

~~~~~常思~~~~~

一点一滴，从小处做起

RS

126 TCP/IP协议 三次握手与四次挥手

2013年09月27日 17:39:03 阅读量: 26430

写评论

收藏

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:

|        |  |   |  |    |  |      |  |    |    |    |  |
|--------|--|---|--|----|--|------|--|----|----|----|--|
| 0      |  | 4 |  | 10 |  | 16   |  | 24 |    | 31 |  |
| 源端口    |  |   |  |    |  | 目的端口 |  |    |    |    |  |
| 序号     |  |   |  |    |  |      |  |    |    |    |  |
| 确认号    |  |   |  |    |  |      |  |    |    |    |  |
| 首部长度   |  |   |  | 保留 |  | 标志位  |  | 窗口 |    |    |  |
| 校验和    |  |   |  |    |  | 紧急指针 |  |    | 填充 |    |  |
| 选项(可选) |  |   |  |    |  |      |  |    |    | 填充 |  |
| 数据     |  |   |  |    |  |      |  |    |    |    |  |
| ...    |  |   |  |    |  |      |  |    |    |    |  |

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, 发起方发送数据时对此进行标记。
- (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=Seq+1。
- (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。
  - (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, 需要客户端和服务端总共发送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, 整个流程如下图所示:

```
sequenceDiagram
    participant Client
    participant Server
    Note over Client: SYN_SENT
    Client->>Server: SYN=1 seq=J
    Note over Server: SYN_RCVD
    Server->>Client: SYN=1, ACK=1 ack=J+1, seq=K
    Note over Client: ESTABLISHED
    Client->>Server: ACK=1 ack=K+1
    Note over Server: ESTABLISHED
```

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, 并将该数据包发送给Server, 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将标志位SYN和ACK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, 如果正确则将标志位ACK置为1, 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, 如果正确则连接建立成功, Client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连接称为半连接 (half-open connect), 此时Server处于SYN\_RCVD状态, 当收到ACK后, Server转入ESTABLISHED状态。SYN攻击就是Client在短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Client的确认, 由于源地址是不存在的, 因此, Server需要不断重发直至超时, 这些伪造的SYN包将占用未连接队列, 导致正常的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攻击, 检测SYN攻击的方式非常简单, 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攻击了, 使用如下命令可以让之现行:

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

#netstat -nap | grep SYN\_RECV

登录 注册

×

https://blog.csdn.net/renzhenhuai/article/details/12105457

2018/8/5

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TCP连接，就是指断开一个TCP连接时，需要客户端和服务端总共发送4个包以确认连接的开。在socket编程中，这一过程由客户端或服务端任一方执行close来触发，整个流程如下图所示：

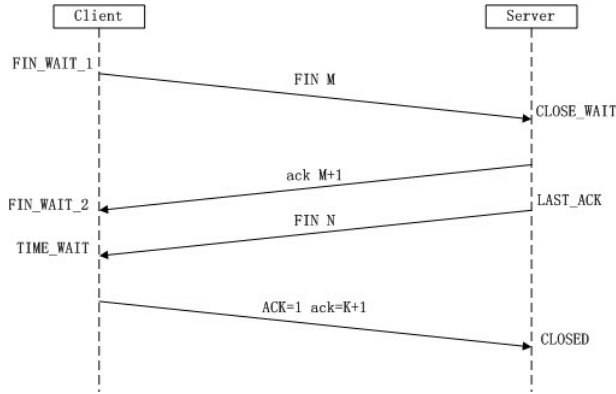


图3 TCP四次挥手

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成数据发送任务后，发送一个FIN来终止这一方向的连接，收到一个FIN只是意味着这一方上没有数据流动了，即不会再收到数据了，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。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上边描述的即是如此。

- （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FIN\_WAIT\_1状态。
- （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SYN相同，一个FIN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LAST\_ACK状态。
- （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Server，确认序号为收到序号+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，具体流程如下图：

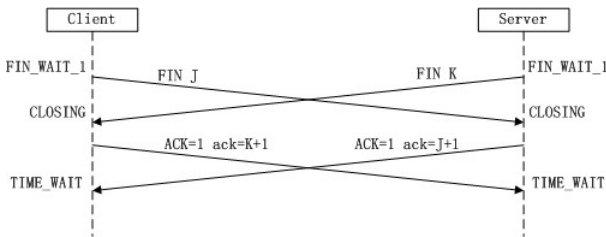


图4 同时挥手

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：

- （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放在一个报文里发送给客户端。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，己方也未必全部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报文给对方来表示同意现在关闭连接，因此，己方ACK和FIN一般都会分开发送。

文章标签：

网络

tcp

个人分类：

网络

相关热词：

tcp和

tcp与

tcp

tcp查看tcp

mqtt与tcp

上一篇

ubuntu下安装Apache+PHP+Mysql

下一篇

PHP防注入漏洞过滤函数

短信接收验证码

短信验证码接收平台有哪些

想对作者说点什么？

我来说一句

TCP为什么需要3次握手与4次挥手

为什么需要“三次握手”

在谢希仁著《计算机网络》第四版中讲“三次握手”的目的是“为了防止已失效的连接请求报文段突然又传...

5.1万