

一种主动网络管理系统模型的分析与研究^{*}

马 燕

(重庆师范大学信息技术系 重庆400047)

摘 要 传统的网络管理存在着一些无法解决的问题,主动网络则为解决这些问题提供了一个新的方法。本文分析了传统网络管理存在的问题及主动网络的结构和管理机制,同时,基于以主动节点为核心的管理,本文还讨论了主动网络管理模式,着重研究了其结构、管理机制、设计轮廓及系统各个部分之间的联系。

关键词 主动网络,网络管理,主动节点

The Research of the Management System Model Based on Active Network

MA Yan

(Dept. of Information Technology, Chongqing Normal University, Chongqing 400047)

Abstract Conventional network management suffers some unconquerable problem. Active network lends a novel method to settle these problems. This paper analyzes the problems of tradition network management, the structure and mechanism of the active network management system. It discusses the pattern of active network management based on the kernel of management of active node, and studies emphatically the structure, management mechanism, design outline and each connection of the management system.

Keywords Active network, Network management, Active node

1 前言

由于目前网络的体系结构在扩展性、网络管理、安全性和可靠性等方面所面临的各种问题,仅仅依靠现有网络的体系结构已不能够很好地解决。因此,对新型网络体系结构进行研究已经是势在必行。主动网络的概念是DARPA(Defense Advanced Research Projects Agency)研究协会于1995年提出来的。它有利于解决目前网络面临的诸多问题,如现有的共享网络结构中引入新技术和新服务的困难,信息穿过多个协议层所需的冗余操作导致的性能下降等问题。主动网络是新型网络体系结构的理想解决方案,是目前网络研究的一个热点。

2 传统的网络管理及存在的问题

目前计算机网络管理多为采用客户/服务器模式的集中式体系结构,其中最主要的代表是简单网络管理协议SNMP。该协议简单、易于实现且具有良好的可扩充性,已成为工业界事实上的网管协议标准。基于SNMP的网络管理系统模型如图1所示。

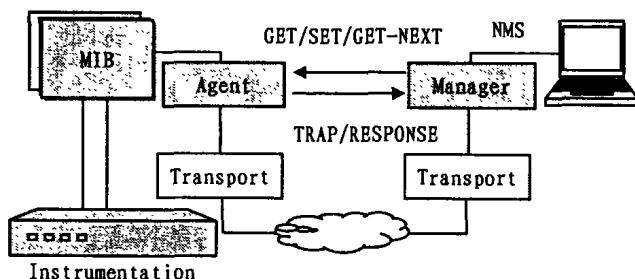


图1 SNMP网络管理模型

在该网络管理系统模型中,被管网络设备由两部分组成:

管理信息库(MIB; Management Information Base)和代理进程(Agent)。管理信息库是对网络被管对象信息的逻辑描述,它包含被管网络设备的配置、状态、错误和性能等方面的信息;代理进程驻留在被管网络设备中,与远程的网络管理系统NMS(Network management Station)通过SNMP协议通信,访问、控制MIB变量。通过SNMP协议,网络管理中心的网络管理系统NMS可以从被管网络设备的MIB中读取设备的配置、性能等状态信息及设置配置信息,同时被管网络设备也可以向NMS报告TRAP错误。

在SNMP结构中,NMS是整个网络管理的核心,它通过轮询被管设备获得该设备上管理信息库MIB中相关变量的值来获得网络的运行状态。

SNMP的网络管理是一种集中式、单序、反应式的模式,随着网络规模和复杂性的增加以及大量异构网络的存在,这种模式已很难适应对大型复杂网络的管理要求:(1)NMS作为信息汇集的中心,成为网络流量的瓶颈;(2)各被管设备的功能不能动态调整;(3)被管设备必须和NMS交换大量信息才能来完成网络管理工作,这既加重了网络的负荷又造成了管理任务的延迟;(4)NMS从各被管设备获取的是大量的原始数据,需要进一步处理才能变为有价值的管理数据。

由于集中式网络管理的以上缺点,近几年来新的网络管理模型不断推出,如基于Web的网络管理、使用智能代理或移动代理、基于CORBA的管理以及分布式管理等。特别是分布式网络管理,它将网络管理的部分工作交由被管理的设备来完成,这样不仅节约了网络的带宽,还可避免网络管理任务的延迟。

3 主动网络的体系结构

主动网络是由一系列被称为主动节点的网络节点构成,这些主动节点为主动报文中的代码提供一个执行环境。主动

^{*}重庆市教委应用基础研究资助项目(020805);“重庆市高等学校优秀中青年骨干教师资助计划”资助项目([2003]2号)。马燕教授,研究方向为计算机网络新技术,多媒体数据库。

网络体系结构的关键在于主动节点(ANN: Active Network Node)的体系结构。因此可以说,主动网络与传统被动网络最基本的区别在于:主动网络中存在主动节点,它们具有执行主动代码,对信包进行计算处理的能力。

主动网络节点的体系结构(如图2所示)从层次上划分,主要由三个组成部分。它们分别是节点操作系统(NOS: Node Operating System),执行环境(EE: Execution Environment)和主动应用(AA: Active Application)。

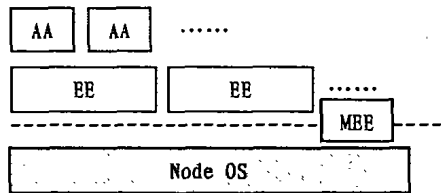


图2 主动网络的参考模型

(1)节点操作系统(NOS):位于主动网络节点的底层,它与主动网络节点的硬件资源接口共同管理和控制对主动网络节点的硬件资源的使用。这些主动网络节点的硬件资源包括节点的计算处理能力、存储器资源和 I/O 处理通道等系统资源。NOS 屏蔽了对这些资源使用的细节问题,给上一个功能层次执行环境(EE)提供使用这些资源的接口。为了提供 QoS 保证,NOS 还应该具有相应的策略和安全信息库。

(2)执行环境(EE):它和下层的节点操作系统(NOS)接口,同时为上层的主动应用(AA)层提供服务。执行环境(EE)是主动网络的主动应用在主动节点上执行的一个临时环境。一个主动网络节点可以具有多个执行环境(EE),每一种执行环境(EE)完成一种特定的功能。EE 为上层的主动应用(AA)层提供可编程的网络接口 API 或者一种虚拟机。

(3)主动应用(AA):最上层是主动应用(AA)层,它完成网络端到端的功能。一个主动应用(AA)由一段程序代码(主动代码)和与主动代码相关的数据、状态参数等组成。通过 EE 对 AA 的调用和执行可以实现用户定制的网络服务。

(4)管理 EE(MEE):负责对主动节点各个 EE、AA 以及各种硬件和软件资源提供一个可编程的管理接口。用户可以通过 MEE 对主动节点的各种状态信息进行收集而实施管理。相当于传统网络节点中嵌入式的 Agent,但它是可编程的接口。

4 主动网络管理系统的结构及实现

4.1 主动网络管理系统的结构

传统的网络管理由于采用集中式管理,无法利用主动网络中的节点的计算能力来管理网络。因此,它们不可能对主动网络实施有效的管理。为了适应主动网络的特点,主动网络的管理模式应能突破传统网络的非对称管理模式,使网络控制与管理工作站及主动节点之间达到一种对待的关系,从而克服传统网络管理中 Manager 端出现的瓶颈问题,也便于业务的动态加载和动态 MIB 的管理与维护。主动网络管理(ANM: Active Network Management)系统结构见图3所示。

从图3可以看出,ANM 体系结构中,主动节点是主动网管所要管理的主动对象。主动节点与控制管理工作站(NMS)之间的通信是一种对等的关系,而不像 SNMP 中客户端与服务端之间的非对等关系。NMS 是网络管理者控制和管理主动网络的界面;主动节点是网管系统的主要管理对象,负责处理主动信包;执行环境(EE)提供了主动信包运行和处理所必需的环境;MEE 负责主动节点的全局管理功能;代码服务器

(CS)负责提供网元设备收集数据所必需的逻辑方法;终端系统使用主动节点的服务运行主动应用。

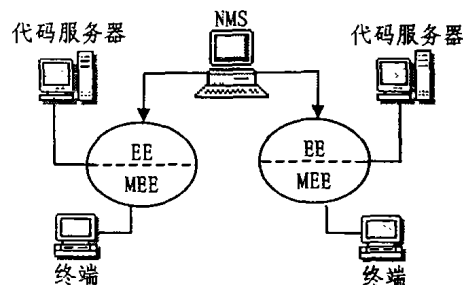


图3 主动网络管理的系统结构

4.2 主动网络管理技术的实现

对于主动网络而言,网络管理也应完成传统网络的配置管理、性能管理、故障管理、安全管理和计费管理五大功能。节点是主动网络的核心。在实际运行中,节点的结构、行为和属性都可能会随时发生变化,因此对主动网络的管理也提出了新的要求。除实现传统的网络管理功能外,主动网络管理主要应实现下述要求:

(1)新的管理业务动态加载:主动网络应具有动态新业务加载的功能。在主动网管系统中,针对新的业务,用户可以编制相应的管理业务代码并将这些管理业务动态地加载到主动节点,不需要变动原有的管理功能、管理信息库等,这有利于新的网管软件、协议的开发。

(2)动态 MIB 的创建与维护:在实际运行中,主动节点的结构和行为可能会发生变化。当一个主动网管系统运行之前,一些被管实体的 MIB 可能还是未知的。为了适应这种特点,主动网管系统必须提供一种可动态创建和维护的 MIB 机制来捕获和表达主动节点的动态结构和行为,可以根据管理业务的需要,动态地构建被管实体的 MIB,并加载到主动节点中,同时动态地进行维护和管理。

(3)与 SNMP 相兼容:主动网管系统应提供动态的 MIB 来实现对节点的管理。因为当主动节点中某些新业务增加使原有的 MIB 信息不够用时,需要增加新的信息来生成一棵新的 MIB 子树,以使网管系统能够从 MIB 中获得管理所需要的信息。

(4)代码移动:主要指网管工作站怎样将主动业务和代码加载进入主动节点。利用主动网络本身技术实现代码移动的方法有两种:离散型和集成型。这两种方法分别适用于美国 Pennsylvania 大学项目组研制出来的基于可编程的交换节点(Programmable Switch)的离散方案,如 Switchware,和 MIT 实验室提出来的基于容器(Capsule)的集成方案,如 Active-ware 的体系结构。

5 一种主动网络管理系统的的设计模型

5.1 管理系统的模型结构

根据主动网络的管理特点及主动网络的结构特征,我们认为一种较好的方案是以节点为核心的管理模式。为此首先将主动网络管理 ANM 系统中的主动节点管理用图4来进行描述。

在图4的节点管理模型中,在 NOS 与 EE 之间增加了节点管理器 ANM(ANet Node Mgr),在节点 NOS(Net OS)与底层之间包含了指令适配器(Instrumentation),实现了对节点的管理。

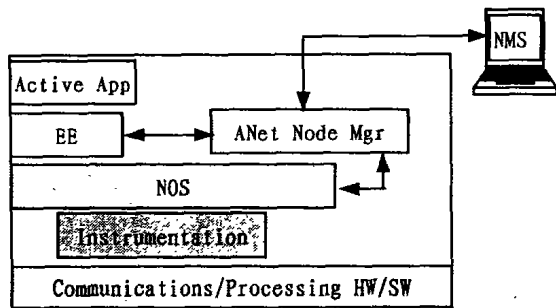


图4 主动网络管理系统框图

节点管理器 ANM 是由一系列 SW(SoftWare)组成,以实现对节点的监视、设置、分析与控制。ANM 与本地节点适配器(通过 Node OS API)一起访问相应数据、设置管理功能和控制相应事件,并通过与 EE 的相互协作实现管理节点 EE 设置、性能并处理运行中出现的问题;调整 SW 使其能够动态地适应主动应用的变化;通过其它的 EE 或 AA 实现对节点配置对象的管理。

节点管理器使 API 在 EE 的控制之下,通过主动应用调整与控制网络资源;主动应用可以监视网络性能并控制网络资源。节点管理器可以和 NMS 一起支持远程管理,通过调整 SW 以适应主动应用的动态变化。

图4是 ANM 的总体框图,现将其结构详细化为图5所示。

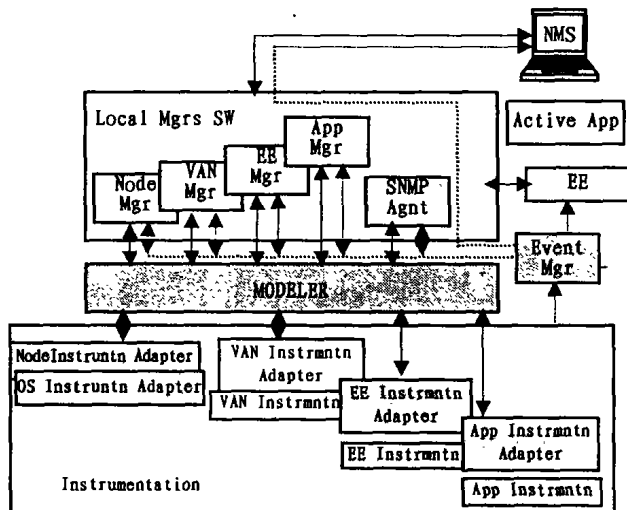


图5 基于节点的主动网络管理模式层次结构

在图5的结构图中,管理系统由节点管理层(Local Mgrs SW)、建模层(MODELER)和指令适配层(Instrumentation)组成,实现对节点的管理、配置、分析与监控。管理系统通过控制节点的 EE 实现对节点的配置、故障和性能管理,通过节点 OS 的 API 发出指令来访问节点的数据、配置节点及操作事件,同时还为 EE 提供一套 API 接口,以使主动应用(App)可以动态地适应与配置网络资源、对网络性能进行监视。

指令适配层(Instrumentation)位于管理系统中的底层,它提供了各种适配器如节点适配器(Node Instrmntn Adapter)、VAN 适配器(VAN Instrmntn Adapter)、EE 适配器(EE Instrmntn Adapter)、App 适配器(App Instrmntn Adapter)等来访问不同类型节点的事件和管理数据,指令适配器通过动态 MIB 来实现访问。

建模层(MODELER)将由底层所采集到的数据统一成网络管理应用程序所能够理解的数据格式。

节点管理层(Local Mgrs SW)负责管理节点的一系列资

源,如 VAN、EE、App、SNMP Agent 等,并与管理工作站 NMS 相连接,可以使网络 NMS 上的软件动态地适应主动应用的变化,也便于实现远程网络管理。

5.2 系统的分层管理设计

(1)指令适配层:当有新的 EE 或 App 加入节点时,通常会有提供访问新的相应服务或性能的接口指令,这些信息都将定入一个可扩展的动态 MIB 中并由此来扩展这些操作。动态 MIB(AMIB)与现在的 SNMP MIB 很相似,不同处是:AMIB 每一个变量有一个特定的方法和调用该方法的约定,指令适配层通过调用该方法来访问相关网络服务的信息;每一个变量都具有相应属性的特征定义,这些属性用以规定轮询形式、访问形式、同步请求等。

指令适配层不但能够正确解释这些接口 MIB 变量,还能够为其中的方法传递必要的参数,从而使建模层和节点管理层能够直接地访问这些 MIB 变量。下面是一个导出 MIB 变量的一个示例:

```
<AMIB>
<EXTEND MIB=url>
<OBJECT name>
  <RPC=rpcl>
  <MIDE=access/>
  <ATTRIB=value/>
</AMIB>
```

示例中的导出操作可以:①将新的对象通过指定的 url 加入到基本 AMIB 中;②由 name 指定其完整名字;③由 access 指定一种访问模式;④value 为对象定义一组特定的应用属性。

(2)数据建模层:数据建模层的主要功能是提供节点的 EE 调用的各种方法。下面给出一个相应的示例:

```
<DML>
<METHOD name>
  <ARGUMENT=type/>
  <URL=url/>
  <COMMENT=description/>
</METHOD>
</DML>
```

示例中功能的名字由 name 给出,节点的变元类型由 type 给出,url 给出了要实现功能的代码地址,description 作为注释。

建模层的总体目标是将管理数据组织成统一的形式,供本地或远程管理应用程序访问和分析网络配置和性能,它能够使管理器设置和控制网络组件及其相关行为。数据建模层分布在网络的各个节点中,可支持一定的重载缓存机制,以提高访问网络管理信息的速度。数据建模层应提供管理信息模式:

①网络拓扑模型,包括物理层、IP 层、虚拟网络层以及这些层之间的关系的拓扑结构。

②节点的网元操作模式,包括节点硬件与软件指令和对一个节点主动网元的操作。

基于上所述,数据建模层设计的关键在于:①改进和组织网络组件的数据模型以实现管理的自动化;②在多节点中统一分布建模层,以通过重载缓存机制实现对建模层的高效访问;③在建模层中通过快速动态的变化来适应主动网元的变化。

(3)节点管理层:节点管理层中各个管理器可实现在运行期间的动态装载与配置,它们可视为运行在 NM EE(网络管理环境)中的 AAs(主动应用)。本层中需要给出管理器的名称、参数、代码服务器地址及有关说明:

```
<LM>
<LOCAL MANAGER manager-name>
  <COMMAND-LINE ARGUMENT=arguments/>
  <URL=url/>
```

```

<COMMENT=description/>
</LOCAL MANAGER>
</LM>

```

节点管理层主要实现本地或远程节点管理的各项功能,它也是采用动态装配方法实现在网络管理中服务的分布。在节点管理层中的各个管理器既可以访问建模层提供的功能,也可以访问动态 MIB 库,它们是主动网络管理系统的基本构成模块。其主要构成模块有:

①节点管理器(Node Mgr):管理本地的硬件、软件与带宽资源,它可以配置和监视这些资源的性能并处理它们的操作故障。

②虚拟网络管理器(VAN Mgr):配置和监控虚拟主动网络的性能,包括创建和配置一个虚拟网络、分配本地资源、监控虚拟网络的性能和故障并进行恢复处理。

③执行环境管理器(EE Mgr):配置和监控 EE 的性能,包括配置 EE 环境、分配 EE 所需资源、将 EE 与 VAN 进行链接、监控 EE 的性能以及主动网元在 EE 中的执行情况。

④主动应用管理器(Active Mgr):可以扩展用以配置和监视各种特定的主动应用。

结束语 网络管理是一个庞大、复杂的系统工程。网络管理涉及到分布式计算、面向对象技术、人工智能等多种领域。

由于主动网络中节点具有独立的计算和处理能力,因此可以实现网络管理的分布化并减少网络中的冗余信息。

本文探讨了一种基于节点的主动网络管理模式的设计,并对其结构、性能与管理机制分析。这种管理模式具有稳定的层次结构、各层独立性强,而且便于扩充,满足了主动网络中节点的灵活性和主动应用的扩展性要求,是一种较为理想的网络管理方案。

参考文献

- 1 Mahon H, Bernet Y, Herzog S, Schnizlein J. Requirements for a Policy Management System, Internet Draft. Nov. 2000
- 2 Al Shaer E. Active Management Framework for Distributed Multimedia Systems. Journal of Networks and Systems Management, 2000, 8(1)
- 3 Kawamura R, Stadler R. Active Distributed Management for IP Networks. IEEE Communications Magazine, 2000, 38(4)
- 4 Salamanca E, Serrat J, Vivero J. Active policy-based management. In: OPENSIG 2001 Workshop, Sep. 2001
- 5 Kiwiior D, Zabele S. Active Resource allocation in Active Networks. IEEE JSAC, 2001, 19(3): 452~459
- 6 Schwartz B Y, et al. Smart Packets: applying Active Networks to Network Management. ACM Transaction on computer Systems, 2000, 18(1)

(上接第59页)

然而在实际系统中,满足上述要求的整体模型是以不同形式出现的,原因就在于各个系统的要求不同。这些特定要求主要被划分成以下几个方面:

(1)表达:在描述查询请求消息时,系统使用的查询语言必须能够充分描述所期望的数据。

(2)复杂度:在一些系统中,返回任意一个结果就足够了,但是在另一些系统中可能需要返回所有的结果。因此前一种系统只需要相对简单的搜索机制,而后一种系统需要复杂的搜索机制。

(3)自治性:每种搜索机制都要定义对等者相对于拓扑结构、数据排列和消息路由的行为,然而,对等者的自治性会受到限制。例如,某对等者可能只想与它的朋友们或者其它在相同组织中可以信任的节点相连接,或者该对等者期望控制那些可以存储它的数据的节点。根据系统的目的和用户需要,搜索机制要与对等者自治程度相一致。

3.2 设计目标

一个设计良好的搜索机制在必须满足系统要求的同时,还要提高以下目标:

(1)效率:当查询消息通过网络广播时,各个节点需要使用 CPU 资源来转发请求,处理它,并使用网络带宽来发送和接收请求,快速响应,因此一般多用带宽、处理能力、响应时间等衡量效率。

(2)服务质量:因为应用的不同,所以可以用不同的标准(如结果数量、响应时间等)来衡量服务质量。服务质量常指用户察觉到的质量,而效率关注达到特定服务程度所利用的资源代价(如带宽)。一般来讲,P2P 网络的 QoS 问题包括以下几个方面:①结果数量:符合要求的结果集的大小;②响应时间:用户需要的信息可能在多个节点同时存放,如何选择一个处理能力强、负载轻、带宽高的节点需要用户考虑。③满意程度:用户可能共享出无用或者违法信息,造成信息垃圾充斥网络。因此,网络应该能够控制用户共享的信息,提高用户获得有用信息的效率。

(3)鲁棒性:鲁棒性指的是失败时的稳定性,强健性,即系统中的对等者离开网络或者搜索失败时仍保持的服务质量和

效率。我们知道,P2P 系统中各个对等者是自治的,节点可以连续地加入或者脱离网络,因此网络规模和网络拓扑结构是不可预知的,变化很快。例如,用户可能会突然关闭其它人正在访问的电脑设备,出现其他人无法访问的情形,同时也增加了大量的不确定带宽,服务器资源和分布式存储的需求,这些动态特性要求网络中的节点能够按照网络信息的变化准确地配置和改变自己,所以必须提高系统的鲁棒性。

结束语 查询机制是 P2P 系统研究的关键问题之一。本文所描述几种常见的查询方法,给出了查询机制的一般要求和设计目标。随着 P2P 用户数量的快速增加,对资源查询机制的扩展性和可用性提出了很高的要求。随着技术和理论的发展,查询机制和实现方法会向着更加智能高效的方向发展,如一些系统采用智能代理等实现手段提高查询效率和网络带宽的利用率。

参考文献

- 1 Napster website. <http://www.napster.com>
- 2 Gnutella website. <http://www.gnutella.com>
- 3 Freenet website. <http://www.freenet.com>
- 4 Crespo A, Garcia-Monila H. Routing indices for peer-to-peer systems. In: Proc. of the Intl. Conf. On Distributed Computing Systems. 2002
- 5 Finding Good Peers in Peer to Peer Networks
- 6 Yang B, Garcia-Monila H. Improving search in peer-to-peer systems. In: Proc. of the International Conference On Distributed Computing Systems. 2002
- 7 Lv Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of Intl. Conf. on Supercomputing. 2002
- 8 Milojevic D S, Kalogeraki V, Lukose R. Peer-to Peer Computing: [Technical Report]. HP Laboratories Palo Alto. HPL-2002-57. 2002
- 9 Stoica I, et al. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. of ACM SIGCOMM'01, Aug. 2001
- 10 Ratnasamy S, et al. A scalable content-addressable network. In: Proc. of ACM SIGCOMM'01, Aug. 2001
- 11 Daswani N, Garcia-Monila H, Yang B. Open Problems in Data-Sharing Peer-to-Peer Systems
- 12 PeerSearch: Efficient Information Retrieval in Peer-to -Peer Networks. HPL-2002-198. July 12th, 2002
- 13 Sylvia R, Scott S, Ion S. Routing Algorithms for DHTs : Some Open Questions[C]. In: 1st Intl. Workshop on Peer-to-Peer Systems. March 2002