

{StarryNet} empowering researchers to evaluate futuristic integrated space and terrestrial networks

一、研究动机与问题定义

1、核心背景

随着 SpaceX(Starlink)、亚马逊(Kuiper)等企业加速部署低地球轨道(LEO)卫星巨型星座，天地一体化网络(ISTN)正成为实现全球泛在接入、低延迟宽带服务的核心技术方向。ISTN 通过整合 LEO 卫星网络与现有地面互联网，可解决偏远地区网络覆盖空白、长距离通信延迟高等问题，同时支撑太空大数据(如地球观测影像)的高效传输与处理。但 LEO 卫星的独特特性也给 ISTN 研究带来了严峻挑战，亟需可靠的实验网络环境(ENE)支撑技术探索与验证。

2、ISTN 的核心特性与研究痛点

ISTN 与传统地面网络的本质差异体现在两方面：

- 高动态性：**LEO 卫星以约 7.8km/s 的速度绕地飞行，单颗卫星对地面某一固定点的可见时间仅数分钟，导致星地连接频繁中断、网络拓扑持续变化；
- 资源受限性：**太空环境中，卫星的带宽、算力、能源等资源稀缺且成本高昂，传统地面网络的资源密集型技术难以直接应用。

这些特性催生了一系列亟待研究的关键问题：如何设计适应高动态性的星地互联拓扑？如何将数千颗卫星融入现有互联网路由体系？如何保证复杂太空环境下的网络韧性？而这些问题的解决，都依赖于一个能精准模拟 ISTN 真实特性的实验环境。

3、现有实验方案的致命缺陷

Category / Tools		(i) Constellation Consistency	(ii) System and Networking Stack Realism	(iii) Flexible and Scalable Environment	(iv) Low-cost and Easy-to-use
Live LSNs or platforms	Live Starlink ([34])	✓	✓	✗	✗
	PlanetLab ([20])	✗	✓	✗	limited
	Emulab ([7])	✗	✓	✗	limited
Simulators and orbit analysis tools	STK ([35])	✓	✗	✓	✗
	GMAT ([11])	✓	✗	✓	✓
	SNS3 ([76])	for GEO only		✓	✓
	Hypatia ([60])	✓	✗	✓	✓
	StarPerf ([61])	✓	✗	✓	✓
Emulators and variations	Mininet ([55,68])	✗	✓	✓	✓
	DieCast ([54])	✗	✓	✓	✓
	Etalon ([69])	✗	✓	✓	✓
STARRYNET (this paper)		✓	✓	✓	✓

Table 1: Comparison of popular network experimentation platforms across different ENE requirements for ISTNs.

现有三类实验方法均无法满足 ISTN 研究的核心需求，形成“真实性-灵活性”的两难困境：

- 真实卫星网络 (Live LSNs)：**以 Starlink 为代表，能提供最真实的网络环境，但存在三大局限——成本极高(单颗卫星造价超百万元)、灵活性极差(无法修改星座参数进行“假设性”实验)、可访问性低(普通研究团队难以获取控制权)；
- 网络模拟器 (Simulators)：**包括 STK、Hypatia、StarPerf 等，通过离散事件建模模拟网络行为，灵活性强且成本低，但抽象度过高——无法运行真实的系统代码和应用程序，忽略了 packet 处理开销、软件错误等实际系统效应，导致实验结果与真实场景偏差较大；
- 网络仿真器 (Emulators)：**如 Mininet、Eetalon 等，基于虚拟化技术构建虚拟网络，可加载真实操作系统和应用程序，但存在两大短板——无法精准模拟 LEO 卫星的高动态拓扑变化，且支撑规模有限(现有方案最多

仅能模拟数百个节点，远不及 Starlink 等星座数千颗卫星的规模)。

3、关键挑战提炼

构建 ISTN 实验环境的核心矛盾是同时实现“高真实性”与“高灵活性”，具体面临三大技术挑战：

- **特性模拟挑战：**地面设备缺乏模拟卫星高动态运动、空间传播特性的能力；
- **规模支撑挑战：**巨型星座的节点数量（数千颗）远超传统实验方案的处理能力；
- **开销控制挑战：**模拟大规模卫星的频繁状态更新（如位置、连接关系）会产生巨大的计算和网络开销，导致实验环境不可用。

二、系统设计解析

StarryNet 的核心设计理念是“真实数据驱动的数字孪生”，通过整合众包数据、模型分析与分布式仿真，在地面虚拟环境中构建与真实 ISTN 行为一致的实验环境。其系统架构如图 1 所示，包含四大核心组件，各组件协同实现“星座一致性、系统真实性、灵活性、低成本”四大目标。

1. 星座观测器 (Constellation Observer, § 4.2)

- **核心功能：**作为 StarryNet 的数据来源核心，通过众包和爬虫技术，全面收集公开的星座相关数据，为实验环境提供真实的数据支撑；
- **数据收集范围：**
 - regulatory 信息：卫星频率分配、轨道协调等官方备案数据（来自 FCC 等机构）；
 - 轨道与卫星数据：卫星轨道参数（TLE 数据）、运行状态、星座布局等（来自 Celestrak、UCS 卫星数据库）；
 - 地面设施数据：全球地面站的地理位置、天线参数、覆盖范围等（来自 SatNOGS、AWS Ground Station）；
 - 用户测量数据：终端用户实测的网络延迟、吞吐量、丢包率等性能指标（来自 Speedcheck、学术机构公开报告）；
- **数据存储：**将收集到的异构数据分类存储于卫星、地面站、用户终端三大数据库，支持高效查询与调用。

2. 星座同步器 (Constellation Synchronizer, § 4.3)

作为 StarryNet“真实性”的核心保障，其目标是让虚拟实验环境与真实 ISTN 在空间、时间、功能上保持高度一致，关键依赖四类混合模型：

(1) 核心模型设计 (§ 4.3.1)

- **星座模型：**采用 Walker 符号 ($N/P/p$) 描述星座宏观结构 (N 为每轨道卫星数、 P 为轨道面数、 p 为轨道相位偏移)，同时通过轨道六要素（倾角、高度、升交点赤经等）精准建模单颗卫星的运动轨迹，结合 SkyField 天文计算库，可计算任意时刻卫星的经纬度、高度 (LLH) 信息，误差小于 100 米；
- **地面站模型：**包含三大关键参数——地理位置（经纬度、海拔）、天线配置（数量、增益、工作频率）、仰角阈值（决定星地通信的可见性，通常设为 $10^\circ - 15^\circ$ ）；
- **网络模型：**
 - 连接策略：支持两种星间连接模式 (+Grid 模式：每颗卫星连接同轨道 2 颗相邻卫星 + 相邻轨道 2 颗卫星；Motif 模式：每颗卫星连接所有可见卫星，形成重复拓扑图案），星地连接支持“最短

- 距离优先”“最长可见时间优先”等策略;
- 性能计算: 传播延迟基于星地/星间物理距离与光速 (3×10^8 m/s) 计算, 链路带宽可由用户根据真实卫星硬件参数(如 Starlink 的 Ka 波段带宽)配置;
- 计算模型: 模拟卫星处理器性能, 支持配置空间级处理器(如 BAE-RAD 系列, 110~466MHz)或商用处理器(如 Raspberry Pi、Jetson TX2), 通过 CPU 频率缩放和时间配额限制, 精准复现卫星的算力约束。

(2) 一致性保障机制

- 空间一致性: 虚拟实验环境的星座规模、卫星位置、星地/星间可见性, 均与真实星座一致, 例如模拟 Starlink S1 星座时, 将严格按照 72 个轨道面、每个轨道 22 颗卫星的配置构建, 且卫星位置实时匹配真实轨道数据;
- 时间一致性: 将时间划分为固定时隙(可配置为 1~3 秒), 在每个时隙内更新卫星位置、连接关系等动态状态, 确保虚拟环境的拓扑变化节奏与真实 ISTN 一致。

3. 星座编排器 (Constellation Orchestrator, § 4.4)

作为 StarryNet “灵活性”与“规模支撑”的核心, 负责调度多台物理机的资源, 构建大规模、可动态调整的虚拟实验环境, 解决单台设备资源不足的问题。

(1) 多机分布式仿真架构

- 架构划分: 分为 1 台资源管理器(Manager)和多台工作节点(Worker), 资源管理器负责全局资源调度、拓扑规划和状态同步, 工作节点负责运行虚拟卫星、地面站等节点实例;
- 节点虚拟化: 采用 Docker 容器技术, 每个容器模拟一颗卫星或一个地面站, 容器内运行完整的 Linux 操作系统和网络协议栈, 确保可加载真实应用程序;
- 链路虚拟化: 通过 Open vSwitch 构建虚拟网桥, 结合 tc 工具动态配置链路延迟、带宽、丢包率等参数, 精准模拟星地/星间链路特性。

(2) 关键技术突破

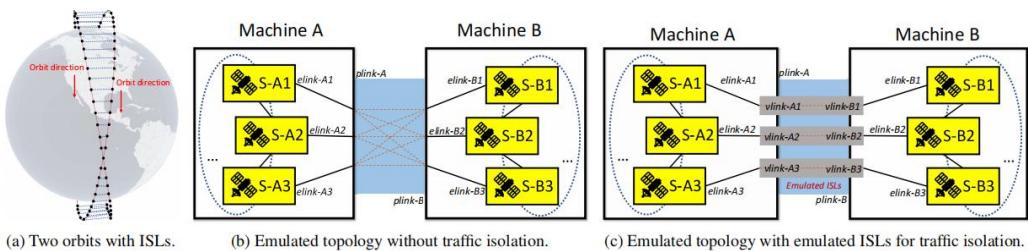


Figure 2: Multi-machine extension to support consistent topology emulation of satellite networks. Machine A and B interconnect by their physical interfaces plink-A/B (plink: physical link, elink: emulated link, vlink: virtual network link created by VLAN).

- 多机拓扑一致性保障: 传统多机仿真易出现“拓扑失真”——当卫星节点分布在不同物理机时, 直接桥接物理网卡会导致星间连接“全连通”, 与真实星座的网格拓扑不符。StarryNet 通过 VLAN 技术为每条星间链路创建独立虚拟链路(vlink), 实现流量隔离, 确保多机环境下的拓扑与真实星座完全一致;
- 高效状态更新机制: 针对卫星动态性带来的频繁状态更新开销, 采用“预测-预生成”策略——基于星座的会合周期(synodic period, 即星座回到相同地面投影的时间), 预生成每个卫星在整个周期内的所有状态更新

事件（如位置变化、连接建立/断开），每个容器通过独立线程读取本地事件列表并执行，避免集中式更新导致的瓶颈，同时通过 NTP 协议实现多机时间同步，确保状态更新时序一致；

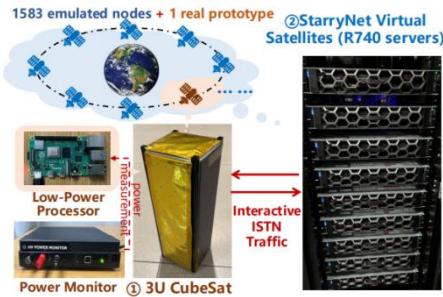


Figure 13: Hardware-in-the-loop testing with STARRYNET.

- **硬件在环支持:** 支持将真实卫星硬件原型（如 3U CubeSat）接入虚拟实验环境，实现“虚拟星座 + 真实硬件”的混合实验，可直接测量真实硬件在不同 workload 下的能耗、性能表现。

4. 统一抽象层 (Unified Abstraction, § 4.5)

为降低研究人员的使用门槛，提供两类简洁易用的 API，隐藏底层复杂的资源调度和模型计算逻辑：

- **环境 API (Env-APIs):** 支持用户加载星座配置文件（如 Starlink 的轨道参数 JSON 文件）、创建/启动/停止实验环境、配置时隙间隔和资源阈值 (Δ ，控制框架占用的 CPU/内存比例，确保预留足够资源给用户实验）；
- **自节点 API (Sat-APIs):** 允许用户程序在虚拟卫星上调用，获取卫星实时状态信息，包括卫星编号、轨道参数、当前位置、相邻可达卫星列表、阳光照射状态等，为开发星上定制化功能（如自适应路由协议）提供支撑。

三、关键技术创新

1. 真实数据驱动的数字孪生构建技术

- **创新点:** 突破传统仿真“模型驱动”的局限，将众包数据与物理模型深度融合，构建 ISTN 的数字孪生体——实验环境的星座规模、运动特性、链路性能均基于真实数据校准，同时保留模型的灵活性，支持用户自定义参数；
- **核心价值:** 首次实现“真实场景还原”与“假设性实验”的统一，既可以复现 Starlink 等真实星座的网络行为，也可以修改轨道高度、星座规模等参数，探索未来星座的优化方向。

2. 大规模星座的轻量化分布式仿真技术

- **创新点:** 通过“容器虚拟化 + VLAN 隔离 + 事件预生成”的组合方案，解决大规模、高动态星座的仿真难题——单台物理机可支持数百个卫星容器，多机集群可扩展至数千个节点，且状态更新开销控制在可接受范围内；
- **关键优化:**
 - 容器轻量化：优化 Docker 容器配置，每个卫星容器的内存占用仅约 20MB，远低于传统虚拟机（数百 MB）；
 - 链路动态调整：基于 tc 工具的链路管理模块优化，实现链路参数（延迟、带宽）的毫秒级调整，匹配卫星运动的高动态特性。

3. 兼具高保真与高灵活性的实验框架设计

- **创新点:** 通过“三层一致性保障”（星座一致性、系统栈一致性、能力一致性）确保高保真，同时通过 API 化设计和参数可配置化实现高灵活性；
- **具体表现:**
 - 一致性：支持运行未修改的 Linux 内核、路由协议（如 OSPF）和应用程序（如 iPerf），实验结果与真实 ISTN 的偏差小于 10%；

- 灵活性：可快速切换 Starlink、Kuiper、Telesat 等不同星座配置，支持星间链路开关、地面站分布调整等多样化实验场景。

4. 硬件在环的跨层实验支撑技术

- 创新点：打通虚拟仿真与真实硬件的交互通道，允许将卫星处理器、星载路由模块等真实硬件接入虚拟星座，直接测量低层级系统效应（如能耗、散热、电磁干扰）；
- 核心价值：解决卫星技术“上天前无法验证真实性能”的行业痛点，降低新技术部署的风险和成本。

四、实验设计与验证

StarryNet 的实验验证围绕“扩展性、保真度、实用性”三大核心目标展开，实验环境基于 8 台 DELL R740 服务器组成的集群（每台配备双 Intel Xeon 5222 处理器、256GB 内存、Ubuntu 20.04 系统）。

1. 扩展性验证

Constellation	Metrics	Height (km)	Constellation Size (number of satellites)	Creation Time (min) Nodes/Links/Total	Avg. CPU (%) Interval = 1/2/3 (s)			Avg. Memory (%) Interval = 1/2/3 (s)			Minimum # of Required Workers
Starlink S1 (72°22, 53°)		550	1584	5.9 [4.6, 10.5]	7.2%	7.0%	6.3%	3.9%	3.5%	3.4%	2
Starlink S2 (72°22, 53.2°)		540	1584	5.9 [4.6, 10.5]	7.2%	7.0%	6.3%	3.9%	3.5%	3.4%	2
Starlink S3 (36°20, 70°)		570	720	3.0 [2.1, 4.9]	1.2%	1.1%	1.0%	2.7%	2.6%	2.6%	1
Starlink S4 (6°58, 97.6°)		560	348	1.9 [1.3, 3.2]	1.0%	1.0%	1.0%	2.7%	2.6%	2.4%	1
Starlink S5 (4°43, 97.6°)		560	172	1.6 [1.2, 3.2]	1.0%	1.0%	1.0%	2.3%	2.3%	2.3%	1
Starlink Full (4408 satellites)		hybrid	4408	13.3 [7.9, 21.2]	39.6%	37.0%	34.3%	10.4%	9.1%	8.9%	7
Kuiper K1 (34°34, 51.9°)		630	1156	4.4 [3.8, 8.2]	2.6%	2.4%	2.3%	3.8%	3.5%	3.2%	2
Kuiper K2 (36°36, 42°)		610	1296	4.7 [4.2, 8.9]	3.9%	3.6%	3.2%	4.0%	3.6%	3.5%	2
Kuiper K3 (28°28, 33°)		590	784	3.2 [2.4, 5.6]	1.3%	1.2%	1.2%	2.7%	2.6%	2.6%	2
Kuiper Full (3236 satellites)		hybrid	3236	5.7 [4.8, 10.5]	24.6%	23.9%	23.2%	6.3%	6.2%	6.2%	6
Telesat T1 (27°13, 98.98°)		1015	351	1.9 [1.3, 3.2]	1.0%	1.0%	1.0%	2.6%	2.5%	2.4%	1
Telesat T2 (40°33, 50.88°)		1325	1320	4.8 [4.2, 9.0]	3.9%	3.7%	3.3%	4.0%	3.6%	3.5%	2
Telesat Full (1671 satellites)		hybrid	1671	3.1 [2.4, 5.5]	7.2%	7.0%	6.4%	4.2%	3.7%	3.6%	3

Table 2: Ability to support mega-constellation emulation with various experimental configurations and system overheads.

- 实验目标：验证 StarryNet 对不同规模星座的支撑能力及系统开销；
- 实验方案：模拟 Starlink、Kuiper、Telesat 三大主流星座的不同壳体（Shell）及全规模配置，测量实验环境的创建时间、CPU/内存开销；
- 关键结果：
 - 规模支撑：可灵活支撑 300–4408 颗卫星的仿真，覆盖从单壳体到全规模星座的需求；
 - 创建时间：小规模星座（300 颗卫星）约 1.6 分钟，全规模 Starlink 星座（4408 颗卫星）约 21.2 分钟；
 - 系统开销：CPU 占用率最高 39.6%，内存占用率最高 10.4%，预留足够资源给用户实验 ($\Delta=50\%$)；
 - 节点需求：全规模 Starlink 星座仅需 7 台工作节点，普通研究团队的云服务器集群即可满足。

2. 保真度验证

通过与真实卫星网络数据、主流模拟器结果对比，验证 StarryNet 的真实性：

(1) 与真实 Starlink 网络对比

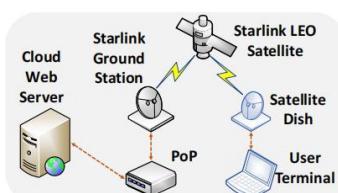


Figure 5: A live Starlink topology.

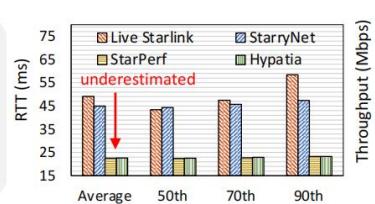


Figure 6: Latency comparison.

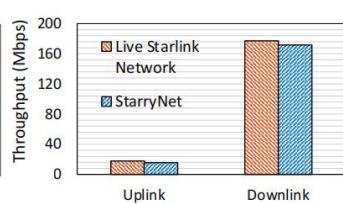


Figure 7: Throughput comparison.

- **实验方案:** 复现 2021 年欧洲 Starlink 实测场景 (用户终端位于克拉根福, 地面站位于法兰克福, Web 服务器位于维也纳), 测量 latency 和吞吐量, 并与真实实测数据、Hypatia、StarPerf 的模拟结果对比;
- **关键发现:**
 - latency: StarryNet 的平均 RTT 为 45ms, 与真实 Starlink 的 47ms 偏差仅 4.3%, 而 Hypatia、StarPerf 的平均 RTT 分别为 32ms、35ms, 低估了真实场景的系统开销;
 - 吞吐量: StarryNet 的上行吞吐量约 20Mbps、下行约 200Mbps, 与真实 Starlink 实测数据完全一致, 而模拟器因无法模拟真实传输协议开销, 无法得到准确结果。

(2) 星间链路 (ISL) 场景验证

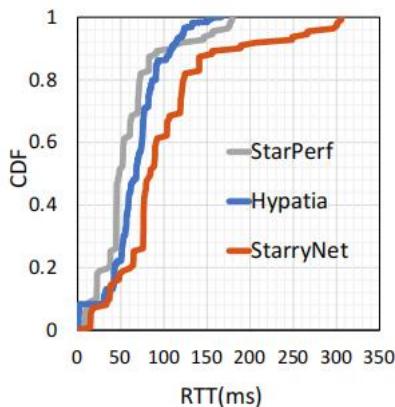


Figure 8: Latency comparison in an ISL-enabled topology.

- **实验方案:** 模拟 ISL 启用后的 Starlink 星座, 测量地面站对之间的 latency 分布, 并与 Hypatia、StarPerf 对比;
- **关键发现:** StarryNet 的 latency CDF 曲线与真实硬件测试结果 (Raspberry Pi 4B、Jetson TX2) 更贴近, 平均 latency 比模拟器高 8-12ms, 这是因为 StarryNet 计入了 packet 处理、链路切换等真实系统开销, 而模拟器忽略了这些因素。

(3) 计算能力模拟验证

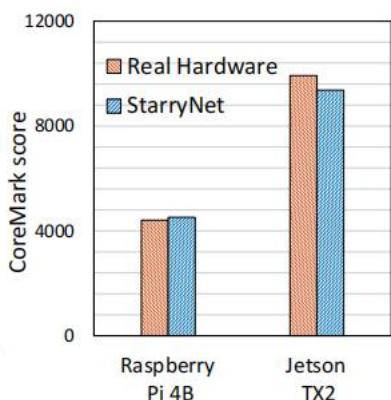


Figure 9: Flexible computation capability.

- **实验方案:** 通过 CoreMark 基准测试, 验证 StarryNet 对不同卫星处理器的模拟能力;
- **关键发现:** StarryNet 模拟的 Raspberry Pi 4B、Jetson TX2 的 CoreMark 得分, 与真实硬件的偏差小于 5%, 可精准复现不同算力约束下的应用性能。

3. 实用性验证: 三大典型案例研究

通过真实研究场景验证 StarryNet 的灵活性和实用性:

(1) 星地互联拓扑设计空间探索

- **实验目标:** 对比四种主流星地互联模式的性能与成本;
- **实验方案:** 基于 StarryNet 构建四种拓扑的实验环境——SRLA (星载中继-最后一公里接入)、SRGS (星载中继-地面站组网)、GSSN (地面站网关-卫星网络)、DASN (用户直接接入-卫星网络), 测量 latency、可达性、

成本等指标；

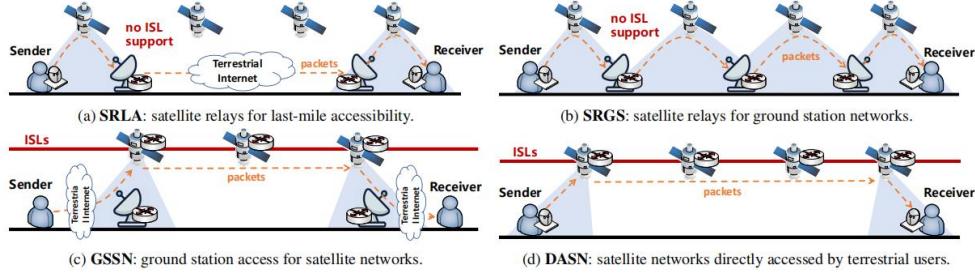


Figure 10: The design-space for various space-ground integration methodologies.

Design	Average end-to-end latency and its breakdown (ms)				Reachability	Frequent Address Update	Operating Cost		
	Inter-Satellite	Space-Ground	Ground	Total			GS	Terminal	ISLs
SRLA	0	76.25	107	183.25	97.00%	X	✓	✓	X
SRGS	0	313.39	0	313.39	51.00%	X	✓	✓	X
GSSN	48.46	38.45	20	106.91	57.40%	X	✓	X	✓
DASN	48.46	37.65	0	86.11	97.50%	✓	X	✓	✓

Table 3: Comparison for different space-ground integration methodologies.

- 关键结论：

- latency 最优：DASN 模式（平均 86.11ms），得益于 ISL 构建的空间直连路径；
- 可达性最优：DASN (97.5%) 和 SRLA (97%)，SRGS 因地面站覆盖不足仅 51%；
- 成本权衡：“弯管”模式（SRLA、SRGS）无需 ISL，部署成本低但 latency 高；ISL 依赖模式（GSSN、DASN）latency 低但部署成本高，且 DASN 存在地址频繁更新的问题。

(3) 网络韧性评估

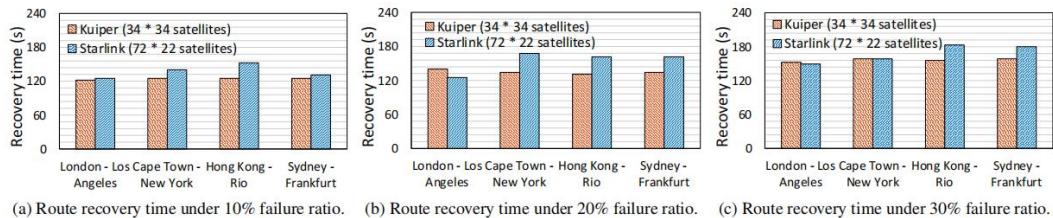


Figure 11: Route recovery time under different constellation-wide failures.

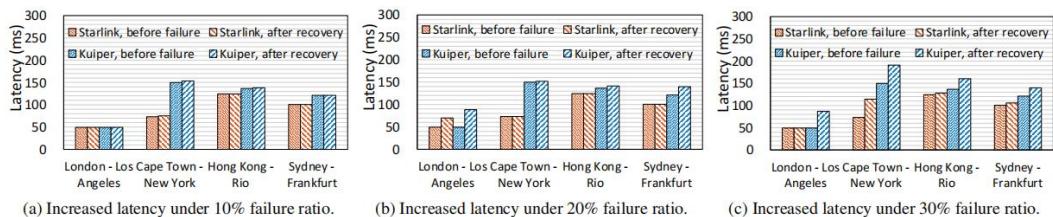


Figure 12: Increased latency after route reconvergence under various constellation-wide network failures.

- 实验目标：分析卫星故障对 ISTN 性能的影响，验证网络韧性；
- 实验方案：模拟 10%、20%、30% 的卫星故障（如 geomagnetic storm 导致的卫星失效），测量路由恢复时间和 latency 变化；

- 关键结论：

- 恢复时间：故障比例越高、星座规模越大，路由恢复时间越长（30% 故障时，Starlink 星座的恢复时间达 240s）；
- latency 变化：故障比例低于 20% 时，latency 仅轻微上升（约 10–15%），因星座的网状拓扑提供了路径冗余；故障比例达 30% 时，latency 急剧上升（约 50%），部分路径因缺乏备份路由中断。

(4) 硬件在环测试

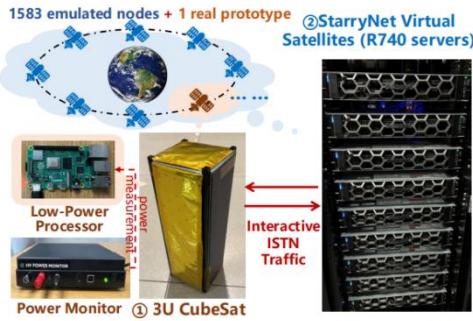


Figure 13: Hardware-in-the-loop testing with STARRYNET.

收敛过程，通过功率计测量能耗；

关键结论：能耗变化：Idle 状态功耗 2.83W，路由收敛时 3.22W，数据传输速率从 100Mbps 提升至 1000Mbps 时，功耗从 4.6W 增至 5.46W；

实用价值：可直接为卫星电源系统设计提供数据支撑，避免因能耗估算偏差导致的任务失败。

五、局限与未来方向

1、现存局限

数据依赖局限：StarryNet 的保真度高度依赖公开数据的准确性，部分关键数据（如 Starlink 的激光 ISL 真实带宽、卫星故障恢复机制）未公开，需用户手动配置，可能影响实验精度；此外，公开 TLE 轨道数据的误差可达 12km，会导致星地可见性、传播延迟的计算偏差；

物理层模拟局限：目前主要聚焦于网络层、传输层实验，对物理层特性（如频谱适配、信号衰减、电磁干扰）的模拟能力有限，无法支撑无线通信物理层相关研究；

规模上限局限：虽然已能支撑数千颗卫星的仿真，但随着星座规模向万颗级扩展（如 Starlink 未来计划部署 4.2 万颗卫星），现有多机调度机制的开销会显著增加，需进一步优化。

未来研究方向

数据与模型优化：持续跟踪真实星座的部署进展，更新数据库；引入机器学习算法，基于有限公开数据校准模型参数，降低数据缺失带来的精度损失；

功能扩展：增强物理层模拟能力，支持频谱资源分配、调制编码方案等实验；引入星上能源模型，模拟太阳能供电、电池储能等能源约束；

性能提升：优化多机资源调度算法，采用容器编排技术（如 Kubernetes）实现动态扩缩容；探索边缘计算与仿真结合的模式，降低大规模仿真的网络开销；

应用生态构建：开发针对 ISTN 研究的专用工具链（如路由协议测试套件、韧性评估工具），并与开源社区合作，扩大用户群体，收集反馈持续迭代。

六、总结与启示

StarryNet 通过“真实数据驱动 + 分布式轻量化仿真”的创新设计，首次突破了传统实验方案“真实性 - 灵活性”的两难困境，构建了一套同时满足“星座一致性、系统真实性、灵活性、低成本”的 ISTN 实验框架。其核心贡献在于：

技术层面：提出了数字孪生驱动的实验环境构建方法，通过四大核心组件的协同设计，解决了高动态、大规模、资源受限的 ISTN 模拟难题；

实践层面：提供了开源、易用的实验工具，降低了 ISTN 研究的门槛，普通

State	Idle	Routing convergence	Data transmission rate (Mbps)				
			100	250	500	750	1000
Power consumption (W)	2.83	3.22	4.6	4.99	5.36	5.45	5.46

Table 4: Power consumption under various ISTN workloads.

实验目标：测量星载路由协议在真实硬件上的能耗开销；

实验方案：将 3U CubeSat 原型（搭载低功耗处理器）接入 StarryNet 的虚拟星座，模拟不同数据传输速率下的路由

研究团队仅需少量云服务器即可开展大规模实验；

研究层面：通过三大案例研究，验证了框架的实用性，为星地互联拓扑设计、网络韧性优化、硬件能耗评估等关键研究提供了可靠支撑。

StarryNet 的成功也带来了重要启示：在复杂网络系统研究中，“数据驱动 + 模型校准”是平衡真实性和灵活性的有效路径；而开源化、易用化的工具链，能加速技术迭代与社区协作，推动整个领域的快速发展。未来，随着星座技术的持续演进和 StarryNet 的不断优化，其有望成为天地一体化网络研究的核心基础设施，助力更多创新技术从实验室走向实际部署。