

EasiDEF: 一种水平化轻量级物联网数据交换协议

侯陈达^{1,2)} 李 栋¹⁾ 邱杰凡^{1,2)} 崔 莉¹⁾

¹⁾(中国科学院计算技术研究所 北京 100190)

²⁾(中国科学院大学 北京 100049)

摘 要 物联网系统中数据交换协议私有化的问题日益突出,异构应用系统之间缺少水平、无缝地交换数据和共享信息的方法.而现有互联网中广泛采用的XML(eXtensible Markup Language)等数据交换协议存在数据冗余和解析复杂的问题,并不适用于资源受限的物联网前端设备.因此文中提出了一种新型的水平化轻量级物联网数据交换协议EasiDEF.该协议根据环境监测类物联网的应用数据特点,将数据类型分为单次请求数据和周期连续上报数据两种常见形式.针对单次出现的数据请求采用三级压缩机制,包括:(1)一种XML文件的序列化方法.该方法基于XML标准设计,保证了数据交换的通用性和水平化.同时序列化的操作使得在低功耗设备中应用XML成为可能;(2)一种将XML文件的字符串标签转化为占用空间小的数值标签的字典映射机制,其中字典的设计面向典型的物联网应用;(3)一种基于LZW(Lempel-Ziv-Welch)的改进数据压缩算法EasiLZW,该算法基于物联网应用字典压缩数据,保证协议的轻量性.而针对周期连续上报数据类型,对连续数据传输提出了一种增量式压缩算法.在保证协议低开销的前提下,EasiDEF可有效压缩XML文件,降低数据传输量.实验结果显示EasiDEF比传统方法提升了5~10倍的压缩效果,能够在物联网的资源受限环境中支持水平化的数据交换.

关键词 物联网;水平化;轻量级;数据交换协议;压缩算法;增量传输
中图法分类号 TP391 **DOI号** 10.3724/SP.J.1016.2015.00602

EasiDEF: A Horizontal Lightweight Data Exchange Protocol for Internet of Things

HOU Chen-Da^{1,2)} LI Dong¹⁾ QIU Jie-Fan^{1,2)} CUI Li¹⁾

¹⁾(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾(University of Chinese Academy of Sciences, Beijing 100049)

Abstract Privatization of the data exchange protocol is a critical problem in the Internet of Things, so heterogeneous systems cannot exchange data and share information seamlessly. Moreover the existing standard protocol like XML is rather heavy load and complicated for the resource-constrained devices in IoT. Therefore this paper proposes EasiDEF, a new horizontal lightweight data exchange protocol for IoT. According to the data feature of IoT application for environmental monitoring, data types are classified as single request data and periodic continuous reporting data. For the single request data, EasiDEF adopts three-phase compression mechanism, including: (1) a serialization method of XML documents. Based on XML the work ensures interoperability for heterogeneous systems. In addition the serialization operation makes it possible to apply XML in low-power devices; (2) a dictionary mapping mechanism which transfers long text string of XML documents to short binary value according to typical IoT applications; (3) a LZW-based improved data compression algorithm named EasiLZW, which uses the IoT dictionary in the

收稿日期:2013-07-12;最终修改稿收到日期:2014-11-03.本课题得到国家“九七三”重点基础研究发展规划项目基金(2011CB302803)、中国科学院战略性先导科技专项项目(XDA06010900)、国家自然科学基金(61003293)和北京市自然科学基金(4112054)资助.侯陈达,男,1987年生,博士研究生,中国计算机学会(CCF)学生会会员,主要研究方向为嵌入式系统和物联网. E-mail: houchenda@ict.ac.cn.李 栋,男,1979年生,博士,副研究员,中国计算机学会(CCF)会员,主要研究方向为无线传感器网络和物联网系统.邱杰凡,男,1984年生,博士研究生,中国计算机学会(CCF)学生会会员,主要研究方向为嵌入式系统和无线传感器网络.崔 莉,女,1962年生,博士,研究员,博士生导师,中国计算机学会(CCF)会员,主要研究领域为传感器技术、无线传感器网络和物联网.

second phase to keep lightweight. Given the periodic continuous reporting data in IoT system, the paper presents an incremental compression algorithm. Accounting for the requirement of low cost in resource-constrained devices, EasiDEF can significantly compress XML documents and reduce the amount of data transmission. Experimental results show that EasiDEF has a 5—10 times promotion in the compression effect compared with conventional methods and supports seamless data exchange in resource-constrained environments.

Keywords Internet of Things; horizontal; lightweight; data exchange format; compression algorithm; incremental transmission

1 引言

近年来物联网技术的飞速发展受到了人们的广泛关注. 国际电信联盟(ITU)^①指出物联网从功能上能够实现“世界上所有的物体都可以通过因特网主动进行信息交换, 实现任何时刻、任何地点、任何物体之间的互联、无所不在的网络和无所不在的计算”. 由此可见物联网概念的出现打破了原有物理世界与信息世界相分离的传统结构, 使得人们不但能够访问和管理互联网中的虚拟资源, 还能够与物理世界中的实体资源互动. 但是现有的物联网应用大多为专有应用, 运行系统相对封闭, 不同系统的数据定义格式不统一, 导致异构系统之间难以进行水平、自由的数据交换和信息共享, 已经成为物联网发展和产业化进程的瓶颈. 因此物联网的应用系统中需要一种新型水平通用的数据交换协议.

从另一方面考虑, 当前互联网已经成为人们获取和共享信息的主要方式, 人们的生活和工作已经离不开互联网. 因此越来越多的需求希望能将各种各样的智能物体也连入互联网中, 以现有的互联网体系去连接物联网应用, 从而大大降低上层应用的开发成本, 提高物联网系统的水平性和通用性. 但是从目前的研究进展看, 物联网应用之间的数据通信和数据交换普遍存在协议隔离和格式差异的问题, 缺少水平化的数据交换协议和应用层访问协议, 不能很好的集成到现有的互联网体系当中.

为了解决物联网系统中数据交换协议私有化的问题, 可以将 XML[®](eXtensible Markup Language) 等在互联网中广泛使用的数据交换协议引入到物联网中. 但是在这一过程中面临着以下的挑战. 首先由于物联网在前端感知网络中通常采用低成本、低功耗的嵌入式设备, 计算资源, 存储资源和网络带宽都严重受限, 对协议和算法的轻量性要求较高. 其次由

于互联网中采用的数据交换协议除了数据内容本身还包含自描述的控制信息, 存在大量信息冗余, 增加了网络传输开销. 同时 XML 技术本身很复杂, 处理 XML 的开销很大. 综合上述两点 XML 处理的复杂性和数据冗余对于低功耗、资源受限的物联网前端感知设备负担过重, 难以在前端设备上设计和实现.

同时需要考虑物联网具有与互联网不同的数据特点. 互联网的 web 请求通常为大量不规律的单次请求, 一次请求的内容一般比较复杂, 包括各种文本和多媒体信息, 数据量大. 但是在物联网应用中, 感知数据往往具有时空相关性, 即具有感知资源相对固定、单次请求的数据量小和连续传输数据相似性大的特点. 特别是在环境监测类物联网应用的感知数据中, 采用 XML 这种自描述的数据交换协议会导致控制信息远远多于感知内容数据, 存在连续传输过程中控制内容大量冗余的问题.

为了解决这些问题, 本文提出了一种基于 XML 的轻量级物联网数据交换协议 EasiDEF. 该协议根据环境监测类物联网应用的数据特点, 将数据类型分为单次请求数据和周期连续上报数据两种类型. 针对单次随机出现的请求数据采用三级压缩机制, 包括: (1) 一种 XML 文件的序列化方法. 该方法基于 XML 标准设计的数据格式保证了数据交换的通用性, 并且序列化的操作使得在低功耗设备中应用 XML 成为可能; (2) 针对现有的物联网典型应用方式, 采用能将 XML 文件的字符串标签转化为数值标签的映射字典机制; 通过字典映射将占用空间较大的字符串转化成二进制数值, 实现了数据压缩; (3) 一种基于 LZW(Lempel-Ziv-Welch) 的改进数据压缩算法 EasiLZW, 该算法基于物联网应用字典压缩数据, 压缩数据保证协议的轻量级特点. 而针对周

① ITU Internet Report 2005: The Internet of Things. <http://www.itu.int/pub/S-POL-IR.IT-2005/e>, 2014. 6. 13

② World Wide Web Consortium (W3C). XML Technology. <http://www.w3.org/standards/xml/> 2014. 6. 13

期连续上报数据的类型,提出了一种增量式压缩传输算法.通过比较连续传输过程中传输文件与标准文件的不同,传输增量 XML 文件,实现 XML 文件的压缩.在保证协议低开销的前提下,EasiDEF 能够大幅压缩 XML 文件,降低数据传输量,为物联网系统前端感知网络提供轻量级水平化的数据交换协议.同时针对互联网中的一般 XML 格式数据集、故宫博物院文物监测系统和无锡太湖水污染监测系统的数据集,设计了 4 种实验场景以验证该数据交换协议的压缩效果和算法开销.

本文的贡献主要集中在以下 3 方面:

(1) 针对单次请求数据类型采用三级压缩机制,有效提高了 XML 的压缩率和处理效率,包括:①一种 XML 文件的序列化方法;②一种字符串标签到二进制数值映射字典机制;③一种基于 LZW 的改进数据压缩算法 EasiLZW.通过这三级压缩机制压缩数据保证协议的轻量级特点.

(2) 针对周期性连续上报数据类型,提出了一种增量式压缩算法,减少了连续数据传输过程中重复数据的冗余传输.

(3) 设计了 4 种实验场景,通过文件压缩率和处理时间开销的对比实验,验证了 EasiDEF 的协议性能和算法开销.

本文第 2 节简要介绍研究背景和相关工作;第 3 节给出 EasiDEF 概述;第 4 节是协议的详细设计和实现;第 5 节是性能与效果实验评价;最后第 6 节给出本文的结论和展望.

2 研究背景与相关工作

目前在数据交换协议方面的工作主要包括互联网通用协议、互联网轻量级协议和物联网轻量级协议 3 类.首先在互联网中广泛采用的协议有 XML、JSON 等协议,尤其以 XML 占主导地位.XML 是为了人类可读性和互操作性而设计的一种通用数据交换协议,因此 XML 在描述数据的过程中采用了大量重复的数据标签,造成了大量数据冗余.一旦这种冗余与应用程序中的元素和属性名结合起来,XML 文件中标签成分的大小可能远远超出文件中实际数据成分的大小.这种数据冗余通常会给网络传输带来负担.

为了降低数据传输量,在互联网中通常采用的方式是先压缩再传输.在实际应用中,Gzip(GNU zip)是互联网中被广泛采用的数据压缩方式,HTTP 协

议(Hypertext Transfer Protocol)即支持 Gzip 的压缩传输.Gzip 是一种开源的文件压缩程序,其核心采用 LZ77 与哈夫曼编码的组合方法,因而 Gzip 是对一般字符串做处理的压缩算法,并没有考虑 XML 的结构化特点,缺少相对应的压缩优化方法.

在学术界很多学者同样提出了一些 XML 压缩方法,以优化 XML 在应用中的性能.文献[1-2]提出的压缩方法主要思想是将 XML 文档拆分成属性名、文本数据和文档结构 3 个不同的部分后分别进行压缩.在压缩每个部分时,还要将数据划分到不同的容器内使得每个容器内的数据具有一定的相关性,然后再根据每个容器内的数据特点进行压缩.Millau^[3-4]采用了差异 DTD(Document Type Definition)树来压缩符合 DTD 定义的 XML 数据,其主要思想是从 DTD 树的解析过程中获得语义信息,通过 XML 结构信息和数据编码形成结构流和内容流来达到压缩的目的.文献[5]用树型结构表示 XML 文档,通过提取公共子树来压缩 XML 数据中的结构冗余.文献[6]给出了第一个支持查询处理的 XML 压缩方法.它用数值来表示元素名和属性名,而数据项则用非自适应的哈夫曼编码来进行压缩.根据某些典型应用仅访问 XML 文档中的某些特定元素的特点,文献[7]中的方法保持应用需要的元素不变而将其他元素用 XSLT(Extensible Stylesheet Language Transformations)转换成 CSV(Comma Separated Values)的格式,进而达到压缩 XML 的目的.2013 年 Lohrey 等学者在文献[8]中提出基于有向无环图的 XML 压缩方法,用无等级的节点标签树来表示 XML 的编码二叉树,减少了标签冗余,实现了数据压缩.然而上述工作 XML 压缩方法的不足都是处理相对复杂,需要大量的计算和存储资源,难以应用在资源受限的物联网前端设备.

2007 年 W3C 标准组织发布了高效 XML 交换格式第一版公开草案 EXI^[9](Efficient XML Interchange).EXI 格式使用由信息和正规语言理论衍生出的一种混合方法,利用相对简单的算法以及少量的数据类型,可靠地产生 XML 事件流的高效编码.虽然 EXI 针对 XML 文件结构做了优化处理,可应用在资源受限设备,但是由于 EXI 的普适性,没有面向物联网的特点进行方法优化,在数据压缩率方面还有提高的空间.

在物联网领域内,目前已有的工作中微软研究院的 Priyantha^[10]等学者于 2008 年发表了首篇关于

物联网 web 服务及其系统架构的论文,文章提到了在能量和计算能力受限的情况下基于传感器节点的 XML 解析方法和压缩算法. UC Berkley 大学的 Dawson-Haggerty^[11]等研究人员在 2010 年提出了首个基于 REST(RePresentational State Transfer)风格的物理信息交换协议 sMap,该协议使用了编码后的 JSON(JavaScript Object Notation)数据描述格式进行数据封装,减少了数据传输量. Castellani 等人在文献[12]中提出了物联网中基于 CoAP 和 EXI 的 web 服务框架. CoAP(Constrained Application Protocol)协议是国际标准制定组织下 CoRE(Constrained RESTful Environment)工作组为受限节点制定相关的 REST 架构风格的应用层协议,以取代 HTTP 协议在受限设备中运行. 文章用 CoAP 协议传输 EXI 格式的载荷数据,在资源受限的节点上验证了 EXI 的可行性. 2013 年 Thoma 等人^[13]提出了基于 REST 的传感器网络,该框架采用 Atom^①和 JSON^②的编码方式减少数据传输,提高网络性能. 同年 Groppe^[14]等研究人员提出了一种嵌入式 XML 语言,该工作通过减少 XML 查询时的多余计算来降低处理 XML 的开销,缺点是该工作并没有减少 XML 的数据冗余.

上述内容介绍了现有水平化数据交换协议的研究现状,所述水平化网络协议栈如图 1 所示. 在互联网中自下而上采用了 TCP(Transmission Control Protocol)、HTTP 和 XML 的水平网络协议体系. 在物联网相关的已有工作中 sMap 采用了 UDP(User Datagram Protocol)、EBHTTP(Embedded Binary HTTP)和 Encoded JSON 结合的水平网络协议体系,也有包括 UDP、CoAP 和 EXI 结合的水平网络协议体系. 本文提出采用 EasiDEF 与 UDP 和 CoAP 构成的新的适用于物联网应用特点的轻量级水平网络协议体系.

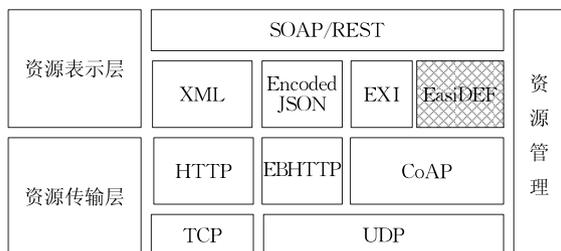


图 1 水平化网络协议栈示意图

综上所述的研究背景和相关工作表明,目前在物联网前端感知网络中需要一套高压缩率和低开销的水平化轻量级数据交换协议.

3 EasiDEF 协议概述

EasiDEF 是一种基于 XML 的物联网轻量级数据交换协议,解决了物联网不同应用系统数据交换协议私有化的问题,同时采用了一定的压缩策略,降低了数据传输量,优化了资源受限网络的性能.

为了能够设计出适合物联网前端感知网络的数据交换协议,首先要分析现有物联网应用系统中前端感知网络通常具的数据特点:(1)在物联网的应用系统中前端感知资源类型相对固定.一旦感知网络被部署,硬件传感器基本不会被修改和更换,所以感知网络中感知资源在一段连续的时间区间内是相对固定的.即使是异主系统,面向同类应用的传感器具有种类、数量、密度等相似性或互补性,如采用通用数据交换协议,则可达资源共享;(2)物联网系统中单次请求数据量一般比较小.因为物联网应用系统的前端感知网络一般采用低成本的嵌入式节点,计算、存储和通信能力非常有限,尤其在环境监测类的物联网应用中,感知数据通常为温度、湿度等环境信息,以数据内容为主,图片等多媒体内容占据较少的比例,所以单次的的数据量通常只在 10~500 byte 范围内;(3)在环境监测等普遍部署的物联网应用中,数据采集往往具有周期性和规律性.应用中用户主动发送请求获取数据的需求较少,数据大多采用用户订阅然后感知节点周期上报的形式,数据的格式和内容在一段时期内基本不变,只有采集的实时数据会变化.对于这类的物联网应用,如果每次都完整传输所有信息,会出现连续数据上报的重复性问题,可以采用增量传输数据的方式进行数据压缩.

以故宫博物院文物监测系统^[15]和无锡太湖水污染监测系统^[16]为例说明以上列出的 3 个物联网系统前端感知网络的数据特点.首先在故宫博物院文物监测系统和无锡太湖水污染监测系统中包含的感知资源类型和对应的数据量级如表 1 所示.从表 1 中可以看出物联网前端感知网络根据实际应用的的不同具有相对固定和专有的硬件感知资源.资源单次通信的数据量比较小,通常只有几十到几百字节.

- ① Nottingham M, Sayre R. The atom syndication format, 2005. <http://tools.ietf.org/html/rfc4287> 2014. 6. 13
- ② Crockford D. The application/json media type for javascript object notation (json). <http://tools.ietf.org/html/rfc4627> 2014. 6. 13

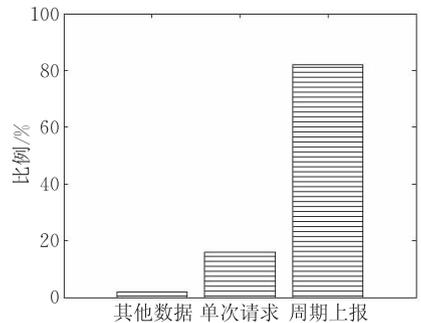
表 1 故宫博物院文物监察系统和无锡太湖水污染监测系统感知资源和对应数据量示例

应用	感知资源	数据量/byte
故宫	温度	20
	湿度	20
	光照	30
太湖	气象组合	200
	水温	20
	浊度	30
	叶绿素	30
	蓝绿藻	30

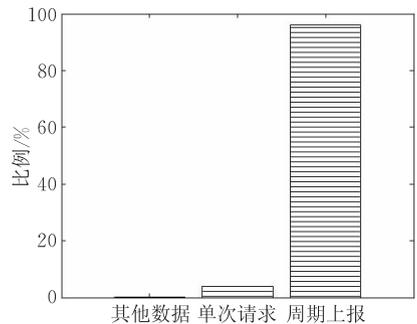
其次本文统计了在无锡太湖水污染监测系统和故宫博物院文物监测系统中 1000 次数据传输中包含的单次请求数据与周期连续上报数据的分布比例,如图 2 所示.

从图 2 中可以看出在物联网的环境监测类系统中数据通信方式以周期上报为主,单次随机请求的需要比较少,占请求总量的 20% 以下. 该统计的结果验证了上述总结的物联网数据特点的第 3 点,即周期性连续上报的数据请求在总体请求中占多数. 在连续数据上报的过程中,相邻两次的上报数据控制内容基本相同,只有实时感知数据发生变化,因此适合在这类应用中采用增量数据传输的方法来压缩数据,降低数据传输量.

根据上述对物联网系统前端感知网络数据特点的总结,本文将应用数据分为单次请求数据和周期连续上报数据两种. EasiDEF 处理流程图如图 3 所示. 对于单次请求数据采用三级压缩机制,包括: (1) XML 文件的序列化. 将格式化的 XML 文件通过序列化的操作转换为字符串,减少 XML 标签冗



(a) 无锡太湖水污染监测系统中请求类型分布



(b) 故宫博物院文物监测系统中请求类型分布

图 2 环境监测系统中请求类型分布图

余,便于后期处理;(2)一种标签数值映射字典机制.设计针对不同物联网应用的标签字典,对要传输的 XML 文件进行字典映射,将多字节字符串标签转换成单字节数值;(3)改进的 LZW 数据压缩算法 EasiLZW.对于周期连续上报数据提出增量式数据传输压缩算法,通过比较传输文件与标准文件的不同,传输增量 XML 文件,达到压缩传输数据,降低传输数据量的目的.总体来说,EasiDEF 通过三级压缩机制和增量传输压缩算法实现面向物联网的轻量级数据交换协议.

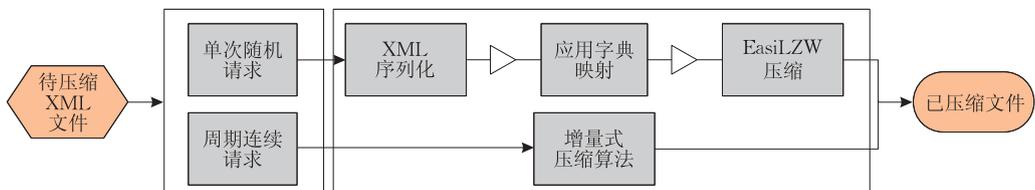


图 3 EasiDEF 的整体处理流程图

4 EasiDEF 设计与实现

本节详细介绍 EasiDEF 协议的设计和实现.

4.1 节介绍如何序列化 XML 文件,序列化的优点和解决的问题. 4.2 节对物联网应用字典的设计进行详细的描述. 4.3 节提出改进的 LZW 算法 EasiLZW. 4.4 节提出一种增量化传输的数据压缩算法.

4.1 XML 序列化

XML 序列化是将文件转换成字符串的过程,其目的是一是压缩文件,减少冗余数据;二是预处理,方便以字符流进行操作.序列化的主要思想是用元素标签(Tag)来表达 XML 数据,包括 XML 文件中的元素(Element)、属性(Attribute)、文本(Text).对于在物联网前端感知网络中不太常用的注释(Comment)等内容进行了精简.图 4 和图 5 给出了一个简单的 XML 文件及其对应序列化后的结果.

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <Region>
    <RegionName>West</RegionName>
    <Sink>
      <SinkName>Wumen</SinkName>
      <FromNet>NetWumen</FromNet>

      <ToDatabase>DbWumen</ToDatabase>
      <IP>10. 22. 0. 234</IP>
      <Port>8080</Port>
    </Sink>
  </Region>
  <success>>true</success>
  <var>
    <moc>IntEnviroment</moc>
    <moi>HouseEnvSensor</moi>
    <code>ID</code>
    <value>101</value>
    <timestamp>2012-04-05
22:46:27</timestamp>
  </var>
  <var>
    <moc>IntEnviroment</moc>
    <moi>HouseEnvSensor</moi>
    <code>Temperature</code>
    <value>30. 53</value>
    <timestamp>2012-04-05
22:46:27</timestamp>
  </var>
</root>

```

图 4 XML 示例文件

```

X4rootX6RegionX10RegionNameT4WestzxX4SinkX8Sink
NameT5WumenzxX7FromNetT8NetWumenzxX10ToData
baseT7DbWumenzxX2IPT1110. 22. 0. 234zxX4PortT48080zx
X7successT4truezxX3varX3mocT13IntEnviromentzxX3moi
T14HouseEnvSensorzxX4codeT2IDzxX5valueT3101zxX9time
stampT192012-04-0522:46:27zxX3varX3mocT13IntEnviro
mentzxX3moiT14HouseEnvSensorzxX4codeT11Temperaturezx
X5valueT530. 53zxX9timestampT192012-04-0522:46:27zxZ

```

图 5 EasiDEF 序列化 XML 后的编码文件

图 5 所示的 EasiDEF 内容是一组标签序列,如“X”标签表示 XML 文件中的元素,“X4root”表示字符长度为 4 的元素“root”,“X4root”是一个完整意义的标签单元.从上述例子可知所有的 XML 内容在 EasiDEF 序列化的过程中都以标签来表示,将元素、属性等内容由一个标签单元来表达.标签单元包括标签本身(Tag)、数据长度(Length)和数据值(Value),即 TLV.因为在 XML 中元素总是以“<elementName>”和“</elementName>”开始和结束,本身就存在信息的冗余,标签化的表示方式消除了这种冗余,在通常情况下有效的减少了数据的长度;重复出现的文本只显示一次,显著地减少了冗余信息,同时压缩了 XML 的数据长度.并且传统方式在解析 XML 时,必须逐个字符的扫描,寻找 XML 元素名称、属性名称、字符文本在何处结束,而 XML 内容的标签化,免除了这一繁琐的工作.

在 EasiDEF 中,文本数据的长度信息总是出现

在文本之前,这样我们可以预先知道将要读取处理的数据长度,需要分配的空间大小,可以整体拷贝,而不像 XML 那样只有读到结束符时才能确定文本数据的大小,所以该序列化模块具有一定的预判功能,可以提前处理数据所需的系统资源.

EasiDEF 完整的结构布局如图 6 所示,由头部(EasiDEF Header)和主体(EasiDEF Body)构成. EasiDEF 头部包括标识 EasiDEF 的区别标志位(Distinguishing Flag)、头部长度(Length)、版本号(Version),EasiDEF 主体则是由一串标签单元构成的,标签“Z”是整个 EasiDEF 结束的标志.

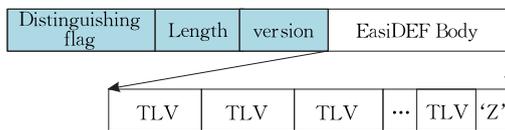


图 6 EasiDEF 的结构布局图

在 EasiDEF 头部,区分标志位占 2 个字节;头部长度是指后续的版本和编码控制信息所占的字节数,用一个字节表示,目前取值为 5;EasiDEF 版本取值 V1.0,版本号为数字 1,占一个字节.对于 EasiDEF 的主体是有多个表达 XML 组成部分的标签单元构成的,最后以“Z”结束.如图 5 所示 EasiDEF 中的数字使用 ASCII 码字符表示,但在实际的 EasiDEF 中是二进制的数值,并非 ASCII 码.

EasiDEF 的标签化方式使得处理器不需要扫描每一个字符,识别 XML 结构的结束符;使用数值代替文本不仅降低信息冗余、压缩数据长度,还使得数据的比较操作更快、更容易;长度前置使得块拷贝成为可能,也避免了处理器动态的分配变长内存.因此 EasiDEF 的解析轻量化,解析时需要的系统资源显著减少,解析的速度得到很大提高.

4.2 标签数值映射字典机制

在这一节本文提出了 EasiDEF 中的标签数值映射字典机制.在上述第 2 节 EasiDEF 概述中总结了物联网系统中前端感知网络的数据特点,即物联网不同应用具有不同的感知资源类型.同类物联网应用拥有大量相似的感知资源,比如在环境监测类的应用中,基本都围绕环境信息:温度、湿度和光照等参数.因此在 XML 文件描述和组织感知数据的时候,有大量相关元素标签是重复和冗余的.

为了解决重复标签问题,充分利用物联网前端感知网络的数据特点,本文提出要在感知节点的本地建立标签数值映射字典.根据对历史数据进行统计和分析,得到出现频率较高的关键词,建立应用字典,将多字节字符串转换成单字节的数值,在 XML

序列化的基础上进一步压缩了 XML 文件. 本节建立的物联网应用字典作为下一节 4.3 中 EasiLZW 的初始化输入字典. 图 7 是以环境监测为背景的物联网应用字典示例.

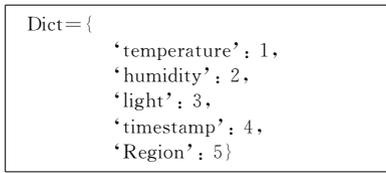


图 7 EasiDEF 物联网应用字典示例

4.3 改进的 LZW 压缩算法 EasiLZW

1984 年 Welch 在文献[17]中提出了 LZW 算法. LZW 算法的压缩原理是通过分析输入字符流, 在压缩的时候自适应地生成字典, 记录此前所有出现过的不重复的字符串. 通过比较当前输入字符流与字典中的字符串来确定输出值, 并更新字典, 其压缩流程中没有传输开销, 计算相对简单. 发送方和接受方都包含初始化字典, 基于已经存在的字典词条建立新的字典词条. 一旦接收到数据, 接收方就可以重新建立字典. 其最大的特点是逻辑简单, 易于实现, 速度快.

但是经典 LZW 算法存在以下 2 个不足: (1) 经典 LZW 算法针对的是无结构的字符流, 没有考虑结构化数据的压缩方法. 结构化数据一般是自描述的, 比如 XML, 数据的结构和内容混在一起. 结构化的数据有很多信息是已知的, 但是 LZW 无法利用; (2) 用 LZW 算法压缩时自适应形成字典的过程需要积累足够的长度, 根据本文第 2 节中所总结的物联网数据特点可知, 物联网系统中存在的信息大多数都是短而小的, 容易造成在压缩时字典无法积累相当的结构化信息, 也就不能有效地压缩文件中的冗余信息.

为了克服上述的不足, 本文提出了 EasiLZW 算法. 其主要思想是编码和解码不再是以单个字符而是以字典中的标准子串为单位进行. 而标准子串是针对物联网应用从 XML 文件中提取出的结构化标签的字符串, 这些字符串组成了 4.2 节中的物联网应用字典. 因此将 4.2 节中生成的典型物联网专有字典作为 EasiLZW 的初始化字典输入算法. EasiLZW 的编码和解码过程如算法 1 和算法 2 所示.

算法 1. EasiLZW 编码算法.

输入: 待压缩的字符串 s

输出: 字符串 s 压缩后的码串 $CODE$

1. $STRING = NULL$
2. WHILE there are still input substring DO

3. $SubSTRING = \text{get input substring}$
4. IF $STRING + SubSTRING$ is in the string table then
5. $STRING = STRING + SubSTRING$
6. ELSE
7. output the $CODE$ for $STRING$
8. add $STRING + SubSTRING$ to the string table
9. $STRING = SubSTRING$
10. END of IF
11. END of WHILE
12. Output the $CODE$ for s

算法 2. EasiLZW 译码算法.

输入: 待解压缩的码串 $CODE$

输出: 完成解压缩的字符串 s

1. $CODE = \text{get first input code}$
2. $SubSTRING = \text{get the translation from } CODE$
3. WHILE there are still input codes DO
4. $CODE_P = CODE$
5. $SubSTRING_P = \text{get the translation from } CODE_P$
6. $CODE = \text{get next input code}$
7. IF $SubSTRING$ is in the string table then
8. output $SubSTRING$
9. $M = \text{get the first unit string in } SubSTRING$
10. add $M + SubSTRING_P$ to the string table
11. ELSE
12. $M = \text{get the first unit string in } SubSTRING_P$
13. output $M + SubSTRING_P$
14. add $M + SubSTRING_P$ to the string table
15. END of IF
16. END of WHILE
17. Output uncompressed s

对于标准子串的匹配问题我们采用滑动窗口的机制来优化标准子串的匹配过程, 如图 8 所示. 该滑动窗口包含两个指针 $P1$ 和 $P2$, $P1$ 是字符流输入位置的当前指针, $P2$ 是搜索标准子串时的超前指针. $P1$ 和 $P2$ 要满足以下 3 个条件: (1) 初始时 $P2 = P1$; (2) 匹配的范围在 $P2$ 到 $P1$ 之间, 滑动窗口的大小 W 为标准子串的最大长度和剩余字符流长度的较小值; (3) 结束时 $P1 = P2 + 1$; 当找到匹配的标准子串时 ($P2 = P1$), 窗口向前滑动 $P2 - P1$ 个字节.



图 8 滑动窗口匹配标准子串示意图

对于 EasiLZW 的算法效率分析如下. 传统 LZW 算法大量依赖快速的字典查找, 若直接检索字典, 代

码执行速度慢,其时间复杂度为 $O(n^2)$. 因此需要对字典检索进行优化,引入 Hash 表能有效提高字符串表的检索效率及整体执行效率. Hash 表的容量与字典的容量均为表中不定长元素的代码数组,用于存放 Hash 值相同的字符串代码.一般 Hash 函数产生的重复值较少,这样检索字符串所需比较的次数将大幅减少,时间复杂度几乎接近于 $O(n)$,从而有效提高了代码的执行效率.改进的 EasiLZW 算法采用物联网应用字典,以标准子串为单位进行字典匹配,删除了匹配率小的词条,并根据字典中每个码字的使用计数值大小进行排序.一方面减少了算法的空间占用,释放了大量无效内存;另一方面,对有序 Hash 表可以采取二分查找,其时间复杂度降为 $O(\log n)$.由此可见,改进算法降低了经典算法的复杂度.

4.4 增量式数据压缩算法

上述 4.1~4.3 节是针对单次请求的数据类型所提出的压缩处理方法,本节针对周期连续上报数据,提出 EasiDEF 增量式的压缩算法.由于在物联网系统中存在较多的周期性数据传输,通常在一段时期内传输的数据类型和结构相对固定,只有实时感知的数据会发生变化.因此如果每次都完整的传输数据包,会造成大量的带宽浪费和能量消耗.为此本文基于 XML 提出一种增量式的数据压缩算法.

增量压缩算法的主要思想是首先要在物联网前端感知节点本地存储一个标准数据文件.该文件可以与程序一起写入感知节点,固化为节点文件.将每次传输的数据文件作为目标文件,然后计算和已经存储在感知终端的标准文件之间的增量变化文件.当两文件间的相似性超过了预先设定的阈值时,就进行增量压缩,计算出一个增量变化文件,最后只需把这个增量变化文件和参考文件存储在感知节点中即可.

增量式数据压缩算法的输入为一个待传输文件(记为 T)和一个参考文件(记为 R),输出为一个增量更新文件(记为 $updateFile$).通过比较待传输文件 T 和参考文件 R ,得到增量比较文件(记为 $VaryFile(T,R)$),该增量比较文件记录了需要周期性更新的数据在原文件中的位置和结构,因此算法可以对照文件 R 的内容对文件 T 进行压缩,将新产生的传感数据 $sensorData$ 按照增量比较文件中记录的数据位置更新,并输出更新后的文件 $updateFile$.其压缩过程如算法 3 所示.

算法 3. 增量式传输压缩算法.

输入:待压缩文件 $newFile$,参考文件 $referenceFile$

输出:增量变化文件 $updateFile$

1. $writeFile(referenceFile)$; //初始化写入参考文件
2. WHILE there are still new sensing data
//有新数据产生
3. IF $referenceFile$ is updated //参考文件已更新
4. $VaryFile(T,R) = Compare(newFile, referenceFile)$;
5. $updateFile = Generate(VaryFile(T,R), new(sensorData))$;
6. $Send(increaseFile)$;
7. ELSE
8. $updateFile = Generate(VaryFile(T,R), new(sensorData))$;
9. $Send(increaseFile)$;
10. END of IF
11. END of WHILE
12. EXIT //没有新数据产生则退出

5 实验结果与分析

本节通过实验分析 EasiDEF 的压缩率和处理速度,以验证该协议的性能表现.因为 EasiDEF 协议是基于 XML 标准设计和实现的,而 XML 的自描述特性可以实现不同系统之间的数据交换,保证了水平性,所以本节没有专门设计实验场景验证协议的水平性,而是把实验的重点放在 EasiDEF 的轻量级验证上.实验基于 ARM 的嵌入式 S3C 6410 的平台设计和实现,该平台 CPU 为 ARM1176JZF-S,主频 664 MHz, RAM 256 M.数据内容基于 CoAP^[18] 协议传输.实验对比的对象包括 XML、GZIP、EXI 和本文的 EasiDEF. XML 为原始文件,GZIP、EXI 和 EasiDEF 都是以 XML 为输入文件经过压缩后的输出文件.GZIP 采用了标准实现方案^①. EXI 采用了 Efficient XML 规范 V1.0 版本^② C 语言版本的实现.在实验的过程中一共设计了 4 类典型应用场景:

场景 1. 互联网中传统的 XML 文件实验.该实验采用了互联网已有的 XML 文件测试集^③,包括以下 4 类:

- (1) TPC-D. 基于交易基准测试集人工生成的

① Gailly J L. Adler M gzip 1.2.4. <http://www.gzip.org/> 2014. 6. 13

② EXI Efficient; An Open Source Implementation of the W3C Efficient XML Interchange (EXI) Format Specification. <http://exificient.sourceforge.net> 2014. 6. 13

③ Hartmut Liefke XMill Examples <http://www.liefke.com/hartmut/xml/data.html> 2014. 6. 13

关系数据库;

(2) Shakespeare. 结构化信息的莎士比亚戏剧注释剧本;

(3) SWISS-PROT. 基因序列和蛋白质数据的信息;

(4) Treebank. 华尔街日报文章的语言注解.

场景 2. 故宫博物院文物监测系统单次请求实验.

场景 3. 无锡太湖水污染监测系统单次请求实验.

场景 4. 故宫博物院文物监测系统数据周期连续上报实验.

实验中采用了压缩率和处理速度两个评价指标. 在本文中压缩率定义为: 压缩率 = (压缩后文档大小 / 压缩前原始文档大小) % . 处理速度定义为事务数 / 秒 Transactions Per Second (TPS); 处理速度 = 事务数量 / 处理事务的总时间.

压缩率衡量算法的压缩效果, 压缩率越小代表压缩效果越好; 处理速度衡量算法的时间开销, 处理速度越大代表算法性能越好. 下文针对 4 个实验场景作逐一的实验结果分析.

5.1 互联网普通应用场景

本文的实验场景 1 对互联网系统中传统的 XML 文件进行处理和压缩, 在 XML 原文件, GZIP、EXI 和 EasiDEF 中对比压缩率和处理速度. 场景 1 压缩率的实验的结果如图 9 所示. 从图中可知 GZIP、EXI 和 EasiDEF 对 XML 原文件都有不同程度的压缩. 在这一实验场景中, 本文的 EasiDEF 压缩效果最好, 压缩范围在 20.01% ~ 54.93%, EXI 压缩效果介于 EasiDEF 和 GZIP 之间, 压缩范围在 36.46% ~ 62.23%, GZIP 的压缩效果最差, 压缩范围在 35.41% ~ 69.74%. 综合看 EXI 与 EasiDEF 的压缩效果接近, 都比 GZIP 有 30% 的效果提升.

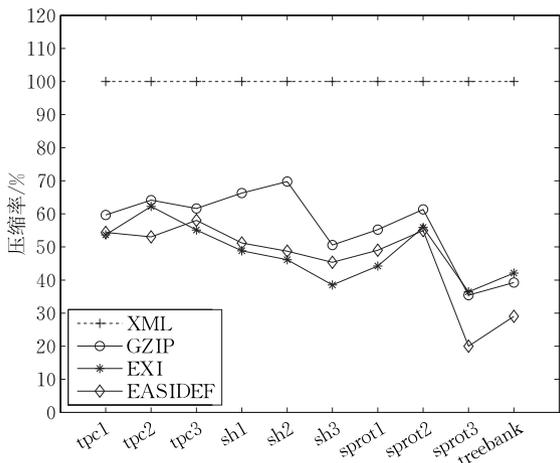


图 9 互联网普通应用单次请求场景: XML、GZIP、EXI、EASIDEF 压缩率对比实验

从场景 1 压缩率的实验结果可知由于 GZIP 是对字符串的压缩算法, 没有针对 XML 的结构化特点进行优化, 所以 GZIP 的压缩效果最差. EXI 则是针对一般 XML 的压缩算法, 对结构化的内容做了处理和压缩, 所以压缩效果比 GZIP 有所提高. EasiDEF 针对 XML 文件进行了 3 步压缩处理, 减少了数据冗余, 压缩效果也要优于 GZIP; 同时说明了 EasiDEF 的水平性, 可以适用于一般的 XML 文件. 但是由于 EasiDEF 面向物联网应用的数据特点进行了优化, 而场景 1 是互联网中的一般 XML 文件, 虽然 EasiDEF 的压缩效果比 EXI 略优, 在此场景下没有充分体现 EasiDEF 在物联网系统中的压缩优势.

图 9 是对场景 1 压缩率的讨论. 提升压缩效果可以降低数据传输量, 节约资源受限节点的通信能量. 同时必须要考虑压缩算法的效率, 本文采用处理速度 (TPS) 来衡量算法的时间开销. 图 10 对场景 1 中的处理速度进行了讨论. 在实验的时候本文以 GZIP 为标准, 对 EXI 和 EasiDEF 做归一化, 例如当 EasiDEF 的处理速度为 200% 时, 表示处理速度是 GZIP 的 2 倍, 即压缩相同文件时 EasiDEF 的时间开销是 GZIP 的一半. 从图 10 可知在多数情况下, EXI 和 EasiDEF 的处理速度要优于 GZIP, 平均提高 250%. 而 EXI 的处理速度又比本文提出的 EasiDEF 平均约有 10% 的提升.

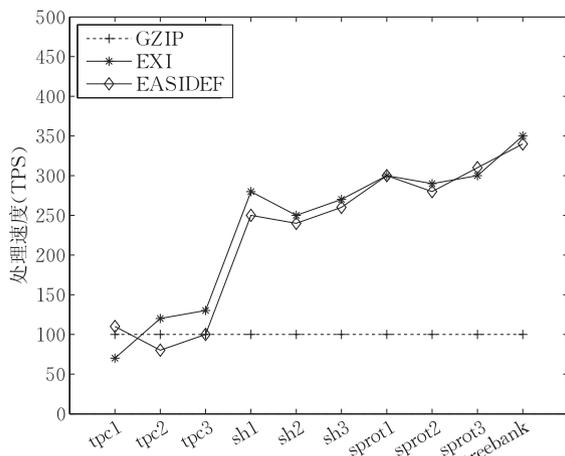


图 10 互联网普通应用单次请求场景: XML、GZIP、EXI、EASIDEF 处理速度 (TPS) 对比实验

场景 1 处理速度的实验结果反映了 GZIP、EXI 和 EasiDEF 压缩算法的不同. GZIP 是基于 LZ77^[19] 与哈弗曼编码的组合物, 在压缩数据时通常要付出一定的代价创建和维护字典, 同时需要遍历二叉树, 因此处理速度最低. EXI 与 EasiDEF 是轻量级协议, 在压缩处理的过程中采用了相对简单的算法, 因

此处理速度要优于 GZIP. 因为本文的 EasiDEF 使用了改进的 LZW 算法,需要对物联网应用字典进行遍历,代价要大于 EXI 的压缩算法,所以处理速度低于 EXI.

5.2 单次请求的实验场景

4.1 节中分析了场景 1 的实验结果,验证了 EasiDEF 在互联网场景下的压缩率高和处理速度大的特点. 因为 EasiDEF 是面向物联网的轻量级数据交换协议,所以本文将场景 2~4 设计在物联网的系统应用中. 具体的场景是基于故宫博物院文物监测系统和无锡太湖水污染监测系统两个实际系统而设计的. 原始 XML 文件封装的感知数据也都来源于这两个实际系统运行中的实时数据.

为了模拟物联网应用系统在实际运行中的情况,本文设计了 5 种网络传输操作:发起连接(connect),激活连接(activate),传输数据(data),关闭连接(deactivate)和断开连接(disconnect). 这 5 种操作组成了物联网应用系统数据请求和响应的一般过程. 实验设计和实现了这 5 中操作的 XML 文件,通过 GZIP、EXI 和 EasiDEF 进行对比实验.

对于单次请求的实验结果如图 11~图 14 所示,在实验场景 2 和 3 下分别测试了 GZIP、EXI 和 EasiDEF 基于故宫博物院文物监测系统和无锡太湖水污染监测系统的感知数据压缩率和处理速度. 从图 11 和图 13 可知 EasiDEF 的压缩效果要明显优于 EXI 和 GZIP. 故宫文物监测系统中 EasiDEF 的压缩率在 8.17%~41.14%, EXI 在 18.62%~47.51%,GZIP 在 16.01%~107.1%. 这里特别注意到 GZIP 会出现放大 XML 原文件的情况. 在 5 个请求操作中,connect, activate, deactivate 和 disconnect 属于控制操作,数据的冗余度比较小. 因

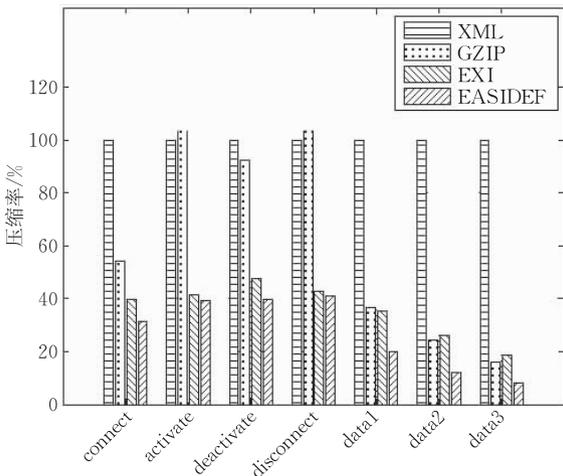


图 11 故宫博物院文物监测系统中单次请求场景:XML、GZIP、EXI、EASIDEF 压缩率对比实验

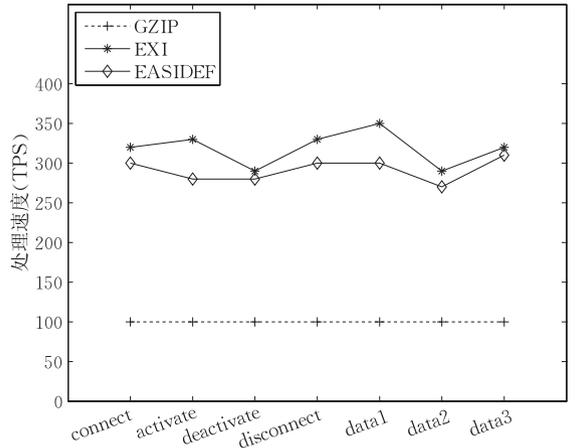


图 12 故宫博物院文物监测系统中单次请求场景:XML、GZIP、EXI、EASIDEF 处理速度(TPS)对比实验

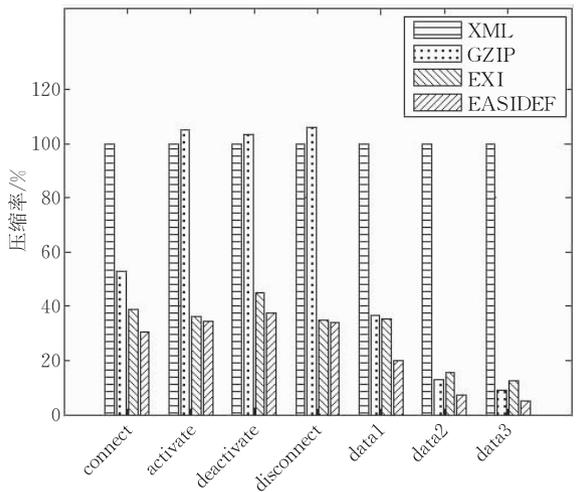


图 13 无锡太湖水污染系统中单次请求场景:XML、GZIP、EXI、EASIDEF 压缩率对比实验

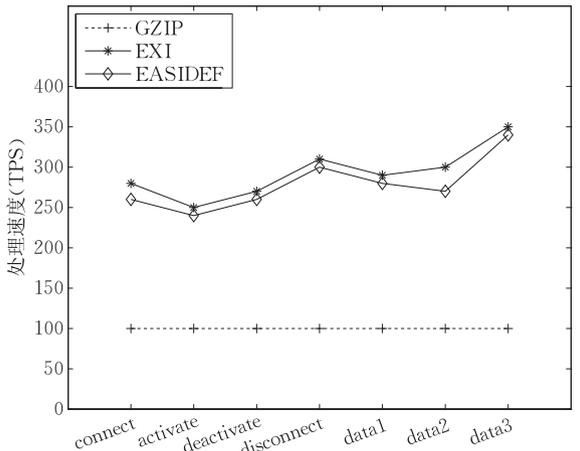


图 14 无锡太湖水污染系统中单次请求场景:XML、GZIP、EXI、EASIDEF 处理速度(TPS)对比实验

此从图 11 和图 13 中可知,前 4 种操作对应的 XML 文件,在压缩率方面 EasiDEF 虽然优于 EXI,但是相对比较接近,没有明显的优势. 对于 data 操作,因

为包含大量的感知数据,描述同类感知数据的标签相同,会产生较多的重复标签元素,造成冗余.因此此时 EasiDEF 的压缩效果要远好于 EXI 和 GZIP,平均约有 2~3 倍压缩效果的提升.

单次请求数据处理速度的结果如图 12 和图 14 所示.从图中可知 EXI 和 EasiDEF 比, GZIP 的处理速度有 2~3 倍的提升, EXI 与 EasiDEF 相比, EasiDEF 的处理速度要低于 EXI 10%~20%,原因已经在 4.1 节中进行了分析.

5.3 周期上报数据实验场景

对于周期上报数据的实验结果如图 15 所示.实验的场景基于无锡太湖水污染监测系统.从图中可知 EasiDEF 的压缩效果要远远好于 EXI 和 GZIP, EasiDEF 的压缩率在 2%左右, EXI 在 10%~20%, GZIP 在 20%左右.这是因为在周期上报的数据传输中, EasiDEF 采用了增量传输的压缩算法,实际传输的数据内容是文件之间的差异,即实时变化的感知数据.而 EXI 与 GZIP 依然是全文件传输,文件内容大量冗余.因此 EasiDEF 比 EXI 和 GZIP 有 5~10 倍的压缩效果提升.

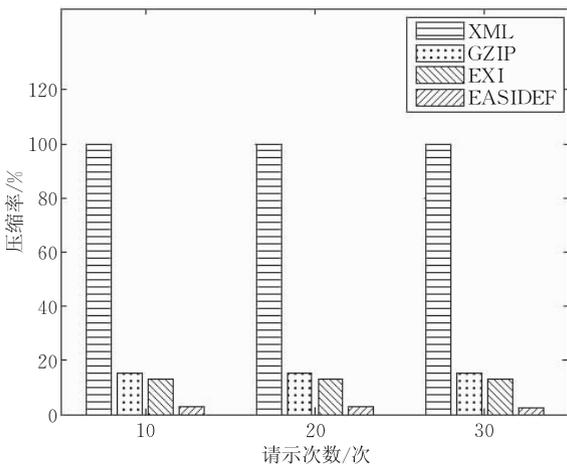


图 15 故宫博物院文物监测系统中周期上报数据的 XML、GZIP、EXI、EASIDEF 压缩率对比实验

6 总结与展望

本文针对现有物联网系统应用中数据交换协议垂直私有化的问题,提出了一种新型的水平化轻量级物联网数据交换协议 EasiDEF.该协议对于单次请求数据和周期连续上报数据两种类型采取了不同的处理方式.针对单次出现的数据请求采用三级压缩机制,有效提高了 XML 的压缩率和处理效率.而针对周期性连续数据上报应用模式,对连续数据传输使用增量式文件压缩算法.在保证协议低开销的

前提下, EasiDEF 能够大幅压缩 XML 文件,降低了数据传输量,为物联网系统前端感知网络提供了水平化轻量级的数据交换协议.

因为本文基于 CoAP 协议传输 EasiDEF 封装好的数据,而 CoAP 协议本身存在因设备休眠导致的节点响应率低等问题.所以在未来的工作中,我们将设计一种新型的应用层传输协议,然后将 EasiDEF 与该协议综合考虑如何提高受限网络的性能表现.

致 谢 审稿专家和编辑为本文提出了宝贵的修改意见和建议,作者在此表示衷心的感谢!

参 考 文 献

- [1] Liefke H, Suci D. XMill: An efficient compressor for XML data//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, USA, 2000: 153-164
- [2] Liefke H, Suci D. An extensible compressor for XML data. *Sigmod Record*, 2000, 29(1): 57-62
- [3] Girardot M, Sundareshan N, Millau: An encoding format for efficient representation and exchange of XML over the Web. *Computer Networks*, 2000, 33(1): 747-765
- [4] Sundareshan N, Moussa R. Algorithms and programming models for efficient representation of XML for Internet applications. *Computer Networks*, 2002, 39(5): 681-697
- [5] Buneman P, Grohe M, Koch C. Path queries on compressed XML//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, Germany, 2003: 141-152
- [6] Tolani P M, Haritsa J R. XGRIND: A query-friendly XML compressor//Proceedings of the 18th International Conference on Data Engineering. San Jose, USA, 2002: 225-234
- [7] Yoshida S, Yahagi H, Odagiri J. CSV compaction to improve data-processing performance for large XML documents//Proceedings of the IEEE International Conference on Data Compression. Snowbird, USA, 2004: 574
- [8] Lohrey M, Maneth S, Noeth E. XML Compression via DAGs//Proceedings of the 16th International Conference on Database Theory. New York, USA, 2013: 69-80
- [9] Peintner D, Kosch H, Heuer J. Efficient XML Interchange for rich internet applications//Proceedings of the IEEE International Conference on Multimedia and Expo. New York, USA, 2009: 149-152
- [10] Priyantha N B, Kansal A, Goraczko M, Zhao F. Tiny web services: Design and implementation of interoperable and evolvable sensor networks//Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. New York, USA, 2008: 253-266
- [11] Dawson-Haggerty S, Jiang X, Tolle G, et al. sMAP: A simple measurement and actuation profile for physical information//Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. Zurich, Switzerland, 2010: 197-210

- [12] Castellani A P, Gheda M, Bui N, et al. Web services for the Internet of Things through CoAP and EXI//Proceedings of the IEEE International Conference on Communications Workshops. Kyoto, Japan, 2011; 1-6
- [13] Thoma M, Kakantousis T, Braun T. REST-based sensor networks with OData//Proceedings of the IEEE International Conference on Wireless On-demand Network Systems and Services. Obergurgl, Austria, 2014; 33-40
- [14] Groppe S, Schütt B, Werner S. Eliminating the XML overhead in embedded XML languages//Proceedings of the 28th Annual ACM Symposium on Applied Computing. Coimbra, Portugal, 2013; 542-547
- [15] Li D, Liu W, Cui L. EasiDesign: An improved ant colony algorithm for sensor deployment in real sensor network system//Proceedings of the IEEE International Conference on Global Telecommunications Conference. Miami, USA, 2010; 1-5
- [16] Li Dong, Zhao Ze, Cui Li, et al. A cyber physical networking system for monitoring and cleaning up blue-green algae blooms with agile sensor and actuator control mechanism on Lake Tai//Proceedings of IEEE the 1st International Workshop on Cyber-Physical Networking Systems. Shanghai, China, 2011; 732-737
- [17] Welch T A. A technique for high-performance data compression. *Computer*, 1984, 17(6): 8-19
- [18] Kovatsch M, Duquenois S, Dunkels A. A low-power CoAP for Contiki//Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems. Valencia, Spain, 2011; 855-860
- [19] Ziv J, Lempel A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 1977, 23(3): 337-343



HOU Chen-Da, born in 1987, Ph. D. candidate. His research interests include embedded system and Web of Things.

LI Dong, born in 1979, Ph. D., associate professor. His research interests include wireless sensor networks and Internet of Things.

QIU Jie-Fan, born in 1984, Ph. D. candidate. His research interests include embedded operating system and wireless sensor network.

CUI Li, born in 1962, Ph. D., professor, Ph. D. supervisor. Her research interests include sensor technology, wireless sensor networks and Internet of Things.

Background

With the rapid development of Internet of Things (IoT) recent years, there is a critical problem to be solved that the lack of unified data exchange format among heterogeneous systems. Because the existing IoT applications are mostly specialized and isolated, their systems are not open to others. Moreover sensing data of these applications are private, so one system cannot communicate with other heterogeneous systems to share information. Thus the current IoT applications require a horizontal lightweight data exchange protocol.

In order to solve the problem of the privatization of data exchange format in IoT, the eXtensible Markup Language (XML) can be introduced to IoT, which is widely used in the Internet and has a good interactive property as the data exchange format. However, there are many challenges while we are doing this. Firstly, size and cost constraints on sensor nodes of IoT result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The technology of the Internet data exchange format XML itself is very complicated, including heavy XML procession and XML redundancy which increases network traffic overhead. When the high weight XML meets low cost sensor nodes, something goes wrong. Secondly, the IoT features different data characteristics from Internet. Internet web requests usually include a large number of irregular single requests and one request is generally complex with the large amount of data. However, in the IoT

applications sensing resources are relatively fixed, while request is periodic continuous and has a small size of data.

Related work about data exchange format for IoT is the Efficient XML Interchange (EXI) format. EXI is a compact representation for XML Information Set that is intended to simultaneously optimize performance and the utilization of computational resources. The EXI format is not optimized for IoT application so the compression ratio can be improved in a new way.

To solve these problems, this paper presents EasiDEF, a new horizontal lightweight data exchange protocol for IoT. Accounting for the requirement of low cost in resource-constrained devices, EasiDEF can significantly compress XML documents. Experimental results show that EasiDEF has a 5~10 times promotion in the compression effect than conventional methods. EasiDEF can reduce the amount of data transmission and reach interoperability among the heterogeneous IoT systems.

This work is supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302803, the Strategic Leading S&T Project of Chinese Academy of Sciences under Grant No. XDA06010900, the National Natural Science Foundation of China under Grant No. 61003293 and the Beijing Natural Science Foundation under Grant No. 4112054.