

SNMP 网络管理中高效轮询方法研究

朱创录^{1,2}

(1. 陕西渭南师范学院 数学与信息科学学院, 陕西 渭南 714000;

2. 陕西渭南师范学院 网络工程技术中心, 陕西 渭南 714000)

摘 要:基于 SNMP 协议的网络数据采集是 TCP/IP 模型网络管理的基础,是互联网环境下网络管理的主要标准。但在应用过程中也存在很多具体问题,特别是在轮询过程中,缺乏统一标准,实现过程中存在效率低、附加流量大等问题,根据现有的网络环境,提出了一种基于优先级控制的轮询方法。该方法采用统计原理,可以根据网络状况动态修正在轮询过程的优先级别,根据不同优先级控制轮询过程的排队。通过测试系统的分析,该方法可以在一定程度上提高轮询的效率,减少关键对象的响应时间,达到有效管理的目的。

关键词:SNMP;轮询;优先级;数据采集

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2012)12-0135-04

Research on Efficient Polling in SNMP Network Management

ZHU Chuang-lu^{1,2}

(1. College of Mathematics and Information Science, Weinan Teachers College, Weinan 714000, China

2. Computer Network Engineering Technology Center, Weinan Teachers College, Weinan 714000, China)

Abstract:SNMP-based data collecting is fundamental for TCP/IP network management, and that is the main standard about network management in Internet. But there is much specific problem in the application process, especially in polling process, such as the lack of uniform standard, the low efficiency phenomenon, the additional communication flows. According to the existing network environment put forward a polling method based on the priority control. The method uses the statistical principle, can fix polling priority level according to the dynamic network status, and can control queue polling based on different priority level. Through the analysis of the testing system, the algorithm can improve the efficiency of the polling, and reduce the response time of the key object. This algorithm can achieve the purpose of effective management.

Key words:SNMP;polling;priority;data collection

0 引言

随着计算机网络技术的不断发展,其应用日益广泛,因此对其管理也不断的发展,SNMP 作为基于 TCP/IP 模型的网络管理协议已经有了很大程度的应用。利用现有的网络管理协议 SNMP 进行设备数据采集时,存在轮询时间长、管理站负担重、效率低等问题^[1,2]。因此需要对现有的 SNMP 轮询方式进行改进以解决传统网络管理模式的不足,在不增加网络传输负担的基础上尽量地提高传输效率是当前要解决的重要问题^[3]。也是对网络管理实时性要求的满足。文中提出了一种基于优先级控制的 SNMP 条件控制轮询算法以实现网络设备的实时数据采集,该方法通过对

代理的状态和在网络中的地位进行分类,同时对 MIB 对象进行设置优先级标签,并且根据网络状况,通过统计原理动态的修正优先级别,从而提高轮询的时效性。并且在实现过程中采用多线程的思路,从而提高轮询效率。

1 基于优先级的条件约束轮询算法

1.1 优先级控制方法

SNMP 的主要性能与 MIB 的搜索效率有关^[4]。按照 SMIv2 (SMI version 2) 的规则, MIB 中所有被管实体对象在概念上组成了一个树状结构, MIB 则为每个对象说明了具体的对象实例,并为每一个事例绑定了一个值,树结构从未命名的根开始^[3,5]。每个被管对象都具有唯一的名字,以对象标识符 (Object Identifier, OID) 作为标识,采用点分形式的整数序列定义。

SNMP 协议在开发过程中充分考虑到它本身的简单性,并充分考虑到高低版本的兼容问题^[6],但这

收稿日期:2012-04-05;修回日期:2012-07-07

基金项目:陕西省教育专项科研计划项目(11JK1067)

作者简介:朱创录(1977-),男,讲师,硕士,主要研究方向为网络应用和网络安全。

也给网络管理系统带来了一些不足:SNMP 协议轮询的方式在一个大型网络中可能会导致网络通信拥塞情况的发生,并且 SNMP 协议把采集数据的负担完全压在了管理者之上^[7];SNMP 代理无法提供要求数据的历史记录;SNMP 协议不能以一种统一通用的数据描述格式保存所有被管理设备的标识、状态和配置等信息^[8]。针对以上不足展开研究,期望解决网络管理中附加数据过多、查询效率低等问题。

针对网络管理中的被管设备对象,主要考虑获取设备当前的最新状态。由于网管系统中所监控的设备是动态变化的,随时可能出现新注册或者因故障而掉线的设备,为能实时地获取所有设备及其拓扑,可以按状态将设备划分为新注册、在线以及掉线三种类型。

```
typedef struct EquipmentAttribute
{
    char * EquipmentName; // 设备名称
    int EID
    //作为一个管理实体的设备唯一索引,不能重名
    int status;
    //新注册、在线以及掉线
} EquATTR, * EATTR;
```

在获取设备信息时,有些信息是相对稳定的,如生产厂家、设备类型、配置信息等,称其为静态属性;有些信息是不断变化的,如上行流量、下行流量、丢包数等,称为设备的动态属性。

设备状况确定后,主要考虑的是对象的类型描述问题,根据动态的网络状况,各个对象值的动态特征也有很大区别^[9]。数据采集方式主要有三种:静态数据采集、基于条件控制的动态采集、其它模式。

```
typedef struct ObjectAttribute
{
    smiOID oid; // 该对象的 OID
    smiLPVALUE m_value; // 该对象的值
    bool IfStatic; // 该 OID 是否为静态
    Int Priority; // 该对象的优先级
    bool Auto; // 是否自动获取信息
    smiUINT32 TimeInterval;
    // 自动获取信息的时间间隔
} OBJECTATTR, * OATTR;
```

对于静态数据采集,在构造对象数据的时候,将 ObjectAttribute 中的 IfStatic 变量设置为 TRUE,那么这些数据采集后,除非重新获取或者批量重新更新的情况下,一般是不需要动态更新的,从而可以减少网络管理给网络造成不必要的负担。

基于条件控制的动态数据采集,由管理站完成,主要用来采集管理代理的实时性能数据,根据不同的应

用需求,可以马上送回用户界面,交由用户处理或者进行某种提示,也可以传送至管理站的数据库接口,由后台数据库进行保留,必要时通过相应的数据库接口传送给用户接口。部分动态数据,控制台程序在判断其是否超过阈值后,将数据以某种形式实时呈现给用户,这种数据通常主要用于监测当前性能,不存储到数据库,不作为历史数据进行查询,或者将部分超出阈值的数据、发生时间、OID 等相关信息记录入日志数据库。通过对相关对象进行属性的扩充达到优化采集的目的,在构造对象的时候,将对象已有的基本信息根据 MIB 的定义保留在数据结构当中,除此,另外增加了新的属性:Priority 用来表示该对象的优先级,Auto 表示该对象的信息是自动获取还是手动更新获取,TimeInterval 表示自动获取的时间间隔。

其它模式,比如事件驱动方法通过对关键事件的预定义,管理代理在定义事件发生时,向管理者发送 Trap 报文^[10],这里不做详细探讨。

关于轮询对象周期的设置,也就是自动获取信息的时间间隔 TimeInterval 的设置主要是基于标准方差分析的方法,也就是说在每一次的轮询之后都会记录结果并参与标准方差运算,观察轮询数据偏离统计的样本平均值的偏差,根据偏差程度去确定优先级的高低,如果对象的轮询参数结果出现经常性的偏离样本平均非常大的情况,就可以判断该对象是需要进行更多次数的轮询,方便管理者应用程序对其进行进一步的处理。

基本方法如下:

1) 初始化 TimeInterval 的值,通常取值为 $(MAX_{(T)} + MIN_{(T)})/2$,也就是 TimeInterval 时间的中间值。

2) 根据统计原理计算取值偏离样本平均值的情况。

$$D(i) = \frac{\sum_{j=1}^i (V_j - \bar{V})^2}{i} \quad (1)$$

$$\bar{V} = \frac{1}{i} * \sum_{j=1}^i V_j \quad (2)$$

在公式(1)和(2)中 V_i 为每次获取的值,也就是轮询结果; \bar{V} 为历史轮询结果的样本平均值; V_j 为历史轮询过程中每次的返回值。这里需要注意的是每次运算 \bar{V} 值的计算并不需要将所有的历史数据都做记录,只需设置 Sum 变量保存历史求和结果即可。

3) 确定轮询周期。

$$TimeInterval(i) = F(D(i)) \quad (3)$$

这一步主要是根据上一步算出的轮询值偏差程度来确定其优先大小。基本思路是:如果轮询值跳动较大,说明该对象动态过程明显,在进行轮询时候,要选

择较少的轮询间隔时间。如果,轮询值不太变化,甚至不变,就可以认为该轮询对象静态特征明显,可以再处理的时候用较长时间进行轮询。关于 F 函数的选择可以根据具体的轮询对象而定,达到修行的目的^[11,12]。

另外一个重要的变量就是轮询对象的优先级 Priority,其值的确定可以由下面的公式得到:

$$\text{Pri} = Co/T(o) \quad (4)$$

其中 Co 为设定值,有两个作用:其一,可以将用户的期望值包含在内;其二,可用作归一化处理。 $T(o)$ 为该对象的轮询周期的时间变量。

1.2 轮询算法描述

该算法是基于统计特征的轮询优先级算法,首先,构造如图 1 所示的循环队列,循环队列中存储对象的 ObjectAttribute 结构体变量,从队列中找出优先级最高的对象作为当前对象。如果得到多个优先级相同的对象,可比较其轮询周期,如果轮询周期较短,进行优先安排。在确定对象信息之后就可以执行 SNMP 的 GetRequest 命令,获取其对象值。之后,就是进行 TimeInterval 值和 Priority 值的计算,为下一次的轮询过程做准备。TimeInterval 值的计算可以采用公式(3),Priority 值的计算采用公式(4)。至此,对于该对象的操作已经完成;下一步操作队列的下一个节点,直至对象循环队列中所有的节点都遍历完成。基于优先级的 SNMP 算法参考图 2 所示。

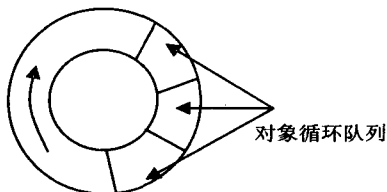


图 1 对象循环队列

2 实验性能分析

为了验证以上算法的有效性,在研究中采用 C++ 语言开发了一个 C/S 结构的网络管理系统,并进行相关的测试工作,测试结果是基于前面所描述的基于优先级的轮询算法。测试基本环境为园区网络,经统计,其中有 800 台设备,划分为 9 个子网。其中,所有设备都是可控设备,也就是说,每台设备均安装 SNMP agent 代理软件以支持 SNMP 协议。实验中主要的测试对象为:IP 地址、MAC 地址、端口数量等 45 个静态 MIB 变量;输入丢包率、输出丢包率、输入差错率、输出差错率、接口输入流量、接口输出流量、接口利用率等 15 个动态 MIB 变量。

在实验系统当中使用基于优先级控制的轮询算法,并结合普通的排队算法进行性能比较,分析了两种方法应用于 SNMP 管理过程中所产生的数据量。

图 3 为管理服务器出口的流量对比,根据图中分析,使用基于优先级等级的算法进行 SNMP 协议的轮询过程中所产生的数据量明显小于通常排队轮询过程所产生的数据量,说明通过采用优先级控制之后,进入网络的用于 SNMP 管理的数据量减少了,这给网络的正常运行减轻了负担,特别是在网络状况不太好的情况下,这种控制方法效果更加明显。

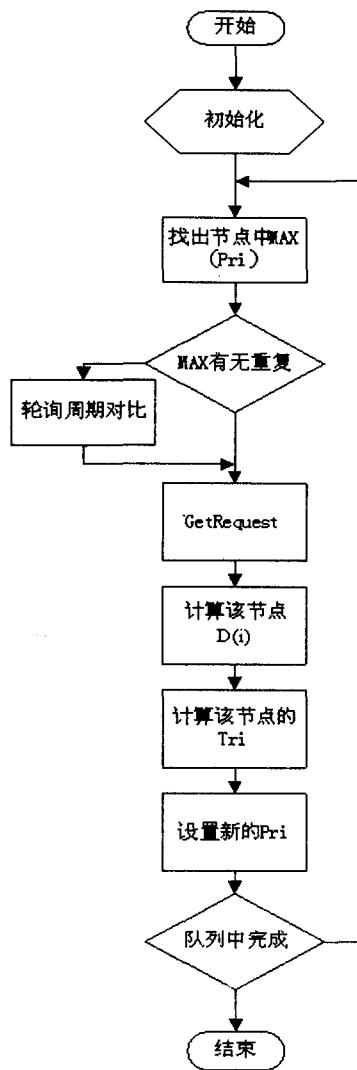


图 2 基于优先级的轮询流程图

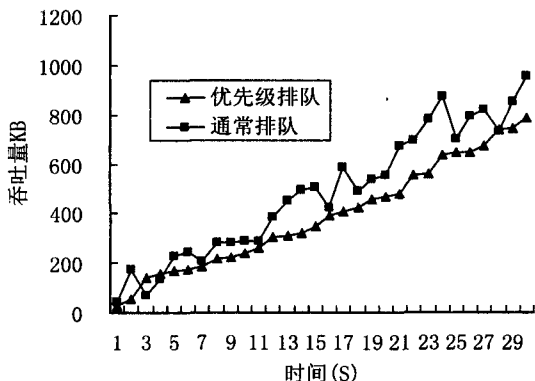


图 3 数据量比较

图 4 为网络中被管设备中 ipInReceives(3) 对象的相应时间比较,从图上可以看出:算法刚启动的短时间内优先级排队算法并无明显优势,但经过一段时间后,优先级排队算法的时间响应明显小于一般排队算法的时间响应。

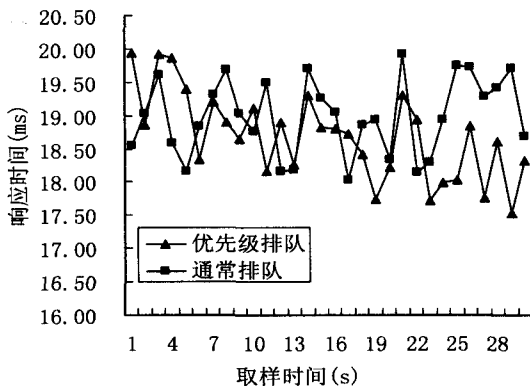


图 4 响应时间比较

综合图 3 和图 4,可以看出,文中所提出的算法,在减小网络管理过程中数据量和缩短访问对象的时间都有一定的作用。

3 结束语

作为网络管理的应用协议 SNMP 对于计算机网络管理发挥了重要作用,但对于动态的网络状态,SNMP 协议在很多方面就显得过于简单。在文中,提出了一种基于动态特征的统计方法,可以根据动态统计的结果修正轮询时间以及优先级从而使轮询过程不断优化。算法在初始状态下是根据相关配置进行周期以及优先级别的设置,随着算法的运行,经过一段时间的自适应调整,这种轮询服务达到最优,另外,可以结合拥塞控制等相关软件接口,达到对网络性能动态调控的目的。

该算法也有不足之处,主要表现在:由于服务器要

实现轮询循环列表的修正并计算优先等级以及轮询周期,这些都运行在监控服务器上,可能对服务器造成一定的负担,这也是需要进一步研究的问题。

参考文献:

- [1] 孙 霞. 基于 SNMP 的远程服务管理模型研究[J]. 计算机时代, 2009(4): 33-34.
- [2] 史爱军. 基于 web 方式的 SNMP 网络管理实现[J]. 信息通信, 2009(6): 54-55.
- [3] Rose M T. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) [S]. RFC1907, 1992.
- [4] Harrington D, Presuhn R, Wijnen B. An Architecture for Describing SNMP Management Frameworks [S]. RFC2271, 1998.
- [5] 武孟军, 徐 龚, 任相臣. Visual C++ 开发基于 SNMP 的网络管理软件[M]. 北京: 人民邮电出版社, 2007.
- [6] 李明江. SNMP 简单网络管理协议[M]. 北京: 电子工业出版社, 2007.
- [7] Martin-Flatin P, Bovert L, Hubaux P. JAMAP: a web-based management platform for IP networks[C]//Proc. of the 10th IFIP/IEEE Intl. Workshop on Distributed System: Operations and Management (DSOM 99). [s. l.]: [s. n.], 1999.
- [8] Hwang K C, Hong J J, Lee K H. A SNMP Group Polling for the Management Traffic [C]//Proc. of the IEEE Region10 Conf. . [s. l.]: [s. n.], 1999: 797-800.
- [9] 刘 翔, 沈明玉. 基于 SNMP 与 Web 的服务监控系统[J]. 合肥工业大学学报(自然科学版), 2008(1): 44-47.
- [10] Stallings W. SNMP, SNMPv2, SNMPv3 and RMON1 and 2 [M]. 3rd ed. [s. l.]: Addison-Wesley Pub Co, 1998.
- [11] 张 彤, 吴世荣. 基于 SNMP 计算机网络流量监控系统研究[J]. 计算机技术与发展, 2011, 21(1): 45-48.
- [12] 吴 昊, 杨 凯, 胡 术, 等. SNMP 跨平台移植和交换机端口 IP 探测[J]. 计算机技术与发展, 2008, 18(11): 21-24.

(上接第 134 页)

- [11] Citro S, McGovern J, Ryan C. Conflict Management for Real-time Collaborative Editing in Mobile Replicated Architectures [C]//Proc of the 13th Australasian Confine Computer Science. [s. l.]: [s. n.], 2007: 115-124.
- [12] Xue Liyin, Orgun M. Locking without Requesting a Lock: A Consistency Maintenance Mechanism in Internet-based Real-time Group Editors [J]. Journal of Parallel and Distributed Computing, 2005, 65(7): 801-814.
- [13] Liu Ling, Li Baizhan, Yang Mingyu. Research on Conflict Model for Collaborative Design in Architecture [J]. Journal of Engineering Graphics, 2006, 45(2): 189-193.
- [14] Ma Chenhua, Lu Guodong, Qiu Jiong. Conflict detection and resolution for authorization policies in workflow systems [J]. Journal of Zhejiang University (Science A: An International Applied Physics & Engineering Journal), 2009, 10(8): 1082-1092.