

基于 SNMP++ 类库的简单网络管理平台的实现

周志成, 王 卓, 汪秉文, 秦肖臻

(华中科技大学 控制科学与工程系, 湖北 武汉 430074)

摘 要: 随着网络的发展, 对网络进行的监控和维护等管理操作也变得更加困难。简单网络管理协议(SNMP)是能够满足网络管理需要的协议。SNMP++ 是 HP 公司开发的一套用于 SNMP 编程的 C++ 库, 它把 SNMP 编程中的相关部分都面向对象化, 从而大大简化了 SNMP 网络管理软件开发。文中先介绍了 SNMP 的概念及框架结构, 列出了与 SNMP 密切相关的管理信息库(MIB)、消息传输机制等, 并配以图例; 接着对 HP 公司所提供的 SNMP++ 类库中的常用类与成员函数作简单介绍。最后讨论了怎样在 VC++ 6.0 平台下利用 SNMP++ 类库开发一个简单的网络管理平台。

关键词: 网络监控; 管理信息库; SNMP++ 类库

中图分类号: TP393.07

文献标识码: A

文章编号: 1005-3751(2006)03-0158-03

Implementation of SNMP++ Based Simple Network Management Platform

ZHOU Zhi-cheng, WANG Zhuo, WANG Bing-wen, QIN Xiao-zhen

(Dept. of Control Science and Techn., Huazhong Univ. of Science and Techn., Wuhan 430074, China)

Abstract: With the development of network, it is more difficult to manage the network. Simple network management protocol(SNMP) is the suitable protocol for the need of network management. SNMP++ is a set of C++ class libraries supplied by HP company. It can extremely simplify the development of network management software. First, this paper introduced the definition and framework of SNMP with graphs. Then, it illuminated common classes and member functions of SNMP++ class libraries. In the end of the thesis, it discussed how to implement an SNMP++ based simple network management platform with the IDE of VC++ 6.0.

Key words: network management; MIB; SNMP++ class library

0 引言

简单网络管理协议 SNMP(Simple Network Management Protocol)是由 IETF(Internet Engineering TaskForce)研发的^[1], 是目前 TCP/IP 网络中应用最为广泛的网络管理协议, 它采用轮询机制, 提供最基本的功能集, 最适合小型、快速、低价格的环境使用, 受到许多产品的广泛支持, 因此成为被广泛接受并投入使用的工业标准^[2]。关于 SNMP 的讨论前人已做了很多努力, 文中主要讨论如何利用 HP 公司所提供的 SNMP++ 类库实现简单的网络管理平台。

1 简单网络管理协议

图 1 是使用 SNMP 的典型配置。整个系统必须有一个管理站(management station)、多个代理站(agent station)

及代理站所维护的管理信息库 MIB(Management Information Base), 管理站与代理站之间通过 SNMP 协议进行消息传输。

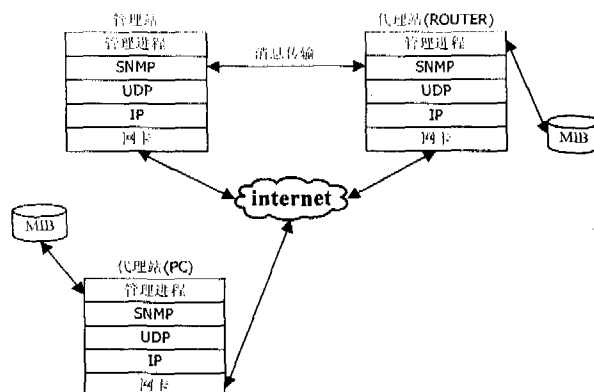


图 1 使用 SNMP 的典型配置

下面简要介绍这几个部分。

1.1 管理站和代理站

在 RFC1157 中描述了 SNMP 所用的是 C/S 模型, 它分别对应着管理站和代理站^[3]。管理站可以是一般台式机也可以是高性能的服务器, 在管理站内运行着管理进程; 代理站为被管理对象, 一般为路由器、交换机等网络设

收稿日期: 2005-06-14

作者简介: 周志成(1973—), 男, 湖南宁乡人, 硕士研究生, 研究方向为网络控制与管理; 汪秉文, 教授, 博士生导师, 研究方向为生产过程综合自动化系统集成优化与决策, 人工智能、专家系统及其应用, 系统故障检测与诊断。

备,也可以是台式机,在代理站内运行着代理进程。要使这种 C/S 模式能够正常工作,管理进程和代理进程必须使用相同的语言(SNMP 协议)^[4]。

1.2 管理信息库

管理信息库(MIB)本质上就是由代理进程维护的一种树形结构的数据库,它保存着能够被管理进程查询和设置的信息。MIB 给出了一个网络中所有可能的被管理对象的集合的数据结构。SNMP 的管理信息库采用和域名系统 DNS 相似的树型结构,它的根在最上面,根没有名字。图 2 是管理信息库的一部分,它又称为对象命名树(object naming tree)。树的节点表示管理对象,它可以用从根开始的一条路径来无二义地识别。

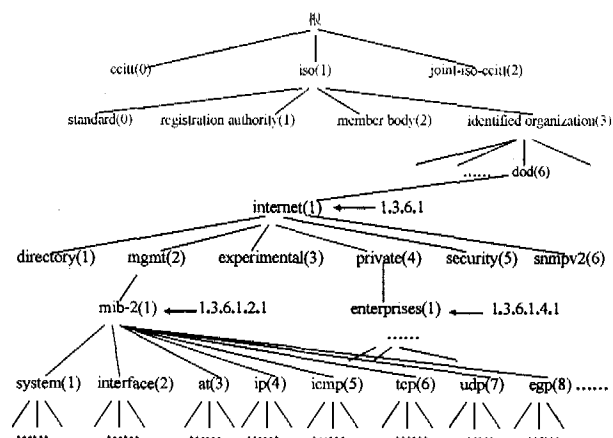


图 2 MIB 树形结构

例如:OID{1.3.6.1.2.1.1}就代表了结点 mib-2 的子对象 system(标号 1),这个对象内保存着与网络设备有关的系统信息^[1]。

事实上,基于 SNMP 协议的网络管理客户软件所要做的就是从 MIB 得到和设定特定对象的信息。例如:如果想计算路由器网络流量就可以访问 MIB 中的 ifOutOctets (16)对象,OID 为{1.3.6.1.2.1.2.2.1.16},该对象保存了一段时间内流出路由器某端口的数据量。同时也可以访问 ifInOctets (10)对象,OID 为{1.3.6.1.2.1.2.2.1.10},该对象保存了一段时间内流入路由器某端口的数据量。

1.3 消息传输

SNMP 规定了 5 种协议数据单元 PDU(也就是 SNMP 报文),用来在管理进程和代理进程之间的交换^[1],如图 3 所示。

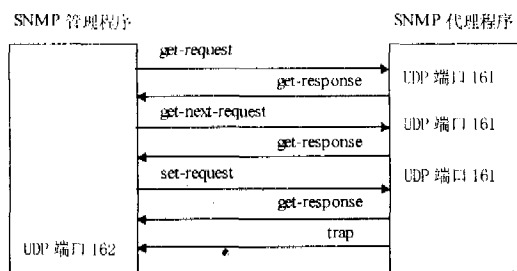


图 3 SNMP 的 5 种报文操作

1) get-request 操作:从代理进程处提取一个或多个对象值。

2) get-next-request 操作:从代理进程处提取紧跟当前对象的下一个对象值。

3) set-request 操作:设置代理进程的一个或多个对象值。

4) get-response 操作:返回的一个或多个对象值。这个操作是由代理进程发出的,它是前面三种操作的响应操作。

5) trap 操作:代理进程主动发出的报文,通知管理进程有某些事情发生。

图 4 是封装成 UDP 数据报的 5 种操作的 SNMP 报文格式。可见一个 SNMP 报文共由 3 个部分组成,即公共 SNMP 首部、get/set 首部、变量绑定。UDP 数据报封装好后还要添加 UDP 首部封装成 UDP 数据报,再添加 IP 首部封装成 IP 数据报。

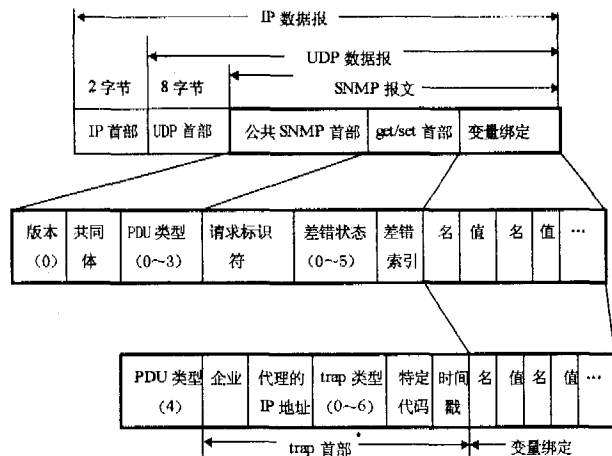


图 4 SNMP 报文格式

HP 公司所提供的 SNMP++ 类库对这 5 种协议数据单元 PDU 做了很好的封装,并提供源码。所以文中 SNMP 报文的这 3 个部分不做详细解释,如想了解该部分内容可以查阅相关资料。

2 实现简单网络管理平台

2.1 SNMP++ 类库简介

SNMP++ 是 HP 公司开发的一套用于 SNMP 编程的 C++ 库,它把 SNMP 编程中的相关部分都面向对象化,因此读者不必了解 SNMP 底层的机制,从而大大简化了 SNMP 网络管理软件开发复杂性和开发难度,也缩短了开发时间^[5]。同时,它是免费的且源代码公开,下载地址: <http://www.agentpp.com/>。

下面对 SNMP++ 中的常用类与成员函数做简单介绍:

1) Address 类:各种地址类的基类,能过各类的构造函数可以生成不同的地址对象。如 IP 地址、UDP 地址。文中下面的例程将用到 UdpAddress 对象及其成员函数 valid()、set_port()。

2) Oid 类:对象标识符 OID(object identifier)类的封装。通过其构造函数和成员函数可以定义和操作 OID 对象。文中下面的例程将用到 OID 对象和 valid() 成员函数。

3) Vb 类:SNMP 变量绑定的封装。在 SNMP++ 中,如图 4 中所示,SNMP 报文中的变量绑定列表可以用 Vb 对象数组表示。Vb 对象可以传递给下面将要介绍的 Pdu 对象从而设置管理信息库 MIB 中的管理对象的值。也可以从 Pdu 对象中取出 Vb 对象而得到 MIB 中的管理对象的值。

文中下面的例程将用到成员函数: set_oid()、get_printable_oid()、get_printable_value()。

4) Pdu 类:SNMP++ 类库对 5 种协议数据单元 PDU 做的封装。文中下面的例程将用到成员函数 get_vb() 和重载后的运算符“+ =”。

5) SnmpTarget 类:抽象目标类。在 SNMP++ 中, SnmpTarget 有两个子类:CTarget(基于共同体的目标类,用于 SNMPv1 和 SNMPv2)和 UTarget(基于用户的目标类,用于 SNMPv3)。文中下面的例程将用到 CTarget 类及其成员函数: set_version()、set_retry()、set_timeout()、set_readcommunity()。

6) Snmp 类:SNMP 类是 SNMP sessions, gets, sets 和 get nexts 的封装。这个类管理 SNMP 的所有资源并提供完整的重试和超时处理机制。文中下面的例程将用到成员函数: socket_startup()、get()、socket_cleanup()。

2.2 VC6.0 编程示例

在编程前最好将下载的源文件先编译生成静态链接库。然后,记住用 #include "snmp_pp/snmp_pp.h" 包含 snmp_pp.h 头文件并将生成的静态链接库加入你的工程中。

下面介绍利用 SNMP++ 实现从路由器中得到 MIB 信息的工作流程:

1)调用 Snmp::socket_startup() //初始化 socket 子系统。

2)创建所需的对象并初始化:

```
UdpAddress address("211.69.205.126") //创建 Udp 地址对象
address.set_port(161) //初始化 Udp 地址对象的端口号
Oid oid("1.3.6.1.2.1.1.1.0") //创建 Oid 对象
Pdu pdu //创建 Pdu 对象
Vb vb; //创建 Vb 对象
vb.set_oid() //将 Oid 对象绑定到 Vb 对象上
pdu += vb //再将 Vb 对象添加到 Pdu 对象的末尾
CTarget ctarget(address) //利用已创建的 address 创建 CTarget 对象
ctarget.set_version(); //设置 CTarget 对象的 SNMP 版本
ctarget.set_retry(); //设置 CTarget 对象的重试次数
```

```
ctarget.set_timeout(); //设置 CTarget 对象的超时时间
```

```
ctarget.set_readcommunity() //设置 CTarget 对象的共同体
```

```
SnmpTarget * target //创建 SnmpTarget 对象指针
```

```
target = &ctarget //将 SnmpTarget 对象指针指向 CTarget 对象
```

```
Snmp snmp(...) //创建 Snmp 对象
```

3)调用 snmp.get(pdu, *target) //向代理站的代理进程请求 MIB 信息。正常情况下,代理进程将发送一个 GetResponse 报文,该返回的 PDU 中将包含变量名和相应值的对偶表。

4)调用 pdu.get_vb(vb,0) //从返回的 PDU 中析出 VB 对象

vb.get_printable_oid() //得到 VB 对象中绑定的 Oid 对象

vb.get_printable_value() //得到 VB 对象中绑定的 Oid 对象的值

程序的运行效果如图 5 所示。

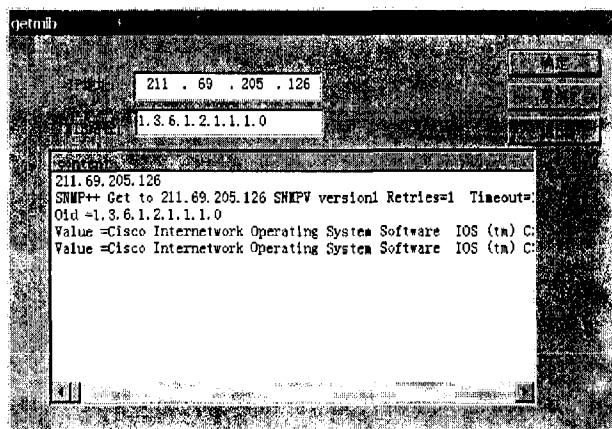


图 5 getmib 程序效果图

3 结束语

主要针对利用 HP 公司所提供的 SNMP++ 类库实现简单的网络管理平台,文中所论述的是笔者写代码的一些经验和体会,在网管程序的具体应用中运用上述知识并实现了预期功能。

参考文献:

- [1] 谢希仁. 计算机网络(第 4 版)[M]. 北京:电子工业出版社,2003.
- [2] Zeltserman D. SNMPv3 与网络管理[M]. 潇湘工作室译. 北京:人民邮电出版社,2000.
- [3] 尹高嵩,范辉. Web 负载测试的研究与应用[J]. 微机发展,2004,14(10):94-95.
- [4] Hamedy S. 简单网络管理协议教程[M]. 胡谷雨,张巍,倪桂强,赖蓉,等译. 北京:电子工业出版社,1999.
- [5] 李向丽,穆玲玲. 基于 IPv6 的 RIPng 路由技术研究[J]. 微机发展,2004,14(10):100-102.