

TCP/IP网络编程(九)

笔记本： 网络编程

创建时间： 2018/11/10 20:24

更新时间： 2018/11/10 21:10

作者： xiangkang94@outlook.com

标签： 第九章(套接字的多种可选项)

1. 套接字的可选项和I/O缓冲大小

- getsockopt & setsockopt 分别用于读取和设置套接字的可选项
- **套接字的类型只能在创建时决定，以后不能在更改**
- SO_SNDBUF & SO_RCVBUF
 - 这两个分别是创建套接字时候生成的I/O缓冲区，这两个关键字可以用来读取和设置缓冲区的大小

2. SO_REUSEADDR

- 发生地址分配错误(Binding Error)
 - time-wait：先断开连接FIN的主机有个**时间间隙**，**在该间隙中端口号被占用，bind时会出错**。在time-wait状态可以继续接收主机B的终止信息，假如没有这个机制，A发送ACK时终止，但是ACK丢失了，则B永远无法接收A的ACK。
 - 系统SOL_SOCKET中的SO_REUSEADDR，默认是FALSE，即time-wait内端口不能重复使用。将值改成TRUE，即可重新使用。

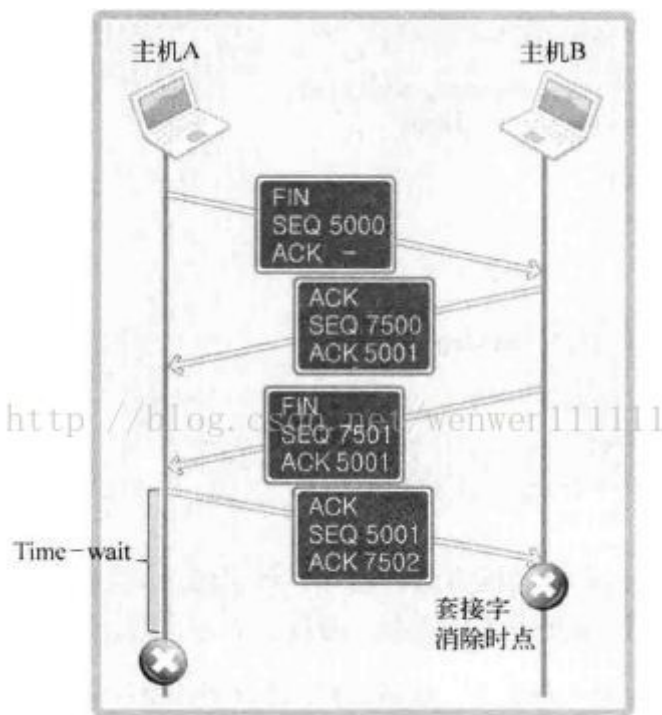


图9-1 Time-wait状态下的套接字

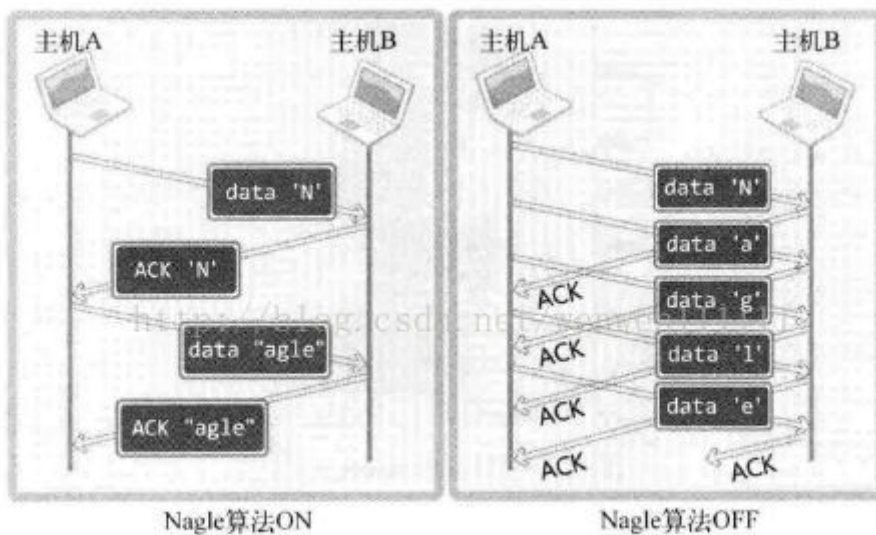


图9-3 Nagle算法

```
optlen = sizeof(option);
option = TRUE;
setsockopt(serv_sock, SOL_SOCKET, SO_RESUREADDR, (void *)&option, optlen);
```

1. Nagle算法：1984年诞生，应用于TCP层，只有当收到上一个数据包的ACK后，Nagle算法才会发送下一数据包。

- TCP套接字默认使用Nagle算法，最大限度缓冲数据，直到收到ACK。
- 对应的，不使用Nagle算法将对网络流量Traffic产生负面影响，影响效率。
- 并不是任何情况都适用Nagle算法，在网络流量未受到太大影响时，不使用Nagle算法传输速度会更快，比如传输大文件数据时。

通常大文件的数据传输不应该使用Nagle算法，因为将大文件传入输出缓冲区并不会花太多的时间，因此，即使不使用Nagle算法，也会在装满输出缓冲区时 候传输数据包。

```
int opt_val = 1; //禁用Nagle算法
setsockopt(sock, IPPROTO_TCP, TCP_NODELAY, (void *)&opt_val, sizeof(opt_val));
```

