计算机科学 2001Vol. 28№. 9

可编程网络计算模型与体系结构

The Computation Model & Architecture of Programmable Networks

唐 寅 王蔚然

(电子科技大学应用所三室 成都 610054)

Abstract Programmable networks[3](PNs) are a new type architecture of networks which allow users or the third part software providers to program it. The architecture of PNs can be customized by using open PNs interfaces(i. e. ,APIs of networks)and a set of services compositions and methods. The critical factors of the research of PNs are the requirement of rapid creation, develipment and management of new services. The new type architecture of networks can optimize the performance of networks, accelerating the development and application of new technologies and protocols by distributing lots of computing tasks into the cheap node of networks. Opensig-[2]v. s. active networks[1](ANs)are the two schools of PNs. This paper presents the concept, model & critical technologies of PNs in detail, then provides the realization approaches of ANs.

Keywords Programmable networks, Opensig active networks, IP networks, Customizing program

1 可编程网络的提出

随着时间的推移,节点的计算能力越来越强大,而花销则越来越低,同时越来越多的网络功能也在实施当中,目的是为用户提供更好的服务(如电信部门的增值服务)。而传统的计算机网络中的网络节点(路由器、网关、交换机)实现的功能本质上讲仅仅是实现数据包的存储转发。因而传统的网络只能被动地将数据从一个节点或终端传输到另一个节点或终端,网络节点只处理数据报头而对数据本身不会进行新的计算或改变。这样的网络又称为被动网络[5]。

然而网络技术的应用迅猛发展,新的技术层出不穷,如第三层交换、WDM、网络安全技术(如防火墙、VPN)等的出现使得这种被动的网络体系带来了新的问题。首先,在传统网络体系中一个技术从提出到推广应用一般要经过以下几个过程:协议的标准化、进入硬件供应商的产品、相关程序的开发和安装调试,所以新的协议、标准受网络原有平台系统的限制得不到快速应用或推广;其次,采用协议重叠[3]的方式开发应用新的协议技术增加了协议层间的冗余开销,降低了网络的性能;另外,新的应用需求往往难以迅速地在现有网络上实现,对数据本身的处理计算工作都限制在终端上,不能充分利用网络的计算能力。为了解决上述这些问题,人们提出了可编程网络的概念。

可编程网络涉及到用户、服务提供者和设备商,覆盖宽带、移动和 IP 网络电信部门。在传统和未来网络 ISPs 的竞争中,能迅速响应市场需求者方能获胜。因此,对新技术的采用将极具挑战并且呼唤能及时产生新的服务和网络技术的先进方法和工具。当前,有巨大需求的特定计算、处理和交换技术等有待解决,新的网络编程环境要求未来的网络体系结构应是开放的、可扩展的和可编程的。

通信硬件(如交换机,路由设备等)与控制软件的分离是使网络可编程的基本要素。但在当前的交换机和路由器集成体系中这种分离难以实现。服务提供者不能访问交换机/路由器的控制环境(如 Cisco 的 IOS 操作系统)、算法(如路由协议)或状态(如路由表、流量表),这使得开发新的网络服务几乎不可能,因为这需要比传统独占系统多许多数量级的灵活性。那么余留的问题是:怎样为第三方控制软件和服务的开发提供"开放环境"。

基于网络操作系统的网络编程环境与网络的可编程性密切相关,它使得提供新服务、开发和管理网络的结构成为可能。未来可编程网络操作系统可能是基于节点操作系统上的主动网络执行环境或支持多种控制体系的开放信令网络内核。这导致了两类可编程网络基本学派:主动网络(ANs)和开放信令(Open Signalling)网络体系。这两种提议都将直接面对同样的问

唐 寅 博士生,主要研究方向:IP 网络、可编程网络、网络安全、信息传输处理。王蔚然 教授,博导,主要研究方向:信息传输处理与多媒体、数据通信、图像处理。

题：怎样能加以控制地、安全地为开发新的体系、服务和协议而开放网络并加速其可编程性。

2 实现可编程网络的基本思想

由 Opensig 组织倡导的开放信令网络体系，建立在一系列国际厂商的支持下。Opensig 建议在通信硬件上开放对交换机或路由器的访问，并提供一系列开放的网络编程接口，以使第三方软件供应者能进入这些电信软件市场。Opensig 认为通过这种"开放"，在交换机上进行新的或特定的服务或体系（即虚拟网络）的开发能得以实现。可编程的具体实现使用被称为开放信令的技术。可编程网络与 QoS 服务保障中的传输、控制和管理有明显的区别。最近，IEEE1520 号工程项目旨在研究采用 Opensig 方法使可编程网络为 ATM 交换机、IP 路由器和移动电信网提供标准化编程接口。这样物理网络设备被抽象为具有知名开放可编程接口的分布式计算模型（如虚拟交换机、交换程序和虚拟基站）。这些开放接口允许服务提供者用中间件工具箱（如，COBAR）操纵网络的状态从而管理和构建新的网络服务。

1994 年至 1995 年 DARPA[7]在讨论未来网络系统的发展时指出，未来的网络应该能动态地支持各种不同的网络服务和新的业务需求，报文数据本身可以被一种语言描述，使其在网络中能得以灵活地计算和处理。从而提出了更加开放、性能更高的主动网络（AN）的概念。AN 组织倡导在现存 IP 网络限制下动态地开发网络新服务。这种动态支持新业务的级别远超过 Opensig。其动态性体现在具有可分配、可执行和转发的主动分组（active packets）概念上。作为一个主动网络的事例，密封舱（Capsules）[5]包含有可执行代码（如，Java 代码）和数据，其中可执行代码移动性是程序传输的主要载体（vehicle）。在这种新的网络体系中，网络由一组称为主动节点的网络节点组成，传输的数据包不同于传统的分组交换数据包，其传输的数据包中含有原始数据和能够执行的小程序，主动网络中的网络节点能对流经本节点的数据包进行计算和处理（包括路由跟踪、数据合并、安全鉴定等），然后将（可能已修改的）数据包发送给其它网络节点；用户可以向网络中加入客户化的程序使网络能迅速提供新的服务。这样，网络节点由被动地传输信息发展为能主动地处理信息，同时用户也由被动地使用网络发展到能主动地参与和定制自己需要的业务。

3 可编程网络模型

3.1 计算和通信模型

可编程网络和传统网络有显著的差异，它通过一系列应用编程接口（APIs）可实现非常广泛的高层服务。图 1 中以三维的形式给出了可编程网络的一个计算模型。这个模型包含了 Internet 参考模型（即应用层、传输层、网络层、链路层），并在其之上扩展了传输、控制和管理平台。这种对传输、控制和管理的划分使我们的计算模型能同时适用于电信网和 Internet。在 Internet 中任一单数据通路都可能贯穿传输（如视频分组）、控制（如资源预留协议—RSVP）和管理（如简单管理网络协议—SMNP）三部分机制。在电信网上通常也支持传输、控制和管理功能 这种划分是依据实现不同网络功能而采用的不同方法和不同的时间伸缩性要求。我们的通用模型能为不同的网络提供一个统一的抽象解释。

通过在网络中注入计算功能可以实现网络服务和可编程性，这种计算功能已不仅仅是传统路由器或交换机的计算能力。为了区分可编程网络和传统网络的概念，我们在通信模型上扩展了计算模型，从而明确地定义了网络体系的可编程性。我们可以把这个通用模型看作由传统的通信模型（包括传输、控制和管理平台）和计算模型组合而成，如图 1 所示。总之，计算模型提供了统一的跨越传输、控制和管理平台的可编程支持，在这些平台里可以对网络体系的各层次（包括应用层、传输层、网络层和链路层）进行编程，或者说通过这个计算模型使可编程性注入到传输、控制和管理平台中。

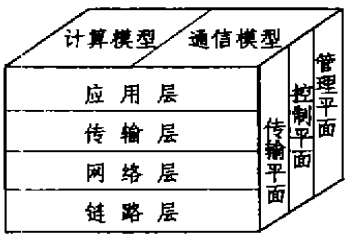

图 1 计算和通信模型

3.2 基本组成

图 1 示出了可编程网络的一个通用体系结构。它是一个三个层模型：最上层为通信模型，中间层为计算模型，最下层为节点硬件（通常指各种交换设备）。通信模型即是可编程网络的网络功能体系（PNA，Programmmable Network Architecture），它是实现网络功能的具体体现者，它提供网络算法（如路由、信令等）、网络服务、网络状态管理等功能。计算模型具体又可分为上下两个层次，上层为网络编程环境（NPE，Network Programming Enviroment），下层为网络节点内核（NK，Node Kernel）。节点内核通常指节点操作系统，它完成节点资源管理工作，因此节点内核仅具有本地意义，即管理单个节点的资源，并同时为多个网络体系共享。网络编程环境为分布式的网络编程服务提供

中间件支持。整个模型提供有两组编程接口，一组是网络编程环境与网络编程体系间的接口；另一组是节点内核与网络编程环境间的接口。这些编程接口都有待标准化以使其具有平台独立的网络编程性。

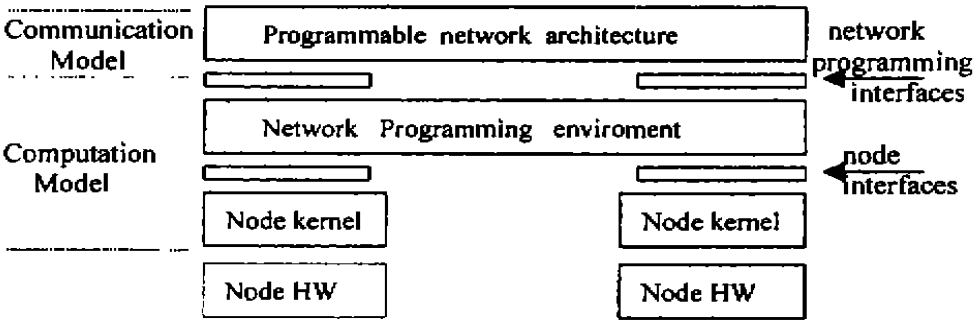

图 2　可编程网络体系结构

对可编程网络的研究集中在此模型的各个部分上，从而引发出了对不同的编程方法、编程性等级和通信机制的研究。与 Opensig 思想一致的有关研究项目重点强调可编程网络 APIs 的定义；其它基于主动网络的项目则集中于对代码移动性和分布式应用的研究，其中动态插件技术正逐步成为构建或扩展新协议和新应用的一种重要方法。下面对图 1 所示的可编程网络模型各部分分别进行较详细的分述。

3.3　可编程网络节点内核

这里节点内核是指运行在交换机/路由器上的底层编程环境。它提供一套有限的节点接口，这些接口支持节点状态（如访问和分配节点资源）的控制和对通信服务（如通信抽象和安全）的请求。节点内核负责分配计算资源（如共享 CPU）和通信资源（如复接器容量），也支持核心安全服务。节点内核运行在各类网络节点、终端系统或设备上，如 IP 路由器、ATM 交换机或基站。

我们通常见到的各种网络操作系统是网络节点内核的一种具体体现。许多节点设备供应商将他们的操作系统集成到他们的交换机和路由器里用来处理网络节点的各种通信功能。如 CISCO 公司的路由器里用的 IOS，ATML 公司的 ATM 交换机里用的 ATMOS 微内核。这些节点操作系统支持各种通信工作，如信令控制、进程管理、进程间通信、数据转发和升级操作系统映象。

3.4　可编程网络编程环境

可编程网络编程环境（NPE）[2]支持网络的构建和网络服务以及网络协议的动态开发。网络编程环境支持不同等级的编程、不同的编程方法、网络技术和应用领域。可编程网络环境的编程实现是通过节点内核提供的编程接口及其分布式工具箱，因而在此意义上说，可编程网络编程环境可以被看作介于可编程网络体系（PNA）和网络节点内核（NK）间的中间件。网络编程环境为网络设计者提供了构建各个网络体系的必要环境和工具，因而在这种意义上，网络编程环境类似于我们开发新应用软件所依赖的 SDK（Software Development Kits）。

3.5　可编程网络体系

网络编程环境为动态编程网络体系提供了必要的支持，但它不提供定义和区分不同网络体系的核心网络算法（如路由算法、信令算法），在这种意义上，它类似于操作系统不提供特定应用算法，而这里的可编程网络体系就相当于特定应用算法。

广义地讲，网络体系包括以下属性：1）各种网络服务，实现一系列分布式网络算法并服务于终端系统；2）网络算法，包括传输、信令/控制和管理机制；3）复合时间尺度，这影响到网络算法的设计；4）网络状态管理，如交换机、路由器、QoS 状态。

4　主动网实现技术

作为可编程网络的一个重要内容，主动网根据其程序代码和数据本身是同时集成到分组中传输还是各自独立地传输到网络节点，可将这种可编程主动网的实现方法分为三类：离散法、集成法以及这两种方法的结合[1,4]。

4.1　离散法

离散法又称主动节点法。网络中预留有二个通道：带内数据传输通道和带外管理传输通道。用户首先将数据处理程序代码通过带外通道传输到所需的主动节点上，然后当主动报文通过带内通道传输到该主动节点时，主动节点通过检查该报文的报头以调用相应的处理程序对它进行处理。新的处理程序代码可以动态地加载到主动节点上。这种方法非常适合需要严格控制所装载程序和客户化程序比较大的场合。

4.2　集成法

集成法又称密封舱法。每个封装体/密封舱内既包

含数据本身也包含一小段程序代码。当封装后的主动数据包到达某个节点后，主动数据包所携带的程序代码被执行，以处理数据包所携带的数据。若该封装体中的程序需要调用的处理方法不在该主动节点上，则可通过点播(on demand)在别的主动节点上下载该方法。这种方法具有更大的灵活性和可编程性，但实现起来也更复杂。

4.3 离散法和集成法结合使用

交换机小程序是离散法和集成法结合使用的一个例子，它由两部分组成，一部分是可编程的密封舱，另一部分是充当主动节点的交换机。使用这种方法的意义是：一方面在某些情况下，密封舱需要某些资源(如路由表等)，而它自己又不可能提供，必须由主动节点(如路由器、交换机)来提供。另一面，在密封舱和主动节点之间划分指令可避免密封舱变得过于庞大而降低网络性能。

总结 Opensig 和主动网从不同角度代表了可编程网络发展的两个基本思想，前者着眼于开放网络体系编程接口(如网络 APIs)以实现网络的可编程性，这是一种相对静态的网络可编程思想；而后者则强调能在网络体系中动态地加载客户化程序(无论是采用离散法、集成法或混合法)，通过网络节点上加载的不同程序来提供定制业务，这是一种更为灵活的方法。可编程网络的出现加速了网络发展的步伐，在可编程网上新的技术标准或协议从提出到应用不再需要漫长的标准化过程。用户可以根据自身需要迅速开发新业务。已有的研究成果表明可编程网的诸多技术能有效地改善网络性能，对 Internet 遗留的问题能提供有效的解决方法，被称为下一阶段网络发展的方向。因此，开展对可编程网的研究对我国的网络技术的发展具有重要意义。

参 考 文 献

1 Tennenhouse D, et al. A survey of active network research. IEEE Commum. Mag, 1997, 35(1): 80~86

2 Adam C M, Lazar A A, Lim K -S, Macroncini F The Binding Interface Base Specification Revision 2 0. OPENSIG Workshop on Open Signalling for ATM, Internet and Mobile Networks. Cambridge, UK, April 1997

3 Biswas J, et al. The IEEE P1520 Stabdards Initiative for Programmable Network Interfaces, IEEE Communications Magazine, Special Issue on Programmable Networks. Oct. 1998

4 Alexander D S, et al. The SwitchWare Active Network Architecture IEEE Network, May/June 1998

5 Wetherall D, et al. Introducing New Internet Services Why and How. IEEE Network, May/June 1998

6 Commentaries on Active Networking and End-To-End Arguments. IEEE Network, May/June 1998

7 Available at: http: // www. darpa. mil/ito/research/anets/

8 Available at: http: // www. cc. gatech. edu/projects/canes/

(上接第 47 页)

有的组状态信息相比较，用以确定组成员是否处在当前最新状态。

2)与时间敏感性组密钥更新策略相结合。当然可以使用时间敏感性组密钥更新策略周期性地强制要求组成员更新密钥，杜绝以上问题的发生，但对于较大型的组，进行一次全面的密钥更新要消耗较多的网络带宽，因此可以合理设置第一种方法的时间间隔 t，使其低于时间敏感性策略的定时器值，这样可以设置较大的定时器值，在每一个 t 间隔内，靠第一种方法来检测密钥更新消息是否及时正确地传递。

3)组成员向组管理者发送组密钥更新消息重传请求

Group Member → Group Manager: gi, mi, r, |H(gi, mi, r)|$_{k_{ai}}$

其中 mi 为组成员标识符，r 为组成员每次发请求时产生的不同随机数且对给定的组成员只准使用一次，k_{ai} 为组成员和组管理者共享的密钥。组管理者证实重传请求后，向指定组成员重新发送组密钥更新消息。

结论 本文给出的两种组密钥更新方案具有可放缩性好、消耗网络资源少、便于高效管理和能适应组成员动态变化等特性，是解决组播密钥更新问题的首选方案，同时针对在组密钥更新过程中易被忽略的组密钥更新消息的正确传输问题提出了有效的解决方法，需要进一步研究的问题是两种方案都靠组管理者集中控制，存在组管理者单点失效和瓶颈问题，如何在 Internet 网上提供一个分布式组播安全的组密钥更新方案是一个非常值得研究的问题。

参 考 文 献

1 Wallner D, Harder E, Agee R. Key Management for Multicast: Issues and Architectures. RFC2617, 1999

2 Harney H, Muckenhirn C. Group Key Management Protocol(GKMP) Specification. RFC2093, 1997

3 Ballardie A. Scalable Multicast Key Distribution. RFC1949, 1996

4 Mittra S. Iolus: A Framework for Scalable Secure Multicasting. Proc. ACM SIGCOMM, 1997

5 Wong C K, Gouda M, Lam S S. Secure Group Communications Using Key Graphs. Proc. ACM SIGCOMM, 1998

6 Blum M, Micali S. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. SIAM J. on Comp., 1984