Evaluating IP Routing on a General Purpose Network Emulation Platform for Space Networking

Jiawei Sun[1], Peng Zhou[2], Kanglian Zhao[1], and Wenfeng Li[1](✉)

[1] Institute of Space-Terrestrial Intelligent Networks, Nanjing University, Nanjing, China
2711025503@qq.com, zhaokanglian@nju.edu.cn, leewf_cn@hotmail.com
[2] Advanced Institute of Information Technology, Peking University, Beijing, China
pzhou@aiit.org.cn

Abstract. Driven by lacking of a low-cost, large-scale, flexible and reconfigurable general purpose network emulation platform for space networking, ISTIN laboratory of Nanjing University designed a flexible and fully reconfigurable space network emulation platform supporting emulation of multiple network architectures and network protocols. We have carried out some space network protocol research based on the platform, such as DTN and IP. This paper constructs a spatial network scenario to evaluate OSPF and OSPF+ routing performance on this platform The emulation results indicate the emulation platform has the ability of IP routing emulation, which guarantees considerable reliability and provides a reference for subsequent emulation work of IP protocols.

Keywords: Space Information Networks · Network emulation · OSPF · Quagga

1 Introduction

Space Information Networks (SIN) are the network systems based on space platforms, such as geostationary (GEO) satellites, medium earth orbit (MEO) satellites, low earth orbit (LEO) satellites, stratospheric balloons, etc., which provide services such as communication, positioning and navigation. Due to the enlargement of network scale and diversity of demands, the emulation platform for SIN research needs to be extensible, flexible and reconfigurable.

So far, there exist two schemes for SIN research. One is to carry out research in the real space network, but the experiment cost is relatively high The other scheme is to use existing network simulators (such as OPNET [1] or NS [2], etc.) for simulation. Compared with on-board experiments, software simulation is easy to implement, but it cannot generate real data flow between nodes. Hence, the reliability of the software simulation cannot be guaranteed.

In order to carry out space network research, the Institute of Space-Terrestrial Intelligent Networks (ISTIN) of Nanjing University developed a scalable network emulation platform for space internetworking [3]. On the basis of retaining reliability, the platform

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 311–320, 2021.
https://doi.org/10.1007/978-3-030-69069-4_26

focuses on enhancing the scalability and reconfigurability so that researchers can deploy and switch network scenarios conveniently and rapidly. The platform supports emulation of multiple network protocols and architectures. Some SIN protocols such as DTN and IP have been emulated on this platform.

Based on this platform [3], IP routing protocols OSPF and OSPF+ [4] are emulated in this paper. Then, we compare routing convergence performance with those of NS simulator to verify reliability of emulation.

The OSPF+ protocol was proposed by Mingwei Xu et al. [4]. Considering the periodicity and regularity of satellite motion, Xu et al. classified space link changes into predictable and unpredictable types. Aimed at predictable link changes, topology prediction module was added to the original routing protocol OSPF and the neighbor state machine module was modified.

The rest of the paper is organized as follows. Section 2 introduces design of the network emulation platform. Section 3 describes the detailed design process of research. Section 4 describes the experimental setup and performance evaluation. Session 5 concludes the paper.

2 Emulation Platform Design

As shown in Fig. 1, the architecture of the emulation platform can be divided into three parts: logical plane, control plane, and data plane.

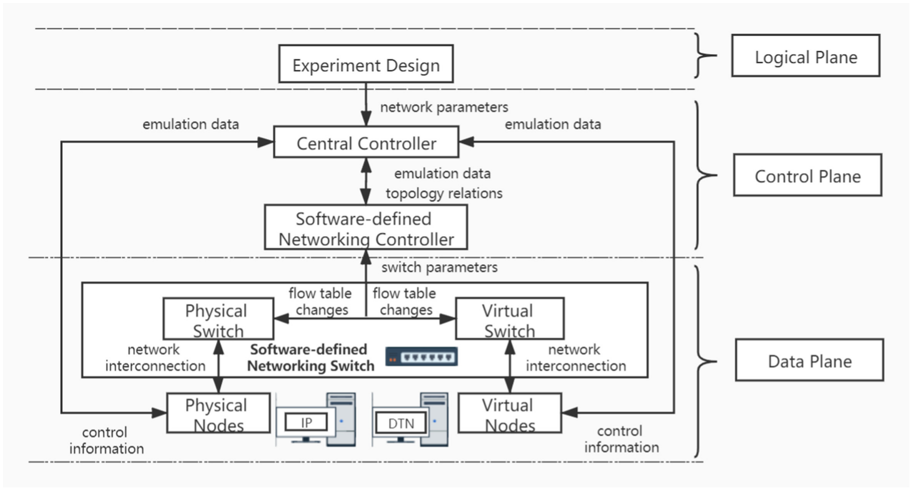


Fig. 1. Architecture of emulation platform.

The function of logical plane is to establish the corresponding network model by modeling software according to the specific SIN scenario. The network model includes connection plans, link states and protocol stacks loaded on the satellites. The Control Plane receives information and parameters from the logical plane and drives the emulations in the Data Plane. The main elements of the Control Plane include central controller and the SDN controller. Data Plane is the principal part of routing emulation, including emulation nodes, switches and other equipments, which is responsible for sending or receiving data, and monitoring the information. The Data Plane mainly includes emulation nodes and devices such as switches.

Various routing protocols or algorithms based on different protocol stacks, such as OSPF and CGR, can be tested and emulated on the proposed emulation platform [3]. The principle of multi-protocol emulation is shown in Fig. 2. Relevant protocol stacks are installed on the nodes according to emulation requirements, and all nodes are connected to the software-defined network switches.

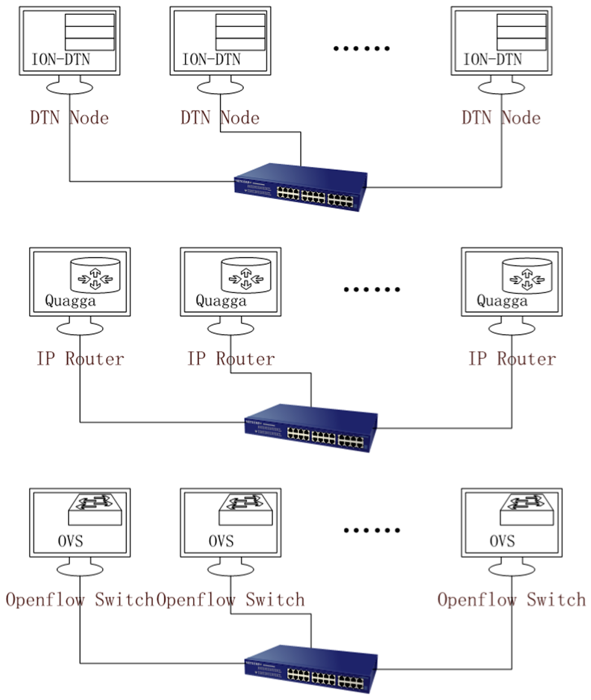


Fig. 2. Emulation of multiple protocols.

3 Routing Emulation Process Design

According to the emulation scheme, the whole emulation testing process can be divided into three phases: experiment preparation, experimental execution and data collection.

3.1 Experiment Preparation

In the experiment preparation stage of emulation, the work is mainly carried out on the logical plane and control plane.

In logical plane, according to the required space routing protocol, an emulation scene from [4] was modeled and the detailed network parameters were obtained. As shown in Fig. 3, a constellation composed by 14 low earth orbit (LEO) satellites is adopted. There are 10 intra-orbit links and 8 inter-orbit links in the constellation. The 10 intra-orbit links are always connected while 8 inter-orbit links are intermittently connected. The link on-off relationships of inter-orbit links in an experiment period (6000s) are illustrated in Fig. 4. A segment represents an interval in the connection state.

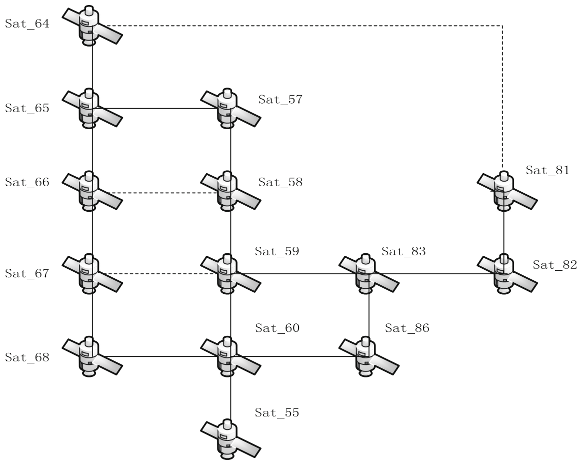


Fig. 3. Emulation scenario [4].

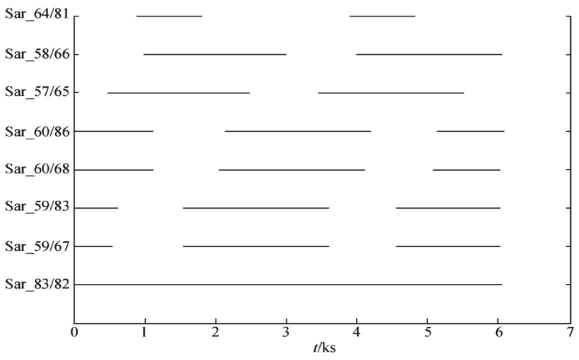


Fig. 4. Connectivity of inter-orbit links [4].

In the control plane, the central controller should be operated and other devices should be available simultaneously before Experiment Execution Phase. Concrete steps are shown in Fig. 5. As mentioned before, network parameters such as connection plans and dynamic link characteristics will be exported to the database of the central controller. Once the network topology changes, it is necessary to carry out Scenario design again and repeat the export steps. In addition, emulation nodes can be constructed from a pre-prepared protocol image and connected to the software-defined network switch. Here, Docker Virtualization Technology (VT) [5] are used to build OSPF/OSPF+ images and nodes. After finishing these preparation work, the central controller is connected to the nodes and switches so that it can send control messages to other devices and collect their feedback.

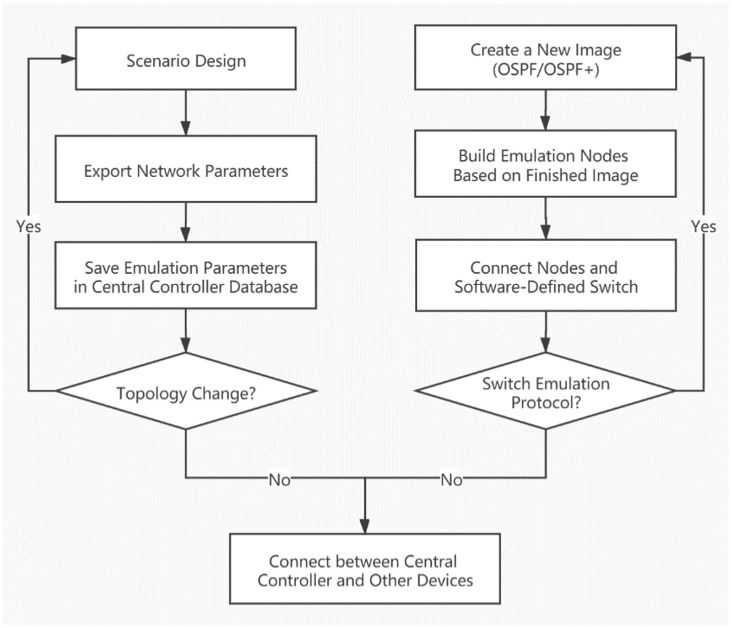

Scenario Design
Export Network Parameters
Save Emulation Parameters in Central Controller Database
Topology Change?
Yes
No
Create a New Image (OSPF/OSPF+)
Build Emulation Nodes Based on Finished Image
Connect Nodes and Software-Defined Switch
Switch Emulation Protocol?
Yes
No
Connect between Central Controller and Other Devices

Fig. 5. Concrete steps of experiment preparation.

3.2 Experiment Execution

In the experiment execution stage, the central controller sends the start signal to the emulation devices. After a node receives the signal, it will run the corresponding protocol process. In our experiment, it is referred to as OSPF/OSPF+ protocol process in the open source routing software Quagga [6]. Meanwhile, the software-defined networking controller will also be signaled. When the network topology changes, the software-defined networking controller will send a set of flow table items to the software-defined network switch. This makes switching network topologies flexible and convenient. The flow table [7] in our experiment is shown in Table 1. In every row, the first and second items represent network cards at both ends of an inter-orbit link, and the third and fourth items represent beginning and ending times of a link connection respectively.

Table 1. Flow table.

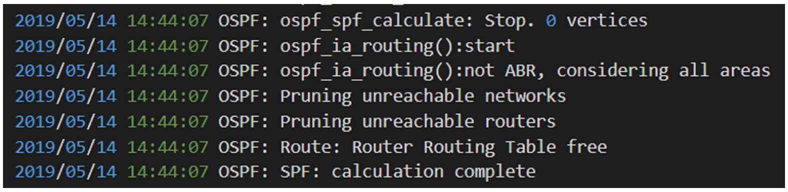
Source node	Dest node	Start time(s)	Stop time(s)
9002	9034	900	1900
9002	9034	3900	4900
9008	9018	1000	3000
9008	9018	4000	6000
9005	9015	400	2400
9005	9015	3400	5400
9026	9032	0	1100
9026	9032	2100	4100
9026	9032	5100	6000
9013	9025	0	1100
9013	9025	2000	4000
9013	9025	5000	6000
9022	9029	0	600
9022	9029	1500	3500
9022	9029	4500	6000
9011	9021	0	600
9011	9021	1500	3500
9011	9021	4500	6000
9030	9036	0	6000

3.3 Data Collection

After completion of data exchanging, it will enter Data Collection Phase. The data to be collected can be obtained from the software-defined network switch and each node. The switch will feed the real-time traffic information back to the central controller. Certainly, some packet capture software such as Wireshark [8] can be applied to analyze data packets. For example, it counts sending and receiving time of data packets at both ends of an end-to-end transmission to calculate the propagation delay and jitter rate etc.. In addition, by obtaining the statistical information of each route forwarding node, real-time neighbor status, route convergence status, and so on can be obtained.

In our research work, the performance indicators of concern were obtained by calculating the time difference recorded in the log file of OSPF/OSPF+ process. Log files of OSPF/OSPF+ process record the timeing of some key events such as state transitions and SPF calculation (see Fig. 6). The details are described in Sect. 4.

Disruption judgement time of OSPF/OSPF+ can be calculated by the following formula (1) and (2). In formula (1) and (2), $t_{linkoff}$ represents the time at which the link is disrupted. When OSPF detects a predictable link-off event, it will change the neighbor state from Full to Down. The time is marked as $t_{Full \rightarrow Down}$. When OSPF+ detects a



```
2019/05/14 14:44:07 OSPF: ospf_spf_calculate: Stop. 0 vertices
2019/05/14 14:44:07 OSPF: ospf_ia_routing():start
2019/05/14 14:44:07 OSPF: ospf_ia_routing():not ABR, considering all areas
2019/05/14 14:44:07 OSPF: Pruning unreachable networks
2019/05/14 14:44:07 OSPF: Pruning unreachable routers
2019/05/14 14:44:07 OSPF: Route: Router Routing Table free
2019/05/14 14:44:07 OSPF: SPF: calculation complete
```

Fig. 6. OSPF process log file.

predictable link-off event, it will change the neighbor state from Full to Leaving. The time is marked as $t_{Full \rightarrow Leaving}$. 5 experiments were carried out for each scenario to reduce the test error.

$$\text{Disruption judgement time}_{OSPF} = \sum\nolimits_{i=1}^{5} \left(t_{Full \rightarrow Down} - t_{linkoff}\right)/5 \qquad (1)$$

$$\text{Disruption judgement time}_{OSPF+} = \sum\nolimits_{i=1}^{5} \left(t_{Full \rightarrow Leaving} - t_{linkoff}\right)/5 \qquad (2)$$

Routing stability of OSPF/OSPF+ is defined as proportion of stable routing time (marked as t_{stab}) within two consecutive link disruptions (marked as t_{total}). Excluding the disruption judgement time (marked as t_{judge}), link state information synchronization time (marked as t_{sync}) and routing calculation time (marked as t_{calc}) in t_{total}, the remaining time is stable routing time. So the calculation formula of routing stability can be described as formula (3). Also, 5 experiments were carried out for each scenario, and the average was finally calculated.

$$\text{Routing stability}_{OSPF/OSPF+} = \left(\sum\nolimits_{i=1}^{5} \frac{t_{total} - t_{judge} - t_{sync} - t_{calc}}{t_{total}} * 100\%\right)/5 \qquad (3)$$

4 Performance Evaluation

Predictable link change times are read into the quagga's OSPF+ module in the form of a configuration file so that the OSPF+ neighbor state machine can trigger a state transition at a specified time. Two performance indicators of OSPF and OSPF+ have been tested and calculated in four scenarios. The bandwidth and delay are configured as 1000 Mbps/25 ms, 1000 Mbps/15 ms, 100 Mbps/15 ms, and 10 Mbps/15 ms respectively. The measured OSPF/OSPF+ disruption judgment time and routing stability (the percentage of time that all nodes in the entire area are converged within two consecutive link disruptions) are shown in the following Table 2 and 3, and Fig. 7 and 8 respectively. The left side represents the results of our emulation, and the right side represents those of NS simulator obtained from [4].

It can be seen that the disruption judgment time of OSPF+ is greatly shortened compared with that of OSPF due to the introduction of the topology prediction mechanism, which saves the time (4 Hello message periods) for judging disconnection between the neighbors as shown in Fig. 9. The neighbor structure of OSPF+ is preserved during

Table 2. Disruption judgement time.

Bandwidth/Delay	OSPF(s)	OSPF+ (s)	Bandwidth/Delay	OSPF(s)	OSPF+ (s)
1000 Mbps/25 ms	37.14	0.37	1000 Mbps/25 ms	41.00	1.00
1000 Mbps/15 ms	35.46	0.46	1000 Mbps/15 ms	41.00	1.00
100 Mbps/15 ms	34.68	0.89	100 Mbps/15 ms	41.00	1.00
10 Mbps/15 ms	33.64	0.94	10 Mbps/15 ms	41.00	1.00

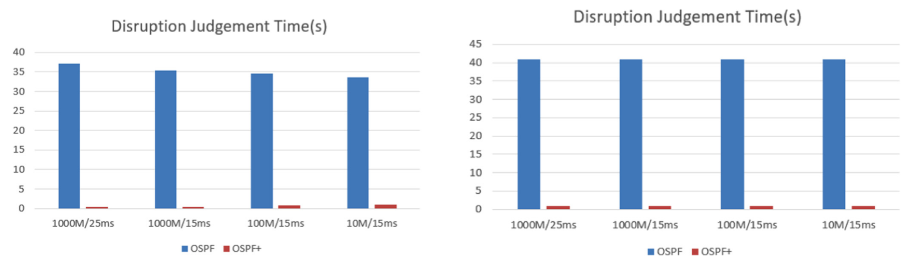


Fig. 7. Disruption judgement time.

Table 3. Routing stability.

Bandwidth/Delay	OSPF	OSPF+	Bandwidth/Delay	OSPF	OSPF+
1000 Mbps/25 ms	88.75%	99.60%	1000 Mbps/25 ms	76%	99.50%
1000 Mbps/15 ms	86.94%	99.63%	1000 Mbps/15 ms	76%	99.50%
100 Mbps/15 ms	86.63%	99.69%	100 Mbps/15 ms	76%	99.50%
10 Mbps/15 ms	87.31%	99.69%	10 Mbps/15 ms	76%	99.50%

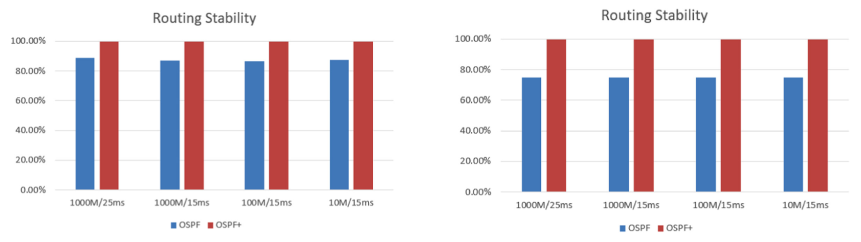


Fig. 8. Routing stability.

the predictable link disruption period. Therefore, when link recovery occurs, the neighbor state machine transforms from Leaving to Exchange directly without handshakes and negotiations. Besides, a new link state database newLSAsList which stores LSAs accepted during the disruption period is added to original OSPF module, so that synchronizing full link state database can be replaced by synchronizing newLSAsList only. These operations reduce the routing reconvergence time by about 14% compared with OSPF when links are restored. The routing stability of OSPF+ is approximately 100%.

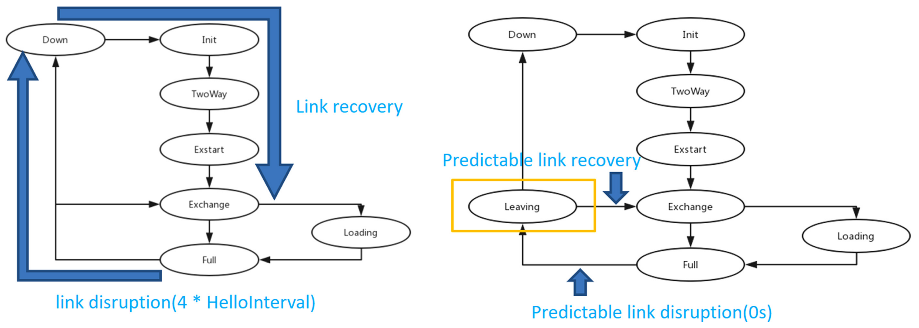


Fig. 9. OSPF/OSPF+ neighbor state machine.

There is a certain gap at the aspect of routing reconvergence time compared with the results obtained from [4]. In the performance evaluation of that paper, routing stability of OSPF is approximately 80% while in our paper the performance indicator is about 85%. Such slight difference can be ignored temporarily being considering the possible differences in software implementation and the introduction of real data flow of the emulation platform. By comparing the simulation results with NS simulation results, it can be seen that the current emulation platform has been able to carry out IP routing emulation, which ensures the reliability to a certain extent.

5 Conclusion

Based on the designed emulation platform, emulation is performed on the newly proposed spatial routing protocol OSPF+. Performance comparison has been made with the traditional intra-domain routing protocol OSPF. It can be seen that for predictable link changes in space network, OSPF+ has a faster link disruption judgment mechanism and shorter routing re-convergence time compared with OSPF. Furthermore, a simple comparison has been carried out between performance indicators obtained from our emulation and NS simulator. Results indicate current emulation platform has the ability of IP routing emulation, which guarantee considerable reliability and provides a reference for subsequent emulation work of IP protocols.

References

1. Zhao, J., Zhu, X.: Research on a hardware-in-the-loop simulation method for wireless network based on OPNET. In: 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, pp. 821–825 (2015)

2. Wang, Z., Cui, G., Li, P., Wang, W., Zhang, Y.: Design and implementation of NS3-based simulation system of LEO satellite constellation for IoTs. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, pp. 806–810 (2018)
3. Lu, T.: An emulation architecture of routing protocol in space information network. Nanjing University (Master's thesis in Chinese) (2018)
4. Xu, M., et al.: Ground-air integrated network intra-domain protocol OSPF+. J. Tsinghua Univ. (Nat. Sci. Ed.) **57**(01), 12–17 (2017). (in Chinese)
5. https://www.docker.com/
6. Chen, Y., He, Y., Zhao, Z., Liang, X., Cui, Q., Tao, X.: DEMO: a quagga-based OSPF routing protocol with QoS guarantees. In: 2018 24th Asia-Pacific Conference on Communications (APCC), Ningbo, China, pp. 5–6 (2018)
7. Yoshino, N., Oguma, H., Kamedm, S., Suematsu, N.: Feasibility study of expansion of OpenFlow network using satellite communication to wide area. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, pp. 647–651 (2017)
8. https://www.wireshark.org/