

doi: 10.3969/j.issn.1006-1576.2011.11.025

## 预定义被管对象层次结构实现分层式运管系统

赵建平<sup>1</sup>, 赵建辉<sup>2</sup>, 刘晓阳<sup>1</sup>, 薛倡新<sup>1</sup>, 段慧芬<sup>1</sup>

(1. 中国卫星海上测控部技术部, 江苏 江阴 214431; 2. 中国人民解放军 63999 部队, 北京 100094)

**摘要:** 针对基于集中式思想请求应答管理模式的 SNMP 协议无法适应分层式的航天测控网络运行管理体系结构要求的问题, 提出预定义被管对象层次结构的方法。通过建立航天测控系统层次定义模型和统一的航天测控网管理信息结构, 利用软件实现对 SNMP 协议的管理站通知报文 (GetInformRequest) 的扩展, 该扩展报文携带了预定义层次结构, 解决了 SNMP 协议不适应分层式体系结构的问题。结果表明: 该方法实现了航天测控运管系统的三层管理结构, 使系统的配置工作简化、灵活, 满足了航天测控网对网络安全的需求。

**关键词:** 简单网络管理协议; 网络运行管理系统; 分层式体系结构; 航天测控网; 管理站; 管理代理; 被管对象

中图分类号: TJ86 文献标志码: A

## Implementation of Layer-Type Network Running Management System Based on Predefining Managed Target Hierarchical Structure

Zhao Jianping<sup>1</sup>, Zhao Jianhui<sup>2</sup>, Liu Xiaoyang<sup>1</sup>, Xue Changxin<sup>1</sup>, Duan Huifen<sup>1</sup>

(1. Technology Department, Satellite Maritime Tracking &amp; Controlling Department of China, Jiangyin 214431, China; 2. No. 63999 Unit of PLA, Beijing 100094, China)

**Abstract:** For simple network management protocol (SNMP) not adapt to the requirements of layer-type network running management architecture because of its centralized-thought request reply management approach, the paper proposes the method of predefining managed target hierarchical structure. By building the hierarchical defining model and the uniform managerial information structure of space instrumentation and command network, expanding SNMP managerial station GetInformRequest with the information of predefined hierarchical structure is realized by the software to solve the problem of SNMP not to adapt to the hierarchical architecture. The result shows that the method implements the three-layered management architecture of space instrumentation and command network running management system, and enables the system configuration simple and flexible which meets the requirements of network security of space instrumentation and command network.

**Keywords:** simple network management protocol; network running management system; hierarchical architecture; space instrumentation and command network; managerial station; managerial agent; managed target

### 0 引言

航天测控网是一个由测控中心(一、二级)、测控站多级子网构成的广域网, 其网络管理系统体系结构是分层式的, 需建立三级管理节点: 一级运管中心、二级运管节点和运管站。其中一级运管中心在网络管理体系结构中处于最高层, 是航天测控运管系统的最高管理机构, 负责整个网络的网络管理事务。一级运管中心管辖二级运管中心, 二级运管中心是航天测控网系统中层管理机构, 是某个区域或部门的最高管理机构, 负责该区域或部门的网络管理事务。二级运管中心管理其下的运管站, 运管站处于网络管理体系的最底层, 是基层管理站点<sup>[1]</sup>。

简单网络管理协议 (simple network management protocol, SNMP) 是目前广泛使用的一种网络管理协议, 因其易于实现、便于扩充等优点, 得到了众

多厂家支持和广泛应用, 已成为网络管理领域中事实上的工业标准。SNMP 协议使用管理者和被管理者的概念, TCP/IP 网际互连中的管理就是通过设置和检查这些变量来实现的<sup>[2]</sup>。被管代理采用 SNMP V2 管理协议, 提供标准 SNMP 接口, 开放 UDP161 端口监听管理者进程发送的管理信息查询请求消息, 开放 UDP 162 端口发送的异常事件报告陷阱消息 (TRAP)<sup>[3]</sup>。

但 SNMP 协议是基于集中式思想的请求应答管理模式, 无法适应航天测控网分层式的网络管理体系结构的要求。因此, 笔者提出了预定义被管对象层次结构的方法来解决这一问题, 其核心思想是扩展 SNMP 协议的管理站通知报文 (GetInformRequest), 使之携带预定义的层次结构信息, 各层管理站将其与本地定义的层次结构进行对比, 根据比对的结果决定该报文的下一跳方向。从

收稿日期: 2011-07-01; 修回日期: 2011-08-12

作者简介: 赵建平(1974—), 女, 福建人, 硕士, 工程师, 从事软件工程及计算机应用研究。

而使 SNMP 协议扩展适应航天测控网运管系统的三层管理结构的需求。

## 1 模型的建立

### 1.1 层次定义模型

建立层次定义模型的目的是通过这个模型描述出整个网络系统的层次结构。由于层次模型通过全网唯一标识的被管对象标识进行定义, 因此称这种方法为被管对象标识法。其原理是为所有纳入运管

系统的被管对象定义包含层次结构的信息, 通过对这个层次结构信息的处理以达到控制信息的流向、实现分层模式的目的。由层次定义模型建立的层次结构信息, 全网统一制定, 最高级别的管理节点(一级运管中心)保存其全部定义, 其它各层的管理者根据其在层次结构中所处位置, 向下保存其中的一个子集。被管对象是一个广义的概念, 包括具有管理者功能的管理站和具有代理功能的被管设备。航天测控运管系统层次定义模型及结构见图 1。

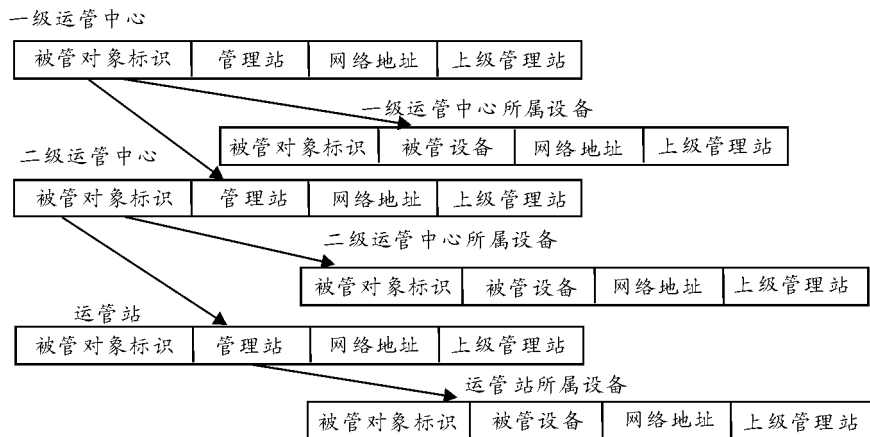


图 1 航天测控运管系统层次定义模型及结构

这是一个可以无限延伸的层次结构, 在实际应用中可以根据需要任意增加层次, 而不必拘泥于航天测控运管系统的三层结构。航天测控运管系统层次定义模型如下：

被管对象标识	被管对象类别	网络地址	上级管理站
--------	--------	------	-------

### 1.2 被管对象及编码

在航天测控运管系统中, 把被管对象分为 2 类:

- 1) 设备/系统类被管对象: 是基本被管对象, 由独立存在的被管设备构成, 比如: 雷达、USB 设备、遥测设备、光学设备、路由器、交换机和计算机等;
- 2) 管理节点类被管对象: 是组合被管对象, 不对应某一具体设备, 由多个基本被管对象组合而成, 比如中心级管理节点、工作站级管理节点、委托代理节点和各专业的专业管理系统设置的管理节点。在系统的信息流程中, 每个被管对象都被分配一个被管对象标识符 (OID), 在系统中是全局唯一的, 采用 4 字段定长分层编码方式, 采用句点分隔的标识方法。因为层次结构信息是通过 SNMP 报文携带的, 被管对象标识的编码规则需要满足 SNMP 协议标准, 所以其编码规则类似 MIB 编码, 也是按设备类别进行编码。编码规则为: 被管对象类别 (2 字段)+ 上级管理节点编码 (1 字段)+ 本级管理节点编码 (1

字段), 其中被管对象类别字段分为管理节点类和被管设备/系统类 2 个类别, 分别对应各级管理节点和各类被管设备或系统; 上级管理节点编码 (中心级) 字段表示该被管对象的上一级管理节点, 一般为中心级管理节点; 本级管理节点编码表示该被管对象在本级的编码。其中一级管理节点的上级管理节点编码固定为 “00”。

### 1.3 统一的信息模型

航天测控运管系统的被管对象构成十分复杂, 不仅包含标准的计算机及网络设备 (通用设备), 还包括众多的测量、通信等专用设备, 而这些专用设备又是航天测控运管系统管理的重点。通用设备采用标准的 SNMP MIB II 定义作为其管理信息库, 专用设备由于其特殊性, 无法使用标准的 MIB II 定义, 须自己定义。航天测控网管系统要求对全系统所有设备进行监视和控制, 且对测控设备的监控是重点, 其管理信息库 (MIB) 应围绕着测控设备, 再将通用设备的 MIB 加入其中, 实现通用设备和专用设备不同格式的 MIB 的形式统一。根据航天测控运管系统被管对象使用的管理信息共有 3 大类: 通用部分 (mib-2 分支)、专用部分中的公共管理信息和专用部分中的专业管理信息。根据航天测控运行管理系

统设备分类标准，测控系统所属的被管设备可分为测控专用设备、通信专用设备、安全专业设备和计算机专用设备。将所有纳入航天测控运管系统的设备，主要按照设备系统、设备类别进行分类，参照 SNMP 协议的管理信息结构(SMI)所制定，通过句点分隔以命名树的形式来标记 MIB 变量，由通用部分和专用部分构成。通用部分采用 SNMP V2 管理信息结构(SMI)的 mib-2 分支，专用部分利用 SNMP V2 管理信息结构(SMI)的私用-企业分支(1.3.6.1.4.1)按照专业进行分类，私用-企业分支的

13526 分支作为航天测控系统运行管理系统专用设备管理入口。航天测控运行管理系统管理信息结构(H-SMI)如图 2。专用设备按参数顺序编码，通用设备采用标准 SNMP MIB II 编码加设备类头的方法实现。航天测控网运行管理应用程序在进行协议处理时，对通用设备的 MIB 采用加头去头的方式实现，即发送时去掉设备类别部分，以标准 MIB 的形式发送到被管设备。接收到设备应答产生的标准 MIB 信息后，再加上设备类头，以航天测控网管系统的 SMI 格式进行保存。

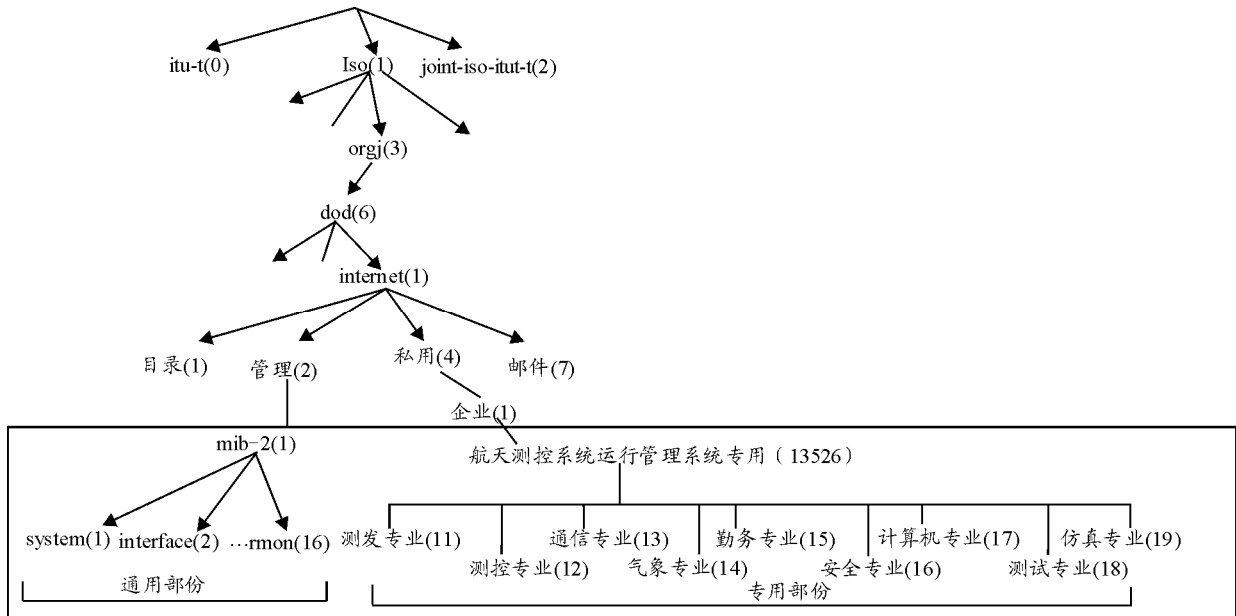


图 2 航天测控运行管理系统管理信息结构(H-MIB)

### 1.4 管理模型

航天测控运管系统采用的管理模型为“管理者—管理者—代理”模型，即管理站间通过对 SNMPv2 的 GetInfromRequest 报文进行扩展，实现在 SNMP 的管理者—代理管理模型上再加上一层管理者。这就要求运管软件应具有对过境的 SNMP 报文具有转发控制能力，起到类似于网关的作用，既阻挡非法访问，又确保运管信息的畅通无阻。通过扩展 SNMPv2 的 InformRequest 指令的功能，使之携带指令方向和发送源等信息，通过指令方向和发送源的级别(一级运管中心、二级运管中心、运管站由高到低)，按转发规则控制运管信息的传递。转发规则规定在发送方向上禁止下级对上级的访问，禁止同级间相互访问，对返回方向的指令则允许通过。各级管理站间采用 InformRequest 协议数据单元(protocol data unit, PDU)实现信息交互，管理站与被管设备间根据采用

GetRequest/GetNextRequest/GetBulkRequest PDU 和 Trap PDU 实现信息交互。下级管理站具有为其上级管理站转发指令和信息的职责，且转发的同时将信息进行本地处理；对于被管设备产生的 Trap 信息，管理站收到处理后，以 InformRequest 指令发往其上级站，然后逐级转发和处理该指令，从而确保每级管理站均可收到被管对象的最新状态，以实现被管对象状态全网同步。

## 2 软件结构及信息交互

### 2.1 SNMP 软件结构

航天测控运管系统从本质上讲是一个基于客户/服务器(C/S)模式的数据库应用软件，有关系统的所有的监测信息、配置信息均存储在后台数据库上<sup>[4]</sup>。航天测控运管软件由监控台(客户端程序)、应用服务器(服务器程序)、数据库服务器 3 部分组成。

1) 监控台即软件的管理控制界面，运管人员通过监控台查看设备状态、进行各种管理，可根据需

要随时启动和退出。2) 应用服务器程序作为后台进程, 系统启动后即一直运行不能中途退出, 执行关键服务并完成所有的管理功能, 负责处理其他管理站或被管设备的进站指令和监控台程序待发送的出站指令。3) 客户端程序和服务器程序之间采用 DCOM(分布式组件对象模型) 技术进行进程间通讯, 实现信息交互, 但进程间通信只起到一个触发作用和交换一些简单的信息, 主要信息的交互还是通过后台数据库进行的<sup>[5]</sup>。4) 数据库服务器对所有的管理数据进行统一的存储、管理和维护, 所有对数据的访问均通过应用服务器提供的统一接口进行。软件结构如图 3。

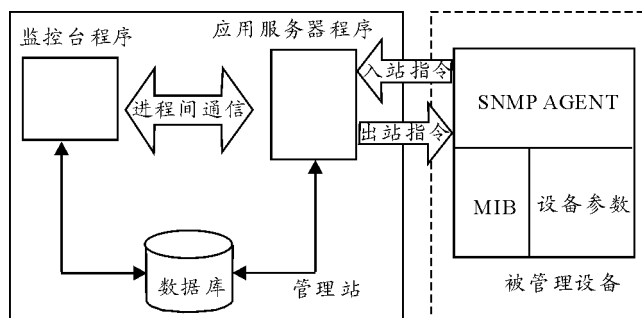


图 3 航天测控运管系统软件结构及信息交互

## 2.2 信息交互

以 SNMP V2 标准协议操作进行运管节点与测控设备之间的网络管理信息交互:

### 1) 被管设备与其所属的管理节点间状态监视

被管设备接收、解析管理者发出的请求信息的能力, 包括 GetRequest、GetNextRequest 和 GetBulkRequest 的 3 类 PDU。被管设备向管理者返回请求结果的能力, 通过 GetResponse PDU 实现。被管设备主动向管理者发送 Trap 消息的能力, 通过 SNMPv2-Trap 实现。被管设备向多个管理者发送 Trap 消息的能力, 通过向不同的管理者分别发送 Trap 消息的方式实现, Trap 本身为单播 UDP 报文, 代理发出的 Trap 消息其 PDU 变量绑定的第 3 个变量为设备自定义的 Trap 变量。

### 2) 参数配置

被管设备具备参数配置能力, 通过接收、解析 SetRequest PDU 实现, 并具备向管理者返回配置情况的能力, 通过 GetResponse PDU 实现。

## 2.3 信息流程

通过被管对象标识法建立的分层式网络管理系统的工作原理是, 当某一管理站需要访问层次结构中定义的某一被管设备时: 1) 首先查找该被管设备的层次定义信息, 如果该设备层次定义信息的“上

级管理站”定义为本地的话, 则可确定该被管对象处在本地管理域中, 则根据其网络地址发送标准的 SNMP GetRequest 指令直接访问即可。2) 如果该设备层次定义信息的“上级管理站”为其它管理站, 则将通知报文发送到该设备定义的“上级管理站”, 至于该报文的以后流向, 由该设备定义的这个“上级管理站”再根据其自身定义的层次结构, 决定该报文的下一目的。依此原理该指令在管理站间不断向前传递, 直至最终的目标。

应答信息也是按相同道理经过管理者的层层传递, 最终返回给发送请求的管理者。在实际应用中管理站的层次不受航天测控网的三级结构限制, 可以任意设置层次结构, 各级管理站均具有相同的管理功能, 只是最高级别的管理站无需做指令转换和转发的工作, 在航天测控运管系统中, 是以数据库表的形式保存这个结构信息的。

## 3 协议扩展与与处理

### 3.1 协议扩展

SNMPv2 在 SNMPv1 的基础上新增管理者到管理者 (GetInformRequest) 的协议操作, 用于提高 SNMP 的分布式处理能力, 该协议操作仅提供了管理站之间的通信手段。对该协议操作加以扩展, 才能实现分层模式下的信息传递。即通过扩展 GetInformRequest 协议操作, 使之携带足够的层次结构信息, 由管理站的网络管理应用程序根据这些信息控制 SNMP 消息在分层式体系结构中的传递。

扩展后的 GetInformRequest 消息格式如图 4。

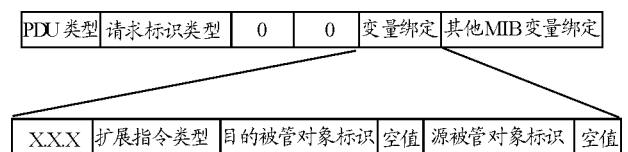


图 4 扩展后的 GetInformRequest 消息格式

第 1 个绑定的 MIB 变量用来标识扩展指令, X.X.X 为在标准的 MIBII 树中没有被使用的变量名, 称之为扩展指令变量, 其值代表扩展指令的类型。第 2 个绑定的 MIB 变量为目的被管对象标识, 用来说明该消息的最终目的地, 其值为空。第 3 个绑定的 MIB 变量为源被管对象标识, 用来说明该消息是由那个管理站初始产生的, 其值为空。

扩展指令类型说明如下:

1) 扩展指令类型为 0 时, 称之为扩展请求指令 (Inform-get), 表示该消息是管理站发出的被管设备请求消息, 类似于标准 SNMP 协议操作的 GetRequest 消息。2) 扩展指令类型为 1 时, 称之为

扩展应答指令 (Inform-response), 表示该 Inform 消息是下级管理站向上级管理站返回的被管设备应答消息, 类似于标准 SNMP 协议操作的 GetResponse 消息。3) 扩展指令类型为 2 时, 称之为扩展 Trap 指令 (Inform-trap), 表示该 Inform 消息是下级管理站向上级管理站转发的被管设备的 Trap 消息, 类似于标准 SNMP 协议操作的 Trap 消息。4) 扩展指令类型为 3 时, 称之为扩展 set 指令 (Inform-set), 表示该 Inform 消息是上级管理站对下级管理站所管设备发出的修改 MIB 信息的消息。类似于标准 SNMP 协议操作的 SetRequest 消息。

### 3.2 协议处理

协议处理是指管理站通过后台应用服务器对 SNMP 协议操作的处理过程, 以实现分层式体系结构的网络管理。

#### 3.2.1 SNMP GetInformRequest 协议处理

该协议操作用于管理者间的信息通信, 当管理站收到该协议操作后, 首先取其变量绑定的第 1 个变量, 以判断该协议操作是否为扩展的协议操作, 如果不是则按正常的管理者间通信处理, 如是扩展的协议操作, 则取其对应的值, 判断是哪一类扩展指令, 具体的处理过程为:

##### 1) 扩展请求指令 (Inform-get)

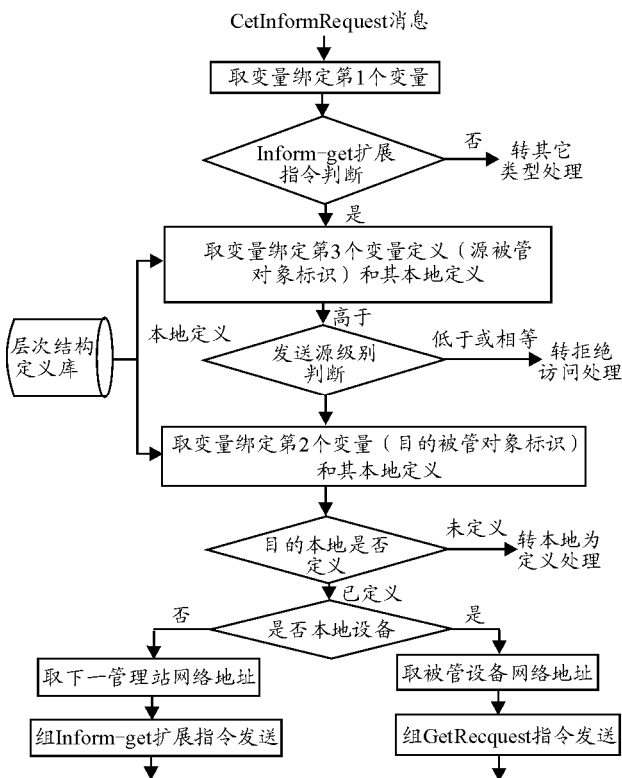


图 5 扩展请求指令 (Inform-get) 处理流程示意图

取变量绑定中的第 3 个变量, 判断该指令的发送者 (源被管对象标识) 在层次结构中所处的位置, 如果同级或下级则该消息被拒绝, 并向发送方发送越级访问的警告报文。如级别高于本地, 则根据第 2 个绑定的变量 (目的被管对象标识) 判断该被管对象是否为本地管理设备, 通过结构定义可判断出该被管对象是被管设备还是管理站。如是本地管理设备则生成标准 SNMP 协议的 GetRequest 消息发往该被管设备, 否则, 从结构定义中取出该被管设备所对应的上级管理站在本地定义的网络地址, 重构 GetInformRequest 指令 (绑定的 PDU 内容不变) 向该管理站发送。如果目的被管对象未在本地定义, 则作废包处理。同时向发送源发送被管对象未定义的出错消息。该指令协议处理流程如图 5。

##### 2) 扩展应答指令 (Inform-get)

取变量绑定的第 3 个变量 (源被管对象标识), 获取其包含的源被管对象标识, 取第 4 个变量及其后绑定的所有 MIB 变量及值, 进行故障处理。处理完成后, 根据获取的源被管对象标识修改本地被管对象状态库。再取出第 2 个变量 (目的被管对象标识), 根据目的被管对象标识判定该消息的下一方向, 如目的被管对象标识就是本地管理站, 则该消息的传递到此终止, 如不是则获取其本地定义的网络地址。继续向上级转发该消息。该指令协议处理流程如图 6。

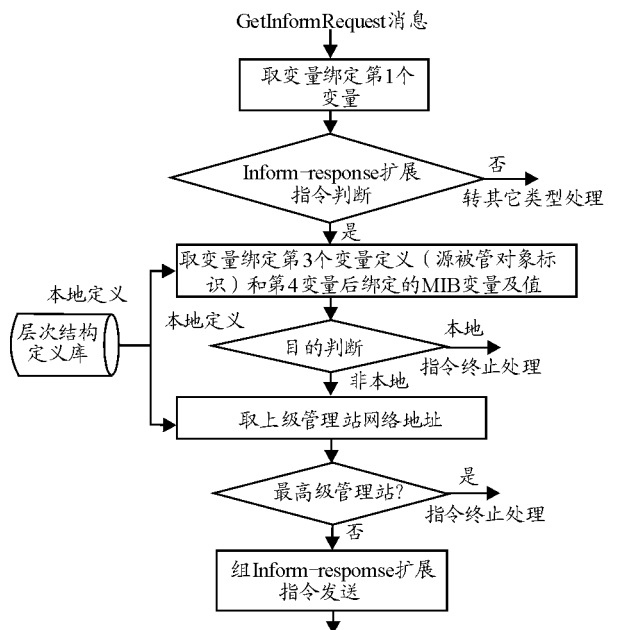


图 6 扩展应答指令 (Inform-response) 处理流程示意图

##### 3) 扩展 Trap 指令 (Inform-get)

取变量绑定的第 3 个变量 (源被管对象标识),

获取其包含的源被管对象标识, 取第 4 变量及其后绑定的所有 MIB 变量及值, 进行故障处理。处理完成后, 根据获取的源被管对象标识修改本地被管对象状态库。然后从层次结构定义库中提取本地定义的上级管理站的网络地址, 重构 Inform-trap 扩展指令 (PDU 绑定不变) 发往上级管理站, 直至最高级管理站。该指令协议处理流程如图 7。

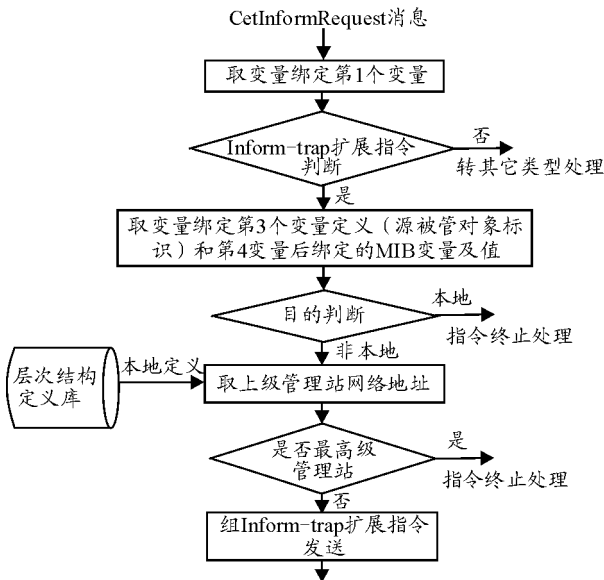


图 7 扩展 Trap 指令 (Inform-get) 处理流程示意图

4) 扩展set指令 (Inform-get)

信息流程同扩展请求指令 (Inform-get), 只是将指令转换为标准 SNMP 协议的 SetRequest 消息发送到被管设备。

3.2.2 SNMP GetResponse 协议处理

该协议处理用于管理站处理所收到的 SNMP GetResponse 消息, 判断所收到的 GetResponse 消息是否是被管设备发出的, 因为只有被管设备对 get 协议操作的应答才包含被管对象的 MIB 信息。采用的方法是: 只响应 Get 产生的应答消息, 对其它指令产生的应答消息不作故障处理。其处理过程是: 当管理站收到上级管理站发送的本地设备请求信息后, 首先保存该 Inform-get 扩展指令中所携带的目的被管对象标识和源被管对象标识, 然后每当收到对 Get 指令的应答消息 Get Response 后取其发送源的网络地址, 通过该网络地址从本地被管对象定义 (层次结构定义) 中取出该地址对应的被管对象标识, 然后对 GetResponse 所携带的 MIB 信息进行故障处理, 根据处理结果更新该被管对象的当前状态。再将该被管对象标识与保存的目的被管对象标识相比较。如果存在, 则该消息是本地管理站为上级管

理站转发的 get 指令所产生的应答消息, 再将保存的源和目的被管对象标识对调, 作为 Inform-Response。扩展指令的目的和源被管对象标识连同 MIB 信息一起构成 Inform-Response 扩展指令发送给请求的上级管理站; 如果不存在, 则为本地管理站自己发出的 Get 指令的应答, 本地处理后不再向上级转发。处理流程如图 8。

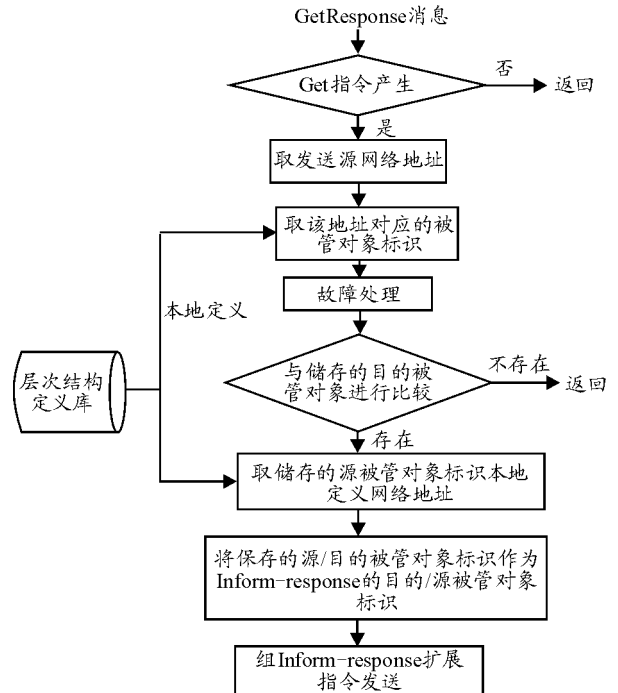


图 8 Response 消息处理流程示意图

3.2.3 SNMP Trap 协议处理

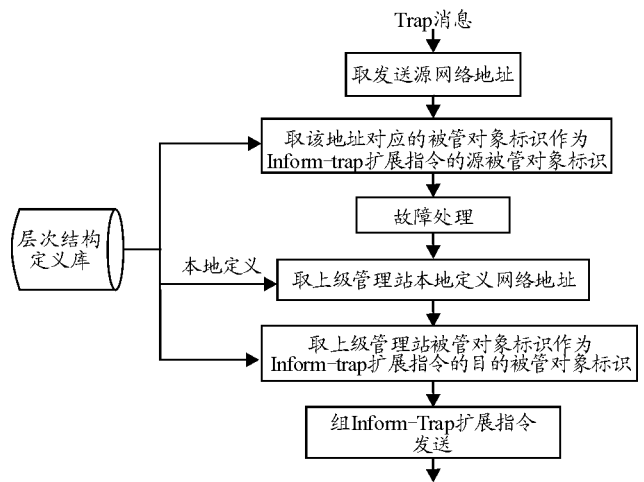


图 9 Trap 消息处理流程示意图

航天测控运管系统规定所有 Trap 信息一律转发至最高级的运管中心, 所以 Trap 消息在本地处理完后, 需立即传至上级管理站。其处理过程是: 当管理站收到被管设备发出的 Trap 消息后, 通过该 Trap 的发送源地址, 从层次结构定义中取出发送源

的被管对象标识(作为 Inform-trap 扩展指令的源被管对象标识),具体步骤同 SNMP GetResponse 协议处理。然后进行故障处理,根据处理结果更新被管对象状态库。再取出上级管理站被管对象标识(作为 Inform-trap 扩展指令的目的被管对象标识)及网络地址,连同该 Trap 的 PDU,共同组成 Inform-trap 扩展指令,发送到上级管理站。每级管理站收到该 Inform-trap 扩展指令后,均向其上级管理站转发,一直发送到本系统的最高级别的管理站为止。处理流程如图 9。

### 4 结论

在分层式运管系统中,各层间的通信是由各层的管理站完成的,每层管理站都根据所在层的位置保存一个向下的层次结构定义的子集,一旦上级管理站的访问信息到达后,将由这个子集的定义来决

定其下一走向。当网络层次结构变更时也只需改变与其直接相连的上下级管理站的网络地址配置即可。因此,该方法不仅简化了系统的配置工作,提高了配置的灵活性,也满足了航天测控网对网络安全的需求。

### 参考文献:

[1] 闫旭. 航天测控网网络管理系统若干问题研究[D]. 武汉: 华中科技大学, 2005.  
 [2] Mark A, Miller P. E. 用 SNMP 管理互连网络[M]. 北京: 电子工业出版社, 2001.  
 [3] 胡玲洁. 扩展 SNMP 网管代理的设计与开发[J]. 长春理工大学学报, 2009, 11(4): 183-184.  
 [4] 武孟军. Visual C++开发基于 SNMP 的网络管理软件[M]. 北京: 电子工业出版社, 2007.  
 [5] 黄苏粉, 张江鑫. 基于 SNMP 的网络管理通信模块的实现[J]. 杭州电子科技大学学报, 2009, 29(5): 101-105.

\*\*\*\*\*

(上接第 89 页)

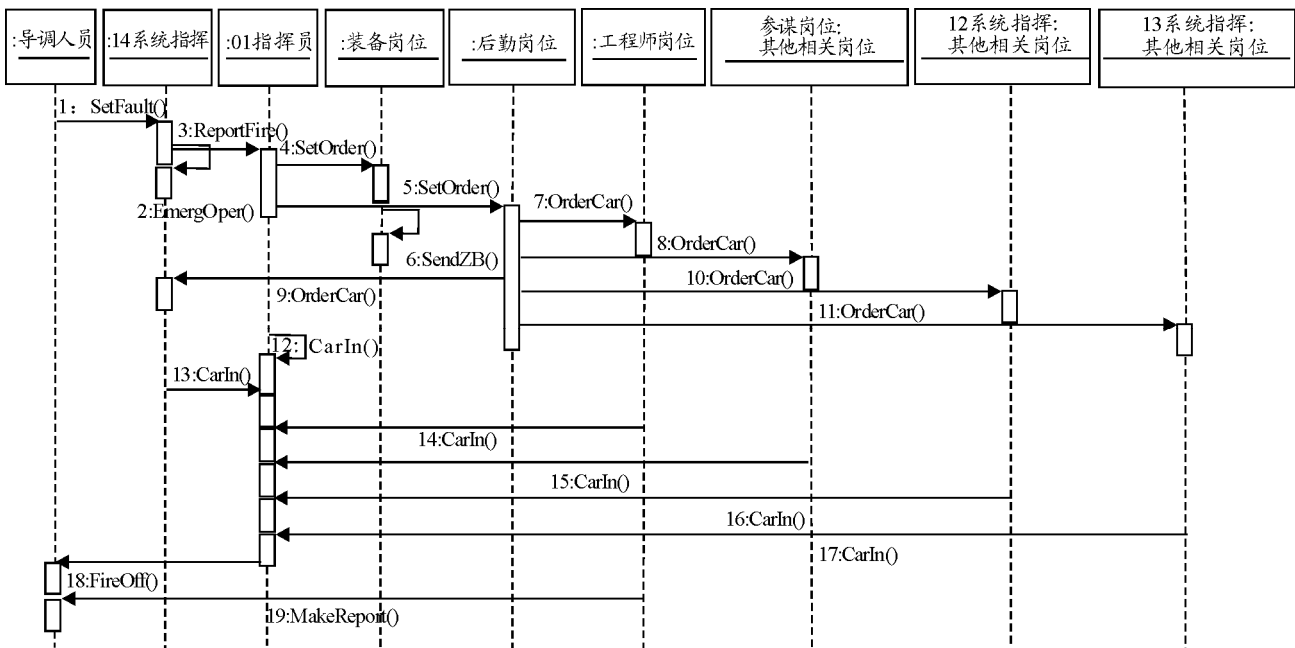


图 5 火工品测试间爆炸突发事件序列图

### 4 结束语

笔者采用 Visual C++作为开发工具,运用美国 MultiGen-Paradigm 公司的虚拟现实开发软件 Vega Prime 作为视景仿真工具,实现了该仿真系统。实际结果证明,基于 UML 进行系统分析与设计,便于从全局上纵览整个软件结构以及整个软件的结构和各分系统结构间的关系。

### 参考文献:

[1] 罗开平, 李一军, 姜维. 基于UML的弹道导弹预警仿真系统建模[J]. 系统仿真学报, 2010(4): 845-849.  
 [2] 刘超, 张莉. 可视化面向对象建模技术—标准建模语言 UML教程[M]. 北京: 航空航天大学出版社, 1999.  
 [3] Jason T. Rof. UML基础教程[M]. 张瑜, 等译. 北京: 清华大学出版社, 2003.  
 [4] 孙培, 李辉. 基于UML的导弹战损仿真系统分析和设计[J]. 科学技术与工程, 2008(7): 1861-1865.  
 [5] 徐宝文, 周毓明, 卢红敏. UML与软件建模[M]. 北京: 清华大学出版社, 2006.