LOARD:分层的开放分布式路由器结构

余 鑫[1] 黄本雄[1] 阮加勇[2]

(华中科技大学电子与信息工程系 武汉 430074)[1] (福建实达网络科技有限公司 福州 350002)[2]

摘 要 现有的路由器体系结构研究中,很大部分是从纯软件角度展开,忽略了目前 ASIC 和 NP 广泛应用的事实。传统的按 USER 和 KERNEL 来划分的方式,已经不能准确地描述真实分布式路由器环境的实际需求。本文针对下一代路由器结构的分布式通信需求和扩展性需求,提出了具备分布式通信机制和平台无关特性的 LOARD 结构。该结构采用虚拟接口、虚拟通道,为分布式路由器的各模块提供了统一的抽象平台,不仅可以适应控制模块的软硬件环境,也可以适应转发模块的软硬件环境需求,对集中式路由器和分布式路由器都具有很好的扩展性和开放性,适合下一代路由器的发展需求。

关键词 分布式,路由器,体系结构,LOARD

LOARD: Layered Open Architecture for Router Distributed

YU Xin[1] HUANG Ben-Xiong[1] RUAN Jia-Yong[2]

(Dept. of Electronics and Information Engineering, Huanzhong University of Science and Technology, Wuhan 430074)[1]

(Fujian Start Network Technology Co., Ltd., Fuzhou 350002)[2]

Abstract Most researches about next generation router focus on the extensible architecture is basing on software. But more and more implements use ASIC and network processor as their data plane solution. The way that separates router's system to USER and KERNEL cannot handle all the problems may occur in distributed environment. This paper proposes an architecture that is platform independent to serve the requirements of distributed communication and extensibility. This architecture called LOARD uses virtual path and virtual interface to provide a transparent platform for both control plane and data plane. It is suitable for next generation routers, which are distributed, paralleling, wire speed, open and extensible.

Keywords Distributed, Router, Architecture, LOARD

1 引言

高速路由器的发展与研究经历了从单处理器共享总线结构、多处理器共享总线结构、多处理器交换结构、到分布式多处理器交换结构的演进过程[3]。纵观其发展趋势,是从低速到高速、从共享总线到并行交换、从集中到分布、从针对性到可扩展。在对路由器结构的整合过程中逐渐形成的并行分布式多处理结构,能满足现有的以及可能出现的各种需求,是下一代高速路由器,包括全光路由器的必然选择。

目前路由器研究的热点内容仍然集中在数据层面,仍然致力于提高数据转发能力和 QoS 能力,对数据层的可扩展性的研究也是目前的一个热点[4~8]。Click[4,5],Router Plugin[6],Scout[17]等均提出了数据转发层的灵活、可扩展的结构,但对控制层和管理,以及分布式环境下的数据层的讨论相对较少,而系统的扩展性、开放性等关键性能的实施也要依赖于控制层。高速路由器的发展趋势是控制管理层的功能仍由通用系统上的软件实现,而数据层的转发功能则由专用的硬件(ASIC),或者是专用的处理器(NP)上的软件来实现。本文立足于分布式环境下路由器的控制层,以及控制层和数据层间的接口,研究适合于下一代路由器分布式处理的软件体系结构,提出了分层结构的分布式开放路由器软件平台的设计思想— LOARD(Layered Open Architecture for Router Distributed)。LOARD 是一套抽象封装,是一套开放 API,也是一种分布式路由器体系结构。

2 路由器体系结构

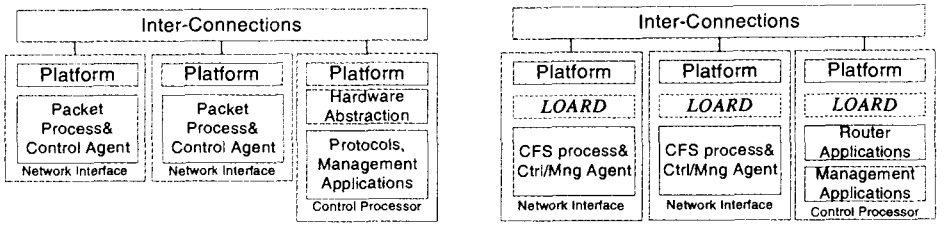
2.1 逻辑结构

路由器在逻辑上一般分为控制层(control plane)和数据层(data plane)两个层面。随着高速路由器的发展和设备的管理性要求的逐步提高,独立的管理层(manage plane)也被单独抽象出来,形成了管理、控制、转发相分离的三层结构。

下一代高速分布式路由器通常包含接口(转发)模块、控制模块、互连模块等基本组件,构成如图 1a 所示的并行多处理分布式结构。数据层(报文处理与转发)被分布到各个接口模块上,但控制层(路由协议、数据表维护等)和管理仍然集中在主控制模块上。在接口模块上实现报文处理,如分类、路由查找、排队、调度、传发等功能,部分 IP 层以下的协议也在接口模块实现,如 ARP、ICMP 等。同时,接口模块还包含控制代理,执行来自控制模块的转发表操作、QoS 参数维护、策略维护等控制命令,以及管理代理,执行来自管理系统的配置、告警、安全、计费等管理命令。在控制模块上实现路由器必须支持的协议,如路由协议(OSPF, RIP, BGP 等)、标记发布协议(LDP)、服务质量控制(CR-LDP, RSVP-TE 等)、管理协议(SNMP 等)、IP 及应用协议(TCP/IP, BOOTP, DHCP 等)[1],

*)国家"863"计划基金资助项目(2002AA001009)。余 鑫 博士生,主要研究方向网络结构与路由结构。黄本雄 教授,博士生导师,主要研究方向下一代网络、通信软件、实时信息处理、网络安全。

以及并行地控制各个分布接口卡的数据表同步与更新。路由器设备的管理维护系统的主控部分也在控制模块上实现。互联模块实现控制模块与接口模块间控制通道，以及接口模块间的高速数据通道等。目前高速路由实现方案几乎无例外的选择了这种结构[3,9~11]。



a) Generic Architecture　　b) Router Distribution with LOARD

图 1　分布式路由器传统结构与 LOARD 结构

本文描述的 LOARD 结构，提出了一套完整的结构和抽象，支持分布式路由器结构中各个主要模块的扩展性、开放性、兼容性、可靠性。它在分布式路由器的各个模块的软件系统中添加了一层抽象层(如图 1b)，对真实的物理环境进行抽象，无论控制层的协议软件、管理软件，还是数据层的控制管理代理软件、转发软件，都可以在统一的抽象平台 LOARD 上运行，使用 LOARD 接口和结构的软件或系统将具备很好的开放性和扩展性。

2.2　LOARD 系统平台结构

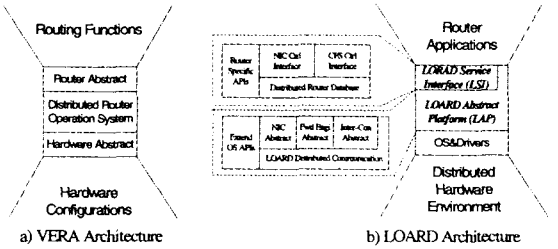
虽然目前路由器中数据层面的高速转发功能已经发展为由硬件来实现，但无论是专用的 ASIC 还是具备部分通用性的 NP，其处理能力的主要优势都在报文分类、路由查找、排队调度等方面。路由器其它部分功能如管理、路由协议等完全可以在低速、通用的 CPU 中运行，事实上这些核心部件通常仍然是由控制模块中通用 CPU 和软件系统来实现的。同时，在分布式环境下控制模块(主模块)与转发模块(分布子模块)间的交互也需要由软件实现，于是目前转发模块中广泛使用的 NP 中除了报文处理引擎外，通常还集成了一个通用微处理器核[13~15]，用以实现上述功能。处理核不同、操作系统的不同，都会导致数据层部分控制功能实现上的变更。为使路由器具备完整意义的扩展性和开放性，分布式环境中的开放软件平台不仅应在控制管理层进行平台抽象，数据层也应有一定程度的平台抽象。

VERA[7]提出了一个完全抽象的虚拟路由器结构，试图使所有软件感知不到具体操作系统和硬件环境。但事实上许多控制和管理功能，如同步、备份等是硬件结构密切相关的。一个完整的路由器抽象，应该分别提供基本的报文处理(路由)抽象、控制结构抽象和管理结构抽象。也就是说，除了将转发子系统进行抽象外，还需要向控制系统和管理系统提供整个路由器平台的抽象。例如，控制系统的各个协议可以不知道路由器的具体物理结构，如网络接口的物理分布，因此可以在路由器整机的级别上抽象，而管理系统必须了解分布系统的物理结构，因此只能在分布式平台的级别上抽象。

基于上述考虑，LOARD 在整体上采用分层的模型，以支持数据和控制管理的不同抽象，以及分布通信机制；同时在 LOARD 的内部分为两层，分别提供路由器级与平台级两个级别的抽象，以支持控制和管理需求。

LOARD 将路由器体系分为如图 2b 所示的四个层面：应用层、路由服务层、抽象平台层、分布式平台环境层。

- 路由应用层(RAL)：所有协议软件、管理软件、路由信息表都位于系统的应用层。
- 服务接口层(LSI)：即路由器抽象层，为应用层提供抽象的路由器环境，以及通信机制、接口机制等的屏蔽，提供统一的路由服务 API。
- 抽象平台层(LAP)：屏蔽真实的系统平台环境，包括操作系统、硬件驱动、模块间的物理连接方式，由此来支持整个平台环境的兼容性和移植性。
- 分布式平台环境(DPE)：包括具体的操作系统、内部硬件连接驱动、网络接口驱动、分布式硬件环境的物理结构等。



a) VERA Architecture　　b) LOARD Architecture

图 2　VERA 结构与 LOARD 结构

3　LOARD 核心组件

3.1　路由器抽象与系统服务—LSI

路由器系统中，起核心控制作用的始终是软件，包括管理、路由协议、路由表维护等工作都只能由软件来完成。也就是说，除了数据层的大部分工作已转移到硬件完成外，其它两层的工作仍停留在软件，而这些软件必然要依赖于一个软件平台或操作系统。通用操作系统无法满足路由器环境下的新需求，专用操作系统又缺乏足够的开放性。因此，将针对路由器的特殊平台需求抽象出来，并与操作系统相独立，是一个开放式路由器系统平台的必然选择。

LOARD 的分层结构中，针对控制管理层的需求划分出专门的路由器服务层 LSI，为控制管理软件提供一个开放并统一的服务接口。针对数据层的需求划分出平台抽象层 LAP，为数据层的转发功能提供物理接口抽象，为控制管理代理提供与主控制类似的运行平台和通信机制。关于 LAP 的平台抽象和分布通信机制将在后面的章节介绍。LSI 的结构如图 2b 所示，主要包括网络接口(NIC)控制接口、分类转发调度(CFS)控制接口、路由器相关 API、分布式数据库。

NIC 控制接口提供对网络接口的设置接口，如设置 IP 地址、获取接口状态等。CFS 控制接口提供对数据层中分类(Classifier)、转发(Forwarder)、调度(Scheduler)等主要组件的控制接口，如更新调度算法、排队方式等。路由器相关 API 则提供在路由软件中最广泛使用的各种算法、数据结构组织、公共库等的调用接口，包括 Radix Tree、Link List、Hash Ta-

ble 操作，VTY 或 CLI 管理界面等。提供这种级别的抽象可以方便地使用不同的转发模块，如 Click 或 Router Plugin 等。也可以采用不同的路由协议软件和库，如 ZebOS、FutureSoft、Zebra 等。

在高速分布式路由器中，各个转发模块不再共享同一个转发表。一种作法是在主控制模块保留一套完整的转发表，而在各转发模块中仅缓存(Cache)转发表中最常用的一部分。LOARD 中分布数据库抽象的提出是为了向控制管理层提供透明的数据库的访问控制接口，如转发表、标记表、策略表等需要提交转发模块使用的关键数据表，并在该抽象的内部完成数据表的同步、更新等功能。这样可以降低系统硬件成本，但维护缓存的工作较复杂，尤其在核心路由器中缓存的更新频率相当高，更新速度容易成为瓶颈而导致效率的降低。随着内存速度的提高和价格的降低，以及更高速、高效处理的要求，各转发模块中也保存完整的转发表，以降低维护缓存的开销[10]。

3.2 开放平台抽象—LAP

路由器系统非常复杂，需要支持的协议、接口、功能都数量巨大，软硬件系统类型众多。在开发路由器的过程中，尤其是分布式路由器中，要进行大量的软硬件的整合工作。如果存在一个开发的系统平台和接口规范，各种符合规范的不同厂商的软硬件产品可以方便的整合，并具备移植性、扩展性、开放性。开放平台的主要手段是虚拟设备、以及操作系统扩展，为实现平台的开放性，进行一定的平台抽象是必须的，这样使得系统软件能够平台无关的适应不同的硬件环境。

在分布式路由器中，数据层和控制管理层可能处在不同的硬件与软件平台上。LAP 可以同时支持这两种环境的平台抽象，如图 3 所示，主要包括操作系统扩展，网络接口抽象，转发引擎抽象，内部互连抽象及分布通信。

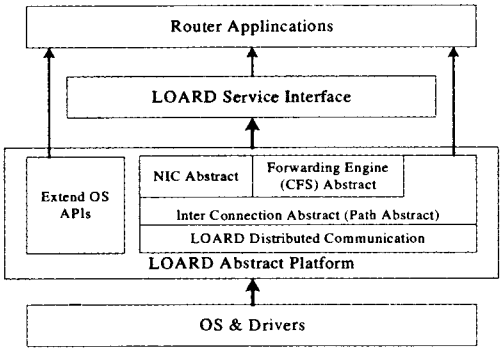


图 3 抽象平台 LAP 结构

在分布式环境下，无论数据层和控制管理层，都不再是一个单一的软件或模块，而是一个物理上分布的系统。真实的网络接口和转发引擎存在于子卡上，而控制管理系统的核心却存在于主卡上，LAP 除了对数据层和控制管理层提供不同的平台外，主卡和子卡上的平台抽象也有区别。LOARD 采用虚拟设备的方式来将数据层和控制管理层结合，并实现系统平台的抽象。网络接口抽象(NIC Abstract)将路由器中所有的网络接口信息收集，并在控制模块和每个接口模块中抽象后提供出虚拟网络接口驱动 VNID(Virtual Network Interface Driver)，这样每个物理模块都面对一个完整的虚拟路由器平台，利于控制层协议软件和数据层转发软件的开放和扩展。转发引擎抽象(Forwarding Engine Abstract)为每一个控制实体(无论是控制模块中的主控制实体，或者接口模块中的控制代理)提供一个虚拟的供控制的转发引擎设备 VFED(Virtual Forward Engine Driver)，所有转发行为控制的命令均通过统一的虚拟转发引擎设备驱动实施。内部互连抽象(Inter Connection Abstract)提供一个虚拟的内部连接设备 VICD(Virtual Inter Connection Driver)，所有软件模块均可以。LAP 将分布式环境完全隔离起来，使路由器中除管理外的每个逻辑实体(协议、分类、排队、调度等)均在一个虚拟的集中式路由器中运行。因此是平台具备非常优秀的透明性，各个系统也具备很好的开放性和扩展性。

3.3 分布式通信机制

高速核心路由器采用的分布式结构引起了通信机制的讨论，主控制模块需要与各个接口模块通信，将转发表和控制信息通知接口模块，同时还要通过接口模块发送路由协议报文，此外各个接口模块之间也要传递要转发的数据报文。可以通过扩展操作系统来实现分布式通信的 DMQ[16] 通信机制，这样一方面不能适应软件和硬件环境的改变，并不具备很好的开放性和灵活性；另一方面增加了操作系统的复杂性，对稳定性也有影响；同时，该通信方式不能满足接口卡之间高速转发数据的需求。实际在分布的多处理路由器环境中，有多种不同性质的信息需要在不同的板卡或模块间传递。完整的通信机制应该能透明的覆盖各种不同性质的信息传递需求，针对不同的通信需求提供不同的通信服务。

分布式路由器环境中的三类主要通信需求包括：FDP(Fast Data Path)、SDP(Slow Data Path)和 MCP(Manage Control Path)。为满足路由器内外的通信需求，同时也支持平台无关特性，LOARD 中的 LAP 针对将分布式路由器环境下的通信需求，将通信连接抽象为一种虚拟的设备 Path(通道)，并进一步将路由器实际的三种通信需求：流量通道、报文通道和控制通道，抽象为快速数据通道 FDP、慢速数据通道 SDP 和控制管理通道 MCP。FDP 用于高速的数据流量通道，在各个网络接口模块间传递报文。该通道要求带宽大、延时低、无阻塞等，实现时可以采用 cross bar、switch fabric 等。SDP 用于相对低速与实时性不强的路由协议、管理协议等本地数据报文通道，在控制管理模块与网络接口模块间传递报文，要求稳定、无丢包。MCP 用于速度更低，但稳定性要求最高的路由器内部的管理维护，以及分布数据库的维护管理，在控制管理模块与所有其它模块间传递报文。实现中 SDP 可以使用专用物理连接如 PCI，也可以使用 FDP 中建立的隧道。MCP 相当于系统的“动脉”，稳定性要求最高，应采用独立的专用连接，如 PCI 或 Ethernet 等。

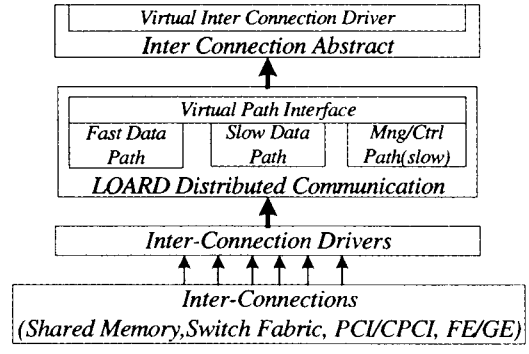


图 4 分布通信 LDC 结构

LOARD分布式通信机制(LDC)的具体结构如图4所示,它是LOARD中最为重要的部件,是整个分布式环境的基础。路由器服务接口LSI需要利用它进行数据库同步、协议报文收发等。LAP更是需要基于它实现网络接口抽象、转发引擎抽象等。应用软件,尤其是管理软件也需要直接利用它进行分布式通信和管理。由于此通信机制中将通道抽象以后提供给上层使用,具体物理连接设备被抽象并封装为虚拟互连设备驱动VICD。LAP将实际的Switch Fabric、PCI、FE等物理连接,抽象为FDP、SDP和MCP这三种虚拟通道,由通道内部维持,为上层软件提供虚拟通道接口VPI(Virtual Path Interface)调用传输功能,物理连接设备和驱动的变更并不影响上层软件,因此除应用软件,LOARD中的其它组件也都具备平台无关特性,这样的分布式平台具备了更广泛、更灵活的扩展性和开放性。

4 原型系统

由于LINUX操作系统的开放性和安全性,以及其强大的网络处理能力,目前在路由器领域中已经越来越趋于使用LINUX,此外的LINUX平台上进行路由器软件的开发更具可移植性和扩展性,因此我们在LINUX平台上进行原型系统的设计和实现,使得LAP的使用充分展现出来。在开发过程中,所采用的LINUX内核版本为2.4.18。

我们基于Linux平台在以太网上实现了提供三种通道的原型系统,转发软件通过FDP的VICD调用高速流量通道,虚拟网络接口VNID通过SDP的VICD实现,控制管理软件通过MCP的VICD调用控制管理通道。

原型系统物理结构如图5a所示,系统由三台PC组成,配置为赛扬1.7G+128M+RTL8139集成网卡,操作系统为Redhat Linux,内核版本为2.4.18。集成网卡用作"路由器"内部互连设备,连接到交换机上。NIC PC另外各添加了一块PCI网卡作为网络接口,分别连接到另两台测试用机上。于是,系统表现为一台具备两个以太网接口的"路由器",如图5b所示。

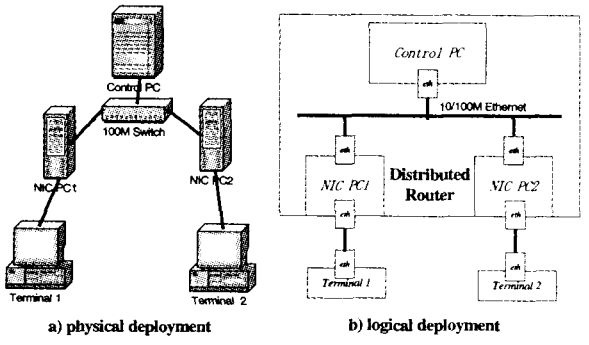


图5 原型系统的物理结构和逻辑结构

具体的代码结构如图6所示,添加了VPI模块实现内部互连通道,添加了VICD模块虚拟网络接口,这两个模块一起实现了对分布环境的屏蔽与抽象。LINUX系统原有的转发模块在本系统中暂时没有改动,这样一方面表明LOARD结构的兼容性与抽象的特点,另一方面表明可以更换转发模块成click等可扩展的转发模块,以达到路由器在数据处理方面的扩展性。

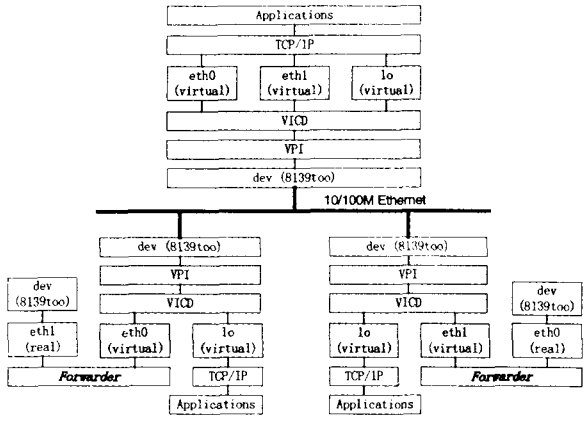


图6 代码结构

本原型系统的实现中,转发、TCP/IP协议和路由协议均使用linux,具备很好的开发性、兼容性和继承性。系统能正常的进行数据转发,数据包从NIC PC1的eth1进入,从NIC PC2的eth0出。64byte的数据报文转发率的17.4%,与单独PC转发时的18.6%相比,性能虽然有所下降,但是可以在提高转发模块的处理能力上得到弥补。同时,采用LOARD构建路由器所获得的扩展性和灵活性,以及对分布环境的支持却是现有系统无法满足的。

结束语 由于路由器厂家众多,其软件平台和软件结构往往作为其核心技术,没有像硬件一样很好地开放和标准化。诸如网络处理器NP,交换矩阵Switch Fabric,网络接口Line Interface等硬件技术都有相应的标准化组织或企业联盟从事标准化工作,因此在硬件平台的构建中往往相对较易。相比之下,路由器软件则要混乱得多,路由协议、操作系统、硬件平台驱动、管理功能的整合要花费巨大的人力,而分布环境却得不到支持。因此,作为路由器发展的一个重要因素,软件平台的抽象化、软件结构的规范化、分布式结构的支持是将来路由器软件发展的一个重要课题。

本文针对下一代高速路由器的发展趋势,从体系结构和分布式平台方面提出了新的分层模型,重点描述了分布式路由器软件平台的结构,提出了路由器服务层的功能模型和平台抽象的两级抽象思想。并采用LINUX系统及接口规范实现了一套原型系统,该系统使路由器物理平台对应用软件系统透明,可以满足分布式环境下的数据层和控制管理层软件的需求。由于LOARD采用两级抽象,并且数据层和控制管理层都支持,不仅具备了click、router plugin等软件路由器所具备的转发层扩展性,也弥补了对分布式环境下运行支持的不足。

参考文献

1 Baker F, Ed. Requirements for IP Version 4 Routers. RFC 1812, Internet Engineering Task Force, June 1995. ftp://ftp.ietf.org/rfc/rfc1812.txt

2 Newman P, Minshall G, Lyon T, et al. IP Switching and Gigabit Routers. IEEE Communications Magazine, 1997, 35: 64~69

3 Aweya J. IP Router Architectures: An Overview. Journal of System Architecture, 2000, 46: 483~551

4 Morris R, Kohler E, Jannotti J, et al. The Click Modular Router. In: Proc. of the 17th ACM Symposium on Operating Systems Principles, December 1999. 217~231

(下转第43页)

当MT端发出数据请求时，如果在本地缓冲区缓存命中，就不用将请求转发给VS，节省了无线带宽；如果未在MT端缓存命中的请求在VS端发生缓存命中，就不用将请求转发给数据库服务器，节省了有线带宽。只有在MT和VS端都没有缓存命中的请求，才通过无线(MT到VS)和有线(VS到服务器)两种信道传送到源服务器。假定在 t 时间内，客户端处于READY状态的MT总数为 M，所有MT端缓存的平均命中率为 H_1，MT对数据项的请求是随机的，其请求到达速率为 λ，VS端缓存的命中率为 H_2，数据项的大小为 B 字节，用 BW_1、BW_2 分别表示本文提出的框架系统节省的无线和有线带宽，则：

$$BW_1 = M * (1 - e^{-\lambda}) * H_1 * B \quad (1)$$

$$BW_2 = BW1 + M * (1 - e^{-\lambda}) * (1 - H_1) * H_2 * B \quad (2)$$

从式(1)、(2)可以看出，MT缓存的数据项个数越多，命中率 H_1 越大，节省的无线和有线带宽也越多；H_2 的大小决定于MT访问数据项的局域性，其值越大，节省的有线带宽就越多。

4.2 维护数据一致性

目前维护无线移动环境下缓存数据一致性主要有同步的IR[4~6]和异步的AS[7]，在不考虑数据源服务器与MSS或VS交换数据更新信息的开销的情况下，VS与IR、AS策略定性比较如表2所示。此外，MT的漫游也是移动环境下维护缓存一致性需要考虑的一个重要因素，IR、AS都是假定所有节点间数据的完全复制来支持MT的移动性，即IR策略中所有MSS广播相同的IR，AS策略中在MSS间复制相同的HLC，而VS自动支持MT在一个公众陆地移动通信网PLMN中的漫游(见图1)，不需要另外的开销。

表2 IR、AS、VS参数对比

比较的参数	IR	AS	VS
MT断开连接的最长时间	W*L	任意	任意
上行请求次数*	0	1	1
访问已缓存数据的延迟	0 to L	0	0
服务器端是否传送未缓存更新数据的 *ID*	Yes	No	No
是否向未进入网络的MT传送更新数据 *ID*	—	Yes	No
是否形成突发的网络流量	Yes	No	No
MT获取已更新数据 *ID* 的计算量	大	小	小
参与维护数据一致性的节点数	多个	多个	1个
MT端缓存一个数据项需要的位数(不包括数据本身)**	64	67	33

*——AS在断开后再次连接后有一次上行传输；VS在已缓存的对象被移出时有一次。

**——假定ID，TS各占4个字节，标记位占1个Bit位

结束语 从微处理器芯片到操作系统，从Web访问到分布式文件系统，缓存技术一直用于降低访问延迟、节省带宽和提高系统的整体性能。本文针对当前运行的GPRS网络和即将来临的3G网络，提出了客户端和验证服务器VS端两层缓存系统框架，有效地降低了大无线移动用户查询数据库环境对无线和有线带宽的占用，并用异方式解决无线环境下缓存一致性问题，有效地减少了参与维护缓存一致性的节点数和MT端的计算量，克服了基于IR报告策略的所有缺点，并通过(1)VS只传送MT已缓存且已更新的数据项的；(2)VS只向已连接进入网络(在线)的MT传送信息；(3)取消时间戳降低传送的信息量；(4)利用用户访问数据的局域性面向组内用户进行广播四种方式来减少异步传输的次数，具有较强的实用性。

参考文献

1 Barbara D. Mobile Computing and Databasee——A Survey. IEEE Transactions on Knowledge And Data Engineering, 1999, 11(1):108~117

2 Wang Z, Das S, Kumar M. SACCS: Scalable Asynchronous Cache Consistency Scheme for Mobile Environments. In: Proc. 23rd Intl. Conf. on Distributed Computing Systems Workshops, Providence, RI, USA, May 2003

3 Lin Y B, Lai W R, Chen J J. Effects of Cache Mechanism on Wireless Data Access. IEEE Transactions on Wireless Communications, 2003,2(6):1247~1257

4 Barbara D, Imielinski T. Sleepers and Workaholics: Caching in Mobile Distributed Environments. In: Proc. of 1994 ACM-SIGMOD Intl. Conf. on Management of Data, Minneapolis, MN, USA, June 1994. 1~2

5 Jing J, Elmagarmid A, Helal A, et al. Bit-Sequences: An Adaptive Cache Invalidation method in Mobile Client/Server Environment. ACM-Baltzer Journal on Special Topics in Mobile Networks and Applications, 1997,2(2):115~127

6 Cao G H. A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. IEEE Transactions on Knowledge and Data Engineering,2003,15(5):1251~1265

7 Kahol A, Khurana S. A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment. IEEE Trans. on Parallel and Distributed Systems, 2001,12(7):686~700

8 Li Y J, Chiang C Y, Liu M T. Effective Web Caching for GPRS Networks. In:Proc. of the Intl. Conf. on Computer Networks and Mobile Computing, Beijing, China , Oct. 2001

9 Bettstetter C, Vogel H, Eberspacher J. GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface . IEEE Communications Survey,1999,2(3):2~14

(上接第33页)

5 Kohler E,Morris R,Chen B J,et al. The Click Modular Router. ACM Tran. on Computer Systems,2000,18(3):263~297

6 Decasper D,Dittia Z,Parulkar G,et al. Router plugins: A software architecture for next-generation routers. IEEE/ACM Trans. on Networking,2000,8(1)

7 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

8 Peterson L,Gottlieb Y,Hibler M,et al. An OS Interface for Active Routers. IEEE Journal on Selected Areas in Communications, 2001,19(3):473~487

9 Keshav S,Sharma R. Issues and trends in router design. IEEE Communications Magazine,1998,36(5):144~151

10 Partridge C,et al. A 50Gb/s IP Router. IEEE/ACM Trans. on Networking,1998,6(3):237~248

11 Semeria C. Internet Backbone Router and Evolving Internet Design. White Paper, Juniper Networks, September 1999. http://www.juniper.net

12 徐恪,吴建平,江勇,徐明伟. 通用路由器软件体系结构研究综述. 小型微型计算机系统,2001(4)

13 IBM Corp. PowerNP NP4GS3 Datasheet. http://www.ibm.com/chips

14 Intel Corp. Intel IXP1200 Network Processor Family. http://developer.intel.com

15 EZchip Technologies Inc. NP-1 Network Processor Product Brief. http://www.ezchip.com

16 徐恪,吴建平,吴剑,徐明伟. 高性能路由器操作系统 HEROS 中的高可用性设计. 863学术年会,2001

17 Montz A B,et al. Scout: A Communication-Oriented Operation System. In:Proc. of the Fifth HotOS, May 1995. 58~61