SPRING W. Mishima
Internet-Draft Y. Fukagawa
Intended status: Informational NTT Communications
Expires: 10 August 2026 6 February 2026

SRv6 SFC Architecture with SR-aware Functions
draft-watal-spring-srv6-sfc-sr-aware-functions-04

Abstract

This document describes the architecture of Segment Routing over IPv6 (SRv6) Service Function Chaining (SFC) with SR-aware functions. This architecture provides the following benefits:

* Comprehensive Management: a centralized controller for SFC, handling SR Policy, link-state, and network metrics.

* Simplicity: no SFC proxies, which reduces the number of nodes and address resource consumption.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Source Packet Routing in Networking Working Group mailing list (spring@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/spring/.

Source for this draft and an issue tracker can be found at https://github.com/watal.

Status of This Memo

Table of Contents

1. Introduction

Segment Routing over IPv6 (SRv6) [RFC8986] enables packet steering through a set of instructions called a segment list. Each SR segment endpoint node provides SRv6 Endpoint Behaviors, including Prefix/ Adjacency Segments, VPNs, and Binding Segments.

Service Function Chaining (SFC) [RFC7665] can be used in various scenarios (e.g. FW, IPS, IDS, NAT, and DPI). SFC based on Segment Routing (SR) is defined in [I-D.draft-ietf-spring-sr-service-programming], which describes some SRv6 Endpoint Behaviors, such as End.AS/AD/AM, are necessary for using SR-unaware functions.

This document describes an architecture for SRv6 SFC with SR-aware functions, which provides comprehensive management of SRv6 network resources and services.

2. Terminology

2.1. Terminology Defined in Related RFCs and Internet-Drafts

The following terms are used in this document as defined in the related RFCs and Internet-Drafts:

* SR, SR Domain, Segment ID (SID), SRv6, SR Policy, Prefix Segment, Adjacency Segment, Anycast Segment, Active Segment, and Distributed/Centralized/Hybrid Control Plane defined in [RFC8402].

* SR Source Node, Transit Node, and SR Segment Endpoint Node defined in [RFC8754].

* SRv6 Endpoint Behavior defined in [RFC8986].

* SFC, SFC Proxy, and Service Classification Function defined in [RFC7665].

* Service Segment, SR-Aware Service, SR-Unaware Service, End.AS, End.AD and End.AM defined in [I-D.draft-ietf-spring-sr-service-programming].

* Headend, Color, and Endpoint defined in [RFC9256].

* Quality of Service (QoS), Service Level Agreement (SLA), and Service Level Objective (SLO) defined in [RFC9522].

* Forwarding Plane, Control Plane, Management Plane, Application Plane defined in [RFC7426].

* Path Computation Client (PCC), Path Computation Element (PCE), and Traffic Engineering Database (TED) defined in [RFC5440].

* BGP Flow Specification defined in [RFC8955]

2.2. Newly Defined Terminology

The following terms are used in this document as defined below:

* Service Function Node: an SR segment endpoint node that provides SR-aware functions as service segments.

* SRv6 Controller: controls SRv6 Forwarding Plane, consisting of a PCE and a Classification Rule Controller.

* Classification Rule Controller: applies sets of SR Policy and flows to SR source nodes.

* Service Function Manager: configures network function instances, enables SR-aware functions as service segments, and collects network metrics.

2.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Design Objectives and Assumptions

3.1. Goals/Objectives

SRv6 SFC Architecture is designed with two main objectives:

* Comprehensive Management: a centralized controller for SFC, handling SR Policy, link-state, and network metrics. When providing SRv6 services, meeting SLAs for each customer is required. These SLAs consist of one or more SLOs such as availability, latency, and bandwidth. In an SRv6 SFC network, service segment provisioning, link-state collection, and SR Policy calculation are required to meet SLOs, respectively.

  [RFC8402] outlines a hybrid control plane that merges a distributed control plane and a centralized control plane. In this hybrid control plane, forwarding information like Node/Adjacency SIDs are advertised mutually by distributed SR nodes via

IGPs such as ISIS and OSPF, while other information like SR Policies, classification rules, and service segments are provided by a centralized controller and manager.

Software-Defined Networking (SDN) [RFC7426] provides centralized management of a network by a controller and a manager. Centralized management reduces operational costs through abstraction and automation. The SDN framework allows users to manage an SR domain without considering the details of a forwarding plane like a topology and node state. Operators can use an SRv6 controller to build SR Policies for SFC and QoS, manage the state of network functions, issue service segments automatically, and specify disaster recovery with protection.

* Simplicity: no SFC proxies, so that reduces nodes and address resource consumption. Network complexity increases operating costs. Generally, using a variety of protocols in a network raises operational costs, including designing, building, monitoring, and troubleshooting.

  Using an SFC proxy may increase forwarding overhead due to additional header manipulations.

3.2. Assumptions

To achieve these objectives, this architecture is based on two main assumptions:

* Straightforward extension of the SRv6 network programming model

  The protocol used in this architecture is compatible with SRv6. This streamlines the operation of services like traffic steering, including SFC, redundancy, and local protection. Standardized protocols such as BGP, PCEP, IS-IS, OSPF, TI-LFA, and Anycast SID are used in this architecture.

  This architecture is SRv6 compliant, enabling support for SR-unaware functions, although SR-aware functions are expected to meet the objective.

* SDN framework compliance and comprehensive management of SRv6 SFC by controllers

  A controller is used to provide comprehensive management. To simplify building and operating, the controller uses standardized protocols and abstracted service interfaces. This also provides programmability by controlling policies that meet a user’s intent including SFC and quality of service (QoS).

4. Overview of Architecture

Figure 1 illustrates an overview of this architecture.

```
+----------------------- Application Plane ----------------------+
|                        User Application                        |
+-----------------------------------|----------------------------+
                                    |
+- Control Plane (SRv6 Controller) -v-----+ +- Management Plane -+
| +--------------+ +--------------------+ | | +----------------+ |
| |Classification| |        Path        | | | |    Service     | |
| |     Rule     | |     Computation    | | | |    Function    | |
| |  Controller  | |    Element (PCE)   | | | |    Manager     | |
| +------|-------+ +-^-------|--------^-+ | | +----------------+ |
+--------|-----------|-------|--------|---+ +---------|----------+
         |           |       |        |               |
+--------|-----------|-------|--------|---------------|---+
| +------v-----------|-------v-+    +-|---------------v-+ |
| |                            |    |      Service      | |
| |       SR Source Node       |----|      Function     | |
| |                            |    |        Node       | |
| +----------------------------+    +-------------------+ |
+------------------- Forwarding Plane --------------------+
```

Figure 1: Overview of SRv6 SFC Architecture with SR-aware Functions

This architecture is based on [RFC7426] and consists of forwarding plane, control plane, management plane, and application plane.

* Forwarding Plane: classifies packets and encapsulates SRH, forwards them, and applies SRv6 Endpoint Behavior.

  - Provides SR-aware function using End.AN.

  - Classifies flows and applies them to a TE application with PBR.

  - Ensures redundancy with anycast.

  - Provides local protection with Fast Reroute (FRR).

* Control Plane: makes decisions about packet forwarding and provides rules for a forwarding plane.

  - Collects link-state including SRv6 locators, prefixes, behaviors, and delays.

  - Calculates and provisions SR Policies.

   - Applies SR Policies to each flow by provisioning flow classification rules.

* Management Plane: deploys and monitors network functions and devices.

   - Sets up network functions.

   - Collects metrics of devices, network functions, and SFC services.

* Application Plane: provides user API for the control/management planes.

   - Offers an interface for operators or customers.

   - Applies intents defined in [RFC9315].

Each component communicates using standardized protocols. These are designed to be loosely coupled and cooperate by using an abstraction layer.

This document suggests handling a control plane by application plane, but a detailed design of an application plane is out of the scope of this document. This is because application plane components and abstraction layers should be designed based on individual network utilization and operator intent. In the following sections, details of a forwarding plane, control plane, and management plane are explained.

5. Forwarding Plane

A forwarding plane provides SFC through packet classification, SRv6 encapsulation, and forwarding. In this architecture, all forwarding plane components are located within the SR domain.

```
 +------------------------------------------------------------------+
 | +-----------+             +----------+            +----------+ |
 | |           | SRv6 Packet | Service  | SRv6 Packet | Service  | |
 | | SR Source |(S2,S1; SL:1)| Function |(S2,S1; SL:1)| Function | |
-->|   Node    |------------>|   Node   |------------>|   Node   |-->
 | |           |             |   (S1)   |            |   (S2)   | |
 | +-----------+             +----------+            +----------+ |
 +-------------------------- SR domain ---------------------------+
```

Figure 2: Forwarding Plane

Figure 2 shows an example of SFC with two network functions. Firstly, the SR source node classifies the flow and encapsulates it with an SRH containing the segment list <S1, S2>. Next, the service function node (S1) receives the packet and applies a network function associated with an End.AN S1. Finally, the service function node (S2) receives the packet and also applies a network function associated End.AN S2, thus achieving SFC.

5.1. End.AN-based Service Segment Provisioning

End.AN provides an SR-aware function.

Functions with the same role MAY be assigned as the same service segment within the SR domain. By using Anycast SIDs, multiple nodes can be grouped as part of the same service segment.

End.AN MAY have optional arguments. This can provide additional programmability by embedding network function instructions in the segment list.

By using virtualized spaces within routers or on generic servers, network functions can be provided at any node in an SR domain. This allows for scaling and flexible redundancy of network functions.

5.1.1. When a Network Function Becomes Unavailable

When a network function becomes unavailable, the node removes the SID from its routing table. If an anycast SID is used, packets are redirected to another node. If no other nodes are available, the node drops the packets and sends an ICMP message (Type 3: Destination Unreachable, Code 0: Net Unreachable).

5.1.2. Anycast Segment

The concept of the Anycast Segment is introduced in [RFC8402]. In the SRv6 SFC, it realizes to provide the same network function segment as the same Anycast Segment. In such cases, the state between network functions MUST be shared mutually.

5.1.3. Fast Reroute

The ordering of network functions in an SRv6 SFC is guaranteed by the segment list, even if an FRR occurs, When an FRR occurs, if the Active segment is an Anycast SID, it MAY be forwarded to another service function node. In such a case, since state synchronization may not have been completed, the network function MUST have a mechanism to handle rerouted packets, such as buffering to wait for synchronization.

5.2. Service Function Chains

In this architecture, each SFC is represented as an SR Policy. The purpose or intent of each SR Policy can be identified using attributes such as color or name.

In general, SFC is achieved by using loose source routing. If both SFC and QoS are desired, they can be achieved by using strict source routing or loose source routing with Flex-Algo SIDs.

5.3. Per-Flow Encapsulation

In an SR source node, which serves as the Service Classification Function, packets are classified on a per-flow basis using PBR and encapsulated with SR Policy. Therefore, the SR source node MUST be capable of identifying packets using at least a 5-tuple or even more detailed information.

In this architecture, aiming for comprehensive management, the Service Classification Function has an API to communicate with the controller.

6. Control Plane

A control plane sets up a forwarding plane by creating SR policies, including SFCs, and applying them to each flow.

```
 +- Control Plane (SRv6 Controller) -------+
 | +--------------+ +--------------------+ |
 | |Classification| |        Path        | |
 | |     Rule     | |    Computation     | |
 | |  Controller  | |   Element (PCE)    | |
 | +------|-------+ +-^-------|--------^-+ |
 +--------|-----------|-------|--------|---+
  Classification link-state SR Policy link-state(Service Segment)
        Rule      (BGP-LS) (PCEP/BGP) (BGP-LS)
  (BGP Flowspec)      |       |        |
 +--------|-----------|-------|--------|-------------------+
 | +------v-----------|-------v-+    +-|-----------------+ |
 | |                            |    |      Service      | |
 | |       SR Source Node       |----|      Function     | |
 | |                            |    |        Node       | |
 | +----------------------------+    +-------------------+ |
 +------------------- Forwarding Plane --------------------+
```

Figure 3: Control Plane

The SRv6 Controller consists of the following two components:

* PCE: provides SR Policies that fulfill SFC/QoS requirements from the headend to the tailend and sends them to the SR source node.

* Classification Rule Controller: provides an Encapsulation Policy that corresponds to a specific flow and SR Policy, and sends them to the SR source node.

6.1. Path Computation Element (PCE)

PCE is a controller that provides SR Policy. As an Active Stateful PCE, it establishes sessions with all PEs in an SR domain and manages SFCs. SR Policies MUST support both explicit and dynamic paths.

For dynamic path computation, the Constrained Shortest Path First (CSPF) algorithm considers not only the SFC but also QoS constraints.

The PCE builds a Traffic Engineering Database (TED) of the SR domain using BGP-LS and installs SR policies via PCEP [RFC5440] or BGP SR Policy [I-D.draft-ietf-idr-segment-routing-te-policy].

To enable dynamic path calculation based on the state of service segments and Network Functions, the BGP-LS Service Segment extension [I-D.draft-ietf-idr-bgp-ls-sr-service-segments] is required.

6.2. Classification Rule Controller

A Classification Rule Controller determines flows to apply specific SFC.

The classification results are advertised to each SR source node as a set of flow, endpoints, and color with an extended protocol based on BGP Flowspec defined in [I-D.draft-ietf-idr-ts-flowspec-srv6-policy].

7. Management Plane

A management plane configures network function instances, enables SR-aware functions as service segments, monitors resources, and collects network metrics. The details of each manager are outside the scope of this document, as the southbound interface of the management plane may be different for each service and hardware architecture.

```
    +-------------------- Function Managers ---------------------+
    | +-----------+ +--------------+ +-----------+ +-----------+ |
    | |  Service  | | Virtualized  | |    VNF    | |  Network  | |
    | |  Function | |Infrastructure| |  Manager  | |  Metric   | |
    | |  Manager  | |   Manager    | |           | |  Manager  | |
    | +-----|-----+ +------^-------+ +-----|-----+ +-----^-----+ |
    +-------|--------------|---------------|-------------|-------+
            |              |               |             |
               Management Plane Southbound Interfaces
            |              |               |             |
    +-------|--------------|---------------|-------------|--------+
    | +-----v--------------v---------------v-------------|------+ |
    | |                  Service Function Node                  | |
    | +---------------------------------------------------------+ |
    +------------------------- SR domain -------------------------+
```

Figure 4: Management Plane

Figure 4 shows examples of managers that MAY be added to a management plane:

* Service Function Manager: provides an SID for a network service and manages this state.

* VNF Manager: handles deployment and scaling of network functions.

  - VNF Manager keeps links redundant and optimize link utilization.

* VIM: monitors hypervisor resources on service function nodes.

  - In SRv6 SFC, a hypervisor managed by a VIM MAY be located in virtualized spaces within routers or on generic servers.

* Network Metrics Manager: collects metrics for SR Policy calculation and evaluation.

  - Metrics are collected from multiple data sources, including IPFIX, TCP statistics, and SRv6 path tracing [I-D.draft-filsfils-spring-path-tracing].

  - Metrics can be used for PCE calculation parameters.

7.1. Service Function Manager

Service Function Manager enables and disables service segments of service function nodes.

The Manager advertises the following parameters to each service function node:

* Behavior: End.AN

* SID: the SID of End.AN (in IPv6 Address format). If service segments support slicing, they are represented as Flex-Algo SIDs.

* Function Name: type of network function

* Action: enable

* TLV:

  - Specification of the Anycast Group: when deploying multiple Network Functions within the same context, it MUST use the Anycast Group TLV to indicate a shared anycast group SID.

  - Allows for the specification of unique parameters and context associated with a particular network function.

8. Normative References

[I-D.draft-filsfils-spring-path-tracing]
Filsfils, C., Abdelsalam, A., Camarillo, P., Yufit, M., Graf, T., Su, Y., Matsushima, S., Valentine, M., and Dhamija, "Path Tracing in SRv6 networks", Work in Progress, Internet-Draft, draft-filsfils-spring-path-tracing-05, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-filsfils-spring-path-tracing-05>.

[I-D.draft-ietf-idr-bgp-ls-sr-service-segments]
Dawra, G., Filsfils, C., Talaulikar, K., Clad, F., Bernier, D., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Guichard, J., and C. Li, "BGP-LS Advertisement of Segment Routing Service Segments", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-ls-sr-service-segments-02, 5 November 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-ls-sr-service-segments-02>.

[I-D.draft-ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., and D. Jain, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-segment-routing-te-policy-26, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-segment-routing-te-policy-26>.

[I-D.draft-ietf-idr-ts-flowspec-srv6-policy]
Wenying, J., Liu, Y., Zhuang, S., Mishra, G. S., and S. Chen, "Traffic Steering using BGP FlowSpec with SR Policy", Work in Progress, Internet-Draft, draft-ietf-idr-ts-flowspec-srv6-policy-08, 1 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-ts-flowspec-srv6-policy-08>.

[I-D.draft-ietf-spring-sr-service-programming]
Abdelsalam, A., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-service-programming-12, 3 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-12>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/rfc/rfc5440>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/rfc/rfc7426>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/rfc/rfc7665>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/rfc/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/rfc/rfc8754>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/rfc/rfc8955>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/rfc/rfc8986>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/rfc/rfc9256>.

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/rfc/rfc9315>.

[RFC9522] Farrel, A., Ed., "Overview and Principles of Internet Traffic Engineering", RFC 9522, DOI 10.17487/RFC9522, January 2024, <https://www.rfc-editor.org/rfc/rfc9522>.

Appendix A. Implementation Experience

Note to the RFC Editor: Please remove this appendix before publication as an RFC.

This appendix is informative and non-normative.

Several components of the architecture have been implemented and validated in IETF Hackathons.

A.1. Flow-based Service Classification

The following functionality has been implemented in GoBGP:

* Advertisement of classification rules using BGP Flowspec extensions for SRv6 Policy.

A.2. SR Policy Computation and Provisioning

The following functionality has been implemented in Pola PCE:

* Centralized SR Policy computation for SFC using a stateful PCE.

* Support for explicit and dynamic SRv6 Policies with loose source routing.

A.3. Service Segment Advertisement

The following functionality has been implemented in GoBGP and ExaBGP:

* Advertisement of VNF and End.AN information using BGP-LS Service Segment extensions.

A.4. Service and Infrastructure Management

The following functionality has been implemented using OpenStack and Ansible:

* Deployment of VNFs.

* Provisioning of SRv6 End.AN behaviors.

Acknowledgments

The authors would like to acknowledge the review and inputs from Mitsuru Maruyama, Katsuhiro Sebayashi, Yuma Ito, and Taisei Tanabe. We partially obtained the research results from NICT’s commissioned research No.JPJ012368C03101 and JST’s CRONOS No.JPMJCS24N9.

Authors’ Addresses

Wataru Mishima
NTT Communications
Japan
Email: w.mishima@ntt.com

Yuta Fukagawa
NTT Communications
Japan
Email: y.fukagawa@ntt.com