A Multi-service Traffic Generation System for Emulation of Space Information Networks

Feng Jiang(✉), Qi Zhang, Yuhang Du, Kanglian Zhao, Wenfeng Li, and Yuan Fang

School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
mf1723024@smail.nju.edu.cn

Abstract. Recently, tremendous progress has been made in the research on space information networks, which also inspires the development of general or dedicated emulation systems. As an important part in emulation of the space information networks, the research on traffic generation system is still in the preliminary stage. In view of the insufficient research in this area, a multi-service traffic generation system dedicated for emulation of space information networks is proposed in this paper, which has the characteristics of scale and flexibility. The system is designed based on software-defined networking technology to decouple the traffic generation part from the traffic transmission part, avoiding the complicated configuration problems in emulation of large scale space information networks. At the same time, the traffic generation part of the system uses a variety of business models, which is helpful for the research in future space information networks.

Keywords: Traffic generation · Business model · Space information networks

1 Introduction

With fast research and development in communications, networking and onboard computing technology, the space information networks system are increasing in complexity and scalability, which requires low cost, flexible, high-fidelity experiment and verification before the expansive production and implementation of the system. However, space industry is high-cost and high-risk, which makes it difficult to build testbeds with a bunch of space nodes. Therefore, it is essential to test and verify the performance of the software and the hardware of space information networks through ground emulation platforms [1]. With reference to the test methods of terrestrial networks, traffic generation is an extremely important part of the entire emulation platform. The traffic generator generates data packets with the control of corresponding business models to emulate the data traffic in the scenario under test, which can effectively customize the traffic parameters to meet the test requirements under different network conditions.

There has been a lot of research on traffic generators in the literature. There are also mainstream traffic generators in the open source community. Iperf [2] is a common traffic generator, which is commonly used as a TCP/UDP performance evaluation tool. By tuning various parameters, the maximum bandwidth of TCP and statistics such as

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 35–47, 2021.
https://doi.org/10.1007/978-3-030-69069-4_4

bandwidth, delay, maximum segment, and maximum transmission unit size can be tested. Harpoon [3] is based on the ON/OFF model of heavy-tail distribution to generate self-similar network traffic. Harpoon has two components: a client thread that issues a file transfer request and a server thread that uses TCP or UDP to transfer the requested file. Its self-configuration process uses file records as input to generate the necessary parameters for configuring the network traffic model to generate artificial traffic with near real-time traffic characteristics. D-ITG [4] (Distributed Internet Traffic Generator) can generate IPv4 and IPv6 traffic, as well as network layer, transport layer and application layer traffic. D-ITG can also simulate network delay and support multiple protocols including TCP, UDP, SCTP, ICMP, etc. D-ITG can simulate service traffic such as VoIP and Telnet by modeling a random model of message size and message intervals. And D-ITG also has a logging function. In addition, many companies develop professional hardware-based traffic generators based on their business needs. For example, Breakingpoint by Ixia or Smartbits by Spirent. These professional hardware traffic generation devices can generate up to 10 Gigabit network traffic, and can accurately customize the bandwidth, packet length and number of network data streams. However, these hardware devices are difficult to reflect the diversity of real network data protocols, and are generally very expensive, mainly used for commercial purposes, and are not suitable for the initial stage of network research.

Three different methods are categorized based on the review of the state-of-the-art in network traffic generation [5]: 1) traffic generation based on network model; and 2) traffic generation based on traffic characteristics; and 3) traffic generation based on application protocol. The advantages of the three methods are summarized, which provides important guidance for the subsequent research in the field of traffic generation. The literature [6] designs the FPGEN flow generator based on FPGA. The traffic generator can send data traffic of Poisson model and Markov model. Because the design writes the configuration message of the data stream directly in RAM, the flow rate can reach 2.5 Gps. Later the literature [7] optimizes the design so that the transmission rate could reach 5 Gps, but the shortcoming was that the IP and port number of the data stream could not be flexibly configured. Literature [8] combines software-defined networking (SDN), uses traffic-based virtual network technology and traffic generation technology based on user behavior model, designs and implements a multi-user network traffic generator that can centrally control traffic generation and physical network resources, and can effectively distinguish network traffic.

However, some of the traffic generation tools described above generally can only generate simple TCP/UDP traffic, and the support for business traffic is relatively simple. One of the current development trends of space information network is the development of network services from a single service to multiple services. There are also many studies on traffic generators supporting multiple services in China, but they are all based on terrestrial network. Based on the discussion and summary of the traditional business traffic model and self-similar business traffic model, the literature [9] designs the main modules of the traffic generator such as the analysis module, business traffic module, traffic generator module, timer module, WinSock interface module, etc. And implements a traffic generator supporting multiple services such as voice traffic, video traffic, and M2M traffic. The literature [10] builds a multi-service traffic generator framework based on Qt on the Linux platform, encapsulates the data structure of the business flow, and generates network traffic in a multi-threaded manner. By separately modeling the HTTP service, voice service and video service, the network traffic sequence with its own self-similar characteristics is generated to realize support for multiple services. In addition, the literature also verified the validity and practicability of the model.

The traffic generators mentioned in the above-mentioned domestic and foreign literatures have their applicable scenarios and characteristics, and the research of these traffic generators is mainly for terrestrial networks running TCP/IP network protocol. The space information network is very different from the terrestrial network. The terrestrial network conditions that can run the TCP/IP network protocol are often better, while the communication environment of the space information network is more harsh. Although in low-orbit networks the TCP/IP protocol can still be run directly due to the low orbital height, as the orbital height increases, the operational performance of the TCP/IP protocol in the space information network will be greatly reduced. Therefore, the above-mentioned traffic generator is not suitable for space information networks.

However, at present, there are still insufficient researches on traffic generators suitable for space information networks. The literature [11] develops a DTN traffic generator "DTN-tg" suitable for space information networks. DTN-tg can generate a constant rate of data traffic that conforms to the DTN protocol. The entire working mode is equivalent to adding a client node and a server node at both ends of the network under test. The generated traffic of the client reaches the server node through the network under test to monitor and measure the performance of the DTN node in the network under test. However, this working mode is essentially equivalent to modifying the topology of the network under test. In the DTN node network, node configuration changes are required. Moreover, the traffic generator does not support the generation of multi-service network traffic.

2 System Structure

The literature [1] builds a simulation platform with authenticity, flexibility and reliability for the space information networks. The simulation platform is mainly composed of main control, simulation node, front-end interface and SDN switch. With the idea that the control plane and the data plane are separated from each other, the simulation platform can carry out true and reliable simulation of the protocol architecture, topology information, link characteristics and node performance in the space information network. In addition, through the GUI interface, users can flexibly and conveniently import or modify simulation scene information, start simulation tasks, monitor scene node status, and obtain experimental result reports. However, the simulation platform is lacking in traffic generation, so the traffic generation system designed in this paper is also an improvement of the simulation platform.

The common traffic generation system is mainly to add sending nodes and receiving nodes at both ends of the network under test, so that the sending node and receiving node have a path through the network under test, so that the traffic generated by the sending node circulates in the network under test. To measure various performance indexes of the tested network. This architecture is simple and easy to use, but as the network scale increases, the cumbersome configuration will greatly affect the flexibility of the architecture.

Therefore, based on the consideration of scale and flexibility, this paper designs a more flexible architecture. As shown in Fig. 1.

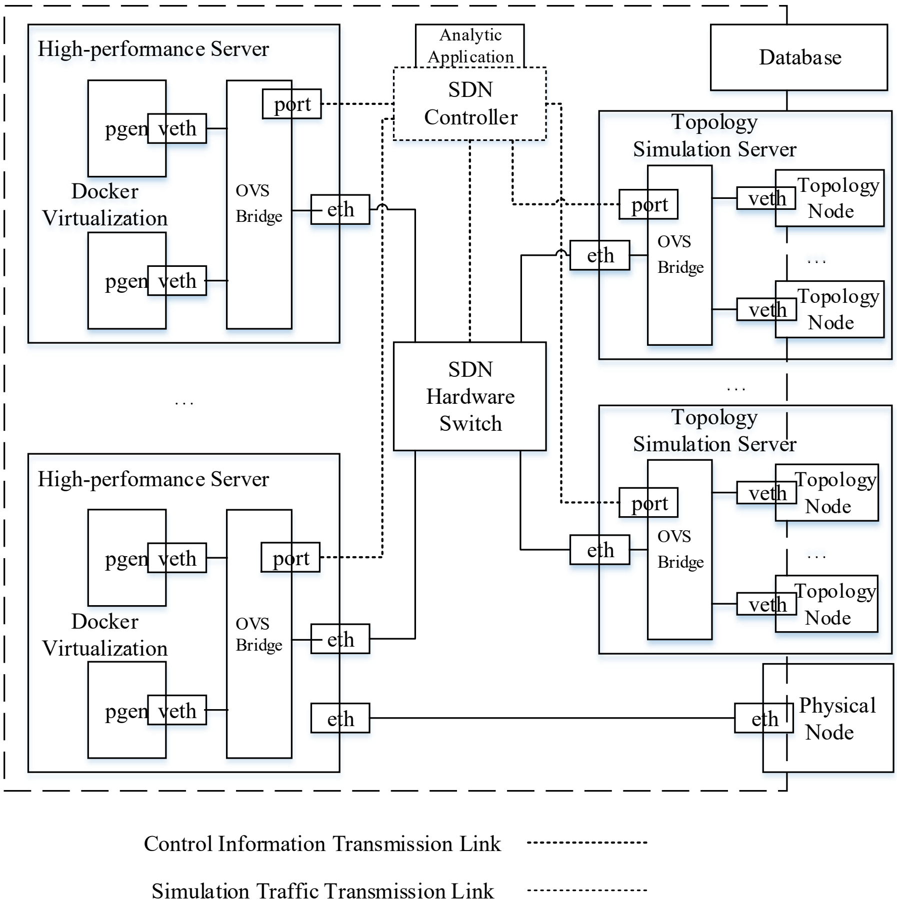

Fig. 1. Architecture diagram of the traffic generation system

The part enclosed by the dotted line in Fig. 1 is the main part of the traffic generation system. It mainly consists of traffic generation nodes (such as pgen in Fig. 1), SDN switches, SDN controllers, etc. Because the system architecture needs to realize large-scale network traffic generation, if a large amount of traffic generation and transmission are implemented in the same server, it will be limited by the server bus speed. Once overload occurs, data loss will occur, affecting the experimental results. Therefore, the traffic generation system adopts a distributed design, placing the traffic generation node in another high-performance server, separating data generation from data transmission, and reducing the load on the server. In addition, the traffic generation system architecture can expand the traffic generation node to multiple high-performance servers, and adopt Kubernetes for cluster management. The unframed part on the right in Fig. 1 is the simulation network part, which can be virtual nodes or physical nodes. In addition, there is a database in the simulation network part, which stores some node information and link information. The simulation network is connected to the virtual switch in the traffic generation system by adding a virtual network port, and the SDN controller is responsible for data forwarding control to realize the interconnection of the two parts. When the traffic generation system is connected to the simulated network, the information of the additional virtual network port will be written into the database as the information of the topology node for subsequent reading and use.

This traffic generation system mainly has three parts of functional design. These are business traffic generation, traffic forwarding control, and network protocol conversion.

The business traffic generation part is implemented in the traffic generation node. First, a traffic model is established according to business characteristics, and then the traffic generation node generates corresponding UDP traffic through Socket network programming and realizes the transmission of the traffic. The third section will introduce the multi-service modeling.

The traffic forwarding control part is implemented by SDN related components. There is an analytic application in the SDN controller in the traffic generation system in this paper, which is used to identify the relationship between the sending and receiving nodes in the parameter configuration file. In addition, the analysis application needs to read the topology node information in the database. After analyzing the obtained information, the source node IP and destination node IP of the injected traffic are discriminated, and the corresponding flow entry information is automatically generated and delivered to the corresponding switch device to ensure that the UDP data traffic generated by the traffic generation node is correct to the corresponding topology node.

The network protocol conversion part is implemented in the topology node part, and is used to process the UDP data packet injected by the traffic generation node. The traffic generation system can be connected to the simulation network of any topology, and supports the operation of different network protocols in the simulation network. For different network protocols in the simulated network, the network protocol conversion will process the received UDP data differently. For the simulation network running the DTN protocol, the relevant experiments in Sect. 4 set the simulation network to run the DTN protocol. First, the topology node acts as the receiving end of UDP traffic. After receiving the UDP data packet, the payload of the UDP data packet is extracted, and the load is encapsulated in accordance with the protocol system running in the simulated network. Finally, a data packet conforming to the simulated network protocol is generated and transmitted in the topology node.

The traffic generation system has the characteristics of low coupling and high flexibility. The low coupling is reflected in the complete decoupling of the traffic generation system and the simulation network. The traffic generation system can be connected to the simulation network of different protocols without modifying any internal configuration. The high flexibility is reflected in the fact that the traffic generation nodes are generated by Docker virtualization, which can achieve lightweight and large-scale deployment, and combine the SDN controller to send flow entries to control the data flow direction, and can flexibly configure the multi-transmission and multi-reception situation between a large number of nodes. It is suitable for large-scale space information network scenarios.

3 Multi-service Modeling

Considering the relative simplicity of the current space information network business, in order to facilitate the establishment of business models, the modeling method used in this paper mainly uses two important characteristics of business traffic: packet interval and packet size. According to the distribution function of packet interval and packet size and other characteristics of business traffic [12], the relevant traffic generation program is realized. At present, the traffic generation system in this paper supports five types of services, namely remote sensing satellite service, voice service, VoIP service, near real-time video service and HTTP service.

The remote sensing business modeling in this paper mainly intercepts the model of the flow acquisition part in the literature [13] and introduces the MMDP process to achieve it. In this experiment, it is assumed that there are four remote sensors in the remote sensing satellite business, so there are five remote sensor states, and each state will only transition to the adjacent state. In addition, the residence time of each state in the remote sensing satellite business conforms to an exponential distribution. At the same time, due to the different usage of remote sensors in various states in the remote sensing satellite business, the corresponding bit rate should also change. In this paper, the basic packet load is set to 50 bytes. According to the current state of use of the remote sensor, the packet load generated by the remote sensing satellite service needs to be increased by a corresponding multiple, so as to indirectly realize the change of the transmission rate. According to the characteristics of the above packets, the algorithm model is established to send a packet of the size of the service characteristics every time interval that conforms to the service characteristics.

In this paper, the traditional modeling method is used to model the voice service, that is, a two-state alternating model is used, which corresponds to the call state and the pause state in voice communication. This model is also called the ON/OFF model. When the model is in the ON state, that is, in the call state, there is traffic generated at this time; when the model is in the OFF state, that is, in the pause state, no traffic is generated at this time. The duration of the ON state and the OFF state conforms to an exponential distribution. In addition, the data traffic generated in the ON state conforms to the Poisson arrival process, that is, the packet interval time conforms to the exponential distribution.

Similar to the traditional voice service, the VoIP service also has two states: active state and inactive state. But unlike the traditional voice service where the two states appear alternately, the VoIP service is modeled using a two-state Markov process. as shown in Fig. 2.

The duration of the active state and the inactive state of the VoIP service conform to the exponential distribution, and their average values are $1/\beta$ and $1/\alpha$, respectively. In the active state, the sending source generates fixed-size packets at specific time intervals. In the inactive state, the sending source generates background data packets with a small amount of data at different specific time intervals. This is different from the traditional

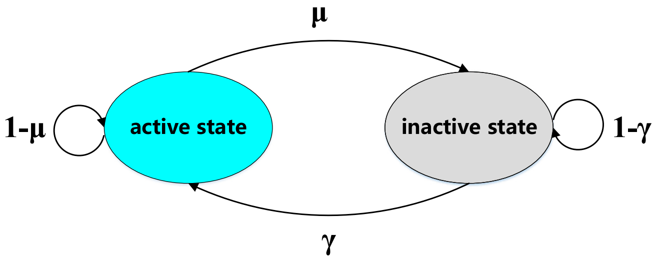

Fig. 2. Markov chain model of VoIP service

voice service. Similar to the heartbeat packet in a long TCP connection, the inactive state tests whether the connection is interrupted by generating a data packet with a smaller amount of data.

Near real-time video service (NRTV) is a type of service that requires high real-time performance. Its QoS requires one-way delay within 10 s, delay jitter as small as possible, and bit error rate less than 1% and so on. The modeling of the NRTV service in this paper is to divide the entire video streaming session into multiple frames. The time interval of arrival of each frame is kept fixed, and each frame can be divided into a fixed number of data packets. The size and arrival interval of these packets conform to the truncated Pareto distribution.

This paper models HTTP business traffic based on web objects. The modeling of HTTP business involves several concepts of web page composition. Suppose an HTTP business process is regarded as a session. When a user runs a web browser, it means that a session starts. When the user disconnects from the Web server, it means that the session ends. Each session contains multiple webpages, meaning that during a session, the user can click on multiple webpages to browse. Each web page will contain multiple objects, these objects can be divided into main objects and embedded objects, and a web page will only have one main object, but there can be multiple embedded objects. The basic flow of the HTTP business model is shown in Fig. 3.

The modeling of HTTP business mainly focuses on the following six parameters: 1) The number of webpages in one session: indicates the number of webpages viewed by the user. 2) The size of the main object in each web page: the main object is a document described by Hypertext Markup Language (HTML), which is the main frame of the entire web page. The size of the main object conforms to the truncated log-normal distribution. 3) Size of embedded objects: embedded objects refer to objects such as pictures, text, and sound embedded in the page frame. The size of the embedded objects also conforms to the truncated log-normal distribution. 4) The number of embedded objects: the number of embedded objects conforms to the truncated Pareto distribution. 5) Reading time: represents the time interval of each webpage, similar to the OFF stage in the ON/OFF model, corresponding to the stage where the user is browsing the webpage. The reading time conforms to the exponential distribution. 6) Parsing time: After sending the main object of the web page to the client, the server will start to send the embedded object part of the web page in sequence after a parsing time. The Parsing time conforms to an exponential distribution.

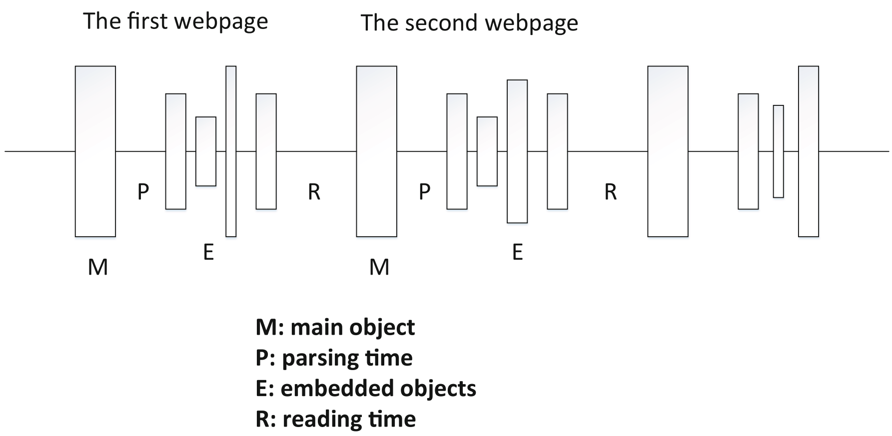

Fig. 3. Basic flow of the HTTP business model

4 Experiment

4.1 Architecture Verification

This experiment is to verify that the traffic generation system under this architecture will not adversely affect the performance of the network under test. Two DTN node topology experiments are designed to measure the delivery time under different link conditions. Two sets of comparative experiments were measured separately: changing the size of the packet loss rate under a fixed one-way delay (250 ms), and changing the size of the one-way delay under a fixed packet loss rate (10e−7). Raw means a two-node transmission experiment using the above system architecture, and mod means a normal two-node transmission experiment. The experimental results are shown in Table 1 and Table 2 below.

Table 1. Delivery time changes with packet loss rate under fixed one-way delay

10e−7		10e−6		5 * 10e−6		10e−5	
Raw	Mod	Raw	Mod	Raw	Mod	Raw	Mod
1.7898	1.7228	3.7347	3.4398	4.1696	4.2874	6.2710	6.6519

Table 2. Delivery time changes with one-way delay under a fixed packet loss rate

250 ms		500 ms		1 s		5 s	
Raw	Mod	Raw	Mod	Raw	Mod	Raw	Mod
1.7898	1.7228	2.7454	2.7854	6.5370	6.7838	30.4498	30.9653

The above experimental results indicate that when the traffic generation system adopting this system architecture measures the performance of the network under test, it will not affect the performance of the network under test. To a certain extent, the correct availability of the traffic generation system using this system architecture is verified.

4.2 Related Experiments

The experimental part of this paper adopts a simplified system architecture, and designs a UI interface for interaction. The UI interface is mainly divided into three parts, namely the parameter setting part, the network performance part and the log display part. As shown in Fig. 4,

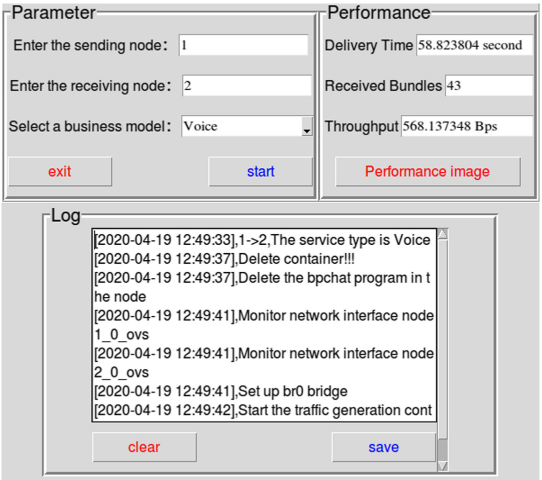

Fig. 4. UI interface

The parameter setting part is composed of three parameter items required for the experiment, which are the sending node number and receiving node number in the tested network, and the traffic service model. The drop-down box for the business model covers the several business models described above. In addition, the OK button in the parameter setting section indicates starting the experiment, and the Exit button indicates exiting the UI interface. The function of the network performance part is to display the relevant performance indicators of the tested network. Specifically, after the end of an experiment, according to the data packet information captured from the network interface of the sending node and the receiving node of the tested network, the three performance indicators of this experiment are counted, which are the delivery time and the number of Bundle and throughput value. In addition, the function of the view performance image button in this part is to display a line graph of each Bundle packet length and a real-time throughput line graph in this experiment. The function of the log display part is to display the experiment process in real time. As the experiment starts, the log display box will show the progress of the experiment in real time. In addition, the progress information will display different colors according to different degrees, general information is displayed in black, key process information is displayed in blue, and error information is displayed in red. At the end of each experiment, the log display box will print out the time of the first Bundle packet sent by the sending node and the time of the last Bundle packet received by the receiving node, and the total size of the bundle packets in the entire network transmission. These statistics can be mutually corroborated with the information in the network performance section. The function of the clear button in the log display part is to clear the contents of the current log box. The function of the save button will save the content displayed in the log display box under the logs folder, with the time as the file name at the time, similar to log2019-12-14-22-12-43.txt.

In this paper, a 4-node space information network is used as the tested network, and the node numbers are identified as 1, 2, 3, and 4, respectively. The space information network adopts a linear topology, that is, data transmission is performed according to node1 → node2 → node3 → node4. The parameter settings are shown in Fig. 4, with node1 as the sending node and node2 as the receiving node in the network under test, using voice service. After the experiment starts, first delete the configuration of the previous experiment, including the old traffic generation node and the bpchat process in the topology node. The bpchat process is originally a program for inter-node communication in the ION software, and the standard input was used as the program input for data transmission. In this experiment, the bpchat program was modified so that it can receive UDP traffic as the server of Socket communication, and perform protocol conversion and encapsulation into bundle packets for transmission on the DTN network under test. Secondly, the Docker technology is used to generate the traffic generation node container, and it is connected to the sending node through the OVS bridge. At the same time, multithreading is started to capture packets from the relevant network interfaces of the sending node and the receiving node. Finally, the bpchat process on the sending node is started, and the traffic generating node runs a packet sending program to transmit UDP traffic conforming to the selected business model to the sending node. After the transmission process is completed, statistical analysis is performed on the captured data packets to display network performance statistics. At this point, a measurement experiment of the flow generation system is ended. Figure 5 is a network performance image obtained in this experiment.

In addition, in order to ensure the accuracy of the traffic collection function in the experiment, this paper simultaneously opens Wireshark in the experiment to capture packets. The captured network interfaces are eth0 of the sending node and eth0 of the receiving node in the topology node. The eth0 is responsible for the transmission of Bundle packets. As can be seen from the captured Bundle packet information, it is true that node 1 (as the sending node) transmitted the bundle packet to node 2 (as the receiving node), which is in accordance with the parameters set in the experiment, and the data packet that has undergone protocol conversion at this time is carried out through the bundle protocol transmission. By filtering bundle packages, you can observe that there are indeed 43 bundle packages displayed, which is consistent with the number of bundle package receptions in the UI interface. Moreover, the start time and end time of the Bundle package also coincide with the log box information. In addition, the load content of the bundle package is also consistent with the load content set in the business model. Through the above comparative analysis, the normal operation of the overall

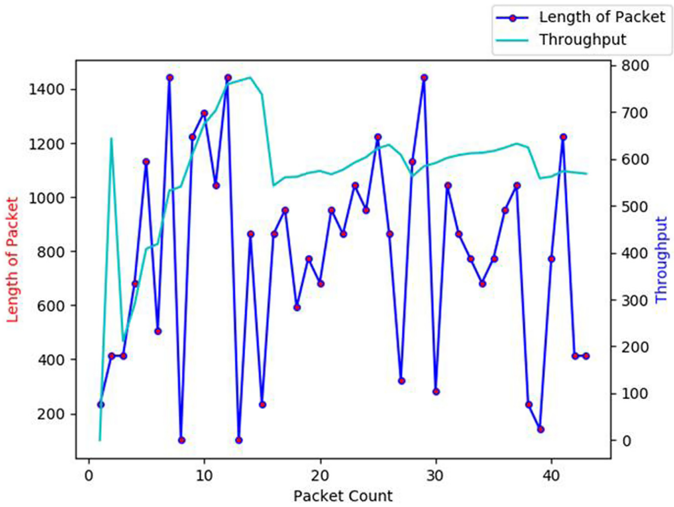

Fig. 5. Network performance image

traffic generation system is ensured, and the accuracy of the relevant module design and operation is verified.

In addition, on the basis of the above experiments, the traffic generation system of this paper is applied to the simulation platform in literature [1]. And a scenario where two ground nodes cannot communicate through a TCP/IP network is simulated, and the space information network is used as a relay network to realize data transmission. The specific experimental realization is that the ground node uploads the relevant UDP service traffic, converts it into Bundle packet and transmits it in the space information network, and finally converts it into UDP packet and sends it to another ground node. In this simulation scenario, there are mainly two nodes representing the ground nodes pgen and recv, and the space information network topology remains unchanged. This experiment separately conducts experiments on the above five types of business traffic. During the experiment, the information of the sending node of UDP traffic and the receiving node of UDP traffic is mainly concerned. In the experiment, Wireshark is used to capture and save the package, and then use Python's pyshark module and matplotlib module to analyze and draw, and get the relevant experiment comparison chart. As shown in Fig. 6.

Figure 6 selects the NRTV service model used. The upper line chart in the figure shows the size distribution of the UDP packets sent by the pgen node to the space information network node. The lower line chart in the figure shows the size distribution of the UDP packets received by the recv node from the space information network node. As can be seen from Fig. 6, the two figures can be completely coincident, indicating that after the UDP data packet sent from the pgen node is transmitted through the space information network, it can be completely restored to the original data packet size and transmitted to the recv node. It is verified that the two terrestrial nodes can be connected through the space information network when the terrestrial network is poor. However, the time attribute of the data packet received by the recv node is inevitably different

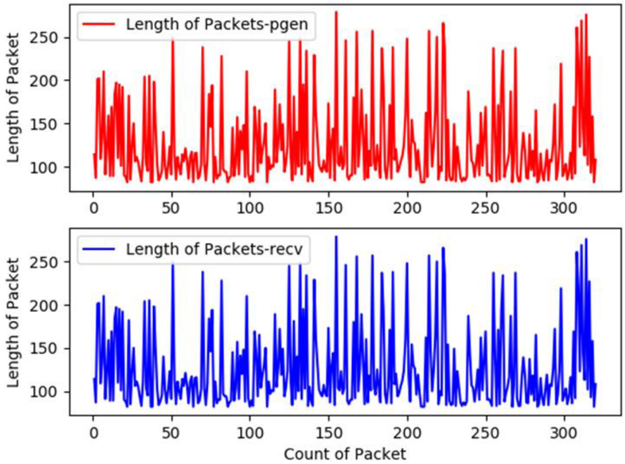

Fig. 6. NRTV service comparison chart

from the sending node. The experimental results show that the traffic generation system designed in this paper can accurately simulate the emergency communication scenario when applied to the simulation platform. Furthermore, it shows that the traffic generation system designed in this paper can effectively access to the simulation platform and undertake the task of traffic generation.

5 Conclusion

This paper studies and analyzes the development status of the traffic generation system, applies emerging network technology, and designs a multi-service traffic generation system architecture suitable for space information networks, which can meet requirements for traffic generation under the large-scale network. The traffic generation system architecture has good low coupling and high flexibility. In addition, in order to solve the problem that the simulation platform cannot simulate business scenarios, the traffic generation system of this paper implements a business traffic module at the traffic generation node, which is used to generate UDP traffic that conforms to the corresponding business characteristics. In addition, this paper has made corresponding verification experiments on the system architecture and operation process, showing that the traffic generation system can be well applied to the simulation platform and undertake the function of traffic generation. The space information network traffic generation system supporting multi-services proposed in this paper has certain reference significance for the subsequent research on multi-service research and traffic generation system in space information networks.

References

1. Lu, T.: An emulation architecture of routing protocol in space information network. Nanjing University (2018)

2. Tirumala, A., Qin, F., Dugan, J., Ferguson, J., Gibbs, K.: iPerf: TCP/UDP bandwidth measurement tool (2005)
3. Sommers, J., Barford, P.: Self-configuring network traffic generation, pp. 68–81 (2004)
4. Botta, A., Dainotti, A., Pescape, A.: A tool for the generation of realistic network workload for emerging networking scenarios. Comput. Netw. **56**(15), 3531–3547 (2012)
5. Wen, O., Zhang, X., Wang, D., Zhang, J., Tang, J.: A survey of network traffic generation. In: International Conference on Cyberspace Technology. IET (2016)
6. Sanh, M., Schmidt, E.G., Giiran, H.C.: FPGEN: a fast, scalable and programmable traffic generator for the performance evaluation of high-speed computer networks. Perform. Eval. **68**(12), 1276–1290 (2011)
7. Wang, X., Wang, Y., Li, P.: An aggregated process-based traffic generator for network performance evaluation. J. Circ. Syst. Comput. **25**, 1650018 (2015)
8. Wang, Z.: The research on virtualization technology of multi-user network traffic generator. Chongqing University of Posts and Telecommunications (2017)
9. Cao, B.: The design and implement of muLti-service traffic generator. Jilin University (2011)
10. Li, K.: Multi-service traffic generator based on Qt. Chongqing University (2016)
11. Amanatidis, T., Malkotsis, A.: DTN-tg A DTN Traffic Generator. In: Koucheryavy, Y., Mamatas, L., Matta, I., Tsaoussidis, V. (eds.) WWIC 2012. LNCS, vol. 7277, pp. 374–379. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30630-3_35
12. Zhang, X., Wang, W.: Multi-user mixed business modeling and business traffic analysis. Beijing University of Posts and Telecommunications (2008)
13. Zhu, Y., Sheng, M., Li, J.: Traffic Modeling and performance analysis for remote sensing satellite networks. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2018)