

tcp_tw_reuse选项的含义如下 (<http://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt>) :

tcp_tw_reuse - BOOLEAN

Allow to reuse TIME-WAIT sockets for new connections when it is safe from protocol viewpoint. Default value is 0.

这里的关键在于“**协议什么情况下认为是安全的**”，由于环境限制，没有办法进行验证，通过看源码简单分析了一下。

=====linux-2.6.37 net/ipv4/tcp_ipv4.c 114=====

```
int tcp_twsk_unique(struct sock *sk, struct sock *sktw, void *twp)
{
    const struct tcp_timewait_sock *tcptw = tcp_twsk(sktw);
    struct tcp_sock *tp = tcp_sk(sk);
```

```
/* With PAWS, it is safe from the viewpoint
   of data integrity. Even without PAWS it is safe provided sequence
   spaces do not overlap i.e. at data rates <= 80Mbit/sec.
```

Actually, the idea is close to VJ's one, only timestamp cache is held not per host, but per port pair and TW bucket is used as state holder.

If TW bucket has been already destroyed we fall back to VJ's scheme and use initial timestamp retrieved from peer table.

```
*/
```

//从代码来看，tcp_tw_reuse选项和tcp_timestamps选项也必须同时打开；否则tcp_tw_reuse就不起作用

//另外，所谓的“协议安全”，从代码来看应该是收到最后一个包后超过1s

```
if (tcptw->tw_ts_recent_stamp &&
    (twp == NULL || (sysctl_tcp_tw_reuse &&
        get_seconds() - tcptw->tw_ts_recent_stamp > 1))) {
    tp->write_seq = tcptw->tw_snd_nxt + 65535 + 2;
    if (tp->write_seq == 0)
        tp->write_seq = 1;
    tp->rx_opt.ts_recent = tcptw->tw_ts_recent;
    tp->rx_opt.ts_recent_stamp = tcptw->tw_ts_recent_stamp;
    sock_hold(sktw);
    return 1;
}
```

```
return 0;  
}
```

总结一下：

- 1) tcp_tw_reuse选项和tcp_timestamps选项也必须同时打开；
- 2) 重用TIME_WAIT的条件是收到最后一个包后超过1s。

官方手册有一段警告：

It should not be changed without advice/request of technical experts.

对于大部分局域网或者公司内网应用来说，满足条件2) 都是没有问题的，因此官方手册里面的警告其实也没那么可怕：)