

一种可定制的异构网络协议解析器的设计

蒋镭沅, 李晓明

(浙江理工大学机械与自动控制学院, 浙江 杭州 310000)

✉1393621017@qq.com; lxmzist@zstu.edu.cn



摘要: 异构网络(Heterogeneous Network)中包含多种使用不同网络协议的设备, 因此设计一种可定制化的网络协议解析器是解决各设备之间信息交互, 降低硬件设备更新和替换成本的重要课题。本文提出使用XML(可扩展性标记语言)作为网络协议解析器的协议描述文件, 利用XML文件的可扩展性、异构性和标签之间的父子层级关系, 实现该网络协议解析器的通用性, 达到更新或更换网络协议时无须重新编写代码, 只需更改或替换XML文件即可; 同时, 支持用户可选当前解析器解析的最小颗粒度为bit或byte, 满足更多设备协议的数据要求。通过实验表明, 本文提出的网络协议解析器满足协议解析的要求。

关键词: 网络协议解析器; XML文件; 异构网络

中图分类号: TP311.5 **文献标识码:** A

A Design of a Customizable Heterogeneous Network Protocol Parser

JIANG Keyuan, LI Xiaoming

(Faculty of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou 310000, China)

✉1393621017@qq.com; lxmzist@zstu.edu.cn

Abstract: Heterogeneous Network contains a variety of devices that use different network protocols. Therefore, designing a customizable network protocol parser is an important issue to realize information interaction between devices and reduce the cost of hardware equipment update and replacement. This paper proposes to use XML (Extensible Markup Language) as the protocol description file of the network protocol parser. Based on its scalability, heterogeneity, and parent-child hierarchy between nodes, the XML file can be used to achieve the generality of the network protocol parser, so that there is no need to rewrite the code when updating or replacing the network protocol, just change or replace the XML file. At the same time, the minimum granularity of the current parser users can choose is bit or byte, which can meet the data requirements of more device protocols. Experiments show that the network protocol parser proposed in this paper meets the requirements of protocol parsing.

Keywords: network protocol parser; XML file; heterogeneous network

1 引言(Introduction)

异构融合网络的研究最早由美国加州大学伯克利分校的BARWAN(Bay Area Research Wireless Access Network)项目提出, 并首次将不同类型的网络组合构成异构网络以支持不同运行网络协议的网络设备^[1]。目前互联网和物联网产业

的繁荣发展也催生了大量的新型协议, 如MPLS、1553总线协议和IP-in-IP等。异构网络为了支持这些新的网络协议, 必须持续不断地更新中转设备的协议解析能力, 以满足不同设备之间的通信需求^[2]。

由于设备来自不同的设备制造商, 且每个设备制造商有

着自己独有的网络协议,网络协议随着产品的更新换代也在不断地更新。而硬件开发周期长和部署成本较高的问题,也意味着异构网络的部署和设计具有实验困难、更新成本高和硬件升级周期长的特点。本文以屏蔽硬件底层协议构建信息交互层的设计思路,利用软件的定义协议的思想^[3]提出了一种基于XML的异构网络协议解析器;设计了解析方式及其对应的XML文件结构,达到只需用户配置XML文件中的标签描述信息就可以实现使用不同网络协议的设备之间的信息交互。本文使用Java语言对解析器进行代码实现,一方面是Java具有平台无关性;另一方面是Java拥有性能稳定且高效的XML解析工具库^[4]。最后通过实验验证了该解析器可以正常工作并具备对不同网络协议设备的数据解析能力。

2 异构网络协议解析器方案设计(Solution design of heterogeneous network protocol parser)

网络协议是为完成计算机网络通信而制定的规则、约定和标准。一个完善的网络协议由语法、语义和时序这三大要素组成。而网络协议解析技术就是按照事先配置的网络协议将数据信息从数据帧中解析出来。

现有的网络协议解析方式可以分为两类,一类是依靠硬件底层对其进行解析;另一类则是依靠软件来实现数据的解析^[5]。前者依靠硬件底层直接对数据进行解析,优点是解析速度快,但是设备更新周期长,对异构融合网络的支持较差;后者则是在应用层对数据进行解析,底层设备只涉及数据的转发,优点是对异构融合网络的支持程度高,不同协议之间只是配置文件的替换,降低了搭建异构融合网络的门槛。

针对异构融合网络中底层设备采用不同协议,不同厂商之间的设备难以互联的问题,本文提出的解析器方案为基于软件的解析方案,用户只需按照解析器提供的标准XML标签来描述当前设备需要接收的网络协议,就可以打通多个设备在异构融合网络中的数据通信壁垒。

解析文件是使用XML语言来描述当前信源设备网络协议的描述文档。该解析文件被应用于本文提出的网络协议解析器中,为宿信和信源设备提供数据解析服务。

网络协议解析器软件由协议管理器(FrameManager)、可解析节点(ParseableNode)、解析链(ParseableList)和解析结果链(ParsedVarList)四大部分组成。协议管理器负责管理当前解析器可解析的所有协议,可解析节点是指解析文件中参与解析的表述节点,解析链则由多个可解析节点组成,而解析结果链则存放了所有解析完成的初始状态的数据。解析链数据结构如图1所示。

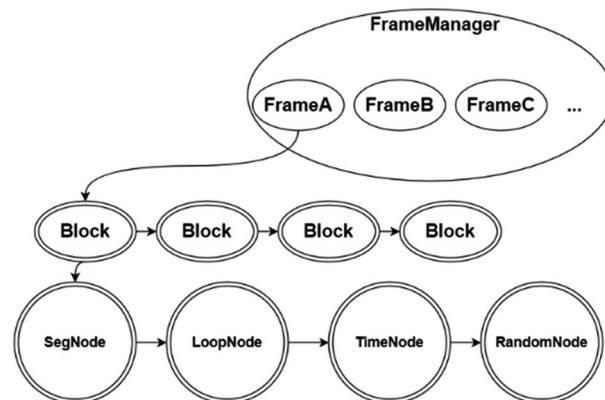


图1 解析链数据结构

Fig.1 Parsing chain data structure

3 XML协议描述文档的设计与实现(Design and implementation of XML protocol description document)

本文提出的网络协议解析器需要在异构融合网络中扮演“解释者”的角色^[6]。在异构融合网络中,设备与设备之间往往不是一对一的关系,而是一对多、多对一的关系。而作为一个异构融合网络往往意味着随着时间的流逝,设备的数量和种类即协议的种类是不断递增和更新的,而不同的协议对数据解析的颗粒度往往不同^[7]。除此之外,由于网络中数据量大且完成一次任务时不同协议切换较频繁,因此对解析器和描述文件便有以下要求:协议可替换和可增减,数据解析颗粒度可伸缩,协议切换要迅速。

为了满足以上设计要求,本文利用XML自身具有的树形结构和节点之间的父子层级关系,可以很好地支持协议的增减替换,并对设备本身的协议按照字段分块,规定最小解析单位,由此达到解析颗粒度的自由选择。

为了满足协议的动态切换和增减,本文利用面向对象的思想设计了一个协议管理器,该管理器在内存中管理多个协议。因为目前计算机中的内存大多使用DDR4内存,而DDR4的传输速率可达2,133—3,200 MT/s^[8]。因此在解析器初始化时,将描述网络协议的XML文件以特定的数据结构存储在内存中可以有效解决计算机IO资源的阻塞和浪费,并采用懒加载的方式对协议文件进行动态扩容。

3.1 XML协议描述文档设计

本文提出的协议描述文件的XML标签主要有根标签frames、协议标签frame、块标签block、最小解析标签seg、可变长字段标签random、循环标签loop和时间标签time。各个标签的主要功能如表1所示。

表1 XML文件标签及功能
Tab.1 XML file nodes and functions

标签	功能
frames	frame管理
frame	代表一种网络协议的类型
block	协议中的某一段，单位为byte
seg	规定数据的解析方式，单位为byte或bit
random	可变长字段专用标签
loop	循环字段专用标签
time	时间字段专用标签

XML文件的树形关系和XML协议描述文档实例如图2、图3所示。

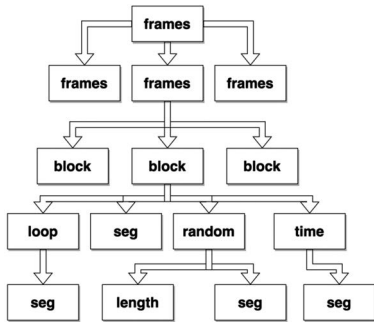


图2 XML标签层级关系

Fig.2 XML element hierarchy relationships

```
<frame name="frameName" id="xx">
<starter var="0x0a,0x12"/>
<block pass="00" name="state" length="1">
<seg pass="00" name="errorCode" type="byte" typesize="1" unit="byte" endian="00"/>
</block>
<block pass="00" name="dataBuffera" length="32">
<loop time="16">
<seg pass="00" name="var" type="short" typesize="2" unit="byte" endian="01"/>
</loop>
</block>
<block pass="00" name="dataBuffera" length="32" shouldCheck="true">
<seg pass="00" name="同步字" type="string" typesize="2" unit="byte" endian="01"/>
<seg pass="00" name="apid" type="string" typesize="11" preBits="0" unit="bit" endian="01"/>
<random pass="00" name="sjq">
<length pass="00" name="len" type="short" typesize="5" preBits="11" end="true" unit="bit" endian="01"/>
<seg pass="00" name="var" type="short" typesize="2" unit="byte" endian="01"/>
</random>
</block>
</frame>
```

图3 XML协议描述文档

Fig.3 XML protocol description file

3.2 解析协议数据结构设计

解析协议的数据结构用于映射当前XML文件中的标签和标签中的属性值，并且按照XML文件中的标签顺序对解析节点进行排列组合。程序中有关解析节点之间的类图关系如图4所示。

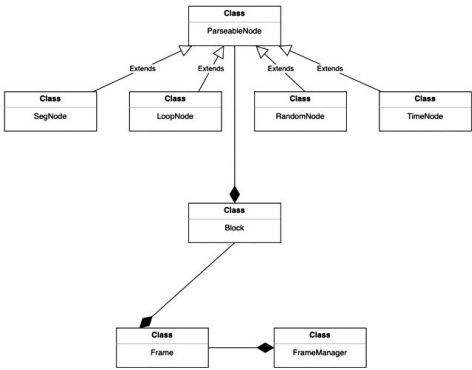


图4 解析节点间的关系

Fig.4 Resolving inter-node relationships

ParseableNode为SegNode、TimeNode、RandomNode和LoopNode的父类，表示可解析单位。SegNode是最小解析对象，也是整个解析链的节点。RandomNode、LoopNode、TimeNode和SegNode之间为聚合关系。所有的解析方式都由SegNode的实例对象完成。一个Frame对象表示一条网络协议，并负责管理多个Block对象；Block对象则管理着ParseableNode的所有子类对象。因此在解析器加载完XML文件后，可以由此在内存中生成一个多分支解析链表，FrameManager对象管理所有加载的Frame对象。

3.3 解析结果JavaBean设计

数据流被解析器成功解析后，所有解析出来的数据都需要打上对应的标识，为后续数据的使用提供键值映射。得到的数据需要经过二次加工或二次内部协议的加工之后才可使用。而当两个网络设备使用的网络协议不相同，则需要描述信息来描述该数据的数据类型、来源、所属帧等信息，因此需要设计一个JavaBean作为该数据信息的载体。该JavaBean的属性内容如表2所示。

表2 ParsedVar字段描述表

Tab.2 ParsedVar field description table

属性名	描述信息
frameName	数据帧名称
frameAddr	数据帧地址
blockName	字段名称
valueName	数据名称
parsed	是否解析标志
valueType	数据类型
deviceName	设备名称
deviceId	设备ID
buffer	数据原字节

4 解析器的设计与实现(Design and implementation of the parser)

本文提出的网络协议解析器作为网络协议的载体，需要从XML协议解析文档中得到单向多分支链表并对其进行管理。数据流到达解析器时，从多个协议中挑选对应的协议对数据流进行解析，因此需要设计一个面向软件的协议解析器来完成协议的加载和数据的解析。本节将利用前面设计的数据结构来设计解析器并描述其解析过程。

4.1 解析器设计

解析器的解析功能会按照当前加载的XML协议描述文档将数据信息从数据流中解析出来。由于数据信息在网络中

的传递可能会出现丢包等问题,因此解析器在解析数据流时需要对收到的数据流进行校验,包括帧头检测、数据长度检测、帧尾检测、CRC校验等^[9],只有检测通过才会进入下一步的数据解析工作。检测不通过时常用的解决方式有两种:丢弃或缓存。前者直接丢弃不完整的数据帧,可能会导致数据的丢失;后者会存储不完整的数据帧并等待帧完整,但是会导致解析器因数据帧长时间没有补全而发生阻塞。由于TCP/IP具有失败重传机制,且为了保证解析器的稳定运行,因此选用丢弃不完整数据帧的失败机制。

4.2 解析器解析流程设计

解析器接收到来自某一设备发来的数据流,首先从数据流中检测是否有帧头信息,一旦检测到帧头信息就开始校验数据长度和帧尾信息是否都符合,如果符合就进入数据的解析流程;否则就丢弃这一帧头继续等待下一帧数据流。解析器在解析过程中对应的状态传递如图5所示。

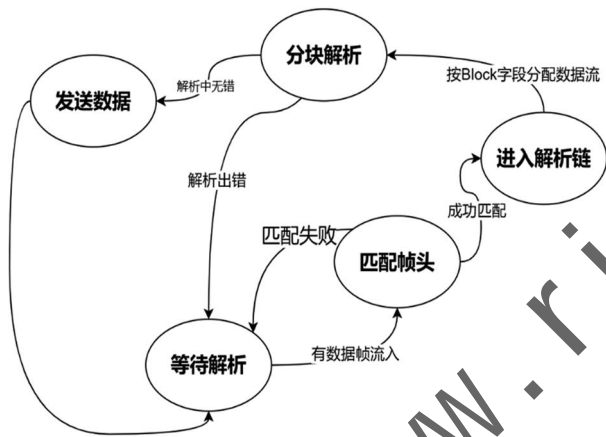


图5 解析器工作状态传递图

Fig.5 Transfer diagram of parser working state

解析流程将按照本文第3节中所设计的数据结构,以协议驱动解析器解析数据流中的数据。解析器会先从FrameManager中匹配协议帧头,匹配成功则开始解析;若没有匹配成功则会进入懒加载机制,从文件中搜索对应的数据帧头;若都没有则放弃解析数据。解析时以block为单位分块解析,若block标签的pass属性的属性值不为1时,则进入解析;否则直接缓存对应字段的原始数据。

解析过程中的数据会使用Parsedvar类的对象存储到链表中,每一个Parsedvar中都包含基本数据信息用于设备的软件层对解析出来的数据进行处理。这时解析器不再关注当前数据信息的去向,而是交给运行在设备中的软件决定。如果需要直接处理就直接处理,如果需要传输到下一个节点的设备就通过当前设备发送。

5 实例测试(Example test)

本文实验使用课题组开发的通用式仪器平台和多盒体组合式现场仪器进行实验测试,使用组合式现场仪器搭建的自动测试设备。自动测试设备用于终端设备模拟和总系统模拟^[10]。为了测试本文提出的解析器的异构性和通用性,通过在模拟总系统上使用该解析器,解析来自拥有不同网络协议的模拟终端设备的网络数据。因此可以将解析器是否成功解析出不同模拟终端设备发送的数据,并成功响应该请求作为该解析器是否达到解析异构融合网络要求的检验标准。

实验设备: X86架构的计算机、通用式仪器开发平台、多盒体组合式现场仪器。

实验设计: 计算机和多盒体组合式现场仪器使用网口连接,通过客户端控制多盒体组合式现场仪器作为模拟总系统向模拟终端设备发送命令,接收模拟终端设备上发出的数据。如果模拟终端设备成功接收到来自模拟总系统的广播数据和指令数据,模拟总系统可以接收到模拟终端设备上传的数据且后面板解析成功并将解析的数据通过客户端展示,则表明本文所提出的解析器符合要求。

本文提出的协议解析器已经作为jar包嵌入通用式仪器开发平台搭建的测试仿真程序的组件库中,并按照当前开发平台的组件开发要求,将解析器以组件的形式对其进行仿真实验。

5.1 实验环境搭建

本实验使用Java的SWT技术搭建了一套地测测试设备客户端,客户端包含需要发送命令的命令树,后面板开启和停止按钮用于控制后面板程序,协议解析表单则记录当前实验中所有通过本文提出的协议解析器应对不同网络协议解析出来的数据,如图6所示。

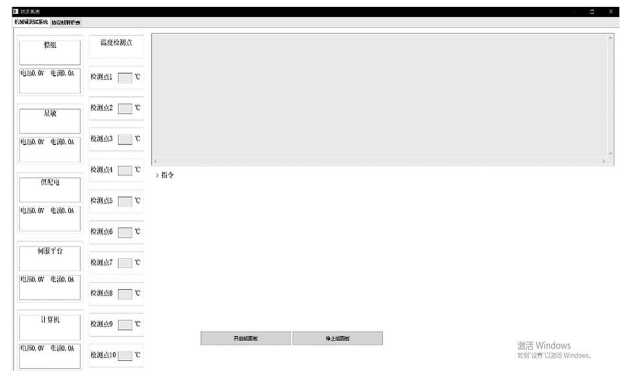


图6 客户端界面

Fig.6 Client interface

利用通用式仪器开发平台搭建后面板程序如图7所示,其中的ProtoParseModule组件就是由本文提出的协议解析器提

供的解析接口所编写的。

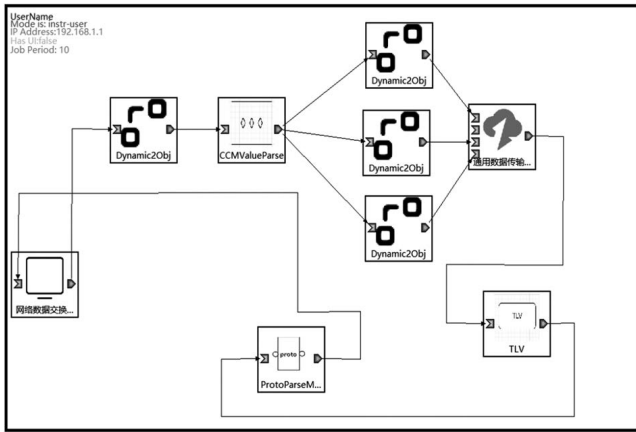


图7 后面板程序

Fig.7 Back panel program

按照本文提出的XML文件标签定义和层级关系编写XML协议描述文件。该协议描述文件的部分内容如图8、图9所示，可以看到数据遥测和数据注入返回内容两者的协议格式完全不同，如果可以使用解析器成功解析出数据信息则表示解析器具有通用性和异构性。

```
<!--数据注入返回内容-->
<frame name="数据注入返回内容" id="02">
  <starter var="0x0a,0x08"/>
  <block pass="01" name="state" length="6">
    <seg pass="01" name="innerInfo" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="00" name="state" length="1">
    <seg pass="00" name="errorCode" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="01" name="invalidData" length="13">
    <seg pass="01" name="var" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="00" name="数据注入返回内容" length="8">
    <seg pass="00" name="检测点1" dev="RMDZCWD1" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点2" dev="RMDZCWD2" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点3" dev="RMDZCWD3" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点4" dev="RMDZCWD4" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点5" dev="RMDZCWD5" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点6" dev="RMDZCWD6" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点7" dev="RMDZCWD7" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="检测点8" dev="RMDZCWD8" type="byte" typesize="1" unit="byte" endian="01"/>
  </block>
</frame>
```

图8 数据注入返回内容网络协议XML片段

Fig.8 Network protocol XML fragment of data injection and contents returning

```
<frame name="数据遥测" id="12">
  <starter var="0x0a,0x12"/>
  <block pass="01" name="state" length="6">
    <seg pass="01" name="innerInfo" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="00" name="code" length="1">
    <seg pass="00" name="errorCode" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="01" name="invalidData" length="13">
    <seg pass="01" name="var" type="byte" typesize="1" unit="byte" endian="00"/>
  </block>
  <block pass="00" name="dataBuffera" length="15" shouldCheck="true">
    <seg pass="00" name="帧头1" type="string" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="帧头2" type="string" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="帧头3" type="string" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="帧头4" type="string" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="同步字" type="string" typesize="2" unit="byte" endian="01"/>
    <seg pass="00" name="循环计数器" type="short" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="数据注入指令接收计数" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="数据注入指令错误计数" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="A" dev="QD" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="A" dev="SM" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="A" dev="SF" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="A" dev="JSJ" type="byte" typesize="1" unit="byte" endian="01"/>
    <seg pass="00" name="A" dev="RK" type="byte" typesize="1" unit="byte" endian="01"/>
  </block>
</frame>
```

图9 数据遥测网络协议XML片段

Fig.9 Network protocol XML fragment of data telemetry

5.2 实验

进行异构网络协议解析器实验测试，流程图如图10所示。



图10 实验流程图

Fig.10 Test flowchart

按照实验流程，通过客户端依次向后面板程序发送指令，后面板程序通过透传的方式向组合式仪器发送指令；组合式仪器模拟各项设备并产回传数据；回传数据按照不同的数据帧格式返回给后面板程序；后面板程序利用本文提出的解析器解析该数据帧，并将解析后的数据链返回给客户端展示。实验结果如图11所示。

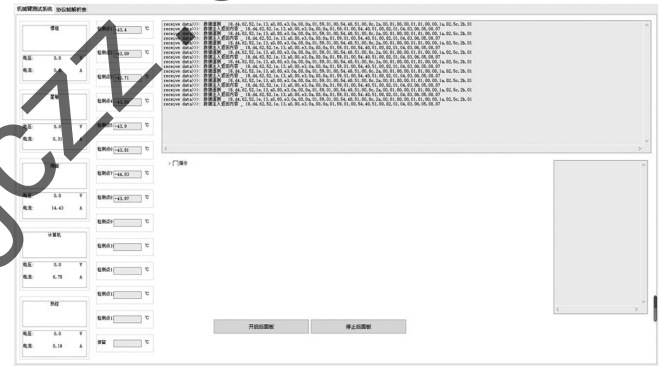


图11 客户端测试结果

Fig.11 Client test result

客户端发送的原始数据如图12所示。

```
//数字量遥测
(byte) 0x0a, (byte) 0x12,
(byte) 0xd4, (byte) 0x18, (byte) 0x52, (byte) 0x62, (byte) 0x13, (byte) 0x1e, (byte) 0x00, (byte) 0xa0, (byte) 0xa0, (byte) 0xa0, (byte) 0xe3,
(byte) 0x0a, (byte) 0x00, (byte) 0x55, (byte) 0x01, (byte) 0x01, (byte) 0x40, (byte) 0x54, (byte) 0x00, (byte) 0x51,
(byte) 0x2a, (byte) 0x0c, (byte) 0x01, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x01, (byte) 0x01, (byte) 0x00, (byte) 0x00, (byte) 0x00,
(byte) 0x02, (byte) 0x1a, (byte) 0x2b, (byte) 0x5c,
//数据注入返回内容
(byte) 0x0a, (byte) 0x08,
(byte) 0xd4, (byte) 0x18, (byte) 0x52, (byte) 0x62, (byte) 0x13, (byte) 0x1e, (byte) 0x00, (byte) 0xa0, (byte) 0xa0, (byte) 0xa0, (byte) 0xe3,
(byte) 0x0a, (byte) 0x00, (byte) 0x55, (byte) 0x01, (byte) 0x00, (byte) 0x01, (byte) 0x40, (byte) 0x54, (byte) 0x00, (byte) 0x51,
(byte) 0x01, (byte) 0x02, (byte) 0x03, (byte) 0x04, (byte) 0x05, (byte) 0x06, (byte) 0x07, (byte) 0x08
```

图12 原始数据

Fig.12 Raw data

从实验结果可以看出，解析器成功解析两个不同网络协议的数据，即数据遥测中的电压电流数据和数据注入返回内容的温度值，并在客户端页面中展示得到电压电流和检测点温度变化数据。通过比对模拟终端设备的发送原始数据，本文提出的异构网络协议解析器解析出的数据和发送端一致。

6 结论(Conclusion)

本文提出的基于XML的网络协议解析器是一种网络数

据解析的软件方法。针对XML文件结构具有异构性的特点,设计了本文提出解析器的XML标签协议描述文档,用户只需要按照指定的格式描述设备的网络协议即可在异构融合网络中实现多个设备的信息交互。实验结果表明,通过该解析器可以成功解析出不同设备的不同应用层协议。通过对协议解析XML文件中标签的组合和标签的属性修改可以完成解析跳过、循环解析和可变长区域的数据解析等复杂情况,验证了本文提出解析器的有效性、通用性,以及对异构融合网络的支持性高等特点。同时,也表明本文所提出的软件解析方法可以在工程研究、测试自动化、工业控制等领域推广使用。

参考文献(References)

- [1] KATZ R H, BREWER E A. Case for wireless overlay networks[D]. Berkeley: Univ. of California, Berkeley, 1996.
- [2] 赵宇,殷树娟,李翔宇.一种可重构以太网数据包解析器中可重构单元的设计[J].计算机工程与科学,2020,42(02):220-228.
- [3] 苗力心.软件定义协议转换关键技术研究[D].郑州:战略支援部队信息工程大学,2020.
- [4] 何卓桁,刘志勇,李璐,等.异构文本数据转换中XML解析方法对比研究[J].计算机工程,2020,46(07):286-293,299.
- [5] 卢宪.广域监控系统的可重构协议解析器设计与开发[D].南京:南京理工大学,2009.
- [6] 王孝龙,刘勤让,林森杰.一种支持异构协议解析的可配置解析器设计[J].计算机应用研究,2018,35(06):1830-1833.
- [7] 张会丽.异构融合网络联合资源分配算法研究[D].重庆:重庆邮电大学,2014.
- [8] 张玮.计算机内存及发展趋势[J].电脑知识与技术,2018,14(35):224-225.
- [9] 王永娜,赵奎,王鸿亮,等.针对异构协议的动态解析器模型[J].计算机系统应用,2017,26(01):251-254.
- [10] 王新亮,陈凯,薛琪琪,等.基于1553B总线的飞控软件测试仿真平台设计[J].计算机测量与控制,2020,28(11):12-15,20.
- [11] 蒋侃沅(1997-),男,硕士生.研究领域:机电系统集成,软件开发.
- [12] 李小明(1976-),男,博士,副教授.研究领域:机电系统集成,软件开发.本文通讯作者.
- [13] 别方法研究与实现[J].计算机仿真,2015,32(12):119-123.
- [14] 李哲,于梦茹.基于改进的HOG与LBP特征值的车标识别方法[J].计算机与数字工程,2019,47(12):3154-3159.
- [15] 王小龙.基于视频图像的车型识别算法研究与实现[D].西安:西安电子科技大学,2014.
- [16] YUAN B, XIA B, ZHANG D. Polarization image texture feature extraction algorithm based on CS-LBP operator[J]. Procedia Computer Science, 2018(131):295-301.
- [17] 郑秋梅,吕兴会,时公喜.基于多特征集成分类器的人脸表情识别[J].中国石油大学学报(自然科学版),2011,35(1):174-178.
- [18] 曲爱妍(1976-),女,硕士,高级工程师.研究领域:图像处理,人工智能,数据安全,数据分析.
- [19] 吴秋玲(1976-),女,博士,副教授.研究领域:信息隐藏技术,无线网络与信息安全.
- [20] 张正(1973-),男,本科,研究员.研究领域:专用网络安全,加解密技术,无线通信.本文通讯作者.
- [21] 梁颖红(1970-),女,博士,教授.研究领域:自然语言处理,智能对话系统.
- [22] 黄晓婷(1990-),女,硕士,工程师.研究领域:图像处理,DSP.

(上接第40页)

析[J].网络安全技术与应用,2021(04):120-121.

[2] 黄晓婷.基于DM8168双核DSP的车型识别系统[D].南京:南京航空航天大学,2015.

[3] 贺敏雪,余烨,程茹秋.特征增强策略驱动的车标识别.中国图象图形学报,2021,26(05):1030-1040.

[4] 张红兵.基于TMS320DM8127的车型识别系统的研究与实现[D].南京:南京航空航天大学,2015.

[5] 陈全义,陈三凤,胡涛,等.基于改进 Linemod-2D 的快速形状匹配算法[J].现代电子技术,2021,44(6):159-164.

[6] 刘欣悦.基于移动端的多特征花卉识别系统[D].长春:吉林大学,2019.

[7] 丁奇安.非限定环境下的车牌识别技术研究[D].芜湖:安徽工程大学,2019.

[8] 姜雪.基于多特征的图像检索方法研究[D].淄博:山东理工大学,2019.

[9] ERAZO-AUX J, LOAIZA-CORREA H, RESTREPO-GIRON A D. Histograms of oriented gradients for automatic detection of defective regions in thermograms[J]. Applied Optics, 2019, 58(13):3620-3629.

[10] 慕春雷.基于HOG特征的人脸识别系统[D].成都:电子科技大学,2013.

[11] 张红兵,李海林,黄晓婷,等.基于车前脸HOG特征的车型识