

基于 ZebOS 的分布式网管软件设计

张增华，张 刚

（中国电力科学研究院，江苏 南京 210003）

摘 要：本文以一种大容量的 OLT 设备网管软件开发为背景，分析了 Ip Infusion 公司的协议软件 ZebOS 的框架结构以及框架内各个模块的通信机制，结合当前主流的开源网管软件 NetSNMP，采用了 Master-Subagent（主代理和子代理方式），实现了 ZebOS 与 NetSNMP 的集成与移植。本网管软件设计，将子代理分布在 ZebOS 模块内部，提高了系统的可扩展性和程序部署的灵活性，总体上降低了系统内部各模块的通信的复杂度，提高了程序可靠性。本文给出的基于主代理和子代理结合的方式，对其它类似的大型设备网管软件开发以及现有设备网管软件开发及改造有一定的参考意义。

关键词：ZebOS；NetSNMP；主代理；子代理；设备网管

0 前言

大型设备开发过程中，系统通常由多个软件模块组成，包括操作系统，功能模块，协议模块等各个功能模块通常分布在多个进程中，设备管理统一的接口管理。SNMP作为一个从网络上的设备收集管理信息的公用通信协议，包括监视网络性能、检测分析网络差错和配置网络设备等。在网络正常工作时，SNMP可实现统计、配置和测试等功能。当网络出故障时，可实现各种差错检测和恢复功能。目前，几乎所有的网络设备生产厂家都实现了对SNMP的支持。在小型网络中，SNMP比较适合进行网络监控；而在大型网络中，SNMP也是一个有效的网络配置工具。可以通过SNMP工具进行设备的管理与配置，对接口的统计和性能度量，对接口链路的状态进行追踪等。

Ip Infusion公司的ZebOS协议包提供高性能，模块化的二、三层交换和路由软件方案，支持 VxWorks, Linux等各种操作系统，在业界有广泛的应用。该协议中间件各个模块独立，可以拆分使用，用户可以根据要求实现各个功能模块的自由组合。在ZebOS协议中间件包中，虽然提供了SNMP管理软件的接口，也提供了部分模块的MIB库设计，本次OLT设备的开发由于涉及到自定义的扩展模块比较多，如系统的环境参数的管理，ONU管理等，如果沿用ZebOS的原有的AagentX设计，总体上扩展改动比较大，定制性不强。因此，在本次大容量OLT设备开发中，采用Master-SubAgent的设计方式，各功能MIB库统一设计，为系统的一致性和程序的可扩展性提供更便利的接口。

1 ZebOS 框架特点

ZebOS模块化，多进程体系架构，每一个协议可以单独使用，可以很容易的被移植和应用大型分布式系统，路由协议的计算可以由多个CPU或板卡来共同完成。这种分布式，多进程的体系结构能大幅提高网络设备整体的稳定性和可靠性。该类型体系结构为分布式网管提供了思路：清晰接口规范定义，明确层次功能划分，使用户的开发人员可以轻松的把单个或多个协议模块顺利的集成于用户自己的软件体系中去。从而大幅提高网络设备软件研发的效率。ZebOS的软件管理平面系统架构如图1所示。

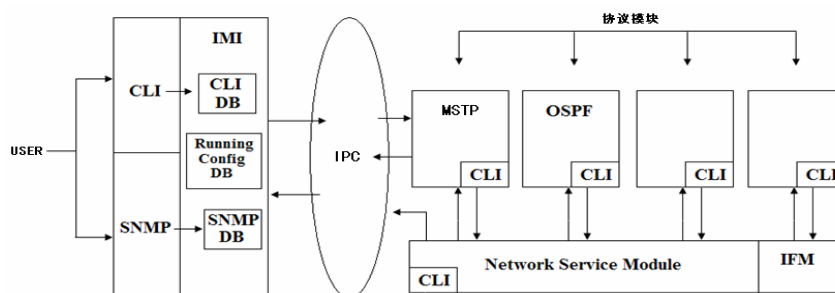


图1 ZebOS软件管理平面架构

2 总体设计思路

该背景项目大容量OLT设备的研发，硬件设计上共14个槽位，每个槽位对应一个主控板或业务板卡。每块板卡上都有独立的操作系统，本次设计及考虑当前的功能需求，也要考虑了以后的功能添加扩展。各个子代理模块分散在各个协议模块或操作系统进程内部。结合ZebOS系统后的SNMP结构设计如图2所示。

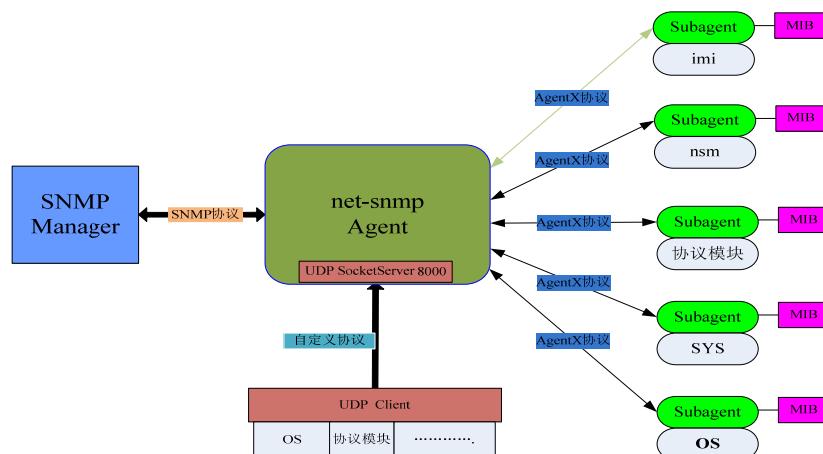


图2 SNMP总体设计结构图

本系统设计特点及本设计有优势如下：

SNMP实现采用Master-SubAgent的模式，该模式包含一个主代理进程和多个子代理进程，子代理进程驻留在主控板中的ZebOS各个协议栈模块中。与ZebOS协议无关子代理如系统环境参数编写单独进程实现数据的读写。

Master和SubAgent之间通过标准接口AgentX进行通信。主代理进程处理SNMP协议，每个子代理进程负责一个特定的MIB视图。主代理进程启动后，负责与SNMP Manager通信，每个的SubAgent启动时在Master中动态注册与注销。通过主代理，该过程对于SNMP管理应用程序是透明的。Master在编译时候仅需要编译基本的MIB库，用户自定义的MIB库随子代理进程发布。

每个协议模块中的Trap信息由于是单向的数据传输，本系统中采用UDP Socket方式实现，即：在主代理中开启一个UDP Socket Server，接收来自各个协议模块和其它与协议模块无关的应用程序的Trap信息。

各子代理将它们的职责信息通告给主代理。每个符合AgentX的子代理运行在各自的进程空间里，因此比采用单个完整的SNMP代理具有更好的稳定性。降低了主代理程序的复杂性。

通过 AgentX协议能够访问它们的内部状态，进而管理站随后也能通过SNMP访问到它们。随着服务器进程和应用程序处理的日益复杂，通过AgentX技术，我们可以利用标准的SNMP管理工具来管理大型软件系统。

3 SNMP 设备网管设计

系统移植与集成主要分为四步：第一步需要将本开源的软件移植到目标板，即将该软件编译后的目标程序在PowerPC处理器上运行；第二步需要将各个协议添加子代理的支持。第三步：实现各个协议模块的内部数据读取，即各个协议模块内部的运行参数和性能参数读取和设置。第四步：告警和事件数据的上报。

对于SNMP通信协议的介绍和MIB库设计，相关的介绍资料比较多，在本文中不做描述。本文主要对该设备网管与ZebOS系统集成、数据采集、告警处理、系统配置这几点进行归类，将系统功能分布在系统对应的功能模块或进程。见表1。

表1 SNMP设备网管设计

| ZebOS 模块 | Authd | imi | nsm | epond | Lacpd |
|----------|----------|--------|-------------|--------------|-------------|
| SNMP 功能 | 1.802.1x | 2.系统参数 | 1.板卡管理 | 1.Pon 相关功能管理 | 1.lacp 协议管理 |
| | | | 2.端口管理 | | |
| | | | 3.vlan 管理 | | |
| | | | 4.mac 管理 | | |
| | | | 5.ntp 管理 | | 2.静态链路聚合 |
| | | | 6.告警管理 | | |
| | | | 7.风暴抑制配置 | | |
| | | | 8.端口流量统计 | | |
| | | | 9.IGMP 协议管理 | | |
| | | | 10.端口镜像管理 | | |

3.1 主代理设计

主代理是一个在可执行SNMP的网络元件上运作的软件，可响应从管理站发出的SNMP要求。它的角色类似客户端/服务器结构（Client/Server）术语中的服务器。主代理依赖子代理提供有关特定功能的管理信息。如果系统当前拥有多个可管理的子系统，主代理就会传递它从一个或多个子代理处收到的请求。

本系统设计中，主代理的主要功能有如下几点：

收集子代理的注册信息。子代理模块启动后，即将当前的OID信息到主代理程序，服务器上注册。见图3。

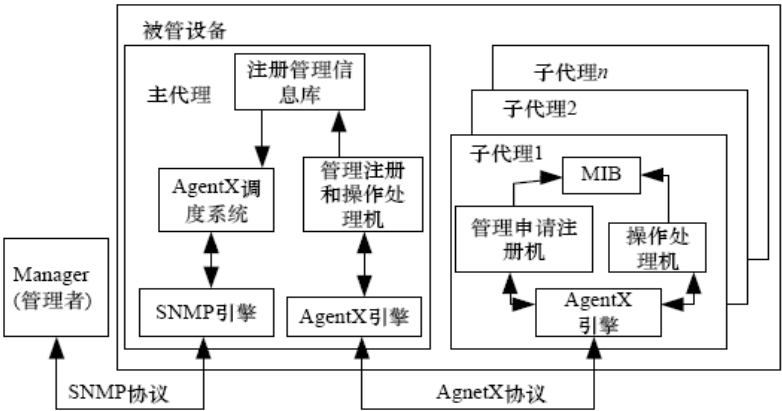
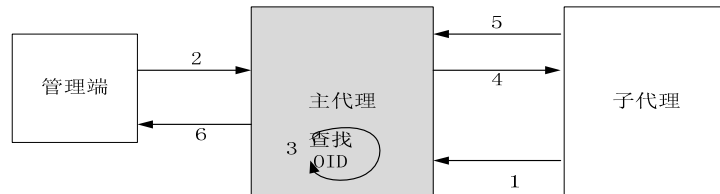


图3 内部通信机制1

主代理响应管理者请求。当一个管理者发送请求后，主代理接收到该信息，并根据请求的OID，将相应的请求发送到对应的子代理模块，子代理模块相应请求并返回结果。如该OID在主代理中未完成注册，主代理返回提示。见图4。



(1) 子代理发起的OID注册

(2) 管理端发起请求

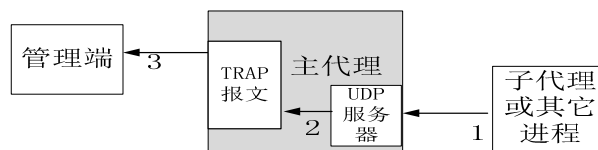
(3) 主代理查找OID所在子代理，将该请求转发给对应的子代理，如没有找到该对应的OID，直接由主代理返回给管理端。

(4) 主代理将该请求转发给子代理。

(5) 子代理返回处理结果给主代理

(6) 主代理将结果发挥给管理端。

主代理转发子代理的告警和事件信息：对于设备内部的告警或事件信息，主代理建立UDP服务器，接收各个子代理模块发送过来的告警或事件信息，主代理接收到信息后，统一标准的格式发送到管理端。见图5。



(1) 子代理或其它进程监听告警或系统事件，发送到主代理的UDP服务器。

(2) UDP服务器接收到告警或事件后，统一格式转发给上层的管理端。

(3) 管理端接收到事件或告警。

3.2 协议模块子代理集成设计

子代理是一个在可执行SNMP的网络元件上运作的软件，执行在特定子系统的特定管理信息库（MIB， Management Information Base）中定义的资讯和管理功能。子代理的一些能力有：

- 1) 搜集主代理的资讯；
- 2) 配置主代理的参数；
- 3) 回应管理者的要求；
- 4) 产生警告或事件。

子代理的设计主要是完成SNMP和各个进程模块的集成和接口调用。

(1) 子代理的集成设计：

- 1) 在编译平台选项中，加入对应的net-snmp依赖库支持：

```
LDLIBS_FLAGS=$(LIBS) $(LD_PATH) `net-snmp-config --agent-libs`
```

- 2) 在每个模块内部，添加根据MIB生成的.h和.c文件

在每个模块内部，以MSTP为例，添加如下函数，启动线程：

```
err = pthread_create(&agtThreadID_mstp,NULL,startMstpAgent,NULL);
```

```
...
```

```

int    startMSTPSubAgent (struct   smiclient_globals * azg)
{
...
    /* initialize mib code here */
    init_l2MSTP();    //启动MSTP模块
...
}

```

调用图见图6。

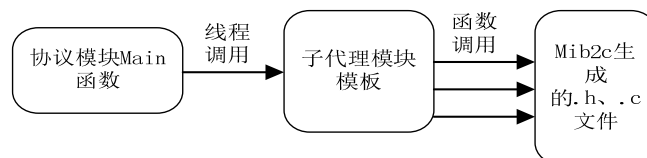


图6 子代理模块集成

3.3 ZebOS 接口设计及调用

ZebOS的接口提供给子代理调用，主要负责ZebOS和子代理之间的数据交互。在MIB中的每个OID节点对应一个函数调用。接口设计的关键是对ZebOS 的处理流程的梳理以及HSL部分接受到数据后的处理流程，如下一个完整的函数调用流程见图7。

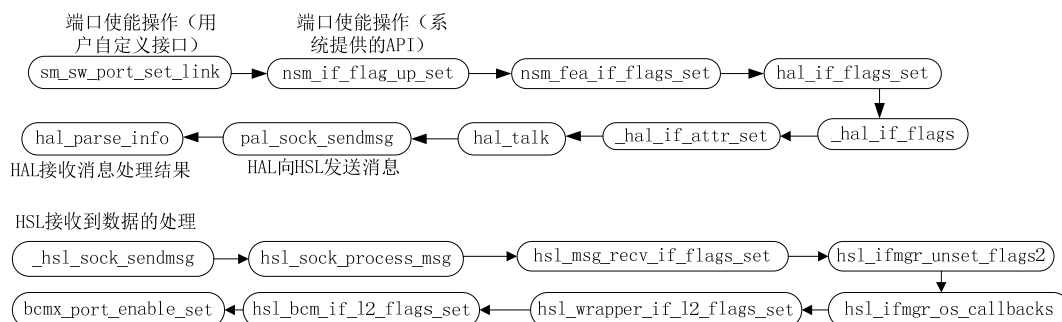


图7 数据处理流程

3.4 告警数据的处理

告警和事件数据的处理，本系统中采用统一出口，各子代理或进程模块在事件或告警点，采用自定义的数据格式，将数据统一汇总到主代理端，在主代理中开配UDP Server，收集各种告警或事件信息，将告警或事件信息归类统一化，由主代理Trap给管理端。

4 结论

大容量OLT设备的分布式代理设计，结合料OLT中的协议中间件ZebOS和系统部署特点，设计出了基于分布式系统的网管系统，该系统已经在项目中得到具体应用，对复杂多变的系统下的网管设计提供了一种解决思路，采用接口设计的思路，对现有的其他类似系统进行SNMP功能添加改造有一定的参考意义，该方法对原有的系统影响最小，对系统的性能要求也比较不高。该设计方法对其它类似的系统有一定的参考意义。

参考文献：

- [1] 姜飞,史浩山,徐志燕,等.一种基于 SNMP 协议的主代理子代理通信机制[J].计算机工程, 2007(11):81-83.
- [2] 岑贤道,安常青. 网络管理协议及应用开发[M].北京:清华大学出版社,1998.

- [3] 石硕,林莉.交换机/路由器及其配置[M].北京:电子工业出版社,2005.
- [4] 李明江.SNMP 简单网络管理协[M].北京:电子工业出版社,2007.
- [5] IP Infusion 《ZebOS Architecture Developer Guide》 November 2011.
- [6] <http://www.net-snmp.org/wiki/index.php/Tutorials>.

作者简介:

张增华（1976—），男，湖北黄冈人，工程师，硕士，主要研究嵌入式软件开发，资源管理系统开发；

张 刚（1966—），男，安徽濉溪人，高级工程师，硕士，主要负责电力通信光缆网相关应用系统的开发。