



专题：星地融合移动通信及卫星互联网

基于OMNeT++的低轨卫星时间确定性 网络仿真平台设计

赵崇尧, 孙耀华, 冯昕澳, 彭木根

(北京邮电大学网络与交换技术全国重点实验室, 北京 100876)

摘要: 为了满足实时和高可靠数据传输需求, 地面网络以太网标准引入时间敏感网络 (time-sensitive networking, TSN), 通过流量整形、流量调度等手段有效降低时延抖动。然而, 地面网络的部署受限于环境因素, 难以覆盖偏远地区和空中服务对象, 无法满足对时延抖动的严格要求。考虑低轨卫星网络具有广域立体覆盖优势, 开展了TSN技术在低轨卫星网络中的适用性评估。利用OMNeT++构建了低轨通信网络时间确定性技术仿真平台, 详细介绍了仿真平台各功能模块的设计原理与工作流程。同时, 针对关键机制进行了深入分析。通过仿真评估, 验证TSN技术在保障低轨卫星网络时间敏感业务端到端时延、时延抖动、数据包到达率的性能优势。

关键词: 低轨卫星网络; 时间敏感网络; 星间确定性传输

中图分类号: TN927.2

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2025140

Design of time-deterministic network simulation platform for low-orbit satellites based on OMNeT++

ZHAO Chongyao, SUN Yaohua, FENG Xin'ao, PENG Mugen

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: In order to meet the demand for real-time and highly reliable data transmission, the Ethernet standard for terrestrial networks introduces time-sensitive networking (TSN), which effectively reduces delay jitter by the means of traffic shaping and traffic scheduling. However, the deployment of terrestrial networks is limited by environmental factors, and it is difficult to cover remote areas and airborne service objects, which can't meet the strict requirements for delay jitter. Considering that low-orbit satellite networks have the advantage of wide-area three-dimensional coverage, the applicability assessment of TSN technology in low-orbit satellite networks was carried out. OMNeT++ was used to construct a simulation platform for time-deterministic technology in low-orbit communication networks, and

收稿日期: 2025-02-27; 修回日期: 2025-04-27

通信作者: 赵崇尧, cyzhao@bupt.edu.cn

基金项目: 中国通信学会青年人才托举项目 (No.2021QNRC001)

Foundation Item: Young Elite Scientist Sponsorship Program by China Institute of Communications (No.2021QNRC001)



the design principles and workflow of each functional module of the simulation platform were introduced in detail. Meanwhile, the key mechanisms were analyzed in depth. Through simulation evaluation, it verifies the performance advantages of TSN technology in guaranteeing the end-to-end delay, delay jitter, and packet arrival rate of time-sensitive services in low-orbit satellite networks.

Key words: low-orbit satellite network, time-sensitive networking, interplanetary deterministic transmission

0 引言

随着卫星制造与火箭发射技术的不断进步，低地球轨道（low earth orbit, LEO）卫星凭借其覆盖范围广、通信时延低、组网灵活等优势成为未来6G天地一体化网络建设中的关键一环^[1-6]。由于低轨卫星网络具有全域全立体空间泛在连接能力，除了传统的尽力而为业务外，偏远地区远程医疗、远程交互式视频业务、跨域无人平台控制等对时延抖动具有严格要求的时间敏感业务也将成为卫星网络承载的重要业务种类。因此，如何为此类时间敏感业务提供大空间尺度确定性传输保障成为提高天基网络服务能力的重要发展方向^[7]。

为了满足地面互联网中时间敏感业务的确定性传输需求，通信协议集合（IEEE 802.1）成立了时间敏感网络（time-sensitive networking, TSN）工作组，旨在扩展标准以太网802.3。TSN的目标是在满足对高实时性和确定性通信需求的同时，保持与传统以太网的兼容性，从而实现关键业务流量与普通以太网流量的融合传输，广泛适用于工业控制、车载通信、航空航天、音视频等多种场景。中国互联网产业联盟（AII）于2019年10月发布了《时间敏感网络（TSN）产业发展报告》^[8]，并于2020年8月发布了《时间敏感网络（TSN）产业白皮书》^[9]，标志着时间敏感网络成为工业互联网基础设施转型升级的新方案。TSN实现确定性低时延传输的关键技术包括流量分类、流量整形、流量调度和流量抢占等。其中流量分类将业务流量分成时间触发（time triggered, TT）流量、音视频桥接（audio video bridging, AVB）

流量和尽力而为（best effort, BE）流量^[10]。其中TT流量具有最高优先级，需要保障其准时到达，实现端到端低时延、低抖动的确定性传输；尽管AVB流量确定性低于TT流量，但仍需要控制有限的端到端时延；BE流量没有严格的时延要求，主要用于数据传输和储存等不需要实时性的应用。流量整形的主要目的是减少网络中的流量突发性和拥塞问题，常见的整形技术包括异步流量整形（asynchronous traffic shaping, ATS）、基于信用的整形（credit-based shaping, CBS）和时间感知整形（time-aware shaping, TAS）。流量调度通过精确控制数据发送的时隙，保证时间敏感流的确定性传输。流量抢占允许高优先级数据包在传输时抢占低优先级数据包，从而解决了高优先级帧需要等待其他低优先级帧传输完毕的优先级反转问题。

然而上述TSN技术主要面向地面相对静态的网络拓扑设计，与地面网络相比，低轨卫星网络具有链路传播时延稳定、拓扑连接关系时变等特性，评估TSN技术在动态卫星网络中的性能成为关键挑战。当前，已有大量研究团队开始推进类似仿真平台的搭建工作。文献[11]设计并实现了基于OPNET的LEO卫星通信仿真平台，研究了卫星多波束天线模型并进行建模，通过部署移动地面节点接收卫星信号的方式进行仿真。文献[12]基于NS-3设计了针对低轨卫星网络路由协议。文献[13]利用OMNeT++模拟卫星通信环境，评估卫星部署方案对系统吞吐量的影响。文献[14]基于OMNeT++和INET仿真框架将拥塞控制算法集成，验证了拥塞控制策略可以适用卫

星通信。

然而,上述低轨卫星网络仿真平台都是基于“尽力而为”的底层逻辑模拟数据包逐跳转发,不支持TSN中的流调度与队列调度机制,无法支撑TSN技术在低轨卫星网络中的性能验证。因此,本文利用OMNeT++构建了低轨卫星通信网络时间确定性技术仿真平台,并介绍了仿真平台中各功能模块的设计方法与工作流程。首先,介绍OMNeT++平台的优势、仿真架构和运行原理,然后,进一步介绍仿真平台中收发端节点、路由交换节点、链路模块基础功能,以及传输控制协议(transmission control protocol, TCP)重传、优先级队列、星间链路传播时延动态模拟等功能的实现方法与工作原理。最后,利用该平台搭建低轨卫星网络并进行TSN方案性能验证,通过统计数据包端到端时延、时延抖动、到达率等各项性能,证明与其他方案相比,TSN技术在低轨卫星网络中能有效保障时间敏感业务的各项服务质量(quality of service, QoS)需求。

1 仿真平台架构

1.1 OMNeT++仿真软件介绍

OMNeT++是一个开源的、模块化的、基于组件的仿真平台,广泛应用于网络建模、仿真与分析。该软件采用C++语言编写,支持多种操作系统运行,并提供了一系列强大的仿真和分析工具,如可视化工具、调试器以及结果分析工具,能够对仿真结果进行深入的分析 and 处理。此外,OMNeT++的文件结构将网络仿真不同层面分离,使仿真运行流程更加清晰易懂。模型行为写在C++文件中,模型拓扑由NED文件定义。此外,在仿真过程中经常要研究改变输入变量对仿真的影响,这些输入写在ini文件中,不属于C++文件和NED文件,这使参数研究变得更简洁。此外,仿真结果很容易用内置的分析工具获取和处理。

通常来说,OMNeT++的工作流程可总结如

下:(1)在创建模型时,需要将进行仿真的系统映射为一个由通信模块组成的层次结构;(2)用NED语言定义模型结构,即仿真系统的网络拓扑;(3)表示协议头的C++类在MSG文件中进行描述;(4)提供合适的文件(NED文件或ini文件)来保存OMNeT++配置和仿真模型的参数。一个ini文件可能包含多个配置信息,这些配置可以相互构建;(5)构建并运行仿真程序,OMNeT++仿真内核支持通过多种用户界面(包括命令行界面和交互式图形界面)与用户进行交互,实现仿真控制与结果可视化;(6)仿真结果被写入输出矢量和标量文件中,在仿真IDE中可以分析和绘制结果(由Pandas和Matplotlib支持的分析工具),查看仿真事件日志。此外,结果文件是基于文本的,可通过R语言、MATLAB或其他工具处理。

当前,OMNeT++已被广泛用于网络仿真,不同领域的研究者为其编写了一系列仿真模型和框架,包括Internet协议、无线网络、移动自组织网络、无线传感器网络、车联网等。大部分模型框架是开源的,并作为独立项目开发。OMNeT++对TSN支持度较高,相关研究有:Simon等^[15]利用OMNeT++仿真平台研究了TSN流调度机制在工业自动化和移动前向网络中的应用;Heise等^[16]提出了一种基于OMNeT++的TSN仿真框架,对TSN中的帧复制与消除机制以及TSN帧抢占机制进行了仿真。

类似的网络仿真软件还包括OPNET和NS-3等。其中,OPNET是一款用于网络建模、仿真和性能评估的商业软件。该软件采用分层建模机制,将网络划分为拓扑层、节点层和进程层,分别对应实际拓扑、设备和网络协议,从而全面反映网络特性。OPNET的显著优势是能够为网络规划与设计提供可靠定量分析支持,进而验证现有方案或对比多种不同设计方案的效果。OPNET被广泛应用于各种网络的性能仿真,Pahlevan



等^[17]使用 OPNET 对 TSN 中 IEEE 802.1Qbv 和 802.1Qci 流量控制机制进行了仿真，并分析其时延特性。NS-2 是 NS 系列早期的开源版本，其建模方式基于 Tcl 脚本和 C++ 扩展，在无线网络等研究中发挥过重要作用。然而其模型架构松散，模块间耦合性差，缺乏图形界面与模块化组织机制，开发复杂、调试效率低，且对 TSN、工业通信等新兴技术支持程度较弱。在新的应用需求背景下，NS-2 逐渐被 NS-3 与 OMNeT++ 取代。NS-3 是一款开源网络仿真软件，支持多种网络协议和技术的建模与仿真，包括无线网络、移动网络、卫星网络、蜂窝网络、数据中心网络、以太网和 IP 协议等。NS-3 基于 C++ 语言编写，并提供 Python 接口，方便用户进行扩展和使用。NS-3 的关键概念包括节点 (node)、信道 (channel)、网络设备 (NetDevice) 和应用程序 (application)。仿真过程通常包括节点生成、网络设备和协议栈安装、应用层协议配置以及节点移动和能量管理等，最后启动仿真并分析结果。Krummacker 等^[18]使用 NS-3 对 TSN 中 IEEE 802.1Qbv 机制进行了仿真分析。

OPNET、OMNeT++ 以及 NS-3 都是基于离散事件的仿真软件，各自具有独特的特点和优势。网络仿真软件及其特点见表 1。首先对比 OPNET 和 OMNeT++，OPNET 是一种商用仿真平台，具有极为丰富的内置仿真库，能够建模多种现成的通信协议、网络拓扑与设备行为，适用于运营商

级别或企业网络的规划和评估。其模型结构以图形驱动为主，基于 C 语言实现特有应用程序接口 (application program interface, API) 开发，支持有限程度的行为重构。然而，OPNET 的模型拓扑结构相对固定，不具备高度参数化和模块化设计能力，限制了其在高度可定制场景中的扩展性。相较之下，OMNeT++ 拥有开放源码、模块化建模、强耦合的消息调度机制及丰富的继承机制，使其在复杂协议建模、分层网络结构设计等方面表现出显著的灵活性。尤其是 OMNeT++ 中的 NED 网络描述语言，结合 C++ 后端，可实现对节点行为、拓扑结构、消息处理逻辑的精准控制。同时，OMNeT++ 在协议栈自定义能力、动态拓扑支持能力、TSN 建模能力等方面显著优于 OPNET，也因其开源特性，具有更低的使用门槛和更高的可扩展性。然后对比 NS-3 和 OMNeT++，NS-3 的仿真内核和模型是不可分割的整体，侧重真实协议栈建模，强调与 Linux 网络协议栈一致的系统行为，更像仿真软件；而 OMNeT++ 更像仿真平台，提供了完备的模块组织机制、事件调度框架和图形化配置接口。开发者可以在其基础上构建个性化仿真框架（如 INET、Veins、Simu5G、CoRE4INET），而不仅限于使用已有模型。正因如此，OMNeT++ 形成了丰富的仿真生态系统，支持从无线通信到有线工业网络的广泛应用，其代码复用能力和多场景

表 1 网络仿真软件及其特点

	OPNET	OMNeT++	NS-3	NS-2
性质	商业软件	开源软件	开源软件	开源软件
语言	C++	C++	C++	C++、TCL 脚本
平台	Windows	Windows、Linux 等	Windows、Linux 等	Windows、Linux 等
图形用户界面	是	是	是	否
分析工具	是	是	是	否
计算效率	高	高	低	中
仿真框架数量	少	多	中	少
支持协议数量	少	多	中	中
支持 TSN 程度	中	高	低	低

兼容性远优于 NS-3。虽然 OMNeT++ 和 NS-3 都是开源软件，但在 NS-3 中针对 TSN 的仿真框架较少，缺乏完备的调度机制、网关行为、时钟同步等模块，不利于进行 TSN 仿真。而 OMNeT++ 支持多个不同的 TSN 仿真框架，可以快速建立 TSN 仿真网络^[10]。同时，在 OMNeT++ 中提供了特定接口实现半实物仿真，可以对 TSN 进行更真实的仿真。因此，本文利用 OMNeT++ 的系列优势，实现低轨卫星时间确定性网络数字仿真平台开发，用于评估 TSN 技术在低轨卫星网络中的性能水平。

1.2 平台架构设计

搭建的低轨卫星时间确定性网络数字仿真平台架构设计如图 1 所示，主要包括场景配置模块、卫星运动模拟模块、路由控制模块、网络状态信息感知模块、路由交换模块，整个仿真过程由离

散时间驱动。具体讲，场景配置模块由网络节点、网络拓扑构成，其中网络节点包括用户终端、LEO 卫星、移动交换中心、网络控制中心，网络拓扑包括星间微波链路和激光链路。用户终端可发送和接收不同优先级的数据包，LEO 卫星可以处理和转发收到的数据包，微波链路和激光链路通过不同速率实现星间通信。卫星运行模块通过轨道文件读取、参数解析、轨道计算获取卫星运行信息。网络状态信息感知模块通过网络测量和故障检测满足用户 QoS 需求。路由交换模块包括时间同步、优先级队列、门控、帧抢占等一系列 TSN 协议。将场景配置模块、卫星运行模块、网络状态信息感知模块、路由交换模块获取到的信息分别输入路由控制模块，进行拓扑管理和链路控制，进而控制星间路由，为用户提供 QoS 保障。

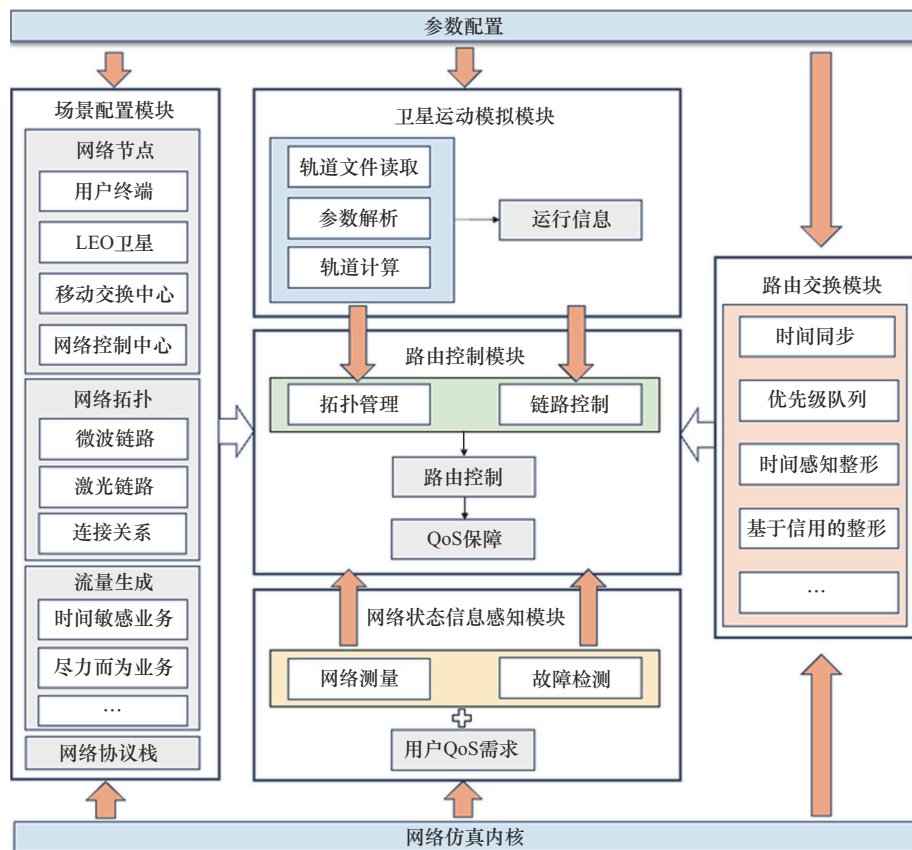


图 1 仿真平台架构设计



1.3 平台运行流程

仿真平台运行流程如图2所示，其中控制节点主要完成路由规划、门控列表生成等工作，之后将路由信息基于分段路由协议封装至源节点，将门控列表配置于各卫星节点。卫星节点根据门控与业务优先级对数据包进行转发，接收端对接收到的数据包进行解析与处理。同时，仿真平台中还进一步考虑星间通信高误码率特性，设计了反馈重传与TCP保障数据包可靠传输。系统每隔一定时间对节点信息进行更新，当仿真时间到达时结束仿真，输出相应仿真结果用于后续分析。

下文针对发送节点、接收节点、链路模块、路由交换节点具备的功能函数进行阐述，同时说明这些节点间的数据接口，从而保障仿真平台按上述流程正常工作。

(1) 发送节点

① 配置文件加载函数：读取XML格式的初始化配置文件，获取数据包大小、数据包发送周期、数据包发送间隔及发送计划等配置信息，并将配置信息写入调度计划。

② 调度计划执行函数：根据配置文件生成的调度计划，依照平台仿真时间执行相应的调度

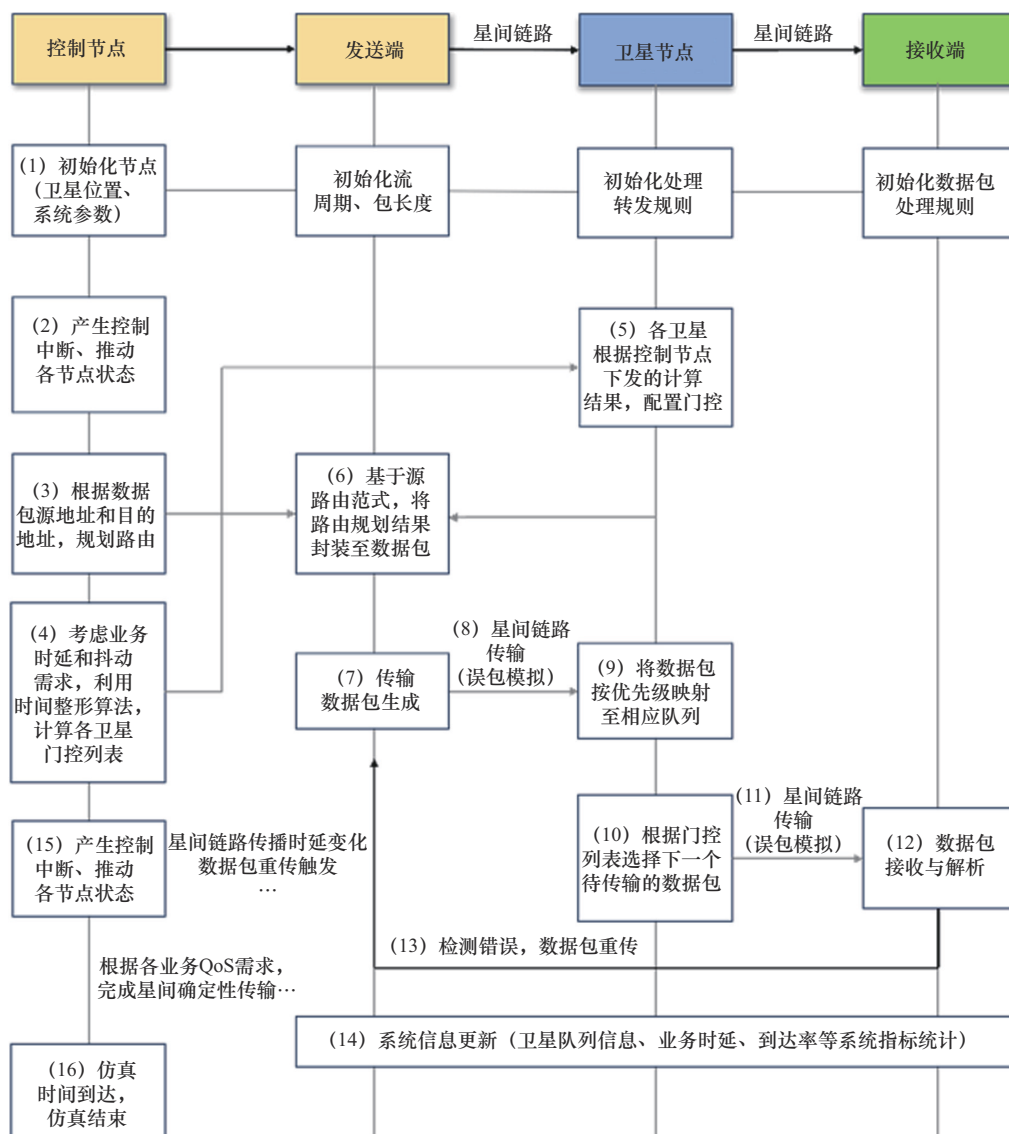


图2 仿真平台运行流程

计划操作。

③ 发送间隔计算函数：根据当前的发包调度计划，计算出距离下一次发包的时间间隔，返回计算的时间间隔。

④ 数据包生成函数：根据当前的发包调度计划，生成配置文件中指定大小的数据包，并设置相应的包信息。

⑤ 优先级定义函数：将生成的数据包放入指定的优先级队列并进行发送，实现数据包优先级定义。

⑥ TCP重传函数：建立已发送数据包队列和已接收数据包队列，通过两队列对比生成数据包传输队列，并对数据包发送时间进行记录，当数据包位于传输队列时间超过数据包重传定时，将数据包放到重传队列，再次发送数据包实现TCP重传功能。

(2) 接收节点

① 接收响应函数：实时监听数据包到达，当数据包到达后，对数据包进行解析，获取数据包携带信息。

② 传输仿真函数：模拟数据包传输过程中丢包、误包等情况，通过随机丢弃一部分数据包，实现数据包传输模拟。

③ TCP反馈函数：针对正常接收到的数据包，对发送端进行接收反馈，确认数据包接收状态，关闭该数据包的TCP重传计时。

(3) 链路模块

① 星间链路模拟函数：模拟星间链路，确定卫星运动规律和链路特性。

② 时延计算函数：根据卫星运动规律和链路特性，计算链路传播时延和传输时延。

(4) 路由交换节点

① 路由文件加载函数：读取XML格式的初始化路由规划配置文件，加载配置文件中转发表与门控表等相关信息。

② 单播地址解析函数：从路由规划配置文

件中提取地址元素，验证并解析每个元素的介质访问控制(media access control, MAC)地址和端口号，并将这些信息存入路由节点的转发数据库，建立单播转发规则。

③ 多播地址解析函数：类似地，解析路由规划配置文件中的元素并建立多播转发规则。多播地址可能对应多个端口，因此在此函数中增加了处理端口列表功能。

(5) 数据接口

各节点功能组成与节点间的数据接口说明如图3所示。首先，发送端节点的配置文件加载函数读取XML格式的初始化配置文件，获取数据包大小、数据包发送周期、数据包优先级及发送计划等配置信息，并将配置信息写入调度计划，调度计划执行函数根据配置文件生成的调度计划，依照平台仿真时间执行相应的调度计划操作，其中，发送间隔计算函数根据当前的发包调度计划，分别计算出距离下一次发包的时间间隔和返回计算的时间间隔；优先级定义函数将生成的数据包放入指定的优先级队列并进行发送，实现数据包优先级定义。数据包生成函数根据时间间隔、优先级值(priority code point, PCP)、数据包大小等配置信息和当前发包调度计划，生成TCP数据包，设置相应的包信息，并将发包状态传递给TCP重传函数，建立已发送数据包队列，与接收端已接收数据包队列对比生成数据包传输队列，并对数据包发送时间进行记录，当数据包位于传输队列时间超过数据包重传计时，将数据包放入重传队列，再次发送数据包实现TCP重传功能。

路由交换节点中的路由文件加载函数读取XML格式的初始化路由规划配置文件，加载配置文件中针对本交换机的路由规划相关信息。入队函数根据PCP、门控等配置信息将生成的数据包放入指定优先级队列并进行发送。待数据包出队列后，地址解析函数从路由规划配置文件中提



取地址元素，验证并解析每个元素的MAC地址和端口号，并将这些信息存入路由节点的转发数据库，建立单播转发规则，并通过转发函数转发至相应模块。

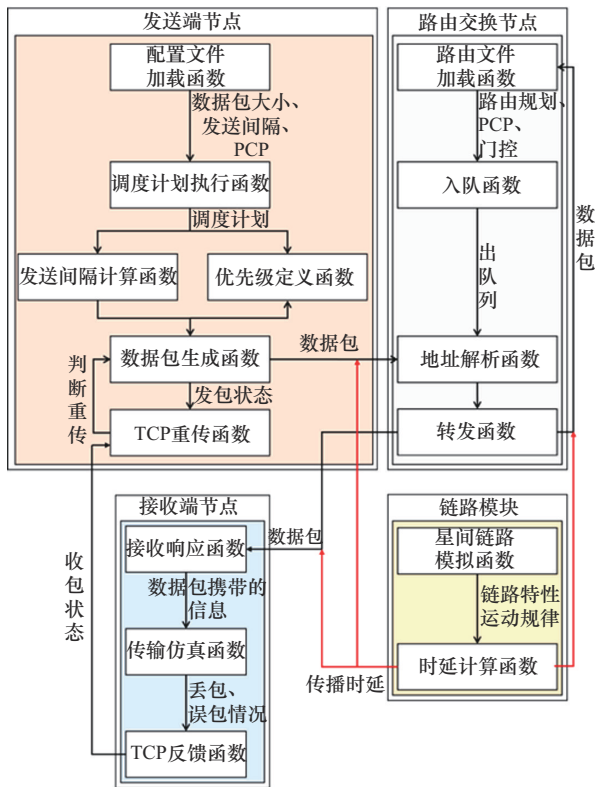


图3 各节点功能组成与节点间的数据接口说明

链路模块中，星间链路模拟函数用于模拟星间链路，确定卫星运动规律和链路特性。时延计算函数根据卫星运动规律和链路特性，计算链路传播时延和传输时延。

接收节点中，接收响应函数实时监听数据包到达，当数据包到达后，对数据包进行解析，获取数据包携带信息。传输仿真函数根据数据包携带信息模拟数据包传输过程中丢包、误包等情况，通过随机丢弃一部分数据包，实现数据包传输过程中的问题模拟。TCP反馈函数根据丢包、误包情况和正常接收到的数据包生成已接收数据包队列，对发送端进行接收反馈，确认数据包接收状态，并关闭该数据包的TCP重传计时。

2 节点和链路模块的工作流程与实现方法

2.1 发送端节点实现原理

仿真平台中的发送端节点不但具备发送数据包的功能，还可以控制发送数据包的时间间隔、包大小以及目的地址，从而完成对TSN相关技术的仿真实现。

设计的发送端节点功能如图4所示。根据OMNeT++仿真平台的运行原理，在仿真开始前，首先设定发送端的发送数据包间隔、数据包大小、目的地址和数据包优先级。然后，在仿真运行期间，定时模块遵循设置的发送间隔，按时调用数据包生成模块，根据设定数据包大小、目的地址、传输优先级生成数据包。最后，将数据包传输到主机网口，发送至传输网络中。

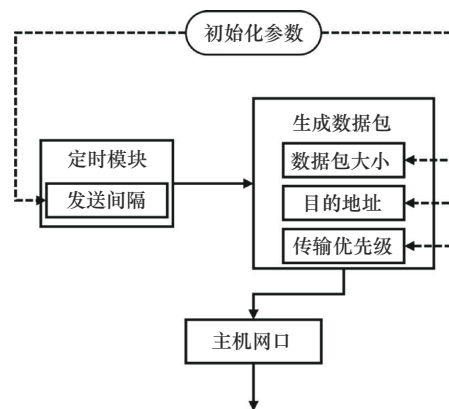


图4 发送端节点功能

数据包周期性发送的实现方式如图5所示。仿真平台首先解析配置文件中的内容，获取发送数据包的循环时间和指定的发送间隔。将解析出来的发送时间以队列形式存储，创建一个变量作为指示当前发送时间的索引。根据索引计算每次发包间隔，根据时间间隔设置自信息，然后根据自信息的到达事件触发发送节点的发包事件。每完成一次发包后，索引值加1，完成整个循环后，索引值设为0。由此实现发送周期和发送间隔的设置。

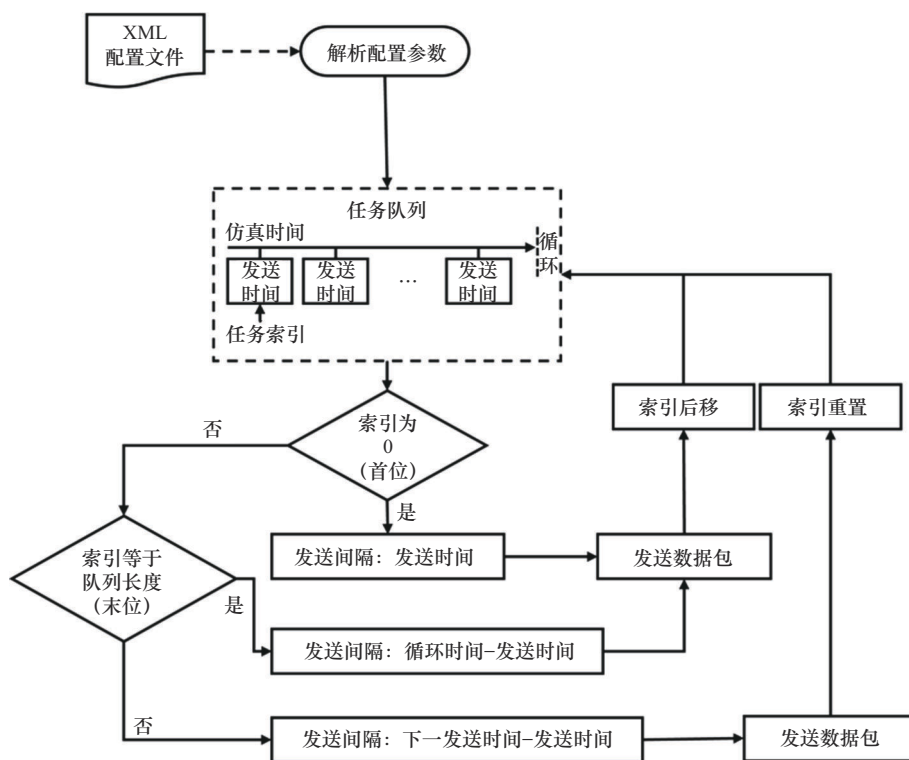


图 5 数据包周期性发送的实现方式

发送数据包的大小由配置参数设置。设置发送数据包的大小如图6所示，首先从配置文件中读取数据包大小，然后根据设置开辟指定大小数据包的内存空间，最后对此数据包设置其他参数并发送出去。

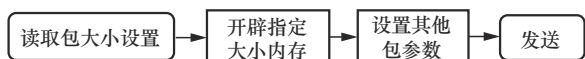


图 6 设置发送数据包的大小

数据包指定发送队列优先级根据配置文件设置。PCP 处理如图 7 所示，首先读取配置文件中针对各发送端配置的 PCP，然后根据 PCP 通过映射表得到优先级队列索引值，最后根据队列索引发送到相应的优先级队列进行传输。

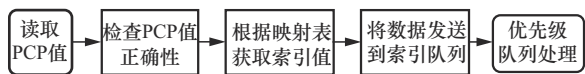


图 7 PCP 处理

2.2 接收端节点实现原理

在仿真平台中，接收端节点需要具备数据包到达检测与内容处理功能，以满足基本的数据包

接收需求并支持后续性能评估与功能扩展。当数据包到达接收端节点时，通过解析数据包内容，提取其源地址、包编号等关键信息，统计端到端时延等仿真性能指标，并为其他模块（如 TCP 重传模型、发送端模块）提供必要参数。同时，为模拟实际网络中的丢包现象，接收端节点还引入一定的随机丢包机制。对于成功接收的数据包，接收端会向发送端发送接收确认信号，以终止该数据包的重传计时过程。接收端节点功能如图 8 所示，确保接收节点在仿真中的功能完整性与扩展性。

数据包接收端的实现如图 9 所示，当接收的数据包到达后，首先解析数据包内容，然后通过生成一个 0~1 的随机数模拟传输过程中丢包、传输超时等现象。若生成随机数小于丢包率，则认为该数据包丢失，将数据包删除，不做其他处理；若生成随机数大于丢包率，则认为接收端成功收到数据包，根据包内容计算该数据包的端到



端时延用于后续结果统计，然后向发送端反馈，并结束收包流程。

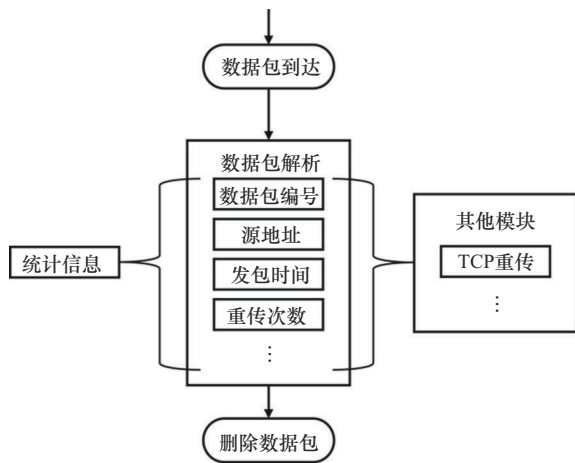


图8 接收端节点功能

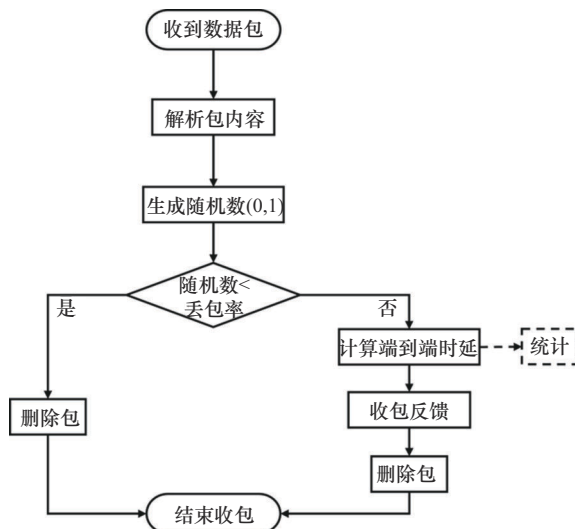


图9 数据包接收端的实现

2.3 TCP重传模块实现原理

用户数据报协议 (user datagram protocol, UDP) 是不可靠的，主要因为它缺乏一些保证数据可靠性的机制，这使得在传输过程中可能出现数据丢失、乱序或重复的情况。与UDP不同的是，TCP提供了一种建立在可靠连接基础之上的字节流量服务。

TCP重传模块需要确保发送端节点发出的数据包能够被接收端节点接收，如果接收端节点未接收到数据包，需要发送节点重传此数据包，直

到接收节点正确收到此数据包。针对数据包的重传需求，此模块需要具备数据包接收判断、重传计时、数据包重传、重传计数等功能，同时为了直观地观察重传情况，还需要相关重传信息输出功能。其中，数据包接收判断用于判断当前接收端是否已接收数据包，并向发送端反馈，确认数据包已接收并避免重复收包；重传计时用于记录数据包传输时间，若超过重传计时且未收到接收反馈，触发重传过程；数据包重传过程用于重新发送指定数据包；重传计数功能可以记录数据包的重传次数，用于传输质量的统计。

设计的TCP重传功能如图10所示。定义重传时间 T_{retran} ，TCP重传模块被分为发送端部分和接收端部分。其中，在发送端任意首次发送的数据包都存在一个重传计时器，根据设定时间检测接收端是否发送接收反馈信号，若收到反馈，则认为接收端已收到数据包，删除此数据包的重传计时器；若未收到反馈，则认为此数据包已经丢包，重传计时器达到 T_{retran} 时，再次发送重传数据包，并重置该数据包的重传计时器，将重传次数加1。在每次重传后，将重传情况输出，便于仿真结果统计。

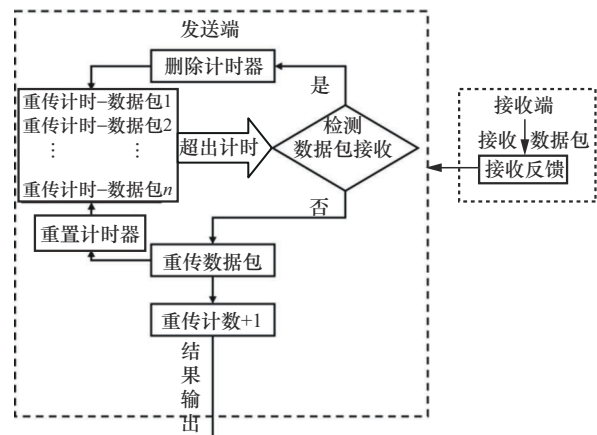


图10 TCP重传功能

TCP重传计时器如图11所示，每次进行发包操作时，首先遍历每个数据包的发包时间，若发现存在数据包发送后等待反馈的时间超过了规定

的重传时间，则跳出遍历循环，执行重传发包操作，并更新数据包发包时间记录；若未发现超时的数据包，则正常发送下一个数据包，并将该数据包的发包时间添加到发包时间记录队列中。

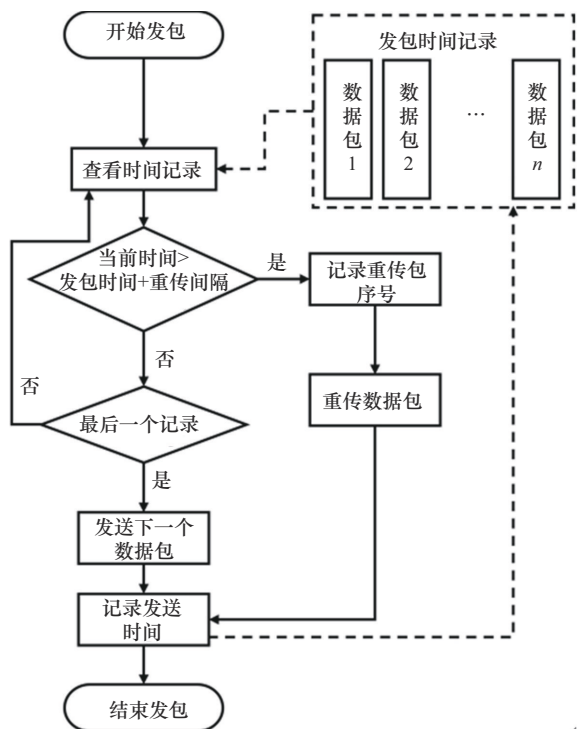


图 11 TCP 重传计时器

2.4 路由交换节点实现原理

TSN 交换机的上层结构主要包括交换机时钟、流调度模块、输出端口队列、时延模块以及下层接口模块，可以转发数据包至下层处理，并调度至相应输出队列进行传输。下层结构主要包括网卡、交换机时钟、中继模块和时延模块等，可完成对来自物理层（physical layer, PHY）和 MAC 层数据的相关处理。下层结构中网卡模块主要对从链路进入的数据包进行初步处理，并将数据包传入 TSN 交换机。

路由交换节点具有对数据包的处理和转发功能，根据数据包的目标 MAC 地址查找转发表，并将数据包转发到正确的物理端口，通常工作在数据链路层（OSI 模型的第 2 层）。路由交换节点会根据

网络拓扑和路由协议（如 OSPF、BGP、RIP 等）生成和更新路由表，并且可以处理多播和广播数据包，确保数据包正确地分发到所有目标节点。

设计的路由交换功能如图 12 所示。路由交换节点从一个端口接收到数据包（也称为帧），然后解析数据包的帧头，读取其中的目标 MAC 地址和源 MAC 地址，将源 MAC 地址和接收数据包的端口关联起来，更新其内部的 MAC 地址表并判断该地址对应端口是否在表中已经记录，这张表用来记录每个 MAC 地址与交换机端口之间的映射关系。如果目标 MAC 地址在表中有对应的端口记录，交换机会将数据包转发到特定端口。如果目标 MAC 地址不在交换机的转发表中，交换机会将数据包广播到除接收数据包端口外的所有端口。广播的目的是确保目标设备能够接收到数据包，并且其响应的帧将返回交换机，这样交换机就能学习到目标设备的 MAC 地址。如果数据包的目标地址是广播地址（FF:FF:FF:FF:FF:FF），交换机会将数据包发送到除接收端口之外的所有端口。交换机会将多播地址的数据包根据多播协议处理，通常会转发到与多播组相关的多个端口。

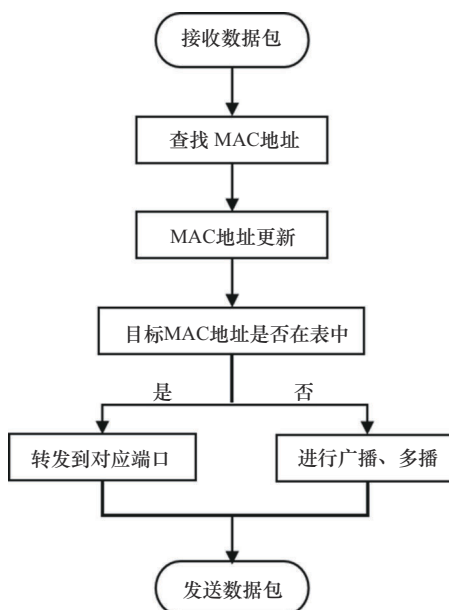


图 12 路由交换功能



TSN 交换机输出队列模块如图 13 所示，排队帧模块将数据包映射至不同优先级队列，交换机根据门控列表进行数据转发。当同一时间端口中有多个队列的流量可传输时，传输选择模块选择优先级更高的流量进行传输。此外，还可通过流调度模块对突发流进行整形，通过限制端口转发速率，有效防止交换机内部或下一跳节点发生拥塞，使突发流量以更均匀的速度平缓向外发送。

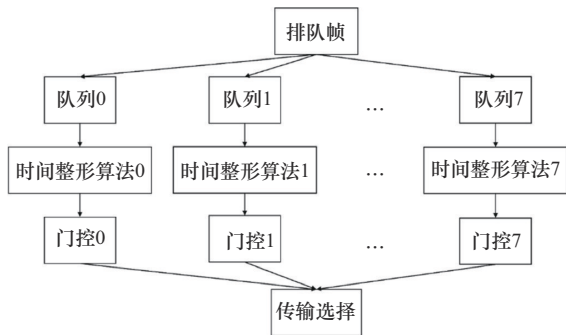


图 13 TSN 交换机输出队列模块

2.5 链路模块实现原理

为实现星间链路动态传播时延的模拟，提出的设计原理动态链路实现如图 14 所示。利用 STK 得到各个时刻的星间距离，并分别除以光速，作为各个时刻星间链的传播时延。在 OMNeT++ 中获取当前仿真时间，在设定时刻改变星间链路传播时延。

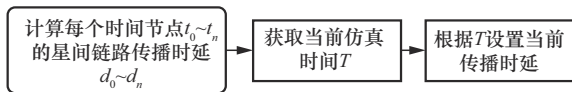


图 14 动态链路实现

3 平台功能验证及仿真分析

仿真平台使用 ubuntu 18.04，安装 OMNeT++

5.6.1 版本，使用 inet 4.1.2 和 nesting 框架。

3.1 平台功能验证

(1) 星间链路传播时延模拟功能验证

以 Walker 星座为例，相同轨道高度相邻两颗

卫星的星间链路空间几何关系如图 15 所示。任意两相邻轨道面的夹角和西侧轨道面升交点与两轨道间北侧交点在地心的夹角^[19]分别为：

$$\gamma = \arccos \left(1 - \frac{(\sin \kappa \cdot \sin \frac{2\pi}{P})^2}{1 + \cos \frac{2\pi}{P}} \right) \quad (1)$$

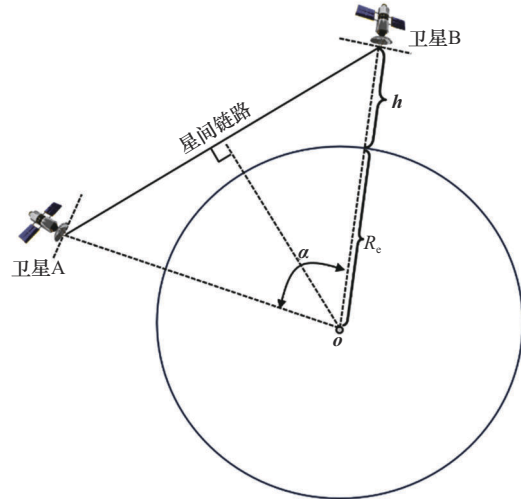


图 15 相同轨道高度相邻两颗卫星的星间链路空间几何关系

$$i = \arcsin \left(\sqrt{\frac{(1 + \cos \frac{2\pi}{P})^2}{2 + 2\cos \frac{2\pi}{P} - (\sin \kappa \cdot \sin \frac{2\pi}{P})^2}} \right) \quad (2)$$

其中， κ 为轨道倾角， P 为轨道面数目，异轨链路地心角可表示为^[19]：

$$\alpha = \arccos \left(-\frac{1 - \cos \gamma}{2} \cos (2\omega t + \Delta\varphi) - \frac{1 + \cos \gamma}{2} \cos (2i - \Delta\varphi) \right) \quad (3)$$

其中， $\Delta\varphi$ 为邻轨两颗卫星间的邻轨相位差， ω 为卫星角速度，分别表示为：

$$\Delta\varphi = \frac{2\pi F}{P \cdot S}, \quad (4)$$

$$\omega = \sqrt{\frac{GM}{(h + R_e)^3}} \quad (5)$$

其中， F 为相位因子， S 为每个轨道上卫星数量， G 为万有引力常数， M 为地球质量， h 为轨道高

度, R_e 为地球半径。卫星 A 到卫星 B 的距离可表示为:

$$d_{A-B} = 2(h + R_e) \sin\left(\frac{\alpha}{2}\right) \quad (6)$$

星间链路传播时延可表示为:

$$d_{prop} = \frac{d_{A-B}}{c} \quad (7)$$

其中, c 为光速。Walker 星座参数见表 2。

表 2 Walker 星座参数

参数	参数值	参数	参数值
P	72	S	22
R_e	6 371 km	h	550 km
F	2	κ	53°
G	$6.67 \times 10^{-11} \text{ N}\cdot\text{m}^2/\text{kg}^2$	M	$5.965 \times 10^{24} \text{ kg}$

链路传播时延动态变化时端到端时延统计情况如图 16 所示, 可以看到其真实曲线动态变化符合理论星间传播时延动态变化。

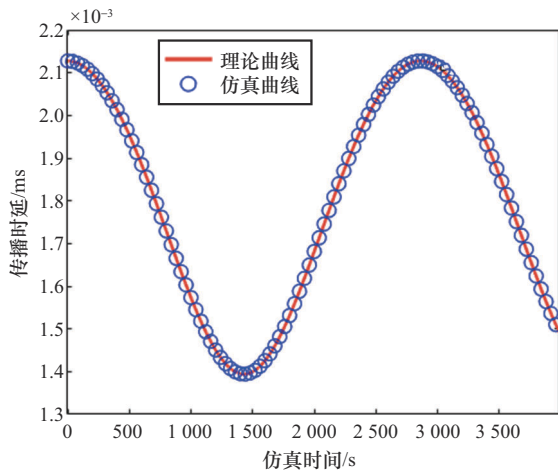


图 16 链路传播时延动态变化时端到端时延统计情况

(2) 数据包重传机制功能验证

通过 OMNeT++ 生成一个简单的卫星网络拓扑。TCP 重传机制验证网络拓扑如图 17 所示, workstation1 和 workstation2 是非实时性通信发送机, 分别发送 PCP 为 6 和 5 的数据包, backupServer 是非实时性通信接收机, 用于接收 PCP 为 5 和 6 的数据包。robotController 是实时性通信

发送机, 发送 PCP 为 7 的数据包, roboticArm 是实时性通信接收机, 用于接收 PCP 为 7 的数据包。satellite1 和 satellite2 为具有路由交换功能的卫星节点, 用于交换和转发数据包。

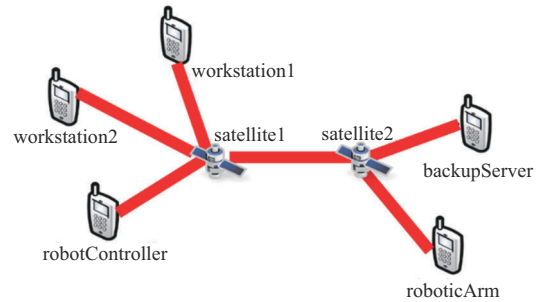


图 17 TCP 重传机制验证网络拓扑

基于图 17 简易网络拓扑, 设置仿真持续时间为 1 s, workstation1 和 workstation2 向 backupServer 发送 BE 流量, robotController 向 roboticArm 发送 TT 流量, robotController 具有重传功能, 重传时间 $T_{retran} = 3.1 \text{ ms}$, 误包率为 5%。星间链路传播时延设为 1 ms, 链路带宽设为 1 Gbit/s。数据包重传机制正确性验证如图 18 所示, 以门控列表算法为例, 未发生重传的数据包与发生重传的数据包时延相差 3.1 ms。综上证明所提重传功能可以正常运行。

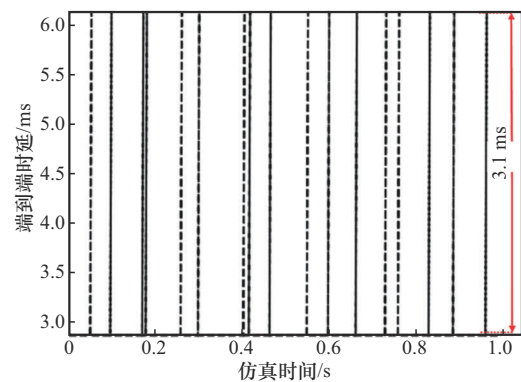


图 18 数据包重传机制正确性验证

3.2 仿真参数

基于如图 19 所示的卫星网络拓扑进行仿真, 路由路径用粗线标注, 其余仿真发包参数见表 3。

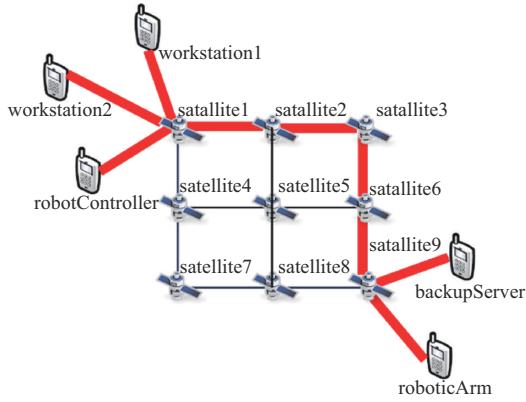


图19 卫星网络拓扑

表3 发包参数

源	目的	优先级	发包间隔	包大小/byte
robotController	roboticArm	7	400 μs	1 500
workstation1	backupServer	6	非周期业务	1 500
workstation2	backupServer	5	非周期业务	1 500

统计每一帧的端到端时延，在不排队的前提下，高优先级数据包每一帧的端到端时延（从robotController到roboticArm）理论值为：

$$d_{tot} = \sum_{i=1}^6 (d_{i,prop} + d_{i,trans}) + 5 \cdot d_{proc} \quad (8)$$

其中， $d_{proc} = 5 \mu s$ 为卫星节点处理时延， $d_{i,prop}$ 为数据包经过第*i*条链路的链路传播时延，利用式(7)进行计算，链路传输时延 $d_{i,trans}$ 定义为数据帧大小与链路传输速率的比值，计算得：

$$d_{i,trans} = \frac{1\,500 \times 8 \text{ bit}}{1 \text{ Gbit/s}} = 12 \mu s \quad (9)$$

为了更直观地体现各方案的平均时延抖动情况，定义时延抖动的计算式如下所示。

$$jitter = d_{tot,max} - d_{tot,min} \quad (10)$$

其中， $d_{tot,max}$ 为接收端接收到所有数据包的最大端到端传输时延， $d_{tot,min}$ 为最小端到端传输时延。将网络吞吐量定义为网络在单位时间内成功传输的数据量，可表示为：

$$Th = \sum \frac{D_i}{T} \quad (11)$$

其中， $I=9$ 为链路个数， $T=1 s$ 为仿真时间， D_i 为时间*T*内通过第*i*条链路的数据量。平均链路利用率定义为网络吞吐量与总信道带宽的比值，可表示为：

$$U = \frac{Th}{I \cdot BW} \quad (12)$$

其中， BW 为总信道带宽。

3.3 仿真结果及分析

在业务需求和资源供给不平衡的情况下，为实现资源的合理分配和充分利用，需要按照某种规则从队列中选择待转发的数据包，这种方法称为队列调度。本文对比了4种常见队列调度方案在无重传时的性能，分别为时间感知整形（time aware shaping, TAS）、帧抢占（frame preemption, FP）、先进先出（first in first out, FIFO）和加权轮询（weighted round robin, WRR）。

TAS方案通过门控列表（gate control list, GCL）实现对各队列开关状态的控制，门开可以传输数据包，门关不可传输数据包。当有多个门打开时，将按优先级从高到低依次进行数据包传输；而FP方案中高优先级数据包能够打断正在传输的低优先级数据包。FIFO方案不对数据包进行分类，采用“尽力而为”的方式，出队顺序与入队顺序相同。WRR方案为每个队列赋予一定权值，之后按优先级依次轮询每个队列并从队列中取出对应大小的数据包进行转发。

首先对TT流量的端到端平均时延与时延抖动进行仿真，TT流量平均时延及时延抖动性能对比如图20所示。TAS方案可根据流量的到达情况进行精细化的门控设计，假设数据包在*T*时刻到达卫星节点，设置门控打开时间段为 $[T, T+20] \mu s$ ，交换机专门将每个周期中的 $20 \mu s$ 预留给robotController发送TT流量。由式(9)可知，数据包传输时延为 $12 \mu s$ ，因此在理论上，数据包在 $20 \mu s$ 内会通过此节点，避免了TT流量与BE流量在输出端口的抢占情况，使平均端到端时延和时延抖动较小。

当采用FP机制时，若TT流量到达卫星节点

时 BE 流量正在传输，将触发帧抢占机制，从而中断正在传输的 BE 流量。当 TT 流量传输完成后，再使 BE 流量继续传输，从而减少 TT 流量的排队时延。因此在该方案下，TT 流量的端到端时延有小幅波动，其确定性略差于 TAS 模式。

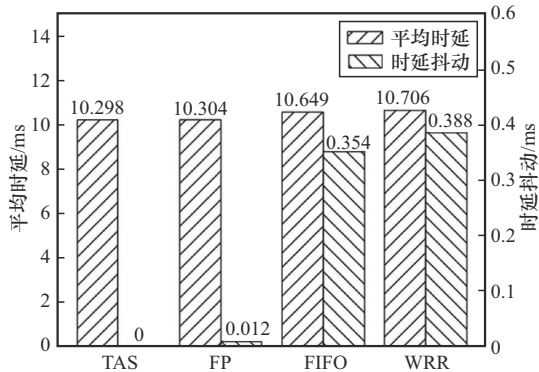


图 20 TT 流平均时延及时延抖动性能对比

FIFO 方案中，由于未设置优先级，3 条流量在一条链路上相互争夺带宽资源，TT 流量更容易在输出端口被先入队的多条 BE 流量阻挡，导致排队时延较高，时延抖动较大。WRR 方案会对各队列进行轮询，依次转发对应数量的数据包，因此 TT 流量通常需要等待其他队列中数据包转发完成后才能转发，进而导致其面临较高的端到端时延与时延抖动。

进一步对网络吞吐量与平均链路利用率进行仿真分析，网络整体吞吐量和平均链路利用率性能对比如图 21 所示。TAS 方案中的 GCL 控制以牺牲链路利用率为代价降低 TT 流量的时延抖动，因此 TAS 方案下整体网络吞吐量和链路利用率较低。与其他方案相比，TAS 方案仅使网络吞吐量减小不到 5%，链路利用率降低不到 3%，但其可以保障 TT 流量具有低时延抖动，因此认为这种性能损失是可以接受的。FP 和 FIFO 方案由于未设置 GCL，链路的传输资源被充分利用，从而网络吞吐量和链路利用率较高。WRR 方案在进行队列调度时，容易出现轮询到某个队列时，该队列中尚无待发送数据包的情况，导致该轮调度资

源浪费，无法有效利用链路带宽，从而造成整体网络吞吐量和链路利用率最差。

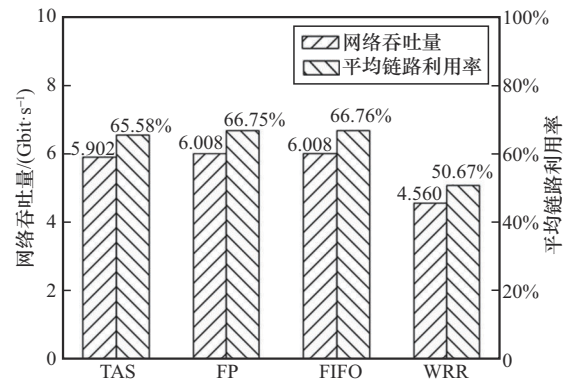


图 21 网络整体吞吐量和平均链路利用率性能对比

不同方案下丢包率和到达率性能对比如图 22 所示。

在图 22 (a) 中，TAS 方案会通过门控设置来降低 TT 流量排队时延，因此会降低 BE 流量的可用带宽，在丢包率和数据包到达率的性能上会差于 FP 和 FIFO 方案，但差距不大。类似地，WRR 方案会将传输资源分配给不同优先级队列，但由于该方案存在轮循到某一队列该队列中数据包还未到达的情况，会浪费一定传输资源，从而使其网络丢包率与到达率为所有方案中最差的。

接下来，在图 22 (b) 中可以发现，TAS、FP 以及 WRR 都能为 TT 流量提供较为充分的传输带宽保障，因此这 3 个方案中 TT 流量到达率都接近 99%。然而，FIFO 方案不区分 TT 流量和 BE 流量，因此该方案无法为 TT 流量提供传输资源保障，当采用该方案时，TT 流量到达率仅为 21.52%。另一方面，也是由于 TAS、FP 以及 WRR 方案能为 TT 流量提供传输带宽保障，这 3 个方案会牺牲 BE 流的传输机会，从而在 BE 流量到达率方面都不如 FIFO。TAS 方案在 BE 流量到达率上仅比 FIFO 减小 5% 左右，但其能提供接近 100% 的 TT 流量到达率以及低时延抖动，因此认为 TAS 方案在 BE 流量到达率方面的性能损失是可以接受的。

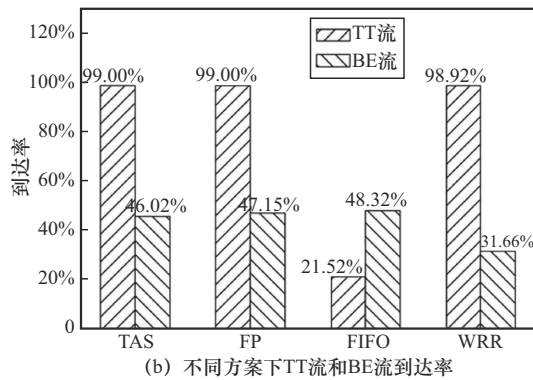
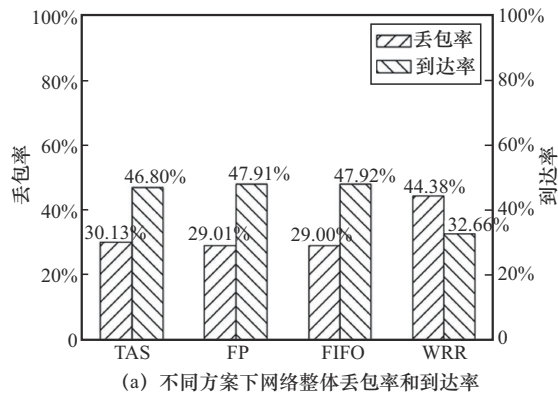


图22 不同方案下丢包率和到达率性能对比

4 结束语

为评估TSN技术在动态卫星网络中的性能水平,本文利用OMNeT++构建了低轨卫星时间确定性网络数字仿真平台,实现了收发端节点、路由交换节点、链路模块等基础功能,并进一步开发了TCP重传、优先级队列、动态星间链路模拟、队列调度与流调度等功能。并通过搭建低轨卫星网络进行仿真验证,通过对比TAS、FP、FIFO以及WRR 4种方案下数据包时延抖动、网络吞吐量、平均链路利用率、丢包率和到达率等指标,验证不同方案的性能水平。仿真结果表明,TAS方案以牺牲链路利用率为代价保证TT流量时延抖动确定性,缺点为会使BE流量丢包率有一定增加。对于严格要求端到端时延的时间敏感业务来说,TAS方案即最优方案,可以带来低时延抖动的确定性保障能力;若业务对端到端时延抖动有一定容忍能力,且同时考虑网络整体吞吐量、链路利用率及BE流量丢包率

等性能指标后,FP方案即最优方案。

未来工作将围绕智能化路径规划、星地协同通信、真实物理层模型构建及TSN标准协议集成等方向展开。随着天地融合网络的发展趋势,考虑网络拓扑变化频繁的特点,未来可引入基于强化学习、图神经网络等人工智能技术的动态路径选择机制,实现更加智能、自适应的多路径调度策略,从而提升动态路由的精度与鲁棒性。进一步完善OMNeT++仿真平台的可扩展性,扩展星地协同通信、跨域数据调度与切换等功能,引入更加真实的星间链路物理层建模,使仿真结果更贴近真实网络部署,提升方案评估的可信度。深入对IEEE 802.1 TSN系列协议的实现机制,拓展包括时间同步、调度机制等标准模块的实现与测试验证,促进研究成果与工业标准的融合对接。进一步提升卫星网络对时间敏感业务的支持能力和资源利用效率,为天地一体融合网络的构建与优化提供有力支撑。

参考文献:

- [1] LAI Z Q, WU Q, LI H W, et al. OrbitCast: exploiting mega-constellations for low-latency earth observation[C]//Proceedings of the 2021 IEEE 29th International Conference on Network Protocols (ICNP). Piscataway: IEEE Press, 2021: 1-12.
- [2] XIAO Z Y, YANG J Y, MAO T Q, et al. LEO satellite access network (LEO-SAN) toward 6G: challenges and approaches[J]. IEEE Wireless Communications, 2024, 31(2): 89-96.
- [3] FENG X A, SUN Y H, PENG M G. Distributed satellite-terrestrial cooperative routing strategy based on minimum hop-count analysis in mega LEO satellite constellation[J]. IEEE Transactions on Mobile Computing, 2024, 23(11): 10678-10693.
- [4] 孙耀华, 彭木根, 赵亚飞, 等. 低轨卫星互联网: 从星地融合迈向通遥一体化[J]. 北京邮电大学学报, 2024, 47(6): 69-98.
SUN Y H, PENG M G, ZHAO Y F, et al. Low earth orbit satellite network: from satellite-terrestrial convergence to the integration of communication, navigation and sensing[J]. Journal of Beijing University of Posts and Telecommunications, 2024, 47(6): 69-98.
- [5] 孙耀华, 江沫泽涵, 彭木根. 人工智能驱动的手机直连低轨卫星通信[J]. 无线电通信技术, 2024, 50(6): 1075-1086.
SUN Y H, JIANG S Z H, PENG M G. Artificial intelligence driven direct-to-handset satellite communication[J]. Radio Communications Technology, 2024, 50(6): 1075-1086.

- [6] 孙耀华, 冯昕澳, 彭木根. 低轨巨型星座组网: 挑战与关键技术[J]. 天地一体化信息网络, 2024, 5(4): 57-74.
SUN Y H, FENG X A, PENG M G. Mega LEO satellite constellations networking: challenges and key technologies[J]. Space-Integrated-Ground Information Networks, 2024, 5(4): 57-74.
- [7] 景毅, 姜春晓, 詹亚锋. 面向卫星通信的6G通感算融合架构、技术与挑战[J]. 无线电通信技术, 2023, 49(1): 12-20.
JING Y, JIANG C X, ZHAN Y F. 6G communication, sensing and computing integration for satellite communication: architectures, technologies and challenges[J]. Radio Communications Technology, 2023, 49(1): 12-20.
- [8] Alliance of Industrial Internet. A survey on industrial development of time-sensitive networking[R]. 2019.
- [9] Alliance of Industrial Internet. A industrial whitepaper on time-sensitive networking[R]. 2020.
- [10] 申雷翼. 基于半实物仿真系统的时间敏感网络调度算法研究[D]. 南京: 南京邮电大学, 2023.
SHEN L Y. Research on time-sensitive network scheduling algorithm based on hybrid simulation system[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2023.
- [11] 曹晖. 低轨卫星通信移动性管理研究及仿真平台设计[D]. 重庆: 重庆邮电大学, 2021.
CAO H. Research on mobility management of low earth orbit satellite communication and design of simulation platform[D]. Chongqing: Chongqing University of Posts and Telecommunications, 2021.
- [12] 陈金玲. 基于NS3的低轨卫星拥塞控制路由仿真平台设计与实现[D]. 北京: 北京邮电大学, 2022.
CHEN J L. Design and implementation of low earth orbit satellite congestion control routing simulation platform based on NS3[D]. Beijing: Beijing University of Posts and Telecommunications, 2022.
- [13] LI X Q, WANG L, LIU L X, et al. OMNeT and Mixim-based protocol simulator for satellite network[C]//Proceedings of the 2011 Aerospace Conference. Piscataway: IEEE Press, 2011: 1-9.
- [14] MARTINO L, DEUTSCHMANN J, HIELSCHER K S, et al. TCP CUBIC with HyStart: congestion control for satellite communication in OMNeT INET[C]//Proceedings of the 2024 IEEE 29th Asia Pacific Conference on Communications (APCC). Piscataway: IEEE Press, 2024: 19-25.
- [15] QOSJA M, MECKEL S, OBERMAISSER R. Discrete-event co-simulation interface for time-triggered organic computing[C]//Proceedings of the 2024 IEEE 12th International Conference on Intelligent Systems (IS). Piscataway: IEEE Press, 2024: 1-7.
- [16] HEISE P, GEYER F, OBERMAISSER R. TSimNet: an industrial time sensitive networking simulation framework based on OMNeT[C]//Proceedings of the 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Piscataway: IEEE Press, 2016: 1-5.
- [17] PAHLEVAN M, OBERMAISSER R. Evaluation of time-triggered traffic in time-sensitive networks using the OPNET simulation framework[C]//Proceedings of the 2018 26th Euro-micro International Conference on Parallel, Distributed and Network-based Processing (PDP). Piscataway: IEEE Press, 2018: 283-287.
- [18] KRUMMACKER D, FISCHER C, ALAM K, et al. Intra-network clock synchronization for wireless networks: from state of the art systems to an improved solution[C]//Proceedings of the 2020 2nd International Conference on Computer Communication and the Internet (ICCCI). Piscataway: IEEE Press, 2020: 36-44.
- [19] 王占伟. 低轨卫星星座的网络拓扑构型设计[D]. 西安: 西安电子科技大学, 2021.
WANG Z W. Network topological configuration design of low-orbit satellite constellation[D]. Xi'an: Xidian University, 2021.

[作者简介]



赵崇尧 (2001-), 男, 北京邮电大学网络与交换技术全国重点实验室硕士生, 主要研究方向为星地融合无线网络。



孙耀华 (1992-), 男, 博士, 北京邮电大学网络与交换技术全国重点实验室副教授, 主要研究方向为低轨卫星通信和无线接入网络智能化。



冯昕澳 (1999-), 男, 北京邮电大学网络与交换技术全国重点实验室博士生, 主要研究方向为低轨卫星通信和巨型低轨星座组网。



彭木根 (1978-), 男, 博士, 北京邮电大学网络与交换技术全国重点实验室教授, 主要研究方向为空间信息通信、通感算一体化、雾无线接入网络等。