The Development of a Lightweight Network Emulator for Large-Scale Space Network Emulation

Fa Chen, Weidong Zhou, Kanglian Zhao, Wenfeng Li(✉), and Yuan Fang

School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, People's Republic of China
{zhaokanglian, yfang}@nju.edu.cn, leewf_cn@hotmail.com

Abstract. Based on the previously developed network emulator named MininetE [1], this paper proposes an improved scheme of lightweight network emulator for large-scale space information network (SIN) emulation. The improved MininetE still has the ability to emulate network link characteristics such as bandwidth, latency and packet loss, while generating real network traffic. Through the setting of network configuration parameters, not only DTN protocol but also IP-based OSPF protocol can be run in the new MininetE, making it possible to build a SIN emulation environment where multiple network protocols coexist. Based on this, a SIN scenario with the improved MininetE is constructed to evaluate the performance of OSPF and the coexistence of OSPF and DTN respectively. The results show that the emulation platform has the DTN and IP-based routing emulation capability and ensures considerable reliability, which provides a reference for further optimization of MininetE emulation platform in the future.

Keywords: Space information networks · Emulation · MininetE · High fidelity · OSPF

1 Introduction

Space Information Network (SIN) is a backbone communication network composed of several satellites and satellite constellations in orbit, which can provide communication services for various space missions. With the development of SIN, it is very necessary to carry out research on new network technology and evaluate network protocols and algorithms. If we conduct the research in a real space network system, testing software needs to be pre-loaded into satellite communication systems, and then launches the test satellites into space. This approach will not only have a long cycle and high cost, bust also may affect existing satellite network systems.

So we need a emulation platform to conduct the research. The traditional physical emulation will inevitably bring high deployment cost, complicated scene parameter configuration and other problems. On the other hand, if we use the existing network simulator (such as OPNET [2] or ns [3]). They are essentially a discrete event simulator. All simulations are driven by discrete events so they are not as real as mininetE.

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 289–298, 2022.
https://doi.org/10.1007/978-3-030-93398-2_29

Therefore, the Institute of Space-Terrestrial Intelligent Networks (ISTIN) of Nanjing University has developed an extensible space interconnection network emulation platform [4]. This emulation platform can be deployed and switched quickly, and provide accurate spatial simulation results. However, due to the high cost, it needs good hardware resources support, which is not realistic for individuals. In order to satisfy the personal ability to carry out some simple experiments and verification on our own laptop, we developed a lightweight emulation platform which is flexible and reconfigurable for spatial information network protocol [1]. On the basis of keeping lightweight, the platform adds adequate isolation, which makes MininetE simulation platform available for spatial information network protocol simulation.

The goals of a usable emulation platform include not only flexibility and support for quick switching between scenarios, but also authenticity, such as function realism and traffic realism.

Through the setting of network configuration parameters, not only DTN protocol but also IP-based OSPF protocol can be run in the new MininetE. Based on this, the performance of OSPF and the coexistence of OSPF and DTN respectively is evaluated in the scenario which is constructed.

2 Architecture Design

Mininet [5] is a lightweight network simulator based on container virtualization technology. However, like most network simulators, Mininet is widely used in the simulation of ground network, but it lacks the simulation support of spatial information network related protocols. Therefore, MininetE emulation platform is developed based on Mininet terminal host. In addition to Mininet providing isolation of network namespace and mount namespace, MininetE adds isolation support of UTS namespace, PID namespace and IPC namespace to virtual nodes. Figure 1(A) shows If double OSPF processes are started simultaneously in Mininet, the second process will fail. However, Fig. 2(B) shows MininetE can run two or more OSPF processes simultaneously. Because of the adding namespace isolation, processes do not interact with other node's processes. Similarly, Fig. 2(C) shows different nodes can run different protocols without affecting each other. All virtual nodes can use different network protocol systems (such as TCP/IP, CCSDS, DTN) according to different target scenarios, so as to achieve flexible switching of different network protocol simulation scenarios.

3 Improvement to MininetE

In the proposed platform, emulation nodes, links and services are necessary for simulation.

Simulation Nodes: It is composed of MininetE hosts and is the main object of network scale expansion. In order to make the node have a real simulation, the increased

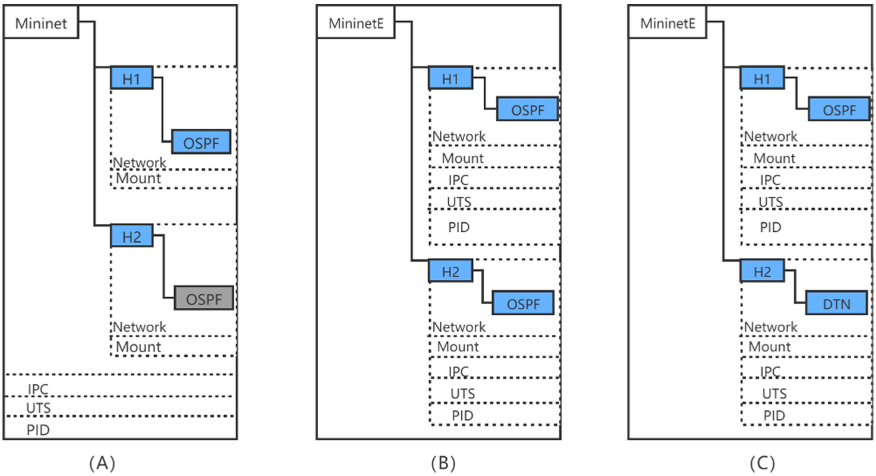


Fig. 1. Comparison of SIN protocol in Mininet and MininetE.

namespace isolation of MininetE can deploy different protocols (such as DTN, TCP/IP) in each simulation node.

Links: Link is necessary for communication between nodes. The on-off of link is determined by the the movement of the satellite. We can use TC NETEM [6] to set the link characteristics.

Switch: We can use ordinary switches or SDN switches, so we can easily control the network topology by the controller.

Controller: The controller is responsible for telling the node and the switch what to do.

In the space information network, due to the movement of satellites, the visibility between satellites may change, and the topology will change accordingly. Switching simulation link is very important. Fortunately, MininetE has an inherent support for software defined (SDN)/Openflow [7]. We develop a link switching method using SDN controller. Our scheme is based on the flow table control of openvswitch by SDN controller. As shown in the Fig. 2, The controller has global topology information. Whenever there is a link state change, it will follow up with the new topology. The controller will send a flow table to the switch to simulate the pass and segment of the link.

As the scale of emulation nodes increases and the nodes in the space information network are in high speed motion, the whole network topology changes dynamically. The connection between nodes is intermittent, and the connection objects of nodes may be switched at some time, and the link parameters will change accordingly. MininetE works by adding different ports, which means different possible neighbor nodes, and switching links by sending flow table.

Figure 3 shows the simulation process. First, We need to determine the simulation scenario. That is to determine the number of simulation nodes, node types, simulation

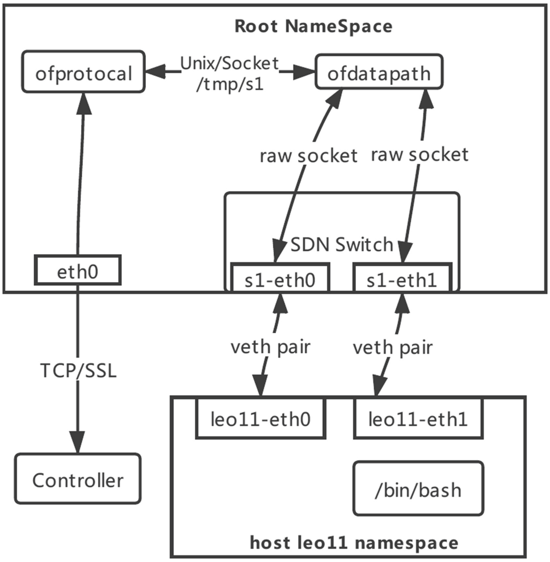


Fig. 2. Controller sending flow table

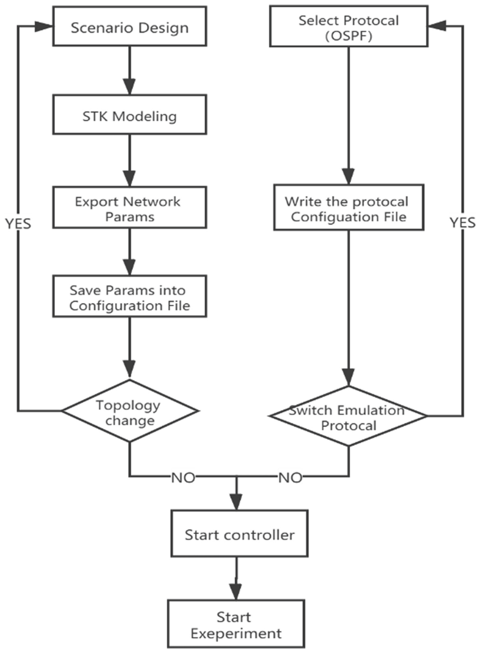


Fig. 3. Simulation process

protocols, etc. After this, we save the network params into configuration file, and then start the controller to begin the simulation.

The simulation node is connected to the SDN switch. Each node runs the program and waits for instructions from the master controller. At the same time, The controller sends a flow table to the switch based on the connection relationship to control the connectivity between the nodes, and tells the nodes what process to run at what time.

4 Experimental Validation

4.1 OSPF Routing Simulation Process

A complete emulation consists of three stages of scene design, experiment operation and data collection.

Scene Design
Firstly, the model is established according to the emulation scene, and detailed simulation parameters can be obtained from such as STK. As shown in the Fig. 4, we build an experiment scenario which contains 14 LEO satellites for space internetworking. The solid line indicates that the links are always connected, and the dashed line indicates that they are intermittently connected. The on-off relationship of the inter orbit link in the experimental period (6000 s) is shown in the Fig. 5. The connection represents the time interval in the connected state. We want to study the performance of OSPF in this satellite network, and show the experimental results on our platform. The 14 simulation nodes are composed of 14 Mininet nodes and connected to SDN switches. After these preparations, the controller is connected to the SDN switch for control.

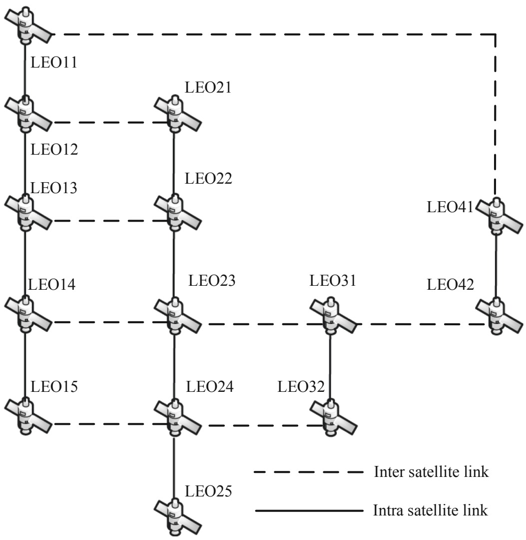


Fig. 4. Satellite network topology in experimental scene.

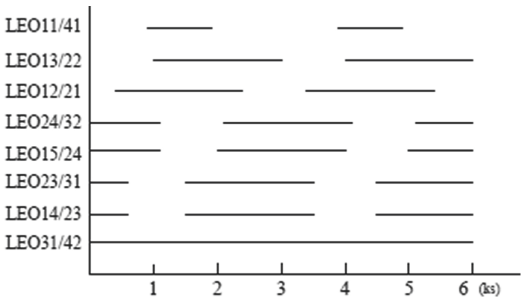


Fig. 5. Inter-satellite link connection relationships

Experimental Operation

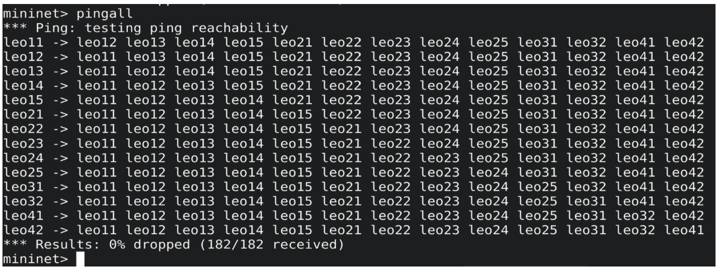
The control plane is mainly composed of the main controller. The control receives information and parameters from the logic plane to drive data emulation. Before the implementation of the experiment, the controller should be in operation, and other equipment should be available at the same time. The simulation parameters are written to the configuration file ahead of time. Once the network topology changes, we need to redesign the scenario and repeat the above steps.

In the experimental phase, the controller triggers the simulation node to start. The node then runs the corresponding process, such as OSPF or DTN. The Quagga [8] used to support OSPF routing protocol and installed on the host. At the same time, the software defined network controller will also receive the signal. The controller has global topology information. When a packet arrives at the OpenFlow switch, the flow table will be matched. If no corresponding flow table entry is found, a packet_in message will be sent to the SDN controller. After the controller makes a decision based on a certain routing algorithm, it sends a flow table to the switch.

Data Collection

In the data collection stage, data can be obtained through switches and nodes. Through some commands, we can obtain real-time information such as neighbor state and routing convergence state. we can obtain the real-time neighbor state, routing convergence state and other information.

As shown in Fig. 6, through the Ping between nodes, any two nodes can communicate with each other, and the result of global router information table proves that MininetE supports OSPF routing protocol.



```
mininet> pingall
*** Ping: testing ping reachability
leo11 -> leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo12 -> leo11 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo13 -> leo11 leo12 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo14 -> leo11 leo12 leo13 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo15 -> leo11 leo12 leo13 leo14 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo21 -> leo11 leo12 leo13 leo14 leo15 leo22 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo22 -> leo11 leo12 leo13 leo14 leo15 leo21 leo23 leo24 leo25 leo31 leo32 leo41 leo42
leo23 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo24 leo25 leo31 leo32 leo41 leo42
leo24 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo25 leo31 leo32 leo41 leo42
leo25 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo31 leo32 leo41 leo42
leo31 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo32 leo41 leo42
leo32 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo41 leo42
leo41 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo42
leo42 -> leo11 leo12 leo13 leo14 leo15 leo21 leo22 leo23 leo24 leo25 leo31 leo32 leo41
*** Results: 0% dropped (182/182 received)
mininet>
```

Fig. 6. Connectivity between nodes

4.2 Hybrid Protocol Emulation

In addition, due to the increasing complexity of space network scenarios, nodes are required to support more than a single network protocol, such as DTN and OSPF routing protocols [9]. It is capable of supporting multiple protocols. As you can see from Fig. 1, MininetE has this ability. In the emulation scenario in Fig. 4, in addition to deploying the OSPF protocol, arbitrarily select several nodes to run the DTN protocol. For example, LEO11 serves as the sending node for data transmission, LEO12 serves as the intermediate node for data forwarding, and LEO13 serves as the receiving node for data. Their communication protocol stack structure is shown in the Fig. 7. The IP layer runs the OSPF routing protocol.

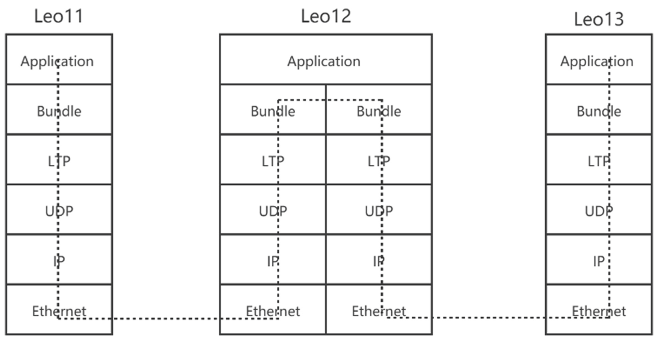


Fig. 7. The node's communication protocol stack

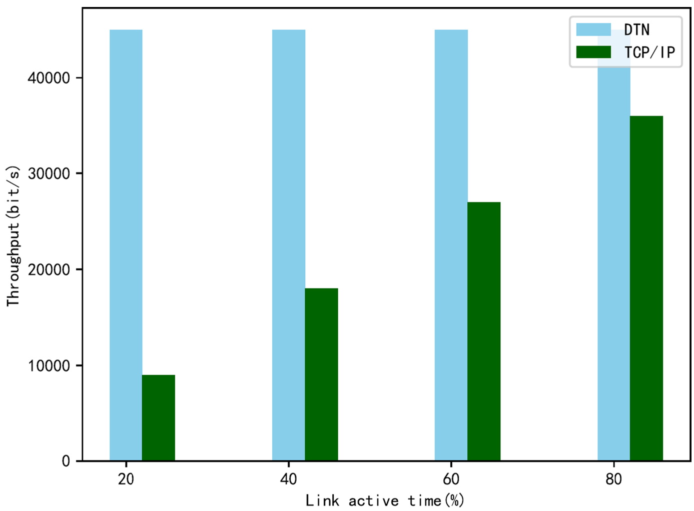


Fig. 8. Data throughput of different link effective time

According to the emulation scenario and experimental configuration, we selected the data throughput of the target node LEO13 as the object of statistics and research, and compared the communication performance of DTN and TCP/IP under different transmission protocols. Figure 8 shows the data throughput rate under the condition that the link time efficiency is 80%, 60%, 40% and 20% respectively [10]. It can be seen that in the deep space communication environment with large delay and high bit error rate, TCP protocol cannot complete the effective transmission of data, and DTN protocol is more suitable to be used in the deep space communication network scenarios.

5 Performance Evaluation

Because the nodes in the space information network are in the state of high speed motion, the whole network topology changes dynamically. The connection between nodes is intermittent, and the connection objects of nodes may be switched at a certain moment. The controller is used to send the flow table to the SDN switch to simulate the link switching and change the network topology. As shown in Fig. 9, The result of traceroute shows that the route converges again. The so-called convergence of routing protocol refers to the process that the routing protocol finds other routers in the network to change and exchange routing entries until all routing entries are exchanged. In this example, as shown in Fig. 3, there is only one hop from leo11 to leo21 at the beginning. When I disconnect the link between leo11 and leo21, OSPF will re select the shortest path for convergence, and the packets will go through leo13, then to leo22, and finally to leo21.

```
root@iZbp1frdqql4a2oq5ctnenZ:/home/admin/mininetE/custom# traceroute 10.0.5.6
traceroute to 10.0.5.6 (10.0.5.6), 64 hops max
  1   10.0.5.6  0.002ms  0.002ms  0.002ms
root@iZbp1frdqql4a2oq5ctnenZ:/home/admin/mininetE/custom# traceroute 10.0.5.6
traceroute to 10.0.5.6 (10.0.5.6), 64 hops max
  1   10.0.2.3  0.002ms  0.002ms  0.002ms
  2   10.0.13.7  0.001ms  0.001ms  0.002ms
  3   10.0.5.6  0.001ms  0.002ms  0.002ms
```

Fig. 9. Route convergence

The judgment interrupt time of OSPF can be calculated by the formula (1), where $t_{linkoff}$ represents time of the link interruption and the $t_{linkoff}$ represents the time nodes to changes the neighbor state from full to down.

$$t_{judge} = t_{Full->Down} - t_{linkoff} \tag{1}$$

We tested several scenarios. The bandwidth and delay are 1000 Mbps/25 ms, 1000Mbps/15ms, 100 Mbps/15 ms and 10 Mbps/15 ms respectively. Measured OSPF interrupt judgment time. The measured OSPF disruption judgment time in MininetE, Docker and NS simulator [11] is shown in the following Table 1.

Table 1. Disruption judgement time.

Bandwidth/Delay	MininetE(s)	Docker(s)	NS(s)
1000 Mbps/25 ms	38.24	37.14	41
1000 Mbps/15 ms	37.13	35.46	41
100 Mbps/15 ms	36.98	34.68	41
10 Mbps/15 ms	36.57	33.64	41

According to Table 1, the judgment interrupt time is basically the same. Considering the possible difference between the software implementation and the introduction of the actual data flow on the emulation platform, this slight difference can be ignored temporarily. By comparing the simulation results of MininetE, Docker and NS, we can see that the improved MininetE can support OSPF and guarantee the reliability.

6 Conclusion

The improved MininetE, which can run emulation scenarios of different protocols on each laptop. With our dynamic topology control method, we can control the spatial network topology. Emulation results show that MininetE has the ability to support DTN network and OSPF protocol. The experimental results are simply compared with docker and NS simulators. The emulation results show that the improved MininetE can support many kinds of spatial network protocols. In the future, more experiments will be carried out to prove that MininetE can meet the requirements of various network protocal and support larger scale emulation.

Acknowledgements. This work is supported by the 13th Five-Year Civil Aerospace Technology Pre Research Project, the Fundamental Research Funds for the Central Univesities under Grant 021014380187 and the National Natural Sciences Foundation of China under Grant 62131012.

References

1. Lin, T., Chen, F., Zhao, K., Fang, Y., Li, W.: MininetE: a lightweight emulator for space information networks. In: Wu, Q., Zhao, K., Ding, X. (eds.) Wireless and Satellite Systems. WiSATS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 357, pp. 48–57. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69069-4_5
2. 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)
3. 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)

4. Lu, T.: An emulation architecture of routing protocol in space information network. Nanjing University. Master's thesis in Chinese (2018)
5. Lantz, B., Heller, B., McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: ACM Sigcomm Hotnets Workshop (2010)
6. tc-netem. [EB/OL]. https://www.linux.org/docs/man8/tc-netem.html
7. The openflow switch. http://www.openflowswitch.org
8. 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)
9. Zhao, K., Wang, R., Burleigh, S., Sabbagh, A., Wu, W., Sanctis, M.D.: Performance of bundle protocol for deep-space communications. IEEE Trans. Aerosp. Electron. Syst. **52**(5), 2347–2361 (2016)
10. Sabbagh, A., Wang, R., Burleigh, S.C., Zhao, K.: Analytical framework for effect of link disruption on bundle protocol in deep-space communications. IEEE J. Sel. Areas Commun. **36**(5), 1086–1096 (2018)
11. Xu, M., et al.: Ground-air integrated network intra-domain protocol OSPF+. J. Tsinghua Univ. (Nat. Sci. Edn.) **57**(01), 12–17 (2017). (in Chinese)