

一种基于分布式网络处理器系统的路由软件平台^{*}

吉 萌^{1,2} 余少华^{1,2}

(华中科技大学计算机学院 武汉 430074)¹ (武汉邮电科学研究院 武汉 430074)²

摘 要 由于网络处理器(NPU)结合了 ASIC 的高转发性能和传统 RISC 处理器的灵活性,目前的大容量网络设备多采用分布式 NPU 的架构。但因为各种 NPU 的体系结构和提供的 API 接口有较大的不同,设计者需要解决路由软件在各种硬件平台的可移植性和可扩展性。本文提出了一种基于分布式 NPU 系统的路由软件平台—EROS,该平台采用模块化和分层化的设计思想,通过数据转发适配层屏蔽了具体的硬件转发行为,简化了上层协议和网管的实现。本文介绍了 EROS 的总体架构和部分实现细节。此外,本文还介绍了 EROS 在实验平台 DEEP 上的实现和测试结果,基于 DEEP 平台的路由器已获得信息产业部高端路由器入网证。

关键词 网络处理器,分布式,数据转发适配层,软件转发引擎

EROS: Extensible Routing Software Platform for Distributed Network Processor Unit Based Systems

Ji Meng^{1,2} YU Shao-Hua^{1,2}

(School of Computer Science, Huazhong University of Sciences and Technologies, Wuhan 430074)¹

(Wuhan Research Institute of Posts and Telecommunications, Wuhan 430074)²

Abstract Since Network Processor Unit (NPU) combines the high performance of ASIC with the flexibility of a traditional RISC, researchers widely construct high-capacity network devices via distributed NPU architecture. But the different implementing approaches and versatile APIs NPU provides may cause tremendous troubles for system software designers. Moreover, there is a trend towards adding new set of services to modern IP routers at an increasingly rapid pace. Our response to this situation is to define and implement a hardware and Operation System independent, distributed and extensible routing software platform called EROS (Extensible Routing Operation System). EROS is a modularized and layered system. It comprises of five logical layers: System Management Layer, Control & Signal Layer, OS Adaptation Layer, Forwarding & Processing Layer and Data Forwarding Adaptation Layer. EROS innovatively utilizes Data Forwarding Adaptation Layer to hide concrete hardware forwarding details and thus can be used in different NPU systems. This paper presents the architecture design of EROS and introduces some of the implementing issues, especially the implementation of Data Forwarding Abstraction Layer. This paper also demonstrates how EROS is implemented on a 128Gbps NPU-based high performance router, which has been approved by China's Ministry of Information Industry. The experiments show EROS could deliver excellent performance without any impacts on the protocol conformability.

Keywords Network processor unit, Distributed, Data forwarding adaptation layer, Software forwarding engine

1 引言

互联网的发展使得对路由器的处理能力需求大幅度增加,目前的高性能路由器具有数百 G 比特甚至 T 比特的交换容量^[1,2]。因此,目前的高性能路由器多采用集中控制和分布转发结合的体系结构来实现对分组的高速转发。而网络处理器(NPU)由于在实现高速转发的同时,具有灵活的可编程性,因此成为高性能路由器的关键组件。

虽然 NPU 的应用部分解决了路由器性能和提供业务能力的瓶颈,研究者发现仍然面临着两大挑战:一、尽管标准化组织正在对 NPU 的接口和 API^[3,4]等进行标准化工作,目前的 NPU 仍然不是标准化的部件,不同的芯片厂商^[5,6]往往采用不同的体系结构和 API 接口,这样就导致需要基于针对不同的 NPU 开发不同的应用软件;二、高性能的路由器需要采

用分布式 NPU 结合大容量交换单元(Switch Fabric)的架构,这就需要软件系统能够应用分布式 NPU 的平台。因此,我们应该解决以下两个问题:1. 如何使路由器的软件架构可以适合于各种不同的 NPU 平台,能够将同一套软件平台可以用于不同的 NPU 硬件平台。2. 该软件平台应该如何设计才能满足多种架构的路由器系统能够同时集中式 NPU 转发或分布式 NPU 转发系统的需求。

目前国内外学术界对可扩展性的路由器软件平台有了一定的研究,这些研究包括了对路由器的抽象以及基于网络处理器上的路由软件。其中,比较有代表性的有麻省理工大学的 XORP^[7]、华盛顿大学的 Router Plugin^[8]、普林斯顿大学的 VERA^[9]和哥伦比亚大学的 Genesis^[10]。但是,它们具有一定的局限性:首先,这些系统大多是基于 Linux 或 BSD 源代码进行开发,软件架构的设计缺少模块化和分层化的考虑。其

^{*} 基金项目:国家高技术研究发展计划(863)2003AA121110、2003AA1Z1180,科技部创新基金 04C26224210719。吉 萌 博士研究生,主要研究领域为光城域网体系结构和高速网络设备设计。余少华 博士,教授,博士生导师,主要研究领域为光因特网和下一代网络体系结构。

次,这些的试验系统基于商用现成的(Commercial Off-The-Shelf)PC平台或网络处理器评估板进行开发,缺少在分布式NPU系统上的验证。第三,这些系统一般只支持单播和组播路由,而缺少对最新应用(如MPLS)的支持。

和学术界的路由器软件系统相比,商用的路由器的软件系统^[1],所支持的功能较为丰富,并且比较成熟,但是仍有一定的不足。这主要是由于这些软件系统同样多是来源于BSD操作系统,缺少模块化和层次化的设计。

本文提出了EROS(Extensible Routing Operation System)路由器软件平台,并介绍了该平台的研究和实现。EROS采用了分层化和模块化设计的思想,将整个软件平台在逻辑上划分为控制信令层、转发处理层、管理配置层、数据转发适配层和操作系统适配层五个层次。EROS的一大创新点在于通过数据转发适配层,用软件转发引擎完成了对路由器数据转发行为的抽象,并过高度抽象化的API接口完成了路由器控制平面和管理平面到数据平面的适配。本文介绍了EROS的总体架构和数据转发适配层的设计和实现。此外,文章介绍了EROS在一个具有128Gbps交换容量的分布式网络处理器平台的实现和实验结果,该平台已获得了信息产业部的人网证。

2 EROS的架构模型和实现

2.1 EROS架构模型

图1为EROS系统的总体结构模型图。EROS采用层次化的设计思想,从逻辑上看,可以划分为五层,即系统管理层(System Management Layer or SML)、操作系统适配层(OS Adaptation Layer or OSAL)、控制信令层(Control & Signal Layer or CSL)、数据转发适配层(Data Forwarding Adaptation Layer or DFAL)和转发处理层(Forwarding & Processing Layer or FPL)。

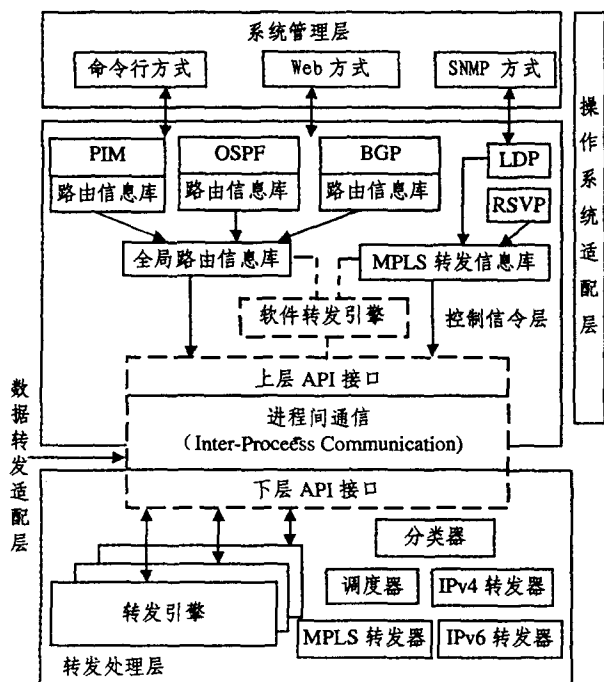


图1 EROS体系结构

2.2 操作系统适配层

OSAL主要是为了控制信令层和系统管理层提供统一的接口,将操作系统的部分功能和常用的函数调用重新进行封装,

这样就对上层应用屏蔽了操作系统的实现细节。此外,它还提供通用的系统调用(如单链表、多链表和哈希的管理),以及对系统内存和缓存区的管理。OSAL的功能由三个功能模块共同实现,即操作系统扩展模块(OS Extension Module)、系统资源模块(System Resources Module)和系统服务模块(System Services Module)。其中,操作系统扩展模块提供操作系统内核所必需的功能,主要包括进程的创建和删除功能、消息队列和信号量功能和基于事件机制的进程通信方式等;系统资源模块提供和系统相关的通用的API调用,包括定时器的管理功能、内存管理功能、缓存区管理和单链、双链和哈希表等功能;而系统服务模块主要提供和系统调试相关的功能,包括日志功能、异常事件处理功能和Trace功能。

2.3 控制信令层

CSL提供系统的路由协议、MPLS信令和业务功能,主要实现控制平面的功能。它一般由运行在一个或多个通用处理器上的软件来实现。CSL主要包括路由模块、MPLS信令模块、服务质量模块、安全策略模块和全局路由信息库。

由于EROS中多个路由协议进程共同访问同一个全局路由表,因此,全局路由信息库需要解决互斥操作问题。通常的实现方式采用锁住整个数据库或需要修改的路由条目所在子树的方法。这种方法在路由更新较慢的情况下问题不大,但不适用于路由更新频繁的网络核心节点。因此,EROS对路由表的更新采用了全新的原子更新方式。如图2,EROS在修改一条特定路由时,首先在空闲内存中创建需要更新的多叉子树的局部,然后再将原来树的指针指向新的多叉树即可。这样,EROS可以在某一个路由进程修改路由时,其它路由进程仍可以修改其它子树,转发进程也可以仍然根据子树内容进行转发。

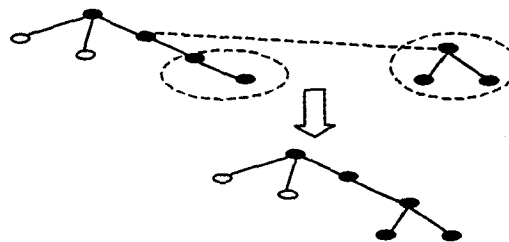


图2 EROS中的路由更新操作

2.4 系统管理层

SML完成对系统配置管理、性能管理、告警管理和安全管理。SML主要完成的功能包括逻辑实体到物理资源的映射、对管理信息库(MIB)数据的统计和分析、性能的查看、告警和日志的记录和用户安全的认证,管理方式主要包括命令行(CLI),简单网络管理协议(SNMP)方式和基于WEB的图形管理界面。

由于EROS是一个支持分布式多处理器的软件平台,SML必须实现分布式的管理才能实现对各个线路接口卡上的管理对象(Managed Object or MO)的管理。EROS采用了SNMP代理中继(SNMP Agent Relay or SAR)技术来实现对各个接口卡上的管理对象的管理。所谓SNMP代理中继,就是运行在主控处理器上的SNMP的主Agent首先判断需要访问的管理对象是否本地能够访问,如果不能访问则将请求(包括对象标识-OID和操作类型)中继到相应的接口卡,并由该卡上的从Agent将操作结果返回。其中,主Agent到从Agent的通信通道和API接口由数据转发适配层提供。

2.5 转发处理层

EROS 中,转发处理层主要实现数据平面的功能,完成基于每个数据包或流的处理,它可以由软件、ASIC 或网络处理器实现,可以采用分布式或集中式的处理方式。FPL 的实现方式决定了路由器的性能和业务能力。

当数据包进入到系统中时,FPL 首先根据数据包入接口的属性来决定应该对其采用何种分类规则(包括 QoS 和 ACL 规则)和转发规则(单播、组播、IPv4 或 IPv6)。然后,相应的分类模块首先完成对该数据包进行分类操作,并送转发模块处理。而转发模块来决定对该数据包查找操作类型是精确操作、最长匹配还是哈希查找,并修改相应包的内容。最后,调度模块将数据包放入到对应的输出队列中,并采用一定的调度算法将数据包发出。

2.6 数据转发适配层

数据转发适配层主要完成对转发平面行为(如网络处理器转发行为)的抽象,并完成控制平面和管理平面到数据平面的适配,对上层应用屏蔽底层硬件实现和体系结构的细节。数据转发适配层是 EROS 的核心部分,是实现软件平台的可扩展性和可移植性的关键。它主要由软件转发引擎(Software Forwarding Engine, SFE)、上层 API 接口(High-level API Interface, HAI)、底层 API 接口(Low-level API Interface, LAI)组成。但这三部分功能并不是在每个系统中都必需的。

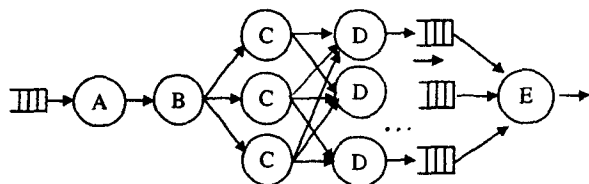
具体说来,FPAL 完成的功能如下内容:软件转发行为抽象、完成转发表(包括单播、组播和 MPLS)和分类规则的实时下载和更新;完成控制平面和数据平面的数据包交换;提供逻辑接口和物理接口的映射和分布;完成网管消息的收集和上报。关于 FPAL 的具体实现,将在本文的第 3 部分做详细介绍。

3 数据转发适配层的研究和实现

如上文所述,数据转发适配层是系统的核心处理部分,它由 SFE、HAI 和 LAI 组成,以下是对每部分研究和实现的具体说明。

3.1 软件转发引擎(SFE)

在目前的高性能路由器中,网络处理器主要完成的是数据平面的转发功能。而在 EROS 中,我们采用软件转发引擎对各种网络处理器的共性进行抽象,并对其转发功能进行模拟,以验证控制软件和网络处理器接口的功能的完整性和一致性。此外,当 EROS 用于基于软件转发的路由器时,SFE 还可以实现数据平面的转发功能。



A 二层接口单元 B IP 接口单元 C 分类单元 D 转发单元 E 调度排队单元
图 3 软件转发引擎数据流

文[9]中对路由器进行了抽象,但该抽象较为简单且不全面。EROS 中的 SFE 定义参考了文[12,13]中对转发单元模型的功能定义,它由以下逻辑单元组成:

二层接口单元(L2 Interface Element):完成二层协议(如 PPP 和 Ethernet)功能的终结。在接收端,完成将数据包从链

路层协议的解封和解复用,在发送端,完成对数据包在对应的链路层协议上的封装和复用。此外,本单元还完成对二层协议信息的统计功能(如 RMON)。

IP 接口单元(IP Interface Element):完成和 IP 层协议属性相关的功能,其中包括:对 IP MTU 的协商、对 IP 包头的完整性检验和操作(包括版本、TTL 和校验和计算)。IP 接口单元可以支持三种协议,IPv4、IPv6 和 MPLS。它同样需要完成对接口统计信息的采集功能。

分类单元(Classification Element):完成对数据包进行分类功能,以支持 QoS 和安全等应用。EROS 中的分类单元采用通过访问控制列表进行流分类的方式,采用基于可配置的多元组(如五元组(源 IP 地址、目的 IP 地址、源协议端口、目的协议端口、协议类型))进行灵活分类。在 SFE 中,有多个分类单元,它和具体的数据流是动态绑定的。

转发单元(Forwarding Element):根据控制平面下发的转发表项实现对数据包的转发处理,查找方式可能是最长匹配查找(Longest Prefix Match)、精确查找(Exact Match)和哈希查找(Hash Searching)。EROS 中的转发单元分为 IPv4 转发单元、IPv4 组播转发单元、IPv6 单播转发单元、IPv6 组播转发单元、MPLS 转发单元和流转发单元等六种类型。

如图 3,当数据包进入系统,SFE 首先对数据包完成链路层的解封和解复用操作,并对其进行 IP 包头的校验;然后,SFE 根据入接口的配置决定该数据包应该和哪一个分类单元和转发单元进行动态绑定,并送给相应的分类器进行处理;分类单元根据多元组规则完成对数据包的流分类处理后,如果该数据包根据规则需被丢弃,则丢弃该包,否则将之送往对应的转发单元进行处理;转发单元根据转发表完成查找和修改等操作,然后送往调度排队单元;调度排队根据控制平面下发的调度排队算法对数据包进行排队和调度,并将该数据包发往出端口。

3.2 上层 API 接口(HAI)

HAI 是 CSL 控制网络处理器数据转发的上层 API 接口。从对 SFE 的功能定义可以看出,网络处理器如果需要正常进行转发,需要从控制平面获取流分类规则、转发表目、调度排队算法的类型和接口相关的配置信息。因此,CSL 需要通过 HAI 这些信息发送到 DFL。

为了实现 SNMP Relay 功能,HAI 还需为主 SNMP Agent 提供网管信息和线路接口卡上的从 Agent 的通信接口 API。此外,作为一个网络设备,CSL 的路由信令等协议包还需要和其它网络上的路由器进行交换。因此,HAI 还应该提供 CSL 和 DFL 之间的发送和接收协议数据包(OSPF、LDP 和 ICMP 等)的 API 接口。

综上所述,HAI 共提供四种类型的 API 接口:1. 转发表项和流规则的更新接口;2. SNMP 代理中继的数据发送和接收接口;3. 协议数据包的发送和接收接口;4. 配置信息更新接口。每一部分具体说明如下。

转发表项和流规则更新接口(Forwarding entry & Flow rule Updating Interface, FFUI):FFUI 提供转发表和流规则的更新接口,主要包括以下内容:单播 IPv4 转发表项的添加、删除和修改;组播 IPv4 转发表项的添加、删除和修改;单播 IPv6 转发表项添加、删除和修改;组播 IPv6 的添加、删除和修改;访问控制列表的添加、删除和修改;流分类规则的添加、删除和修改;MPLS 标记映射表项(用于标记边缘路由器)的添加、删除和修改;MPLS 转发等效类到标记映射表项(用于

标记交换路由器)的添加、删除和修改。由于高性能路由器往往采用多个接口卡的方式,转发表目需要在主控卡和各个接口卡之间实现同步。因此,FFUI的表目更新接口包括表目的单播、组播和广播,且同时提供单条表目的更新和多条表目的批量更新(Bulk Update)。

SNMP 中继数据发送和接收接口 (SNMP relay Transmitting& Receiving Interface, STRI);STRI的功能较为简单,它提供主 SNMP 代理和从 SNMP 代理之间的数据通道,它包括发送和接收两个 API 接口。由于部分管理信息需要从多个接口卡获取后在主控卡汇总,STRI 同样需要提供单播、组播和广播的 API 接口。

协议数据包的发送和接收接口 (Protocol packet Transmitting& Receiving Interface, PTRI);PTRI 的功能和 STRI 相似,它包括 CSL 和 DFL 之间协议包的发送和接收的 API,它只需要提供单播 API 接口。

配置信息更新接口 (Configuration Information Update Interface, CIUI);CIUI 定义了对接口的配置信息和排队调度算法配置的更新 API 调用函数,它包括:队列调度算法的配置、逻辑接口的 IPv4 转发使能和禁止、逻辑接口的 IPv6 转发使能和禁止、逻辑接口的标记交换功能使能和禁止、逻辑接口标记边缘路由器国内功能使能或禁止和区分服务功能使能或禁止等。由于部分配置信息需要同步到所有接口,CIUI 需要支持单播和广播功能。

3.3 底层 API 接口(LAI)

LAI 是 CSL 和 SML 控制网络处理器数据转发的底层 API 接口。它的 API 接口种类和 HAI 相同,包括四种类型的 API 接口:1. 转发表项和流规则的更新接口;2. SNMP 中继的数据发送和接收接口;3. 协议数据包的发送和接收接口;4. 配置信息更新接口。区别在于,HAI 中的 API 由主控处理器上的上层应用进行调用,并将之发送到线路接口卡。而

LAI 的 API 是在线路接口卡上运行,调用转发引擎(如网络处理器)提供的底层函数将具体的表项或配置信息写入到转发引擎(如网络处理器)能够访问的空间。HAI 和 LAI 采用我们开发的进程间的通信(IPC)方式进行交换,该 IPC 支持单播和组播的可靠传输。

4 实验和结果分析

4.1 实验平台及实现

本文的实验平台采用我们开发的分布式多业务平台 DEEP (Distributed Extensible sErVICES Platform, DEEP)。DEEP 采用分布式网络处理器结合大容量的交换矩阵的体系结构,具有 128Gbps 的交换容量,可以作为高端的路由交换设备和宽带接入服务器。DEEP 最多支持十二个线路接口卡,可支持快速以太网、千兆以太网和 POS 等接口。图 4 是该设备的体系结构简图,DEEP 路由引擎、线路接口卡和交换矩阵组成,其中各个线路接口卡之间的数据通道通过大容量的 Crossbar 交换矩阵连接;而路由引擎和线路接口卡的控制通道通过管理总线连接。而线路接口卡采用双网络处理器架构,实现高速的分布式转发和数据平面的操作。目前,我们采用 Intel 公司的 IXP1200 网络处理器作转发处理,主要是考虑到该网络处理器具有灵活的可编程性,而且基于它的研究和应用已较为成熟,符合我们的要求。

EROS 的控制信令层和系统管理层软件运行在路由引擎上,包括路由协议和 MPLS 信令,以及 SNMP 主 Agent 和 CLI 网管软件;数据转发适配层的上层 API 接口和软件转发引擎也运行在路由引擎上;而 EROS 转发处理层的功能由 IXP1200 上的微码完成,目前我们实现了单播路由转发器、组播路由转发器、MPLS LER 转发器和 MPLS LSR 转发器的功能;底层的 API 接口和 SNMP 从 Agent 运行在 IXP1200 内嵌的 StrongArm 核上。

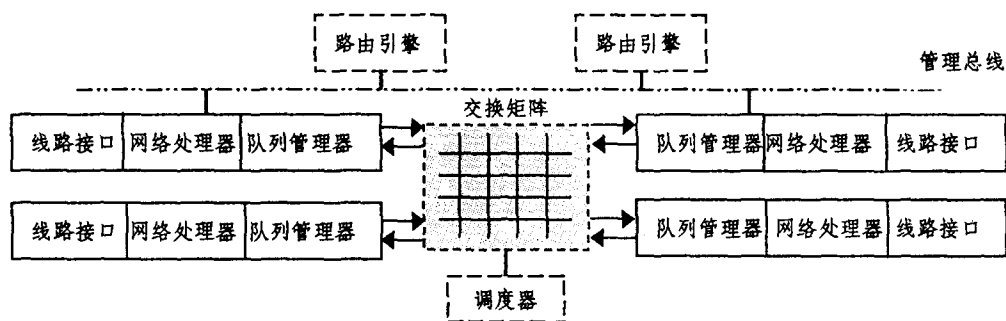


图 4 DEEP 体系结构图

4.2 实验结果

目前,我们在 EROS 中已实现了以下功能:RIPv1/v2c、MPLS、OSPF、BGP 和 PIM SM 协议,以及 SNMP 和 CLI 的网络管理功能。针对这些功能,我们主要测试了 EROS 的协议性能和协议一致性。在性能测试上,我们主要测试了大路由表(>256k)路由抖动下的转发性能。因为在路由抖动时,路由引擎会频繁地通过数据转发适配层向各个线卡上的网络处理器通告路由转发表目,其测试结果可以反映 EROS 的控制信令层和数据转发适配层的效率。

我们采用了安捷伦公司的 RouterTest 对 DEEP 平台上的 EROS 协议性能进行了测试,其中,IXP200 的 Fcore 频率为 200MHz。具体实验结果如表 1。

表 1 大路由表抖动下的转发性能

| 性能测试项 | 大容量路由表下的抖动性能 | |
|------------------|--------------|----------|
| | 转发性能 | 时延 |
| 300k 路由表下转发性能 | 79.5% | 167.33μs |
| 300k 路由表下 BGP 抖动 | 79.5% | 167.36μs |
| 300k 路由表下 IGP 抖动 | 79.5% | 167.38μs |

注 1:测试线路接口卡为双端口千兆卡

注 2:测试包长为 64+78+228+740+1508 的混合流量,路由更新速率为每秒钟 10000 条

我们对 DEEP 平台的协议一致性也进行了测试,测试仪采用安捷伦公司的 QARobot 和 MIB Browser,测试例是基于《路由器设备技术规范-高端路由器》和 QARobot 上的标准测试例,具体结果如表 2。

表 2 EROS 协议一致性测试结果

| 一致性测试 | 测试项 | 通过项 | 不支持项 |
|------------------------------|-------|-------|---------------------|
| BGP4 路由协议测试 | 73 项 | 73 项 | 0 项 |
| RIPv1 路由协议的测试 | 9 项 | 9 项 | 0 项 |
| RIPv2 路由协议的测试 | 10 项 | 10 项 | 0 项 |
| PIM-SM 组播路由协议测试 | 20 项 | 20 项 | 0 项 |
| OSPFv2 路由协议的测试 | 59 项 | 59 项 | 0 项 |
| MPLS LDP 协议测试 ^[1] | 190 项 | 231 项 | 16 项 ^[2] |
| SNMP 网络管理 | 12 项 | 12 项 | 0 项 |

注 1: LDP 测试采用模式十二

注 2: LDP 不支持项均为 ATM LSR 测试项

4.3 结果分析

从表 1 可以看出, DEEP 平台的千兆线卡在大路由表抖动时时延保持基本不变, 性能稳定, 这说明 EROS 平台数据转发适配层有良好的性能, 在重载(300k 路由, 每秒 10000 表项更新)时做到了控制平面和转发平面的信息实时同步。其中, 转发表项和流规则接口能够正常的完成对 IPv4 转发表项的编码和解码。双端口千兆性能未达到线速并非因为表项未能及时同步, 而是因为 IXP1200 的处理能力瓶颈为 2.4MPPS^[5], 而双端口千兆在 64byte 包长时线速所需要的性能为 2.98MPPS。此外, 我们在实现中采用的 IXP1200 的 Fcore 的工作频率为 200MHz。如果我们采用工作频率为 232MHz 的 IXP1200, 双端口千兆的转发性能理论上将提高 15%, 可以在混合包长情况下达到 95% 以上。

而从协议一致性的测试结果可以看出, EROS 平台在分布式网络处理器平台上有效地实现了协议的一致性, 已成功地实现了 SNMP、单播路由、组播路由和 MPLS 功能。整个系统虽然由多个线路接口卡和路由引擎组成, 但仍然保证了逻辑上的完整性。这说明数据转发适配层中的协议数据包的发送和接收接口工作正常, 能够实现协议包在路由引擎和线路接口之间的交互, SNMP 中继代理可以通过 API 接口正常访问线卡上的 MIB 变量。而 MPLS LDP 测试例中有十六项不支持是因为 DEEP 平台不支持 ATM 接口, 所以与其相关的测试项无法进行测试。

结论 本文提出并完整地论述了适用于分布式网络处理器系统的 EROS 路由软件平台, 它通过数据转发抽象层屏蔽了具体的硬件转发实现, 可以适用于各种网络处理器的硬件平台。它有效地解决了以往硬件平台的改变导致软件大幅改动的难题。同时, EROS 采用模块化和分层化的设计思想, 可以适合于各种不同的操作系统平台, 满足了可移植性和可扩展性的要求。

在 DEEP 上的测试结果表明 EROS 可以有效地应用于分布式网络处理器系统, 并在满足协议一致性的同时具有良

好的协议性能, 具有较大的实用价值。基于 DEEP 平台的路由器系统已获得信息产业部的高端路由器网证。实验结果表明由于受到网络处理器性能的限制, 在大路由表抖动下, 双千兆端口的性能无法达到线速。因此, 采用更高速率(10Gbps)的网络处理器和进一步增加应用功能(MPLS VPN)将是我们下一阶段的深入研究课题。

参考文献

- 1 Cisco Carrier Routing System Technical Paper. Cisco Systems. June 2004. <http://www.cisco.com>
- 2 Juniper T-Series Routing Platform Datasheet. Juniper Networks. 2003. <http://www.juniper.net>
- 3 IP Unicast Forwarding Services API Implementation Agreement. Network Processor Forum, June 2004. <http://www.npforum.com>
- 4 Stream Interface Implementation Agreement. Network Processor Forum, Oct. 2002. <http://www.npforum.com>
- 5 IXP1200 Network Processor Datasheet. Intel Corporation. Sept. 2000. <http://www.intel.com>
- 6 IBM PowerNP NP4G3 Network Processor Solution Product Overview, IBM Microelectronics Division, April 2001. <http://www.ibm.com>
- 7 Handley M, Hodson O, Kohler E. XORP: An Open Platform for Network Research. First Workshop on Hot Topics in Networks, Princeton, New Jersey, 2002
- 8 Decasper D, Dittia Z, Parulkar G, Plattner B. Router Plugins: A Software Architecture for Next Generation Routers. IEEE/ACM Transactions on Networks, Feb. 2000, 8(1): 2~15
- 9 Karlin S, Peterson L. VERA: An Extensible Router Architecture. In: Proc. of the 4th Intl. Conf. on Open Architectures and Network Programming (OPENARCH), April 2001. 3~14
- 10 Kounavis M E, Campell A T, Chou S, Modoux F, Vicente J, Zhuang H. The Genesis Kernel: A Programming System for Spawning Network Architectures. IEEE Journal on Selected Areas in Communications, March 2001, 19(3): 511~526
- 11 Bollapragada V, Murphy C, White R. Inside Cisco IOS Software Architecture. Indianapolis, IN 46290 USA. Cisco Press. 2000
- 12 Khosravi H, Anderson T. Requirements for Separation of IP Control and Forwarding. RFC3654, Nov. 2003
- 13 Yang L, Haipern J, Gopal R. ForCES Forwarding Element Model. IETF draft. Feb. 2004
- 14 Yang L, Haipern J, Anderson T, Gopal R. Forwarding and Control Element Separation (ForCES) Framework. IETF draft, Jan. 2004
- 15 Rekhter Y, Li T. A Border Gateway Protocol 4 (BGP-4). RFC1654. July 1994