An Architecture of Programmable Network based on Adaptive SRv6 Policy

Hyunkyung Yoo, SungHyuk Byun, Sunhee Yang, and Namseok Ko
Mobile Core Network Research Section
Electronics and Telecommunications Research Institute(ETRI)
Daejeon, Republic of Korea
hkyoo@etri.re.kr, shbyun@etri.re.kr, shyang@etri.re.kr, nsko@etri.re.kr

***Abstract*—SRv6 is a source-based routing architecture in which a node steers packets by the service and topological requirements. Using programmable segments, SRv6 enables end-to-end service delivery to satisfy the network constraints of various services. In this paper, we present the architecture of programmable network in which the forwarding path is classified by the service flow and the adaptive path is decided according to the network load. We propose the adaptive SRv6 policy that dynamically selects the optimal segment lists among candidate path by monitoring the network performance. The adaptive SRv6 policy provides the flow-level path optimization that satisfies network QoS for each service flow.**

***Index Terms*—Segment Routing, SRv6, SRv6 policy, candidate-path, SID list, adaptive BSID, Network Programmability**

I. Introduction

With the expansion of technologies such as 5G, IoT, edge computing, and AI(Artificial Intelligence)/ML(Machine Learning), most applications require the network's high reliability and availability. Therefore, it is necessary to find the network resources and provide the optimal service paths in the network considering to the various QoS characteristics of applications. Network programming can steer packets by the QoS requirement through segments, that is, forwarding instruction set.

A representative network programming technology is segment routing(SR). IETF(Internet Engineering Task Force) SPRING (Source Packet Routing in Networking) working group standardizes the structures, the network programming, and the use cases of segment routing. The Segment Routing over IPv6 (SRv6) network programming enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header [1].

SRv6 features have been deployed in public, large-scale commercial networks, including Softbank, Rakuten, China Telecom, Iliad, LINE Corporation, China Unicom, CERNET2, China Bank and Alibaba [2]. Softbank and Rakuten have deployed a nationwide SRv6 network to offer 5G services to customers. China Telecom, Iliad, and NOIA Networks apply SRv6 to a large-scale IP backbone network. LINE Corporation and China Unicom introduce SRv6 to cloud data center. Alibaba uses SRv6 to provide the Predictable Network services. Meanwhile, domestic network vendors are looking for new SRv6 use cases and are defining the requirement to apply SRv6 in cloud data center. SRv6 is designed to use flexible segments according to service characteristics and network plans. It can provide traffic management and network monitoring using programmable path [3].

In this paper, we present the architecture of programmable network platform that classifies the forwarding path by the service flow and provides the SRv6 policy for adaptive path. It provides the dynamic path optimization by selecting the candidate path of SR policy by monitoring the network performance. The remainder of this paper is organized as follows. In Section II, we introduce the architecture of programmable network platform for providing programmable services according to the QoS characteristics of service flow and network load. Section III describes the adaptive SRv6 policy based on adaptive BSID. Finally, we summarize and conclude the paper in Section IV.

II. THE ARCHITECTURE OF PROGRAMMABLE NETWORK

SRv6 based platform can control the network path by selecting the optimal network resources according to the QoS characteristics of applications. We defined the SRv6 based Service Programmable Network (SPN) architecture [4]. We present the extension of SPN architecture for supporting programmable services in Internet Service Provider(ISP) network and Data Center network.

Figure 1 is an extended Programmable Network Platform architecture to support programmable P4 switch. Programmable Network Platform is composed of SPN applications, controller, and multiple nodes. The SPN nodes perform the segment routing in SRv6 service provider domain. They forward IPv4/IPv6 packets based on the path defined in the Segment Routing Header(SRH) [5]. And, they manage the SRv6 SID(segment Identifier) and network information. They are operated on SONiC (Software for Open Networking in the Cloud) OS [6] on a programmable P4 (Programming Protocol-Independent Packet Processors) switch [7].

The SPN controller is an extension of the open source ONOS (Open Network Operating System) SDN controller [8]. It is charge of the configuration and management of SPN nodes. It consists of SBI(Southbound Interface), Controller Core, Controller Application, and NBI(Northbound Interface) blocks. The SBI is an interface between a controller and

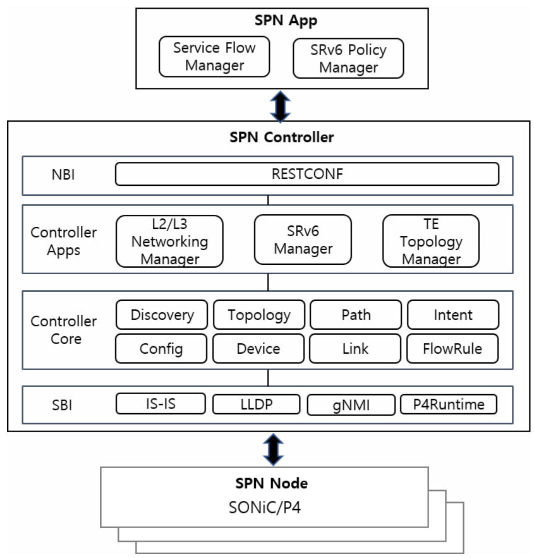

Fig. 1. Architecture of Programmable Network Platform

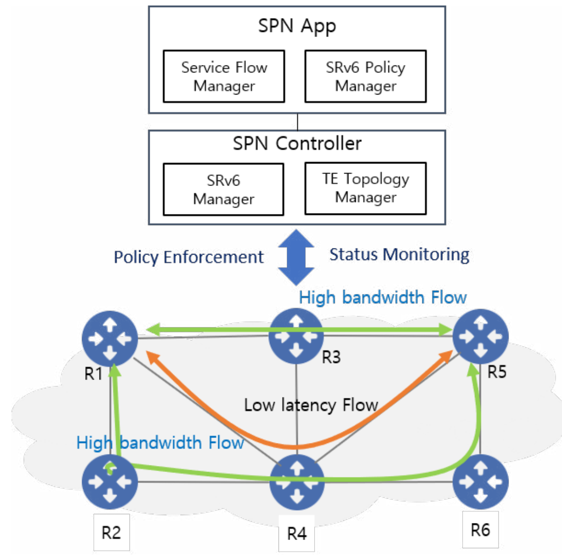

Fig. 2. Path Classification of Service Flow

SPN nodes. Through this interface, controller gathers network topology and manages forwarding tables of nodes. SBI protocols are defined IS-IS, LLDP, gNMI, and P4Runtime. IS-IS((Intermediate System to Intermediate System)) protocol is an Interior Gateway Protocol(IGP) that gathers the link state information. We use FRRouting as an IS-IS protocol. LLDP(Link Layer Discovery Protocol) is link discovery protocol used by network nodes and gNMI(gRPC Network Management Interface) is generic API to read and write configuration state of nodes. P4 is a language for programming the data plane of network nodes and P4Runtime is runtime control API for P4-defined data planes. NBI is an interface between a controller and SPN applications. We use a RESTCONF(Representational State Transfer Configuration Protocol).

Controller core is the core functionality set of controller. Controller Core functions are device, link, topology, flowrule, path, intent, config, and discovery. Device function manages the information of network devices/nodes. Link function manages the inventory of links. Topology function manages the network graph information of nodes, links, and ports. FlowRule function manages the match/action flow rules installed on nodes and provides flow metrics. Path function computes paths between nodes or between hosts using the topology graph. Intent function allows applications to specify their network control in form of policy. Config(Configuration) function configures the network topology information. Discovery function discovers the network topology of nodes, links, and hosts.

Controller applications consist of L2/L3 networking manager, SRv6 manager, and TE topology manager. They provide L2/L3 networking, SRv6 functions and topology management using core functions. The L2/L3 networking manager is designed for data center networking. It gathers the network topology with LLDP and provides IPv4/IPv6 Forwarding table for routing. SRv6 manager manages the SID information of each node and controls the policy of headend node. TE topology manager collects topology and TE(Traffic Engineering) information of nodes. It computes path using topology graph and TE information. Also, it recalculates the path in case of logical topology change, link failure, and adaptive BSID.

There are service flow manager and SRv6 policy manager in SPN applications. The service flow manager classifies the incoming service flow, and assigns the color by the service QoS requirement. It manages QoS information for flow type. SRv6 policy manager computes the forwarding paths between nodes or between hosts from global topology. And it generates and sends SRv6 policy that composed candidate paths to the headend node.

III. ADAPTIVE SRv6 POLICY

SRv6 policy is an ordered list of segments that represent a source-routed policy [9]. It is identified through the tuple ⟨headend, color, endpoint⟩. The headend is the node where the policy is implemented and the endpoint indicates the destination of the policy. The color is the value that associates the SRv6 policy with an intent or objective [10]. The headend uniquely identifies the SRv6 policy using color and endpoint, and steers the service flow.

Our platform offers the differentiated path considering the QoS requirement of incoming service flow. We define an adaptive SRv6 policy that selects the dynamic path by network load during packet forwarding. The process of adaptive SRv6 policy is as follows: classification of service flow, path computation, generation of SRv6 policy, and path re-computation for adaptive policy.

Figure 2 shows the concept to classify the path by the service flow type. When service flow manager application receives the service requests, it classifies the incoming service flow and assigns the color by the service QoS requirement.

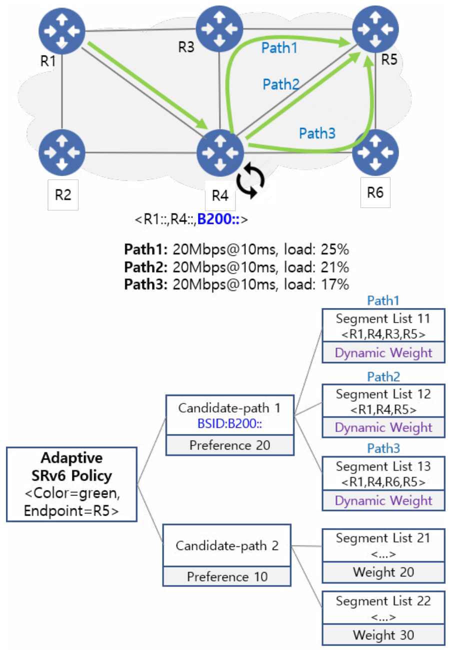

Fig. 3. Path Selection by Adaptive BSID

For example, we assign orange color to the low-latency services such as factory automation, automated driving, drone piloting, and remote surgery. And we assign green color to the video streaming and VR/AR services requiring high bandwidth. In addition, it can be classified by QoS requirement such as low cost, guaranteed security, and energy efficiency.

After classification of service flow, TE topology manager computes optimal candidate paths based on the recent topology graph and TE. The basic metrics of path computation are link delay, bandwidth, and load. For supporting a dynamic path by network load, we use TWAMP (Two-Way Active Measurement Protocol) for real-time measurement and statistics of link performance [11]. Depending on the service flow type, the weight of delay or bandwidth metric is applied differently. That is, low latency flow sets the minimum delay, and high bandwidth flow sets the maximum bandwidth. Among the computed candidate paths, SRv6 policy manager application defines and sends an active path to the headend node. The headend forwards it to the network.

One SRv6 policy has candidate paths and each candidate path has a weighted SID lists. The candidate path of the higher preference becomes the active path. And the SID list of a higher weight is selected as next segment. For adaptive SRv6 policy, we introduce the adaptive BSID, that is, extension of Binding Segment(BSID) that is replaced to SID lists in a specific node. In adaptive BSID, SID list has dynamic weight. When network node receives the adaptive BSID, it recalculates the real-time network load and selects the SID list dynamically.

Figure 3 shows the path selection by adaptive BSID based on dynamic network load. Its policy is ⟨color=green, endpoint=R5⟩. The candidate path1 has segment list 11, 12, 13 with dynamic weight. Each segment list is as follows:

- segment list 11 : ⟨R1::,R4::,R3::,R5::⟩
- segment list 12 : ⟨R1::,R4::,R5::⟩
- segment list 13 : ⟨R1::,R4::,R6::,R5::⟩

R1 is headend node that encapsulates SID lists such as ⟨R1::,R4::,B200::⟩in SRH. R1 node forwards this packet to R4 node. As R4 node receives the packet whose active segment is "B200::", recalculates the network load of segment list 11, 12, and 13 with dynamic weight. As a result, each network load is 25%, 21%, and 17%. Among them, R4 selects the segment list 13(path3) with least network load. It encodes ⟨R6::,R5::⟩as next SID and forwards it to R6. Using adaptive SRv6 policy, We can provide the best path according to the change of network state.

IV. Conclusion

We proposed an architecture of programmable network in which the forwarding path is classified by the service flow and the best path is decided according to the network load. We introduced the concept of adaptive SRv6 policy holding adaptive BSIDs with dynamic weights. It enables the load-balanced path optimization by dynamically selecting a segment list of SRv6 policy. Our programmable network platform can provide network programmability using the flexible segments of adaptive SRv6 policy. For further study, we plan to implement the adaptive SRv6 policy and develop useful services.

Acknowledgment

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [23ZR1100, A study of hyper-connected thinking Internet technology by autonomous connecting, controlling and evolving ways].

References

[1] C. Fisfils et al, "Segment Routing over IPv6 (SRv6) Network Programming," Internet Requests for Comments, RFC Editor, RFC8986, Feb. 2021.

[2] S. Matsushima et al, "SRv6 implementation and deployment status," IETF Internet-Draft, April. 2022.

[3] C. Fisfils et al, "Segment Routing Architecture," Internet Requests for Comments, RFC Editor, RFC8402, Jul. 2018.

[4] Yoo, Hyunkyung, et al, "A Service Programmable Network Architecture based on SRv6," IEEE 13th International Conference on Information and Communication Technology Convergence (ICTC), 2022.

[5] C. Fisfils et al, "IPv6 Segment Routing Header (SRH)," Internet Requests for Comments, RFC Editor, RFC8754, Mar. 2020.

[6] SONiC (Software for Open Networking in the Cloud), https://github.com/sonic-net/SONiC

[7] P4 (Programming Protocol-Independent Packet Processors), https://github.com/p4lang

[8] ONOS (Open Network Operating System), https://github.com/opennetworkinglab/onos

[9] Ventre, Pier Luigi, et al, "Segment routing: a comprehensive survey of research activities, standardization efforts, and implementation results," *IEEE Communications Surveys & Tutorials 23(1)*, pp. 182-221, 2020.

[10] C. Fisfils et al, "Segment Routing Policy Architecture," Internet Requests for Comments, RFC Editor, RFC9256, July. 2022.

[11] Rakesh Gandhi et al, "Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks," IETF Internet-Draft, Aug. 2023.