IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2019, accepted March 6, 2019, date of publication March 20, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906397

NPVT: Network Protocol Validation Testbed for Integrated Space-Terrestrial Network

ZENGYIN YANG[1,2], HEWU LI[2,3], QIAN WU[2,3], AND JIANPING WU[1,2], (Fellow, IEEE)
[1]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]China Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[3]Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China

Corresponding author: Qian Wu (wuqian@cernet.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61832013, in part by the Beijing Municipal Science Technology Commission under Grant Z171100005217001, and in part by the Beijing National Research Center for Information Science and Technology under Grant 20031887521.

ABSTRACT With the development of the terrestrial network and the increase in demand for providing global Internet access, using the integrated space-terrestrial network (ISTN) has attracted significant attention to provide the global Internet service. However, ISTN faces a unique set of challenges, such as time-varying link quality and dynamic link connection in its space network, and the significant differences between space network and the terrestrial network of ITSN. These unique characteristics may impose an adverse impact on network protocols of ISTN. The focus of this paper is to design a network protocol validation testbed (NPVT) for ISTN. Using the static hardware devices with real network protocols, the proposed testbed was designed to emulate the dynamic space network based on the satellite operational data, and then to provide a near-real space network environment for the validation of diverse network protocols. To further support network protocol validation when space network and terrestrial network are integrated, we integrated the testbed with the terrestrial operational next-generation Internet (i.e., CERNET2) and mobile communication network at the backbone level. The extensive experiments demonstrated that NPVT is capable of validating diverse network protocols in complicated and dynamic scenarios.

INDEX TERMS Integrated space-terrestrial network (ISTN). network protocol, validation, testbed.

I. INTRODUCTION

With the expansion of human activities, the demand for providing Internet service in remote, marine, and aerial environments has increased dramatically. Consequently, building the integrated space-terrestrial network (ISTN) has been receiving significant attention from multiple points. For instance, the traditional space network operators (e.g., Iridium [1] and Inmarsat [2]) have begun to integrate their satellite networks with terrestrial network to provide IP-based service; the Internet content providers (e.g., Google [3]) have started to scale their the terrestrial services over space network, and mobile communication network (e.g., 5G) are integrating with the space-based networks for ubiquitous coverage [4].

Generally, ISTN is a global networking system composed of space network and terrestrial network. Terrestrial network consists of the terrestrial Internet and 2G/3G/4G/5G mobile communication networks; space network consists of various satellite systems. The key to achieving integration of space network and terrestrial network is the integration of network protocols. When the space network and terrestrial network use the same protocol to provide the Internet service, the integration of both networks is realized naturally. However, the network topology of space network deviates significantly from that of terrestrial network. While the links between terrestrial routers are almost static, link connectivity and link quality between routers in space network change frequently, which may impose some unexpected impact on network protocols in the terrestrial network or space network of ISTN. It is important to understand the behavior of network protocols in ISTN and make some helpful suggestions for improvement.

Testbed is a popular tool that can test network protocols without significant fidelity loss. It applies validated hardware devices with real network protocols to emulate the ISTN. Because both devices and network protocols have

The associate editor coordinating the review of this manuscript and approving it for publication was Vittorio Camarchia.

been validated fully, the fidelity of testbed is directly related to its design.

We have developed a high-fidelity network protocol validation testbed (NPVT) to analyze the behavior of diverse network protocols in ISTN. When we designed the testbed, the following four goals were in mind:

- To support new and emerging network protocols, the testbed should be designed to support Internet protocol version 6 (IPv6).
- The networking of testbed should change automatically like that of a real-world space network, especially, the dynamic link connectivity and the time-varying link quality that have a direct influence on the performance of network protocols.
- The backbone-level interconnection between the emulated space network and the operational terrestrial network should be achieved as the oncoming ISTN does.
- The testbed should automatically deploy diverse network protocols without manual operations.

The challenge and focus of this paper are to develop an architecture for the testbed tailored for testing various network protocols. The architecture consists of a user plane, a control plane, and a data plane. The user plane transforms the topology dynamics of space network into a series of commands using the satellite operational data, and then the control plane uses these commands to change, in real-time, the networking relationships between hardware devices and network protocols in the user plane. The real time acquisition of emulating the real satellite system is the basis of testbed. By dynamically changing the flow rules in programmable switches and the queuing rules in emulated space routers, the testbed emulates dynamic link connectivity and time-varying link quality of the space network. The deployment of network protocols relies on an *Agent* in each emulated satellite that interacts with network protocols in a different working environment. To further support network protocol validation when space network and terrestrial network are integrated, we have designed a terrestrial-space integrated system (TSIS) for the testbed to securely interconnect the real terrestrial network.

Extensive experiments ensure the efficiency and the scalability of NPVT. NPVT can automatically emulate various space networks, and deploy diverse network protocols, and integrate itself with the real terrestrial networks. In addition, we validate the scalability of testbed's control plane and data plane. Results demonstrated that NPVT has high scalability.

In summary, we have four-fold contributions:

- We propose an architecture for NPVT, in which we can flexibly change the networking of testbed.
- We provide detailed methods for reliably emulating the dynamics of the space network.
- We design a TSIS to integrate the emulated space network with the terrestrial next-generation Internet (CERNET2) [5] as well as the mobile communication network.

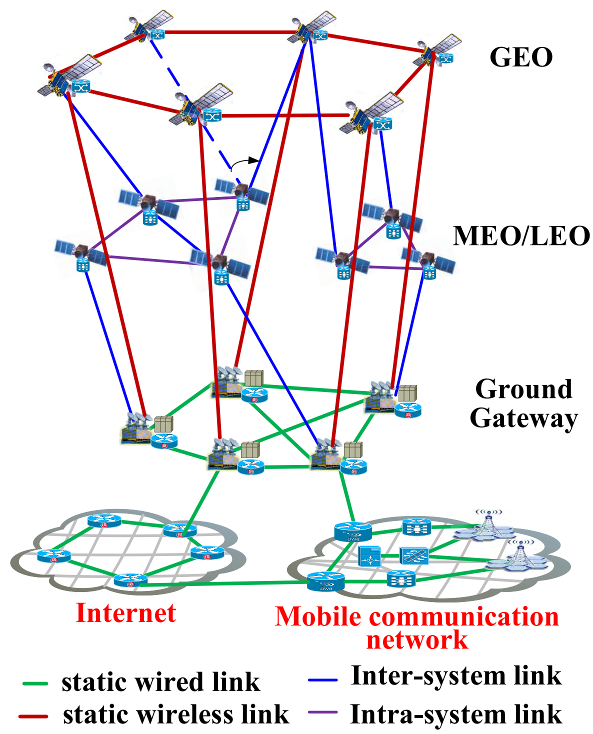


FIGURE 1. Architecture of ISTN.

- We validate the efficiency and the scalability of the testbed.

The rest of this paper is organized as follows. Section II gives an overview of ISTN, and Section III summarizes the motivations and challenges of designing the NPVT. Section IV and section V provide the architecture of NPVT and the key technologies, respectively. Section VI presents the testbed setup and the testbed validation. Finally, section VII presents an overview of related testbeds, and section VIII concludes this paper.

II. BACKGROUND AND MOTIVATION

A. OVERVIEW OF ISTN

To provide global Internet service, many network operators have focused on the integrated space-terrestrial network (ISTN). Generally, ISTN is an integrated network of space network and terrestrial network, as shown in Fig. 1. The terrestrial network consists of Internet and 2G, 3G, 4G, and 5G mobile communication networks. The space network is a networking system of multiple satellite networks. The satellite network consists of geosynchronous orbit (GEO), medium earth orbit (MEO), low earth orbit (LEO) satellites, and ground gateways. These network nodes are interconnected to form a global mesh network using satellite-to-satellite links and satellite-to-gateway links. At the same time, the space network connects with the terrestrial network using one or more ground gateways. Global users have the flexibility to access both terrestrial network and space network. Communication between these users is effectively routed between three classes of networks.

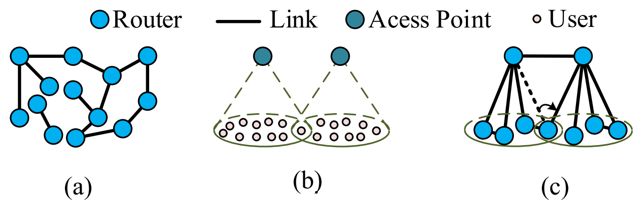


FIGURE 2. Topologies of different networks in ISTN.

There exist three classes of topologies corresponding to different networks in the ISTN. A typical topology for the Internet is shown in Fig. 2 (a). Although this type of network topology builds many point-to-point link connections among routers, the entire network does not exhibit a high degree of dynamic behavior. The topology changes typically occur on the order of hours to days.

The topology of the access network of mobile communication network can be represented by the point-to-multipoint mode, as shown in Fig. 2 (b). In this topology, one access point applies one or more antenna beams to connect hundreds of users. In addition, the connection between the access point and users has a certain degree of mobility, but it minimally affects the network topology of a mobile communication network. This is because the mobility mainly occurs within a router of the core network of mobile communication network. The topology of the core network of mobile communication network is similar to the topology of terrestrial Internet.

The space network combines with the characteristics of the above networks, as shown in Fig. 2 (c). In this type of network, each satellite or ground gateway acts as a router, which builds many point-to-point links with its neighbors using multiple antennas. The networking among space routers is similar to that among the Internet's routers. This networking is different from the user access between a satellite and multiple terminals. The latter one works like point-to-multipoint mode of the mobile communication network.

In addition, the space network has a high degree of mobility, which is the result of constant and predictable movement of the satellite. The satellite's movement directly leads to the change of the relative position of space routers, and further cause the changes in link quality and link connectivity. (1) **Time-varying link quality.** The space link is essentially a wireless link whose quality is very susceptible to link distance because of the high signal attenuation caused by long distances between space routers. Link quality can be described with some key link parameters, namely, high link delay, severe packet loss, and low interface bandwidth. The link quality also varies with the relative distance between satellites. (2) **Dynamic link connection.** The dynamics of link connection are caused by the satellite's movement. When a space router flies out of its peer router's coverage, it will disconnect its link to the peer router, and meanwhile, it waits for its broken link to be available or attempts to peer with the new router. With the network scale increasing, the change frequency of link connectivity increases exponentially.

B. NETWORK PROTOCOLS

Corresponding to the three networks of ISTN, there are three classes of protocol suites, namely, the transmission control protocol / Internet protocol (TCP/IP) suite for terrestrial Internet, CCSDS (consultative committee for space data systems) standards for space network [6], and 3G/4G/5G standards for mobile communication network.

To achieve the integration of these three networks of ISTN, the IP and its related network protocols are preferable. Fig. 3 shows an example of the network protocol architecture of ISTN. IP and related network protocols are designed to interconnect these heterogeneous networks in ISTN. Applying these protocols to ISTN can accelerate the construction of ISTN. When the space network and the terrestrial network of ITSN use the same protocol to provide the same Internet service, the integration of space network and terrestrial network is realized naturally. In addition, the use of IP and related protocols allows the space network to provide general IP connectivity for different services (data, voice, remote sensing data) using the common equipment, which can significantly increase the flexibility of satellite operation. Finally, because these protocols have been widely validated, the cost and risk of designing new network protocols for ISTN can be reduced.

In space network of ISTN, the existing CCSDS protocols including advanced orbiting systems (AOS) [7] and IP over CCSDS [8] are used to support the implementation of IP and related network protocols. IP over CCSDS is a standard published by CCSDS. Furthermore, the DTN (delay-tolerant networking) [9] is used to support the scenarios where the end-to-end transmission cannot be guaranteed.

The integration with terrestrial Internet and mobile communication network has been studied fully. That integration is, thus, not taken as our core concern in the paper.

III. MOTIVATIONS AND CHALLENGES

A. MOTIVATIONS

Before using IP and related network protocols to integrate space network with terrestrial network, it is important to study the protocols thoroughly. Because the IP-based network protocols are originally designed for almost static networks, these network protocols may behave inefficiently in some cases of ISTN.

To best examine the behavior of network protocols, we choose the tool of the testbed, as opposed to numerical simulation or the use of an actual satellite with onboard processing. Numerical simulation is the easiest method, as it does not require the use of real network devices or real network protocols. However, it can lead to a significant loss of fidelity because it uses a list of hypothetical methods to convert hardware devices and network protocols into incomplete forms. For instance, the behavior of a network protocol is converted into a series of discrete events without simulating the full scope of the protocol. The onboard-processing satellite is a reliable tool. However, a satellite launch requires significant time and the cost is very high. More importantly, the network

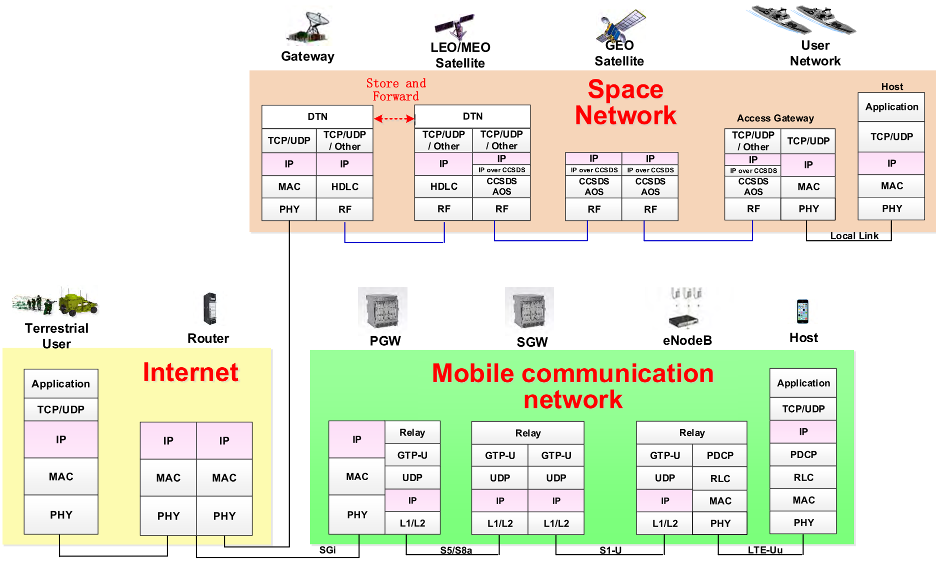


FIGURE 3. An example of the network protocol architecture of ISTN.

scale is limited by the number of onboard-processing satellites. The testbed, which combines the advantages of numerical simulation with the onboard processing functions of a satellite, can test network protocols reliably at a lower cost. It uses the hardware devices and the real network protocols to emulate the space network. Both the hardware devices and protocols have been widely validated. Naturally, the fidelity of the testbed is directly related to the design of the testbed.

B. CHALLENGES

However, designing a high-fidelity testbed is not an easy task, owing to the wide differences between terrestrial network and space network of ISTN. The following challenges are considered. (1) Using the static terrestrial hardware devices and softwares to emulate the dynamic space network requires the networking relationship of static hardware in the testbed to be frequently re-arranged to emulate the dynamic behavior of the space network. Once link connectivity of the real-world space network changes, the testbed needs to disconnect/reconnect the links between static hardware. Similarly, when the link quality of the real-world space network varies, the quality of the emulated link needs to be changed correspondingly. Moreover, these emulation methods must not interfere with the normal behavior of the network protocols. (2) There exist multiple combinations of network protocols to be initialized and maintained. When the testbed is set up, it needs to initialize the corresponding network protocols according to the experimental requirements. It is then required to re-deploy protocols automatically once the network topology of the testbed changes. For instance, when two emulated space routers build a new link, the testbed should automatically deploy the network protocols' neighboring relationship attached to that link. (3) The integration between the emulated space network and the real terrestrial network is required to overcome the wide differences between both networks. The emulated space network is a private AS (autonomous system) [10], whereas the terrestrial network is a public AS [10]. Therefore, methodologies are needed to eliminate the barriers to the interconnection between private AS and public AS. Moreover, it is non-trivial to coordinate the time progress in real networks and emulated space networks. (4) The procedures needed to configure these terrestrial devices to emulate space network do not seem so easy. The testbed needs to correctly get/change the working status of devices in testbed according to the emulation requirement.

IV. THE TESTBED ARCHITECTURE

The testbed is designed to provide the near-real ITSN environment for the validation of diverse network protocols. The NPVT's architecture consists of three parts, namely, user plane, control plane, and data plane. The user plane transforms experimental requirements into command information, and then the control plane dynamically adjusts the networking relationship of hardware devices and softwares in the data plane to emulate the ISTN. The detailed architecture of the testbed is shown in Fig. 4 and discussed as follows.

A. USER PLANE

The user plane is to generate the command information according to the experimental requirements including satellite operational data, network protocols for validation, performance indexes, and so on. The topology dynamics of space network are calculated by taking advantage of the predictability of satellite movement. The results of that calculation are formatted as a series of files that are time-ordered. When the timing of a formatted file is satisfied, the user plane generates the command information corresponding to the network topology described by the file. The command

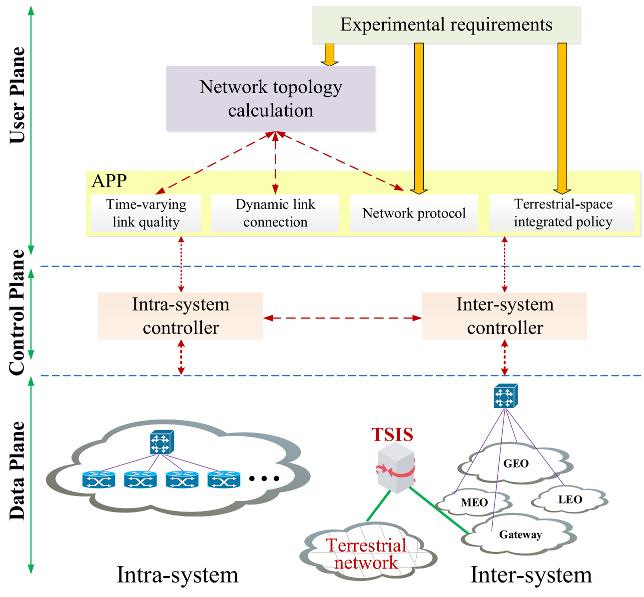


FIGURE 4. Architecture of testbed.

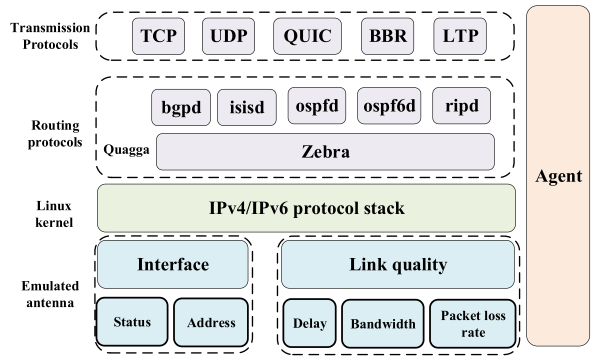


FIGURE 5. Protocol stack of the emulated space router.

information mainly consists of four types: *time-varying link quality*, *dynamic link connection*, *network protocol*, and *terrestrial-space integrated policy*.

B. CONTROL PLANE

The control plane is responsible for implementing commands generated by the user plane into the data plane. All elements in the data plane, including hardware devices and network protocols, are controlled by the control plane. The function of the control plane is undertaken by two types of controllers: the intra-system controller and the inter-system controller. The intra-system controller implements the control commands within the satellite system. The inter-system controller sets the control commands between satellite systems and the integration policy between terrestrial network and space network.

C. DATA PLANE

The data plane is used to emulate the ISTN directly. It is composed of the emulated space network, the terrestrial network, and a terrestrial-space integrated system (TSIS).

The space network is emulated using a collection of PC-based routers as well as programmable switches. PC-based router consists of the *Linux* system and *Quagga* routing software, whose architecture is shown in Fig. 5. *Quagga* is a routing software that implements many routing protocols [11]. These routing protocols are used to interconnect any two space routers. Apart from traditional transport protocols, such as the transmission control protocol (TCP) and the user datagram protocol (UDP), the latest proposed transport protocols (such as Google's BBR [12] and QUIC [13]) are also deployed in the emulated space router. When the testbed conducts the scenarios where end-to-end transmission cannot be guaranteed, the DTN Lickliter transmission protocol (LTP) [14] can be implemented using the software of interplanetary overlay network (ION) [15]. The network cards in emulated space router are used to emulate the antennas of space router. One network card has an interface and represents a satellite's antenna. All network cards are connected with each other using the programmable switch. The *Agent* in emulated space router is designed to receive commands of the controller and to deploy these commands into the corresponding module in such router.

The terrestrial network mainly consists of the terrestrial Internet and 3G/4G mobile communication networks. TSIS not only interconnects with emulated space network and terrestrial network, but also implements the security isolation policies to protect both networks.

D. DISCUSSION

The complexity of the overall system mainly consists of calculating the topology dynamics of space network and emulating the dynamic space network. The first part mainly is the implementation complexity of mathematical models. That is an easy task. The second part is a complicated task that requires the systematic designs for emulating the dynamic space network using the static hardware devices installed with the real network protocol. This complexity mainly consists of the difficulties of the testbed's designs to configure these devices. The architecture of NPVT is similar to the concept of software-defined networking (SDN) [16]. That can simplify the procedures needed to configure these terrestrial devices as the concept of SDN does. But, their design goal or detailed methods are different. The basic concept of SDN is to control the routing between users by deploying the forwarding rules into the programmable switch. The basic idea of designing the testbed is to change the dynamic networking between routers in ISTN, including the link quality and link connections, the deployment of network protocols, and the integration with the terrestrial network.

Since the softwares used in tested such as ns2/STK, Quagga, ION, and Linux system are open-source, the realization cost of testbed is mainly determined with the cost of hardware devices, which is also positively related to the scale of testbed. In any case, the cost of hardware devices is so far smaller than the cost of onboard-processing satellites.

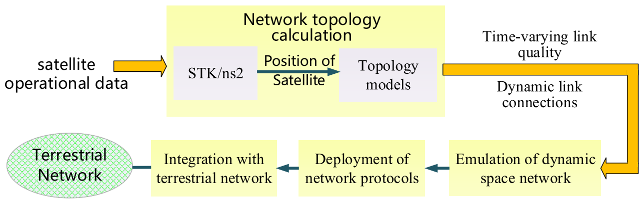


FIGURE 6. The data flow of the testbed.

V. DESIGNS

Fig. 6 shows the data flow among the core designs of testbed. The design of network topology calculation transforms satellite operational data into topology dynamics, including the time-varying link quality and dynamic link connection. Using these topology dynamics, the testbed automatically emulates the dynamic space network, deploys network protocols, and integrates with the terrestrial network.

A. NETWORK TOPOLOGY CALCULATION

In this module, there are two core jobs. One is to calculate the time-varying position of satellite using the satellite parameters. The other is to calculate the link connection and the link quality according to the relative position between satellites.

According to Newton's law of gravitation, the satellite will move in its orbit at the fixed rate of $|V|$ with time-varying direction. The mathematical calculation of satellite position can be simplified as follows:

$$P_c = P_0 + V * t \tag{1}$$

where P_0 and P_c are the initial and current position of satellite respectively, and t is the running time. V represents the speed of satellite movement. There exist some tools to conduct that calculation, such as the satellite module of network simulator 2 (ns2) [17] and satellite tool kit (STK) [18].

Because the satellite movement is predictable and the space environment is simple, the link quality and link connection can be calculated easily. These calculations consist of two steps. The first step is to figure out the visibility relationship between space routers. Then, the time-varying link quality and dynamic link connection are calculated according to these visibility relationships. Some previous works have been proposed to calculate the dynamic link connection, such as WPM-TD (weighted perfect matching based topology discovery) model [19]. The link quality is calculated by three typical channel models, which are the C. Loo model [20], the Corazza model [21], and the Lutz model [22].

Besides, the space network's topology dynamics also can be modeled using real-world trace data. For instance, STK can use the trace data of satellite to detailedly calculate the time-varying position of satellite and visibility relationship between satellites.

B. EMULATION OF DYNAMIC SPACE NETWORK

Emulating the topology dynamics of space network is a primary factor in designing a reliable testbed. The basic idea is to re-arrange the networking relationship of static devices in the testbed when the network topology of the real-world space network changes.

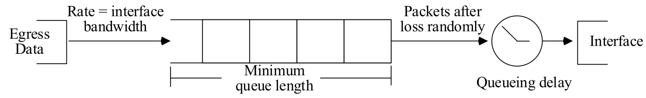


FIGURE 7. Emulation of link quality.

1) TIME-VARYING LINK QUALITY

To achieve the flexibility of emulation without compromising reliability, the time-varying link quality is emulated using the software-based queueing policy and the hardware-based memory in network card.

Each network card deploys a queue (the memory) to store the egress data. When the link quality varies, we rewrite the queueing policy of the queue. The policy can be described with four parameters, namely, queueing delay, packet loss rate, interface bandwidth, and minimum queue length, as shown in in Fig.7.

The first three parameters represent the crucial factors of link quality, which are link delay, packet loss rate, and interface bandwidth, respectively. The last parameter (minimum queue length) is the basis of emulating link quality. The value of minimum queue length is calculated with the equation 2.

$$Minimum\_queue\_length > \frac{max\_delay * max\_bandwidth}{MTU * 8} \tag{2}$$

where $Minimum\_queue\_length$ is the queue length, max_delay is the maximum delay among all possible link delays, and $max\_bandwidth$ is the maximum bandwidth among all possible bandwidths. MTU (Maximum Transmission Unit) is the largest length of the frame. Its default value is 1500 bytes, but we can alter the value according to the experimental requirements.

To avoid the queue overflow when deploying the queueing policy, the minimum queue length should more than its theoretical value. For example, for an emulated link with a delay of 120 ms and a bandwidth of 100 Mb/s, the value of minimum queue length show be set more than 1000. Fig. 8 shows the effect of queue length on the link throughput. With the queue length increasing, the throughput of the emulated link will firstly increase and then remain unchanged. For an emulated link with a delay of 120 ms and a bandwidth of 100 Mb/s, when the value of queue length is 1000 (the theoretical value of minimum queue length), the throughput of the link is very close to the set bandwidth of emulated link. When the value of queue length becomes larger than 1000, the throughput of emulated link will increase slightly and then remain unchanged.

To deploy the queueing policy automatically, we design a *Shell* script to execute the system call of *Linux* system automatically. When the *Agent* in emulated space router receives

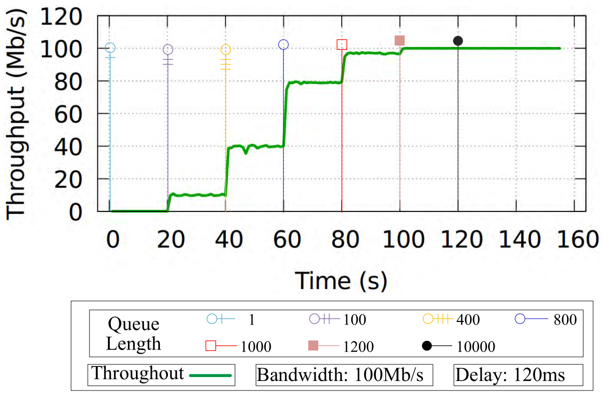


FIGURE 8. Effect of queue length on the throughput of emulated link.

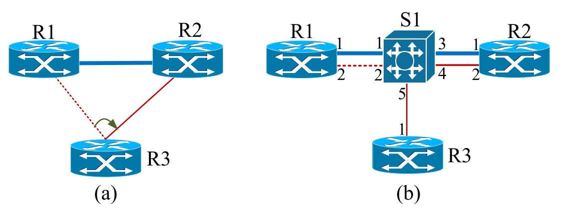


FIGURE 9. Emulation of dynamic link connection.

the command from the control plane, it calls this script to change the link quality.

2) DYNAMIC LINK CONNECTION

To emulate the dynamic link connection, we transform the link connection into the forwarding rule between ports in programmable switch. Every router uses multiple interfaces to access the programmable switch, and every interface represents an antenna. This design is to emulate the fact that a space router adopts many antennas to connect with the same number of space routers. The dynamics of link connection are emulated by deleting/adding the corresponding rule in the programmable switch.

When a link disconnects, the controller deletes the corresponding rule in programmable switch. When a new link reconnects, the new rule is added to the switch. As shown in Fig. 9, when router R3 disconnects with router R1 and connects with router R2, the programmable switch S1 deletes the bidirectional flow rule between S1-2 and S1-5 ports in programmable switch S1 and adds the bidirectional flow rule between S1-5 and S1-4 ports in S1. Because the forwarding rule is a map between two physical ports in the programmable switch, we can emulate the connectivity of link connections in the physical layer easily.

However, the emulation of building a new link may fail if we do not take extra actions. For instance, if an in-use interface in emulated space router (R2-1) or an in-use port in the programmable switch (S1-3) is assigned to accept the new link between R2 and R3, a failure will occur. To avoid failure, we design three tables to transform the logical link connection into the forwarding rule correctly. (1) *Bitmap* records the interface utilization of emulated space router in real time. (2) *Interface-based adjacent matrix* describes the neighboring relationship between interfaces in different emulated space router. That is, replacing '1' in the traditional adjacent matrix with the ID of a spare interface in the *Bitmap*. (3) *Mapping table* builds the one-to-one mapping between the interface of the emulated space router and the port of the programmable switch. It remains unchanged after the testbed is set up. Consequently, replacing interfaces in the *Interface-based adjacent matrix* with ports in the *Mapping table* can generate the correct forwarding rule for building a new link.

Fig. 10 shows an example of the generation process of the forwarding rule when R3 disconnects from R1 and connects with R2. It firstly updates the *Interface-based adjacent matrix* using the information of adjacent matrix and *Bitmap*. In *Interface-based adjacent matrix*, the values of (R1, R3) (corresponding to the link from router R1 to router R3) and (R3, R1) are set 0, the value of (R2, R3) is set the port-ID of R2 (i.e., 2), and the value of (R3, R2) is set the port-ID of R3 (i.e., 1). And then the *Interface-based adjacent matrix* is transformed into the correct forwarding rule using the information of *Mapping table*. The forwarding rules of (R1, R3) and (R3, R1) are set (0,0), and the forwarding rule of (R2, R3) is set (S1-4,S1-5) (from the port-ID S1-4 to the port-ID S1-5) and the value of (R3, R2) is set (S1-5,S1-4) (it is also the adverse forwarding rule of (R2, R3)). Finally, the *Bitmap* is updated according to the old *Bitmap* and *Interface-based adjacent matrix*.

C. AUTOMATIC DEPLOYMENT OF NETWORK PROTOCOLS

The deployment of network protocols consists of two parts, i.e., protocol tuning and protocol maintenance.

Because the experimental scenarios are diverse, there may be various combinations of network protocols for testing in the testbed. To simplify the process of protocol tuning, we install almost all of the network protocols into the emulated satellite in advance. Then, the protocol tuning can be achieved easily by enabling the corresponding protocols. The testbed can afford the test for network protocols from the network layer to the application layer, such as, 6over4, NDP (Neighbor Discovery Protocol) [23], OSPF (Open Shortest Path First Interior Gateway Protocol) [24], BGP (Border Gateway Protocol) [25], ICMPv6, and DHCPv6 at the network layer; TCP, UDP, BBR and QUIC at the transport layer; HTTPs, FTP and SSH at the application layer.

The protocol maintenance is used to make the network protocols in different emulated space routers work together. Specifically, it re-deploys the networking relationships of network protocols when the network topology of real-world space network changes. To achieve the automatic deployment of the network protocols, we implement a *python* script in the *Agent* of emulated space router. This script parses and deploys the command information from the user plane. It changes its working environment to interact with the network protocols based on the received command information. To allow the

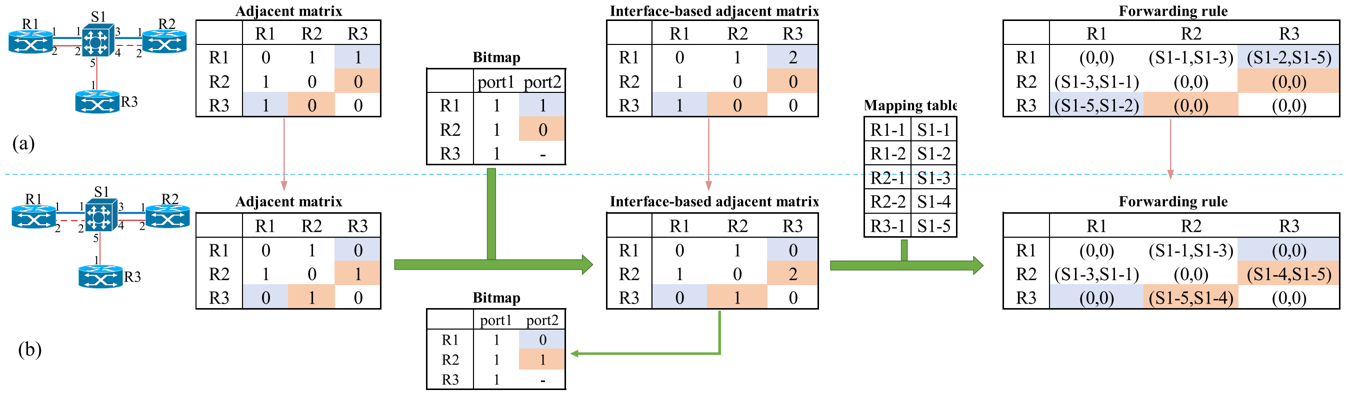


FIGURE 10. Generation precess of forwarding rule.

interaction with diverse network protocols, the command information is formulated as follows.

ptype	*working*	*cmd*	*cmd*	$\cdots$

where *ptype* indicates the protocol type (such as NDP, OSPF, BGP, TCP, BBR, and QUIC), *working* represents the working environment for conducting the configurations of *ptype*, and *cmd* is the command of configurations. The *python* script uses the information of *ptype* and *working* to change its working environment, and then conduct the content of *cmd*.

D. INTEGRATION WITH TERRESTRIAL NETWORKS

The TSIS (terrestrial-space integrated system) is designed to interconnect the emulated space network with the real terrestrial networks. The architecture of TSIS is shown in Fig. 11, where the eBGP (External Border Gateway Protocol) session is used to interconnect both the emulated space network and the real terrestrial networks (Internet and mobile communication network).

The integration of emulated space network and the operational terrestrial networks mainly relies on the interconnection between private AS and public AS. The emulated space network acts as a private AS, but it is assigned to a provider-independent IPv6 address space. When it advertises its IPv6 prefixes to terrestrial networks, TSIS will strip the private AS ID of these prefixes away from the AS_PATH attribute. The upstream ASes in terrestrial networks will learn these prefixes but do not ware the existence of private AS. On the contrary, the emulated space network will learn the AS_PATH attribute of public ASes in the terrestrial networks.

The terrestrial Internet is deployed using the operational next-generation Internet (CERNET2), and the mobile communication network is deployed using the commercial 3G/4G access network of cellular network and emulated core network. The PDN GateWay (PGW) of cellular network interconnects with the TSIS system. The TSIS system acts as the core network of cellular network and the integration point of cellular network and emulated space network. Because the

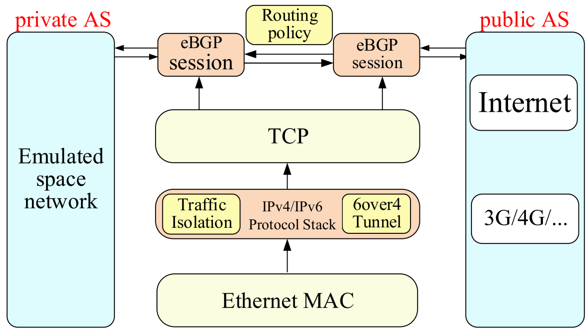


FIGURE 11. Architecture of TSIS.

testbed is a pure IPv6 network and the commercial 3G/4G access network is almost IPv4 network, the intra-site automatic tunnel addressing protocol (ISATAP) [26] is used to transmit IPv6 packets of the user over the 3G/4G access network to the emulated space network. ISATAP is an automatic IPv6 transition mechanism meant to transmit IPv6 packets of the user over an IPv4 network.

Furthermore, it is non-trivial to coordinate the time progress in real networks and emulated space networks. The Network Time Protocol (NTP) is used to maintain the time progress of every device in the emulated space network. Then the time clock of emulated space network keeps pace with that of real networks.

When deploying some emulations related to the integration of terrestrial network and space network, the testbed can use the inter-system controller to change the terrestrial-space integrated policies, such as ACL, 6over4, and traffic isolation.

VI. TESTBED VALIDATION

A. TESTBED SETUP

1) THE LAYOUT OF TESTBED

In this section, we provide an overview of the testbed, which is built using the general hardware devices and open-source softwares.

Fig. 12 shows a representation of the layout of the testbed. The emulated space network consists of three GEO satellites,

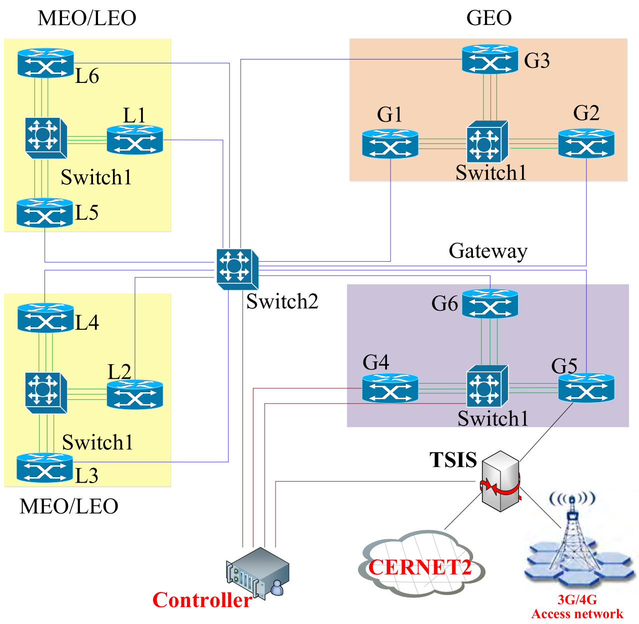


FIGURE 12. Topology of testbed.

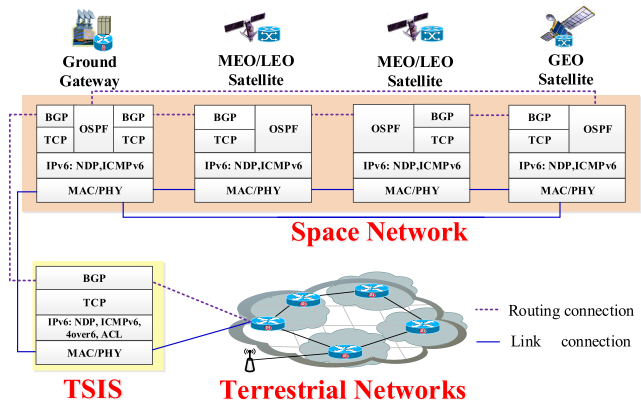


FIGURE 13. Example of network protocol implementation.

three ground gateways, and six MEO/LEO satellites. The satellite or ground gateway is emulated by a Dell PowerEdge server with eight Intel Pro 1Gbps network cards. Each server installs with the open-source *Linux* system and *Quagga* routing software. Each router is equipped with eight network cards. Three network cards are used to emulate intra-system links, four network cards are used to emulate inter-system links, and the last network card is used to connect with the controller.

To emulate the dynamic link connection, we use two 48-port SDN switches to interconnect these emulated space routers. The SDN switches are represented by *switch1* and *switch2* in Fig. 12. *switch1* is used to interconnect space routers within the satellite system, and *switch2* is used to interconnect space routers between different systems. Two RYU controllers control two switches respectively. The controller also is scaled to control all emulated space routers. Controllers and emulated space routers are interconnected using a normal switch *switch*3. For brevity, some link connections are not shown in this figure. Controllers are implemented in two Lenovo Thinkpad PCs. The time synchronization for all nodes is based on the Network Time Protocol (NTP).

Additionally, we deploy TSIS to achieve backbone-level integration with the real terrestrial network CERNET2 [5] and 3G/4G access network. CERNET2 is a large-scale and operational testbed with pure IPv6. The interconnection between testbed and CERNET2 is a backbone-level relationship, instead of the network access, which means that routing protocols deployed in the emulated space network and CERNET2 can share routing information with each other. The 3G/4G access network is commercial ZTE equipment, which is a pure IPv4 network.

2) NETWORK PROTOCOL TUNING

Fig. 13 shows an example of network protocol tuning. In each router, the TCP/IPv6 protocol stack, NDP, OSPF, and BGP are configured. The TCP/IPv6 protocol stack is used to support the implementation of other protocols. ICMPv6 is a management protocol for sharing the control message. ICMPv6, NDP, OSPF, and BGP are the core network protocols used to interconnect the whole network. NDP is used to build a direct networking relationship between two neighboring routers. OSPF is applied to construct the networking relationship of all space routers within the same satellite system, while BGP is used to build the networking relationship between two space routers that belong to different systems. To focus on the test of IP and related network protocols, the functions of AOS and IP over CCSDS are performed using the Ethernet. In the future, we will add these protocols to the protocol stack of space router.

3) SATELLITE OPERATIONAL DATA

We use real climate observation data and satellite parameters to conduct emulations.

The observation data is collected by NSMC (National Satellite Meteorological Center) [27] using Chinese satellites such as the FÄŞngynÃžn (FY) series, or rented satellites such as NOAA (USA), MTSAT (USA), etc. The collected data is used to analyze weather events. Taking the weather event of a typhoon as an example, the related data is collected by three GEO satellites (FY2D, FY2E, FY2F) and two LEO satellites (FY3C, NOAA). The satellite parameters are given in Table 1. Three ground gateways, which are located at Beijing, Guangzhou, Urumqi, are used to receive satellite data and convey the data to the terrestrial network. Fig. 14 shows a snapshot of the space network. The green thick line represents the visibility relationship between climate satellites, and the pink thin line represents the orbit of climate satellite.

In the testbed, three GEO satellites (FY2D, FY2E, FY2F), two LEO satellites (FY3C, NOAA) and three ground gateways (Beijing, Guangzhou, Urumqi) are emulated using G1, G2, G3, L1, L2, G5, G4, and G6, respectively. All satellites

TABLE 1. Parameters of climate satellites.

Satellite	Type	Inclination (°)	Right ascension of ascending node (°)	Argument of perigee (°)	Altitude (km)	Data Size	Generation interval of data (min)
FY2D	GEO	4	65.5	313.4	42151	xx	60
FY2E	GEO	1.8	65	212	42151	xx	60
FY2F	GEO	1.3	77	226	42151	xx	60
FY3C	LEO	98.7	21	73	7207	xx	41
NOAA	LEO	99.2	323	337	7225	xx	25

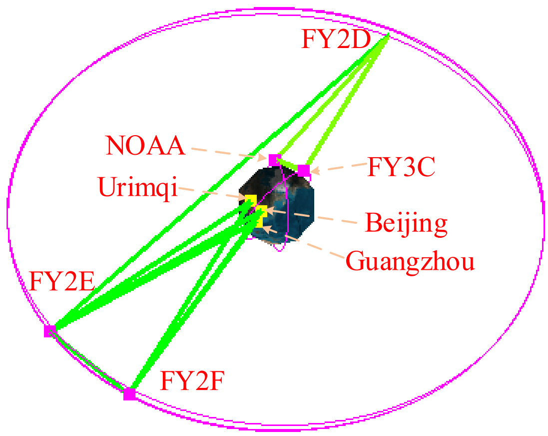


FIGURE 14. A snapshot of space network.

and ground gateways are orchestrated into the networking mode. That is, all satellites and ground gateways can build the link connection if the link condition is allowed. In the networking mode, satellites can send their observation data in almost real time. Two PCs, which are attached to FY3C and NOAA satellites, respectively, are used to emulate the sending of observation data. One PC, which is implemented in the campus wireless network of Tsinghua University, is used to emulate the receiving of observation data. Both the campus network and the testbed are interconnected via the global terrestrial network indirectly. The observation data is sent from PC to server using a reliable protocol, i.e. TCP.

B. BASELINE VALIDATION

In this section, we validate the basic functions of the testbed, by exploring the emulations of real climate satellite data. The resulting data demonstrates that the testbed is capable of emulating the dynamic space network and integrating with the terrestrial network.

1) EMULATION OF THE DYNAMIC SPACE NETWORK

The correctness and timeliness of emulating dynamic space network are the critical factors for validating the accuracy of testbed. If the network topology of testbed is very different from that of the real-world space network, the testbed cannot be qualified for its job. If the timeliness is not guaranteed, the networking of emulated space network in testbed falls behind the topology of the real-world space network.

Corresponding to the designs of the testbed, the emulation of the dynamic space network has the following main aspects, namely, the emulation of time-varying link quality, the emulation of dynamic link connectivity, and the re-deployment of

TABLE 2. Emulation error (%) of link quality.

	25^{th}	50^{th}	75^{th}	95^{th}
Interface bandwidth	0.01	0.03	0.07	0.12
Link delay	0.15	0.17	0.19	0.24
Packet loss rate	0.70	1.56	2.88	6.35

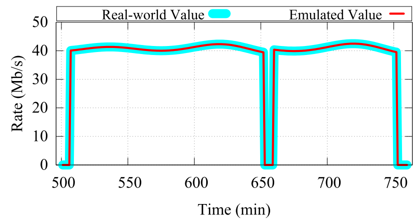


FIGURE 15. Emulation of interface bandwidth.

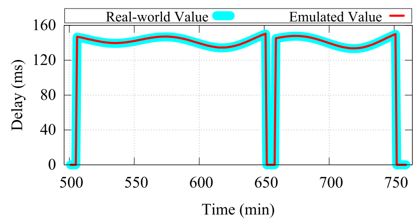


FIGURE 16. Emulation of link delay.

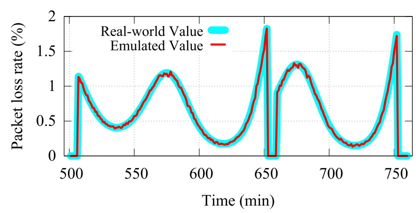


FIGURE 17. Emulation of link loss.

network protocols when network topology changes. In detail, the emulation of link quality consists of emulations of interface bandwidth, link delay, and packet loss rate.

Correctness: For the validation tests, the inter-system links between FY3C and GEO satellites are selected. We observe the emulation result during the experiment time from 500 min to 760 min.

Time-varying link quality. Figs. 15, 16 and 17 show the emulation results of interface bandwidth, link delay, and packet loss rate. The line of emulated value overlaps with the line of the real-world value that is calculated using the module of *Network Topology Calculation*. That means both values are almost the same.

TABLE 2 shows the 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentiles of emulation error of link quality. Results indicate that all of the emulation errors are very small. The emulation error of loss packet rate is slightly high. That is mainly because, its real-world value is very small, and the small amount of packet loss will increase the error rate.

TABLE 3. Changes of path from FY3C to terrestrial network.

Time (min)	Link connectivity	Path	Status
500	link (L1, G3) disconnects	L1, L2, G3, G2, G5, TSIS	---
507	link (L1, G1) reconnects	L1, G1, G4, G5, TSIS	Changed
550	link (L2, G3) disconnects	L1, G1, G4, G5, TSIS	Unchanged
653	link (L1, G1) disconnects	L1, L2	Unreachable
660	link (L1, G3) reconnects	L1, G3, G2, G5, TSIS	Changed
711	link (L2, G2) reconnects	L1, G3, G2, G5, TSIS	Unchanged
745	link (L2, G2) disconnects	L1, G3, G2, G5, TSIS	Unchanged
754	link (L1, G3) disconnects	L1, L2	Unreachable

Dynamic link connectivity and network protocol. The result of routing information is used to describe the correctness of the emulation of dynamic link connectivity and the deployment of network protocols. When the link connectivity of the real-world space network changes, the link connectivity in the testbed will change and the network protocol will re-deploy correspondingly, and then the network protocol will automatically calculate the new routing information corresponding to the new link connectivity. Only when both link connectivity and routing protocol in the testbed work properly, the routing information matches with the real-world link connectivity.

TABLE 3 shows changes of the path from FY3C (L1) to terrestrial network. The result data indicates that network protocols can calculate the shortest path when link connectivity changes. When L1 connects with the GEO satellite (G1, G2, or G3), the network protocol will select the path passing link between L1 and the GEO satellite, owing to this path being the shortest. When L1 disconnects from the GEO satellite, network protocol will select the path passing link between L1 and L2, owing to this path being reachable. When both L1 and L2 disconnect from the GEO satellite, the network protocol will only calculate the path between L1 and L2, owing to the path from L1 to terrestrial network being unreachable.

Timeliness: The timeliness is evaluated with the deployment time of the commands to emulate *time-varying link quality*, *dynamic link connection*, and to re-deploy *network protocol*.

Table 4 presents the 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentiles of deployment time. Results indicate that the deployment time is very small, ranging from a few milliseconds to tens of milliseconds. The timeliness can be competent for emulating the dynamic space network. There are three reasons for that: (1) Since the link quality only varies a little in the real-world space network during a short time, the emulation of *time-varying link quality* brings little distortion. (2) Compared with the time of building a new link in real-world satellites (approximately several tens of seconds), the deployment time of *dynamic link connection* in testbed is very short and can be ignored. (3) The time for the network protocol to discover the topology changes (the time of network convergence) is several seconds or tens of seconds. The deployment time of *network protocol* is also negligible.

TABLE 4. Deployment time (ms) of emulating dynamic space network.

	25^{th}	50^{th}	75^{th}	95^{th}
Link quality	21.8	30.9	55.3	65.2
Link connection	3.2	5.3	6.6	13.1
Routing protocol	14.9	14.8	14.9	16.6

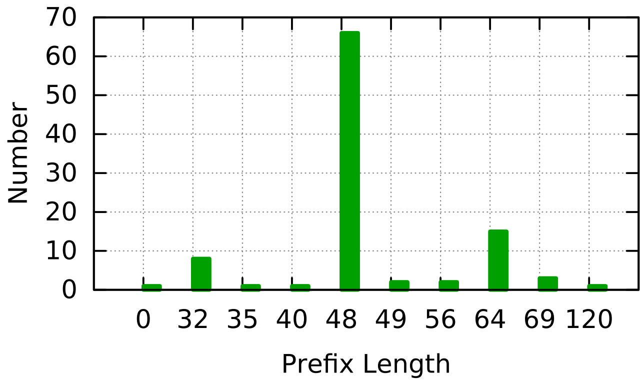


FIGURE 18. Distribution of IPv6 prefixes in space router.

2) INTEGRATION WITH TERRESTRIAL NETWORK

We validate the integration between emulated space network and terrestrial network, by observing the routing information in each emulated space router and TSIS. If that backbone-level integration is achieved, the terrestrial network and space network can share the routing information with each other.

Fig. 18 shows that every emulated space router learns 100 IPv6 prefixes of the terrestrial network. The routing information indicates that the backbone-level integration with both networks is achieved. Compared with the IPv6 prefixes in the core router of terrestrial network (approximately tens of thousands), the IPv6 prefixes learned by the emulated space network are very small-scale. That is because many inactivated prefixes of the terrestrial network are aggregated into the 0-bit prefixes (:: /0).

C. SYSTEMATIC VALIDATION

In this part, we conduct a systematic validation on the efficiency of the testbed using the data transmissions of FY satellites.

To analyze the data transmission, we introduce the non-networking mode as a comparison object of the networking mode. In the non-networking mode, the satellite sends its observation data to the ground only when the satellite passes over the gateway. Because the satellite moves relative to the gateway, it cannot always communicate with ground gateway, which results in the latency of several hours for sending the data until it moves into the coverage area of a gateway. The testbed orchestrates the dynamic network topologies of space networks in two modes using the same devices. Meanwhile, the emulated space networks of two modes interconnect the terrestrial network using TSIS.

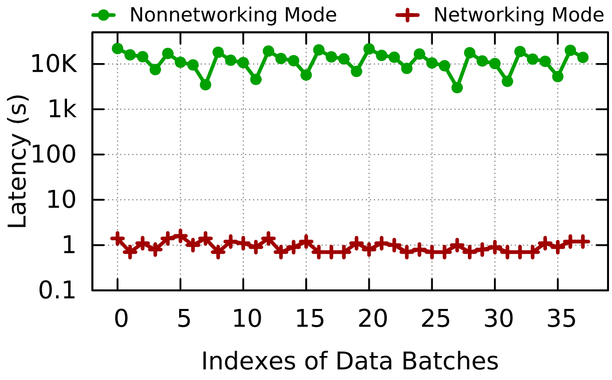


FIGURE 19. Transmission delay of FY3C's data batches.

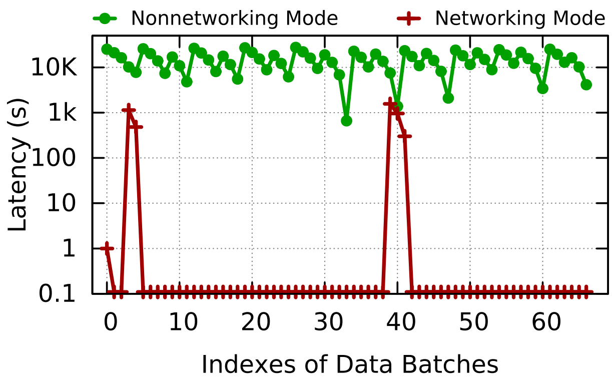


FIGURE 20. Transmission delay of NOAA's data batches.

Figs. 19 and 20 reveal an obvious common characteristic that the latency in networking mode is reduced by more than an order of magnitude compared with that in the non-networking mode. The latency of each data batch is reduced from several hours to one second. We notice that there exist some points where the latency of NOAA's data batches is approximately 1000 s. That is because, when the observation data is generated, the NOAA satellite is not in the coverage area of GEO satellites or gateways and then the NOAA satellite has to wait for a while to send data.

The success of data transmission presents the reliable validation of the testbed. Firstly, the orchestration of two modes validates that the testbed can arrange different dynamic space networks and emulate their dynamics. Secondly, the result that networking mode dramatically decreases the latency of sending data, proves that the deployment of network protocols can calculate the up-to-date path among multiple routers. Finally, the successful communication between users in space network and terrestrial network shows that the integration between space network and terrestrial network is achieved.

D. TESTBED SCALABILITY

For this test, we are interested in the scalability of the testbed. The scalability is analyzed from the viewpoints of testbed's data plane and control plane.

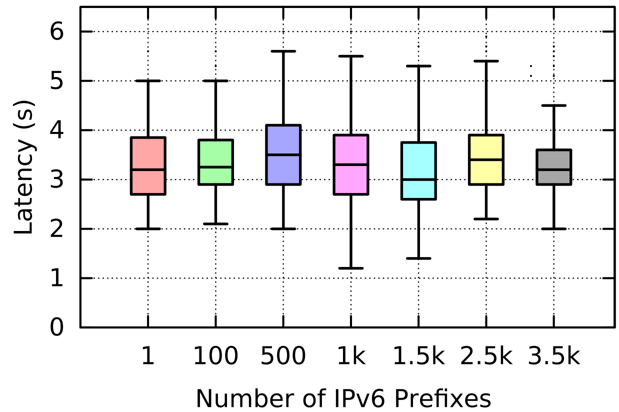


FIGURE 21. Distribution of transmission delays of FY3C's data.

1) SCALABILITY OF THE DATA PLANE

In the data plane, the network scale can be increased by installing more hardware devices. However, the cost of doing this is very high. In this test, the scale of the routing table in every space router is used to validate the scalability of the data plane.

The scale of routing table is a critical factor of network scale. It increases as the network scale increases. A larger routing table means that the router needs more resources to determine an efficient route entry when it forwards a packet. If the scale of routing table exceeds the resource limitation of the router, some abnormal phenomenon that degrades the performance of data transmission will occur. This includes an increase in the time consumed for routing table lookup, the loss of some route entries, and even some routers go down. In other words, if the data transmission suffers from severe performance degradation when we increase the number of announced prefixes, then the testbed cannot be scaled to support a large number of satellites.

When we increase the scale of the routing table, we observe the data traffic of FY3C and NOAA satellites. Both satellites send all the observation data at one time. This configuration allowed us to exclude the influence of topology changes of both space network and terrestrial network. The number of IPv6 prefixes announced by the space network is set to 1, 100, 500, 1000, 1500, 2500, and 3500. All space routers will learn these announced prefixes and install them into the routing table.

Fig. 21 presents the minimum, the first quartile, the second quartile (the median), and the third quartile as well as the maximum of FY3C's transmission delays. The data indicate that announcing a large number of network prefixes does not increase transmission delay. Besides, the success of data transmission implies that no route entry is lost, and no router is down.

We also conducted a similar test with NOAA's data shown in Fig. 22. Results also indicated that the transmission performance is not affected. That means the testbed can be capable of emulating the space network with thousands of routing entries. If a space router has eight network prefixes

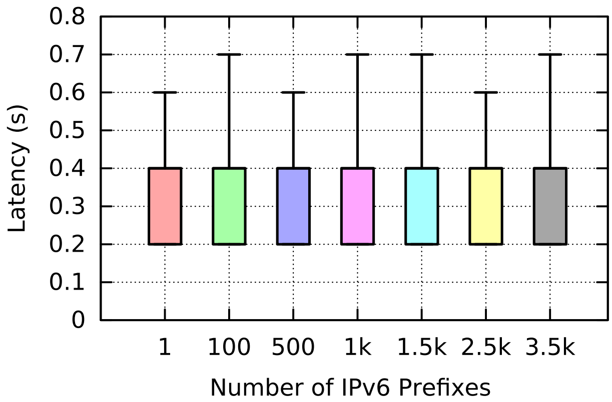


FIGURE 22. Distribution of transmission delays of NOAA's data.

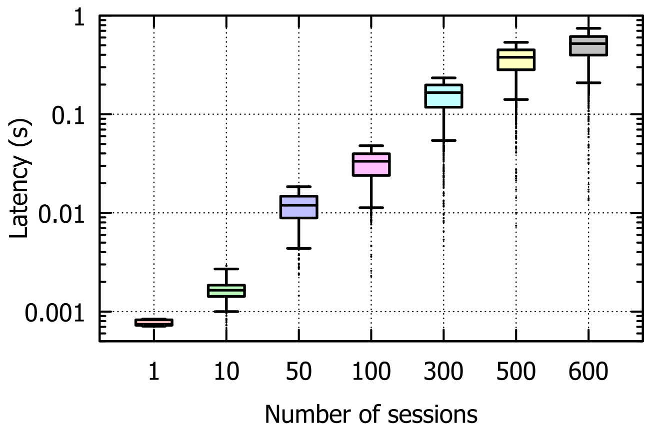


FIGURE 23. Effects of the number of emulated satellites.

(each network card is assigned a prefix), the data plane of the testbed can emulate at least 437 space routers.

2) SCALABILITY OF THE CONTROL PLANE

In the control plane, there exists one central mechanism, i.e., the controller. With the scale of the data plane increasing, the latency of the controller implementing commands may increase.

To test the performance of the controller, we use an inter-system controller and the emulated FY3C satellite. The controller applies many parallel TCP sessions to communicate with FY3C satellite. One TCP session represents a controlled space router. The number of sessions is set to 1, 10, 50, 100, 300, 500, and 600. We record the latency of each session to send and conduct commands.

Fig. 23 shows that the transmission delay of commands is minimal even when the number of parallel sessions reaches several hundred. However, the latency of a single controller increases as the number of emulated space routers does. Therefore, when the scale of the testbed is very large, the implementation of multiple collaborative controllers may be a good choice.

VII. RELATED WORKS

Currently, some researchers have proposed to build the testbed for emulating space network or terrestrial network.

A. TESTBED FOR SATELLITE PLATFORM / USER ACCESS

Marchese and Perrando [28] have proposed a testbed for emulating the packet switched satellite. The testbed adopts a central emulator (called the Elaboration Unit), to emulate the satellite to support the forwarding of IP packets. However, the interconnection between emulated satellite and the gateway is static, and the routing protocols are not deployed in their emulate satellite. Dubois *et al.* [29] proposed an OpenSAND platform for the DVB-RCS satellite system (Digital Video Broadcasting-Return Channel via Satellite), which mainly emulates the multi-beam of the satellite and the access between multiple ground gateways and a satellite. Alphand *et al.* [30] also designed a SATIP6 emulated satellite platform to support the implementation of IPv6 in the DVB-RCS satellite system.

Guo *et al.* [31] proposed an Inter-planetary emulator [31] based on the NETShaper emulation tool, which provides an emulation layer between the network card and Linux's IP stack to emulate the link quality. The emulation method of link quality is similar to our method of queueing policy, but the time-varying feature of link quality is not considered in their designs. Chertov *et al.* [32] proposed an MSET designed for emulating the Time Division Multiple Access (TDMA) between a satellite and multiple terminals. The testbed creates a TermAgent Click element in the emulated satellite to emulate the framing schedule. Authors from Aerospace Corporation [33] designed a space network testbed (SNTB) to test the behavior of BGP in a local-area network (LAN) composed of the emulated satellite and multiple terminals. The mobility between satellite and terminals is emulated by changing the neighboring relationship among BGP peers, instead of the real link connection/disconnection.

The main difference between our testbed and the above testbeds is that our testbed is designed for the integrated network of space network and terrestrial network. The existing space network testbeds mainly focused on fields of satellite platform [28]–[30] or user access between a satellite and ground terminals [31]–[33]. Their considered scenarios work like the user access of mobile communication network, as shown in Fig. 2 (b), in which the role of satellite network is an access network of the terrestrial network. Besides, the networking of multiple satellites in their testbed is not considered deeply.

B. TESTBED FOR SPECIFIED NETWORK PROTOCOLS

At present, some articles have presented the dedicated platforms to validate their proposed network protocols in space network. Li *et al.* [34] proposed a KVM [35] based platform to validate a proposed software-defined architecture for ISTN. The platform is designed by mapping the visibility relationship between space routers (calculated by STK) to the emulated space network consisting of virtual machines. However, the platform mainly emulates a snapshot of the space network, and the network topology was almost static. In addition, the visibility relationship calculated by STK is not the space network topology. The transform from visibility relationship to space network topology requires further modeling using the movement pattern of the satellite. Some previous works [19], [36] also have proposed a platform based on Mininet [37] to test their proposed network protocols. The platform emulates the dynamic space network by changing the interface address or the neighbor relationships of the network protocol. However, because the design of this platform interferes with the working state of the network protocol, the platform can only test some network protocols that are transparent to the network layer, such as BGP.

Besides, the above two platforms are based on virtualization technology to emulate the space network. Although the virtualization technology can simplify the design of the testbed, it limits the testbed's capability. For instance, when network protocols in multiple routers send or process packets simultaneously, the abnormal packet loss will occur, owing to the instantaneous load of the host being very high. Therefore, these testbeds are mainly qualified for testing special network protocols.

Last but not least, the existing testbeds have not achieved an efficient integration with the terrestrial network.

C. TESTBED FOR TERRESTRIAL NETWORK

Designing a testbed for the terrestrial network has been studied fully, such as GENI [38] and Emulab [39]. GENI is an open infrastructure for at-scale networking and distributed systems research and education that spans the US [38]. Emulab is designed to support different networking scenarios over the same hardware through consistent use of virtualization and abstraction. When the network scenario is built, the network topology will remain almost static. Because the network topology of the space network constantly changes, these testbeds cannot emulate the dynamics of space network directly.

VIII. CONCLUSION

In this paper, we have presented a high-fidelity testbed for ITSN, named NPVT. The testbed was designed to provide a near-real environment to validate diverse network protocols in ITSN. In this testbed, the complicated and dynamic space networks can be emulated reliably, diverse network protocols can be deployed, and the integration between itself and the terrestrial next-generation operational network (CERNET2) and the mobile communication network was achieved. We validated the efficiency and scalability of the testbed using real climate satellite data. As future work, the onboard-processing satellite will be integrated into the testbed to enrich onboard experiment scenarios.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their valuable comments. They would also like to thank L. Jun, C. Yazheng, and Z. Qi from Tsinghua University for joining the discussion and writing of this paper.

REFERENCES

[1] *Iridium Next: The Bold Future of Satellite Communications*. Accessed: Jan. 2019. [Online]. Available: https://www.iridium.com/network/iridiumnext
[2] *Inmarsat and Cisco Create Satellite Services Alliance*. Accessed: Jan. 2019. [Online]. Available: https://www.inmarsat.com/news/inmarsat-and-cisco-create-satellite-services-alliance/
[3] *Oneweb Satellite Constellation*. Accessed: Jan. 2019. [Online]. Available: https://en.wikipedia.org/wiki/OneWeb_satellite_constellation
[4] *5G; Study on Scenarios and Requirements for Next-Generation Access Technologies, Version 14.2.0, Release 14*, document TR 38.913, ETSI, Sophia Antipolis, France.
[5] J. Wu, J. H. Wang, and J. Yang, "CNGI-CERNET2: An IPv6 deployment in China," *Acm Sigcomm Comput. Commun. Rev.*, vol. 41, no. 2, pp. 48–52, Apr. 2011.
[6] *The Consultative Committee for Space Data Systems*. Accessed: Jan. 2019. [Online]. Available: https://public.ccsds.org/default.aspx
[7] *AOS Space Data Link Protocol. Issue 3. Recommendation for Space Data System Standards (Blue Book)*, Standards CCSDS 732.0-B-3, Washington, DC, USA, CCSDS, 2015.
[8] *IP over CCSDS Space Links. Issue 1. Recommendation for Space Data System Standards (Blue Book)*, Standards CCSDS 702.1-B-1, Washington, DC, USA, CCSDS, 2012.
[9] S. Burleigh *et al.*, "Delay-tolerant networking: An approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.
[10] J. Mitchell, *Autonomous System (AS) Reservation for Private Use*, document RFC 6996, Internet Engineering Task Force (IETF), Fremont, CA, USA, 2013.
[11] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," *IEEE Netw.*, vol. 28, no. 2, pp. 42–48, Mar./Apr. 2014.
[12] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, pp. 58–66, 2017.
[13] A. Langley *et al.*, "The QUIC transport protocol: Design and internet-scale deployment," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 183–196.
[14] S. B. M. Ramadas and S. Farrell, *Licklider Transmission Protocol— Specification*, document RFC 5326, Internet Engineering Task Force (IETF) Network Working Group, Fremont, CA, USA, 2008.
[15] S. Burleigh, "Interplanetary overlay network: An implementation of the DTN bundle protocol," in *Proc. 4th IEEE Consum. Commun. Netw. Conf.*, Jan. 2007, pp. 222–226.
[16] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.
[17] *The Network Simulator—NS-2*. Accessed: Jan. 2019. [Online]. Available: https://www.isi.edu/nsnam/ns/
[18] *Systems Tool Kit (STK)*. Accessed: Jan. 2019. [Online]. Available: https://www.spacefoundation.org/what-we-do/space-certification/products/systems-tool-kit-education
[19] Z. Yang, H. Li, Q. Wu, and J. Wu, "Topology discovery sub-layer for integrated terrestrial-satellite network routing schemes," *China Commun.*, vol. 15, no. 6, pp. 42–57, Jun. 2018.
[20] C. Loo, "A statistical model for a land mobile satellite link," *IEEE Trans. Veh. Technol.*, vol. 34, no. 3, pp. 122–127, Aug. 1985.
[21] G. E. Corazza and F. Vatalaro, "A statistical model for land mobile satellite channels and its application to nongeostationary orbit systems," *IEEE Trans. Veh. Technol.*, vol. 43, no. 3, pp. 738–742, Aug. 1994.
[22] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, and W. Papke, "The land mobile satellite communication channel-recording, statistics, and channel model," *IEEE Trans. Veh. Technol.*, vol. 40, no. 2, pp. 375–386, May 1991.
[23] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, *Neighbor Discovery for IP Version 6 (IPV6)*, document RFC 4861, Internet Engineering Task Force (IETF) Network Working Group, Fremont, CA, USA, 2007.
[24] R. Coltun, D. Ferguson, J. Moy, and A. Lindem, *OSPF for IPV6*, document RFC 5374, Internet Engineering Task Force (IETF) Network Working Group, Fremont, CA, USA, 2008.
[25] Y. Rekhter, T. Li, and S. Hares, *A Border Gateway Protocol 4 (BGP-4)*, document RFC 4271, Internet Engineering Task Force (IETF) Network Working Group, Fremont, CA, USA, 2006.
[26] D. T. F. Templin and T. Gleeson, *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*, document RFC 5214, Internet Engineering Task Force (IETF) Network Working Group, Fremont, CA, USA, 2008.

[27] *National Satellite Meteorological Center China Meteorological Administration*. Accessed: Jan. 2019. [Online]. Available: http://www.nsmc.org.cn/en/NSMC/Home/Index.html

[28] M. Marchese and M. Perrando, "A packet-switching satellite emulator: A proposal about architecture and implementation," in *Proc. ICC*, Apr./May 2002, pp. 3033–3037.

[29] E. Dubois *et al.*, "Opensand: Open-source satcom emulator," in *Proc. Kaconf*, 2017, pp. 1–4.

[30] O. Alphand, P. Berthou, T. Gayraud, S. Josset, and E. Fromentin, "SATIP6: Satellite testbed for next generation protocols," in *Proc. Broadband Satellite Commun. Syst. Challenges Mobility*, 2004, pp. 53–62.

[31] Z. Guo, B. Malakooti, K. Bhasin, and A. Holtz, "Design of accurate and efficient network emulation systems with application to inter-planetary networks," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Apr. 2006, pp. 895–900.

[32] R. Chertov, D. Havey, and K. Almeroth, "MSET: A mobility satellite emulation testbed," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[33] B. Etefia, V. Swaminathan, J. Train, and J. Hant, "Emulating a space-based router," in *Proc. IEEE Aerosp. Conf.*, Mar. 2010, pp. 1–14.

[34] T. Li, H. Zhou, H. Luo, and S. Yu, "SERvICE: A software defined framework for integrated space-terrestrial satellite communication," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 703–716, Mar. 2018.

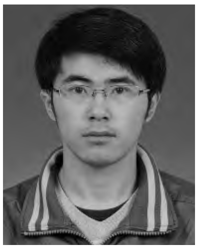
[35] *Kernel Virtual Machine*. Accessed: Jan. 2019. [Online]. Available: http://www.linux-kvm.org

[36] Z. Yang, H. Li, Q. Wu, and J. Wu, "Analyzing and optimizing BGP stability in future space-based internet," in *Proc. IEEE 36th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2017, pp. 1–8.

[37] B. Lantz, B. Heller, and N. Mckeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, Oct. 2010, Art. no. 19.

[38] *Geni*. Accessed: Jan. 2019. [Online]. Available: https://www.geni.net/

[39] *Emulab*. Accessed: Jan. 2019. [Online]. Available: https://www.emulab.net/portal/frontpage.php



ZENGYIN YANG received the B.S degree from Xidian University, Xi'an, China, in 2014. He is currently pursuing Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing. His current research interests include routing, the next-generation Internet architecture, network protocols, testbed, and integrated space-terrestrial networks.

HEWU LI received the M.S. and Ph.D. degrees in computer science from Tsinghua University, in 2001 and 2004, respectively. He is currently an Associate Professor with the Institute for Network Sciences and Cyberspace, Tsinghua University, and the Director of the Wireless and Mobile Network Research Laboratory, Network Research Center. His research interests include mobile wireless network architecture, integrated space-terrestrial networks, testbed, broadband wireless access technology, and mobility architecture in next-generation networks.

QIAN WU received the M.S. and Ph.D. degrees in computer science from Tsinghua University, in 2002 and 2006, respectively, where she is currently an Associate Professor with the Institute for Network Sciences and Cyberspace. Her research interests include the next-generation Internet architecture and protocols, integrated space-terrestrial networks, mobile and wireless networks, multipath transfer, and mobile multicast.

JIANPING WU (F'11) was also the Chairman of Asia Pacific Advanced Network, from 2007 to 2011. He is currently a member of the Chinese Academy of Engineering and a Professor with the Department of Computer Science, Tsinghua University, where he is also serving as the Chairman of the Department of Computer Science and the Dean of the Institute for Network Sciences and Cyberspace. He is also serving as the Director of the Network Center and the Technical Board of China Education and Research Network, and the National Engineering Laboratory for Next Generation Internet, a member of the Advisory Committee of the National Information Infrastructure for Secretariat of State Council of China, and the Vice President of the Internet Society of China. He has devoted himself to the research of computer network including technology research, network engineering, and cultivating talents for many years. He has led his team to do an in-depth study of network design, network engineering, network core equipment development, and network architecture.