量不为负，交换合法。

(4)循环反转(loop reversal) 改变循环进行的方向，经常和别的迭代空间变换技术相结合，因为它改变了依赖向量。

(5)条块分割(strip mining) 可以用来调整并行操作的粒度。

(6)循环收割(loop shrinking) 当一个循环当中存在着依赖，阻碍了迭代的并行执行，只要依赖距离大于1，依然可以开发出一定程度的并行性。考虑一个紧嵌套循环，假设数组的依赖距离均为常数，且最小依赖距离为k，可以k为步长，将迭代空间划分成若干块，每块内部可并行执行，块与块之间可以串行执行。

(7)循环分布(loop distribution) 将一个单一的循环分成若干个子循环，每个子循环有和原循环相同的迭代空间，它的循环体是原循环体的一个子集。循环分布可用来产生紧嵌套循环，所产生的子循环往往具有更少的依赖性。循环分布不能对循环体中的语句作任意划分，分布是有条件的，其算法比较复杂。

其它诸如循环合并(loop fusion)，循环归一(loop coalescing)等变换对程序的并行性都有一定的意义，这里不再一一介绍。

结束语 设计一个OO并行编程系统，本身是一件很复杂的工作，一般应该使它达到四个目标：高效率、易使用、移植方便以及广泛的应用适应性。

为了实现这些目标，有很多问题需要进一步研究，例如：(1)如何确定用户和系统之间的任务界面；(2)OO继承机制以及对象作为参数对并行性分析的影响；(3)如何利于并行性识别设计OO程序中间形式描述；(4)任务的粒度控制和优化问题，包括权衡通讯与并行；(5)通讯和同步的优化问题；(6)移植问题，不仅要考虑代码的移植，还应考虑性能的移植问题；(7)在开发数据并行性时，如何解决数据的自动分布问题。

(下转第13页)

(上接第63页)

除了吞吐量的提高，PF_ATM协议栈还有其它一些优点。它为建立点到多点连接提供了一个简单接口，这个特征对电视会议系统等应用来说是诱人的。基于Socket的接口还使得应用进程能够在一个连接的生存期内改变QOS参数，改变流控约定等等，这对实现交互式的动态电视会议系统也是十分有用的。

结束语 为了充分利用ATM的高带宽、服务质量约定等特性，我们提出了一个新的ATM专用协议栈，给出了该协议栈的语义描述，并通过模拟一个视频会议系统验证了该协议栈的优越性。实验模拟结果表明，这种新的协议结构有效地利用了ATM的优良特性，尤其适合多媒体数据传输应用。

鸣谢 本文的写作和实验模拟得到上海邮电管理局科技处的大力协助，在此表示感谢。

参考文献

[1] M. Laubach, Classical IP and ARP over ATM, Internet RFC1577, Jan. 1994

[2] B. Manning and R. Colella, DNS NSAP Resource Records, Internet RFC1706, Oct. 1994

[3] ATM Forum LANE_SWG, "LAN Emulation Over ATM Specification" V1.0, Jan, 1995

[4] 4.4BSD Programmer's Reference Manual, Apr. 1994

[5] D. Saha, et al., A Video-conferencing tested on ATM: Design, Implementation, and Optimizations, Proc. IEEE Conf. on Multimedia Computing and Systems, Apr. 1995

[6] R. Black and S. Crosby, Experience and Results from the Implementation of an ATM Socket Family, Technical Report of Cambridge University Computer Laboratory, Oct. 1993

[7] 方震、龙浩、吴时霖，ATM局域网仿真体系结构的研究，《小型微型计算机系统》

[8] 方震、龙浩、吴时霖，ATM网络的流量控制，《计算机科学》，№.3，1997

一种新的ATM专用协议栈的语义描述

The Semantics of a New ATM-Oriented Protocol Stack

方 震 叶小萌 王 睿

(复旦大学计算机科学系 上海200433)

摘 要 For a smooth upgrading to ATM from current applications, especially for a thorough exploitation of the attractive features ATM provides, we propose a new ATM-oriented protocol stack with its key semantics. The API is the popular Socket interface. Implementational results show that this new protocol architecture makes use of ATM features satisfactorily, and promises a good basis for multimedia applications

关键词 ATM, Socket, TCP/IP, Protocol family, NSAP, Segmentation/Reassembly

一、引言

ATM(异步传输模式)提供了诸如高带宽、服务质量(QOS)约定等一系列优良特性,可望成为未来B-ISDN的首选基础传输技术。要使ATM技术真正得到广泛应用,在进入直接开发ATM应用程序的阶段之前,须保证从现有传统网络技术到ATM的平滑过渡。目前,传统网络与ATM互连有两种主要方式:LANE(局域网仿真)和IP and ARP Over ATM。LANE是ATM Forum推荐的ATM接口规范,其主要优点是对多种网络层协议的支持,但明显的缺点是:不能使用ATM的QOS属性;最大帧尺寸受以太网1500字节MTU的限制,影响了文件传输效率;只能使用UBR(未定义位速率)和ABR(可用位速率)服务,不能使用CBR(恒定位速率)和VBR(可变位速率),不适合对时延敏感的多媒体数据的传输;不同的虚拟网不能共用同一个虚连接。对于第二种互连方式,现有的大部分产品以IP Over ATM方式运行于ATM之上,它与LANE方式的区别在于,重新实现一个新的数据链路层,而LANE是仿真数据链路层中的MAC子层。它将IP直接映射到ATM层能克服LANE模型的某些限制,减少地址转换的开销,可以在逻辑子网一级定义QOS。但单纯的IP Over ATM方式的缺点是,不能支持除了IP之外的SNA, NetBios, IPX等等其他网络层协议;事实上的标准RFC1577没有定义如何处理广播与多目广播,与上面两种方案不同,我们的途径是提供ATM API(应用程序接口)。

二、协议结构

我们的目标是,在保持现有应用程序可用性的前提下,在ATM连接接口上同时支持TCP/IP等传统协议族和新的专门面向ATM的协议族,发展新的多媒体应用环境。目前,TCP/IP等传统协议族支持应用进程间包数据通信,但未考虑下层ATM传输网络的存在;面向ATM的协议族支持建立应用进程驱动的ATM虚连接,按照QOS需求传送数据,包括AAL1虚电路、AAL5有序包等多种不同的数据类型。新的多协议栈结构(这里我们仅以TCP/IP为例,未考虑其他现有协议)与当前基于Socket的应用程序接口兼容,以确保现有应用程序可以毫无问题地运行,并且为QOS参数的描述和传递提供了相应的功能扩展。

网络通信的主要瓶颈之一是协议数据单元(PDU)的处理开销,这在很大程度上又是由于现有协议分层结构的大量冗余信息所致。如在PF_INET协议族看来,ATM是无异于802.3/以太网或802.5/令牌环的另一种链路层标准,为了向所有链路层用户提供统一接口,PF_INET协议栈体系就不得不忽略ATM的诸特点,从而造成ATM层和AAL层的功能在上层协议中重复出现。然而,PF_ATM协议族则专门面向ATM传输网络,其协议栈十分简单,例如,PF_ATM提供应用进程至应用进程的直接虚连接,应用进程PDU的解复用在网络接口处根据连接标识直接进行,而在PF_INET这类"通用型"协议结构中则实际上要经由一个解复用栈。在PF_

ATM 中，协议栈的唯一公有功能是分段和重组，其它处理都根据连接句柄在各个连接上按照特定的需要实施。这样，PF_ATM 的协议数据单元处理开销大大降低。另一方面，连接句柄可帮助完成各个连接特定的处理需求，如提供接近硬件极限的快速数据信道，提供复杂的差错控制、流量控制机制等等。PF_ATM 的另一优点是控制流与数据流的分开，这使得数据传输机制简单、快速，而控制机制可以较为复杂和有效。我们利用这个特点建立设备至设备的内核数据通道，大大提高了多媒体传输应用程序的性能。

图1是本协议并入现有协议结构的示意图，即采用 Socket 接口，并定义一个面向 ATM 的专用协议族。可以看到，图1中包容了两条不同的协议体系：ATM 专用协议族和 IP 协议族。应用程序可以经由 IP 协议族(PF_INET)，透明地访问到 ATM 网络。应用程序也可以经由 ATM 专用协议族(PF_ATM)直接访问 ATM 的传输服务，与 ATM 专用协议族 PF_ATM 相应的地址族称为 AF_ATM。在发送方，将通过 ATM 连接传送的 PDU 直接从应用进程交由 ATM 适配层进行分段，分成53字节的 ATM 信元；在接收方，ATM 信元重组后直接上交 Socket 层。下面我们先叙述经由 PF_ATM 协议族的应用程序接口语义要素，再给出协议栈结构的较为详细的说明。

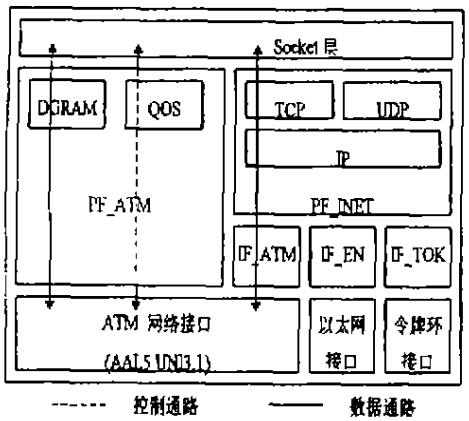

图1 协议结构

2.1 寻址

连接建立阶段的一个问题是 ATM 地址的获得，因为不能要求应用程序和用户给出20字节的 ATM NSAP(或8字节的 ISDN E.164)地址。我们拟采用 Internet 域名服务(DNS)为应用进程提供名字解析，即 DNS 被扩展为支持 NSAP Resource Records (Internet RFC1706)，DNS Server 维护 NSAP 库，受理 DNS Client 发出的主机名解析查询请求，给出 ATM NSAP 地址。DNS Client 和 DNS Server 之间的通信可以按照正常的 TCP/IP 协议进行。

2.2 控制

在传统的 Socket 接口中，控制和数据是在同一个 Socket 上传送的，通常用 ioctl( )或 setsockopt( )功能调用来传送控制参数，但是这些函数对于从协议模块至应用进程的"上行"控制信息流无能为力。在最近的 4.4BSD Socket 中，sendmsg( )和 recvmsg( )函数所用的信息结构新增加了若干控制域，从而这些函数可以在应用进程和协议模块之间传递控制信息，但是在数据处理与控制由不同进程担负的场合下难以胜任这种控制信息和数据信息的复用处理。例如，在远程会议中，控制部分由会议控制实体负责，而数据是由另一个不同的进程或硬件设备产生和使用的，所以我们建立一个不同于数据传递 Socket (D-Socket)的控制 Socket (C-Socket)。

在 ATM 协议族中，C-Socket 与 QOS 协议模块相连，该 Socket 控制一个或多个 D-Socket，并支持应用进程与 QOS 模块间双向的信息传递。QOS 模块可以通过 C-Socket 向应用进程发送信息，反映连接状态及网络环境的变化。

如果用一个 ATM PVC(永久式虚连接)来负责两台主机之间所有的数据传输，由于每一个上层的较大 PDU 都直接分段产生一批突发信元，当网络负载加重时，传输速率便不堪承受，所以我们对每一次数据传输都使用一个 ATM SVC(交换式虚连接)。在 ATM 层，每个连接有一个 association 记录，每个 Socket 与一个 association 记录相关联。association 结构包括若干控制域，如 VCI(虚连接标识)、接收方进程指针等。对于普通的连接，接收进程在将所收到的信元以 AAL 服务数据单元(SDU)上交给 Socket 层之前先要完成诸 AAL 层功能的处理。每个 ATM 网络接口对应一个特殊的 association 用于 ATM 虚信令，对这些 association，接收进程通过 C-Socket 把这些信元原封不动地上交。当通信双方是本地通信时，与该连接两端相对应的两个 Socket 可以互相直接访问，这时 association 记录其实并无存在的必要，这一点与 TCP Socket 是相似的；当双方是通过远程连接通信时，核心层通过 ATM 层的 association 结构来辨识每一个 ATM 连接。

2.3 数据报服务

采用 CBR,UBR 或 VBR 的多媒体数据流可以采用 AAL5的数据报服务实现.ATM 数据报 Socket 与 IP 协议族的数据报 Socket 不同,后者可以向多目的发送数据报,或者从多目的接收数据分组;而我们的 ATM 数据报 Socket 与 ATM 虚连接语义紧密相关,点对点连接是双向的,点对多点连接却是单向的.与一个虚连接相联的 Socket 可以将数据再定向到某个设备。下例说明了 ATM 数据报连接的建立过程。

源主机:创建一个 D-Socket 并把它与一个本地端口号联编,该本地端口也作为 ATM 连接建立信令的一个应用进程标识.在 TCP 和 UDP 中,这个本地端口是众所周知的或是由操作系统动态分配的。

```
data_sock=socket(AF_ATM,SOCK_DGRAM,0);
bind(data_sock,(local_ATM_address,local_port));
```

源主机用 C-Socket 建立一个与目的站点之间的 ATM SVC,SVC 建立所需要的信息包括目的主机的地址、应用进程端站点信息,以及前向和反向连接的传输参数规定。本地主机的信息通过将 D-Socket 作为一个参数传递来获得。

```
ctl_sock=Socket(AF_ATM,SOCK_QOS,0);
sendmsg (ctl_sock, (SETFLOWSPEC, data_sock,
    flow_spec,flow_spec_len));
sendmsg (ctl_sock, (SETUP, data_sock, (dest_
    ATM_address,dest_port)));
```

这里 sendmsg 调用用来传送控制信息的结构变量(4.4 BSD 的 msghdr).这些控制信息的传送也可以利用 write 或 send 函数来实现。

QOS 模块在 C-Socket 上为每个打开的连接维护一个连接状态信息,一旦连接建立,QOS 模块就注册一个相对应的连接句柄,并向应用进程发送一个状态消息,使应用进程获得该连接句柄。

对于点对多点通讯的情形,虚连接建立后可以增、减端站点,这只需向 C-Socket 写入相应的消息。

```
sendmsg (ctl_sock, (ADDPARTY, data_sock, (new_
    dest_address,new_dest_port)));
sendmsg(ctl_sock,(DROPPARTY,data_sock,(dest_ad-
    dress,dest_port)));
```

目的主机:创建一个 D-Socket,将它与一个本地端口号联编,然后创建一个 C-Socket,向该 D-Socket 发出"接收 ATM 呼叫建立请求"消息,并在 C-Socket 上等待呼叫建立请求。

```
listen_sock=socket(AF_ATM,SOCK_DGRAM,0);
bind(listen_sock,(local_ATM_address,local_port));
ctl_sock=socket(AF_ATM,SOCK_QOS,0);
sendmsg(ctl_sock,(SETRCVHANDLE,listen_sock));
```

当收到连接建立请求后,QOS 模块在 C-Socket 上向有关进程发送连接建立消息.应用进程可以用 accept 调用接受这个连接,这时创建了一个新的 D-Socket,accept 调用将使 QOS 模块产生一个相应的"ATM 呼叫建立请求接受"消息:data_sock=accept(listen_sock,&remote_endpoint);应用进程也可以通过 C-Socket 通知 QOS 模块:拒绝接受此请求,即 sendmsg(ctl_sock,ctl_sock,(REJECTSVC,data_sock))。

2.4 ATM 网络接口

ATM 网络接口设备与 ATM 硬件作用,并向上层用户提供一个数据链路层接口,负责维护 ATM 虚连接,提供对 ATM 信令的访问机制。

ATM 网络接口设备可以由一个或多个实体(核心用户或普通用户)打开,解复用就由 ATM 网络接口设备完成。RS/6000的 AIX 操作系统(BSD UNIX)还对核心用户提供一种 UPCALL 机制,为 ATM 设备驱动程序注册句柄,以进行诸如接收消息、状态更新、发送完成等功能。这个机制可以用来象支持传统协议一样支持 ATM 专用协议族,并且提供基于虚连接的解复用及针对特定连接的数据处理。在控制通路上,根据 BLLI 信息,传入的 ATM 呼叫建立消息被定向到合适的用户模块,连接一旦建立,用户模块就可以为 ATM 设备注册面向特定连接的句柄了。

2.5 IP 接口

IP 协议族通过 IF_ATM 网络接口层获得支持。网络接口层向用户提供无连接的数据链路服务,根据逻辑链路控制(LLC)信息对传入的数据包解复用。当下层是 ATM 设备时,该接口层将无连接的 LLC 服务转化为 ATM 虚连接,这种转换的标准之一是 RFC1577,不赘述。

2.6 ATM 专用协议栈接口

PF_ATM 协议族直接建立在 ATM 网络设备之上,所以无需经由一个网络层接口.QOS 模块打开一个与 ATM 设备的信令连接,为呼叫建立信息创建一个句柄.利用包含呼叫建立信息的 BHLI 中的应用进程端口号,把传入的呼叫建立信息定向至相应的应用进程.呼叫成功建立后,QOS 模块把刚获得的连接句柄以 DGRAM 插入相应的 D-Socket,但仍为状态函数保留该句柄,以能够接收连接控制信息.并通过 C-Socket 向应用程序发送连接控制信息。

对于 ATM 数据报连接,ATM 网络设备执行 AAL5的分段和重组功能。当一个数据分组要送出

时，DGRAM 模块把连接句柄和该数据分组一起传递，指示 ATM 设备通过某特定虚连接发送数据分组；当有数据分组传入时，ATM 网络设备根据 VC 号对分组进行解复用，并把它与代表该 VC 的连接句柄一起传给 DGRAM 模块，然后 DGRAM 模块根据连接句柄把分组放入某个应用程序的 Socket 缓冲区内。

三、对 ATM 专用协议栈的模拟

为了评价 ATM 专用协议栈的效率，我们按照新的协议栈结构模拟了一个视频会议系统。

3.1 系统配置

系统主要硬件组成是：IBM RS/6000工作站（64M 主存，33MHz 主频），DEC ATMworks 750网卡，MMT 适配器。ATM 卡完成 AAL5的功能，如分段和重组，它可支持多达2048条 ATM 虚电路，采用的是基于信用（credit-based）的流控算法。MMT 适配器支持30帧/秒速率的双工实时声/像压缩/解压，任一时刻可同时有32个解压流和1个压缩流。整个系统由一个专用的数字信号处理器（DSP）控制。操作系统是 AIX。

系统的通信结构包括：各会议成员间的控制通道，音频、视频和数据通道。这些连接没有严格的端对端延迟要求，但要求可靠的端对端传输，它们各自的 QOS 要求差别较大。应用程序通过传统的 TCP/IP 栈建立控制连接，通过扩展的协议栈配置音频和视频等连接。应用层实体 VAPE（Video-Audio Processing Entity）控制音频/视频设备和系统中的多媒体数据流。

3.2 对新协议结构的效率评价

如图2的粗线所示，首先我们采用 AAL5＋UDP/IP（PF_INET）的方案在各 VAPE 之间传输声像数据。在发送方，MMT 将音频、视频数据捕获、数字化，压缩，由 DSP 打包，向驱动程序发送一个中断，表示发送就绪，驱动程序再向 VAPE 发一个信号。VAPE 收到这个信号后，读入数据，然后通过 UDP/IP Socket 连接传送给对等的 VAPE。类似地，在接收方，VAPE 通过 UDP Socket 从网络接口读入数据，利用 MMT 驱动程序的写功能调用写入 MMT 的缓冲区。采用上述这种通讯方式，则每交换一个数据包在发送方和接收方都需两次系统调用（从 MMT 读、写入 Socket 和从 Socket 读、向 MMT 写）。

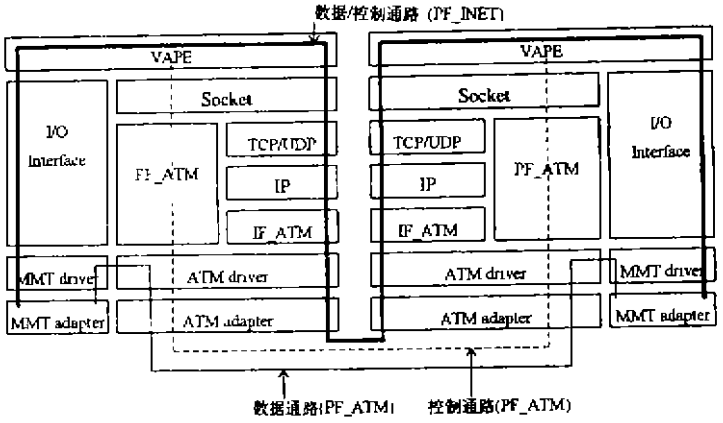

图 2 模拟的视频会议系统中控制通道和数据通道示意图

作为对比，我们又采用 PF_ATM 协议栈，将控制和数据通道分开，如图2中的虚线和细实线。VAPE 仍然控制数据传送，但不直接干涉数据的移动。数据通道直接将对等的 MMT 适配器相连，数据传送在内核中以一种近于自治器的方式进行。在接收方，从以该 MMT 为端点的 VC 主存中的多缓冲区链，拷贝到 MMT 上的双端口缓冲区。

建立数据通道的过程是，首先，VAPE 之间通过 Socket 接口建立数据连接，然后 VAPE 打开的 VC 的 ATM 句柄与 MMT 设备相联，从而把数据连接的端点重定向到了 MMT 适配器。上述过程显然需要对 MMT 驱动程序的扩充，故我们为 MMT 驱动程序增加了打开和关闭 ATM 连接的功能。

控制流和数据流的分离使得应用进程对数据传输是透明的，从而内核中数据的传输无需第一种方案中的四次系统功能调用，这不但消除了跨越异域边界的数据拷贝开销，而且减少了上下文转换开销。这种优化方案中数据通道的延迟仅仅是从 MMT 适配器到 ATM 接口这一段传递。不难发现，在发送方两种方案的时延差值比接收方来得大，即 PF_ATM 协议方案在发送方优势更明显，这是本系统发送和接收两端的不对称性造成的。在发送端，MMT 产生的数据直接拷贝到网络接口缓冲区中；而在接收端，送给 MMT 的数据报先要拷到系统主存中，然后才进入 MMT 的缓冲。这种不对称的设计是由于 MMT 的缓存不足，它有4MB 的发送缓存，却只有4KB 的接收缓存。

（下转第29页）