

### 知识点1【TCP三次握手四次挥手】

### 知识点2【Linux查看TCP的各类状态值】

### 知识点3【TCP的状态宏翻译】

ESTABLISHED 套接字已建立连接

SYN\_SENT 套接字正在积极地尝试建立连接

SYN\_RECV 从网络接收到一个连接请求

FIN\_WAIT1 套接字已关闭，连接正在关闭

FIN\_WAIT2 连接已关闭，套接字正在等待远程端关闭

TIME\_WAIT 套接字关闭后等待处理仍在网络中的数据包

CLOSE 套接字没有被使用

CLOSE\_WAIT 远程端已经关闭，等待套接字关闭

LAST\_ACK 远程端关闭，套接字关闭。等待确认

LISTEN 套接字正在侦听传入的连接

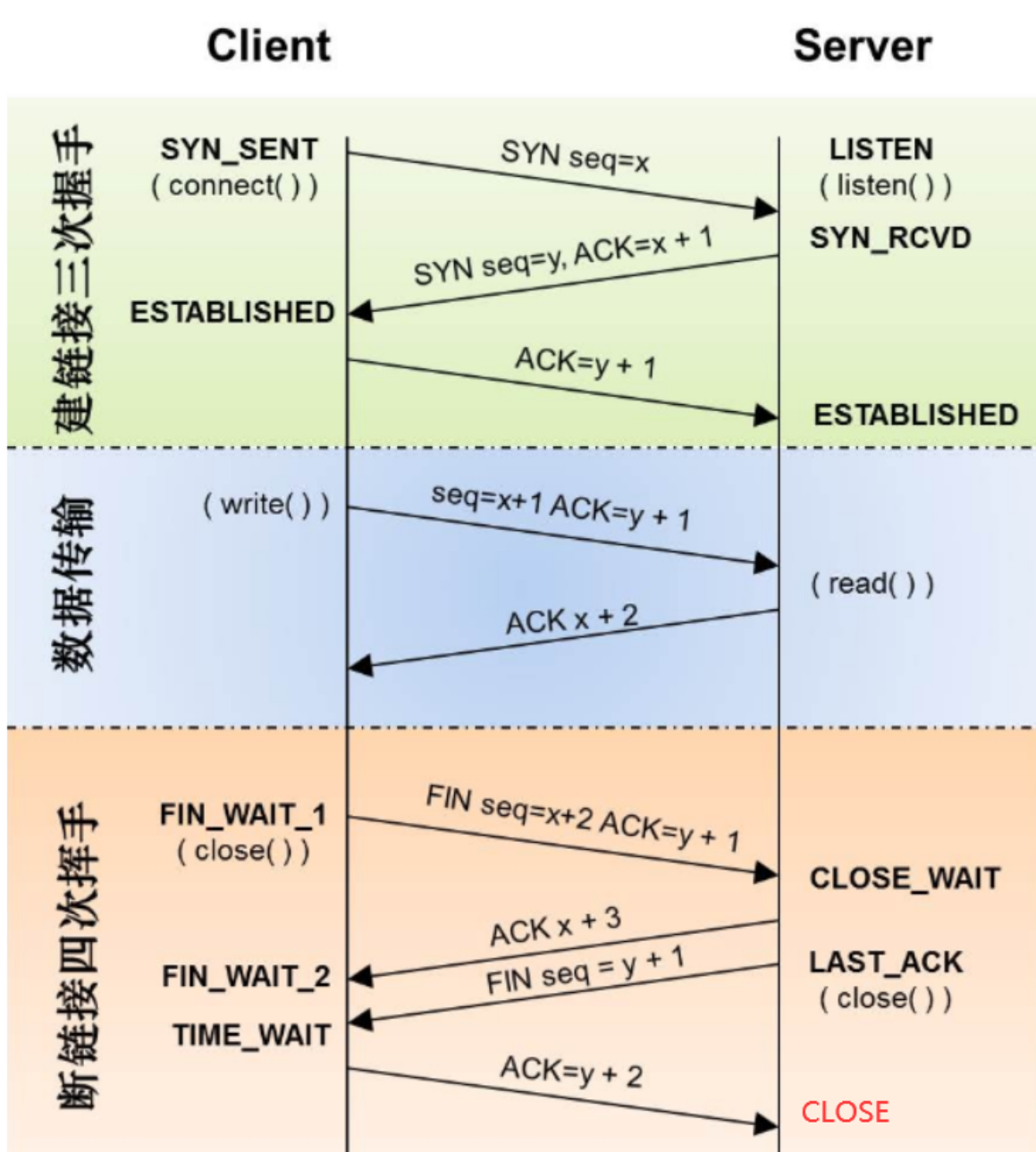
CLOSING 两个套接字都关闭了，但我们仍然没有发送所有的数据

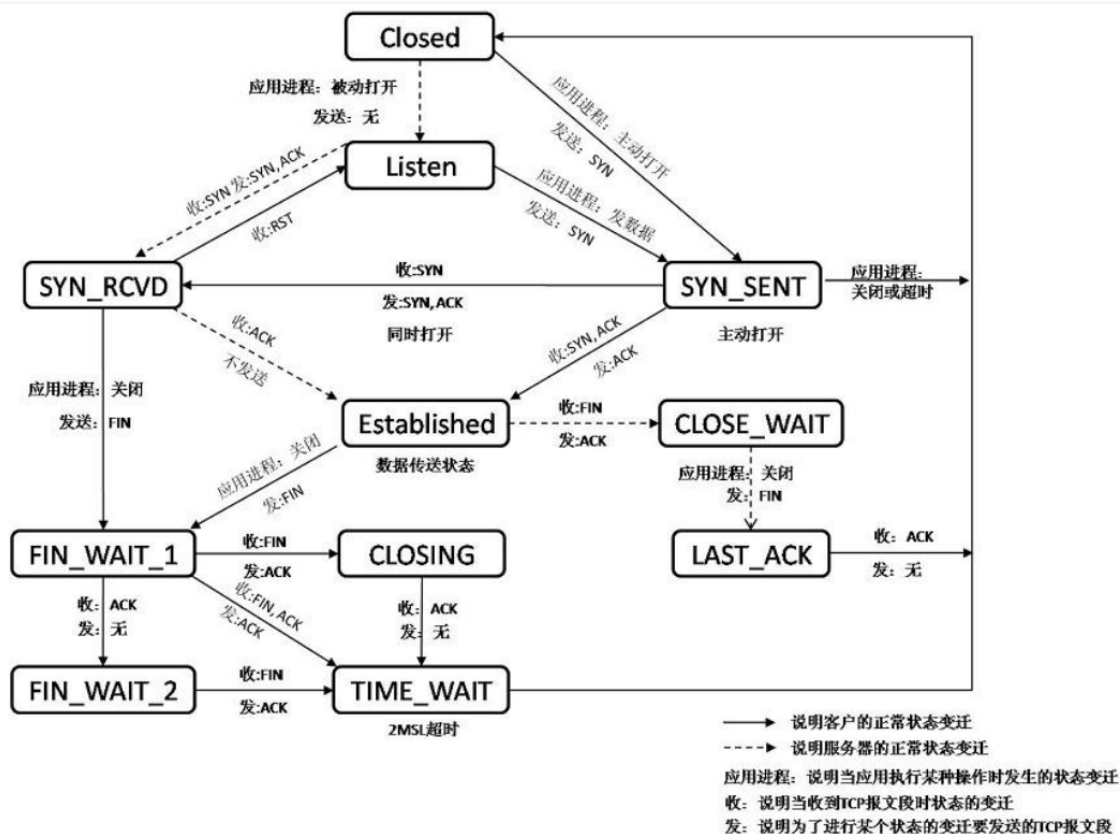
### 知识点4【注意】

1、为什么要等待呢？

2、为什么TIME\_WAIT 状态还需要等 $2 * \text{MSL}$ 秒之后才能返回到CLOSED 状态呢？

## 知识点1【TCP三次握手四次挥手】





## 知识点2 【Linux查看TCP的各类状态值】

```
1 man netstat
```

### State

The state of the socket. Since there are no states in raw mode and usually may be left blank. Normally this can be one of several values:

#### ESTABLISHED

The socket has an established connection.

#### SYN\_SENT

The socket is actively attempting to establish a connection.

#### SYN\_RECV

A connection request has been received from the network.

#### FIN\_WAIT1

The socket is closed, and the connection is shutting down.

## 知识点3 【TCP的状态宏翻译】

**ESTABLISHED** 套接字已建立连接

**SYN\_SENT** 套接字正在积极地尝试建立连接

**SYN\_RECV** 从网络接收到一个连接请求

**FIN\_WAIT1** 套接字已关闭，连接正在关闭

**FIN\_WAIT2** 连接已关闭，套接字正在等待远程端关闭

**TIME\_WAIT** 套接字关闭后等待处理仍在网络中的数据包

**CLOSE** 套接字没有被使用

**CLOSE\_WAIT** 远程端已经关闭，等待套接字关闭

**LAST\_ACK** 远程端关闭，套接字关闭。等待确认

**LISTEN** 套接字正在侦听传入的连接

**CLOSING** 两个套接字都关闭了，但我们仍然没有发送所有的数据

## 知识点4【注意】

### 1、为什么要等待呢？

为了防止这种情况：A接到B的释放连接请求后会发送一个确认信息，但是如果这个确认信息丢了，也就是B没有收到确认释放连接，那么B就会重发一个释放连接请求，这时候A还处于TIME\_WAIT状态，所以会再次发送一个确认信息。

### 2、为什么TIME\_WAIT 状态还需要等 $2*MSL$ 秒之后才能返回到CLOSED状态呢？

IA2因为虽然双方都同意关闭连接了，而且握手的4个报文也都发送完毕，按理可以直接回到CLOSED 状态（就好比从SYN\_SENT 状态到ESTABLISH 状态那样），但是我们必须假想网络是不可靠的，你无法保证你最后发送的ACK报文一定会被对方收到，就是说对方处于LAST\_ACK 状态下的SOCKET可能会因为超时未收到ACK报文，而重发FIN报文，所以这个TIME\_WAIT 状态的作用就是用来重发可能丢失的ACK报文