

Research Article

Chen Gang*, Chen Guangyu, Tong Xin, Ren Qiaoyan, and Kuang Dongmei

Application of SDN-IP hybrid network multicast architecture in Commercial Aerospace Data Center

<https://doi.org/10.1515/astro-2022-0223>

received March 04, 2023; accepted April 26, 2023

Abstract: The increasing amount of massive data generated by commercial spacecraft in orbit puts forward higher and higher requirements for the stability, reliability, and computing power of computer systems for commercial aerospace data center. Data center computer systems are gradually transforming from X86 architecture and IP network model to a platform model with cloud computing and software-defined network (SDN) technology. This article proposes a new network architecture based on a unicast/multicast protocol for data interaction between the SDN and IP network. There are three main contributions of this article. The first is that the architecture proposed in this article aims to reduce end-to-end transmission latency and packet loss. The second is to improve the flexibility of system configuration and precise control when the SND controller state changes. The third is to verify the feasibility of deploying network architecture in a real data center environment.

Keywords: SDN, hybrid network, multicast

1 Introduction

The computer system of a commercial aerospace data center is the information hub with the most intensive data resources and the most frequent data exchange. It undertakes the tasks of system connection, data transmission and storage, and information processing and analysis. With the development of commercial space missions, the data center based on cloud computing technology will gradually develop into an intelligent center with a complete system and a safe, reliable, flexible, controllable, and efficient operation. Cloud technology (Babbar *et al.* 2021) is a way to platformize, productize, and resource IT technology and resources and deliver them to users in the form of services. The cloud computing platform has the advantages of cost intensification, security intensification, and management intensification, which will greatly improve the resource utilization, service bearing, data mining, and innovative application capabilities of the data center computer system, as well as the security and reliability of data. On the basis of building foundation resource pool based on mainstream cloud technology and using virtualization technology, the commercial aerospace data center is planned and built from multiple aspects, such as computing virtualization, data storage virtualization, network virtualization, and virtual security management, based on actual business needs. It is an independent information processing system that has the characteristics of loose coupling of network resources and virtualization of IT equipment, and ultimately meets the systematic requirements (Mohamed *et al.* 2021).

Compared with IP networks, software-defined network (SDN) has the advantages of improving the intelligent edge forwarding capability of the control layer, efficient bearing capacity of the backbone network, openness, and collaboration of the network, which are often applied in cloud data centers (Sharma 2021, Feamster *et al.* 2014). SDN can change the tightly coupled architecture in IP networks and enhance the level of network resource pooling.

* **Corresponding author: Chen Gang**, State Key Laboratory of Astronautic Dynamics, Xi'an, Shaanxi, China, e-mail: chengangxxwl@sina.com

Chen Guangyu: State Key Laboratory of Astronautic Dynamics, Xi'an, Shaanxi, China, e-mail: guangyuchen11@163.com

Tong Xin: State Key Laboratory of Astronautic Dynamics, Xi'an, Shaanxi, China, e-mail: tongxink@qq.com

Ren Qiaoyan: State Key Laboratory of Astronautic Dynamics, Xi'an, Shaanxi, China, e-mail: 479446466@qq.com

Kuang Dongmei: State Key Laboratory of Astronautic Dynamics, Xi'an, Shaanxi, China

By introducing SDN and virtualizing resources based on the internal physical network of cloud data centers, cloud data center network capacity can be synthesized into a unified network capacity pool, which can not only alleviate the pressure of scalability and flexibility faced by cloud data centers when carrying multiuser services but also improve network intensification capabilities and finally realize automated deployment configuration of cloud data center resources, rapid service rollout, and flexible expansion of intelligent networking capabilities (Alssaheli *et al.* 2021).

SDN transforms the network from hard to soft, enhances the adaptability and support of the network to virtualization, cloud computing, and other technologies, as well as the centralized control of the network, and finally improves the service support capability of the network for the service (Chaokun *et al.* 2014). SDN is used to reshape the network architecture inside the aerospace measurement and control data center and gradually form the aerospace measurement and control communication network architecture with the central cloud as the core, the main communication nodes as the hubs, and the remote measurement and control stations as the edges (Wibowo *et al.* 2017). The central SDN can be divided into network control layer, network forwarding layer, and computing access layer. The network control layer consists of access controller (AC) controllers and an FabricInsight (FI) intelligent analyzer, which connects upward with the cloud platform, providing service-oriented interfaces, and controls network devices downward, abstracting the requirements of service deployment and network operation and maintenance into network services. The network forwarding layer is composed of network devices such as switches and routers, including three regions: cloud network (center), pipeline network (main communication nodes), and terminal network (remote measurement and control stations). The network forwarding layer is not only the data center network for server and storage access but also the network node for wide-area network and terminal access.

At present, there are still some elements to be improved in SDN technology in terms of network device chip capability, standard use of OpenFlow protocol, and security protection of controllers, so SDN will not completely replace IP network, but is also certainly a strong complement to the existing network. As a necessary auxiliary function of network equipment, SDN has appeared; such network equipment can not only carry forwarding control through SDN but also manage and maintain equipment through the IP network and can also use traditional manual methods to deploy services to network equipment. Although SDN is seen as the “next big thing” in networking technology, it will not replace the existing network equipment in commercial aerospace data center. However, it can be used a way to deploy network

equipment based on inherent needs. For example, “SDN + High Performance Networking” can make commercial aerospace data center cloud platforms more resilient.

According to the pattern of three regions: cloud network (center), pipeline network (main communication nodes), and terminal network, this article introduces SDN in central cloud platform and still uses IP network technology in the main communication nodes and terminal network, which inevitably results in the coexistence of SDNs and IP networks (Stallings 2015). Multicast technology achieves efficient point-to-multipoint data transmission in IP network, which can effectively save bandwidth, control network traffic, reduce data source load, and reduce network load. So multicast technology is widely used in the process of commercial aerospace data interaction (Quinn and Almeroth 2001, Oliveira and Pardalos 2005, Caiyun *et al.* 2014, Lan and Xuezhi 2000, AlSaeed *et al.* 2018, Huang *et al.* 2016, Kotachi *et al.* 2019). On the one hand, SDN controller can access the real-time network topology. On the other hand, users can replace multicast routing algorithms rapidly and flexibly according to their own requirements. This article studies the connectivity and stability of data interaction between SDN and IP networks (involving complex scenarios such as SDN and IP networks, IP networks across SDN to IP networks, and so on) and solves the difficulties of data interaction between the two types of networks based on multicast protocols. The research in the article lays the technical foundation for the application of SDN in commercial aerospace data center and provides strong practical guidance.

2 Concept and technical mechanism of SDN technology

2.1 Concept of SDN technology

SDN is a new network architecture proposed by Martin and Nick McKeown, who are part of the Clean-Slate research group at Stanford University. SDN is an implementation of network virtualization (Stallings 2015). SDN is divided into data plane, control plane, and application plane from bottom to top (or from south to north). The core technology of SDN’s architecture is to separate the control plane from the data plane and mask the differences between the underlying physical forwarding devices through standards so as to realize resource virtualization. At the same time, flexible and open upward interfaces can be used by the application layer to configure the network on demand and use network resources. SDN defines and controls network with software-

programmable openness at both control and data levels, which can make network smarter as a pipeline.

The horizontal and vertical orientations in the computer system are standard and open; there are hardware, drivers, operating systems, programming platforms, and application software from bottom to top in the computer system; and programmers can easily create all kinds of applications. Compared with computer systems, the meta-data interaction between networks in IP network is only standard and open in the horizontal direction and relatively closed in the vertical direction, making it relatively difficult to deploy services and create applications. SDN technology provides standardization and programmability in the vertical direction of the network, thereby reducing equipment load and O&M costs and improving network resource utilization. SDN is a revolution in network field, which provides a new experimental path for the research of new network architectures and greatly promotes the development of internet technology.

2.2 Intellectual property core

The data plane is an SDN built according to different rules among various network elements such as switches. The

control plane, with controller as logical center, holds global network information and is responsible for controlling various forwarding rules, running logical policies, and maintaining network-wide view. The data plane communicates with the control plane using the SDN control data plane interface (CDPI, also known as the Southbound Interface). CDPI has a unified communication standard, mainly using the OpenFlow protocol to distribute forwarding rules in the controller to forwarding devices. The application plane includes various SDN-based network applications, and users do not need to care about the technical details of the underlying equipment, and can quickly deploy new applications through programming alone. The communication between the control plane and the application plane is handled by the SDN north-bound interface (NBI), which allows users to customize development according to actual needs. The east-west interface is responsible for communication between multiple controllers. All of these interfaces are open.

The OpenFlow protocol used in the CDPI interface in SDN is characterized by matching forwarding rules based on flow. Each switch maintains a flow table and forwards according to the forwarding rules in the flow table, and the establishment, maintenance, and distribution of the flow table are all done by the controller. Applications invoke various network resources required through NBI to achieve

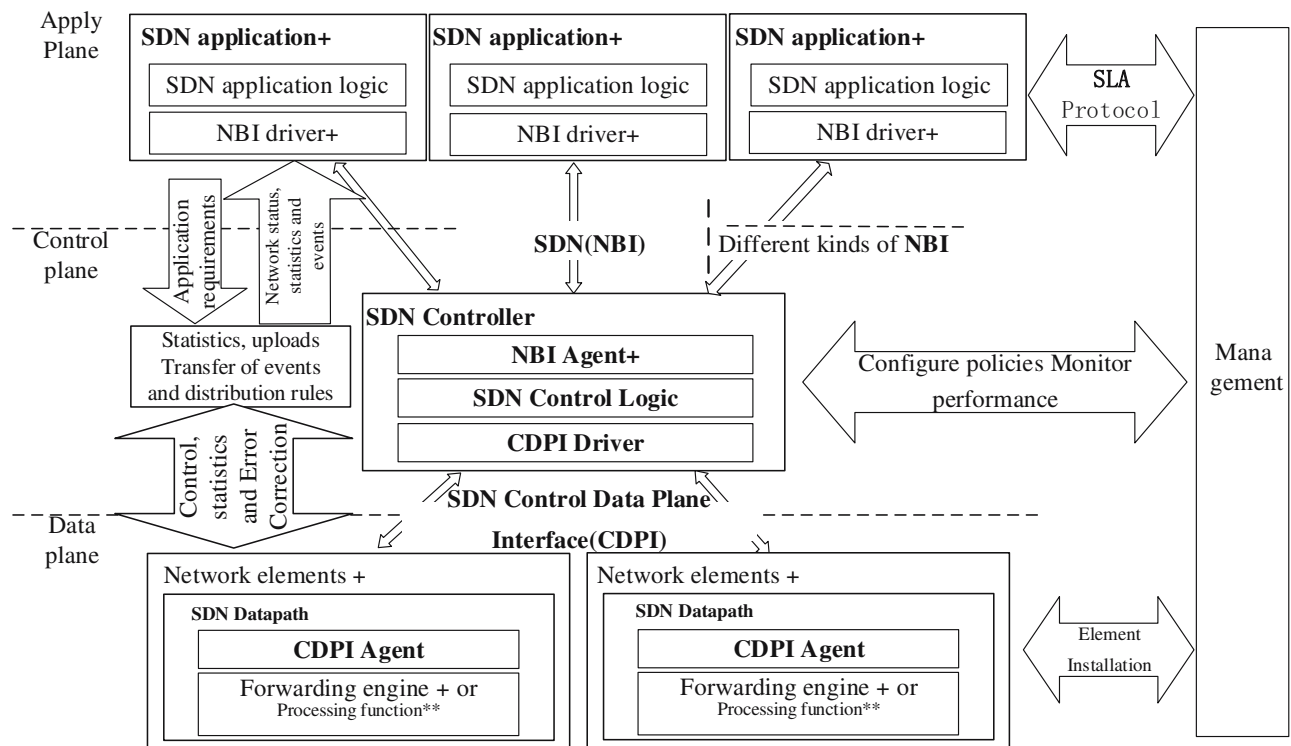


Figure 1: SDN implementation mechanism.

rapid configuration and deployment of the network. The east–west interface enables the controller to provide technical assurance for load balancing and performance improvement. The mechanism of SDN is shown in Figure 1.

As shown in Figure 1, the controller abstracts the entire network into network service and provides users with easy-to-use NBIs by accessing CDPI interfaces and calling corresponding network channel resources, which not only facilitates customized privatization applications but also realizes logical management of the network. CDPI sends forwarding rules from the network operating system to network devices without affecting the control layer and above logic. NBI allows third party to develop network management software and applications, providing managers with more options. The abstract nature of network allows users to select different network operating systems according to their needs without affecting the normal operation of physical devices.

3 Research on SDN and IP network convergence technology

With the continuous expansion of network scale, there are many complex protocols in closed networks, which increase the difficulty of network planning and customization optimization. At the same time, in IP networks with complex services, if business requirements change after service deployment, it is very cumbersome to modify the network configuration of the corresponding devices, which cannot meet the requirements of flexible and agile service deployment.

The inherent defects of IP technology make it difficult to meet the above requirements. SDN technology also has certain defects in chip support capability, standard use of OpenFlow protocol, and security protection of the controller. Therefore, the coexistence of the two types of networks and the reliable transmission of data has become the focus of industry research. At present, the mainstream research on reliable data transmission between two types of networks includes RouteFlow, IMISA (interconnection mechanism for IP subnet and SDN subnet in autonomous system) architecture, and controller border gateway protocol (BGP) components.

3.1 RouteFlow

RouteFlow (Balaji *et al.* 2022), proposed by CPqD, a Brazilian telecommunication research and development center,

is one of the first ways to interconnect SDN with IP networks. RouteFlow mainly includes three parts. The first part is RouteFlow, an application running on the SDN controller NOX, written and implemented in C++, and responsible for the interaction between NOX controller and RouteFlow Server. The second part is RouteFlow Server, which is responsible for the virtual network to interact with the NOX controller and maintains a database for storing network information. The third part is virtual network environment, in which RouteFlow clones the SDN data plane in the virtual network environment, and each OpenVswitch (Yong *et al.* 2017) switch in each SDN corresponds to a virtual machine (VM) in the virtual network, with the same number of interfaces and connections. In virtual network, the VM runs the open-source routing engine Quagga (Nascimento *et al.* 2011, Linux Foundation 2022), which emulates IP network router functionality.

When the OpenVswitch switch in SDN receives a routing protocol message from the IP network, the message is sent to NOX controller, which sends the message to virtual network according to the pre-established port mapping relationship, and the routing engine Quagga processes it according to the IP network routing method. The processing result is forwarded by the NOX controller to the OpenVswitch switch and then sent to the IP network. By converting the interaction between IP network and SDN into interaction between IP network and virtual network in RouteFlow, the interconnection between IP network and SDN is indirectly realized. The architecture of RouteFlow is shown in Figure 2.

In the RouteFlow implementation, each OpenVswitch switch must generate a Quagga VM in the RouteFlow environment. When SDN is large, the overhead of building VM with RouteFlow will be significant, which limits the scope of RouteFlow's applications. Moreover, RouteFlow only uses OpenVswitch switches to achieve distributed protocol

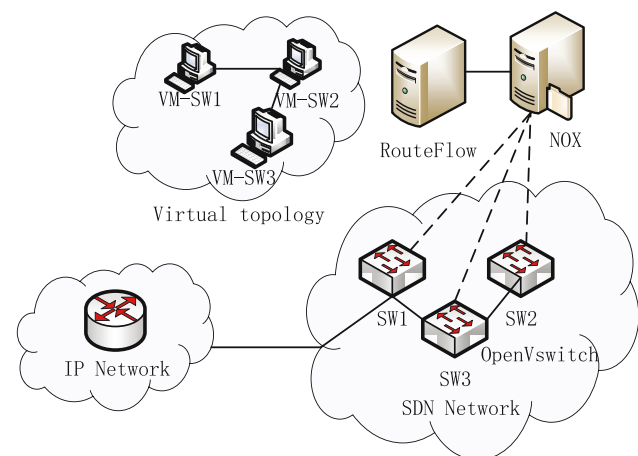


Figure 2: Topology diagram of RouteFlow mode (Balaji *et al.* 2022).

calculation and data forwarding functions, and the NOX controller is only used to relay protocol messages between traditional routers and virtual routers and does not achieve centralized control of controller to SDN.

RouteFlow implements unicast protocol-based data interaction between two types of networks but does not have multicast protocol-based data exchange capabilities.

3.2 IMISA architecture

The IMISA architecture was proposed by Yanwei and Zheng (2014). It mainly composed of three parts. The first part is the converter, which is deployed at the SDN area boundary and can interact with traditional routers at layer 3 based on the open shortest path first (OSPF) protocol, as well as with SDN controller as an OpenVswitch. The second part is the OSPF routing application, which runs on the SDN controller POX and has OSPF packet parsing and construction functions.

In an autonomous system (AS) range that includes SDN and IP network, the POX controller parses and constructs OSPF protocol message, and the SDN boundary converter interacts with the border router of the IP network between the two networks, so that the IP network in the AS can sense the SDN and the SDN can sense the IP network, which enables the interconnection of SDN and IP networks within AS. The topology of the IMISA approach is shown in Figure 3. The IMISA approach controls the border switch through the POX controller, which simulates a router to interact with the IP network based on the OSPF protocol. The IMISA approach enables SDNs to interconnect with IP networks based on OSPF protocol when they are used as edge networks but does not realize the ability to relay traditional IP networks using SDNs.

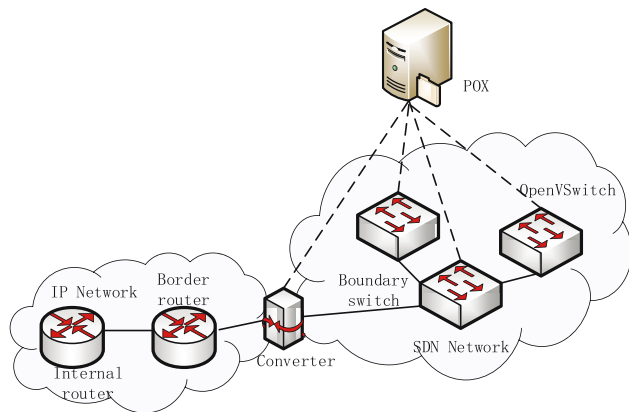


Figure 3: Topology of IMISA (Yanwei and Zheng 2014).

The IMISA approach achieves unicast protocol-based data interaction between two types of networks but does not have multicast protocol-based data interaction capabilities.

3.3 Controller BGP component

Controller BGP component approach was proposed by Lin *et al.* (2013) of Tsinghua University. This method is characterized by running an SDN-IP interconnect application module on top of the SDN controller, which mainly includes two parts. The first part is BGP routing module, which is responsible for processing BGP messages and storing routing information into the routing information database. The second part is Flow table entry pre-installation module, which calculates and installs the flow table based on the routing information learned from BGP. The controller BGP component approach topology architecture is shown in Figure 4. The SDN-IP interconnect application module is added to the SDN controller, which has complex functions and a large workload.

The controller BGP component approach implements the unicast protocol-based data interaction between the two types of networks, but does not have the data exchange capability based on multicast protocols.

In summary, although the three technical approaches have certain technical characteristics and advantages, none of them has the ability to interact with data based on multicast protocols between the two types of networks. Therefore, they cannot be directly applied to the commercial aerospace communication system designed in this article in the mode of cloud network (cloud platform and SDN), pipe network (IP network), and terminal network (IP network).

To solve the above problems, this article designs a SDN-IP hybrid networking multicast architecture to realize the data interaction function based on unicast/multicast protocols. The method in this article mainly includes three

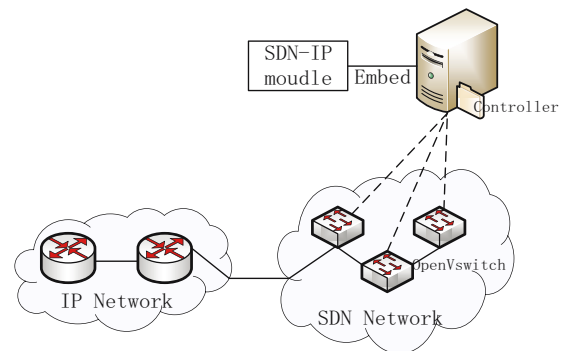


Figure 4: Controller BGP component solution architecture (Lin *et al.* 2013).

parts. The first part is to solve the problem that adding BGP routing modules to the SDN controller tends to increase the load on the controller. The core idea is to use Quagga (running BGP applications) as the Routing Engine to replace the BGP Routing Synchronization module of the SDN controller, and synchronize the BGP update messages processed by the Routing Engine to the SDN controller through a predefined format, so as to achieve the goal of reducing the load on the controller. The second part is to address data interactions based on unicast protocols. The SDN-IP unicast management module is designed on the SDN controller to process routing information, maintain route convergence between AS, and realize data interaction between SDN and IP network or other SDNs based on unicast protocol. The third part is to solve data interaction based on multicast protocol. The SDN-IP multicast management module is designed on the SDN, which is responsible for handling multicast member management and multicast route management, performing intra-SDN routing calculation, generating intentions, and delivering flow tables to SDN switches so as to realize the data interaction between the two networks based on multicast protocols.

4 Overall architecture design

The overall design of the SDN-IP hybrid networking multicast architecture is shown in Figure 5. The architecture of this article is based on the SDN controller open network operating system (ONOS) Controller (Berde *et al.* 2014). In order to implement multicast data interaction between IP network and SDN, we designed two main parts: the SDN unicast management module and the SDN multicast management module.

According to the BGP protocol specification, the network is divided into two AS. The first area is the autonomous domain AS1, composed of IP network devices. The second area is the SDN autonomous domain AS2, consisting of ONOS directors, border switches, and routing engines.

In AS1, routers interact according to the BGP protocol specification, and send routing information of AS1 to BGP Speaker in the domain. In AS2, the ONOS Controller discovers the module through the topology to obtain SDN connection information. The BGP speaker in AS1 and the routing engine in AS2 transmit routing information by establishing an external BGP (EBGP) connection, and the routing engine and ONOS Controller pass routing information by establishing internal BGP (IBGP) connection. In AS1, the BGP Speaker sends routing information to routing engine through BGP update message, then the routing engine updates routing information table after receiving routing information and sends the updated routing information to ONOS Controller through BGP update message, which is processed by SDN unicast routing management module. Submodule of protocol independent multicast (PIM) module (PIM multicast routing management module) and internet group management protocol (IGMP) module (IGMP group member management module) in SDN multicast management module process the corresponding protocol message, and the submodule of multicast routing management (MCAST) module maintains multicast routing table. The above three submodules jointly maintain group member joining, and multicast tree grafting and pruning. A submodule of the multicast routing forwarding (MFWD) module maintains intention and sends flow table to border switch. All of the above processes finally realize multicast protocol-based data interaction between SDN and IP network.

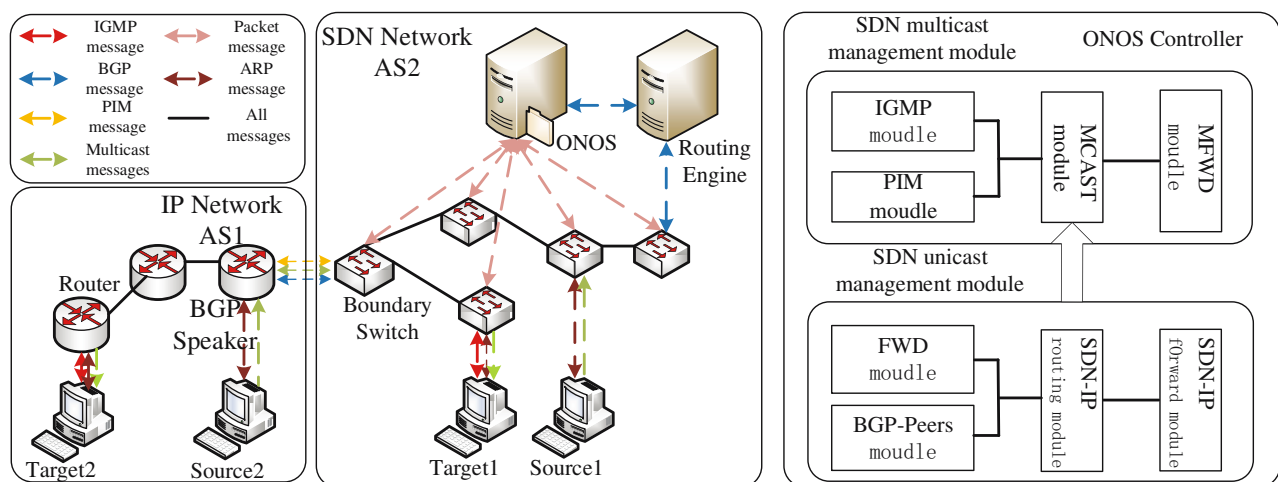


Figure 5: Controller BGP component solution architecture.

4.1 Composition and function of SDN unicast management module

The SDN unicast management module mainly includes hop-by-hop forwarding (FWD) module, BGP-Peers connection management module, SDN-IP unicast routing module, and SDN-IP unicast forwarding module.

- 1) FWD module. It maintains EBGP connection from BGP speaker in IP network to routing engine in SDN, and forwards BGP message with destination address of SDN hop-by-hop to routing engine. Most of the EBGP connections in IP networks are based on static configuration files, and static configuration files of SDN need to change with the topology of SDN, which is difficult to implement. In this article, FWD module is used to establish EBGP connection between any two physically reachable nodes in SDN by hop-by-hop forwarding and generating intention.
- 2) BGP-Peers connection management module. It maintains IBGP connection between routing engine and ONOS Controller in SDN, and listens for IBGP messages between ONOS Controller and routing engine.
- 3) SDN-IP unicast routing module. It listens to BGP update messages in BGP-Peers connection management module and parses information about routes in the messages to maintain the unicast routing table. After receiving the BGP update message, it extracts network layer reachability information (NLRI) and NEXT_HOP information from Path Attributes. NLRI information is the address of the reachable network, which is composed of IP address and network mask. NEXT_HOP information is the IP address of the next-hop router that reaches NLRI.
- 4) SDN-IP forwarding module. It generates intention based on the unicast routing table and compiles and delivers flow table. It listens to unicast routing module, and when there is a new unicast routing table change, it generates Multi-Point-to-Single-Point intentions according to the changes and saves them in the intent table. When unicast data arrive at the border switch, it is forwarded to SDN-IP unicast forwarding module through Packet_in message, then calls intention service to query the matching intention, compiles the flow table based on the intent, and delivers it to the border switch.

4.2 Composition and function of SDN multicast management module

The SDN multicast management module mainly includes IGMP module, PIM module, MCAST module, and MFWD module.

- 1) IGMP module. (a) Parses IGMP packets of the SDN and manages the addition of group members within the SDN. (b)

Extract multicast session information (multicast source [S] and multicast group address [G]) in IGMP protocol packets received by the ONOS controller. (c) Record the SDN switch port, which receives IGMP protocol message as the multicast data O-port (Out-port). (d) Combine the above information into a triplet <Switch/O-Port,S,G>. (e) Store <Switch/O-Port,S,G> in MCAST multicast routing management module.

If the multicast source information is parsed to be empty, the host sends the session away from the multicast, finds the S and MCAST multicast routing tables through G, and deletes the corresponding triplet <Switch/Port,S,G> from the multicast routing table. The specific implementation process is shown in Algorithm 1.

Algorithm 1: IGMP

for all received IGMP packet **do**

```

if < S, G > ≠ ∅
| MCAST ← < Switch/Port,S,G > ;
else
| delete < Switch/Port,S,G > from MCAST;
end
end

```

- 2) PIM module. (a) Parse PIM-Join/Prune message of SDM and tell the upstream/downstream routers of SDN that a host joins/leaves a multicast session <S, G> to complete the multicast tree expansion/pruning of the mixed network. (b) Extract the multicast session information (multicast source [S], multicast group address [G]) of the protocol packet when ONOS Controller receives a PIM-Join message from IP network, record the SDN switch port (Switch/Port) that receives the protocol packet as the multicast data export port, and combine the above information into a triplet <Switch/Port, S, G>, and store it to MCAST module. (c) Query MCAST multicast routing table when ONOS Controller receives PIM-Prune message and delete the corresponding triplet <Switch/Port,S,G> from the multicast routing table. The specific implementation process is shown in Algorithm 2.

Algorithm 2: PIM

for all received PIM packet **do**

```

if PIM Join then
| MCAST ← < Switch/Port,G-J,S-J >;
else
| delete < Switch/Port,G-P,S-P > from MCAST;
end
end

```

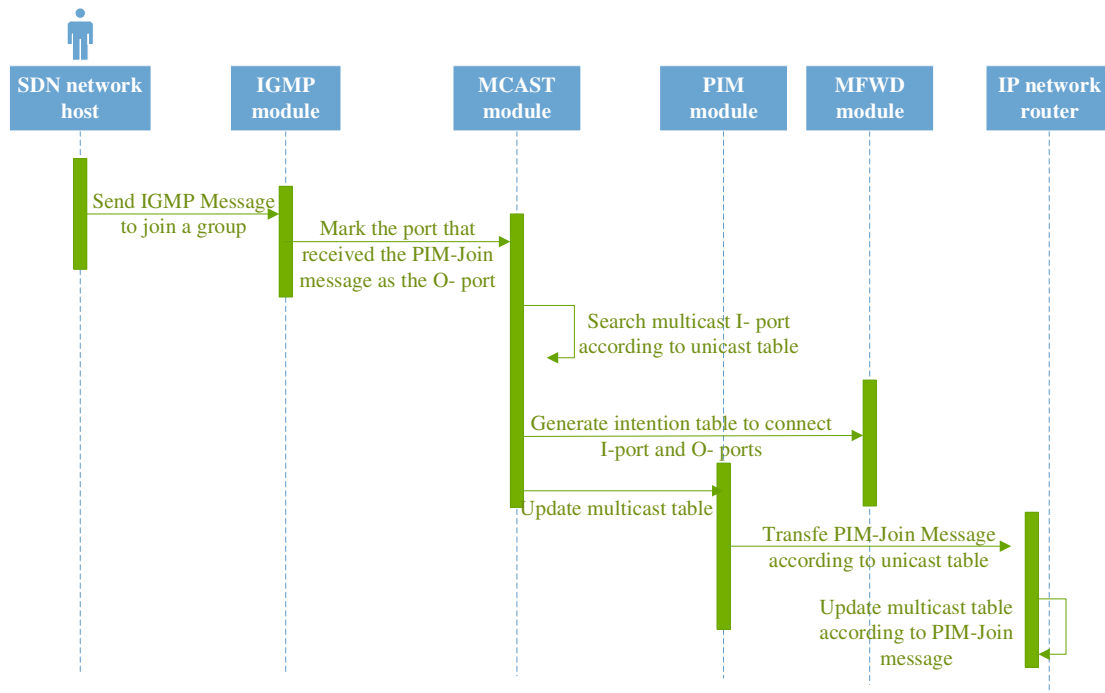


Figure 6: Flow chart of SDN host joining IP network multicast group.

- 3) MCAST module. (a) Maintain the multicast routing table. (b) Be responsible for the storage of the multicast routing table in SDN. (c) Provide adding, deleting, querying, and modifying of multicast table and multicast source's entry and output ports.
- 4) MFWD module. (a) Generate an intent based on the multicast routing table. (b) Compile the forwarding flow table. (c) Listen to the change information of the MCAST multicast routing table. (d) Generate an intent based on the changed content and saves it. (e) Find the generated intent table when receiving multicast traffic. (f) Compile the forwarding flow table to the border switch according to the intent.

4.3 Process of joining multicast group in multiple scenarios

4.3.1 Scenario of SDN host joining IP network multicast group

As shown in Figure 5, the process of joining an IP network multicast group to an SDN host consists of five main steps. (i) Host in SDN sends IGMP join request message to border switch, which sends the request message to ONOS Controller through Packet-in message (ii) ONOS Controller calls IGMP module to parse the message information (multicast

source [S] and multicast group), marks Switch/Port as multicast data O-port, combines the above information into a triplet $\langle \text{Switch/O-Port}, S, G \rangle$, and stores the MCAST module. (iii) MFWD module generates intention according to the change generated by the multicast information of MCAST module, so that the grafting of the multicast tree is realized from SDN to IP network. (iv) The MCAST module searches the Switch/I-Port (In-port) of multicast source by looking up the unicast routing table, i.e., the switch port directly connected to the next-hop router from SDN to multicast source, then adds IGMP to request packet and sends it to switch, and finally forwards it to the next-hop router of IP network. (v) When multicast traffic arrives, it searches intention table, sends the flow table, and forwards the multicast traffic received at the I-port to all multicast O-ports. Now, it can realize the interconnection of multicast paths within SDN. The linkage process between the above modules is shown in Figure 6.

4.3.2 Scenario of IP host joining SDN multicast group

As shown in Figure 5, the process of joining SDN multicast group for a host in IP network consists of six steps. (i) IP network Target Host 2 in the IP network sends IGMP join request message to the connected router. (ii) After receiving the IGMP message, the router parses it, locates the port of SDN border switch according to the routing table, and sends

PIM Join message to the switch port. (iii) After receiving the PIM Join message, the switch sends the received PIM join message to ONOS Controller through Packet-in message. (iv) ONOS Controller calls PIM module to parse the message information (multicast source [S] and multicast group address [G]), marks the port Switch/O-Port as the multicast data O-port, combines the above information into a triplet <Switch/O-Port, S, G>, and stores it in the MCAST module. (v) MFWD module generates intention according to the change of routing table, and it implements the grafting of multicast tree from SDN to IP network. (vi) When multicast data arrive, it looks up the intention table and sends the flow table, then forwards the multicast data received at the I-port to all multicast O-ports, and finally implements the conduction of multicast paths within the SDN. The linkage process between the above modules is shown in Figure 7.

4.3.3 Scenario of IP A host joining IP B multicast group across SDN

As shown in Figure 8, the process of joining IP network B multicast group across SDN by a host A in IP network consists of seven steps. (i) The target2 in IP network A sends IGMP join request message to the connected router. (ii) After receiving the message, the router searches the port of SDN boundary switch according to the routing table and sends PIM Join message to the switch port. (iii) After receiving the PIM Join message, the switch sends the message to ONOS

Controller through Packet_in message. (iv) ONOS Controller calls the PIM module to parse the message information (multicast source [S] and multicast group address [G]), marks this port SwitchO-Port as the multicast data O-port, combines the above information into a triplet <SwitchO-Port, S, G>, and stores it in the MCAST module. (v) MFWD module generates intention according to the change of multicast information in MCAST module. (vi) The MCAST module searches the Switch/I-Port of the multicast source by looking up the unicast routing table and sends the PIM Join message to the I-port, so that the designated router of IP network B can receive this message and the grafting of multicast trees is realized. (vii) When multicast data arrives it looks up the relevant intention table and sends the flow table, and forwards the multicast data received at the I-port to all multicast O-ports to enable the multicast paths within the SDN, so as to realize the conduction of multicast paths from IP network A hosts across the SDN to IP network B. The linkage process between the above modules is shown in Figure 9.

4.4 Process of leaving multicast group in multiple scenarios

4.4.1 Scenario of SDN group members leaving multicast group of external IP network

As shown in Figure 5, the process of SDN multicast member leaving external IP network multicast group mainly includes

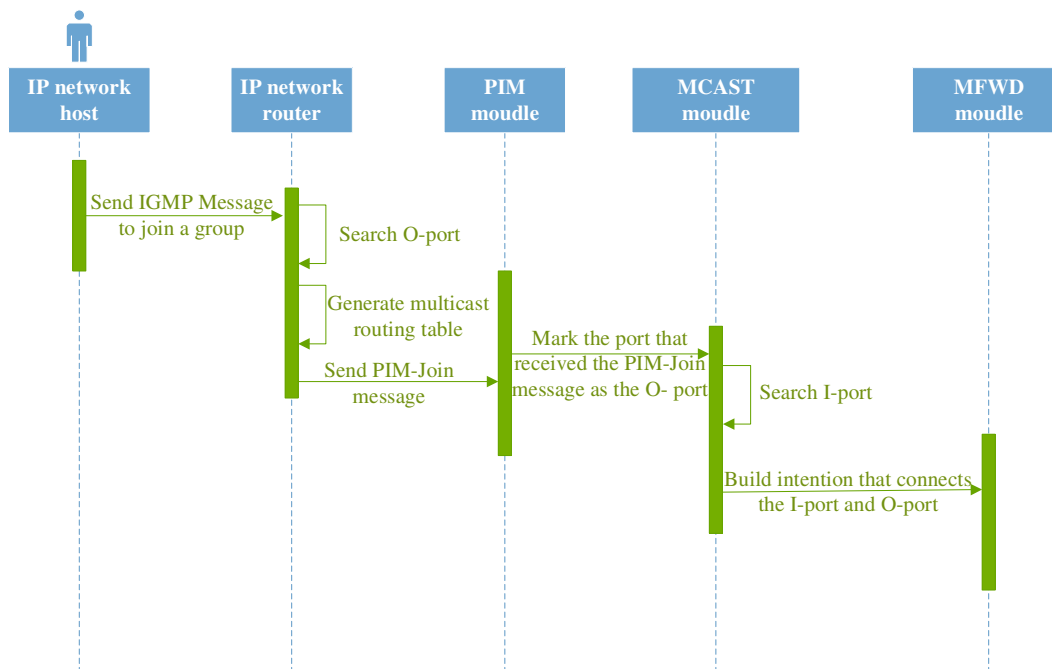


Figure 7: IP network host joins SDN multicast group.

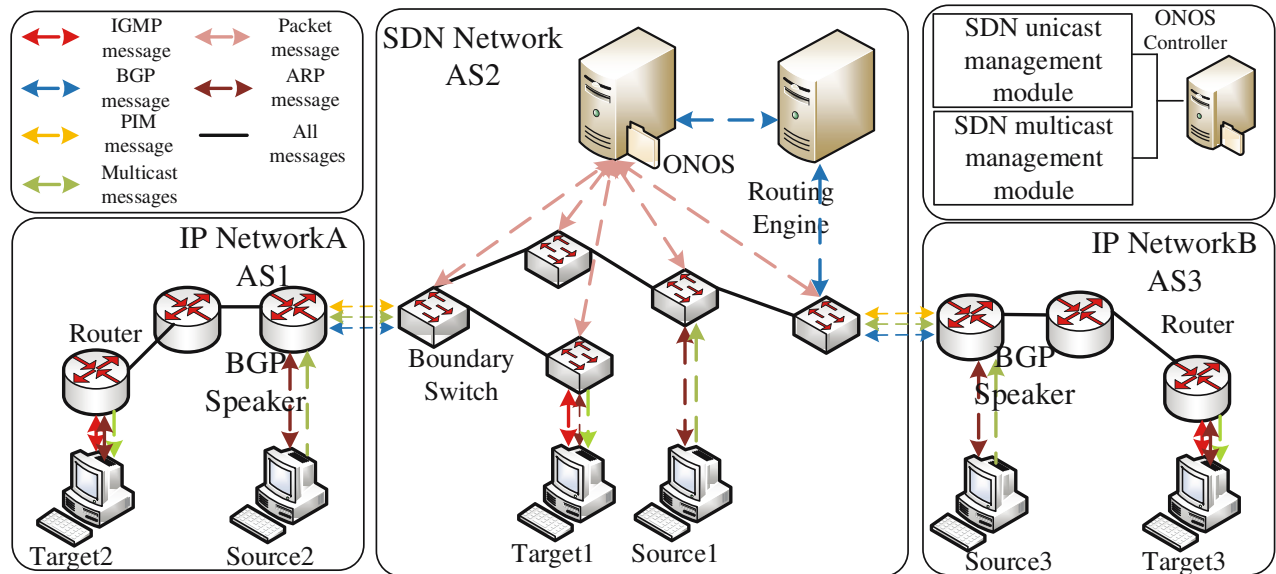


Figure 8: Schematic diagram of the scenario where host A of IP network joins multicast group B of IP network across SDN.

five steps. (i) SDN multicast member sends IGMP leave request message to the border switch. (ii) The switch sends the received message to the ONO Controller through Packet-in message. (iii) ONOS Controller calls IGMP module to parse the message information (multicast source [S] and multicast group address [G]). (iv) ONOS Controller queries the multicast routing

table, PIM module generates the corresponding PIM-Prune message and sends it to multicast source I-port. (v) MCAST module deletes the relevant routing table, updates the intention, and sends the flow table to the border switch to prune the SDN from the IP network multicast tree. The linkage process between the above modules is shown in Figure 10.

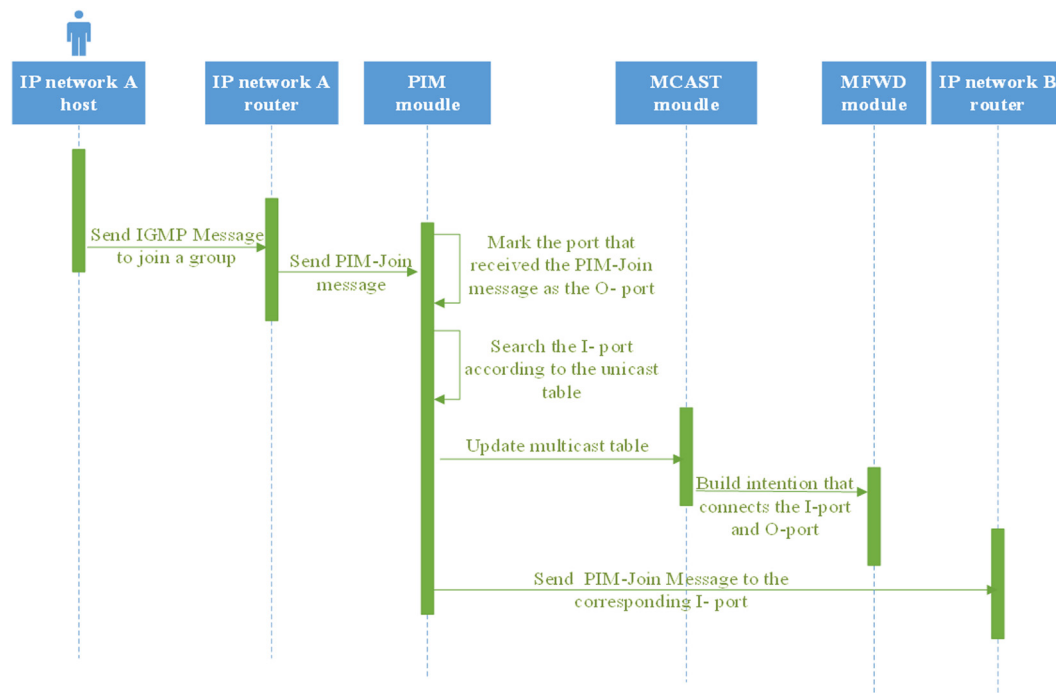


Figure 9: Flow chart of IP network host joining IP network multicast group across SDN.

4.4.2 Scenario of IP group members leaving SDN multicast group

As shown in Figure 5, the process for IP network group member to leave SDN multicast group consists of five main steps. (i) IP network group members send IGMP leave request message to the connected router. (ii) After receiving IGMP message, the router looks up the SDN boundary switch according to routing table and sends PIM-Prune packets to that switch port. (iii) After receiving the PIM-Prune message, the switch sends the request message to ONOS Controller by sending a Packet-in message. (iv) ONOS Controller calls PIM module to parse the message information (multicast source [S] and multicast group address [G]). (v) ONOS Controller queries the multicast routing table, and MCAST module deletes the relevant routing table, updates the intention, and sends the flow table to the border switch to prune multicast tree from IP network to SDN. The linkage process between the above modules is shown in Figure 11.

4.4.3 Scenario of IP group A members leaving IP network B multicast group across SDN

As shown in Figure 8, the process of IP network group member leaving an IP network multicast group across SDN consists of six steps. (i) IP network group member

send IGMP message to the connected router. (ii) After receiving IGMP message, the router looks up the SDN boundary switch according to routing table and sends PIM-Prune message to that switch port. (iii) The switch sends the PIM-Prune message to ONOS Controller through Packet-in messages. (iv) ONOS Controller calls PIM module, marks the port as O-port, and parses the message information (multicast source [S] and multicast group address [G]). (v) ONOS Controller queries the multicast routing table, and MCAST module deletes the relevant routing table, updates the intention, and sends flow table to border switch. (vi) The switch forwards PIM-Prune message to I-port, so that the designated router of IP network B can receive the messages and the pruning of multicast tree from IP network A to IP network B across the SDN is completed. The linkage process between the above modules is shown in Figure 12.

5 Deployment scheme and simulation test

We built a test environment to verify the function and performance of the data interaction based on the multicast protocol in the SDN-IP hybrid networking multicast architecture, which includes interconnection, end-to-end delay,

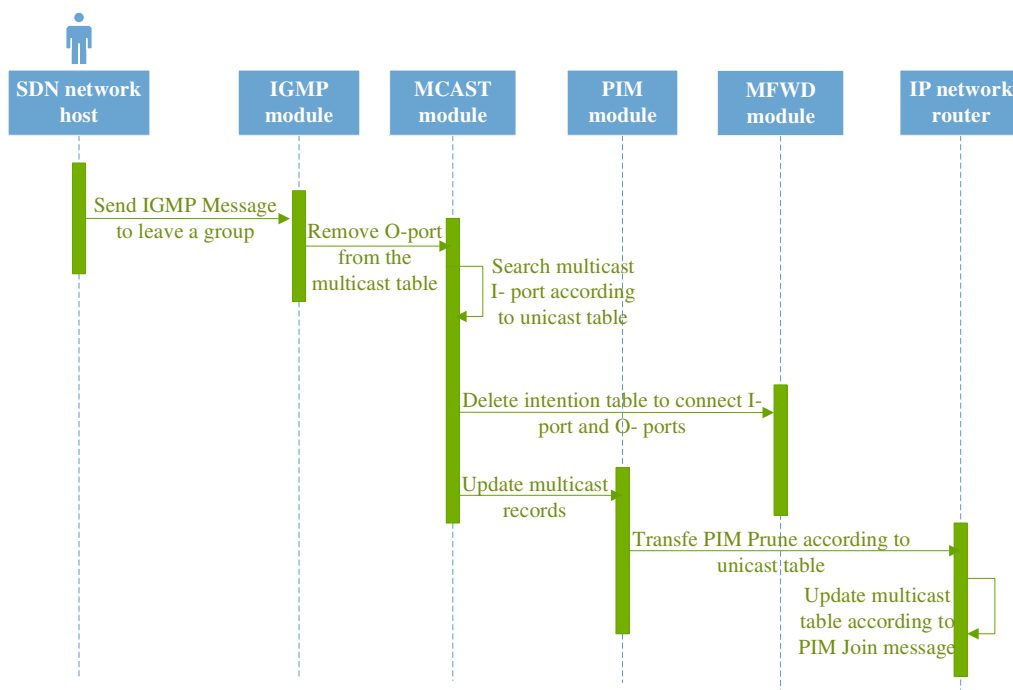


Figure 10: Flow chart of SDN group members leaving IP network multicast group.

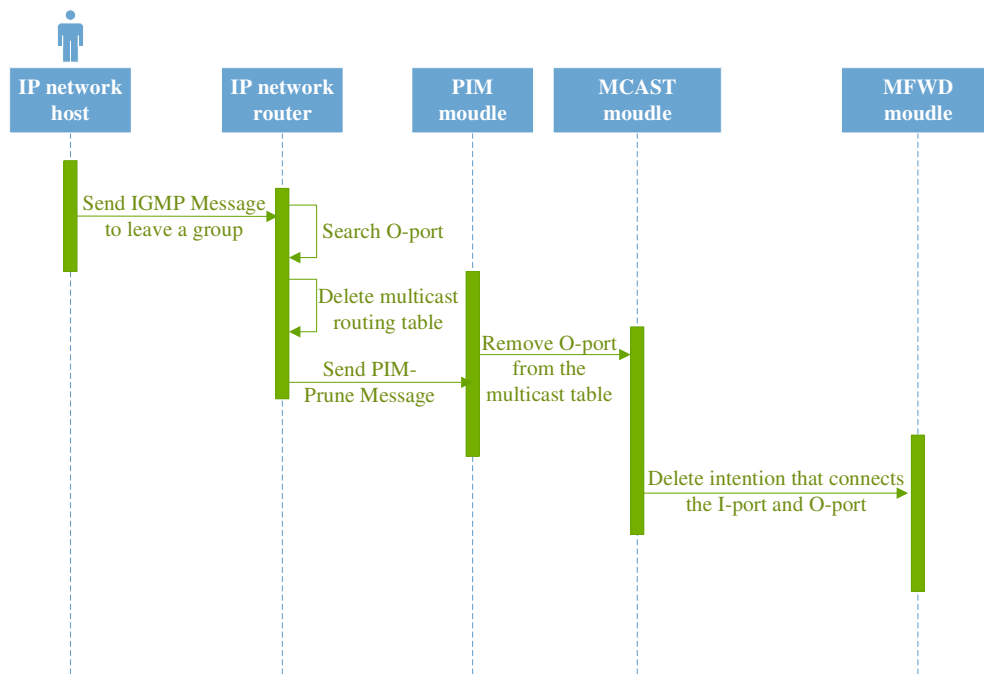


Figure 11: Flow chart of IP network group members leaving SDN multicast group.

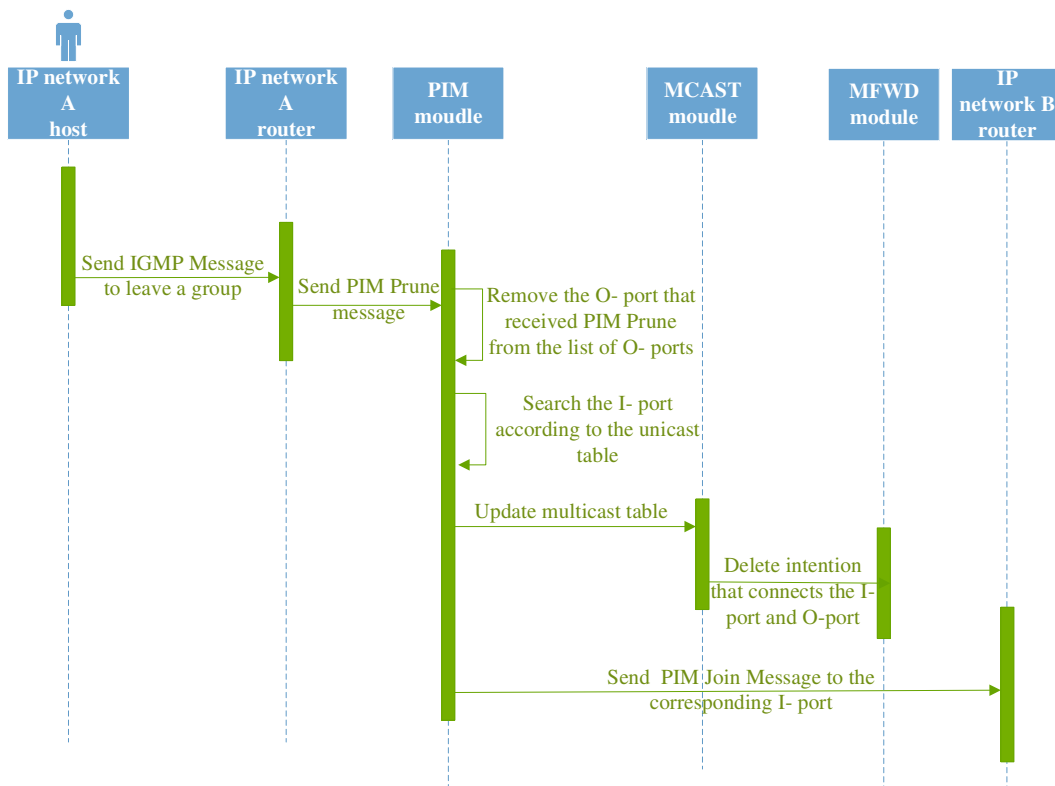


Figure 12: Flow chart of IP network A group members leaving IP network B multicast group across SDN.

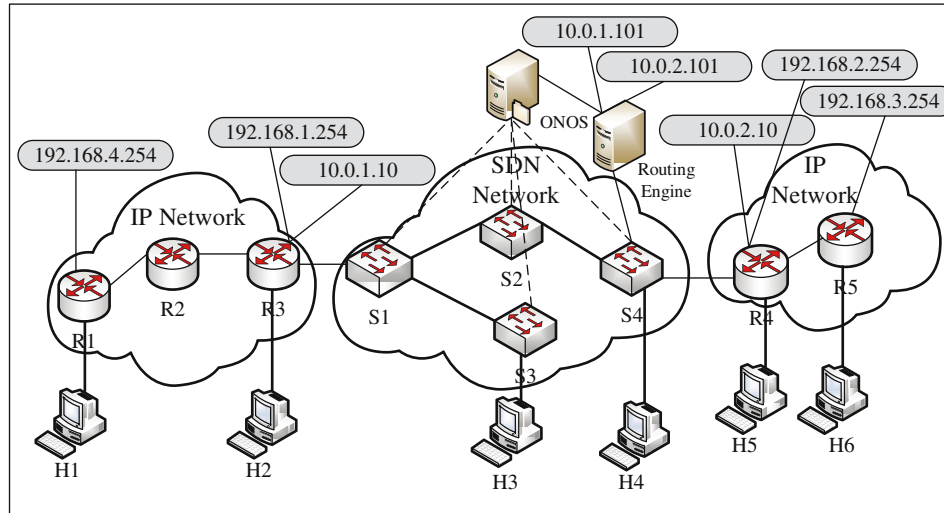


Figure 13: Test environment topology diagram.

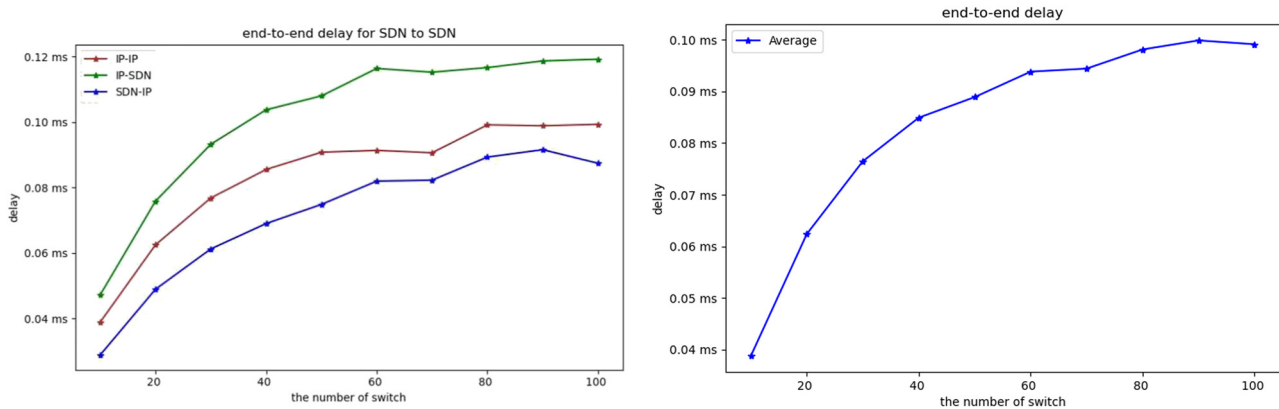


Figure 14: End to end delay test curve under three scenarios.

and packet loss test of the multicast architecture in multiple scenarios. The test environment mainly includes data interaction based on the multicast protocol in the SDN-IP hybrid networking multicast (Figure 13).

5.1 End-to-end delay test

End-to-end delay is the time from when the multicast source sends data to the time of receiver successfully receives data, which is a very sensitive parameter in multicast protocol. In this article, we mainly test the end-to-end delay in three scenarios for data interaction based on multicast protocol in SDN-IP hybrid networking multicast architecture: SDN source host to IP network target host, IP network source host to SDN target host and IP network source host to IP network target host across SDN. The

measurement method is as follows: putting a timestamp in the message sent by the multicast source, parsing the timestamp after receiving the message, and subtracting the timestamp on the basis of the receiving time to calculate the delay between the multicast source and the receiver. The test results are shown in Figure 14, from which it can be seen that the end-to-end delay of the above three scenarios is within 0.2 ms and that the average end-to-end delay of the above three scenarios is within 0.1 ms, which can meet the requirements of commercial aerospace missions for network transmission delay.

5.2 Packet loss rate test

Packet loss rate refers to the ratio of the number of lost packets to the total amount of data sent during data

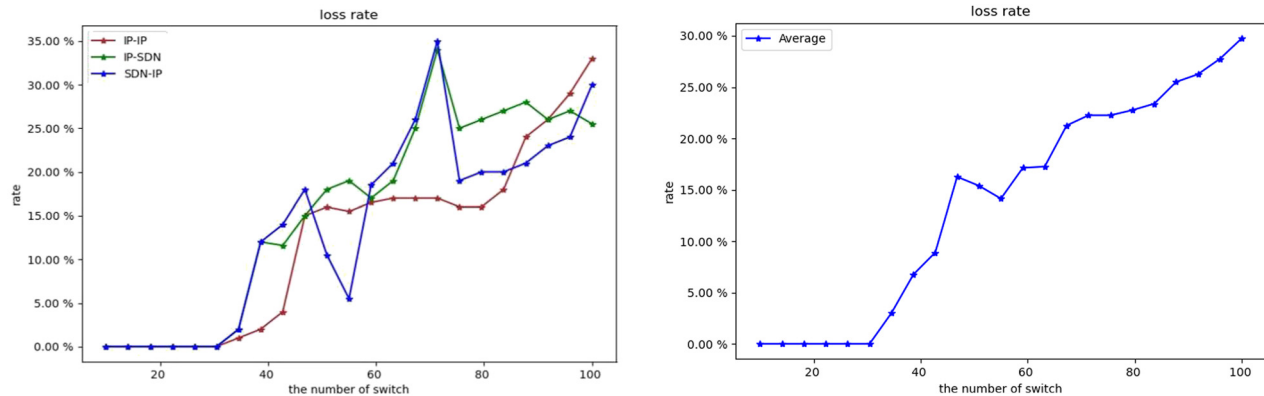


Figure 15: Packet loss rate test curve under three scenarios.

transmission. In this article, the packet loss rate test scenario is consistent with the end-to-end delay test. An Iperf client is started on each multicast source (sender) to send user data protocol (UDP) packets with network traffic from 2 to 100 Mbps, and the destination multicast receiver (receiver) binds the multicast group using Iperf UDP server mode to receive UDP packets. The packet loss rate of SDN in the case of single controller is shown in Figure 15. It can be seen that packets start to be lost when the network traffic is greater than 30 Mbps, and the average packet loss rate is 30% when the network traffic is greater than 100 Mbps. After analysis, the packet loss rate is 0 when the network traffic is less than 30 Mbps. The reason for packet loss after greater than 30 Mbps is that ONOS Controller is overloaded. The above packet loss problem can be solved by deploying multiple controllers mode.

6 Conclusions

SDN is applied to the cloud platform in the data center. It can realize automatic deployment and configuration of resources, rapid service rollout, and flexible expansion so as to alleviate the scalability and flexibility pressure faced by large-scale cloud data center when carrying multi-user services. However, SDN cannot completely replace IP networks, so there will be a coexistence of SDNs and IP networks. It is challenging to achieve SDN and IP network for data interaction based on multicast protocol. In this article, we design a new network architecture for data interaction based on unicast/multicast protocols. In order to solve the key technical difficulties of data interaction based on multicast protocol in the above two types of networks, this article designs and constructs a new network architecture based on single/multicast protocol for data interaction between the

two networks in multiple scenarios by creating unicast management module and multicast management module on the SDN controller. Our architecture has low end-to-end latency and low packet loss, which is conducive to deployment and implementation in actual data center environments and provides a solid technical foundation for the future application of SDN technology in commercial aerospace data center cloud platforms.

Funding information: The authors state no funding involved.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interest.

References

- AlSaeed Z, Ahmad I, Hussain I. 2018. Multicasting in software defined networks: A comprehensive survey. *J Netw Comput Netw.* 104:61–77.
- Alssaheli O, Zainal Abidin Z, Zakaria N, Abal Abas Z. 2021. Implementation of network traffic monitoring using software defined networking Ryu controller. *WSEAS Trans Syst Control.* 16:270–277.
- Babbar H, Rani S, Singh A, Abd-Elnaby M, Choi BJ. 2021. Cloud based smart city services for industrial internet of things in software-defined networking. *Sustainability.* 13(16):8910.
- Balaji G, Paul J, Timo T. 2022. Quagga Routing Suite. In <http://www.nongnu.org/quagga/>.
- Berde P, Gerola M, Hart J, Higuchi Y, Kobayashi M, Koide T, Lantz B, O'Connor B, Radoslavov P, Snow W, et al. 2014. ONOS: towards an open, distributed SDN OS. *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*; 2014 Aug 22; Chicago (IL), USA. ACM, 2014. p. 1–6.
- Caiyun D, Ying Z, Weiguo D. 2014. Research on specified source multicast in space TT C communication network. *Radio Engineering.* 44(9):3.

- Chaokun Z, Yong C, Heyi T, Jianping W. 2014. State-of-the-art survey on software-defined networking (SDN). *J Softw.* 26(1):62–81.
- Feamster N, Rexford J, Zegura E. 2014. The road to SDN: An intellectual history of programmable networks. *Comput Commun Rev.* 44(2):87–98.
- Linux Foundation. 2022. Open vSwitch. <http://openvswitch.org>.
- Huang L, Zhi X, Gao Q, Kausar S, Zheng S. 2016. Design and implementation of multicast routing system over SDN and sFlow. 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN); 2016 Jun 4–6; Beijing, China. IEEE, 2016, p. 524–529.
- Kotachi S, Sato T, Shinkuma R, Oki E. 2019. Multicast routing model to minimize number of flow entries in software-defined network. 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS); 2019 Sep 18–20; Matsue, Japan. IEEE, 2019.
- Lan T, Xuezhi Z. 2000. Research of IP multicast communication mode and correlative problems. *J Xian Technol Univ.* 20(3):209–214.
- Lin P, Hart J, Krishnaswamy U, Murakami T, Kobayashi M, Al-Shabibi A, Wang K-C, Bi J. 2013. Seamless interworking of SDN and IP. *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*; 2013 Aug 12–16; Hong Kong, China. ACM, 2013. p. 475–6.
- Mohamed A, Hamdan M, Khan S, Abdelaziz A, Babiker SF, Imran M, Marsono MN. 2021. Software-defined networks for resource allocation in cloud computing: A survey. *Comput Netw.* 195:108151.
- Nascimento MR, Rothenberg CE, Salvador MR. 2011. Virtual routers as a service: the RouteFlow approach Leveraging software-defined networks. *Proceedings of the 6th International Conference on Future Internet Technologies*; 2011 Jun 11–15; Seoul, Korea. p. 34–37.
- Oliveira CA, Pardalos PM. 2005. A survey of combinatorial optimization problems in multicast routing. *Comput Oper Res.* 32(8):1953–1981.
- Quinn B, Almeroth K. 2001. IP multicast applications: Challenges and solutions. Technical report.
- Sharma R. 2021. A review on software defined networking. *Int J Sci Res Comput Sci Eng Inf Technol*, pp. 11–14.
- Stallings W. 2015. *Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional, pp. 560.
- Wibowo FX, Gregory MA, Ahmed K, Gomez KM. 2017. Multi-domain software defined networking: research status and challenges. *J Netw Comput Appl.* 87:32–45.
- Yanwei S, Zheng C. 2014. IMISA: Interconnection mechanism for IP subnet and SDN subnet in autonomous system. *J Commun.* 35:76–81.
- Yong T, Weizhe W, Wenying W. 2017. The design and implement of the interconnected architecture of SDN and traditional IP network. *Comput Eng & Sci.* 39(12):7.