

基于 XML 技术的网络配置管理系统

章 勋, 杨家海, 王继龙, 张 辉

(清华大学信息网络工程研究中心, 北京 100084)

摘 要: 传统的网管系统无法适应由网络技术的不断发展而带来的诸多问题。该文讨论了将可扩展标记语言(XML)技术应用于网络管理的相关问题, 引入了 XML 协议转换网关, 以解决基于 XML 技术的网络管理系统在现行网络环境下的部署和使用问题, 实现基于网管系统界面的统一配置管理功能。

关键词: 网络配置管理; 简单网络管理协议; 可扩展标记语言

XML-based Network Configuration Management System

ZHANG Xun, YANG Jia-hai, WANG Ji-long, ZHANG Hui

(Network Research Center, Tsinghua University, Beijing 100084)

【Abstract】 With the rapid development of the network technologies, conventional network management systems are hard to meet the increasing requirements. This paper studies the adoption of XML technology in network management, introduces an XML gateway to make the XML-based network management system deployable in legacy SNMP-based network environments, and implements a universal configuration manager which can provide unified, GUI-based configuration service.

【Key words】 network configuration management; SNMP; eXtensible Markup Language(XML)

1 概述

随着互联网的飞速发展, 网络规模不断扩大, 网络复杂性不断增加, 对网络管理的要求越来越高。传统的网络管理是基于 IETF 在 1988 年提出的 SNMP 协议及其相关规范。SNMP 以其简单和实用性已经成为事实上的工业标准, 形成了当前普遍基于 SNMP 协议的网络管理框架。

但是, 由于 SNMP 协议本身固有的问题, 这一框架在信息模型、协议安全性、传输协议等方面有许多无法克服的缺陷^[1]。因此, SNMP 已不能对日益复杂的现代化网络进行有效的管理, 在配置管理方面的缺陷尤为明显, 人们迫切需要新的网络管理模型。

可扩展标记语言(eXtensible Markup Language, XML)的出现为构建新的网络管理模型提供了契机。目前学术界对于 XML 技术在网络管理中应用的研究大致可以分为 3 类。第 1 类是 XML 和 SNMP 集成管理, 如 J.P. Martin-Flatin 提出的 SNMP MIB 到 XML DTD 文档的转换模型^[1], F. Strauss 开发的“libsmi”工具^[2]; Avaya Labs 用来读写 SNMP agent 中的管理信息的 XML 接口^[3]。第 2 类是基于 XML 的管理架构, 如 J.P. Martin-Flatin 在文献[1]中提出的 WIMA 和 Hong 等提出的基于 XML 和 HTTP 协议的 XML-based Management 架构^[4]。第 3 类是采用 Web Services 技术来进行基于 XML 的网络管理, 这方面的代表是 IRTF 的 NMRG 和 OASIS 这两个组织。

总的来讲, 使用 XML 技术来进行网络管理是未来的趋势, 但是在目前 SNMP 设备广泛部署的形势下, 不可能实现纯的基于 XML 的网络管理, 采用 XML/SNMP 网关的方案是既能兼顾当前现实又能获得基于 XML 网络管理的某些好处的选择之一。但是, 综观目前的研究, 一是少有成熟的实现, 二是简单地使用 SNMP SET 来对设备进行配置, 在当前的网络环境下, 无法对设备进行有效配置。

本文提出了笔者的解决方案: 采用一个 XML 到其他协议的转换网关, 既能管理传统的基于 SNMP 代理的网络设备, 又能管理基于 CLI(Telnet/SSH)的设备, 基于这一网关实现了一个既能监控网络又能配置网络的配置管理系统。

2 XML 技术与网络管理

XML^[5]是一种元标记语言, 在 1998 年由 W3C 定义为在 WWW 上进行文档交换的标准。XML 及其相关技术的优点使它非常适合于用来进行大规模网络的管理工作。其特点如下:

(1) 可以采用 XML Schema 或者 DTD 来建立管理信息模型, 具有良好的灵活性和可扩展性, 很好地表达信息之间的关联。

(2) 使用广泛部署的协议, 如 TCP 和 HTTP 来可靠地传输大量数据。

(3) DOM 和 SAX 为 XML 文档的处理提供了标准接口。

(4) XPath 表达式能用于管理信息的过滤。

(5) 能使用 XSLT 对 XML 文档进行处理。

3 基于 XML 的协议转换网关

3.1 架构设计

XML 协议转换网关的主要任务是: 把 HTTP 协议传输的 XML 请求转换为相应的 SNMP 操作或者 CLI 命令, 并把得到的响应也编码成 XML 文档, 通过 HTTP 协议返回给请求者。一方面, 网关接受包含 XPath 表达式的 HTTP GET 请求, 对请求进行解释, 生成 SNMP 请求发送给设备; 在收到响应之后, 应用 XPath 表达式进行过滤, 得到请求者需要的数据。

基金项目: 国家自然科学基金资助项目(60473083); 国家“863”计划基金资助项目(2005AA103110-2)

作者简介: 章 勋(1983-), 男, 硕士研究生, 主研方向: 网络运行与管理; 杨家海, 研究员; 王继龙, 副教授; 张 辉, 助理研究员
收稿日期: 2007-02-18 **E-mail:** zhangxun00@mails.tsinghua.edu.cn

另一方面,请求者以 HTTP POST 方式向网关发送配置请求的 XML 文档,网关将请求翻译 CLI 命令,通过 Telnet/SSH 协议登录到远程设备上执行,并把响应编码成 XML 文档返回给请求者。网关的结构及数据流程如图 1 所示。

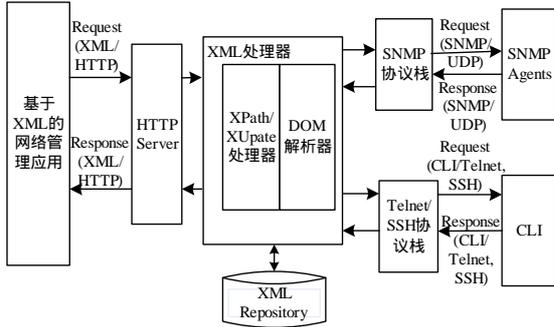


图 1 系统架构及数据流程

3.2 Get 操作的设计与实现

目前,对设备配置和运行信息的获取大部分还是采用 SNMP 方式进行的。因此采用网关中的 XML/SNMP 转换来实现从远程 SNMP 设备获取管理信息。

首先,需要把 SNMP MIB 信息用 XML 格式来进行描述。在 SNMP 协议中,采用的是结构化信息(SMI)来建立管理信息模型;在这里,采用 XML Schema 来定义表示 SNMP MIB 的 XML 文档的结构。SMI 与 XML Schema 的对应关系如下所示:

SMI MIB	XML Schema
MIB 对象(OBJECT-TYPE)	XML Schema 元素(xsd:element)
对象名称	元素名称(name)
对象元素序列(SEQUENCE)	元素子序列(xsd:sequence)
对象索引(INDEX)	元素属性(xsd:attribute)

XML/SNMP 网关实现为一个 Java Servlet,它实现了 Servlet 的 doGet()方法来接受 HTTP GET 类型的请求,工作流程描述如下:

- (1)接受 HTTP GET 请求,其中 Query String 是 s。
- (2)根据请求和 s,创建 GatewayDataRequest 对象 gwRequest。
- (3)对 gwRequest 中的每个 ContextGetRequest 对象 currContextRequest,执行下面的(4)~(6),全部执行完之后执行(7)。
- (4)根据 Schema 库,解析 currContextRequest 对象的 XPath 中请求的 MIB 变量,并得到相应的 OID。
- (5)执行 SNMP 操作(GET, GETNEXT)获取 MIB 变量的值。
- (6)对结果应用 XPath 表达式,过滤得到请求者请求的信息。
- (7)对每个 ContextGetRequest 对象返回的结果,合并到一个 XML 文档中。
- (8)通过 HTTP 协议返回 XML 文档。

例如,要获得某台路由器上所有活动的接口的描述,则向网关发出如下 HTTP 请求:

```
http://127.0.0.1:18080/gateway/servlet/gateway?get=/snmp-data/context[@ipaddr="166.111.200.5" and @community="xxx"]/ifEntry
[ifOperStatus="up" and (ifOutOctets > 0 or ifInOctets > 0)]/ifDescr
```

得到的响应是这样的一个 XML 文档(部分省略):

```
<snmp-data>
<context ipaddr="166.111.200.5" port="161" community="xxx">
<IF-MIB:ifEntry ifIndex="1">
<IF-MIB:ifDescr>TenGigabitEthernet2/1</IF-MIB:ifDescr>
</IF-MIB:ifEntry>
...
</context>
</snmp-data>
```

3.3 Set 操作的设计与实现

出于安全方面的考虑,目前几乎所有网络设备上都禁止了 SNMP Set 功能,管理员一般通过 CLI 方式(Telnet/SSH)登录到远程设备终端执行命令来对设备进行配置。在这里,通过网关中的 XML/Telnet 功能实现对设备的远程配置。

采用以下的形式化语言对命令库 Schema 进行描述,提出一些定义。

定义 1 命令库的 XML 文档是一个九元组:

$$(A_e, A_a, A_n, A_v, subelem, attrs, name, value, e_{commands})$$

其中, A_e 是所有元素的集合; A_a 是所有属性的集合; A_n 是名字的集合,包括元素和属性的名字; A_v 是值的集合,包括元素和属性的值; $subelem$ 是一个二元关系, $subelem \subseteq A_e \times A_e$, 如果 $e_1, e_2 \in A_e$, 那么 $(e_1, e_2) \in subelem$ 表示 e_1 是 e_2 的子元素; $attrs$ 是一个二元关系, $attrs \subseteq A_a \times A_e$, 如果 $a_1 \in A_a, e_1 \in A_e$, 那么 $(a_1, e_1) \in attrs$ 表示 a_1 是 e_1 的属性; $name$ 是一个二元关系, $name \subseteq A_n \times (A_e \cup A_a)$, 如果 $n_1 \in A_n, i_1 \in A_e \cup A_a$, 那么 $(n_1, i_1) \in name$ 表示 n_1 是 i_1 的名称; $value$ 是一个二元关系, $value \subseteq A_v \times (A_e \cup A_a)$, 如果 $v_1 \in A_v, i_1 \in A_e \cup A_a$, 那么 $(v_1, i_1) \in value$ 表示 n_1 是 i_1 的值; $e_{commands}$ 是命令库的 XML 文档的根元素,其中, $e_{commands} \in A_e, commands \in A_n, (commands, e_{commands}) \in name$ 。

定义 2 值的集合 $A_v = A_{cv} \cup A_{vv}$, A_{cv} 是常量的集合, A_{vv} 是变量的集合。

定义 3 E_x 表示名字为 x 的元素的集合,其中 $e \in A_e, E_x = \{e | (x, e) \in name\}$ 。

定义 4 $ATTR_x$ 表示名字为 x 的属性集合,其中 $a \in A_a, ATTR_x = \{a | (x, a) \in name\}$ 。

而且约定: e_x 是名称为 x 的一个元素,其中 $x \in V_n, e_x \in E_x; a_x$ 是名称为 x 的一个属性,其中 $x \in V_n, a_x \in ATTR_x$ 。

那么命令库 Schema 中元素和属性之间的关系为

$$(E_{Command}, e_{commands}) \subset subelem$$

$$(\{e_{processes}, e_{params}\}, e_{Command}) \subset subelem$$

$$(E_{process}, e_{processes}) \subset subelem$$

$$(E_{param}, e_{params}) \subset subelem$$

$$(\{a_{name}, a_{os}\}, e_{command}) \subset attrs$$

$$(a_{cmd}, e_{process}) \subset attrs$$

$$(\{a_{name}, a_{defaultvalue}\}, e_{param}) \subset attrs$$

此外,对于变量的集合 A_{vv} ,只规定了格式,没有规定其类型和值。用户必须在 e_{params} 中定义变量。笔者规定:如果变量 $v_1 \in A_{vv}$,那么 v_1 形式必须是 "%...%", v_1 需要在执行命令前被赋值。

例如，要实现在 Cisco IOS 路由器上修改 hostname 的配置功能，在命令库中应该包含这样的语句：

```
<command name="setHostname" os="Cisco IOS">
  <processes>
    <process cmd="hostname %newname%"/>
  </processes>
  <params>
    <param name="newname" defaultvalue="router"/>
  </params>
</command>
```

管理程序需要调用这条操作指令时，把包含如下内容的 XML 文档以 POST 方式发送给网关：

```
<request>
  <operation name="setHostname" os="Cisco IOS">
    <parameter name="newname" value="THURouter"/>
  </operation>
</request>
```

3.4 性能分析

通过协议转换网关访问设备上的 MIB 变量，以及直接通过 SNMP 协议访问相同的 MIB 变量，测量两种方式下的响应时间与网络流量，并进行比较，如表 1 所示。

表 1 响应时间与流量比较

方式	响应时间/ms		网络流量/B	
	SNMP	Gateway	SNMP	Gateway
system Group	163	176	653	1 083
interface Group	6 438	6 985	38 089	44 166

从表 1 可以看出，通过协议转换网关获取 MIB 变量的响应时间稍长，这是因为网关要把收到的 XML 请求转换成 SNMP 协议的请求，接收到 SNMP 响应后还要编码成 XML 文档，需要额外的时间进行处理，这是难以避免的；而且通过协议转换网关获取 MIB 变量时产生的网络流量也要更大，这主要是因为 XML 标记产生的。事实上，如果采用 HTTP 协议的 gzip 对 XML 文档进行压缩后传输，可以大大减少网络流量，不过这样做的代价是响应时间会有所增加。

4 配置管理系统

在实现了上述协议转换网关的基础上，设计和实现了一个大规模网络配置管理系统。该系统管理的对象是部署了 SNMP Agent 以及配置了基于 Telnet/SSH 协议的 CLI 的网络设备。与传统的网络配置管理系统相比，该系统不仅能获取网络配置，监控网络运行，还能对网络进行配置，不会受到设备禁用 SNMP SET 这一现状的限制。系统架构如图 2 所示。

用户发出获取配置信息的请求时，系统根据请求的 MIB 变量生成 XPath 表达式，并通过 Get 方式访问网关的 Servlet；从网关获取 XML 格式文档的响应后，对其采用 DOM 进行解析获取 MIB 变量的值。用户发出更改设备配置请求时，系统根据发出的请求生成格式化的 XML 文档，并通过 Post 方式发送给网关；从网关获取到执行结果的 XML 文档后，把结果展示给用户界面。

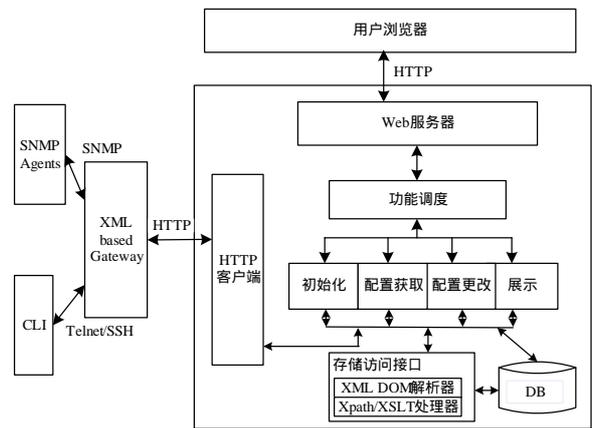


图 2 配置管理系统架构

系统中的设备配置管理主页面集成了对设备进行配置管理的主要功能和常用操作。页面的上半部展现该设备的基本信息，下半部是该设备的端口信息列表。此外，在页面中可以选择要对设备进行的配置操作，如修改 Hostname、修改 Banner、配置 ACL 以及执行命令等；也可以对某个端口执行配置操作，如修改端口的描述、配置端口 IP 地址以及禁用和启用端口等；还可以选择更新或者删除设备的配置信息。

5 结束语

本文设计和实现了一个基于 XML 的多协议转换网关，通过它可以对现在广泛部署的 SNMP 协议设备以及基于 CLI 的网络设备进行有效管理。并在此基础上，实现了一个大规模网络配置管理系统，与传统的配置管理系统相比，该系统不仅能获取设备配置信息和运行状态，还能有效地对设备进行配置。目前该系统已经试用于中国下一代教育和科研计算机网 CERNET2 主干网的实际网络管理中。

下一步考虑在协议转换网关中监听 SNMP Trap 消息，并格式化为 XML 文档发送给网管应用程序，基于此可以开发更为丰富的网管应用。另外，对系统性能(响应时间，网络流量等)的改进也是工作的方向。

参考文献

- [1] Flatin J P M. Web-based Management of IP Networks and Systems[D]. Lausanne: Swiss Federal Institute of Technology, 2000.
- [2] Strauss F. A Library to Access SMI MIB Information[EB/OL]. [2006-12-10]. <http://www.ibr.cs.tu-bs.de/projects/libsmi/>.
- [3] Avaya Labs Research. XML-based Management Interface for SNMP Enabled Devices[EB/OL]. (2001-09-08). <http://www.research.avayalabs.com/user/mazum/Projects/XML/>.
- [4] Ju H, Choi M, Han S, et al. An Embedded Web Server Architecture for XML-Based Network Management[C]//Proc. of IEEE/IFIP Network Operations and Management Symposium. [S. l.]: IEEE Press, 2002.
- [5] Bray T, Paoli J, McQueen C M S, et al. Extensible Markup Language (XML) 1.0 (Second Edition)[R]. W3C Recommendation, 2000-10.