OpenFlow Protocol Extension in SDN-based Satellite Networks

Zhanqi Xu[1], Miaomiao Feng[1], Jianxin Lv[2], Fan Yang[1], and Guo Chen[1]
[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China
[2] Dept. of advanced research, Fiberhome Communication Ltd., Wuhan, 430073, China
Email: zqxu@mail.xidian.edu.cn; 593833805@qq.com; ljx70@fiberhome.com; fany@xidian.edu.cn; 962332430@qq.com

Abstract**—**The function enhancements of different OpenFlow visions in match fields and statistics are paving the way for making OpenFlow as the most important southbound interface. However, in existing studies of software defined hybrid terrestrial-satellite networks, no attention has been paid to the multi-granularity switching used in GMPLS and performance parameters transmission of the wireless link while both aspects are of significance for the ever-increasing satellite application in mobile and rural scenarios. Therefore, an extension scheme of OpenFlow in these two aspects was proposed. By modifying the matching module, adding microwave port properties and new message type associated with microwave link parameters, the scheme was able to support the satellite multi-granularity switching and transmission parameter acquisition of wireless interfaces. The simulation prototype was built to check the feasibility of new message delivery.

Index Terms**—**Software Defined Satellite Networking (SDSN), OpenFlow, Protocol Extension, Simulation Implementation, Network Function Virtualization (NFV)

I. Introduction

To satisfy the requirements for more fine-grained switching, more flexible configuration and management [1], Software Defined Networking (SDN) [2] was introduced to satellite networks. The proposed architecture included Software-Defined Satellite (SDS) [3], Software defined satellite networks (SDSN) [4], OpenSAN [5] and SDN/NFV-based satellite networks [6], which aimed at solving such problems as inflexible configuration and update, complicated management and poor versatility of satellite networks. Since OpenFlow protocol is one of the most important protocols used widely between a controller and an SDN-enabled switch, its investigation is therefore inevitable and significant.

As the primary southbound interface protocol of SDN, the primitive OpenFlow protocol can only apply to Ethernet for communication between a controller and forwarding devices. Thus, such a protocol requires to be enhanced or extended when employing it to other networks. There are some researches about the extension on OpenFlow protocol based on the physical characteristics of optical networks and Optical Transport Network (OTN) equipment[7], and other improvements on matching and forwarding to support optical networks[8], [9]. In addition, some scholars have extended OpenFlow protocol to wireless networks [1012]. For example, to monitor and control a satellite, B. W. Chen proposed an OpenFlow extension scheme on satellite networks [13], extending the flow table matching module of OpenFlow protocol, and embedding the programmable module onto a satellite. However, it did not address the implementation. Since the diversity of different bandwidth requirements is changing from a wavelength of a feeder link to several tens Kbps of a conventional user, satellite networks should support multi-granularity switching of nodes. To guarantee users' Quality of Service (QoS), it is imperative and important to obtain the transmission parameters of feeder links for detecting the link transmission quality so that shifting between a microwave link and an optical link, and changing the modulation order are feasible. However, little attention has been paid to the OpenFlow applicability to satellite networks.

For promoting the integration of the terrestrial networks and SDSN, this paper proposed such an OpenFlow protocol extension scheme for SDSN as to enhance the function of the southbound interface protocol. This extension addressed the two aspects of multi-granularity switching and wireless transmission characteristics for terrestrial-satellite network.

II. Software Defined Satellite Networking

To construct the SDSN architecture shown in Fig. 1, SDN is introduced into the satellite networks by embedding SDN-enabled function onto satellite nodes and placing SDN controllers on the ground networks. This architecture contains the terrestrial MPLS subnet, ATM subnet, IP subnet, other networks, and networked GEO satellites that are connected via inter-satellite microwave and laser links. We use heterogeneous gateways to connect the four terrestrial networks to the satellite network. These gateways can be divided into two types: Label Switching Router (LSR) and Label Edge Router (LER), implementing the function of core forwarding and edge accessing in the MPLS domain, respectively.

Manuscript received December 15, 2018; revised July 2, 2019.
This work was supported by the Natural Science Foundation of China (No.61572391) and 111 Project Base (B080038).
Corresponding author email: zqxu@mail.xidian.edu.cn.
doi:10.12720/jcm.14.8.721-726

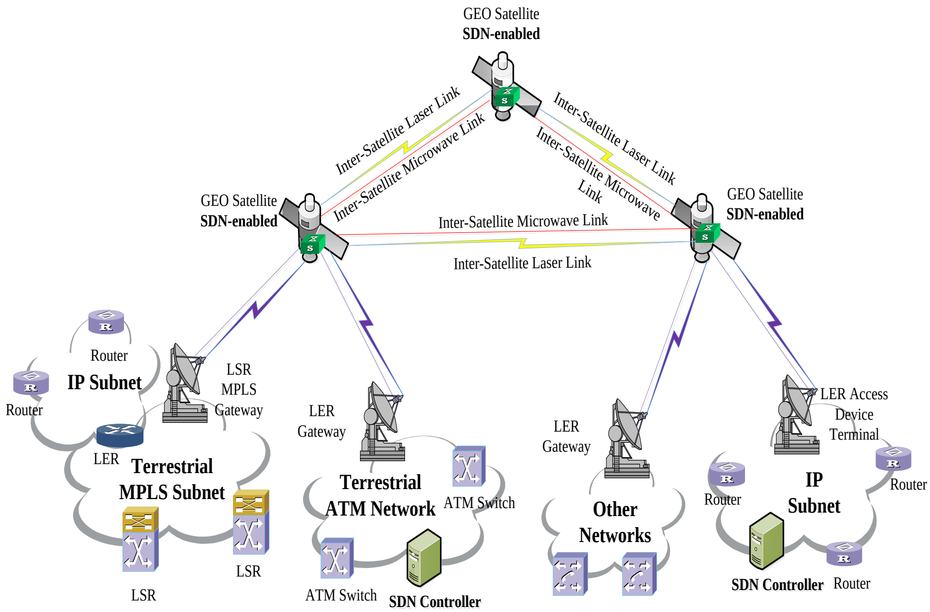


Fig. 1. The architecture of software defined satellite networking

The satellite nodes act as forwarding devices in the infrastructure layer, which supports IP, ATM and optical switching with different granularities according to the rules of flow table uploaded by the ground controller. The forwarding devices inside the subnets perform forwarding operation in turn with the policy issued by the controller. An SDN controller is responsible for the connection establishment, modification and release in hybrid satellite networks. It also performs the control and decision-making of the whole network according to its global view via Openflow protocol, monitoring the network status and scheduling satellite resources dynamically and rationally. Compared with traditional satellite networks, SDSN could make it feasible to provide finer granularity of network resource management, higher QoS to end users, network scalability, and lower capital expenditures (CAPEX) and operational expenditures (OPEX) as well.

A satellite node needs to provide multi-granularity switching, such as packets, cells, frames, time slots, wavelengths, wavebands, and fibers. It offers the dynamic resource allocation directly on optical field or electronic field, and provides varieties of services flexibly. Moreover, hybrid terrestrial-satellite networks adopt wireless transmission in the satellite-ground user link and the satellite-ground feeder link, along with the coexistence of laser links and microwave links between a satellite and the ground station. To use network resources effectively and provide better QoS to users, we need to know the characteristic of wireless channels in real time. However, OpenFlow protocol was proposed specially for terrestrial Ethernet, therefore the existing OpenFlow protocol cannot be directly used to satellite networks. For providing the features mentioned for satellite networks, the OpenFlow protocol needs to be extended and enhanced. To support the multi-granularity switching of satellite nodes and characteristic acquisition of wireless transmission channels, OpenFlow protocol can be extended from two aspects of matching modules and adding new messages for the interaction of a controller and forwarding devices.

III. Extension of Matching Module

As the core of hybrid terrestrial-satellite networks, satellite nodes should not only support traditional IP switching, ATM switching and circuit switching, but also provide various switching forms of beam, intermediate frequency (IF), port, optical fiber, optical wavelengths, etc[14]. To implement the multi-granularity switching in SDN-enabled satellite nodes, we propose to extend the multiple flow tables matching of the framework extension in OpenFlow protocol.

A. *Multiple Tables Matching Framework*

Fig. 2 describes the matching framework of multi-granularity switching achieved by multiple flow tables. Each SDN switch uses the structure multiple flow tables, and decomposes the matching process into several steps, forming pipeline processing where each table implements switching of different granularities. The table ID increases by one each time from 0, and the switching granularity decreases as the pipeline processing continues.

The matching process is as follows: 1) when the user data arrives at the input port, it enters Table 0 for matching. If its key field matches the corresponding entry, we perform the relevant action specified by the flow table entry. 2) it will proceed to the next step, going to the next table to continue matching or exiting to end the pipeline. Contrarily, if there is no matching entry, each packet will be sent to the controller, waiting for the controller to issue the policy for further processing.

0 shows the specific structure of the *Payload Granularity* field in the multiple flow tables matching framework. It represents the current load types or the

switching granularity of a bearer or carrier. If the smaller switching granularity exists, then jump into the appropriate flow table to continue the pipeline processing; otherwise, stop the pipeline and forward to the output without preamble. The *Hierarchy* field gives the current switching level defined in TABLE I. Each remaining filed of *WB* (WaveBand), *WL* (WaveLength), *SDH* (Synchronous Transfer Hierarchy), *ATM*, *IP* is a flag, indicating whether there are other finer switching granularities on the current switching granularity. Note that the switching from a lager granularity to a finer one is *WB*, *WL*, *SDH*, *ATM* and *IP*, which indicates a switch may contain six switching stages at most. Except for the *WB*, each of the remaining four switching techniques may be embedded onto or transferred by the technique that locates at its front or left position. For example, an SDH frame can be transferred by a specific wavelength, and a Virtual Container (VC) or time slot switching can occur within an SDH frame. There are different switching granularities at VC level, ranging from VC4, VC3 and VC12 for Europe and China if needed. In addition, an ATM cell or IP packet could be extracted from a VC. Accordingly, ATM or IP switching will perform further.

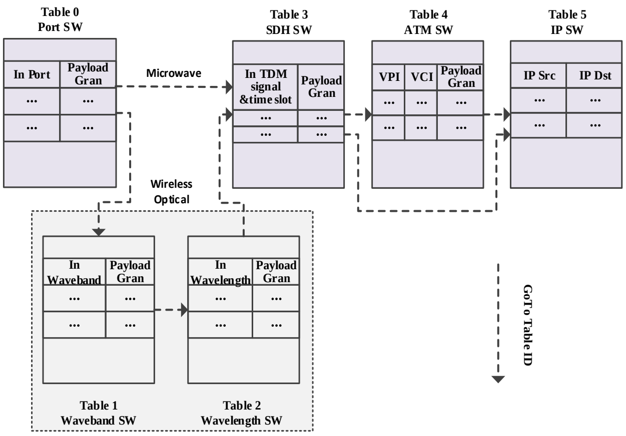


Fig. 2. The flow tables of multi-granularity switching on satellites

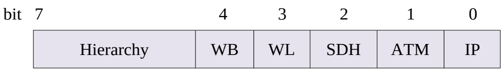


Fig. 3. Specific structure of *Payload Granularity* field

TABLE I. SWITCHING HIERARCHY

Code	Switching Granularity
000	Port switching
001	Waveband switching
010	Wavelength switching
011	SDH switching
100	ATM switching
others	Reserved for future use

B. *Single Flow Table*

The specific parameters of the single flow table are extended in accordance with the characteristics of each switching granularity in the multiple tables matching framework. From the aspects of the match field, counter and action, Fig. 4 shows the details of the each extended switching table, which gives the specific parameters of each switching table according to the property of its switching granularity. Match fields can perform the matching of each switching traffic, the counters are responsible for matching count, and instructions implement relevant actions. We can perform different switching granularities by using this table, ranging from the Port, Waveband, Wavelength, and TDM slot to the packet/cell level.

Taking ATM switching as an example, 0 illustrates the extended ATM switching flow table including three extended fields: match field, counter, and instruction. The match field includes virtual path identifier (*VPI*), virtual channel identifier (*VCI*), and *Payload Granularity*. We implement VP or VC switching by checking VPI or VPI+VCI, respectively. The counter counts the number of ATM cells matched, while the instruction is used to modify the set of actions or pipeline processing associated with ATM switching.

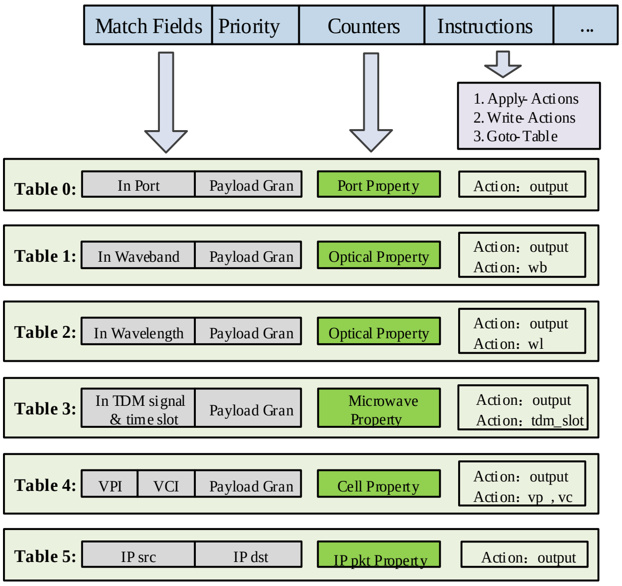


Fig. 4. Extension of the detailed each table

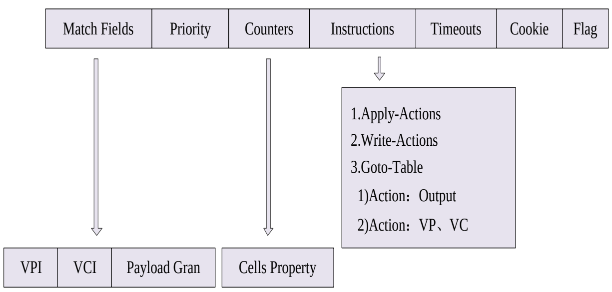


Fig. 5. Diagram of an ATM switching table

In the ATM switching table, the related actions are defined according to the ATM switching features. We perform VP switching and VC switching, so the ATM cells can be forwarded to the relevant output ports. Associated with different switching granularities of satellite nodes, different actions defined could make the packets to output in a corresponding manner. For example, the VPI or VPI+VCI of an incoming ATM cell are updated accordingly when such an ATM cell is output. 0(a) demonstrates the new action types and 0(b) utilizes OFPAT_VC as an example to explain its structure definition.

```
enum ofp_action_type {
    OFPAT_WB            = =30 ;
    OFPAT_WL            = =31 ;
    OFPAT_TDM_SLOT      = =32 ;
    OFPAT_VP            = =33 ;
    OFPAT_VC            = =34 ;
}
```

(a) New action types

```
Struct ofp_action_vc {
    uint16_t type;        /*OFPAT_VC. */
    uint16_t len;         /*Length is 16. */
    uint16_t vci;         /*Output VCI. */
    uint16_t vpi;         /*Output VPI, the high 4 bits are 0.*/
    uint32_t max_len;     /*Max length to send to controller. */
    uint8_t  pad[4];      /*Pad to 64 bits. */
};
OFP_ASSERT(sizeof(struct ofp_action_vc) = = 16);
```

(b) OFPAT_VC structure definition

Fig. 6. New action and its structure definition

C. *Flow Table Use Case*

The multiple tables matching framework is employed to the SDSN shown in Fig. 1. The user in the left IP subnet sends IP packets from the heterogeneous networks gateway to the GEO satellite nodes via microwave links. The packets are matched multiple flow tables within each satellite node successively, and then are forwarded to the corresponding output port until the destination of the correct IP subnet is reached ultimately. Given the case of IP over SDH, Fig. 7 presents the flow table diagram of the lower left satellite node.

```
Table=0, in_port=1, payload_gran=0x1f, actions=goto_table=3
Table=3, in_tdm_slot=1, payload_gran=0x7f, actions=tdm_slot:2, goto_table=5
Table=5, dl_dst=00:00:00:00:00:35, payload_gran=0xbf, actions=output:2
```

Fig. 7. Flow table diagram of an SDN-enabled satellite node

IV. EXTENSION ON MICROWAVE PORT FEATURES

In the hybrid terrestrial-satellite network, the laser link coexists with the microwave link of satellite-ground and inter-satellite. The microwave link with lower construction cost and higher transmission reliability has been widely used, while the laser link has the advantages of larger available bandwidth, less limitation to frequency allocation, and fine confidentiality. However, it can be relatively affected by weather conditions, such as cloud, fog, rain, and snow. Therefore, the two kinds of links compensate each other greatly. OpenFlow V1.4 has been revised to support for optical port features, thus appending microwave port features to it can be also valuable, which supports the configuration and monitoring of microwave transmission channel.

Considering the features of ports available in a datapath, the field *OFPPF_MICROWAVE* representing microwave link feature should be added to the structure *ofp_port_features*. For the specific port properties extension, there are three parts that contain the microwave port modification characteristic for port configuration, the statistical characteristic for port monitoring, and the description characteristic for port identification. The OpenFlow protocol provides extension domains for the above three features. With reference to the existing optical port properties, we design three extension fields for the structure *ofp_port_features*, namely,

1) ofp_port_mod_prop_experimenter
2) ofp_port_stats_prop_experimenter
3) ofp_port_desc_prop_experimenter.

```
struct ofp_port_desc_prop_ microwave{
    unit16_t   type;            /*  OFPPDPT_MICROWAVE.   */
    unit16_t   length ;         /*  Length in bytes of this property. */
    unit8_t    pad[4];          /*  Align to64 bits. */

    uint32_t   supported;       /*  Features supported by the port.   */
    uint32_t   min_tx_freq      /*  Minimum TX Frequency. */
    uint32_t   max_tx_freq;     /*  Maximum TX Frequency. */
    uint32_t   tx_grid_freq;    /*  Tx Grid Spacing Frequency.  */
    uint32_t   min_rx_freq;     /*  Minimum RX Frequency. */
    uint32_t   max_rx_freq;     /*  Maximum RX Frequency. */
    uint32_t   rx_grid_Freq;    /*  Rx Grid Spacing Frequency.  */
    uint16_t   min_tx_power;    /*  Minimum TX power.  */
    uint16_t   max_tx_power;    /*  Maximum TX power.  */
};
OFP_ ASSERT( sizeof( struct ofp_port _desc_prop_experimenter ) = =40 )
```

Fig. 8. Microwave port transmission features

0 gives the specific definition of the transmission characteristic parameters of a microwave port, mainly including the maximal-and minimal-frequency, grid spacing and power for a receiver and a transmitter. Note that the meaning or corresponding explanation of each variable is followed itself. Similarly, 0 presents the microwave port modification characteristic as an example to describe its definition.

```
enum ofp_port_mod_prop_type {
    OFPPMPT_ETHERNET          =0        /* Ethernet property. */
    OFPPMPT_OPTICAL           =1        /* Optical property. */
    OFPPMPT_MICROWAVE         =2        /* Microwave property. */
    OFPPMPT_EXPERIMENTER      =0xFFFF   /* Experimenter property. */
} ;
struct ofp_port_mod_prop_experimenter {
    uint16_t  type;           /* OFPPMT_MICROWAVE. */
    uint16_t  length;         /* Length in bytes of this property. */
    uint32_t  microwave_id;   /* Microwave ID which takes the same form
                                   as in struct ofp_microwave_header. */
    uint32_t  micro_type;     /* Microwave defined. */

    uint32_t  freq_band;      /* Waveband used, L, S, C, X, Ku, K, or Ka. */
    uint32_t  freq;           /*  Center  frequency. */
    uint32_t  freq_offset;    /* The signed frequency offset. */
    uint32_t  grid_span;      /* Grid size of this port. */
    uint32_t  tx_power;       /*Tx power setting. */
} ;
```

Fig. 9. Microwave port modification features

V. NEW MESSAGE EXTENSIONS

OpenFlow protocol was originally designed for Ethernet, and its controller-to-switch messages can merely collect the link transmission parameters of terrestrial wired networks. However, satellite networks use wireless transmission of laser and microwave links, it is thus necessary to obtain the parameters of a wireless transmission channel.

A. *New Message Definition*

In SDSN, there is the message interaction between a ground controller and SDN-enabled satellite nodes. For the delivery of satellite channel parameters, we propose to add two new messages to OpenFlow protocol shown in 0, which are the request and reply of satellite channel parameters. The second contains eight channel transmission parameters, so each satellite node could transport its real-time status to the ground controller by either answering to the inquiry request from the ground controller or by trigging events, such as in each constant period of time or when some parameters are beyond the thresholds configured in advance.

```
struct ofp_satparams_request{
    uint32_t experimenter;
    uint32_t sat_type;           /*Enum ofp_sat_type{
                                       OFPST_SAT_PARAMS_REQUEST;
                                       OFPST_SAT_PARAMS_REPLY
};*/
};
OFP_ASSERT(sizeof(struct ofp_satparams_request == 8));
struct ofp_satparams_reply {
    uint32_t experimenter;
    uint32_t sat_type;
    uint32_t bandwidth_total;  /*Total Transmission Bandwidth.*/
    uint32_t bandwidth_used;  /*Occupied Bandwidth.*/
    uint32_t EIRP;               /*Effective Isotropic Radiated Power(dBw).*/
    uint32_t Nf;                 /*Noise Figure(dB),typical value is 7dB.*/
    uint32_t Te;                 /*Equivalent NoiseTemperature (K),typical
                                       value is 1000K.*/
    uint32_t C_N;                /*Carrier-to-Noise Ratio.*/
    uint32_t Ws;                 /*Saturation Flux Density (dBW/m^2).*/
};
OFP_ASSERT(sizeof(struct ofp_satparams_reply == 36));
```

Fig. 10. New message of two types

B. *New Message Validations*

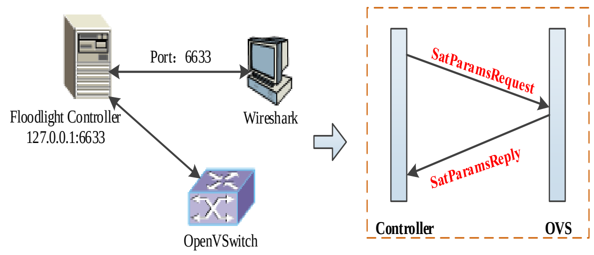
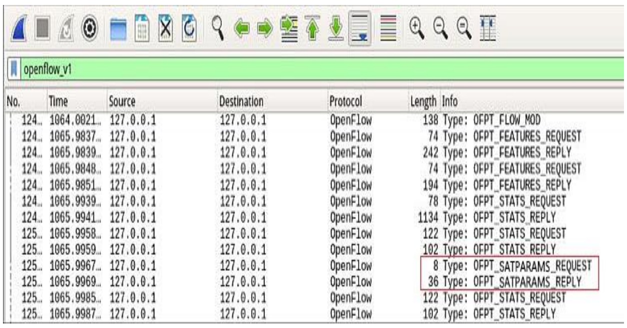


Fig. 11. Prototype system construction and expected results



No.	Time	Source	Destination	Protocol	Length	Info
124…	1064.0021…	127.0.0.1	127.0.0.1	OpenFlow	138	Type: OFPT_FLOW_MOD
124…	1065.9837…	127.0.0.1	127.0.0.1	OpenFlow	74	Type: OFPT_FEATURES_REQUEST
124…	1065.9839…	127.0.0.1	127.0.0.1	OpenFlow	242	Type: OFPT_FEATURES_REPLY
124…	1065.9848…	127.0.0.1	127.0.0.1	OpenFlow	74	Type: OFPT_FEATURES_REQUEST
124…	1065.9851…	127.0.0.1	127.0.0.1	OpenFlow	194	Type: OFPT_FEATURES_REPLY
124…	1065.9939…	127.0.0.1	127.0.0.1	OpenFlow	78	Type: OFPT_STATS_REQUEST
124…	1065.9941…	127.0.0.1	127.0.0.1	OpenFlow	1134	Type: OFPT_STATS_REPLY
125…	1065.9958…	127.0.0.1	127.0.0.1	OpenFlow	122	Type: OFPT_STATS_REQUEST
125…	1065.9959…	127.0.0.1	127.0.0.1	OpenFlow	102	Type: OFPT_STATS_REPLY
125…	1065.9967…	127.0.0.1	127.0.0.1	OpenFlow	8	Type: OFPT_SATPARAMS_REQUEST
125…	1065.9969…	127.0.0.1	127.0.0.1	OpenFlow	36	Type: OFPT_SATPARAMS_REPLY
125…	1065.9985…	127.0.0.1	127.0.0.1	OpenFlow	122	Type: OFPT_STATS_REQUEST
125…	1065.9987…	127.0.0.1	127.0.0.1	OpenFlow	102	Type: OFPT_STATS_REPLY

Fig. 12. New message communication validation using wireshark

For verification of the correctness and feasibility of the added message, we set up a small prototype system shown in 0, which contains an Openvswitch, a floodlight controller and a Wireshark terminal with Ubuntu 14.04 environment. We add the proposed message to source package and use Wireshark tool to capture the communication packet. The result shows the success delivery of request-reply pair messages between the controller and the SDN switch. 0 shows the message interaction between the controller and the virtual switch. When the controller sends *OFPT_SATPRAMAS_REQUEST* message requesting the transmission parameters of satellite wireless channel, we see that the switch returns *OFPT_SATPRAMAS_REPLY* message to report its own status, confirming the successful delivery of the new message.

VI. Conclusions

In this paper, multiple tables feature of OpenFlow protocol is extended to support the multi-granularity switching of satellite nodes, and the new message type is added to collect the transmission parameters of satellite wireless channel, which has the reference value for the SDSN realization. In addition, SDN open source software is used to build a test system that validates the new appending messages transfer between the ground controller and the satellite forwarding devices in SDSN.

Our next works will be: 1) an implementation scheme of an on-board switch for the SDN scenario; 2) check the added message, and add more parameters to the proposed structure, such as the spot beam and its carrier identifier; 3) how to trade off the implementation complexity and performance; 4) the architecture of an on-board switch suitable for the NFV scenario; 5) evaluating the performance of the space and ground hybrid network[15] with the SDN controlled and the number of users more than 10^4.

References

[1] S. Xu, X. W. Wang, and M. Huang, "Software-Defined next-generation satellite networks: Architecture, challenges, and solutions," *IEEE Access*, vol. 6, pp. 4027-4041, Feb. 2018.

[2] M. Jammal, T. Singh, A. Shami, *et al.*, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, no. 11, pp. 74-98, July 2014.

[3] X. N. Yang, J. L. Xu, and C. Y. Lou, "Software-Defined satellite: A new concept for space information system," in *Proc. Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, Hangzhou, China, 2012, pp. 586-589.

[4] Z. Tang, B. Zhao, W. R. Yu, *et al.*, "Software defined satellite networks: Benefits and challenges," in *Proc. Computing, Communications and IT Applications Conference (ComComAp)*, Beijing, China, 2014, pp. 127-132.

[5] J. Bao, B. Zhao, W. Yu, *et al.*, "OpenSAN: A software-defined satellite network architecture," in *Proc. ACM Conference on SIGCOMM*, Chicago, Illinois, USA, 2014, pp. 347-348.

[6] T. Li, H. Zhou, H. Luo, *et al.*, "Using SDN and NFV to Implement Satellite Communication Networks," in *Proc. International Conference on Networking and Network Applications (NaNA)*, Hakodate, Hokkaido, Japan, 2016, pp. 131-134.

[7] H. W. Li, Z. Y. Zhao, W. Q. Sun, *et al*., "SDN Plus: Architecture and extension for using SDN to control future wireless networks," *Designing Techniques of Posts and Telecommunications*, vol. 58, no. 10, pp. 44-50, Oct. 2015.
[8] M. Chen, G. Shou, Y. Hu, *et al*., "Enabling software-defined optical networks based on OpenFlow extension," in *Proc. 20th Opto-Electronics and Communications Conference (OECC)*, Shanghai, China, 2015, pp. 1-3.
[9] M. Bahnasy, K. Idoudi, and H. Elbiaze, "Software-defined DWDM optical networks: OpenFlow and GMPLS experimental study," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Austin, TX, USA, 2014, pp. 2173-2179.
[10] S. Yamashita, A. Yamada, K. Nakatsugawa, *et al*., "Extension of OpenFlow protocol to support optical transport network, and its implementation," in *Proc. IEEE Conference on Standards for Communications and Networking (CSCN)*, Tokyo, Japan, 2015, pp. 263-268.
[11] M. Yan, J. Casey, P. Shome, *et al*., "AetherFlow: Principled wireless support in SDN," in *Proc. IEEE 23rd International Conference on Network Protocols (ICNP)*, San Francisco, CA, USA, 2015, pp. 432-437.
[12] V. Nascimento, M. Moraes, and R. Gomes, *et al*., "Filling the gap between software defined networking and wireless mesh networks," in *Proc. 10th International Conference on Network and Service Management (CNSM) and Workshop*, Rio de Janeiro, Brazil, 2014, pp. 451-454.
[13] B. W. Chen, "method and apparatus based on the OpenFlow extension protocol used for software defined satellite networking," CN Patent 106060858A, Oct. 26, 2016.
[14] F. Ma, "Research on extension and prototype implementation of the signaling protocol in hybrid satellite and terrestrial networks," M.S. thesis, Telecom. Eng., Xidian Univ., Xi'an, China, 2016.
[15] C. Chen, S. Xie, X. Zhang, and Z. Ren, "A new space and terrestrial integrated network architecture aggregated SDN," *Journal of China Academy of Electronics and Information Technology*, vol. 10, no. 5, pp. 450-454, Oct. 2015.

Zhanqi Xu received his B.S., M.S. and Ph.D. degrees in communication and electronic system from Xidian University, China in 1984, 1987, and 1997, respectively. He is the senior member of both the China Institute of Communications (CIC) and China Computer Federation (CCF), and an IEEE member. Dr. Xu had a one-year postdoctoral study at Hong Kong University of Science and Technology during the turn of this century. Since 2000, He has been with the state key laboratory on Integrated Services Networks (ISN), Xidian University, China, where he is currently a Professor. His interested areas include optical networks, space information networks, and communication networks modeling and Internet of Things (IoTs).

Miaomiao Feng received the B.S. degree in Communication Engineering and the Master degree in Communication and Information System from Xidian University, China, in 2014 and 2017, respectively. Her research interests include satellite networks, software defined networks and network resource assignment.

Jianxin Lv is a distinguished technic expert of Fiberhome Communications Technology Company (FHCTC), Ltd. He has been engaged in optical fiber communication technology research and product development for more than 30 years. He has conducted the products development in optical transport equipment of FHCTC, such as Automatically Switched Optical Network (ASON) and Packet Enhanced Optical Transport Network (POTN). He has extensive experience in 40G/100G high-speed optical transmission technology research and equipment development.

Yang Fan holds a BS in Electronic Engineering in 1995, Master degree in Electronic Engineering in 1998 and Ph.D in Information and Communication Engineering in 2002, and all are from Xidian University, China. He is currently an associated professor with the state key laboratory on Integrated Services Networks (ISN), Xidian University, China. His study areas are optical network, green networks and high speed switching.

Guo Chen received his Master degree in Communications and Information Systems from Xidian University, China, in 2018. The focus of his studies was on Software Defined Networking and programmable dataplane prototype, advancing research on packet processing with FPGAs. He is currently a software engineer in Shenzhen City, China.