



Why move to EVPN and Segment Routing ?

Dominique Schaeffer,
Director of IP Consulting Engineering EMEA

MENOG 2025, Dubai, Nov 27th – 28th

NOKIA

IXPs challenges

A big L2 switch

Only BGP route exchanges and Data traffic is desired

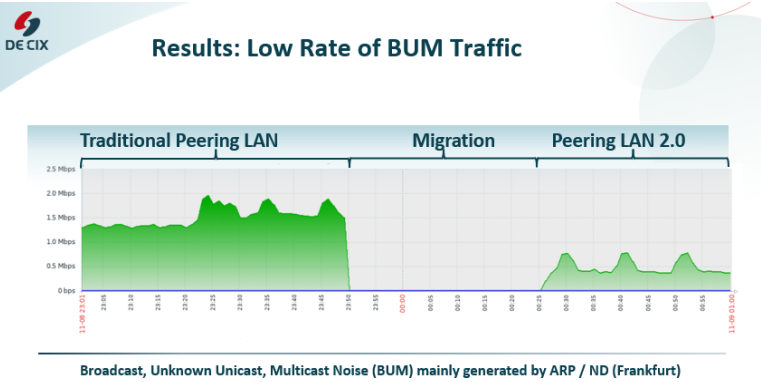
Noise (unwanted protocols, BUM, etc.) can be extremely intrusive and must be avoided

Fast convergence and redundancy needed in case of failure



EVPN with proxy-ARP/ND to the rescue

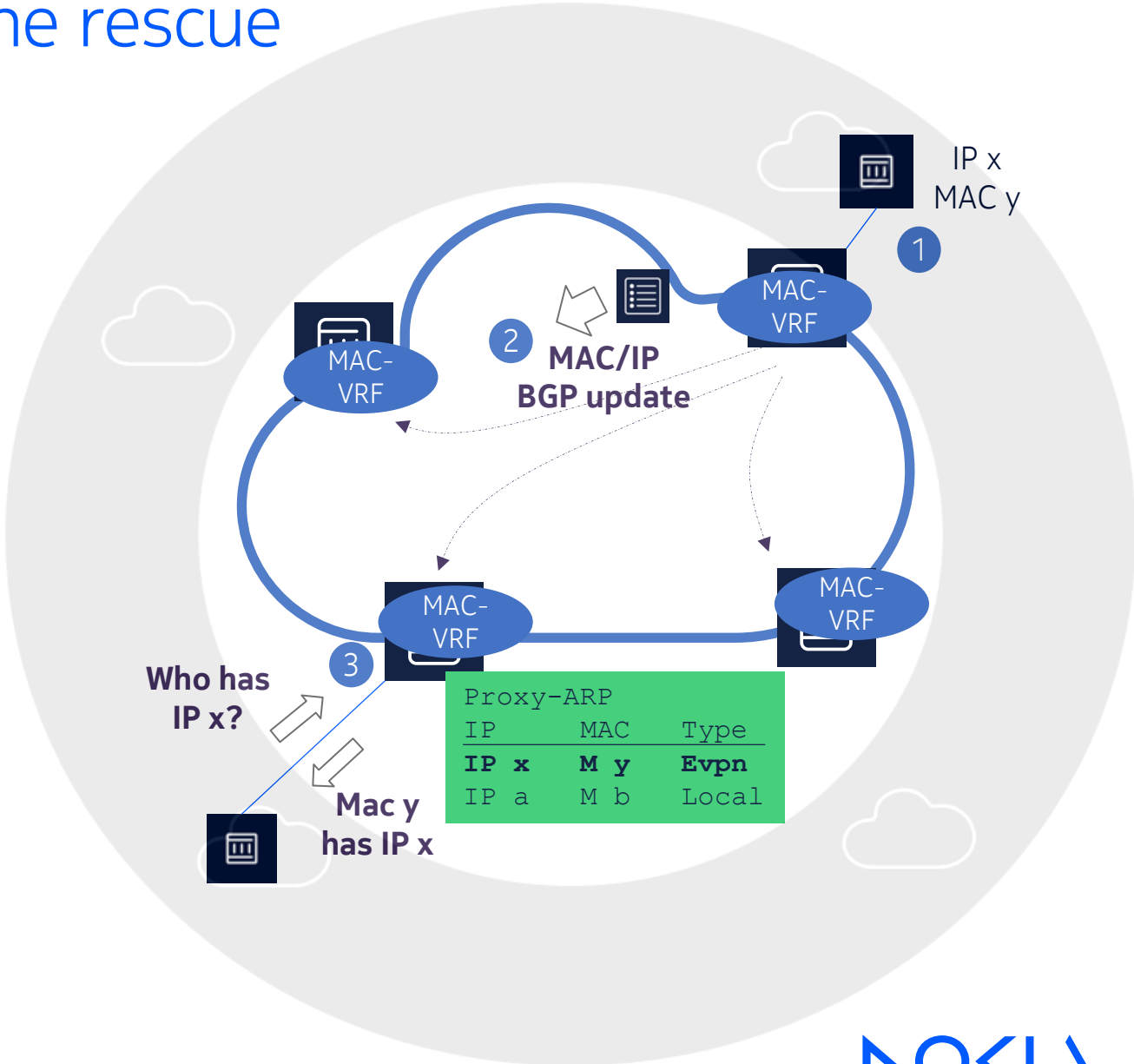
EVPN with proxy-ARP/ND reduce significantly the BUM traffic.



Source : EVPN Innovations for Internet Exchange Providers (IXPs), Thomas King, MPLS SD&AI NET WORLD Paris 2024



Source : IX.br evolution to SRv6 EVPN, Fabio Pessoa Nunes, Innovations for Internet Exchange Providers (IXPs), LACNOG

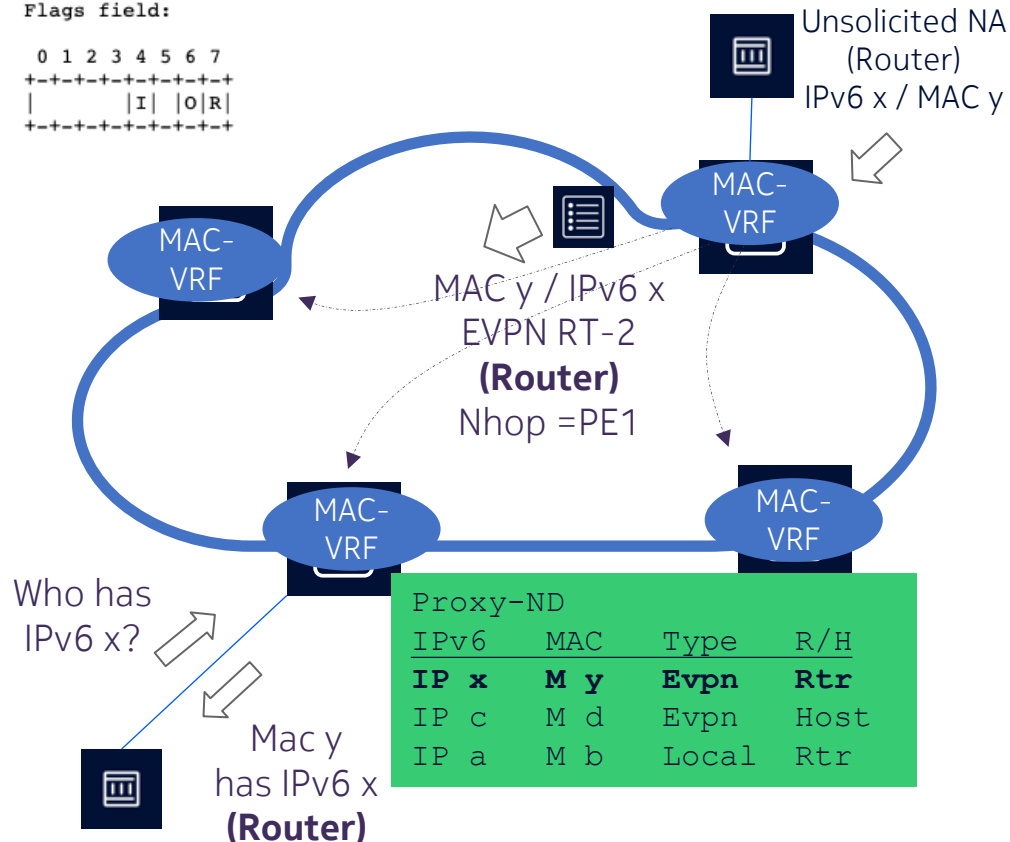


Extended proxy-ARP/ND handling in RFC 9047 and RFC 9161

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type=0x06 | Sub-Type=0x08 | Flags (1 octet) | Reserved=0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Reserved=0                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Flags field:

```
0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+
| I | O | R |
+---+---+---+---+---+---+---+---+
```



Issues in proxy-ND prior to RFC 9047 / 9161

- PEs snoop NA msgs + flags that have impact on how hosts and routers build their neighbour cache (RFC 4861)
- R (Router) flag : responder can route packets
- O (Override) Flags: responder owns an anycast IPv6 – should not override
- Proxy-ND can learn these flags but cannot convey this info in EVPN

Issues in proxy-ND/ARP with static entries

- PE1 learns IP1/M1 dynamically
- PE2 is statically configured with IP1/M2
- Both PE1/PE2 advertise same IP/MAC route info
- PE3 receives info from both -> which one to pick?

Solution: R, O and I flags advertised along with MAC/IP routes

- R=1 indicates the entry is a router, R=0 a host. This allows the PEs to advertise routes for host and routers at the same time
- O flag is propagated from NA -> EVPN -> NA
- Static and Configured proxy-ARP/ND entries are advertised with I=1. The I flag has an impact on the selection on Rx

Maximum security for IXPs

Configured IPs and MAC lists

- Ensure an IP is bound to one MAC out of an authorized list
- Dynamic IP/MAC learning can be turned off

EVPN flood options

- Prevent any flooding of unknown ARP Requests or GARPs towards EVPN tunnels – similar for ND unknown NS and unsolicited NA messages

Sponge MAC

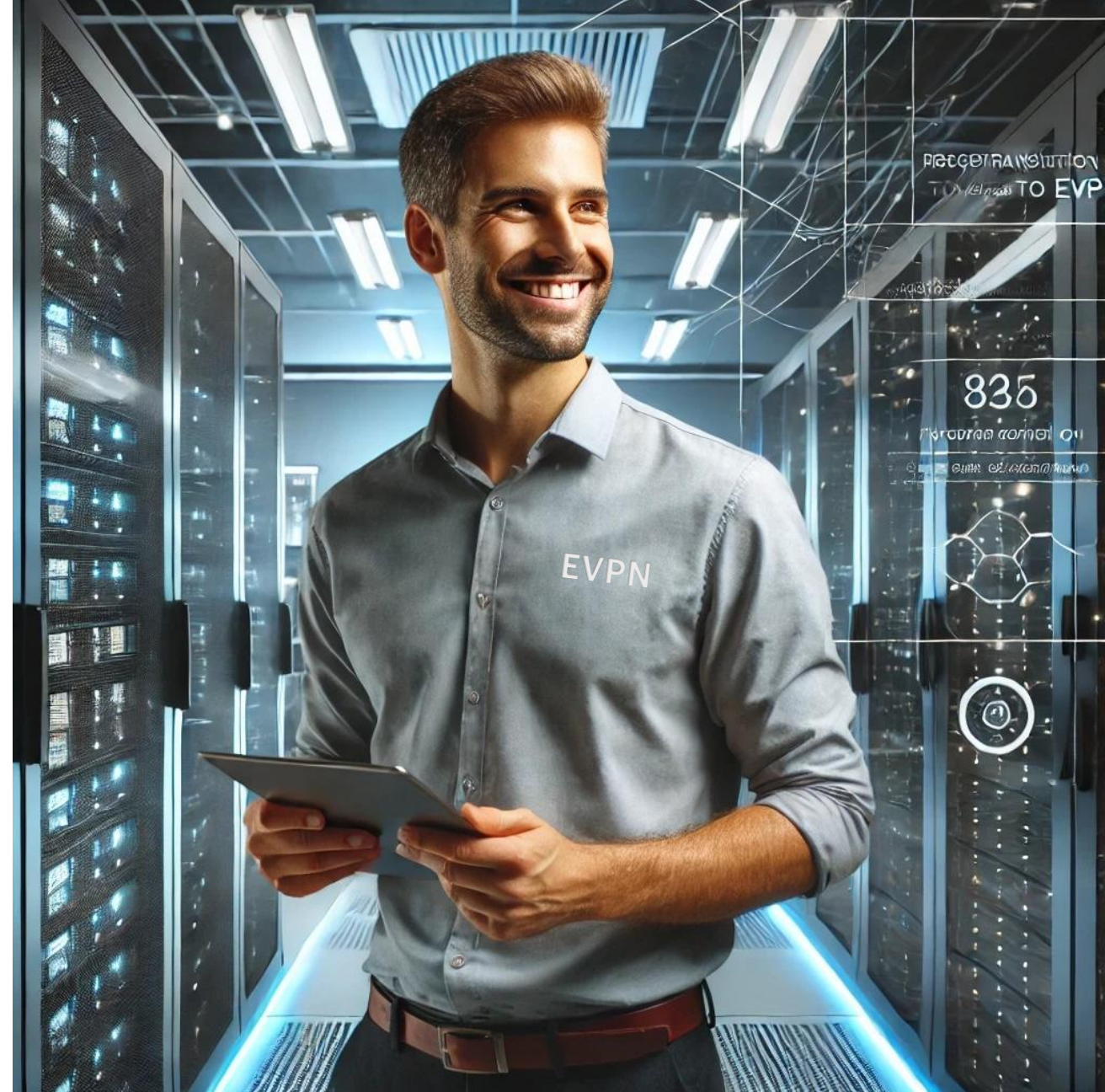
- When a sponge MAC is configured, ARP Requests or NSs from an unauthorized IP address are:
 - not learned in proxy-AR/ND table
 - replied with the configured sponge MAC address
- Any IP not configured as static-arp/nd IP can be considered unauthorized
- Sponge MAC can be configured as static MAC and blackholed or sent to collector



Why move to EVPN ?

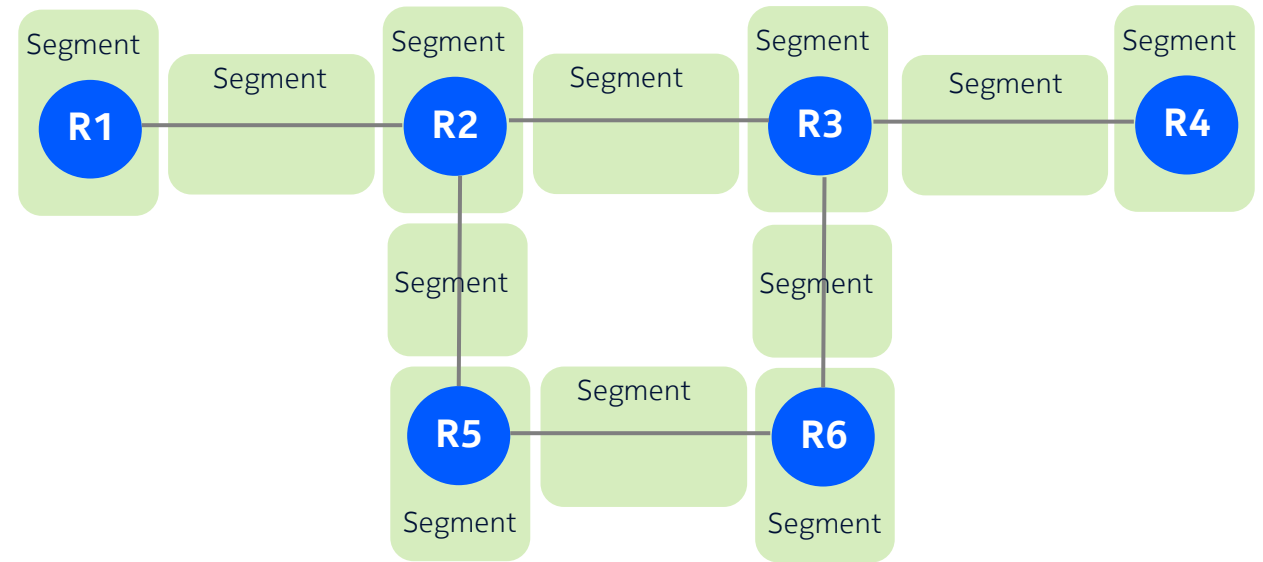
Key advantages

- Remove BUM noise traffic
- Enhanced security
- Single-active / all-active multi-homing
- Future proof protocol
- Ready for SRv6
- Automation capabilities
- Smooth interworking and integration with VPLS
- Full tooling for port hygiene
- Loop protection with black-hole MAC



Segment routing introduction

- Segment Routing (SR) provides a tunneling mechanism via source routing
- A segment can represent a node, link, or service
- Topological segments are advertised by routing protocols (IS-IS and OSPF)
- A SR tunnel can be encoded as
 - A single **MPLS label** or a stack of MPLS labels
 - A single **IPv6 address** or several IPv6 addresses in the IPv6 Extension header (Segment Routing Header).
- The segments can be thought of as a set of instructions executed from the ingress PE



SR-MPLS and SRv6 comparison

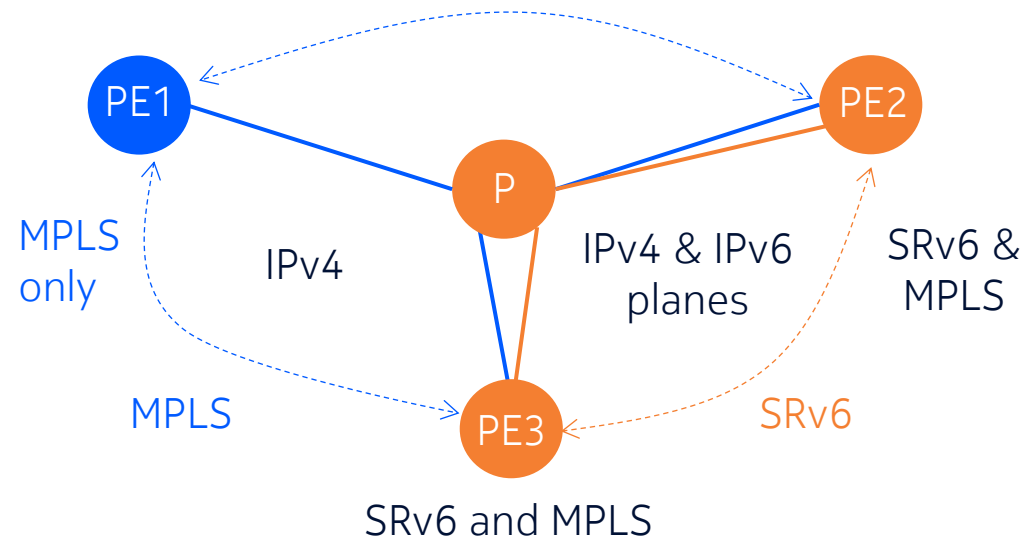
	SR-MPLS	SRv6 μ SID
IP Infrastructure	IPv4 or/and IPv6	IPv6 only
Traffic Load Balancing	Hash or Entropy label	IPv6 header flow-id field
Services	Any service over MPLS	Supported : BGP IP-VPN and EVPN Not supported: T-LDP VPWS, VPLS
Traffic engineering	FlexAlgo, PCEP, BGP SR-Policy	FlexAlgo PCEP, BGP SR-Policy (standards still in progress)
Interworking with legacy	BGP-LU or SR/LDP mapping server No impact on services	Services-GW to ease migration from SR-MPLS to SRv6 Requires T-LDP service migration to EVPN/IP-VPN
Intermediate nodes	Intermediate node need to support MPLS	Intermediate nodes need to be IPv6 native only (if no traffic engineering)
Skills & training	Familiar to planning and operations	IPv6 address/locator planning, SRv6 coupled overlay/underlay operations is very different.
Field proven	Vast majority of multi-vendor deployments	Few multi-vendor deployments

Migration to SRv6

Multiple options

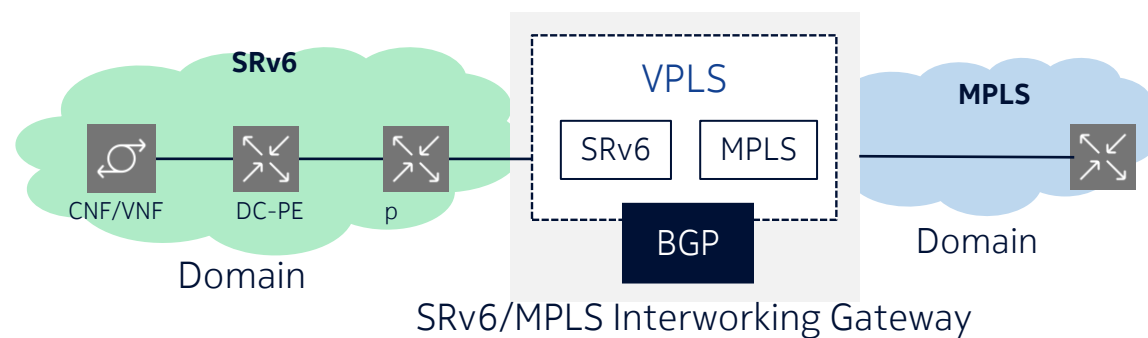
Option 1 : Dual planes model

- Multi-instance SRv6 and MPLS capable platforms transition to SRv6 transport progressively, while using MPLS towards nodes that do not support SRv6



Option 2 : Interworking gateway model

- Upgrade one domain at a time, while introducing Interworking-GW at domain boundary.



Why move to Segment Routing ?

Key advantages

- Increase network resiliency by leveraging TI-LFA
- Simplify troubleshooting as MPLS label or IPv6 address is consistent in the domain (Node or Prefix SID) (for non compressed SRv6)
- Good visibility of the traffic leveraging statistics per SID (for MPLS)
- Building flexible services by leveraging traffic engineering capability
- Possibility to implement a centralized controller for full control
- No more LDP and LDP/IGP sync issue; taking advantage of IGP extension
- Control plane simplification
- Easier load balancing (for SRv6)



Want to learn more?

Free book

Segment Routing book:

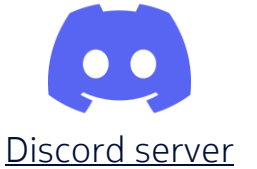
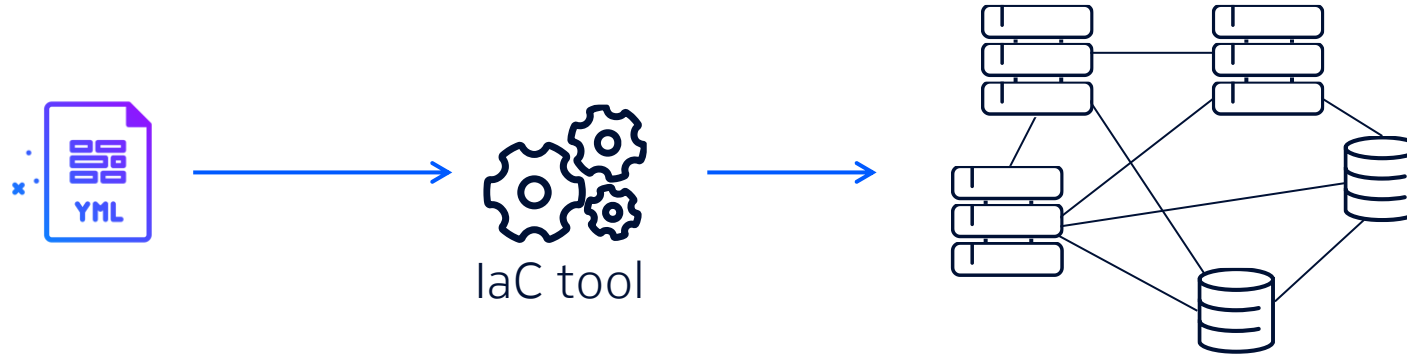
<https://doc.sros.dev/SR-with-SROS>



Testing peering configuration with Containerlab

Bringing declarativeness to networking labs

IT

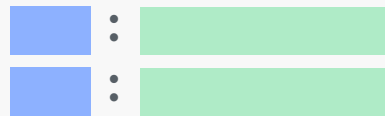


Network Labs

```
name: mylab
```

```
topology:
```

```
  nodes:
```

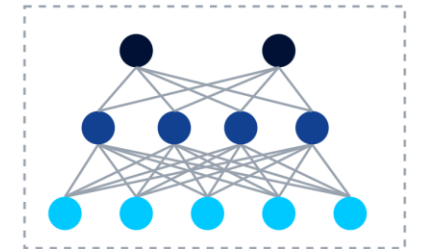


```
  links:
```



CONTAINERlab

<https://containerlab.dev>

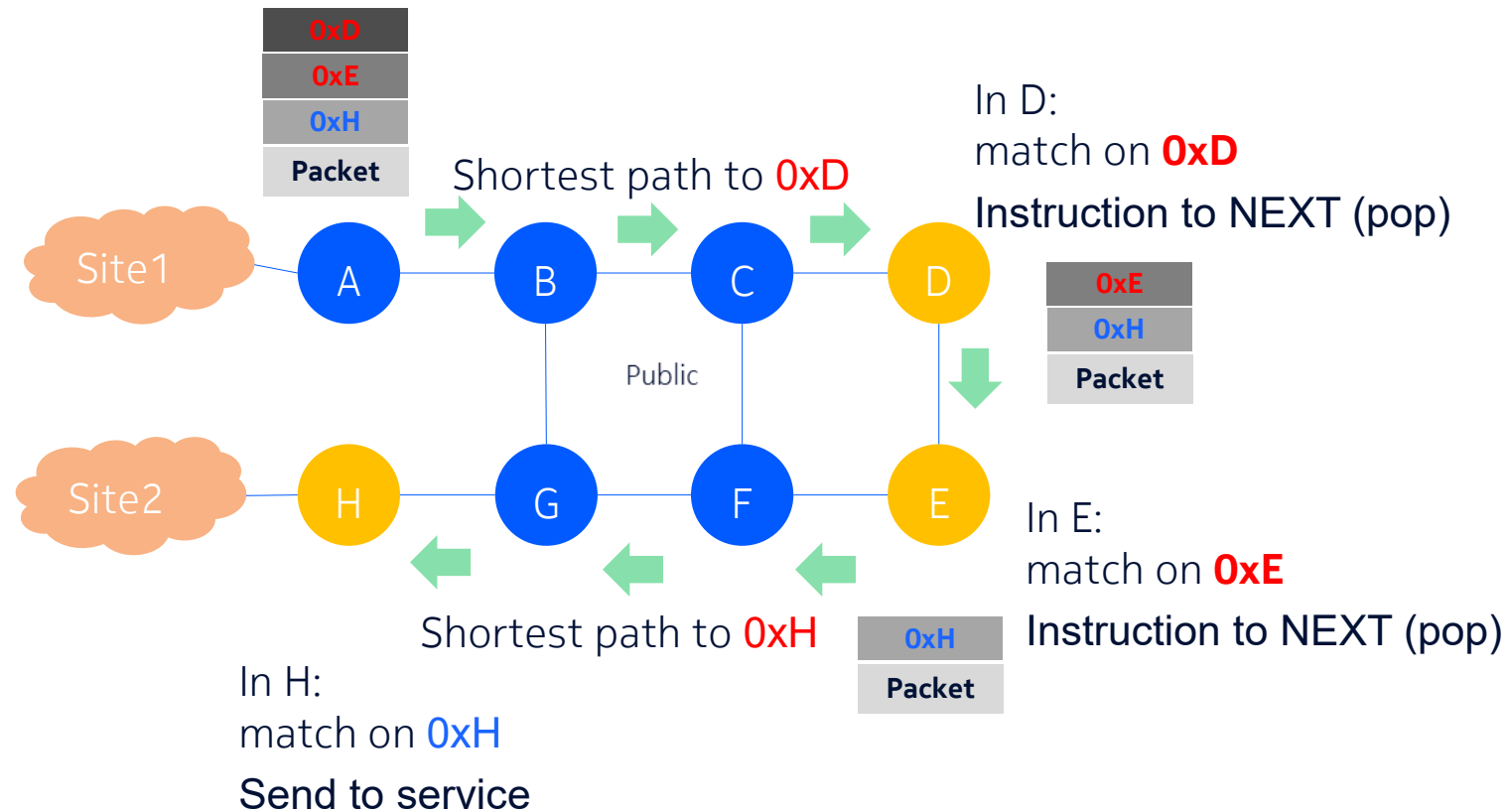


NOKIA

Backup slides

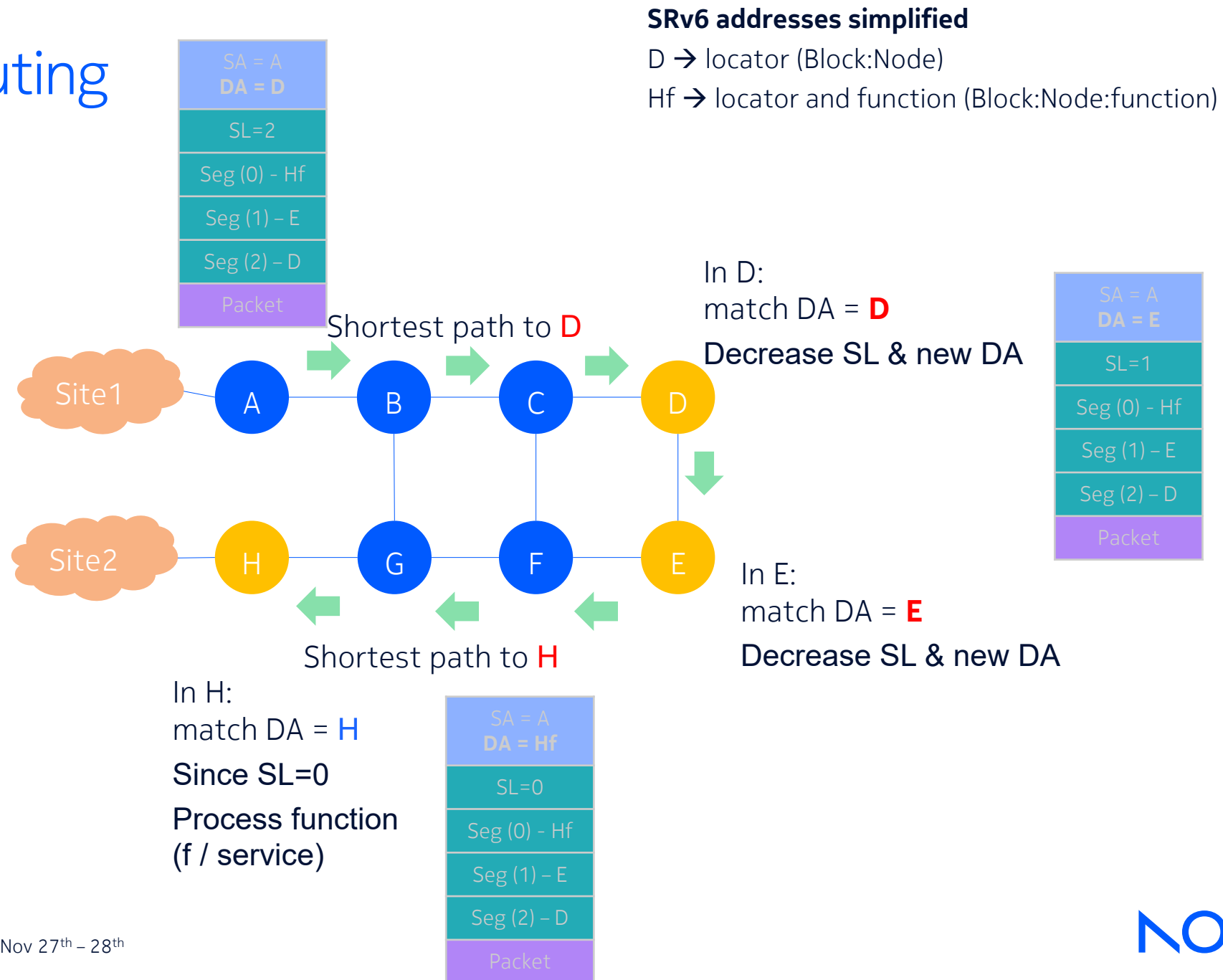
Segment Routing

MPLS data plane



Segment Routing

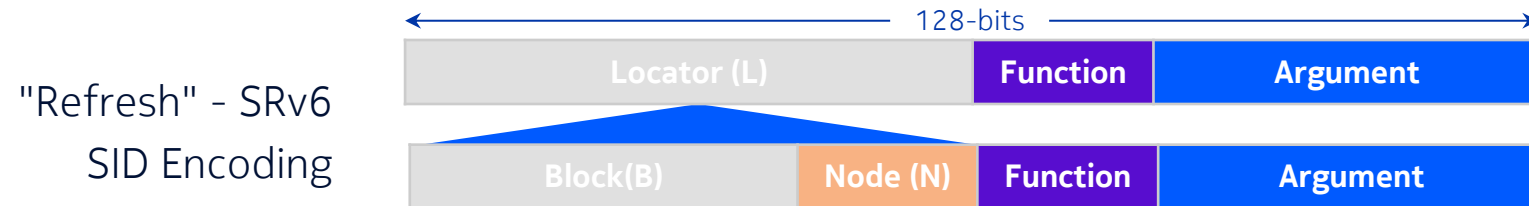
SRv6 data plane



SRv6 SID refresh

SID represented by an IPv6 address

- Represented by an IPv6 address
- Locator / Block / Node / Function

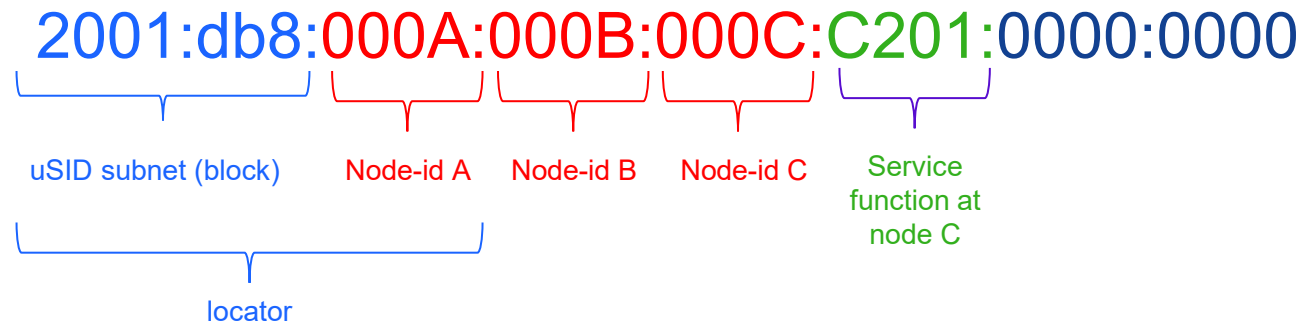


- SRv6 advantages:
 - Routable
 - Easy to cross domain boundaries

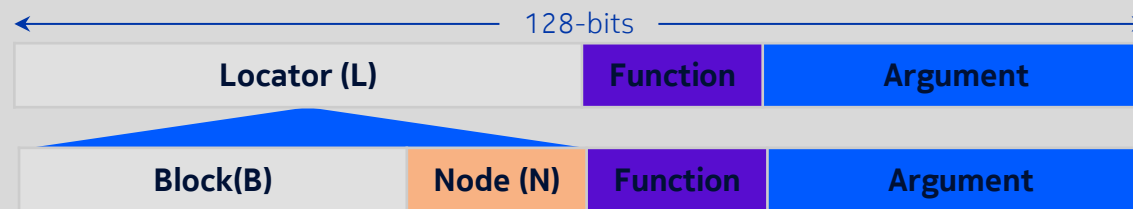
Micro SID

Base principles

- 2 - Let's concatenate the identifiers
- Identifiers can be concatenated in 128 bits (uSID Carrier) and placed in IPv6 DA (or in SRH)



"Refresh" - SRv6
SID Encoding



Micro SID

Base principles

- 1 - Let's make Segment Identifiers shorter than 128 bits
- Identifiers are 16-bit long
- Represent the same instructions (topological/service) as SRv6
- Exist under an address **block** (like SRv6)
- 16-bit addressing space used to
 - globally identify **nodes** (and allow routing to happen)
 - identify **local instructions** (adjacencies, services)

Global uSIDs

- Configurable size
- In SR-OS *SID multiplier* is configurable and determine the size of the Global uSID (GIB)*

Remaining uSIDs are **local instructions***

- Adjacency SIDs, service termination

Example with SID multiplier 40



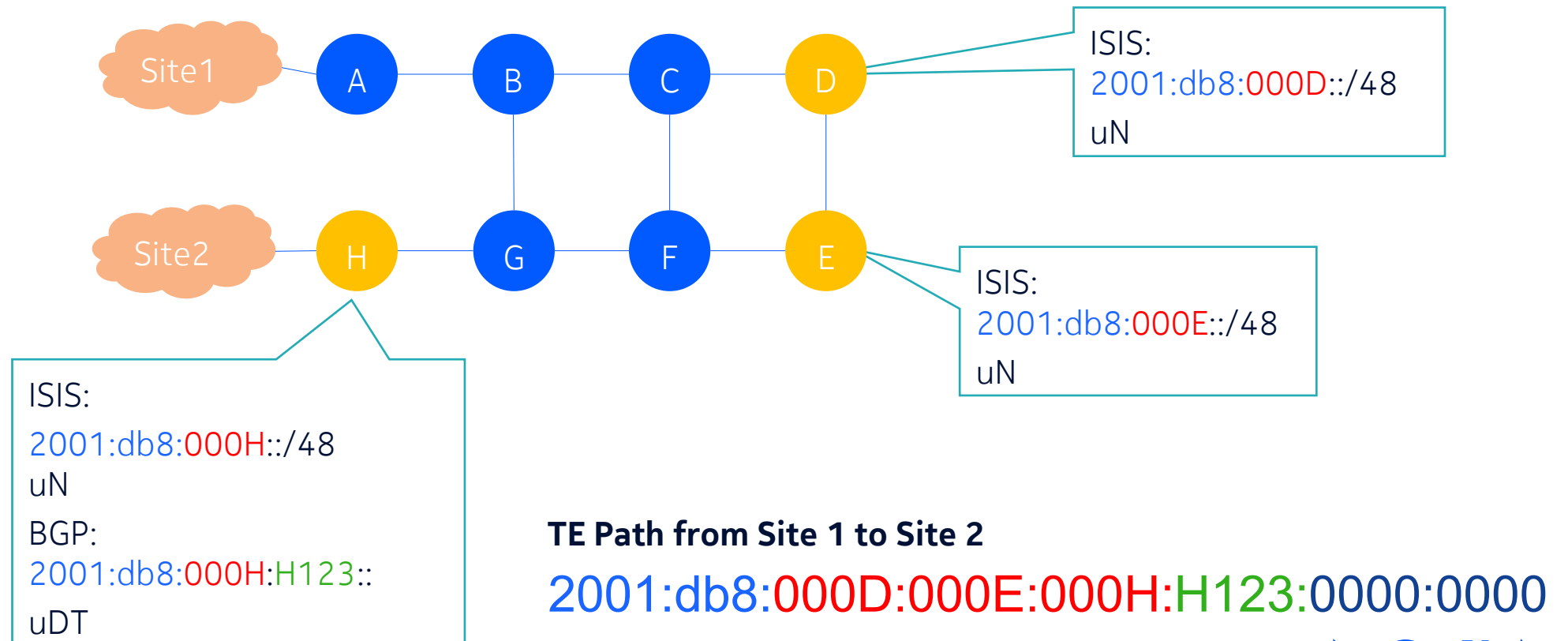
SID multiplier	Start of GIB (hex)	End of GIB (hex)	GIB Count	Start of LIB (hex)	End of LIB (hex)	LIB Count
16	0001	3FFF	16383	4000	FFFF	49151
20	0001	4FFF	20479	5000	FFFF	45055
24	0001	5FFF	24575	6000	FFFF	40959
28	0001	6FFF	28671	7000	FFFF	36863
32	0001	7FFF	32767	8000	FFFF	32767
36	0001	8FFF	36863	9000	FFFF	28671
40	0001	9FFF	40959	A000	FFFF	24575
44	0001	AFFF	45055	B000	FFFF	20479
48	0001	BFFF	49151	C000	FFFF	16383
52	0001	CFFF	53247	D000	FFFF	12287
56	0001	DFFF	57343	E000	FFFF	8191
60	0001	FFFF	61439	F000	FFFF	4095

Size of Global uSID and local instructions compared to configured SID multiplier

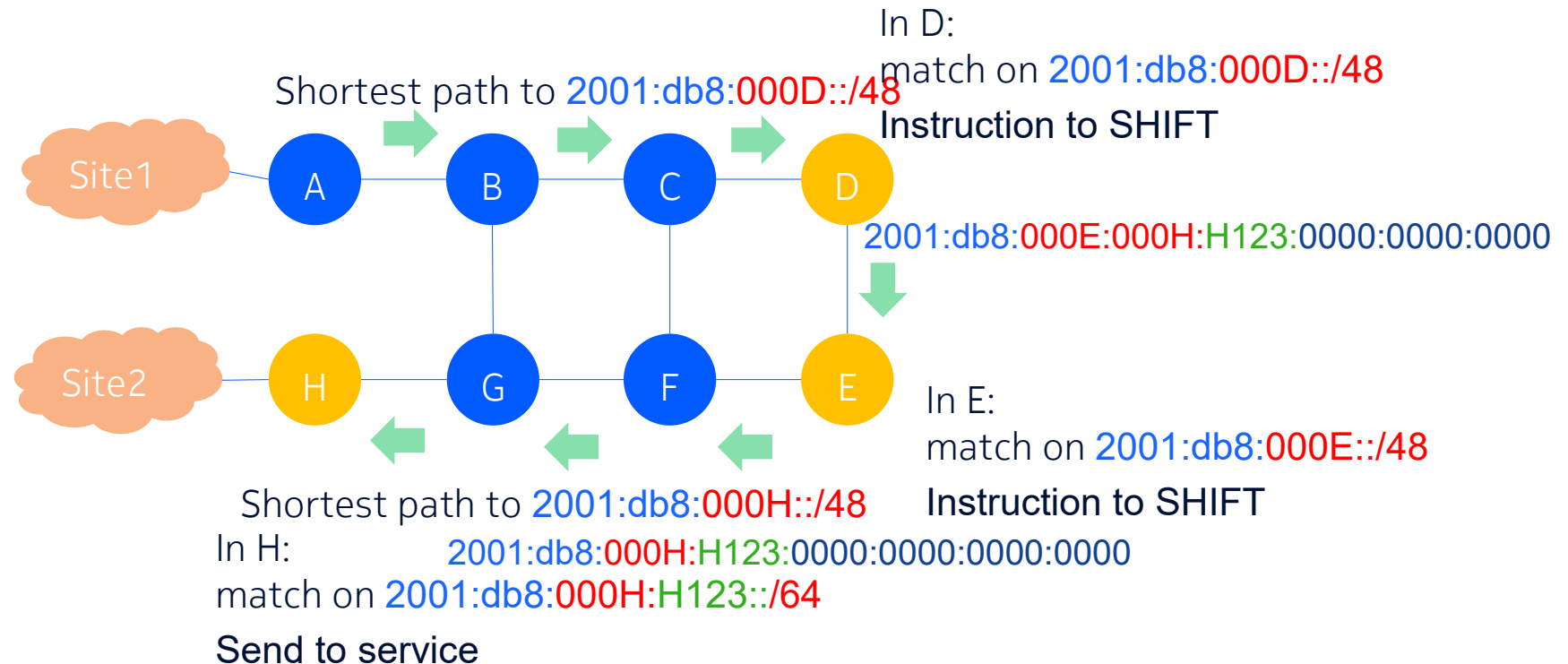
Micro SID

Operation

- Routing merely depends on which addresses have been advertised and which entries are in router FIBs
- Micro SID defines specific instructions associated to FIB entries



Micro SID Operation



`2001:db8:000D:000E:000H:H123:0000:0000`

Maximum security for IXPs

```
*[ex:/configure service vpls "1"]
A:admin@PE-2# info
fdb {
    static-mac {
        mac 00:00:00:00:00:01 {
            sap 1/1/c1/1:504
            monitor forward-status
        }
        mac 00:00:00:00:01:01 {
            sap 1/1/c1/1:504
            monitor forward-status
        }
        mac 00:00:00:00:00:02 {
            sap 1/1/c2/1:504
            monitor forward-status
        }
        mac 00:00:00:00:02:02 {
            sap 1/1/c2/1:504
            monitor forward-status
        }
    }
}
proxy-arp {
    admin-state enable
    dynamic-populate true
    send-refresh 120
    flood {
        gratuitous-arp false
        unknown-arp-req false
    }
    dynamic-arp {
        restrict-non-configured-ip-addresses true
        sponge-mac 00:de:ad:de:ad:00
        ip-address 10.0.0.1 {
            mac-list "list-1" // contains 00:00:00:00:00:01 and 00:00:00:00:01:01
        }
        ip-address 10.0.0.2 {
            mac-list "list-2" // contains 00:00:00:00:00:02 and 00:00:00:00:02:02
        }
    }
}
```

Configured IPs and MAC lists

- Ensure an IP is bound to one MAC out of an authorized list
- Dynamic IP/MAC learning can be turned off

EVPN flood options

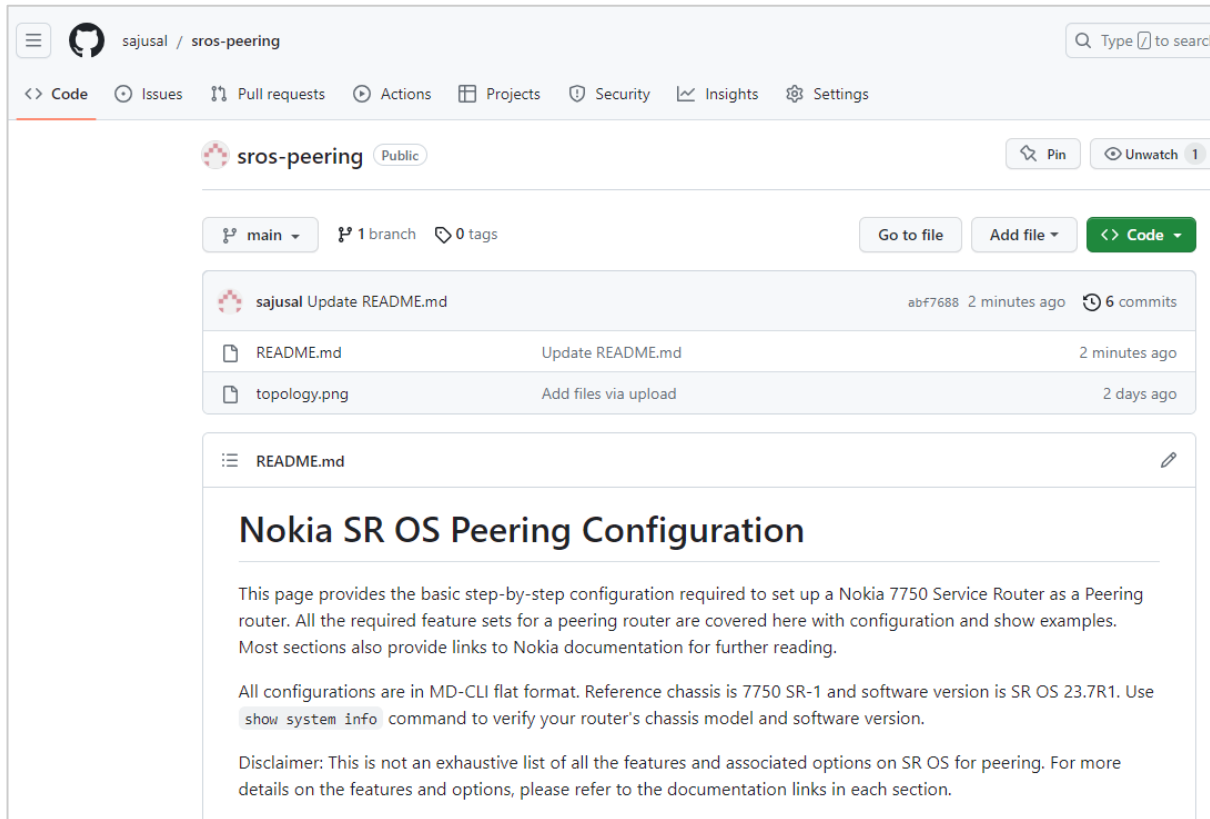
- Prevent any flooding of unknown ARP Requests or GARPs towards EVPN tunnels – similar for ND unknown NS and unsolicited NA messages

Sponge MAC

- When a sponge MAC is configured, ARP Requests or NSs from an unauthorized IP address are:
 - not learned in proxy-AR/ND table
 - replied with the configured sponge MAC address
- Any IP not configured as static-arp/nd IP can be considered unauthorized
- Sponge MAC can be configured as static MAC and blackholed or sent to collector

Peering reference configuration

<https://github.com/sajusal/sros-peering>



The screenshot shows the GitHub repository page for 'sajusal/sros-peering'. The repository is public and has 1 branch and 0 tags. The main branch is 'main'. The repository contains a README.md file, a topology.png file, and a commit history showing updates to README.md and topology.png. The README.md file is titled 'Nokia SR OS Peering Configuration' and provides a step-by-step configuration guide for setting up a Nokia 7750 Service Router as a Peering router. It includes a disclaimer that this is not an exhaustive list of all features and options on SR OS for peering.

Route Policies

Routing policies control the size and content of the routing tables, the routes that are advertised, and the best route to take to reach a destination.

For more details on route policy configuration and options, visit [SR OS Route Policies Documentation](#).

In these examples, we are creating AS path lists, community and prefix lists.

```
/configure policy-options as-path "PEERING" { expression "64503" }
/configure policy-options as-path-group "BOGON" { entry 10 expression ".* 0 .*" }
/configure policy-options as-path-group "BOGON" { entry 20 expression ".* [64496-64511] .*" }
/configure policy-options as-path-group "BOGON" { entry 30 expression ".* 65535 .*" }

/configure policy-options community "LARGE-PEER" { member "65100:100" }
/configure policy-options community "SMALL-PEERS" { member "65200:200" }
/configure policy-options community "SMALL-PEERS" { member "65400:.*$" }
/configure policy-options community "SMALL-PEERS" { member "65500:.*" }
```

RPKI

SR OS supports RPKI for BGP prefix origin validation.

In this example, we are configuring a RPKI session with an external server and then enabling prefix origin validation under the BGP group. We are also configuring BGP to not use any routes whose origin is invalid.

For more details on RPKI implementation, visit [SR OS RPKI Documentation](#).

```
/configure router "Base" origin-validation rpki-session 172.31.1.2 { admin-state enable }
/configure router "Base" origin-validation rpki-session 172.31.1.2 { local-address 10.10.1.4 }
/configure router "Base" origin-validation rpki-session 172.31.1.2 { port 8282 }
/configure router "Base" bgp group "eBGP-Peering" { origin-validation ipv4 true }
/configure router "Base" bgp group "eBGP-Peering" { origin-validation ipv6 true }
/configure router "Base" bgp best-path-selection { origin-invalid-unusable true }
```