SRv6 uSID Data Center Use Case

Case Study

Gyan Mishra
“IT Technologist & Innovation Specialist”
Associate Fellow – Network Design
April 19th 2023

verizon

End-to-end cross-domain policies

- **Current solutions rely on:**
  - NVO in the DC (VxLAN, NVGRE, GENEVE, …)
  - MPLS in the Core
  - DPI for host intent classification and protocol conversion at domain boundaries
- **Expensive CAPEX/OPEX**
- **DPI incurs in performance and scale bottleneck**

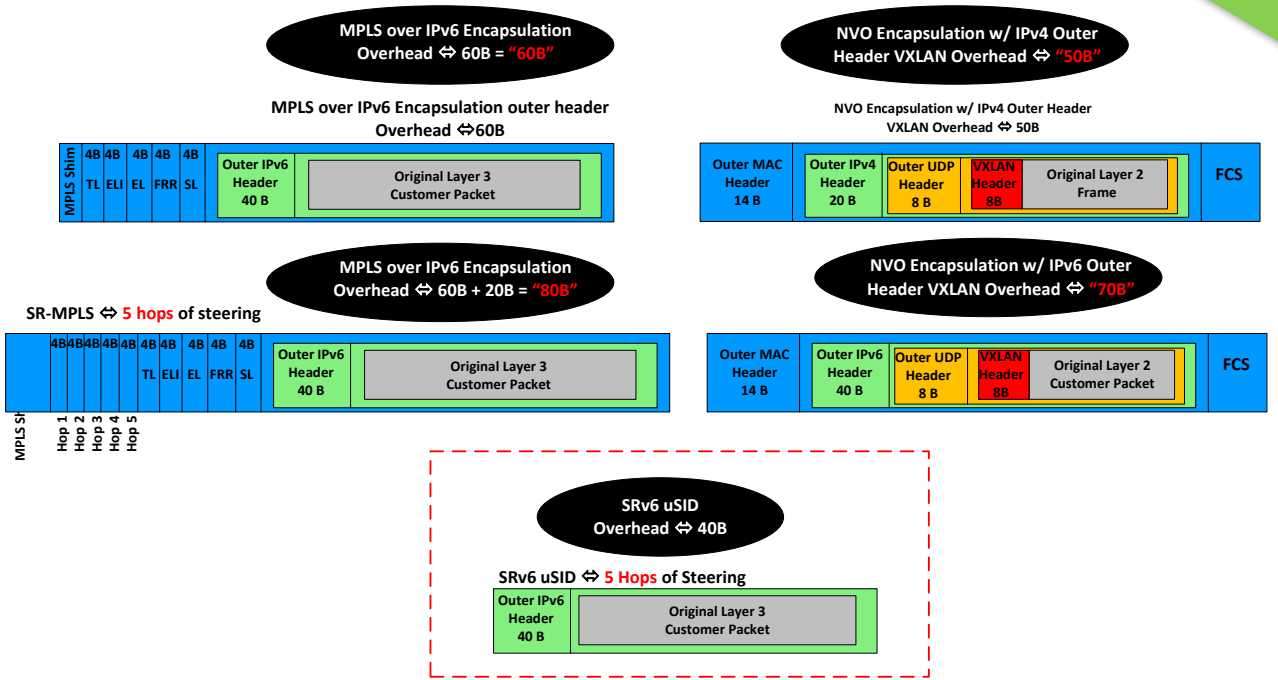
Load-balancing

- **VXLAN and MPLS rely on hacks to encode entropy**
- **MPLS Entropy Label is sub-optimal, difficult to find, implementations differ,…**
- **VXLAN encodes entropy within a sub-range of the UDP Source Port**

uSID provides optimum HW entropy (shallow, fixed offset) with the IPv6 Flow Label ✔

Fiber tax (MTU overhead)

- **NVO encapsulations incur in expensive MTU overhead**



Minimal overhead with uSID ✓

Host Networking

- **MPLS failed to reach the host**
- **We want to extend the fabric to the host**
- **SRv6 uSID provides several alternatives:**
  - Router-in-container (xRD, SONiC, Nokia, Juniper cRPD)
  - eBPF/Cilium (with GoBGP for the control plane)
  - FD.io VPP/Calico (with FRR control plane)
  - Native Linux Kernel (with FRR control plane)

VNFs

- **NVO (VXLAN, NVGRE, GENEVE) do not support service chaining**
- **All require complex PBR engineering, state at every hop, and processing to link services together in a service chain**

uSID provides ultra simplicity service chaining ✔

Traffic Engineering in the DC:

- **We need Traffic Engineering for certain flows within the Data Center!**
  - Selective steering of elephant flows in the DC to avoid “hot spots”
  - Bandwidth upgrade transitional periods where links are not the same bandwidth and UCMP load balancing
  - Excluding Link & Nodes experiencing congestions hot-stops
  - Mission critical mice-flows that require low-latency & jitter tolerance

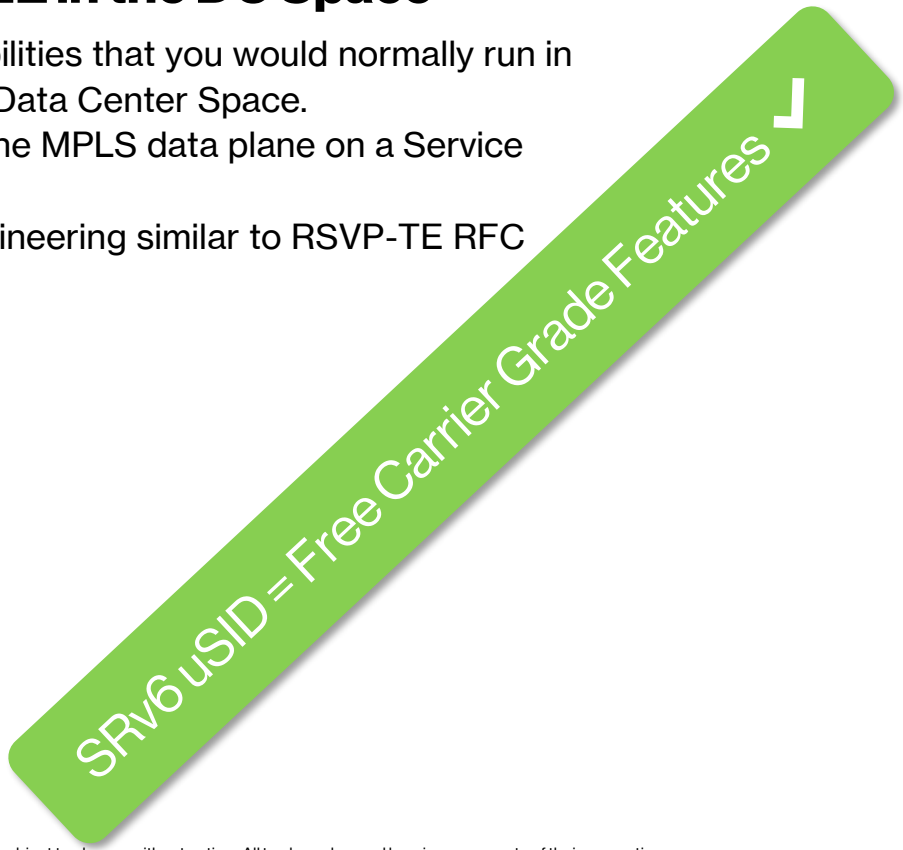
Proprietary technology

- **VxLAN schemes often include proprietary elements**
  - Each implementation defines their own bits for influencing load-balancing or learning processes.

SRv6 uSID is 100% open/IETF Standard ✔

SRv6 uSID ⇔“What you get for FREE in the DC Space”

- SRv6 gives you Carrier grade feature rich capabilities that you would normally run in the Core network, you now have available in the Data Center Space.
- With SRv6 you get similar features provided by the MPLS data plane on a Service provider network including:
  - MPLS Data plane capabilities of traffic engineering similar to RSVP-TE RFC 3209
  - IP-VPN capabilities RFC 4364
  - BGP EVPN capabilities RFC 7432
  - BGP NVO overlay capabilities RFC 8365
  - Global Table routing
  - Native IP Data plane
  - Network Slicing
  - Flex Algo
  - SR-PM Performance Measurement
  - Path Tracing
  - Traffic Engineering capabilities



Vendor, Merchant & SONiC maturity

- **SONiC support**
  - Rich support in SAI/SONiC/FRR stack
- **Merchant SRv6 uSID**
  - Broadcom Jericho/Jericho2
  - Broadcom Trident4
  - Broadcom Tomahawk5
  - Cisco Silicon One

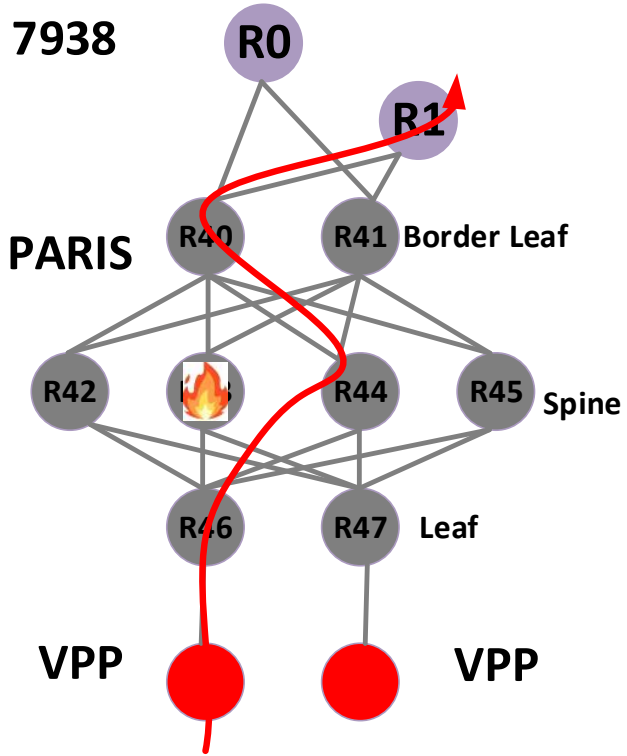
Demo time!

- **Use-case 1: SRv6 uSID DC fabric**
  - Host-to-Host
  - Policy programmed from Linux Kernel & VPP host
- **Use-case 2: Inter-DC**
  - Host-to-Host across the metro
  - Metro with several planes, and FlexAlgo
- **Full demo available in here: https://youtu.be/w7R53ni8ATk**

Use-Case 1: SRv6 uSID Intra-DC steering

DC-2 BGP Only DC RFC 7938
Node number = ASN

SR Algo 0 (All links)
SR Algo 0 -Latency)
SR Algo 0-Latency



Use-Case 1: SRv6 uSID DC Fabric Packet Capture & Screen Scrapes

SRv6 uSID IPv4 payload steer DC-2 Paris ⇔ Core ⇔ DC-1 Berlin using VPP Host attached to DC fabric

SRv6 uSID IPv4 payload steering policy

```
vpp#sr policy add bsid 40::40 next fc00:0:44:40:4:64:66:e000 encap

vpp#sr steer l3 10.0.0.66/32 via bsid 40::40

vpp# show sr policies
SR policies:
[0].-   BSID: 40::40
        Behavior: Encapsulation
        EncapSrcIP: fc00:0:46:1::3
        Type: Default
        FIB table: 0
        Segment Lists:
        [0].- < fc00:0:44:40:4:64:66:e000 > weight: 1

vpp# show sr steering-policies
SR steering policies:
Traffic         SR policy BSID
L3 10.0.0.66/32 40::40
```

SRv6 uSID IPv6 payload steer:

```
vpp#sr policy add bsid 41::41 next fc00:0:44:40:4:64:66:e000  encap

vpp# sr steer l3 fc00:0:66::1/128 via bsid 41::41

vpp# show sr policies
SR policies:
[1].-   BSID: 94::94
        Behavior: Encapsulation
        EncapSrcIP: fc00:0:46:1::3
        Type: Default
        FIB table: 0
        Segment Lists:
        [1].- < fc00:0:45:41:66:e000:: > weight: 1
-----------
vpp# show sr steering-policies
SR steering policies:
Traffic         SR policy BSID
L3 fc00:0:66::1/128     41::41
```

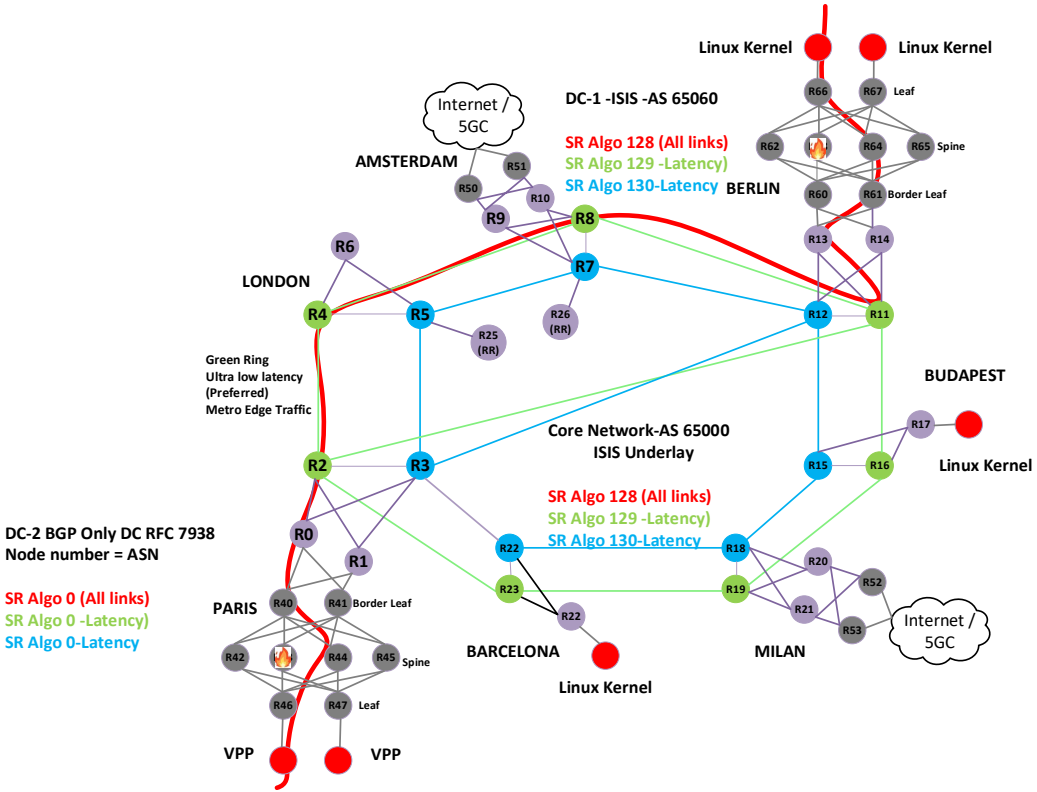
IPv4 payload packet capture xrd61-xrd64

```
04:41:56.724814 IP6 fc00:0:46:1::3 > fc00:0:64:66:e000::: IP 10.11.46.2 > 10.0.0.66: ICMP echo request, id 113, seq 463, length 64
04:41:57.724271 IP6 fc00:0:46:1::3 > fc00:0:64:66:e000::: IP 10.11.46.2 > 10.0.0.66: ICMP echo request, id 113, seq 464, length 64
```

IPv6 payload packet capture xrd61-xrd64:

```
04:55:37.285381 IP6 fc00:0:46:1::3 > fc00:0:64:66:e000::: IP6 fc00:0:46:2::2 > fc00:0:66::1: ICMP6, echo request, seq 368, length 64
04:55:38.285012 IP6 fc00:0:46:1::3 > fc00:0:64:66:e000::: IP6 fc00:0:46:2::2 > fc00:0:66::1: ICMP6, echo request, seq 369, length 64
```

Use-Case 2: SRv6 uSID Inter-DC Drawing



Use-Case 2: SRv6 uSID Inter-DC Packet Capture & Screen Scrapes

SRv6 uSID IPv4 payload steer DC-1 Berlin ⇔ Core ⇔ DC-2 Paris using Linux host attached to DC fabric

SRv6 uSID IPv6 payload steering policy:

root@ubuntu-linux-srv6:sudo ip route add 10.0.0.46/32 encap seg6 mode encap segs fc00:0:64:61:4:44:46:e000 dev ens7
root@ubuntu-linux-srv6:/home/cisco# ip route
default via 192.168.122.1 dev ens8 proto dhcp src 192.168.122.88 metric 100
10.0.0.0/24 via 10.10.66.2 dev ens7 proto static
10.0.0.46 encap seg6 mode encap segs 1 [fc00:0:64:61:4:44:46:e000] dev ens7 scope link----------->SRv6 uSID steering programmed IPv4 payload

SRv6 uSID IPv4 payload steer capture DC-2 Paris:

xrd41-xrd44

20:30:23.274808 IP6 fc00:0:66:1:5054:2ff:fe41:b107 > fc00:0:44:46:e000::: srcrt (len=2, type=4, segleft=0[|srcrt]
20:30:24.275510 IP6 fc00:0:66:1:5054:2ff:fe41:b107 > fc00:0:44:46:e000::: srcrt (len=2, type=4, segleft=0[|srcrt]

SRv6 uSID IPv6 payload steering policy:

root@ubuntu-linux-srv6:sudo ip -6 route add fc00:0:46::1 encap seg6 mode encap segs fc00:0:64:61:4:44:46:e000 dev ens7
root@ubuntu-linux-srv6:/home/cisco# ip -6 route
::1 dev lo proto kernel metric 256 pref medium
fc00:0:46::1 encap seg6 mode encap segs 1 [fc00:0:64:61:4:44:46:e000] dev ens7 metric 1024 pref medium--->SRv6 uSID steering programmed IPv6 payload

SRv6 uSID IPv6 payload steer capture DC-2 Paris:

xrd44-xrd46

20:40:32.890109 IP6 fc00:0:66:1:5054:2ff:fe41:b107 > fc00:0:46:e000::: srcrt (len=2, type=4, segleft=0[|srcrt]
20:40:32.890994 IP6 fc00:0:46::1 > fc00:0:66:1:5054:2ff:fe41:b107: ICMP6, parameter problem, code-#4, length 176