IPv6 Operations and Deployment Scenarios over SDN

Chia-Wei Tseng[1,2], Sheue-Ji Chen[1], Yao-Tsung Yang[1,2], Li-Der Chou[2,4], Ce-Kuen Shieh[3,4], Sheng-Wei Huang[3]

[1]Broadband Network Laboratory, Chunghwa Telecom Laboratories, Taoyuan, Taiwan
[2]Dept. of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan
[3]Dept. of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
[4]National Center for High-performance Computing, Hsinchu, Taiwan
chiawei@cht.com.tw, sjc491@cht.com.tw, yaovct@cht.com.tw, cld@csie.ncu.edu.tw, shieh@ee.ncku.edu.tw, swh@ee.ncku.edu.tw

***Abstract*—IPv6 is a technology that provides enormous address space and end-to-end communication, features that are required in the context of the device automation integration for future network. The transition to IPv6 holds the future of the internet infrastructure. Software-defined networking (SDN) defines a new concept for computer networks that can separate and provide abstract elements of network devices. IPv6 SDN has the potential to revolutionize the network design, construct and operate networks to achieve more efficient business network agility. In this paper, we will discuss the main architectures of SDN and illustrate how IPv6 can be deployed and integrated in SDN technologies using OpenFlow mechanisms. We will also discuss the IPv6 impact on link performance and deployment scenarios.**

Keywords—IPv6, SDN, OpenFlow, Network deployment

I. INTRODUCTION

IPv6 is now being deployed globally with increasing momentum as IPv4 address exhaustion is becoming a reality for ISPs and network providers. According to google's statistics in April 2014 showed that over 3% of the google users are using native IPv6 to access google. IPv6 marks the beginning of the future network. It provides resources and functions to make software defined networking, and network virtualization easy to scale.

SDN is an approach to building computer networks that separates and abstracts elements of these systems. Several works have addressed the architecture of total control of the network including the SDN layer and network layer [1], [2], [3]. Currently, SDN is realized in OpenFlow [4]. Its design goal is to enable researchers to innovate in the computer science and related fields with networking and communication needs. SDN provides an opportunity to redesign networks by separating the software that controls the network from the network elements. OpenFlow separates the data plane from the control plane, defining an OpenFlow switch. The OpenFlow 1.2 provides basic support for IPv6. The OpenFlow 1.3 expands the IPv6 essential features including access control, quality of service and tunneling support. With IPv6 support in OpenFlow, implementers can now take full advantage of IPv6 over SDN environments. With IPv6, it is easier to create service-oriented overlays and possible to create a level of persistency with these overlays that can't be created in an IPv4 context.

Currently the majority of the world's traffic flows is still over IPv4, though that is expected to change over time as carriers and enterprises IPv6 adoption is gaining popularity. The press is full of SDN start-ups that try to solve the networking issues of ISPs and Enterprise in optimizing their networks but none of them is deploying IPv6 especially after the IPv4 address space has exhausted. Even if SDN makes abstraction of the IP layer, it still depends on stable and routable IP addressing which is difficult to be achieved using IPv4. It's clear that SDN and IPv6 are both crucial for the long-term vision of the future network. IPv6 is necessary for end-to-end communication; SDN enables infrastructure flexibility and both play into network agility.

Our major contributions are summarized as follows:

- We define the IPv6 operations and deployment scenarios over SDN environment. This paper provides the basic guide for IPv6 SDN deployment.
- We demonstrate the IPv6 can coexist with IPv4 over SDN environment without impact the IPv4 performance, and the OpenFlow can provide a secure network environment for IPv6 end node needs.

The rest of the paper is organized as follows: in Section II, the background and related works are addressed. Section III is to describe the operations of the IPv6 over SDN, and Section IV presents the IPv6 deployment scenarios over SDN. Section V presents experiment results in a real enterprise environment. The last section concludes this paper and addresses potential future works in the future networks.

II. BACKGROUND AND RELATED WORKS

The current Internet has created more and more problems as it evolves with rapid development. Many of these problems need to be overhauled from the network architecture design. For this very reason, the deployment of a new architecture becomes imperative. The new connectivity scenarios encompass the need for end to end communication, which is revolutionizing the modern network revolution. The shortage of IPv4 addresses has created widespread use of private address spaces, which are not directly accessible from the

Internet. IPv6 offers the potential to build a much more powerful Internet with vastly larger scale compared with the current network. But the security and administrative personnel tasked with defending IPv4 networks have not kept pace with the growth of IPv6. Another issue with the current network is the difficulty of easy and scalable functional enhancement. Most device manufacturers develop network device software along with their proprietary hardware to perform operations appropriate for each communication. Besides these technical limitations, the requirements of new scenarios comprise aspects such as energy consumption, economic and social needs have not been taken into consideration. How to control and manage communication networks has become the most important task of the current network architecture [5], [6], [7].

[8] proposed a Software Defined Approach to Unified IPv6 Transition. The propose approach unifies the variety of IPv6 transition mechanisms. Combine the existing IPv6 protocol and the SDN to provide the smarter and reliable network communication architecture. [9] proposed an IPv6 Virtual Network Architecture (VNET6) to support flexible services in IPv6 network. IPv6 is a critical protocol in VNET6. The VNET6 is adaptive to video service with high bandwidth and low tendency and improves quality of experiences to users. [10] proposed a SDNv6 architecture, SDNv6 is an idealized service-oriented network architecture based on IPv6 and virtualization technologies. The SDNv6 motivates a network architecture composed of reliable virtual entities, plug-and-play access with auto-configure and flexible service clouds over a physical network.

Combining the IPv6 and SDN provides the features listed as follows:

- Scalability: IPv6 SDN can provide flexibility in how the network can be used and operated. Services developers, either users or operators will have more selections and freedom for their service creation development cycles without vendor lock-in increasing quicker time to market, minimizing the costly dependence on vendors.
- Operational Savings: IPv6 SDN can lower operating expenses with open source platform with easy configuration and lower network construction costs.
- Better management: IPv6 SDN architecture alters the outdated methods of manual configuration, offering speedy provision connections in a computerized manner through separation and abstraction. IPv6 SDN can simplify IP address management and provide a secure environment for end node needs.

III. IPv6 Operation Behaviors over SDN

A. *Configuration of IPv6 Host*

Most operating systems (e.g., Windows 7, Apple Mac and Linux OS) are already default IPv6 ready. To perform address configuration on IPv6 host over SDN environment, several methods, including: static addressing, dynamic addressing via StateLess Address AutoConfiguration (SLAAC) with DHCPv6 (Stateful), SLAAC with DHCPv6 (Stateless), or

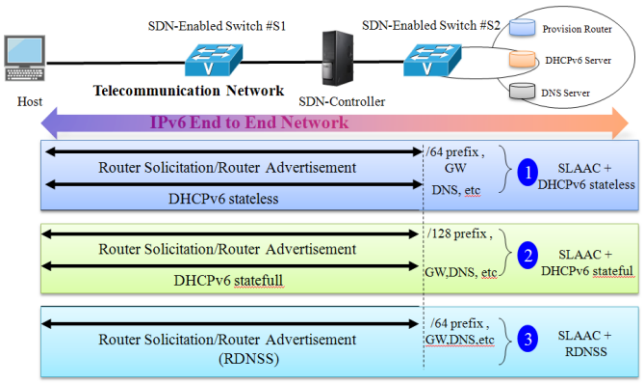

Fig. 1 Dynamic Autoconfiguration Mechanisms of IPv6 host over SDN

SLAAC Recursive DNS Server (RDNSS) are the current methods for this IPv6 address configuration management. IPv6 static addressing works exactly the same as IPv4 static addressing. Figure 1 shows the dynamic autoconfiguration mechanisms on IPv6 host in SDN environment. IPv6 SLAAC is implemented on the IPv6 client by listening for Router Advertisement (RA) and then taking the prefix that is advertised to form a unique address that can be used on the network. The details of IPv6 SLAAC mechanism in SDN is described as follows:

1) when the host uses SLAAC to access SDN, the host will send out RS (Router Solicitation)packet ;

2) SDN switch S1 will first match flow table using the information in RS packet. If no match is found, the RS packet will be forwarded to SDN controller using the OpenFlow packet-in message ;

3) SDN controller will recognize the network provision router and send out RS packet using OpenFlow packet-out message to S2 switch;

4) Upon receiving this packet, S2 will decode RS packet and forward this packet to network provision router that it is directly connected with;.

5) The network provision router will respond with a RA as soon as the router receives the RS packet.

6) S2 will send RA using the OpenFlow packet-in message to SDN controller ;

7) After processing the packet-in message, SDN controller will send RA using packet-out message to S1 and the S1 will forward this packet to the host that it is directly connected with;

8) As soon as the host receives RA, it will form IPv6 address using the prefix in this RA.

For extra data such as DNS server information, DHCPv6 implementation is required. Based on the IPv6 host networking requirements, DHCPv6 implementation is divided into DHCPv6 stateless and DHCPv6 stateful. The DHCPv6 Stateless and DHCPv6 Stateful are two existing IETF standard mechanisms defined to provide IPv6 networking information for end hosts. The behavior of DHCPv6 message exchange in the SDN environment is similar with the message exchange in SLAAC. If no match is found in the flow table entry for the received DHCPv6 packets , SDN-enabled switch will use

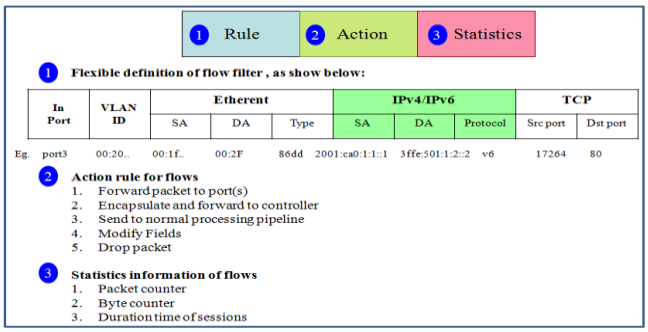

Fig. 2 An example of the flow table in OpenFlow switch

OpenFlow to exchange DHCPv6 message exchange with DHCPv6 server through SDN controller.

In SLAAC RDNSS, the DNS server information is comprised in the RA packet, with the SLAAC mechanism, the host can autoconfigure their internet connectivity parameters simultaneously. With these IPv6 dynamic autoconfiguration mechanisms, the address assignment and management become simpler in IPv6 SDN environment.

B. Behavior of the Packets Forwarding and Routing

Figure 2 is an example to depict the flow table in the OpenFlow switch. When a packet arrives at a switch in a conventional network, rules that are built into the switch's proprietary firmware will decide the where to forward the packet. A flow table entry is identified by its match fields and priority, the match fields and priority are used in combination to identify a unique flow entry in the flow table. For example, a switch may use the Eth type、IPv4/IPv6 address or input port of the packet to decide whether to send the packet using the output action policy. The IPv6-specific features that can be used is the flow label, the IPv6 packet header information can be used for flow tracking and for maintaining state across federated SDN domains.

For interoperability with the existing IP-based networks, SDN applies RouteFlow [11] to enable network to run Layer 3 routing interact with existing IP-based legacy network. Figure 3 illustrates the concept of RouteFlow architecture. RouteFlow is composed of an OpenFlow controller application, an independent RouteFlow server, and a virtual network environment that can clone the connectivity of a physical infrastructure and runs IP-based routing engines (e.g. Quagga).

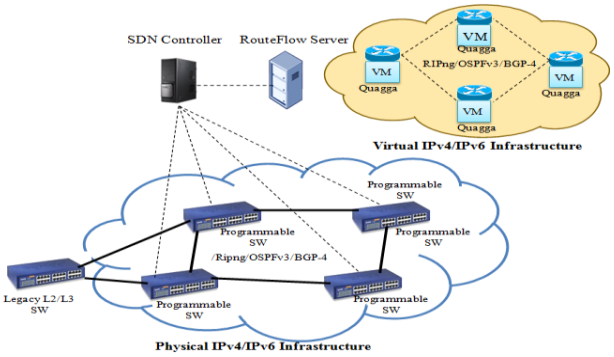

Fig. 3 RouteFlow Architecture

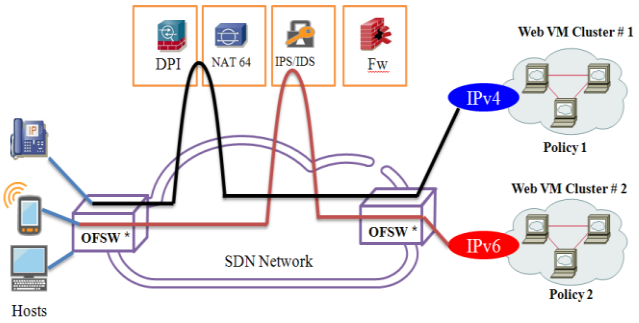

Fig. 4 Service Chaining for IPv6 Transition

C. Behavior of the Service Chaining

The concept of service chaining is built on SDN principles. A service chain simply consists of a set of network services, such as the authentication, authorization, transitions, security and other functions. These network functions can be implemented as part of a dynamic service chains where host's traffic flow is processed by various service functions thus avoiding the need for deploying different physical network elements. Figure 4 is an example of service chain for IPv6 transition. Due to the IPv4 address exhaustion, many operators have deployed the IPv6 transition technologies such as Network Address Translation (NAT). In this example, the service chaining can route the certain ipv6 host traffic through NAT64 service. The traffic traversing a NAT64 function may go through different types of IP address domain. As a result, packets processed by the service processing function are in IPv4. The OpenFlow switch in this scenario should be able to identify the flow and process the forwarding function based on the OpenFlow policy control.

To conclude, the dynamic service chaining enables network operators to provide differentiated and innovative services in their network based on their operating policies.

IV. IPv6 Deployment Scenarios over SDN

A. IPv6 SDN Broanband Network Scenario

The SDN impact on service providers has been rapid, global and significant. For service providers, this represents both a challenge and an opportunity. The current Broadband network is facing the difficulty of easy and scalable functional enhancement. Most device manufacturers must develop network device software along with their proprietary hardware to perform specific operations for each communication. The current Broadband architecture is also very expensive to deploy and maintain. Each service connects to a different device at home (e.g., PC, Notebook and TV) environment, often provisioned with a different IP address. Multiple IP service platforms quickly deplete capital budgets, and the service overlays they spawn result in complex provisioning and capacity planning in the access and aggregation networks. IPv6 migration is also an area of concern for service providers because legacy network devices (e.g., BRAS, Home Gateway and WiFi AP) lack the performance and are not capable of the carrier-grade network address translation (NAT) scaling to enable flexible IPv4-to-IPv6 migration.

The major drawbacks of the traditional Broadband Network are listed as follows:

- Lack of sufficient IPv4 and the performance and scale of a carrier-grade NAT.
- Limitations of proprietary software and hardware embedded in network elements.
- The existing IPv6 transition mechanisms require costly end-to-end network upgrades and managing a large number of devices with a variety of transitioning protocols.

In order to satisfy the stringent performance and scalability for IPv6 deployment and real time broadband services control, the IPv6 SDN can incorporate the service chaining functions for dynamic policy control and the IPv6 SLAAC mechanism for network host provision requirements.

Figure 5 illustrate the IPv6 SDN Broadband network architecture. In this case, multivendor network device configuration (e.g., IP-based Network device) can be controlled by standard interfaces (e.g., OpenFlow). With these programmable network SDN-enabled devices, the network operators can develop their own operations, administration and maintenance (OA&M) program, allowing customization and optimization, reducing the overall capital and operational costs. In the IPv6 SDN Broadband network scenario, the difference services can be partition into different slices. FlowVisor [12], an implementation of SDN, is a specialized controller using OpenFlow as a hardware abstraction layer between the control and forwarding paths. FlowVisor can create slices of network resources (e.g., data, voice and video) and delegates control of each slice to a different controller. The slices is a set of flows (called flowspace). When receiving a packet, the switch can decide which flowspace contains it, and hence which slice it belongs to. The orchestrator enables adaptive network resources allocation in IP layers based on the real time traffic to achieve efficient network resources control.

In the IPv6 SDN Broadband network scenario, the availability of globally IPv6 addresses for all interfaces simplifies SDN implementations. On the flip side, SDN can also make the transition to IPv6 simpler. In IPv6 SDN environment, operators can smoothly migrate services through service chains while making the management of IPv6 environments more deterministic.

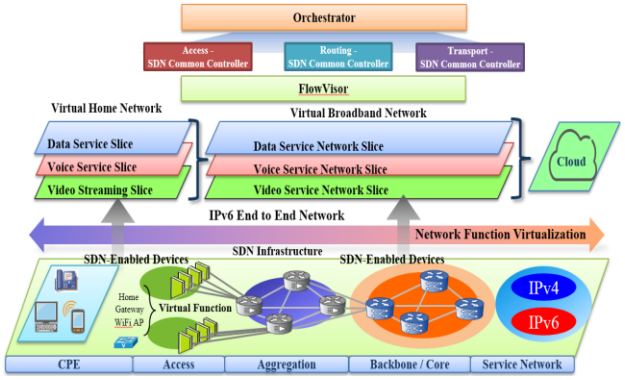

Fig. 5 IPv6 SDN Broadband Network Deployment

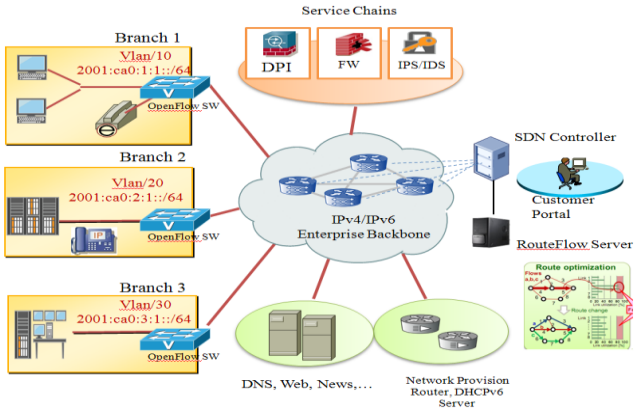

Fig. 6 IPv6 SDN Enterprise Campus Network Deployment

B. IPv6 SDN Enterprise Campus Network Scenario

The enterprise campus network is traditionally defined as a hierarchical model comprising the core, distribution, and access layers. Each element in the architecture has a specific set of functions or services that it offers and a specific role to play in each of the design. With the hierarchical design of the enterprise campus, the problems in one area of the network often impact the entire network. Simple add and move changes in one area have to be carefully planned to avoid affecting other parts of the network. Similarly, a failure in one part of the campus quite often affects the entire campus network. Besides, in the current enterprise network environment, most operating systems (e.g., Win 7 and Win8) are already default IPv6 ready, making IPv6 autoconfiguration easy to deploy while posing another big threat for IPv6 security that also needs to be addressed.

The major drawbacks of the traditional Enterprise Campus Network are listed as follows:

- Complicated host IP address and network topology maintenance and management that will increase network management operation cost.
- Simple and easy IPv6 autoconfiguration with network security challenges.

IPv6 SDN enterprise network architecture is proposed to solve the traditional enterprise network deployment issues. Figure 6 illustrate the IPv6 SDN Enterprise campus network. Every branch network in different enterprise network is connected with the IPv6 SDN core network using OpenFlow switch. With RouteFlow server design, SDN controller can support dynamic network configuration for quick enterprise network deployment. In this case, most of the enterprise network devices are IPv6 ready. Enterprise network can leverage this IPv6 support to enable IPv6 autoconfiguration function to connect to the Internet without manual configuration. For IPv6 security management, IPv6-specific flow label and secure service chains can be used. Any unauthorized IPv6 user will be detected and controlled using these mechanisms.

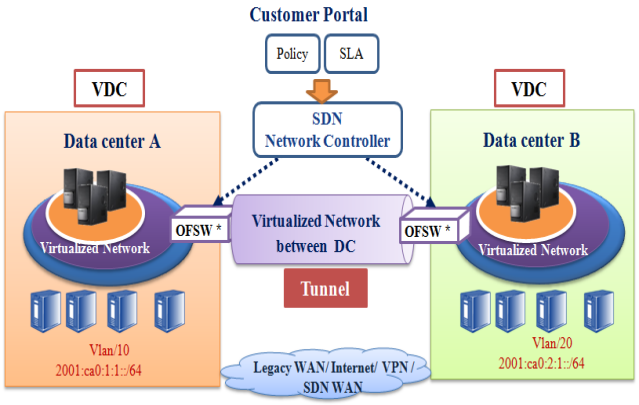

Fig. 7 IPv6 SDN Data Center Network Deployment

C. *IPv6 SDN Data Center Scenario*

As cloud business becomes more pervasive, distributed and large-scale data centers hosting a multitude of customers who must be kept securely segmented from each other will eventually become the standard. Besides, these data centers need to handle the very large number of addresses present on the internet. How to control and manage these Virtual Machines (VMs) has become the most important task of the current datacenter architecture.

The major drawbacks of the traditional data center are listed as follows:

- VMs IP address maintenance and network VLAN limits will become challenges as the data center scales, particularly when physical isolation is required or in public internet environments.
- As clouding computing grows or shrinks with customer's business, computing and storage resources need to be allocated with major operational overhead.

To overcome the traditional data center network deployment challenges, the IPv6 SDN data center architecture is proposed. In Figure 7, large numbers of virtual machines are expected in this virtual network domain. The virtual machines may be located on multiple data centers. This type of networking requires simple and robust autoconfiguration features. With the standard IPv6 feature, the network provisioning can automatically assign IPv6 addresses and perform device numbering to each virtual machine providing the interconnection between VMs. For performance and reliability issues, the traffic from the virtual machines needs to be inter-connected or aggregated over the network connections/tunnels that have been discovered and provisioned using IPv6. The OpenFlow switch can be leveraged to aggregate the VMs traffic over a selected set of network connections which provides minimal delay and bandwidth guarantees.

V. Experiment Results

To test the IPv6 over SDN environment, NEC's Programmable Flow Switch[13] is used in a real network environment. This switch is be able to provide the minimal IPv6 functionality, support match on IPv6 header fields, and execute actions on those fields (e.g., ethertype=0xffff86dd, IPv6add =2001:ca0:68:164::22/64). The experiment is running over IPv4/IPv6 dualstack environment. The IPv6 performance is measured and compared with IPv4. IxChariot 6.7[14] is used to evaluate end to end network transmission performance. We use script to generate IPv4 and IPv6 traffic on Microsoft Windows 7 simultaneously. This test script is designed with UDP and sends 1MB data from sender (endpoint in subnet1) to receiver (endpoint in subnet 2) continuously, then waits for the acknowledgment.

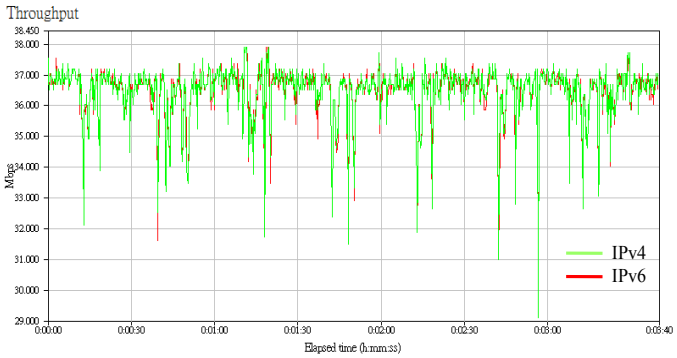

Fig. 8 Throughput comparison between IPv4 and IPv6 on the SDN environment

Figure 8 shows the throughput comparison of IPv4 and IPv6 experiment result. The result shows that throughput for IPv6 and IPv4 running over SDN environment have marginal difference and IPv6 will not impact the existing IPv4 network performance.

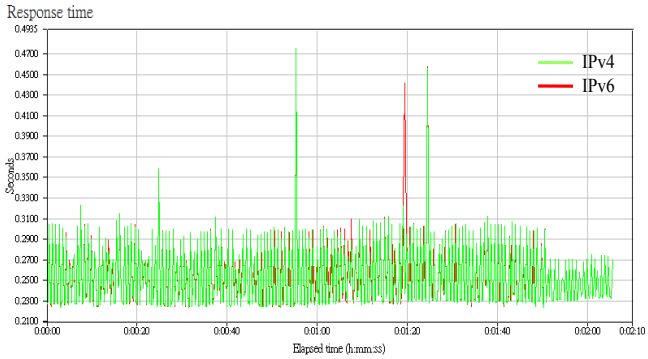

Fig. 9 Response Time comparison between IPv4 and IPv6 on the SDN environment

Figure 9 shows the response time comparison of IPv4 and IPv6 experiment result. The difference in response time of an IPv4 and an IPv6 is small. To verify OpenFlow security control function for IPv6 traffic, a whitelist policy is used. Using the OpenFlow whitelist policy, unauthorized IPv6 users who are attempting to access this network will be detected. When the whitelist policy is activated, IPv4/IPv6 packet will be transmitted normally. But after 1:50 when the whitelist policy is disabled, the IPv6 traffic will be blocked after the policy is modified.

Observations from the experiment results are listed as follows:

- IPv6 and IPv4 can coexist in SDN environment and enabling IPv6 will not impact the existing IPv4 traffic.
- OpenFlow-based SDN environment can secure the IPv6 deployment using whitelist mechanism.

Although the whitelist function evaluated in our experiment is very simple, but the concept of the whitelist function is important. For example, the rogue IPv6 router advertisement problem can be resolved by the illegal messages filter policy using OpenFlow to secure the IPv6 SLAAC scheme used in the SDN environment. The IPv6 DDOS attacks can be mitigated using traffic statistics gathered by OpenFlow source IP or port policies control. In addition to the above security functions mechanisms, service chains function is also the good solution to harden IPv6 security.

Conclusion and Future Works

IPv6 and SDN have emerged as a new paradigm of networking. In this paper we discuss the IPv6 operations and the deployment scenarios over SDN environment. IPv6 SDN can minimize many hardware restrictions and provide programmable network topologies to support variables services. The experiment results show the IPv6 and IPv4 can coexist over SDN environment, and the SDN technology can secure the IPv6 deployment.

Our future work will involve more experiments and analyses of these deployment scenarios. The impact of the IPv6 transition technology over SDN and study the capacity influenced by different network architecture will be also taken into consideration.

Acknowledgment

The work described in this paper was supported in part by Chunghwa Telecom Laboratories and National Science Council of the Republic of China. (Project No. NSC 100-2218-E-008-012-MY3 and NSC 102-2221-E-008-039-MY3)

References

[1] Myung-Ki Shin, Ki-Hyuk Nam, Hyoung-Jun Kim, “Software-defined networking (SDN): A reference architecture and open APIs,” ICT Convergence (ICTC), 2012 International Conference, Daejeon, South Korea, Oct. 2012.

[2] Staessens, D. Sharma, S. Colle, D. Pickavet, and M. Demeester, P. , “Software defined networking: Meeting carrier grade requirements,” Local & Metropolitan Area Networks (LANMAN), Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on, pp.1-5, Oct. 2012.

[3] Nunes, B. Mendonca, M. Bguyen, X. Obraczka, K. and Turletti, T., ”A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” IEEE Communications Surveys & Tutorials, pp.1-18, Jan. 2014.

[4] OpenFlow.[Online].Available: archive.openflow.org

[5] Sachin Sharma, Dimitri Staessens, Didier Colle, Mario Pickavet and Piet Demeester, ”Automatic Configuration of Routing Control Platforms in OpenFlow Networks,” Special Interest Group on Data Communication (SIGCOMM), 2013 International Conference, Hong Kong, China, August. 2013.

[6] L.-D. Chou, Y.-S. Chen, Y.-T. Yang, T.-C. Chang, C.-K. Shieh, S.-W. Huang, “Implementation of Virtual Network Management System with SLA on NetFPGA,” Proceeding of the 14th Asia – Pacific Network Operations and Management Symposium (Best Paper Award), Seoul, Korea, Sep. 2012. Project Number：NSC 100-2218-E-008-012-MY3

[7] L.-D. Chou, Y.-T. Yang, W.-P. Chang, T.-C. Chang, Y.-M. Hong, C.-K. Shieh, S.-W. Huang, “The Implementation of Multilayer Virtual Network Management System on NetFPGA,” Proceedings of the First International Workshop on Future Internet and Cloud Networking(FICN), Tainan, Taiwan, pp.988-991, 7-9, Dec. 2011. Project Number：NSC-100-2218-E-008-012-MY3

[8] Wenfeng Xia, Tina Tsou, Diego Lopez, ”A Software Defined Approach to Unified IPv6 Transition,” Special Interest Group on Data Communication (SIGCOMM), 2013 International Conference, Hong Kong, China, August. 2013.

[9] Xiaohan Liu, Gang Qin, Shuangjian Yan, Ze Luo, Baoping Yan, “VNET6: SOA Based on IPv6 Virtual Network,” Cloud and Service Computing (CSC), 2012 International Conference, pp.32-39, Nov. 2012.

[10] C. -W. Tseng, Y.-T. Yang, L.-D. Chou, “An IPv6-Enabled Software-Defined Networking Architecture,” Proceeding of the 15th Asia – Pacific Network Operations and Management Symposium, Hiroshima, Japan, Sep. 2013. Project Number：NSC 99-2221-E-008-041-MY3 and NSC 100-2218-E-008-012-MY3.

[11] M. Ribeiro Nascimento, C. Esteve Rothenberg, M. R. Salvador, C. Corrêa, S. Lucena and M. F. Magalhães., “Virtual Routers as a Service: The RouteFlow Approach Leveraging Software-Defined Networks,”. In 6th International Conference on Future Internet Technologies 2011 (CFI 11), Seoul, Korea, June 2011.

[12] Rob Sherwood, Glen Gibb, KK Yap, Guido Appenzeller, Nick McKeown, Guru Parulkar., ”FlowVisor: A Network Virtualization Layer,” DT Inc. R&D Lab, Standford University, Nicira Networks, Tecp. Rep. OPENFLOW-TR-2009-1, Oct. 2009.

[13] NEC ProgrammableFlow.[Oneline].Available: www.necam.com/SDN/

[14] IxChariot.[Online].Available: http://www.ixchariot.com