

图书基本信息

书名：<<TCP/IP详解 卷3：TCP事务协议、HTTP、NNTP和UNIX域协议>>

13位ISBN编号：9787111075684

10位ISBN编号：7111075684

出版时间：2000-9

出版时间：机械工业出版社

作者：史蒂文斯

页数：252

译者：胡谷雨

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

本书是“TCP/IP详解系列”的延续。

主要内容包括：TCP事务协议，即T/TCP，这是对TCP的扩展，使客户-服务器事务更快、更高效和更可靠；TCP/IP应用，主要是HTTP和NNTP；UNIX域协议，这些协议提供了进程之间通信的一种手段

。

当客户与服务器进程在同一台主机上时，UNIX域协议通常要比TCP/IP快一倍。

本书同样采用了大量的实例和实现细节，并参考引用了卷2中的大量源程序。

适用于希望理解TCP/IP如何工作的人，包括编写网络应用程序的程序员以及利用TCP/IP维护计算机网络的系统管理员。

作者简介

作者：(美国)史蒂文斯(Seves,W.R.) 译者：胡谷雨 等

书籍目录

译者序

前言

第一部分 TCP事务协议

第1章 T/TCP概述

1.1 概述

1.2 UDP上的客户-服务器

1.3 TCP上的客户-服务器

1.4 T/TCP上的客户-服务器

1.5 测试网络

1.6 时间测量程序

1.7 应用

1.8 历史

1.9 实现

1.10 小结

第2章 T/TCP协议

2.1 概述

2.2 T/TCP中的新TCP选项

2.3 T/TCP实现所需变量

2.4 状态变迁图

2.5 T/TCP的扩展状态

2.6 小结

第3章 T/TCP使用举例

3.1 概述

3.2 客户重新启动

3.3 常规的T/TCP事务

3.4 服务器收到过时的重复SYN

3.5 服务器重启动

3.6 请求或应答超出报文段最大长度MSS

3.7 向后兼容性

3.8 小结

第4章 T/TCP协议 (续)

4.1 概述

4.2 客户的端口号和TIME_WAIT状态

4.3 设置TIME_WAIT状态的目的

4.4 TIME_WAIT状态的截断

4.5 利用TAO跳过三次握手

4.6 小结

第5章 T/TCP协议的实现：插口层

5.1 概述

5.2 常量

5.3 sosend函数

5.4 小结

第6章 T/TCP的实现：路由表

6.1 概述

6.2 代码介绍

6.3 radix_node_head结构

6.4 rtenry结构

6.5 rt_metrics结构

6.6 in_inithead函数

6.7 in_addroute函数

6.8 in_matroute函数

6.9 in_clsroute函数

6.10 in_rtqtime函数

6.11 in_rtqkill函数

6.12 小结

第7章 T/TCP实现 : 协议控制块

7.1 概述

7.2 in_pcbaddr函数

7.3 in_pcbconnect函数

7.4 小结

第8章 T/TCP实现 : TCP概要

8.1 概述

8.2 代码介绍

8.3 TCP的protosw结构

8.4 TCP控制块

8.5 tcp_init函数

8.6 tcp_slowtime函数

8.7 小结

第9章 T/TCP实现 : TCP输出

9.1 概述

9.2 tcp_output函数

9.2.1 新的自动变量

9.2.2 增加隐藏的状态标志

9.2.3 在SYN_SENT状态不要重传SYN

9.2.4 发送器的糊涂窗口避免机制

9.2.5 有RST或SYN标志时强制发送报文段

9.2.6 发送MSS选项

9.2.7 是否发送时间戳选项

9.2.8 发送T/TCP的CC选项

9.2.9 根据TCP选项调整数据长度

9.3 小结

第10章 T/TCP实现 : TCP函数

10.1 概述

10.2 tcp_newtcpcb函数

10.3 tcp_rtlookup函数

10.4 tcp_gettaocache函数

10.5 重传超时间隔的计算

10.6 tcp_close函数

10.7 tcp_msssend函数

10.8 tcp_mssrcvd函数

10.9 tcp_dooptions函数

10.10 tcp_reass函数

10.11 小结

第11章 T/TCP实现 : TCP输入

11.1 概述

11.2 预处理

11.3 首部预测

11.4 被动打开的启动

11.5 主动打开的启动

11.6 PAWS : 防止序号重复

11.7 ACK处理

11.8 完成被动打开和同时打开

11.9 ACK处理 (续)

11.10 FIN处理

11.11 小结

第12章 T/TCP实现 : TCP用户请求

12.1 概述

12.2 PRU_CONNECT请求

12.3 tcp_connect函数

12.4 PRU_SEND和PRU_SEND_EOF请求

12.5 tcp_usrclosed函数

12.6 tcp_sysctl函数

12.7 T/TCP的前景

12.8 小结

第二部分 TCP的其他应用

第13章 HTTP : 超文本传送协议

13.1 概述

13.2 HTTP和HTML概述

13.3 HTTP

13.3.1 报文类型 : 请求与响应

13.3.2 首部字段

13.3.3 响应代码

13.3.4 各种报文头举例

13.3.5 例子 : 客户程序缓存

13.3.6 例子 : 服务器重定向

13.4 一个例子

13.5 HTTP的统计资料

13.6 性能问题

13.7 小结

第14章 在HTTP服务器上找到的分组

14.1 概述

14.2 多个HTTP服务器

14.3 客户端SYN的到达间隔时间

14.4 RTT的测量

14.5 用listen设置入连接队列的容量

14.6 客户端的SYN选项

14.7 客户端的SYN重传

14.8 域名

14.9 超时的持续探测

- 14.10 T/TCP路由表大小的模拟
- 14.11 mbuf的交互
- 14.12 TCP的PCB高速缓存和首部预测
- 14.13 小结
- 第15章 NNTP : 网络新闻传送协议
- 15.1 概述
- 15.2 NNTP
- 15.3 一个简单的新闻客户
- 15.4 一个复杂的新闻客户
- 15.5 NNTP的统计资料
- 15.6 小结
- 第三部分 Unix域协议
- 第16章 Unix域协议 : 概述
- 16.1 概述
- 16.2 用途
- 16.3 性能
- 16.4 编码举例
- 16.5 小结
- 第17章 Unix域协议 : 实现
- 17.1 概述
- 17.2 代码介绍
- 17.3 Unix domain和protosw结构
- 17.4 Unix域插口地址结构
- 17.5 Unix域协议控制块
- 17.6 uipc_usrreq函数
- 17.7 PRU_ATTACH请求和unp_attach函数
- 17.8 PRU_DETACH请求和unp_detach函数
- 17.9 PRU_BIND请求和unp_bind函数
- 17.10 PRU_CONNECT请求和unp_connect函数
- 17.11 PRU_CONNECT2请求和unp_connect2函数
- 17.12 socketpair系统调用
- 17.13 pipe系统调用
- 17.14 PRU_ACCEPT请求
- 17.15 PRU_DISCONNECT请求和unp_disconnect函数
- 17.16 PRU_SHUTDOWN请求和unp_shutdown函数
- 17.17 PRU_ABORT请求和unp_drop函数
- 17.18 其他各种请求
- 17.19 小结
- 第18章 Unix域协议 : I/O和描述符的传递
- 18.1 概述
- 18.2 PRU_SEND和PRU_RCVD请求
- 18.3 描述符的传递
- 18.4 unp_internalize函数
- 18.5 unp_externalize函数
- 18.6 unp_discard函数
- 18.7 unp_dispose函数
- 18.8 unp_scan函数

18.9 unp_gci函数

18.10 unp_mark函数

18.11 性能 (再讨论)

18.12 小结

附录A 测量网络时间

附录B 编写T/TCP应用程序

参考文献

缩略语

编辑推荐

《CP/IP详解(卷3):CP事务协议.HP.P和UIX域协议》同样采用了大量的实例和实现细节，并参考引用了卷2中的大量源程序。

《CP/IP详解(卷3):CP事务协议.HP.P和UIX域协议》适用于希望理解CP/IP如何工作的人，包括编写网络应用程序的程序员以及利用CP/IP维护计算机网络的系统管理员。

《CP/IP详解(卷3):CP事务协议.HP.P和UIX域协议》的特点是内容丰富，概念清楚且准确，讲解详细，例子很多。

作者在书中举出的所有例子均在作者安装的计算机网络上经过实际验证。

在《CP/IP详解(卷3):CP事务协议.HP.P和UIX域协议》的最后，作者给出了许多经典的参考文献，并一一写出评注。

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>