

基于策略和主动网技术的系统响应时间研究^{*}

王中生¹, 徐东升², 王建国¹

(1. 西安工业大学 计算机科学与工程学院, 陕西 西安 710032;

2. 榆林学院 计算机科学系, 陕西 榆林 719000)

摘要: 指出了传统网络管理存在的弊端, 在分析主动网技术和基于策略管理逻辑结构的基础上, 提出了一种基于策略和主动网技术相结合的网管逻辑体系结构来实现高效的分布式网络管理。对基于该逻辑结构网管系统的响应时间进行了理论分析, 并通过仿真实验验证了理论分析的正确性。该逻辑结构可实现管理策略实施的自动化、管理策略定义和传播机制的共享, 为定制动态的网管服务提供了支撑。

关键词: 主动网络; 策略管理; 响应时间

中图分类号: TP393.07

文献标识码: A

Study on the system response time based on the strategy and active network technology

WANG Zhong Sheng¹, XU Dong Sheng², WANG Jian Guo¹

(1. Computer Science and Engineering College, Xi'an Technological University, Xi'an 710032, China;

2. Department of Computer Science, Yulin College, Yulin 719000, China)

Abstract: This paper analysis the shortcomings of traditional network management, and constructed a policy based on active network technology and distributed network management. It also analysis the theory of response time and verified the correctness of the theory by simulation experiment. This logical structure management can be implemented automatically, it's definition and transmit mechanism can be shared. It provide a better support for the dynamic customization of network management.

Key words: active network; strategy management; response time

当今计算机网络管理是以 CS(Client/Server) 为流模式的集中式管理。在基于 SNMP 协议的计算机网管系统中, 网管工作站 NMS(Network Management Station) 是整个网络的集中控制点, 它通过轮循被管设备中的代理 (Agent) 读取被管理对象 MIB (Management Information Base) 中每个变量的值, 从而综合获得当前网络的状态, 并确定需要的控制行为。基于 SNMP 协议的网管结构特别适合于设备自身处理能力不足而网络流量不大的情况。面对网络的飞速发展, SNMP 网管模型暴露出许多缺陷: (1) NMS 负责所有网管信息的处理, 当网络扩大时, 有可能成为系统的瓶颈^[1]; (2) 没有定义 MIB 中数据的处理方式, 为了能够适应各种处理和监控方式, MIB

定义不得不尽量详尽, 传输时占用较多带宽资源; (3) NMS 只是盲目地收集原始数据, 实际的处理往往只用到部分数据, 造成资源浪费; (4) 驻留在设备上的 Agent 没有任何处理功能, 即使最简单的处理, 也必须将数据传送到 NMS, 等待返回的结果^[2]; (5) Agent 的功能在安装之后是固定的, 不能按照系统状态和管理策略变化进行调整。

由于集中式管理的种种缺陷, 人们又提出了基于策略管理^[3]的体系结构。该体系结构的一个重要目标是实现面向服务质量 QoS 和安全的、系统管理任务的自动化。它通过策略的制定、仲裁和执行对网络进行智能化的管理, 它的两个主要组成部件是策略执行点 PEP(Pol-

* 基金项目: 陕西省教育厅专项经费(陕教资 2008(147)号); 国家基金委留学基金(留金出[2006]3085)

Enforcement Point)和策略决定点 PDP(Policy Decision Point)。其中,PEP 一般处于节点上,负责执行管理策略,对网络起实质的管理作用;而 PDP 则一般处于存放策略的策略服务器上,主要负责仲裁策略,即决策使用哪项策略进行管理,当 PEP 收到某项管理任务的指示或消息时,它将该指示或消息封装成一个要求进行策略仲裁的请求发送给 PDP;PDP 经过对相关信息的检测与评估,进行策略仲裁,并从策略服务器中获取相应的策略返回给 PEP;PEP 接受策略应答并执行该项策略以完成预期的管理任务(PEP 也可能拒绝执行该项策略)。一个网络节点上也可以配置一个 PEP 和一个本地策略决策点 LPDP(Local PDP),此时,策略服务器上需配置一个全局策略决策点 GPDP (Global PDP)。PEP 发出请求时,LPDP 先作处理,如果信息不足,则发送给 GPDP 处理^[4]。

为了解决传统网管的弊端,本文结合主动网络技术和策略管理技术,提出了一种基于策略管理和主动网络技术相结合的分布式网络管理体系结构,并对该体系结构系统响应时间进行分析研究和实验仿真。

1 基于策略和主动技术的结合

基于策略的管理为智能性的计算机网络管理提供了新的解决方案,为实现面向 QoS 和安全的、系统管理任务的自动化提供了可能。但该结构还存在以下的问题需要解决或完善:首先是该结构着重考虑了基于 RSVP 协议的路由器上实施网络管理,但是当用于其他类型的服务质量技术时,PEP 的调节功能如何实现没有涉及;其次,PEP 在每次执行策略实行管理控制的时候,都需要临时动态地请求 PDP 从策略库中获取合适的策略,这对于大量的具有相同性质的网络管理功能来说,将浪费不少的时间和网络带宽资源;第三是实现管理控制的策略可能出现冲突,如何仲裁策略冲突和解决冲突,该结构中并没有体现。

主动网络(AN)^[5-6]是一种可计算的模型。它与传统 IP 网络最大的区别是其主动包可携带数据和可执行的主动代码,主动节点具有计算能力,用户可以根据管理应用的需要对网络进行编程,实现了网络应用快速增长的需要。主动网的概念应用到网络管理领域,既可以实现网络的分布式管理,又可以灵活、动态地派发新的网络管理功能^[7]。

主动网络技术是一个可对网络中间节点(主动节点)进行编程、计算的技术。其执行环境(EE)^[8]具有解析主动代码、执行主动代码以完成管理控制的功能,这一点可兼容基于策略管理中的 PEP 的功能。且 EE 只与封装主动代码的协议相关(如主动网络封装协议 ANEP^[9]、主动 IP 协议 Active IP^[10]等),而用具体的主动代码来支持区分服务 Diff-Serv 正是主动网络技术的优点之一。

为了解决上述的第 2 和 3 问题,我们利用主动网络

本身的特点和主动代码插入机制,提出了一种新的网管逻辑体系结构,该结构与传统的网管结构相似,但通过利用主动技术和策略管理技术扩展了网元,使得大量的网管功能可以在被管节点本地得到完成,从而实现分布式的网络管理。大量的中间信息也不需要传递给 NMS,从而在一定程度上节省了带宽资源。此时,NMS 不再是各个被管节点的各种状态信息的集中处理点,而仅仅是各个被管节点处理结果的汇集点以及有关全局管理功能的处理地点,从而大大减轻了 NMS 的负担。

从本质上说,一段主动代码就是一段应用程序,从管理的角度看,一段主动代码以及与其相关的数据和状态参数就是一个管理策略。所以,存放各种主动代码的代码服务器(CS:Code Server)实际上也就是基于策略管理体系结构中的策略服务器,它是策略(主动代码以及与其相关的数据和状态参数)的源。用户可以定制各种合适的管理策略(实际上就是开发进行管理控制的应用程序),通过主动代码分发机制发送到网络上的节点进行执行而实现管理控制,从而实现用户动态地定制网管服务,为新的应用快速地在网管中应用提供了支持。

各种主动代码(策略)分布式的存放在专门的 CS 中,当需要执行某个主动代码时,执行环境 EE(兼容 PEP)检测所需要的主动代码在本地代码库中是否存在,若存在,就立即调用,否则就请求从 CS 中下载并存储在本地的代码库中,以便以后可以直接启动使用。这种方式有效地解决的上述的第 2 个问题,相同的策略只需要从 CS 中获取一次,从而可以避免频繁地获取策略而浪费不少的时间和网络带宽资源。

每个节点上配置一个策略决策点(称为本地策略决策点 LPDP)和一个或多个 EE(兼容 PEP),当该节点上的软监测器(软监测器实际上是一组监测主动代码,它可能在网络上不停地流动,也可能驻留在该主动节点上)发现某个事件,如负载超限,LPDP 就作出决策,通知相应的 EE(PEP)从本地代码库或从代码服务器中调用指定的代码执行而实现管理任务。

2 系统响应时间分析

本文提出的网管系统和管理上有两种方式:第一是事件管理处理方式,即某个节点中的软监测器发现了某个事件,并且能够由本地的 LPDP 仲裁决策,由 EE 马上执行策略而处理,处理后主动向 NMS 发送处理结果。这种管理方式由于是节点主动发起的管理,称之为节点主动式。第二是 NMS 主动向相关的节点发送主动包进行某项管理任务,如拓扑发现,或者是涉及到全局决策的管理任务,如路由改变等。这种管理方式由于是 NMS 主动发起的管理,称之为 NMS 主动式。

在此,暂不考虑节点的处理时间,而只考虑网络上主动包的传输时间。由于响应时间取决于网络上传输的

数据流量和传输速率,所以先分析在执行管理任务时,网络上的传输流量。在使用的主动代码分发机制中,从代码服务器下载的主动代码可存储在节点本地的代码库中,以后可直接从本地调用,所以对于大量重复使用相同主动代码的管理任务而言,所需要的主动代码只在网络上传输一次,因此也不考虑下载主动代码的传输流量。

设在应用层上传输的数据流量为 X ,则网络上实际传输的数据流量为 $X'=\alpha(X)+\beta(X)X$,其中 $\alpha(X)$ 决定于网络传输的控制信息, $\beta(X)$ 决定于主动包的封装协议(我们使用 ANEP)。则 $X'=(\frac{\alpha(X)}{X}+\beta(X))X$,令 $\lambda=\sigma(X)=\frac{\alpha(X)}{X}+\beta(X)$,则 $X'=\lambda X$ 。我们称 λ 为主动包的加权值。

2.1 节点主动式的系统响应时间分析

这种情况比较简单,此时的响应时间是指事件发生开始到 NMS 收到处理后的结果之间的时间。我们不考虑节点的处理时间,所以我们认为节点一经监测到某个事件,立即生成相应的主动包(包含处理的结果)上报给 NMS。设 X 为上报主动包的数据量,NMS 与节点之间的平均“跳数”为 M ,主动包的加权值为 λ ,则网络上的数据流量为 $Y=\lambda MX$ 。设网络传输速率为 v ,则响应时间为: $T=\frac{Y}{v}=\frac{\lambda MX}{v}$ 。

2.2 NMS 主动式的系统响应时间分析

这种情况较为复杂,为了方便分析系统的响应时间,清楚地表明该管理系统的主动包的传输情况,我们将主动包传输表示成如图 1 所示。主动包传输流程是:首先 NMS 产生一个包含有主动代码的主动包,该主动包指示完成某项管理功能;然后将该主动包发送到第一个节点上;节点中的 EE 解析主动包,从本地代码库或从代码服务器中调用指定的代码执行而实现管理任务;然后生成新的主动包(有完成管理任务的结果和原来的主动信包组成),并将新的主动包分发到下一个节点,直到所有节点处理完成后,将最后生成的主动包上报给 NMS。

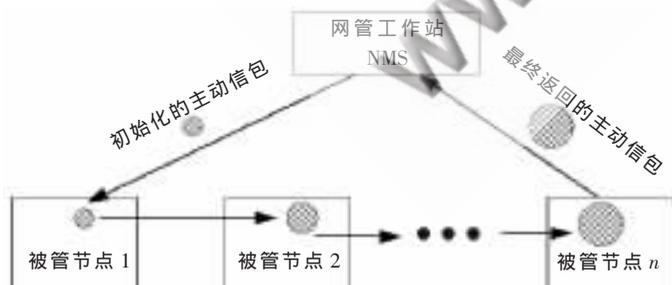


图 1 主动信包流动示意图

此时的响应时间包括 NMS 发出主动包开始到 NMS 收到最后一个节点生成的主动包(包含所有节点完成管理任务后的结果)之间的时间。

系统的数据流量分成三个部分,第一部分是 NMS

发送的初始化主动包的流量,用 Y_{NMS} 表示;第二部分是节点之间传输的数据流量,用 Y_{NODE} 表示,第三部分是最终返回给 NMS 的主动包的流量, Y_{NMS}' 表示。

设主动包的原始大小为 X_0 ,每个节点完成管理任务后的平均结果流量为 X ,被管节点的个数为 N ,NMS 与节点之间的平均“跳数”为 M 则:

$$Y_{NMS}=\lambda MX_0;$$

$$Y_{NMS}'=\lambda M(X_0+NX);$$

用 $Y_{NODE,n}$ 表示从第 $n-1$ 个节点传输到第 n 个节点的处理结果流量。当 $n=1$ 时,表示只有一个节点,不存在节点之间的传输流量,所以 $Y_{NODE,1}=0$,当 $n>1$ 时,从第 $n-1$ 个节点传输到第 n 个节点的处理结果流量是前 $n-1$ 个节点的所有处理结果流量之和,所以,

$$Y_{NODE,n}=\lambda \sum_{i=1}^{n-1} X=(n-1)\lambda X;$$

因此 N 个节点之间传输的主动包流量 Y_{NODE} 为:

$$Y_{NODE}=(N-1)\lambda X_0+\sum_{i=2}^N Y_{NODE,i} \\ = (N-1)\lambda X_0+\frac{N^2-N}{2}\lambda X$$

所以,设网络传输速率为 v ,则系统的响应时间 T 为:

$$T=\frac{Y_{NMS}+Y_{NODE}+Y_{NMS}'}{v} \\ =\lambda \frac{2(2M+N-1)X_0+(M+N-1)NX}{2v}$$

从上式可以看出,影响系统响应时间的参数有:节点与 NMS 之间的平均“跳数” M ,被管节点的个数 N ,初始化的主动包大小 X_0 ,节点处理后的平均结果流量以及网络的传输速率 v 。下面通过实验具体分析节点与 NMS 之间的平均“跳数” M ,被管节点的个数 N ,以及网络的传输速率 v 对系统响应时间的影响。

3 仿真实验结果

我们采用仿真的方法进行实验验证上述推导的响应时间。实验的网络环境是内部局域网,使用 PC 机仿真被管节点。在整个实验过程中,在网络上传输的全部是主动包。主动包的封装格式(暂不考虑安全方面的参数项)遵循 ANEP 协议,如图 2 所示。第一项是“ANEP 报头”是遵循 ANEP 协议而封装,具体报文头可参见文献 [7];第二项是“主动代码指针”,指示应当调用的主动代码,该指针是一个全局标示符;第三项是“数据”,具体包括调用主动代码所需要的数据以及处理的状态数据和返回给 NMS 的结果数据,所以该项是变长的。实验过程中对节点发送“Ping”应用主动包,返回的结果是节点的 IP 地址和是否能够 Ping 通的状态标志。

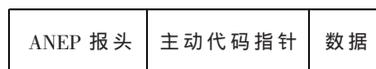


图 2 主动信包格式

首先,我们分析节点与 NMS 之间的平均“跳数” M ,被管节点的个数 N 对系统响应实现的影响。分别实验了 M 为 1, 5, 10 的情况,实现结果如图 3 所示。本实现中,仿真的最大节点个数 N 为 30, NMS 发出的初始化主动包的大小 X_0 为 78 Bytes, 节点处理后的返回结果大小 X 为 6 B, 网络的传输速率为 10 Mb/s。

从图 3 可看出,当“跳数” M 一定时,节点个数 N 对响应时间的影响交大,且当 N 增加到一定的个数时,响应时间呈加速的形式增加,从上述研究推导的公式子可以看出,响应时间是被管节点个数 N 的二次多项式函数。可见实验结果与推导的公式是相吻合的。

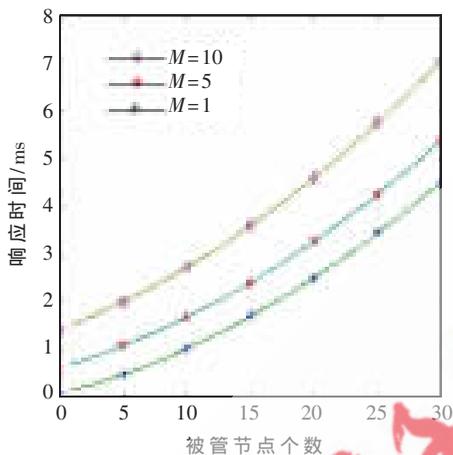


图 3 跳数和节点个数对响应时间的影响

其次,分析网络传输速率 v , 被管节点的个数 N 对系统响应时间的影响。分别实验了 v 为 2 Mb/s, 8 Mb/s 和 10 Mb/s 情况,实现结果如图 4 所示。本实现中仿真的最大节点个数 N 为 30, 跳数 M 为 5, NMS 发出的初始化主动包的大小 X_0 为 78 B, 节点处理后的返回结果大小 X 为 6 B。

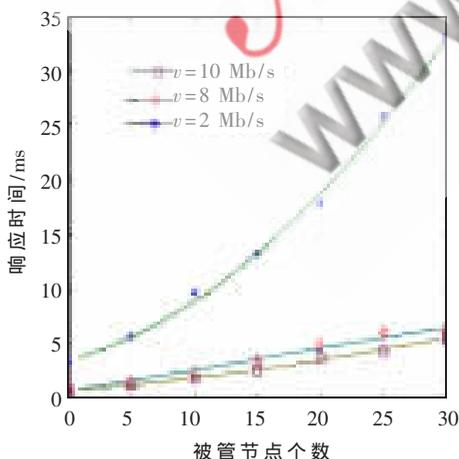


图 4 传输速率和节点个数对响应时间的影响

从图 4 可看出,网络传输速率 v 越大,系统的响应时间就越小。这与上述推导的公式也是相吻合的,从公式可以看出,当其它参数保持不变时,系统的响应时间与网络的传输速率成反比。

主动网是当前网络体系结构研究的热点,主动网技术已经被应用到许多网络中(如多播、网络拥塞等),把主动网技术应用到网络管理中,是主动网应用的一个新领域。本文提出了一种基于主动网技术和策略管理相结合的网管逻辑体系结构以实现高效的分布式网络管理,同时对该结构系统的响应时间进行了研究分析,并对研究结果进行实验仿真。这种网管逻辑结构集中了主动网技术和策略管理的优点,克服了策略管理的缺点;可实现管理策略实施的自动化,管理策略定义和传播机制的共享,为动态定制网管服务提供了支撑;大量的管理任务可以在节点本地得到处理,从而减轻了 NMS 的负担,提高了系统的响应时间,减少了网络流量,节省了网络带宽资源。

参考文献

- [1] 郭健,吴伟明,张爱霞.面向业务的 NGN 综合网管系统的研究[J].数据通信,2005(3):9-12.
- [2] 李瑛,王建国.基于主动代码的面向业务的网络管理体系结构研究.西安工业大学学报,2006(6):16-18.
- [3] WATSON T J. Policy-based Diffserv on Internet servers: The AIX approach. IEEE Internet Computing, 2000,4(5):75-80.
- [4] 徐东升,袁飞云,赵政文.主动网技术和策略相结合的网络管理模式[J].信息安全与通信保密,2006,5.
- [5] TENNENHOUSE D L, SMITH J M, SINCOSKILE W D, et al. Wetherall, A Survey of Active Network Research. IEEE Comm. Mag, 1997,35(1):80-86.
- [6] CALVERT K L. Directions in Active Network. IEEE Comm. Mag., 1998,36(10):72-78.
- [7] SCHWARTZ B, et al. Smart Packets for Active Network, BBN Technologies draft, <http://www.net-tech.bbn.com/smt-pkts/smart.ps.gz>, 1998.
- [8] CALVERT K L. Architecture Framework for Active Networks Version1, Active Network Working Group, Draft, July, 1999.
- [9] ALEXANDER D S, et al. Active Network Encapsulation Protocol. Draft, <http://www.cis.upenn.edu/SwitchWare/ANEP>, July 1997.
- [10] WETHERALL D, TENNENHOUSE D. The Active IP Options. In Proceedings of the 7th ACM SIGOPS European Workshop, Connemara, Ireland, Sep 1996.

(收稿日期:2009-03-11)