Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Routing in Future Space-Terrestrial Integrated Networks with SatNet-OSPF

Hongcheng Yan*[1] | Liang Qiao[1] | Wei Wu[1] | Juan A. Fraire[2,3] | Dong Zhou[1] | Luming Li[1] | Yong Xu[1]

[1]Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing 100094, China
[2]Univ Lyon, Inria, INSA Lyon, CITI, F-69621 Villeurbanne, France
[3]CONICET - Universidad Nacional de Córdoba, Argentina

Correspondence
*Hongcheng Yan, Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing 100094, China. Email: hongcheng.yan@gmail.com

Summary

Connectivity in satellite networks is governed by the spacecraft nodes' orbital dynamics together with the planet's continuous rotation where ground nodes are located. The resulting time-dynamic but predictable topology demands the design of specific distributed routing schemes. However, terrestrial Internet routing schemes' maturity, proven scalability, and efficiency shall be leveraged whenever possible to facilitate space-terrestrial integration while reducing risk and costs. In line with this reflection, we introduce SATNET-OSPF: a backward-compatible satellite extension for the widely used Open Shortest Path First routing protocol. The key features of SATNET-OSPF are a) accurate routing interface mapping to inter-satellite links and ground-to-satellite links, b) accelerated link up/down event detection adapted to space-specific wireless technologies, c) proactive routing and forwarding mechanism to take advantage of predicted link down events, and d) low memory footprint topology model to efficiently propagate the forthcoming space connectivity events via constrained telecommand links. Leveraging existing IPv6 and OSPFv3 open-source stacks, we implemented SATNET-OSPF in an actual space router comprising a space-grade SPARC V8 CPU and a radiation-hardened FPGA. Furthermore, we present the details of an emulation test bench supporting various configurations with COTS terrestrial OSPF routers that enabled a realistic performance evaluation of the SATNET-OSPF. Results show that SATNET-OSPF reduced OSPFv3 routing protocol overhead by up to 31%; shortened the link event detection delay by four orders of magnitude; decreased the routing outage by a factor of 22; and ensured flooding control message generation and forwarding times, as well as routing computing time, satisfy the original requirements (192 ms, 37 ms, and 17 ms, respectively).

KEYWORDS:
Space and Satellite Networks, OSPFv3, Space Router

1 | INTRODUCTION

The space sector is thriving. More complex missions are being deployed for Earth observation, communications, and navigation purposes [1,2,3]. Increasing complexity is enabled by recent technological advances emerging from miniaturization in electronics and advanced software techniques [4]. Smaller satellites (e.g., nano-satellites or CubeSats) are becoming increasingly popular, unlocking a so-called democratic space [5]. However, to achieve more ambitious objectives regarding broader coverage, higher performance, and increased reliability, in-orbit resource sharing and data transfer via

inter-satellite links are on the hype [6]. These systems are interconnected with terrestrial networks in so-called space-terrestrial integrated networks, which have captured the attention of the space research and engineering communities [7,8].

The core challenge of space-terrestrial networks is the topology dynamics. Due to the relative motion of space and terrestrial nodes, satellite-to-satellite and ground-to-satellite links cannot be maintained persistently [9]. Instead, point-to-point communication links need to switch regularly, which results in a time-evolving space network topology. Furthermore, due to the harsh space environment, space links are prone to failures, which generates further fluctuations in the topology [10]. As a result, the space-terrestrial topology dynamics comprise two kinds of transformations: (*i*) predictable changes triggered by the line-of-sight variations between spacecraft and ground nodes, and (*ii*) non-predictable changes provoked by link and node failures.

Resulting end-to-end space-terrestrial multi-hop paths exhibit the same time-dynamic characteristic, which needs to be adequately captured and reacted upon by the routing protocol. While most routing algorithms have been developed to consider one link effect or the other, there are minimal alternatives for simultaneously dealing with the topology dynamics while ensuring a straightforward space-terrestrial networking integration.

Such an integrated, dynamic, and decentralized [11] routing technology is crucial to realize future networked space systems. The transport layer has already identified this from the U.S. National Defense Space Architecture [12]. The main challenge is identifying and implementing the algorithmics to autonomously predict and detect the topology changes while carefully adjusting the forwarding logic.

In this work, we tackle this ambitious objective by extending the most widely used routing algorithm on the Internet: Open Shortest Path First (OSPF)[1]. Without modifying the core OSPF logic, we design SATNET-OSPF: a specific framework that can provide the required space-terrestrial networking capabilities. The main features of SATNET-OSPF are a) adequate OSPF interface mapping to the space-terrestrial network, b) link up and down detection mechanisms, c) proactive forwarding table management based on forthcoming topology events, and d) a low footprint topology model to facilitate the configuration of SATNET-OSPF routers. The core value of the contribution is that OSPF remains unchanged: we can leverage existing and well-tested OSPF implementations with minimal risk and effort. We test this hypothesis with Quagga's [14] open-source OSPF code, framed in SATNET-OSPF and then embedded in a real space router hardware implemented with a space-grade CPU controlling a space-hardened FPGA. The resulting space router prototype is then integrated into a specific test bed we designed to mimic the space-terrestrial network dynamics. A series of test cases are designed to validate the features above, from which realistic measurements are summarized and analyzed. To sum up, the contributions of this paper are:

- Topology model: a space network topology model based on an adjacency list that achieves higher memory efficiency.
- SATNET-OSPF: an enhanced OSPFv3 wrapper that adapts it to route in highly dynamic space-terrestrial networks.
- Space router: We implement SATNET-OSPF on a flight-ready space router for which we introduce the architecture.
- Test bed evaluation: We evaluate the effectiveness of SATNET-OSPF through realistic tests with real hardware.

The remainder of this paper is organized as follows. Section II presents relevant related works. SATNET-OSPF is described in detail in Section III, including the topology model and critical mechanisms for interface selection, link detection, and link event prediction. Section IV presents the space router details and discusses how SATNET-OSPF was embedded in it. The space-terrestrial test bed and the test case results and analysis are summarized in Section V. Finally, conclusions and future work are laid out in Section VI.

2 | RELATED WORK

The state-of-the-art routing logic for space-terrestrial integrated networks can be classified into four gross research domains: a) IP-Based Routing in Space, b) space-specific routing, c) delay-tolerant networking routing, and d) software-defined networking routing.

2.1 | IP-Based Routing in Space

Terrestrial routing is widely based on Internet Protocol (IP) routing, which attracted the space networking research community as it presents a high technical maturity level due to its massive adoption. Furthermore, keeping IP as a standard protocol between the space and ground segment ensures a transparent space-terrestrial integration. Applying terrestrial IP routing to space networks is an attractive choice for the research community.

Extensive experiments have been carried out in orbit to test the validity of IP protocol in space. One is Cisco's Low Orbit Router (CLEO) [15]. The Internet Routing in Space (IRIS) [16] is also remarkable. The main limitation of these efforts is that IP-based routing assumes topological changes are

[1]This work is an extended and archival quality version of previous publication [13]. Besides comprehensive content, we add the new topology model and a more detailed explanation of SATNET-OSPF features

unpredictable. Thus, the studied protocols acted reactively toward link establishment and termination events. As discussed in [17], directly applying IP routing to space systems renders high protocol overhead and slow routing convergence.

To solve this issue, Bantan et al. proposed a space-based OSPF routing protocol [18]. This contribution exploits a so-called *sleep state* to the neighbor state machine of the core OSPF protocol to handle predictable link state changes. Nevertheless, this related work only provided a theoretical evaluation of the idea without giving quantitative evaluation metrics. On the other hand, Xu proposed an OSPF variant with a topology prediction feature [19]. Likewise, the solution extended the neighbor state machine to prepare it for future links. Finally, Fischer et al. [10] proposed a predictable link-state IP routing. However, it is based on a customized link-state routing. In all these three related works, authors modified the standardized OSPF state machine and did not indicate nor demonstrate if the update would break retro-compatibility with legacy OSPF versions. In other words, space-terrestrial integration was not considered.

2.2 | Space-Specific Routing

Routing in space-specific systems has kept busy a significant sector of academia and standardization committees led by national space agencies such as the Consultative Committee for Space Data Systems (CCSDS). Since the original Iridium Low-Earth Orbit (LEO) satellite system to the latest Starlink mega-constellation system, many routing protocols have been proposed and discussed [20].

Specifically, space-routing techniques are divided into single-shell and multi-shell constellation configurations. For the former, several routing solutions have been based on virtual topology snapshots [21,22]. In contrast, others followed the so-called virtual nodes paradigm [23], where satellites are temporarily associated with a virtual node in a fixed geographical position. On the other hand, routing protocols for multi-layer satellite networks generally exploit a group-based routing approach [24]. Finally, given the non-balanced traffic distribution of satellite networks, specific routing considering novel load-balancing techniques has also been proposed [25].

The main limitation of these space-specific routing techniques is that they remain predominantly abstract. In most cases, they are evaluated in simulation environments that do not consider the complete protocol stack, thus ignoring concrete technical issues that arise when integrating with the terrestrial network. Furthermore, it is worth noting that commercial satellite networks in LEO (e.g., Iridium, OneWeb, Starlink, etc.) do solve the integration issue. Still, given the proprietary nature of these deployments, the strategy is not public.

2.3 | Delay-Tolerant Networking Routing

Delay Tolerant Network (DTN) is a family of networks designed to cope with the high-latency and disruptions characterizing deep-space networks and other so-called challenged networks (e.g., underwater). When considered for the space domain, the main algorithm for routing in DTN is the Contact Graph Routing (CGR) [26,27]. CGR was initially proposed by NASA and then integrated into the Schedule Aware Bundle Routing (SABR) [28] standardized by the CCSDS.

The core capability of CGR is the consideration of future topology changes, which are imprinted in a *contact plan* [29]. The contact plan is processed into a contact graph over which modifications of the Dijkstra algorithm can be executed. CGR has been implemented in the open-source Interplanetary Overlay Network (ION) [30] and has been validated in several space missions [31].

Although CGR can handle predictable topological changes effectively, it does not consider unpredictable topological changes as required for the space-terrestrial network. Only recent research attempted to derive an opportunistic extension that has so far not performed as expected [32,33,34]. Moreover, integrating CGR with IP terrestrial networks is a challenge yet to be tackled.

2.4 | Software-Defined Networking Routing

Software Defined Networking (SDN) is an increasingly popular research topic where a centralized controller orchestrates routing. Thanks to the centralized and global vision of the overall network status, a global-optimum forwarding policy can be achieved [35]. Indeed, adaptations of SDN-based techniques to the space domain are available in the literature.

Among these, Bao et al. [36] proposed to place the SDN controller in a Geostationary Earth Orbit (GEO) satellite with reliable links and comprehensive coverage over LEO satellites. Also, Shi et al. [37] explored a distributed placement of the controller in three locations: GEO satellites, high-altitude platforms (HAPS), and ground nodes.

Even though the efforts in distributing SDN controllers, the centralized spirit of SDN routing proved to be poor in terms of system robustness compared with distributed routing [38]. Furthermore, centralized decision-making presents a slow response to network failures, which are reacted upon an SDN controller located in a remote network node.

2.5 | Toward Space-Terrestrial Integrated Routing

This paper argues that IP and DTN routing are at two extremes regarding predictable and unpredictable topological changes. In other words, the IP routing mechanism considers unpredictable topological changes, while DTN routing mainly considers predictable topological changes. This is likely the result of the originating scenarios of these two routing technologies: terrestrial Internet and deep-space.

The current state-of-the-art indicated two possible paths: a) extending DTN routing to cope with unpredictable events and b) extending IP routing to cope with predictable topology changes. Since IP is based on a more mature protocol stack, we focus on the latter with the spirit of not changing standardized logic but framing it into an adequate framework that bridges the terrestrial technology into the space segment.

Terrestrial IP routing protocols are mainly divided into two categories: *distance vector* routing and *link state* routing [39]. On the one hand, the link information in distance vector routing needs to be transmitted to the surroundings node by node, so it has the disadvantages of slow convergence and significant protocol overhead. On the other hand, the routers in link state routing autonomously detect the link state and flood the link state changes to the entire network. In both cases, all routers have a database describing the topology state of the entire network and use a unified algorithm to calculate the shortest path. However, link state routing delivers fast convergence and lower protocol overhead. Therefore, we choose to work with link-state routing.

The most widely-used terrestrial link-state routing protocols are Open Shortest Path First version 2 (OSPFv2) for IPv4 and Open Shortest Path First version 3 (OSPFv3) for IPv6 [40]. Due to the limited IPv4 address resources, the terrestrial network is transitioning from IPv4 to IPv6. At present, the applicability of IPv6 to space has been verified through mentioned CLEO [15] and IRIS [16] projects and supported by recent research [41,42]. As a result, we focus on OSPFv3. In the next section, we present SATNET-OSPF: our novel approach toward space-terrestrial routing leveraging OSPFv3.

3 | SATNET-OSPF

In this section, we describe SATNET-OSPF, which adopts four measures to enable and improve the performance of OSPFv3 in space networks. Firstly, we model the time dynamic topology and imprint it in a memory-efficient format. Secondly, we analyze the characteristics of space links and conclude that point-to-point interface types can reduce protocol overhead; Thirdly, as regards unpredictable link-down events, the decoding lock flag is introduced into the detection process of link state changes to speed up the detection process. Finally, to adequately capture predictable link-down events, we propose an efficient routing and forwarding control mechanism that sets the link down in advance to trigger the route convergence process while updating the network-wide forwarding table synchronously with a delay.

3.1 | Topology Model

Space networks generally store topological information as topology *snapshots* [10,22]. A new topology snapshot is generated when a new link is established, or an established connection is broken. The topology snapshot approach divides the time-varying space network topology into a series of static network topologies during a time interval in a Finite State Machine (FSM) fashion. However, the snapshot method divides the continuous intra-plane (contacts between neighboring satellites in the same orbit) and terrestrial links into discrete time intervals. As a result, the snapshot model requires a large amount of redundant information. If the model needs to be shared with satellite nodes, it will consume significant bandwidth in constrained telecommand, telemetry links, and valuable onboard memory.

We propose an alternative solution to approach space-terrestrial routing where topological information is exchanged among nodes efficiently. Our topology model is based on an *adjacency list*, as illustrated in Fig. 1. In this case, for each node s, each adjacent node u of node s represents a link denoted as $Link(s, u)$. Our model considers elements to describe the properties of the link: a) a flag indicating the periodic nature of the link, b) the period of the link, and c) the link establishment time interval list. The resulting topological model is strategically designed to consider the temporal dynamics inherent to satellite constellations. This includes both space-to-space and space-to-ground connections. This is important in the unique attributes of these links, which are characterized by continuous cycles of establishment and termination. This pattern ultimately leads to periods of episodic communication.

Specifically, the *isP* element indicates whether the $Link(s, u)$ is periodic. The specific period value is indicated in the *period* element if it is periodic. The *link establishment interval list* element is then used to indicate the time intervals during which $Link(s, u)$ is established. Let us denote the time interval as $TI_1 = \{t|T_0 \leq t < T_1\}$. Then, the link establishment interval is $\sum_i TI_i$. In case the contact is persistent (i.e., continuous visibility or cabled terrestrial link), the link establishment interval list shall contain a single interval, i.e., $TI = \{t|0 \leq t < \infty\}$. If the link is periodic, the interval list shall only store the information of one period. The information on subsequent periods can be calculated by iterating over the period value. Finally, the periods of different links can be maintained independently.

To wrap up, our space network topology model is based on an adjacency list, allowing a reduced amount of data to represent the time-dynamic aspects of the space-terrestrial topology. This is particularly appealing to fit the time-evolving and predictable changes of satellite constellations.

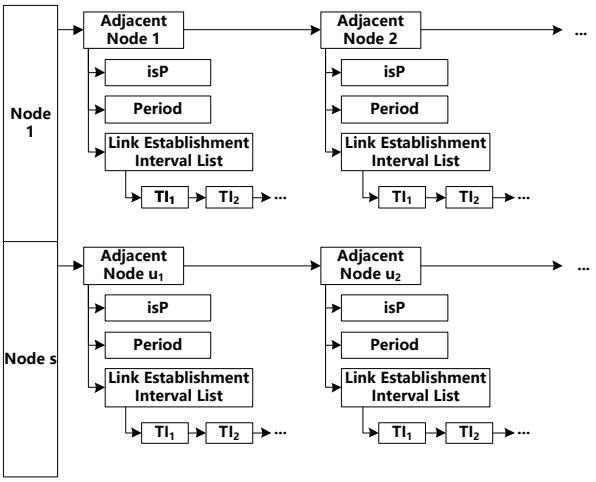


Figure 1 Space-terrestrial network topology model based on adjacency list.

We leverage this model to imprint the forthcoming topology in a data-efficient format to configure our SATNET-OSPF framework, further described in the following subsections.

3.2 | Interface Type for Space Links

Because of the distances involved, space links are typically constrained in bandwidth. This is an issue for OSPF-based protocols that need to calculate the routing table on the premise of synchronizing the link state database of the entire network. As a result, it is mandatory to study how to reduce OSPFv3 interaction overhead in SATNET-OSPF. In this section, we take the first step by choosing the appropriate interface type for space links.

The interface types for representing links in OSPFv3 include *i*) point-to-point, *ii*) broadcast, *iii*) NBMA (non-broadcast multiple access), *iv*) Point-to-Multi-Point, and *v*) virtual links. For example, in terrestrial networks, links between routers usually utilize Ethernet cables, which are modeled as broadcast links. However, space wireless links have different characteristics and require careful attention to classify them correctly.

Space links are divided into inter-satellite links (ISL), ground-to-satellite links (GSL), and user data links. ISLs can be further divided into a) intra-plane links, which connect neighboring satellites in the same orbital plane; b) inter-plane links, which connect neighboring satellites in adjacent orbits; and c) inter-layer links, which connect neighboring satellites in different layers, as illustrated in Fig. 2. Since the links modeled by OSPFv3 are those between routers, we focus on ISLs and GSLs.

To maximize the throughput over large distances, high-gain directional antennas typically support both ISLs and GSLs. Gimbals and body-pointing approaches are often leveraged to allow spacecraft to point to the destination end-point dynamically. Thus, there is generally only one node at either end of the ISL or the GSL. Even if the ISLs and GLSs are dynamically switched, at a certain moment, there is still only one router node at either end of the link.

As a result, we claim ISLs and GSLs can adopt the simplest point-to-point interface type rather than the broadcast interface type.

Notably, selecting the point-to-point interface type allows us to avoid the election process of Designated Router (DR) and Backup Designated Router (BDR) as required for the broadcast or NBMA link. Moreover, since network link state advertisements (Network-LSA) are unnecessary in point-to-point links, the resulting protocol interaction overhead between routers is significantly reduced.

3.3 | Fast Link State Detecting Mechanism

In OSPF terminology, a link state can be either *up* or *down*. When the routing protocol detects a link state change, it propagates the update to the rest of the network. While the goal is to keep a consistent network-wide topological view, the detection and propagation of the update incur in a period during which route loops and data loss might occur due to conflicting route table information. We analyze link-up and link-down events in the following subsections to specify which adaptations must be considered for SATNET-OSPF.

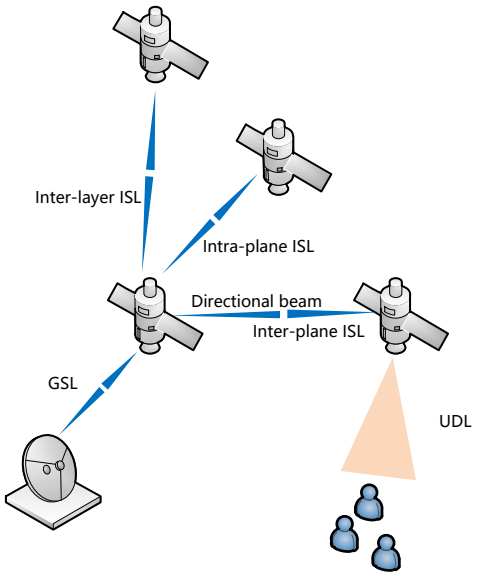


Figure 2 Space links are divided into inter-satellite links, ground-to-satellite links, and user data links.

3.3.1 | Link-Up Event

When a link transits from down to up, the OSPF protocol will either a) keep the original transmission path or b) switch to a better transmission path enabled by the new link. Because the old and new paths are still valid, the route convergence process will not generate loops or data loss in most cases. As a result, link-up events do not require adaptation when considered for SATNET-OSPF. This crucial insight significantly enhances the applicability of our contribution to the dynamic environment of satellite constellations. We defend this claim utilizing an 8-node network example presented in Fig. 3.

Initially, at time T_0, the link between node 4 and node 8, denoted as $Link(4,8)$, is down. In this topology, traffic from node 1 to 8 will flow along $Path2$ $(1-2-5-6-7-8)$ rather than $Path1$ $(1-2-3-4-8)$, which would be the shortest if $Link(4,8)$ were not down.

Then, at time T_1, $Link(4,8)$ changes its state to up. Node 4 detects the link-up event at time T_2 and updates its routing table (and the forwarding table) through the OSPFv3 protocol at time T_3. Also, node 4 generates a Link State Update (LSU) packet and floods it to node 3. Node 3 receives the LSU packet and updates its routing table at time T_4. Next, node 3 floods the LSU packet to node 2. Node 2 updates its routing table at time T_5. By the time T_6, all nodes receive the LSU and update their routing and forwarding table. However, nodes 1, 5, 6, and 7 keep their routing table to destination node 8 unchanged.

During the detection process of the $Link(4,8)$ up event ($[T_1, T_2]$) and the routing table update process ($[T_2, T_3]$) no data is lost. The main reason was: no traffic was flowing via the non-existing $Link(4,8)$. Furthermore, the LSU flooding process ($[T_3, T_6]$) will not generate routing loops during a link-up event. The timeline of the whole route convergence process is presented in Fig. 5. The period $[T_1,T_2]$ is the link state detection time, and the period $[T_1,T_6]$ is the routing convergence time.

3.3.2 | Link-Down Event

While frequent link-up events can follow standard OSPF processes without generating issues, link-down events demand more careful consideration. When a link transits from an up to a down state, the OSPF routing protocol must switch original transmission paths via such links to a new one over a different link. During the route convergence period, the original transmission path might remain a valid option for some nodes in the network. Therefore, data loops and loss can occur in this case. To illustrate the phenomena, we continue the example from the previous section in Fig. 4.

Node 1 is sending data to node 8 via $Path1$ $(1-2-3-4-8)$, but $Link(4,8)$ switches to a down state at T_1, which is detected by node 4 and node 8. We next focus on the process starting at node 4 since it is at the upper stream of $Path1$. Node 4 detects the link down event at time T_2 and updates its routing table through the OSPF protocol at time T_3. Next, node 4 will flood the LSU message announcing the link down event to node 3, which will update its local routing table at time T_4. The same update flows from node 3 to node 2 at time T_5. Finally, at time T_6, the route convergence process completes; namely, the data transmission path from node 1 to node 8 is switched from $Path1$ to $Path2$ $(1-2-5-6-7-8)$.

For the data sent from node 1 to node 8 during the period $[T_1,T_5]$, since the routing table of the nodes on the data transmission path is inconsistent with the actual network state, data loss will occur. Specifically:

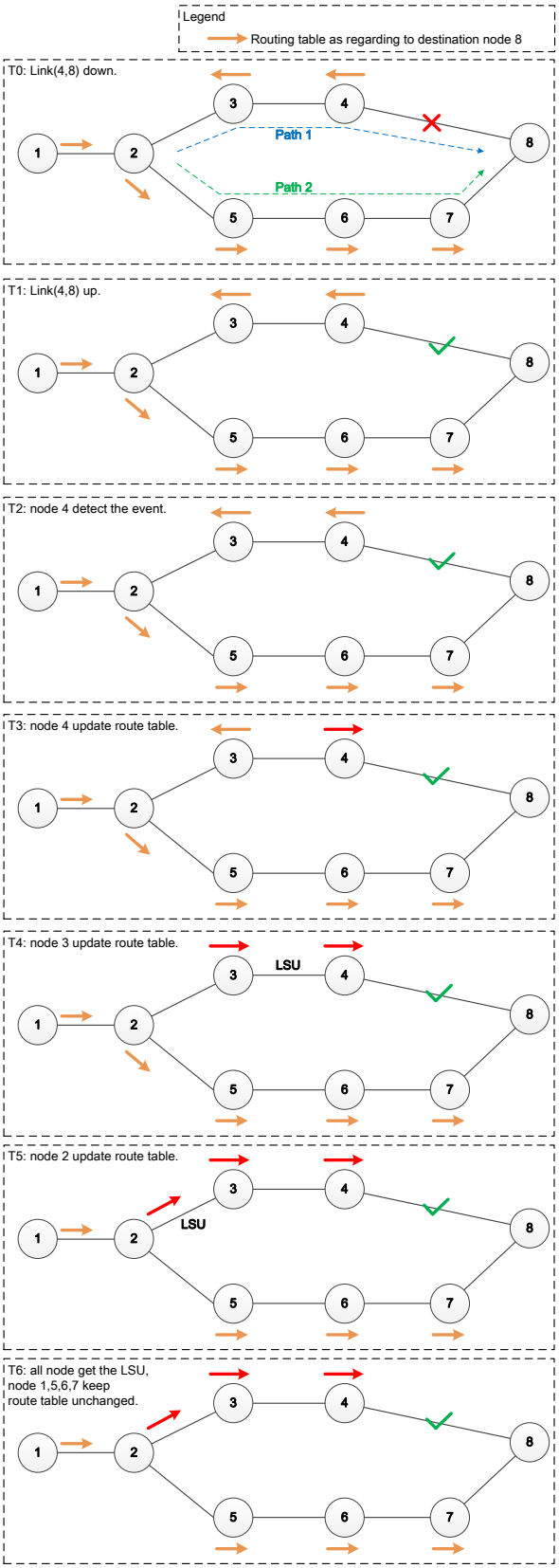


Figure 3 The response process of OSPF after the link is up.

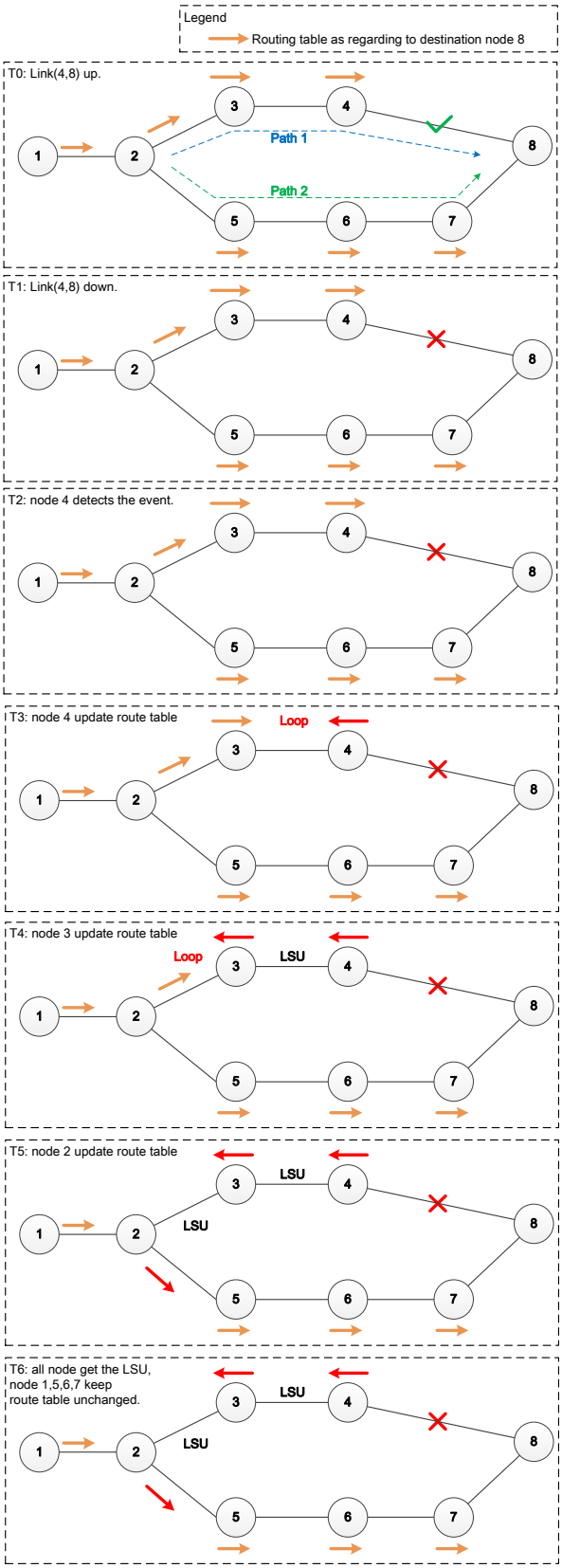


Figure 4 The response process of OSPF after the link is down.

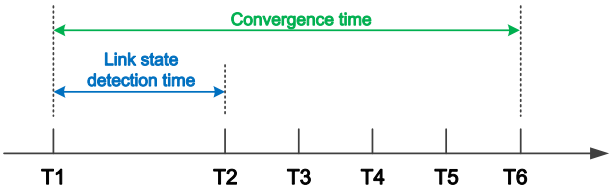


Figure 5 Timeline of the route convergence process.

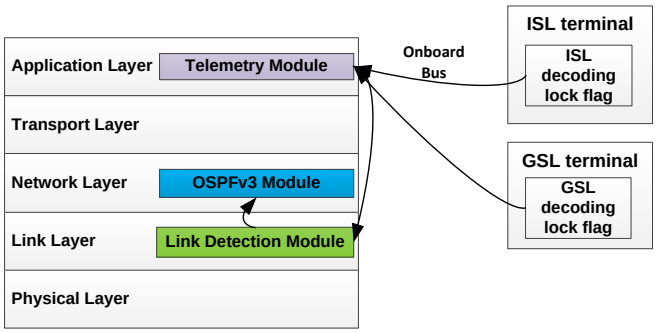


Figure 6 Fast link state detecting mechanism of ISLs and GSLs.

- During the period $[T_1,T_3]$, data received by node 4 continues to be sent to 8 via $Link(4,8)$, as the routing table was not yet updated, resulting in data loss.
- During the period $[T_3,T_4]$, data between node 3 and 4 is looped due to the inconsistent routing tables, causing packets to be dropped because of the exceeded hop limit count.
- During the time period $[T_4,T_5]$, due to the inconsistent routing table between node 2 and node 3, the same routing loop issue will provoke data loss.

3.3.3 | Fast Link State Detecting Mechanism

This section will focus on reducing the link state detection time $[T_1,T_2]$.

The OSPFv3 protocol uses periodical Hello packets to detect the link state change. If the nodes at both ends of the link receive the Hello packet sent by the other party, the link is considered *up*. Otherwise, if nodes do not receive the Hello packet within 4 times the Hello packet interval, the link is considered to be *down*, triggering the route convergence process. Generally, the Hello packet interval of OSPFv3 is 10 seconds so the link-down detection interval can take up to 40 seconds.

Because of the resulting high delay in link state detection, terrestrial Ethernet interfaces usually leverage the link status information provided by the physical layer chip. For example, when we unplug an Ethernet cable, the Ethernet chip immediately detects it and sends a report to the upper layer.

The ISLs and GSLs of space networks generally adopt a wireless link instead of an Ethernet link. Inspired by the Ethernet approach, we propose to profit from the sequential communication process of the wireless GSL and ISL transponders, i.e., a) the bit synchronization locking, b) the frame synchronization locking, and c) the decoding synchronization locking. In other words, SATNET-OSPF utilizes the decoding lock flag of the ISLs and GSLs to indicate the link up and down.

The fast link state detecting mechanism of the ISLs and GSLs we designed is shown in Fig. 6. The Telemetry Module on the satellite regularly collects the decoding lock flag of ISLs and GSLs through the onboard satellite bus, e.g., 1553B, CAN, serial. The decoding lock flag is then sent to the Link Detection Module, which reports the link status to the OSPFv3 Module. As a result, fast detection of link status similar to terrestrial Ethernet is achieved in SATNET-OSPF.

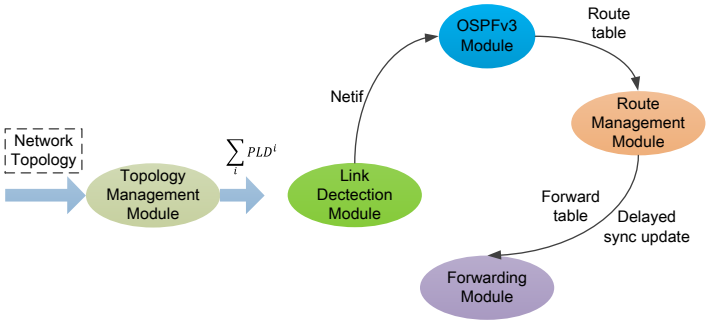


Figure 7 Routing and Forwarding control mechanism for Predictable link down events (RFP).

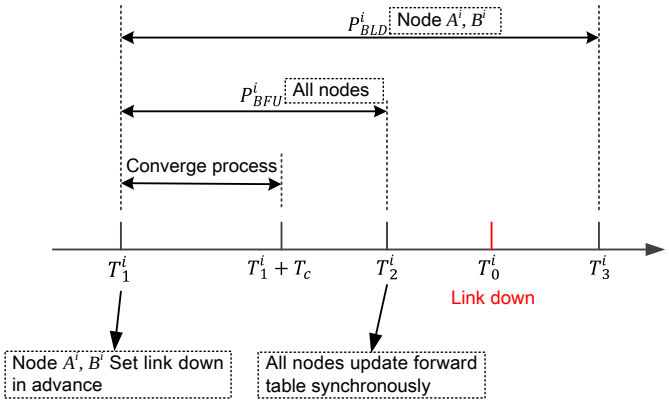


Figure 8 RFP action throughout the PLD time points.

3.4 | Routing and Forwarding Control Mechanism for Predictable Link Down Events

Due to the relative motion of nodes in space networks, the ISLs and GSLs must be dynamically switched, leading to a continuous (predictable) update of the network topology. Moreover, due to the harsh space environment, space links and nodes are prone to failure, which will also cause (unpredictable) changes in the network topology. However, predictable link updates occur far more frequently than failures.

Because the topology of a terrestrial network is relatively stable, the OSPFv3 routing protocol only considers unpredictable topology changes and lacks effective handling measures for predictable topology updates.

This section proposes a proactive Routing and Forwarding control mechanism for Predictable link-down events (RFP). RFP performs reliable routing and forwarding when the link-down events are predictable. The flow diagram of RFP is presented in Fig. 7. The RFP mechanism requires the cooperation of the Topology Management Module (TMM), the Link Detection Module (LDM), the OSPFv3 Module, the Route Management Module (RMM), and the Forwarding Module.

We define a Predictable Link Down event i as $PLD^i(X, A^i, B^i, T^i_0, T^i_1, T^i_2, T^i_3)$, where X is the link expected to change to a down state, A^i, B^i are the nodes at both ends of link X, T^i_0 is the time when the predictable link down event occurs, $T^i_1 = T^i_0 - T_c - 2 * dT$, $T^i_2 = T^i_0 - dT$, $T^i_3 = T^i_0 + dT$, T_c is the route convergence time of the whole network, and dT is a system-wide safe-margin parameter. For any PLD^i, the relationship of different time points and the action of RFP is shown in Fig. 8.

Blocking period of Forward table Update (BFU)

We define $P^i_{BFU} = \{t | T^i_1 \leq t < T^i_2\}$ as the Blocking period of Forward table Update (BFU) corresponding to predictable link down event PLD^i. During the BFU period in SATNET-OSPF, the RMM of all nodes in the space network (not only nodes A^i, B^i) will receive the new routing table, but they won't immediately impact it into the forwarding table. Instead, the forwarding table will be delivered to the Forwarding Module at a later moment defined by the BFU. Only after the BFU period expires does the RMM of all nodes send the forwarding table to the Forwarding Module. Thanks to the BFU mechanisms, this happens synchronously at time T^i_2 (see Fig. 8), thus allowing a fast and synchronous network-wide response to a predictable link state update. Indeed, the BFU allows the new routing table to be generated in all nodes before it becomes active in the network.

Blocking period of Link Detection (BLD)

We define $P^i_{BLD} = \{t | T^i_1 \leq t < T^i_3\}$ as the Blocking period of Link Detection (BLD) corresponding to predictable link down event PLD^i. During the BLD period, the LDM of nodes A^i and B^i will no longer notify the OSPFv3 Module via the network interface (Netif) of the latest up-down status of link X. Once the BLD period is over, the Link Detection Module of node A^i and B^i will resume notifying the OSPFv3 Module.

3.4.1 | RFP Mechanism

The RFP mechanism will distribute the space network topology introduced in section III-A to each node of the space network in advance. The distribution source can be the ground control center or a satellite that propagates and plans the space network topology. The TMM of each node in the space network will extract and determine all the predictable link-down events for the whole network ($\sum_i PLD^i$) according to the received space network topology. Then, each node will take corresponding measures according to $\sum_i PLD^i$. The overall step-by-step RFP process is as follows:

1. The LDM of node A^i and B^i at both ends of link X will set link X to be down in advance at time T^i_1 (the actual down event of link X will occur at time T^i_0).
2. Nodes A^i and B^i at both ends of link X enter the BLD period P^i_{BLD} synchronously. During the BLD period, the LDM of nodes A^i and B^i will no longer notify the OSPFv3 Module via Netif with the latest up-down status of link X. At this time, the LDM assumes that link X is down while it is still up.
3. All nodes in the entire network (not only nodes A^i, B^i) enter the BFU period P^i_{BFU}. During the BFU period, nodes refrain from impacting the routing table received by RRM into the forwarding table.
4. At the time $T^i_1 + T_c$, the entire space network completes the routing convergence process, and the RMM of all nodes has already received the newest route table.
5. At the time T^i_2, all nodes in the network exit the P^i_{BFU} period, meaning all RMMs will transform the routing table into the forwarding table, which is synchronously provided to the Forwarding Module (i.e., no inconsistency).
6. At time T^i_0, the predictable link-down event PLD^i occurs. But the network-wide forwarding table has been updated in advance, and the data transmission path has already bypassed link X (i.e., no routing loops or data loss).
7. At time T^i_3, node A^i and B^i at both ends of the link X exit the BLD period P^i_{BLD}, and resume notifying the OSPFv3 Module via Netif with the latest up-down status of link X.

If we apply the RFP mechanism to the scenario in Fig. 4, the evolution of the data flow becomes the one in Fig. 9, where the routing loop and subsequent data loss are removed. Node 4 and 8 will be informed the $Link(4, 8)$ will be down since T_0. Both nodes will compute T_1, T_2, and T_3 in advance and execute RFP steps 1) to 7). Thanks to the blocking periods BFU and BLD, both nodes will perform the link-down detection in advance (at T_1, converging at $T_1 + T_c$) and update the routing and forwarding table synchronously (at T_2) so that traffic flows via $Path2$. At T_0, the $Link(4, 8)$ will effectively switch to a down state without impacting the traffic. At the end of the RFP process, both nodes resume nominal link status report to OSPF via the Netif interface (at T_3).

The differences in handling the predictable link-down event between standard OSPFv3 and our enhanced SATNET-OSPF are shown in Fig. 10. The standard OSPFv3 protocol needs to detect the predictable link-down event after the link passively is down at T_0. Thus, the switch to $Path2$ occurs after convergence is achieved (at $T_0 + T_c$). However, during the route convergence process, data transmission will become unreliable. Instead, the SATNET-OSPF enhancement will set the link to be down at the time of T_1 and complete the route convergence process in advance. With SATNET-OSPF, the entire network will update the forwarding table synchronously and switch to $Path2$ by T_2.

Some border-line remarks on the operation of RFP:

1. BFU and BLD overlap: when two or more predictable link-down events are very close, the BFU and BLD periods may overlap. Then, the BFU and BLD periods will be the union of these periods, respectively, i.e., $P_{BFU} = \sum_i P^i_{BFU}$, and $P_{BLD} = \sum_i P^i_{BLD}$.
2. BFU and unpredictable events: if an unpredictable link-down event occurs within the BFU period P^i_{BFU}, the corresponding routing table update must also be delayed until the end of the BFU period. While this will impact the performance, the chances of failures within the BFU period are relatively low.

With these considerations at hand, the RFP mechanism will not change the standard operation of OSPFv3, so the compatibility of SATNET-OSPF with standard OSPFv3 is kept.

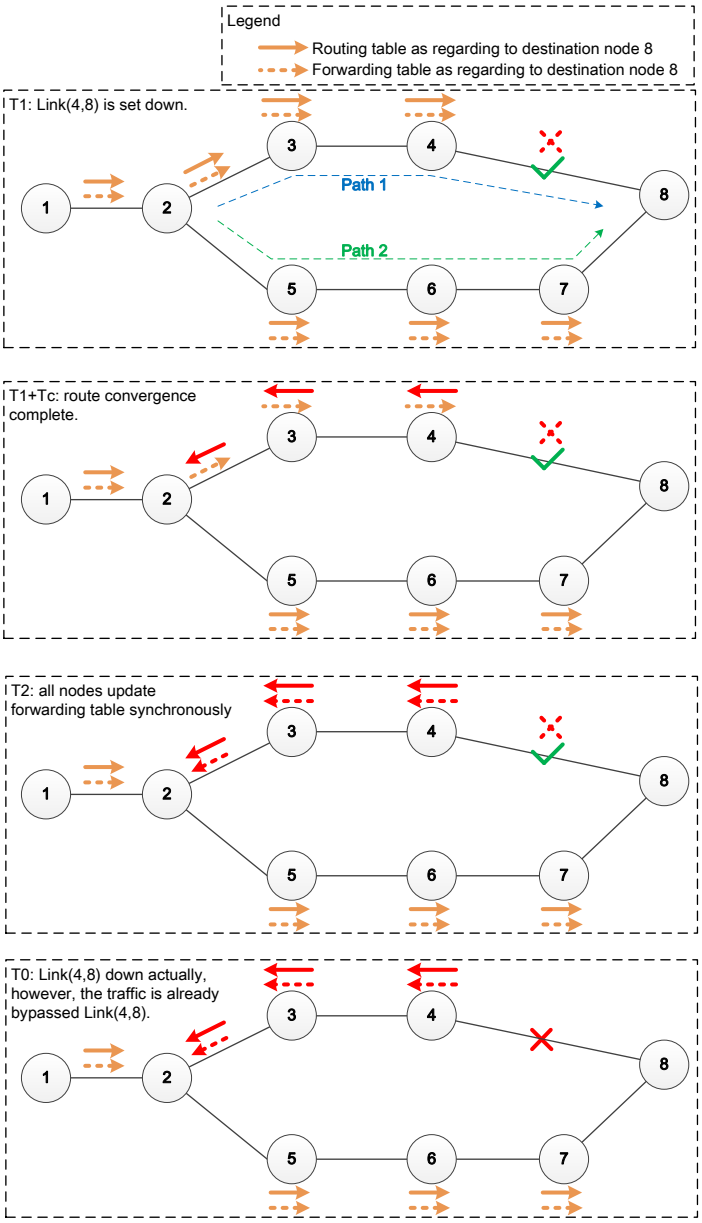


Figure 9 The change of routing table and forwarding table of each node at each time point.

4 | SPACE ROUTER IMPLEMENTATION

To accurately gauge the performance of our proposed solution, we have implemented the SATNET-OSPF routing protocol on a space router that is prepared for flight. This space router utilizes a two-layer architecture, composed of CPU and FPGA [43,44], and is capable of managing 2 bi-directional Ground Station Links (GSLs) and 4 bi-directional Inter-Satellite Links (ISLs). The architecture of this space router is illustrated in Fig. 11.

Our space router employs cutting-edge space hardware. We have opted for a space-grade SPARC V8 processor for the CPU, operating at a clock frequency of 64 MHz. The processor is equipped with Error Detection and Correction (EDAC) capabilities, enabling it to correct one-bit errors and detect two-bit errors. The primary role of the CPU is to manage and operate the control plane. Regarding the FPGA, we have selected a space-grade Xilinx Virtex-5 FPGA comprising 13 million logic cells and 10 Mbit Block RAM. The FPGA is tasked with handling high-speed data plane forwarding.

On the software front, the CPU operates on the VxWorks operating system. The IPv6 protocol stack is provided by LwIP [45 42]. For the OSPFv3 protocol stack, we employ the Quagga routing suite [14], utilizing ospf6d to facilitate OSPFv3 protocol functions. Quagga interfaces with the LwIP-provided API to send and receive IPv6 packets. At the application layer, the Space Packet Protocol [46] is used to encapsulate Telemetry (TM) and

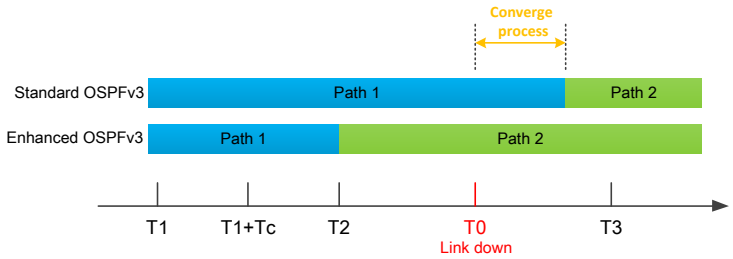


Figure 10 The differences in handling the predictable link down event between standard OSPFv3 and SATNET-OSPF.

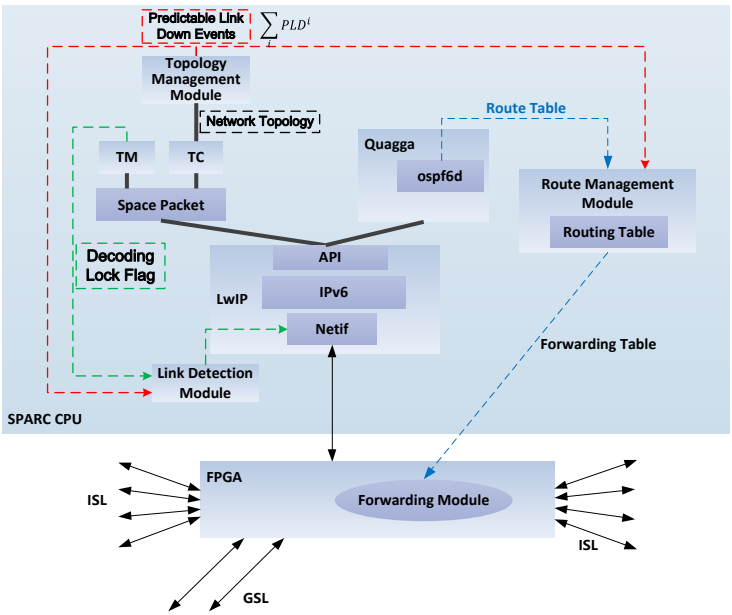


Figure 11 Space router architecture.

Telecommand (TC) services. This space packet is then enveloped within the UDP protocol to communicate with the underlying IPv6 stack. Before the operation, the space network topology is dispatched to each node's Topology Management Module (TMM) via Telecommand.

Drawing on the telemetry data, the Topology Management Module will extract and schedule all the predictable link-down events of the network ($\sum_i PLD^i$) using the received space network topology. Subsequently, the Link Detection Module will receive the comprehensive, predictable link-down events of the network $\sum_i PLD^i$ from the Topology Management Module (as indicated by the red dashed line in Fig. 11). Using this information, it will compute the Blocking period of Link Detection (BLD) $\sum_i P^i_{BLD}$. If outside the BLD period, the Link Detection Module will determine whether the space link is active or inactive by using the decoding lock flag received from the Telemetry (TM) module (as depicted by the green dashed line in Fig. 11). Consequently, the link status will be updated in real-time via the network interface (Netif) of LwIP, which, in turn, will update the ospf6d module. However, within the BLD period, the Link Detection Module will refrain from updating the link status to the Netif of LwIP (and, therefore, to ospf6d).

Conversely, the Routing Management Module will receive all the predictable link-down events from the Topology Management Module (as represented by the red dashed line in Fig. 11). This data will be employed to determine the Blocking period of Forward table Update (BFU) $\sum_i P^i_{BFU}$. If outside the BFU period, the Routing Management Module will convert the updated routing table received from ospf6d into a forwarding table. It will then promptly send this forwarding table to the Forwarding Module of the FPGA. However, within the BFU period, the Routing Management Module will halt the route table's delivery to the Forwarding Module of the FPGA until the final moment, when the BFU period concludes.

5 | PERFORMANCE EVALUATION

This section evaluates the performance of the SATNET-OSPF protocol using a space network emulation test bench. The test bench comprises various Commercial-Off-The-Shelf (COTS) OSPF routers and the SATNET-OSPF space router prototype we design, enabling a broad set of space-terrestrial emulation scenarios. The performance assessment is segmented into three distinct experimental categories: 1) overhead, 2) route convergence time, and 3) outage time. Each of these categories is examined below.

Overhead (Section 5.1) Taking into account the scarcity and high cost of bandwidth in space links, we assess the protocol overhead of SATNET-OSPF in Section 5.1. In the interest of precision, SATNET-OSPF messages are embedded within realistic CCSDS protocols for this evaluation.

Route Convergence Time (Section 5.2 to Section 5.5) The *route convergence time* represents another significant performance metric for SATNET-OSPF. This time period encompasses the interval from when a change in link state occurs to when the route table is updated across all network nodes.

In the context of OSPFv3 and SATNET-OSPF, nodes have the ability to detect alterations in the link state. Upon such an occurrence, they generate a link-state advertisement (LSA), which serves to notify other nodes in the network about the event. When neighboring nodes receive the LSA, they first ensure it is not a redundant copy before flooding the LSA to other nodes via remaining interfaces, excluding the interface from which the LSA was received.

Upon receipt of a valid LSA, nodes update their local link-state database and initiate the shortest path first calculation to update the OSPFv3 routing table. To ensure fast convergence, LSA flooding is carried out instantly, preceding the shortest path calculation. The route convergence process is complete when the last node concludes the shortest path first calculation.

Consequently, the route convergence time involves (*i*) the detection time of the link state change, specifically when changing to a down state in the case of unpredictable events, (*ii*) the time taken for LSA generation, (*iii*) the time for LSA forwarding, (*iv*) the signal propagation time, and (*v*) the time taken for route computation. Each of these components is individually measured and discussed in Sections 5.2 to 5.5.

Outage Time (Section 5.6) Ultimately, we assess the route outage time for SATNET-OSPF, particularly in instances of predictable link-down occurrences. In order to do this, we plan link updates through telecommand messages and compare the outage times between SATNET-OSPF and the conventional OSPFv3, as detailed in Section 5.6.

5.1 | Protocol overhead

We first evaluate the performance of the protocol overhead of SATNET-OSPF. The test scenario is shown in Fig. 12.

We use a Protocol Translation Gateway (PTG) to interconnect the space router with the terrestrial COTS router[42]. The PTG realizes the conversion between the Advanced Orbiting System (AOS)[47] protocol of the space link layer and the Ethernet link protocol of the terrestrial link layer. Thus, the space router and the COTS router are connected at the network layer level. The TLK1 interface of the PTG corresponds to the E1 interface of the PTG. For the COTS router R1, we choose the MSR2600[48] from H3C. Link 1 of the space router is connected to the TLK1 interface of the PTG through the TLK2711 interface, while the G0/1 port of R1 is connected to the E1 interface of PTG through an Ethernet cable. The G0/0 port of R1 is connected to a test PC. We mirror the traffic of the G0/1 port to the G0/0 port of R1 and use Wireshark on the test PC to monitor the OSPFv3 interaction process between the COTS R1 and the space router SR1.

We did two experiments to evaluate the differences in protocol overhead between the point-to-point interface and the broadcast interface types. During the first experiment, we set the OSPFv3 interface type of Link 1 of SR1 and the G0/1 port of R1 to broadcast interface type. For the second experiment, we set the OSPFv3 interface type of Link 1 of SR1 and the G0/1 port of R1 to the point-to-point interface type. Since higher protocol overhead means longer convergence time and higher protocol traffic, we compare the route convergence time, total protocol traffic overhead, and average protocol traffic overhead during the initial route establishment process for both experiments.

The experiment results are summarized in Table 1. It can be seen that compared with the broadcast interface type, the initial convergence time and the total protocol traffic overhead of the point-to-point interface type can be reduced by 12.5% and 31.4%, respectively. On the other hand, the average protocol overhead can be reduced from 3.01 Kbps to 2.36 Kbps. By checking the router's Link State Database, it can be found that the broadcast interface type test case has one more Network-LSA. Because of the extra Network-LSA on the broadcast link, the protocol interaction takes more time, and the total protocol traffic overhead and the average protocol overhead increase. Lower protocol overhead means SATNET-OSPF will consume less network communication resources and have lower impact to the user traffic.

5.2 | Detection time of unpredictable link down events

Our second experiment focused on the detection time of unpredictable link-down events. The test bench configuration is shown in Fig. 13.

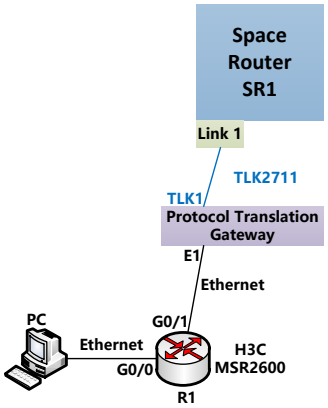


Figure 12 Protocol overhead experiment setup.

Table 1 Protocol overhead experiment results.

	Broadcast int.	Point-to-point int.	Improvement
Convergence time (s)	8	7	12.5%
Overhead (Bytes)	3086	2118	31.4%
Overhead (Kbps)	3.01	2.36	21.6%

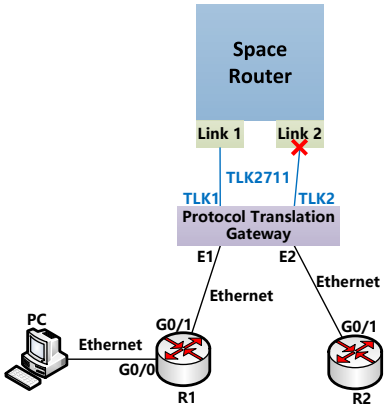


Figure 13 Detection time of unpredictable link down events experiment setup.

The TLK1 of the PTG links to E1 of the PTG, while the TLK2 of the PTG links to E2 of the PTG. Link 1 of SR1 is connected to TLK1 of the PTG, while Link 2 of SR1 leads to TLK2 of the PTG. The G0/1 port of R1 is connected to E1 of the PTG, and The G0/1 port of R2 is connected to E2 of the PTG. The G0/0 port of R1 is connected to a test PC. The status of the space router SR1 is monitored through the Telemetry downloaded to the test PC.

The decoding lock flag of Link 1 and Link 2 is set to be locked in the initial state. Then, after the SR1 has completed the route convergence process with R1 and R2 through Link 1 and Link 2, we set the decoding lock flag of Link 2 to be unlocked to test the detection time of an unpredictable link-down event. We download the recorded time when Link 2 is down and the timestamp when the routing table is updated to the test PC via Telemetry. Then, the difference between these times is the detection time of the unpredictable link-down event.

The test results are summarized in Table 2. We conduct a total of 10 tests. We record the time stamp for each test when the link is down (T_1) and when the route table is updated (T_2). Then the detection time is $T_2 - T_1$. From the test result, we can see that the detection time of the unpredictable link-down event is 0.192 seconds (192 milliseconds) on average, which is much faster than using the hello mechanism of OSPFv3, which is 40 seconds at most.

Table 2 Detection time of unpredictable link down event experiment results.

Run	Link down time	Route table update time	Detection time
1	138.831 s	138.876 s	0.045 s
2	427.934 s	427.98 s	0.046 s
3	473.238 s	473.544 s	0.306 s
4	517.835 s	517.881 s	0.046 s
5	571.086 s	571.545 s	0.459 s
6	638.737 s	638.782 s	0.045 s
7	688.187 s	688.546 s	0.359 s
8	721.438 s	721.547 s	0.109 s
9	762.088 s	762.547 s	0.459 s
10	790.789 s	790.835 s	0.046 s
		Average	0.192 s

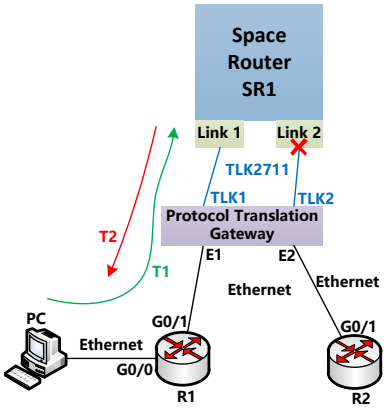


Figure 14 Flood generation delay experiment setup.

5.3 | Flood generation delay

In this subsection, we measure the flood generation delay and the experiment scenario is shown in Fig. 14. The setup is similar to the one used for the detection time of unpredictable link-down events. We mirror the traffic of the G0/1 port to the G0/0 port of R1 and then use Wireshark on the test PC to monitor the traffic between R1 and SR1. We also use a Telecommand sent from the test PC to set Link 2 of SR1 to be down instead of putting the decoding lock flag (The space router SR1 supports both modes of link status control).

Initially, Link 1 and Link 2 of SR1 are both up. After SR1 and R1 have completed the route convergence process, a Telecommand, which will set the Link 2 of SR1 to a down status, is sent to the space router SR1 from the test PC via R1 to SR1 at time T_1 (the green arrow in Fig. 14). Then the space router SR1 will generate an LSU packet and flood it to the COTS router R1 at time T_2 (the red arrow in Fig. 14). We use Wireshark on the test PC to monitor the traffic between R1 and SR1. We denote when the Telecommand appears on the G0/1 port as T_1 and when the LSU packet appears on the G0/1 port as T_2. Then We denote $T_2 - T_1$ as the flood generation delay once the link is down.

The test results of the flood generation delay experiment are summarized in Table 3. We conducted a total of 5 runs, and the average value of the flood generation delay on the implemented space router measured was 192ms.

Table 3 Flood generation delay experiment results.

Run	Link down time	LSU packet time	Flood generation delay
1	328 ms	514 ms	185 ms
2	714 ms	910 ms	196 ms
3	217 ms	404 ms	188 ms
4	466 ms	661 ms	195 ms
5	626 ms	823 ms	197 ms
		Average	192 ms

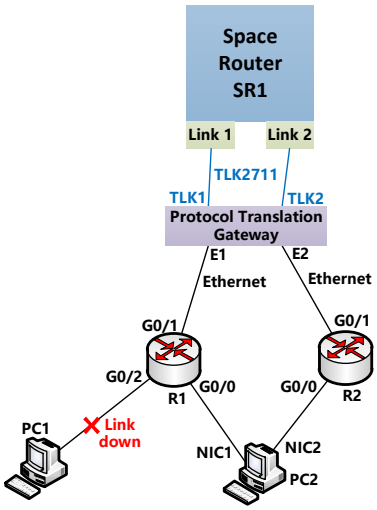


Figure 15 Flood forwarding delay experiment setup.

5.4 | Flood forwarding delay

The fourth experiment we conducted was the flood forwarding delay test. The experiment setup is shown in Fig. 15.

In this case, Link 1 of SR1 is connected to TLK1 of the PTG, and the E1 of the PTG is connected to the G0/1 port of R1. Link 2 of SR1 is connected to TLK2 of the PTG, and the E2 of the PTG is connected to the G0/1 port of R2. The G0/0 port of both R1 and R2 are connected to Network Interface Card (NIC) 1 and 2 of the test PC2, respectively. The G0/2 port of R1 is connected to the test PC1. We mirror the traffic of G0/1 of R1 to G0/0 of R1 and the traffic of G0/1 of R2 to G0/0 of R2; then, on the test PC2, we use two Wireshark instances to monitor the traffic of both the G0/1 port of R1 and the G0/1 port of R2.

Initially, the test PC1 is connected to G0/2 of R1. Then, after the space router SR1 has completed the route convergence with the COTS router R1 and R2, we unplug the Ethernet cable between the test PC1 and G0/2 port of R1. At this moment, R1 will detect that the G0/2 port is down, generate an LSU packet 1, and flood it to SR1 via the G0/1 port. After the SR1 receives the LSU packet 1 from Link 1, it will flood LSU packet 2 to COTS router R2 via Link 2. Both Wireshark monitoring NIC1 and NIC2 will capture these flooding messages. We denote when the LSU packet 1 appears on the G0/1 port of R1 as T_1 and when the LSU packet 2 arrives on the G0/1 port of R2 as T_2. Then, the flood forwarding delay of the implemented space router can be calculated as $T_2 - T_1$.

We conducted 5 runs, and the experiment results are shown in Table 4. The average flood forwarding delay of the implemented space router is 37ms.

5.5 | Route computation time

The fifth experiment we conducted was the route computation time of the implemented space router. The test setup is presented in Fig. 16.

Table 4 Flood forwarding delay experiment results.

Run	LSU-1 time	LSU-2 time	Flood forwarding delay
1	80 ms	112 ms	32 ms
2	547 ms	580 ms	33 ms
3	265 ms	307 ms	42 ms
4	520 ms	554 ms	34 ms
5	256 ms	298 ms	42 ms
		Average	37 ms

The shortest path calculation is the most time-consuming part of the OSPFv3 protocol (and thus SATNET-OSPF). We download the computation time of the shortest path calculation of ospf6d to the test PC through Telemetry. We designed three scenarios to test the route computation time, illustrated in Fig. 16. From left to right, we connect additional routers (i.e., we join R2 in scenario 1, R3 in scenario 2, and R4 in scenario 3). Furthermore, we evaluate the route update time when these routes are disconnected. The final goal is to analyze the route computation time of the OSPFv3 protocol on the implemented space router in these three scenarios with increasing topology sizes.

We recorded the route computation time of the OSPFv3 protocol when the additional router node was added and removed from each scenario. Each experiment was conducted 5 times. The test results are listed in Table 5. The test results show that the route computation time when the additional router node was connected (conn.) is higher than that of the disconnected (disc.). This is expected since the newly connected router makes the network larger. Also, as the number of nodes in the network increases, the route computation time remains within reasonable bounds, and the route computation time on a space-grade 64 MHz SPARC V8 processor of a 5-node network is not higher than 17 ms.

5.5.1 | Route Convergence Time

As mentioned, by obtaining the detection time of link down event, the flood generation delay, the flood forwarding delay, and the route computation time, we can estimate the route convergence time of the space network connected by the space router using SATNET-OSPF. The route convergence process is illustrated in Fig. 17.

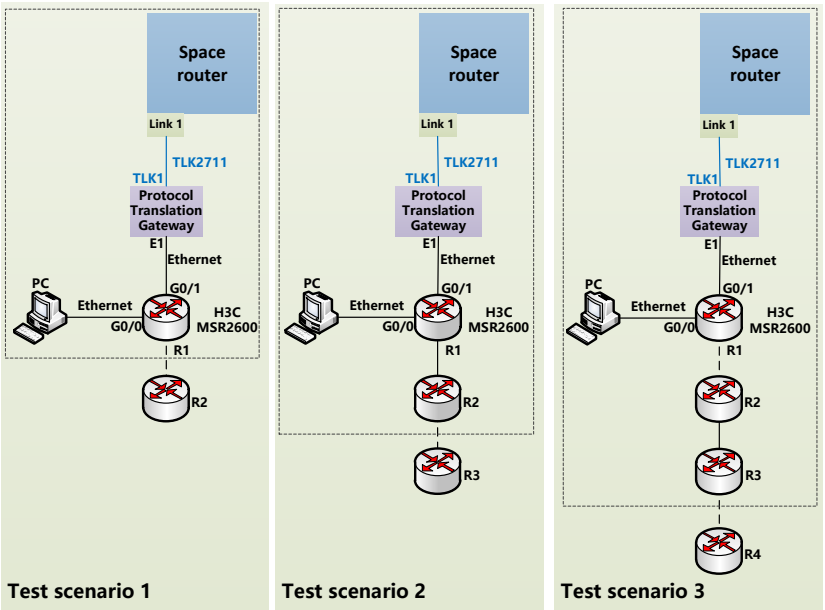


Figure 16 Route computation time experiment setup.

Table 5 Route computation time experiment results.

Run	Scenario 1 (disc.)	Scenario 2 (disc.)	Scenario 3 (disc.)
1	10 ms	12 ms	14 ms
2	10 ms	12 ms	14 ms
3	11 ms	12 ms	14 ms
4	11 ms	12 ms	14 ms
5	11 ms	12 ms	14 ms
Avg.	10.6 ms	12 ms	14 ms
Run	**Scenario 1 (conn.)**	**Scenario 2 (conn.)**	**Scenario 3 (conn.)**
1	13 ms	13 ms	17 ms
2	12 ms	13 ms	17 ms
3	13 ms	14 ms	16 ms
4	13 ms	13 ms	16 ms
5	13 ms	14 ms	16 ms
Avg.	12.8 ms	13.4 ms	16.4 ms

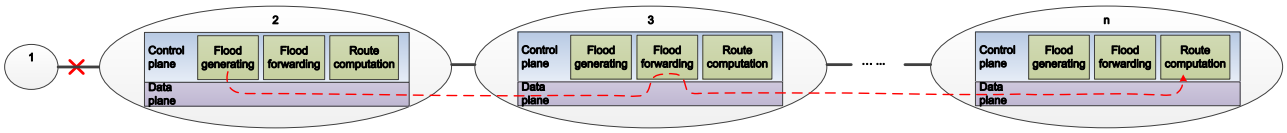


Figure 17 Route convergence process.

First, node 2 detects the link state down and generates the LSU packet. Then, the next hop floods (forwards) the LSU packet to the next node before conduct the shorted path first calculation. Finally, the last node n, which lies farthest from the original node 2, receives the LSU packet. With this information, node n executes the route computation routine.

Consequently, if T_d is the detection time of link down event (e.g., 192 ms), T_{fg} is the flood generating delay (e.g., 192 ms), T_{ff} is the flood forwarding delay (e.g., 37 ms), T_{comp} is the route computation time (e.g., 17 ms), N_h is the hop count of the longest path, and T_p is the accumulated space link propagation delay of the longest path, the resulting route convergence time T_c of the whole space network can be computed as follows:

$$T_c = T_d + T_{fg} + T_{ff} * N_h + T_p + T_{comp}$$

For space networks with different scales, the detection time of link down event T_d, the flood generating delay T_{fg} and the flood forwarding delay T_{ff} remain stable: this is the main characteristic of the SATNET-OSPF routine on the implemented space router hardware. However, the hop count of the longest path N_h, the accumulated space link propagation delay of the longest path T_p, and the route computation time T_{comp} will vary with the scale of the space network. The first two factors, N_h and T_p can be obtained through analytical calculations, while T_{comp} can be obtained through emulation.

5.6 | Routing outage during predictable link-down events

The sixth experiment we conducted was the route outage imposed by the predictable link-down events and the RFP mechanism. The experiment setup is illustrated in Fig. 18.

We use two space routers, SR1 and SR2, and three COTS routers to build the test bench. SR1 and SR2 are interconnected through the TLK2711 interface via Link 3. Link 1 of SR1 is connected to the TLK1 of PTG1, while the E1 of PTG1 is connected to G0/1 of R1. The G0/0 of R1 is connected to the test PC1. Link 1 and Link 2 of SR2 are connected to the TLK1 and TLK2 of the PTG2. The E1 of the PTG2 is connected to G0/1 of R2, while the E2 of the PTG2 is connected to G0/1 of R3. The R2 and R3 are interconnected through G0/2. The G0/0 of R2 is connected to the test PC2. The G0/0 of R3 is connected to the test PC3. The PC1 and PC2 are used for the ping application, while the PC3 is used to receive the Telemetry of SR1 and SR2 to monitor the network status.

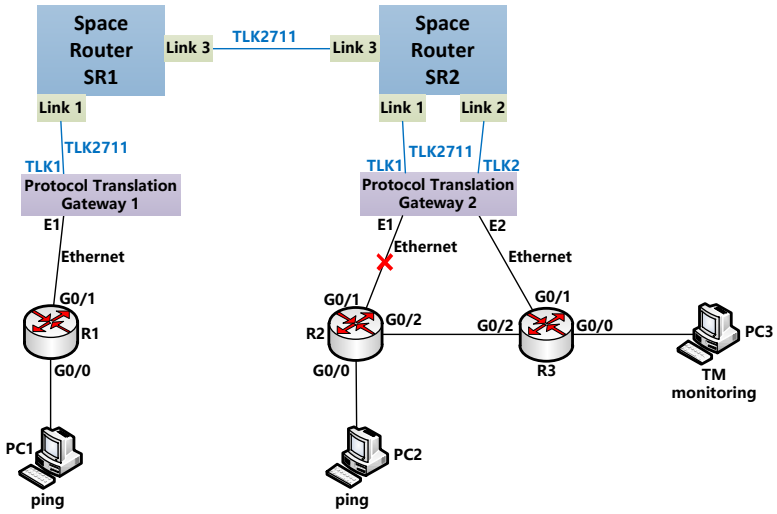


Figure 18 Route outage during predictable link-down events experiment setup.

Table 6 Route outage during predictable link-down events experiment parameters.

Parameter	Value (second)
dT	10
T_c	10
T_0	363000000
T_1	362999970
T_2	362999990
T_3	363000010
Δt	0.001

We assume that the link between R2 and SR2 will be down at 363000000 seconds, i.e., this is a predictable link-down event. Then we test and measure the routing outage of standard OSPFv3 and SATNET-OSPF when this event occurs.

The test parameters are summarized in Table 6. The Δt value indicates the time interval between subsequent pings. To facilitate the observation of the system response during the test, dT and T_c are set to a more considerable value of 10 seconds. According to the network condition, these two parameters can be set to smaller values in a real space system.

The experiment steps of enhanced OSPFv3 are as follows:

1. At time 362999940s, the space network topology information is sent as a Telecommand to SR2. The SR2 will then extract the information of the link between R2 and SR2, which will be down at 363000000s seconds.
2. At time 362999950s, we confirm that the link between R2 and SR2 is up, and SR1, SR2, R1, R2, and R3 have completed the route convergence process.
3. At time 362999960s, PC1 starts to ping PC2 at Δt intervals. Through the Telemetry received on the test PC3, we can determine the data transmission path at this time is $Path1$ ($PC1^{-}R1 - SR1^{-}SR2^{-}R2^{-}PC2$).
4. At time 362999970s, through the Telemetry received on the test PC3, it is confirmed that SR2 sets the link between SR2 and R2 to be down in advance. At the same time, SR2 enters the BLD period for $Link(SR2, R2)$, but the data transmission path is still $Path1$.
5. At time 362999980s, via more Telemetry received on the test PC3, we confirm that the data transmission path is still $Path1$ since the SR2 is in the BFU period.

6. At time 362999990s, through the new Telemetry received on the test PC3, we confirm that the data transmission path is switched to $Path2$ ($PC1^{-}R1 - SR1^{-}SR2^{-}R3^{-}R2 - PC2$) and the SR2 exits the BFU period.

7. At time 363000000s, we disconnect the link between R2 and SR2 (actually unplug the network cable between R2 and PTG2).

8. At time 363000010s, we confirm that SR2 exits the BLD period for $Link(SR2, R2)$.

9. We stop PC1 from pinging PC2, confirm the number of sent packets N_s and the number of response packets N_r during the ping process, and calculate the route unavailable time as $(N_s - N_r) * \Delta t$ as a route outage indicator;

The test steps of standard OSPFv3 are similar to that of SATNET-OSPF except that the SR2 will not set the link between SR2 and R2 to be down in advance. Also, the router SR2 will switch to $Path1$ after the link between SR2 and R2 is brought down.

We conducted 10 runs for standard OSPFv3 and SATNET-OSPF, respectively. The results of this last experiment campaign are shown in Table 7. The results show that during a predictable link-down event, the average route outage of standard OSPFv3 is 137.1ms. In contrast, the average outage time of SATNET-OSPF protocol is 6.2ms. Thus, the route availability of SATNET-OSPF is improved by about 22 times compared to standard OSPFv3. In the standard OSPFv3 test scenario, the COTS router R2 utilizes the Ethernet physical layer status to speed up the link-down detection process. Even under this condition, SATNET-OSPF performs significantly better than standard OSPFv3.

Table 7 Route outage during predictable link-down events experiment.

Standard OSPFv3					
Run	**Sent**	**Received**	**Dropped**	**Drop %**	**Outage**
1	24295	24150	145	0.5968%	145 ms
2	27654	27541	113	0.4086%	113 ms
3	23804	23635	169	0.7100%	169 ms
4	21079	20927	152	0.7211%	152 ms
5	18786	18658	128	0.6814%	128 ms
6	21749	21588	161	0.7403%	161 ms
7	18118	17967	151	0.8334%	151 ms
8	23380	23227	153	0.6544%	153 ms
9	24871	24778	93	0.3739%	93 ms
10	22659	22553	106	0.4678%	106 ms
				Avg.	137.1 ms
SATNET-OSPF					
Run	**Sent**	**Received**	**Dropped**	**Drop %**	**Outage**
1	54428	54423	5	0.0092%	5 ms
2	51171	51169	2	0.0039%	2 ms
3	55619	55615	4	0.0072%	4 ms
4	57037	57031	6	0.0105%	6 ms
5	54401	54391	10	0.0184%	10 ms
6	56280	56275	5	0.0089%	5 ms
7	55696	55689	7	0.0126%	7 ms
8	56675	56667	8	0.0141%	8 ms
9	54809	54800	9	0.0164%	9 ms
10	54123	54117	6	0.0111%	6 ms
				Avg.	6.2 ms

6 | CONCLUSIONS AND FUTURE WORK

This paper sheds light on the growing interest in space-terrestrial integrated networks. We have detected unique challenges in topology dynamics and mapped them to specific adaptations required in OSPFv3 to achieve seamless integration with Internet routing. We coined our framework SATNET-OSPF. Thanks to a careful selection of features, we kept the compatibility of SATNET-OSPF with legacy OSPF implementations. We could use existing software implementations and construct our flight-ready space router using space-grade hardware elements. We presented an appealing test bench connecting it with legacy COTS OSPF routers to run a series of performance measurements using real IP-based traffic. Results proved that our SATNET-OSPF solution is a solid path towards future space-terrestrial integrated networks.

In future work, we plan to keep running performance benchmarking with particular attention to CPU utilization and memory consumption of the space router. Furthermore, we plan to develop our testing framework to evaluate and demonstrate real traffic that simulates Earth observation and communication missions. We aim to investigate how the routing algorithm would impact user traffic or network performance. This includes key metrics such as delay, jitter, throughput, and network resource utilization. We believe these improvements will provide a more accurate and comprehensive representation of the system's performance under realistic operational conditions. Furthermore, because of its multi-layer cooperation, open communication environment, and time-dynamic but predictable topologies, the security aspects of SATNET-OSPF must be considered carefully. Indeed, threat analysis of the proposed SATNET-OSPF will be investigated in our future work. Of course, we look forward to seeing our space router flying in future near-Earth networked missions.

References

1. Zhao Q, Yu L, Du Z, et al. An Overview of the Applications of Earth Observation Satellite Data: Impacts and Future Trends. *Remote Sensing* 2022; 14(8): 1863.
2. Fraire JA, Iova O, Valois F. Space-Terrestrial Integrated Internet of Things: Challenges and Opportunities. *IEEE Communications Magazine* 2022.
3. Ge H, Li B, Jia S, et al. LEO enhanced global navigation satellite system (LeGNSS): progress, opportunities, and challenges. *Geo-spatial Information Science* 2022; 25(1): 1–13.
4. Stock G, Fraire JA, Hermanns H, Cruz E, Isaacs A, Imbrosh Z. On the Automation, Optimization, and In-Orbit Validation of Intelligent Satellite Constellation Operations. *arXiv preprint arXiv:2210.11171* 2022.
5. Sweeting M. Modern small satellites-changing the economics of space. *Proceedings of the IEEE* 2018; 106(3): 343–361.
6. Kodheli O, Lagunas E, Maturo N, et al. Satellite communications in the new space era: A survey and future challenges. *IEEE Communications Surveys & Tutorials* 2020; 23(1): 70–109.
7. Shang B, Yi Y, Liu L. Computing over space-air-ground integrated networks: Challenges and opportunities. *IEEE Network* 2021; 35(4): 302–309.
8. Guo H, Li J, Liu J, Tian N, Kato N. A survey on space-air-ground-sea integrated network security in 6G. *IEEE Communications Surveys & Tutorials* 2021; 24(1): 53–87.
9. Feldmann M, Fraire JA, Walter F, Burleigh SC. Ring Road Networks: Access for Anyone. *IEEE Communications Magazine* 2022; 60(4): 38–44.
10. Fischer D, Basin D, Eckstein K, Engel T. Predictable mobile routing for spacecraft networks. *IEEE Transactions on Mobile Computing* 2012; 12(6): 1174–1187.
11. Fraire JA, Gasparini EL. Centralized and Decentralized Routing Solutions for Present and Future Space Information Networks. *IEEE Network* 2021; 35(4): 110–117.
12. SDA . Transport. 2022. https://www.sda.mil/transport/. Accessed 2022-09-01.
13. Yan H, Qiao L, Niu J, et al. On Dynamic Routing Technology for Space Network. In: 2021 International Conference on Space-Air-Ground Computing (SAGC). IEEE. ; October 23-25, 2021; Huizhou, China: 106–111.
14. Jakma P, Lamparter D. Introduction to the quagga routing suite. *IEEE Network* 2014; 28(2): 42–48.
15. Wood L, Shell D, Ivancic W, et al. CLEO and VMOC: enabling warfighters to task space payloads. In: MILCOM 2005-2005 IEEE Military Communications Conference. IEEE. ; October 17-20, 2005; Atlantic City, NJ: 3052–3058.

16. Florio MA, Fisher SJ, Mittal S, et al. Internet routing in space: Prospects and challenges of the IRIS JCTD. In: MILCOM 2007-IEEE Military Communications Conference. IEEE. ; October 29-31, 2007; Orlando, FL, USA: 1–6.

17. Wang J, Xu F, Sun F. Benchmarkinng of routing protocols for layered satellite networks. In: . 2 of *The Proceedings of the Multiconference on" Computational Engineering in Systems Applications"*. IEEE. ; October 04-06, 2006; Beijing, China: 1087–1094.

18. Bantan N, Kkan J. Space OSPF: An area hierarchic routing protocol for routers in motion. In: 25th AIAA International Communications Satellite Systems Conference (organized by APSCC). ; April 10-13, 2007; Seoul, South Korea: 3104.

19. Xu M, Xia A, Yang Y, Wang Y, Sang M. Intra-domain routing protocol OSPF+ for integrated terrestrial and space networks. *Journal of Tsinghua University (Science and Technology)* 2017; 57(1): 12–17.

20. Azúa R. dJA, Calveras A, Camps A. Internet of Satellites (IoSat): Analysis of Network Models and Routing Protocol Requirements. *IEEE Access* 2018; 6: 20390-20411. doi: 10.1109/ACCESS.2018.2823983

21. Werner M. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE journal on selected areas in communications* 1997; 15(8): 1636–1648.

22. Gounder VV, Prakash R, Abu-Amara H. Routing in LEO-based satellite networks. In: 1999 IEEE Emerging Technologies Symposium. Wireless Communications and Systems (IEEE Cat. No. 99EX297). IEEE. ; April 12-13, 1999; Richardson, TX, USA: 22–1.

23. Ekici E, Akyildiz IF, Bender MD. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Transactions on networking* 2001; 9(2): 137–147.

24. Akyildiz IF, Ekici E, Bender MD. MLSR: a novel routing algorithm for multilayered satellite IP networks. *IEEE/ACM Transactions on networking* 2002; 10(3): 411–424.

25. Liu J, Luo R, Huang T, Meng C. A load balancing routing strategy for LEO satellite network. *IEEE Access* 2020; 8: 155136–155144.

26. Fraire JA, De Jonckere O, Burleigh SC. Routing in the space internet: A contact graph routing tutorial. *Journal of Network and Computer Applications* 2021; 174: 102884.

27. Burleigh S. Contact Graph Routing. Internet-Draft draft-burleigh-dtnrg-cgr-01, Internet Engineering Task Force; 2010.

28. CCSDS . *Schedule-Aware Bundle Routing, CCSDS 734.3-B-1*. Washington DC, USA: Consultative Committee for Space Data Systems (CCSDS) . 2019.

29. Araniti G, Bezirgiannidis N, Birrane E, et al. Contact graph routing in DTN space networks: overview, enhancements and performance. *IEEE Communications Magazine* 2015; 53(3): 38–46.

30. Burleigh S. Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol. In: 2007 4th IEEE Consumer Communications and Networking Conference. ; January 11-13, 2007; Las Vegas, NV, USA: 222-226

31. Mukherjee J, Ramamurthy B. Communication technologies and architectures for space network and interplanetary internet. *IEEE communications surveys & tutorials* 2012; 15(2): 881–897.

32. Burleigh S, Caini C, Messina JJ, Rodolfi M. Toward a unified routing framework for delay-tolerant networking. In: 2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE). IEEE. ; September 26-28, 2016; Aachen, Germany: 82–86.

33. D'argenio PR, Fraire JA, Hartmanns A, Raverta F. Comparing Statistical and Analytical Routing Approaches for Delay-Tolerant Networks. In: International Conference on Quantitative Evaluation of Systems. Springer. ; 2022: 337–355.

34. Raverta FD, Fraire JA, Madoery PG, Demasi RA, Finochietto JM, D'Argenio PR. Routing in Delay-Tolerant Networks under uncertain contact plans. *Ad Hoc Networks* 2021; 123: 102663.

35. Xu S, Wang XW, Huang M. Software-defined next-generation satellite networks: Architecture, challenges, and solutions. *IEEE Access* 2018; 6: 4027–4041.

36. Bao J, Zhao B, Yu W, Feng Z, Wu C, Gong Z. OpenSAN: a software-defined satellite network architecture. *ACM SIGCOMM Computer Communication Review* 2014; 44(4): 347–348.

37. Shi Y, Cao Y, Liu J, Kato N. A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks. *IEEE Network* 2019; 33(1): 29–35.

38. Vissicchio S, Tilmans O, Vanbever L, Rexford J. Central control over distributed routing. *SIGCOMM Comput. Commun. Rev.* 2015; 45(4): 43–56. doi: 10.1145/2829988.2787497

39. Tanenbaum AS. *Computer networks*. Pearson Education India . 2003.

40. Coltun R, Ferguson D, Moy J, Lindem A. OSPF for IPv6, RFC 5340. 2022.

41. Jaff EK, Pillai P, Hu YF. IP multicast receiver mobility support using PMIPv6 in a global satellite network. *IEEE Communications Magazine* 2015; 53(3): 30–37.

42. Yan H, Li L, Qiao L, et al. Design and Ground Test of an IPv6 Node for Space Network. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC). IEEE. ; December 11-14, 2020; Chengdu, China: 373–377.

43. Zhou D, Shen X, Li K, Feng G, Wang L. Design and Implementation of High-Performance Space Router Based on FPGA. In: 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN). ; June 12-15, 2019: 704-708

44. Janković S, Smiljanić A, Vesović M, et al. High-Capacity FPGA Router for Satellite Backbone Network. *IEEE Transactions on Aerospace and Electronic Systems* 2020; 56(4): 2616-2627. doi: 10.1109/TAES.2019.2951187

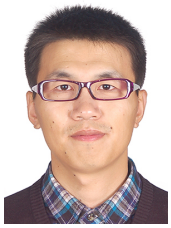
45. LwIP . lwIP - A Lightweight TCP/IP stack - Summary. 2022. https://savannah.nongnu.org/projects/lwip/. Accessed 2022-09-01.

46. CCSDS . *Space Packet Protocol, CCSDS 133.0-B-2*. Washington DC, USA: Consultative Committee for Space Data Systems (CCSDS) . 2020.

47. CCSDS . *AOS Space Data Link Protocol, CCSDS 732.0-B-4*. Washington DC, USA: Consultative Committee for Space Data Systems (CCSDS) . 2021.

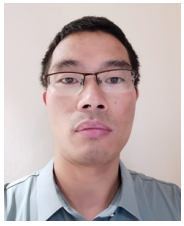
48. H3C . Products & Technology- H3C MSR2600 Router Series- H3C. 2022. https://www.h3c.com/en/Products_Technology/Enterprise_Products/Routers/Multi-Service_Router_Series/H3C_MSR2600/. Accessed 2022-11-27.

AUTHOR BIOGRAPHY



Hongcheng Yan received the Ph.D. degree in flight vehicle design from China Academy of Space Technology, Beijing, China, in 2015. He is now a senior engineer at the Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing, China. His current research interests include network protocol design for space networks.

Liang Qiao received the B.E. degree in computer science and engineering from Beihang University, Beijing, China, and the M.S. degree in computer science from the University of Southern California, Los Angeles, CA, USA. He is currently pursuing a Ph.D. degree in control science and engineering at Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is also an engineer at the Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing, China. His research interests are in spacecraft electronic systems and satellite constellation networks.



Wei Wu received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2010. He is now a senior engineer at the Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing, China. His current research interests include OBDH systems and space network router design.

Juan Fraire is a researcher and professor at INRIA (France) and CONICET-UNC (Argentina), and a guest professor at Saarland University (Germany). Core topics of his interest are near-Earth and deep-space networking and informatics, adding up to more than 70 published papers in international journals and leading conferences. Juan is the founder and chair of the Space-Terrestrial Internetworking Workshop (STINT) and participates in diverse joint projects with space agencies (e.g., NASA, ESA, CONAE) and companies in the space sector (e.g., D3TN, Skyloom).

Dong Zhou received a master's degree in power electronics and electric drive from Beijing Jiaotong University, Beijing, China, in 2008. He is now a senior engineer at the Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing, China. His current research interests include network protocol design for space networks.

Luming Li received the master's degree in radio and mobile communication system from the University of Hertfordshire, Hatfield, Hertfordshire, UK, in 2014. He is now an engineer at the Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing, China. His current research interests include spacecraft electronics and information system design.

Yong Xu received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2011. He is a full professor at the Beijing Institute of Spacecraft System Engineering in Beijing, China. His primary research interests include designing spacecraft electronic information systems, onboard information processing, and image compression.