

基于 SCTP 的客户/服务器的设计与实现

冉春玉, 张广军, 屈力, 周鹏, 薛果

(武汉理工大学 计算机学院, 湖北 武汉 430070)

摘要:随着 IP 网向多业务网的发展, 新一代运输层协议 SCTP 会逐渐替代 TCP 在下一代网络中获得更加广泛的应用。分析了 SCTP/IP 协议栈中数据传输过程, 介绍了 SCTP 协议所提供开发人员的套接口编程模型, 并给出一个在 Unix/Linux 系统上利用一到多形式套接口模型开发客户/服务器应用的示例。

关键词:SCTP; socket; 客户/服务器

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2007)06-0127-03

Design and Implementation of Client and Server Based on SCTP

RAN Chun-yu, ZHANG Guang-jun, QU Li, ZHOU Peng, XUE Guo

(College of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract: With the development of IP network towards multi service network, new SCTP protocol will be utilized wide instead of TCP in next generation network. Analyze data transmission process in SCTP/IP protocol stack in the article. Introduce the socket programming model for developers using SCTP/IP protocol suite. And give an example about client and sever application using one to many socket model on Unix/Linux platform.

Key words: SCTP; socket; client/server

0 引言

SCTP(Stream Control Transmission Protocol)是一个新的、可靠的面向数据报的运输层协议。在2000年已由 IETF 标准化为 RFC2960。IETF RFC2960 中定义的是^[1]:SCTP是可靠数据报传输协议,它运行于提供不可靠传递的分组网络上,如 IP 网。SCTP 发展了 UDP 和 TCP 两种协议的长处,它一方面增强了 UDP 业务并提供数据报的可靠传输;另一方面,SCTP 的协议行为类似于 TCP 并克服 TCP 的某些局限。SCTP 提供与 UDP 和 TCP 相似的服务。它保留了数据报的边界,而与此同时检测丢失的数据、重复的数据以及失序的数据。并像 TCP 那样给应用层提供可靠性、排序,以及全双工的数据传输服务。SCTP 还有拥塞控制和流量控制机制。SCTP 在客户和服务器之间建立关联,能够在所连接的端点之间提供多个流,每个消息各自可靠地按序投递。一个流上的某个消息丢失不会阻塞其他流上消息的投递。它还多宿提供传输级支持,使得单个 SCTP 端点能够支持多个 IP 地址,该特

性可以实现网络的容错。

1 SCTP 的服务

SCTP 主要是为最近引入 Internet 的一些应用程序而设计的,如 IUA(在 IP 上运行 ISDN)、M2UA 和 M3UA(电话信令)、H.248(媒体网关控制)、H.323(IP 电话)等,这些应用都需要一些比 TCP 所能提供的还要复杂的服务。SCTP 提供了这种增强的性能和可靠性。SCTP 向应用层提供的服务主要有如下方面^[1]:

(1)进程到进程的通信。应用进程访问特定的端口得到 SCTP 协议提供的服务。SCTP 使用 TCP 空间中所有熟知端口,也有一些外加的端口号,如 SIP 协议用 5060 端口, IUA 协议使用 9990 端口, M2UA 和 M3UA 协议分别用 2904 和 2905 端口。

(2)多重流。在传送音频或者视频的实时数据时, TCP 就满足不了性能要求, SCTP 允许在每一个连接中使用多重流服务,即关联。当某一个流被中断时,其他的流还能够继续交付其数据,满足了高性能传输的需求。

(3)连接多个网络。SCTP 关联支持多重归属服务,允许每一端使用多个 IP 地址,可以连接到不同的本地网络,在这种容许故障的处理方法中,当一条路径

收稿日期:2006-10-03

作者简介:冉春玉(1945-),男,湖北武汉人,教授,硕士研究生导师,研究方向为网络及网络数据库。

出现故障时,其他的接口可以用于数据不中断地交付。对于发送或者接受实时数据时,这种容许故障的处理方法很有用。

(4)全双工通信。每一个 SCTP 都有发送缓存和接受缓存,因而分组可以双向发送。

(5)面向连接的服务。SCTP 是面向连接的协议,当客户向服务器进程发送数据时,就发生以下情况:首先,两个 SCTP 彼此建立关联;其次,数据在两个方向交换;最后,关联终止。

(6)可靠的服务。SCTP 是可靠的运输层协议,在数据传输过程中使用确认机制来检查到达的数据的安全和可靠。

2 SCTP 数据传输过程

SCTP 名称中的“流”是指本协议能够处理每个关联中的若干个用户数据报流,并且为每个流中的用户数据报提供有序传送。“流”由消息序列组成,一个 SCTP 消息由 SCTP 头和数据两部分组成,它被封装在 IP 数据包中在 IP 网上传输。SCTP 的消息主要有 INIT, INIT ACK, COOKIE ECHO, COOKIE ACK, SACK, DATA HEATBEAT, DATA HEATBEAT ACK, SHUTDOWN, SHUTDOWN ACK, SHUTDOWN COMPLETE 等。

SCTP 联结的数据传输主要经历关联建立、数据传输和关联终止三个阶段^[2]。

2.1 关联建立

SCTP 中的关联概念要比 TCP 中的连接概念含义更广,一个关联的两个 SCTP 端点都向对方提供一个 SCTP 端口号和一个 IP 地址列表,这样每个关联都由两个 SCTP 端口号和两个 IP 地址列表来识别,在一个关联内的拥塞控制机制与 TCP 的拥塞控制机制类似。一个关联是由多个单向的流组成,各个流之间相对独立,可以单独发送数据而不受其他流的影响,也可以共同实现用户数据的有序递交。流的建立和拆除过程相对独立、简单。而关联的建立过程相对而言就比较复杂,是依靠四个消息 INIT, INIT ACK, COOKIE ECHO, COOKIE ACK, 通过四次握手方式来建立,具体过程如下:

(1)服务器调用 socket, bind 和 listen 函数被动打开;

(2)客户调用 connect 主动打开,发送 INIT 消息,告诉服务器客户的 IP 地址清单、初始序列号、分组的起始标志、客户请求的外出流数目和能支持的外来流数目。

(3)服务器发送 INIT ECHO 对客户消息确认,同

时包含一个状态魔饼。状态魔饼包含服务器用于确认本关联有效所需的所有状态,引入这种机制的目的在于增强协议的安全性,防止拒绝服务和伪装等潜在的攻击。

(4)客户以一个 COOKIE ECHO 消息回射服务器的状态魔饼。

(5)服务器以一个 COOKIE ACK 消息确认客户回射的魔饼是正确的,至此关联建立。

2.2 数据传输

SCTP 关联建立后,就可以在该关联上进行数据传输。内核将服务器应用进程缓冲区中所有数据拷贝到 SCTP 套接口的发送缓冲区。SCTP 取套接口缓冲区中的数据并把它发送到对端 SCTP,其过程基于 SCTP 数据传送的所有规则。在客户端读取数据的过程相反。SCTP 数据块的交互遵循选择应答机制,应答 SACK 按“延迟应答”方式发送,每收到两个消息发送一次。流量控制和拥塞控制部分沿用了 TCP 的算法。

2.3 关联终止

SCTP 使用 SHUTDOWN, SHUTDOWN ACK, SHUTDOWN COMPLETE 三个消息,以迅速地关闭联结。关联的一端关闭某个关联时,另一端必须停止从应用进程传入新的用户数据并发送。关联关闭请求的接收端发送完已经排队的数据后,执行关联的关闭。

3 SCTP 套接口编程模型

SCTP 套接口分为两类:一到一套接口和一到多套接口。在决定使用哪种接口形式时,需要考虑应用程序的多个因素,包括:服务器程序是迭代的还是并发的、服务器希望管理多少套接口描述字、应用进程希望维护多少个连接状态以及优化关联建立的四次握手过程中使得能够在其中第三个(也可能是第四个)分组交换用户数据是否重要等。

3.1 一到一形式套接口模型

一到一形式的套接口交互类似于 TCP 套接口^[3],如图 1 所示^[2]。服务器启动后,打开一个套接口,bind 一个地址,然后就等着 accept 客户关联。一段时间后客户启动,它也打开一个套接口,并通过四次握手方式建立与服务器的一个关联,假设客户向服务器发送一个请求,服务器处理该请求后向客户发回一个应答。这个循环持续到客户发起终止该关联为止。这样主动关闭关联之后,服务器或者退出,或者等待新的关联。

一到一形式 SCTP 套接口是一个类型为 SOCK_STREAM,协议为 IPPROTO_SCTP 的网际套接口,即协议族为 AF_INET 或者 AF_INET6。

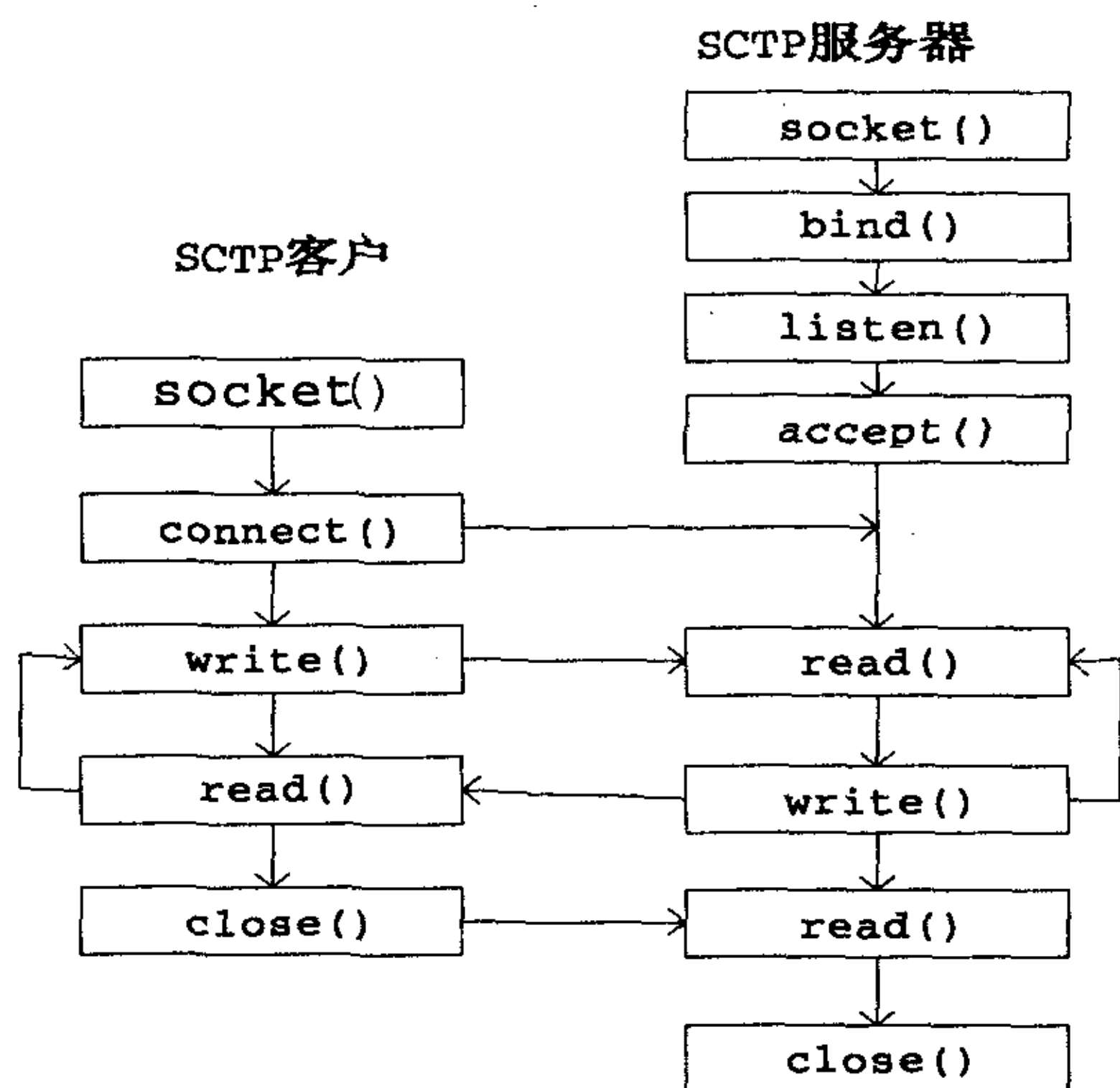


图1 SCTP一到一形式的套接口模型

3.2 一到多形式套接口模型

一到多形式给应用程序开发人员提供这样的能力:编写的服务器程序无需管理大量的套接口描述字。单个套接口描述字将代表多个关联,就像一个套接口能够从多个客户接收消息那样。在一到多式套接口上,用于标识单个关联的是一个关联标识。关联标识是一个类型为 `sctp_assoc_t` 的值,通常是一个整数。它是一个不透明的值;应用进程不应该使用由内核早先给予的任何关联标识。

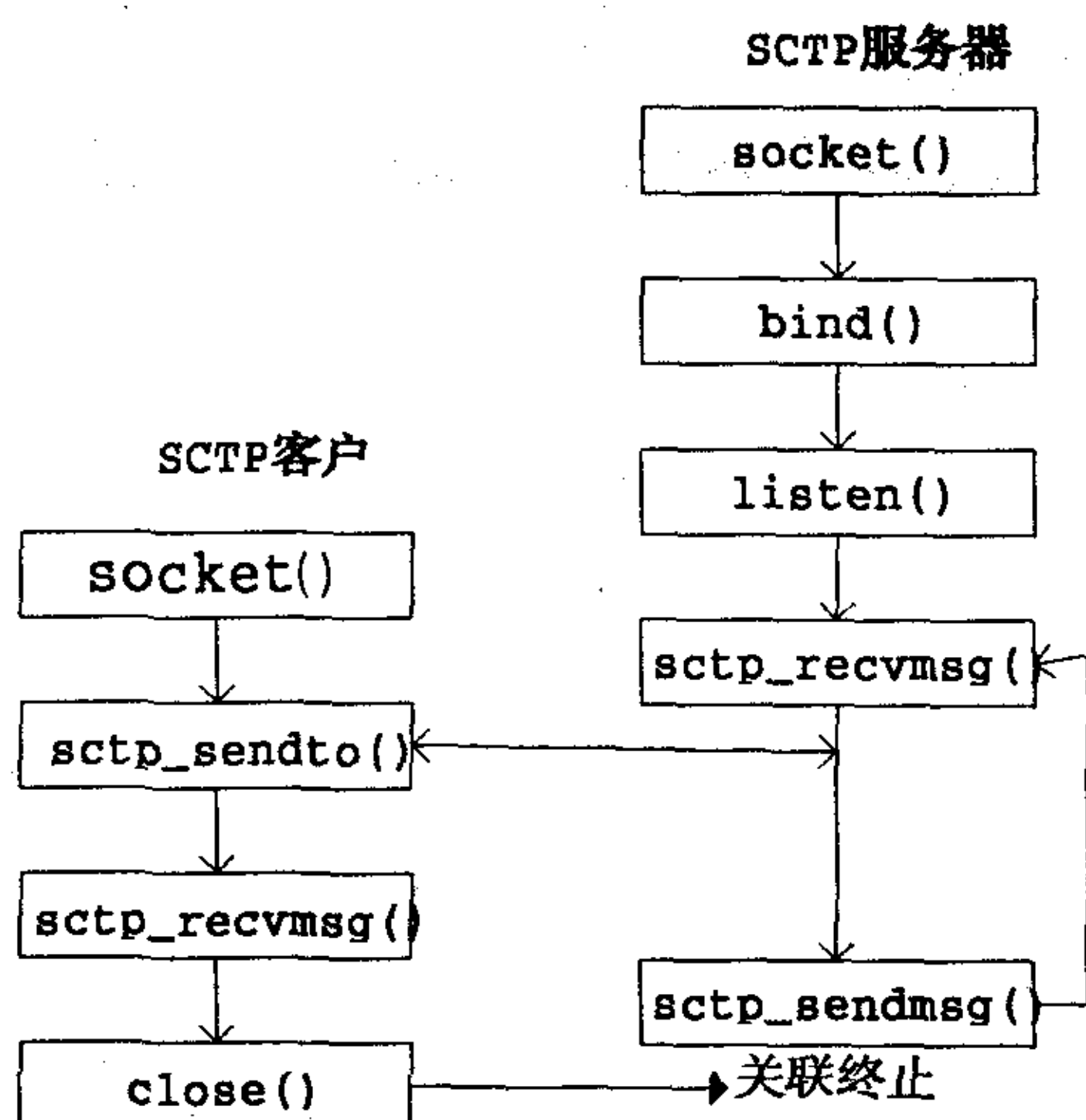


图2 SCTP一到多形式的套接口模型

SCTP一到多套接口的编程模型如图2所示^[2]。服务器启动后打开一个套接口,bind一个地址,调用listen来监听以允许客户建立关联,然后就调用 `sctp_rcvmsg` 阻塞于等待第一个消息的到达。客户启动后也打开一个套接口,并调用 `sctp_sendto`,它导致隐式建立关联,而数据请求由四次握手的第3个分组捎带给服务器。服务器收到该请求后进行处理并向该客户发回一个应答。客户收到应答后关闭其套接口,从而终止其上的关联。服务器循环回去接收下一个消息。

一到多形式 SCTP 套接口是一个类型为 `SOCK_SEQPACKET`, 协议为 `IPPROTO_SCTP` 的网际套接口,即协议族为 `AF_INET` 或者 `AF_INET6`。

4 SCTP 客户/服务器实现

用基本的函数编写一个完整的一到多形式 SCTP 客户/服务器。设计单纯使用 SCTP 的应用程序需要安装内核补丁或升级操作系统。限于篇幅,只给出程序中的关键步骤。实验环境如图3所示。客户程序运行在 Solaris 10 系统上,服务器运行在 Red Hat Linux 9.2 系统上,通过局域网连接。

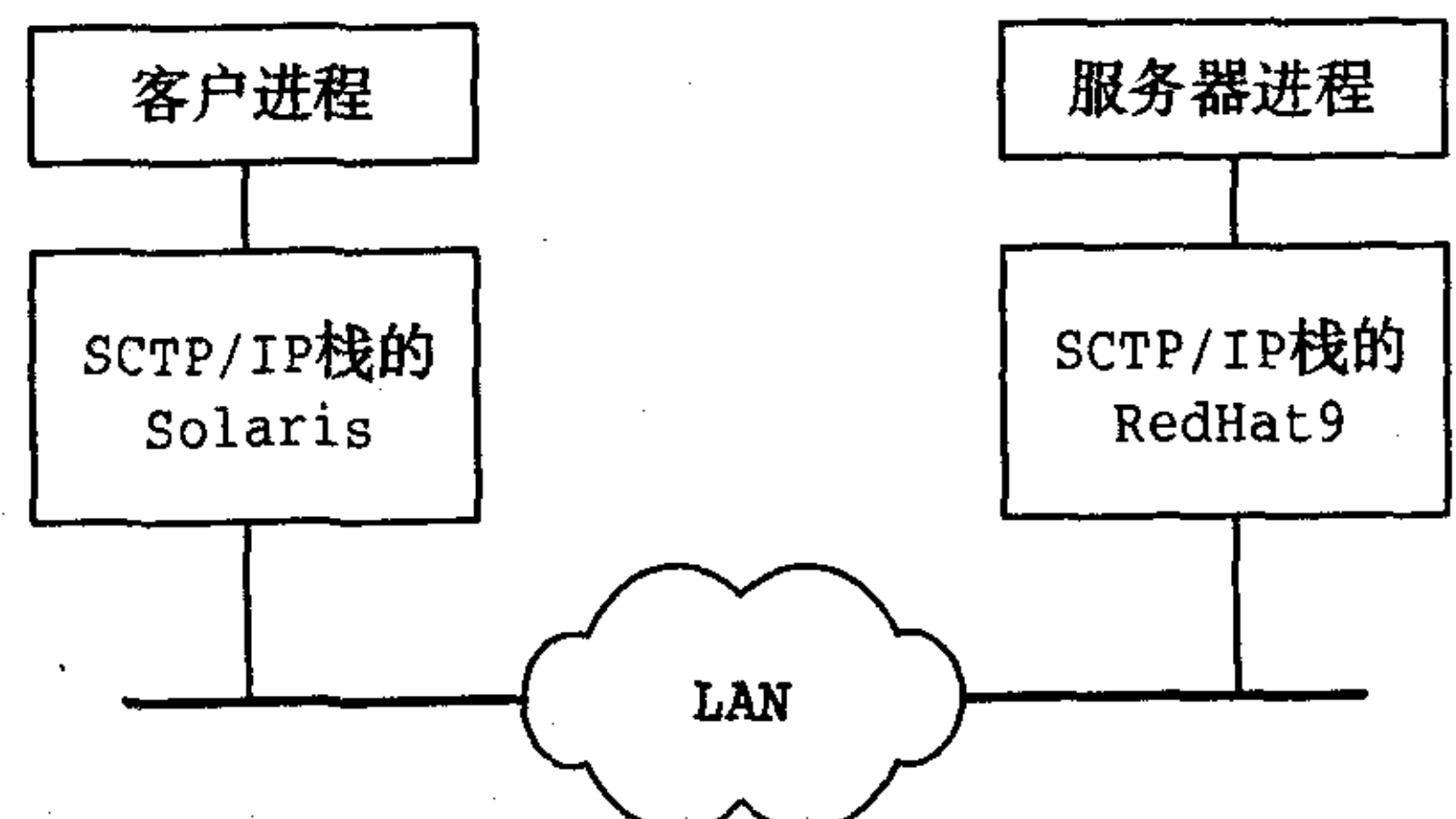


图3 SCTP客户/服务器实验环境

4.1 应用程序的服务器端关键代码^[4,5]

```
/* 创建一个 SCTP 一到多套接口 */
sock_fd = Socket (AF_INET, SOCK_SEQPACKET, IPPROTO_SCTP);

/* 捆绑通用地址和服务器的众所周知端口 */
bzero (&servaddr, sizeof (servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htonl (INADDR_ANY);
servaddr.sin_port = htons (SEVER_PORT);
bind (sock_fd, (SA *) &servaddr, sizeof (servaddr));
/* 预定通知,开启外来关联 */
events.sctp_data_io_event = 1;
setsockopt (sock_fd, IPPROTO_SCTP, SCTP_EVENTS, &events, sizeof (events));
listen (sock_fd, LISTENQ)
for ( ; ; ) { /* 服务器主处理循环,初始化客户套接口地址结构的大小,进入阻塞状态,等待客户消息 */
    len = sizeof (struct sockaddr_in);
    rd_sz = sctp_rcvmsg (sock_fd, readbuf, sizeof (readbuf), (SA *) &cliaddr, &len, &msg_flags);
    /* 服务器发送消息 */
    sctp_sendmsg (sock_fd, readbuf, rd_sz, (SA *) &cliaddr, len, sri.sinfo_ppid, Sri.sinfo_flags, sri.sinfo_stream, 0, 0);
}
```

4.2 应用程序的客户端关键代码^[4,5]

```
/* 验证参数并创建一个套接口 */
if (argc < 2) err_quit ("missing argument - use % host [echo] \n", argv[0]);
```

(下转封三)

$\Delta T_{正常} = [0 \ 0.0311 \ 0.0044 \ 0.0215 \ 0.0087]$

用阶数 $p = 5$ 的模型对正常和故障状态下的几组振动信号进行估计, 每组数据的长度也为 $N = 1024$ 点, 所得的模型系数如表 4 和表 5 所示。

表 4 正常状态下的模型系数表

	a_1	a_2	a_3	a_4	a_5
1	1.0000	-0.0831	-0.0070	-0.2128	0.1281
2	1.0000	-0.0804	-0.0030	-0.2074	0.1219
3	1.0000	-0.1102	-0.0078	-0.1837	0.1201

表 5 故障状态下的模型系数表

	a_1	a_2	a_3	a_4	a_5
1	1.0000	-0.5390	0.1254	-0.0232	0.2494
2	1.0000	-0.5120	0.1216	-0.0457	0.2914
3	1.0000	-0.5348	0.1407	-0.0460	0.2561

表 4 中数据为用第三档正常齿轮的振动测试信号所求得的模型系数。表 5 中数据为用同一档坏齿轮的振动测试信号所求得的模型系数。从表 4 中的数据可以看出, 正常状态的模型系数全都落在容差范围之内, 也即属于 $T_{正常} \pm \Delta T_{正常}$; 从表 5 中的数据可以看出, 故障状态的模型系数都或多或少地有几个系数不落在容差范围之内。从实验 1 和实验 2 中可以看出, 根据模型系数的取值可以区分正常和故障状态的齿轮, 从而实现了齿轮箱的故障检测。

4 结 语

在分析了 AR 模型系数的基础上, 提出了基于“系

数-故障”的故障检测思想的新方法, 该方法通过判断模型系数是否落在正常状态的容差范围内, 能有效地检测齿轮的工作状态, 且这种方法具有算法简单、结果直观等优点。

参考文献:

[1] 金少先, 金中石. 齿轮振动的边带分布特征与故障诊断实例[J]. 振动工程学报, 1999, 12(3): 429-433.

[2] 徐 科, 杨德斌, 徐金梧. 小波变换在齿轮局部缺陷诊断中的应用[J]. 机械工程学报, 1999, 35(3): 105-107.

[3] 高清维, 李川奇, 庄镇泉. 齿轮箱振动信号的复杂度分析[J]. 电子测量与仪器学报, 2002, 16(2): 1-4.

[4] 王华民, 陈 霞, 安 钢, 等. 基于高阶累积量的齿轮箱故障诊断研究[J]. 机械强度, 2004, 26(3): 247-249.

[5] 王世一. 数字信号处理[M]. 北京: 北京理工大学出版社, 2004: 323-325.

[6] Burg J P. Maximum entropy spectral analysis[D]. Stanford, CA: Dept. of Geophysics, Stanford Univ., 1975.

[7] Marple S L. A new autoregressive spectrum analysis algorithm[J]. IEEE Trans on ASSP, 1980, 28: 441-454.

[8] Kay S M. Modern Spectral Estimation: Theory and Application[M]. Englewood Cliffs, NJ: Prentice Hall, 1988.

[9] Marple S L. Digital Spectral Analysis with Applications[M]. Englewood Cliffs, NJ: Prentice Hall, 1987.

[10] 胡广书. 数字信号处理[M]. 北京: 清华大学出版社, 2000: 365-366.

(上接第 129 页)

```
if(argc > 2){printf("echoing messages to all streams \n");echo_
to_all=1;}

sock_fd = socket(AF_INET, SOCK_SEQPACKET, IP-
PROTO_SCTP);

/* 设置服务器地址:从表达格式转换成数值格式,并和端
口号组合成请求的目的地 */
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
servaddr.sin_port = htons(SERV_PORT);

/* 设置一到多 SCTP 套接口的通知预订 */
setsockopt(sock_fd, IPPROTO_SCTP, SCTP_EVENTS,
&evnts, sizeof(evnts));

/* 调用函数来请求服务 */
serviceRequest(stdin, sock_fd, (SA *) $servaddr, sizeof(serv-
vaddr));

close(sock_fd);
```

5 展 望

SCTP 是为传输信令业务流而开发的。但是它本身所具有的、优于 UDP, TCP 的一些先进协议机制, 如

选择性确认、快速重传、无序递交和支持多种网络特性等, 使得 SCTP 能够在一定程度上满足高性能传输的要求。这会给它带来更为广泛的应用需求。目前已有各种操作系统支持 SCTP, 如 Linux, AIX, solaris, Free BSD 和 Windows。SCTP 会成为下一代 IP 网络面向连接的可靠的传输层协议而取代 TCP 的地位, 在下一代网络以及下一 VoIP 业务网中会有更为广泛的应用空间。

参考文献:

[1] IETP RFC 2960. Stream Control Transmission Protocol[S]. 2000.

[2] IETF RFC 3257. Stream Control Transmission Protocol Applicability System[S]. 2000.

[3] Forouzan B A, Fegan S C. TCP/IP 协议族[M]. 谢希仁译. 北京: 清华大学出版社, 2006.

[4] Stevens W R. UNIX 环境高级编程[M]. 尤晋元等译. 机械工业出版社, 2002.

[5] Haviland K, Gray D, Salama B. UNIX 系统编程[M]. 舒明译. 电子工业出版社, 2003.