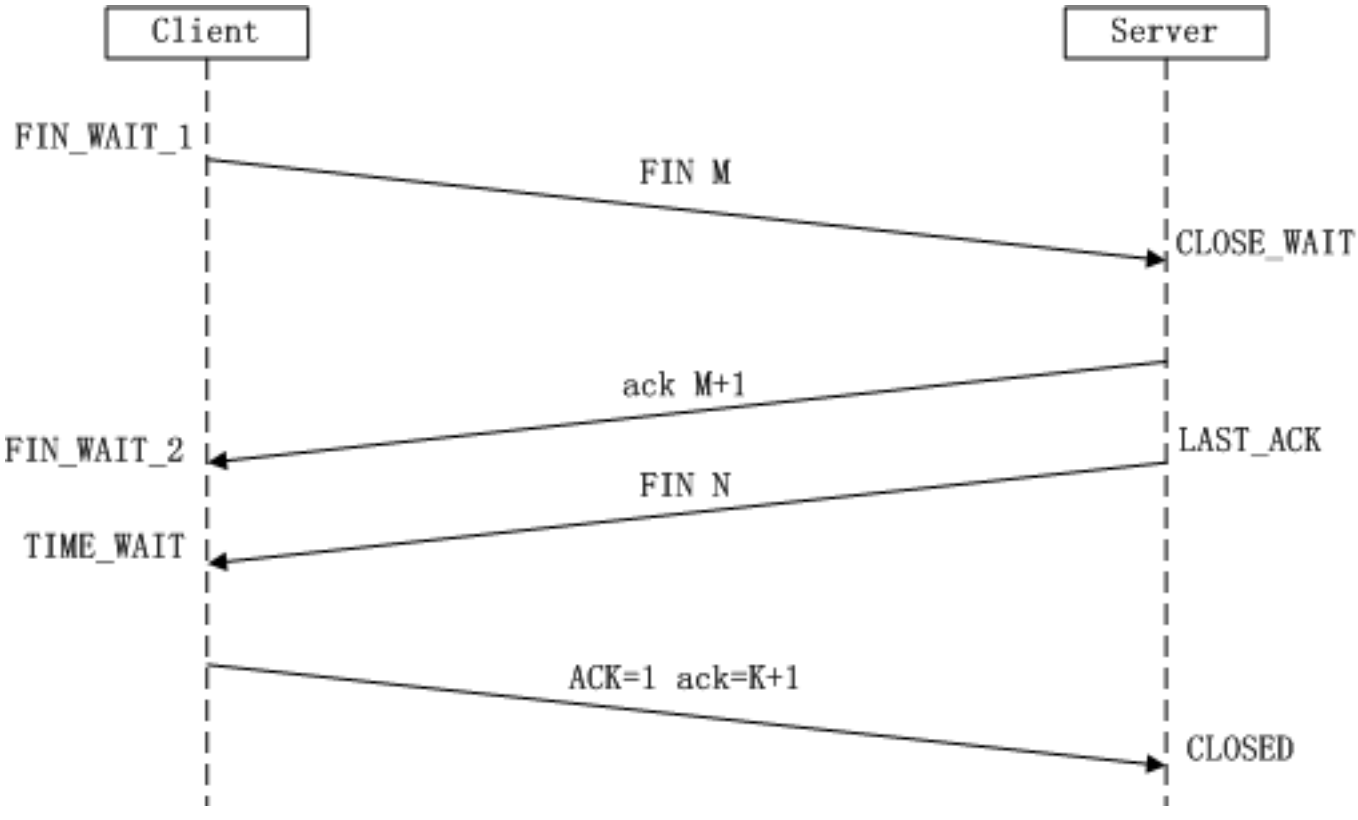
SYN攻击：

在三次握手过程中，Server发送SYN-ACK之后，收到Client的ACK之前的TCP连接称为connect），此时Server处于SYN_RCVD状态，当收到ACK后，Server转入ESTABLISHED状态。SYN攻时间内伪造大量不存在的IP地址，并向Server不断地发送SYN包，Server回复确认包，并等待Clien地址是不存在的，因此，Server需要不断重发直至超时，这些伪造的SYN包将产时间占用未连接SYN请求因为队列满而被丢弃，从而引起网络堵塞甚至系统瘫痪。SYN攻击时一种典型的DDOS攻的方式非常简单，即当Server上有大量半连接状态且源IP地址是随机的，则可以断定遭到SYN攻令可以让之现行：

```
#netstat -nap | grep SYN_RECV
```

三、四次挥手

三次握手耳熟能详，四次挥手估计就所谓四次挥手（Four-Way Wavehand）即终止T断开一个TCP连接时，需要客户端和服务端总共发送4个包以确认连接的断开。在socket编程中户端或服务端任一方执行close来触发，整个流程如下图所示：



示志位ACK置为1，
接建立成功，Client

半连接（half-open
攻击就是Client在短
t的确认，由于源
队列，导致正常的
攻击，检测SYN攻击
击了，使用如下命

TCP连接，就是指
中，这一过程由客

图3 TCP四次挥手

由于TCP连接时全双工的，因此，每个方向都必须要单独进行关闭，这一原则是当一方完后，发送一个FIN来终止这一方向的连接，收到一个FIN只是意味着这一方向上没有数据流动了据了，但是在这个TCP连接上仍然能够发送数据，直到这一方向也发送了FIN。首先进行关闭的闭，而另一方则执行被动关闭，上图描述的即是如此。

（1）第一次挥手：Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FI

（2）第二次挥手：Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与占用一个序号），Server进入CLOSE_WAIT状态。

（3）第三次挥手：Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入L

（4）第四次挥手：Client收到FIN后，Client进入TIME_WAIT状态，接着发送一个ACK给Ser到序号+1，Server进入CLOSED状态，完成四次挥手。

上面是一方主动关闭，另一方被动关闭的情况，实际中还会出现同时发起主动关闭的情况图：

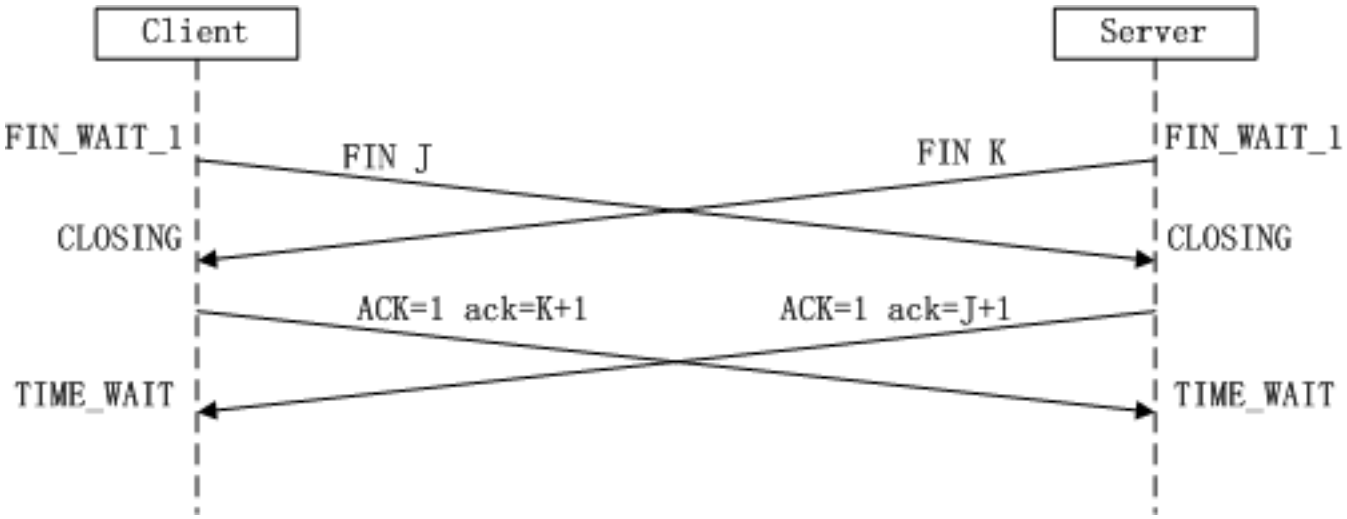


图4 同时挥手

流程和状态在上图中已经很明了了，在此不再赘述，可以参考前面的四次挥手解析步骤。

四、附注

关于三次握手与四次挥手通常都会有典型的面试题，在此提出供有需求的XDJM们参考：

完成数据发送任务
，即不会再收到数
的一方将执行主动关

N_WAIT_1状态。
SYN相同，一个FIN

LAST_ACK状态。
ver，确认序号为收

况，具体流程如下

关于三次握手与四次挥手通常都会有典型的面试题，在此提出供有需求的朋友们参考：

（1）三次握手是什么或者流程？四次握手呢？答案前面分析就是。

（2）为什么建立连接是三次握手，而关闭连接却是四次挥手呢？

这是因为服务端在LISTEN状态下，收到建立连接请求的SYN报文后，把ACK和SYN放在一个端。而关闭连接时，当收到对方的FIN报文时，仅仅表示对方不再发送数据了但是还能接收数据数据都发送给对方了，所以己方可以立即close，也可以发送一些数据给对方后，再发送FIN报文现在关闭连接，因此，己方ACK和FIN一般都会分开发送。（建立连接是相互的，而断开连接是不

已剪辑自: http://blog.csdn.net/renzhenhuai/article/details/12105457

个报文里发送给客户
，己方也未必全部
给对方来表示同意
不是相互的）