RAA 2021 Vol. **21** No. **11**, 279(10pp) doi: 10.1088/1674-4527/21/11/279

http://www.raa-journal.org *http://iopscience.iop.org/raa*

Research in Astronomy and Astrophysics

Simulation analysis of a method to improve data-transmission performance of Nanshan 26 m Radio Telescope based on Software-Defined Networks

Jie Wang[1,2,4], Hai-Long Zhang[1,3,4⋆], Na Wang[1,3⋆], Xin-Chen Ye[1,4], Wan-Qiong Wang[1], Jia Li[1], Meng Zhang[1,2] and Ya-Zhou Zhang[1,2]

[1] Xinjiang Astronomical Observatory, Chinese Academy of Sciences, Urumqi 830011,China; *zhanghailong@xao.ac.cn, na.wang@xao.ac.cn, wangjie@xao.ac.cn*

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[3] Key Laboratory of Radio Astronomy, Chinese Academy of Sciences, Nanjing 210008, China

[4] National Astronomical Data Center, Beijing 100101, China

Received 2021 March 1; accepted 2021 August 12

Abstract Data Center of Xinjiang Astronomical Observatory (XAO-DC) commenced operating in 2015, and provides services including archiving, releasing and retrieving precious astronomical data collected by the Nanshan 26 m Radio Telescope (NSRT) over the years, and realises the open sharing of astronomical observation data. The observation data from NSRT are transmitted to XAO-DC 100 km away through dedicated fiber for long-term storage. With the continuous increase of data, the static architecture of the current network cannot meet NSRT data-transmission requirements due to limited network bandwidth. To get high-speed data-transmission using the existing static network architecture, a method for reconstruction data-transmission network using Software-Defined Networks (SDN) is proposed. Benefit from the SDN's data and control plane separation, and open programmable, combined with the Mininet simulation platform for experiments, the TCP throughput (of single thread) was improved by ∼24.7%, the TCP throughput (of multi threads) was improved by ∼9.8%, ∼40.9%, ∼35.5% and ∼11.7%. Compared with the current network architecture, the Latency was reduced by ∼63.2%.

Key words: data transmission — observation data — data center — virtual machine

1 INTRODUCTION

With the rapid development of the information field, technologies such as big data, artificial intelligence, cloud computing, and data mining have been gradually applied to astronomy and have achieved many results. Information technology has provided a reliable foundation for the analysis and processing of astronomical data. Presently, with the telescope array, and the wideband or multi-beam receiver, the astronomical observations produce massive, high-precision, high-quality, and highly reliable data at an unprecedented speed, indicating that astronomy has entered a data-intensive era (Cui et al. 2015). The massive astronomical observation data make research in astronomy a data-centred and data-driven scientific activity. Taking a single antenna radio telescope (Zhang et al. 2017) as an example, the annual data rate of the Green Bank Telescope (GBT) (O'Neil et al. 2004) is about 1.4 PB, since the Versatile Green Bank Spectrometer (VEGAS) terminal was mounted (MacMahon et al. 2018). Effelsberg generates approximately ∼25 TB (Winkel et al. 2010) of archived data annually. Parkes has generated more than 100 TB of archived data (Hobbs et al. 2011) since its establishment. For the Five-hundred-meter Aperture Spherical radio Telescope (FAST), the 19 beam receiver generates ∼10 PB pulsar data with the original data generation rate of 38 GB/s (Yue & Li 2019). Responding to the data-intensive astronomical-research era, China is actively promoting the development of big data research in astronomy. In 2019, the National Astronomical Data Center[1] (Cui et al. 2020) (NADC), a logically unified

⋆ Corresponding author

[1] `https://nadc.china-vo.org/`

service platform, became the first national scientific data center covering the data generated by major astronomical observation equipment in China. NADC has more than 10 PB of existing astronomical data[2]. With such huge astronomical data resources, the tension of IPv4 network link resources is becoming increasingly prominent. Problems such as insufficient network protocol address space and limited network bandwidth (Li et al. 2016) results in insufficient data-transmission because the IPv4 network architecture cannot transmit large-scale astronomical data and transmission of observation data at a constant rate is difficult. Large-scale astronomical data-transmission occupies large bandwidths, affecting data-transmission efficiency and is not time efficient during transmission. Thus, fast and efficient transmission of astronomical data generated by observations is a difficult problem to be solved in astronomical research. In this paper, we introduce a method to improve the NSRT data-transmission performance using SDN, and intend to provide new ideas for solving the problem of astronomical data-transmission.

2 SDN

2.1 SDN Architecture

SDN originated from the Clean Slate research project[3] of Stanford University in 2006, aiming to 'reshape the Internet'. Martin Casado formally proposed the concept of SDN (Casado et al. 2006) in 2006. SDN decouples the control plane from the data plane, separating the control and forwarding functions of data packets during transmission (McKeown et al. 2008). The control plane controls multiple devices and logically attains centralised control and programmability. It can quickly, simply and dynamically adjust traffic within the controllable range of the entire network to meet changing needs. The basic architecture of SDN is shown in Figure 1.

The SDN architecture consists of five elements: application plane, control plane, data plane, SDN northbound interfaces (NBIs) and SDN control-data-plane interface (CDPI) (Kreutz et al. 2014). The application plane embodies the SDN programmability and communicates with the controller through the NBIs. It decomposes the application functions into executable fine-grained and sends them to the SDN controller, which implements the application functions. The control plane is the core of the SDN architecture that provides a centralised view of the entire network (Astuto et al. 2014) while processing network traffic, forwarding data packets and controlling traffic according to specified fine-grained instructions. It communicates with the data plane through the CDPI. The data plane consists of various switches, including physical and virtual switches. SDN realises the decoupling of the control and data planes, respectively.

2.2 OpenFlow Protocol

The OpenFlow protocol[4] (McKeown et al. 2008) is a CDPI protocol for communication between the SDN control and data planes, respectively. It consists of both a flow and a group table for performing packet query and forwarding functions and an OpenFlow channel to an external controller. The flow table integrates both the FIB and MAC tables of the IPv4 architecture and is composed of match fields, priority, counters, instructions, timeouts and cookie. Each flow table entry defines forwarding rules including key fields such as port number, VLAN, L2 / L3, etc. As can be seen from Figure 2, the flow table matching process of the OpenFlow protocol-supported switches after receiving match fields is extracted from the data packet, and Table 0 to Table n are matched with the flow table items successively until matching (Hu et al. 2014). For unsuccessfully matched packets, the table-miss flow entry specifies either sending the packet to the controller or discarding it directly.

2.3 Necessity and Advantage

The transmission of NSRT observation data requires large bandwidths, and local traffic burst leads to long-tailed distributions of statistical characteristics. Presently, the NSRT data-transmission network maintains the traditional architecture, limited in bandwidths and difficult to modify network functions, resulting in low data-transmission efficiency and link congestion. Moreover, all switches in the traditional network architecture are heterogeneous, so the network cannot be controlled globally. SDN separates the control plane from the data plane, software-programmable network architecture, controls the traditional distributed network equipment and migrates to the controllable devices, making the network infrastructure can be abstract the network services and applications, finally through a mode of open and programmable software to realise the automatic control function of the network. SDN improves the scalability and flexibility of the NSRT data-transmission network.

[2] https://nadc.china-vo.org/html/aboutus.html
[3] http://cleanslate.stanford.edu/
[4] http://www.opennetworking.org/

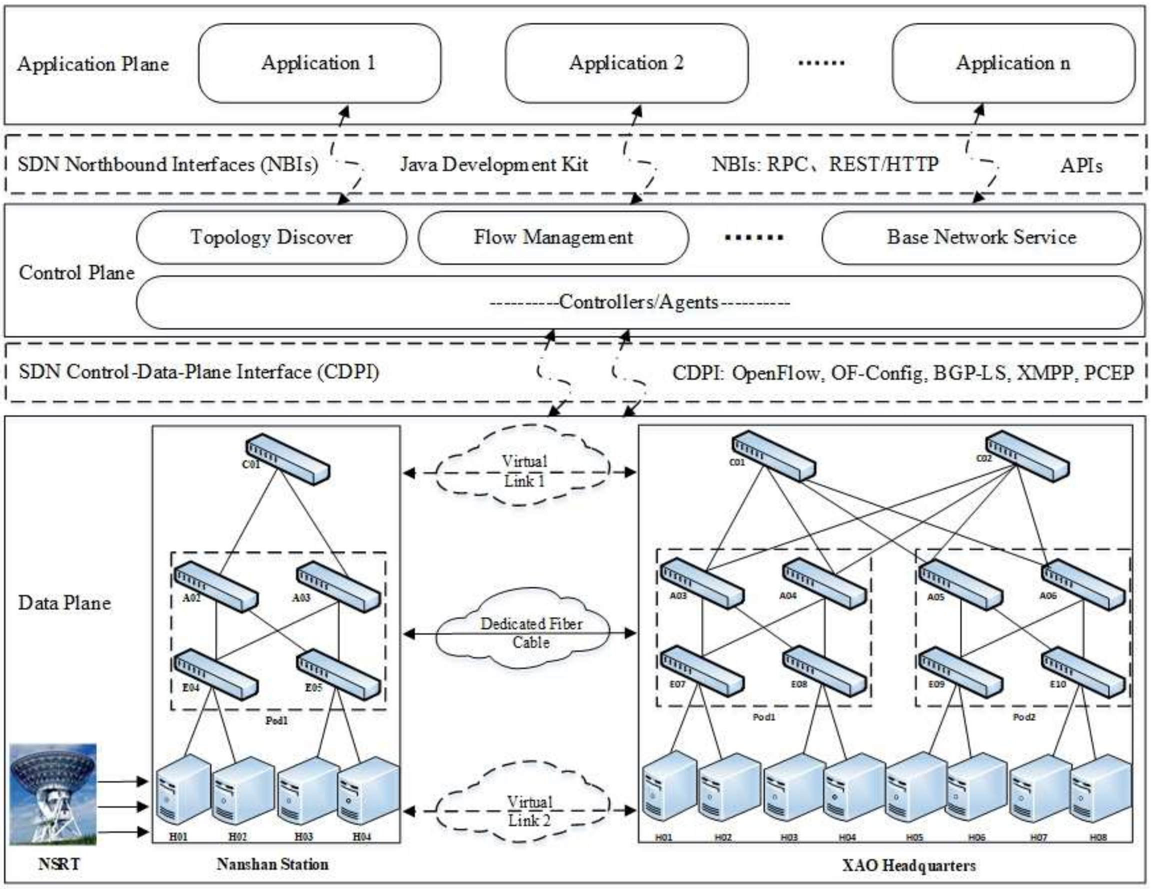

Fig. 1 The basic architecture of SDN.

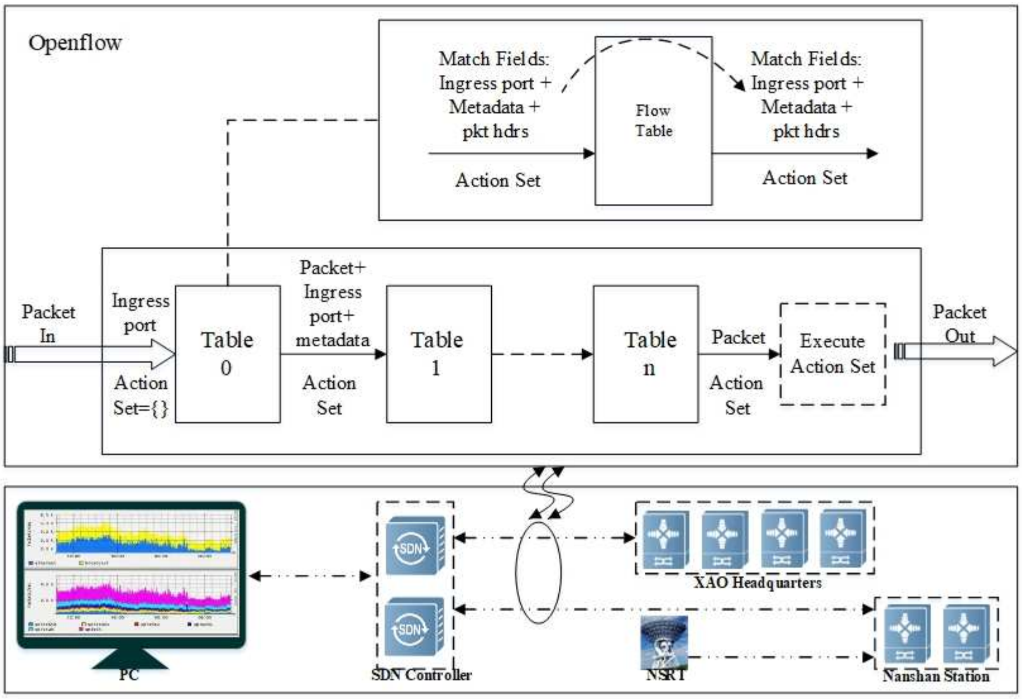

Fig. 2 Processing flow of packet matching multi flow tables.

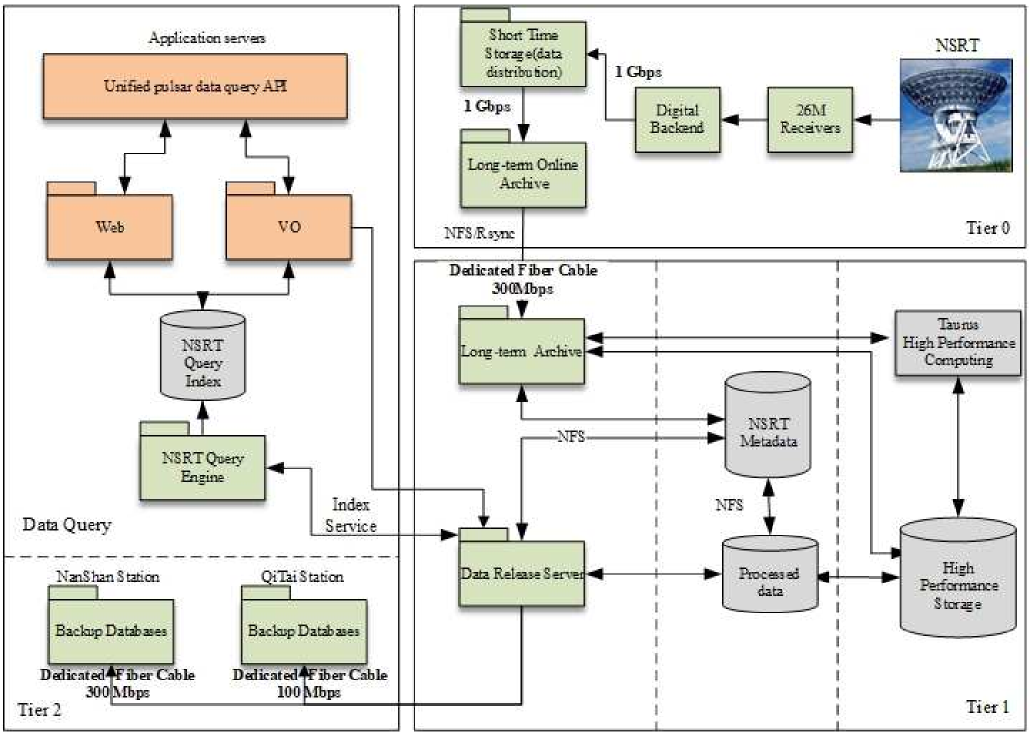

Fig. 3 NSRT data archiving and releasing system.

Table 1 Configuration of Simulation Experiment

Parameter	Value	Parameter	Value
Bandwidth(Nanshan Station to XAO)	300Mbps	Bandwidth(Others)	1Gbps
Controller monitoring period	1s	Topology 1	Openflow switches:10 SDN controllers:1
Maximum buffer queues of switch	1000	Topology 2	Openflow switches:2 SDN controllers:1
Iperf-Execution Flow Size	100Mbps-1Gbps	Topology 3	Openflow switches:10 SDN controllers:2
Ping-Execution Flow Size	1Kbps-100Kbps	Topology 4	Openflow switches:2 SDN controllers:2

3 NSRT DATA-TRANSMISSION NETWORK RECONSTRUCTION DESIGN

3.1 Status of the Data-transmission Network

From Figure 3, radio signals collected by the data acquisition system, preprocessed and stored in real-time (Zhang et al. 2019). Then, all the data will be backed up remotely to XAO headquarters through a 300 Mbps dedicated fibre. Presently, RSYNC is adopted to achieve data synchronisation. However, the bandwidth of astronomical and other data cannot be separated, and RSYNC only records one type of log information, hindering status judgement during mass astronomical data-transmission. Additionally, some disadvantages of single-path transmission are limited network resources and uneven resource utilisation. Once the single path is congested or other conditions occur, the transmission performance declines, significantly reducing the transmission rate of the astronomical data. When single-path failure occurs, several packets will be lost, which makes the demands of astronomical data-transmission difficult to meet. In this regard, the collective control and dynamic programmability of SDN can be employed to adjust network resources to meet the demands for astronomical data-transmission.

3.2 SDN Controller Deployment

Figure 4(a) shows that the networks of both XAO headquarters and Nanshan station are taken as logical switches, and a single SDN controller is deployed to maintain a logical topological view of the entire network, including the control network devices at the data plane as one functional unit. This network deployment method

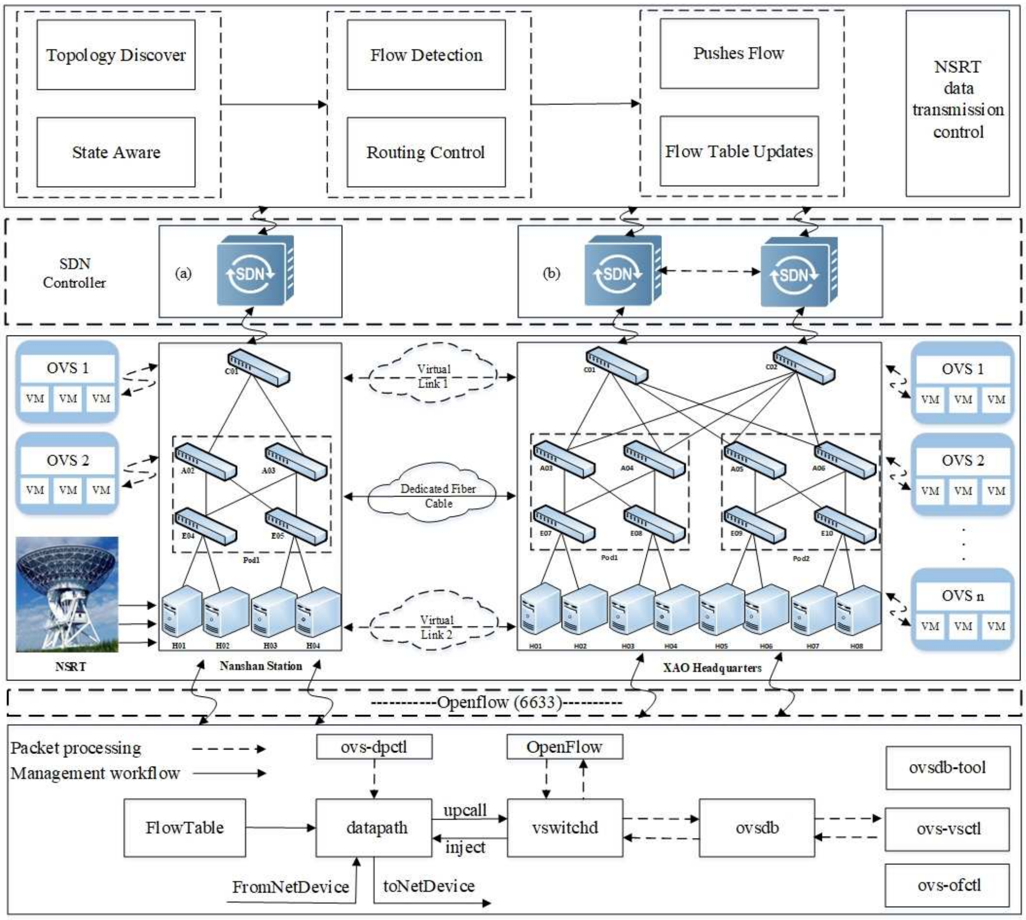

Fig. 4 SDN controller deployment.

automatically or customarily optimises the network equipment, and quickly adjusts the flow dynamics of the transmitted data. This network deployment method is independent of any special software. As shown in Figure 4(b), when multiple SDN controllers are deployed, they treat the networks of the XAO headquarters and the Nanshan station as two independent logical switches. The SDN controller of both the XAO headquarters and the Nanshan station, respectively, are deployed to maintain their respective logical topology views without considering the underlying logical topology.

The module design is as follows:

(1) Topology discovery module: The SDN controller sends packets to the NSRT transmission network through the link layer discovery protocol (LLDP), and obtain the entire network topology.

(2) State aware module: Sends state request packets to switch by periodically transferring the port- and flow-state requests, then obtains the link state information and data flow statistics information of the switches.

(3) Flow detection module: From the data flow information collected by the state aware module, the flow through the switches is periodically polled for N times. The NSRT data-transmission traffic is detected and allocated to the path with more link space, then stored in the memory.

(4) Routing control module: Controls flow based on the information from the flow detection module. If a traffic situation occurs during the NSRT data-transmission, the shortest path algorithm is adopted to route the link state information obtained using the state aware module.

(5) The routing information calculated using the routing control module is sent to the switches through the pushes flow module, which updates the flow table in switches. When transmitting on the NSRT through switches, the packets are matched with the new flow table before completing the data-transmission process.

3.3 Data Forwarding Mechanism

From Figure 5, the OpenDayLight was used to develop and control the entire process. Server 1 (10.0.0.1) sends a request to Server 5 (10.0.0.5), and the OpenFlow switch-Nanshan station (SW1) sends the request to the controller through the packet-in message. The controller receives the

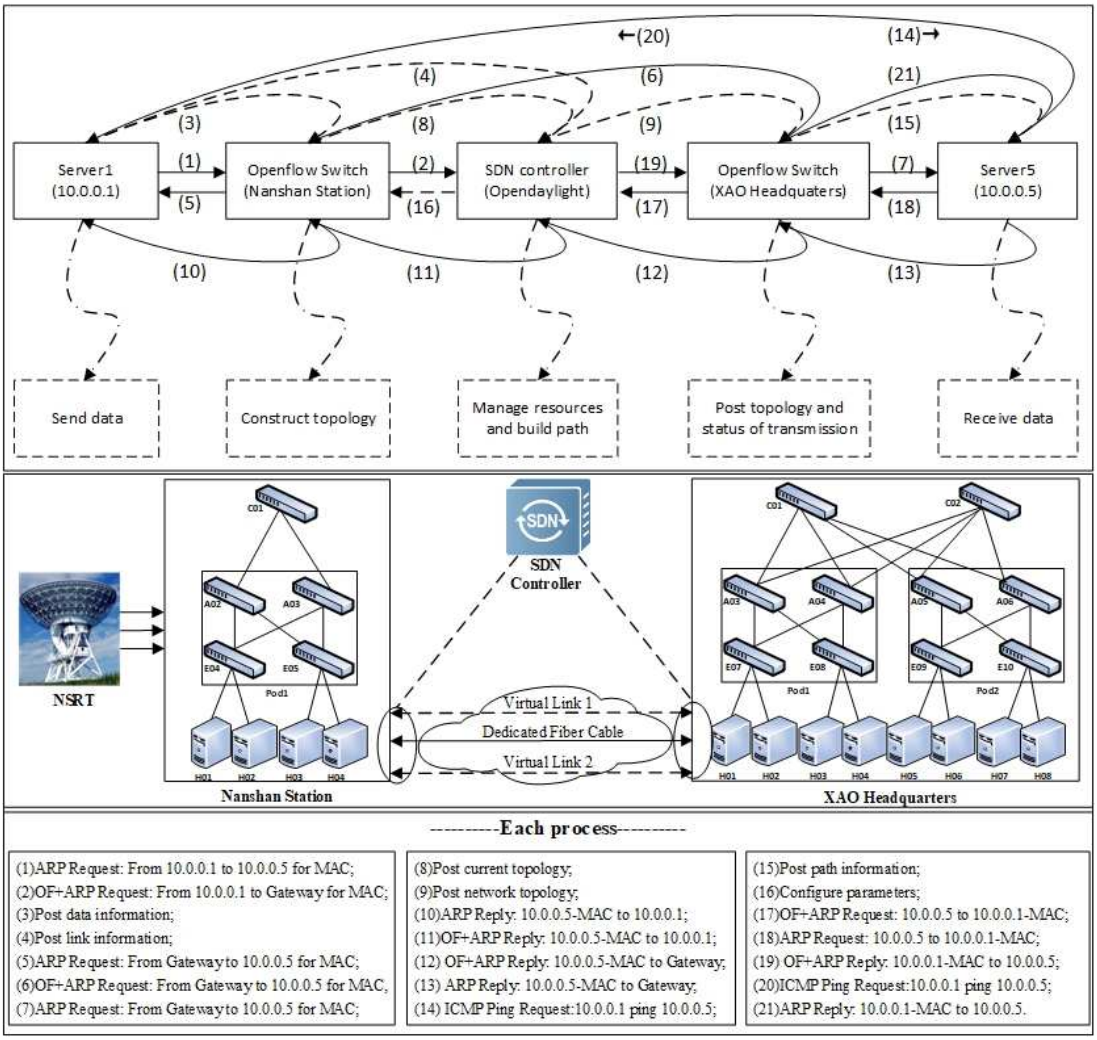

Fig. 5 Data forwarding mechanism.

message, changes the source IP to the default gateway IP address and issues it. Server 5 responds to the request and sends the response to Openflow switch-XAO headquarters (SW2). After encapsulation, it sends the response to the controller through the packet-in message. The controller sends to SW1 through the Openflow protocol and gets the location information of Server 5 after receiving the response. From the network topology calculation, we obtain the data forwarding path to Server 5 and Server 1. After Server 1 receives the response, it completes the parsing process, constructs the ICMP PING request packet and sends the packet to Server 5. SW2 then sends the packet-in message to the controller upon receiving the ICMP PING request from Server 1. The controller gets the response and returns the result, and forwards the port to SW2 via the OpenFlow protocol. Server 5 sends the ICMP Response package to SW2 and SW2 forwards to SW1 according to the match-forwarding table. Then, SW1 sends to the corresponding port of Server 1 according to the match-forwarding tabl.

4 NSRT DATA-TRANSMISSION NETWORK SIMULATION

4.1 Simulation Environment and Tools

Two computers with Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz, 64 GB memory and Intel(R) Core(TM) i5-3470 CPU @ 3.20GHZ, 8 GB memory was used in the experiment, one of which simulated the network of the XAO headquarters and the other simulated the network of the Nanshan station. Both computers build a virtual machine; the operating system was Ubuntu 18.04.1 (Linux version 5.3.0-46-generic). The controller used the OpenDayLight as the core of the SDN architecture (version: distribution-karaf-0.6.1-Carbon) with high availability, modularity, scalability and multi-protocol support. The SDN data plane

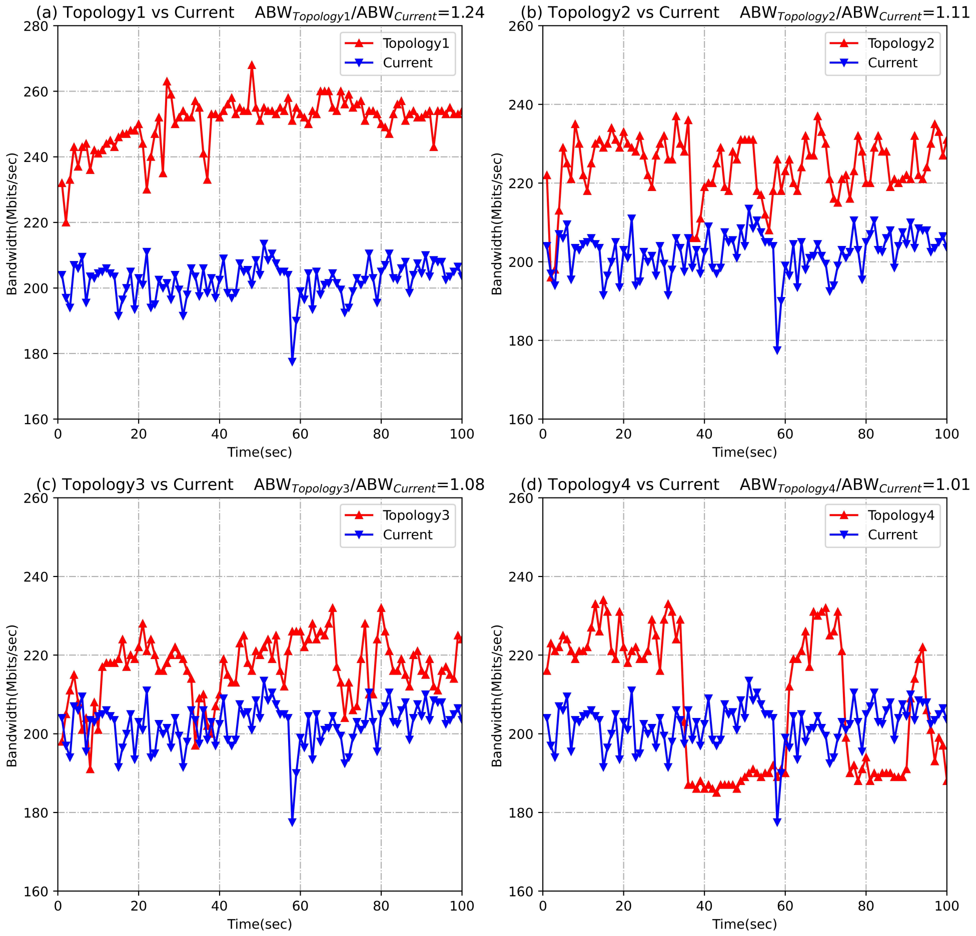

Fig. 6 TCP throughput test of single thread.

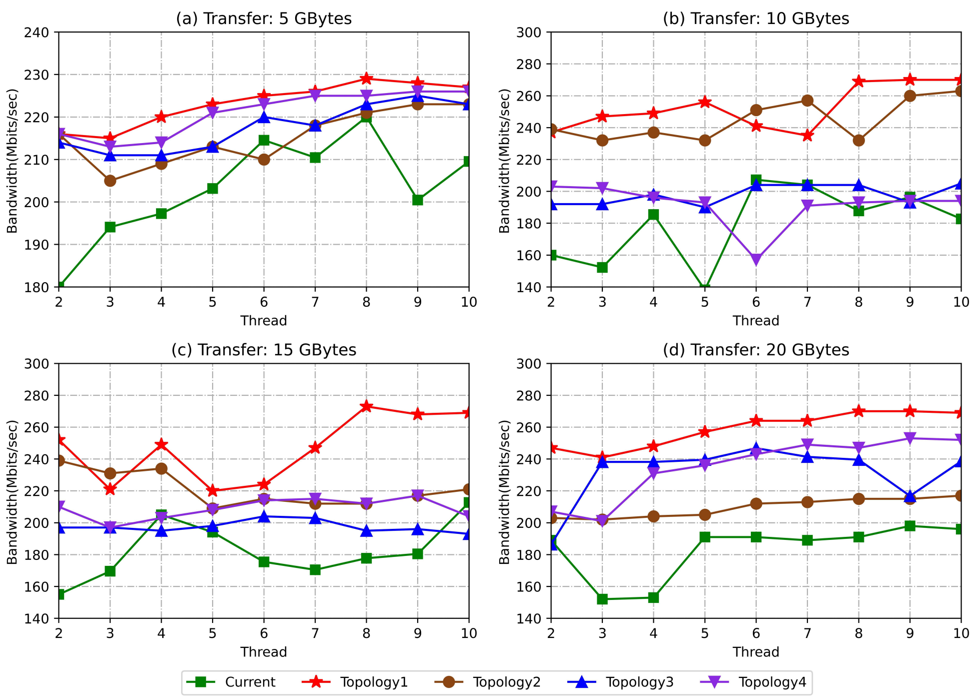

Fig. 7 TCP throughput test of Multi threads.

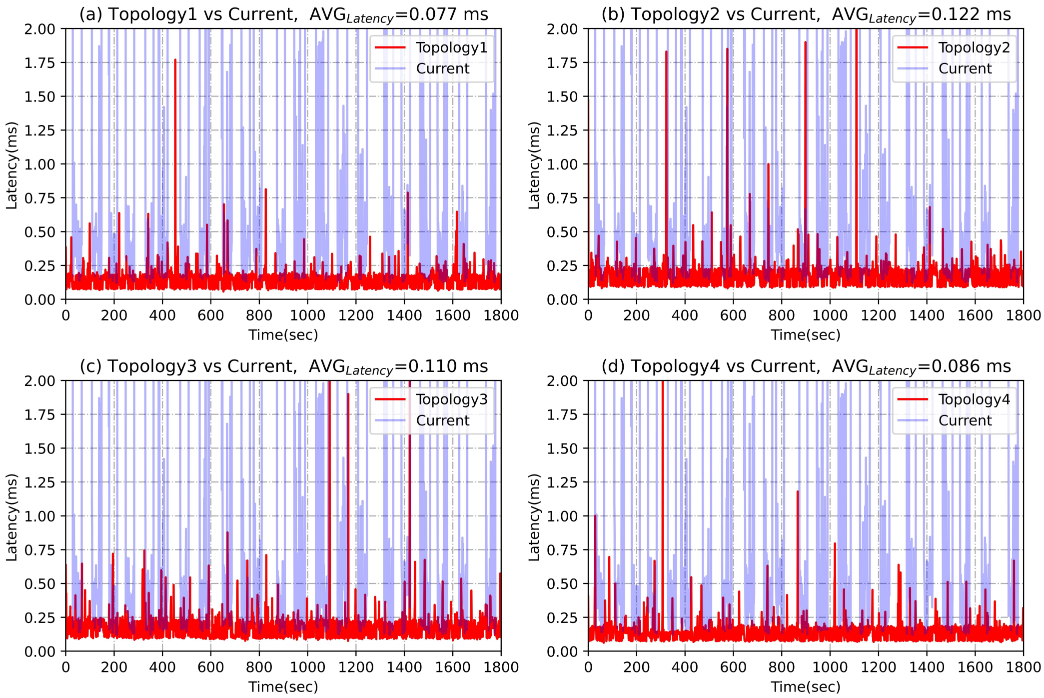

Fig. 8 Latency test.

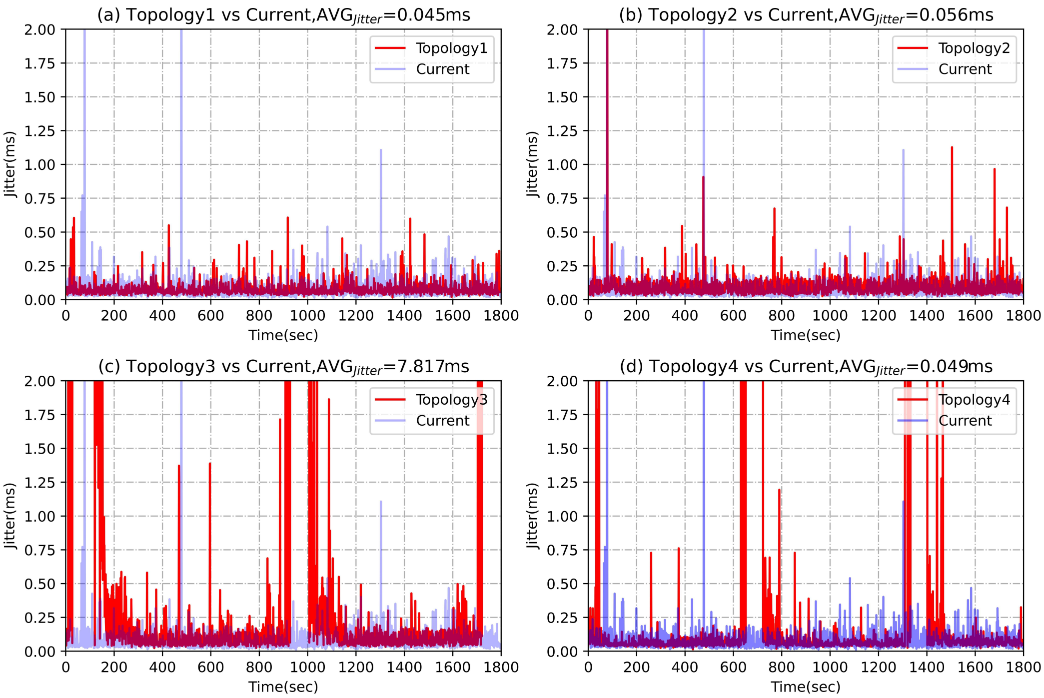

Fig. 9 Jitter test.

used Mininet 2.2.2, Open vSwitch 2.5.5 and Openflow 1.3 for simulation. Meanwhile, we tested the current network architecture between both computers, labelled current and compared it with Topology 1–4.

4.2 Simulation Implementation

Using a simulation environment built on Mininet, the link bandwidth from the Nanshan station to the XAO headquarters is set to 300 Mb/s, and the remaining link bandwidth is set to 1 Gb/s. The large flow required in the experiment is generated using python script, executed by the iperf function, and the size ranges from 100 Mbps–1 Gbps; the small flow is generated using a python script executed by the ping function, and the size ranges from 1–100 Kbps. The SDN controller OpenDayLight monitoring period is 1 s. The configuration of the simulation experiment is shown in Table 1.

4.3 Test Results

4.3.1 TCP throughput

TCP throughput is an important indicator for evaluating TCP behaviour and performance. It represents the maximum Nanshan station-to-XAO packet transmission rate. The formula for calculating the TCP throughput is as follows:

$$\mathrm{Throughput} = \left(\sum_{i=1}^{M} \mathrm{Throughput}_i \right) / M, \qquad (1)$$

where $\mathrm{Throughput}_i$ is the throughput of the stream *i*, and *M* is the amount of effective data in the NSRT transmission network. The performance of the four topologies (1–4) was tested in groups, with the test transmission data volume was set to 5120 MB. We execute python script developed by Iperf to test the single-thread TCP throughput. From Figure 6, the Server 1 of the Nanshan station sends packets to the Server 5 of the XAO headquarters. We compare the TCP throughput of the first 100 seconds. Results show that the TCP throughput of topology 1 is higher than that of the other three topologies.The ratios of $\mathrm{ABW}_{\mathrm{Topology}(1-4)}$ to $\mathrm{ABW}_{\mathrm{Current}}$ (ABW: Average Bandwidth) are 1.24, 1.11, 1.08 and 1.01, respectively.

When testing TCP throughput for the multi-thread, the transmission volume is 5, 10, 15 and 20 GB, respectively, and the number of threads lies between 2 and 10. Set the bandwidth from the Nanshan station to XAO be 300 Mbps, and the parameters of the other bandwidths remain 1 Gbps. We execute python script to test the multi-thread and compare it with the current transport. From Figure 7, topology 3 has a lower elapsed time than the other topologies.

4.3.2 Latency

Latency is an important indicator for evaluating the performance of the transmission network, indicating the delay of the packets from the NSRT-server to XAO-DC. The low delay indicates high forwarding rate of the transmission network. The formula for calculating the latency is as follows:

$$\mathrm{Latency} = \mathrm{Average} \left(T_{\mathrm{end}} - T_{\mathrm{start}} \right), \qquad (2)$$

where T_{start} and T_{end} denote the time when the packet is sent from the NSRT-server and the time when the packet reaches the XAO-DC, respectively. The test results of the latency measures are shown in Figure 8. Selecting Server 1 of the Nanshan station and Server 5 of the XAO headquarters to run the test 1800 seconds, the results show that the latency of topology 1 is lower than the other three topologies, and topology 1 has a more stable test result. The $\mathrm{Average}_{\mathrm{Latency}}$ of topologies (1–4) are 0.077, 0.122, 0.110 and 0.086 ms.

4.3.3 Jitter

Jitter is also an important indicator for evaluating the stability of data-transmission. It represents the change in delay during the data-transmission of packets and reflects the magnitude and frequency of the delay fluctuations in data packet forwarding. The lower jitter value indicating that the data-transmission is stable. The formula for calculating jitter is as follows:

$$\mathrm{Jitter} = \sum_{i=1}^{n} \left(\mathrm{Latency}_i - \mathrm{Average}_{\mathrm{Latency}} \right) / N, \qquad (3)$$

where $\mathrm{Latency}_i$, $\mathrm{Average}_{\mathrm{Latency}}$, and *N* represent the latency of the packet *i*, the average delay of the packet *n* in the forwarding process, and latency of the packet *n*, respectively. Figure 9 is the result of the jitter test of the UDP packets. We execute python script to send the UDP packets from Server 1 of the Nanshan station to Server 5 of the XAO headquarters. The $\mathrm{Average}_{\mathrm{Jitter}}$ of topology (1–4) is 0.045, 0.056, 7.817 and 0.049 ms, respectively.

5 CONCLUSIONS

The application of SDN technology to NSRT data transmission at XAO can improve network performance and effectively solve the problems of fast transmission, dynamic adjustment and bandwidth limitation of astronomical data under the current network architecture, as verified by simulation experimental tests. The observation data of NSRT in the simulation test can be quickly and stably transmitted to XAO-DC through the SDN reconfigured network. Subsequently, corresponding functions will be developed in the SDN controller to adjust and optimize the network scheme according to different data transmission situations.

Acknowledgements This work is supported by the National Natural Science Foundation of China (NSFC, Grant Nos. 11803080, 11873082 and 12003062), the National Key Research and Development Program of China (2018YFA0404704), the Youth Innovation Promotion Association, Chinese Academy of Sciences (CAS), the program of the Light in China's Western Region (2019-XBQNXZ-B-018). Data resources are supported by China National Astronomical Data Center (NADC). This work is supported by Astronomical Big Data Joint Research Center, co-founded by National Astronomical Observatories, CAS. The algorithms in this paper have applied Taurus High Performance Computing Cluster of Xinjiang Astronomical Observatory, CAS during the testing process.

References

Astuto, B. N., Ca, M. M., Xuan, N. N., Obraczka, K., & Turletti, T. 2014, IEEE Communications Surveys & Tutorials, 16

Casado, M., Garfinkel, T., Akella, A., et al. 2006, in USENIX Security Symposium, 49, 50

Cui, C., Yu, C., Xiao, J., et al. 2015, Chinese Science Bulletin, 60, 445

Cui, C., Tao, Y., Li, C., et al. 2020, Astronomy and Computing, 32, 100392

Hobbs, G., Miller, D., Manchester, R. N., et al. 2011, PASA, 28, 202

Hu, F., Hao, Q., & Bao, K. 2014, Communications Surveys & Tutorials IEEE, 16, 2181

Kreutz, D., Ramos, F. M., Verissimo, P. E., et al. 2014, Proceedings of the IEEE, 103, 14

Li, C., Cui, C., He, B., et al. 2016, Journal of Computer Applications, 036, 25 (in Chinese)

MacMahon, D. H. E., Price, D. C., Lebofsky, M., et al. 2018, PASP, 130, 044502

McKeown, N., Anderson, T., Balakrishnan, H., et al. 2008, ACM SIGCOMM computer communication review, 38, 69

O'Neil, K., Radziwill, N., & Maddalena, R. 2004, in Astronomical Society of the Pacific Conference Series, 314, Astronomical Data Analysis Software and Systems (ADASS) XIII, eds. F. Ochsenbein, M. G. Allen, & D. Egret, 70

Winkel, B., Kalberla, P. M. W., Kerp, J., & Flöer, L. 2010, ApJS, 188, 488

Yue, Y., & Li, D. 2019, Big Scientific Data Management. BigSDM 2018. Lecture Notes in Computer Science, 11473, 6

Zhang, H., Ye, X., Zhao, Q., et al. 2017, SCIENTIA SINICA Physica, Mechanica & Astronomica, 47, 059505, (in Chinese)

Zhang, H., Demleitner, M., Wang, J., et al. 2019, Advances in Astronomy, 2019, 1