

复杂网络环境下 SNMP 和 CMIP 集成管理技术

Integrated management technologies of SNMP and CMIP in complex network environment

胡卫华 王艳清 (北京化工大学 信息科学与技术学院 北京 100029)

摘要:网络管理协议是网络管理系统重要组成部分,简单网络管理协议(SNMP)和公共信息管理协议(CMIP)是目前两种重要的网络管理协议,这两种管理协议在信息结构定义和通信机制上存在很大差异,论文对这两种管理协议的集成技术进行了研究,并重点讨论用基于 SNMP 协议的管理系统管理基于 CMIP 协议的网络设备的委托代理技术的一种实现。

关键词:简单网络管理协议(SNMP) 公共信息管理协议(CMIP) 委托代理

1 引言

网络在物理区域上日益扩大已经不仅仅是网络管理需要面对的唯一问题,不同品牌、不同型号的网络设备,不同实现方式的网络应用程序聚集在同一网络中给网络管理带来了又一难题。一致性无疑是降低网络管理系统复杂度,减少管理成本的有效选择。在实现一致性的过程中产生了两个独立而又相似的网络管理协议,即 SNMP(简单网络管理协议)和 CMIP(公共管理信息协议)。

SNMP 的优点是设计简单,容易实现,可扩展性强,但安全性差,功能较弱。CMIP 的功能强大,但是它需要占用大量网络资源,因此仅在电信等少数范围内有所应用。但如果弃粗存精将两者有机结合,那么网络管理将会从统一的管理协议方案中获益,即保护了现有网络管理系统的商业投资,又节省了开发时间。

2 SNMP 与 CMIP 集成管理方案

目前,SNMP 和 CMIP 协议集成管理方案主要有两种:一种是协议共存(coexistence),另一种策略是协议互通(interworking)。

2.1 协议共存

协议共存的思想主要是将 SNMP 和 CMIP 协议共存于同一管理系统中。使用的主要的策略有双协议栈方法、混合协议栈方法、通用应用程序接口(APIs)方法。

双协议栈方法主要思想是在一个管理站上通过支持 SNMP 和 CMIP 两种体系结构的协议栈来统一管理该网络中支持 SNMP 和 CMIP 协议的网络设备,如多协议主机、路由器、交换机等。混合协议栈主要思想是将管理协议加载到不同的网络传输层协议上。即将基于 TCP/IP 的 SNMP 加载到 OSI 模型上,而将建立在 OSI 模型之上的 CMIP 应用在 TCP/IP 协议上。混合协议栈的目标是使网络通信协议相对管理应用层变的透明,使管理者的主要精力放在选取最恰当的管理协议上。通用应用程序接口(APIs)是一个目标比较宏大的解决方案。该方法“基于一种多厂商的网络管理平台”^[7],“提供一个公共的 API 集合来屏蔽特定的管理协议、数据定义及用户接口的细节和差异,以便于管理应用的开发”^[7]。

2.2 协议互通

协议互通的思想是在管理者和被管设备之间通过一个委托代理实现管理信息的转换和管理协议功能的映射,使得支持 CMIP 协议的管理设备可以执行基于 SNMP 协议的管理进程的各种管理命令,而基于 SNMP 协议的管理进程也可以正确接收、解析支持 CMIP 协议的管理设备发来的报警信息。反之亦然。

2.3 集成策略比较

双协议栈方法的前提是被管网络设备支持协议栈中的各个协议,所以该方法对管理多协议设备很实用。混合协议栈方法正被一些官方组织开发研究,如

CMOT (CMIP over TCP/IP) 协议和 RFC1418 (SNMP over OSI)。通用应用程序接口 (APIs) 方法有利于第三方软件的开发和集成,但复杂度非常高。委托代理可以在不改变现有管理网络的基础上对支持其他协议的设备进行管理,有效降低管理成本,所以比较容易被用户接受。

很明显,SNMP 与 CMIP 委托代理技术分为两种,一种是 CMIP 管理方向 SNMP 代理方转换,另一种是 SNMP 管理方向 CMIP 代理方转换。由于 SNMP 协议设计简单、容易实现,网络信息流量压力小、可扩展性强等特点使得其成为现代网络管理应用中默认的标准。但随着 TMN 概念在电信上的推广,支持 CMIP 协议的物理设备正慢慢拥有自己的市场。无论从商业用户投资的角度上讲还是从网络管理者改动现有管理系统的程度上讲,用户更倾向于由 SNMP 管理方向 CMIP 代理方转换。

3 SNMP 与 CMIP 委托代理实现方案

SNMP 管理方向 CMIP 代理方转换模型如图 1 所示,涉及的主要技术点包括:怎样使 CMIP 协议的管理信息结构映射为 SNMP 协议的管理信息结构;怎样使 CMIP 协议识别的被管对象映射为 SNMP 协议识别的被管对象;怎样使 SNMP 和 CMIP 协议相互正确解析对方发出的数据包。

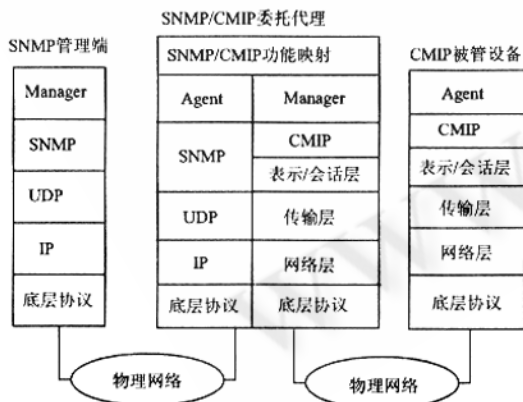


图 1 SNMP/CMIP 委托代理模型

3.1 管理信息结构的映射

SNMP MIB 只支持标量和标量的二维矩阵。所以 SNMP 协议的管理信息都以简单变量和表的形式出现。在这种情况下采用一个标量和一个二维表的组合

来映射一个 CMIP 被管对象类。简单标量用来记录该被管对象类实例的个数,二维表用来映射类结构。类的属性名构成二维表的列名,类的实例构成二维表的行。为了在二维表中体现类的继承关系,二维表将有一列记录该类的父类名。为了支持被管对象实例的添加、删除操作,二维表还将有一列记录该行实例的状态。以某台打印机的打印队列为例,CMIP PrintJob 类的映射如图 2 所示。这里要注意到 SNMP 的 MIB 是预定义生成的,可以在管理信息中增删管理对象的实例,但要增删管理对象,那就要重新编译 MIB 树。

3.2 管理对象的映射

网络管理一般先要确定管理的是哪台网络设备,然后再确定管理的是被管设备的哪个参数。SNMP 和 CMIP 在识别被管对象上有着明显的不同。SNMP 靠 IP 寻址来确定被管网络设备,再通过一棵针对该设备的预定义命名树来确定被管网络设备内的某一被管参数。这就意味着对一个被管网络来说,被管设备内的管理对象命名仅在设备内是唯一的。CMIP 的命名却有极大的灵活性,被管网络中的被管设备以一种相互包含的形式存在于一棵包含树上,每个被管网络设备的全局命名通过把从根到该被管设备的路径上每个对象实例的相对名连接起来构成的。被管设备内的被管参数是以该被管设备类的参数或子类的参数形式出现。被管参数实例命名也是将相对名连接到被管设备实例命名尾部构成全局名。

所以在委托代理中对应 SNMP 和 CMIP 的被管对象实例时将分两步,首先对应被管设备,其次对应被管设备中的被管对象。

采用 IP 和 OSI 系统网络设备实例名对应表在 SNMP 和 CMIP 之间映射被管设备。由于 SNMP 被管设备包含自身的一棵 MIB 树,委托代理必须知道要将 CMIP 被管设备内包含的各个被管对象类映射到哪棵 MIB 树上,所以 IP 和 OSI 系统网络设备实例名对应表中还应存在 MIB 树的对应信息。对应表结构如图 3 所示。

在对应被管设备中的被管对象时要维护的也是一张转换表,在 SNMP MIB 树上某 table 名加该 table 中 id 属性值可以唯一对应一个 CMIP 设备中的被管类的实例,在此基础上 table 中其他属性就可以对应到 CMIP 设备中被管类的实例的属性上了。

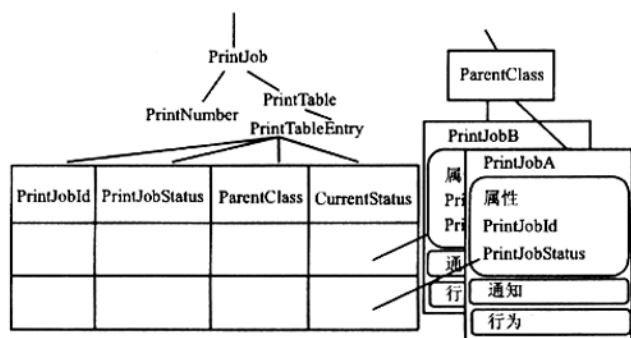


图 2 管理信息结构映射

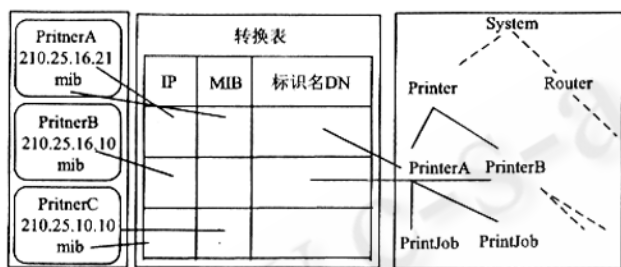


图 3 被管设备映射表

由此可以看出,如果委

托代理负责管理多个 CMIP 设备,甚至是一个 CMIP 小型网络,那委托代理所要维护的转换表会很大,为了确保 SNMP 管理操作 PDU 正确传输给目标设备和 CMIP 应答 PDU 正确对应原 SNMP 请求所要求的处理开销同样会非常大,效率并不乐观。如果委托代理和 CMIP 代理处于同一设备中,委托代理所要维护的信息量相对减少,执行转换的针对性加强,转换相对容易实现。

3.3 通信数据包的映射

3.3.1 SNMP 管理方与 CMIP 代理方查询/应答数据包转换

SNMP 管理方的查询操作 GetRequest - PDU, GetNextRequest - PDU, GetBulkRequest - PDU 转化为数个 CMIP 的 M-GET 操作, 转化 M-GET 操作的个数由 SNMP 管理方的查询操作变量绑定列表中变量的个数决定。即 SNMP 查询的每一个变量都将由一个独立的 CMIP M-GET 操作完成。

在转化过程中,CMIP PDU 中操作 ID 字段的值应和 SNMP 请求的 request-id 字段的值相同,从而标识 CMIP M-GET 操作是转化的哪个 SNMP 请求,此外 SNMP 请求 PDU 变量绑定字段的有序性还决定了

CMIP M-GET PDU 是有序的, 所以 CMIP M-GET PDU 中应包含该 PDU 转化的是第几个变量绑定的索引信息。由于转化后的 CMIP M-GET PDU 有确定的查询目标, CMIP 的 scoping 和 filtering 字段将不使用。

对于 SNMP v1 来说 GetRequest - PDU 请求是原子性,所以转化后对应的 CMIP M-GET 只要有一个没成功返回,委托代理就将放弃所有值,返回含错误信息的 GetResponse - PDU。对于 SNMP v2 来说 GetRequest - PDU 请求没有原子性要求,委托代理将成功的 CMIP M-GET PDU 中的值提取出来,按 CMIP M-GET PDU 中包含的变量绑定的索引值将提出的变量绑定值组成 SNMP GetResponse - PDU,在 CMIP M-GET 操作失败的变量绑定字段填写相应的错误信息,并将失败的 CMIP M-GET 操作的索引值填写到 SNMP GetResponse - PDU 的 error-index 字段中。

对于返回的错误信息,CMIP 和 SNMP 出入很大,CMIP 有更加详细的错误报告。但这不影响 SNMP 的操作,可以将 CMIP 的错误信息归类,对应 SNMP 的错误报告。

GetNextRequest - PDU, GetBulkRequest - PDU 和 GetRequest - PDU 转化过程类似,只是委托代理要支持 GetNextRequest - PDU 的按字典顺序检索。

3.3.2 SNMP 管理方与 CMIP 代理方设置 / 应答数据包转换

SNMP 对被管对象实例参数的修改,被管对象实例的增删操作,让被管对象实例执行某项简单动作将全部由 SetRequest - PDU 完成, SetRequest - PDU 将分别被映射成 CMIP 的 M-SET、M-CREATE、M-DELETE 和 M-ACTION 操作。

SetRequest - PDU 是原子性操作, 只要一个 SetRequest - PDU 中有一个要操作的变量没有完成就宣告本次操作失败。为了确保 SetRequest - PDU 是原子性的, 将 SetRequest - PDU 转化成数个 CMIP 操作, 并且每个 CMIP 操作只对原 SetRequest - PDU 绑定列表中的一个变量绑定做 M-SET、M-ACTION、M-CREATE 和 M-DELETE 中的一种操作。

在转化过程中,CMIP PDU 中操作 ID 字段的值也和 SNMP 请求的 Request - Id 字段的值相同,并且 CMIP PDU 要记录所操作的变量在原 SNMP PDU 中的索引值。

在转换 SetRequest - PDU 为 M - SET 过程中,如果由于其他转换操作的失败而造成本已成功的 M - SET 转换放弃,委托代理必须有能将改变过的值复原。就要求委托代理在执行 M - SET 前对原值做保存,并支持恢复操作。

在转换 SetRequest - PDU 为 M - CREATE 或 M - DELETE 过程中,要注意 CMIP 被管理对象的继承关系。一个被管理对象在没有子继承者的情况下才允许被执行删除操作。所以委托代理必须对 SNMP 列表中行的父子关系做分析判断后才可以转换 M - CREATE 或 M - DELETE 操作。

3.3.3 CMIP 代理方向 SNMP 管理方事件报告数据包的转换

CMIP 事件报告以事件驱动,而 SNMP 采用轮询的方式。由于 CMIP 语意上要比 SNMP 丰富并允许对事件驱动进行说明,所以 CMIP 事件报告应在委托代理处解析,并将说明包含在一个 SNMP Trap 中发送。由于事件驱动方式不一样,委托代理会将收到的 CMIP 事件报告在自己被管理站轮询到时发送。

4 结束语

由于两种协议的被管对象定义思想不同,管理信

息传输方式不同,在同一网络中集成两种协议的开销相当大,同时造成了部分功能的损失,但委托代理技术扩大了 SNMP 管理系统的管理范围。随着电信网络的日益成熟和管理技术的慢慢发展,委托代理技术越来越受到研究人员和商家的认可和支持。

参考文献

- 1 ITU-T Recomm. X.711. Common Management Information Protocol Specification[S].
- 2 ITU-T Recomm. X.710. Common Management Information service definition[S].
- 3 RFC 1905 Protocol Operations for Version 2 of the Simple Network Management Protocol [S].
- 4 RFC 1157 Simple Network Management Protocol[S].
- 5 W. Richard Stevens. TCP/IP Illustrated Volume 1: The Protocols[M]. 第一版,北京:机械工业出版社,2005:270-292.
- 6 郭军,网络管理[M],第二版,北京:北京邮电大学出版社,2003:25-73 102-193.
- 7 蔡皖东、周兴社,SNMP 和 CMIP 集成管理技术的研究[J],小型微型计算机系统,1999,20(5):338-392.