ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol.12 (2022) No.1, pp.229-240
DOI:10.33168/JSMS.2022.0116

A Network Federation Scheme for Inter-Domain SDN Communications

Dongkyun Kim , Yong-Hwan Kim

Korea Institute of Science and Technology Information, Daejeon, South Korea.
mirr@kisti.re.kr)

Abstract. Recently, a variety of the integrated service platforms demand the innovative network technologies such as software-defined networking (SDN). However, it is problematic and complicated to provide interconnectivity and reachability between inter-domain SDNs. Thus, this paper proposes a scheme for the inter-SDN federation and communication based on the abstracted network information exchange and automated peer network configuration for the integrated network services. The proposed scheme is implemented and verified over the de-facto wide-area SDN infrastructure (KREONET-S) which is deployed on Korea Research Environment Open Network (KREONET), a representative national research network in Korea. The proposed method is implemented and experimented based on the distributed and layer-2 oriented network architecture by using the abstracted message exchanges between local SDN controllers in order to provide low message overheads and high federation scalability. Our research work contributes to more efficient inter-SDN communications by means of the data modeling abstraction of the local SDN network information and the simplified interactions between individual controllers by the automated configurations through the specific application interfaces. The proposed scheme also contributes to the development of the integrated service platforms which are demanded to provide IoT, Big Data, AI, and cloud services in the tightly coupled manner over the inter-SDN domains.

Keywords: SDN, Federation, Inter-Domain, Inter-SDN, KREONET-S

1. Introduction

These days, the integrated service platforms are demanded to provide IoT, Big Data, AI, and cloud services (Hyun et al., 2020), based on the advanced networking capability in the tightly coupled manner. However traditional networks hardly accommodate the proper network characteristics required for advanced applications because of the existing hardware-oriented, static, and manually-operated network environment. Software-defined Network (SDN) is an innovative network technology to enable software-centric, dynamic, automated and intelligent network environment (McKeown et al., 2008; Nadeau et al., 2013), using the centralized network control and management system, which is derived from the control plane and data plane separation architecture of SDN. However, SDN needs additional methods for the inter-domain SDN (inter-SDN) communications in the same way that the inter-domain networking and routing over the Internet is generally supported by Border Gateway Protocol (BGP).

In the early stage, an individual SDN domain connects to the Internet by configuring a specific SDN-IP gateway to peer with another SDN domain over the layer-3 network (Lin et al., 2013; Lin et al., 2014). This method exploits BGP protocol, which causes packet loss and performance degradation problems due to the delayed routing update issues (Lin et al., 2014; Gupta et al., 2015) when it comes to the inter-SDN communications. Therefore, several studies suggest the modified BGP protocols to improve the routing update methods over the inter-domain SDNs (Gupta et al., 2015; Kotronis et al., 2012), while other research work proposes the routing information exchange schemes by use of the East-West interface of the SDN architecture (Lin et al., 2015; Lin et al., 2014). However, the suggested inter-SDN methods that incorporate the routing information updates should cope with the several problems such as multi-field prefix matching, end-to-end routing between SDN domains, and multi-path data transmission (Zhou et al., 2017).

Furthermore, in terms of the applicable areas, the proposed inter-SDN communication scheme is able to be applied for the several application areas such as IoT, Big Data, AI, and cloud services where the service resources are not only provided in each of the distributed SDN domains, but demanded to have the SDN domains connected for the integrated services based on the on-demand virtual network embedding over the separate virtual network operators (Wang et al., 2016). For instance, a great deal of the distributed IoT sensors in multiple SDN domains should connect to the cloud data center in an individual SDN domain using the dedicated and high-performance network bandwidth embedded by the virtual network for the high performance data transfer and Big Data analysis, which is associated with the machine learning technologies (Fisher et al, 2013).

In the meantime, a global centralized controller can be placed to control the communication between the multiple controllers of the different SDN domains

based on a research work (Xie et al., 2019) that allows each local controller to rapidly exchange control messages with the relatively low costs and high data transmission performance by establishing layer-2 network connections to other controllers. The research issue of this work is that there are considerable message overheads and scalability problem due to the several interconnections for the data communication among global and local controllers. In this context, we propose an inter-SDN federation scheme based on the distributed and layer-2 (L2) oriented network architecture using the direct and simplified message exchanges between the local SDN controllers (with the global controller excluded) in order to provide low message overheads and high federation scalability. Our research work contributes to more efficient inter-SDN architecture by means of the data modeling abstraction of the local SDN network information and simplified interactions between controllers by use of the automated configurations and the specific application interfaces described in the next section.

2. Federation System Architecture

In this Section, we introduce the general architecture and procedures of inter-SDN federation system. Moreover, it is explained how end-hosts in one SDN domain perform data communications with another SDN domain based on the proposed abstracted data model and automated configuration procedures including the virtual end-host creation using a specific address resolution protocol (ARP) handling method.

2.1. Architecture and Procedures

The proposed inter-SDN federation system assumes an integrated SDN communication architecture that consists of multiple SDN domains as shown in Fig. 1, where the detailed internal components in an SDN domain are indicated as follow: 1) physical network (N: layer-2 based transmission network including end-host, switch, and link), 2) federation application (A, integrated control application of inter-SDN federation system), 3) controller/cluster (C, abstracted information exchange between SDN domains and admission control of the network and end-host), 4) data store (D, optional, abstracted information storing as an alternative store of controller/cluster (C)). In addition, the data plane of different and individual SDN domains should be connected physically using the layer-2 link, while the control plane of each controller/cluster needs to be reachable via layer-3 network (a.k.a. Internet). It is also supposed in the system architecture that each SDN controller should acquire the network topology information automatically from the physical network via southbound interface, and the application program should collect and modify network event information via northbound interface of the controller.

The internal components in the SDN domain performs the following detailed

procedures for the inter-SDN federation:
(1) (A ⇒ C) providing the abstracted data modeling schema (JSON, XML) for the controller/control cluster (C) at the runtime of federation application, where the schema contains: a controller IP address, a border switch identifier, a border switch port identifier, a list of the end-host identifier (e.g., MAC/IP address, etc.)

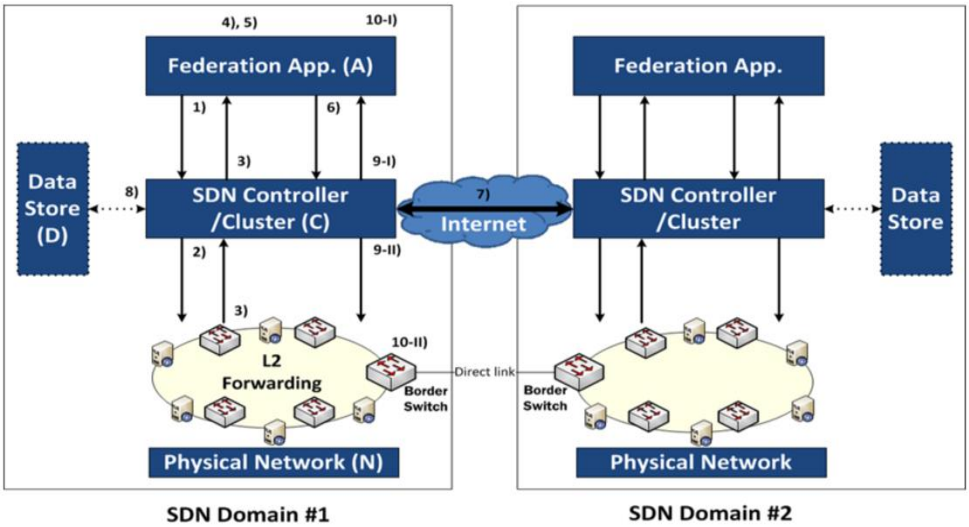


Fig. 1: Inter-SDN Federation Process and Architecture

(2) (C ⇒ N) requesting the physical network (N) topology information
(3) (C ⇒ N, N ⇒ C ⇒ A) providing the N topology information
(4) (A) selecting a list of the end-hosts to be federated with the peer SDN domain using the application interfaces such as graphical user interface (GUI), REST API, and command line interface (CLI)
(5) (A) generating network federation information based on the pre-defined data modeling schema (e.g., JSON)
(6) (A ⇒ C) delivering a command to execute the inter-SDN federation
(7) (Between C) exchanging SDN federation information after acknowledging the peer domain's federation request via a secured Internet connection using REST API calls between SDN controller/clusters of each domain
(8) (C) storing the exchanged network information as an integrated data model
(9) (C ⇔ D, optional) backing up the federated network information
(10) (C ⇒ A) delivering an acknowledge message for the federation request (as an execution command)
(11) (C ⇒ N, border switch) generating physical network connections for the federated end-hosts of the peer SDN domain at the specified port of the border switch
(12) Completing the inter-SDN federation procedure

2.2. ARP Handling for Virtual End-host Creation

Basically, the IP address to MAC address resolution should be carried out before a source end-host sends data to a destination end-host over the L2-based SDN environment (e.g., OpenFlow-based network). Therefore, each end-host uses ARP to find a MAC address of the destination end-host and transmits data by looking up the flow rules corresponding to the MAC address. However, ARP can only acquire the MAC address of the end-hosts connected in the same subnet or in the same SDN domain where a controller provides proxy ARP services.

In this context, this paper proposes an ARP handing method to create the virtual (bogus) end-host in an SDN domain (e.g., SDN domain #1 in Fig. 1). The inter-SDN federation system generates the virtual end-host which connects to a port of the border switch in an SDN domain, where the virtual end-host is an acknowledged physical end-host in the peer SDN domain (e.g., SDN domain #2). Thus, when the SDN controller receives an ARP request message from the source end-host that wants to communicate with the acknowledged end-host in the peer SDN domain, the controller generates an ARP reply message including the MAC address mapped to the destination IP address of the peer end-host, and delivers the message to the source end-host so that L2-based data transmission starts. After the data transmission and the inter-SDN federation session is completed, the proposed ARP handling method is disabled and all the virtual end-hosts in the SDN domain are removed.

In this way, an SDN controller maintains a peer domain's acknowledged end-host information in the SDN domain by exploiting the border switch (which has the layer-2 connectivity to the peer domain's border switch) as a proxy device for the peer end-hosts. For example, if a source end-host in the SDN domain A wants to transfer data to a destination end-host in the SDN domain B using OpenFlow as an SDN southbound interface, the SDN controller in the domain A receives a PACKET-IN message and simply makes a flow rule to forward the packet to the border switch in the domain A, where the (bogus) destination end-host connects virtually. After this procedure, the packet arrives at the SDN domain B via the border switch in the SDN domain A, then it is delivered to the (real) physical destination end-host using the flow rules generated by the SDN controller in the domain B.

3. System Implementation and Experiments

The proposed inter-SDN federation system is implemented and verified based on the well-known open source SDN controller, Open Network Operating system (ONOS, 2021; Berde et al., 2014), and ONOS-based Virtual Dedicated Network (VDN) system (Kim, D et al., 2017; Kim, D et al., 2018). Also, the data plane network is configured for simulation and experiments using Mininet (Mininet, 2021)

in two different servers each of which is used to construct the individual SDN domains where the control plane and data plane are connected via an Internet link and a 1Gbps dedicated link, respectively, between two SDN domains. In addition, the implemented inter-SDN federation system is integrated with the VDN system for the bandwidth-guaranteed data transmission. The proposed inter-SDN federation system provides five principal functionalities as follows:

- Simplified and abstracted network federation information management using ONOS and JSON model, based on the following information: a public IP address of controller, a border switch identifier, a border switch port identifier, a list of acknowledged end-hosts
- Inter-domain SDN connection request, acknowledgement, and release using the above network federation information
- Acknowledged end-host registration and configuration in association with the VDN system
- Proxy configuration for end-to-end communication between individual SDN domains
- Web based graphical user interface (GUI) and REST API for the inter-SDN federation management

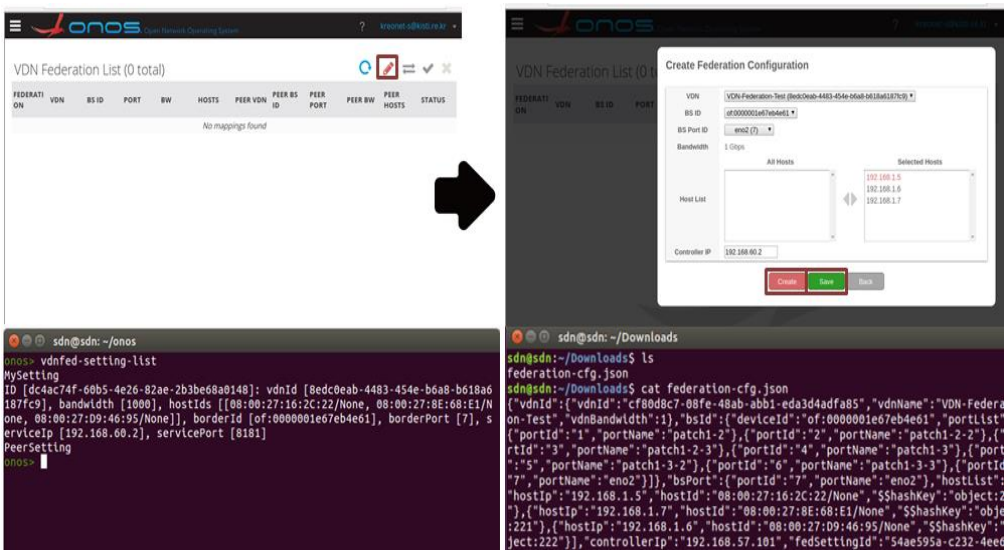
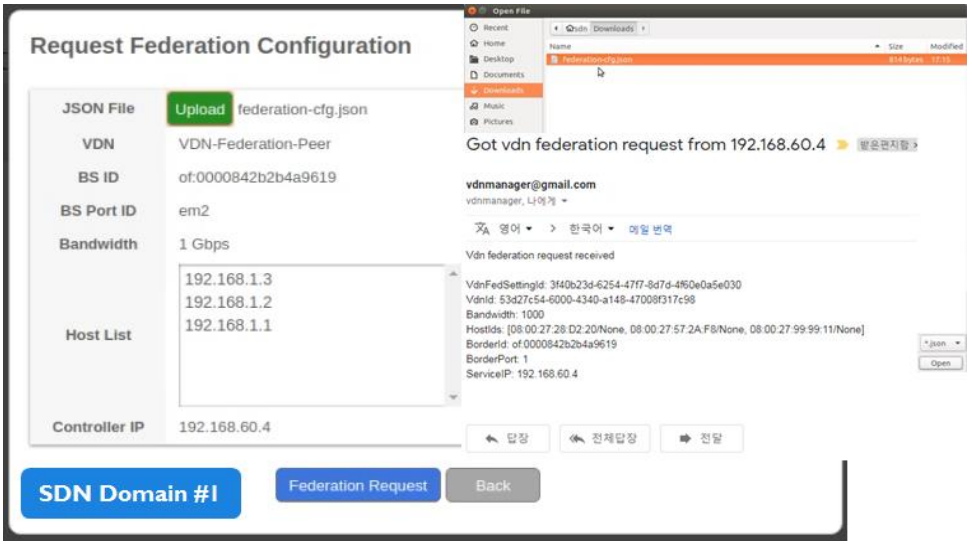


Fig. 2: Graphical User Interface to Generate SDN Federation Information

In order to federate the inter-domain SDNs, a virtual dedicated network (VDN) should be created by the VDN system in each SDN domain in advance, and the individual domains need to be connected through a dedicated link. Then, each domain's abstracted network federation information is generated in the ONOS memory and the JSON data model as well. The abstracted JSON dataset is used to

deliver network federation information to the peer SDN domain later. The Fig. 2 indicates the user interface (GUI) to generate the network federation information, where each VDN contains the specific (acknowledged) end-hosts or every end-host in the SDN domain.

The pre-generated JSON dataset including the network federation information is delivered to the peer SDN domain's federation application by uploading the JSON dataset using the GUI as shown in Fig. 3, or using REST API calls. And the peer domain's JSON dataset is brought to the originated SDN domain in the same



way. After that, each SDN domain's administrators can confirm the SDN federation requests based on the automated notification via e-mail in Fig. 3.

Fig. 3: Graphical User Interface for the Inter-SDN Federation Request

When the federation request is confirmed in two domains (SDN domain #1 and SDN domain #2 in Fig. 4.), each domain's pre-created VDN is automatically updated to include a border switch, and the requested end-hosts of the peer SDN domain (SDN domain #2 in Fig. 4) which are virtually created and connected to the border switch in the SDN Domain #1 as the virtual bogus end-hosts as shown in Fig. 4. At this time, the proposed ARP handling method is applied to achieve the successful data transmission between the federated SDN domains. It is identified that the inter-SDN federation procedures are completed in Fig. 4. It is also indicated that the end-hosts in the different SDN domains can communicate with each other after the distributed SDN domains are federated in Fig. 5.

In the meanwhile, the proposed federation system conducts the simple, direct, and abstracted network federation information exchanges, and performs the related configurations independently in each SDN domain. Therefore, multiple SDN domains (more than two domains) can also be federated based on the proposed

system architecture and design, which makes the federation more scalable with the reduced message overheads.

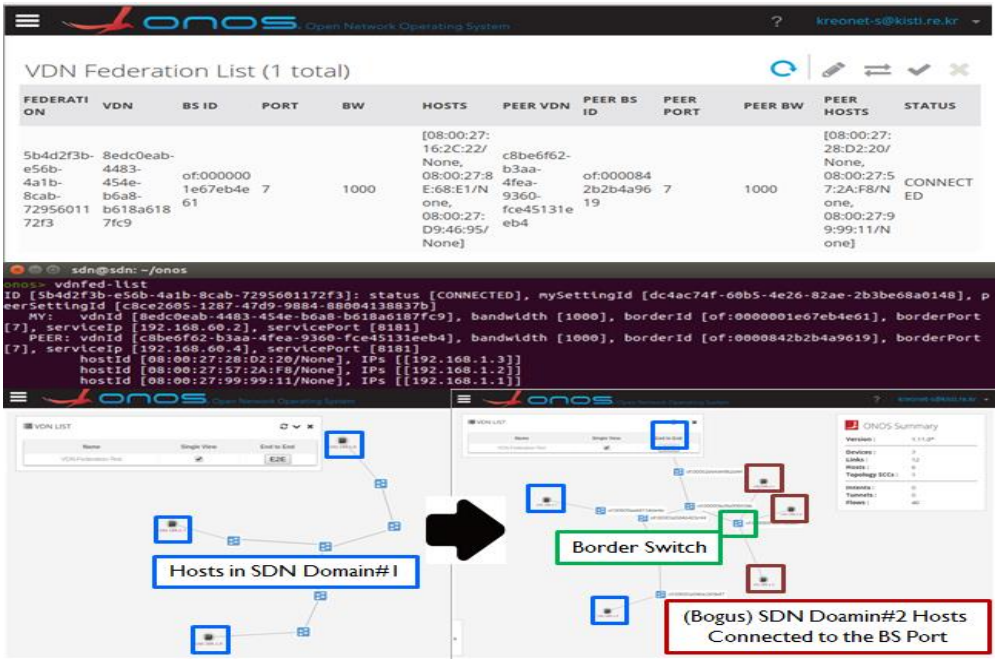


Fig. 4: Inter-SDN Federation and VDN Update Results including Peer SDN End-hosts

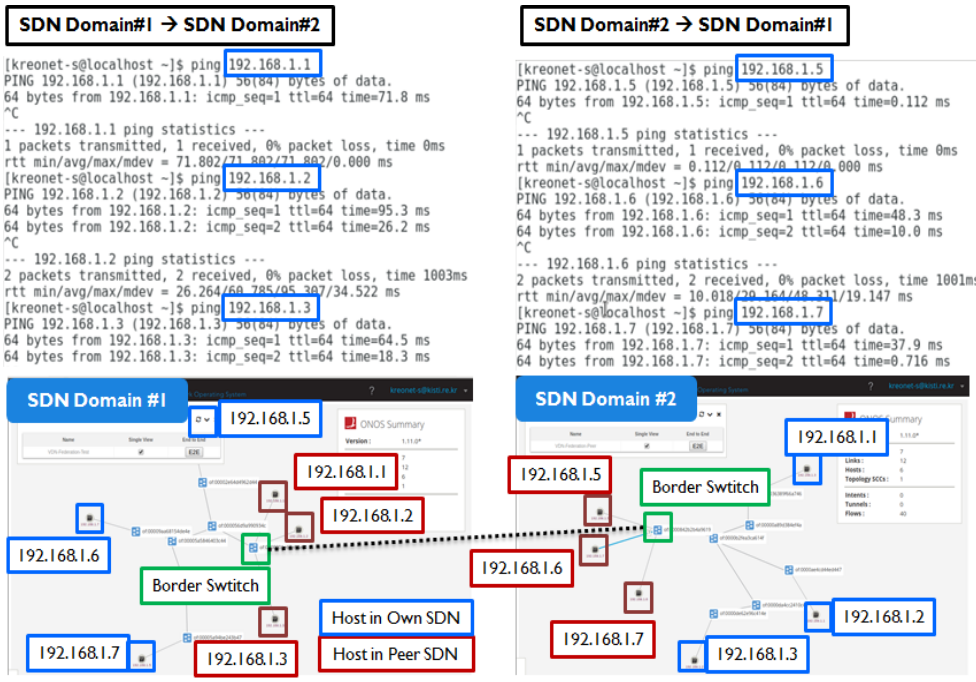


Fig. 5: End-to-end Communication Experiment based on the Proposed Federation System

For instance, supposing the SDN domain #3 has a connection to the existing SDN domain #2, and the end-hosts in the domain #3 want to communicate with the end-host A in the domain #1 and the end-host B in the domain #2, the multiple domain federation can be made with the message exchange simplified as follows: 1) the federation system of the domain #3 generates abstracted network federation information and requests for the domain #2 federation, 2) the federation system of the domain #2 generates its own abstracted federation information containing the end-host A and end-host B, then requests for the domain #3 federation (here, note that there is no additional message exchange between domain #2 and domain #1), 3) both of the domain #2 and domain #3 confirm the requests from each other.

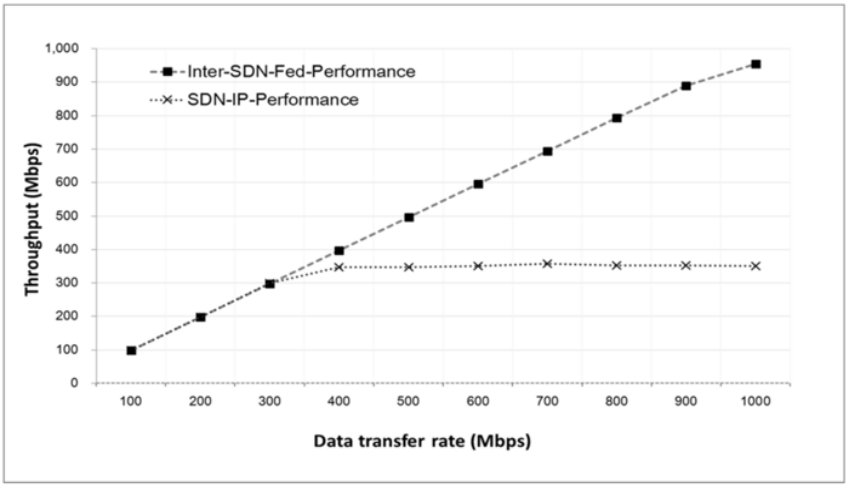


Fig. 6: Inter-SDN Federation Performance Experiment Results

Fig. 6 describes the network performance comparison between the proposed federation architecture and the previous Internet-based SDN-IP architecture using BGP, which are implemented on the wide-area SDN infrastructure in Korea (KREONET-S, 2021). The well-known performance measurement tool, iperf3, is used for the performance experiment where the result indicates that the better network transmission rate is achieved by the proposed scheme as shown in Fig. 6. According to the performance experiment, similar results are measured in the case of TCP and UDP traffic. The BGP-based network (SDN-IP) performance is saturated at around 350Mbps, while the inter-SDN federation network performance increased to around 950Mbps in proportion to the data transmission rate up to 1Gbps. The performance of the inter-SDN federation network is enhanced and guaranteed based on the dedicated virtual networking between source and destination end-hosts connected to the SDN domains, as provisioned by the proposed inter-SDN federation system. By comparison, the BGP-based network performance appears to be unguaranteed primarily due to the routing issues raised by the abovementioned related work.

4. Conclusion

This paper proposed an inter-SDN federation system architecture and design to provide the layer-2 based interconnectivity between the distributed and multiple SDN domains by the abstracted network information exchange and automated federation process configurations for the low message overhead and high federation scalability. The proposed work is experimented and verified over the de-facto wide-area SDN infrastructure in Korea (KREONET-S), resulting in the network performance enhancement compared to the legacy network protocol. The major

limitation of our study is that only a couple of SDN domains are used for the inter-SDN federation experiment. Therefore, in order to study the multiple-domain extensibility of the proposed scheme, our future work includes more experiments and verification in the joint research work with the new SDN domains such as the national research network in China (CSTNET, China Science and Technology Network) and other R&E organizations in the global research network community.

Acknowledgement

This research was supported by Korean Institute of Science and Technology Information (KISTI).

References

Berde, P. et al. (2014). ONOS: towards an Open, Distributed SDN OS. *Proc. 3rd Workshop on Hot Topics in Software Defined Networking*. 1-6.

Fisher, A. et al. (2013). Virtual Network Embedding: A Survey. *IEEE Commun. Survey & Tutorials*. 15(4), 1888-1906.

Gray, K. and Nadeau, T. D. (2013). SDN: Software Defined Networks. New York: O'Reilly Media Inc.

Gupta, A. et al. (2015). SDX: A software defined Internet exchange. *ACM SIGCOMM Comput. Commun. Rev*. 44(4), 551-562

Hyun, Jin-Woo et al. (2020). A Study on the IoT LED Streetlight Convergence Technology for Smart City Service. *Journal of Next-generation Convergence Technology Association*, 4(2), 135-143.

Kim, D. et al. (2017). Cloud-centric and Logically Isolated Virtual Network Environment based on Software-Defined Wide Area Network. *Sustainability*. 9(12), 2382.

Kim, D. et al. (2018). Logically Isolated Group Network for Virtual Convergence Environment over SD-WAN. *The Journal of Supercomputing*. 74(2), 1-11.

Kotronis, V. et al. (2012). Outsourcing the routing control logic: Better Internet routing based on SDN principles. *Proc. 11th ACM Workshop on Hot Topics in Networks*. 55-60.

KREONET-S (2021). Daejeon: Korea Institute of Science and Technology Information. Available from http://www.kreonet-s.net/

Lin, P. et al. (2013). Seamless interworking of SDN and IP. *ACM SIGCOMM Comput. Commun. Rev*, 43(4), 475-476.

Lin, P. et al. (2014). Interworking with SDN using existing BGP. *Proc. The Ninth Int. Conf. Future Internet Technol*. 1-2.

Lin, P. et al. (2015). A west-east bridge based SDN inter-domain testbed. *IEEE Commun. Mag*. 53(2): 190-197.

Lin, P. et al. (2014). WE-bridge: West-east bridge for SDN inter-domain network peering. *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*. 111-112

McKeown, N. et al. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Compute. Commun.*, 38(2), 69-74.

Mininet (2021). Mininet Project Contributors. Available from https://mininet.org/

ONOS (2021). Open Network Operating System. Menlo Park: Open Network Foundation. Available from https://opennetworking.org/onos/

Wang, J. et al. (2016). A multi-domain SDN scalability architecture implementation based on the coordinated controller. *Int. Conf. IEEE Cyber-Enabled Comput. And Knowledge Discovery*. 494-499.

Xie, J. et al. (2019). A Survey of Machine Learning Techiques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Commun. Survey & Tutorials.* 21(1), 393-430.

Zhou, H. et al. (2017). SDN-LIRU: A lossless and seamless method for SDN inter-domain route updates. *IEEE/ACM Trans. Networking*. 25(4), 2473-2483.