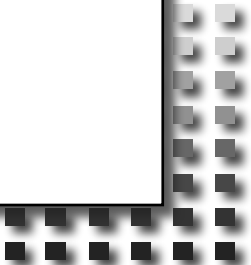




IETF Hackathon - BGP-LS Extensions for SRv6 Service Chaining

**IETF 121
2–3 November 2024
Dublin, Ireland
Yuta Fukagawa**



Hackathon Plan

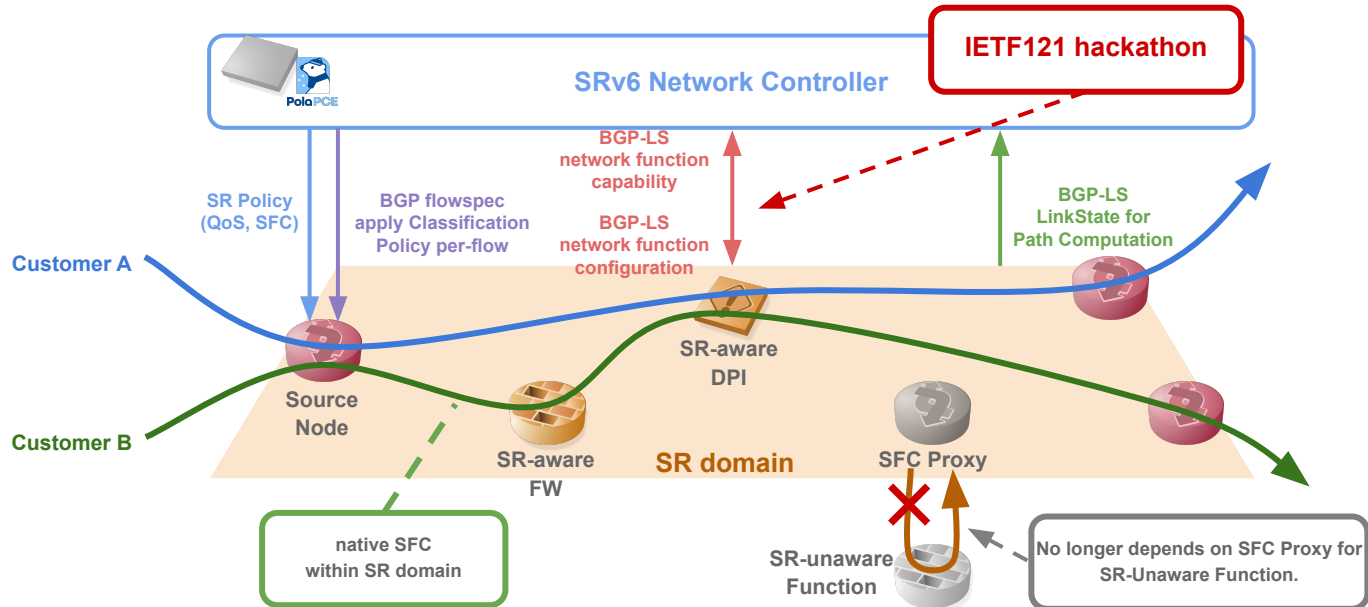
Objective: enable the controller to recognize, configure, activate/deactivate Service Functions on SRv6 nodes.

- **BGP-LS extensions for SRv6 SFC Architecture**
 - **BGP-LS Advertisement of Segment Routing Service Segments**
([draft-ietf-idr-bgp-ls-sr-service-segments](#))
 - **Implementation for Controlling Network Function State via BGP-LS**
(The I-D for this is not written yet; it will be developed later and is not included in this Hackathon.)
- **Hackathon Goal: Implement BGP-LS extensions in the following OSS.**
 - Implement TLVs to [GoBGP](#), an BGP daemon written in Go Language
 - Implement additional [Wireshark](#) dissectors

SRv6 SFC Architecture

The concept that comprehensive controller manages the entire SRv6 SFC network.

The controller collects information about Service Segments and apply SR Policy based on the information about available Network Functions.



BGP Service Segment

Using Service Segment, SRv6 controller will be able to get Network Function status for service-based Traffic Engineering.

I-D: <https://datatracker.ietf.org/doc/draft-ietf-idr-bgp-ls-sr-service-segments/>

Type (2 octet)
Length (2 octet)
Service Type (ST) (2 octet)
Flags (1 octet)
Traffic Type (1 octet)
RESERVED (2 octet)

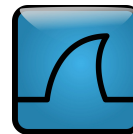
Service Chaining (SC) TLV

Type (2 octet)
Length (2 octet)
Opaque Type (2 octet)
Flags (1 octet)
Value (variable)

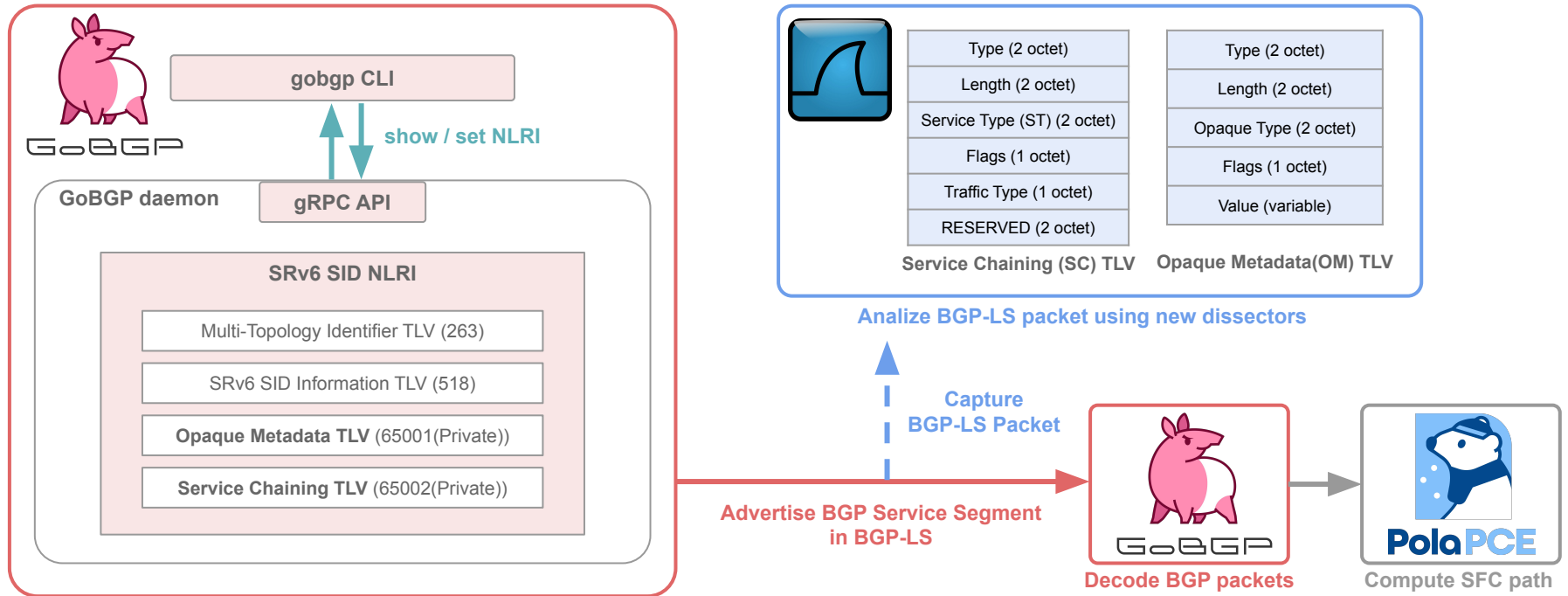
Opaque Metadata(OM) TLV

Implementations (1/2)

- **GoBGP** (<https://github.com/osrg/gobgp>)
 - OSS BGP daemon written in Go
 - Add support related in BGP-LS SRv6 Architecture
 - **RFC 9514**: SRv6 SID NLRI, SRv6 SID Information TLV, Multi Topology TLV
 - **draft-ietf-idr-bgp-ls-sr-service-segments**: Service Chaining TLV, Opaque Metadata TLV
- **WireShark** (<https://www.wireshark.org/>)
 - OSS network protocol analyzer
 - Add dissector for BGP-LS Extension in this hackathon
 - **draft-ietf-idr-bgp-ls-sr-service-segments**: Service Chaining TLV, Opaque Metadata TLV
- **Pola PCE** (<https://nttcom.github.io/pola/>)
 - OSS Path Computation Element
 - written by our team (NTT Com)



Implementations (2/2)



What got done

- **GoBGP**
 - **COMPLETED: SC TLV**
 - <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-ls-sr-service-segments-02#section-2>
 - **COMPLETED: OM TLV**
 - <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-ls-sr-service-segments-02#section-2>
 - **COMPLETED: SRv6 SID NLRI**
 - <https://datatracker.ietf.org/doc/html/rfc9514#name-srv6-sid-nlri>
 - **WIP: GET / POST NLRI via gRPC API**
 - Interoperability test was succeeded with following Wireshark dissector
- **Wireshark**
 - **WIP: The dissector for parsing SC TLV and OM TLV has been completed.**

Wrap-up / Our Code

- **GoBGP**: <https://github.com/yfskyline/gobgp>
- **Wireshark Dissector**: <https://gitlab.com/yfskyline/wireshark>

Members

- **Yuta Fukagawa**, Kanagawa Institute of Technology, fukagawa@nw.kanagawa-it.ac.jp
- **Wataru Mishima**, Kanagawa Institute of Technology, mishima@nw.kanagawa-it.ac.jp
- **Taisei Tanabe**, National Institute of Informatics, tanabe@nii.ac.jp
- **Yuya Tajima**, NTT Communications, yuya.tajima@ntt.com

This work was partially supported by JST, CRONOS, Japan Grant Number JPMJCS24N9.

Appendix



I E T F

SRv6 SFC Architecture

The concept that comprehensive controller manages the entire SRv6 SFC network.

The controller collects information about Service Segments and apply SR Policy based on the information about available Network Functions.

