A Space Information Service Forwarding Mechanism Based on Software Defined Network

Zhou Wei* and Baokang Zhao
College of Computer, National University of Defense Technology, Changsha, 410073, China
{zhouwei14, bkzhao}@nudt.edu.cn

Abstract

Space information network includes multi-system integration that involves real-time acquisition, transmission, and processing over the information systems. Space information network is an important strategic direction of China's 13th Five-Year Plan for the innovation of science and technology, which is also considered to be the current focus area of national development. Efficient carrying and the information integration amongst different information services are the major challenges in the space information networks, which are constantly affecting the performance of system integration. Motivated by these challenges, in this paper, a Space Information Service (SIS) forwarding mechanism is proposed that is completely based on Software-Defined Network (SDN). According to the complexity of the information service demand, the designing of label tunnel control mechanism and packet/label mixed forwarding methodologies were executed by referring the concepts of SDN. Simulation results revealed that the SIS could efficiently achieve high information traffic delivery along with the management of the network tunnel.

Keywords: Satellite, Space Information Networks, SDN, Services, Geosynchronous Equatorial Orbit

1 Introduction

Space information network supports various types of space platforms as a carrier through an efficient integration of multi-systems [5].Space information network includes Geosynchronous Equatorial Orbit (GEO) communications satellites, navigation satellites, remote sensing satellites, jets, ground users, and ground control centers, etc. According to composition of the space information network, relay satellites and communication satellites operating in different orbits form the backbone of the space-based network, which is the core part to the system. Remote sensing satellites and navigation satellites utilize space-based backbone network to transmit the service data, and the whole system is controlled and maintained by the network operation control center installed on the ground [10] [7] [11].

Figure 1 illustrates one sample scenario of space information network. According to the property of each subsystems, the network can be divided into the core network, access network, user subnet and user control center.

- Core network consists of GEO communication/relay satellites.
- Access network consists of the MEO and LEO communication satellites. User subnets can access the network either through the access network or directly through the GEO communications/relay satellites.

Journal of Internet Services and Information Security (JISIS), volume: 7, number: 3 (August 2017), pp. 48-60

[*] Corresponding author: College of Computer, National University of Defense Technology, Changsha, 410073, China, Tel: +86-0731-84575820

- User subnets includes remote sensing satellites, remote sensing satellites subnet, space stations, ground network and an individual terminals.
- This bullet point statement must be continued with the previous statement or it should be rectified separately.

The resources in space wireless channel are limited in contrast to the terrestrial networks' transmission medium such as cables, twisted pair, fiber, etc. When the transmissions are extended to the field of space, system will be affected by the ionosphere, thermal radiation, earth magnetic field, space environment, and mobility [8]. Because of these characteristics/factors of the vast airspace, the data transmission is continuously suffering from a long delay problem.

In terrestrial networks, IP-based next-generation satellite network has many shortcomings such as the slow convergence of the distributed routing algorithm, lack of global vision, complex network configuration, non-flexible configuration updates, and lack of management of network services. In addition, the characteristics of satellite networks also bring more challenges to network performance. The network dynamic topology attributes to more unstable routing, and as well the interaction of longer satellite distance and satellite over-the-top transmission of data makes slow updates in global network routing. Thus, the problem of utilizing the space information network requires an efficient solution that can provide a highly effective and flexible service delivery process.

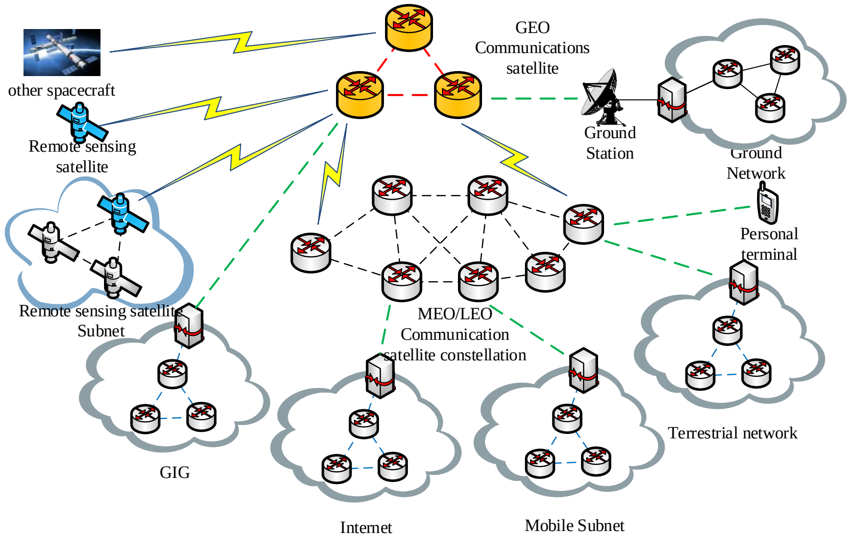


Figure 1: An illustration of the space information network.

Targeting at the heterogeneity of space information network, the characteristics of multitype traffic and heterogeneous demand of subnet performance are analyzed. In this paper, the authors proposed a novel SIS forwarding mechanism for achieving the demands of multi-service forwarding of space information network. This mechanism mainly includes numerous advantages: fully combine the characteristics of space information network system, adopt the software defined network method, realize the centralized and flexible control of the network [14] [17]; use the packet /label hybrid forwarding method, labelbased forwarding avoids complex routing table queries and hide the heterogeneity of multiple Layer-2, effectively achieve quality of service of multi-service. Finally, the simulation results revealed that SIS could realize the efficient and flexible forwarding of multi-service of space information network.

The rest of paper is structured as follows. Section II gives a brief about the related works. In section III, the system architecture is elaborated in detail. Section IV illustrates the experimental settings and simulation results of our work. Finally, section V concludes our work and discuss some of the future scope in this work.

2 Related Work

In 1996, NASA had a considerable size of the wide area monitoring network. Nascom, PSCN, NSI, AEROnet, EOSDIS and Bnet combined to form NASA integrated services network. In 2004, NASA released a detailed report on the network architecture and technology for the future space information network formation [3]. The United States proposed the Global Information Grid plan (GIG) for 2020 to achieve end-to-end seamless connectivity capabilities and facilitating with future wide-area spatial information sharing.

In 2009, for the first time, Intelsat14 high-orbit satellites were equipped with space routers to achieve space routing and testing for switching technology. China is vigorously promoting the rapid development of space technology, published as “national space infrastructure, long-term development plan (2015-2025)” and other related strategic documents. Space information networks heavily rely on the channel utilization. An efficient mechanism can overcome the issue of delayed transmission in the information space networks [16]. These networks suffer from the same issues as observed by the conventional networks, which can be overcome by utilizing the virtualization technology [18]. Routing is another paradigm to be handled with these software enabled networks for sustaining the connectivity between the hosts [6].

Bertaux et al. [2] presented a detailed study on the implantation of the SDN-based network visualization for broadband satellite networks. The authors discussed various use-cases to demonstrate the utilization and effectiveness of the SDN-based satellite networks. Li et al. [15] utilized the capabilities of SDN to form an efficient satellite network. The authors primarily focused on the feasibility of using SDN technology for satellite communication networks. Barritt and Eddy [1] focused on the SDN enhancement for LEO satellite networks. The authors used SDN technology to overcome the issues of dynamic routing protocols in LEO communications. However, their technique do not include any transmission criteria and tunnel formation to allow efficient data delivery between the hosts.

Irrespective of the use of SDNs, Jia et al. [12] provided an approach for collaborative data downloading using inter-satellite links. The authors improved the connectivity between the earth stations and satellite by improving the overall throughput of the network. Guo et al. [9] developed a directional routing algorithm for space networks by avoiding non-directional and blind transmissions. Chenji et al. [4] developed a cognitive architecture for space communication networks by integrating different technologies such as deep learning, cognitive radios and security modules. However, these approaches can be further improved by lowering the amount of decisions involved in identifying appropriate links by using a centralized software-controlled node.

3 Proposed Technique: SIS

This section presents various components of the proposed SIS model. SIS adopts SDN system architecture to achieve high-speed delivery of space information network service data. The following parts describe the details of each of these components.

3.1 Software Defined Network System Architecture

Based on the idea of software defined network centralized control, the core level is located on the control plane, as shown in Figure 2. The control plane, which is considered to be the most crucial part of the system architecture is also seen as the most complex part of an entire system. This part uses a link discovery protocol to obtain the global network state, including the network node topology and on-off states of dynamic nodes. The link discovery module uses the Link Layer Discovery Protocol (LLDP) to dynamically maintain global topology information. The global topology is maintained by the topology class. The shortest path is selected based on the priority and business modules. On the other hand, the routing module uses the topology class information to generate the corresponding service forwarding policies.

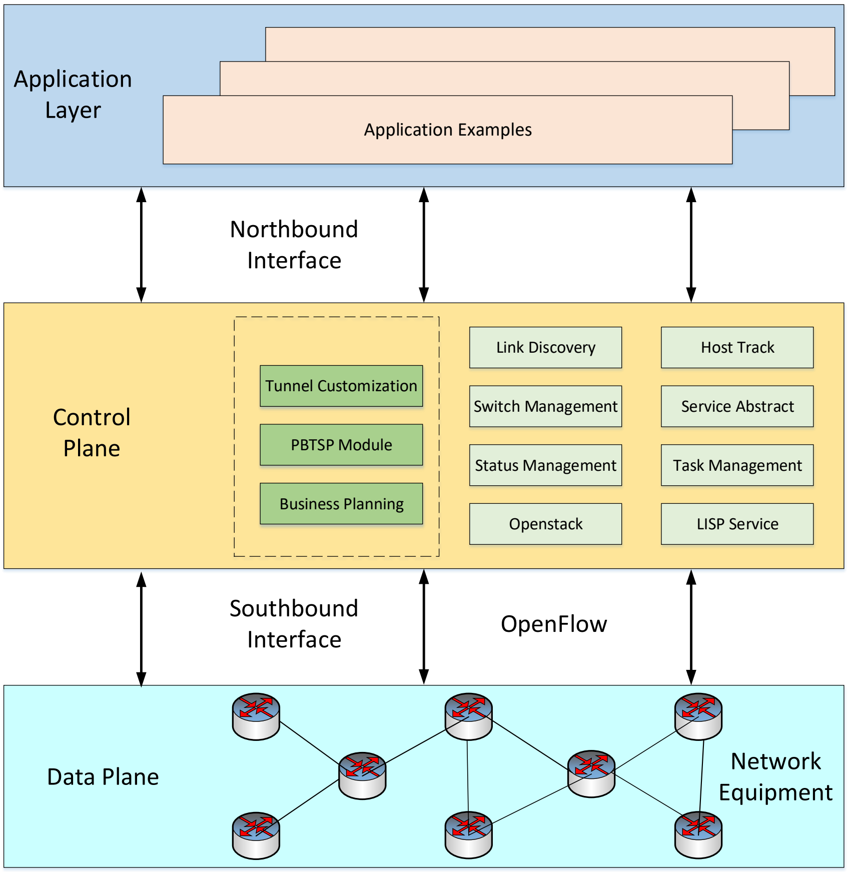


Figure 2: An illustration of the software defined network system architecture

In the initial case, the packet in the space information network depends on the routing protocol. Due to the diversity in the types of services in the space information network, the demand for service quality is higher than that of the terrestrial networks. Therefore, in the process of follow-up packet transmission, the dynamic control tunnel customization depends on the traffic characteristics of different tunnels. In addition, it adopts the flow control method on the basis of flow identification to determine the packet/label hybrid forwarding mode. The research module includes tunnel customization module, PBTSP(based on Priority and Business Types of the Shortest Path) module, and business planning module.

3.2 Tunnel Customization Module

Different types of space information services are available in the tunnel for different bandwidth, priority needs, and control plane establishes a number of virtual channel connections. In the initial phase, the system statically configures several tunnels. According to the service requirements and initial tunnel, packet forwarding process forms a service label path and includes it between the Layer 2 header and the IP header, as shown in Figure 3. The Label field is 24 bits, TTL (Time To Live) field is 12 bits, and the packets are forwarded according to the corresponding label.

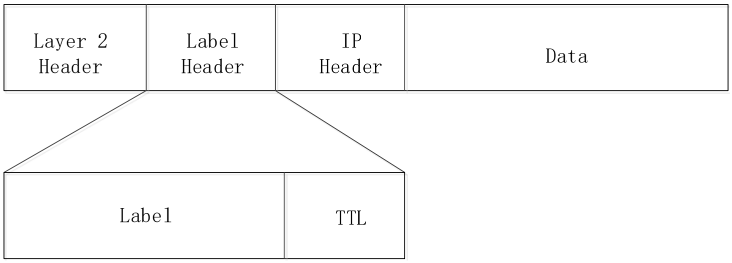


Figure 3: An illustration of the label header structure.

Table 1: Parameters of tunnel structure

parameter	Word length(byte)	Features
Tid	2	Tunnel ID
Hdpid	8	Tunnel Header Data plane ID
Tdpid	8	Tunnel Tail Data plane ID
Resbw	4	Tunnels reserved bandwidth
Priority	1	Tunnel priority
Traffictype	1	Tunnel traffic type

The tunnel structure includes unique identifier of the tunnel (Tid), data plane node identifier located at the head of the tunnel (Hdpid), data plane node identifier located at the end of the tunnel (Tdpid), size of the current reserved bandwidth of tunnel (Resbw), current tunnel label path priority (Priority), and the tunnel traffic types (Traffictype), as shown in Table 1. The tunnel direction is oriented by the end-to-end data plane node. The initial stage is the system startup, which includes the static establishment of the tunnel. According to the structure of the tunnel customization module, the basic parameters of the tunnel are set and the service tunnel is formed. Next phase includes the setting of the data plane node identifiers (dpid), backbone routers (BR), and edge router (ER). BR nodes are the core components of the space information network, and ER nodes connect to a wide variety of subnets.

After the tunnel parameters are determined, the tunnel establishment process is performed. The working of the tunnel formation is presented as a flowchart in Figure 4.

As soon as, tunnel initialize parameters, function “applyConfTunnels” is called to achieve tunnel route calculation. The function involves adding of tunnels one by one, setting of tunnel parameters to meet different way from beginning to ending along with the current configurations of the tunnel. After the initial tunnel parameters are read, the label path computation procedure calls PBTSP modules on the basis of the priority and reserved bandwidth for determining the best path for the tunnel. The path calculation process determines the need to re-schedule the tunnel path and strip the tunnel.

An illustration of the path generation is shown in Figure 5. Two tunnels are designed; one between BR1 and BR0, and the other between BR1 and BR3. Tunnel connection implies a virtual link, non-solid physical link, and the physical process through the multiple BR nodes. According to the physical node, PBTSP module calculates the shortest path, which is considered as a virtual tunnel node.

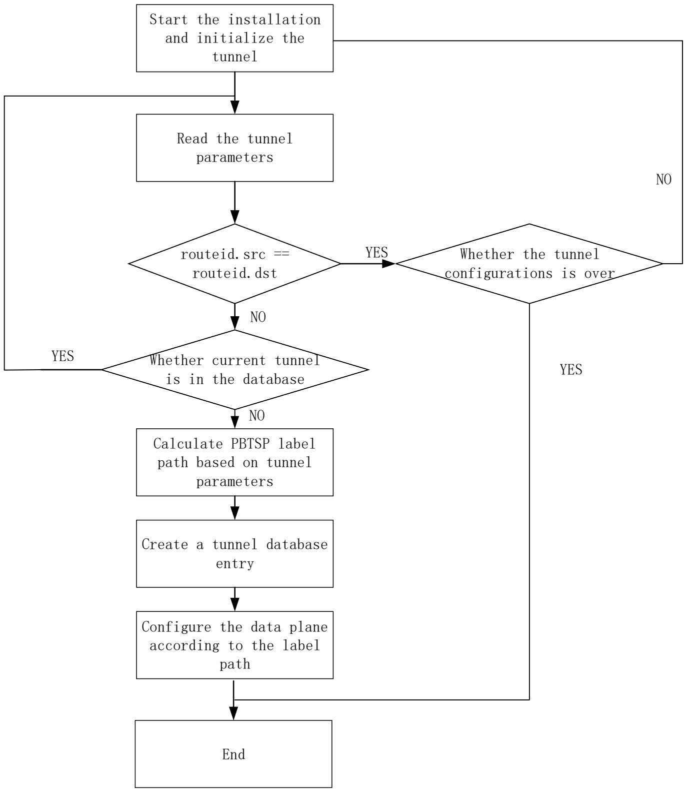


Figure 4: Tunnel establishment flow chart.

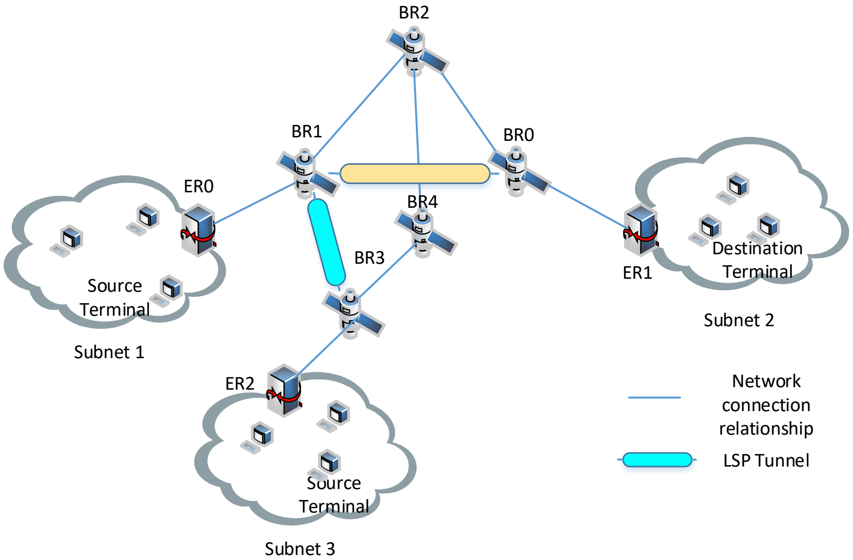


Figure 5: An illustration of tunnel to establish a network topology map.

3.3 PBTSP Module

PBTSP module combines topology information from initial data of the tailored tunnel by using Dijkstra shortest path algorithm [13]. This data is utilized for service computing to meet the bandwidth and priority of the label path. In the data plane, when a packet stream cannot determine the flow of data to the interface, it sends OFPT_*PACKET*_IN to the controller through the Southbound interface, and the controller calls *HANDLE _PACKET _IN* function, after which the packet stream is processed.

In the process of label path calculation, according to the packet stream flows and priority needs, it is necessary for an initial tunnel to judge the bandwidth capacity and traffic types using the function prototype named as tunnel_fit. When the tunnel meets the requests of the demand, it uses Dijkstra algorithm to calculate the routing path. Using the OpenFlow protocol, flow table entries of data plane node can be configured to generate the related forwarding entries. An illustration of the PBTSP basic flowchart is illustrated in Figure 6.

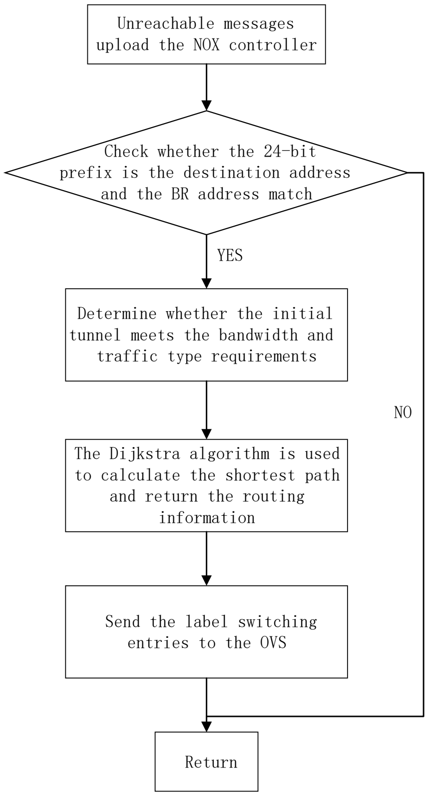


Figure 6: PBTSP basic flowchart.

3.4 Business Planning Module

The different types of services in the space information network include voice services, data services, video services, image services. Different types of services possess different Quality of Service (QoS) requirements in terms of bandwidth, delay, packet loss rate, etc. as shown in Table 2.

In terms of bandwidth, data services and image services are mainly obtained from remote sensing satellites, where the data transmission rate of existing remote sensing satellites are mostly a few 100 Mbps. In addition, The data rates for voice and video services are about tens and hundreds of kbps. It

Table 2: Different types of Quality of Service (QoS) requirements

business type	bandwidth	Time delay	Packet loss rate	Examples of services	Business coding
Voice services	low	low	high	Voice telephone	0x0001
Data services	high	high	medium	Remote sensing	0x0002
Video services	medium	low	high	video conference	0x0004
Image business	high	high	medium	Remote sensing image	0x0008

can be clearly seen from Table 2 that the bandwidth requirement of voice and video services is smaller than that of data and image services.

The voice and video services have high demanding requirements on real-time and require lower delay. Data services require high reliability of data transmission to ensure the correct transmission of data, whereas voice and video services can tolerate a certain degree of packet loss. To support the realtime applications, voice and video services use label switching, whereas data and image services use packet exchange policies.

4 Experimental Setup and Results

The experimental evaluation process of the service forwarding mechanism for the space information network is conducted using different emulations. By simulating the real environment scene of the space information network, the core network, the access network and the user subnets are visualized. The network internal controller node and data plane nodes are designed by specifying the role for each node.

Topology emulation environment, as shown in Figure 7, comprises a single controller mainly located in the internal functioning of GEO satellites.

BR characters are usually seen as a part of the core network backbone nodes, which is mainly responsible for service data forwarding. Part of the access network nodes that contain ER characters are responsible for different types of user's access subnets. The user terminal subnets (such as personal terminals) are the data source terminals and destination terminals. During the phase of evaluation testing, the source terminal sends packet messages to the destination terminal with a certain bandwidth and type. The packets are classified according to the requirements of the service demands, and the related test items are finally verified.

Controller starts the phase and sends the related control commands to each of the above components for initiating the phase of transmission. Initial exchange process is between the controller and the data plane node, which runs according to the capabilities of OpenFlow protocol. It maintains a connection and information matching between the controller and the data plane nodes for establishing the transmission agreement. As shown in Figure 8, Wireshark is used to capture the OpenFlow messages.

According to the tunnel parameter configurations, the initialization phase loads the configuration parameters to initialize the tunnel. The test topology includes H21, H22 and H23 from different subnets of the different network. ER0, ER1, ER2 are the boundary data plane nodes located at the boundary of subnets and core network. BR0, BR1, BR2, BR3 and BR4 form the core backbone of the network. The initial configuration of the tunnel takes place inside the controller. After reading the configuration information, the controller establishes the corresponding tunnel.

In this network topology, two tunnels are established. These tunnels are one-way static tunnels with numbers bearing 0x7e00 and 0x7e01. The former allows the service types 0x0001 and 0x0002, and the later allows any type of service types included. After determining the basic information, the controller sets the data plane node's flow table through OFPT_FLOW_MOD. This is done by calculating the optimal shortest path based on the label PBTSP constraints.

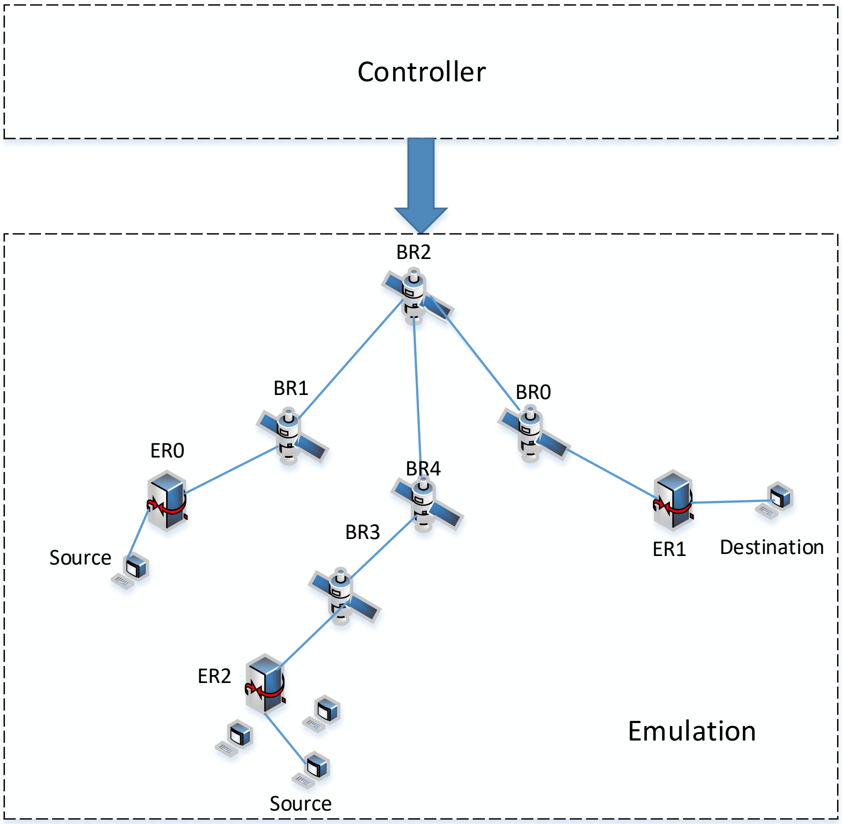
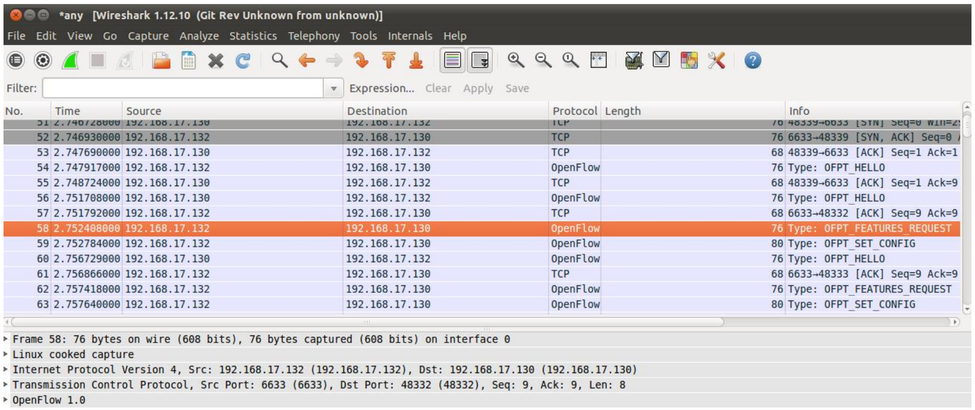


Figure 7: Emulations: network topology



```
No.  Time        Source          Destination     Protocol Length Info
  51 2.746728000 192.168.17.130  192.168.17.132  TCP       76 48339→6633 [SYN] Seq=0 Win=2
  52 2.746930000 192.168.17.132  192.168.17.130  TCP       76 6633→48339 [SYN, ACK] Seq=0
  53 2.747690000 192.168.17.130  192.168.17.132  TCP       68 48339→6633 [ACK] Seq=1 Ack=1
  54 2.747917000 192.168.17.132  192.168.17.130  OpenFlow  76 Type: OFPT_HELLO
  55 2.748724000 192.168.17.130  192.168.17.132  TCP       68 48339→6633 [ACK] Seq=1 Ack=9
  56 2.751708000 192.168.17.130  192.168.17.132  OpenFlow  76 Type: OFPT_HELLO
  57 2.751792000 192.168.17.132  192.168.17.130  TCP       68 6633→48332 [ACK] Seq=9 Ack=9
  58 2.752408000 192.168.17.132  192.168.17.130  OpenFlow  76 Type: OFPT_FEATURES_REQUEST
  59 2.752784000 192.168.17.132  192.168.17.130  OpenFlow  80 Type: OFPT_SET_CONFIG
  60 2.756729000 192.168.17.130  192.168.17.132  OpenFlow  76 Type: OFPT_HELLO
  61 2.756866000 192.168.17.132  192.168.17.130  TCP       68 6633→48333 [ACK] Seq=9 Ack=9
  62 2.757418000 192.168.17.132  192.168.17.130  OpenFlow  76 Type: OFPT_FEATURES_REQUEST
  63 2.757640000 192.168.17.132  192.168.17.130  OpenFlow  80 Type: OFPT_SET_CONFIG

▸ Frame 58: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.17.132 (192.168.17.132), Dst: 192.168.17.130 (192.168.17.130)
▸ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 48332 (48332), Seq: 9, Ack: 9, Len: 8
▸ OpenFlow 1.0
```

Figure 8: An illustration of the data plane node controller and connection establishment packet capturing process.

The optimal paths of the two tunnels are as follows:

- Tunnel 0x7e00: tunnel path BR1-BR2-BR0, configures the label forwarding table in the data plane corresponding to the path information.
- Tunnel 0x7e01: tunnel path BR1-BR2-BR4-BR3, configures the label forwarding table in the data plane related to the path information.

After successfully establishing the tunnels, the source host uses the packetgen.py to send the related traffic packets across various subnets. According to the characteristics of the two tunnels, the host sends three service data traffics from H21 to H22, which are numbered as 0x0001, 0x0002, 0x0008 respectively. H21 belongs to the subnet 192.168.10.0/24, H22 hosts on the segment 192.168.1.0/24, and H23 belongs to subnet segment 192.168.30.0/24 respectively. Network topology traffics are shown in Figure 9. H21 transmits the same three types of data traffics to H23. According to tunnel setup paths, 0x7e01 tunnel allows all types of traffic, and therefore, the flow through this tunnel is fully utilized by the label forwarding process. Since, 0x7e00 tunnel runs only 0x0001 and 0x0002, the H21 to H23 utilizes these two services to meet the needs of data traffic forwarding label. 0x0008(for service data flow) does not belong to the tunnel type, hence, it is used for packet forwarding.

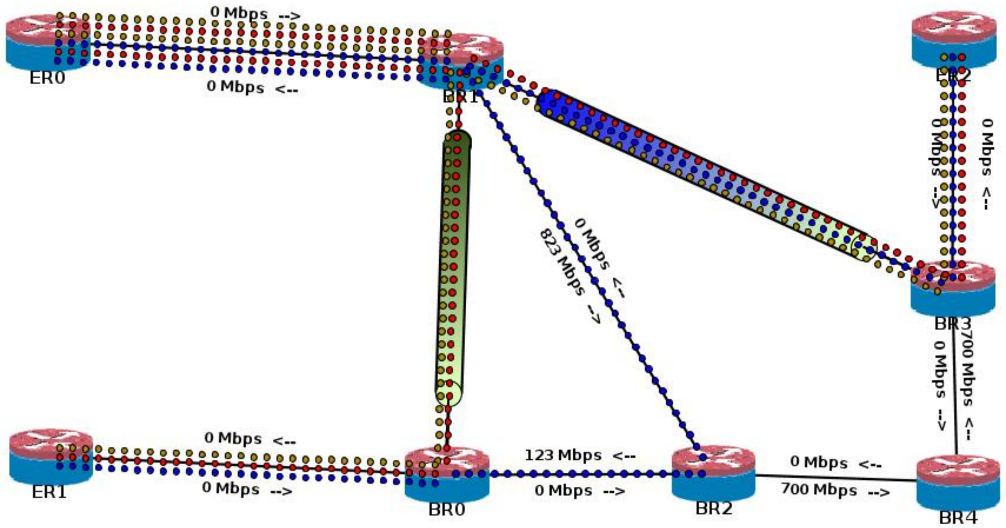
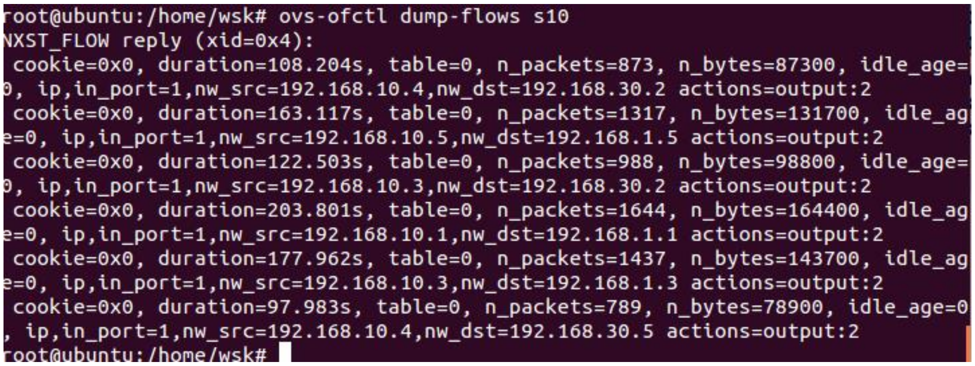


Figure 9: An illustration of the tunnel establishment process.

The data plane node entry submissions are based on the OFPT_FLOW_MOD function of the OpenFlow protocol. Note that, the tunnel establishment between the core backbone data plane nodes follows BR to BR relationship. Since, ER0 data plane has no internal flow table for forwarding rules, therefore, it sends OFPT_PACKET_IN by OpenFlow protocol packets to the controller, and the controller uses handle_packet_in to process it. After the controller calculates the forwarding mode, it sends the entry to the corresponding data plane node.

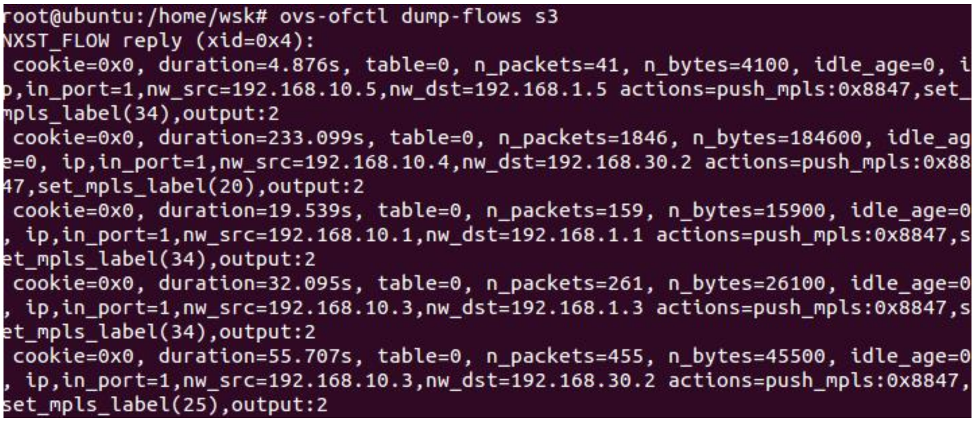
As shown in Figure 10, flow table contents of ER0 can be found through the command of ovs-ofctl dump-flows, indicating that, it has received the controller's control information. It matches the packets according to the three tuples namely, incoming port, destination IP address and source IP address. In addition, it determines the outgoing port corresponding to the packet, which is no more a part of the tunnel.

According to the custom tunnel, when the packet is forwarded from ER0's port 2, the tunnel goes into the follow-up label forwarding process. Label forwarding is mainly performed on the data plane node corresponding to the tunnel. After entering the tunnel, BR1 node adds the label header (Ethernet header type field replaces 0x8847), and the new forwarding entries, as shown in Figure Figure 11. Subsequent traffic transfer process can be conducted in various ways by using BR label switching node along with the tunnel. Eventually forwarded to the corresponding ER node, packets are dismantled from the



```
root@ubuntu:/home/wsk# ovs-ofctl dump-flows s10
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=108.204s, table=0, n_packets=873, n_bytes=87300, idle_age=
0, ip,in_port=1,nw_src=192.168.10.4,nw_dst=192.168.30.2 actions=output:2
 cookie=0x0, duration=163.117s, table=0, n_packets=1317, n_bytes=131700, idle_ag
e=0, ip,in_port=1,nw_src=192.168.10.5,nw_dst=192.168.1.5 actions=output:2
 cookie=0x0, duration=122.503s, table=0, n_packets=988, n_bytes=98800, idle_age=
0, ip,in_port=1,nw_src=192.168.10.3,nw_dst=192.168.30.2 actions=output:2
 cookie=0x0, duration=203.801s, table=0, n_packets=1644, n_bytes=164400, idle_ag
e=0, ip,in_port=1,nw_src=192.168.10.1,nw_dst=192.168.1.1 actions=output:2
 cookie=0x0, duration=177.962s, table=0, n_packets=1437, n_bytes=143700, idle_ag
e=0, ip,in_port=1,nw_src=192.168.10.3,nw_dst=192.168.1.3 actions=output:2
 cookie=0x0, duration=97.983s, table=0, n_packets=789, n_bytes=78900, idle_age=0
, ip,in_port=1,nw_src=192.168.10.4,nw_dst=192.168.30.5 actions=output:2
root@ubuntu:/home/wsk# 
```

Figure 10: An illustration of ER0 flow table contents.



```
root@ubuntu:/home/wsk# ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=4.876s, table=0, n_packets=41, n_bytes=4100, idle_age=0, i
p,in_port=1,nw_src=192.168.10.5,nw_dst=192.168.1.5 actions=push_mpls:0x8847,set_
mpls_label(34),output:2
 cookie=0x0, duration=233.099s, table=0, n_packets=1846, n_bytes=184600, idle_ag
e=0, ip,in_port=1,nw_src=192.168.10.4,nw_dst=192.168.30.2 actions=push_mpls:0x88
47,set_mpls_label(20),output:2
 cookie=0x0, duration=19.539s, table=0, n_packets=159, n_bytes=15900, idle_age=0
, ip,in_port=1,nw_src=192.168.10.1,nw_dst=192.168.1.1 actions=push_mpls:0x8847,s
et_mpls_label(34),output:2
 cookie=0x0, duration=32.095s, table=0, n_packets=261, n_bytes=26100, idle_age=0
, ip,in_port=1,nw_src=192.168.10.3,nw_dst=192.168.1.3 actions=push_mpls:0x8847,s
et_mpls_label(34),output:2
 cookie=0x0, duration=55.707s, table=0, n_packets=455, n_bytes=45500, idle_age=0
, ip,in_port=1,nw_src=192.168.10.3,nw_dst=192.168.30.2 actions=push_mpls:0x8847,
set_mpls_label(25),output:2
```

Figure 11: An illustration of BR1 flow table contents.

label header. In addition, subsequent packet forwarding based on the IP is performed, and packets are forwarded to the corresponding host node.

5 Conclusion and Future Work

This paper introduces the space information service (SIS) forwarding mechanism based on the software defined network. The authors presented the existing software-defined network technology framework, and subsequently, utilizes the properties of control and data planes to form the tunnel control and space data plane for the efficient delivery of packets. The emulation results revealed that the label tunnel control can dynamically control the tunnel through the periodic flow state feedback information of the data plane nodes . It is done based on the associated tunnels with different priorities, reserved bandwidth and service types. Mixed packet forwarding mechanism based on packet/label, modifies the flow table entries in the data plane node according to a certain policy for the realization of flexible traffic control and efficient delivery. In the future, on the basis of improving the pre-function, we will focus on the reliability of forwarding, sdn network management security.

References

[1] B. J. Barritt and W. Eddy. Sdn enhancements for leo satellite networks. In *Proc. of the 34th AIAA International Communications Satellite Systems Conference (ICSSC'16), Cleveland, Ohio, USA*, pages 5755–5764.

AIAA, October 2016.

[2] L. Bertaux, S. Medjiah, P. Berthou, S. Abdellatif, A. Hakiri, P. Gelard, F. Planchou, and M. Bruyere. Software defined networking and virtualization for broadband satellite networks. *IEEE Communications Magazine*, 53(3):54–60, March 2015.

[3] K. Bhasin and J. Hayden. Developing architectures and technologies for an evolvable nasa space communication infrastructure. Technical Report 20040086724, NASA Glenn Research Center, May 2004.

[4] H. Chenji, G. Stewart, Z. Wu, A. Javaid, V. Devabhaktuni, K. Bhasin, and B. Wang. An architecture concept for cognitive space communication networks. In *Proc. of the 34th AIAA International Communications Satellite Systems Conference (ICSSC'16), Cleveland, Ohio, USA*, pages 5728–5741. AIAA, October 2016.

[5] J. Du, C. Jiang, Q. Guo, M. Guizani, and Y. Ren. Cooperative earth observation through complex space information networks. *IEEE Wireless Communications*, 23(2):136–144, April 2016.

[6] S. El Alaoui and B. Ramamurthy. Routing optimization for dtn-based space networks using a temporal graph model. In *Proc. of the 2016 IEEE International Conference on Communications (ICC'16), Kuala Lumpur, Malaysia*, pages 1–6. IEEE, May 2016.

[7] R. Ferrús, H. Koumaras, O. Sallent, G. Agapiou, T. Rasheed, M.-A. Kourtis, C. Boustie, P. Gélard, and T. Ahmed. Sdn/nfv-enabled satellite communications networks: Opportunities, scenarios and challenges. *Physical Communication*, 18:2:95–2:112, March 2016.

[8] J. Glückler, E. Lazega, and I. Hammer. Exploring the interaction of space and networks in the creation of knowledge: An introduction. In *Knowledge and Networks*, volume 11 of *Knowledge and Space*, pages 1–21. Springer, Cham, January 2017.

[9] L. Guo, X. Wang, Y. Liu, P. Han, Y. Xie, and Y. Tan. Directional routing algorithm for deep space optical network. *China Communications*, 14(1):158–168, January 2017.

[10] F. He, Q. Liu, T. Lv, C. Liu, H. Huang, and X. Jia. Delay-bounded and minimal transmission broadcast in leo satellite networks. In *Proc. of 2016 IEEE International Conference on Communications (ICC'16), Kuala Lumpur, Malaysia*, pages 1–7. IEEE, May 2016.

[11] Y. Hu, Q. Zhou, J. Li, Y. Zhang, J. Li, and T. Cui. Architecture and protocols for satellite sdh networks. *International Journal of Future Computer and Communication*, 5(5):210–213, October 2016.

[12] X. Jia, T. Lv, F. He, and H. Huang. Collaborative data downloading by using inter-satellite links in satellite networks. *IEEE Transactions on Wireless Communications*, 16(3):1523–1532, May 2017.

[13] Y. Y. G. Jianya. An efficient implementation of shortest path algorithm based on dijkstra algorithm [j]. *Journal of Wuhan Technical University Of Surveying And Mapping (WTUSM)*, 3(4):208–211, January 1999.

[14] C.-S. Li and W. Liao. Software defined networks. *IEEE Communications Magazine*, 51(2):113–114, February 2013.

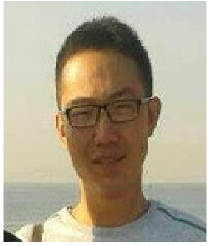
[15] T. Li, H. Zhou, H. Luo, Q. Xu, and Y. Ye. Using sdn and nfv to implement satellite communication networks. In *Proc. of the 2016 International Conference on Networking and Network Applications (NaNA'16), Hakodate, Japan*, pages 131–134. IEEE, July 2016.

[16] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh. Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view. *IEEE Communications Surveys & Tutorials*, 18(4):2442–2473, May 2016.

[17] Z. Tang, B. Zhao, W. Yu, Z. Feng, and C. Wu. Software defined satellite networks: Benefits and challenges. In *Proc. of the 2014 IEEE Computing, Communications and IT Applications Conference (ComComAp'14), Beijing, China*, pages 127–132. IEEE, October 2014.

[18] Q.-Y. Yu, W.-X. Meng, M.-C. Yang, L.-M. Zheng, and Z.-Z. Zhang. Virtual multi-beamforming for distributed satellite clusters in space information networks. *IEEE Wireless Communications*, 23(1):95–101, February 2016.

Author Biography



Wei Zhou received the master degree in the School of Computer Science, National University of Defense Technology in 2017. Currently he is taking a Ph.D's course at NUDT. His research interests include protocols, security issues in computer networks and machine learning.

Baokang Zhao is an Associate Professor in the School of Computer Science, National University of Defense Technology, where he received a Ph.D. degree in Computer Science in 2009. He has served as a program committee member for several international conferences and a reviewer for several international journals (including TC and TCAD). He serves on the editorial board of the Journal of Internet Services and Information Security. His current research interests include system optimization, protocols, algorithms, and security issues in computer networks. He is a member of the ACM, IEEE, and CCF.