

基于 1553B 和 SpaceWire 组合网络的星载 数据管理软件设计

于俊慧, 穆 强, 牛跃华, 张亚航
(北京空间飞行器总体设计部 北京 100094)

摘要: 为同时满足对于星务管理数据和高速载荷数据的传输要求, 国内外的飞行器趋于使用 1553B 和 SpaceWire 组合网络来进行星上数据传输。复杂而灵活的数据传输要求使得卫星数据流设计中引入了大量标准协议, 同时大大增加了星载数据管理软件的复杂度。对一种基于 1553B 和 SpaceWire 组合网络的星载数据管理软件的协议体系进行了介绍, 通过对协议进行分析, 改进了已有的软件框架, 形成了一种基于标准协议的分层的软件架构, 以空间包为单位实现星内数据交互, 实现了软件应用层与数据链路协议的分离。软件实现结果表明, 合理的协议分层和有效的接口封装提高了软件的可拓展性和可移植性, 有利于软件产品化。

关键词: 星载数据管理; 1553B; SpaceWire; 星载软件

中图分类号: V556.4 **文献标识码:** A **文章编号:** CN11-1780(2021)04-0005-07

DOI: 10.12347/j.ycyk.20210324002.

引用格式: 于俊慧, 穆强, 牛跃华, 等. 基于 1553B 和 SpaceWire 组合网络的星载数据管理软件设计[J/OL]. 遥测遥控, 2021, 42(4): 53–59[20XX-XX-XX]. <http://ycyk.brit.com.cn/ycyk/article/abstract/20210324002>.

Onboard OBDH software design based on combined network of 1553B and SpaceWire

YU Junhui, MU Qiang, NIU Yuehua, ZHANG Yahang

(Beijing Institute of Spacecraft System Engineering, Beijing 100094, China)

Abstract: Spacecraft tends to use combined network of 1553B and SpaceWire to meet the transfer need of both platform management data and payload data. Lots of standard protocols are applied to provide variable and flexible data service, which makes onboard software much more complicated. A protocol set of onboard OBDH software applying combined network of 1553B and SpaceWire is introduced in this article. Based on analysis on the protocols, the software frame as well as the route-table on board is improved. It is proved that the software is much easier to transform and maintain by applying reasonable layer and encapsulation, which can do good to software production as well.

Key words: OBDH; 1553B; SpaceWire; Onboard software

DOI: 10.12347/j.ycyk.20210324002

Citation: YU Junhui, MU Qiang, NIU Yuehua, et al. Onboard OBDH software design based on combined network of 1553B and SpaceWire [J/OL]. Journal of Telemetry, Tracking and Command, 2021, 42(4): 53–59[20XX-XX-XX]. <http://ycyk.brit.com.cn/ycyk/article/abstract/20210324002>.

引 言

随着航天任务的不断发展, 星载软件规模越来越大, 软件系统的复杂度日益增加, 对星地、星内数据传输的要求也不断提高。目前, 空间飞行器的数据处理系统通常需要处理两种数据: 对时间较为敏感的星务管理数据以及对带宽有较高要求的载荷数据^[1]。传统的高速载荷数据传输通常采用 LVDS 电平标准进行, 但 LVDS 只能进行点对点的数据传输, 不能形成高速数据网络。SpaceWire 是一种高速低耗的网络连接, 在空间飞行器中得到了广泛应用, 但 SpaceWire 中使用的虫洞路由使得其网络中的

数据传输延迟难以预估^[2]。因此, 目前国内外卫星趋于使用传统的 1553B 总线实现低速的星务管理数据传输, 同时构建高速 SpaceWire 网络用于多个高速载荷之间的数据传输, 进而实现星内高低速混合网络的构建^[3]。

复杂的任务和数据传输模式, 使得星载数据管理软件的设计复杂度越来越高。为了满足日益复杂的需求, 在星地、星内传输协议中越来越多地引入国外标准协议, 比如空间数据系统咨询委员会 CCSDS 的相关标准协议、欧洲标准化合作组织 ECSS 的相关标准协议等^[4]。本文介绍了一种基于 1553B 和 SpaceWire 组合网络的星载数据管理软件的协议体系, 并在此基础上, 介绍了分层的星载数据管理软件设计情况。通过适当的分层和构件封装, 将上层应用与传输协议分离, 有效地提升了软件的可移植性和可复用性, 提高了软件开发效率和软件可靠性。

1 协议体系

星载数据管理分系统作为星上数据交互的中心, 负责了星务管理数据的组织、汇集和转发、星上网络管理、部分载荷数据存储等功能。星务管理数据通常包括地面上行的遥控指令、星上下传的遥测数据以及部分星内的交互数据(比如时间信息等), 这些数据传输对实时性要求比较高, 通常需要明确传输延迟或以固定时序发送。数据管理软件从测控分系统接收上行遥控数据, 根据遥控协议格式, 完成遥控数据的获取和校验。经过校验的遥控数据或者在数据管理分系统被执行, 或者通过 1553B 总线转发至相应终端执行。随着卫星功能越来越复杂, 遥控数据的种类变得越来越复杂, 同时由于卫星测控上行带宽有限, 遥控协议需能为用户提供灵活服务的同时, 尽可能提高上行数据传输效率。同时, 数据管理软件还从 1553B 总线获取星上各终端的遥测数据, 连同自身采集的遥测数据一起按照遥测协议进行组织, 然后发送至测控分系统下传地面。遥测数据用于向地面反馈卫星状态, 可以细分为实时遥测、突发遥测、延时遥测等多种类型, 不同类型的遥测传输要求也各不相同, 其目的在于快速、准确、全面地向用户表征卫星健康状态和任务执行情况。载荷数据主要是星上载荷设备、各类传感器产生的业务数据, 各终端按照 SpaceWire 网络协议要求组织数据, 发送至存储模块暂存, 之后通过数传分系统下传至地面。这些数据对于传输的实效性要求不高, 但数据量较大, 对总线带宽要求较高。

通过上述分析可以看出, 为了同时满足高低速数据传输的要求, 星载数据管理软件涉及大量的协议处理, 包括星地数据链路协议(遥控、遥测协议)、1553B 总线协议以及 SpaceWire 总线协议。为了提供灵活、多样的用户服务, 这些协议均采用或借鉴了标准协议。本节以某星载应用为例, 对上述协议进行简单介绍。

1.1 遥控协议

遥控协议采用了标准的分包遥控协议^[5]。星载数据管理软件需要处理的空间数据协议格式为遥控传送帧。遥控传送帧由帧主导头、帧数据域和帧差错控制域组成, 格式如图 1 所示。星上通过虚拟信道来区分不同的服务类型, 即在整个任务期间, 一个虚拟通道只能支持一种服务。传送帧数据域由一个或多个遥控包首尾相接构成。星载数据管理软件需对数据完成校验后, 提取传送帧数据域中的遥控包进行处理或转发。

分包遥控协议相对于传统 PCM 遥控协议的优势非常明显: 一方面通过虚拟信道的使用, 使得不同类型的数据可以复用同一个物理信道, 提高了用户使用的灵活性; 另一方面在分包遥控中, 遥控帧为变长且其中可以插入多个遥控包, 而不再需要像 PCM 遥控帧一样进行填充, 提高了信道使用效率。

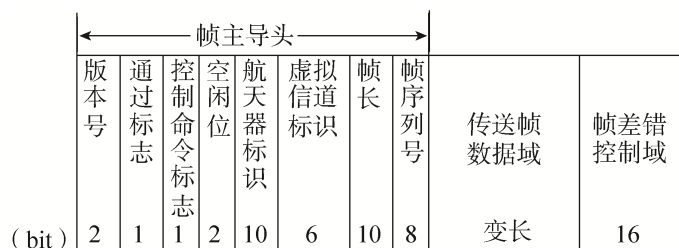


图 1 遥控传送帧标准数据格式

Fig. 1 Standard data structure of telecommand transfer frame

1.2 遥测协议

遥测数据采用 CCSDS AOS 推荐的信道访问数据单元 (CADU) 格式^[6], 如图 2 所示。AOS 传输帧由传输帧帧头、传输帧插入域, 传输帧数据域、传输帧尾四部分组成; 传输帧数据域包含一个 M_PDU (多路复用协议数据单元) 或一个空闲数据。在某一虚拟信道同时只能存在一种上述数据单元, 特定虚拟信道上传送的数据单元在整个任务阶段不变。

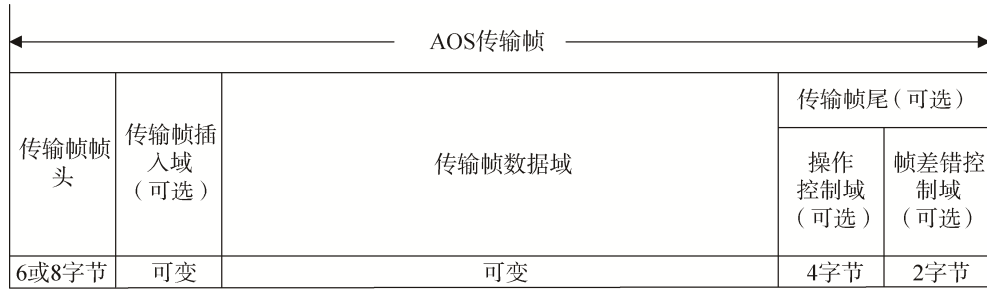


图 2 AOS 传输帧结构
Fig. 2 Data structure of AOS transfer frame

其中, 业务数据装载于传输帧数据域, 采用 M_PDU 的形式传输。M_PDU 由连续的包组成, 超过最大 M_PDU 长度的包会被分开, 完全填满 M_PDU, 并且剩余部分从同一个虚拟信道的新 M_PDU 开始。下一个 M_PDU 继续由级联的包组成, 直到它溢出为止, 其格式如下:

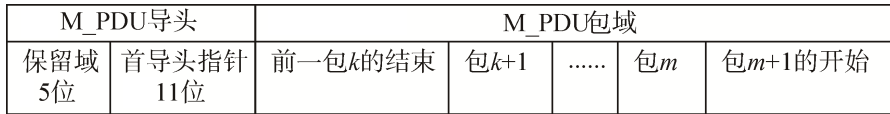


图 3 M_PDU 数据格式
Fig. 3 Data structure of M_PDU

AOS 遥测协议为用户提供了非常灵活的数据传输方式, 已经广泛用于我国的航天器。通过对虚拟信道的划分, 可以允许不用种类的数据以不同的优先级和调度方式来占用物理信道; 对于相同虚拟信道中的不同源包, 可以通过对遥测包周期进行调度, 实现对不同数据传输速率的控制; 另外, 传输帧插入域还可以实现为时间敏感信息提供时延可控的下传服务。

1.3 1553B 总线协议

1553B 总线协议采用了 ECSS 提出的基于时间同步的 1553B 总线通信协议, 在原有美军标的基础上, 提供了时间服务、通信同步服务、置数/取数服务、数据块传输服务以及终端管理服务等五种服务^[7], 其中通信同步服务是强制服务, 其它四种服务均依赖于通信同步服务, 五种服务间的依赖关系如图 4 所示。

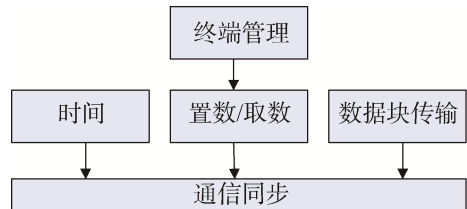


图 4 服务依赖
Fig. 4 Service reliance

基于时间同步的 1553B 总线通信协议就是通过通信同步服务, 将一个时间周期内的总线数据传输划分为 N 个通信帧, 在每个通信帧中, 又进一步将数据传输细分为预分配带宽和未预分配带宽。通过对于不同优先级的数据使用不用服务的方式, 可以实现获得固定传输延迟或者获得可靠握手的数据传输效果。同时, 通过合理约定数据通信方式, 可以达到总线资源合理分配的效果, 如图 5 所示。通信同步服务和时间服务为时间同步的 1553B 总线通信协议的基础服务, 而数据传输则根据其优先级和时序安排分别采用置数/取数或者数据块传输服务进行, 数据传输的最小数据格式为空间包。

基于时间同步的 1553B 总线通信协议改变了传统 1553B 总线碎片化的通信方式, 整个总线通信为集中式管理, 通过合理的资源分配可以提高总线带宽使用率; 同时, 通过对置数/取数、数据块传输等上层

服务行为的约定, 总线管理器不再需要对每个总线终端进行识别, 只需要通过数据类型来区分使用的服务, 有利于系统的迁移和扩展。

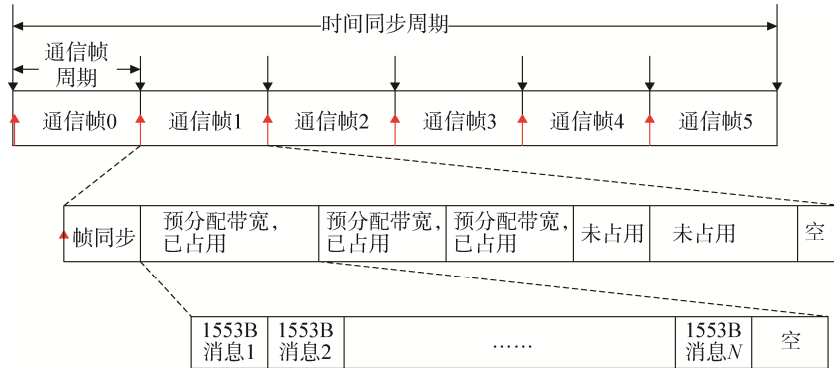


图 5 通信帧分解图示
Fig. 5 Decomposition of communication frame

1.4 SpaceWire 网络协议

SpaceWire 网络数据链路接口协议遵循 ECSS-E-ST-50-12C 标准《SpaceWire-Links, nodes, routers and networks》的规定。

卫星 SpaceWire 总线网络中传输的数据主要包括两类：

- ① 业务类数据：通常为连续大块的数据传输，包括载荷观测产生的业务数据、平台监测产生的业务数据、上行注入数据、下传地面数据等；
- ② 控制类数据：通常为多种类、高频度、变长度的短数据包，包括配置设备工作模式的数据，设置工作状态的数据等，此类数据为安全关键性数据，具有高可靠性的要求。

按照 SpaceWire 网络中传输的数据性质，相应的将网络传输业务划分为两类：

- I 类数据业务：提供卫星业务类数据传输服务，用于结点设备通过 SpaceWire 网络传输连续大块数据，主要是星上传感器产生的测量数据。对于上注的大块数据，在数据内部设计容错校验手段以增强可靠性。
- II 类数据业务：提供卫星控制类数据传输服务，用于传输安全关键性数据，包括星上遥测数据备份、相机积分时间数据，姿态与轨道数据、导航信息数据、配置路由单元和各结点设备 SpaceWire 控制器芯片的路由表数据等。

I 类数据业务传输采用 SpaceWire 总线网络的串行传输通用协议 STUP (Serial Transfer Universal Protocol)，ECSS-E-ST-50-11C 标准指定协议标识号为 0xEF。为了便于使用和业务拓展，SpaceWire 网络传输的 I 类业务数据包格式分为两级，第一级采用 STUP 协议数据包基本格式，第二级在协议数据包的装载数据区中，定义星上用户数据结构，由数据文件标识和虚拟信道数据单元 (VCDU) 构成，其协议数据包格式如图 6 所示。

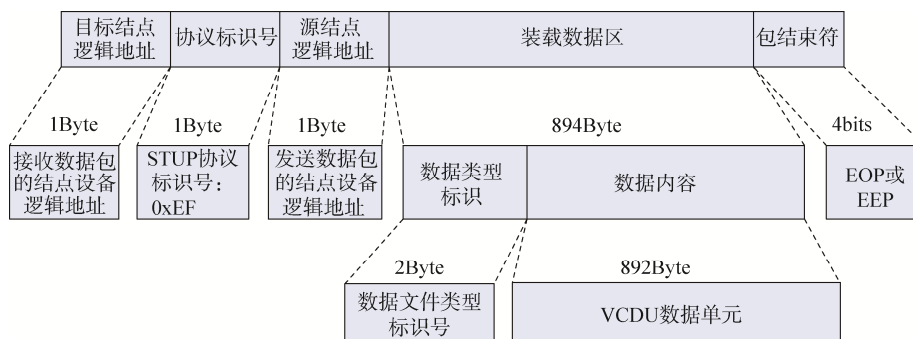


图 6 I 类业务数据包格式
Fig. 6 Data form applying service type I

II 类数据业务传输采用 SpaceWire 总线网络的远程存储访问协议 RMAP (Remote Memory Access Protocol), ECSS-E-ST-50-11C 标准指定协议标识号为 0x01。SpaceWire 网络传输的 RMAP 协议数据包分为两类:

- ① 写入命令包, 用于发送卫星 II 类业务数据;
- ② 读取命令包, 用于上位机读取 SpaceWire 控制芯片的内部状态。各数据包格式基于 RMAP 协议标准确定。

写入操作用于 SpaceWire 网络中的发起端结点向目标端结点中指定子地址写入零字节或多字节的数据。写入命令包的包头和数据区均采用 CRC 校验机制, 目标结点可对数据包的正确性进行检查, 包头中的指示域指定命令数据包的属性和处理方式。写入命令包的格式如图 7 所示。用户数据部分视数据类型不同做了二级协议封装, 对于星上遥测数据保存, 采用的是 VCDU 格式。

目标结点逻辑地址	协议标识号 (0x01)	指示域	授权域
源端结点逻辑地址	事务标号 (MS)	事务标号 (LS)	扩展子地址 (0x00)
子地址 (MS) (0x00)	子地址 (0x00)	子地址	子地址 (LS)
数据长度 (MS)	数据长度	数据长度 (LS)	包头CRC
用户数据	用户数据	用户数据	用户数据
用户数据	用户数据
用户数据	数据CRC	EOP	

图 7 写入操作命令包格式
Fig. 7 Write command packet form

也就是说, 对于业务数据, 无论采用 I 类业务还是 II 类业务, 其用户数据装载的数据格式均为 VCDU, 而 VCDU 中的有效数据区也采用 M_PDU, 其基本单元也为空间包, 如上文中的图 3 所示。

在这种协议下, I 类业务可以尽可能地利用带宽, 提高传输速率; II 类业务则提供了带确认的数据传输方式。同时, 这两类业务中用户数据装载的 VCDU 中的虚拟信道可以用于区分星上的不同数据类型, 为用户提供灵活的传输服务的同时满足载荷数据传输的带宽要求。

2 软件设计

为了实现应用软件与操作系统和硬件平台以及链路协议与上层应用的解耦, 星载软件趋向于使用分层的软件架构, 常见的星载软件一般分为应用层、中间件层、操作系统和驱动层、物理层等。通过在层级之间进行端口封装, 一定程度上使得上层应用程序的开发接口更加标准化, 便于软件的多人协同开发和软件组装。从第 1 章的协议体系介绍中可以看出, 无论是星务管理数据涉及的遥控、遥测和 1553B 总线协议, 还是载荷数据传输涉及的 SpaceWire 总线通信协议, 其有效数据单元都采用了空间包格式, 这就使得星内标准化的数据传输成为可能。从硬件端口接收到的数据, 通过对数据链路协议的解析, 最终转化为空间包的形式向应用层传递, 而应用层产生了数据也以空间包的形式向下传递, 通过数据链路协议的进一步封装后, 转化为满足协议接口的数据格式, 进而输出给硬件端口, 如图 8 所示。

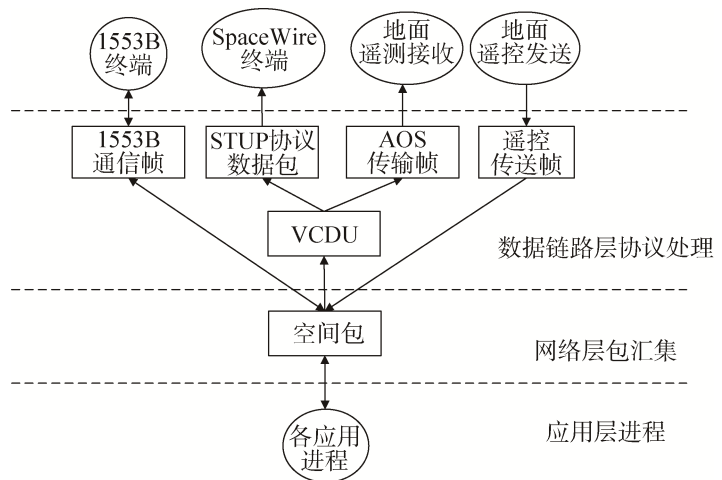


图 8 协议分层示意图
Fig. 8 Schematic diagram of protocol layer

2.1 软件框架设计

已有很多研究, 尝试利用标准协议对星载软件进行通信协议的分层^[4,8]。文献[4]建立了星内路由, 利用构件技术对数据链路层协议进行封装, 建立了分层的协议雏形。本软件的设计借鉴了这种分层理念, 对其中的数据链路协议进行了补充和完善, 进一步对各层级之间以及应用层各进程之间的数据交互格式进行标准化封装, 形成了基于标准协议的分层软件体系架构, 如图 9 所示。数据链路层协议均以构件的形式完成封装, 通过标准接口向上层提供服务。在这种体系接口下, 应用层各进程间、应用层与数据链路层之间, 均通过网络层路由, 以空间包为基本单位进行数据交互。

2.2 星内路由设计

在这种软件设计框架下, 网络层完成了空间包的汇聚和路由。星内所有需要传输和交互的空间包在此汇集和缓存, 然后以空间包包头中的 APID (应用过程标识)^[9] 字段作为标志进行路由, 进而将空间包分发至预定的目的地。可以看出, 空间包路由功能为星上数据流转的中枢, 高效可靠的路由信息设计对于提高软件可靠性非常重要。文献[4]中提出了一种路由信息设计方法, 但这种设置对于每个空间包只支持一个目的地, 且其服务类型与目的地编制方式之间存在数据冗余, 难以实现标准化配置的自动化生成。

为了实现标准化配置, 真正做到型号间无路由逻辑更改, 拓展可用的目的地, 设计了如下的路由信息:

```

struct RouteInfo
{
    unsigned16 apid; /*被路由源包的 APID*/
    SERVICETYPE service_type; /*下个链路要采用的服务类型*/
    unsigned8 addr_rt[ADDR_RT_NUM];/*总线目的地*/
    unsigned8 addr_ap[ADDR_AP_NUM];/*本地进程目的地*/
    unsigned8 addr_vc [ADDR_VC_NUM];/*是否下行信道传输*/
    unsigned8 addr_spw[ADDR_SPW_NUM];/*是否 SpaceWire 总线传输*/
};
    
```

对于每个空间包, 利用其 APID (应用过程标识) 字段对其目的地进行表征, 查询其在各个数据链路 (包括本地其他进程、1553B 总线、下行遥测或者 SpaceWire 网络) 上的地址。若发往本地其他进程, 则可以直接调用操作系统提供的进程间消息接口, 将空间包发送至目的地进程。若发往 1553B 总线终端, 则需进一步调用 1553B 协议处理构件, 将其封装至 1553B 通信帧中, 在下一个 1553B 通信时隙中发送。若发送往 SpaceWire 网络终端 (终端地址用其逻辑地址标识), 根据 STUP 协议要求对其进行进一步封装, 然后调用 SpaceWire 网络接口函数, 在端口空闲时进行发送, 网络路由器可以通过识别 STUP 包头中的逻辑地址, 将数据路由至相应目的地。若发往地面遥测, 则调用遥测构件将源包组织到相应的虚拟信道缓存中, 等待下行。当某个空间包需要向多个地址进行发送时, 可靠性设计保证只有在所有地址都已准备好的情况下再对此包进行发送, 保证了传输的可靠性。网络路由的设计也进行了标准化封装, 各型号间使用时, 只需要更改路由配置信息即可。

3 实现效果

利用 C 语言进行编程, 对上述的软件设计进行了实现。利用地面工控机、1553B 总线仿真卡、SpaceWire 终端仿真机等设备, 对软件功能和协议实现情况进行了验证。试验结果证明软件功能性能满

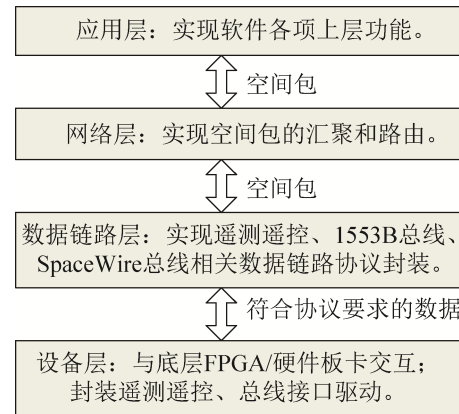


图 9 基于协议的分层软件架构
Fig. 9 Layered software construction based on standard protocol

足要求。通过对各层次的接口封装、对通信协议的封装,使得软件具有很好的可拓展性和可移植性。无论是硬件平台发生变化还是通信协议发生改变,只要更换其中的协议封装构件,同时保证接口的一致性即可。软件应用层之间无深度耦合关系,进程间接口统一,便于软件组装。同时,通过不断对成熟模块进行构件化封装,软件复用和产品化程度得到了有效提升。

4 结束语

本文介绍了一种基于 1553B 和 SpaceWire 组合网络的星载数据管理软件的协议体系。在对这些协议进行分析的基础上,提出使用空间包作为星上数据流转的最小单元,改进和实现了基于协议分层的星载数据管理软件。通过协议分层和接口封装,使得链路层协议实现与上层应用软件设计解耦,有效提升了软件的可拓展性和可移植性,促进了软件复用和产品化。

参考文献

- [1] PARKES Steve, ARMBRUSTER Philippe. SpaceWire: Spacecraft onboard data handling network[J]. Acta Astronautica, 2010, 66(1-2): 88-95.
- [2] ECSS. ECSS-E-ST-50-12A, Space Engineering-SpaceWire-Links, nodes, routers and networks[S]. 24 January 2003.
- [3] 朱维. 航天器综合电子系统技术研究[D]. 上海: 上海交通大学, 2013.
ZHU Wei. System design of spacecraft avionic [D]. Shanghai: Shanghai Jiao Tong University, 2013.
- [4] 张亚航, 袁珺, 于俊慧, 等. 一种基于星内路由的航天器数管软件框架设计[J]. 航天器工程, 2015, 24(6): 70-74.
ZHANG Yahang, YUAN Jun, YU Junhui, et al. Design of spacecraft OBDH software framework based on onboard route[J]. Spacecraft Engineering, 2015, 24(6): 70-74.
- [5] ESA. Packet telecommand standard[S]. ESA PSS-04-107, Issue2, 1992.
- [6] CCSDS. CCSDS 732.0-B-2 AOS Data Link Protocol. Recommendation for space data system standards[S]. Washington D.C.: CCSDS, 2006.
- [7] ECSS. ECSS-E-ST-50-13C, Space engineering - interface and communication protocol for MIL-STD-1553B data bus on board spacecraft[S]. Netherlands, 2008.
- [8] 何熊文, 孙勇. 一种卫星数管中心计算机软件的工程实现[J]. 航天器工程, 2007, 16(5): 47-53.
HE Xiongwen, SUN Yong. Engineering realization of software in central terminal unit of satellite data management system[J]. Spacecraft Engineering, 2007, 16(5): 47-53.
- [9] CCSDS. CCSDS 133.0-B-1 Space packet protocol. Recommendation for space data system standards[S]. Blue Book Issue 1. Washington D.C.: CCSDS, 2003.

[作者简介]

- 于俊慧 1988年生, 硕士, 高级工程师, 主要研究方向为空间数据系统和星载软件设计。
穆强 1979年生, 硕士, 高级工程师, 主要研究方向为空间数据系统和航天器数据管理。
牛跃华 1982年生, 硕士, 研究员, 主要研究方向为空间数据系统和航天器数据管理。
张亚航 1985年生, 硕士, 高级工程师, 主要研究方向为空间数据系统和空间信息安全。

(本文编辑: 潘三英)