

船舶通信报文实时动态按位解析方法^①

郭梓暘^{②*} 姬庆庆^{** ***} 肖创柏^{*}

(^{*} 北京工业大学信息学部 北京 100124)

(^{**} 中国科学院大学 北京 100049)

(^{***} 中国科学院计算技术研究所虚拟现实实验室 北京 100190)

摘 要 在船舶系统中,随着报文数量的增加,无法对报文进行统一的配置。现有的报文解析技术存在着解析效率低,无法实时解析等缺点。针对上述问题,文章提出了一种基于数据库的报文配置方法和实时解析提取过滤的方法,该方法利用数据库对报文和其中的数据字段信息进行管理,并可以根据数据字段起始终止位数实现以位作为最小解析单位对报文进行快速解析和提取过滤。通过实际应用表明,此方法可以对报文进行统一配置,并提高了报文实时解析和提取过滤的效率,能满足实际需要。

关键词 动态配置, 实时解析, 报文过滤, 数据提取

0 引 言

船舶系统是一个耦合度高,各设备之间相互关联的复杂通信系统。目前,船舶系统内部各设备之间用以太网技术实现数据的通信^[1]。在各种设备开发调试的过程中,需要进行多次通信测试以保证设备运行的正确性^[2]。若在船舶航行中出现网络故障,仅仅依靠船员根据他们的经验和专业素养来进行故障判断,极有可能造成不可估量的损失。因此,无论是在船舶开发调试的过程中还是船舶网络故障发生时,如何快速对通信报文进行解析和调试都尤为重要^[3,4]。

目前船舶通信系统中实时通信的数据量较大,报文种类繁多,且报文针对新型设备的对接需要经常对报文数据字段的内容进行更改,所以对于如何动态地配置报文数据字段进行解析以及如何快速实时地通过解析对应数据从而滤掉无效、错误和用户“不关心”的网络原始数据的要求也在不断提高^[5]。

传统的报文解析方法,即以字符串解析的方式,在程序中对每一条报文都写一段解析代码从而实现了对数据报文中的数据字段逐个进行解析^[6]。这些解析代码逻辑相似,代码繁琐,且仅能以字节为最小单位进行解析。当报文中的数据字段发生变化时,只能对相应的解析代码进行修改。

近年来,许多学者针对特定的业务逻辑和应用场景对报文解析的方法进行了优化。Wang 等人^[7]提出了一种通过使用现场可编程门阵列(field programmable gate array, FPGA)等硬件来实现报文实时快速的解析方法,但该方法不易进行扩展且灵活性差。陈星亮^[8]提出了一种利用多线程技术提高报文数据解析的时效性的高效报文解析方法,但该方法只能作用于线下数据,无法进行实时解析。吴建斌等人^[9]提出了一种基于可扩展标记语言(extensible markup language, XML)的多格式报文解析方法,该方法可以对报文数据进行动态解析,但报文量过多时 XML 文件在读取与更改时会造成大量的时间开销和内存开销^[10]。上述各种报文解析方法虽然

① 国家自然科学基金(61501008),北京市科技计划(Z171100004717001)和北京市自然科学基金(4162007)资助项目。

② 男,1994 年生,硕士生;研究方向:通信技术,信号处理及计算机应用软件;联系人,E-mail: 416218173@qq.com

(收稿日期:2018-10-10)

能实现报文的解析,但都存在不足。目前的船舶通信系统中,报文种类繁多,报文的数据字段也经常会为了满足其他接口的需要进行修改。在开发调试或者维修检测的过程中,需要实时对特定的数据字段进行提取或过滤。在解析时,数据的规模 and 大小也往往不可预知,且为了满足实时性的要求,对于数据的处理只能是一次扫描,无法回溯^[11]。

本文在充分了解船舶通信系统实际应用特点的基础上^[12],提出了一种多协议船舶通信报文动态解析与快速提取过滤数据的方法。利用该方法既可以动态配置报文格式以位作为最小单位进行解析,也可以实现对某些特定的数据字段进行提取操作或过滤操作。

1 报文配置

1.1 报文定义

用户数据报协议 (user datagram protocol, UDP) 是一种无连接的传输层协议,提供简单不可靠的信息传送服务。

由于 UDP 报文在传输的过程中延迟小,且数据传输效率高,船舶通信系统中一般采用 UDP 报文进行通信。在 2 个设备 A 和 B 间交互时,设备 A 先给设备 B 发送 1 条命令报文,在设备 B 收到设备 A 的命令报文后,设备 B 向设备 A 发送确认报文,表明已经收到了设备 A 向设备 B 发送的命令报文。若设备 A 在一段时间内并未收到设备 B 发送来的确认报文,则重新向设备 B 发送命令报文,以此来保证通信之间的稳定性^[13]。

船舶通信报文结构如图 1 所示,船舶通信 UDP 报文主要由 UDP 报文头和 UDP 数据两部分组成。HeadData1, HeadData2, ..., HeadDataN 代表报文头中的信息字段,包括发方互联网协议 (Internet protocol, IP) 地址、发方端口号、目标 IP 地址、目标端口号及报文长度等信息。UDP 数据是真正需要传递的数据信息,UDP 数据的前 4 个字节为报文标识 (identification, ID), 报文 ID 可以唯一确定一条报文^[14]。对报文进行解析时,结合报文 ID 即可根据这条报文所定义的数据字段对接收的数据进行解析。Data1, Data2, ..., DataN 代表报文中的数据字

段,数据字段可由使用者自行进行定义^[15]。

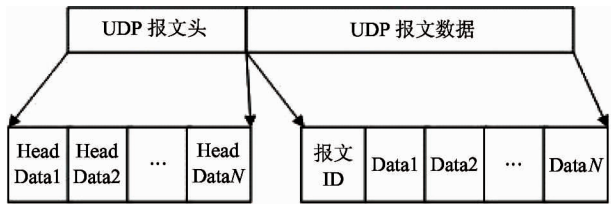


图 1 船舶通信报文结构

报文数据字段规范化定义如表 1 所示,报文数据字段根据需要对它的数据进行规范化定义,以此来保证报文配置模块可以满足所有报文的定义要求。其中字段名称、字段类型和长度是必填项,其他的数据根据字段类型需要而填写。常见的字段类型有整数型、小数型、枚举型、字符串型和循环型等。整数型有最大值和最小值要求,小数型需要确定其精度,枚举型需要确定每一解析值的准确含义,循环型需要知道详细的循环解析次数。起始位数和终止位数则是根据每个数据字段在该条报文中的位置由报文配置模块计算所得,确定了起始位数和终止位数可以快速确定这一字段在该条报文中的位置,由此获得指定数据。这样的优势在于不用循环解析出所有数据,便可以对报文数据进行快速提取和实时过滤。

表 1 报文数据字段规范化定义

数据特性	含义
Name	字段名称
Type	字段类型
BeginIndex	起始位数
EndIndex	终止位数
Length	长度
Precision	精度
Max	最大值
Min	最小值
BitEnum	枚举值
Loop	循环解析次数

1.2 数据库设计

报文配置以及报文解析所用到的数据库总体设计如图 2 所示。

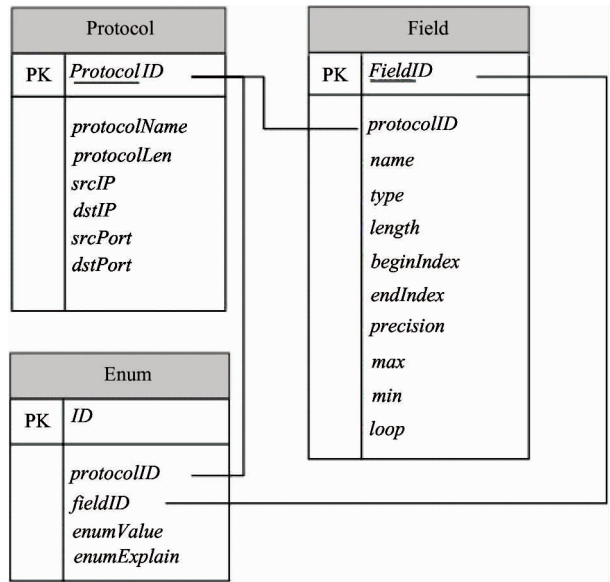


图 2 数据库的总体设计

表 2 至表 4 分别列出数据库中每张表中的具体数据及含义。

报文表如表 2 所示,报文 ID 作为报文的唯一标识,为该表主键。报文名称和长度为必填项,为每一条报文的基本信息。IP 地址和端口号作为选填项,可用于验证报文发送的正确性。

字段表如表 3 所示,字段在该报文中的顺序 ID 作为字段数据的唯一标识,为该表主键。报文 ID 为外键,用于确定该条数据字段所对应的报文。字段名称、字段数据类型、字段长度、字段在报文数据中的起始和终止位置为必填项,是每一个数据字段的基本信息。数据数值精度、最大值、最小值和重复次数作为选填项,根据数据类型的需要进行填写。

表 2 Protocol 报文表

名称	数据类型	长度	能否为空	备注
<i>protocolID</i>	Int	4	否	报文 ID,主键
<i>protocolName</i>	nvarchar(50)	50	否	报文名称
<i>protocolLen</i>	Int	4	否	报文长度
<i>SrcIP</i>	nvarchar(16)	16	是	源 IP 地址(字符串)
<i>dstIP</i>	nvarchar(16)	16	是	目的 IP 地址(字符串)
<i>srcPort</i>	Int	4	是	源端口号(数据范围 0 ~ 65 533)
<i>dstPort</i>	Int	4	是	目的端口号(数据范围 0 ~ 65 533)

表 3 Field 字段表

名称	数据类型	长度	能否为空	备注
<i>FieldID</i>	int	4	否	字段顺序 ID,主键
<i>protoID</i>	int	4	否	报文 ID,外键
<i>name</i>	nvarchar(50)	50	否	字段名称
<i>type</i>	nvarchar(50)	50	否	字段数据类型
<i>Length</i>	Int	4	否	字段长度
<i>BeginIndex</i>	int	4	否	字段在报文数据中的起始位置(单位:位)
<i>EndIndex</i>	int	4	否	字段在报文数据中的终止位置(单位:位)
<i>Precision</i>	Float	4	是	数据数值精度
<i>min</i>	float	4	是	最小值
<i>max</i>	float	4	是	最大值
<i>Loop</i>	Int	4	是	重复次数

枚举表如表 4 所示,枚举值 ID 作为枚举值的唯一标识,为该表主键。报文 ID 和字段顺序 ID 作为

外键,用于确定该枚举值所对应的报文中的数据字段位置。枚举值和枚举值所对应的解释为必填项。

表 4 Enum 枚举表

名称	数据类型	长度	能否为空	备注
ID	int	4	否	枚举值 ID, 主键
ProtoID	int	4	否	报文 ID, 外键
FieldID	int	4	否	字段顺序 ID, 外键
EnumValue	int	4	否	数字型枚举值
Enum_explain	nvarchar(50)	50	否	枚举值所对应的解释

1.3 报文配置方法

针对传统报文配置需要专业化人员对数据库文件进行配置带来的通用性差的特点,本文在进行报文配置时,设计了报文配置模块,提供良好的操作界面,用户在填写指定信息后,该模块便可以将所有的数据字段信息写入数据库中。报文配置模块在 Windows 平台下采用 C#语言开发,结合 C#中的 dev 界面插件可以给用户提供一个非常直观的编辑界面。使用者可根据需要对报文的数据字段进行添加和修改操作,该模块为报文的动态解析和数据的实时过滤提供服务。

报文配置模块的具体工作流程如图 3 所示。

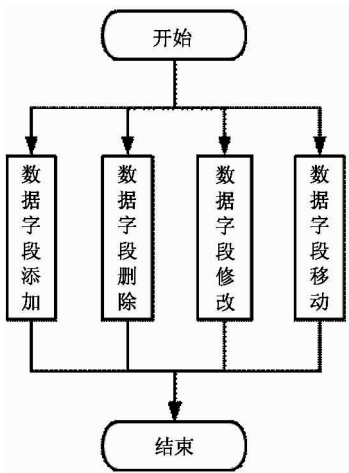


图 3 报文配置模块流程图

(1)启动报文配置模块,根据报文的唯一标识报文 ID 从数据库中取出该条报文下的所有数据字段信息,并根据数据字段的先后顺序以表格的形式显示。

(2)若想要添加数据字段,点击数据字段添加按钮,确定字段类型并填写该字段类型需要的字段信息完成数据字段的添加。完成添加后,模块会根

据已知数据字段的总长度和新添加的数据字段的长度来计算出新添加的数据字段的起始位数和终止位数并显示在数据字段表格中。

(3)若想要修改已添加数据字段的信息,选中待修改数据字段并点击修改按钮,根据数据字段的类型重新填写需要的字段信息。完成修改后,模块会根据每一条数据字段的长度重新计算待修改数据字段之后的每一条数据字段的起始位数和终止位数,并对待修改数据字段的信息进行刷新,显示在数据字段表格中。

(4)若想要修改已添加数据字段的顺序,选中待修改数据字段并点击上移或下移按钮。模块会对包含当前操作的数据字段之后的每一个数据字段重新计算其起始位数和终止位数,并显示在数据字段表格中。

(5)点击完成按钮结束对当前报文数据字段的配置并保存,模块根据最终数据字段表格显示的内容对数据库进行更改。

2 报文解析方法

2.1 按位获取数据

目前,报文中的很多数据字段都以位作为最小单位,从而避免了以字节作为最小单位解析时补位所造成的数据冗余,极大程度保证了报文中的每一位数据都是真实有效的。接收到的报文是以字节数组的形式存储在缓存中的,解析的首要任务是通过每个数据字段的起始位数和终止位数,从原始报文字节数组中获取该字段数据值,再根据数据字段的类型进行下一步处理。按位拼接字段数据值算法根据数据字段的位数循环判定,通过字符串与操作和拼接操作获取字段数据值,具体的算法描述如表 5 所示。

表 5 按位拼接字段数据值算法

按位拼接数据算法
全局变量: <code>_and[]</code> : 与操作数数组
输入:
<code>I</code> : 起始位数
<code>EI</code> : 终止位数
<code>rawData[]</code> : 原始报文字节数组
输出:
<code>handleData</code> : 字段数据值
Start:
<code>bit = EI - BI + 1</code> // 计算位数长度
<code>rawBBI = BI / 8</code> // 计算起始字节
<code>RawBEI = EI / 8 + 1</code> // 计算终止字节
For <code>i</code> from 0 to <code>rawBEI - rawBBI</code>
Do
IF <code>i == 0</code> // 计算第一字节
THEN 若字段数据值在第一个字节内, 字段数据值为原始报文中第 <code>rawBBI</code> 字节的值右移 $(8 - BI \% 8)$ 位后的最后 $(bit - 1)$ 位的值。否则, 字段数据值拼接原始报文中第 <code>rawBBI</code> 字节后 <code>m_index</code> 位的值。
ELSE IF <code>i > 0 && i < rawBEI - rawBBI - 1</code> // 计算中间字节
THEN 字段数据值拼接原始报文中第 <code>rawBBI + i</code> 字节的值
ELSE
THEN 字段数据值拼接原始报文中第 <code>rawBBI + i</code> 字节的值右移 <code>m_index</code> 位的值
ENDIF
End
按位拼接数据算法

2.2 报文解析

通过报文配置模块配置后, 可对已配置好的所有报文进行解析。解析的具体流程如图 4 所示。

(1) 利用 WinPcap 对数据包进行获取^[16-18], 获取的报文放入报文缓存中。若报文缓存不为空, 则取出缓存中的第一条报文进行解析。

(2) 从 UDP 头解析出报文的长度, 再从 UDP 头结尾处向右移动 4 位解析出报文的唯一标识报文 ID。根据报文 ID, 从报文表中找到该条报文的具体内容。若从报文库中找不到该条报文 ID 或报文的实际长度与报文配置的长度不符, 则报出相关错误。

(3) 通过报文 ID 从数据字段表中找到该条报文下的所有数据字段, 根据每个数据字段的起始终止位数和类型以此进行解析。例如某数据字段是补

码形式的浮点型, 则先从该数据字段表中取出起始位数、终止位数、最大值、最小值和精度。通过起始位数和终止位数, 对数据进行拼接, 拼接完毕后将补码形式转化为对应的十进制数字, 然后与精度值相乘。最后对此结果进行判定, 若在最小值和最大值区间内, 则继续进行下一数据字段的解析, 否则报出相关错误。

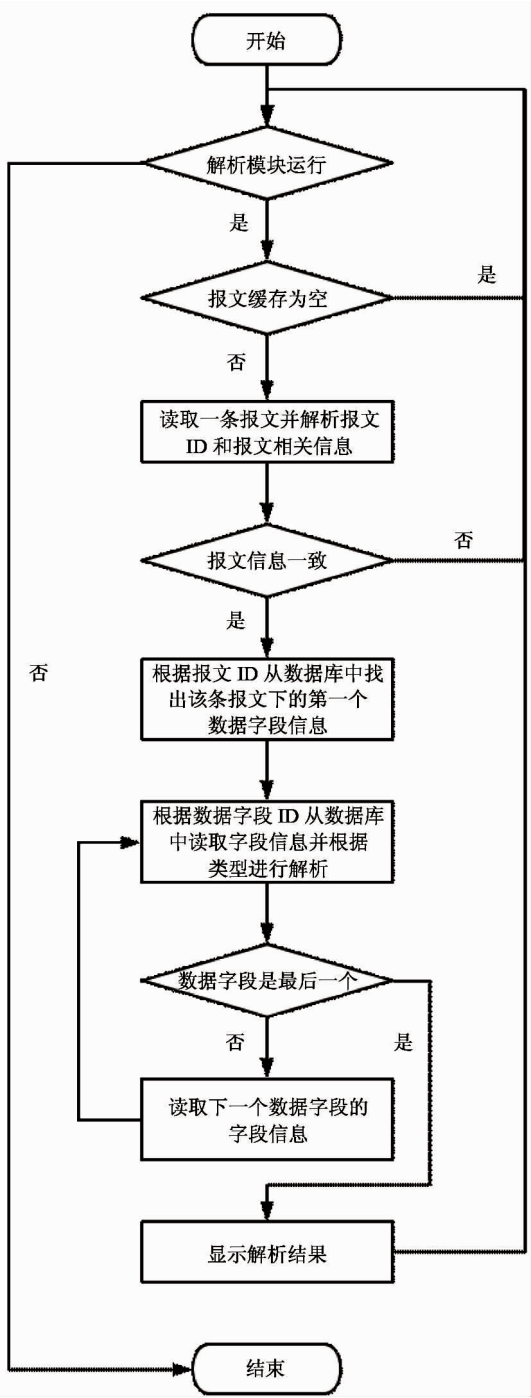


图 4 报文解析流程图

(4)显示解析结果。

2.3 报文提取与过滤

在船舶设备开发调试或维修的过程中,经常会对一个或者多个数据字段进行监控,从而更加精准地判断数据的走向或者以此来排查一些偶发的事件。单靠手动通过解析后的报文对某些数据字段进行提取或者逐条从中发现问题所在,会消耗大量的时间,极有可能带来一些严重的后果。基于上述报文配置模块,本文提出一种报文实时提取过滤的方法,可以实时对指定数据进行提取或过滤,便于用户对数据的分析以及异常发现。具体流程如图 5 所示。

(1)通过人机交互界面从报文数据库中选择需要提取或过滤的报文和其中的数据字段。若需要以某一个条件对数据字段进行过滤,还需要设置此字段的过滤条件,例如这个数据大于某值等。

(2)若报文缓存不为空,则取出缓存中的第一条报文进行解析。

(3)从 UDP 头解析出报文的长度,再从 UDP 头结尾处向右移动 4 位解析出报文的唯一标识报文 ID。若该报文 ID 未被选择,则不进行任何解析操作。否则,根据报文 ID,从报文表中找到该条报文的详细内容。若从报文库中找不到该条报文 ID 或报文的实际长度与报文配置的长度不符,则报出相关错误。

(4)通过报文 ID 从数据字段表中找到该条报文下所选取的需要提取或过滤的数据字段,根据这些数据字段的起始终止位数直接定位到对应数据,并进行解析。

(5)若进行提取操作,则显示解析结果,并将所选待提取的数据字段进行输出。

(6)若进行过滤操作,则根据解析结果与过滤条件进行判定。满足条件则显示这条解析结果,不满足条件则不进行显示。

3 性能指标测试

为了确保动态配置和实时解析过滤方法的实用性,本文根据实际情况模拟了一个测试环境,并在此

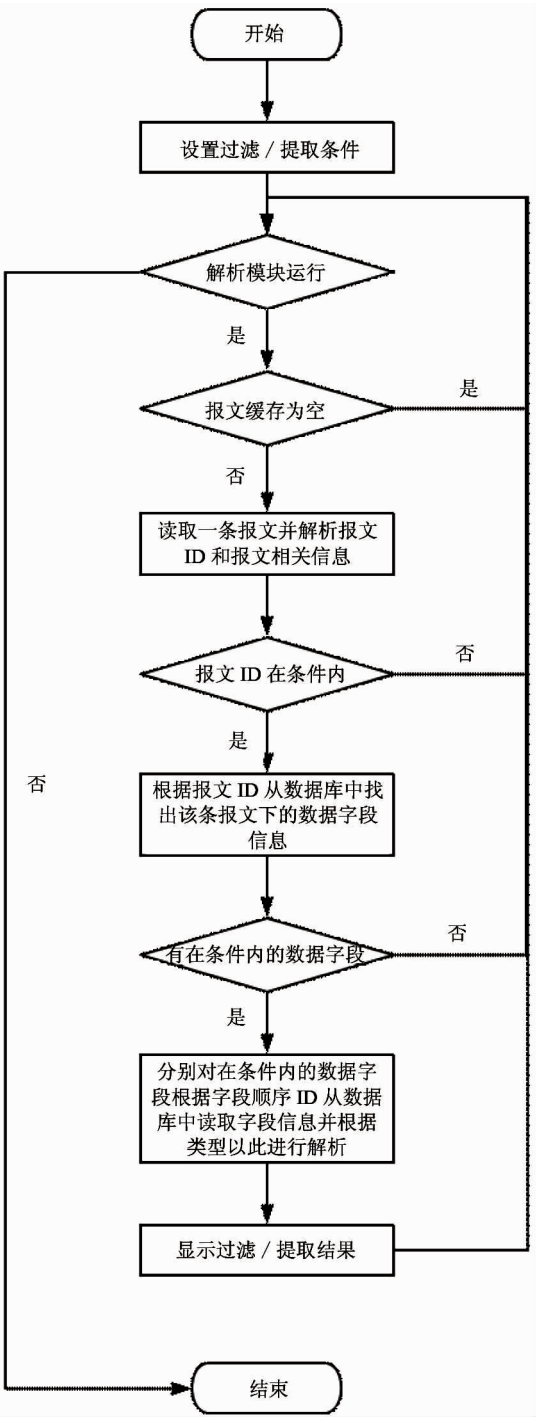


图 5 报文实时提取过滤流程图

环境中分别对实时解析、过滤和提取三个功能进行测试。

上述测试环境中,硬件部分包括 3 台计算机,10 个子设备,1 个交换机。3 台电脑中,1 台作为整个系统的控制计算机,可对所有的子设备进行控制;1 台作为报文模拟发送计算机,持续模拟发送带有数据的报文;1 台作为解析计算机,有上述报文配置模

块,报文解析模块以及报文提取过滤模块。所有设备间都使用以太网进行连接,所有设备和电脑都通过交换机连接到同一局域网下。

实验过程如下:

(1)启动报文动态配置模块,对所有实验中用到的报文进行配置。实验中用到的报文分为两种,一种为船舶设备每隔一段时间给解析计算机发送的设备状态报文,另一种为报文模拟发送计算机模拟发送的带有很多数据信息的报文,报文中的数据可任意进行设置。

(2)通过控制计算机启动整个实验的通信环境。船舶设备的状态报文每秒向解析计算机发送 2 条。报文模拟计算机模拟发送 100 条带有数据的报文,每秒向解析计算机发送 50 条,且这些报文中的数据自行定义便于过滤以及检测。

(3)启动报文解析模块,捕获 100 000 条数据报文。

(4)设置过滤条件,启动报文过滤模块,捕获 100 000 条数据报文。

(5)设计提取条件,启动报文提取模块,捕获 100 000 条数据报文。

性能测试指标如表 6 所示。

表 6 性能测试指标

性能	指标
每条报文处理时间	60 ~ 70 μ s
吞吐量	14 200 ~ 16 600 条/s
每秒处理报文数	5 020
包捕获率	99.9%
过滤正确率	99.5%
提取准确率	99.6%

从实验中可以看出,对于单条报文的处理时间远小于 1 ms,完全可以满足船舶通信系统实时解析的需求。船舶系统的子设备大多以每秒 2 条至 50 条的频率进行通信,实验中模拟 100 个设备以每秒 50 条的频率进行通信,包捕获率为 99.9%,说明解析模块工作基本稳定。丢失的少量数据包有可能是因为在网络不稳定的时候导致丢失,也有可能是在旧缓冲区满了之后开辟新缓冲区的过程中没有成功

地将这部分数据包捕获至缓冲区内。

通过对过滤功能和提取功能的实验,其正确率和准确率均大于 99.5%,说明这两个模块可以实时地对数据进行解析以及提取或过滤操作,极大程度上保证过滤后结果的准确性。没有成功过滤或提取的报文可能是因为报文在解析的过程中由于报文数据信息与数据库中填写的数据不一致,提取与过滤模块自动将这些报文当做无效报文所导致。对于船舶设备来说,每秒接收的报文在 2 条至 50 条不等,偶尔丢失掉几条报文并不会对过滤或提取到的结果产生影响。

4 结 论

本文提出了一种多协议船舶通信报文动态解析与快速提取过滤数据的方法。用数据库对所有的报文进行统一管理,并通过报文配置模块完成对报文的动态配置。报文的解析方法与提取过滤方法也可以让用户实时对报文进行解析,对所需数据字段进行提取、过滤,且处理速度较快能够满足开发调试以及维护需要。相对于传统的解析方法来说,该方法有以下三方面的优势:第一,以位作为最小单位进行解析,避免了以字节作为最小单位解析时补位所造成的数据冗余,极大程度保证了报文中的每一位数据都是真实有效的;第二,通过报文配置模块和解析模块针对船舶报文的业务逻辑实现报文的自动解析,减轻了相关报文通讯类软件的开发成本;第三,根据船舶系统的需求,实现了实时过滤和提取的功能,处理时间能满足需要。

本文所提出的方法已在民用船舶上试用并取得了理想的报文处理结果,但目前可配置的数据字段类型可能还不全面,现阶段在相关代码部分已经留出了相应的开发接口,未来将针对这些问题进行改善,从而使本方法在更多船舶上或其他工业领域得到推广应用。

参考文献

[1] Adhane G W, Kim D S. Distributed control system for ship engines using dual fieldbus[J]. *Computer Standards & Interfaces*, 2017, 50:83-91

- [2] An H, An W G, Zhao W T. Reliability analysis of ship structure system with multi-defects [J]. *Science China Physics Mechanics & Astronomy*, 2010, 53 (11) :2113-2124
- [3] Wu Y , Yao J , Liu B . A reliability analysis method of network structure reconfiguration for phased mission system [J]. *Cluster Computing*, 2018 (2) :1-9
- [4] Chao, Chiatso M, Wu, et al. A clustering- and probability-based approach for time-multiplexed FPGA partitioning [J]. *Integration the Vlsi Journal*, 1999, 38 (2) :245-265
- [5] Li Y, Qin X. A hybrid fault diagnosis model in distributed application management [J]. *Journal of Computer Research & Development*, 2010, 47 (3) :455-462
- [6] 孙俊男, 刘明哲, 徐皑冬, 等. 基于 IEC60870-5-104 远动规约的 PLC 通信模块的设计与实现 [J]. 高技术通讯, 2016, 26 (4) :389-395
- [7] Wang X, Li X U, Xiao L I, et al. Design and realization of GOOSE decoding module based on FPGA [J]. *Power System Protection & Control*, 2015, 43 (24) :129-135
- [8] 陈星亮. 海洋水文气象报文自动解析系统的设计与实现 [A]. 中国海洋学会. “一带一路”战略与海洋科技创新——中国海洋学会 2015 年学术论文集 [C]. 中国海洋学会:中国海洋学会, 2015, 9:149-157
- [9] 吴建斌, 邹佩江. 基于 XML 的多协议报文解析的方法与实现 [J]. 电子测量技术, 2017, 40 (3) :168-172
- [10] Jansen D, Buttner H. Real-time ethernet: the EtherCAT solution [J]. *Computing & Control Engineering*, 2004, 15 (1) :16-21
- [11] Jiang Q, Chakravarthy S. Stream data processing: a quality of service perspective [J]. *Advances in Database Systems*, 2009, 36:9-21
- [12] Luo Y, Chan-Lea W U, Xiong W C. System hierarchy method based on self-resolving message protocol [J]. *Computer Engineering*, 2012, 38 (4) :79-81
- [13] Zhang Z X, Shang X W. Describing and interpreting method of variant-length protocol data [J]. *Ship Electronic Engineering*, 2015, 35 (5) :62-64
- [14] Tindell K, Burns A, Wellings A J. Calculating controller area network (can) message response times [J]. *Control Engineering Practice*, 1995, 3 (8) :1163-1169
- [15] 杨海涛, 刘琼. 局域网环境下 Peer-to-Peer 应用的识别与监测 [J]. 高技术通讯, 2008, 18 (7) :666-672
- [16] 刘德胜, 黄芝平, 唐贵林, 等. 基于 WinPcap 的网络化测试通信层设计 [J]. 自动化仪表, 2012, 33 (7) :69-72
- [17] Zhou J F. WinPCap based experimental system for protocols of computer network [J]. *Research & Exploration in Laboratory*, 2009, 28 (12) :68-70
- [18] Ren H Z, Nian M. Method for high speed network packet capture based on DPDK [J]. *Computer Systems & Applications*, 2018, 27 (6) :242-245

Method of dynamic real-time parsing and filtering for ship communication messages

Guo Ziyang^{*}, Ji Qingqing^{** ***}, Xiao Chuangbai^{*}

(^{*} Faculty of Information Technology, Beijing University of Technology, Beijing 100124)

(^{**} University of Chinese Academy of Sciences, Beijing 100049)

(^{***} Virtual Reality Laboratory, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

Abstract

In the ship system, as the number of message increases, the message cannot be uniformly configured. The existing message parsing techniques have the disadvantages of slow parsing efficiency and inability to parse in real time. Aiming at the above problems, this paper proposes a database-based message configuration method and real-time parsing, extracting and filtering method. The proposed method uses the database to manage the message and the data field information therein. The message can be quickly parsed and filtered with bits as the minimum parsing unit according to the starting digits and ending digits of the data field. The actual application shows that the proposed method can uniformly configure message and improve the efficiency of real-time parsing and extracting or filtering of packets to meet actual needs.

Key words: dynamic configuration, real-time parsing, message filtering, data extracting