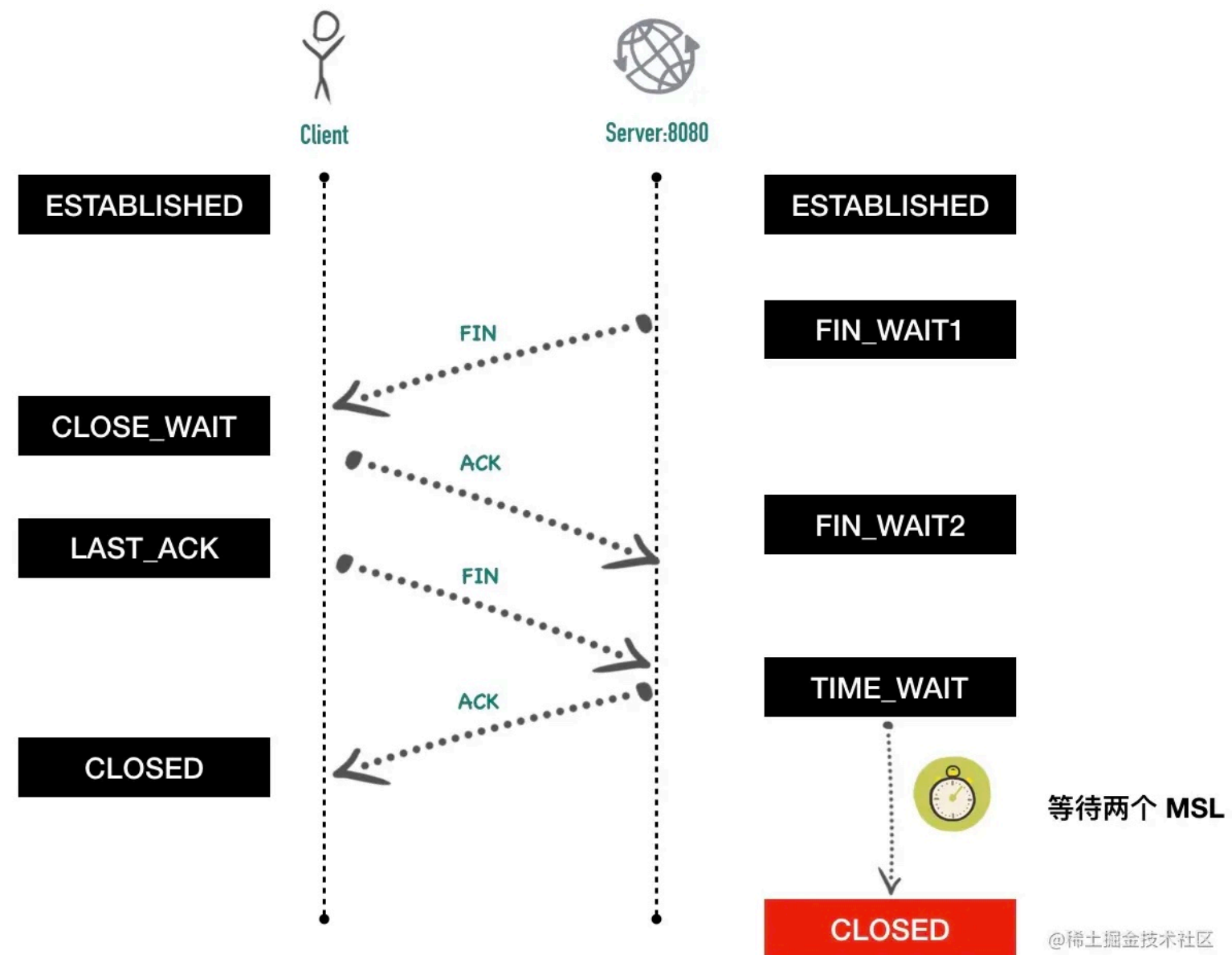




怎么老是出现“地址已经被使用” | 聊聊 Socket 选项之 SO_REUSEADDR

再看四次挥手

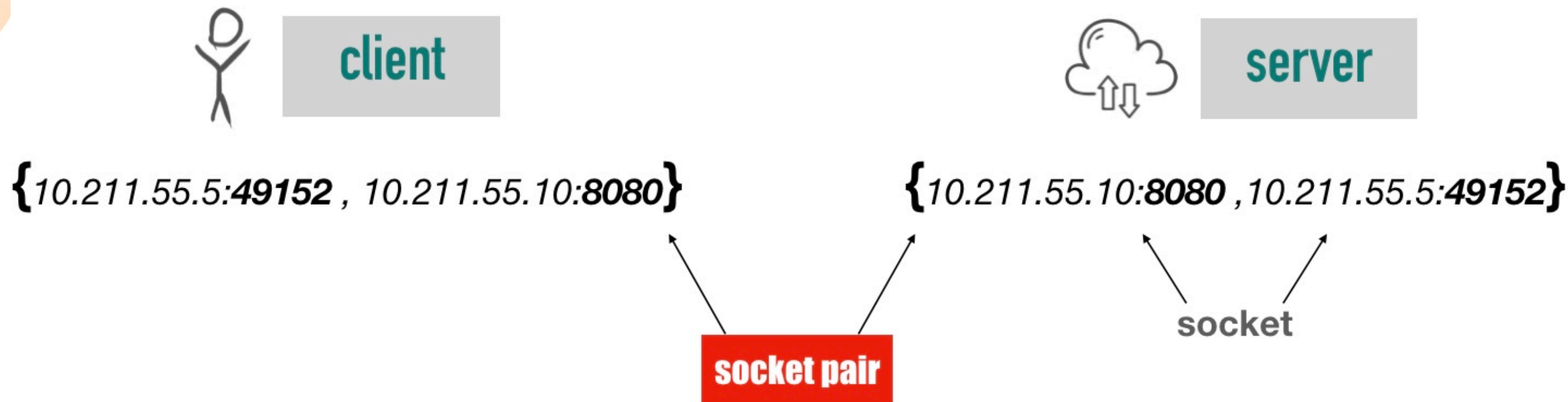


实验环节

```
public class ReuseAddress {  
    public static void main(String[] args) throws IOException {  
        ServerSocket serverSocket = new ServerSocket();  
        // setReuseAddress 必须在 bind 函数调用之前执行  
        serverSocket.setReuseAddress(false);  
        serverSocket.bind(new InetSocketAddress(8080));  
        System.out.println("reuse address: " + serverSocket.getReuseAddress());  
        while (true) {  
            Socket socket = serverSocket.accept();  
            System.out.println("incoming socket..");  
            OutputStream out = socket.getOutputStream();  
            out.write("Hello\n".getBytes());  
            out.close();  
        }  
    }  
}
```

为什么需要 SO_REUSEADDR 参数

服务端主动断开连接以后，需要等 2 个 MSL 以后才最终释放这个连接，重启以后要绑定同一个端口，默认情况下，操作系统的实现都会阻止新的监听套接字绑定到这个端口上。



TCP 要求这样的四元组必须是唯一的，但大多数操作系统的实现要求更加严格，只要还有连接在使用这个本地端口，则本地端口不能被重用（bind 调用失败）

SO_REUSEADDR 默认值

启用 SO_REUSEADDR 套接字选项可以解除这个限制，默认情况下这个值都为 0，表示关闭。

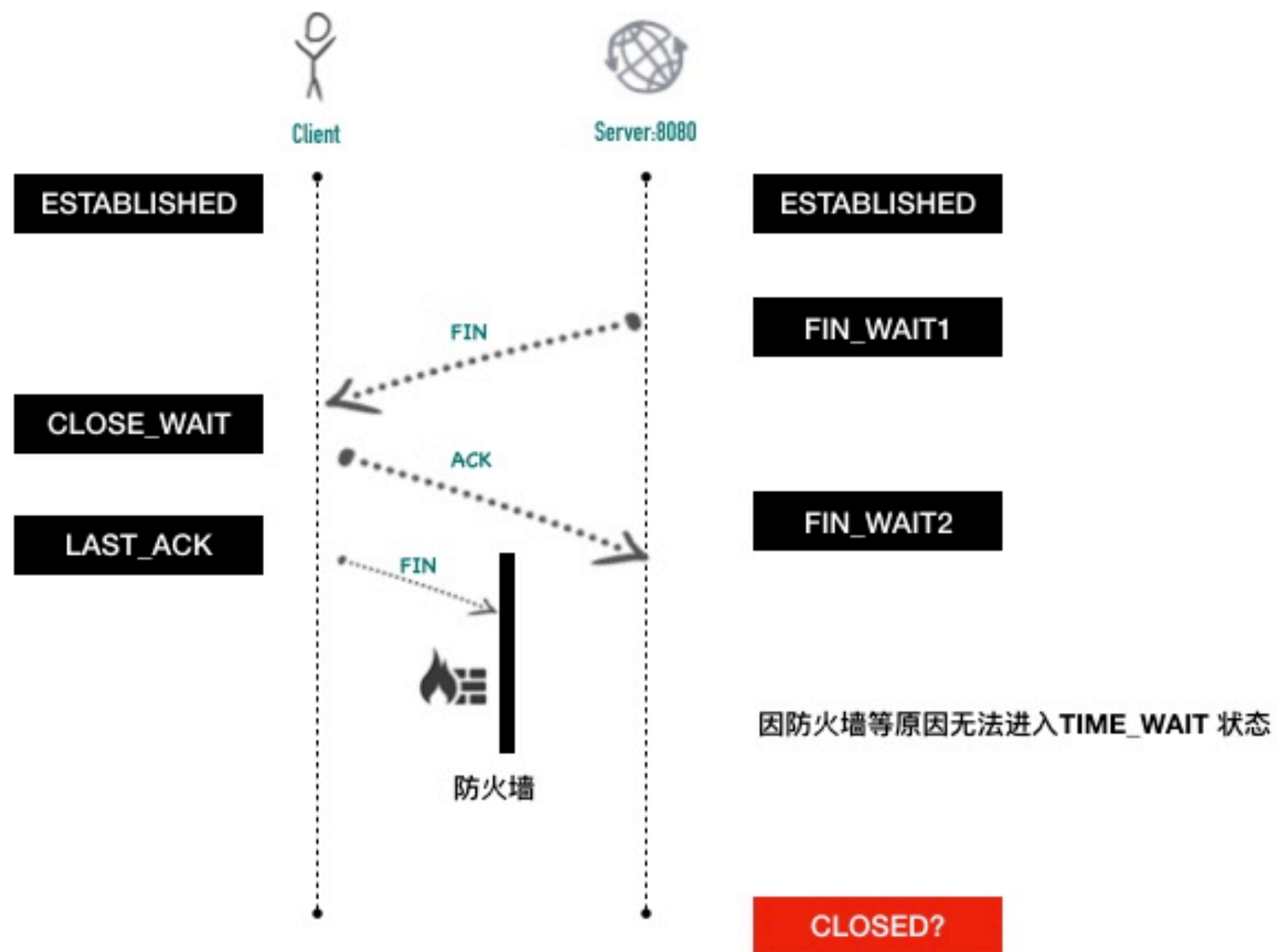
在 Java 中，reuseAddress 不同的 JVM 有不同的实现，在我本机上，这个值默认为 1 允许端口重用。但是为了保险起见，写 TCP、HTTP 服务一定要主动设置这个参数为 1。

是不是只有处于 TIME_WAIT 才允许端口复用?

查看 Java 中 `ServerSocket.setReuseAddress` 的文档，有如下的说明

```
/**
 * Enable/disable the {@link SocketOptions#SO_REUSEADDR SO_REUSEADDR}
 * socket option.
 * <p>
 * When a TCP connection is closed the connection may remain
 * in a timeout state for a period of time after the connection
 * is closed (typically known as the {@code TIME_WAIT} state
 * or {@code 2MSL} wait state).
 * For applications using a well known socket address or port
 * it may not be possible to bind a socket to the required
 * {@code SocketAddress} if there is a connection in the
 * timeout state involving the socket address or port.
 * /
```

假设因为网络的原因，客户端没有回发 FIN 包，导致服务器端处于 FINWAIT2 状态，而非 TIMEWAIT 状态，那设置 SO_REUSEADDR 还会生效吗？



实验步骤 (1)

现在有两台机器c1（充当客户端），c2（充当服务器）。在客户端c1 利用防火墙拦截掉所有发出的 FIN 包：

```
sudo iptables --append OUTPUT --match tcp --protocol tcp --dport 8080 --tcp-flags FIN FIN --jump DROP
```


实验步骤 (2)

在c1 上使用 `nc c2 8080` 发起 tcp 连接，随后杀掉 c2 的进程，因为服务端收不到客户端发过来的 FIN 包，也即四次挥手中的第 3 步没能成功，服务端此时将处于 `FIN_WAIT2` 状态。

```
ya@c2 ~$ sudo netstat -lnpa | grep 8080
```

```
tcp6          0      0 10.211.55.10:8080      10.211.55.5:39664      FIN_WAIT2      -
```

实验步骤 (3)

将 SO_REUSEADDR 设置为 1，重复上面的测试过程，将发现不会出现异常。将 SO_REUSEADDR 设置为 0，则会出现 Address already in use 异常。

为什么通常不会在客户端上出现

通常情况下都是客户端主动关闭连接，那客户端那边为什么不会有问题呢？

因为客户端都是用的临时端口，这些临时端口与处于 TIME_WAIT 状态的端口恰好相同的可能性不大，就算相同换一个临时的临时端口就好了。

小结

SO_REUSEADDR 是一个 Socket 选项，主要就是用来告诉操作系统内核，如果端口已被占用，但是 TCP 连接状态位于 TIME_WAIT（不仅仅是），可以重用端口。

记住：所有 TCP 服务端程序，一定要在 bind 之前设置 SO_REUSEADDR 选项，不然可能会出现无法在短时间内重启服务端程序。

作业

对于 UDP 协议来说，SO_REUSEADDR 选项有没有作用？

