

Research Article

Cloud-Based Experimental Platform for the Space-Ground Integrated Network

Xiaofeng Wang , **Xinyu Chen** , **Haiyang Ye** , **Yuan Liu** , and **Guizhu Zhang** 

School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China

Correspondence should be addressed to Xiaofeng Wang; wangxf@jiangnan.edu.cn

Received 13 August 2020; Revised 27 September 2020; Accepted 5 October 2020; Published 21 October 2020

Academic Editor: Weizhe Zhang

Copyright © 2020 Xiaofeng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The space-ground integrated network (SGIN) is an important direction of future network development and is expected to play an important role in edge computing for the Internet of Things (IoT). Through integration with an SGIN, IoT applications can provide services with long-distance and wide-coverage features. However, SGINs are typical large-scale and time-varying networks for which new network technologies, protocols, and applications must be rigorously evaluated and validated. Therefore, a reliable experimental platform is necessary for SGINs. This paper presents a cloud-based experimental platform for the SGIN context named SGIN-Stack. First, the architecture of SGIN-Stack, which combines the Systems Tool Kit (STK) and OpenStack, is described. Based on this architecture, a seamless linkage between OpenStack and STK is achieved to realize synchronous, dynamic, and real-time network emulation for an SGIN, and the dynamic differential compensation technology and a random number generation algorithm are applied to improve the emulation accuracy for satellite links. Finally, an emulation scenario is constructed that includes six space-based backbone nodes, sixty-six space-based access nodes, and a ground station. Based on this emulation scenario, experiments concerning the satellite link delays, bit error ratio (BER), and throughput are carried out to prove the high fidelity of our SGIN-Stack platform. Emulation experiments involving satellite orbital maneuvers and attitude adjustments show that SGIN-Stack can be used for dynamic and real-time SGIN emulation.

1. Introduction

Space-ground integrated networks (SGINs) [1, 2], space-terrestrial integrated networks (STINs) [3], and space-air-ground integrated networks (SAGINs) [4] are highly heterogeneous complex networks that include a space-based backbone network, a space-based access network, and a ground-based backbone network. The goal in proposing an SGIN is to construct an integrated, expandable, cross-supported, and reconfigurable space-ground information system for applications.

SGINs have a wide range of applications in many fields, especially in the field of edge computing for the Internet of Things (IoT). For instance, Wang et al. [5] proposed to transform the traditional satellite into a space edge computing node, which takes less time and consumes less energy; Chen et al. [6] proposed a SAGIN edge/cloud computing architecture for offloading the computation-intensive applications. With SGINs, edge computing for IoT has transcended phys-

ical limitations and realized the “Internet of Everything” in the real sense [7, 8]. For example, in [9], the authors introduced related research results on SGINs for the construction of maritime networks to enable the development of IoT services in the sea environment. A new space-air-ground (SAG) IoT network paradigm was proposed in [10], where the authors introduced related key technologies to meet the needs of a variety of IoT fields, such as aviation and smart city construction.

Since SGINs are large-scale complex networks with heterogeneity and time-variability features, the integration of SGIN technology into the IoT also presents new technical requirements, such as requirements on resource management, traffic distribution, and protocol optimization [10, 11]; therefore, many new technologies and protocols continue to emerge. For instance, Zhao et al. [12] proposed an identity-based authentication scheme called MASIT for SAG IOT (SIOT). Liu et al. [13] proposed a multibeam satellite industrial IoT (IIOT) based on nonorthogonal multiple access (NOMA)

to achieve wide-area coverage and improved service performance. However, any such new technologies, protocols, and IoT application services must be rigorously evaluated and validated before an SGIN is actually deployed [11]; therefore, a reliable experimental platform for SGIN is necessary [14].

As cloud platforms and virtualization technologies have evolved to be capable of supporting scalable and high-fidelity virtual networks, network emulation based on cloud platforms and virtualization has become an emerging research trend [15, 16]. For example, based on a commodity cloud platform, a network protocol emulation platform called CloudNet was implemented in [17]; by means of a lightweight virtualization technique called LXC containers, CloudNet is scalable to thousand-node emulated networks. At present, the OpenStack cloud platform [18], which incorporates state-of-the-art technologies, such as software-defined networking (SDN) [19], full virtualization (kernel virtual machine, KVM), and lightweight virtualization (Docker), has become the foremost mainstream cloud platform in the industry, with features such as high efficiency and stability; consequently, it is considered a good choice for developing emulation platforms [15, 20].

An SGIN is a typical large-scale and time-varying network with variable delay and bit error ratio (BER), unstable end-to-end paths between satellites, and satellite maneuvering capabilities [4, 15, 21]. Consequently, traditional cloud platforms cannot meet the emulation requirements for an SGIN [15]. Although an OpenStack-based network emulation platform named EmuStack was proposed in [15] for the real-time network emulation of delay-tolerant networks (DTN), including the time-varying emulation of satellite network characteristics (e.g., bandwidth, delay, and BER), EmuStack ignores the maneuverability of satellites, which necessitates the real-time emulation of dynamical changes in the emulated SGIN (e.g., dynamic satellite management strategies [21]). To address this problem, an emulation platform for SGINs named *SGIN-Stack* is proposed in this paper.

SGIN-Stack is based on OpenStack, and through the incorporation of technologies, such as SDN, KVM, and Docker, SGIN-Stack achieves high fidelity in emulating the nodes of an SGIN and high throughput and flexibility in emulating the SGIN's links. To ensure emulation fidelity for links, the emulation source data of SGIN-Stack, which include the orbit data of the satellites, an atmospheric rain attenuation model, and communication calculation for the satellite links, are computed using the Systems Tool Kit (STK) [22], which is a leading analysis software package developed by Analytical Graphics, Inc. (AGI) of the USA that can be used to dynamically design and simulate multilayer satellite networks [23]. SGIN-Stack implements a seamless and high-efficiency linkage between OpenStack and STK to blend the advantages of both. Through this linkage between OpenStack and STK, SGIN-Stack can realize dynamic, accurate, and real-time SGIN emulation. The emulation network generated by SGIN-Stack on OpenStack is called the SGIN emulation network. The SGIN emulation network can be dynamically driven by the accurate data computed by STK. Besides, for any dynamic changes in the SGIN emulation network, such as satellite orbital maneuvers and attitude adjust-

ments, STK can be further leveraged to recompute the corresponding data in real-time. Furthermore, SGIN-Stack integrates not only the Linux Traffic Control (TC) tools [24] and SDN techniques (similar to EmuStack [15]) but also the dynamic differential compensation technique and a random number generation algorithm by means of OpenStack. With these techniques, SGIN-Stack can more accurately emulate variable delay and BER characteristics as well as unstable end-to-end paths between satellites.

Overall, the main advantages of SGIN-Stack are as follows:

- (1) Through the seamless and high-efficiency linkage between OpenStack and STK, SGIN-Stack can realize not only the realistic emulation of SGINs but also the real-time emulation of dynamic changes in SGINs, such as unstable paths between satellites and satellite maneuvering
- (2) By virtue of the proposed dynamic differential compensation technique and random number generation algorithm, SGIN-Stack can emulate satellite links with improved accuracy
- (3) SGIN-Stack integrates two existing virtualization methods, that is, KVM and Docker, to leverage the unique benefits of each. Space-based backbone satellite nodes and gateway station nodes are emulated by KVM, which can meet the high throughput requirements of space-based backbone satellite links. Space-based access satellite nodes are emulated by Docker, which can achieve fast deployment and efficient utilization of physical resources

This paper is organized as follows. First, in Section 2, previous works related to SGIN emulation are introduced. Second, the architecture of SGIN-Stack is described in Section 3. Several core implementations for the accurate, dynamic, and real-time emulation of SGINs are described in detail in Section 4. In Section 5, an SGIN scenario is constructed with SGIN-Stack, and the corresponding experimental results are discussed. Finally, we conclude this paper and describe directions of future research in Section 6.

2. Related Work

STK [22] is a commercially available software package for various satellite performance analyses, including the validation of new theoretical models [25], the detailed animation of spacecraft simulations [26], the calculation of satellite coverage information [27, 28], the design of network topologies [29], and the evaluation of satellite maneuvers [30]. STK has strong modeling and analysis capabilities for satellite networks but lacks a network performance analysis mechanism [31]; hence, it cannot effectively emulate a satellite network in regard to network protocols, dynamic throughput, and secure communication methods [28].

A method of designing satellite constellation topologies in STK and exporting them to the ns-2 discrete event simulator was proposed in [29] for evaluating the network

performance of satellite networks. In [32], STK was combined with the QualNet network simulator to model communication with Internet protocols. The methods used in [29, 32] combine the advantages of both STK and network simulators and thus are more promising for evaluating SGINs. However, because network simulators have difficulties emulating complicated environments [16], providing relevant and comprehensive details [15], and generating realistic data flows [14], the predominant trend of evaluation technology has shifted from simulations to experimental testbeds [16].

On the basis of SDN and virtualization technologies, a scalable, reconfigurable, and flexible network emulation platform for space information networks was proposed in [14]. This platform can emulate the physical characteristics of satellite nodes (such as the CPU frequency, memory, and operating system) and the characteristics of the links between nodes (such as the traffic rate, BER, and delay), where the source data for the link characteristics are derived from STK. Our work differs from this platform in that we attempt to emulate not only these variable characteristics but also the maneuvering of satellites.

An OpenStack-based DTN emulation platform named EmuStack was proposed in [15], in which lightweight virtualization (Docker) is used to support a large-scale topology (up to hundreds of nodes). EmuStack integrates TC and STK with OpenStack to emulate the characteristics of satellite links. In addition to using TC and STK, our work attempts to further improve the emulation fidelity for satellite links through methods such as dynamic differential compensation and a random number generation algorithm. Moreover, our work implements a more seamless linkage between OpenStack and STK than that in EmuStack in order to emulate dynamic changes in SGINs.

TUNIE, a testbed for evaluating DTNs, was designed in [16]. In this testbed, virtualization and SDN technologies are exploited to implement a realistic and reliable DTN environment, and time-varying bit rates, error rates, and transmission delays are implemented by controlling data transmission. TUNIE requires the further integration of abundant models for the emulation of an SGIN, which is a highly heterogeneous and complex network.

3. Architecture of SGIN-Stack

Figure 1 illustrates the overall architecture of SGIN-Stack, which is based on the OpenStack cloud architecture. The OpenStack cloud architecture mainly includes a control node, a network node, and multiple compute nodes. The control node is responsible for managing OpenStack, the network node provides L2 and L3 network services for OpenStack, and the compute node is responsible for the operation of virtual machines. The architecture of SGIN-Stack includes an STK simulation node, a network node, and several physical emulation nodes. The network node corresponds to the OpenStack network node and provides network services for SGIN-Stack. The various modules of the physical emulation nodes are integrated into all OpenStack compute nodes. The STK simulation node is built on a host.

In the design of the overall platform architecture, several ideas originating from SDN are adopted. The STK simulation node acts as the application plane to generate simulation information source data. According to the simulation information source data, the physical emulation node acts as the control plane to manage the traffic of the SGIN emulation network. The various virtual network devices in the SGIN emulation network in OpenStack form the data plane.

The architecture of SGIN-Stack is based on an emulation concept relying on multiscale integration combining full and lightweight virtualization technologies. Space-based backbone nodes and gateway stations have high throughput and are emulated using full virtualization technology. Space-based access nodes with lower throughput requirements are emulated using lightweight virtualization technology. The integration of the two virtualization technologies not only ensures the emulation performance but also realizes the efficient utilization of physical resources.

The three main components of the architecture depicted in Figure 1 are described in detail in sections 3.1, 3.2, and 3.3.

3.1. STK Simulation Node. The STK simulation node mainly consists of STK11 and MATLAB R2013a. The STKX module is designed to provide two-way communication between STK and MATLAB. STK is used to simulate the satellite nodes and ground nodes. The network simulated in STK is called the SGIN simulation network. Each MATLAB module obtains the relevant data from STK via the STKX module and either proceeds with further processing or transmits relevant commands to STK to control the SGIN simulation network in STK in reverse. The functions of each MATLAB module are described as follows:

The topology monitor module is responsible for monitoring the dynamic changes in network topology in STK. It monitors the list of objects in STK by means of the “stkObjNames” command, compares the results against the existing list of objects to monitor the dynamic behaviors of object addition and deletion in STK, and sends the information on any such dynamic changes in STK to the topology linkage control module in each physical emulation node. The topology linkage control module changes the topology of the SGIN emulation network in accordance with the information from the topology monitor module. Thus, the topology monitor module and the topology linkage control module are linked to realize the real-time and dynamic emulation of the network topology.

The main task of the STK report acquisition module is to obtain reports of data on link characteristics. Its main function is to extract a report of link performance data from STK via the “stkAccReport” command and process these data to generate the link characteristics, such as delay and packet loss rate. These link characteristics are then sent to the link characteristics linkage control module via the north-bound interface of the corresponding physical emulation node in order to set the link characteristics in the SGIN emulation network. Thus, the STK report acquisition module and the link characteristics linkage module are linked to realize the real-time and dynamic emulation of the link characteristics.

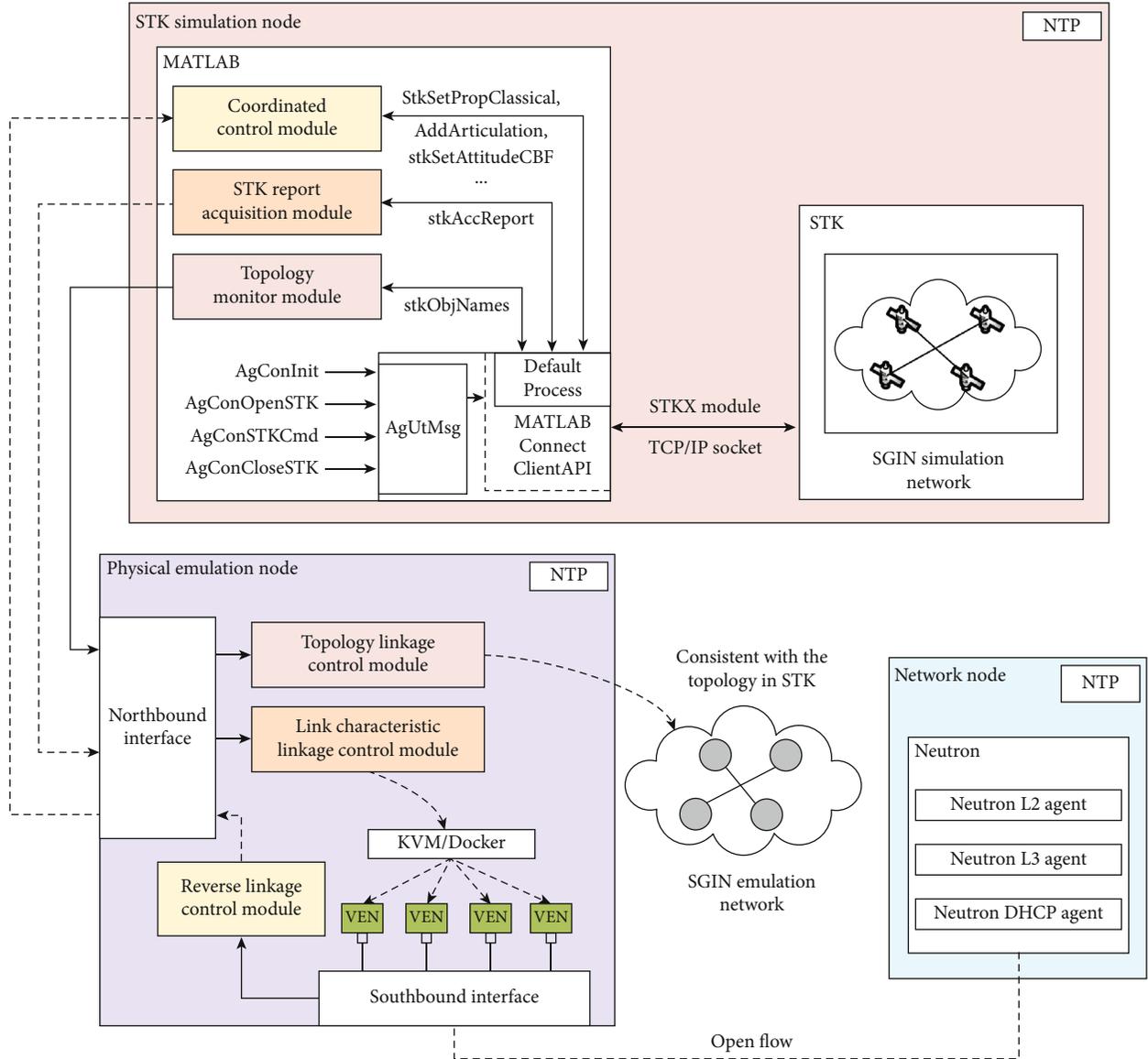


FIGURE 1: Architecture design of SGIN-Stack.

In accordance with the instructions from the reverse linkage control module in the physical emulation node, the coordinated control module controls the orbital maneuvers, attitude, and solar panels of the corresponding satellite in STK through commands such as “`stkSetPropClassical`,” “`AddArticulation`,” and “`stkSetAttitudeCBF`.” The linkage between the coordinated control module and the reverse linkage control module enables a more flexible dynamic reconfiguration of the SGIN emulation network in SGIN-Stack.

3.2. Physical Emulation Node. Each module of a physical emulation node is mainly based on OpenStack. All nodes of SGIN-Stack adopt the Network Time Protocol (NTP), and the control node of OpenStack is used as a unified time axis, thus ensuring the real-time operation and synchronization of each node in SGIN-Stack. The emulation nodes corresponding to satellites and ground stations in the SGIN emulation

network are called virtual satellite nodes and virtual ground station nodes. The links between virtual satellite nodes are called virtual satellite links.

The physical emulation node establishes communication with the STK simulation node and network node through the northbound interface and southbound interface, respectively. The northbound interface is the NBI (northbound interface) in the SDN architecture, which can realize the communication between the STK simulation node and the physical emulation node. The southbound interface is the CDPI (control-data-plane interface) in the SDN architecture, which can realize communication between the physical emulation node and the network node, and then realize communication between the physical emulation node and virtual network devices of the SGIN emulation network.

The function of each module is described as follows:

The topology linkage control module is responsible for maintaining consistency with the topology of the SGIN

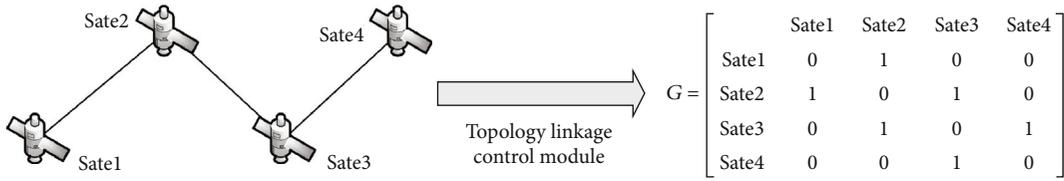


FIGURE 2: A simple satellite network topology.

simulation network in STK. For instance, if an additional satellite is added in STK, this module can perceive such an event, send a creation request to the Nova scheduler service of OpenStack, send a remote procedure call request to the Nova compute service, obtain an image of the created node from the Glance service, and start the node on the compute node's hypervisor; afterward, the network service Neutron will assign a hardware device, such as a network card, and an Internet Protocol (IP) address to the virtual satellite node.

The link characteristic linkage control module dynamically controls the link characteristics of the corresponding virtual satellite node, such as the delay, packet loss rate, and bandwidth, in accordance with the data sent by the STK report acquisition module in the STK simulation node.

The reverse linkage control module mainly sends corresponding commands to the coordinated control module of the STK simulation node in accordance with the commands issued by the virtual satellite node to realize the reverse control of the satellite in the SGIN simulation network in STK, such as the control of orbital maneuvers, the satellite attitude, and the solar panels.

3.3. Network Node. The network node corresponds to the network node in OpenStack. The Neutron service is integrated on the network node, which provides network services for OpenStack. The design goal of the Neutron service is "Networking as a Service," which is to provide network support for the OpenStack platform and realize software-based network resource management. Neutron services include the Neutron L2 agent, Neutron L3 agent, and Neutron Dynamic Host Configuration Protocol (DHCP) agent.

As Neutron's core agent, the Neutron L2 agent is responsible for providing layer 2 network functions, including management of network, subnet, and port resources. Commonly used Neutron L2 agents include Linux Bridge and OpenvSwitch (OVS). SGIN-Stack uses OVS, which supports the OpenFlow protocol, to provide the Layer 2 connectivity to the virtual satellite nodes and virtual ground station nodes.

The Neutron L3 agent is a service agent of Neutron, which can provide routing services of Layer 3. This agent uses Linux IP stack, route, and iptables to implement network traffic between virtual machines in different networks in the intranet, as well as the routing and forwarding of network traffic between virtual machines and the external network.

The Neutron DHCP agent is a service agent of Neutron, which is responsible for providing DHCP services. This agent dynamically allocates IP addresses to virtual machines through Dnsmasq. Dnsmasq is an open-source program that provides DHCP services and domain name system (DNS) caching.

4. Implementation of SGIN-Stack

This section describes the core methods and modules of SGIN-Stack in detail and shows how the emulation platform can dynamically modify the networks topology and satellite link characteristics. Sections 4.1, 4.2, and 4.3 explain the principles of the three core modules of physical emulation nodes.

4.1. Topology Linkage Control. The topology linkage control module is responsible for keeping the topology of the SGIN emulation network in OpenStack consistent with the topology of the SGIN simulation network in STK. MATLAB acts as a listener, receiving the topology of the SGIN simulation network from STK and directing OpenStack to alter the topology of the SGIN emulation network when the topology changes in STK. After receiving the connection matrix from the topology monitor module in MATLAB, the topology linkage control module dynamically changes the topology during emulation. Figure 2 shows a simple topology and its corresponding connection matrix. A connection between two nodes is represented by a value of one in the connection matrix; if there is no connection, the corresponding matrix element is zero otherwise.

The topology linkage control module can dynamically control the topology of the SGIN emulation network by configuring the OVS flow tables of OpenStack. The module acquires information on each virtual satellite node that refers to the device tap in OVS, including the media access control (MAC) address, the virtual local area network (VLAN) label, and the IP address. A device tap is a kind of virtual Ethernet device in the second layer of the Open Systems Interconnection (OSI) model, which is implemented by software in the host kernel. When a satellite link is switched from the normal communication state to the interrupt state, the virtual link between the device tap and OVS can be disconnected by the "del-port" interface of OpenStack. The corresponding VLAN label and flow table rules are deleted, and the link cannot transmit any data. When a link is switched from the interrupt state to the normal communication state, in addition to reestablishing the link through the "add-port" interface of OpenStack, a flow table rule that matches the Address Resolution Protocol (ARP) and IP packets is created by the topology linkage control module.

4.2. Link Characteristic Linkage Control

4.2.1. Controlling Bandwidth and Delay. The delay of a satellite link varies dynamically because the length of the link is constantly changing, and the resulting changes in delay lead to dynamic changes in throughput. In SGIN-Stack, there are two locations where the bandwidth and delay can be

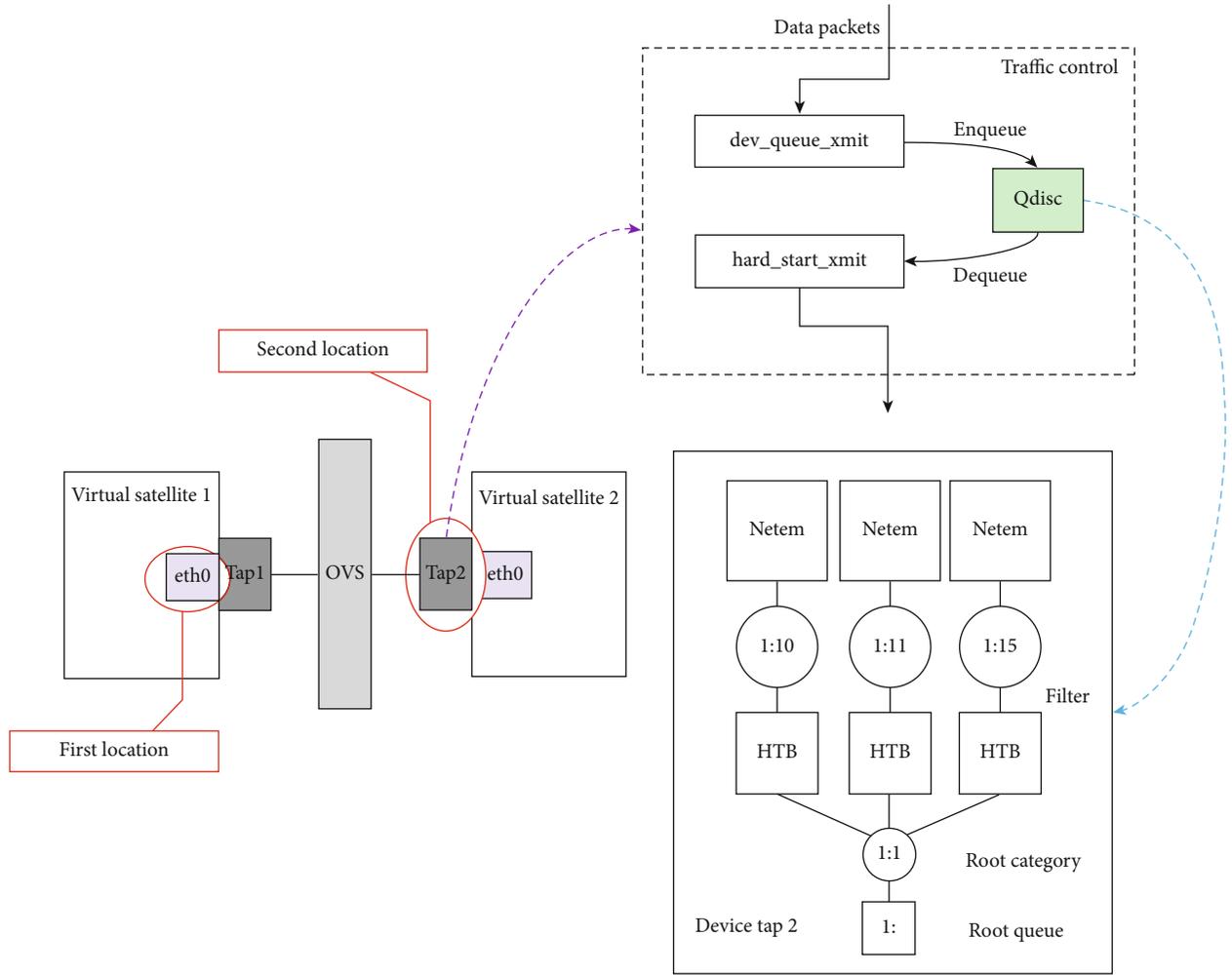


FIGURE 3: OVS-based virtual satellite 1 and virtual satellite 2 link.

set: the first is the network interface card (NIC) driver in the virtual nodes, and the second is the device tap in OVS. The link characteristic linkage control module uses the second location because it is more efficient to control the device tap to set the bandwidth and delay. If the first location were to be chosen, the link characteristic linkage control module would require more run time because it would need to send commands to the virtual nodes to limit the rate, and this additional time would influence delay emulation. Note that satellite links are directional, which means that the emulation of link A-B is different from that of link B-A. Figure 3 depicts an OVS-based virtual link from “virtual satellite 1” to “virtual satellite 2.” The device tap of “virtual satellite 1” will be controlled by the link characteristic linkage control module because TC in Linux can effectively control only the outflow of traffic.

The link characteristic linkage control module uses TC in Linux to add Hierarchical Token Bucket (HTB) rules to the device tap in OVS to limit the bandwidth of the virtual satellite link. The queuing discipline (QDisc) is a queuing rule located in the network device of the host. It is mainly divided into unclassifiable QDisc and classifiable QDisc. After the data

packet enters the network device, it will be shaped according to the QDisc configured for the network device. HTB is a classifiable QDisc that implements flow control in cooperation with TC in Linux. Packets needing to be sent are in the send buffer in the kernel space in “virtual satellite 1”; packets sent to “virtual satellite 2” will first be filtered by iptables. When packets reach device tap 2 in OVS, the function “dev_queue_xmit” in the kernel is used to release the packets, enqueueing them in the QDisc buffer to shape them. Then, the function “hard_start_xmit” is used to send packets to the NIC driver. The link characteristic linkage control module can call the network emulator (Netem) module of the Linux kernel to dynamically limit the delay of the device tap. Netem is a network simulation module provided by the Linux kernel, which can simulate complex Internet transmission performance in a local area network, such as bandwidth, delay, and packet loss. The entire queue of the device tap has a structure similar to a tree, as shown in Figure 3. The data packets first pass through the root queue of the device tap and then enter the root category. A category can have multiple subcategories. The link characteristic linkage control module adds HTB queue rules to limit bandwidth, similar to Netem rules.

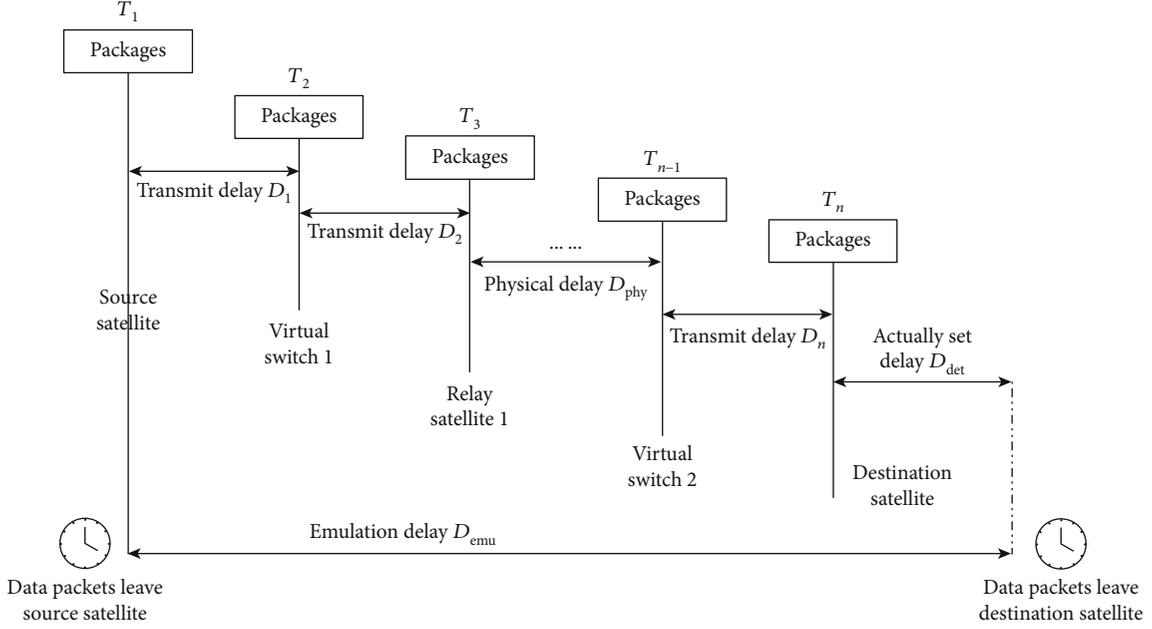


FIGURE 4: Principle of satellite link delay compensation.

4.2.2. Delay Compensation. The satellite links in SGIN-Stack include intrahost virtual links and crosshost virtual links. For an intrahost virtual link, the two virtual satellites connected by the link are hosted in the same physical emulation node; for a crosshost virtual link, the two virtual satellites are hosted in different physical emulation nodes. Either the transmission delay within the physical emulation node or the transmission delay of the underlying physical link that connects the two physical emulation nodes will reduce the fidelity of emulation of the satellite link delay. In the case of a complex underlying network, the error caused by the underlying physical link delay is often greater than the error caused by the transmission delay within a physical emulation node. To address the above two link transmission delays, a dynamically configurable differential delay compensation method is designed to achieve high-fidelity emulation of the satellite link delay.

In Figure 4, T_1 is the time at which the source satellite sends a message to the destination satellite; T_2 is the time at which the data packets arrive at the layer 2 switch on a physical node or arrive at a relay satellite; T_3 is the time at which the data packets arrive at a relay satellite; T_n is the time at which the data message reaches the destination satellite; $D_1, D_2, D_3, \dots, D_{n-1}$ are the transmission delays of the intermediate links; D_{phy} denotes the total transmission delay of the underlying physical link; D_{emu} is the actual emulation delay obtained by the traditional TC tool; D_{set} is the set delay; D_{error} is defined as the total virtual link transmission delay. The specific relations are defined as shown in equation (1):

$$\begin{cases} D_i = T_{i+1} - T_i, i \in [1, n) \\ D_{emu} = T_n - T_1 + D_{set} \\ D_{error} = \sum_{i=1}^{n-1} D_i + D_{phy} = T_n - T_1 \end{cases}, \quad (1)$$

$$D_{set} = D_{emu} - D_{error} = D_{emu} - (T_n - T_1). \quad (2)$$

The dynamic differential compensation technology will compensate for the transmission delay in the intermediate link, that is, the time of compensating D_{emu} , so that the final emulation delay is closer to the set delay D_{set} , as shown in equation (2). It can be seen from equation (2) that differential compensation of the transmission delay can be realized only when the start time T_1 and the arrival time T_n are known. The dynamic differential compensation technology completes differential compensation by setting the netfilter queue (NFQUEUE) of iptables. NFQUEUE is a target in iptables rules; it can hand over the data packet to the user space for further processing. The specific implementation method is as follows: A message sending queue and a message receiving queue were set on the source satellite and destination satellites' network card, respectively. When the data packet is sent to the network card of the source satellite, the packet sending queue automatically matches the data packet that matches the source IP as the source satellite IP and the destination IP as the destination satellite IP. The qualified data packet is sent into the NFQUEUE sending queue, and then, the queue obtains the current timestamp T_1 , intercepts the data packet in the user mode, and encapsulates T_1 at the tail of the data part of the data packet. When the data packet reaches the network card of the destination satellite, the message receiving queue automatically matches the data packet that matches the source IP as the source satellite IP and the destination IP as the destination satellite IP. The qualified data packet enters the NFQUEUE receiving queue, and then, the queue will parse the data part of the data packet in the user mode to obtain T_1 . Finally, the queue obtains the current timestamp T_n and dynamically compensates for the link transmission delay to achieve higher satellite delay emulation fidelity.

Input: $lower_limit$, $upper_limit$

$lower_limit$ is the lower limit on the random number to be generated by the link characteristics linkage control module; $upper_limit$ is the upper limit on the random number to be generated by the link characteristics linkage control module

Output: $Random$

```

1 Initialize the algorithm;
2  $Threshold \leftarrow$  A constant;
3  $Index \leftarrow$  0;
4  $Random\_lib \leftarrow$  An empty array;
5 for  $numa\_node$  in  $(0, Threshold)$  do
6    $Random\_num \leftarrow$  LCG( $lower\_limit$ ,  $upper\_limit$ );
7   Append  $Random\_num$  to  $Random\_lib$ ;
8  $Index \leftarrow$  INT(LCG(0,  $Threshold$ ));
9  $Random \leftarrow$  Value in  $Random\_lib$  whose index in the array is  $Index$ ;
10 return  $Random$ ;

```

ALGORITHM 1: Random number generation algorithm.

4.2.3. *Controlling the BER.* In our emulation platform, the BER is emulated by discarding packets. When a packet is transmitted by a sender, the link characteristic linkage control module generates a random number; this random number will be compared against the BER of the current virtual satellite link. If the random number is within the BER range, then it is assumed that a bit error has occurred in the packet, and that the packet should be discarded; as a result, the packet will not enter the buffer queue. Otherwise, the packet enters the buffer queue. Therefore, the key to accurately controlling the BER is to use a suitable random number generation algorithm.

Computers generate only pseudorandom numbers; therefore, random numbers generated by a computer exhibit a certain correlation. Algorithm 1 describes the detailed steps of how a random number is generated in SGIN-Stack. This algorithm improves the linear congruential generator (LCG) to reduce the correlation of the generated random numbers.

First, the LCG algorithm is initialized, and two parameters named $lower_limit$ and $upper_limit$ should be provided. Specifically, $lower_limit$ is the lower limit on the random number to be generated by the link characteristic linkage control module, and $upper_limit$ is the upper limit on this random number. Since the range of BER is between 0 and 1, the value of $lower_limit$ is 0, and the value of $upper_limit$ is 1. Then, we set a constant named $Threshold$ to define the size of the alternative random number library ($Random_lib$). The $Threshold$ cannot be too large or too small. If the $Threshold$ is too large, the time required to generate the random number will be too long, and if it is too small, the correlation of the generated random numbers will be too high. Tests show that the random numbers generated by Algorithm 1 with a $Threshold$ of 10 exhibit good randomness and are produced within an acceptable time. Then, LCG ($lower_limit$, $upper_limit$) will generate a random number between $lower_limit$ and $upper_limit$, which will be added to the alternative random number library. This step will be iterated until the alternative random number library is filled. Finally, the LCG algorithm is used again to generate a random integer between 0 and $Threshold$, which is assigned to $Index$. Finally,

the random number whose index value is $Index$ in the alternative random number library is selected as the random number value to be output.

The theoretical BER value of the virtual link is obtained by the STK simulation node and then sent to the link characteristic linkage control module. When a data packet is sent by a sender, the link characteristic linkage control module will call the random number generation algorithm to generate a random number. If the random number is within the theoretical BER range, the data packet will be discarded; otherwise, the data packet will enter the buffer queue. The link characteristic linkage control module realizes the emulation of BER through this algorithm.

4.3. *Reverse Linkage Control.* Emulations of satellite orbital maneuvers, satellite solar panels, satellite attitudes, and space-borne cameras are critical for several SGIN scenarios, such as space-based access networks. While OpenStack does not provide any services pertaining to the above considerations, STK can fully compensate for this deficiency. The orbit of a satellite changes as required by its control system, which causes it to perform orbital maneuvers. Transfers of satellites from a high orbit to a low orbit or vice versa and adjustments of satellite surveillance areas based on users' needs require orbital maneuvers. An orbital maneuver is achieved by adjusting six parameters of the satellite's orbit, namely, the semimajor axis, eccentricity, perigee central angle, right ascension of the ascending node (RAAN), orbital inclination, and near-point angle. STK's attitude control module can solve the problem of controlling the attitude of an object in the three-dimensional graphics window in STK. With this attitude control module, a three-dimensional object in STK is no longer a constant model but an object with multiple functions, such as rotation over time. Meanwhile, STK provides standard pose definitions and can also use externally input gesture files to enable a variety of analytical tools for calculating the effects of attitude changes on other parameters.

Figure 5 shows the specific process of the two-way linkage control in SGIN-Stack. First, the STK simulation node completes the initialization process of the SGIN simulation

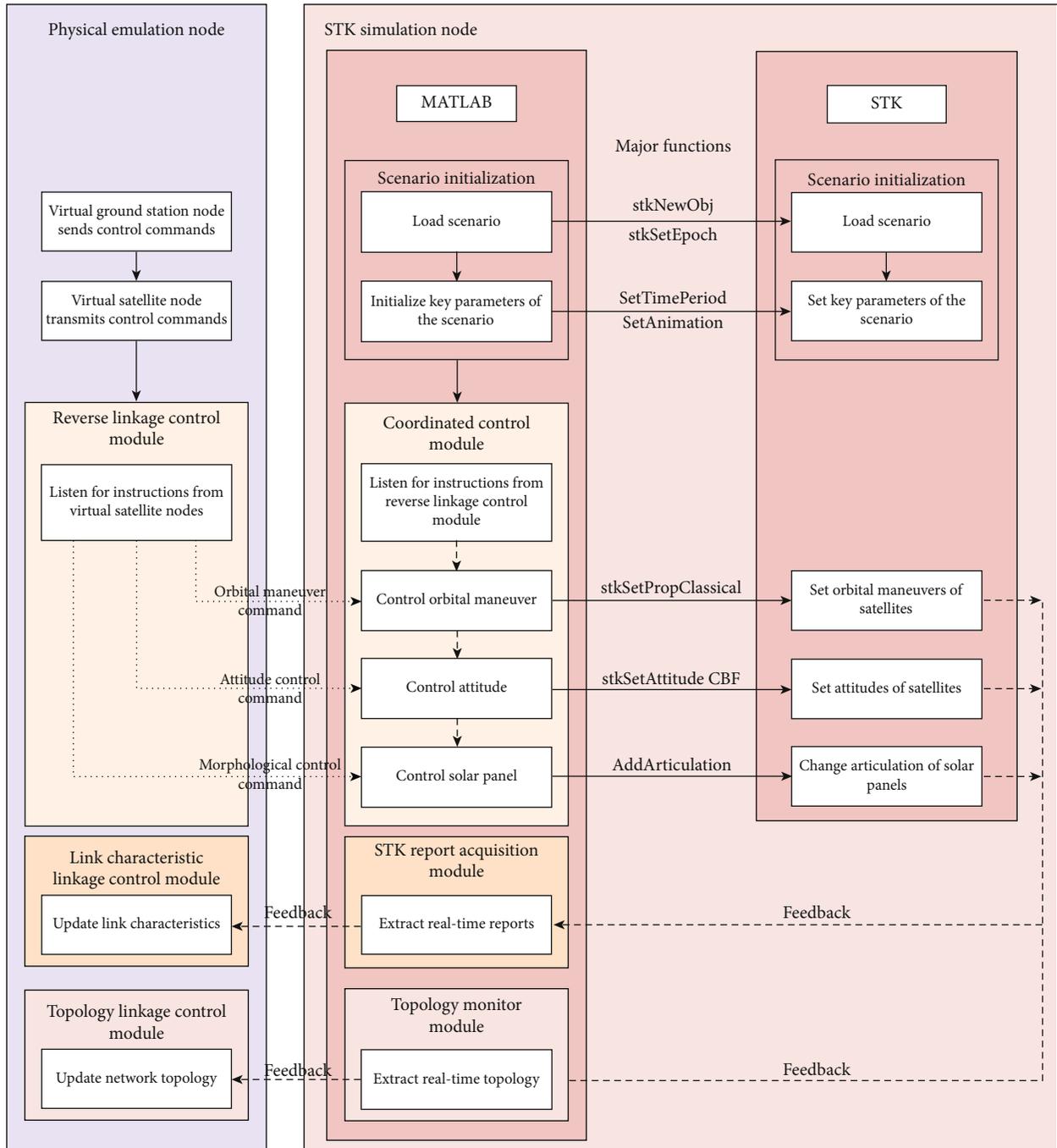


FIGURE 5: Two-way dynamic linkage between the SGIN simulation network and the SGIN emulation network.

network, that is, the scenario initialization part in Figure 5, including the two processes of loading the scenario and initializing the key parameters of the scenario. When the virtual ground station node in the physical emulation node sends a control command to the virtual satellite node, the reverse linkage control module can capture this behavior and send the control command to the corresponding submodule in the coordinated control module of the STK simulation node, according to the type of control command. The control commands include orbital maneuver control commands, attitude control commands, and morphological control commands.

The coordinated control module contains submodules corresponding to these three types of control commands. The submodules will call related functions according to the control commands to drive the satellites in the SGIN simulation network to perform related operations. The reverse control process may cause changes in the topology or link characteristics of the STK simulation network, and these changes will be fed back to the STK report acquisition module and topology monitor module in real-time and then to the link characteristic linkage control module and topology linkage control module in the physical emulation node. Finally, the topology

TABLE 1: Parameters of the satellite constellations.

6*parameters	Space-based backbone network	Space-based access network
Semimajor axis (km)	42166.3	7158.14
Eccentricity ratio	0	0
Inclination (degree)	0	86.4
Argument of perigee (degree)	0	0
RAAN (degree)	20, 80, 140, 200, 260, 320	30, 60, 90, 120, 150, 0
Mean anomaly (degree)	0	Odd orbital planes: 0, 32.72, 65.45, 98.18, 130.91, 163.63, 196.36, 229.09, 261.82, 294.54, 327.27 Even orbital planes: 16.36, 49.09, 81.82, 114.54, 147.27, 180, 212.73, 245.45, 278.18, 310.91, 343.63

and link characteristics in the SGIN emulation network will be updated in real-time.

Let us take orbital maneuver control as an example. The process is as follows: first, the reverse linkage control module receives a parameter adjustment command. Then, the coordinated control module receives the corresponding command from the reverse linkage control module and uses the function “stkSetPropClassical” to control the orbital maneuver of the relevant satellite in the SGIN simulation network. As a result, SGIN-Stack updates the SGIN emulation network, including the network topology and virtual satellite link characteristics, based on the new SGIN simulation network.

5. Experiments

In this section, a typical SGIN topology is emulated using our SGIN-Stack platform, and two experiments are performed. In the first experiment, the intermittence, bandwidth, delay, and BER of all links in an SGIN are emulated in real-time to show the high fidelity of SGIN-Stack. In the second experiment, a simulation scenario considering satellite orbital maneuvers, attitudes, and solar panels is constructed to show that SGIN-Stack can perform the dynamic and real-time network emulation for an SGIN.

5.1. A Typical SGIN Topology. We first describe a typical SGIN topology that can meet the communication needs of terrestrial users anywhere in the world. The topology consists of a space-based backbone network, a space-based access network, and a ground network. Terrestrial users can connect to the SGIN to make satellite phone calls and use navigation services from anywhere via the space-based access network. The space-based backbone network is responsible for long-distance communication and information exchange.

According to the design requirements above, the design objectives of the space-based backbone network and the space-based access network are as follows:

- (1) The space-based backbone network should be designed to cover the entire world and, in particular, be able to continuously cover China
- (2) The space-based access network should be able to cover China and the surrounding areas and should

be easy for terrestrial users to access from anywhere, including high-latitude areas

In accordance with the design goals, our SGIN topology consists of sixty-six low earth orbit (LEO) satellites, six geostationary orbit (GEO) satellites, and a ground station. The space-based backbone network is formed by the six GEO satellites, all of which are in geostationary orbit over the equator, meaning that the eccentricity, inclination, and mean anomaly of these GEO satellites are all 0. These six GEO satellites are evenly distributed in the orbit plane, and the longitudes of the ascending nodes are 20°, 80°, 140°, 200°, 260°, and 320°. In this way, the space-based backbone network can cover the whole world (except for the polar regions). The space-based access network is composed of the sixty-six LEO satellites, which are distributed in six orbital planes. For their orbits, we refer to the parameters of the Iridium constellation, which can achieve global coverage. The inclination is set to 86.4, and the longitudes of the ascending nodes are 30°, 60°, 90°, 120°, 150°, and 0°. There are eleven LEO satellites in each orbit, which are evenly distributed in the orbital plane, and the difference in mean anomaly between adjacent satellites is 32.72°. The specific parameters of satellites forming the space-based access network and the space-based backbone network are shown in Table 1. Figure 6 displays both overall and partial maps for this SGIN scenario.

The dilution of precision (DOP) is commonly used in the satellite navigation to quantify the effects of the precision of positional measurement. The geometric DOP (GDOP) is used to determine how errors in positional measurements will affect a satellite’s final estimated state. The ideal value of the GDOP is 1; this value indicates a high confidence in precision. STK has a built-in DOP figure of merit (FOM) to measure the geometry of satellites and the quality of coverage. There are a number of DOPs to choose from, including the GDOP, which measures the DOP for the entire navigation solution (with positional and clock-related components).

Due to the number of experiments performed and the limited available space, we present only several representative figures to analyze the GDOP of the space-based access network. In Figure 7, there are many visible satellites and little variation in the GDOP with longitude at Lat80. In Figure 7(a), the GDOP at Lat80Lon10 varies from 1.413 to

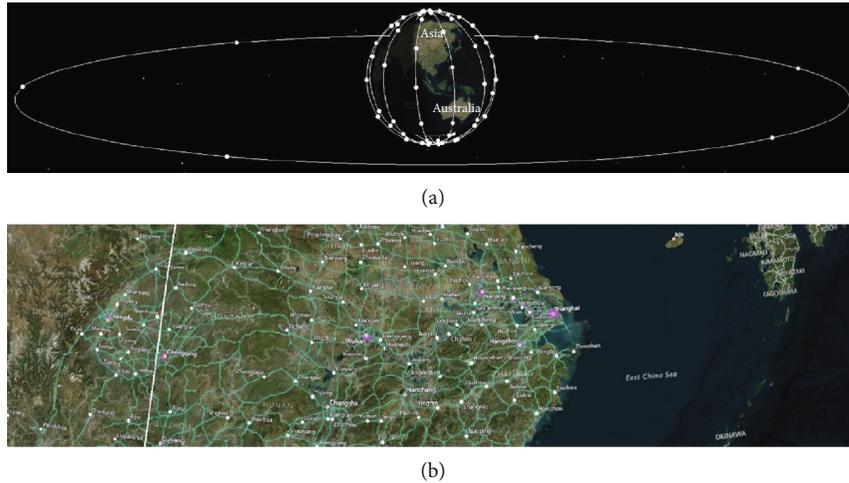


FIGURE 6: SGIN scenario in STK. (a) Global scenario. (b) Partial scenario.

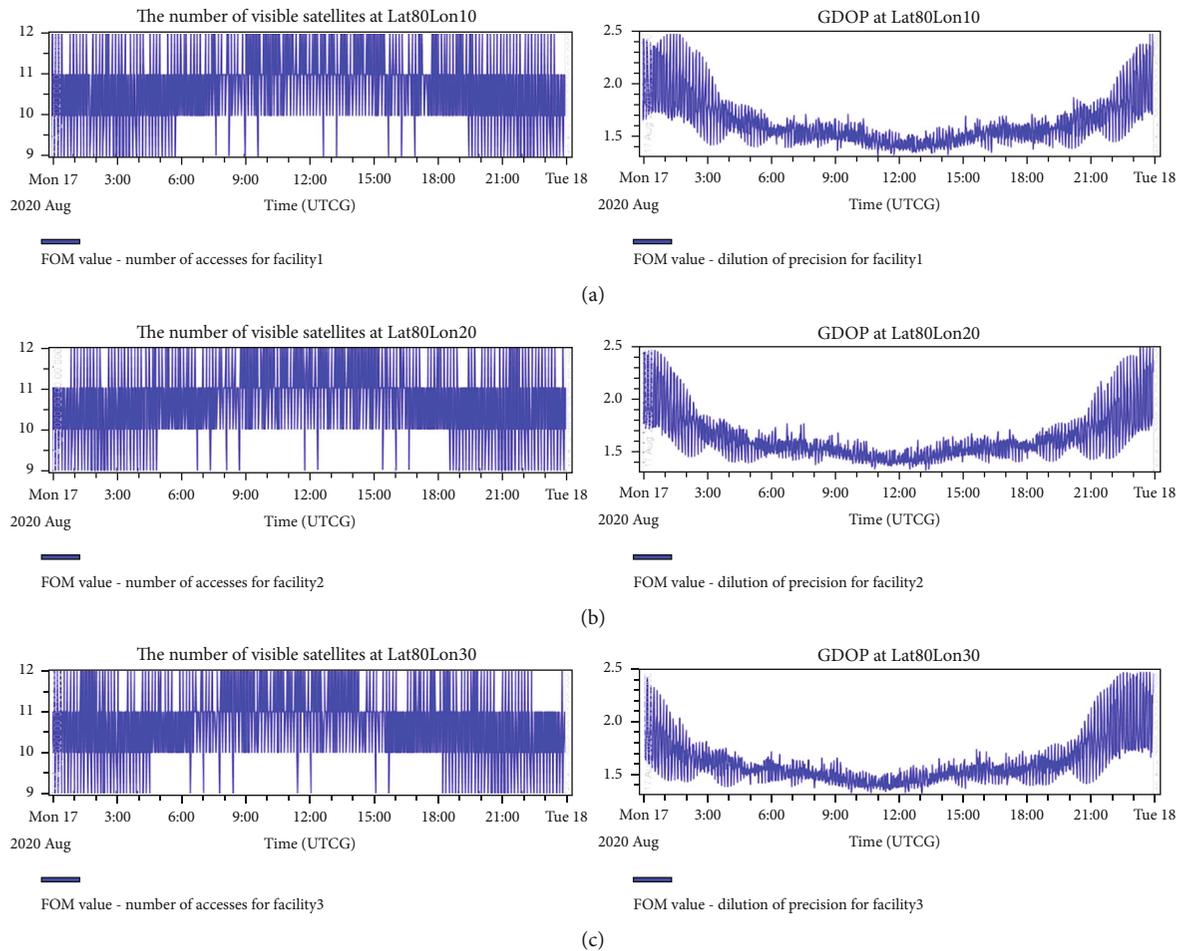


FIGURE 7: Number of visible satellites and GDOP at Lat80. (a) Lat80Lon10. (b) Lat80Lon20. (c) Lat80Lon30.

2.916, with a mean value of 1.778; in Figure 7(b), the GDOP at Lat80Lon20 varies from 1.621 to 3.103, with a mean value of 1.972; and in Figure 7(c), the GDOP at Lat80Lon30 varies from 1.621 to 3.113, with a mean value of 1.970.

In Figure 7, the shape variations of the GDOP are similar at the same latitude line, with little time migration. Thus, we

observe that changes in longitude have little impact on the number of visible satellites and the GDOP. Because the mean GDOP values in these three high-latitude regions are also below 2, it can be concluded that the space-based access network can meet the needs of communication and positioning at high latitudes.

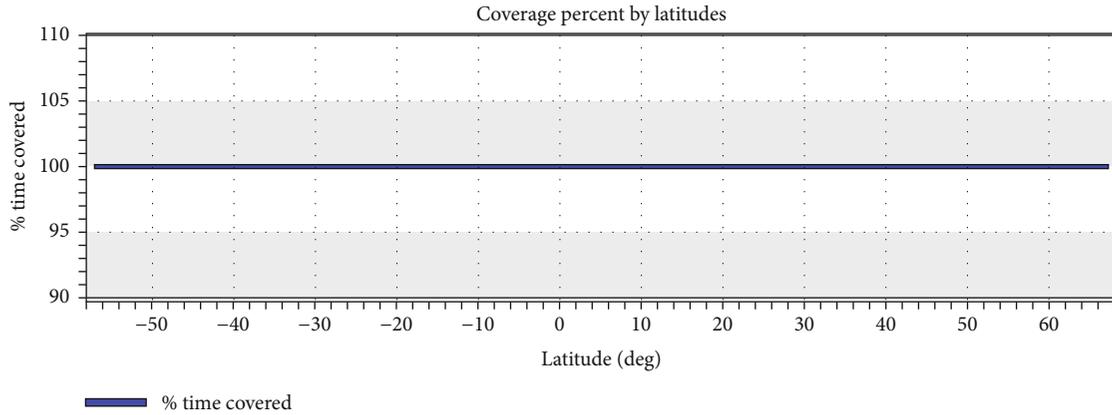


FIGURE 8: Coverage percent by latitude of the space-based backbone network.

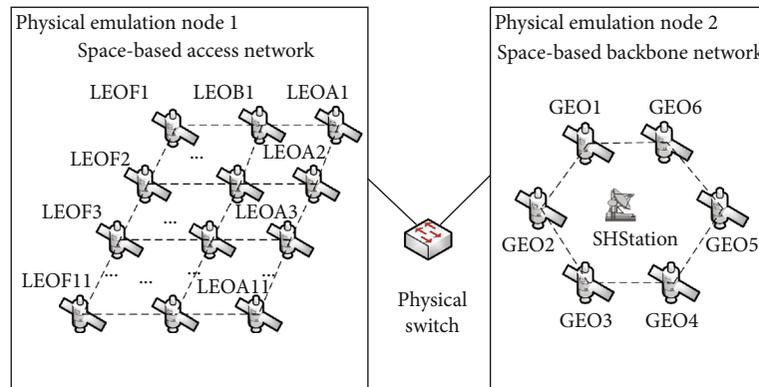


FIGURE 9: Topology of the SGIN experimental scenario.

Figure 8 displays the coverage percentage of the GEO constellation. This diagram shows that the space-based backbone network can cover the globe and meets the design requirements.

5.2. Emulation Using SGIN-Stack. To evaluate and demonstrate the capabilities of our platform, the SGIN scenario that is described in detail in Section 5.1 is emulated in this section.

5.2.1. Experimental Environment. Our SGIN-Stack platform is based on OpenStack Mitaka and consists of one STK simulation node, one network node, and two physical emulation nodes. The STK simulation node is integrated into a host with an Intel (R) Core (TM) i7-6700 processor and 8 GB of RAM. OpenStack Mitaka contains one control node, one network node, and two compute nodes. The typical hardware of an OpenStack Mitaka node is a Dell PowerEdge R830 server with two Intel (R) Xeon (R) E5-4620 v4 processors and 64 GB of RAM. The four nodes of OpenStack Mitaka are interconnected via a 10 gigabit switch, and all four nodes run CentOS 7 Linux. Experiments show that a single Dell PowerEdge R830 server can host seventeen KVM virtual satellite nodes or several hundred Docker virtual satellite nodes.

5.2.2. Emulated SGIN Topology. The topology of the SGIN emulation network is shown in Figure 9. One of the physical emulation nodes uses a lightweight virtualization (Docker)

approach to deploy the space-based access network in which the satellites are distributed over six orbital planes with 11 satellites per orbit, for a total of 66 satellites. Another physical emulation node adopts a fully virtualized mode (KVM) to deploy the space-based backbone network, which consists of six geosynchronous satellites and one ground station (SHStation).

The emulated network consists of 268 satellite links in total, which can be divided into three main categories: inter-satellite links (ISLs), interorbital links (IOLs), and gateway-satellite links (GWLs). An ISL is a link between two satellites in orbits of the same altitude; ISLs can be further divided into an interplane link and intraplane link. An interplane link is the link between satellites in different orbital planes, while an intraplane link is a link between satellites in the same orbital plane. In contrast, an IOL is a link between two satellites in orbits of different altitudes. A link between a satellite and the ground station is called a GWL. The link emulation module in the emulation platform uses multithreading to obtain the link emulation data for all links from storage and dynamically control the link characteristics of all virtual links.

5.2.3. Emulation of Delay. Figure 10 compares the actual emulated delay values and the corresponding theoretical values for five typical satellite links in the SGIN from

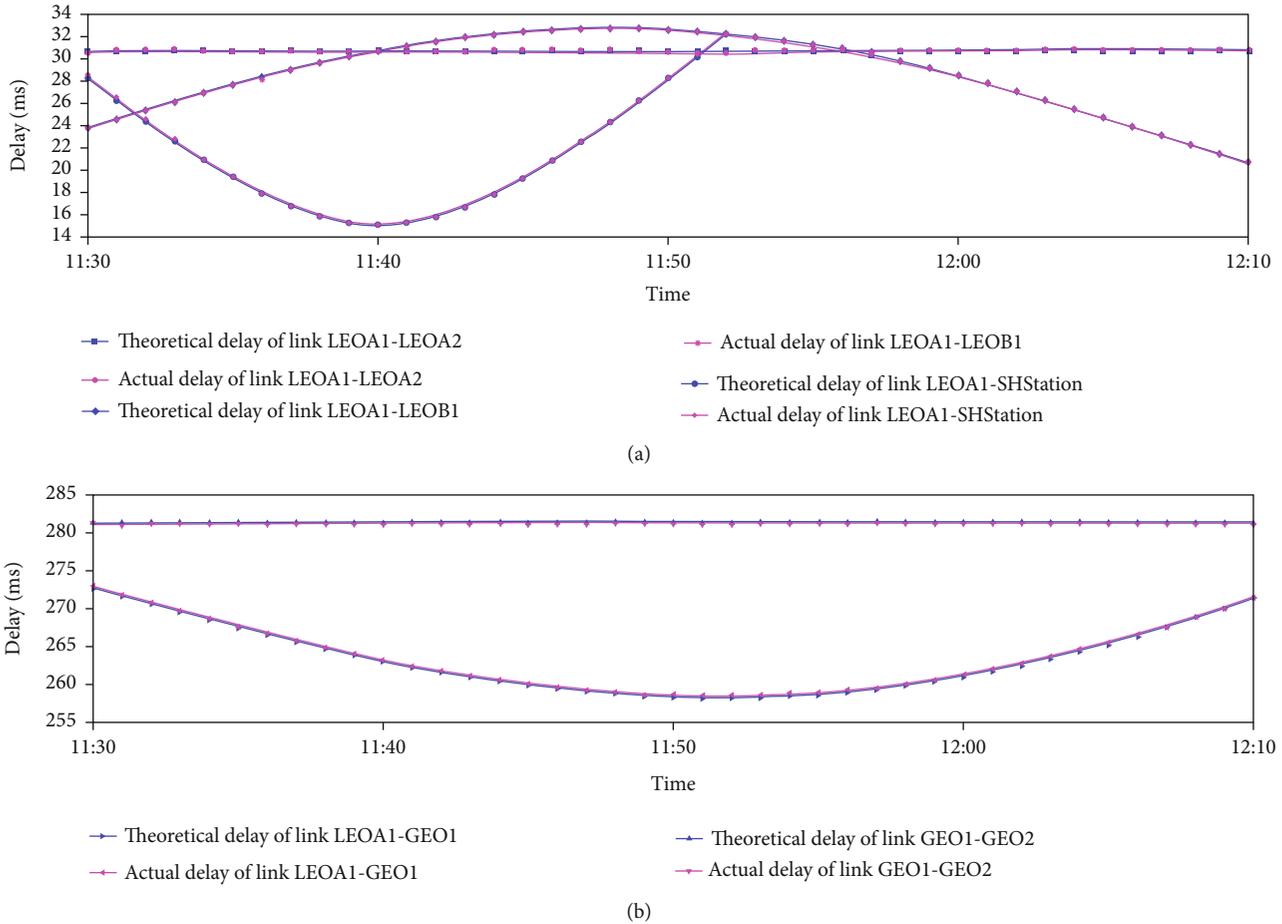


FIGURE 10: Comparison of theoretical and actual delays.

11:30:00 to 12:10:00. The five groups of satellite links are LEOA1-LEOA2, LEOA1-LEOB1, LEOA1-SHStation, LEOA1-GEO1, and GEO1-GEO2. Figure 10(a) shows the delay curves of the links LEOA1-LEOA2, LEOA1-LEOB1, and LEOA1-SHStation. Figure 10(b) shows the delay curves of the links LEOA1-GEO1 and GEO1-GEO2. During the experiment, the theoretical delay and the actual delay of each group are recorded at the same time every minute. Links LEOA1-LEOA2 and GEO1-GEO2 are intraplane ISLs, link LEOA1-LEOB1 is an interplane ISL, link LEOA1-GEO1 is an IOL, and link LEOA1-SHStation is a GWL. Since the links LEOA1-LEOA2 and GEO1-GEO2 are ISL and intraplane links, the two satellites are relatively static in each case, and the delay curves of links LEOA1-LEOA2 and GEO1-GEO2 are almost straight lines. The delay curves of the other links are uneven because the satellite pairs connected by those links are relatively dynamic and may even be invisible to each other at certain times. A link will be in the interrupt state if one satellite is invisible to the other, e.g., in the case of link LEOA1-SHStation in Figure 10 at 11:52. According to statistical calculations, in the case of real-time dynamic emulation, the delay error rates of the five groups of satellite links are 0.31%, 0.3%, 0.24%, 0.11%, and 0.04%. The total average error rate

is 0.2%. The results show that the delays in the emulation platform closely match the theoretical delays in STK.

EmuStack merely uses the TC tool in the Linux kernel to create a Netem queue in the network card of the source satellite to emulate delay. To compare the emulation performance of EmuStack and SGIN-Stack integrated with the dynamic differential compensation technique, three typical links were selected, and the delay emulation performance of the two solutions was tested. Each group sends 100 data packets for testing. The experimental results are shown in Figure 11. As shown in Figure 11, the IOL represented in panel (a) is the link between LEOA3 and GEO3, and the link delay is set to 70 ms; the space-based access ISL represented in panel (b) is the link between LEOA1 and LEOA3, and the link delay is set to 40 ms; the space-based backbone ISL represented in panel (c) is the link between GEO1 and GEO3, and the link delay is set to 80 ms. The emulation results of SGIN-Stack integrated with the dynamic differential compensation technique are compared to those of EmuStack with traditional TC. The average error of the delay emulation in SGIN-Stack is approximately 0.1 ms. The maximum error of EmuStack is approximately 0.44 ms, and the overall error fluctuates within a range of approximately 0.4 ms.

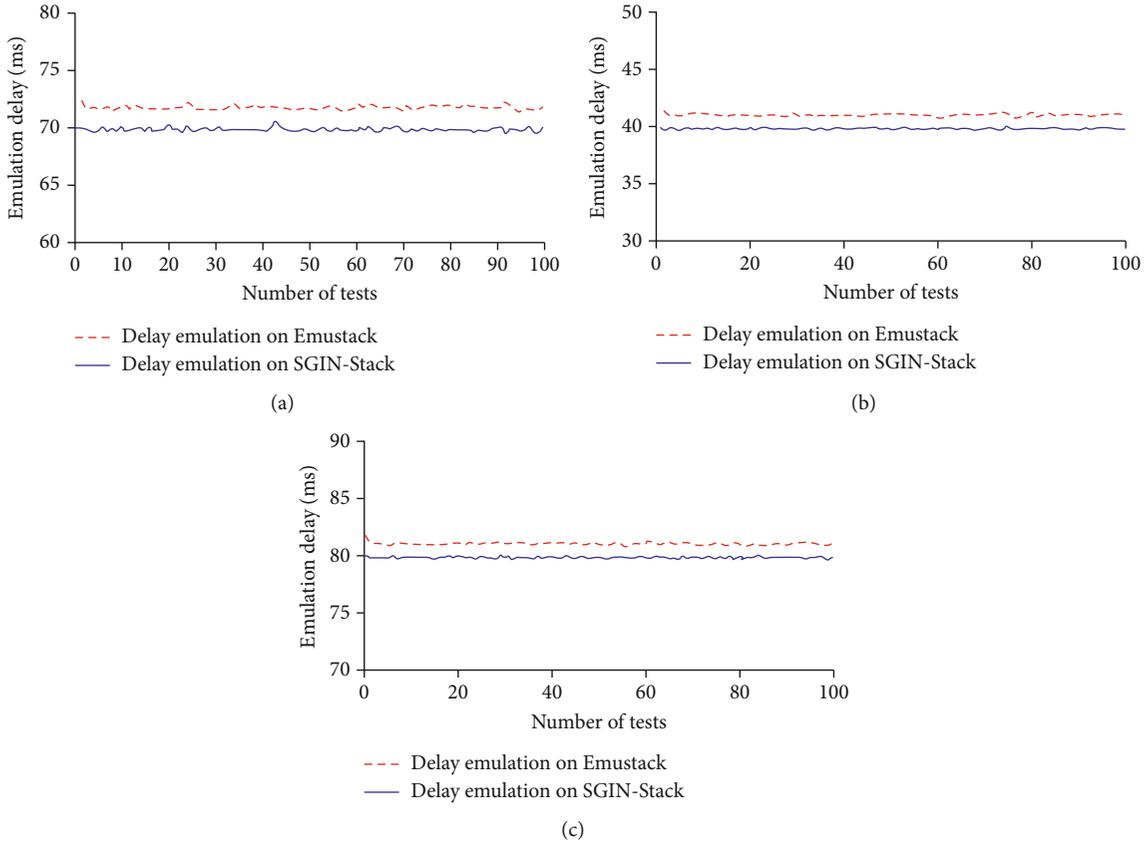


FIGURE 11: Comparison of theoretical and actual delays. (a) IOL 70 ms, (b) Space based access network ISL 40 ms, (c) Space based access network ISL 80 ms.

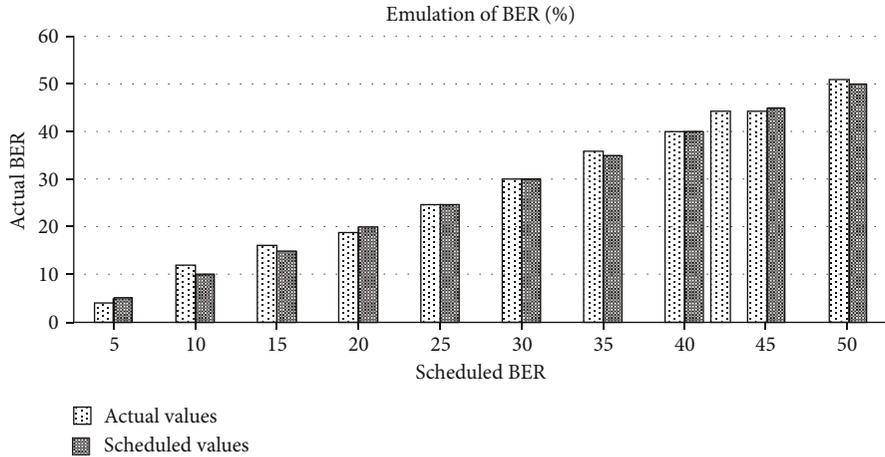


FIGURE 12: Emulation of the BER.

Experiments show that SGIN-Stack has higher emulation accuracy than EmuStack. It can be seen that SGIN-Stack integrated with the dynamic differential compensation technology can improve the emulation fidelity of delay by compensating the transmission delay of virtual and physical links.

5.2.4. *Emulation of the BER.* To verify the accuracy of BER, it is necessary to send a certain number of packets, measure the

probability of packet loss, and then compare the result to the scheduled BER. Satellite link LEOA1-LEOA2 is taken as an example here; the results for other links are similar. The experimental results are shown in Figure 12. The scheduled BER values range from 5% to 50%, and 100 packets are sent for each evaluation. The actual emulated values of the BER are compared against the scheduled BER values. The results show that the proposed method of random number generation is feasible.

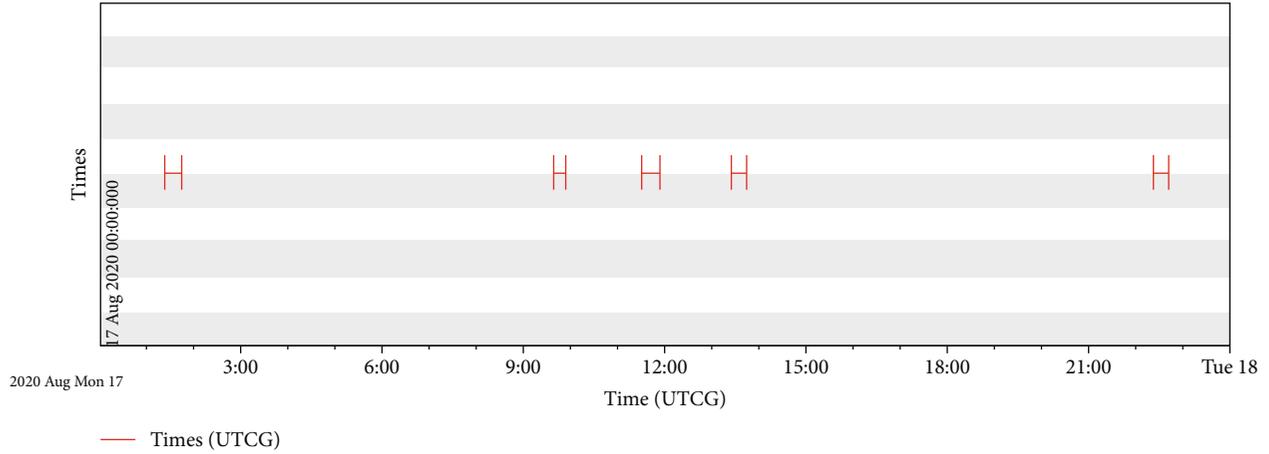


FIGURE 13: Access times between LEOA1 and SHStation.

5.2.5. *Emulation of Intermittence and Bandwidth.* Link LEOA1-SHStation is selected to compare the bandwidth and intermittence between theory and practice. The access times between LEOA1 and SHStation in STK are shown in Figure 13. The emulation period is from 00:00:00 to 24:00:00; there are five access times in this period, and each access lasts approximately twenty minutes. Other parameters of this link are shown in Table 2.

Figure 14 shows the rates and access times on SGIN-Stack. Based on this figure, we can observe that the access behavior between LEOA1 and SHStation is emulated accurately; there are five access times in this period, and each access lasts approximately twenty minutes. The throughput values plotted in Figure 14 are the mean values over one minute, and any delay reduces the throughput. As a result, the throughput is below but close to 200 kB/s. The result shows that the throughput and intermittence of a link in the SGIN closely match those in STK; this finding proves that the platform supports the emulation of intermittent links in an SGIN.

To verify the emulation accuracy of the bandwidth, under the premise that the satellite link LEOA1-SHStation is connected and the delay is 0 ms, five groups of theoretical bandwidth values are set, and the average value is taken for each group of 10 tests. The five groups of theoretical bandwidth values are 10 Mbps/s, 50 Mbps/s, 100 Mbps/s, 200 Mbps/s, and 500 Mbps/s. Figure 15 shows the experimental results of the emulation of bandwidth. The average actual throughput of the five groups tested are 9.6 Mbps/s, 47.8 Mbps/s, 95.5 Mbps/s, 190.9 Mbps/s, and 476.3 Mbps/s. The average error rate is 4.43%. It can be seen that SGIN-Stack has high bandwidth simulation accuracy.

5.3. *Example of Reverse Linkage Control Emulation.* Two typical examples of the reverse linkage control emulation scenario are proposed to verify the correctness and reliability of the reverse linkage control between OpenStack and STK. The first experiment is the reverse linkage control process of the satellite orbit maneuver. The second experiment is

TABLE 2: Link characteristics of the link LEOA1-SHStation.

Link	Bandwidth	Delay	Packet loss rate
LEOA1 => SHStation	200 kB/s	8 ms to 17 ms	0% to 1%

the reverse linkage control process of the satellite attitude and solar panel adjustment.

The satellite orbit maneuver experiment takes the satellite “test” as an example. The satellite “test” and “test1” are sent into orbit at a perigee altitude of approximately 192 km, and their apogee altitude is approximately 570 km. Table 3 shows the initial states of “test” and “test1.” The satellite “test” initially follows a fixed operating orbit, and “test1” always follows a fixed operating orbit. In addition, there is a satellite link named test-test1 between “test” and “test1.” When the ground station node sends an orbital maneuver command to “test,” it performs the corresponding orbital maneuver.

Figure 16(a) describes in detail the process of an orbital maneuver for transitioning from an elliptical orbit to a circular orbit. The elliptical and circular orbits are highlighted in green and red, respectively. Figure 16(b) shows the satellite’s specific changes in longitude, latitude, and altitude. The satellite is subjected to orbital control at approximately 6:45 a.m. The momentum of the satellite “test” undergoes a positive change, and the orbit changes from an elliptical orbit to a circular orbit with a radius of approximately 570 km. In Figure 17, the delay of the virtual satellite link test-test1 in SGIN-Stack is shown: before 10:45 a.m., the link delay curve is gentle, whereas after 10:45 a.m., the curve shows a sudden increase because the satellite “test” is in the process of executing an orbital maneuver, and the link length of test-test1 is increasing.

The satellite attitude and solar panel adjustment experiment take the satellite “test1” as an example. At the initial moment, the attitude angles of the body coordinate system

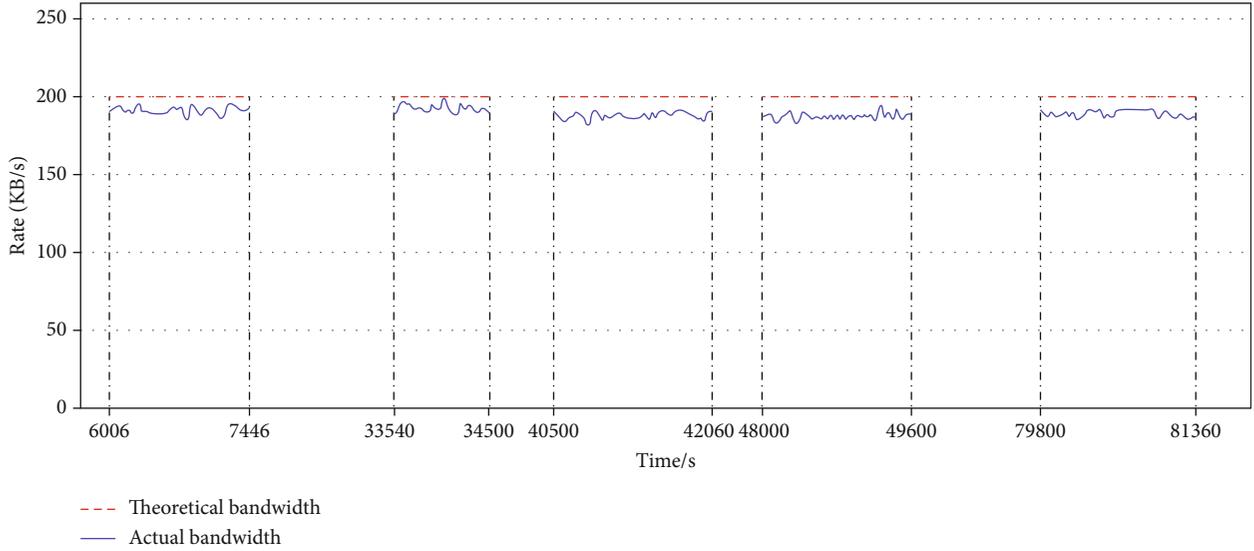


FIGURE 14: Bandwidth and intermittence of the link LEOA1-SHStation.

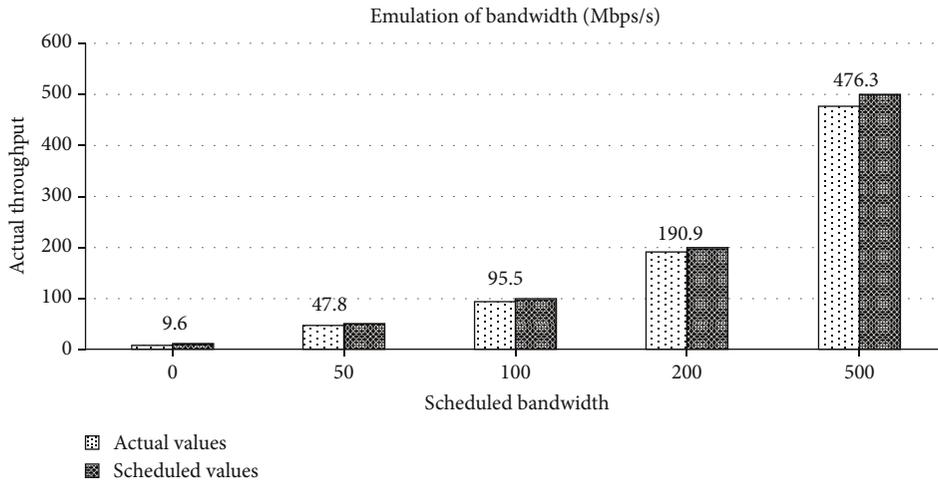


FIGURE 15: Emulation of the bandwidth of the link LEOA1-SHStation.

TABLE 3: Initial states of the satellites “test” and “test1”.

Initial state	Satellite “test”	Satellite “test1”
Semimajor axis	6771 km	6771 km
Coordinate type	Keplerian	Keplerian
Eccentricity rate	0.02953	0.02953
Inclination	28 deg	28 deg
Mean anomaly	0 deg	20 deg

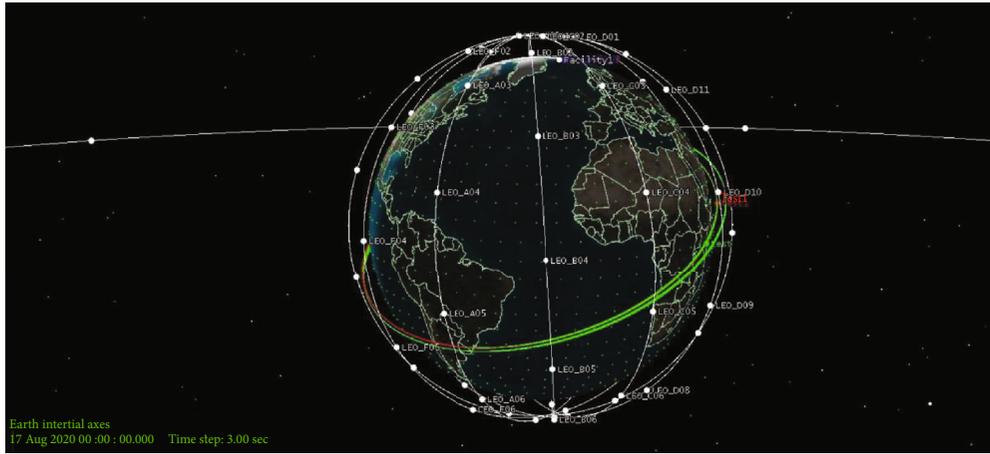
of the satellite “test1” in the SGIN simulation network relative to the orbital coordinate system are (for Figures 18(a)–18(c)) $\psi = 20^\circ$, $\theta = 7^\circ$, and $\phi = 7^\circ$. When the virtual ground station node in the SGIN emulation network sends an instruction to open the solar panel to the virtual satellite node corresponding to the satellite “test1,” the reverse linkage control module of SGIN-Stack will send the instruction to the

coordinated control module of the STK simulation node. The coordinated control module will drive the satellite “test1” in the STK simulation network to perform the solar panel opening operation according to the command. After the solar panels of the satellite “test1” are unfolded, the attitude angles of the satellite “test1” become (Figure 18(d)) $\psi = 0^\circ$, $\theta = 2^\circ$, and $\phi = 2^\circ$.

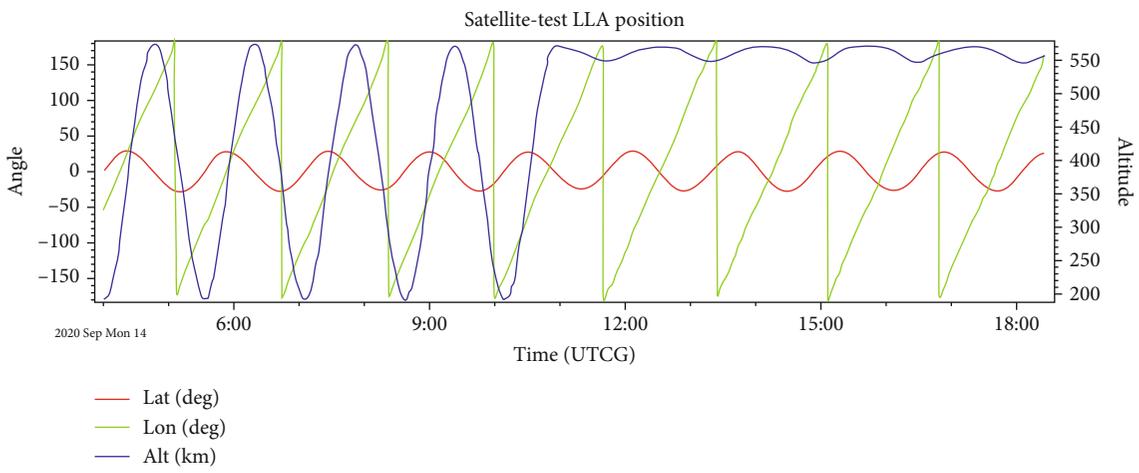
The above two experiments show that SGIN-Stack can reversely control the satellite nodes in the SGIN simulation network in the STK simulation node, including orbital maneuvers, attitudes, and solar panels, to realize the emulation of reverse linkage control.

6. Conclusions

As a current hot spot, SGINs are expected to be widely used in IoT. Before an SGIN is actually deployed, a reliable



(a)



(b)

FIGURE 16: Satellite “test.” (a) Orbital maneuver. (b) Latitude, longitude, and altitude (LLA) position.

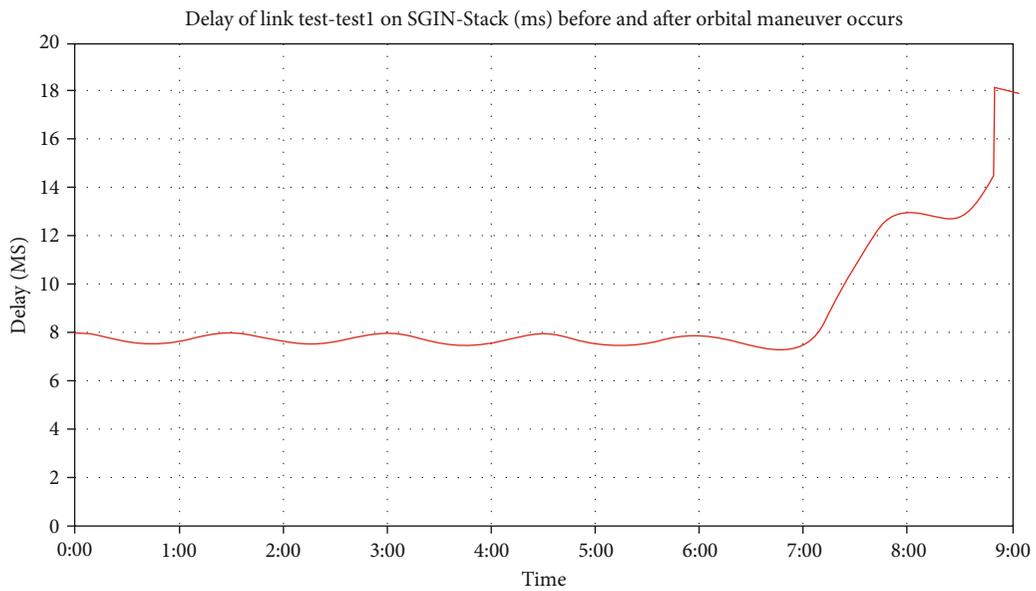


FIGURE 17: Delay of the satellite link test-test1 in SGIN-Stack.

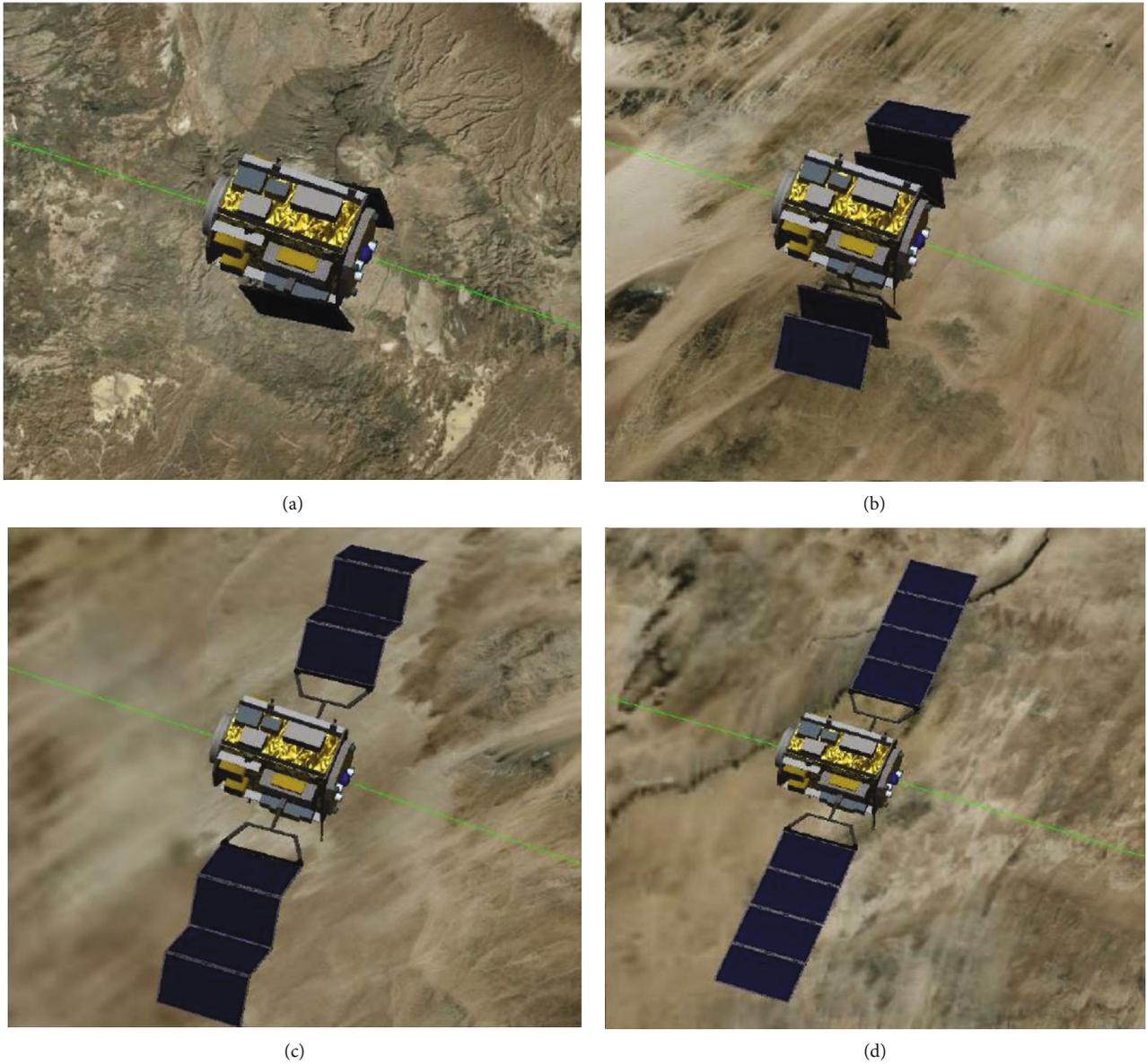


FIGURE 18: Deployment of solar panels and attitude angles of the body coordinate system. (a–c) $\psi = 20^\circ$, $\theta = 7^\circ$, and $\phi = 7^\circ$. (d) $\psi = 0^\circ$, $\theta = 2^\circ$, and $\phi = 2^\circ$.

emulation test platform is needed to test related technologies and protocols. In this paper, we propose a real-time, flexible, and dynamic SGIN emulation platform called SGIN-Stack based on OpenStack. The newly developed emulation platform is designed to consider dynamic topological characteristics, such as variable delays and BERs, unstable end-to-end paths between satellites, and maneuvering of satellites. Through the design of a two-way dynamic linkage, a seamless connection between OpenStack and STK is achieved in SGIN-Stack. Through this two-way dynamic linkage, SGIN-Stack can emulate satellite link characteristics, satellite orbital maneuvers, satellite attitudes, and solar panel deployment dynamically and accurately. In this way, the topology, new protocols, and IoT applications of an

SGIN can be rigorously tested and verified on SGIN-Stack before actual deployment.

Through the emulation of SGINs, relevant application tests for IoT services can be conducted in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 61972182, 61672264] and the National Key R&D Program of China [grant number 2016YFB0800305].

References

- [1] Z. Yi, X. Du, Y. Liao, and L. Cao, "A quality-of-service-aware dynamic evolution model for space-ground integrated network," *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, Article ID 155014771772864, 2017.
- [2] Y. Xiao, T. Zhang, and L. Liu, "Addressing subnet division based on geographical information for satellite-ground integrated network," *IEEE Access*, vol. 6, pp. 75824–75833, 2018.
- [3] Y. Shi, Y. Cao, J. Liu, and N. Kato, "A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks," *IEEE Network*, vol. 33, no. 1, pp. 29–35, 2019.
- [4] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Effects of correlated multivariate FSO channel on outage performance of space-air-ground integrated network (SAGIN)," *Wireless Personal Communications*, vol. 106, no. 1, pp. 7–25, 2019.
- [5] Y. Wang, J. Yang, X. Guo, and Z. Qu, "Satellite edge computing for the Internet of Things in aerospace," *Sensors*, vol. 19, no. 20, article 4375, 2019.
- [6] X. Cheng, F. Lyu, W. Quan et al., "Space/aerial-assisted computing offloading for IoT applications: a learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.
- [7] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting internet of remote things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, 2016.
- [8] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO satellite constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.
- [9] K. L. A. Yau, A. R. Syed, W. Hashim, J. Qadir, C. Wu, and N. Hassan, "Maritime networking: bringing internet to the sea," *IEEE Access*, vol. 7, pp. 48236–48255, 2019.
- [10] T. Hong, W. Zhao, R. Liu, and M. Kadoch, "Space-air-ground IoT network and related key technologies," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 96–104, 2020.
- [11] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.
- [12] B. Zhao, P. Liu, X. Wang, and I. You, "Toward efficient authentication for space-air-ground integrated Internet of things," *International Journal of Distributed Sensor Networks*, vol. 15, no. 7, Article ID 155014771986039, 2019.
- [13] X. Liu, X. Zhai, W. Lu, and C. Wu, "QoS-Guarantee Resource Allocation for Multibeam Satellite Industrial Internet of Things with NOMA," *IEEE Transactions on Industrial Informatics*, p. 1, 2019.
- [14] T. Lu, W. Zhang, X. Ni et al., "A scalable network emulation architecture for space internetworking," in *2016 IEEE International Conference on Communication Systems (ICCS)*, pp. 1–5, Shenzhen, China, December 2016.
- [15] H. Li, H. Zhou, H. Zhang, B. Feng, and W. Shi, "Emustack: an openstack-based dtn network emulation platform (extended version)," *Mobile Information Systems*, vol. 2016, 15 pages, 2016.
- [16] Y. Li, P. Hui, D. Jin, and S. Chen, "Delay-tolerant network protocol testing and evaluation," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 258–266, 2015.
- [17] A. Dutta and O. Gnawali, "Large-scale network protocol emulation on commodity cloud," in *2014 IEEE Global Communications Conference*, pp. 1114–1119, Austin, TX, USA, December 2014.
- [18] Openstack <http://www.openstack.org/>.
- [19] S.-h. Zhang, X.-x. Meng, and L.-h. Wang, "SDNForensics: a comprehensive forensics framework for software defined network," in *Proceedings of the International Conference on Computer Networks and Communication Technology (CNCT 2016)*, Atlantis Press, December 2016.
- [20] X. Wang, M. Zhai, and G. Zhang, "Research on high-fidelity router emulation technologies based on cloud platform," in *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pp. 1–4, Tokyo, Japan, October 2018.
- [21] T. Li, H. Zhou, H. Luo, and S. Yu, "SeRvICE: a software defined framework for integrated space-terrestrial satellite communication," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 703–716, 2018.
- [22] Stk <https://agi.com/products>.
- [23] J. Qi, Z. Li, and G. Liu, "Research on coverage and link of multi-layer Satellite Network based on STK," in *2015 10th International Conference on Communications and Networking in China (ChinaCom)*, pp. 410–415, Shanghai, China, August 2015.
- [24] B. Hubert, *Linux advanced routing & traffic control HOWTO*, vol. 1, Netherlabs BV, 2002.
- [25] F. Xhafa, X. Herrero, A. Barolli, L. Barolli, and M. Takizawa, "Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem," *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1086–1100, 2013.
- [26] S. McCamish and M. Romano, "Simulation of relative multiple spacecraft dynamics and control with MATLAB-SIMULINK and Satellite Tool Kit," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, p. 6805, Hilton Head, SC, USA, June 2007.
- [27] V. B. S. S. I. Dutt, G. S. B. Rao, S. S. Rani, S. R. Babu, R. Goswami, and C. U. Kumari, "Investigation of GDOP for precise user position computation with all satellites in view and optimum four satellite configurations," *The Journal of Indian Geophysical Union*, vol. 13, no. 3, pp. 139–148, 2009.
- [28] S. C. Spangelo, J. W. Cutler, A. T. Klesh, and D. R. Boone, "Models and tools to evaluate space communication network capacity," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2387–2404, 2012.
- [29] P. Muri, J. McNair, J. Antoon, A. Gordon-Ross, K. Cason, and N. Fitz-Coy, "Topology design and performance analysis for networked earth observing small satellites," in *2011-MILCOM 2011 Military Communications Conference*, p. 1940, Baltimore, MD, USA, November 2011.
- [30] M. Yin, J. Li, X. Wang, and H. Baoyin, "A rapid method for validation and visualization of agile earth-observation satellites scheduling," *Astrodynamic*, vol. 2, no. 4, pp. 325–337, 2018.

- [31] A. N. U. Medina and G. Qiang, "QoS routing for LEO satellite networks," in *Pervasive Computing and the Networked World. ICPCA/SWS 2012. Lecture Notes in Computer Science, vol 7719*, Q. Zu, B. Hu, and A. Elçi, Eds., pp. 482–494, Springer, Berlin, Heidelberg, 2012.
- [32] D. R. Beering, S. Tseng, J. L. Hayden et al., "RF communication data model for satellite networks," in *MILCOM 2009-2009 IEEE Military Communications Conference*, pp. 1–7, Boston, MA, USA, October 2009.