维普资讯 http://www.cqvip.com

Active Network 系统的结构分析与研究*)

Analysis and Exploration of the Active Network architecture

马 燕

(重庆师范学院物理学与信息技术系 重庆400047)

Abstract The model of transfers and disposal of current network does not satisfy the requirement of increasingly complex structure and numerous operation of network, active network is becoming the hotspot of research at present. On the analysis of structure of active network, this paper analyses the structure and model of transfers and disposal of two sort of active network. Finally ,the paper researches the architecture and circulation of RANI and presents the simulating environment of experiment testing.

Keywords Active network, Activeware, Switchware, RANI

现有的网络系统基于"端到端"的模型体系,是1984年由H. Salzer、D. Reed 和 D. Clark 提出来的。在这种网络中,相关的计算和控制是在发送和接收"端点"进行的,在网络的结点上不对包进行处理。虽然在网络中路由器和交换机可以改变包头,但是对用户的数据不做任何处理,因此目前的网络系统又称为是"被动网络"。

Active Network 即主动网络,是1995年由 DARPA(Defense Advanced Research Projects Agency)开发由关于网络未来发展方向的讨论会议上提出来的。Active Network 是对传统的基于"端到端"模型体系的突破,它赋予了网络"编程"的功能,即网络的行为是可通过编程控制的。因此,整个网络是分布式的计算体系,其作用超出了"通信子网"的范畴。

一、Active Network 的体系结构

Active Network 是由一系列的主动节点(Active Node)构成的,每个节点都运行着一个节点操作系统(Node OS)和一些可执行环境 EEs(Execution Environments),以及相应的接口。其结构如图1所示。

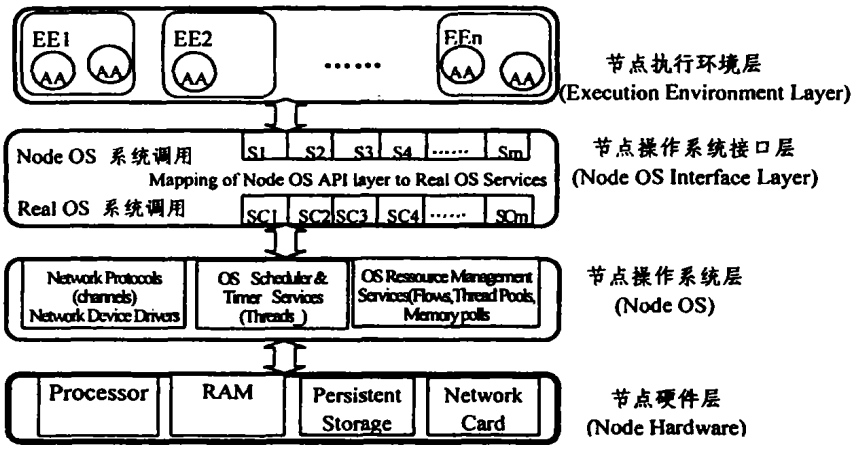

图1 主动网络节点的结构

节点的层次结构由四层构成:硬件层,即节点的物理层;节点操作系统(Node OS)及相应的接口;执行环境(EE)和主动应用(AA)。在上述的结构中,Node OS 提供了最基本的网络功能和资源管理、存取控制、代码转换和节点安全等机制,它还为 EE 提供安全保障、在主动节点之间实现分组路由。EE 定义了一个虚拟机 VM(Virtual Machine)和可编程接口。它由主动包中的指令控制,负责对到达节点的主动包进行解释和处理。EE 类似于计算机系统中的 Shell 程序,为用户提供端到端的支持。从图1中可以看出,Node OS 屏蔽了资源管理和 EE 环境等细节。主动包所携带的代码 EE 中执行,并使用 Node OS 所提供的网络传输信道来收发包。在节点中,AA 是一个应用程序,它在 EE 的环境中执行,以实现用户对网络主动节点资源的存取。

在主动网络中,用户可以根据需要动态扩展和定制网络的服务功能,具有网络的可编程性,支持代码的可移动性,具有与网络用户的可编程接口以及移动代码对主动节点资源的安全性访问机制等。

目前,Active Network 有两种典型的结构体系,一种是基于容器(Capsule)的集成方案,如 Activeware 体系结构,它主要应用于面向无连接的网络;另一种是基于可编程的交换节点(Programmable Switch)的离散方案,如 Switchware,它适合于面向连接的网络。

1.1 Activeware 系统的结构

Activeware 是由 MIT 实验室提出来的一种基于 ANTS

*)本文得到重庆市教委应用基础研究项目资助。

维普资讯 http://www.cqvip.com

(Active Node Transfer System)的主动网络体系结构,在ANTS的设计中,包的帧不仅包含了数据,而且还包含了一个代码段。这样,包含有数据和代码的包被形象地称为容器。在容器中的代码可以被主动网络中的节点识别并执行。

一个 Activeware 网络系统由若干个主动节点构成,在 Activeware 系统的中间节点上,除了现有的路由选择和交换功能外,还应新增三个组成元素:代码调用机制,其功能是将代码从容器中调入节点;其二是临时执行的环境,其功能是存储容器中的执行代码;其三是存储空间,用来存储节点在执行代码过程中产生的各种数据。当一个 Capsule 包到达节点时,Capsule 中的代码将从 Capsule 中分离出来送入到临时执行环境中并被执行,在执行中可以访问节点的存储资源,代码执行的结果可以生成新的若干 Capsule 并传送到网络其它节点上,如图2所示。

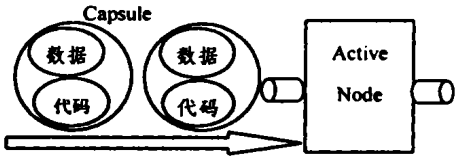

图2 基于 Capsule 的传输方案

一个 Activeware 节点包含三个主要的组成部件:容器编程模型—将 Capsule 划分成不同的层次结构以区别其代码、协议等;节点操作系统—控制 Capsule 中代码在节点中的运行并实现各种资源的保护和按照授权方式访问各种资源;代码传输机制—提供一种能够自适应节点连接而做出相应调整的自适应机制。图3是 Capsule 组成的格式:

Corresponding IP header			ANTS-specific header			High layers	
源地址	目的地址	TTL	报文	类型	前一节点地址	报头类型	用户数据

图3 Capsule 组成的格式

1.2 Switchware 系统的结构

Switchware 是美国 Pennsylvania 大学的 Switch Ware 项目组研制出来的主动网络系统,它是基于可编程的 Switch Ware 交换机方案,主动节点由具有一个可编程元件来实现交换功能,如图3a 所示。在这种方案中,包的格式保留现有的不变,但是提供了一个机制将用户程序调入网络中节点上。这些程序暂时存储在节点的存储空间中,包到达节点时,节点就会根据包头的信息来决定调用相应的用户程序来处理。如用户要对驻留节点程序进行修改或撤销时,可通过发送信息包来通知节点。其结构如图3b 所示。

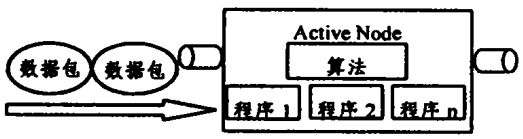

图3a 基于 Switchware 的传输方案

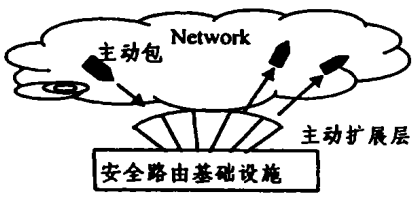

图3b Switchware 结构

在 Switchware 的结构中,有程序代码和数据两种类型的包,当主动节点接收到包时,首先对包的头部进行检查,如果包属于程序代码类型的,则对其进行安全验证后向节点注入代码。如果包属于数据类型,则节点根据头部信息选择节点中的相关程序对其数据进行处理。因此,节点能够提供比传统的简单路由表更为复杂、灵活的功能。在对这种机制研究的基础之上,Switchware 设计了具有主动网络程序设计的语言 PLAN(Programming Language of Active Network),它用于在包中描述用户程序功能并可以调用节点低层功能模块,并利用 PLAN 建立了一个主动互连网 PLANet。

在图3b 中,节点的主要功能是由主动扩展层提供的,为了实现节点的安全性、平台无关性和扩充各种性能,Switchware 选择了 Caml 语言,它是基于 ML 语言的扩展,类似于 Java 语言。

为了实现与传统的 IP 网络的兼容性,Switchware 最近提出了一种 PLAN-P 方案,它是在 PLAN 基础上扩展的一种语言,采用 PLAN-P 的主动网络极大程度地支持网络的编程能力和扩充功能。PLAN-P 系统是基于 IP 传输网络结构的,并且不需要改变包格式。它利用现有基于 IP、UDP 和 TCP 网络的高层服务功能,因此 PLAN-P 支持网络结点的编程但不具有对包的编程能力。其结构如图4所示。

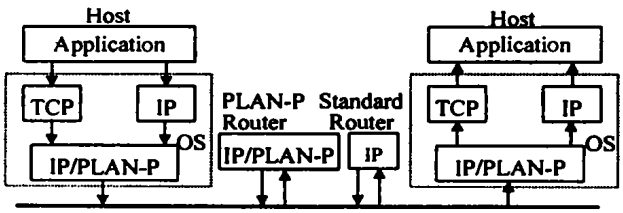

图4 PLAN-P 结构

PLAN-P 网络系统中由标准路由和 PLAN-P 路由组成,它不需要网络中每一个路由支持 PLAN-P 语言,网络中的 PLAN-P 路由中加载有 PLAN-P 的路由表,它提供了 PLAN-P 程序,主动包经由 PLAN-P 节点处理并传输。在该系统中,节点所接收的各种包可由同一个 PLAN-P 程序分类处理,而不像 PLAN 系统中一个程序只处理一种包。

二、RANI 主动网络系统结构的研究

RANI(Rutgers Active Network Initiative)是由美国 Rutgers 的 New Jersey 大学近年来提出来的一种主动网络结构。该系统的特点在于它是基于目前流行的 TCP/IP 网络设计的,因此它与现行的网络有很好的兼容性。我们对该网络体系进行研究,并给出一个模拟该系统的环境。

RANI 的主动节点可通过运行于 Windows NT 上的 Java 程序来实现基主动服务功能,它是处于网络协议的应用层(Application),因此运行是在 TCP/IP 协议栈中实现,其层次结构如图5(a)所示。

在该网络系统中,每一个节点都可与其它任意的节点实现包的交换。包有两种类型:一种是不需要主动服务的包,称为是被动型(Passive)包;另一种是主动型(Active)包。主动节点根据包头信息来判断包的类型。被动包到达节点时,节点只需要进行简单的存储和类似传统网络的传输处理。对于主动型网络则根据请求的类型提供主动服务。节点的结构如图5b 所示。

维普资讯 http://www.cqvip.com

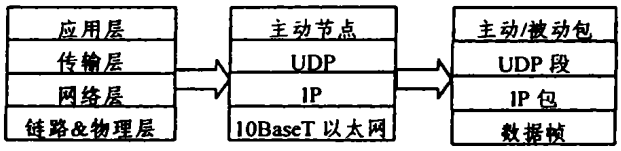

图5a RANI 的协议栈

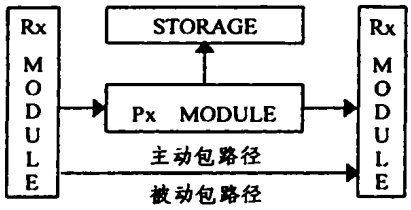

图5b RANI 节点的结构

在图5b 中，Rx 和 Tx 分别是节点的接收和发送端，主动包通过 Px 模块的功能提供相应的服务环境，服务程序则存放在 STORAGE 模块中。被动包则提供类似于传统的节点服务功能。为达到此目的，在节点中需要对包的类型进行识别。节点的处理过程如图6所示。

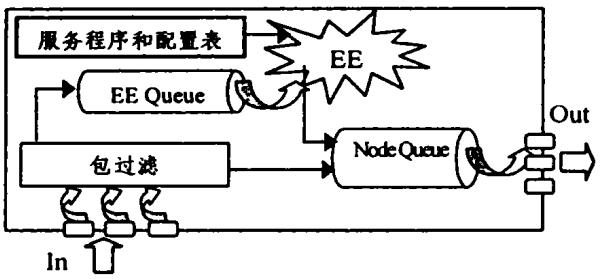

图6 RANI 节点对包的过滤

在图6中，EE 是主动包处理的环境，进入节点的包经过滤后，需要处理的包进入 EE 中，经对报头检查后调用相应的服务处理程序，而被动(Passive)包经排队后到节点的输出端。

主动包的格式如图7所示。

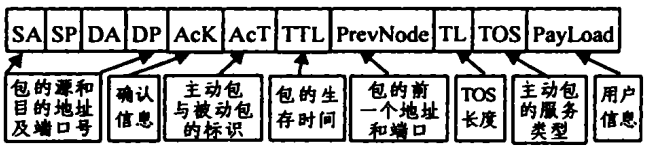

图7 Active Packet 包格式

2.1 RANI 节点传输过程

凡是到达 RANI 节点的包都要进行过滤检查，如果包的目的地址与本节点的 IP 地址相同，就将包交给本节点的应用程序处理。对于经检查后目的地址不是本节点的包，则进行下一步处理。对于被动包，交由队列 Node Q 中向前传送，它传送的机制由其所检查到的目的地址决定。而对于主动包，则交由队列 EE Q 中进入到 EE 模块中实现相应的主动服务。为了实现两个队列 Node Q 和 EE Q 的传输平衡，在设计中安排了两个队列：一个是快速队列(Fast Track)；另一个是慢速队列(Slow Track)。由于节点要对主动包的包头进行检查和处理需要花费较长的时间，如果仅设置一个队列则对被动包的传输不利。具体实现过程如图8所示。在图8中，对于到达目的处的主动包先交由 EE 环境处理后才交本地的 Application 程序。

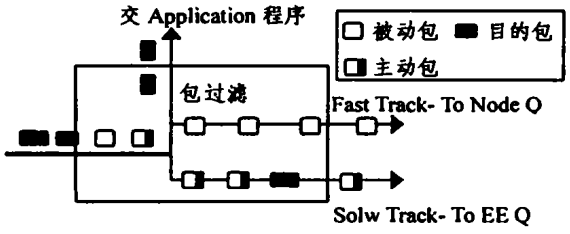

图8 RANI 节点的接收模式

2.2 RANI 节点处理过程

在 RANI 节点上的主动服务是按照等级存贮的，它们提供了可供服务的等级接口和相应的处理模式。节点根据主动包要求服务的等级来调用相应的服务。为实现管理，在节点上存储了一个可供服务的表列。该表列以 Hash 表格式保存，它以服务名作为关键字，以等级的描述符作为 Hash 值，主动服务在 EE 中执行，EE 按照 FIFO 的方式从 EE 队列中取出主动包。在主动包的 TOS 域中提供了要求主动服务的类型。节点根据服务名查找 Hash 表(具体处理的过程见图9)，并按照下列情况处理：

1)在 Hash 表中未能找到主动服务的请求，意味着该主动服务未装载在节点上，EE 试图从主动包中装载相关的主动服务(图9中路线1所示)。

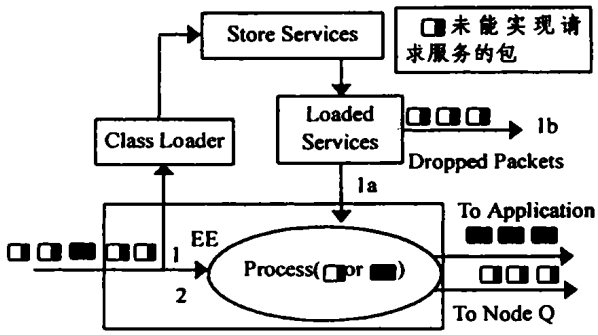

图9 RANI 节点的处理模式

(1)如果 EE 装载主动服务成功，更新 Hash 表，并且从主动包中调用相应的服务等级程序，以后凡是主动包有这类服务请求，则可以直接调用并进行处理(图9中路线1a 所示)。

(2)如果 EE 装载主动服务不成功，则放弃对此包的处理。以后对这种不能实现主动服务的主动包按照传统节点方式处理。这类包在所有的节点上都不能实现主动服务(图9中路线1b 所示)。

2)在 Hash 表中能找到主动服务的请求，就根据 Hash 表中的返回值直接按照服务的等级调用相应的主动服务程序(图9中路线2所示)。

下面是通过一个试验网络的模拟处理过程：

RANI 节点的处理包括对节点输入主动包和被动包、监视节点队列和测试虚拟链路。多个包在模拟 UDP 或 TCP 源的包生成器支持下产生并注入节点，用户可以选择包的数量、包的平均注入率和包的猝发规模。对节点的设置包括有创建(或撤销)虚拟链路、管理路由表和设置队列参数。主动网络中所有节点地址都是由唯一的 IPv4地址加以区别。在主动节点运行时，它与其它的链路可通过用户界面来创建，每一个链路创建成功后才会自动地加入到路由表中。路由表就作为包含了目的地址的 Hash 表来使用，它将端口号作为关键值，将链路上各节点服务程序名作为 Hash 值。当创建新链路或撤销

维普资讯 http://www.cqvip.com

链路时就更新 Hash 表的内容。也可通过手工设置路由表来改变网络中节点路由配置。

为了说明路由表的结构，我们采用图10所求的拓扑结构，节点之间的链路用 VLINK 标记。表1中显示了节点1和节点4的路由表结构。在表中节点的 IP 地址和链路的端口分配是任意的。图11是节点1上的用户界面，节点上使用 Windows NT 作为操作系统，采用 Java 语言编程实现。

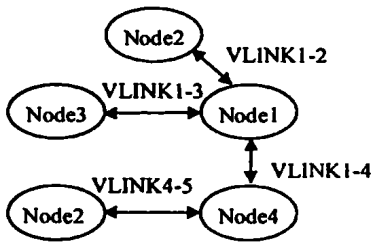

图10 模拟实验网络拓扑

表1 模拟实验网络系统中节点1和节点4的路由表

NODE1		NODE2	
Key	Vlaue	Key	Vlaue
Node2 128.6.43.52:2000	VLINK1-2	Node1 128.6.43.20:6000	VLINK1-4
Node3 128.6.30.3:4000	VLINK1-3	Node2 128.6.43.52:2000	VLINK2-4
Node4 128.6.21.18:3000	VLINK1-4	Node3 128.6.30.3:4000	VLINK3-4
Node5 128.6.21.19:3000	VLINK1-5	Node5 128.6.21.19:3000	VLINK5-4

RANI 主动网络基于菜单方式实现网络的编程，为用户提供一个通信应用程序和包处理的统一环境，信任操作可以允许用户调用新的服务程序或更新一个已存在的服务程序。通过图形界面可以使用户方便地进行设置和操作，可以通过重新启动服务来适应节点的拓扑结构改变。但由于 RANI 将主动节点建立在 TCP/IP 协议栈之上，因此使得对主动包的处理速度较慢。此外，由于假定网络可靠、链路稳定、路由表的静态性，而这些情况都与实际有差别的。

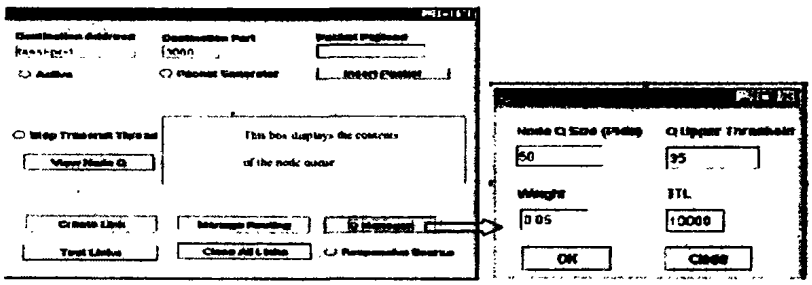

图11 在节点1上对 RANI 测试的用户界面

主动网络是一种崭新的网络结构，它采用的主动技术涉及到编译技术、操作系统、网络技术等各个方面。它使得网络可以动态地配置和动态控制，极大地提高了网络的性能并增加了网络的灵活性和扩展性，并为宽带网络的发展提供了广阔的前景。目前虽然还没有实用的产品推出，但是它得到了广泛的关注，现在正在对其关键性的技术如路由、资源分配、安全性、开发语言和平台等进行研究，可以相信，主动网络已经开始改变了传统网络的概念，它对未来技术的影响将起非常重要的推动作用。

参考文献

1 Hicks M, et al. PLAN Programmer's Guide. University of Pennsylvania, May 2000

2 Wetherall D, et al. Introducing New Internet Services: Why and How. IEEE Networks Magazine, May/June 1998

3 Moore O T, Nettles S M. Towards Practical Programmable Packets. Technical Report MS-CIS-00-12, University of Pennsylvania, May 2000

4 Schwartz B, et al. Smart Packets: Applying Active Networks to Network Management. ACM Trans. on Computer Systems, 2000, 18(1): 67～88

5 Alexander D S, et al. The SwitchWare Active Network Architecture. IEEE Network Magazine, 1998, 12(3): 29～36

6 Braden B, et al. Introduction to the ASP execution environment: [Technical report, USC/Information Science Institute]. Feb. 2000

7 Scott Alexander D, et al. The SwitchWare Active Network Architecture. IEEE Network Magazine, 1998, 12(3): 29～36

(上接第110页)

用类型，所采取的负载平衡策略可能会有很大的差异，因此在具体实现中，应充分考虑应用的特殊性，只有选择和设计最适合特定应用的方案，才能获得负载平衡的最佳效果。从这个角度来说，并不存在一种统一的最佳方案。随着实际需求中对负载平衡效率和精确度的更高要求，负载平衡硬件化的趋势也会越来越大。

参考文献

1 Brewer E. Clusters: Multiply and Conquer. Data Communications. Jul. 1997

2 Fox A, Gribble S, Chawathe Y, Brewer E A. Cluster-Based Scalable Network Services. In: Proc. of SOSP '97, St. Malo, France, Oct. 1997

3 Bryhni H, Klovning E, Kure Q. A Comparison of Load Balance Techniques for Scalable Web Servers. IEEE Network, Jul.-Aug. 2000. 58～64

4 Mosedale D, Foss W, McCool R. Lessons Learned Administering Netscape's Internet Site. IEEE Internet Comp., Mar.-Apr. 1997. 28～35

5 Cisco-Scaling the Internet Web Server. http://www.cisco.com. Jun. 2000

6 Andresen D, et al. SWEB: Towards a Scalable WWW Server on MultiComputers. In: Proc. 10th Int'l. Parallel Processing Symp., HI, Apr. 1996

7 Schroeder T, Goddard S, Ramamurthy B. Scalable Web Server Clustering Technologies. IEEE Network, May-Jun. 2000. 38～45

8 Gan X, Schroeder T, Goddard S, Ramamurthy B. LSMAC vs. LSNAT: Scalable Cluster-based Web Servers. Cluster Computing: the Journal of Networks, Software Tools and Applications, 2000, 3(3): 175～185