# 第8章　T/TCP的实现：TCP概要

## 8.1　概述

本章内容覆盖了T/TCP对TCP数据结构和函数所做的全局性修改。增加了两个全局变量：tcp_ccgen，即全局CC计数器，以及tcp_do_rfcl644，这是一个标志变量，说明是否选用CC选项。TCP的协议交换记录项也作了修改，以支持隐式的打开和关闭。另外还在TCP控制块中增加了4个变量。

对tcp_slowtimo函数也作了简单修改，以便能够测量每个连接的持续时间。给定一个连接的持续时间，如果持续时间短于MSL，则如4.4节所述，T/TCP将截断TIME_WAIT状态的保持时间。

## 8.2　代码介绍

T/TCP没有增加新的源文件，但是需要一些新的变量。

### 全局变量

图8-1中给出了T/TCP新增加的全局变量，在各个TCP函数中都会用到。

| 变　量 | 数据类型 | 说　明 |
|---|---|---|
| tcp_ccgen | tcp_cc | 要发送的下一个CC值 |
| tcp_do_rfc1644 | int | 如果为真(默认)，发送CC或CCnew选项 |

图8-1　T/TCP新增的全局变量

在第3章我们给出了一些有关tcp_ccgen变量的例子。在6.5节中也提到了tcp_cc的数据类型是用typedef定义的，是无符号长整数。tcp_cc变量值为0，表示它尚未定义。tcp_ccgen变量总是这样存取的：

```
tp->cc_send = CC_INC(tcp_ccgen);
```

其中cc_send是TCP控制块的新字段(见后面的图8-3)。宏CC_INC是在<netinet/tcp_seq.h>中定义的：

```
#define CC_INC(c)    (++(c) == 0 ? ++(c) : (c))
```

由于这个值是在使用之前增加的，因此，tcp_ccgen要初始化为0，且它的第一个有用值为1。

为了按照模运算比较CC的值，定义了四个宏：CC_LT、CC_LEQ、CC_GT和CC_GEQ。这四个宏与卷2第649页定义的四个SEQ_*xx*宏完全一样。

变量tcp_do_rfcl644与卷2中介绍的变量tcp_do_rfcl323相似。如果tcp_do_rfcl644为0，TCP不会向对方发送CC或CCnew选项。

### 统计量

T/TCP新增了5个计数器，如图8-2所示。它们加在tcpstat结构中，卷2第638页对这个结构有介绍。

| tcpstat字段 | 说　明 |
|---|---|
| tcps_taook | TAO正确时接收到SYN |
| tcps_taofail | 接收到带有CC选项的SYN，但TAO测试失败 |
| tcps_badccecho | CCecho选项错误的SYN/ACK报文段 |
| tcps_impliedack | 隐含着对前一次连接的ACK的新SYN |
| tcps_ccdrop | 因为无效的CC选项而丢弃的报文段 |

图8-2　在tcpstat 结构中新增的T/TCP统计量

程序netstat必须经修改才能打印这些新字段的值。

## 8.3 TCP的protosw结构

我们在第5章提到过，TCP的protosw记录项inetsw[2](卷2第641页)的pr_flags字段在T/TCP中作了修改。新的插口层标志 PR_IMPLOPCL必须包括在内，已有的标志PR_CONNREQUIRED和PR_WANTRCVD也必须包含在内。在sosend中，如果调用进程给出了一个目标地址，这个新的标志允许对一个未建连接的插口调用 sendto，并且如果指定了MSG_EOF标志，它所起的作用是发出一个PRU_SEND_EOF请求而不是PRU_SEND请求。

对protosw记录项所作的修改中有一个不是T/TCP所需的，即定义了tcp_sysctl函数作为pr_sysctl字段。这就允许系统管理员用前缀为net.inet.tcp的sysctl程序来修改能够控制TCP操作的一些变量值(卷2介绍的Net/3代码仅仅支持sysctl程序通过ip_sysctl、icmp_sysctl和udp_sysctl函数对IP、ICMP和UDP的一些变量进行控制)。在图12-6中给出了tcp_sysctl函数。

## 8.4 TCP控制块

TCP控制块中新增了四个变量，卷2第643~644页说明了TCP控制块的tcpcb结构。我们在图8-3中仅给出了新的字段，并非整个结构。

| 变　量 | 数据类型 | 说　明 |
|---|---|---|
| t_duration | u_long | 以500ms为单位的连接持续时间 |
| t_maxopd | u_short | MSS加上通常选项的长度 |
| cc_send | tcp_cc | 发送给对等端的CC值 |
| cc_recv | tcp_cc | 从对等端中接收到的CC值 |

图8-3　T/TCP在tcpcb 结构中新增的字段

t_duration用于确定T/TCP是否可以截断TIME_WAIT状态的保持时间，见4.4节的讨论。当控制块创建时它的值为0，由tcp_slowtimo(8.6节)每过500 ms加1。

t_maxopd是为了代码的方便而设的。它的取值是已有t_maxseg字段的值加上TCP选项通常所占用的字节数。t_maxseg是每个报文段中的数据字节数。例如，在MTU 为1500字节的一个以太网上，如果时间戳和T/TCP都用上了，t_maxopd将为1460，t_maxseg则为1440。

它们之间的差值 20字节是由 12字节的时间戳选项加上 8字节的 CC选项(图2-4)造成的。t_maxopd和t_maxseg都是在tcp_mssrcvd函数中计算并记录的。

最后两个变量来自RFC 1644，在第2章给出了有关这三个变量的例子。如果一个连接的两端主机都用了CC选项，cc_recv的值将为非0。

在TCP控制块的t_flags字段中新定义了6个标志，如图8-4所示，是对卷2的图24-14中的9个标志的补充。

| t_flags | 说 明 |
|---|---|
| *TF_SENDSYN* | 发送SYN(隐藏的半同步连接状态标志) |
| *TF_SENDFIN* | 发送FIN(隐藏的状态标志) |
| *TF_SENDCCNEW* | 主动打开时发送CCnew选项而不是CC选项 |
| *TF_NOPUSH* | 不发送报文段，只清空发送缓存 |
| *TF_RCVD_CC* | 当对端在SYN中发送了CC选项时设置该标志 |
| *TF_REQ_CC* | 已经/将在SYN中申请CC选项 |

图8-4 T/TCP新增的t_flags 及其取值

不要把T/TCP中的两个标志TF_SENDFIN与TF_SENTFIN混淆，前者表示TCP需要发送FIN，而后者表示已经发出FIN。

> TF_SENDSYN和TF_SENDFIN这两个名字源于Bob Braden的“T/TCP的实现”。FreeBSD实现中将这两个名字改为TF_NEEDSYN和TF_NEEDFIN。我们选用了前面的名字，因为已经用新的标志来表示是否需要发送控制标志，如果选用后面的名字就会误解为需要接收SYN或FIN。然而请注意，因为选用了这样的名字，T/TCP的TF_SENDFIN标志和已有的TF_SENTFIN标志(表明TCP已经发出了FIN)仅有一个字符之差。

我们将在下一章的图9-3和图9-7中分别介绍TF_NOPUSH和TF_SENDCCNEW标志。

## 8.5 tcp_init函数

所有的T/TCP变量都不需要显式的初始化，因此卷 2中介绍的tcp_init函数没有变化。全局变量tcp_ccgen是没有初始化的外部变量，按照C语言的规则，它的默认值为0。这样做不会出错，因为在8.2节中定义的宏CC_INC是先对该变量加1，然后再用，因此在重启动后，tcp_ccgen的第一个有用值是1。

T/TCP也要求在重启动时将TAO缓存全部清空，由于在重启动时要初始化IP路由表，所以TAO缓存不需要专门处理。在路由表中每增加一个新的rtentry结构，rtrequest要将该结构初始化为0(卷2第489页)。这就意味着rmxp_tao结构中3个TAO变量的默认值都为0(图6-3)。为新主机创建新的TAO记录项时，T/TCP需要将tao_cc的值初始化为0。

## 8.6 tcp_slowtimo函数

两个TCP定时函数中有一个增加了一行：每次处理 500 ms定时器时，要对每个TCP控制块的t_duration字段执行加1操作，卷2第666页给出了tcp_slowtimo函数。下面这一行

```
tp->t_duration++;
```

加在这个图的第94~95行之间。这个变量的用途是测量每个连接的长度，以500ms为单位。如果连接持续时间短于MSL，TIME_WAIT状态的保持时间就可以截断，在4.4节中已经讨论过。

与这项优化有关的工作是在<netinet/timer.h>头文件中定义了下面这个常量：

```
#define TCPTV_TWTRUNC   8    /* RTO factor to truncate TIME_WAIT */
```

我们在图11-17和图11-19中将可以看到，如果T/TCP连接是主动关闭，并且t_duration的值小于TCPTV_MSL(60个500 ms，即30秒)，那么TIME_WAIT状态的保持时间就是当前重传超时(RTO)乘以TCPTV_TWTRUNC。在局域网环境中，RTO通常为3个500ms，即1.5秒，这将使TIME_WAIT状态的保持时间缩短到12秒。

## 8.7 小结

T/TCP新增了两个全局变量(tcp_ccgen和tcp_do_rfcl 644)、4个TCP控制块字段和5个TCP统计结构计数器。

tcp_slowtimo函数也作了修改，以500ms为时间单位计量每个TCP连接的持续时间。这个持续时间决定了T/TCP能否在主动关闭时截断TIME_WAIT状态的保持时间。