

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2021

Z. Ali
C. Filsfils
P. Camarillo
Cisco Systems
D. Voyer
Bell Canada
S. Matsushima
Softbank
February 21, 2021

**Building blocks for Slicing in Segment Routing Network
draft-ali-spring-network-slicing-building-blocks-04.txt**

Abstract

This document describes how to build network slice using the Segment Routing based technology. It explains how the building blocks specified for the Segment Routing can be used for this purpose.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 21, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction.....	2
2	Segment Routing Policy.....	3
2.1	Flex-Algorithm Based SR Policies	4
2.2	On-demand SR policy	5
2.3	Automatic Steering	6
2.4	Inter-domain Considerations	6
3	TI-LFA and Microloop Avoidance.....	7
4	SR VPN.....	7
5	Stateless Service Programming.....	7
6	Operations, Administration, and Maintenance (OAM).....	8
7	QoS.....	8
8	Stateless Network Slice Identification.....	9
8.1	Stateless Slice Identification in SRv6.....	9
8.2	Stateless Slice Identification in SR-MPLS.....	10
8	Orchestration at the Controller.....	10
9	Illustration.....	10
10	Security Considerations	10
11	IANA Considerations	11
12	References	11
12.1	Normative References	11
7.2	11
13	Acknowledgments	11
14	Contributors	11

1 Introduction

As more and more Service Providers and Enterprises operate a single network infrastructure to support an ever-increasing number of services, the ability to custom fit transport to application needs is critically important. This includes creating network slices with different characteristics can coexist on top of the shared network infrastructure.

Network Slicing is meant to create (end-to-end) partitioned network infrastructure that can be used to provide differentiated connectivity behaviors to fulfill the requirements of a diverse set of services. Services belonging to different Network slices can be wholly disjoint or can share different parts of the network infrastructure. Network Slicing is one of the requirements in 5G [3GPP 23501].

Segment Routing enables Service Providers to support Network Slicing without any additional protocol, other than the SR IGP extensions. The network as a whole, in a distributed and entirely automated manner, can share a single infrastructure resource along multiple virtual services (slices). For example,

one slice is optimized continuously for low-cost transport; a second Slice is optimized continuously for low-latency transport; a third Slice is orchestrated to support disjoint

Internet-Draft Network Slicing Using SR

services, etc. The optimization objective of each of these slices is programmable by the operator.

The Segment Routing specification already contains the various building blocks required to create network slices. This includes the following.

- . SR Policy with or without Flexible Algorithm.
- . TI-LFA with O(50 msec) protection in the slice underlay.
- . SR VPN.
- . SR Service Programming (NFV, SFC).
- . Operation, Administration and Management (OAM) and Performance Management (PM).
- . QoS using DiffServ.
- . Stateless Network Slice Identification
- . Orchestration at the Controller.

Each of these building blocks works independently of each other. Their functionality can be combined to satisfy service provider's requirement for the Network Slicing. An external controller plays an important role to orchestrate these building blocks into a Slicing service.

This document elaborates on the attributes of each of these building blocks for Network Slicing. The document also highlights how services in each Slice can benefit from traffic engineering, network function virtualization/ service chaining (service programming), OAM, performance management, SDN readiness, O (50 msec) TI-LFA protection, etc. features of SR while respecting resource partitioning employed over the common networking infrastructure.

The document equally applicable to the MPLS and SRv6 instantiations of segment routing.

The following subsection elaborates on each of these build blocks.

2 Segment Routing Policy

Segment Routing (SR) allows a headend node to steer a packet flow along any path without creating intermediate per-flow states [[I-D.ietf-spring-segment-routing-policy](#)]. The headend node steers a flow into a Segment Routing Policy (SR Policy).

I.e., the SR Policy can be used to steer traffic along any arbitrary path in the network. This allows operators to enforce low-latency and / or disjoint paths, regardless of the normal forwarding paths.

Internet-Draft Network Slicing Using SR

The SR policy is able to support various optimization objectives [I-D.[draft-filsfils-spring-sr-policy-considerations](#)]. The optimization objectives can be instantiated for the IGP metric ([[RFC1195](#)] [[RFC2328](#)] [[RFC5340](#)]) xor the TE metric ([[RFC5305](#)], [[RFC3630](#)]) xor the latency extended TE metric ([[RFC7810](#)] [[RFC7471](#)]). In addition, an SR policy is able to various constraints, including inclusion and/or exclusion of TE affinity, inclusion and/or exclusion of IP address, inclusion and/or exclusion of SRLG, inclusion and/or exclusion of admin-tag, maximum accumulated metric (IGP, TE, and latency), maximum number of SIDs in the solution SID-List, maximum number of weighted SID-Lists in the solution set, diversity to another service instance (e.g., link, node, or SRLG disjoint paths originating from different head-ends), etc. [I-D.[draft-filsfils-spring-sr-policy-considerations](#)]. The supports for various optimization objectives and constraints enables SR policy to create Slices in the network.

SR policy can be instantiated with or without IGP Flexible Algorithm feature. The following subsection describes the SR Flexible Algorithm feature and how SR policy can utilize this feature.

2.1 Flex-Algorithm

Flexible Algorithm enriches the SR Policy solution by adding additional segments having different properties than the IGP Prefix segments. Flex Algo adds flexible, user-defined segments to the SRTE toolbox. Specifically, it allows for association of the "intent" to Prefix SIDs. [I-D.[ietf-lsr-flex-algo](#)] defines the IGP based Flex-Algorithm solution which allows IGPs themselves to compute paths constraint by the "intent" represented by the Flex-Algorithm.

The Flex-Algorithm has the following attributes:

- . Algorithm associate to the SID a specific TE intent expressed as an optimization objective (an algorithm) [I-D.[ietf-lsr-flex-algo](#)].
- . Flexibility includes the ability of network operators to define the intent of each algorithm they implement.

- . By design the mapping between the Flex-Algorithm and its meaning is flexible and is defined by the user.
- . Flexibility also includes ability for operators to make the decision to exclude some specific links from the shortest path computation, e.g.,

Internet-Draft Network Slicing Using SR

- o operator 1 may define Algo 128 to compute the shortest path for TE metric and exclude red affinity links.

- o operator 2 may define Algo 128 to compute the shortest path for latency metric and exclude blue affinity links.

A Network Slice can be created by associating a Flexible-Algorithm value with the Slice via provisioning.

Flex Alg leverages SR on-demand next hop (ODN) and Automated Steering for intent-based instantiation of traffic engineered paths described in the following sub-sections. Specifically, as specified in [\[RFC8402\]](#) the IGP Flex Algo Prefix SIDs can also be used as segments within SR Policies thereby leveraging the underlying IGP Flex Algo solution.

2.2 On-demand SR policy

Segment Routing On-Demand Next-hop (ODN) functionality enables on-demand creation of SR Policies for service traffic. Using a Path Computation Element (PCE), end-to-end SR Policy paths can be computed to provide end-to-end Segment Routing connectivity, even in multi-domain networks running with or without IGP Flexible-Algorithm [I-D.[draft-ietf-spring-segment-routing-policy](#)].

The On-Demand Next-hop functionality provides optimized service paths to meet customer and application SLAs (such as latency, disjointness) without any pre-configured TE tunnel and with the automatic steering of the service traffic on the SR Policy without a static route, autoroute-announce, or policy-based routing.

With this functionality, a Network Service Orchestrator can deploy the service based on their requirements. The service head-end router requests the PCE to compute the path for the service and then instantiates an SR Policy with the computed path and steers the service traffic into that SR Policy. If the topology changes, the stateful PCE updates the SR Policy path. This happens seamlessly, while TI-LFA protects the traffic in case the topology change happened due to a failure.

2.3 Automatic Steering

Automatically steering traffic into a Network Slice is one of the fundamental requirement for Slicing. That is made possible by the "Automated Steering" functionality of SR. Specifically, SR policy can be used for traffic engineer paths within a slice, "automatically steer" traffic to the right slice and connect IGP Flex-Algorithm domains sharing the same "intent".

A headend can steer a packet flow into a valid SR Policy within a slice in various ways [I-D.[draft-ietf-spring-segment-routing-policy](#)]:

- . Incoming packets have an active SID matching a local Binding SID (BSID) at the headend.
- . Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.
- . Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- . Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

2.4 Inter-domain Considerations

The network slicing needs to be extended across multiple domains such that each domain can satisfy the intent consistently. SR has native inter-domain mechanisms, e.g., SR policies are designed to span multiple domains using a PCE based solution [I-D.ietf-spring-segment-routing], [I-D.ietf-spring-segment-routing-central-epe]. An edge router upon service configuration automatically requests to the Segment Routing PCE an inter-domain path to the remote service endpoint. The path can either be for simple best-effort inter-domain reachability or for reachability with an SLA contract and can be restricted to a Network Slice.

The SR native mechanisms for inter-domain are easily extendable to include the case when different IGP Flex-Algorithm values are used to represent the same intent. E.g., in domain1 Service Provider 1 (SP1) may use flex-algo 128 to indicate low latency Slice and in domain2 Service Provider 2 (SP2) may use flex-algo 129 to indicate low latency Slice. When an automation system at a PE1 in SP1 network configures a service with next hop (PE2) in SP2 network, SP1 contacts a Path Computation Element (PCE) to

find a route to PE2. In the request, the PE1 also indicates the intent (i.e., the Flex-Algo 128) in the PCEP message. As the PCE has a complete understanding of both Domains, it can understand the path computation in Domain1 needs to be performed for Algorithm 128 and path computation in Domain2 needs to be

Internet-Draft Network Slicing Using SR

performed for Algorithm 129 (i.e., in the Low Latency Network Slice in both domains).

3 TI-LFA and Microloop Avoidance

The Segment Routing-based fast-reroute solution, TI-LFA, can provide per-destination sub-50msec protection upon any single link, node or SRLG failure regardless of the topology. The traffic is rerouted straight to the post-convergence path, hence avoiding any intermediate flap via an intermediate path. The primary and backup path computation is completely automatic by the IGP.

[I-D.[draft-bashandy-rtgwg-segment-routing-ti-lfa](#)] proposes a Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at protecting node and adjacency segments within 0(50 msec) in the Segment Routing networks. Furthermore, [I-D. [draft-bashandy-rtgwg-segment-routing-uloop](#)] provides a mechanism leveraging Segment Routing to ensure loop-freeness during the IGP reconvergence process following a link-state change event.

As mentioned earlier, Network Slicing in Segment Routing works seamlessly with all the other components of the Segment Routing. This, of course, includes TI-LFA and microloop avoidance within a Slice, with the added benefit that backup path only uses resources available to the Slice. For example, when Flexible Algorithm is used, the TI-LFA backup path computation is performed such that it is optimized per Flexible-Algorithm. The backup path shares the same properties as the primary path. The backup path does not use a resource outside the Slice of the primary path it is protecting.

4 SR VPN

Virtual Private Networks (VPNs) provides a mean for creating a logically separated network to a different set of users access to a common network. Segment Routing is equipped with the rich multi-service virtual private network (VPN) capabilities, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN

(EVPN). The ability of Segment Routing to support different VPN technologies is one of the fundamental building blocks for creating slicing an SR network.

Internet-Draft Network Slicing Using SR

5 Stateless Service Programming

An important part of the Network Slicing is the orchestration of virtualized service containers. [I-D.[draft-xuclad-spring-sr-service-chaining](#)] describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. It introduces the notion of service segments. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of the service, running either on a physical appliance or in a virtual environment, is associated with a segment identifier (SID) for the service. These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined with topological SIDs to achieve service programming while steering the traffic through a specific topological path in the network. In this fashion, SR provides a fully integrated solution for overlay, underlay and service programming building blocks needed to satisfy network slicing requirements.

6 Operations, Administration, and Maintenance (OAM)

There are various OAM elements that are critical to satisfy Network Slicing requirements. These includes but not limited to the following:

- . Measuring per-link TE Matric.
- . Flooding per-link TE Matric.
- . Taking TE Matric into account during path calculation.
- . Taking TE Matric bound into account during path calculation.
- . SLA Monitoring: Service Provider can monitor each SR Policy in a Slice to Monitor SLA offered by the Policy using technique described in [I-D.[draft-gandhi-spring-udp-pm](#)]. This includes monitoring end-to-end delays on all ECMP paths of the Policy as well as monitoring traffic loss on a Policy. Remedial mechanisms can be used to ensure that the SR policy conforms to the SLA contract.

7 QoS

Segment Routing relies on MPLS and IP Differentiated Services. Differentiated services enhancements are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. [[RFC2475](#)] defines

Internet-Draft Network Slicing Using SR

an architecture for implementing scalable service differentiation in the Internet. This architecture is composed of many functional elements implemented in network nodes, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions including metering, marking, shaping, and policing.

The DiffServ architecture achieves scalability by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviors to aggregates of traffic depending on the traffic marker. Specifically, the node at the ingress of the DiffServ domain conditions, classifies and marks the traffic into a limited number of traffic classes. The function is used to ensure that the slice's traffic conforms to the contract associated with the slice.

Per-hop behaviors are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams. Specifically, per class scheduling and queuing control mechanisms are applied at each IP hop to the traffic classes depending on packet's marking. Techniques such as queue management and a variety of scheduling mechanisms are used to get the required packet behavior to meet the slice's SLA.

8 Stateless Network Slice Identification

Some use-cases require a slice identifier (SLID) in the packet to provide differentiated treatment of the packets belonging to different network slices.

The network slice instantiation using the SLID in the packet is required to work with the building blocks described in the previous sections. For example, the QoS/ DiffServ needs to be observed on a per slice basis. The slice identification needs to be topologically independent and stateless.

8.1 Stateless Slice Identification in SRv6

[I-D.[draft-filsfils-spring-srv6-stateless-slice-id](#)] describes a stateless encoding of slice identification in the outer IPv6 header of an SRv6 domain. As defined in [RFC8754](#) [[RFC8754](#)], when an ingress PE receives a packet that traverses the SR domain, it encapsulates the packet in an outer IPv6 header and an optional SRH. Based on a local policy of the SR domain, the Flow Label field of the outer IPv6 header carries the SLID. Specifically, the SLID is added in the 8 most significant bits of the Flow Label field of the outer IPv6 header. The remaining 12 bits of the Flow Label field are set as described in [section 5.5 of \[RFC8754\]](#) for inter-domain packets. Based on the local policy of the SR domain, the draft also uses one of the bits in the Traffic Class field of the outer IPv6 header to indicate that the entropy label contains the SLID.

The network slicing mechanism described in [I-D.[draft-filsfils-spring-srv6-stateless-slice-id](#)] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SRv6 overlays.

8.2 Stateless Slice Identification in SR-MPLS

[I-D.[draft-decraene-mpls-slid-encoded-entropy-label-id](#)] describes a similar stateless encoding of slice identification in the SR-MPLS domain. Specifically, the document extends the use of the Entropy Label to carry the SLID. The number of bits to be used for encoding the SLID in the Entropy Label is governed by a local policy of the SR domain. Based on the local policy of the SR domain, the draft uses one of the bits in the TTL field of the Entropy Label to indicate that the Entropy Label contains the SLID.

The network slicing mechanism described in [I-D.[draft-decraene-mpls-slid-encoded-entropy-label-id](#)] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SR-MPLS overlays.

A controller plays a vital role in orchestrating the SR building blocks discussed above to create Network Slices. The controller also performs admission control and traffic placement for slice management at the transport layer. The SDN friendliness of the SR technology becomes handy to realize the orchestration. The controller may use PCEP or Netconf to interact with the routers. The router implements Yang model for SR-based network slicing.

Specification of the controller technology for orchestrating Network Slices, services and admission control for the services is outside the scope of this draft.

9 Illustration

To be added in a later revision.

10 Security Considerations

This document does not impose any additional security challenges.

Internet-Draft Network Slicing Using SR

11 IANA Considerations

This document does not define any new protocol or any extension to an existing protocol.

12 References

12.1 Normative References

[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/info/rfc2119>>.

7.2. Informative References

[I-D.ietf-spring-segment-routing-policy] Filss, C., Sivabalan, et al, "Segment Routing Policy For Traffic Engineering", [draft-ietf-spring-segment-routing-policy](#) (work in progress).

[I-D.ietf-lsr-flex-algo] P. Psenak, et al, [draft-ietf-lsr-flex-algo](#), work in progress.

[RFC8402]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC8402](#).

[I-D.[draft-filsfils-spring-sr-policy-considerations](#)]

Filsfils, C., et al. [draft-filsfils-spring-sr-policy-considerations](#) (work in progress)

[RFC8754]

Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-16](#) (work in progress), February 2019.

[I-D.[draft-filsfils-spring-srv6-stateless-slice-id](#)] Filsfils, C., et al. [draft-filsfils-spring-srv6-stateless-slice-id](#), work in progress.

[I-D.[draft-decraene-mpls-slid-encoded-entropy-label-id](#)] Decraene B., Filsfils, C., Henderickx W., Saad T., Beeram V., work in progress.

13 Acknowledgments

14 Contributors

Francois Clad
Cisco Systems, Inc.
fclad@cisco.com

Internet-Draft Network Slicing Using SR

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Email: cf@cisco.com

Pablo Camarillo Garvia
Cisco Systems, Inc.
Email: pcamaril@cisco.com

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca