

62-66

计算机网络

网络协议

仿真

设计

14

计算机科学 Vol.20 No.1 1993

# 面向多机环境的网络协议仿真与测试

于忠民 赵宏 刘积仁

TP393

(东北工学院计算机系, 沈阳110006)

## 摘要

如何在多机环境下单独开发调试计算机网络某一层协议软件, 保证其正确性、易维护性及高效性是网络开发者所遇到的一个问题。本文通过对网络协议软件的一般开发方法的研究, 提出了在多机环境下利用层间插件机制建立网络协议的仿真测试环境的思想, 并介绍了应用Socket进程通讯机制设计协议仿真与测试环境的基本方法及以此方法设计的NPSTE (Network Protocol Simulation & Testing Environment) 系统的实现。

网络协议软件的开发, 通常是按照 ISO 标准进行的。由于 ISO 为用户提供了标准化的协议原语调用接口<sup>[1-5]</sup>, 使得各层协议具有良好的透明性, 且可以独立进行设计。目前, 在协议的规范描述、自动生成、验证及测试等方面已取得了很大进展, 提高了协议的开发效率和质量。但由于协议各层具有相对独立性, 而且开放的接口较多, 各层协议又都是自封闭的, 只通过相互的接口原语进行联系, 从而很难在网络环境还不完善时对某层协议单独进行调试或模拟运行。特别是当多个开发者同时进行多层协议软件的开发时, 这种情况更易出现。所以用户必须等到

其它层协议完成后一同连接运行调试, 或者利用构造的特殊环境来运行调试自己的协议, 这样不仅推迟了协议错误查找阶段, 又不易区分所出错误的原因, 给协议开发者带来了不便, 影响了协议开发效率和质量。

针对上述问题, 如果能在协议实现之前就对各层协议单独进行运行调试, 及早发现并纠正错误, 则可以大大提高网络协议开发效率。近来, 在网络协议工程研究领域, 从协议描述到协议实现已提出许多方法, 并建立了各种开发环境, 提高了协议描述、纠错及实现等过程的自动化程度和可靠性。但许多方法仍处于探索中。本文提出一种比较直观的方法 (图1), 它近于实际, 又易于实现, 这就是建立协议仿真测试环境, 解决某层协议单独开发时调试协议软件的困难。

### 一、多机环境下的网络协议仿真测试

如图2所示, 多机环境下的网络协议仿真测试环境主要是为协议提供一个完整的运行调试环境, 它允许用户单独地开发和调试某一层协议, 即在其它相关层协议还未完成时, 借助特定的环境调试本层协议。这需要首先将被开发层协议与仿真测试环境相连,

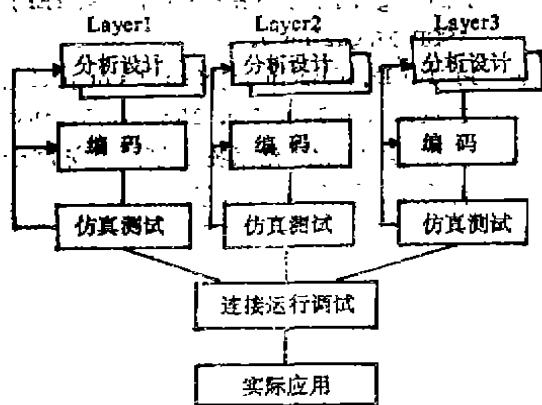
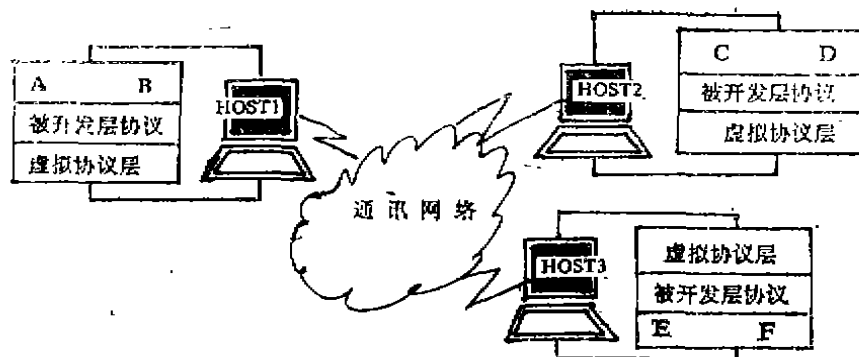


图1. 带有独立仿真调试的开发方法



(其中:A、B、C、D、E、F为仿真测试驱动用户,HOST1、2、3为仿真主机)

图2.多机环境下的协议仿真测试

并且设置运行环境,通过驱动用户及虚拟层协议的建立组装环境,将被开发层协议封装在一个可仿真运行的环境中。

图2中表示了有三个不同HOST的环境。也可以在一个HOST上仿真多个主机环境。仿真测试环境主要包括:

(1) **仿真测试驱动用户** 其主要任务是负责组装协议的原语帧,然后将其交给被开发层协议,驱动被开发层协议运行一个上层协议原语。同时通过收听进程接收从被开发协议来的原语,并将其存入收发队列供用户查阅。驱动过程可以通过下述方式实现:

- 人机交互方式:通过驱动用户交互地编辑调用原语。
- 自动驱动方式:系统自动从事先定义的驱动序列中读取调用原语。

(2) **被开发层协议** 用户开发的并准备调试的协议。

(3) **虚拟协议层** 是为被开发层协议提供的下层协议仿真部分,每当被开发协议向下进行原语调用时,虚拟协议层将根据原语中的地址信息将其解释成相应的原语,然后发送到目的主机的相应进程。

(4) **通信网** 建立于UNIX系统Socket机制基础上,保证仿真测试环境中各个HOST的进程之间的信息传递,各层协议间的信息交互都是通过Socket包的传递实现的。

## 二、仿真测试环境的内部机制

NPSTE是一个网络协议仿真测试环境,它应用了Socket技术,根据ISO所提供的七层协议标准,构造了一组新的网络协议层插件接口(Slot),实现了协议原语级动作的仿真与测试。NPSTE在保持了原有协议原语调用接口的基础上,扩充了用于协议仿真测试的机制,可模拟被开发层协议的上层及下层协议用户,通过原语调用触发被开发协议,将独立的协议连接起来,并为被开发层协议提供一个完整的运行环境,实现协议的仿真及测试。

如图3所示,上、下层Slot是被开发层协议与邻层协议的原语调用接口,发送Slot将本层协议发向其它层(或被模拟的)协议的原语转发给目的层协议,而接收Slot则将收到的原语送给本层协议,触发相应的动作。NPSTE通过Slot将各自独立的协议层连接起来,使得下述各种协议能通过通讯连接起来。上、下层协议用户可以是以下形式:1)被模拟的某层协议;2)被开发的某层协议;3)已实现的某层协议。

总之, NPSTE具有如下特点:

- 通用性: Slot接口采用了ISO标准。
- 灵活性: 当完成协议仿真测试后,可方便地拆除Slot,形成实际的协议;另外, Slot还可在协议原语级扩充其它功能。
- 透明性: 用户只需要保证协议调用原语参数的正确性。

- 直观性：NPSTE利用窗口为每一协议进程分派一监视窗口，动态地跟踪各层协议。

- 独立性：各个网络结点和各层协议是相互独立的。

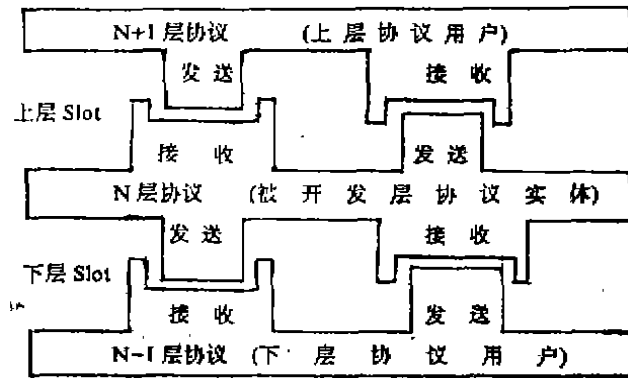


图3. Slot接口

### 1. 进程通讯机制

Socket是由4.1BSD系统提供的一种进程通讯机制，它可以实现不同主机下的不同进程间的通讯<sup>[6,7]</sup>，Socket实际上是一开放的传送层协议编程平台，NPSTE为每个主机的协议进程通讯链路分派不同的端口，各端口以主机名和端口号标识信息的名字，实现数据的发送和接收。

在UNIX 4.1BSD系统下，有Virtual Circuit、DataGram、RawSocket三种方式，分别属于有连接(Connect)和无连接(Connectless)两种通讯方式，每次可发最大数据长度为8.5K字节，并有限定的接收缓冲空间。NPSTE采用了有连接的DataGram方式实现各层协议进程间的通讯。

为了更加透明和方便地使用Socket，NPSTE构造了一个新的库NewSocket，并向UNIX系统用户开放。它在原来的Socket机制的基础上，隐含了Socket原有的一些复杂的数据结构，使用户只需了解目的进程的主机名、相应的通讯端口号和本进程要发送或接收的数据(或数据区)及其长度，就可以实

现进程间的通讯。NewSocket库的主要函数有：

初始化 `int init-socket(char*host-name, int local-port)`

关闭 `close-socket()`

等待读 `socket-data *wait-socket(int wait-time)`

扫描读 `socket-data *read-socket()`

循环读 `socket-data *listen-socket()`

发送 `int socket-send(char*dest-host, int dest-port, char*DATA, int data-len)`

### 2. Slot 机制

NPSTE提供了一开放式的Slot函数库，用户可以按ISO的协议原语调用原则与Slot库进行连接。Slot是建立在Socket基础上的协议原语接口，它可以连接于被开发的ISO标准的协议并为其提供接口服务。发送Slot将原语调用参数

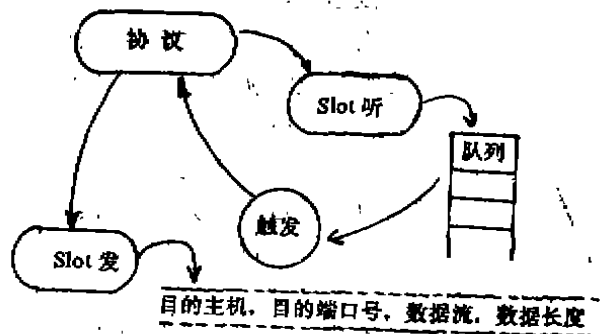


图4. Slot的内部机制

合并为流数据发送到目的进程，而接收Slot将听到的数据分解成相应的原语加入队列中，并根据接收的原语触发本层协议自动机动作。图4表示了NPSTE的Slot内部机制。

### 三、协议的仿真与测试

#### 1. 协议仿真

NPSTE提供了被开发层协议以外的其它层协议的仿真功能，为被开发层协议建立了上、下层协议支持环境。仿真协议层收到一帧后，根据本层的自动机原理及本帧所含地址向目的层协议发送相应的帧。

对上层协议用户的仿真，NPSTE采用人

机交互或自动的原语调用方式向下层协议发一原语请求，驱动下层协议产生相应动作。由于NPSTE系统是建立于窗口环境上的，因此具有良好的用户界面。NPSTE在用户驱动窗口中提供了标准原语发送及接收命令，用户可以选择要发送的原语参数，如目的主机、目的用户、原语类型及要发之数据，组装成要发送的帧；同时，系统为用户保存每次连接过程中所收到的原语及发送出去的原语，用户可以用相应的命令列出队列中的原语，或查看某一原语中的信息。

NPSTE对协议的仿真限于原语级，而将原语的内部具体处理过程封闭在一个黑箱中，从而可实现协议外部功能的仿真。

### 2. 协议测试

NPSTE在Slot机制中也可以增加协议原语的执行检测机制（图5）。测试驱动控制器从驱动用户或事先编辑的原语序列中输入一原语，向被测试层协议发一原语请求，驱动被测试协议自动机动作，并完成被测试协议的上层调用原语的检测；而协议检测器则主要跟踪被测试协议的原语执行过程，监视它对上、下层原语调用所做出的响应，并将响应结果与相应的原语测试原则相对照，以测试其正确性与否，如测试其是否超时或出现错误应答等。

NPSTE在驱动序列中，规定了原语的类型、发向目的、数据、超时信息等；而测试

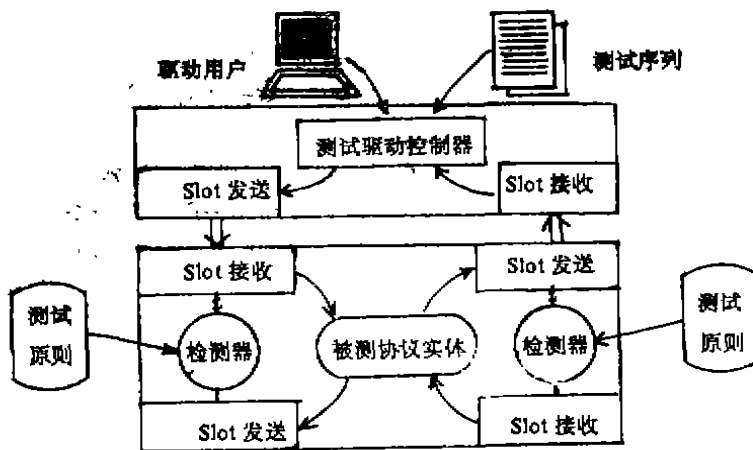


图5. NPSTE的协议测试机制

原则主要包括了正确的协议对各原语的响应情况，即指出了每个原语的下个动作的可能结果。

### 3. 协议容错处理的测试

为了能更生动地仿真测试协议运行时的各种情况，NPSTE可以提供一种制造错误原语的功能，用户可以在驱动序列或通过驱动窗口选定要产生的错误类型，NPSTE在原语处理中自动插入相应的错误，以测试协议在错误产生情况下的处理能力。

### 四、实现环境

NPSTE作为计算机网络协议开发支持环境PIDE中的一个部分，现已在SUN工作站上初步建成，可实现网络层和传送层协议的仿真调试环境。NPSTE中的协议可在不同的主机上运行，也可在中心主机上建立的虚拟主机上运行。中心主机是NPSTE的主控中心，它为被开的协议和仿真的协议以及驱动用户提供跟踪窗口，协议的运行情况，

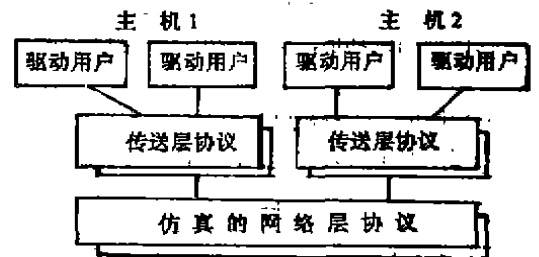


图6 对传送层协议的仿真调试

如原语请求帧的形成、发送及帧的接收及应答等原语的流动，可以直观地在窗口上显示出来，从而使用户可及时地发现所调试协议的正确与否。如图6表示了实现环境中，在被开发的传送层协议上建立两个用户结点进行传送层协议仿真调试的情况。

在上述仿真调试过程中，主机HOST1上的用户

HOST1 USER1	HOST2 USER2
用户层窗口:	
TCONrequest to HOST2 USER2	TCCONindication from HOST1 USER1
TCONconfirm from HOST2 USER2	TCCONresponse to HOST1 USER1
传送层窗口:	
TCONrequest from HOST1 USER1	NCONindication from HOST1
NCONrequest to HOST2	TCCONindication to HOST2 USER2
NCONconfirm from HOST2	TCCONresponse from HOST2 USER2
TCONconfirm to HOST1 USER1	NCONresponse to HOST1

USER1和HOST2上的用户USER2间通过传送层进行通讯时,用户可交互地发送原语调用命令,并观察其响应结果,查看收发队列内容。通过各层的原语可随时从跟踪窗口上显示出来。例如,上面是在一次连接中,原语流在用户层和传送窗口上的显示情况。

NPSTE的实现有助于网络协议软件的独立开发,具有实际意义。在该环境下,我们已开发了ISO传送层和网络层协议,并对其进行了仿真调试运行。从使用情况看,其效率比原始的手工调试方法有明显提高;此外,NPSTE还可以在原语级上进一步扩充,加深理论研究,以便提供更加完善的协议仿真测试环境,进一步提高网络协议的开发效率。

感谢 作者在研究开发过程中,得到了张伟和张立东博士的大力帮助,在此表示感谢。

#### 参考文献

- [1] Information processing system-Open System Interconnection Transport service definition ISO 8072, 1988.
- [2] Information processing system-Data Communications-Network service definition, ISO 8348, 1988.
- [3] Information processing system-Data Communications-Network service definition, Addendum 1, Conn-
- nectless mode transmission ISO 8348/ADD1, 1988.
- [4] Information processing system-Data Communications-Network service definition, Addendum 2, Network layer addressing ISO 8348/ADD2, 1988.
- [5] A.S.Tanenbaum, Computer Network, 1988.
- [6] David Coffield and Doug shepherd, "Tutorial guide to Unix sockets for network communications", Computer communications, Vol 10 no 1 feb, 1987.
- [7] W.Richard Stevens, UNIX Network Programming, Prentice Hall Software Series, 1990
- [8] B.Wolfinger, O.Probnik, "Simulation of Protocol Layer of Communication in Computer Network", 《Computer Network and Simulation I》.
- [9] 肖可 胡道元, "OSI标准化测试系统的实现和研究", 《计算机网络基础研究论文集》.
- [10] 刘积仁 赵宏 李华天, "计算机网络协议开发支撑环境—PIDE", 《通信学报》, Vol.12 No.3, May, 1991