

一种主动网络管理系统结构的分析与研究^{*}

邹显春 张为群

(西南大学计算机与信息科学学院 重庆 400715)

摘要 主动网络是一种新型的网络体系,它不仅可以传递数据包,而且还可以执行数据包中特定的运算任务。主动网络为用户提供了可编程的接口,用户可通过网络中的节点动态地注入所需的服务。但由于传统网络管理不适应主动网络管理,不能发挥主动网络的分布式计算能力。论文讨论了一种基于节点的主动网络管理模型,分析了该模型的结构、管理机制和设计要点,并对网络拓扑发现和流量进行了分析。

关键词 主动网络,网络管理,主动节点

Study of the Hierarchy Management Model Based on Active Network Node

ZOU Xian-Chun ZHANG Wei-Qun

(Faculty of Computer and Information Science, Southwest University, Chongqing 400715)

Abstract Active network is a new framework where network nodes not only forward packets, but also perform customized computation on the packet flowing through them. It provides a programmable interface to the user where users dynamically inject services into the intermediate nodes. However, the traditional prototype of network management does not accommodate to the management of active networks, it cannot utilize the distributed copulation capabilities that active networks provides. This paper analyses the structure and mechanism of the active network management system, introduces a pattern of active network management, and studies the structure, management mechanism, design outline and each connection of the management system. The paper also studies the network topology discovery and traffic.

Keywords Active network, Network management, Active node, Network management system

主动网络 AN(Active Network)是针对传统网络发展中所遇到的实际问题而提出来的。主动网络是一种全新的网络计算模型,它赋予网络“编程”能力,允许用户向网络节点插入定制的程序或在报文插入程序代码,以便修改、存储或重定向网络中的数据流。主动网络是一种区别于传统网络的全新体系结构,为网络的快速发展提供了一个契机。

1 主动网络管理

1.1 主动网络管理

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,无法发挥和体现主动网络的优越性能。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^[1],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的动态 MIB 的管理与维护。

1.2 主动网络管理需求分析

主动网络管理也应完成诸如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)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 ActiveWare 的体系结构。

2 主动网络管理系统的结构设计

2.1 主动网络管理系统结构

根据主动网络的管理特点及主动网络的结构特征,我们提出了一种主动网络的管理模式。该管理基于节点为管理核心,充分利用了主动网络的主动性、分布性和智能性,以实现主动网络的分布式智能管理,其框图如图 1 所示。

^{*}重庆市信息产业发展基金(200401021);西南大学校改项目(290-411021)。邹显春 副教授,主要研究领域为计算机网络、软件工程;张为群 教授,计算机网络、软件工程。

在图1中,对节点的管理程序和监视程序是由 ANMS (Active Network Management System)采用 ANEP 标准封装成包并送到 ANEP 自适应数据鉴定器和监视器 Daemon(Data Adaptive Evaluator and Monitor),再由 Daemon 将包注入到网络中。图2中的P即为管理程序(Network Management Program),它是可传送的 Smart 包;图中的D即为 Daemon,它驻留在网络中的节点中。Smart 包在网络中的传送有端到端(end-by-end)或逐段转接(hop-by-hop)两种方式,在前一种方式下,Smart 包中的程序代码仅在目的端被执行。而在后一种方式下,Smart 中的程序代码可在源端、目的端及中间的所有段被执行。Smart 包中包含代码的程序可以将网络中任何一个主机运行结果带回到源端。

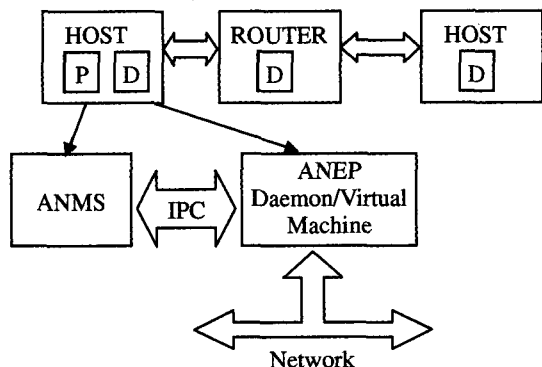


图1 主动网络管理系统的总体结构

在节点中,操作系统中安全策略数据库(Security Policy Database)和安全执行引擎(Security Enforcement Engine)保障了结点的安全。当AA或者EE中的实体需要结点操作系统提供服务时,由EE向NOS发出请求消息,该请求消息中带有主体标识符用以指明请求的生成者。结点操作系统收到消息后,将它提交给安全执行引擎,若通过了身份验证,再检查安全策略数据库,最后执行相应操作完成执行环境所请求的服务。

2.2 ANMS 结构与组成

ANMS(Active Network Management System)仅位于管理节点上,它是整个网络管理的核心。ANMS负责整个系统的控制,将管理员制定的管理策略付诸实现。网络管理员通过ANMS启动管理任务,负责网络系统中参数的设置和运行状况的监控。在ANMS中,由管理员通过命令方式,运用主动管理系统的各种AA,调用相应的模型以生成主动包注入到主动网络中,这些AA包括主动网络节点信息的获取、网络监控、分析和配置主动网络中各节点的应用程序。图2是ANMS(主动节点管理站)的结构图。

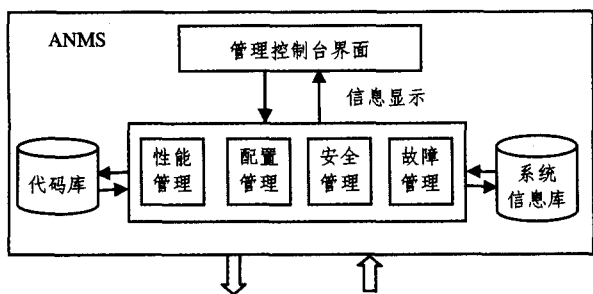


图2 主动管理 ANMS 结构模型

在ANMS中,由管理员通过命令方式,运用主动管理系统的各种AA,调用相应的模型以生成各种管理报文注入到

主动网络中,这些AA包括主动网络节点信息的获取、网络监控、分析和配置主动网络中各节点的应用程序。

ANMS应具有的功能有:(1)拓扑探测:ANMS应具有探测网络拓扑结构、生成结构的视图等功能,并尽可能维护网络的最新拓扑结构;(2)网络资源管理:ANMS负责网络资源的分配,以保证资源的合理使用;(3)系统的管理和维护;(4)安全控制:一是要保证ANMS上存储的信息安全,二是通过对ANMS发出的封装体进行数字签名,以证明其合法身份。

ANMS的管理功能由四个模块组成:

(1)用户界面:即网络管理员与ANMS之间的接口,它将网络中的资源信息、服务信息、网络运行状况和部署、数据流量等显示给管理员,如图3所示。管理员可以通过此界面发布网络管理命令。

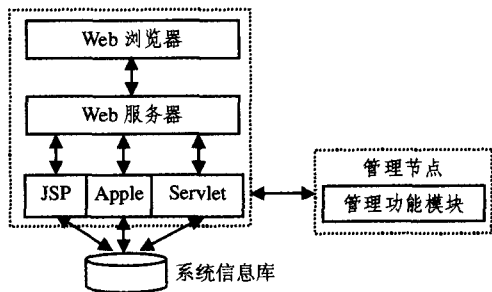


图3 用户界面接口示意图

每一个主动节点都有一个接收命令端口,本地系统利用这个端口来发布管理命令,这一过程是通过Servlet实现的。下面是一个发布管理命令的Servlet程序片段:

```
try{
    Socket antTeller=new Socket("10.10.10.101",205);
    PrintStream teller_stream=new PrintStream(new BufferedOut-
        putStream(antTeller.getOutputStream()));
    Teller_stream.println("app anm.configure.topology.TopAppli-
        cation");
    Teller_stream.flush();
    Teller_stream.close();
} Catch(Exception e){
    System.out.println(e);
    System.exit(0);
}
```

(2)管理功能模块:它是ANMS系统的核心,负责整个网络功能的调度和收集信息的处理,它包括对网络的性能、配置、安全和故障等进行调度与管理。

(3)接口模块:提供了访问信息库的方法,可以实现对信息库的打开、创建、查找等功能。本系统采用JDBC的方式来管理模块与数据库之间的连接。下面是一个访问数据库的Status的实例,它访问数据库MySQL:

```
java.sql.Connection conn;
java.sql.Statement stmt;
java.sql.ResultSet rs;
Class.forName("org.gjt.mm.mysql.Driver");
conn=DriverManager.getConnection("jdbc:mysql://10.10.10.101/
    ANMDB","owner","passwd");
stmt=conn.createStatement();
String sql="SELECT * FROM Status";
rs=stmt.executeUpdate(sql);
.....
rs.close();
stmt.close();
conn.close();
```

(4)系统信息库:用于记录网络资源信息,如网络配置信息、网络资源列表、当前可用资源等;此外,还要记录网络中各个节点的信息。

2.3 主动节点管理器 ANM

为实现对节点的管理,在NOS与EE之间增加了节点管

理器 ANM(ANet Node Mgr),它实现主动网络节点的 Daemon 功能,为节点提供一个安全可靠的运行环境。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。

ANM(ANet Node Mgr)作为主动网络管理的一部分,利用与被管理的主动节点之间的 EE 和 NOS 交互行为实现对节点的管理与控制,从而为管理程序提供共享的监视管控制节点的功能^[5]。ANM 的主要功能有:(1)提供对节点的监视、控制功能;当 ANMS 为完成某项监控功能而发出一个封装报文时,ANM 应保证封装体在当前节点上能够正确执行,同时 ANM 在安全允许的范围内应该如实报告本节点的运行状况。(2)安全控制:对 ANMS 发来的封装体和用户的身份进行合法性的鉴别,根据本地节点的安全机制决定可以对节点资源实施何种访问与控制等。

图 4 是 ANM 的总体流程图,当 ANM 启动后,首先要进行初始化工作,主要包括读取配置文件中的数据以配置 ANM 和 MIB 中的变量;注册 ANM 与远程 ANMS,初始化 ANM 的端口等。然后,ANM 进入本地管理子模块。

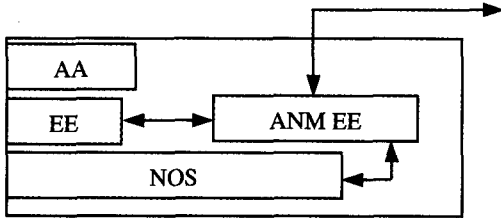


图 4 主动网络管理系统中的主动节点

本地管理子模块由一个无限循环体构成,在这个循环内接收远程 ANMS 系统配置 ANM 模块的封装体报文,根据报文类型交由各类报文的处理句柄处理。本地管理子模块要为其它应用程序提供访问被管理资源的原语。在具体实现中,我们先定义了一个抽象类,该类声明了 get 原语和 set 原语。其中 set 原语用于设置被管理资源的值,get 原语用于获得当前被管理的资源值。其格式为:

get 原语: errorStatus=get(nodeIdentify, digest, objectId, value)

set 原语: errorStatus=set(nodeIdentify, digest, objectId, value)

其中,errorStatus 是原语执行的返回代码,errorStatus=0 表示执行原语成功。

我们利用主动 Capsule 对 Mib 进行设置、获得等操作,Set 操作主要包括 SnmpSetCapsule、SnmpSetProtocol、SnmpSetApplication 等类组成。在类 SnmpSetCapsule 中实现了 ANMSSet 方法。在对网络元素的 MIB 访问中,我们使用了 AdventNet SNMP API,它提供了使用 Java 访问 MIB 的接口。在这里仅给出 ANMSSet 的实现。

```
Public boolean ANMSSet (SnmpSetCapsule c){
//.....
AddressIPv4 destAddr;
String snmpOut, community, port, destination, oid, value;
SnmpTarget target = new SnmpTarget();
//取得 Set 操作参数
port = c.port;
community = c.community;
try
```

```
{ String rfcMib = (SNMP.mibCacheDir() + "/RFC1213-MIB");
target.loadMibs(rfcMib);
target.setTargetHost(n.getAddress());
target.setTargetPort(Integer.parseInt(port));
target.setCommunity(community);
oid = c.oid;
target.setObjectID(oid);
value = c.value;
// SET 操作
snmpOut = target.snmpSet(value);
}
catch (Exception e)
{ //.....
}
```

在原语的执行过程中,它首先调用安全控制子模块进行安全性检查,向安全控制子模块传递的参数是请求资源访问的节点标识符和验证此节点身份的密文。

3 ANEP 封装体及转发模式

封装体是由管理节点的 ANMS 向主动网络中节点所发出的主动报文,其格式采用 ANEP 封装。它使用四个 ANEP 的可选项:源地址、目的地址、完整性校验和和鉴定可选项^[6]。鉴定可选项的细节如图 5 所示,它主要识别报文的发送者,包含了一个数字签名和一个公开密钥认证。各个域的说明如下:

(1)ID 域包含一个 IPv4 或 IPv6 地址,用 ID 类型和长度域标识,其值与在 ANEP 源地址可选项的源地址域中的值相同;

ANEP 可选项头部			
ID 类型	签名类型	论证类型	ID 长度
签名长度	论证长度	负载长度	
ID			
签 名			
认 证			

(2)签名域是一个数字签名,用数字签名类型和长度域标识,这个智能包的数字签名算法的有效类型可以是哈希算法或是 MD5(第 5 类报文摘要算法);

(3)认证域遵循 X.509 公开密钥认证,用标识类型和长度域标识。该认证域包含有 IPv4 或 IPv6 地址的值。

结束语 主动网络管理体现了主动网络的思想,将一部分网络管理功能动态地分布在主动节点上,充分利用了主动节点的计算能力,使节点能够自动发现、解决问题,从而极大地优化了网络管理。本文讨论了一种主动网络的管理模型,该模型中各个模块相互独立、任务明确,而且在每个层都可以动态更新以适应主动网络中主动节点的易变性和主动应用的扩展性,因此网络管理的稳定性和扩展性都大为提高,适应了现代网络管理的需要。

参考文献

1 Moore O T, Nettles S M. [EB/OL]. Towards Practical Programmable Packets; [Technical Report MS-CIS-00-12]. University of Pennsylvania, May 2000

2 Di Fatta G, Lo Re G. an Evolution of the Internet. Active Networks [J]. In: Proc. of AICA2001 - 39th Annual Conference, Cernobbio, Italy, 2001

3 Marshall I W, et al. Active management of multiservice networks. In: Proc. IEEE NOMS, 2000, 981~983

4 Al Shaer E. Active Management Framework for Distributed Multimedia Systems [J]. Journal of Networks and Systems Management, 2000, 8(1): 49~72

5 Brunner M, Stadler R. Service Management in Multi-Party Active Networks [J]. IEEE Communications Magazine, 2000, 38(3): 281~286

6 Kiwior D, Zabele S. Active Resource allocation in Active Networks [J]. IEEE JSAC, 2000, 19(3): 452~459