

一种主动网络管理系统结构模型的研究

钟国祥^{1,2} 邱玉辉² 马燕³

(重庆教育学院 重庆 400067)¹ (西南大学计算机与信息科学学院 重庆 4000715)²

(重庆师范大学物理学与信息技术学院 重庆 400047)³

摘要 随着网络规模的不断扩大,为了提高网络的可靠性和管理水平,需要进一步改进原有的网络管理的体系结构。而基于 SNMP 协议的传统网络管理体系已不能适应规模日益扩大网络的需要。本文讨论了一种基于节点的主动网络管理框架体系,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 主动网络管理, ANMS, 报文转发, 拓扑发现

Design of the Model of the Management System Based on Active Network

ZHONG Guo-Xiang^{1,2} QIU Yu-Hui² MA Yan³

(Chongqing Educational College, Chongqing 400067)¹ (Faculty of Computer and Information Science, Southwest University, Chongqing 400715)²

(Faculty of Physics and Information Technology, Chongqing Normal University, Chongqing 400047)³

Abstract With the quick expansion of network, the centralized strategies based on ANMP are not appropriate in large-scale networks. 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 management, ANMS, Packets transmitting, Topology discovery

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。基于主动网络的管理技术是将主动网络与网络管理相结合的新型网络管理技术,将势必加大网络管理的现代化进程。

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。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、动态性和智能性,以实现主动网络的分布式智能管理,其框图如图 1 所示。

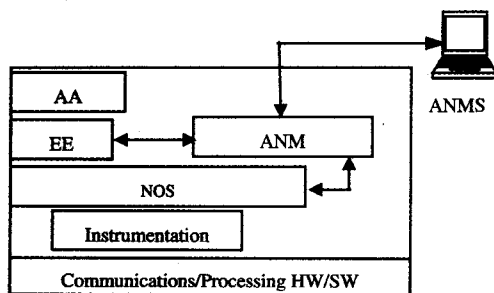


图 1 主动网络管理系统框图

在图 2 的管理模型中,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。ANM 是由一系列 SW(SoftWare)组成,以实现节点的监视、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。

2 一种主动网络管理系统 ANMS 的结构设计

2.1 主动网络管理系统 ANMS

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。

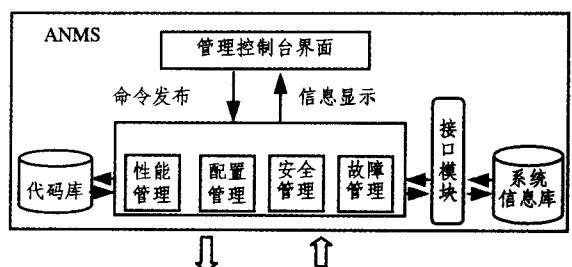


图 2 ANMS 的结构图

ANMS 应具有如下功能。(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)系统的管理和和服务;(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。

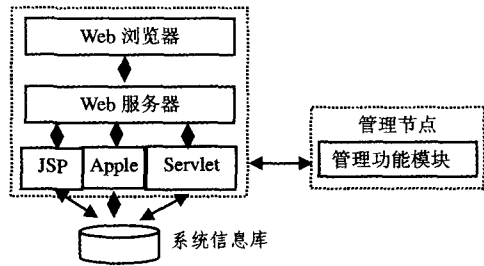


图 3 用户界面接口示意图

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-
    putSream(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.ecit(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

实现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("ofg.git.mm.mysql.Driver");
conn=DriverManager.getConnection("jdbc:mysql://10.10.10.101/
ANMDB","owner","paswd");
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。

2.2 主动节点管理器 ANM

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 为完成某项监控制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。

ANM 在启动后,需要与三个部分进行通信:(1)与节点操作系统 NOS 交互。通过 NOS API 访问本地资源如 CPU、内存使用情况、资源配置、路由表的更改等;(2)与 EE 的交互。访问 EE 的配置和性能等,掌握每个 EE 占用的 CPU 资源、内存等,从而平衡多个 EE 之间的资源分配;(3)与 ANMS 交互。为 ANMS 提供接口,使 ANMS 通过它完成对节点配置的访问与控制。

图 4 是 ANM 的结构图。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。ANM 主要的组成有:

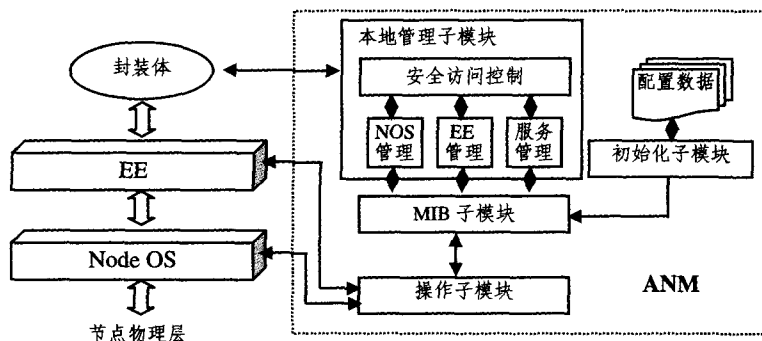


图 4 ANM 的结构图

(1)本地管理子模块。负责对节点 NOS、EE、服务等进行管理,向远程 ANMS 提供访问节点资源的机制并与 ANMS 实现通信。在初始化启动后,它维护节点资源信息、为 ANMS 访问节点资源提供服务原语、接收 ANMS 的服务请求、检测节点性能和配置参数等。它采用了 MIB 的命名方式,以对象标识符来命名被管对象。它提供了两个服务原语:

get 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 操作主要由 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.nocacheDir() + "/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。

(2)安全访问控制子模块。其功能是保护被管理节点上的资源不受到非法访问。ANMS 所发出的封装体报文应提供所访问节点的标识符和经过密钥计算的摘要。该模块通过封装体报文中的标识符搜索访问权限库,利用权限库中的公钥来识别身份,并根据权限库中的权限列表来判断所请求的资源是否允许被访问。

(3)MIB子模块。沿用 SNMP 中命名的资源管理方式,以对象标识符来命名被管理对象。它将 MIB 对象映射成节点的本地变量,将对外统一的管理功能接口映射成节点内部的管理功能函数。

(4)操作子模块。它通过 NOS 的 API 和 EE 提供的接口来访问它们的配置、性能数据,调用它们的配置函数。

3 主动网络管理模块功能的设计

配置管理模块的功能主要是监控网络和系统的配置信息保存数据,维护所有网络元素的最新配置清单,允许网络管理员查询、修改网络元素的配置。网络管理员可以根据网络的运行状态改变设备的一些当前设置,从而控制网络的运行。除对单个设备的管理之外,配置管理还要能够发现和管理设备之间的连接关系,即发现网络的拓扑结构。

网络配置信息主要包括:网络设备的拓扑结构、网络设备的 IP 地址、网络设备的运行特征等,配置管理通过修改被管理对象的属性来控制被管对象。配置管理的对象是一个逻辑的概念,可以是软件,也可以是硬件。本系统中的配置信息数据库的定义使用关系型数据库的方法。数据库中的每一项代表了相应网络元素的某种特性。数据库的结构反映了网络元素之间的关系。整个配置管理模块结构如图 5 所示。图中配置信息数据库用来保存网络的配置信息,拓扑发现和重构模块提供自动拓扑发现功能,并将所发现的路由器、主机、端口

以及其他网络设备的信息保存在配置数据库中,同时可以在配置数据库中的信息和网络拓扑关系不一致的情况下,对网络拓扑结构进行修正。定期更新配置信息模块提供定期更新路由器设备、主机及其端口的配置信息。配置节点信息模块主要完成由管理员或者其它模块发出的设置或者修改网络设备的配置信息功能。

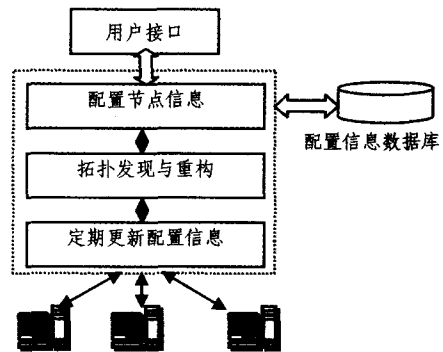


图 5 配置管理模块结构图

4 主动报文的封装及转发模式

3.1 ANEP 报文封装格式

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 6 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:

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			

图 6 ANEP 报文的可选项结构

(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;

(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。

3.2 报文的转发

封装体是网络中实现分布式管理的主要元素,它的报文是相当灵活的,可以在传输过程中根据某个节点的状态信息进行计算,决定报文的转发方向、向管理节点返回封装体报文的类型及其携带的数据信息。这种灵活性是通过转发例程来实现的,例程存在于各个节点中,目前将转发例程分为 4 类。

(1)One-One 模式:One-One 的转发模式是一种最简单的转发模式,它又根据包是否在所经过节算分成两种结构:一是所经中间节点不执行,其方式与目前的端到端通信方式类似。

(下转第 49 页)

originator (o=)
 contact (c=)
 media_description (=)

这些与 IP 地址或端口联系的字段必须被修改得与目标协议簇一致。这些地址包括在发送到 UFWDD 的映射请求的 SDP 部分。

④ Content_length 当一个消息体(SDP) 被修改后, 包长也要重新计算。

⑤ VIA VIA 头会插入到请求消息, 从响应消息里删除。

(2) UFWDD (UDP 域转发)

这个功能实体管理 IPv4 和 IPv6 地址空间, 进行网络地址翻译。从 IPv4/IPv6 收到的数据包使用由网关管理的 IPv6/IPv4 地址表分配的 IPv6/IPv4 地址转发给相应的 IPv6/IPv4 主机。接收到的 UDP 包请求格式如下:

MAPPED 0.0.0.0; 0 0.0.0.0: 0

[::]: 0[3ffe: 3ff: 1be0: 273: 3e0e: 1bcf: 0ace: 86ab]:
 10003 2

这里有 5 个字段参量: 源地址/端口, 入地址/端口, 出地址/端口, 目标地址/端口, 相邻端口数。UFWDD 用“MAPPED”格式回答源映射请求:

MAPPED 0.0.0.0; 0 210.41.35.199; 70001

[3ffe: 4ff: 1be0: 273: 18d0: 6568: a9bc: 1f00]; 7005

[3ffe: 3ff: 1be0: 273: 3e0e: 1bcf: 0ace: 86ab]; 10003

2

在这个例子中,UFWDD 从 210.41.35.199; 70001 收到数据包, 属于 IPv4 地址/端口, 将这个数据包通过 IPv6 输出地址/端口[3ffe: 3ff: 1be0: 273: 3e0e: 1bcf: 0ace: 86ab]: 10003 转发到 IPv6 目标地址[3ffe: 4ff: 1be0: 273: 18d0: 6568: a9bc: 1f00]: 7005, 并忽略被请求的源地址/端口。为了使接收到的目标地址返回的数据包可达, 同时还要建立一张反向映射表, 这样, 从出地址收到的来自目标地址的数据包就可以通过入地址发到源地址。只有当来自源地址的第一个

数据包被接受后, 反向映射表才会被建立。

(3) 控制协议

为了请求分配地址, 在协议簇间映射都通过 UDP 进行。这就允许不同结构下的组件相互通信。在图 1 中, 除了网关还有 SIP 代理。每个代理都表示混合 I 网络中一边的 SIP 服务提供商。例如, 一个 SIP 服务提供商如 iporg, 一个来自 IPv4/IPv6 网络的请求, 目的为 iporg 的用户, 会被定向到 IPv4/IPv6 网络中的 iporg 代理, 当代理在呼叫发起的网络(如 IPv4) 中时, 不能确定用户位置, 将会向网关转发请求, 在经过修改转换之后, 会再次转发到位于 IPv6 网络中的 iporg 代理, 这些功能模块被统一放置于 SIP_PGW 中。当然, 这样的结构无疑会增加网关的复杂性和网关的负载, 尤其在处理大量的 SIP 消息和可能执行的诸如 CPL 一类的服务时, 或是类似的翻译和路由数据之外服务。但是这样的结构可以使网关的生产商独立提供信令与媒体数据包的转接设备, 减轻 SIP 服务商的压力, 使他们专注于处理媒体业务, 而不必考虑网络间的接口问题。

结束语 本文提出了一种有效连接 IPv4 和 IPv6 网络之间的通信方法。这种解决方案是基于用户域的, 而不是核心域, 可以有更高的执行效率, 但是需要 2 个 SIP 代理, 这增加了管理的开销。下一步我们将代理服务器和网关融合, 使用更底层的 NAT-PT 进行 IP 头的转换, 以取得更高的效率。

参考文献

- 1 Rosenberg J, Schulzrinne H, Camarillo G. SIP: Session Initiation Protocol. RFC3261, IETF, 2002-06
- 2 Holdrege M, Srisuresh P. Protocol Complications with the IP Network Address Translator (NAT). RFC3027, 2001-01
- 3 Deering S, Hinden R. Internet Protocol (Version 6). RFC 2460, IETF, 1998-10
- 4 Handley M, Jacobson V. SDP: Session Description Protocol. RFC2327, 1998-04
- 5 Gilligan R E. Nordmark; Transition Mechanisms for IPv6 Hosts and Routers. RFC2893, 2000-08
- 6 RFC3303; Middlebox communication architecture and framework
- 7 RFC2824; Call processing language framework and requirement

(上接第 46 页)

二是沿传输路径计算的转发模式, 在这种转发模式中, 封装体报文按照指定的每到一个中间节点时, 就要在该节点执行其携带的程序。通过这种执行方式, 管理节点可以把集中的任务分发到沿整个传输路径上去执行。

(2) One-Multi 模式: 在这种转发模式中, 多个同样的封装体报文同时从一个节点发出, 这些封装体报文发向不同的目的节点并且在该节点执行。这种转发模式可以用于信息的广播(如拥塞位置的检测)、子网的控制。

(3) BFST(Breadth First Search Traversing) 转发模式: 这是一种并行控制模式。当封装体报文到达一个主动节点后, 它被转发到与当前节点直接相连的邻居节点。到达下一个节点时同样按照将该报文直接转发给相邻的节点。显然, 经过一次转发后, 网络中将出现很多该封装体报文的副本。当这些副本到达下一个节点后, 它们又被复制, 并转发到它们的邻居节点, 报文副本依次转发下去, 直到遍历完整个网络。

(4) DFST(Depth First Search Traversing) 转发模式: 这是一种串行控制模式。在该模式中, 封装体报文到达一个主动节点后, 它被转发到与当前节点直接相连的一个邻居节点。当它到达这个邻居节点后, 它又被转发到该邻居节点的一个邻居节点, 依次转发下去, 直至遍历完整个网络。

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. Active Networks [J]: an Evolution of the Internet. In: Proc of AICA2001 - 39th Annual Conference, Cernobbio, Italy, Sept 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: 49~72
- 5 Brunner M, Stadler R. Service Management in Multi-Party Active Networks [J]. IEEE Communications Magazine, March 2000, 38(3): 281~286
- 6 Kiwior D, Zabele S. Active Resource Allocation in Active Networks [J]. IEEE JSAC, March 2000, 19(3): 452~459