

面向导航星座的星间传输协议测试系统设计

刘骥铭, 冯旭哲, 岳宇航

(国防科技大学 机电工程与自动化学院, 长沙 410073)

摘要: 星间网络协议是卫星通信的基础; 目前, 随着网络技术的发展, 网络协议的功能越来越复杂, 使得协议的测试越来越重要; 星间网络协议中处于核心地位的传输层协议是在测试中受到重点关注; 由于导航星座的条件限制, 现阶段难以对其使用的传输协议进行整网测试, 因此需要一个协议测试系统来满足测试需求; 在分析了导航星座通信网络特点与星间传输协议的基础上, 设计了传输协议测试系统; 该测试系统对星间网络协议的传输层协议进行实现, 并通过数据总线来模拟数颗卫星之间的数据通信; 测试结果表明, 测试系统能够满足星间传输协议的测试需求。

关键词: 导航星座; 星间传输协议; 协议测试

Design of Inter-satellite Transmission Protocol Testing System for Navigation Constellation

Liu Qiming, Feng Xuzhe, Yue Yuhang

(School of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, China)

Abstract: Inter-satellite network protocol is the basis of satellite communications. At present, the function of network protocol became more and more complex with the development of network technology, it makes the testing at protocol very important. The transport layer protocol which is at the core of inter-satellite network protocol receives great attention. Due to limited conditions of navigation constellation, currently, the entire network testing can not be done on transmission protocol. Therefore, a protocol testing system is needed to meet the test requirements. Based on analysis of navigation constellation communication network features and inter-satellite transmission protocol, a protocol testing system is designed. This system realize the transport layer protocol of inter-satellite network protocol and simulate communication between multiple satellites through the data bus. The test result shows that, the protocol testing is able to meet the test requirement of inter-satellite transport protocol.

Keywords: navigation constellation; inter-satellite transmission protocol; protocol testing

0 引言

星间网络协议是卫星通信的基础。传输层是网络协议中一个重要的协议层, 传输层协议处于网络协议的核心地位。其保证数据流的正常运行, 实现数据的可靠性保证、时效性保证、流量控制等功能。

传输协议在实现过程中, 可能会由于实现者对协议的不同理解或者实现过程中的非形式化因素, 导致协议实现的功能与协议规范的要求不一致^[1]。故在传输协议投入使用之前必须进行协议测试, 以确定协议实现是否与其协议规范相一致。

导航星座是按照要求分布在多个轨道的卫星群整体, 其各个卫星协同作业, 主要完成卫星导航任务。本文关注的是导航星座的星间传输协议, 其应用于导航卫星节点之间的数据传输。在导航星座的建设过程中, 在单颗卫星的设计研制阶段, 卫星测试阶段以及在卫星正式组网运行之前时, 如何对星上使用的传输协议进行测试是一个重要问题。由于卫星网络没有完整组网, 不具备传输协议的卫星整网测试条件。因此需要设计一个传输协议测试系统, 通过实现星间传输协议并模拟星间数据通信, 对卫星节点上的星间传输协议进行测试。

1 导航星座卫星网络

1.1 卫星通信网络

导航星座是一类较为复杂的卫星星座, 其精密度很高, 由多颗导航卫星组成。导航卫星之间通过无线链路来实现星座组网, 形成卫星通信网络。在导航星座中, 导航卫星提供的功能是一致的, 所以各个导航卫星在本质上可以看作是对等的。因此可以将导航星座卫星通信网络理解为一个具有多个对等节点的无线网络^[2]。

本文所针对的卫星通信网络接入模式为时分多址接入模式(TDMA), 在 TDMA 模式下, 时间被划分为等同的时隙, 卫星通信采用同一频率, 不同的扩频码。在每个时隙内, 每颗卫星都有各自不同的建链对象, 卫星之间拥有互不干扰的无线链路, 并且每颗卫星同时有且仅有一条链路^[3]。到了下一个时隙, 每颗卫星都会断开链路, 并按规律切换建链对象。某卫星轨道在不同时隙内无线链路的状态如图 1 所示。

时隙之上的单位称为超帧, 超帧包含了固定数量的时隙。超帧内的时隙都会按照一定规律变化。图 2 表示某颗卫星按某种规律的时隙变化图。

在这个例子中, 每个超帧有 3 个时隙, 在一段时间内, 每个超帧的内容一样, 可视为一个周期。该卫星在超帧内根据时隙的变化先后与 3 颗卫星分别建链。同样的, 其他卫星都有各自的时隙变化规律。

收稿日期: 2015-07-14; 修回日期: 2015-09-11。

作者简介: 刘骥铭(1991-), 男, 湖南衡阳人, 硕士研究生, 主要从事卫星通信方向的研究。

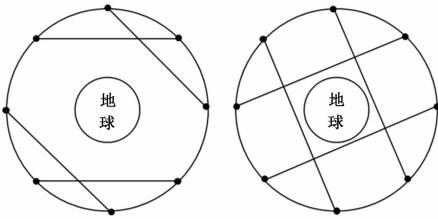


图 1 某轨道在不同时隙内的链路状态

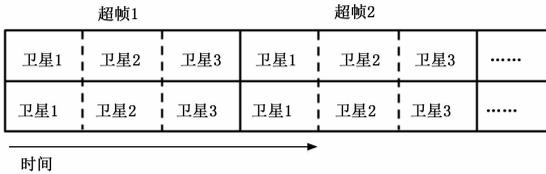


图 2 某颗卫星的时隙变化图

1.2 星间传输协议

传统地面互联网中, TCP/IP 协议是使用最广泛的网络协议。但是由于 TCP 协议是针对地面网络开发的传输协议。在具有误码率高、传输时延大、链路不对等特性的空间环境中应用效果不佳。因此关于空间环境的网络协议得到了大力的研究与发展。

在关于空间数据传输的协议中, 最著名的就是 CCSDS 提出的 SCPS 协议。SCPS 协议以 TCP/IP 协议栈为模型并基于 TCP/IP 协议进行了改进^[4], 目的是解决由空间网络的特性带来的传输可靠性保证方面的问题, 从而适用于空间通信环境。

SCPS 协议包含了一系列不同功能的协议, 图 3 说明了协议分层结构与分层功能。

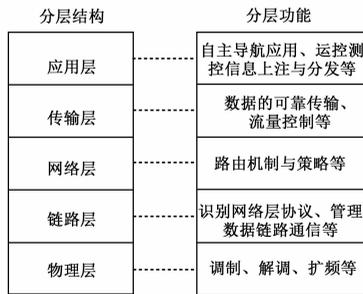


图 3 协议分层结构与分层功能

本文主要研究的是网络协议中处于核心地位的传输层协议, 其他层的协议不作过多考虑。传输层协议主要功能如下。

- 1) 数据的可靠传输: 包括数据的应答、数据的转发与重传、重传定时设置等^[2]。
- 2) 网络拥塞控制与流量控制: 在网络拥塞时控制数据传输, 维护网络性能。

2 测试系统总体设计

2.1 测试系统功能

在分析了卫星通信网络特点与传输协议的基础上, 设计一个协议测试系统来验证协议实现是否能按照协议标准正确实现数据传输。由于传输层协议不涉及信息传输的介质与接口问题, 所以测试系统采用数据总线来模拟星间的无线链路进行数据的传输。

因此, 测试系统设计为在地面上使用, 能够模拟导航星座的组网, 利用数据总线传输模拟卫星节点间的数据传输。其主要实现以下功能:

- 1) 星间数据生成, 测试系统根据传输协议数据格式, 产生测试数据;
- 2) 星间数据传输, 测试系统可以通过总线传输测试数据;
- 3) 星间数据处理, 测试系统实现星间传输协议, 对测试数据进行处理, 依据协议返回相应的数据;
- 4) 星间数据保存, 对各类数据进行保存, 用于事后对数据进行观察分析;
- 5) 虚拟节点模拟, 模拟多个等效通信节点。

2.2 测试系统设计

测试系统的目的是对星间传输协议进行测试, 而实际上, 传输协议是以软件实现的形式存在与卫星的数据处理单元中。因此, 本文选取的实际测试连接对象是一个星间传输数据处理单元。该数据处理单元作为一个被测通信节点, 其实现了星间传输协议, 并配备有处理器、星间数据传输总线芯片与接口、数据存储器等, 具备数据传输、数据处理、数据存储等功能。

在 TDMA 模式下, 由于每颗卫星在一个时隙内有且仅有一条链路, 那么只需要与被测对象建立一条通信链路。在时隙改变时, 只要改变建链的节点即可, 对于传输层来说, 通信链路可以保持不变。考虑到测试系统具有模拟多个通信节点的功能, 因此, 对于被测节点来说, 只需要采用一套设备就可以实现模拟导航星座网络的数据传输, 为被测节点提供一种完备的传输协议测试手段。

根据实际条件与应用需求搭建了测试平台。测试系统与被测对象通过数据总线进行通信。各类数据经过数据总线实时传至 CPU, CPU 运行测试软件对数据进行实时处理。用户可以通过外接设备实现人机交互。图 4 是测试系统的测试连接图。

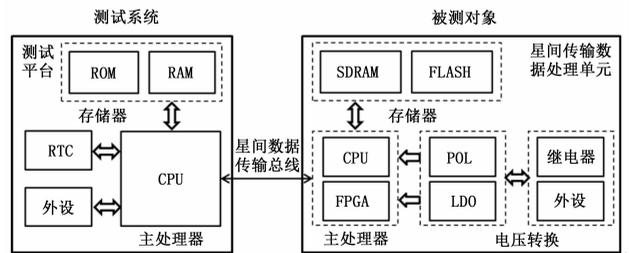


图 4 测试连接图

测试系统模拟出多个等效通信节点, 测试对象作为一个被测节点, 它们均使用同样的传输协议。整个系统初始化之后, 处于等待用户指令的状态。用户根据测试用例, 在测试系统应用程序中配置参数。配置完成后, 测试系统根据输入的参数生成测试任务, 并通过总线将测试任务告知被测对象, 最后打开测试数据文件, 开始测试任务。测试进行时, 测试系统会按照 TDMA 模式工作, 并按照测试任务进行数据的收发。直到测试数据发送完毕或者用户发送停止指令, 测试结束。

3 测试系统软件设计

3.1 软件主要功能

为满足测试需求, 测试系统软件应具有下列功能:

- 1) 能够配置测试参数, 能够载入链路规划表, 读取并解析规划表中所包含的通信节点信息、传输速率信息、收发状态

信息，并由这些信息建立测试场景；

2) 读取源数据，并根据规划表的解析结果，产生测试数据，驱动数据总线进行任务信息与测试数据的传输；

3) 为每一个虚拟通信节点分配独立的数据缓存，数据在发送或者接收后，能够按照通信节点、收发时间、数据类型等信息进行分类保存；

4) 分通信节点统计显示各类业务数据的数据量、应答计数、转发计数、重传计数、错误数据种类与错误数据计数；

5) 产生系统时间，作为测试系统的时间基准。用户能够自行设置系统时间。

3.2 软件结构设计

测试系统应用程序在 Visual C++ 环境下开发，采用了模块化设计，由用户界面、数据存储模块、数据生成模块、链路规划模块、传输协议实现模块、总线通信模块与通信节点模拟模块组成。各模块协同工作以实现软件主要功能。图 5 是软件结构示意图。

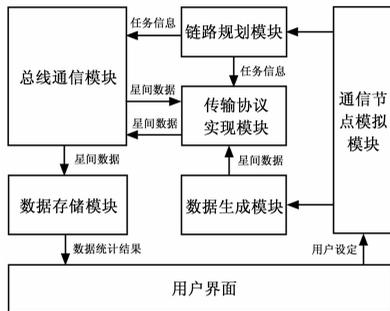


图 5 软件结构示意图

其中用户界面包括配置界面和统计界面。用户通过配置界面配置参数以实现对各个模块直接或间接的控制，能够实现不同的测试场景，执行不同的测试用例，提高了测试系统的测试覆盖率。统计界面用于显示得到各类数据的统计结果。

考虑到卫星网络处于 TDMA 模式下，程序设计为以时间触发中断的结构。测试系统应用程序根据系统时间进入各个中断处理程序，使得通信过程严格按照时间进行。这在确保了良好的软件结构与程序可控性的同时简化了编程与调试。

3.3 软件设计流程

测试系统软件的工作流程如图 6 所示。

测试系统软件初始化首先完成了数据总线设备的初始化，并产生了系统时间，作为系统的时间基准。整个通信过程严格按照时间进行，这要求时间基准必须精确可靠，以确保数据传输的实时性与可靠性。

然后在用户界面中载入规划表，规划表是用户根据测试用例在测试前制作。测试系统对规划表进行解析，并结合配置参数，生成任务信息。任务信息中包含了通信节点号，通信速率，数据类型、数据量以及收发状态等信息。由这些信息，测试系统可以建立测试场景。同时将源数据按照传输协议帧格式加上帧头信息产生测试数据。测试系统根据当前使用的任务信息切换至当前使用的通信节点、传输速率与收发状态并将任务信息告知被测对象。被测对象同样切换至对应状态后才开始测试数据的传输。图 7 为规划表的格式。

被测对象收到测试数据后进行数据处理。首先对数据进行误码校验，丢弃误码并作记录。然后解析数据帧头信息，以确

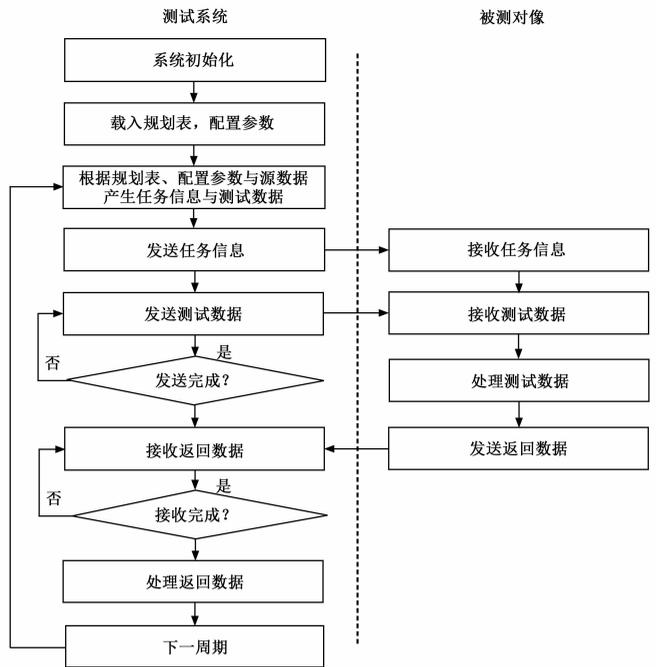


图 6 软件工作流程图

	A	B	C	D
1	目标卫星节点	是否卫星节点	收发状态	速率状态
2	0xc1	是	收	1
3	0xc1	是	发	1
4	0xc2	是	收	2
5

图 7 规划表格式

定数据的类型，判断是否为应答数据、是否需要应答、是否为转发数据，并将数据存入对应的缓存区。最后按照传输协议中的应答机制、重传机制、转发机制对数据做进一步处理，返回数据。

测试系统接收返回数据完成后进行数据解析与处理，并将此周期内双方的数据传输结果记录并显示在用户界面的统计界面。然后进入下一周期，重复上述过程。直到测试数据发送完毕或者用户发送停止指令，测试结束。

4 测试系统测试

4.1 测试用例说明

为了对测试系统进行测试，设计了一个典型的测试用例。

测试系统模拟 0xc1—0xc6 共 6 个通信节点，被测对象作为通信节点 0xe3。然后制作规划表，设定六个节点与被测节点按时间节拍轮流通信，将通信速率设为 50 帧/周期。

发送的测试数据有遥控数据、运控数据与遥测数据，其中遥控数据与运控数据需要应答。为 0xc1、0xc2 节点各配置 1680 帧遥控数据，将目的地址设为 0xe3，并在 0xc2 节点的数据中使用错误的 CRC；为 0xc3、0xc4 节点各配置 1400 帧运控数据，将目的地址设为 0xe3，并在 0xc4 节点的数据中使用错误的 CRC；为 0xc5、0xc6 节点配置 840 帧遥测数据，将 0xc5 节点数据的目的地址设为 0xc1，0xc6 节点数据的目的地址设为 0xc2。三类数据的帧速率均设为 10 帧/周期。

(下转第 60 页)

