

面向物联网环境的网络管理消息转换机制研究

吴黎兵^{1,2}, 刘冰艺¹, 聂雷¹, 何炎祥¹, 魏学将¹, 杨科¹

¹(武汉大学 计算机学院, 武汉 430072)

²(武汉大学 软件工程国家重点实验室, 武汉 430072)

E-mail: wu@whu.edu.cn

摘要: 针对物联网的特点, 分析当前 SNMP 协议难以满足物联网网络管理发展的现状, 结合 NETCONF 协议在物联网网络管理的优势, 提出将 NETCONF 用于物联网网络管理. 结合物联网特征建立基于 NETCONF 的管理端、代理端, 提出并实现了一种网络管理消息转换机制, 该消息转换机制具有自适应能力, 将基于 XML 的管理消息自适应地转换为当前代理需要的报文格式. 消息转换器主要针对当前网络环境下广泛应用的网络管理标准 SNMP 协议, 实现了与 SNMP 的兼容, 探讨了与 ANMP 和其他网络管理协议的转换方法. 最后, 通过实验证明了这种消息转换机制的可行性和有效性, 为物联网环境下融合异构协议的网络管理平台提供理论依据和参考.

关键词: 物联网; SNMP; NETCONF; 自适应能力; 消息转换机制; 异构协议; 网络管理平台

中图分类号: TP393

文献标识码: A

文章编号: 1000-1220(2015)01-0001-06

Research on Translation Mechanism of Network Management Message in Internet of Things

WU Li-bing^{1,2}, LIU Bing-yi¹, NIE Lei¹, HE Yan-xiang¹, WEI Xue-jiang¹, YANG Ke¹

¹(School of Computer Science, Wuhan University, Wuhan 430072, China)

²(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

Abstract: With the features of the Internet of Things (IoT), the current SNMP protocol can hardly keep up with the development of network management of IoT, which the NETCONF protocol can qualify. In these circumstances, we propose the NETCONF protocol could be applied to network management of IoT according to its strengths in it and develop the manage-side and agent-side based on NETCONF. More importantly, we achieve an adaptive message transformation mechanism, which can automatically convert XML-based messages to the messages needed by particular agent. Focused on the widely used SNMP at present, we achieve the compatibility of it and explore the feasibility of ANMP and some other protocols in network management. Finally, the experiments confirm the feasibility and availability of this message transformation mechanism, providing a theory basis and reference for network management platform based on heterogeneous networks and protocols in the future.

Key words: Internet of Things (IoT); SNMP; NETCONF; adaptive; message transformation mechanism; heterogeneous protocols; network management platform

1 引言

物联网在当前互联网的基础上有了许多新的特点, 因此对于物联网网络管理提出了新的需求. ISO 定义的网络管理 5 个系统管理功能在物联网时代难以满足新的需求, 传统的基于简单网络管理协议(SNMP, Simple Network Management Protocol)的网络管理系统, 不管在扩展性方面还是管理效率方面的局限性日益突出, 因此迫切需要新的网络管理系统模型.

可扩展标记语言^[1](XML, eXtensible Markup Language)的出现为构建物联网环境下的网络管理模型提供了可能. 基于 XML 的网络管理系统具有其他网络管理系统无法比拟的特性, 这些特性使得它非常适用于物联网网络管理.

基于上述优势, 为了统一规范基于 XML 的网络管理, IETF 在 2006 年提出了基于 XML 技术的 NETCONF^[3]

(RFC4741) 协议. NETCONF 的提出不仅使基于 XML 新一代网络管理配置方面的功能得以加强, 形成了结构明晰的规范, 也使得 XML 网络管理的效率得到显著提升. 基于 NETCONF 的网络管理已经得到广泛认可, 事实上一些 IT 公司也开发并实现了支持 NETCONF 的相关系统, 如 Juniper 公司的 JUNOS^[4]产品、Cisco 公司的 IOS^[5]产品, 相关开源产品有 en-suite^[6]的 yencap + manager 和 yuma^[7]等. 对于物联网这种融入各种异构网络的网络系统, 更加迫切需要基于 NETCONF 的网络管理系统实现对网络设备跨网络、跨系统的高效管理.

物联网相对于传统的互联网具有自身的一些显著特征, 比如, 网络拓扑变化很快; 网络节点可以高速移动; 节点间的链路状态变化频繁; 节点能量、计算能力、存储能力有限等. NETCONF 取代事实上的工业标准 SNMP 协议将会是一个漫长的过程, 有必要通过一种转换机制, 实现既能管理 SNMP 的

收稿日期: 2013-09-02 收修改稿日期: 2013-09-30 基金项目: 国家自然科学基金项目(61272112, 61472287)资助. 作者简介: 吴黎兵, 男, 1972 年生, 博士, 教授, 研究方向为网络管理、无线传感器网络等; 刘冰艺, 男, 1990 年生, 博士研究生, 研究方向为无线传感器网络、网络管理; 聂雷, 男, 1989 年生, 博士研究生, 研究方向为无线传感器网络、网络管理; 何炎祥, 男, 1952 年生, 博士, 教授, 研究方向为可信软件、分布并行计算和软件工程; 魏学将, 男, 1981 年生, 博士研究生, 研究方向为物联网; 杨科, 男, 1988 年生, 硕士研究生, 研究方向为分布式计算和网络管理.

网络设备,又能对基于其他网络协议的设备进行有效管理.将 NETCONF 应用于物联网网络管理,首先必须实现对 SNMP 的兼容,能将基于 NETCONF 的管理报文转换为 SNMP 管理报文.目前,协议转换研究吸引了许多网络管理专家的注意,但是大多数停留在理论阶段,或者研发的系统扩展性较差,针对物联网这一复杂网络环境下的消息转换方法研究很少.本文设计的消息转换机制可以实现对目前广泛采用的 SNMP 协议、非 SNMP 协议(比如 ANMP、NETCONF)的支持,并且结合物联网特点设计 NETCONF 管理端、代理端,便于今后在物联网这个大环境下实现网络设备的统一管理.

2 相关工作

在转换网关方面,为了弥补 SNMP 协议在网络管理扩展性以及效率上的不足,文献[8]提出一种 SNMP MIB 到 XML 的转换算法,并将转换算法应用于 SNMP-XML 转换网关.文献[9]重点讨论了 SNMP-XML 翻译网关中的 MIB 转换技术,并实现了 MIB 文件到 XML 文件的转换.文献[10]提出一种基于 NETCONF 的网络管理系统对 SNMP 和 CLI 设备进行管理的方法,文章主要分析数据模型的转换以及消息映射.文献[11]提出了一种通用网关模型,实现基于 XML 网络管理对 SNMP 代理和非 SNMP 代理的统一管理.

在 NETCONF 协议的分析和应用开发方面,文献[12]通过实验分析证明了 NETCONF 在复杂网络环境下的强大性能,实验结果显示了 NETCONF 相对于 SNMP 和 CLI,在网络管理上更高效、更安全、扩展性更强、更容易开发新的应用.文献[13]对 NETCONF 的三种建模语言 XML Schema、Relax NG 和 YANG 进行了对比分析,得出尽管 YANG 是专门为 NETCONF 设计的建模语言,但是仍然有些地方考虑不足,比如转换工具设计得并不完善.

上述文献大多数停留在理论架构阶段,或者仅仅对某个单一功能进行实现,例如文献[8]提出的转换器安全性不高,数据在传输过程中容易被劫取,而且使用的 XML 管理报文没有形成统一规范.文献[9-10]主要针对数据模型转换,文献[11]主要停留在管理消息转换的架构设计上,在实现方面只做了简单的描述.尽管 NETCONF 有众多优点,但是 SNMP 作为事实上的网络管理工业标准有很多不可替代的特性,比如它的简易性、实用性,以及它对设备的实时监控性优于 NETCONF 协议,这些决定了未来 SNMP 协议将长期存在于网络环境中,单纯地将 NETCONF 应用于物联网不切实际.本文基于 NETCONF 协议,提出面向物联网网络管理的消息转换机制,实现对 SNMP 的兼容,同时也考虑了对其他网络管理协议的支持和系统的扩展性,为物联网环境下网络设备的统一管理提供了参考.

3 基于 NETCONF 网络管理架构

3.1 基于 NETCONF 管理端

管理端一共包含 3 个模块,如图 1 所示,分别是交互界面、管理消息处理层、会话通信层.

交互界面负责与管理员进行信息交互.消息处理层是管理端的核心模块,负责将管理消息封装成基于 NETCONF 的管理报文,并传送给会话通信层.另外还负责验证收到的报文

格式,解析出操作结果.内容层封装是根据采用的数据模型对报文进行封装.操作层封装、RPC 封装则根据用户选择的操作类型将报文封装到相应的 RPC 报文中去.会话通信层对应 NETCONF 逻辑模型中传输层,负责将管理消息传输给消息转换器,并等待消息转换器的响应,将响应结果返回给消息处理层.

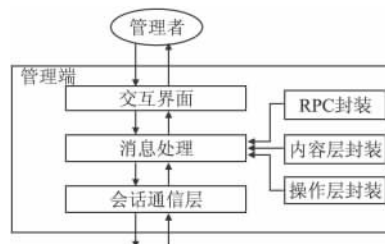


图 1 管理端架构设计

Fig. 1 Architecture of manager-side

Service 作为管理端与消息转换器之间报文的传递,符合在物联网下网络管理消息传递的特性要求,更容易实现跨平台、跨设备、跨网络对网络设备的监管.

消息转换器对发送过来的管理报文进行相应转换,使得网络管理端可以与不同类型的代理端进行通信,消息转换器以 Web Service 的方式发布,实现与管理端交互,并且直接与物联网环境下不同类型的代理进行信息交互.

消息转换器的整体架构如下页图 2 所示.主要分为三个功能模块,即消息分类器 SNMP、报文转换模块,其它代理报文转换模块.

3.2.1 NETCONF-SNMP 管理消息转换

负责将 NETCONF 管理报文转换为 SNMP 管理报文的转换器主要由 6 个模块构成,分别是请求分析器、MIB-XML 翻译器、XML DOM、XML 询问器、trap 处理模块(由 trap 接收器、trap 分析器和 trap 过滤器组成)以及报文生成器.

请求分析器结合 XML Schema 判断管理报文的合法性,并且结合操作类型映射表提取操作类型. MIB-XML 翻译器负责将 SMI MIB 转换为 XML,转换后的 XML,可以实现查找操作对象对应的 OID 和操作对象的映射. XML DOM 是转换器的核心,对 XML 文件进行分析,将得到的设备地址、操作类型、操作对象 OID 等,传给 SNMP 轮询器,还负责接收 trap,实现 trap 中参数映射后,传递给 XML 报文生成器. SNMP 轮询器将得到的参数包装成 SNMP PDU 传给 SNMP 代理,并且接受来自 SNMP 代理的响应.为了减少管理端与转换器之间的通信流量,对 SNMP trap 处理上采用过滤机制,trap 处理器由 3 个模块组成,将接收到的 trap 进行分类,并建立分级制度来判断紧急程度,最后过滤掉一些重复或者失效的 trap 报文,并传递给 XML DOM 报文生成器生成相应的响应报文或通知报文后传给 NETCONF 管理端.

NETCONF-SNMP 转换器也可以实现对 ANMP 代理的兼容性,这在于 ANMP 使用与 SNMP 相同的 PDU 格式,而且同样使用 UDP 作为传输协议来发送 ANMP 消息.在数据收集和控制方面,ANMP 扩展了 SNMP MIB 以便记录 Ad Hoc 网络特有的信息.若要对 ANMP 代理进行管理,首先管理端载入对应的 MIB 文件,消息分类器判断管理报文中的 IP 地址,若对

3.2 消息转换器架构

本文的消息转换器基于 Web Service 进行通信.基于对 XML 的广泛接受,Web Service 成为使用标准传输、编码和协议来交换信息的应用程序.选择 Web

应的是 ANMP 代理时, 将管理报文传给转换器, 可以实现对 ANMP 代理对应设备的有效管理。

3.2.2 对其他网络管理协议的支持

考虑到物联网环境下存在少量其他网络管理协议, 本文加入了一种支持其他网络管理协议的理论构架, 下面对这种架构进行阐述。

如图 2 所示, 架构主要有四部分组成, 分别是数据表、适配器、消息产生器、Trap 接收器。其中三个数据表和三个适配器是架构的关键, 设备信息表记录代理的相关信息, 用以区分管理端与哪个代理通信; 私有数据表将 NETCONF 的属性映射为代理的私有属性; 操作表则将 NETCONF 的操作类型映射为代理的私有操作类型。报文适配器实现报文格式对应; 传输适配器负责转换器与多种代理进行通信; trap 适配器则负责代理如何主动与管理端进程通信, 通知管理进程有某些事情发生。

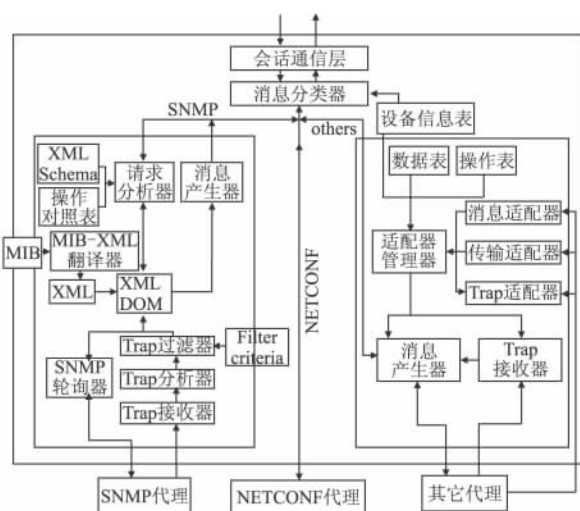


图 2 消息转换器架构

Fig. 2 Architecture of information translator.

当管理端与代理通信时, 转换器首先载入该代理的 XML 配置文件, 生成三个数据表, 分别是设备信息表、操作类型表、私有数据表。通过数据表生成上述三个适配器的具体实现, 适配管理器负责初始化适配器并在合适的时候调用适配器。在适配管理器的协调下, 消息产生器就能将管理端传递过来的 NETCONF 配置报文通过适配器转化为代理能够识别的 PDU, 返回的 PDU 也能通过管理消息产生器转换为基于 NETCONF 的响应报文。

3.3 代理端架构设计

目前 Cisco, Juniper 等网络设备生产商都实现了基于 RFC4741 的代理, 并嵌入到了它们最新的路由器当中。

NETCONF 采用 XML 进行数据传输和模块表达, 具有较强的可扩展性, 网络设备提供商可以使用此协议获取、设置所有的配置数据, 适合物联网下的不同类型设备快速添加和高效管理。这些功能很大一部分依赖于代理端的实现, 如何将基于 NETCONF 的代理端有效应用到不同类型的网络设备, 并对其进行网络管理是一个迫切需要解决的问题, 这关系到 NETCONF 在物联网网络管理的生命力。本文将代理分为三种

类型, 分别是 SNMP 代理、NETCONF 代理和其他代理。

本节结合物联网的特征分析了 NETCONF 代理架构设计, 按照 RFC4741 中规定, 一个基本的 NETCONF 代理分四层结构来设计。另外代理必须完成对能力特性、三种数据库状态和事件通知的支持。基于 NETCONF 的代理架构如图 3 所示。

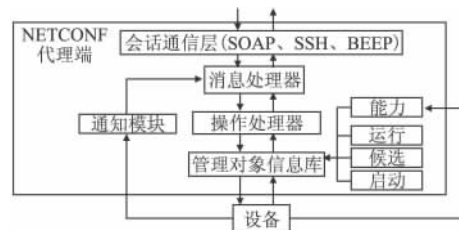


图 3 NETCONF 代理端架构

Fig. 3 Architecture of agent-side.

会话通信层负责与管理端交互, 代理端启动后会监听来自管理端的管理消息。消息处理器接受来自会话通信层的管理消息后, 能够解析出操作类型和操作对象并传给操作处理器, 也能将操作处理器操作后的结果封装成基于 NETCONF 的响应报文。操作处理器执行消息处理器解析出来的具体操作。管理对象信息库中配置信息状态分为 3 个阶段, 对应 3 种数据库状态: 运行状态 (running)、启动状态 (start up) 和候选状态 (candidate)。能力 (capabilities) 是 NETCONF 的新特性, 这种特性允许客户端发现服务端支持的协议扩展集, “能力”的提出丰富了基本操作集, 增加新的操作使代理端扩展性得到提高。被管理设备将能力集传递给管理数据库存储, 在管理端发出 “HELLO” 报文后, 传递给管理端以告知管理端被管设备支持的协议扩展集。通知 (Notification) 模块负责将被管理设备主动发出的消息传递给消息处理器, 再由消息处理器封装后, 通过会话通信层传递给管理端。

本文并没有将重点放在讨论 “能力” 和 “通知” 这两种特性上, 但这两种特性对于将 NETCONF 应用到物联网环境中至关重要, 因为物联网环境需要 “能力” 来支持可扩展性, “通知” 来支持对被管设备的实时监控, 这也是我们未来工作讨论的重点。

4 转换算法流程设计

4.1 数据模型分析

与 SNMP 相比, NETCONF 是一个全新的 XML 配置管理协议。NETCONF 协议在概念上分为四层, 分别是内容层、操作层、RPC 层和传输协议层。目前操作层、RPC 层、传输协议层都有相应的标准和规范, 内容层并没有给出具体要求, 允许单独定义 NETCONF 数据模型, 使得它的灵活性增强, 但是另一方面也阻碍了 NETCONF 的广泛应用。数据模型定义与传统数据模型的转换是目前各大国际化标准研究的重点和热点。2009 年, NETMOD 工作组提出将 YANG^[14] 作为标准的 NETCONF 数据建模方法目前还处于讨论和验证中。现有的数据模型语言, 如 XML Schema Definition (XSD) 和 Relax NG 可以用于其数据模型。正因为 NETCONF 基于 XML, 所以 XSD 是其中一种比较理想的选择, 本文也是采用 XSD 作为数据建模语言来开展实验。IETF 组织, 特别是 NETCONF 工作组将 XML

Schema 列入考虑之中。

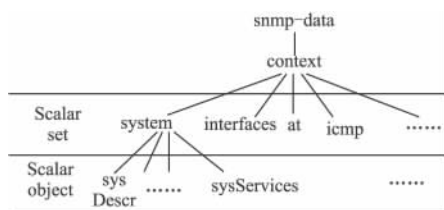


图4 管理数据模型

Fig.4 Management data model.

德国开发的 LIBSMI^[15] 是比较成功的案例,得到了广泛

的认可,它将 MIB 树转换为公认较好的四层结构,如图 4 所示。前两层与具体 MIB 无关,只有下两层才依赖于具体 MIB,第三层为表结点的 ENTRY 结点,或者是标量结点的容器结点,第四层是叶子结点。对于非 MIB 树形式的管理对象也可以建立这样的四层结构模型,这样统一了网络被管对象的描述。本文在实验中便采用了该四层结构来规范管理对象数据。

4.2 关键类设计

为了实现基于 XML 的 NETCONF 报文到 SNMP 报文的转换,需要对 NETCONF 报文进行解析。基于第 2 节中的转换器架构,本文设计的关键类有: ParseXML 类、XmlDocument 类、Operations 类以及 UdpTarget 类,关键类图如图 5 所示。

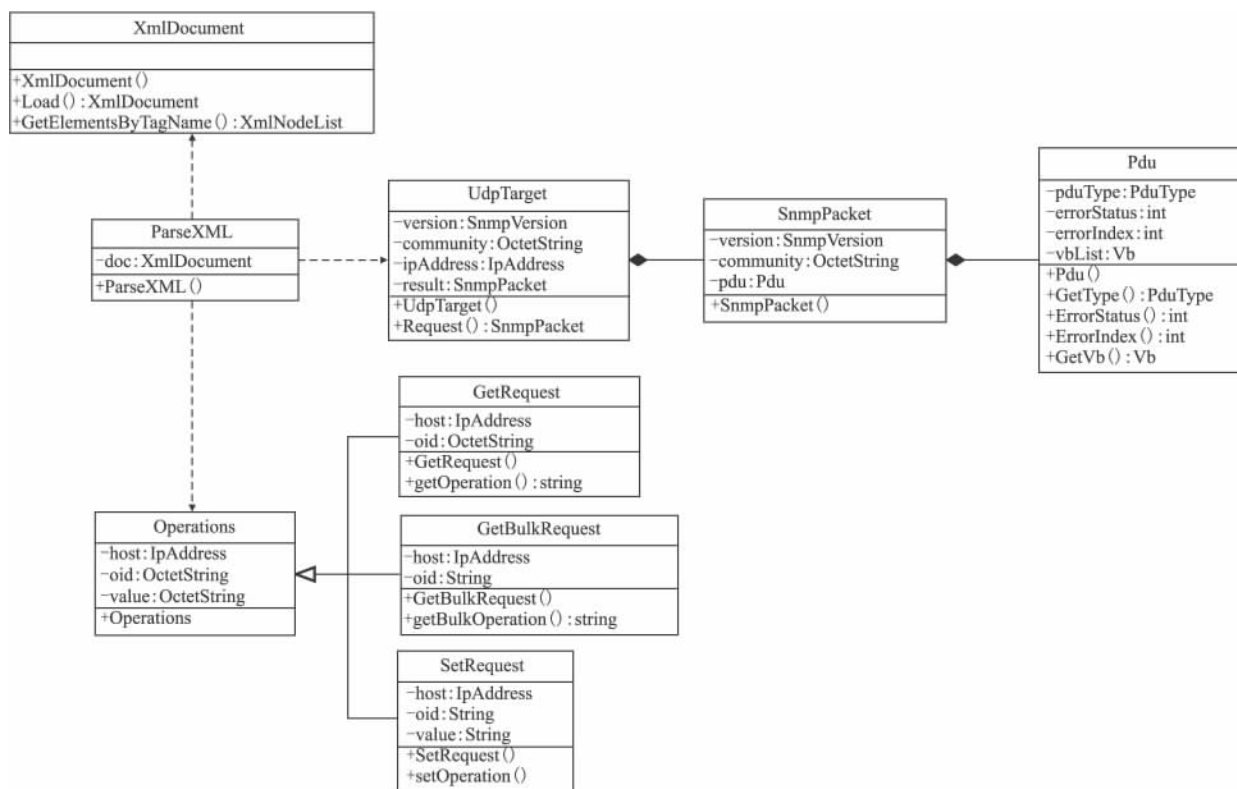


图5 关键类图

Fig.5 Class diagram

ParseXML 类用于对 NETCONF 报文进行解析,其实现依赖于其他三个关键类; XmlDocument 类主要用来提取基于 XML 的 NETCONF 报文的关键信息,如设备 IP 地址、操作对象 OID、操作类型等; Operations 类提供三种基本 SNMP 操作: GetRequest、GetBulkRequest 和 SetRequest,这为实现 SNMP 的基本功能提供了可能; UdpTarget 类主要用来构造 SNMP PDU,并依据具体操作发送相应 SNMP 请求报文。

4.3 转换算法流程设计

转换流程如图 6(见下页)所示。

5 实验与分析

5.1 实验环境和参数

采用 Visual Studio 2010 作为开发平台,编程语言是 C#。实验选择了四台机器,一台机器作为基于 NETCONF 的网络管理端,一台机器作为报文转换器,一台机器作为 SNMP 代

理,一台作为 NETCONF 代理,SNMP 代理端和 NETCONF 代理端分别通过 RS232 与汇聚节点相连,通过 USB 与和 RFID 读卡器相连。汇聚节点通过 ZigBee 协议与静态节点相连。实验环境如图 7(见下页)所示。

5.2 运行实例

本文选取 SNMP 代理和 NETCONF 代理进行实验,目标是通过基于 NETCONF 管理端是获取设备的 sysUpTime。

点击“import object of management”,管理对象以树形结构显示在对象浏览器窗口中选择管理设备。输入代理的 IP 地址再分别选择操作类型,数据库状态,操作对象,点击“createMessage”,管理端产生基于 NETCONF 管理报文并显示在发送窗口中。点击“sendMessage”,发送报文,等待转换器响应后,响应报文显示在报文接收窗口中。

若对应的是 SNMP 代理,接收窗口直接显示结果,发送报文以及返回报文如图 8(见下页)所示。

若对应的是 NETCONF 代理,接收窗口显示基于 NETCONF 的 rpc-reply 报文,发送报文及返回报文如图 9 所示。若对应的是 NETCONF 代理并且发送报文缺少 message-id,则返回基于 NETCONF 的 rpc-error 报文,封装在 rpc-reply 中,发送

和返回的报文如图 10 所示。

5.3 实验分析结果与改进

图 11 和图 12 分别表示系统测试 20 组,每组 100 次随机 get-config 操作的响应时间比较。测试的响应时间类型分别是

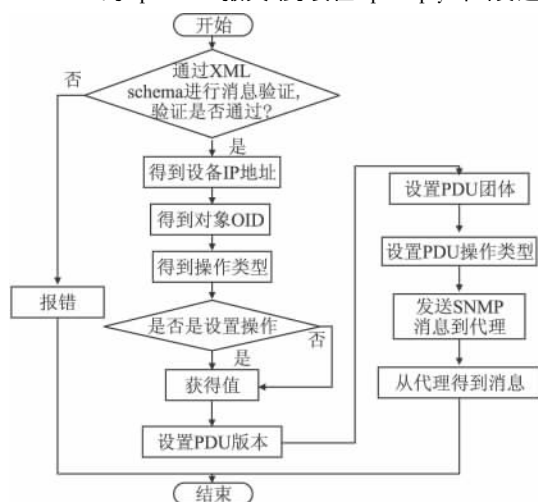


图 6 算法流程

Fig. 6 Algorithm flowchart.



图 8 从 SNMP 代理获取 SysUpTime

Fig. 8 Obtain SysUpTime from SNMP agent



图 9 从 NETCONF 代理获取 SysUpTime

Fig. 9 Obtain SysUpTime from NETCONF agent

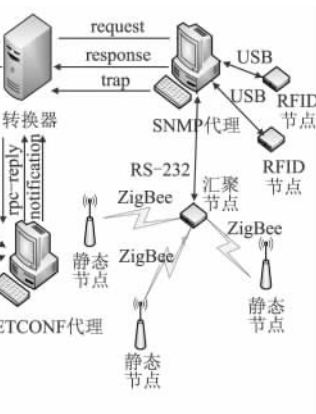


图 7 实验环境

Fig. 7 Experimental environment



图 10 缺少 message-id 的管理报文

Fig. 10 A message missing message-id

响应总时间 T_{all} 、管理端到转换器传输时间 T_{MT} 、转换器处理时间 T_{TR} 、转换器到代理端响应时间 T_{TA} 。本文在测试时,第一次响应时间作为特例不考虑,这是因为转换器以 Web 服务的方式发布,通过 TCP 三次握手与管理端建立连接,而转换器与代理之间通过 UDP 通信,UDP 不需要建立连接。由图 11 和图 12

可知,四种类型响应时间在一个相对稳定的区间内波动,但是 T_{all} 和 T_{MT} 明显高于 T_{TR} 和 T_{TA} 。经过分析,这主要是因为两方面原因:首先基于 TCP 的报文传输时间明显高于基于 UDP 的报文传输时间,另外在包的大小方面,基于 NETCONF 的管理端发送的是 XML 报文,相对于 SNMP 只需变量绑定要复杂。

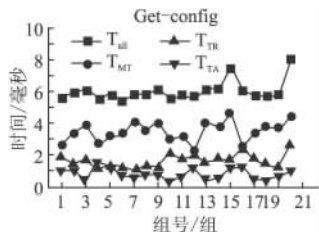


图 11 Get-config 响应时间比较图

Fig. 11 Comparison of Get-config response time

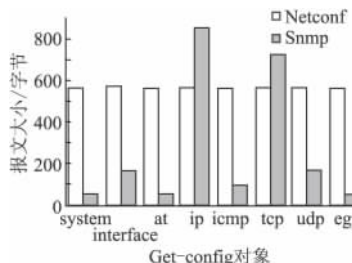


图 12 报文大小比较图

Fig. 12 Comparison of package size

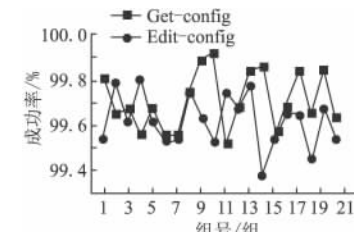


图 13 系统成功率

Fig. 13 Success rate of the system

图 12 表示的是管理端发送的报文与经过转换后的报文大小的变化比较,实验只考虑 TCP 或 UDP 报文段的数据部分。本文测试用 mib-2 中前 8 个基本对象组,对于 MIB 树中非

叶子结点的实例,转换器自动调用 GetBulkOperation 与代理交互。由图 13 可知,NETCONF 管理端的管理报文大小几乎是稳定不变的,通过转换器转换后的 SNMP 管理报文大小的变化

波动相对较大. 通过分析, 这主要因为考虑到 8 个基本对象组中标量对象的个数不尽相同, 权衡代理的处理时间和转换后报文数量后, 实验将 SNMPv2 GetBulk 报文的 NonRepeaters 字段设置为 0, MaxRepetitions 字段设置为 20. 对于标量对象大于 20 的对象组需要发送多个 GetBulk 请求. 今后这部分可以优化, 根据实际需要自动生成这两个参数来减少转换后的报文大小. 测试转换前后报文大小变化为今后转化器位置部署提供了参考.

图 13(见上页) 表示的是随机产生 20 组, 每组 10000 次管理报文得到正确响应报文的成功率, 失败的操作将会返回错误类型. 通过实验发现两种操作的成功率稳定在 99.4% 以上. 通过返回的错误类型发现, 失败与否主要与当前网络和系统状况有关. get-config 操作的成功率总体上高于 edit-config 的成功率, 这是因为配置操作相对于取值操作受限更多.

实验选择了响应时间、报文大小变化、成功率来评价系统的性能. 实验主要考虑的是 NETCONF 管理端与 SNMP 代理的兼容性. 数据模型采用的是在 4.1 节介绍的基于 XML 的四层结构. 即将 MIB-2 库通过 smidump 转换成实验所需要的模型. 实验过程中还发现, 在转换过程中存在 MIB 树中间结点丢失的情况, 由此可见转换 MIB 信息表达效率不够精确.

6 结 论

为了在物联网环境中部署一种基于 NETCONF 的统一网络管理系统, 考虑到当前 SNMP 在配置性能、安全性和扩展性方面的不足, 以及 SMI 语法复杂和灵活性差等问题, 文章提出了面向物联网环境下网络管理消息转换机制, 它能够自适应地对各种设备进行管理, 特别是将这种转换机制通过 Web Service 的方式提供给管理端, 在安全性、可扩展性、配置效率上有很大程度提高. 本文还提出了管理端、转换器、代理端的设计架构, 通过实验验证了系统对 SNMP 代理的兼容性, 能够对 SNMP 设备进行有效管理.

未来工作中我们将研究重点集中在数据模型、可扩展性、被管设备监控性能优化三个方面, 最终实现物联网环境下网络设备的统一高效管理.

References:

- [1] Straub F, Klie T. Towards xml oriented Internet management [C]. Proceedings of IEEE International Symposium on Integrated Network Management 2003: 505-518.
- [2] Wu Xiao-qiong, Xiao De-bao, Chang Ya-nan. Snmp mib data conversation based on netconf [J]. Computer Engineering, 2010, 36(2):

64-66.

- [3] Enns R et al. Netconf configuration protocol [R]. RFC4741, Network Working Group 2006.
- [4] Juniper JUNO Software [EB/OL]. http://www.juniper.net/products_and_services/junos/ 2012.
- [5] Cisco. IOS software [EB/OL]. http://www.cisco.com/web/CN/products/products_netsol/ios/index.html/ 2012.
- [6] IETF. NETCONF working group [EB/OL]. <http://www.ops.ietf.org/netconf/> 2012.
- [7] Yuma. YANG-based unified modular automation tools [EB/OL]. <http://sourceforge.net/projects/yuma/> 2012.
- [8] Jeong-Hyuk Y, Hong-Taek J, Hong J W. Development of snmp-xml translator and gateway for xml-based integrated network management [J]. International Journal of Network Management, 2003, 13(4): 259-276.
- [9] Wu Li-bing, Fu fan, He Yan-xiang et al. Research on mib data conversation in xml-based network management [J]. Journal of Wuhan University of Technology (Information & Management Engineering), 2011, 33(6): 851-854, 863.
- [10] Wu Bao-zhen, Chang Yanan. Integrating snmp agents and cli with netconf-based network management systems [C]. Proceedings of Computer Science and Information Technology 2010: 81-84.
- [11] Xiao De-bao, Zhao Zhi-xia, Xiong Lei. Research and implementation on model of general translation gateway based on netconf [J]. Journal of Huazhong Normal University 2007, 41(4): 515-519.
- [12] Yu J, Al Ajarmeh I. An empirical study of the netconf protocol [C]. Proceedings of Networking and Services (ICNS) 2010: 253-258.
- [13] Cui Hui-yang, Zhang Bin, Li Guo-hui et al. Contrast analysis of netconf modeling languages: xml schema, relax ng and yang [C]. Proceedings of Communication Software and Networks 2009: 322-326.
- [14] Bjorklund M. Yang-a data modeling language for the network configuration protocol (NETCONF) [S]. RFC6020, IETF 2006.
- [15] libsmi-a library to access SMI MIB information [EB/OL]. <http://www.ibr.cs.tu-bs.de/projects/libsmi/smidump> 2012.

附中文参考文献:

- [2] 吴小琼, 肖德宝, 常亚楠. 基于 NETCONF 的 SNMP MIB 数据转换 [J]. 计算机工程, 2010, 36(2): 64-66.
- [9] 吴黎兵, 付帆, 何炎祥, 等. 基于 XML 网络管理的 MIB 转换方法研究 [J]. 武汉理工大学学报(信息与管理工程版), 2011, 33(6): 851-854, 863.
- [11] 肖德宝, 赵志霞, 熊磊. 基于 NETCONF 的通用转换网关模型的研究和实现 [J]. 华中师范大学学报(自然科学版), 2007, 41(4): 515-519.