电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS

论文题目 **面向天地互联的 TCP/IP 与 CCSDS 协议转换设计与仿真研究**

学科专业	航空宇航科学与技术
学　　号	**201921100120**
作者姓名	姚 欢
指导教师	蒋大钢　副教授
学　　院	航空航天学院

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

分类号 V556.8 密级 公开

UDC 注1 629.7

学 位 论 文

面向天地互联的 TCP/IP 与 CCSDS 协议转换设计与仿真研究

姚 欢

指导教师 蒋大钢 副教授

电子科技大学 成 都

李 雪 副教授

电子科技大学 成 都

申请学位级别 硕士 学科专业 航空宇航科学与技术

提交论文日期 2022 年 4 月 15 日 论文答辩日期 2022 年 5 月 26 日

学位授予单位和日期 电子科技大学 2022 年 6 月

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Design and Simulation of TCP/IP and CCSDS Protocol Conversion for Space-earth Interconnection

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Discipline **Aeronautical and Astronautical Science and Technology**

Student ID **201921100120**

Author **Yao Huan**

Supervisor **Associate Prof.Jiang Dagang**

Associate Prof.LiXue

School **School of Aeronautics and Astronautics**

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：魏欢　　日期：2022年5月26日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：魏欢　　导师签名：

日期：2022年5月26日

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

摘　要

随着我国深空探测活动的不断深入，天地互联互通成为保障航天任务顺利开展的关键环节。目前，地面网络普遍采用 TCP/IP 协议，空间网络一般使用 CCSDS 标准协议，两大协议体系彼此独立。

为了在空间网络中使用成熟的地面通信协议，本论文开展了 IP over CCSDS 的协议转换方案研究，即在空间链路中使用 CCSDS AOS 协议来传输 IP 数据包，并从天地互联网络的链路层、网络层和传输层开展详细设计。

在链路层方面，针对 AOS 协议承载 IPv4 数据包的问题，使用了 AOS 封装来解决，设计了 AOS 传输帧的结构；对于天地数据交互时 IP 数据包的长度问题，基于数据包的实际长度采取切割和拼接的操作来解决。此外，为了解决紧急信息在天地互联网络中的快速传输问题，在链路层的 Mac 协议中引入了数据优先级的发送方式。

在网络层方面，针对航天器运动导致的链路中断、无法正常发送数据包的问题，使用了移动 IP 技术，弥补了常规 IP 技术在空间中的不足，减少了链路中断时间。

在传输层方面，针对空间链路高误码率等导致的 TCP 协议性能下降问题，使用 SCPS-TP 协议来代替 TCP 协议，该协议在 TCP 协议首部选项部分扩展，在数据重传机制通过 SACK、SNACK 以及在拥塞控制算法上使用 Vegas 等手段解决了常规 TCP 协议在空间网络中吞吐量下降、丢包率提高、拥塞窗口较小问题。

本文使用 OMNeT++创建了地月网络仿真，对上述研究进行了全面验证。结果表明：从链路层来看，发送端未封装 IP 数据包和接收端经过 AOS 封装和解析之后的数据包一致，验证了 AOS 协议的可行性以及对于数据包处理过程的正确性，同时使用了数据优先级的仿真结果表明不同数据源的碰撞次数减少 50%以上，同时使用该方式后传输时延也较之前减少了约 1s；从网络层来看，使用了移动 IP 技术后，解决了链路中断无法发送数据包的问题，链路中断时间也减少了约 3s；从传输层来看，SCPS-TP 协议在吞吐量上比 TCP 协议提高了约 150bps，拥塞窗口比 TCP 协议提升了 15~20MSS，以上结果说明在空间网络中 SCPS-TP 协议性能优于 TCP 协议，更加适合 IP over CCSDS 的协议转换模式。

本文天地互联转换协议设计对于我国开展近地空间和地月的信息具有一定借鉴意义，为未来对空间通信系统的探究给予了理论和技术上的帮助。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

关键词：TCP/IP ，CCSDS，SCPS-TP，移动 IP，AOS，IP over CCSDS

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

ABSTRACT

As China's deep space exploration activities continue to progress, the interconnection between heaven and earth is a key link to ensure the smooth implementation of space missions. At present, the ground network generally uses TCP/IP protocol, while the space network generally uses CCSDS standard protocol, and the two protocol systems are independent of each other.

In order to use mature terrestrial communication protocols in space networks, this thesis carries out research on the protocol conversion scheme of IP over CCSDS, i.e. using the CCSDS AOS protocol to transmit IP packets in space links, and carries out detailed design from the link layer, network layer and transport layer of the interconnected network between heaven and earth.

In the link layer, for the problem of AOS protocol carrying IPv4 packets, AOS encapsulation is used to solve the problem, and the structure of AOS transmission frames is designed; for the problem of the length of IP packets during the data interaction between heaven and earth, the cut and splice operation is adopted to solve the problem based on the actual length of the packets. In addition, in order to solve the problem of fast transmission of emergency information in the heaven and earth interconnection network, a data priority sending method is introduced in the Mac protocol of the link layer.

In the network layer, mobile IP technology is used to compensate for the lack of conventional IP technology in space and to reduce link interruption times, in response to the problem of link interruptions caused by spacecraft movement and the inability to send packets properly.

In the transport layer, in response to the performance degradation of the TCP protocol caused by the high BER of the space link, the SCPS-TP protocol is used to replace the TCP protocol, which is partially extended in the TCP protocol prefix option, and solves the problems of throughput degradation, packet loss rate increase and smaller congestion window of the conventional TCP protocol in the space network by means of SACK and SNACK in the data retransmission mechanism and the use of Vegas in the congestion control algorithm.

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

In this thesis, a simulation of the Earth-Moon network was created using OMNeT++ to fully validate the above study. The results show that: from the link layer, the unencapsulated IP packets on the sender side and the packets on the receiver side after AOS encapsulation and parsing are identical, verifying the feasibility of the AOS protocol and the correctness of the packet processing process, while the simulation results using data priority show that the number of collisions between different data sources is reduced by more than 50%, and the transmission delay is reduced by more than 2s after using this method. From the network layer, the use of mobile IP technology solves the problem of link interruption inability to send packets, and the link interruption time is also reduced by about 3s; from the transport layer, SCPS-TP protocol improves about 150bps in throughput and 15-20MSS in congestion window than TCP protocol, the above results show that SCPS-TP protocol outperforms TCP protocol in space network. The above results show that the SCPS-TP protocol outperforms the TCP protocol in spatial networks and is more suitable for the protocol conversion mode of IP over CCSDS.

This thesis has some implications for the design of interconnection protocols between heaven and earth for the development of near-Earth space and Earth-Moon communication and data interaction in China, and gives theoretical and technical help for future exploration of space communication systems.

Keywords: TCP/IP,CCSDS, SCPS-TP, Mobility IP, AOS,IP over CCSDS

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

目 录

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第一章 绪 论

1.1 论文的背景与意义

近年来，月球探测一直各国航天发展的重要方向，2017 年美国自 1969 年登月后第二次宣布重返月球的计划，将于 2024 年发射载人飞船进行登月；欧盟于 2019 年开始逐步开展月球采矿的技术研究，预计在 2025 年开始进行相关探测任务。中国作为航天大国，月球探测活动也在按计划进行，2004 年我国正式立项探月项目，并于 2007 年开始发射“嫦娥”系列月球探测器，2020 年随着嫦娥 5 号成功将 2kg 月球土壤带回，标志着我国“绕”、“落”、“回”的三步走战略实现。

随着我国各种航天任务不断增多，目前的航天器和地面站的测控通信技术已经难以满足日益增长的需求，为航天器提供高效良好的服务。所以天地互联的通信体系的建设就显得极其重要。目前以 TCP/IP 协议为代表地面通信网络相当成熟，已经成为地面通信网络的标准协议，深空通信的发展趋势是利用地面网络的优势，实现天地互联互通。但是地面通信和深空通信是两个相对独立的协议体系，由于结构和协议的不同，给天地互联信息传输带来了不少的困难。为了解决这个问题，我国在十三五规划中提出了《民用航天“十三五”技术预先研究第二批项目指南》，该指南明确了天地互联的重点方向：不再局限于近地空间，开启月球等距离更加遥远的深空探测，同时建立一个基于中继卫星的信息交互的标准技术系统，用来替代原来的测控体制，利用地面成熟发展的网络技术优势，实现天地互联的通信协议转换，为大规模商用和民用提供接入空间网络以及获取信息的服务[1]。

本文基于以上背景开展，依托于《深空通信与信息公共服务体系研究》项目，仔细研究了国际上常用的空间通信协议 CCSDS 协议[2]，并根据项目的实际需求提出了 IP over CCSDS 的天地互联的协议转换的设计方案。项目中提出了地月空间网络的模型，在地月传输中存在着误码率高、时延大、前向和反向链路不对称的问题、而且地月中继卫星有着高速移动的特性，可能会导致链路长时间中断的问题、同时空间链路与地面链路不同，基本为无线链路，存在地面链路协议无法很好地应用在空间网络中的问题、最后在进行数据传输时，不同的数据具有不用的优先级，可能改变协议的工作方式和处理结果。针对以上问题，本文在构建天地互联的网络时，对传输层进行一定的改进，提升了数据的传输效率；在网络层使用了移动 IP 技术，缩短了链路中断的时间，减少了传输层的重传次数；在链路层使用了 CCSDS AOS 协议，该协议提供了封装 IP 数据包的功能，使地面成熟的 IP 协议能够和空间网络协议融合为一体，实现了天地互联的协议转换；本文根据 IP over CCSDS 进行仿真研究，在地月空间网络中构建出天地互联的协议转换设计，使用

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

数据优先级的发送模式使协议转换系统的功能更加完善，为地面网络与空间网络进行端到端的数据传输以及确保各项航天任务的完成提供了技术支持。

1.2 国内外研究与现状

1.2.1 国外研究及发展现状

CCSDS 协议委员会（Consultative Committee for Space Data System）在 1982 年由美国牵头成立，主要成员为美国和欧盟等国家和地区的航天部门[3]。在成立之后的一段时间里，CCSDS 首先推出了常规在轨系统（Common Orbiting System，COS）和分包遥测（Telemetry，TM）、分包遥控（Telecommand，TC）的技术建议书，前者是应用于近地空间网络和深空探测任务，后者使用了虚拟信道复用的技术，即实际空间信道在逻辑上被分为很多个虚拟信道，提高空间的信道利用率，之后 CCSDS 又陆续推出了包括射频、航天器轨迹、无线电外测、数据传输单元标准等[4]。

1986 年 CCSDS 推出了高级在轨系统（Advanced Orbiting Systems，AOS），AOS 是对 COS 的拓展，它的应用场景是在载人空间站、空间实验室以及肩负特定复杂任务的航天器[5]，相比于一般的航天任务，需要用到 AOS 协议的航天任务执行了更加严格的数据标准。

从上个世纪九十年代开始，CCSDS 对于之前发布的 TM、TC 和 AOS 协议进行修补和扩展，形成了一个层次清晰的空间链路层协议模型，称之为空间包协议（Space Packet Protocol，SPP）。为了适应空间环境的特性，如空间链路经常中断、长延迟、误码率高、空间链路不对称等一系列问题，1999 年 CCSDS 发布了空间通信协议规范（Space Communication Protocol Specification，SCPS）协议族[6]，主要包括 4 个协议，分别是关于网络、安全、传输和文件方面的有关协议和规定，其中本文主要研究的内容是空间传输协议（SCPS Transmission Protocol，SCPS-TP）[7]，上述四个协议第一个对应网络层，第二个和第三个协议对应传输层，最后一个对应应用层。2003 年针对空间网络的链路层，CCSDS 制定发布了 AOS 空间链路层协议，该协议建议所有的空间传输数据均采用统一的 CCSDS 包，可以是同步也可以是异步，让不同信道、不同格式的数据使用同一种打包方式，形成一个个虚拟信道数据传输单元，可以共享同一个物理信道。2012 年 CCSDS 发布了 IP over CCSDS Space Links 蓝皮书，并提出了近距-1 空间链路协议，2018 年 CCSDS 发布了关于空间数据系统标准的建议草案，该标准的提出了是为了规定空间数据链路的安全性，它定义通过了操作空间链路所需要的密钥管理、安全管理管理以及数

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

据结构。

在应用方面，自 CCSDS SCPS 被提出以来，美国航天局的实验室一直将其作为标准协议执行每一次航天任务，2000 年美国科技公司也开始推进 SCPS 协议的商业化进程，推出了使用该协议的软件 SkipWare，在军队得到一定规模的使用。美国 MITRE 公司利用地面网络的扩展性和稳健性，针对天地互联的特点，成功实现了 CCSDS SCPS 协议与 TCP 协议的转换，弥补了 TCP 协议在空间通信的不足。除此之外，加拿大 Xiphos 公司推出了采用超文本传输协议 1.1（Hyper Text Transfer Protocol，HTTP），支持快速重传、报头压缩，快速重传的功能的 SCPS-TP 协议的网管。但是，以上有关 SCPS 协议都是根据空间通信环境的特殊性而开发的，无法和地面网络的协议直接进行连接使用，所以有必要对地面网络和空间网络的协议转换方面研究。

为了实现了天地一体化通信，美国 NASA 开展了 OMNI 项目的研究，该项目目的在于将地面成熟 TCP/IP 协议使用到空间中的卫星、登月飞船、空间站等航天器中，虽然该项目认为空间环境的影响不足以妨碍 IP 协议的使用，这一点与 CCSDS 的指导思想是不一致的，但是该项目也得到了不错的实验结果。

1.2.2 国内研究及发展现状

中国从上个世纪九十年代开始关注 CCSDS 协议的发展，同时也对 CCSDS 协议进行研究。1996 年中国专门建立了一个 CCSDS 小组，研究 CCSDS 协议的最新进展，进行了“CCSDS 体制空间数据系统”的研究，规划了未来我国自主空间站以及中继卫星使用 AOS 体制；1999 年，中国科学院成立了“天基综合信息网关技术预先研究室”项目组，对 CCSDS 协议框架进行研究，并对 CCSDS 信息传输协议进行仿真演示，取得了一定的阶段性成果，同年五月，在理论研究取得成果的前提下，我国发射实践五号卫星，对于 AOS 系统进行了验证；在 2008 年我国发射“天链一号”数据中继卫星[8]，使得神州七号载人飞船的测控通信能力加强，这颗中继卫星使用了 CCSDS 链路层的协议，2008 年 6 月，中国加入 CCSDS 组织，正式成为该组织的第十一个成员，这实际上说明了 CCSDS 协议将成为中国进行天地互联信息传输的标准协议；2013 年工信部为天地互联信息传输设立了一个学术交流的平台；2015 年 6 月，中国工程院提出了对于天地一体化的初步想法和设计；2018 年国家十三五规划中明确提出了对于构建深空通信服务体系的要求，在要求中我国开始将对 CCSDS 协议研究的重点放在了天地互联一体化，这将是未来空间网络的发展趋势。

从上面的发展态势来看，未来的航天任务必然会越来越复杂。低轨卫星的数

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

量不断增多，星间链路的拓扑结构也随之更加复杂；空间站内部的网络规模越来越大，点对点的数据传输很难应对这种局面，所以建立天地互联的一体化网络协议是迫在眉睫，不容忽视的。除此之外，空间任务也需要天地互联的网络协议，而且鉴于空间环境较为恶劣，比如链路的非对称性以及高误码率等，航天器使用的通信协议就不能和地面 TCP/IP 协议簇的完全一致，因此就需要进行协议转换设计来解决 CCSDS 协议和 TCP/IP 协议不兼容的问题，同时协议转换也能带来好处，比如和地面网络更加兼容，利用现有成熟的技术优势降低了空间组网的成本。

CCSDS 协议在我国航天测控中只是应用在小部分的卫星和飞船中，大部分航天器依旧使用 PCM 测控体制，并且缺乏 IP 技术如何在地月通信中的问题研究、SCPS 协议和地面网络协议连接的问题研究，而且目前国内的研究很少关注重要数据在天地互联网络的快速发送的问题。

1.3 本文主要研究内容

本文的主要内容是基于天地互联互通的通信需求，以 TCP/IP 协议与 CCSDS 协议之间的转换为研究目标，在实现协议转换时参考 CCSDS 建议，提出了 IP oover CCSDS 的协议转换方式。在 OMNeT++网络仿真平台上，构建出地月空间网络的模型，在地月空间网络中验证了 AOS 协议，对空间中卫星等高速移动的节点，本文提出了移动 IP 的解决方案，并在平台中进行了仿真实现，空间通信中的高误码率、长时延等对 TCP 协议产生了严重的影响，本文使用了 SCPS-TP 协议来进行改善，并在 OMNeT++中对两者进行对比仿真，从仿真结果来看，SCPS-TP 协议在吞吐量、丢包率等性能指标上都强于 TCP 协议，最后在仿真场景中实现了高优先级数据的快速传输，改善了网络性能。

本文的创新性主要为以下 2 个方面：

（1）在 OMNeT++平台上对于 IP over CCSDS 设计出了通用性强且具有可扩展性的协议转换模型，并且对该模型进行仿真实现，协议的性能和指标达到了预期的目标。

（2）在 IP over CCSDS 的协议转换方式中加入了数据优先级的发送模式，解决了重要数据需要快速传输的问题。

1.4 本论文的结构安排

本文的章节结构安排如下：

第一章主要叙述了课题的研究背景和意义，展现了关于 CCSDS 协议的国内外发展历史和研究现状，说明了之前研究内容的不足之处，由此引出本文的研究重

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

点和内容，在本节的最后对本文的结构进行了说明。

第二章根据协议转换涉及的理论知识进行介绍，主要包括了 TCP 协议首部、关键机制和拥塞控制算法等，对 CCSDS 协议和 AOS 协议进行了说明，对 CCSDS 物理层和数据安全做了补充。

第三章提出了基于 IP over CCSDS 的天地互联的协议转换方式，构建了协议转换模型，对空地数据交互进行了说明，AOS 帧结构进行了设计，对 IP 数据包的处理进行了说明，之后由于常规路由在空间环境中的不适应，本文提出了移动 IP 的技术方案，再之后提出了 TCP 协议在空间通信存在的一些问题并给出了解决方案 SCPS-TP 协议，设计了 SCPS-TP 协议的一些关键机制，最后对数据优先级的划分和发送进行了说明。

第四章首先介绍了本文使用的仿真软件，设计了仿真场景，验证了 AOS 协议和移动 IP 技术，对 TCP 协议和 SCPS-TP 协议进行了仿真比较，仿真研究了使用数据优先级的发送方式。

第五章对全文的研究工作做了总结，对一些可以改善的部分进行了补充说明。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第二章 相关理论知识

2.1 TCP/IP 协议介绍

2.1.1 概述

TCP/IP 协议族是广泛应用于地面因特网的标准通信协议，包含在开放互联模型（Open System Interconnection Reference Model，OSI）的七层网络模型中[9]，其中传输控制协议（Transmission Control Protocol，TCP）对应于运输层或是传输层，而网际互联协议（Internet Protocol，IP）对应于网络层。IP 协议主要负责将传输层的数据进行封装，根据 IP 地址，把数据包传送对应的主机，但是这种网络协议并没有包含可靠传输的机制，无法保证数据传输的质量，在使用 IP 协议传输数据的过程中会有很大概率发生数据丢包、数据重复以及数据错位的情况。针对 IP 协议的设计局限性使用 TCP 协议来解决[10]，使用 TCP 协议的通信节点在数据交互之前会相互发送握手信号，来创建连接的关系，在主机的套接字之间传输数据，在交互完毕之后发送消息来删除连接关系，并通过拥塞控制等一系列手段改善通信状况。图 2-1 是分层的 TCP/IP 协议族与 OSI 体系结构比较，前者较后者减少了表示层、链路层等，但是包含的通信协议没有改变，只是表达更加简洁。

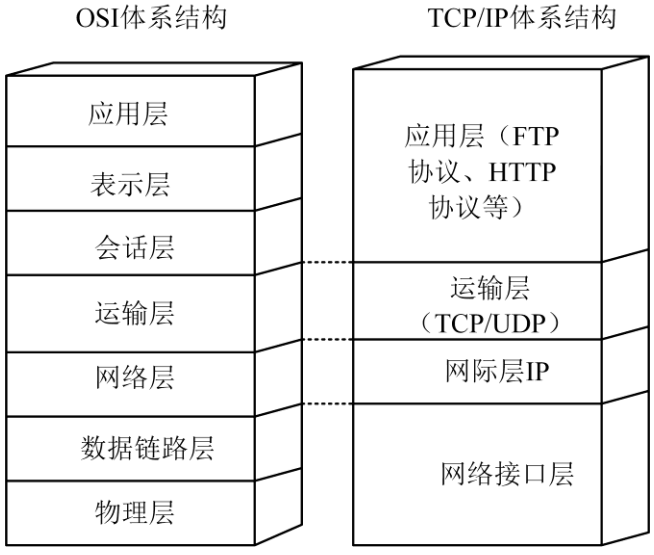

图 2-1 TCP/IP 四层模型与 OSI 七层模型

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.1.2 TCP 首部

TCP 协议传送的数据是一个个数据段（segment）,一个 TCP 分段是由数据和首部构成的,首部相当于 TCP 协议给与数据的标识,下面本文简单来讨论一下 TCP 首部的结构以及作用，如图 2-2 所示为 TCP 首部的格式。TCP 段是适合发送的数据形式，实际上是由字节流组成的。

图 2-2 TCP 首部格式

16 位端口号：源端口号表明数据从哪台主机发过去的，目的端口号表明数据要发送到哪台主机。

32 位序号：TCP 协议在传输段时会给段中的每一个字节都分配一个 32 位的编号，比如一个 TCP 数据分段的序号是 1，这个分段共有 100 个字节，那么本分段的第一个字节序号为 1，最后一个字节序号为 100。

32 位确认号：表示对于收到的字节的确认，为接收到的字节序号的下一个。比如一台主机收到了另一台主机序号为 550 的字节,那么需要给出的确认号为 551，表示着序号 551 之前的字节已经收到了。如果确认号为 N，表示序号为 N-1 的字节数据已经正确收到了。

4 位头部长度：表示首部的长度，一般如果不使用选项部分，则首部长度为 20 字节，首部长度最长为 60 字节（（2^4-1）*4）。

紧急 URG：当 URG 为 1 时表示此时需要传输的数据比较紧急，可以不按照原本的字节流顺序发送。

确认 ACK：ACK=1 确认号才有效，为 0 则表示确认号无效。

推送 PSH：PSH=1 时表示通信的一方希望另一方能够对自己的命令快速响应，此时接收到命令的一方会立马创建一个 TCP 分段进行发送，而不用等到缓存队列满了再发出，一般很少使用该指令。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

复位 RST：表示 TCP 因为一些故障出现差错，需要断开连接，重新建立连接，此时 RST 置为 1。

同步 SYN：这是一个通信双方同意建立连接的握手信号，在 TCP 连接的前 2 个报文中需要建立连接的双方都会将 SYN 置为 1。

16 位窗口：这个值表明通信的一方的缓存区还能接收的数据量，以字节为单位，最大值为 65535 byte（2^{16}-1）。

16 位校验和：主要用来检测 TCP 分段是否发生了数据错位的情况。如果校验和不正确，那么接收数据的一端将不对出错的分段确认收到，并丢弃该分段。

16 位紧急指针：用来表示分段中紧急数据的字节大小。

选项：用来扩展 TCP 协议的内容和功能，比如可在选项部分添加选择确认收到的机制（SACK），选项部分的最大值为 40 字节[11]。

2.1.3 TCP 协议关键机制

（1）TCP 的建立连接和释放连接

当 2 个应用程序通过 TCP 进行通信时，一个应用程序是客户端，另一个则是服务器。服务器会开始连接之前开启控制机制，之后客户端会向服务器的端口和 IP 地址发送请求连接的 TCP 段。在服务器和客户端之间的连接建立之后，它们之间就可以进行全双工的通信。如果两者之间不再有数据需要发送，那么应用程序就会关闭端口，但是应用程序还是可以接收来自其他连接的数据。当通信双方都关闭端口时，此时连接就会断开。

当打开连接时，客户端会发送一个同步字段（Synchronize sequence number，SYN）为 1 的 TCP 段，此时在 TCP 段首部的 32 位序号为此次连接传输数据的第一个序号，之后客户端处于 SYN-SENT（同步已发送）阶段当中。这个序列号不能是常量，因为这样一来，此连接中可能会错误的接收到之前版本的连接中的数据段，故大多数 TCP 实现根据系统时钟选择初始序列号。随后服务器收到请求连接的分段，会给客户端发送一个确认分段（Acknowledgment，ACK）并进入 SYN-RCVD（同步接收）阶段，该分段会携带自己的序列号和 SYN，客户端在收到这个响应的 ACK 段后，会给服务器发送一个响应分段并处于 ESTABLISHED 阶段（建立连接），服务器在收到这个分段后也会进入 ESTABLISHED 阶段，此时通信双方的形成了连接的关系，这个过程称为三次握手。

当断开连接时，客户端和服务器都可以选择主动关闭，这里以客户端先执行关闭程序为例。客户端会发送一个 FIN 段告知服务器，客户端应用程序已经没有需要发送的数据了，之后便处于 FIN-WAIT-1 阶段（第一阶段的等候状态），服务

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

器会响应客户端发送 ACK 段，之后就处于 CLOSE-WAIT 阶段（等待关闭），客户端从第一阶段的等候转向第二阶段的等候即 FIN-WAIT-2 阶段，一段时间后服务器也会向客户端发送 FIN 段，用于通知客户端服务器应用程序已经没有数据需要发送了，之后便处于 LAST-ACK 阶段（最后确认），等待客户端的确认信息，客户端在收到此分段后发出 ACK 分段，从第二阶段的等候转变到 TIME-WAIT 阶段，计时器从 2 MSL 开始倒计时，计时器归零后从等待阶段转变到 CLOSED 阶段，而服务器在收到确认信息之后就会进入 CLOSED 阶段。这个过程称为 TCP 四次挥手的过程。如图 2-3 所示。

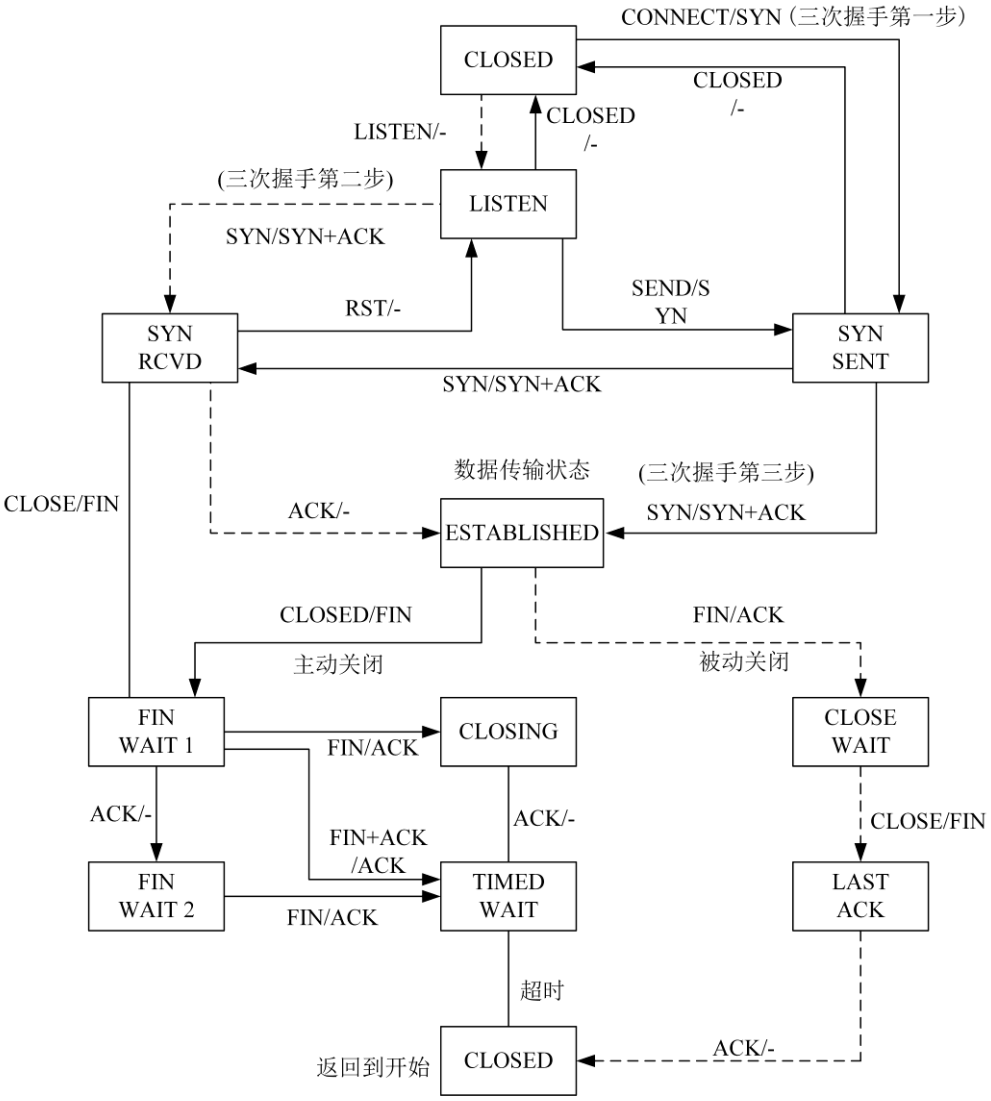

图 2-3 TCP 连接管理状态机

（2）差错控制

①持久性计时器

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

当接收端缓冲区容量耗尽时，它会在 ACK 分段中把窗口字段的大小置为 0 来停止发送端发送数据。当数据进入了应用程序当中，它会重复发送最后一个 ACK 分段，并在分段中将窗口字段更新，达到重新开始发送数据的目的。如果这个 ACK 分段丢失了，发送端无法得知数据是否已经达到了接收端。为了避免这种情况，发送端在接收到窗口字段大小为 0 的分段时，会开启一个持久性计时器，如果计时器在窗口增加之前失效，发送端将会发送一个带有 1 字节数据的探测段，在接收端的 ACK 分段中会有当前的窗口大小的信息。

②重传机制

当发送端发送一个 TCP 分段后，它会开启一个重传计时器。如果 ACK 分段在计时器失效之前达到，那么计时器就会停止计时，否则发送端将会认定该分段丢失并重发。如果计时器的设定时间（Retransmission Timeout，RTO）过长，那么丢失的分段会引起很高的延迟；如果时间太短，那么接收端就会收到许多无用重复的分段。为了使 TCP 传输的最优化，计时器的时间必须是动态的调整。Jacobson 提出了一个比较实用的公式：

$$RTO = RTT + 4 * D \tag{2-1}$$

其中 RTT 和 D 是测量的平滑往返时间（Roundtrip Time）和平滑平均偏差，它们的初始值都设为 0，并根据以下公式在每次接收到 ACK 分段时更新：

$$RTT = \alpha * RTT + (1-\alpha) * M \tag{2-2}$$

$$D = \alpha * D + (1-\alpha) * |RTT - M| \tag{2-3}$$

其中 M 是分段发送和确认到达之间的时间，α 是平滑因子，一般取 7/8。

在计算 RTT 时可能会存在一个问题：如果重传计时器超时了，分段被重新发送，那么接收端发送的 ACK 分段可能是对之前分段的确认，也有可能是对重传分段的确认。为了避免这种混淆 RTT 计算的情况，如果分段一旦被重传那么不会去更新 RTT[12]。

2.1.4 TCP 拥塞机制

流量控制允许发送方在接收方因内存限制而无法接收时减慢传输速度。如果发送方将大量数据传输到网络中，可能会使网络节点的处理能力过载，因此数据包会在网络层丢失。为此，在发送方维护拥塞窗口（Congestion Window，CWND）[13]，该窗口值表示发送方在接收 ACK 之前允许的最大网络传输量大小。拥塞窗口的大小是通过监测网络的状态来决定的。

拥塞控制有 4 种算法用来更新拥塞窗口和增加拥塞控制的效率，分别是慢启

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

动、拥塞避免、快速重传以及快速恢复[14]。如图 2-4 所示。

慢速启动意味着当连接打开时，发送方最初以低速率发送数据。这意味着初始窗口最多为 2*MSS（Maximum Segment Size）。如果没有数据包丢失，那么拥塞窗口会迅速增加，在每次传输中会增加一倍。当检测到数据包丢失时，拥塞窗口重置为 1MSS，并再次应用“慢启动”。

拥塞避免算法的主要思想是缓慢增大拥塞窗口，窗口增大的趋势是呈线性的，当拥塞窗口增长太大可能会导致网络拥堵，在拥塞避免阶段每经过一次往返时间，发送数据一方的 CWND 就会增加 1MSS，这里并不是无限增大会有一个门限值 *ssthresh* 。一开始时在没有数据丢失的情况下，CWND 值会迅速增大，当 $CWND \leqslant ssthresh$ 时，本文提到四种拥塞算法的前两种都可以使用，但是当 $CWND > ssthresh$ 时，TCP 协议建议应用第二种拥塞控制机制而不是慢开始机制。

快重传算法的目的是通过接收方发送重复的 ACK，让发送方知道接收方某个分段没有收到，这时发送方立即就会重新发送丢失的数据，这个时间不超过 RTO，就不会引发网络拥塞，可以让网络的吞吐量提升 20%。

快恢复算法是指当检测到 3 个重复确认时，不应用慢启动算法，不用让 CWND 从 1MSS 再开始重新增大。可以从某一个值开始恢复，这样可以在短时间内重新达到之前的发送量。

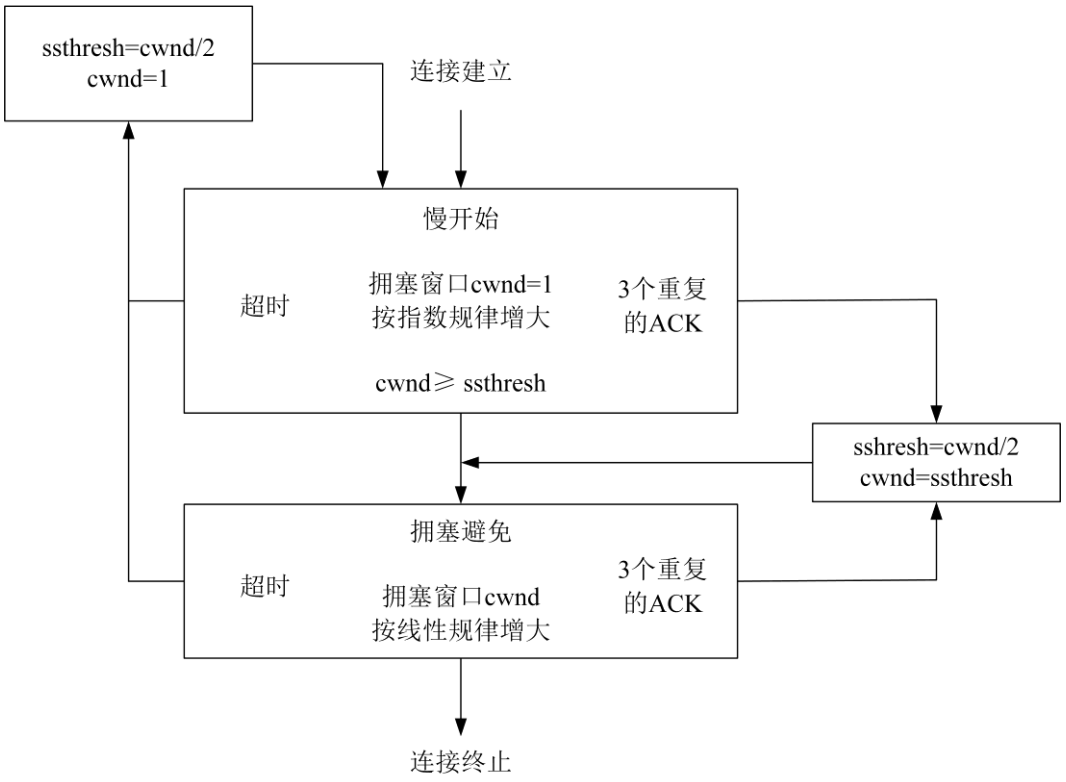

图 2-4 TCP 拥塞控制流程图

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.1.5 IP 协议

IP 协议是位于网络层的协议，负责将 TCP 数据封装打包，根据目的地址发送到相应主机上去。其中较为重要的概念就是 IP 地址，IP 地址表示网络中节点的标识符，一台主机在得知另一台主机的 IP 地址后可以通过路由器、链路将消息发送过去。IP 地址一般是由 1~3 字节长度的网络字段，和相应 3~1 字节长度的主机字段组成，当网络中的主机 A 和主机 B 的网络字段的值是同样的，则不需要进行路由器转发直接交付即可，否则需要使用路由器查找路由表进行转发，直到交付到目的主机为止[15]，路由器在转发时经过的链路可能不仅仅是地面以太网，也有可能是空间无线链路。但是上述 IP 地址也存在着问题，比如存在许多地址浪费的现象、使用起来难以变通等，为了解决这些问题，IP 协议将网络字段再扩展一个子网字段，在引入了子网字段之后，IP 地址就变成了网络字段，子网字段和主机字段的组织结构。

对于网络中的每一个节点而言，它们的 IP 地址都是唯一的，包含在 IP 数据包中。路由器的任务就是根据 IP 地址找出下一跳的节点并转发该 IP 数据包。但是 IP 数据包的长度是可变的（20~65535 字节），可能会大于链路层的可发送的数据包长度，需要对 IP 数据包进行切割分成多个片段传输，再经由目的主机收集片段并组装成完整的数据报，向上交给传输层。

2.2 CCSDS 协议介绍

CCSDS 协议族的结构大致与 TCP/IP 协议族模型类似，从上至下共有 6 层，分别是应用层、传输层、网络层、数据链路层、同步信道编码层、物理层，其中数据链路层和同步信道编码层都属于链路层[16]。下面依次对 CCSDS 网络每层常用的协议进行介绍。其结构组成如图 2-5 所示。

应用层：主要包括 CFDP、LDC、FTP、SCPS-FP、ASP[17]。其中第一个和第三个协议都是关于文件传输规范的，CFDP 协议是跨层设计的，也可以用在传输层，LDC 是为了尽量传回有价值的数据同时也尽可能的少占用卫星的存储空间和链路带宽[18]，ASP 是具体的应用协议。

传输层：主要协议有 SCPS-TP、TCP、UDP 和 SCPS-SP。其中 SCPS-TP 协议是针对空间通信存在的一些问题，如链路容易中断、传输延迟高、信噪比低等，专门定制的 CCSDS 协议，对比 TCP 协议，SCPS-TP 协议在一些方面做了改进，这部分内容将在第三章中详细叙述。

网络层：主要协议有 SSP（空间包协议）、SCPS-NP、IP 协议。其中 SSP 和 SCPS-NP 是 CCSDS 的定制协议。网络层可以使用封装技术，让地面的 IP 数据报

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

可以在空间链路中传输。

链路层：CCSDS 协议将链路层划分为 2 层，前者主要包括 TMSDLP 协议、TCSDLP 协议、AOS 协议、Proximity-1 SLP，其中前两个分别对应着遥测和遥控的体制，而后者包含 3 个内容：TMSCC、TCSCC 和 Proximity-1 SLP，其中前两个都是关于信道编码的规定，Proximity-1 SLP 跨越了整个链路层[19]。需要说明的是 CCSDS 链路层协议可以直接传输 IPv4 数据包、SCPS-NP 数据包等，其他类型数据需要用到 CCSDS 提供的 ENCAP 服务才能在空间信道中被复用传送[20]。

物理层：首先是 Proximity-1 SLP 该协议横跨了链路层和物理层，再就是射频与调制系统主，主要是对空间信号使用的频段和调制做出了规定。

总的来说 CCSDS 协议是一个包含多个协议的庞大组织体系，在空间任务中使用哪些协议是由任务的具体情况来决定。每种协议有着各自的数据格式（PDU）。

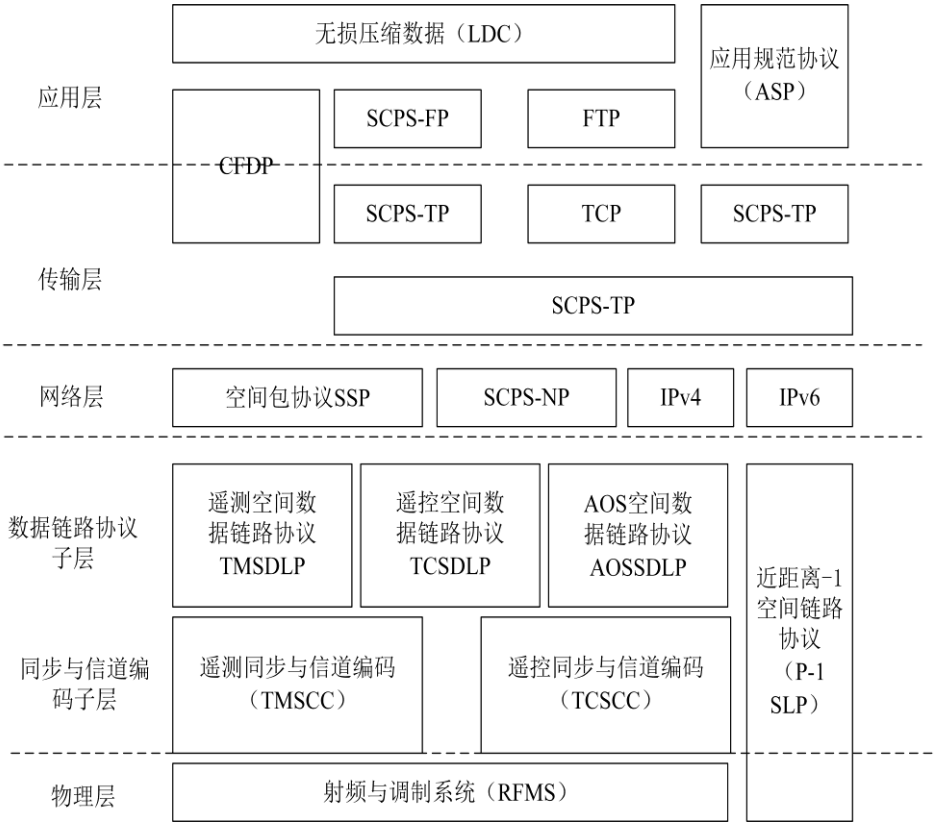

图 2-5 CCSDS 协议模型

2.3 CCSDS 链路层

AOS 是用于地对空、空对地和空间链路通信的 CCSDS 链路层协议，在 COS 协议的基础上开发出来的，一般用在中低数据速率的航天器上，是面向复杂的航天任务。AOS 在数据处理和交换方面以标准的格式处理，凭借其 7 种服务类型满

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

足了不同航天器的业务需求。

2.3.1 CCSDS 主网

CPN（CCSDS 主网）是 CCSDS AOS 空间数据管理系统的核心，提供端到端的数据通信服务。CPN 由地面和空间两大部分组成，地面部分主要是地面子网，空间部分主要是星载子网和空间链路子网（SLS），其中空间链路子网是 CPN 的最重要的组成部分[21]。图 2-6 为 CPN 的结构，从图中可以看到空间链路子网起到了一个桥梁的作用，把地面和空间的通信网络连接起来，如果是与地面子网连接则是 CPN 空地配置，否则的话则是 CPN 空空配置。

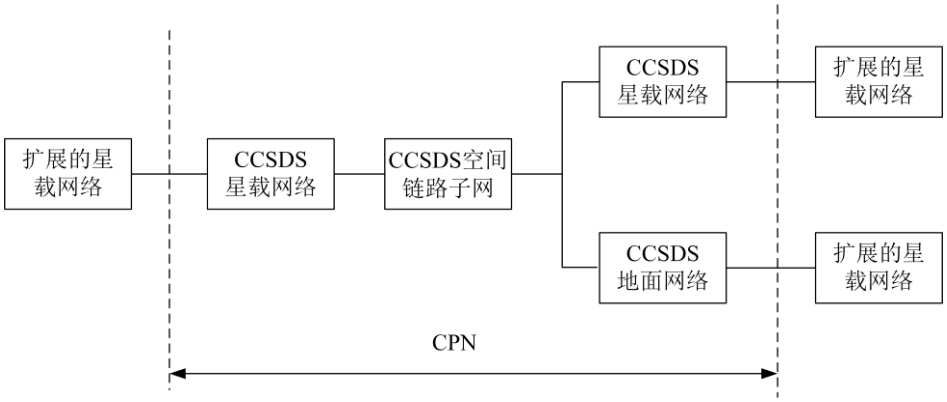

图 2-6 CCSDS 主网核心结构

扩展的星载网络：为 CPN 的空间段部分，主要为航天器，如月球探测器、中继卫星等。如果以上的内部电路系统足够复杂，那么可以视为单独的网络[22]。

CCSDS 地面网络：主要由地面设备系统和数据接口组成。

CCSDS 空间链路子网（SLS）：在 CCSDS 星载网络和 CCSDS 地面网络中起到连接的作用，是 CPN 的核心部分。图 2-7 为 SLS 的结构图，其中 SLS 的重要思想是虚拟信道，它是指一个物理上的实际信道可以容纳多个逻辑信道，每个逻辑信道都可以用来进行数据通信，并且每个逻辑信道都会有相应的标志，这样可以提高空间信道的利用率。

根据航天任务的不同需求，CPN 提供了 2 种类型的服务，一种是端到端的服务，即数据从 CCSDS 主网的空间网络端点发送到地面网络端点或是从地面到空间[23]，如处理语音、电视及科学等不同信源的数据，该服务类型包括 2 种服务路径服务和网间服务，另一种是 SLS 业务，即在子网中提供点到点的数据传输服务，例如 Digital Audio、Magnetic Recording Replay 等，点到点业务提供了封装服务、位流服务、多路复用服务、插入服务、虚拟信道数据单元服务和虚拟信道访问

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

（VCA）服务[24]。综上，CPN 总计可以提供八种服务，来满足不同用户的数据传输需求，每种业务都对应了不同的业务数据单元（SDU）。

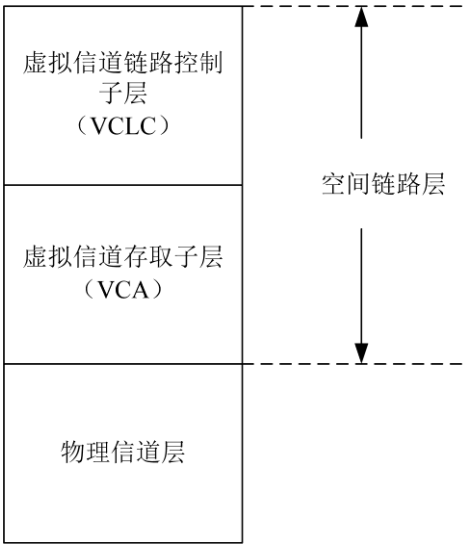

图 2-7 SLS 结构

2.3.2 AOS 协议概述

AOS 空间数据链路协议提供七项服务，表 2-1 对这些服务从四个方面进行了描述。AOS 协议定义了三种服务种类（异步、同步和周期），它们决定了用户提供的 SDU 怎么样通过空间链路在 PDU 中传输[25]。

表 2-1 AOS 链路层协议服务概述[26]

服务	服务种类	服务数据单元	业务接入点（SAP）地址
虚拟信道包（VCP）	异步	包	GVCID+包版本号
位流	异步或周期	位流数据	GVCID
虚拟信道访问（VCA）	异步或周期	VCA_SDU	GVCID
虚拟信道帧（VCF）	异步或周期	传输帧	GVCID
主信道帧（MCF）	异步或周期	传输帧	MCID
插入	周期	IN_SDU	物理信道名称
虚拟信道操作控制域（VC_OCF）	异步或周期	OCF_SDU	GVICD

在异步服务中，服务用户提供的服务数据单元与服务提供者生成的传输帧，这二者的传输关系不存在时序关联，用户可以在其希望的任何时间请求数据传输，

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

但服务提供者可能会对数据生产速率施加限制。在同步服务中，服务数据单元的传输与信道所有传输帧的释放同步。而周期服务是同步服务的一种特殊情况，其中服务数据单元以恒定速率传输。

在 AOS 传输帧头中有三个标识符字段：虚拟通道标识符（VCID）、航天器标识符（SCID）和传输帧版本号（TFVN）。其中第二个和最后一个标识符组成了主通道标识符（MCID），主通道标识符和传输帧头中第一个标识符字段组成了全局虚拟通道标识符（GVCID）[27]。因此，可以得出如下关系：

MCID = TFVN + SCID

GVCID = MCID + VCID = TFVN + SCID + VCID

物理信道中每个虚拟通道都由 GVCID 标识。因此，虚拟信道由拥有一致的 GVCID 的传输帧构成。AOS 空间链路信道之间的关系如图 2-8 所示。

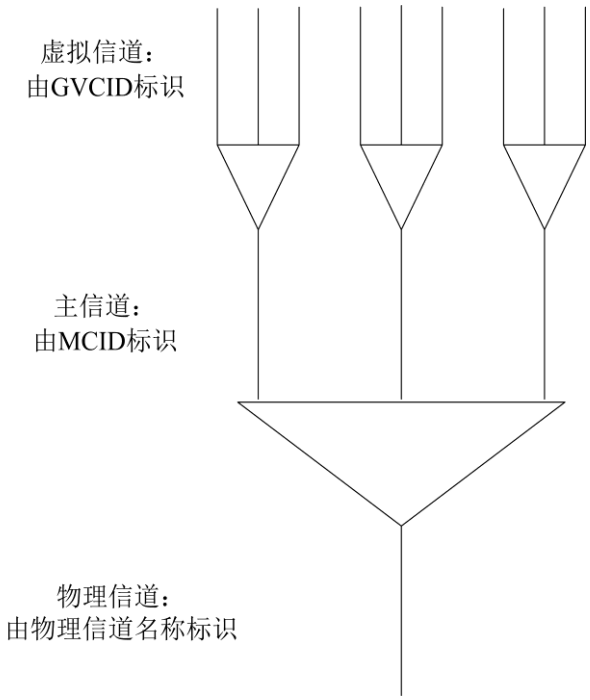

图 2-8 信道之间的关系

2.3.3 AOS 传输帧

AOS 的 7 种服务类型都是为了传输不同来源的数据包而提出的，本文要实现的就是 CCSDS 空间链路中传输 IP 数据包，即 IP over CCSDS，从本文论述的重点来说就是 IP over AOS。想要 IP 数据包在 CCSDS 空间链路上传输需要先将 IP 协议数据单元（IP_PDU）添加 CCSDS 规范的扩展部分（IPE），之后将扩展后的 IP 数据包进行 CCSDS 封装，最后再放在 AOS 传输帧进行传输。如图 2-9 所示。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

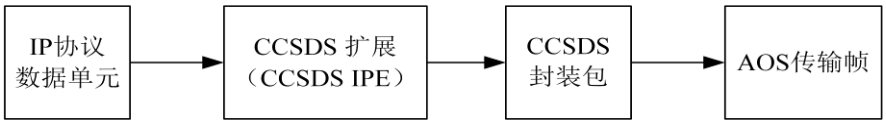

图 2-9 IP 数据报传输流程

图 2-10 中为 IP 数据包的封装过程示意图。封装首部、IPE 首部和 IP 数据包数据域组成 AOS 帧，其中 ENCAP Head（封装首部）、IPE 和 IP 数据组成了 M_PDU 分组区[28]。图 2-10 下半部分为 IP 数据包封装成的 AOS 帧。

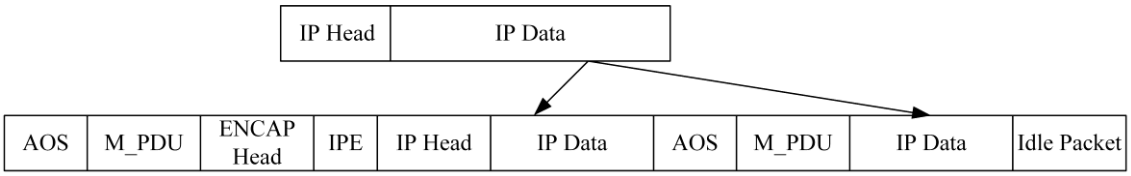

图 2-10 IP 数据包封装

2.3.4 AOS 业务等级

针对空间网络信道误码率较高，为了提升传输效率，CCSDS AOS 提出了 3 种不同的通信业务等级，来进行数据检错和纠错。

1 级服务：AOS 传输使用编码虚拟信道数据单元（CVCDU）格式，使用了 ARQ 重传机制和 RS 纠错编码，可以保证数据出错概率低，其数据单元的丢失率小于 10^{-12}。

2 级服务：AOS 传输采用 CVCDU，使用了 RS 纠错码，数据出错的概率比 1 级服务低，但是仍然很高，保证数据序列行不被破坏，其数据单元的丢失率小于 10^{-7}。

3 级服务：AOS 传输采用 CVCDU，没有差错控制机制，依赖信道本身的性能，数据出错概率不高不低，用户数据不完整的可能性高，其数据单元的丢失率小于 10^{-7}。

2.4 CCSDS 物理层

2.4.1 概述

CCSDS 物理层由射频与信号调制系统、Proximity-1 物理层组成。射频与信号调制系统使用在地面站、中继卫星和航天飞船上，从技术标准、指导意见和程序设计这个 3 个方面来介绍系统的构成和使用，技术标准主要包括了载波调制、天线极化方式、信号编码等技术指标，指导意见主要包括了频率利用、能量限制、

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

解调方法、操作程序、测试建议等技术细节，程序设计则包含了设计工具和程序算法。表 2-2 为 CCSDS 物理层的信号调制和射频参数。

表 2-2 部分信号调制和射频参数

名称	参数
相位调制方式	使用残留边带
子载波频率	8 或 16kHz
地空多普勒频移	2GHz：±80 kHz 7GHz：±300 kHz
信号的最大载波抑制	10dB
工作频段	X、Ka 双频
增益要求	X 频段大于 49dB；Ka 频段大于 61dB

2.4.2 技术难点

物理层的技术难点在于天线技术。X/Ka 双频段天线技术可大大减少整个有效载荷的体积，具有先天优势，其难点在于要兼顾 X 和 Ka 两个频段的设计因素，并同时保证天线在两个频段的高效率辐射性能。

通过对国内外高增益可展天线研制情况的回顾，可以看出网状反射面可展天线是目前最为成熟的可展天线方案，特别是周边桁架式和径向肋式的网状可展天线方案，由于其较完备的设计理论，是深空探测高增益天线设计方案的首选。但对于地月天线系统而言，由于其工作频率高达 Ka 频段，对反射面型面精度要求很高。因此地月传输来说，不论是径向肋形式还是周边桁架式的可展天线，都面临着反射面型面精度控制困难，展开结构刚柔混合多体建模复杂，以及天线环境适应性和地面验证试验困难等诸多问题，需要对其开展更为细致的研究和仿真分析，方能满足该项目天线指标要求。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.5 CCSDS 数据安全

在 CCSDS 蓝皮书中包含了关于加密安全算法的建议，用于数据加密、认证加密。采用正确的标准算法将实现安全的互操作性，并且可以降低使用安全服务的成本，这些加密算法可以为数据系统提供机密性和身份验证的保护。CCSDS 并不要求在哪一层使用加密算法，根据航天任务的不同，加密可能在如 IPSec 一样在网络层之上实现，也可能用于数据链路层，甚至是物理层（批量加密），它也可以在多个层同时使用。CCSDS 推荐在航天任务中使用 AES 加密算法，对于加密密钥大小做出了规定，目前使用的密钥位数为 128bit，在以后的任务中建议使用 256bit 的密钥，在算法操作模式上建议使用计数器模式，在认证加密方面 CCSDS 推荐 GCM（伽罗瓦/计数器模式）模式，GCM 既可以在硬件上也可以在软件上快速认证加密，而且可以并行化和流水线化，有利于航天器在空间执行任务，不要使用无关的、一次性的位填充。

空间链路数据安全协议是一种用于空间任务的数据处理方法，这些任务需要对传输帧的内容进行身份验证或保密。该协议在保证数据安全的同时，也可以保证数据传输的服务质量，它也是可扩展的，能够支持任意数量的虚拟信道的数据传输。本文使用 AOS 协议作为链路层协议，安全协议能够为 AOS 提供服务，如图 2-11 所示。

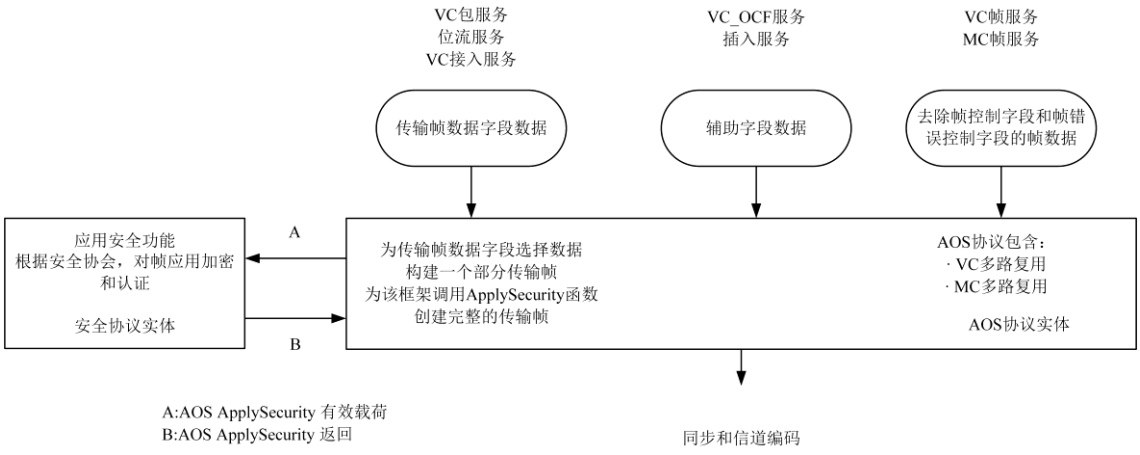

图 2-11 安全协议对于 AOS 服务的支持

从图 2-11 中看出安全协议为 AOS 传输帧数据字段数据提供了认证、加密和认证加密的服务，所以它对虚拟信道包服务、位流服务和虚拟信道访问提供了全面保护，安全协议不为 AOS 辅助字段提供加密的保护。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.6 本章小结

本章主要 TCP/IP 协议和 CCSDS 协议进行了一些介绍。前者主要内容为 TCP 协议的首部、关键机制和拥塞机制，IP 协议的地址组成，后者介绍了空间链路层 AOS 协议，着重叙述了 AOS 协议的服务类型、传输帧以及业务等级，对 CCSDS 物理层做了一些简单介绍，包括技术参数和技术难点，最后对 CCSDS 数据安全做了了解。对于 TCP 协议和 AOS 协议的描述，为下一章的传输层的改进和链路层的设计打下了基础。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第三章 天地互联的协议转换设计

3.1 基于 IP over CCSDS 的协议转换方式

从第一章和第二章的分析可以知道，当地面和空间采用各自的网络协议，在进行数据传输时，是需要进行协议转换，本文采用天地互联的协议转换设计，因为天地一体化的网络协议会使天地通信变得方便高效，同时还可以使用地面网络中已经成熟的技术减少开发的成本，使系统的鲁棒性和拓展性大大提高。

所以综上原因，在结合 CCSDS 蓝皮书的建议上本文提出 IP over CCSDS 的天地互联的协议转换方式，在网络层应用地面成熟 IP 协议，在链路层应用范围广的 AOS 协议。

根据 CCSDS 出版物，得知 IP over CCSDS 传输数据包有三种方法[29]，如图 3-1 所示。

（1）直接进行帧封装，即不加任何处理地把 IP 数据包置于 CCSDS 帧中；

（2）利用 CCSDS AOS 的封装包服务；

（3）用户定义的串联数据流封装；

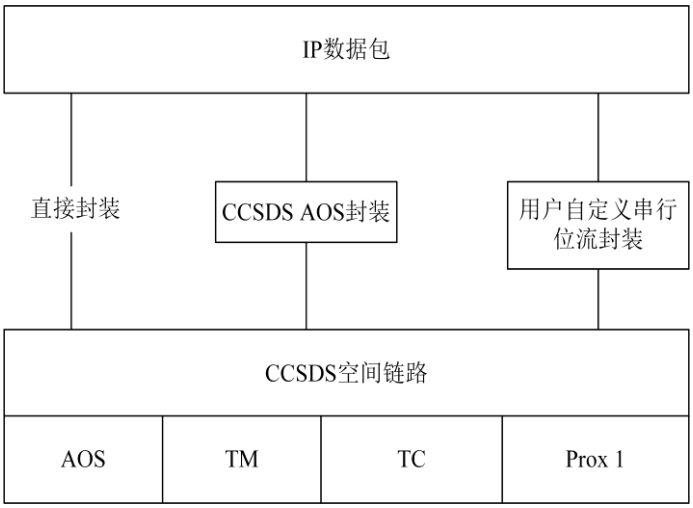

图 3-1 IP over CCSDS 传输 IP 数据包的方式

从中看出 IP over CCSDS 传输 IP 数据包的方式主要有两种形式，包的形式和数据流的形式。如果以数据流的形式传输会存在几种缺点，一是会破坏数据包的完整性，二是是需要进行 IP 数据包的重建并且还要设置同步标志位。因此，本文采用的是第二种方式，即 CCSDS AOS 封装包的形式。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

图 3-1 主要是 IPv4 数据包的传输方式，随着 IPv6 的提出和普及应用，CCSDS 也相应的对 IP 数据包的传输方式做出调整和完善，两种类型的 IP 数据包传输方式如图 3-2 所示[30]。

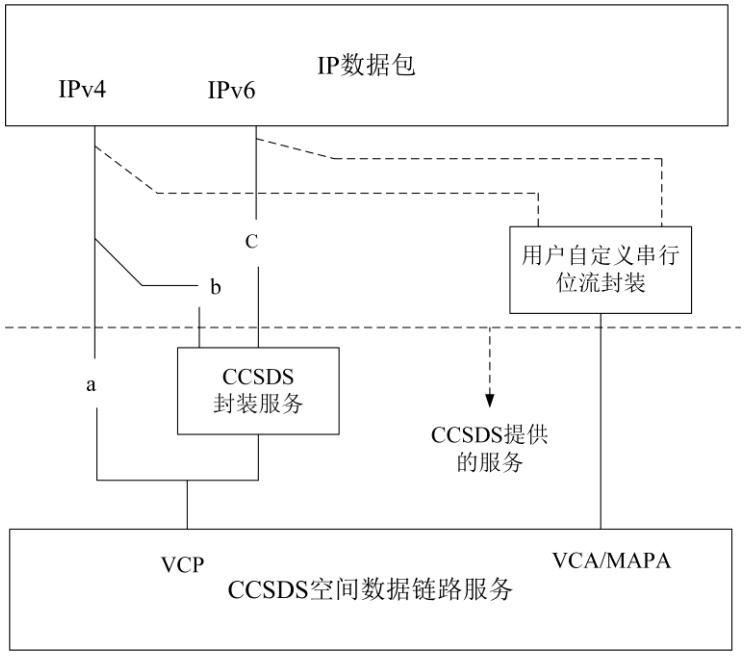

图 3-2 更加完整的 IP 数据报传输方式

a 方式是将 IPv4 数据包通过复用传输的方式将其置于空间链路帧中传输，该方式使用的是虚拟信道包服务（VCP）。

b 方式用到了 CCSDS 封装服务，是把 IPv4 数据包按照一定次序封装在封装业务数据单元（E_SDU）中，在空间链路中进行传输。

c 方式也是利用了 CCSDS 封装服务，是把 IPv6 数据包按照一定次序封装在 E_SDU 当中，在空间链路中传输。

本文选择的是 IPv4 数据包进行传输，即 a 方式。图 3-2 中虚线部分为不推荐的方式。

3.2 协议转换模型和空地数据交互

在提出了 IP over CCSDS 的协议转换方式后，还需要构建具体的网络协议模型，该模型主要涉及到空间航天器模型的构建、地面测控站网络模型的构建和天地之间的通信传输。地面测控站网络模型主要是利用 TCP/IP 协议族，而空间网络模型主要是 CCSDS 协议，并针对空间环境的特殊性，对网络的传输层 TCP 协议进行了改进，以实现更加高效的数据传输。天地互联的网络拓扑如图 3-3 所示。图

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

中航天器网络系统和地面指挥中心之间通过 IP over CCSDS 空间链路连接，指挥中心再通过专用 IP 与地面站发送信息。

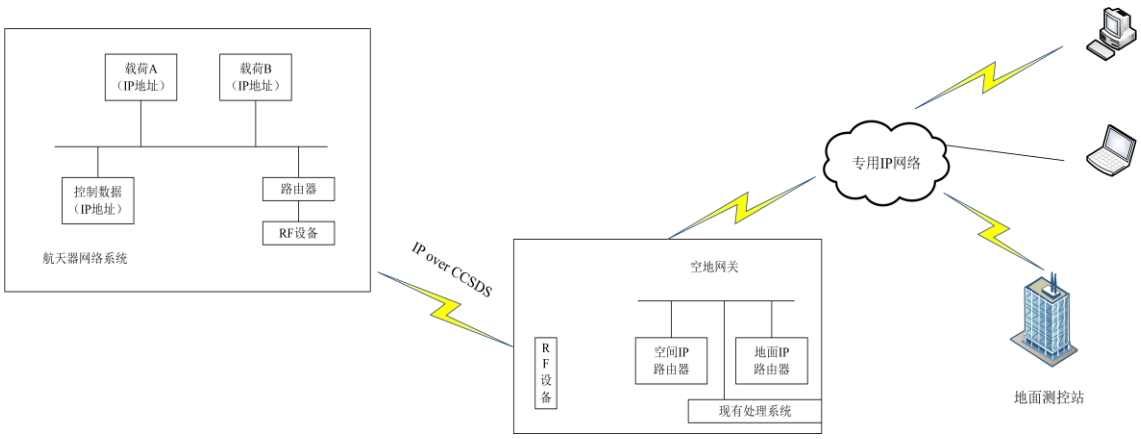

图 3-3 天地互联的网络拓补图

3.2.1 天地互联总体协议栈模型

由于 CCSDS 协议族中包含很多协议，根据不同的航天任务可以选择不同协议组成空间网络，来满足用户需求，所以存在多种空间网络协议组合。空间环境比地面特殊，存在信号弱，时延大，误码率高等问题，所以本文着重考虑空间网络协议的选择，尤其是传输层协议，它将直接影响协议转换的性能，除此之外本文还关注了网络层以及空间链路层，由于本文并不涉及到硬件方面，所以物理层就不再叙述。

（1）传输层

传输层的 TCP 协议是可靠的传输协议，在地面网络应用广泛，但是在空间中传输数据可能会存在卫星链路延时长，TCP 协议无法判断丢包原因的问题，为了在航天器与航天器之间，航天器与地面之间进行有效可靠的数据交互，本文在传输层使用 SCPS-TP 协议，这是 CCSDS 针对空间网络设计的传输层协议，该协议将在本章第五节被详细叙述。

（2）网络层

网络层本文选择 IP 协议，IP 协议可以屏蔽链路层及以下的差异，而且 IP 协议技术成熟，在地面网络应用广泛，采用该协议可以降低天地互联网络的开发成本，同时 CCSDS 空间链路层 AOS 协议提供封装 IP 数据包的服务，将 IP 数据包封装成传输帧，使 IP 协议在空间链路中应用起来更加便捷。CCSDS 协议针对网络层开发出了 SCPS-NP 协议，不选择该协议的原因是需要改变地面网络设备的终端协议，不如 IP 协议可以大规模使用成熟的商业设备。但是空间网络中卫星、空间站等节

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

点都是处在移动的状态中，可能会从一条链路中跑到另一条链路中去，这会使 IP 数据包无法达到正确的节点，所以有必要对 IP 技术进行一些改进，即移动 IP 技术，这一部分内容本文将在 3.4 节详细叙述。

（3）数据链路层

因为在链路层空间和地面网络存在较大不同，所以在链路层不能使用和地面网络相同的协议。CCSDS 在链路层开发出了 4 种协议，其中 TM（遥测）和 TC（遥控）协议只适用于单向的数据传输，在航天器和地面通信交互性方面不足，而 Proximity-1 协议只能用于空间段的航天器的通信传输，因此数据链路层我们选择使用 CCSDS AOS 协议，本章将在第二节对 AOS 的服务类型和帧结构进行设计。

综上本文天地互联的协议转换设计是以 IP over CCSDS 的转换方式作为基础，传输层采用 SCPS-TP/TCP 协议，网络层使用 IP 协议，数据链路层应用 CCSDS AOS 协议，实现天地互联传输 IP 数据包[31]，IP over CCSDS 从本质上说就是在空间网络中传输 IP 数据包。图 3-5 为天地互联的协议转换的设计模型。

图 3-4 空间系统中包括各种航天器，如卫星、空间站、月球着陆器等，用来传输数据完成科研任务，地面系统主要指地面测控站和用户。地面系统通过地面路由器和地面网关，将控制信息传递到航天器，而航天器的遥测数据等通过通信协议转换等一系列操作将数据转换成能够在网络层使用的 IP 数据包，经由空间路由器和空地网关到达地面测控站。

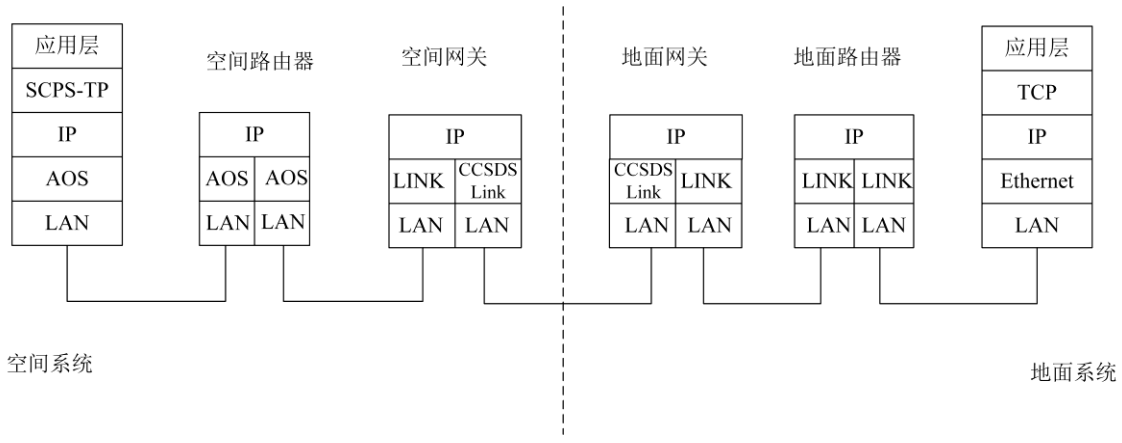

图 3-4 天地互联网络协议栈模型

3.2.2 天地数据交互

从以上分析可以知道，在网络协议转换中最重要的是网关功能的实现，在天地互联的网络协议中就是指地面网络和空间网络如何进行数据交互。

当地面网络向空间网络发送数据时，先是经过 IP 数据包的处理模块，检查 IP

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

地址，经过地面路由器转发至地面网关，在经过链路层时需要进行 AOS 帧的封装，即调用 AOS 协议的虚拟信道包服务，当 AOS 帧到达空间网络并且被接收时，空间网络接收系统对 AOS 帧进行解析服务，得到 IP 数据报后根据地址在经由空间路由器转发到目标航天器，完成数据传输。

当目标航天器需要向地面测控站或是指挥中心发送测量数据时，需要先将 IP 数据报封装成 AOS 传输帧，之后根据 IP 地址转发到相应的空间路由器，之后通过路由器到达空间网关，当 AOS 帧到达地面网络时，需要 AOS 帧解析的服务，从帧的数据字段获得 IP 数据包，在根据 IP 地址，经过地面路由器转发到相应的地面测控站。图 3-5 为空地数据交互的过程。

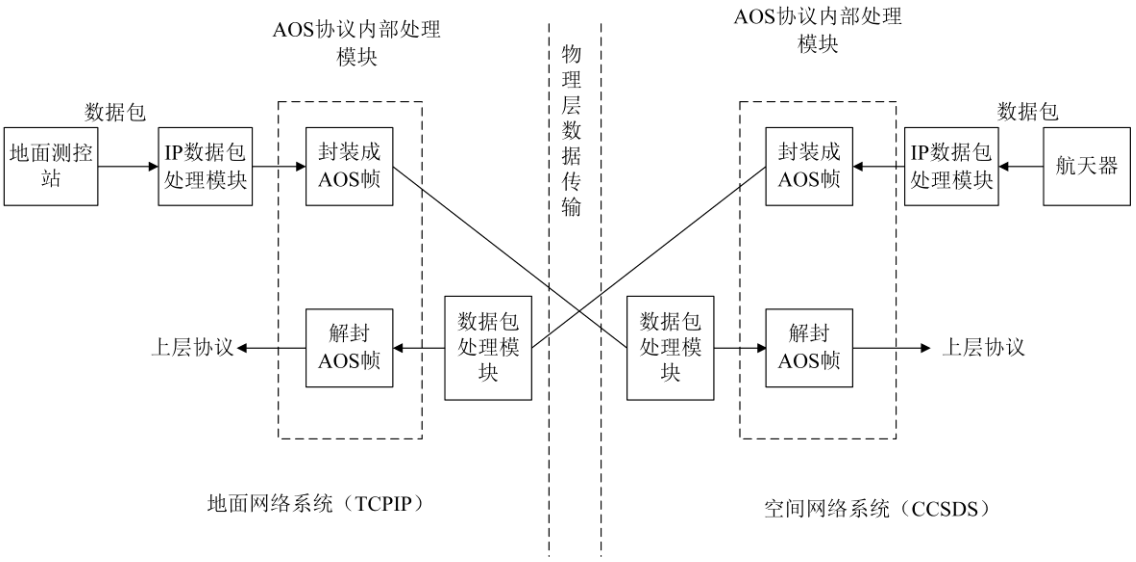

图 3-5 天地数据交互过程

3.3 链路层 AOS 协议的设计

CCSDS 在 AOS 协议蓝皮书中提供了对 IP 数据包的传输支持即在链路层可以应用 IP 数据包，并将其与 CCSDS 源格式数据包视为同样的级别。本文使用 IPv4 数据包不需要使用封装数据单元，而是使用多路复用的服务即可在空间链路中进行传输，IPv4 数据单元在 CCSDS 中视为 M_SDU，再使用多路服用 M_PDU 后在空间链路子网中传输。

空间网络较为重要的是空间链路层协议，因为空间网络的链路特性与地面网络差异很大，链路不对称的问题很突出，而链路层又涉及到天地互联的数据交互，IP 数据包的封装和解析等，以下章节将重点论述 AOS 传输帧的设计和数据包的相关处理过程。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.3.1 AOS 传输帧设计

AOS 传输帧包含 5 个主要字段，并且按照以下顺序连续放置：

a）传输主帧头（长度 6 或 8 个字节，强制选）；

b）传输帧插入区（长度可变，非必选）；

c）传输帧数据字段（长度可变，强制选）；

d）操作控制字段（长度 4 个字节，非必选）；

e）帧差错控制字段（长度 2 个字节，非必选）[32]；

对于空间链路上的通信信道来说，AOS 传输帧所占据的字节数是恒定的，传输帧长度的改变可能会导致接收机失去同步。其帧结构如图 3-6 所示。

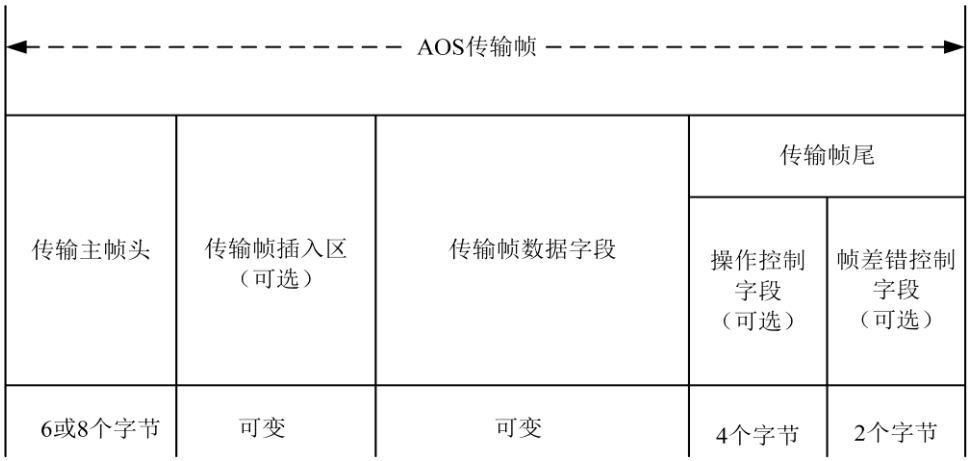

图 3-6 AOS 传输帧结构

传输主帧头也包含 5 个部分，分别是主信道标识（10 个比特，强制选）、虚拟信道标识（6 个比特，强制选）、虚拟信道计数（3 个字节，强制选）、信号字段（1 个字节，强制选）、帧头差错控制（2 个字节，非必选）。其格式如图 3-7 所示。

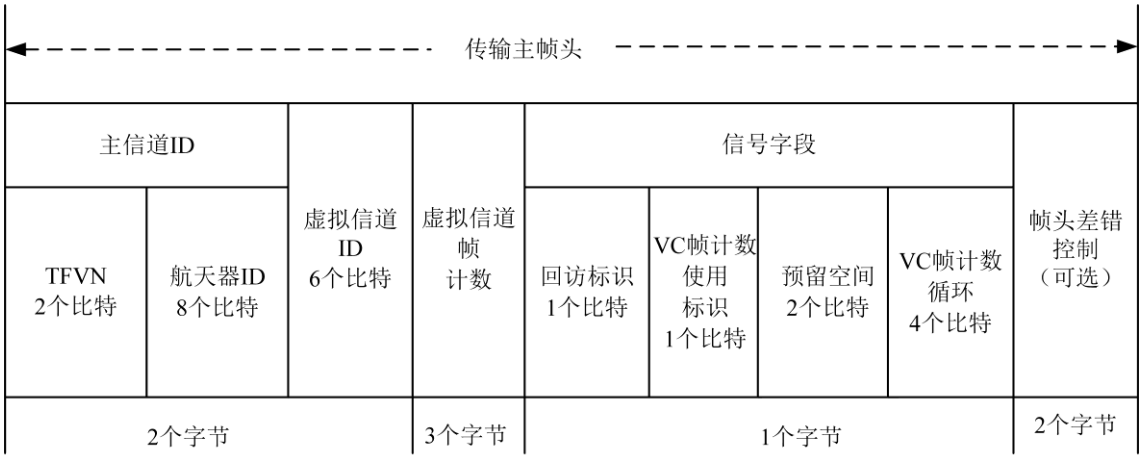

图 3-7 传输帧主帧头结构

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

需要说明的是 TFVN 一般取二进制值“01”；关于虚拟信道 ID 最多可以设置 64 个（2^6=64），若只使用一个虚拟信道则 ID 值为“000000”，若是虚拟信道传输的数据是空闲数据，则虚拟信道 ID 值为“111111”；虚拟信道帧计数字段包含虚拟信道内传输的帧计数值（模值为 16777216）；回放标识主要用来区分实时传输帧和回访的传输帧，因为它们可能使用了相同的虚拟通道，一般用“0”表示实时传输帧，“1”表示回访的传输帧；VC 帧计数使用标识字段为 0 表示未使用 VC 帧计数循环字段且接收器忽略该字段，为 1 表示表示使用 VC 帧计数循环字段且由接收器解释；预留空间由 CCSDS 保留，以供将来定义，它的典型值为“00”；VC 帧计数循环：每次 VC 帧计数返回到 0 时，VC 帧计数周期应增加，VC 帧计数循环有效地将 VC 帧计数从 24bit 增大到 28bit；帧头差错控制主要用来对主信道 ID、虚拟信道 ID 和信号字段进行纠正编码保护[33]。

本文使用 M_PDU（多路复用协议数据单元）来填充图 3-6 中传输帧数据字段，使用 M_PDU 的原因是该数据单元的数据域可由 CCSDS 格式数据包填充。M_PDU 的格式如图 3-8 所示。可变部分为 M_PDU 数据域即包区，为 CCSDS 包填充。M_PDU 头部包含了预留位，该值由默认的五个零来填充，第一头指针指向了首个 CCSDS 包中的第一个字节位置。

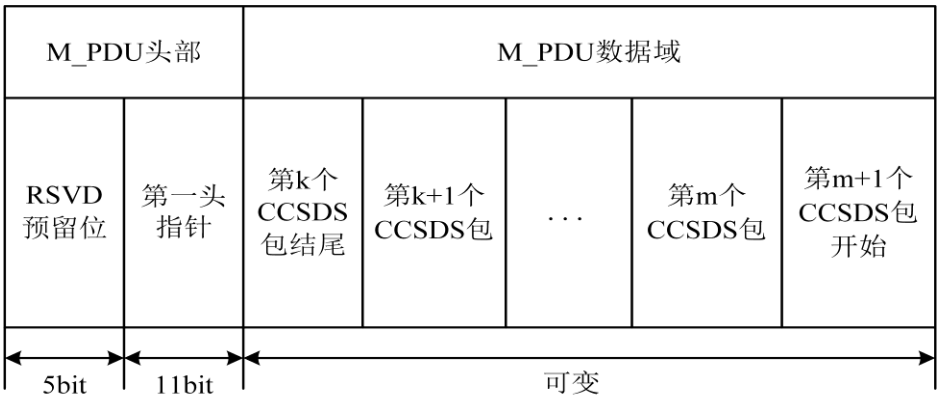

图 3-8 M_PDU 格式

根据 2.3.4 节 AOS 业务等级共有 3 级，本文仿真设计选用第三级，没有差错控制和编码方式，但是这里预留了差错字段位置，使用最简单的帧结构进行实现。综上，本文设计的 AOS 传输帧总长度为 65 字节，这里的传输帧插入区为管理层设置，本文不用设置，所以传输帧主帧头的长度为 5 字节，传输帧数据字段长度为 58 字节，总的 AOS 传输帧格式如图 3-9。其包括了主帧头、数据字段和差错控制字段，数据字段由 M_PDU 来填充。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

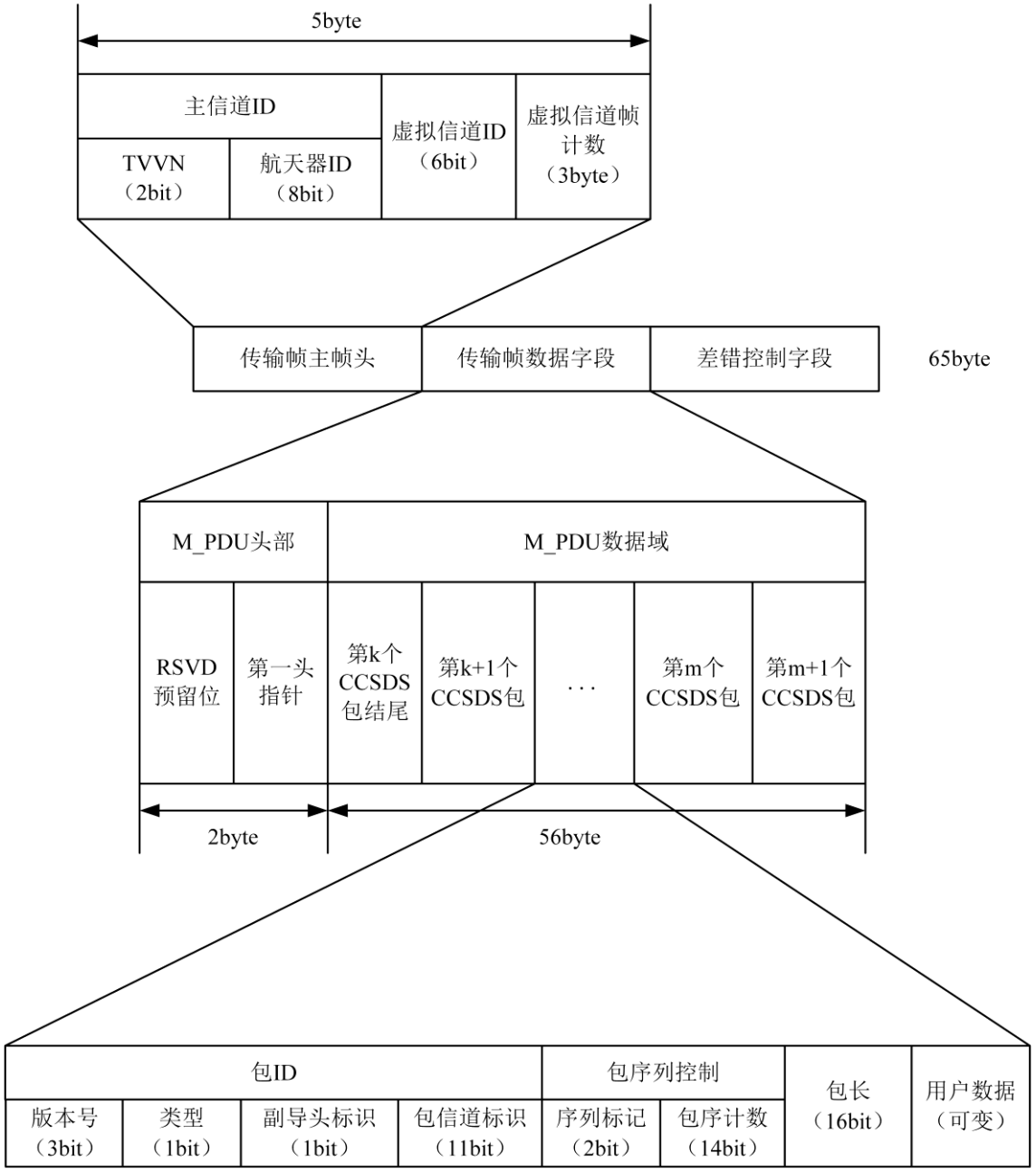

图 3-9 总的传输帧结构

3.3.2 IP 数据包的处理

在 2.3.2 节中提到了 AOS 的 7 种服务，本文挑选其中较为重要的几种服务来实现：虚拟信道包服务（VCP）、虚拟信道访问服务（VCA）、虚拟信道操作控制域服务（VC_OCF）、虚拟信道帧服务（VCF）、主信道帧服务（MCF）和插入服务[34]。IP 数据包在发射端的处理流程如图 3-10 所示。

本文使用了 IPv4 的数据包，如果想要使用 IPv6 数据包只需要在包处理模块之前加一个封装模块。IPv4 数据包经过包处理模块之后成为长度固定的 M_PDU，进

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

入虚拟信道后经过虚拟信道帧模块变成 VC_PDU，之后通过 MCF 服务和 AOS 帧生成模块，最终成为完整的 AOS 传输帧。

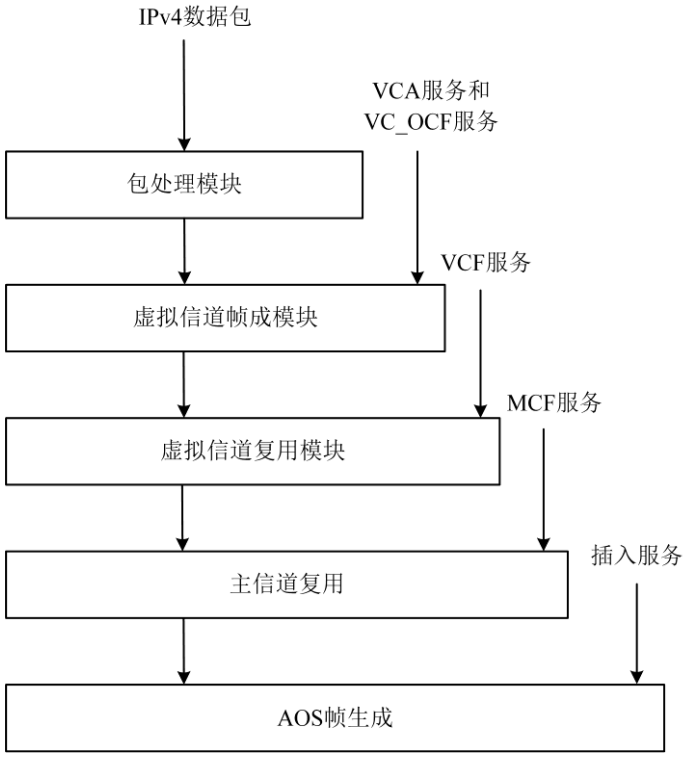

图 3-10 IP 数据包在发射端处理流程

IP 数据包的位数为 16 位，其占据的字节数在一定范围内变化（0~64KB），长度因数据部分而变化，AOS 传输帧的长度是固定的，当 IP 数据包的占据的字节长度大于 M_PDU 包区长度的话，就要对 IP 数据包进行分割成能够放入 M_PDU 长度的块。

假设 M_PDU 的字节数为 MB，封装首部的字节数为 EB，IPE 的字节数为 IB，IP 数据包的字节数为 IPB，则由 2.3.3 节内容可推出 M_PDU 的包区字节数为 B=MB-EB-IB。

对 IP 数据包的分割流程如下：

（1）读取 IP 数据包得出其字节数 IPB，与 M_PDU 的包区字节数 B 比较大小，当 IPB≤B 时，就直接把 IP 数据包放入包区中，如果不够的话需要填充空闲数据；当 IPB＞B 时，需要对 IP 数据包分割操作，把字节数为 B 的 IP 数据从 IP 数据包中拿出放入包区中，IP 数据包的字节数变为 IPB-B；

（2）判断剩余字节数 IPB-B 和 M_PDU 的字节数 MB 的大小；

（3）当剩余字节数 IPB-B≤MB 时，将剩余 IP 数据包放入包区，可以生成 AOS 帧了；

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

（4）当剩余字节数 IPB-B＞MB 时，把字节数为 MB 的 IP 数据从 IP 数据包分割出放入包区内，生成 AOS 帧，此时剩余字节数为 IPB-B-MB，再返回到（2）；

（5）依次执行（2）、（3）和（4），一直到 IP 数据包处理完毕。

IP 数据包分割的流程图如图 3-11 所示。

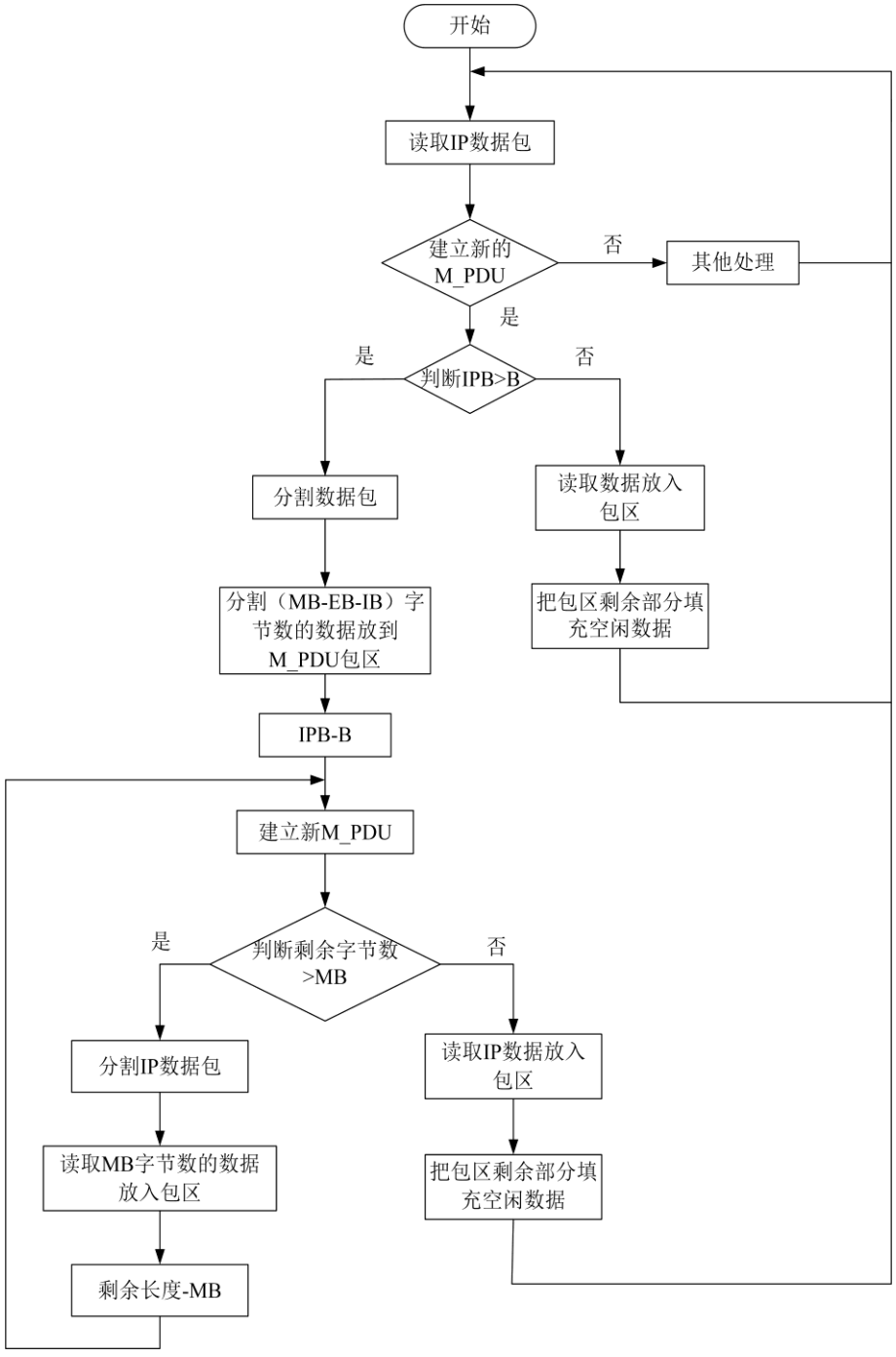

图 3-11 IP 数据包分割处理流程

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

但是当 IP 数据包占据的字节数很少，一个 M_PDU 包区可以放置多个 IP 数据包，所以需要对 IP 数据包拼接操作。

假设任意一个 IP 数据包的字节数为IPB_n，n 个 IP 数据包的总字节数为$X_n = IPB_1 + IPB_2 + \cdots + IPB_n$，M_PDU 的包区字节数为 B=MB-EB-IB，这里需要设置一个 M_PDU 包区的剩余字节数的门限值为 T。具体的拼接流程为：

（1）读取 IP 数据包，我们将包区剩余字节数$B - X_n$和IPB_{n+1}比较大小，当$B - X_n \geqslant IPB_{n+1}$时，直接将第 n 个放入包区中，n 增加 1 后，再比较$B - X_{n+1}$和IPB_{n+2}的大小。

（2）如果$B - X_n < IPB_{n+1}$，我们比较$B - X_n$和门限值 T 的大小，当$B - X_n > T$时，要对进行 IP 数据包分割处理；当$B - X_n \leqslant T$时，直接将包区剩余部分填充空闲数据，生成 AOS 帧。

（3）不断重复（1）和（2）直到 IP 数据处理完毕。处理流程如图 3-12 所示。

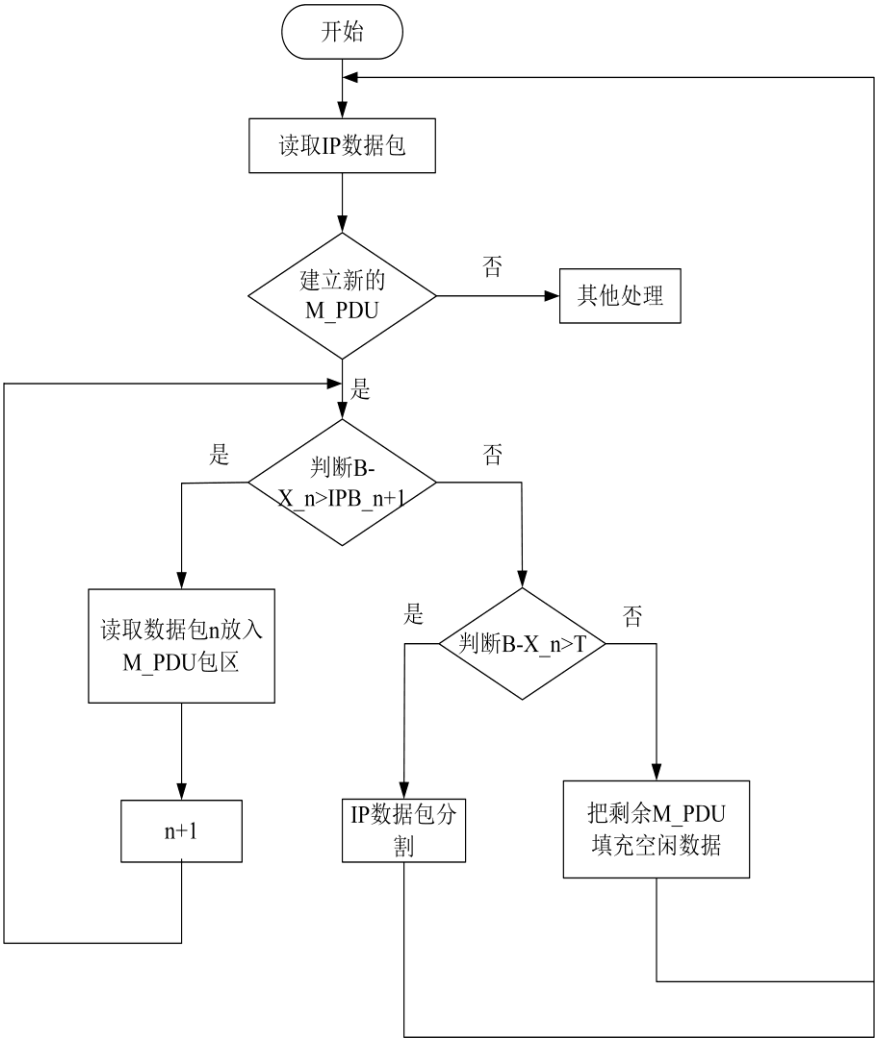

图 3-12 IP 数据包拼接处理流程

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

以上论述的是在发送端对 IP 数据包的切割和拼接的过程，在接收端我们要从 AOS 传输帧提取 IP 数据包，当 AOS 传输帧进入接收端缓存区之后，按照顺序从其中提取 M_PDU，之后再从数据单元中提取 IP 数据包。从 M_PDU 中提取数据包是根据其第一头指针和包内的长度来划分的。第一头指针指出了 M_PDU 中第一个 CCSDS 包的位置（主导头第一个字节位置），如果传输帧 N 的 M_PDU 分组区域中的最后一个分组溢出到同一虚拟信道（N<M）的帧 M 中，则帧 M 中的第一个报头指针忽略分割分组的剩余部分，并指示从帧 M 开始的下一个分组的开始。这里长度区域表示为 len。关于从 M_PDU 中提取 IP 数据包的流程如下：

（1）从缓存区中读取 M_PDU；

（2）读取 M_PDU 中的第一头指针 Fh，如果 Fh 为“所有 1 减去 1”则为空闲数据，丢弃即可；如果 Fh 为“全 1”，则提取 M_PDU 数据域全部数据置于上次剩余数据之后，提取下一个 M_PDU；如果不是以上 2 种情况，则需要提取 Fh 之前的数据，和之前一次的剩余部分数据进行拼接，成为一个完整的 IP 数据包。

（3）检查 M_PDU 剩余的尚未处理的数据 rest，如果 rest 的长度 restLen<4，读取 rest 到缓冲区内，再提取一个 M_PDU；如果 restLen>4，则 Fh 到 Fh+len 为完整的 IP 数据包，之后 Fh 加 1 赋给 len；

（4）重复（1）（2）（3）知道数据被提取完。IP 数据包提取流程如图 3-13 所示。

3.3.3 其他模块处理

在上一个小节中，本文着重论述了在发送端和接收端关于 IP 数据包的处理过程，包括 IP 数据包的切割和拼接，这一小节我们再对发送端的其他处理模块进行一些说明。

在发送端处理完 IP 数据包之后，进入虚拟信道帧生成模块，该模块的主要目的就是形成 AOS 传输帧的基本结构。大致的过程就是对 M_PDU 不加处理直接置于 AOS 帧的数据字段，之后创建 AOS 帧的主帧首部，为每个虚拟信道独立创建一个虚拟信道帧计数，并将其置于主帧首部中去[35]。如果有一个特定虚拟信道的 VC_OCF 服务用户，则用户提供的 OCF_SDU 应放置在操作控制字段中。

之后就是进入了虚拟信道复用模块，该模块的主要功能就是用于复用一个主信道的不同虚拟信道的传输帧，同时虚拟信道复用功能应以管理设置的适当顺序复用从虚拟信道生成功能实例和虚拟信道帧服务用户接收到的传输帧，但是 CCSDS 没有指定用于排序的算法，可以由项目需求根据优先级、释放率和同步时序要求等因素来确定。当仅仅存在一个主信道时，则虚拟信道复用功能将创建一

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

个 OID 传输帧（即空帧），以在释放时间没有可用于传输的有效传输帧的情况下保持数据不断流。OID 传输帧应将其 VCID 设置为“全 1”的默认值。没有必要为 OID 传输帧保持虚拟通道帧计数[36]。

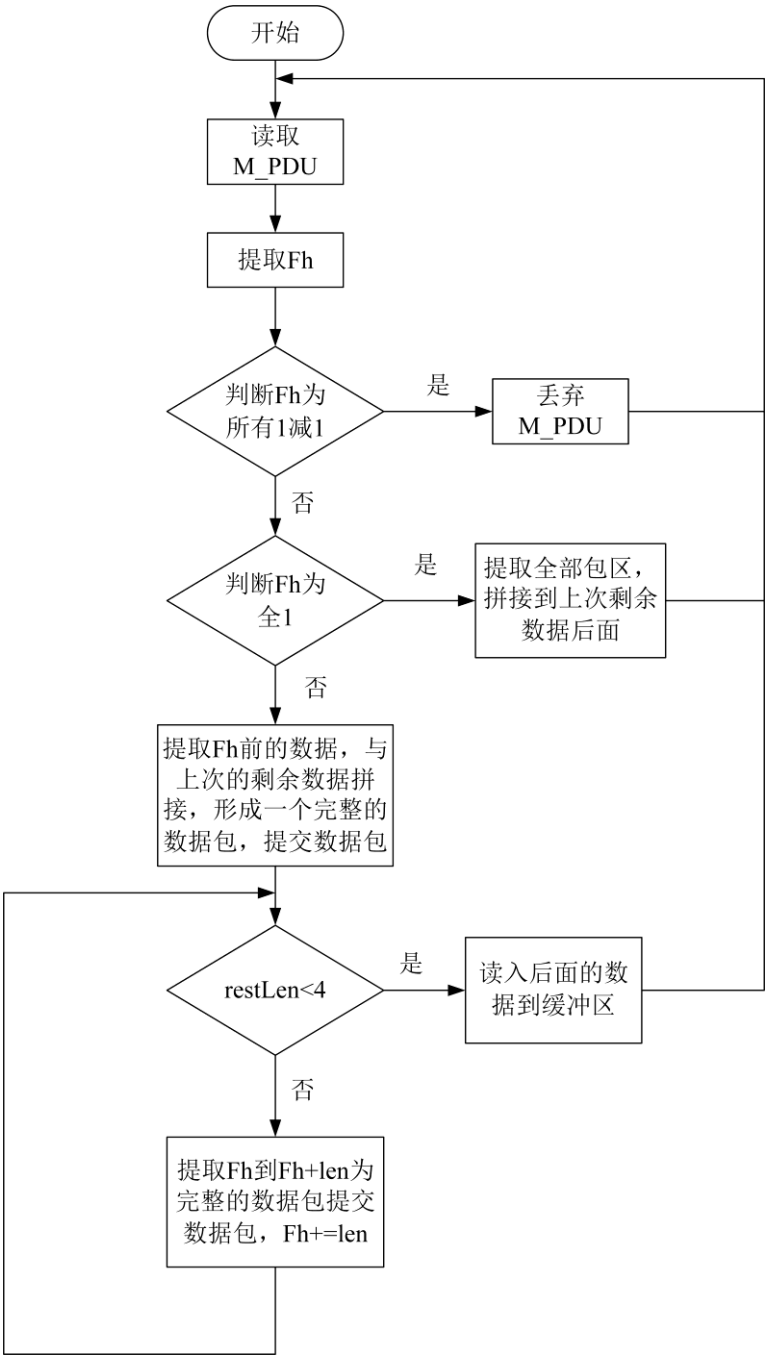

图 3-13 IP 数据包提取处理流程

在虚拟信道复用模块之后是主信道复用模块，该模块主要作用是复用一个物理信道的不同主信道的传输帧，和虚拟信道复用模块一样，需要根据项目的优先

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

级、释放率和同步时序要求确认排序的算法，而且在没有可用于传输的有效帧的情况下，主信道复用功能可以创建一个 OID 传输帧来保持数据不断流，OID 帧可以设置和虚拟信道复用的 OID 相同的参数。

在主信道复用模块之后就是所有帧生成模块即 AOS 帧生成模块，该模块的主要功能是将 IN_SDU（插入业务数据单元）插入到物理信道的传输帧中，它还可以执行 CCSDS 标准推荐的错误控制编码。IN_SDU 应定时以对应于传输帧释放时间的恒定间隔到达，IN_SDU 放入传输帧的插入区，保持字节对齐。如果存在帧头错误控制字段，则应使用 RS 编码过程生成校验位并添加到传输帧主头。

经过以上的数个模块处理之后就可以生成完整的 AOS 传输帧，传输帧经过空间链路到达空间网络，在接收端需要对传输帧进行解析，其过程为数据包的提取、虚拟信道接收、虚拟信道解复用、主信道解复用、AOS 帧接收，图 3-14 描述了数据从图的底部流向顶部，表明了接收端对于 AOS 帧的数据处理。

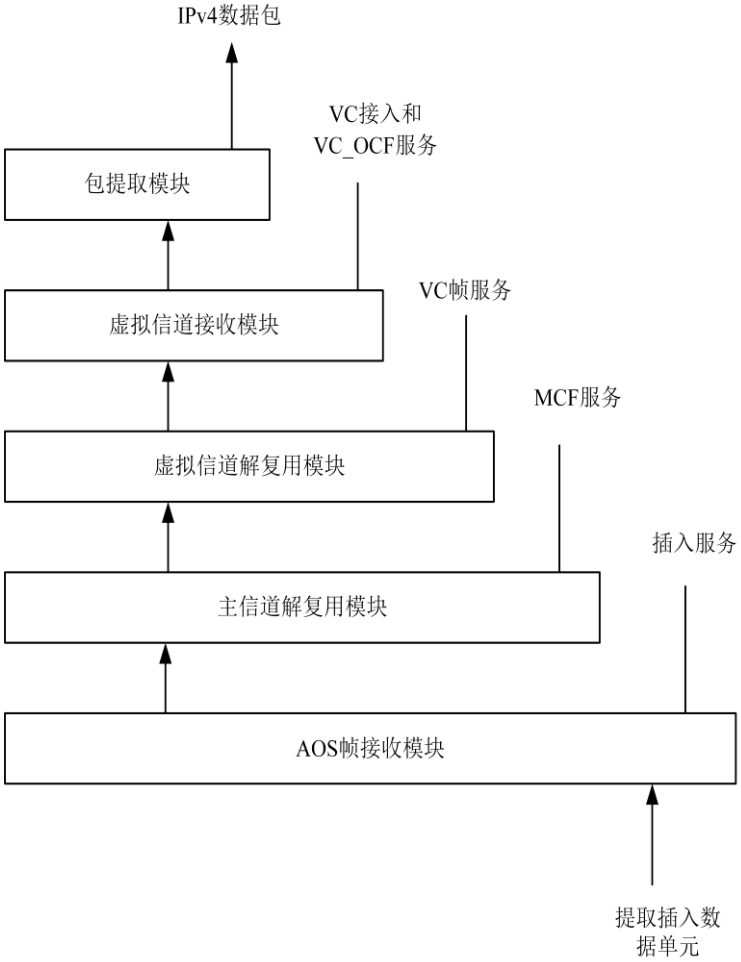

图 3-14 IP 数据包在接收端的处理流程

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.4 移动 IP 技术

从 3.3 节得知 AOS 协议是可以传输 IP 数据包的，然而对于卫星等航天器而言具有很高的移动速度，当地面测控站向卫星发送指令时，卫星此时可能已经运行到别的空间网络中，无法接收到信息，自然也无法得到来自地面网络的 IP 数据包，为了能够在在这种情况下继续使用 AOS 协议，即 IP over CCSDS 的协议转换方式，本文参考了地面网络中类似状况的处理方法，选择使用移动 IP 技术减少链路中断的时间，减小卫星的移动性带来的性能损失。

3.4.1 常规 IP 的局限性

在常规的 IP 协议中，都处于相同子网的节点只要知道对方的 IP 地址，就可以正常的发送和接收数据报，如果其中一个节点移动到其他子网时，需要设置特定主机路由或者是改变节点的 IP 地址才能接收数据报，但是两者都不适用于空间网络，前者会使路由表存储太多特定路由地址而导致内存溢出，后者可能会让节点之间的通信中断。

IP 地址由网络字段、子网字段以及主机字段组成，如果 N 个主机的网络字段和子网字段相同，那么可以认为这 N 个主机都是处于同一条链路上。同一条链路上的主机只需要通过网络字段和子网字段来进行路由选择，因此路由器仅仅保存网络字段和子网字段即可。但是如果需要进行通信两个节点不在同一条链路上，就无法通过网络字段和子网字段发送数据报。为了解决以上 IP 路由的问题，移动 IP 技术产生了。

3.4.2 移动 IP 的设计

移动 IP 技术的目的就是移动节点不需要进行改变 IP 地址，同时不用让通信中断，也可以正常的发送和接收数据报。

移动 IP 中包括 3 类节点：本地代理（Local Agent，LA）、外部代理（External Agent，EA）、移动节点（Mobile Node，MN）。

本地代理是处在 MN 初始时所处的链路上且可以发送代理消息的路由器，外部代理处在其他链路上且可以支持 MN 的路由服务的主机（路由器），移动节点是具有可移动性的通信设备，如卫星。

初始时 MN 的网络字段和子网字段与 LA 所处链路是一致的，当 MN 运动到 EA 所处链路上时会被分配转交地址，转交地址和外部链路的网络字段和子网字段是一致的[37]。

移动 IP 的工作内容包括三个部分：代理发现、转交地址的注册和取消、数据

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

收发

（1）代理发现

LA 和 EA 定期会在本身的链路上发送代理消息，代理消息中包含该代理的 IP 地址，如果 MN 接收到代理消息可以说明其所在的链路存在代理路由器，MN 将本身的 IP 地址网络前缀与代理消息中的网络前缀比较，相同的话则 MN 位于本地链路，否则位于外部链路。若 MN 移动到外部链路时，会从 EA 发布的代理消息中获得一个转交地址。

（2）转交地址的注册

转交地址的注册是 MN 把获得的转交地址告知 LA。MN 将注册请求的数据包发送给 EA，EA 在通过校验后会将请求转发给 LA。LA 也是在校验通过后将 MN 在 LA 获得的地址与转交地址之间的联系缓存到地址表中去，之后就向 EA 发送，EA 会把两个地址之间的联系添加到本地缓存中并且转发请求确认的数据包给 MN。当 MN 回到本地链路上时，会向 LA 发送取消注册的请求，LA 会格式化地址表中 LA 赋予 MN 的地址和转交地址的联系。

（3）数据收发和取消

当 MN 处于外部链路且需要接收 IP 数据包时，LA 会从链路中对该数据包实施抓包的操作，在抓取到该包后将转交地址作为目的地址，对该包进行封装，此时的源地址为 MN 从 LA 获得的地址即初始地址，EA 收到 LA 发送的数据包后，把包实施解封的操作，解封之后可以读取到目的地址为初始地址的数据包，交付路由器进行转发，选择一个合适的接口发送给 MN；当 MN 需要发送数据包时，此数据包的源地址为初始地址，目的地址为接收 MN 数据包的主机 IP 地址，会对该包进行封装，之后包的源地址会变成转交地址，目的地址会变成 EA 地址，EA 收到数据包后会对该包实施解封的操作，之后再封装，把包的源地址变成 EA 地址，目的地址变成 LA 地址，LA 收到数据包后，对其解封装，得到原本的数据包，读取目的地址发送到对应主机所在网络的路由器上。

当与 MN 进行通信的节点距离 MN 较近，而与 LA 相距较远时，此时节点需要经过 LA 所在的网络到达 MN，这样的话不如直接通过 EA 达到 MN 方便，这就是移动 IP 的三路由问题，这时就需要将其路由方式进行改变，由原来通过 LA 的间接路由更改为通过 EA 的直接路由的方式。

移动 IP 的大致流程如下：

①LA 和 EA 发布代理消息，确定 MN 的位置，若 MN 的网络前缀与 LA 相同，则 MN 处于本地链路，若 MN 的网络前缀与 EA 相同，则 MN 处于外部链路；

②若 MN 处于外部链路，获取转交地址，进行注册，当回到本地链路时，取

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

消注册；

③当 MN 处于外部链路且有其他链路的数据包向其发送时，则被 LA 截获，进行封装处理，转发给 EA，再由 EA 交付给 MN；

④当 MN 处于外部链路且需要向其他链路的主机发送数据包时，则由 EA 进行处理之后，转发给 LA，再由 LA 处理发送到目的主机。其工作流程如图 3-15 示。

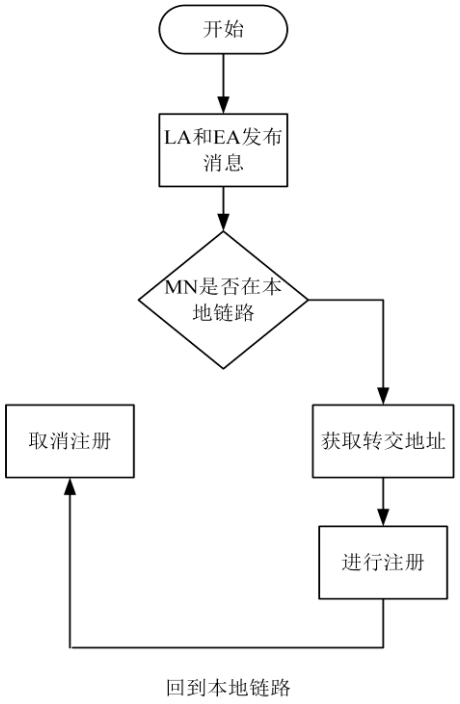

图 3-15 (a) 移动 IP 的注册流程

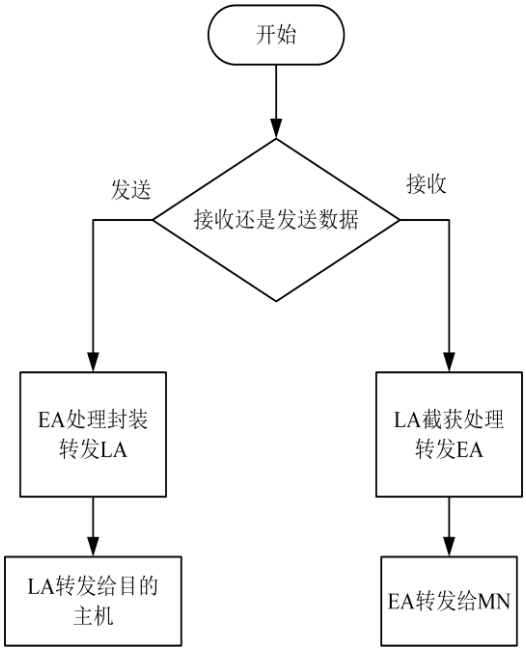

图 3-15(b) 移动 IP 的数据收发

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.5 针对传输层的改进-SCPS-TP 协议

3.4 小节对移动 IP 进行了介绍，解决了航天器移动带来的问题，但是 IP 协议无法保障数据的可靠性，只是将数据包从网络的节点发送另一个节点，而 TCP 协议是应用在地面网络传输层中的协议，是一种可靠的传输协议，是面向字节流的，TCP 协议字节流中的每一个字节都被给予一个编号，接收到数据的一端对接收到的字节都会发送 ACK 应答[38]，TCP 的性能表现会直接影响 IP over CCSDS 的协议转换效率。在地面网络中，传输延时和误码率都很低，所以一般出现数据包丢失的问题，TCP 协议会默认是网络拥塞的缘故，通过计算 ACK 包来启动拥塞控制和流量控制，但是在空间网络中，传输延时长、前向和反向链路不对称，误码率高，使得 TCP 协议的性能受到不好的影响。表 3-1 为空间网络和地面网络的一些区别。

表 3-1 地面网络和空间网络的区别

因素	地面网络	空间网络
RTT（往返时间）	一般为几毫秒或是几十毫秒	几百毫秒
BER（误码率）	10^{-9}~10^{-10}	10^{-5}~10^{-7}
链路特性	一般连续	经常中断
数据丢失原因	网络拥塞	链路中断、误码率高、RTT 高等
上下行速率比	1:1	1:10~1:1500

SCPS-TP 协议是 CCSDS 针对空间通信推出的传输层协议，根据航天任务的不同需求，SCPS-TP 协议一共具有 3 种服务模式来满足，分别是完全可靠、高效可高以及最低可靠，其中最低可靠没有连接管理的机制，是使用 UDP 协议来完成，高效可靠依赖 SCPS-TP 协议，而完全可靠的传输方式只能由 TCP 协议完成[39]。SCPS-TP 协议是对 TCP 首部的选项部分进行扩展，完成 TCP 协议的改进，其主要改进部分如下：

（1）选择性确认（Selective Acknowledgment，SACK）机制

在发送一段连续的字节流时，如果部分序列没有达到接收端，TCP 协议会重新发送整段序列，导致传输效率低下，SCPS-TP 协议会有选择的进行确认，接收端通知发送端所有已经成功到达的分段，因此发送端只需要重新传输实际丢失的分段。

（2）选择性否认确认（Selective Negative Acknowledgement，SNACK）机制

SANCK 机制是 SACK 的基础之上实现的，其基本思想是将在传输过程的错误

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

信息封装在一个 SANCK 包中，传送给发送端，这样可以避免频繁的重传，最大限度利用空间链路。

（3）拥塞控制 Vegas 算法

Vegas 算法主要是通过比较期望的数据速率和实际速率的差异来调整拥塞窗口的大小[40]，相比与 TCP Reno 算法，Vegas 算法能够获得 40%~69%的吞吐量的提升，以及降低 20%~49%的丢包率，具体内容将在 3.5.4 节叙述。

除了上面 3 个主要改进之外，SCPS-TP 协议还在首部的选项部分对窗口值进行了修改，使滑动窗口变大，增加了上下行链路的下载速度。

3.5.1 空间通信中存在的一些问题

（1）传输延时长

地面网络一般是有线信道，信号的 RTT（往返时间）很短，一般为几毫秒或是几十毫秒，而空间网络的 RTT 为地面网络的十几倍，静止轨道的航天器传输延时一般为 250ms，地面站处理数据的时间为 50ms，所以 RTT 可以达到 550ms，这对 TCP 协议而言会严重影响其性能。

传输延时长首先会影响拥塞窗口的增长。如 2.1.4 节所述 TCP 协议开始通信的算法是慢启动，在慢启动阶段是通过计算到达的 ACK 应答包来实现拥塞窗口的缓慢增大的[41]，一旦通信双方的传输延时过长，就会引起 ACK 包的传输速度增长缓慢，导致拥塞窗口不能按照算法所期待的方式快速变大，也会使得拥塞控制算法长时间停留在慢启动阶段，发送的数据很少，空间链路的信道利用率不够高，浪费了链路带宽。

延时长还会影响 TCP 数据重传。在一般通信中，数据会出现丢失的情况，当发送数据的一方在 RTO 失效后还没有收到 ACK 包，TCP 就会开启重传机制，但在空间网络中因为长延时，一般设置 RTO 的值都会很大，所以重传数据会需要很长时间。而且 TCP 协议默认丢包的原因是网络拥塞，一旦出现丢包的情况，TCP 协议就会缩小拥塞窗口，降低数据速率，回到慢启动的状态，这样需要很长时间链路带宽才能得到充分利用。

除此之外，延时长还会影响 TCP 协议的吞吐量，吞吐量与 RTT 之间成反比关系，关系式如公式 3-1 所示。

$$Throughput = WIN/RTT \tag{3-1}$$

其中 WIN 指的是发射窗口。当 RTT 为 550ms，发射窗口的最大值为 65535 字节，那么按照公式 3-1 计算，吞吐量不会超过 1Mbps。

（2）前向和反向链路的不对称

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

在 TCP 通信中，一般将发送方到接收方的链路称为前向链路，而接收方到发送方的链路成为反向链路。为了信号的发射功率足够大，空间网络的前向链路带宽大于反向链路，前向链路用来发射数据报，反向链路是用来发射对数据包的应答包，因为前向和反向链路的不对称，反向链路的带宽小，应答包不能在规定时间达到，导致拥塞窗口的增大速率变小，影响 TCP 协议吞吐量。而且应答包不能及时达到，也会引起 RTO 超时，使得拥塞算法进入慢启动或是避免的阶段，缓慢增大拥塞窗口，空间链路带宽利用率很低。除了以上之外，链路的不对称会使部分应答包丢失，数据在重传的时候会出现不按顺序的问题，导致数据突发。

（3）误码率高

一般地面通信的误码率都是在 10^{-9} 左右，空间通信的误码率比地面高好几个数量级，可以达到 10^{-5}~10^{-7}，如前向修正算法对地面误码有一定的帮助，但是对空间通信这样的高误码率是没有办法的。而且空间信道受到天气、温度、湿度、多径等影响，误码率还会进一步提高。由于误码率造成的数据丢失，会导致发射窗口缩小，数据发送速率减小。而且当拥塞窗口出现多个数据包丢失的情况，TCP 拥塞算法会将 CWND 值减小一半，如果数据包继续丢失，那么 CWND 值继续减小，会影响到发射窗口的大小，根据公式 3-1，TCP 协议的吞吐量也会下降，通信双方将要等待很长时间才能将传输状态恢复到正常的水平。

3.5.2 选择性确认机制设计

在地面网络中，TCP 协议一般采用累计确认的方式，即对于一段连续到达的字节流，TCP 协议只会对最后一个字节发出 ACK，表明小于该序列号的所有字节已经全部接收到了，如果字节流没有按照顺序达到，其中一部分数据包丢失，引发快速重传机制，重新发送全部的数据包，造成数据的重复发送，地面网络的链路质量好，RTT 短，误码率低，所以数据重新发送的影响小，而空间网络 RTT 长，数据重新发送所需的时间长，并且 SCTP-TP 协议扩大窗口值，重新发送数据比较多，所以窗口中数据出错的概率大大提升，又因为空间误码率很高，窗口中可能会出现多个数据包丢失的情况。

SACK 机制就是针对这一情况而设计的，接收端可以对非连续的字节流接收，并将把已经接收到的字节流序列号范围和 ACK 确认信息封装在一个 SACK 包中，发送端在收到 SACK 包后就会知道有一段序列的字节流丢失了，发送端只用发送丢失的数据包即可，不用将重复的数据包再次发送，节省了空间链路的带宽资源，也提高了传输效率。图 3-6 为 TCP 协议的重传机制示意图，未使用 SACK 机制。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

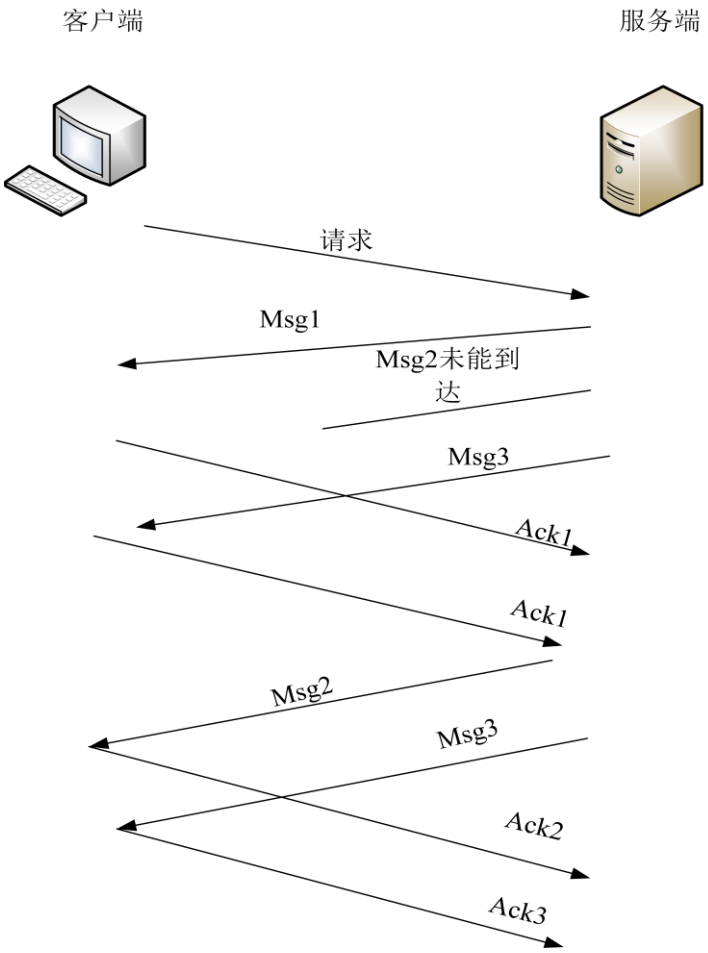

图 3-16　TCP 的重传机制

如图 3-16 所示，客户端向服务端请求传输消息，共有三个消息分段。Msg1 到达客户端给出响应 Ack1，Msg2 在传送过程中丢失，Msg3 正常到达客户端，由于 TCP 采取累计确认的机制，所以在发出 Ack1 后，客户端准备接收 Msg2，但是该消息丢失，只接收到 Msg3，客户端继续发送重复确认 Ack1。服务端在收到 2 个 Ack1 后，将 Msg2 和 Msg3 进行重传，Msg3 被重复接收。

图 3-17 所示的使用 SACK 机制的 SCPS-TP 重传。客户端在接收到 Msg3 后，给服务端发送一个重复的 Ack1 和 SACK3，表明已经接收到 Msg3，服务端在收到该应答包后，可以确认除了 Msg2 之外的数据包均已正常收到，只需要重新传输 Msg2。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

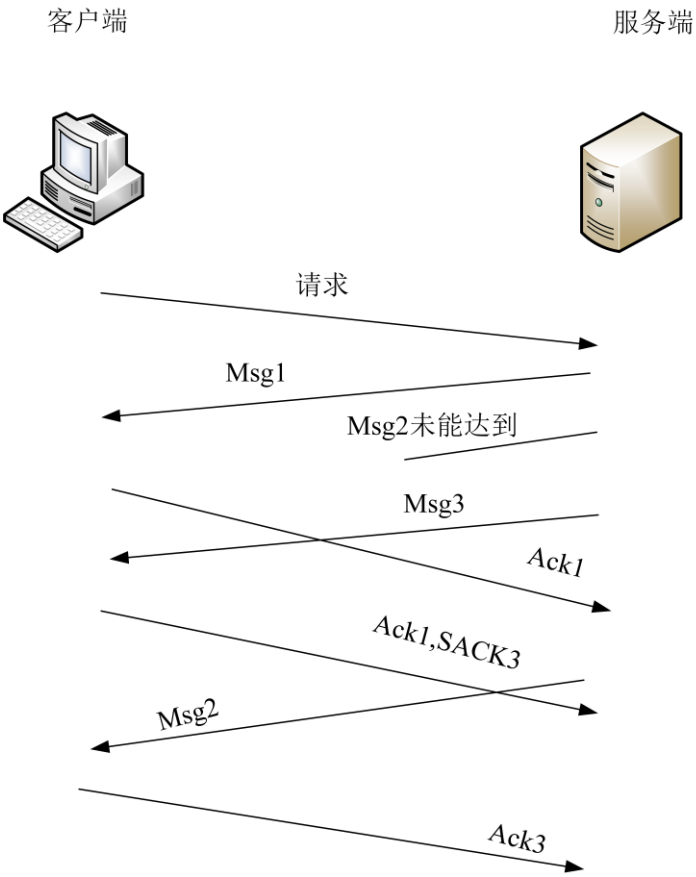

图 3-17 SCPS-TP 协议的 SACK 机制

SACK 内容通常由 2 个部分组成，一是 Sack-Permitted Option（允许选择），允许选择包括类型值（kind），一般为 4，以及长度值（length），一般为 2，长度值的单位为字节，所以允许选择的长度为 2 个字节，类型值和长度值各占一个字节。二是详细的 SACK 内容。SACK 允许选择如图 3-18 所示。

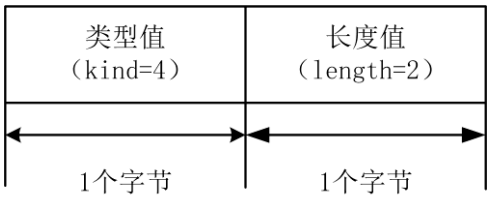

图 3-18 SACK 允许选择

如果通信双方想要使用 SACK 机制，必须在建立连接的前两个 TCP 包中，明确指定允许选择，该允许选择是在 TCP 首部的扩展选择实现。

SACK 具体的内容也包括 2 个部分，其一是类型值（kind=5）和长度值，长度值由 SACK 内容确定，一般不超过 40 个字节，二是具体的选择确认字节流信息。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

而且 SACK 选项一般和时间戳一起使用，因此 SACK 具体信息一般不超过 3 组不连续的字节流。SACK 选项的具体内容如图 3-19 所示。

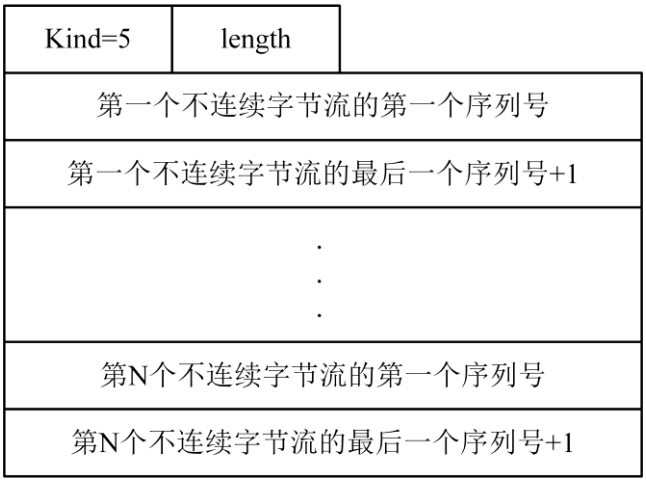

图 3-19 SACK 选项内容

图 3-20 为 SACK 在 OMNeT++中（网络仿真软件，会在第四章详细介绍）控制台的主要日志信息。

```
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Inserted SACK entry: [53752..54288)
INFO (TcpConnection)ClientServer.server.tcp.conn-5:1 SACK(s) added to header:
INFO (TcpConnection)ClientServer.server.tcp.conn-5:0. SACK: [53752..54288)
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Sending: .1000 > .1025: ack 53216 win 7504 options NOP NOP SACK
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Staying in state: ESTABLISHED (no FSM event)
```

图 3-20(a) SACK 机制日志信息

图 3-20(a)表示开启 SACK 机制，将 SACK 信息添加到 TCP 的首部，对于序号 53752-54288 字节流的确认。

```
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:rexmitQ: [53216..55360) enqueueSentData [53216..53752)
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Sending: .1025 > .1000: [53216..53732) (l=516) ack 50012 win 7504
```

图 3-20(b) 客户端重发信息

图 3-20(b)表示客户端重新发送序号 53216-53752 的字节流。

```
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Connection 192.168.1.2:1000 to 192.168.1.1:1025  on socketId=5  in ESTABLISHED
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Seg arrived: .1025 > .1000: [53216..53752) (l=536) ack 50012 win 7504
```

图 3-20(c) 信息达到服务端

图 3-20(c)表示序号 53216-53752 的字节流达到服务端。

3.5.3 SNACK 机制设计

SNACK 机制是 SCPS-TP 协议针对 RTT 长和前反向链路不对称的问题提出的

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

改进方法，地面上的 TCP 协议发送的 ACK 包中往往只有一个窗口的一个错误字段，若是在窗口内包含了多个错误字段，数据的接收端就需要发送多个带有错误字段的 ACK 包[42]，地面网络延时低，TCP 协议性能几乎不受影响，但是空间网络延时高，且前向和反向链路极不对称，这对 TCP 协议的传输效率产生非常不好的影响。而 SANCK 机制会将多个错误信息打包一起发给数据发送端，就不用发送多个 ACK 包了，这样提高了空间通信反向链路的利用率。

但是如果出现多个窗口都都有很多错误信息，那么一个 SNACK 包还不足以覆盖所有错误，因为 TCP 首部选项的长度不会超过 40 字节，因此接收端还是需要发送多个 SNACK 包才能对所有窗口信息的确认。

```
kind=TCPOPTION_SELECTIVE_NEGATIVE_ACKNOWLEDGEMENTS;
length=8;//2 + 6
unsigned int holeoffset;
unsigned int holeoffset_size;
unsigned int bit_vector;
```

图 3-21 OMNET 中 SNACK 选项设计

从图 3-21 中可以看出，SNACK 选项由 5 个部分组成[43]：

（1） 选项类型

选项类型字段为 SNACK 选项的必选项，处于选项字段的首个字节位置，该字段值为 21。

（2） 选项长度

选项长度字段为 SNACK 选项的必选项，处于类型字段后的首个字节位置。选项长度字段应包含选项使用的总字节数：

a）如果不使用 SNACK 比特向量位，那么选项总字节数为 6；

b）如果使用 SNACK 比特向量位，那么选项的总字节数是 6 加上 SNACK 比特向量位的字节数[44]；

SNACK 比特向量位的长度是一个具体问题，可能因为 SNACK 选项的不同而不同。

（3）空洞偏移量（Hole1 Offset）

空洞偏移量字段为 SNACK 选项的必选项，处于在选项长度字段后的首个和第二个字节的位置。该字段表明第一个错误空洞与确认号之间的偏移。偏移值以 MSS（最大段的大小）为单位，它的值为偏移序列字段值减去确认字段值，再把这个差值除以最大段中包含的用户字节数（即组成 1 个 MSS 的字节数）来计算[45]，如公式 3-2 所示。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$Hole1\ Offset = \frac{Offset_{seq} - ACK_{seq}}{MSS} \tag{3-2}$$

如果公式无法除尽，那么将余数添加到数据空洞的大小中。

（4）空洞大小（Hole1 Size）

空洞大小字段为 SNACK 选项的必选项，处于偏移量字段后的首个和第二个字节的位置。该字段表明选择中第一个错误空洞的大小。空洞大小字段以 MSS 为单位，通过数据空洞的长度除以最大段中包含的用户字节数，如公式 3-3 所示。

$$Hole1\ Size = \frac{Hole1\ Offset}{MSS} \tag{3-3}$$

如果公式除不尽，那么结果大小向上取整。

（5）SNACK 比特向量（SNACK Bit-Vector）

SNACK 比特向量字段是非必选的，如果 SNACK 包含该字段，那么处于空洞大小字段之后的首个字节位置。该字段表示了在接收端空洞大小字段之后中丢失数据的内容：

a）SNACK 比特向量字段中每个“0”表示接收端重新排列的队列中相应 MSS 大小块中的缺失数据，“1”则表示对应 MSS 块大小的数据正确接收；

b）如果有必要，应将“0”添加到最后一个“1”的右侧，来保证 SNACK 比特向量位在数据范围上终结，即字节能够对齐。

如果想要使用 SNACK 功能，那么必须在 SYN=1 的分段中明确表示支持 SCPS SNACK 选项，在空间网络中，TCP 会出现无序传输的情况，一般都会采取延迟发送 SNACK 包以达到减少乱序数据分段导致的不好影响，但是具体要等待多长时间才发送 SNACK 包，视分段丢失的实际状况而定，除非不会出现乱序分段，否则 SNACK 延迟发送都是有益的。SCPS-TP 协议会确保 SNACK 选项的数据包有较长的时间从接收端送往发送端，并在下一个 SNACK 包达到发送端之前，发送端进行数据重传。在通信环境极度恶劣的状况下，可能会出现 SANCK 选项的数据包消失在空间链路中，在这种情况下，是需要为之前的 SANCK 选项中错误 Hole1 传输一个 SNACK 数据包。

数据的发送端在收到 SNACK 选项后，应重新传输填充数据空洞的所需要的所有分段，发送端可以通过首先重新传输与 Hole1 对应的分段，如果存在 SNACK 比特向量字段，那么发送端可以左移比特向量位，直到最后一个“1”被移出。从而重新传输与该过程中遇到的每个“0”对应的分段，下面用一个例子来说明 SNACK 的传输过程。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

如图 3-22 显示了在接收端的数据队列，其中*RCV.NXT*指的是下一个预期的序列号，表示该序号之前的所有消息已经没有任何错误地被接收端收到，沿着箭头方向的每两个竖线之间的间隔代表着 1 个 MSS。在这个数据队列中存在 3 个数据空洞，表示没有正确接收到的片段，它们的位置分别是*RCV.NXT*处之后的 3 个数据空洞，然后是 4 个正确接收的片段；1 个数据空洞，后面跟着 2 个正确接收的片段；然后是 2 个数据空洞，1 个正确接收的片段。

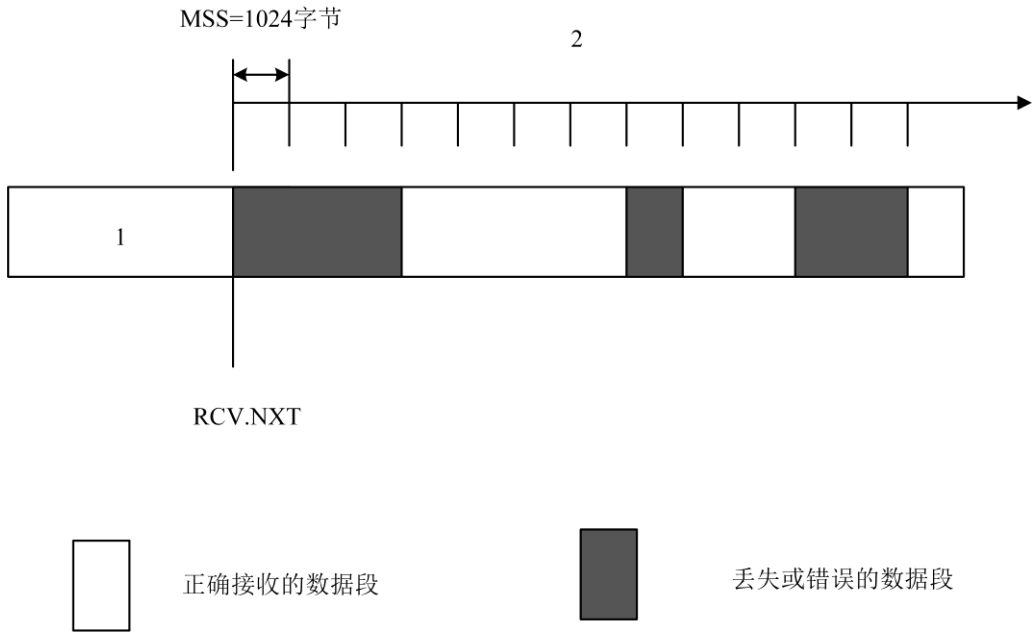

图 3-22 SNACK 数据序列

图 3-22 中 1 代表已经确认的序列号，2 表示接收到的数据但是还没有确认。

若是通过 SNACK 选项中的比特向量字段来表示图 3-22 的数据序列，其结果如图 3-23 所示。其中选项类型（Type）为 21，总的字节数（Length）为 8（6 个字节大小的必选项字段加上 2 个字节大小的比特向量字段），空洞偏移（Hole1 Offset）量为 0，因为已经确认的序列号后面就是数据空洞，之后是 4 个正确接收的片段，则表示为“1111”，1 个数据空洞，表示为“0”，2 个正确接收的片段，表示为“11”，2 个数据空洞表示为“00”，1 个正确接收的片段，表示为“1”，后面再补 6 个“0”，所以比特向量字段为“11110110 01000000”。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

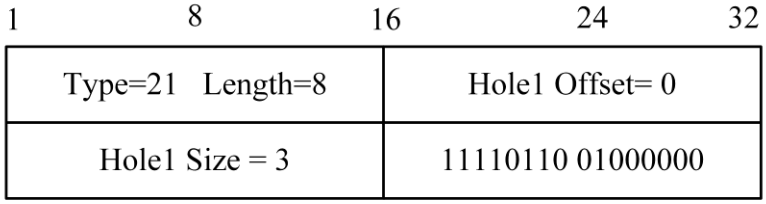

图 3-23 SNACK 比特向量表示

如果不使用 SNACK 比特向量字段，那么三个 SNACK 包才能表示完整数据丢失或是错误的信息，如图 3-24 所示。在空间链路带宽资源有限的情况下，使用比特向量字段能够提升传输效率。

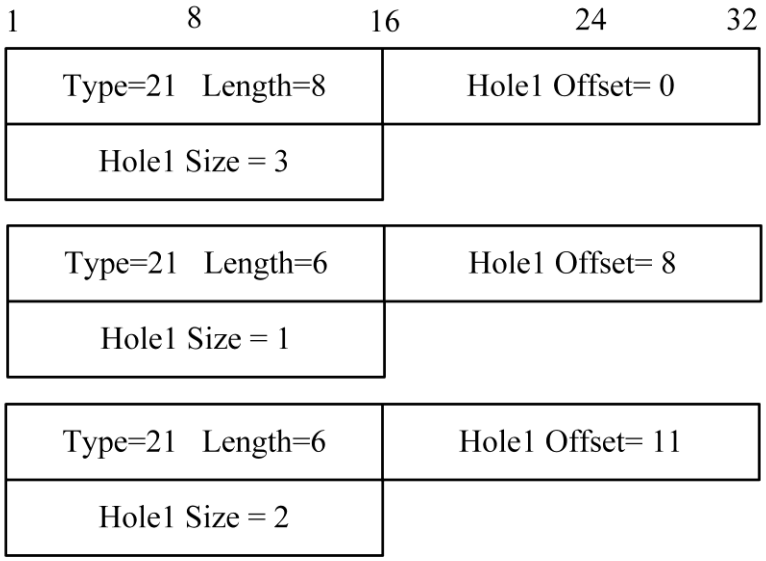

图 3-24 不使用比特向量字段的 SANCK 选项

图 3-25 为 OMNeT++中 SNACK 机制的仿真流程，如果没有数据丢失的情况，则直接发送 ACK 包即可，否则判断是不是新的丢失数据，若不是，进入 SNACK 机制。本文使用了定时器来控制 SNACK 的延迟发送。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

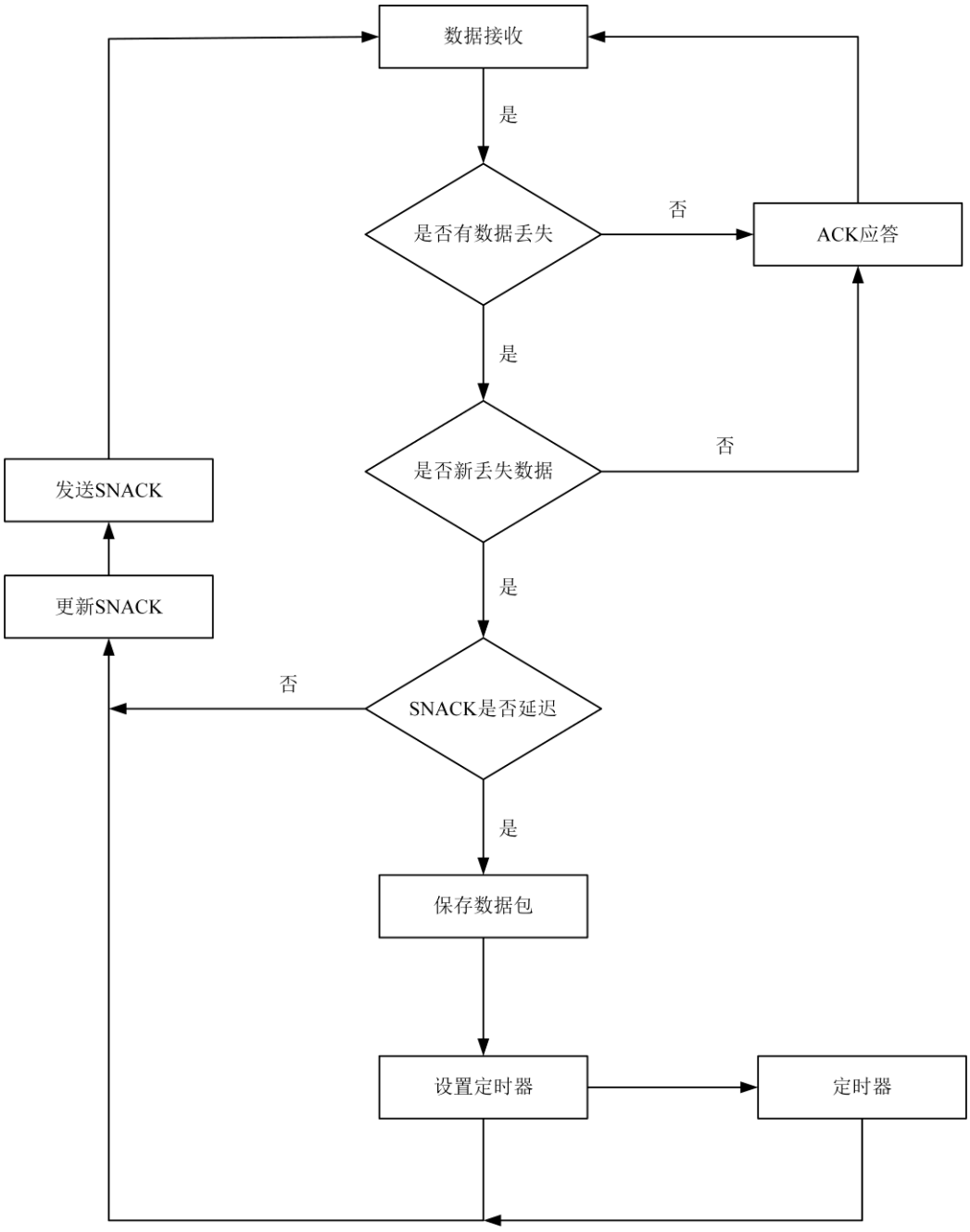

图 3-25 SANCK 机制仿真流程

本文在 OMNeT++中对 SNACK 机制进行仿真，下面是其主要的日志信息。

```
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Connection 192.168.1.2:1000 to 192.168.1.1:1025  on socketId=5  in ESTABLISHED
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Seg arrived: .1025 > .1000: [54288..54824) (l=536) ack 50012 win 7504
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:TCB: snd_una=50012 snd_nxt=50012 snd_max=50012 snd_wnd=7504 rcv_nxt=53216 rcv_wnd=7504 snd_cwnd=536
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Processing ACK in a data transfer state
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Processing segment text in a data transfer state
WARN (TcpConnection)ClientServer.server.tcp.conn-5:check_rcv_nxt=53752n=lrecord succeedl
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Out-of-order segment, sending immediate ACK
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Sending: .1000 > .1025: ack 53216 win 7504
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Staying in state: ESTABLISHED (no FSM event)
```

图 3-26(a) SNACK 日志信息

图 3-26(a)表示接收端接收到序列号为丢失数据包之后的数据包（乱序数据

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

包），会将乱序数据包写入 SNACK 机制中，此时发送端会依然正常发送数据。

```
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Connection 192.168.1.2:1000 to 192.168.1.1:1025  on socketId=5  in ESTABLISHED
INFO (TcpConnection)ClientServer.server.tcp.conn-5:Seg arrived: .1025 > .1000: [54288..54824) (l=536) ack 50012 win 7504
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:TCB: snd_una=50012 snd_nxt=50012 snd_max=50012 snd_wnd=7504 rcv_nxt=54288 rcv_wnd=7504 snd_cwnd
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Processing ACK in a data transfer state
DETAIL (TcpConnection)ClientServer.server.tcp.conn-5:Processing segment text in a data transfer state
INFO (TcpConnection)ClientServer.server.tcp.conn-5:rcv_nxt changed to 55360, (delayed ACK disabled) sending ACK now
WARN (TcpConnection)ClientServer.server.tcp.conn-5:write snack to headerrecord[i]=4Snack holeoffset=2,holeoffset size=1,bit vector=8i=1sum=4some of
```

图 3-26(b) SNACK 日志信息

一段时间后接收端在 ACK 包中写入 SNACK 信息，告知发送端哪些数据已经收到，不需要重传。

```
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:Connection 192.168.1.1:1025 to 192.168.1.2:1000  on socketId=4  in ESTABLISHED
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Seg arrived: .1000 > .1025: ack 55360 win 7504 options Snack
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:TCB: snd_una=53216 snd_nxt=56968 snd_max=56968 snd_wnd=7504 rcv_nxt=50012 rcv_wnd=7504
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Tcp Header Option(s) received:
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:Option type 21 (Snack), length 8
WARN (TcpConnection)ClientServer.client1.tcp.conn-4:snack received SNACK received:
WARN (TcpConnection)ClientServer.client1.tcp.conn-4:holeoffset=0, holeoffsetsize=1  bitvector=3 snackNo=53752
WARN (TcpConnection)ClientServer.client1.tcp.conn-4: first=536
WARN (TcpConnection)ClientServer.client1.tcp.conn-4:[53752,54288) have been acked snackNo=54288
WARN (TcpConnection)ClientServer.client1.tcp.conn-4:[54288,54824) have been acked snackNo=54824
WARN (TcpConnection)ClientServer.client1.tcp.conn-4:[54824,55360) have been acked
DETAIL (TcpConnection)ClientServer.client1.tcp.conn-4:Processing ACK in a data transfer state
INFO (TcpConnection)ClientServer.client1.tcp.conn-4:Updating send window from segment: new wnd=7504
```

图 3-26(c) SNACK 日志信息

发送端接收到 SNACK 信息，读取 TCP 首部，对丢失的数据包进行重传。

3.5.4 Vegas 拥塞算法设计

Vegas 是 SCPS-TP 协议默认的拥塞控制算法，是用来解决因为网络拥塞而导致的数据丢包的问题。Vegas 算法的核心思想是通过测量 RTT 来计算实际吞吐量和期望吞吐量之间的差值，来控制拥塞窗口的大小和调整发送数据的速度[46]，在网络拥塞发生之前减少数据丢包，提升空间网络的传输效率。Vegas 算法的具体实现设计如下所示：

首先需要测量通信双方所有的 RTT，并取其中的最小值，一般 RTT 最小值为 TCP 建立连接后发送第一个数据分段，即在路由器的缓存区队列不为空之前。这个最小值记作 $RTT_{\min}$。本文将期望吞吐量记作 $Throughput_{\exp}$，实际吞吐量记作 $Throughput_{act}$，当前的 RTT 记作 RTT_{now}。

$$Throughput_{\exp} = \frac{CWND}{RTT_{\min}} \tag{3-4}$$

$$Throughput_{act} = \frac{CWND}{RTT_{now}} \tag{3-5}$$

$Throughput_{\exp}$ 和 $Throughput_{act}$ 之间的差值为：

$$\Delta = (Throughput_{\exp} - Throughput_{act}) \cdot RTT_{\min} \tag{3-6}$$

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

再使用两个门限值α和β与Δ进行比较，一般将这两个门限值分别设为 1 和 3。若$\Delta<\alpha$，则说明当前网络并不拥堵，可以适当增大 CWND 值；若$\Delta>\beta$，则说明当前网络拥塞状况比较严重，接收缓存区可能已经溢出了，需要减小 CWND 值，降低数据发送速率；若是$\alpha<\Delta<\beta$，则说明当前网络状况比较稳定，可以保持现在的 CWND 值不变。总结以上成为公式如下：

$$CWND(t+\delta t)=\begin{cases} CWND(t)+1 & \Delta<\alpha \\ CWND(t) & \alpha<\Delta<\beta \\ CWND(t)-1 & \Delta>\beta \end{cases} \tag{3-7}$$

Vegas 算法还对重传机制进行了改进。传统 TCP 协议在数据丢失之后会给发送重复确认，当缓存区收到 3 个同样 ACK 包后，发送端才能确定在链路中数据丢失了，进而进行数据包的重传，同时也会将拥塞窗口的值缩减为原本的 1/2，按照线性的方式增长，这种依赖 ACK 的方式在空间网络中很不好（RTT 长和误码率高）。在 Vegas 算法中，发送端不需要等待 3 个同样的 ACK 包，只需要接收到 2 个形同的确认，就可以知道发生了丢包，拥塞控制就不用进入避免阶段而是直接进入到快速恢复的阶段，提升了在空间传输的效率以及带宽利用率。

Vegas 算法适合空间网络状况的慢启动。在地面的 TCP 协议中，拥塞窗口的增长缘于 ACK，发送端每进行一次的确认收到之后，拥塞窗口的值就会扩大为之前的两倍，这种指数增长的速度会很快使拥塞控制从慢启动到达拥塞避免，这种快速增长对于空间网络的缓存区很不利，大量的数据包可能会很快充满缓存，导致网络拥堵和数据丢失。Vegas 针对这一问题增长拥塞窗口的速度进行了减慢，Vegas 每间隔一个 RTT 才增加窗口值。因为空间传输的 RTT 较长，所以缓解了 CWND 增长过快的隐患。

Vegas 算法的仿真伪代码如下：

```
//更新 baseRTT 即最小 RTT
state->v_baseRTT = newRTT;
// 计算实际吞吐量
uint32_t actual = rttLen / newRTT;
// 计算期望吞吐量
expected = (uint32_t)((state->snd_nxt - firstSeqAcked) + std::min(state->snd_mss - acked, (uint32_t)0)) / state->v_baseRTT;
// 计算实际和期望吞吐量差值
uint32_t diff = (uint32_t)((expected - actual) * SIMTIME_DBL(state->v_baseRTT)
```


(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
+ 0.5);
    // 慢启动
    if (diff > 1 * state->snd_mss) { // gamma
    }
    //差值大于 beta 时，拥塞窗口缩小
    if (diff > 4 * state->snd_mss) { // beta
    state->v_incr = -state->snd_mss;
    }
    //差值小于 alpha 时，拥塞窗口增大
    else if (diff < 2 * state->snd_mss) { // alpha
    state->v_incr = state->snd_mss;
    }
```

3.6 数据优先级

在前文中，本文设计了天地互联的协议转换，主要针对了传输层、网络层和数据链路层的协议进行描述，并提出了一些改进的措施，整个天地互联网络的基本组织框架形成，本小节主要考虑到空间网络中一些重要事件的发生，如航天员的身体健康出现突发状况、卫星受外界干扰偏离轨道等，这些突发事件的重要数据信息必须要在第一时间传送到地面站，让地面站能够有足够的时间做出反应，采取相应措施，因此不同的数据在空间节点中需要定义为不同的优先级，并且将优先级的信息封装在传输帧中，让优先级高的数据先发送。

3.6.1 数据优先级的划分

在空间网络中一般有 3 种类型的数据：数据量最多的测控数据、空闲数据和重要且紧急的事件数据。其中优先级最低的就是空闲数据，这种数据并不包含任何有用的信息；比空闲数据优先级高一级的是测控数据，这种数据一般是卫星的测距码等；优先级最高的数据是重要且紧急的事件所携带的数据。

本文使用了 IEEE802.11 协议中的 MAC（Media Access Control）协议来承载表示数据优先级的任务，在 IEEE802.11 协议中 MAC 帧主要由 4 个部分组成的，分别是帧控制字段（Frame Control）、持续时间字段（Duration）、地址字段（Address）、帧体字段（Frame Body）[47]。本文在帧控制字段和持续事件字段之间插入一个 3 字节的标记字段（tag）用来定义数据的优先级。在 tag 字段中前 2 个字节为 0x1200，如果节点收到的数据帧的 tag 字段值为 0x1200，则说明该数据帧的数据是定义有

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

优先级的数据，否则说明该数据帧的数据为一般的数据。图 3-27 为在 MAC 帧中插入 tag 标识字段。

本文使用 2 个比特的数据来标识数据的优先级，如果 tag=00，说明数据帧中的数据的优先级是最低的，在信道忙碌时，可以选择不可靠的方式传输该数据帧；如果 tag=01，说明数据帧的数据的优先级介于最低和最高之间，该数据帧携带了一般的测控数据；如果 tag=10，说明数据帧的数据的优先级是最高的，在多个节点需要使用信道发送数据是，应该优先发送该数据帧。

当网络中的一个节点接收到来自另一个节点的带有 tag 标识的数据时，先是检查 tag 字段的值，根据 tag 值把接收数据放置到缓存队列的对应位置，如果 tag 值表示的优先级最高，那么把接收数据放在队列的最前端；在节点将接收的数据发送出去时，按照接收队列里数据的位置，将数据存入发送队列中，依照队列顺序发送数据即可。

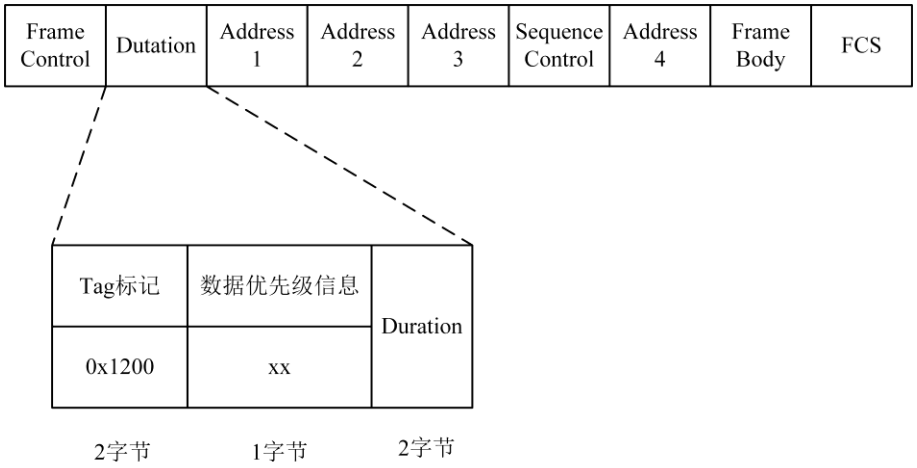

图 3-27 在 MAC 帧中插入 tag 标识字段

3.6.2 优先级数据的发送

在 MAC 协议中有 2 个重要的参数：竞争窗口（Competition window value，CV）和等待时间。在节点第一次发送数据时，初始的 CV 值为所有 CV 值中最小的一个，即$CV_{\min}$，如果有其他节点也要发送数据那么就会发生碰撞，CV 值就会增大一倍，节点会进入等待状态，若是等待时间结束仍然发生碰撞，则 CV 值继续翻倍，直到增大到 CV 值的极限即$CV_{\max}$，当节点的数据发送成功时，CV 值设置为最小的值。其中等待时间T_{wait}和 CV 的关系如公式 3-8 所示。

$$T_{wait} = Random(\ ) \times SlotTime \tag{3-8}$$

其中 Random()是一个随机函数，它的值分布在$[0, CV-1]$中，而 CV 的值是在

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$[CV_{\min}, CV_{\max}]$中，SlotTime 为 2 个数据帧发送之间的时隙。从公式 3-8 中可以得知，如果 CV 的值越小则等待时间越小，数据就能够优先进入信道。

为了确保优先级高的数据能够先进入信道发送，本文根据数据的优先级定义了不同大小的 CV，分别是CV_{00}、CV_{01}和CV_{10}对应了 tag 值为 00，01 和 10 的优先级，它们的初始大小关系如式 3-9 所示。

$$CV_{00\min} > CV_{01\min} > CV_{10\min} \tag{3-9}$$

当网络节点的 tag=10 时，它的竞争窗口值为$CV_{10\min}$，是 3 个 CV 中最小的一个，相较于其余 2 个优先级的数据，该节点的等待时间大概率较小从而能够更快进入信道开始发送。需要注意的是，即使节点发送数据时信道没有其他节点的数据，此时仍然需要等候些许时间，在等候结束后，如果信道仍然没有其他数据占据，则可以发送。

因此，不同优先级的数据帧在传输时获得不同大小的竞争窗口，这种措施让优先级较高的数据可以等待发送的时间变短，率先进入信道，在一定程度上减少节点碰撞的次数，提升网络的性能表现。

3.7 本章小结

本章首先提出了基于 IP over CCSDS 的天地互联的协议转换方式，并根据该协议转换方式构建出了协议转换的总体模型， 根据本文论述的重点选出了构建协议栈模型的不同协议，重点在于传输层的协议和空间链路层的协议，并且对天地互联网关中的数据交互进行描述。之后对空间链路层 AOS 协议进行了设计，主要包括 AOS 传输帧结构的设计、IP 数据包的处理以及其他模块处理，对于空间中航天器的移动性对于 IP 数据包影响，本文提出了 IP 技术的方案，之后由于 IP 协议的不可靠的性质，本文在传输层使用 TCP 协议来保证数据的可靠传输，根据空间通信的一些问题，本文提出了传输层 TCP 协议的改进—SCPS-TP 协议，改进主要在于选择性确认机制、SNACK 机制和 Vegas 算法，综上本章完成了对天地互联的协议转换设计，在本章的最后本文对数据优先级的需求和实现进行了描述，为第四章的协议转换仿真打下了基础。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第四章 协议转换设计的仿真

基于 IP over CCSDS 的协议转换设计的仿真需要使用相关仿真软件，本文根据深空通信服务体系建设的项目需求使用了 OMNeT++网络仿真软件，协议转换的仿真是根据 OSI 五层协议模型，重点仿真的内容在于传输层和空间链路层协议。本文仿真的模型主要包括了地面系统和空间系统（卫星和月球着陆器等），地面系统主要是地面测控站通过天地互联的网关与空间网络中的航天器进行数据交互，本章先是对 OMNET++仿真软件进行介绍，之后会对仿真场景的构建、仿真结果和协议性能指标进行更加详细的论述。

4.1 OMNeT++软件介绍

本文使用的 OMNeT++软件进行的协议仿真，OMNeT++ 是一个模块化的、基于组件的 C++ 编程语言仿真库和框架，该软件的目的是对网络模型的创建并且对其仿真。其中“网络”的含义更广泛，包括有线和无线通信网络、片上网络、排队网络等。同时软件作为独立项目开发的模型框架提供特定领域的功能，例如对互联网协议、无线自组织网络、传感器网络等的支持等。OMNeT++使用了 Eclipse 作为自己的编辑器并对其进行一定的改变，提供了仿真运行的图形环境，集成了 SysC，扩展了数据库，可以实现实时网络仿真。在科学研究和工业生产中 OMNeT++作为一个网络仿真平台，被科学工作者广泛使用，拥有了一个数量众多的用户社区。OMNeT++为常用的网络模型提供了模块架构，模块使用 C++ 编程语言实现功能，然后使用网络描述语言(NED)组装成更加复杂的功能模型。OMNeT++对用户界面的使用具有很好的支持，并且由于其模块化架构，仿真内核（和模型）可以轻松嵌入到应用程序中[48]。

OMNet++的主要以下几个部分组成：

- 仿真内核库（C++）
- NED 拓扑语言
- 基于 Eclipse 平台的仿真 IDE
- 交互式模拟运行时的 GUI（Qtenv）
- 用于仿真执行命令行界面（Cmdenv）
- 实用程序（makefile 创建工具等）
- 文档、样本模拟等

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4.1.1 重要概念

（1）模块的概念

OMNeT ++ 模型由使用消息通信的模块组成。活动模块称为简单模块（simple modules），它们是用 C++编程语言描述，使用模拟类库。简单模块可以组合为复合模块（compound modules），在 OMNeT ++中整个模型称为网络，其本身就是一个复合模块。消息（messages）可以通过跨模块的连接传递，也可以直接发送到其他模块，模块与模块之间通过门（gate）连接。简单模块与复合模块之间的关系如 4-1 所示[49]。

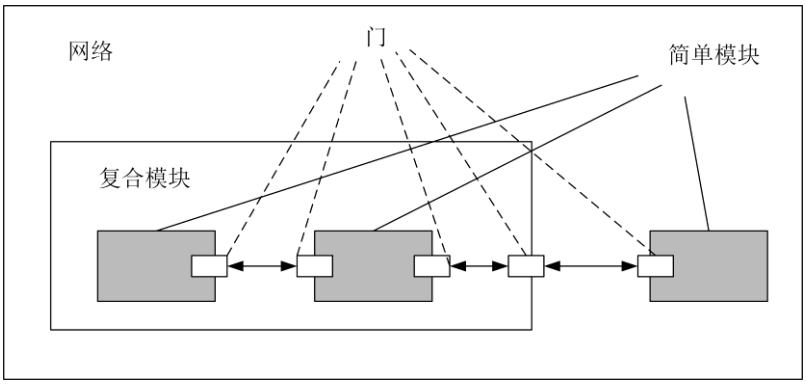

图 4-1 简单模块和复合模块

OMNeT++中提供了 NED 语言来描述模块的结构，NED 允许用户声明简单模块，并将它们连接和组合成复合模块，用户还可以将一些复合模块标记为网络。通道（channels）是另一种组件类型，它的实例可以在复合模块中使用。软件中提供了许多的模块类型，如前文中提到的简单模块等，并且用户还可以自定义模块类型，使用在结构组织更加复杂、功能更丰富的网络模型中。

（2）消息、门、连接和参数

模块通过交换消息进行通信。消息在仿真模拟中具有多种类型，比如 2 台主机之间的链路层中的帧、网络层的数据包或是具有实时移动性的物体等。消息可以包含数组、队列、栈、二叉树等较为复杂的数据结构。简单模块有 2 种发送消息的方式，一是直接发送，二是在 NED 文件中使用定义的路径。消息是 OMNeT++的核心概念，在 OMNeT++中消息由 cMessage 类及其子类 cPacket 表示。cPacket 用于通信网络中的网络数据包（帧、数据报、传输数据包等），cMessage 用于其他一切。用户可以自由地对 cMessage 和 cPacket 进行子类化，以创建新类型和添加数据。

当模块收到消息时，模块的“本地模拟时间”提前。消息可以来自另一个模

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

块或同一个模块（自我消息用于实现计时器）。

门是模块的输入接口与输出接口，消息通过输出门（output）发送，通过输入门（input）到达。在 OMNeT++中有三种类型的门：input、output 以及 inout，后者本质上是一个粘合在一起的输入和输出门。一个门，无论是输入还是输出，只能连接到另一个门。（对于复合模块的门，这意味着一个“外部”连接和一个“内部”连接。）可以单独连接输入输出门的输入侧和输出侧[50]。

连接是模块之间产生联系的关键，一般都是在单个模块的层次级别内才能被定义和创建，可以用在复合模块中通过门连接 2 个独立的简单模块，也可以连接另一个复合模块组成一个网络模型（如图 4-1）。连接定义在复合模块的“连接”部分。连接不能跨越层级；可以连接两个子模块门，一个子模块门和父模块门的“内部”，或父模块的两个门（尽管这很少有用），但不可能连接到父模块外部或复合子模块内部的任何门。

为了便于对通信网络进行建模，可以使用连接对物理链路进行建模。连接支持以下参数：Datarate（数据速率）、Propagation delay（延迟）、Error rate（误码率）和 Packet error rate（包错误率），并且可能被禁用。连接参数和底层算法被私有声明在信道对象的内部，无法被访问和修改，但是 OMNeT++允许用户可以根据仿真的需要赋予参数不同的值来构建不同性质的信道。

由于模型的层次结构，消息通常通过一系列连接传输，在简单模块开始产生，通过路径达到目的简单模块中，复合模块就像模型中的“木箱子”，在其内部领域和外部世界之间传递消息。

参数（parameters）可以在模块中存在，参数可以 NED 文件或是在 omnetpp.ini（配置文件）中配置具体的数值。参数可用于自定义简单的模块行为，并对模型拓扑进行参数化。其中参数可以采用字符串、数字或布尔值，也可以包含 XML 数据树。数值包括使用其他参数和调用 C 函数的表达式、来自不同分布的随机变量，以及用户交互输入的值。此外，在 OMNeT++中用户可以使用参数声明简单模块的数量、门的个数以及通过何种方式连接等，不同的参数就会产生不同的网络结构[51]。

综上，我们可以总结出使用 OMNeT 进行网络仿真的一般步骤：

1）使用 NED 语言描述模块的结构（包括模块的门、参数等），多个模块形成了网络拓扑；

2）使用 C++语言进行各种网络行为的代码编写，包括消息发送、消息删除、确认消息收到、消息重发等；

3）编写 ini 文件，在配置文件中给参数分配具体的数据，比如信道延迟分配

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2ms，误码率分配 10^{-6} 等，启动仿真，在 Qtenv 用户界面运行仿真实例。

4.1.2 INET 框架

本文仿真借鉴了 INET 协议框架，下面对该框架进行一些介绍。

INET 是基于 OMNeT++实现的一个包含五层网络模型的 C++框架，它包含了网络建模过程中常用的复合模块，如主机、交换机和路由器等，这些网络模块通常是由几个通信协议来实现，如主机就是包含了传输层 TCP 协议、UDP 协议，链路层 PPP 协议，MAC 协议等。一个网络模型内部是由许多主机、路由器等复合模块组成的，同时还定义了模块的门和一些参数，如模块是否具有移动性，模块之间的通信距离等，这些参数通常是在 NED 文件中描述，也可以在 ini 文件进行赋值。

INET 框架基本实现了五层网络中一些很常用的通信协议，如应用层的 http 协议，传输层的 TCP 协议、UDP 协议以及 SCTP 协议，网络层的 ARP 协议、ICMP 协议、IPv4 协议和 IPv6 协议，链路层的 ppp 协议、Mac 协议和 IEEE80211 协议，关于物理层 INET 提供了噪声模拟、通信缓存、传播速度或是时间等定制模块，除了常用的通信协议，该框架定义了模块移动性、仿真可视化、能量模型、路由以及缓存队列等属性。

INET 框架结构是按照文件夹分成组织的，非常像 Java 的包。INET 中的包大致按照 OSI 模型来进行组织的，顶层包括 inet.applications、inet.transport、inet.networklayer 和 inet.linklayer，其他的包为 inet.base、inet.util、inet.world、inet.mobility 和 inet.nodes，这些包对应着 INET 源代码树中的 src/applications/、src/transport/等目录，其中 src/目录对应于 INET 包，由 src/package.ned 文件定义。顶层包中的子目录一般对应着具体的协议和协议族，简单模块的实现是同名的 C++类，源文件放置在与 NED 文件相同的目录中[52]。

4.2 仿真场景的构建

4.2.1 网络拓扑模型构建

天地互联的协议转换仿真涉及到地面网络、空间网络、天地网关以及地面和空间的数据传输等，地面网络协议的仿真可以借鉴 INET 框架来进行实现主要是由 TCP/IP 协议族构成，地面网络的主要节点是地面站，空间网络的仿真主要是通过修改 INET 框架来完成主要是由 CCSDS 协议族构成，空间网络的主要节点包括地月中继卫星和月球着陆器，地面网络的地面站通过地月中继卫星与月球着陆器进

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

行通信指挥，月球着陆器也是经地月中继卫星把测量数据发送到地面站。地月空间网络拓扑图如图 4-2 所示。

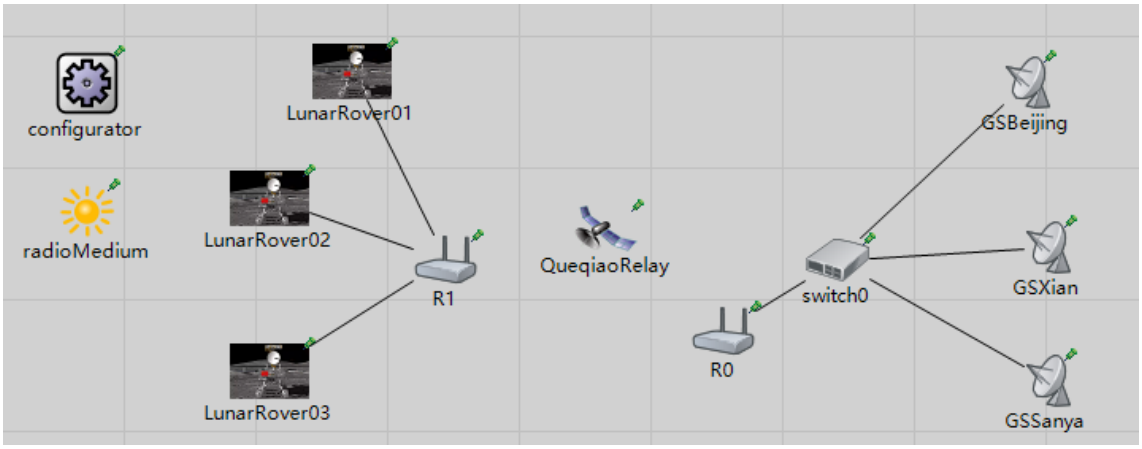

图 4-2 地月空间网络拓扑图

图 4-2 中右侧有三个月球着陆器，分别是 LunarRover01、LunarRover02 和 LunarRover03，拓扑图中间有一个地月中继卫星 QueqiaoRelay，拓扑图左侧有三个地面站西安、北京和三亚，图中 configurator 为 IPv4 配置模块，2 个路由器分别为 R0 和 R1，一个交换机 switch0。这几个网络节点的位置参数如表 4-1 所示。

表 4-1 网络节点位置参数

节点	位置
LunarRover01	经度-19.5°，纬度 44°
LunarRover02	经度 180°，纬度 45°
LunarRover03	经度-52°，纬度 43°
QueqiaoRelay	轨道高度 1200km，轨道倾角 89.0°
GSBeijing	东经 120°，北纬 40°，海拔 36.6m
GSXian	东经 90°，北纬 18°，海拔 36.6m
GSSanya	东经 110°，北纬 18°，海拔 36.6m

3 个地面站之间使用 TCP/IP 协议进行通信，它们通过以太连接到交换机 switch0，switch0 再与无线路由器 R0 相连，R0 会把来自地面站的消息转发给中继卫星 QueqiaoRelay，通过空间链路与无线路由器 R1 进行数据传输，R1 根据 IP 地址把消息转发给对应的着陆器，着陆器与 R1 也是通过以太（Ethernet）连接的，

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

接收到指令后执行探测活动。R0 和 QueqiaoRelay 与 R1 和 QueqiaoRelay 这 2 段空间链路都是使用的 CCSDS AOS 协议，IP 数据包在路由器被封装成 AOS 传输帧，通过射频把数据发送在无线信道上，达到中继卫星之后经卫星缓存转发，到达月球路由器。月球着陆器向地面站发送测控信息为反过程，这里不再叙述。图 4-3 为地月空间网络的协议栈模型。

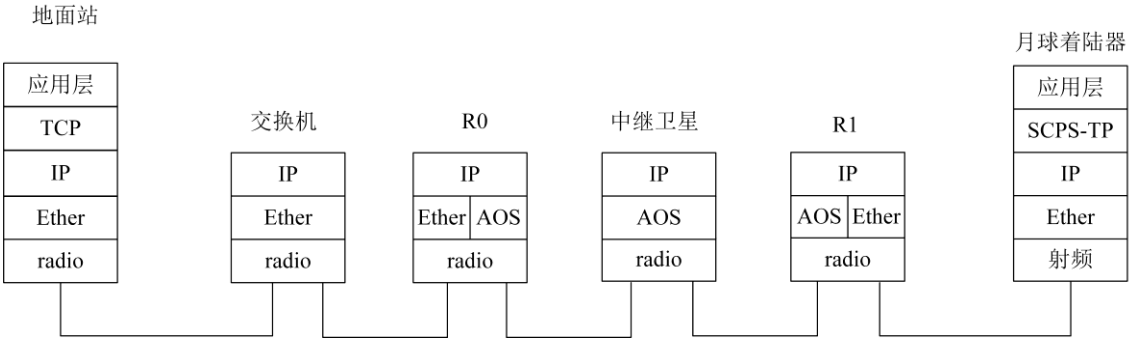

图 4-3 地月空间网络协议栈模型

4.2.2 节点模块的结构

本文仿真中我们主要创建了 3 个节点类型 Lander、RelaySatellite 和 GroundStation 分别对应着月球着陆器、中继卫星和地面站，这 3 个节点都是几个简单模块组成的复合模块。着陆器节点模块由 5 个简单模块构成，其结构图如图 4-4 所示。

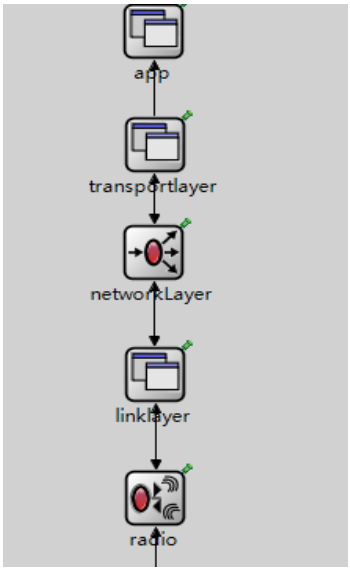

图 4-4 着陆器模型结构

从图 4-4 中可以看出该模块结构对应着 TCP/IP 网络模型，从上到下分别是应

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

用层、传输层、网络层、链路层和网络接口层，分别对应着着陆器模型的 app 子模块、transportlayer 子模块、networklayer 子模块、linklayer 子模块以及 radio 子模块。

（1）app 子模块

本文的研究重点不是应用层协议，所以选择了一个应用层模板 tcpApp，它仅仅表示了传输层的上层结构，有着与 SCPS-TP 协议进行数据交互的接口。

（2）transportlayer 子模块

该模块是本文的研究重点，INET 在 OMNeT++中提供了一个传输层模块，该模块主要包括了 TCP 协议和 UDP 协议，SCPS-TP 协议是在 TCP 协议的基础上实现的，所以本文要对 TCP 协议在 OMNeT 中的内容做一些叙述。

TCP 协议的主要内容包括：

TcpConnection.msg：此消息文件主要定义了一些 TCP 协议连接时的参数，比如 SYN、ACK、FIN 等，以及 TCP 的连接管理如 2.1.3 小节所述，还有 TCP 的发送队列 TcpSendQueue 和接收队列 TcpReceiveQueue。

TcpAlgorithm：主要包括了 TCP 协议的几种算法，包括 TcpNewReno、TcpReno、TcpTahoe 以及 TcpWestwood 算法。 这些算法主要实现了消息重传和拥塞控制，如慢启动、拥塞避免和快速重传等。

本文的 SCPS-TP 协议的 SCAK 机制、SNACK 机制和 Vegas 控制算法就是在以上 TCP 连接和算法上完成的。

（3）networklayer 子模块

该子模块主要包括 IP 协议、ARP 协议（Address Resolution Protocol）、ICMP 协议（Internet Control Message Protocol）和 IGMP 协议（Internet Group Management Protocol）。IP 协议主要负责将传输层的分段（segment）加入一些控制信息封装成数据包（packet），再交给链路层即可，或者是把链路层的传输帧（frame）进行解析，提取出用户数据再封装成 IP 数据包。

ARP 协议则是一个地址解析协议，主要作用是将 IP 数据包中的目的 IP 地址进行翻译，成为 MAC 地址，才能把数据发送到正确的主机上。如果节点想要向 MAC 地址未知的节点发送 IP 数据包，它会在以太网上广播 ARP 帧，被请求的节点会发送包含自己 MAC 地址的 ARP 帧响应请求的节点，请求节点收到响应后，更新其 ARP 缓存，使用得到的 MAC 地址发送 IP 数据包。

ICMP 协议是是用来报告和诊断网络错误的一种协议，可以在主机和路由器之间传输消息，这些消息包括路由地址是否可以达到、网络是否能够通信等，它与 TCP 协议不同，它不用传输用户数据也不能和 IP 数据报分离，因此该协议是不可

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

靠传输，在 INET 框架中提供了 sendErrorMessage()函数来发送错误的 IP 数据报信息。

当网络中的主机需要和同在一个子网中的路由器建立多播组的通信关系时，需要使用到 IGMP 协议。IGMP 负责将多播组成员身份信息从主机分发到路由器，当主机的一个接口加入多播组时，它将向路由器发送该接口的 IGMP 报告，还可以在接口离开多播组时发送报告。

（4）linklayer 子模块

链路层主要使用了 Ethernet 和 MAC 协议，以太网数据的传输速率为 10Mbps，采用全双工通信。在 INET 框架中支持了主要的以太网技术和设备类型的支持，比如仿真使用的交换机，与在物理层工作的无源集线器和中继器不同，交换机在数据链路层工作[53]，并在连接的子网之间中继帧。

MAC 协议中本文定义了 4 种类型的 MAC，分别是 GeneralMac、SatToSatMac、SatToTerMac 和 TerMac，用在着陆器的 MAC 是第一种，即通用性的 MAC，它声明了 4 个输入输出门，用于接收和发送消息，还定义了一个数据速率的参数 bitrate，默认值为 8Mbps，该 MAC 模块较简单，没有提供封装和解封装的功能，使用 nextSendTime 参数来进行媒体访问控制，该参数为 IP 数据报的长度除以 bitrate 得到。

（5）radio 子模块

本文定义了 7 种不同的 radio 子模块，分别是 GeneralRadio、GSRadio、LanderRadio、SatToGSRadio、SatToSatRadio、SatToTerRadio 和 TerRadio。在着陆器中使用的就是第一种，即通用的 radio 子模块。该子模块定义了 4 个输入输出门用于收发数据，声明了 3 个参数 destRadio、destNode 和 propagationSpeed，分别是目的 radio 模块、目的节点和传播速度，这里的传播速度默认为光在真空中的速度，即 299792458m/s。

地月中继卫星模型的结构如图 4-5 所示。图 4-5 中包含了 5 个简单模块，其中 networklayer 子模块已经在之前的段落中介绍了，这里主要介绍 AOS 子模块。AOS 子模块实现了 AOS 协议的基本功能，主要包括包处理模块、虚拟信道帧生成模块、AOS 帧生成模块和同步模块，包处理模块是通过队列存储 IP 数据包，根据其长度进行切割和拼接，放入 M_PDU 包区中；虚拟信道帧生成模块生成 AOS 帧的基本结构；AOS 帧生成模块是在虚拟信道帧模块的工作之后，填充 AOS 帧结构的剩余部分，主要是差错控制模块；同步模块主要是通过在传输帧添加一个标识 ASM，接收端来识别 ASM 完成帧同步。以上为发射端 AOS 协议的实现，在接收端为反过程。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

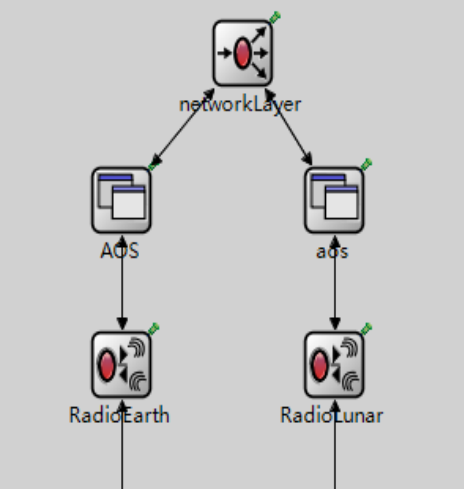

图 4-5 中继卫星模型结构

RadioEarth 子模块和 RadioLunar 子模块，前者是用来和 R1 进行消息的收发，后者是用于和 R0 通信。

地面站发送的消息从 RadioEarth 子模块进入中继卫星，先是经过 AOS 子模块进行 AOS 同步接收，从 AOS 帧中提取 IP 数据包，根据包中的目的 IP 地址进行转发，转发之前进入 aos 子模块进行 AOS 帧生成的步骤，完整的 AOS 帧经 RadioLunar 子模块发往 R1 路由器，R1 解帧之后根据目的地址将 IP 数据包通过以太网送达对应的着陆器，着陆器发送数据到地面站为反过程。

图 4-6 为地面站模型结构，地面站模型也包含了 5 个简单模块，从上至下分别是 app 模块、transport 模块、networklayer 模块、linklayer 模块和 redio 模块，对应着 OSI 五层模型，与月球着陆器不同的是在传输层，月球着陆器使用的是 SCPS-TP 协议，而地面站上使用的就是一般的 TCP 协议。

除了以上 3 类主要的节点模型，地月网络的拓扑图中还有几个节点。LA0、LA1 和 EA 都属于无线路由器，在 INET 中该类节点具有 2 种输入和输出的门，分别是 radio 和 ethg，对应于无线和有线连接，该节点是基于 IEEE80211 协议实现，并且具有可移动性，可在 mobilityModule 设置；configurator 模块是用来分配 IPv4 地址的，可以默认配置或是在 XML 文件中手动配置，如果是前者那么 IP 地址就是从 10.0.0.0 开始到 10.255.255.255 按顺序分配；radioMedium 是一个无线媒介，在 INET 中只要使用了无线节点，就需要配置它，该模块表示所有通信节点共享物理介质，它可以考虑信号衰减、干扰等物理现象，本文使用的模型忽略了信号的衰减和干扰；switch0 是基于以太交换机实现的，每个节点都通过全双工线路直接连接到交换机，默认使用 CSMA（Carrier Sense Multiple Access，载波侦听多路访

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

问），信道利用率高，它是属于链路层的一种设备。

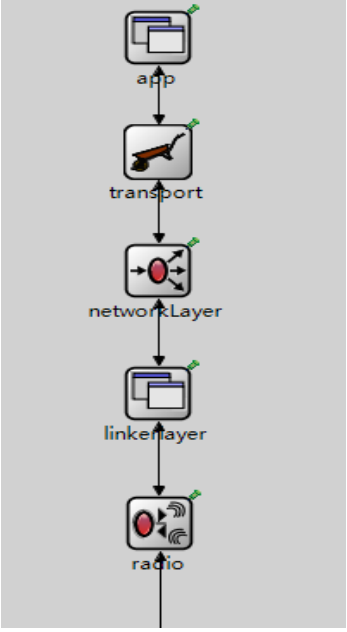

图 4-6 地面站模型结构

由于在真实的空间环境中继卫星具有移动性，本文在 OMNeT++中设计了一个简单模块 SatJ2Mobility 来实现卫星的移动，该模块中有几个参数 semiMajorAxis（卫星长半轴）、eccentricity（离心率）、raan（右升交点角）、perigeeArgument（近地点幅角）、meanAnomaly（平近点角），这里本文简化了仿真模型，把卫星的运行轨道设置为正圆，因此 semiMajorAxis 设置为 7578km，其余参数都为 0。

4.3 链路层 AOS 协议的仿真验证

本小节是关于 3.4 节 AOS 协议的验证，地面站北京和月球着陆器 LunarRover01 的 AOS 帧的内容，如果上述结果一致，则实现了 IP over CCSDS，即 TCP/IP 协议与 CCSDS 协议的转换。

本次验证我们设置 IP 数据包的长度为 64 字节，每 100ms 从地面站北京发送一个 IP 数据包，一共发送 10 个数据包来验证 AOS 协议的可行性。北京的 IP 地址为 192.168.31.169，端口号为 64480，月球着陆器 LunarRover01 的 IP 地址为 192.168.31.211，端口号为 49787，数据速率为 2000bps，误码率设为 0。运行仿真，我们在 debug 模式下查看尚未进行 AOS 帧封装的 IP 数据包，如图 4-7 所示。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

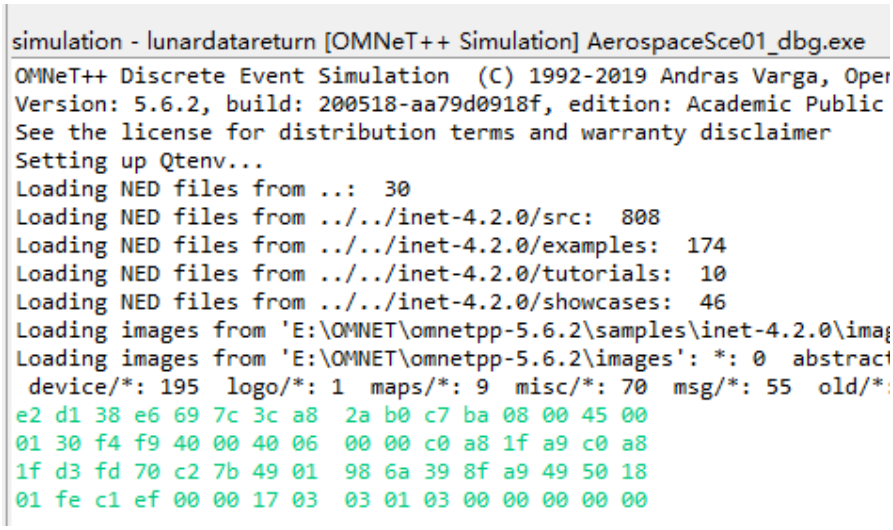

图 4-7 发送端的 IP 数据包

图 4-7 中有颜色的字体为 IP 数据包内容，其中第二行数字“c0 a8 1f a9”北京 IP 地址的十六进制，第二行至第三行数字“c0 a8 1f d3”为着陆器的 IP 地址，数字“fd 70”为北京端口号，“c2 7b”为着陆器端口号，该数据包的 TCP 序列号为第三行数字“49 01 98 6a”，即 1224841322。在进行封装 AOS 帧时，会把 IP 数据包的数据部分提出出来，放入帧中，AOS 帧长度为 65 字节，如图 4-8 所示。

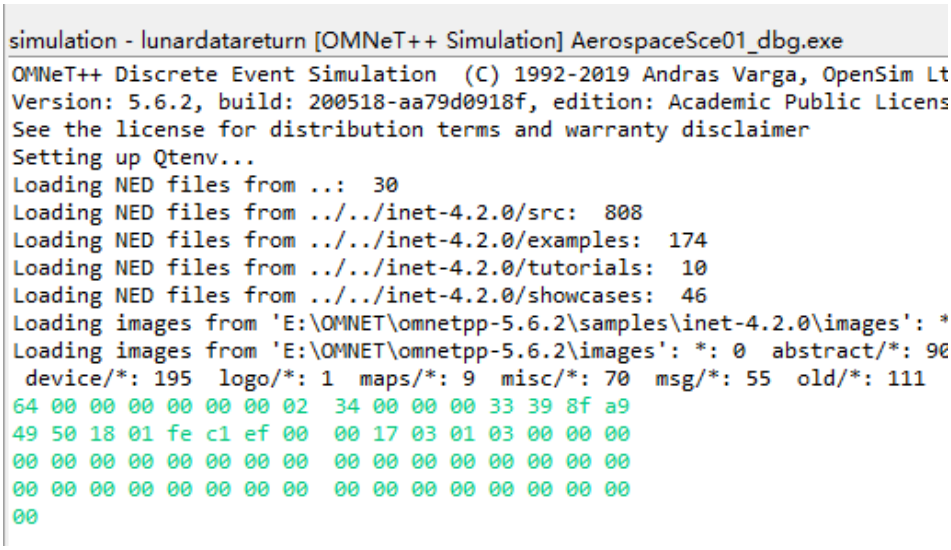

图 4-8 AOS 帧内容

从图 4-8 中可以看到，AOS 帧提取了 IP 数据包的数据部分，对于剩余部分进行填充 0。之后再正常运行仿真，在控制台窗口可以看到达到着陆器的 IP 数据包如图 4-9 所示。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
0000  e2 d1 38 e6 69 7c 3c a8  2a b0 c7 ba 08 00 45 00
0010  01 30 f4 f9 40 00 40 06  00 00 c0 a8 1f a9 c0 a8
0020  1f d3 fd 70 c2 7b 49 01  98 6a 39 8f a9 49 50 18
0030  01 fe c1 ef 00 00 17 03  03 01 03 00 00 00 00 00
```

图 4-9 接收端的 IP 数据包

可以看到接收端着陆器的 IP 数据包与发送端地面站北京的 IP 数据包一致，验证了 AOS 协议的正确性，验证了 AOS 封装和解析的操作可以实现，为下一节传输协议的仿真奠定了基础。

4.4 移动 IP 的仿真实现

4.4.1 移动 IP 模块的设计

switch0 和路由器 R0 作为外部链路，Sanya 作为外部代理 EA，初始时刻 MN 处于本地链路，和 LA 具有相同的网络前缀，这里 MN 的本地 IP 地址在 XML 文件设置为 192.168.31.169，LA 的 IP 地址为 192.168.31.168，EA 的 IP 地址为 192.167.31.169，R0 的 IP 地址为 192.0.0.0。仿真示意图如图 4-10 所示。

在 OMNeT++中本文设计一个移动 IP 的模块，该模块提供了移动 IP 技术实现需要的几个 class（类）和函数，下面对这几个类和函数做一些阐述。

class ICMP：该类是一个消息类，用在移动 IP 代理发现阶段。来自本地代理，外部代理的代理广播、移动节点的请求消息都是 ICMP 类型的消息。该类包括了 ICMP 类型变量（广播消息或是请求消息）、IP 地址变量、LA 标志位、EA 标志位、注册请求标志位以及 EA 可提供的转交地址。

class registrationMessage：该类定义了移动 IP 向 LA 请求注册的消息类型，注册消息一般内置于 UDP 数据报。该类包括了注册类型变量（请求或是回复）、转交地址、LA 地址、EA 地址、请求注册消息的存活时间、64 位的消息 ID。

class mobileNode：该类定义了移动节点，它包括了本地的 IP 地址、MAC 地址和转交地址。

class correspondentNode：该类定义的节点是用来和 MN 通信的，本文仿真中通信节点为 GSXian。该类包括了自身的 IP 地址。

class localAgent：该类主要是用来在本地链路中执行 MN 移动性管理，比如将分组转发到 MN 所在的外部链路中的 EA 中去。该类主要包括了 LA 地址和移动绑定表（存储本地 IP 地址和转交地址的映射）。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

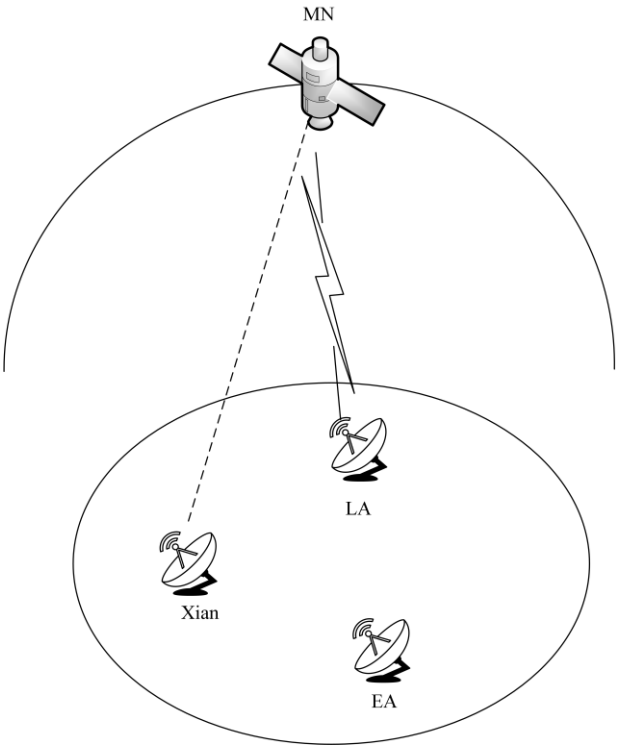

图 4-10 移动 IP 仿真示意图

class external Agent：该类定义了外部链路中的一个实体，它帮助 MN 实现移动性管理，例如接收发往位于其链路中的 MN 的数据报。该类主要包括了 EA 地址和访问列表。

class datagram：该类包含 MN 和通信节点之间发送的数据，它包括了源地址、目的地址和生存时间。数据报在发送时需要进行封装，EA 收到封装的数据报并将其解封，转发到 MN。

以上为本文定义的关于移动 IP 的几个主要类，下面在主函数中本文实现了几个函数来完成移动 IP 的基本仿真。

configuration(ICMP_t& agentDiscovery, network& n)：初始化移动 IP 的仿真环境，使用代理消息广播的方式来发现 MN。

agentDiscovery(MN m, LA l, EA e, ICMP_t agentMethod)：这一函数主要实现移动 IP 代理发现的功能，它广播一条 ICMP 类型的消息来发现 MN。

registerMN(MN &m, LA &l, EA &e)：该函数实现当 MN 处于外部链路时在 LA 上注册的功能。

indirectRouting(MN m, LA l , EA e, correspondentNode CN)：该函数主要实现通信节点（GSXian）向 MN 发送数据报。数据报会被路由到 EA 所在网络，但是 LA 会拦截，他会将封装的数据报（隧道）转发给其绑定表中的 MN 的外部代理，LA

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

将解除封装的数据报转发给 MN，通信节点并不知道移动节点位于外部链路中。

directRouting(MN m, LA l , EA e, correspondentNode CN)：直接路由解决了三角路由的问题。要实现直接路由需要在会话开始之前了解 MN 的转交地址，通过直接向 LA 询问实现的。

4.4.2 仿真结果的分析

LA 和 EA 之间相距 100km，MN 以 10km/s 的速度从 LA 向 EA 移动，GSXian 向月球着陆器 LunarRover01 发送数据包，每个 1s 发送一个，一共发送 20 个，当 MN 处于本地链路时，GSXian 是先向 LA 发送，再通过 MN 转发到 LunarRover01；当 MN 移动到外部链路时，此时通信会中断，因为 MN 通过代理消息发现其网络前缀与当前链路的网络前缀不一致，MN 获得转交地址，向 LA 请求注册，之后 LA 会截取目的地址为本地 IP 地址的数据包，通过隧道技术发送到 EA，EA 再经过 R0 发送到 MN，MN 转发到 LunarRover01。图 4-11 是接收端 LunarRover01 在不同时刻接收到的数据包变化图。

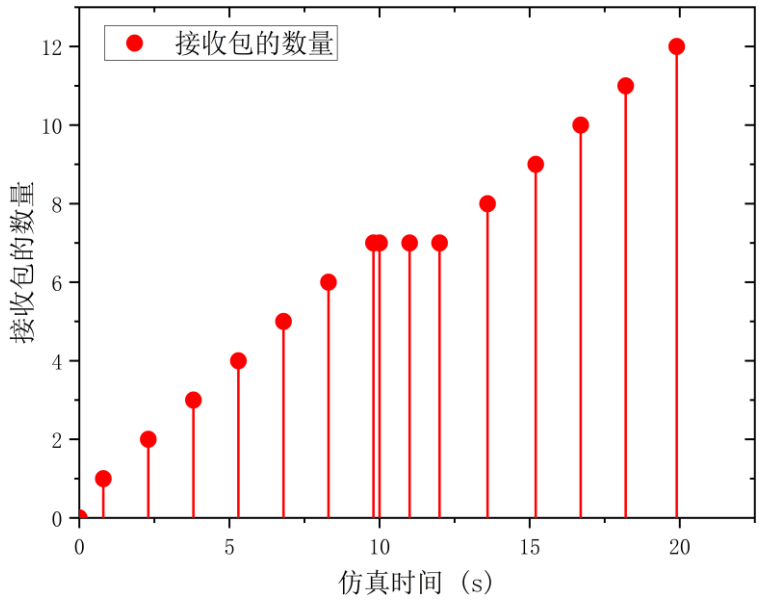

图 4-11 接收端数据包数量变化

这里将信道 delay 设置为 0.25ms 从图中可以看到，第一个数据包在大概 0.8s 时达到接收端，之后每个大概 1.5s 后达到一个数据包，如果不使用移动 IP 大概 10s 时数据包的数量不再增长，之后链路会中断约 5s，在使用了移动 IP 后，MN 需要向 LA 进行注册操作，暂时无法转发数据包，在大概 2s 后数据包再次开始变化，大概间隔 1.6s 增长一个数据包，因为 LA 拦截数据包进行封装操作之后，不是直接发送给 R0，而是发送给 EA，EA 再发送给 R0，这是间接路由，之后当 MN 和

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

GSXian 距离较近时，是通过直接路由方式传送，间隔大概在 1.5s 左右。

4.5 传输层 SCPS-TP 协议仿真与仿真结果

本节主要是针对 3.5 节的仿真实现，即 SCPS-TP 协议的仿真，INET 框架中提供了 TCP 的协议模型，我们对于 SCPS-TP 协议的仿真主要是在修改 INET 框架的基础之上实现的，对于 3.5.1 小节关于的空间通信存在的问题即传输延时长、高误码率、前向和反向链路不对称，本文提出了 3 种改进机制来提高空间传输效率，即选择性确认机制、SNACK 机制和 Vegas 算法，下面本文将会比较在不同误码率的影响下 TCP 协议和 SCPS-TP 协议的吞吐量和丢包率的变化，这可以直接反映出两者的在空间网络中性能差异；以及在不同的误码率下两者的拥塞窗口的变化。

4.5.1 吞吐量比较仿真和分析

OMNET 里有 3 种数据传输模式，bytestrem、object 和 bytecount。其中 bytestream 是传输真实的消息字节，TCP 会将字节分成 MSS（分段最大的尺寸）块大小，当作 TCP 分段的有效载荷进行传输，本文选择它作为仿真的数据传输模式。

在 OMNeT++中本文通过设置信道模块 DatarateChannel 来实现地月数据传输的信道模型，该信道模块一共有 4 个比较重要的参数，分别是 delay（传输时延）、datarate（数据速率）、ber（误码率）、per（数据包错误率），具体的参数设置表 4-2 所示。

表 4-2 吞吐量仿真参数

仿真参数	数值
delay(s)	0.25s
datarate(bps)	2000（前向），1000（反向）
ber	10^{-8},10^{-6}
per	0
AOS 帧长度（byte）	65

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

此外，DatarateChannel 模块还提供了几个统计参数，其中就包括了吞吐量（Throughput），吞吐量的计算是通过 sumPerDuration()函数实现的，该函数计算到目前位置发射窗口的字节数之和，除以持续时间（RTT），并输出计算结果，单位为 bps。

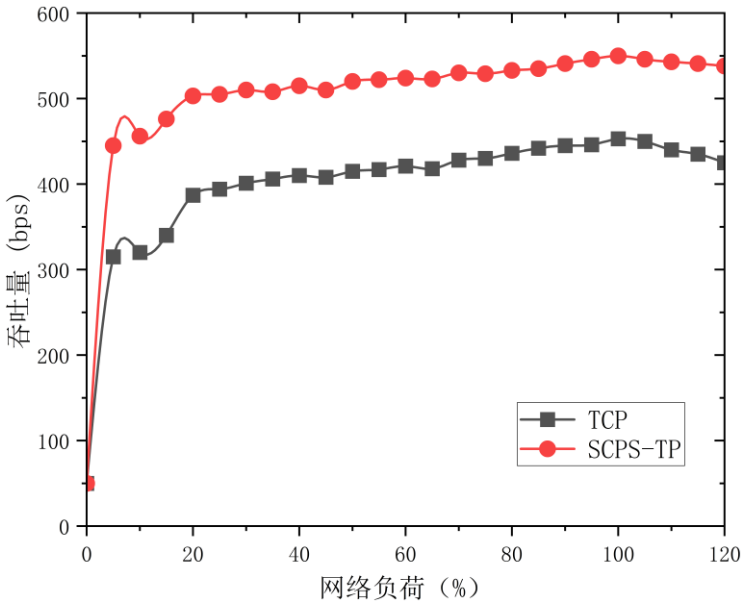

图 4-12 误码率为 10^{-8} 时吞吐量比较

图 4-12 中 TCP 协议和 SCPS-TP 协议的吞吐量比较，这里的吞吐量主要是月球着陆器的吞吐量，本文通过该节点的传输层协议分别设置成 TCP 协议和 SCPS-TP 协议来进行比较的。可以看出后者的整体性能是优于前者，主要的原因包括 2 个：一是 SCPS-TP 协议的发送窗口比 TCP 协议大，在本次仿真中我们将发送窗口由原来的 16 比特扩展到 30 比特，所以在开始发送时使用 SCPS-TP 协议可以传输更多的数据量，根据公式（3-1）可知，吞吐量也跟着变大；二是因为 SCPS-TP 协议使用了 Vegas 拥塞算法可以根据网络负荷动态的调整拥塞窗口的大小，可以减少数据丢包的现象，不用数据重传，也是可以提升吞吐量。从图 4-12 中，我们还可以看到当网络负荷超过 100%时，即出现网络拥塞时，SCPS-TP 协议和 TCP 协议的吞吐量均下降，但是可以看出前者由于使用了 SNACK 机制让吞吐量下降的程度小于后者。

图 4-13 的误码率为 10^{-6}，比图 4-12 提升了 2 个数量级，TCP 协议的吞吐量下降了很多，曲线也比较抖动，SCPS-TP 协议吞吐量也收到影响但是不如 TCP 协议那么剧烈。TCP 协议吞吐量下降的主要原因是误码率的提高导致数据传输丢包率增大，又 RTT 时间长，使数据重传的时间变长同时拥塞窗口变小，拥塞控制算法

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

不断在慢启动和拥塞避免阶段循环，发射窗口的数据量变少。SCPS-TP 由于使用了 SACK 机制，重传丢失数据即可，又加上 SNACK 机制，使用数据空洞将发射窗口中错误信息打包一起发送，不用多次发送带有错误信息的 ACK 包，在高误码率的情况下改善了吞吐量下降的现象。

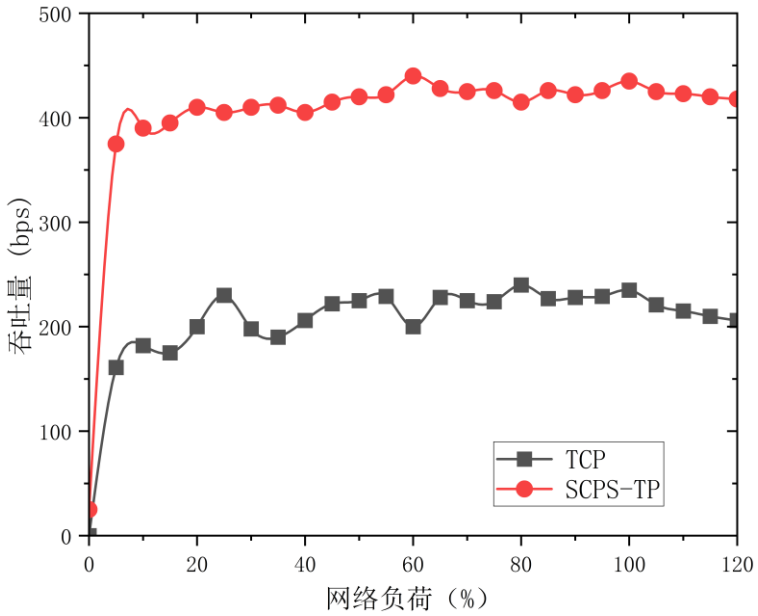

图 4-13 误码率为 10^{-6} 时吞吐量比较

4.5.2 丢包率比较仿真和分析

本小节针对月球着陆器的数据丢包率进行仿真和分析，通过计算丢失数据包的数量占预期接收数据包的比例得出数据丢包率，本文将设置不同的误码率下统计丢包率来比较 SCPS-TP 协议和 TCP 协议的性能。具体的仿真参数如表 4-3 所示。

在 OMNeT++中本文通过设置地面站节点的发送包的数量 numSent 和月球着陆器节点的接收包的数量 numRecieved，两者的差值就是丢失的数据包数量 numLost，丢包率 lostPacketRate=numLost/numSent，在 initialize()函数初始化后，再以关键字 WATCH 监视即可。图 4-14 为 TCP 协议和 SCPS-TP 协议的丢包率比较。

从图 4-14 中可以看出当误码率为 10^{-8}，TCP 协议和 SCPS-TP 协议的丢包率都很低不会超过 1%，随着误码率的不断提升信道质量逐渐恶劣，丢包率也没有超过 5%，然而误码率一旦超过 10^{-5} 时，两者的丢包率直线上升，在误码率为 10^{-4} 时 TCP 协议的丢包率达到 30%左右，超过了丢包率为 25%的 SCPS-TP 协议，这表明了 SNACK 机制在空间通信中的优越性能，从 4.4.1 小节和本小节来看，SCPS-TP 协议在高误码率和长时延的情况下，其吞吐量和丢包率的表现均优于 TCP 协议。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

表 4-3 丢包率仿真实验参数

仿真参数	数值
delay(s)	0.25
datarate(bps)	2000（前向），1000（反向）
ber	10^{-4}、10^{-5}、10^{-6}、10^{-7}、10^{-8}
per	0
AOS 帧长度（byte）	65

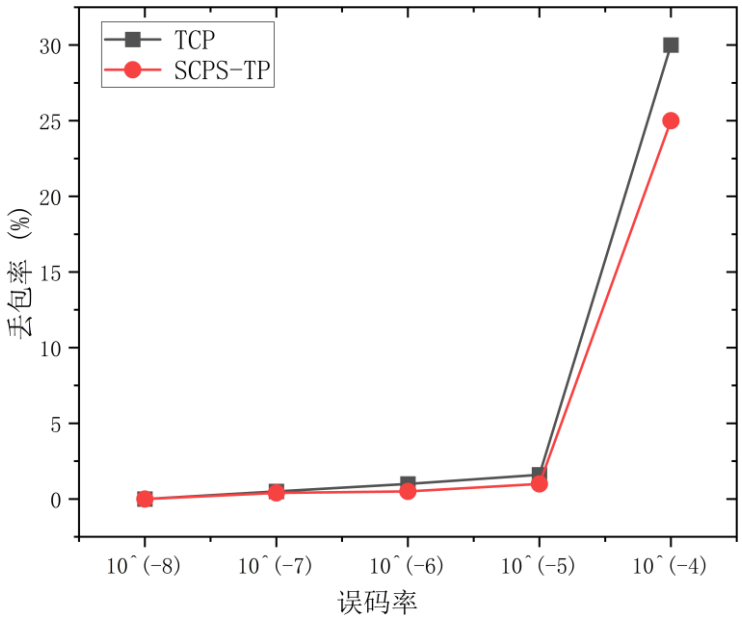

图 4-14 TCP 协议和 SCPS-TP 协议丢包率比较

4.5.3 拥塞窗口比较仿真和分析

在 TCP 协议中可以知道拥塞窗口值（CWND）的大小和数据丢包有关系，当没有丢包时，CWND 值会快速增大，如果有丢包的话，拥塞控制算法会选择拥塞避免而导致 CWND 值增长很缓慢，使发射窗口的数据量很少，所以 CWND 是衡量 TCP 协议的重要指标之一，这一小节本文将比较 TCP 协议和 SCPS-TP 协议在不同误码率的条件下，CWND 的变化情况。

在 OMNeT++中，TCP 协议在 TcpConnection.ned 中提供了 cwnd 的信号参数，这个参数记录了拥塞窗口值的变化，它的默认值为 1MSS，在仿真开始后记录该参数的变化情况即可得出结论。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

图 4-15 为误码率为 0 时，TCP 协议和 SCPS-TP 协议的 CWND 值的变化，可以看到没有丢包产生时，CWND 的值增长很快且两个协议的变化曲线几乎重合。

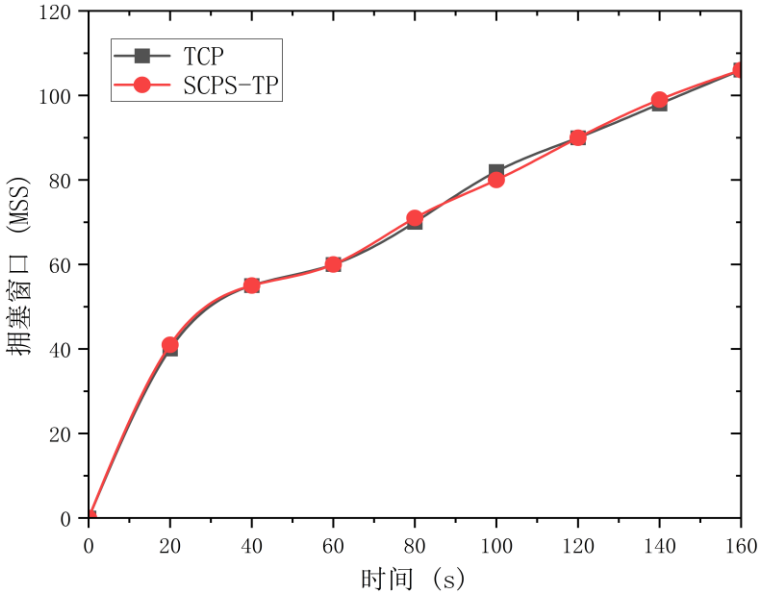

图 4-15 误码率为 0 时的拥塞窗口值的变化

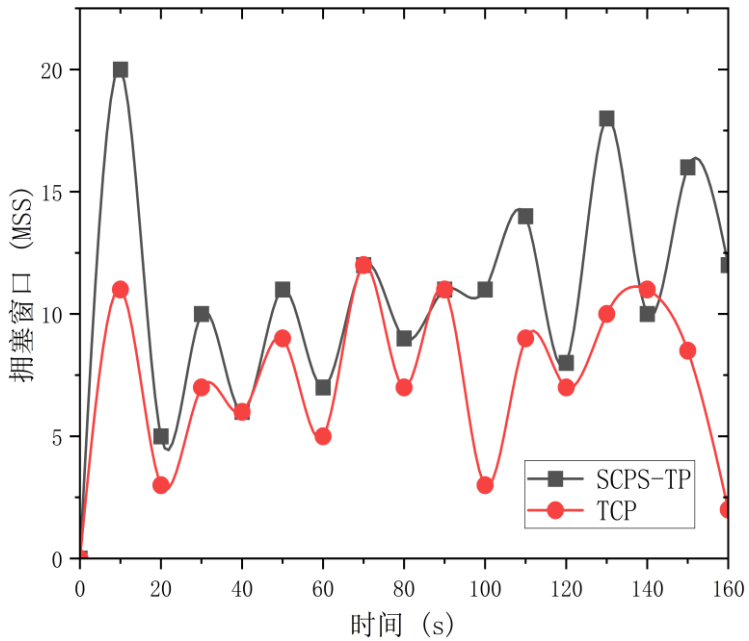

图 4-16 误码率为 10^{-5} 时的拥塞窗口的变化

图 4-16 当误码率为 10^{-5} 时，会有丢包的产生，由于空间的时延长，TCP 协议的 ACK 包不能及时达到，所以拥塞窗口值不会增长得很大，最大时也只有 12MSS 左右，而 SCPS-TP 协议由于采用了 Vegas 拥塞算法，可以动态调整拥塞窗口，期

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

望吞吐量和实际吞吐量的差值较 TCP 协议小，根据公式（3-7）可以得知 CWND 不断增大 1 个 MSS，而且由于使用了选择性确认，不用发送很多个 ACK 包，所以总体上 SCPS-TP 协议的拥塞窗口值是大于 TCP 协议，最大值有 20MSS，平均值也有 12MSS 左右，性能比 TCP 协议好。

从图 4-17 中我们可以看到，与图 4-16 相比较，整体上拥塞窗口值都提升了，这是因为误码率的降低，SCPS-TP 协议的 CWND 最大值没有超过 45MSS，而 TCP 协议则没有超过 25MSS，整体上前者比后者要稳定一些，TCP 协议会把所有的丢包原因归结于网络拥塞，是不管信道是否恶化，根据拥塞控制算法就会减小拥塞窗口，这忽视了空间链路的特性，而 SCPS-TP 协议考虑了空间链路的特殊性，在出现丢包时会默认是链路的恶劣条件导致的，所以它的拥塞窗口会比较稳定，这是复合空间通信的实际情况的。

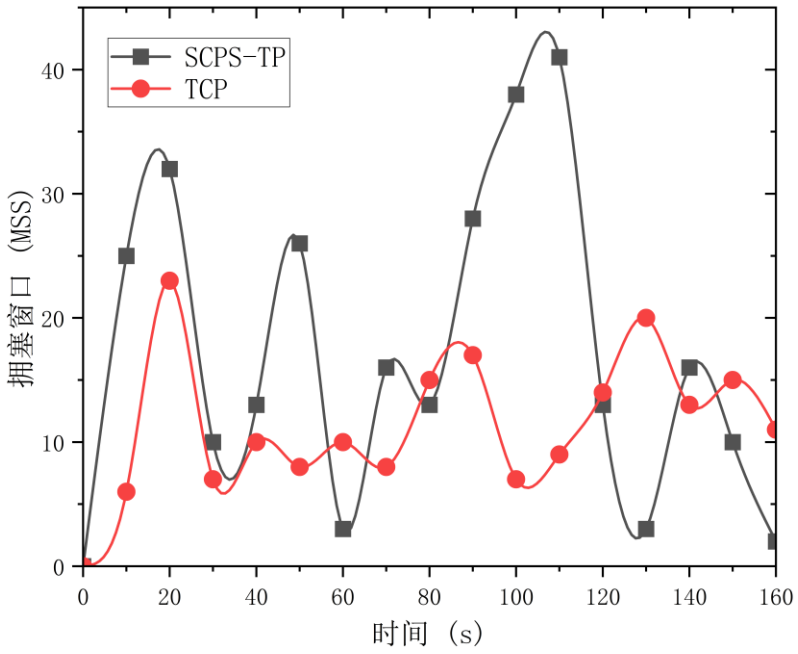

图 4-17 误码率为 10^{-7} 时的拥塞窗口的变化

4.6 关于数据优先级的仿真

4.6.1 Mac 模块的设计

在地月传输的场景中，3 个月球着陆器 LunarRover01、LunarRover02 和 LunarRover03 分别发送 tag=00，01 和 10 的带有优先级标识的数据，向无线路由器 R1 发送。本文对 3 个着陆器的 GeneralMac 进行拓展，其拓展的主要内容如下：

（1）定义 Mac 帧的头部结构体，包括了 destAddr（目的地址）、srcAddr（源

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

地址）、duration（信道使用的时间）、tag（数据优先级字段）；定义了私有成员变量 CV_min，CV_max、SlotTime（时隙主要是由物理层来决定的，但是本文研究没有涉及到物理层，所以这里把该参数定义为定值）和 t_wait（等待时间）；

（2）定义了初始化函数 initial_cv()，该函数实现了当数据是具有优先级数据的标识时，对于竞争窗口进行赋值，当 tag=00，CV=3*CV_min，当 tag=01，CV=2*CV_min，当 tag=10；CV=CV_min；

（3）定义 class Timer，该类主要包含了关于等待计时器的几个控制函数，如 start()函数开启计时器，pause()函数暂定计时器和 recovery()函数恢复计时器。start()函数的部分伪代码如下：

```
Timer::start(int cv,int idle)
{
    paused_=0;
    Scheduler &s= Scheduler::instance();
    stime=s.clock;    //记录当下时间
    t_wait=(Random::random()%cv)* SlotTime;     //等待时间=CV*时隙值
    if(idle==0)   //信道忙碌
    paused_=1   //暂停计时器
    else{
        assert(t_wait>=0);     //计时器尚未归零
        s.schedule(this,&intr,t_wait);  //计时器倒计时
    }
}
```

4.6.2 仿真结果的分析

在本次仿真中目的节点为 R1，源节点有 3 个分别发送不同优先级的数据，这 3 个不同源不同优先级的数据竞争从着陆器到无线路由器的信道，使用信道提供的参数 datarate 来控制网络中流量，其主要参数如表 4-4 所示。

首先在没有使用数据优先级的条件，如果此时 3 个着陆器节点竞争信道的使用权，网络节点并不会等待相邻节点发送数据完毕再发送自己的数据，而是不断地重发数据，这样会导致节点碰撞得更加激烈。在使用了根据数据优先级进行发送的策略后，节点会依据各自数据的优先级等待不同的时间，在同样的数据速率下减少了节点的碰撞次数。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

表 4-4 数据优先级主要仿真参数

参数	值	参数	值
仿真时间（s）	200	CV_{10min}	15
数据包大小（bit）	512	代理协议	TCP
CVmin	15	时隙时长（us）	32
CV_{max}	1023		
CV_{00min}	45		
CV_{01min}	30		

从图 4-18 中可以得知，节点在使用了基于优先级的数据发送方式后，节点之间的碰撞次数较没有使用数据优先级之前明显减少了，而且随着发送速率的增大，碰撞概率更低了，有些情况下，节点的碰撞次数较之前减少了 50%以上。

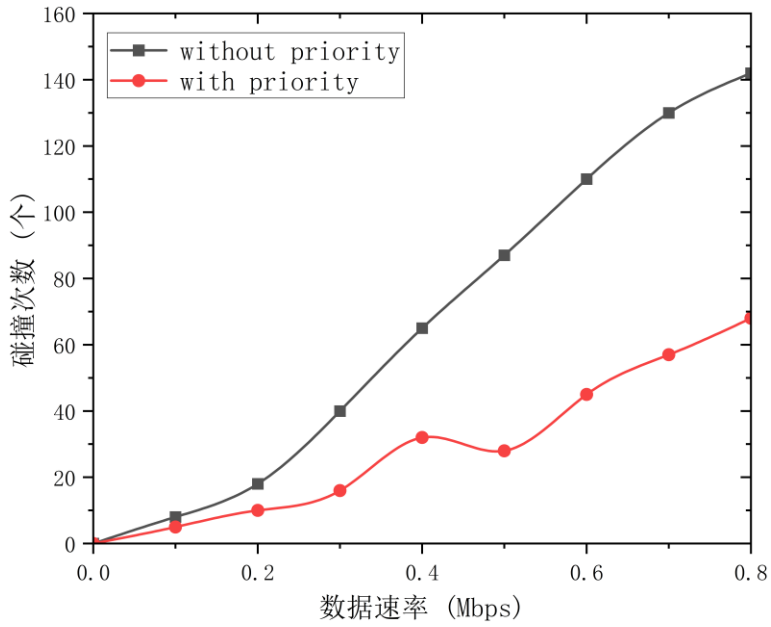

图 4-18 使用优先级数据发送方式前后碰撞次数

月球着陆器节点到地球地面站的端到端时延也是一个衡量数据优先级的指标，这里仿真的时延指的是 3 个月球着陆器到地面站北京的传输时延，即从着陆器应用层接口产生数据的时刻和到达北京测控站时刻的差值，图 4-19 为不同优先级的数据的端到端时延。

从图 4-19 中可以得知，当发送数据的速率增大时 3 个不同优先级的数据端到端时延基本上都是增大的，tag=00 的数据时延增加的最为明显，因为此数据的优

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

先级最低，CV 值最大，等待的时间也是最长的；tag=10 的数据时延较其他 2 种优先级的数据的时延增长最小，因为此数据的优先级最高，CV 值最小，等待时间大概率也是最小的，所以它能够优先获取信道的使用权，通过以太信道发送出去。

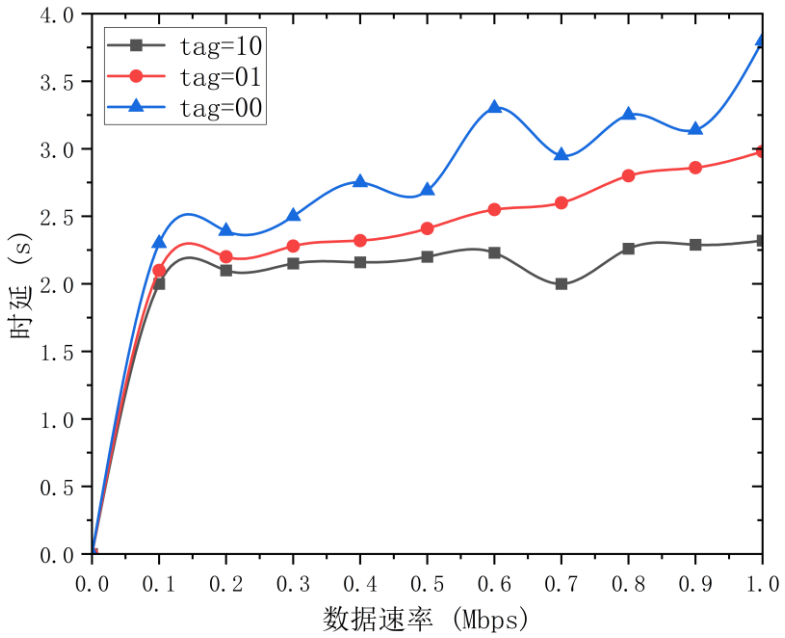

图 4-19 不同优先级数据端到端时延

从以上的仿真结果和分析来说，使用了数据优先级的发送方式之后可以有效降低节点之间发送数据的碰撞次数，也能减少月球节点到地面站的传输时延，提高了空间链路的利用率和网络吞吐量。

4.7 本章小节

本章主要对天地互联的协议转换设计进行仿真验证和协议性能指标的评价。先是对仿真软件 OMNeT++进行了介绍，包括了几个重要概念，从而引出了仿真的基本步骤，之后就是对协议转换设计了一个地月网络的仿真场景，包括了地面站、中继卫星和月球着陆器几个关键节点，并对节点模型使用的协议进行了论述，在此基础上我们对 AOS 协议进行仿真验证，结果说明了在此仿真场景下本文实现天地互联的协议转换设计，之后本章对移动 IP 进行仿真设计，并统计了接收端数据包的数量来说明了该技术的工作过程，再对对协议的性能在吞吐量、丢包率和拥塞窗口方面进行了评估，结果表明了 SCPS-TP 协议在空间通信环境下改善了协议转换系统的表现，最后对数据优先级的发送方式进行仿真，其结果表明采用该方式后可以降低传输时延，改善了空间网络的性能。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第五章 全文总结与展望

5.1 本文总结

本文研究依托于我国十三五规划中关于民用航天技术预先研究项目深空通信与服务体系研究，着眼于数据交互和信息传输需求，对通信协议进行深入的研究，完成了面向天地互联的 TCP/IP 协议和 CCSDS 协议转换，实现了在空间网络中传输 IP 数据包的需求，本文的研究内容主要包括以下几个方面：

（1）提出了基于 IP over CCSDS 的天地互联网络的设计方式

本文在研究 CCSDS 红皮书的基础上，以及根据项目的需求，结合地面网络成熟的 TCP/IP 协议，提出了 IP over CCSDS 的天地互联的协议转换方式，即在 CCSDS 空间链路中传输 IP 数据包，从 CCSDS 出版物提出的三种传输方法中选择了基于 AOS 协议的 CCSDS AOS 封装，将 IP 数据包组装成 AOS 传输帧，并构建出天地互联的总体协议栈模型，解决了地面网络和空间网络数据交互的需求。

（2）对协议转换进行了仿真验证

针对协议转换设计本文在 OMNeT++中构建了地月网络的仿真场景，设计了地面站、中继卫星和月球着陆器等节点，根据 CCSDS AOS 红皮书进行了 AOS 帧结构的设计，对于 IP 数据包在封装成传输帧和从 AOS 传输帧提取 IP 数据包时存在的一些问题，如 IP 数据包的切割和拼接等进行了说明，并在仿真软件中进行 AOS 帧的实现，验证了 AOS 协议的可行性，即实现了 IP over CCSDS 的协议转换。

（3）对移动 IP 技术进行了仿真实现

对于空间通信中卫星的高速移动的链路切换导致 IP 路由问题，本文设计了移动 IP 的技术，通过实现代理发布、请求注册和隧道等功能，解决了因为链路切换使节点之间无法通信的问题，并且解决了移动 IP 中三角路由问题，让协议转换系统具有鲁棒性。

（4）改进了传统 TCP 协议在空间通信中的性能

在空间通信中存在时延长、误码率高、前向和反向链路不对称的问题，严重影响了 TCP 协议的传输效率，因此本文针对项目的需求研究了 CCSDS SCPS 红皮书，设计了 SACK 机制、SNACK 机制和 Vegas 算法，即实现了 SCPS-TP 协议，并在 OMNeT++中进行了 TCP 协议和 SCPS-TP 协议的比较仿真，仿真结果表明 SCPS-TP 协议在吞吐量、丢包率和拥塞控制方面的空间性能均优于 TCP 协议，SCPS-TP 协议更适合空间通信。

（5）实现了数据优先级的协议功能

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

在一般的地月传输中并没有使用数据优先级的发送方式，本文在实现了协议转换后，引入了数据优先级的发送方式，在发生紧急事件的情况下可以优先将事件数据发送到地面站，使整个协议转换系统更加完整，仿真结果表明数据优先级降低了节点间发送数据碰撞的概率，同时减少了端到端的传输时延。

5.2 后续展望

创建深空通信服务体系是一个长期的艰巨的任务，本文设计的协议转换的系统还是较为简单，对于 CCSDS 协议的实现还不够，在下一阶段的研究希望能在以下几个问题开展深入探究：

（1）针对 AOS 协议提供 7 种服务，本文实现了最为基本的，只能传输数据，对于语音、视频、图片等信息没有涉及，而且为了方便实现 AOS 帧，并没有进行差错控制和 ARQ 协议方面的研究，在高误码率下无法很好提升传输的效率。

（2）对 IP 技术的移动局限性，本文只是进行初步的移动 IPv4 的研究和仿真，但是移动 IPv4 同时也存在一些问题，如 IPv4 的地址空间较少，对于空间网络中越来越多的节点来说是不太充分，而且 MN 切换时需要在 LA 进行绑定跟新，跟新完成之后才能绑定，空间网络中延时较长，导致传输效率低下。

（3）对于 SCPS-TP 协议的实现还是不够充分，比如还可以对 SCPS-TP 协议的首部进行压缩，变成 TCP 协议的一半，这对于传输带宽小的反向链路来说是极其友好的；还有可以初始的拥塞窗口进行扩大，有利于提升传输的数据量，拥塞窗口值在初始时是 1MSS，可以根据实际情况，适当的扩大初始的窗口值；Vegas 算法在前向和方向链路不对称时，由于反向链路测量的 RTT 长，导致拥塞窗口值很小，浪费了链路带宽，下一步的研究计划可以针对这个问题对 Vegas 算法进行一定的改进。

（4）本文对于 CCSDS 应用层协议只是提供了接口，并没有具体的协议，下一步可以对 CFDP、FTP 等应用层协议进行研究。

（5）本文只是实现了最为基本协议转换，仿真场景有些简单，节点的数量不够多，对于卫星组网和星间链路等情况没有考虑，而且缺少对于空间路由算法的设计。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

致　谢

研究生三年的学习生活还有几个月就要结束了，在成电不知不觉度过了七年的时间，回想起研一入学时的场景彷佛还是不久之前的事情，这三年的时光里在学习和生活中遇到了不少贵人，在我需要帮助的时刻给与善意和助力，在本文的最后部分，我想向他们表示真诚的谢意。

首先感谢我的两位导师李雪教授和蒋大钢副教授，感谢李老师对我的接纳，让我有幸走入科研的道路，感谢蒋老师在平时的学习中适时的解惑，您的指导使我反思许多，收获成长，还有感谢您在毕业论文的整体设计上的指导。感谢两位老师，在我迷茫困惑时给与提醒和帮助，衷心地感谢您们。

感谢辅导员林伯先老师，林老师为人热心，能够站在学生的角度为我们考虑，谢谢您在生活上对我的帮助和照顾。

感谢我的师兄薛林山博士在科研项目宽带协同导航中的指导，在实现信号的跟踪和捕获过程我遇到了许多困难，感谢你不厌其烦与我们讨论，帮助我解决问题，还有谢谢师兄在毕业设计上给我指明了方向，你的乐观精神也让我感悟不少，感谢你。我在做项目中有不少困惑，感谢我的同学董佳林耐心地给我解释，同时指出了我学习方式的不足之处，让我意识到自己的问题，谢谢你。还要感谢已经毕业的师兄任纪缘，谢谢你在毕业之前对我在项目上的指引，让我进入了测控通信的世界。

感谢重庆大学张寅森和王璐璐同学在协议转换设计上给与的帮助。

最后感谢我的父母和家人们，谢谢你们一直以来的理解和支持。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

参考文献

[1] 国防科工局.民用航天“十三五”预研第二批项目指南表.B0110 深空通信与信息公共服务体系研究[EB/OL].2018-7-17, http://www.cnsa.gov.cn/n6758823/n6758839/c6802367/content.html.

[2] 郭林峰. 空间通信网络的连接计划设计研究[D].重庆邮电大学,2020.18-22.

[3] 李严. IP over CCSDS 空间组网链路传输协议设计[D].西安电子科技大学,2011.9-11.

[4] 赵运弢. 面向流量相关性的高级在轨系统复用及优化技术[M].国防工业出版社,2018.

[5] 王若楠. AOS 中基于泊松流的帧生成与虚拟信道调度算法研究[D]. 沈阳理工大学, 2016.

[6] 叶晓国, 肖甫, 孙力娟,等. SCPS/CCSDS 协议研究与性能分析[J]. 计算机工程与应用, 2009, 45(4):4.

[7] 王金苗, 叶建设, 许鹏文. 卫星链路上 SCPS-TP 协议中的 SNACKTCP 性能分析[J]. 中国科技信息, 2011(10):1.

[8] 董绍进. LEO 卫星星座路由算法研究与仿真[D]. 国防科学技术大学,2011.17-19.

[9] 宋敬彬. Linux 网络编程[M]. 北京：清华大学出版社, 2010.

[10] 李领治. 实用计算机网络教程[M]. 清华大学出版社, 2012.

[11] 谢希仁. 计算机网络第七版[M]. 北京：电子工业出版社,2017.219-241.

[12] Klaus Wehrle, Jochen Reber. INET Framework for OMNeT++[R]. the University of Karlsruhe.2013.89-111.

[13] W.RICHARDSTEVENS. TCP/IP 详解.卷 1,协议[M]. 机械工业出版社, 2000.

[14] 黄永峰, 李星. 计算机网络教程[M]. 北京：清华大学出版社, 2006.

[15] 苏钦. 大学计算机基础教程[M]. 北京：人民邮电出版社, 2007..

[16] 孙志颖. 天地一体化网络协议的研究与仿真[D].西安电子科技大学,2012.7-9.

[17] 蒋立正. IP over CCSDS 空间组网通信关键技术研究[D].中国科学院研究生院（空间科学与应用研究中心）,2009.7-14.

[18] 潘彩平. HLA-RTI 平台下的 CCSDS 物理层仿真系统研究[D].沈阳理工大学,2018.13-15.

[19] 王翔,申志伟,朱肖曼,高松林,郭烁.卫星互联网组网技术研究[J].信息通信技术,2021,15(02):36-43+64.

[20] 田野. 高级在轨系统中的多路复用与差错控制技术研究[M]. 科学出版社, 2014.

[21] 陆静. 空间数据传输领域高级在轨系统 ISO 国际标准的航天应用研究[D]. 北京邮电大学, 2009.

[22] 王为奎, 黄强, 张孝虎. 高级在轨数据系统的研究与应用[C]// 第二届中国卫星导航学术

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

年会电子文集. 2011.

[23] 黄姝娟. 空间实验室数据管理系统网络通信协议的研究[D].西北工业大学,2005.9-30.

[24] 闫朝星,付林罡,谌明,朱至天.天地信息网络协议融合技术综述[J].遥测遥控,2020,41(06):30-38.

[25] 李勇. 基于 CCSDS 的空间网络通信协议研究与实现[D].电子科技大学,2011.16-24..

[26] Recommendation for Space Data System Standards. AOS Space Data Link Protocol: CCSDS 732.0-B-4[S]. BLUE BOOK October 2021.

[27] 白海斌. 空天信息专用网络高速网关设计与实现[D].西安电子科技大学,2015.36-40.

[28] 邓浩. 深空环境下 AOS 协议研究[D].华中科技大学,2013.14-23.

[29] Recommendation for Space Data System Standards. IP over CCSDS Space Links: CCSDS 702.1-B-1[S]. BLUE BOOK September 2012.

[30] 张胜磊,刘淑茜,张坚.基于 IP over CCSDS 的空间通信网络天地一体化研究[J].载人航天,2008(03):22-25+42.

[31] 郑东旭. 卫星导航系统传输带宽优化技术研究[D]. 沈阳理工大学.2021.

[32] 杜长刚. IP over CCSDS 适配器高速接口电路设计与实现[D]. 西安电子科技大学.2014.

[33] 王博. 符合 CCSDS 标准的遥测采编和遥控译码单元[D]. 中国科学院大学.

[34] 吴迎松. AOS 演示验证系统接收端设计与实现[D]. 内蒙古工业大学, 2009.

[35] 卢婷. 高速实时/回放分级复接器的设计与实现[J]. 空间技术部, 2008.

[36] 张庆君, 郭坚, 董光亮. 《空间数据系统》(第二版)[J]. 国际太空, 2016, 000(010):73-73.

[37] 李思. 基于 IPsec 协议的移动 IP 网络安全方案研究[D].东北大学,2015.14-26.

[38] 王军. 卫星通信中基于带宽估计的 TCP 传输控制算法研究[D].华中科技大学,2008.64-66.

[39] 丁锐. SCPS-TP 协议在空间通信中的研究与仿真[D].电子科技大学,2011.26-54.

[40] 秦楠. 基于多 Agent 系统的网络拥塞控制若干问题的研究[D].同济大学,2007.36-53.

[41] 黄科. TCP 协议在空间通信中的应用研究[D].国防科学技术大学,2009.20-22.

[42] 杨丽圆. 卫星网络中传输层确认机制的研究[D]. 沈阳理工大学.

[43] Recommendation for Space Data System Standards. Space Communications Protocol Specification(SCPS0)-Transport Protocol(SCPS-TP): CCSDS 714.0-B-2[S]. BLUE BOOK October 2006.

[44] 姚向朋. 卫星网络 TCP 算法研究[D].华中科技大学,2007.

[45] 李雪梅. 天地一体化异构网络融合技术研究[D].西安电子科技大学,2018.56-76.

[46] 袁翰林. 异构无线网络中 TCP 性能的研究与改进[D]. 南京邮电大学, 2012.

[47] 卢由. 超短波通信协议 MAC 层控制帧解析及退避算法仿真[D]. 电子科技大学, 2016.

[48] https://omnetpp.org/intro/

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[49] 王俊义 ... OMNeT++网络仿真[M]. 西安：西安电子科技大学出版社, 2014.

[50] 赵永利, 张杰. OMNeT++与网络仿真[M]. 人民邮电出版社, 2012.

[51] Kim Barrett, Bob Cassels. OMNeT++ Simulation Manual[R]. New York,NY,USA,2015.

[52] Klaus Wehrle, Jochen Reber. INET Framework for OMNeT++[R].10-58.

[53] 李海燕, 周克兰, 吴瑾. 大学计算机基础[M]. 清华大学出版社, 2013..

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net