



CO/DC

Network Transformation

Daniel Voyer
Technical Fellow
March 2017

What is Bell Canada ?

Our Origins

- Oldest in World (1880)
 - We really did invent the phone
- Largest in Canada
- Public
- Multiple ventures
 - Wireline, Wireless, Media, Enterprise, etc.
 - Satellites, Sports teams,

Network 3.0 is a journey to...

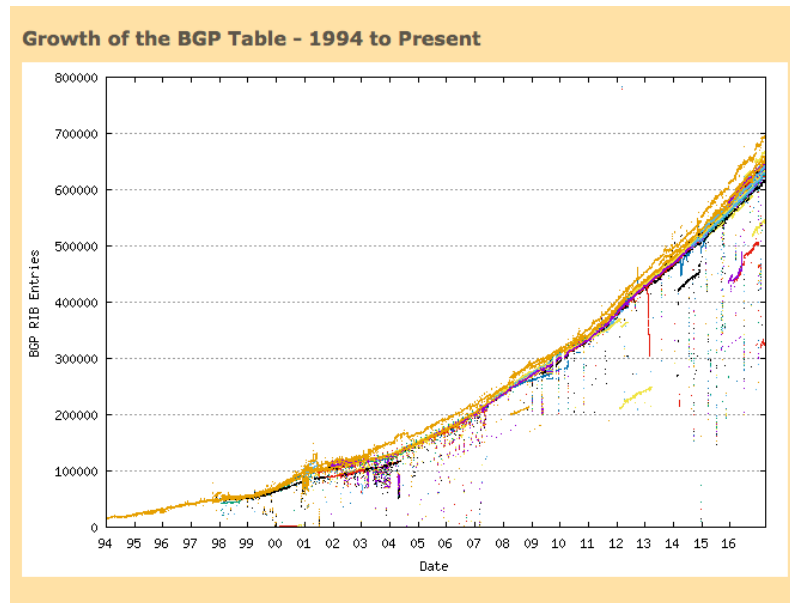
Transform how Bell delivers the best customer experience with seamless access to a software-driven, cloud-based ecosystem

Network 3.0

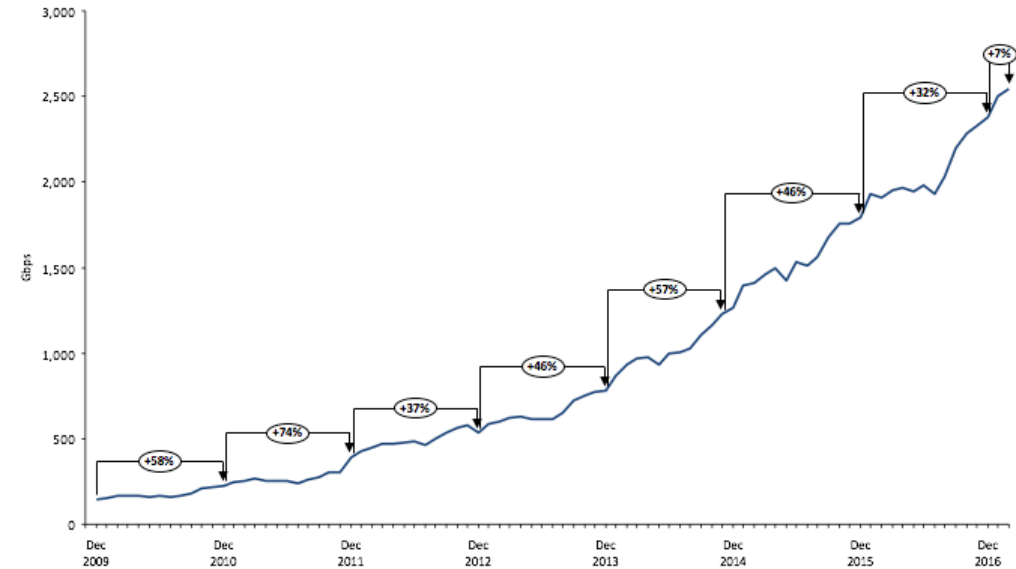
- SP Transformation
 - Culture
 - Processes
 - Technology
- New mode of operations (cloud)
- New competition (OTT)
- New services (NFV)
- Agility

Challenges - Internet traffic is growing

- Internet grow exponentially
- Physical Networks are static and requires long cycle migration changes



Source <https://bgp.potaroo.net>

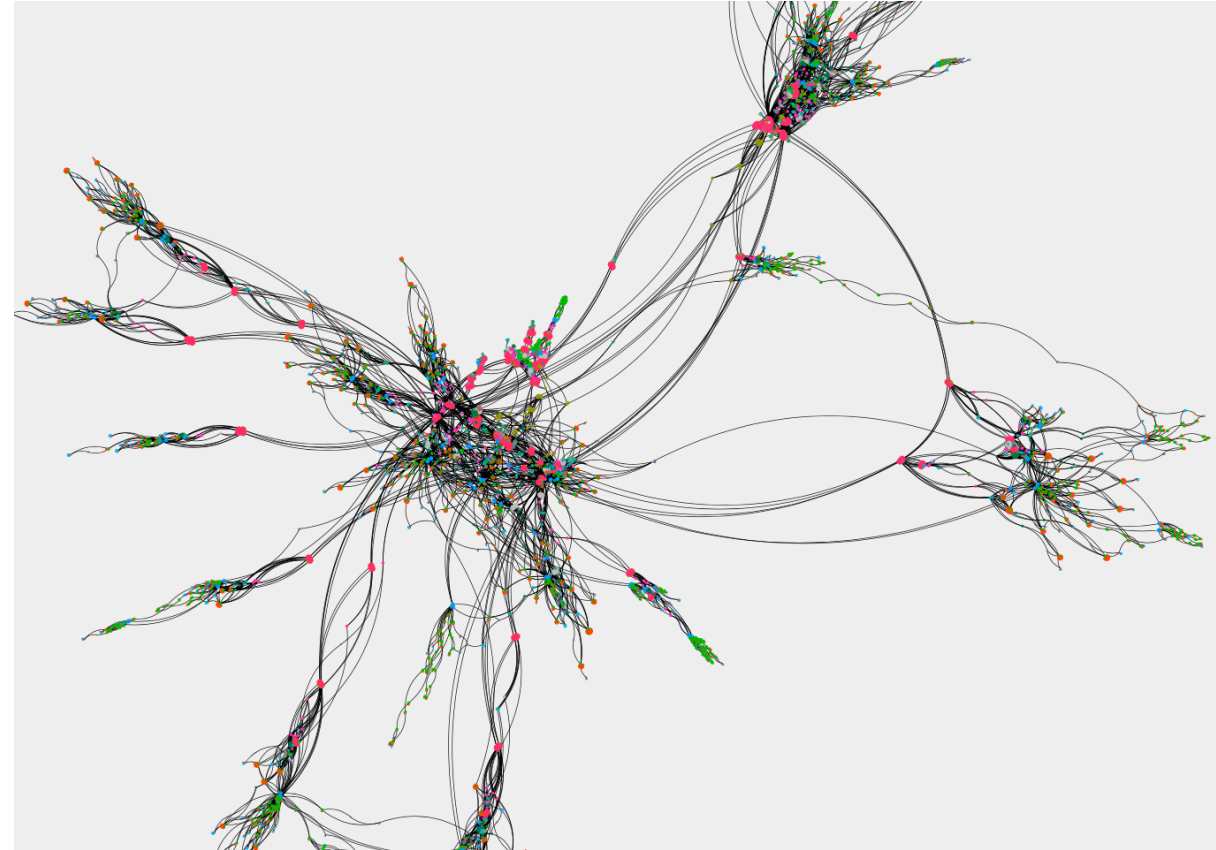


- Hit TCAM limits
 - August 2014 widespread outages
- Cost more money \$\$\$

Growing faster than we can adapt – and pay for ...

Challenges – Bell's Own Complexity

- Many independent MPLS domains today
 - Long provisioning cycles
 - Can take up to 3-4 weeks with tools (or longer) to engineer
 - No E2E Traffic Engineering
 - Complex with state in the network
 - Static and hardcoded, it's always on
 - No E2E OAM
 - Not always aware that tunnels are failing
 - Poor visibility of the state of the tunnels
 - hop by hop troubleshooting



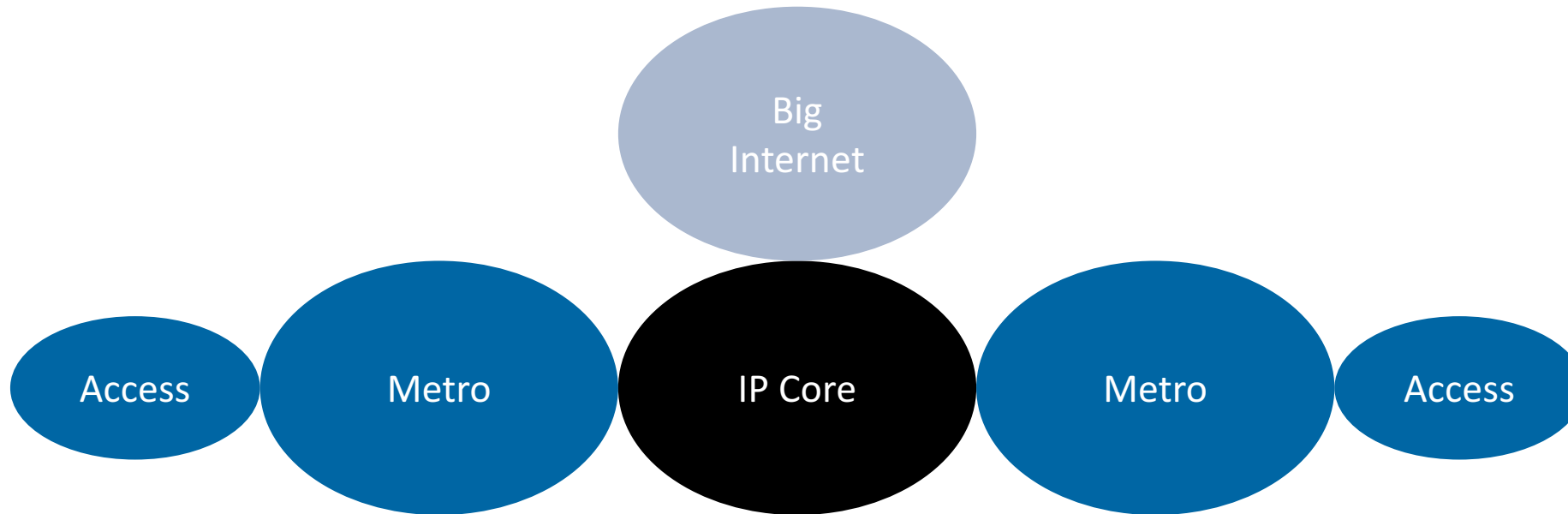
Without simple and efficient traffic engineering, how do you manage this

A Need for a New Architecture

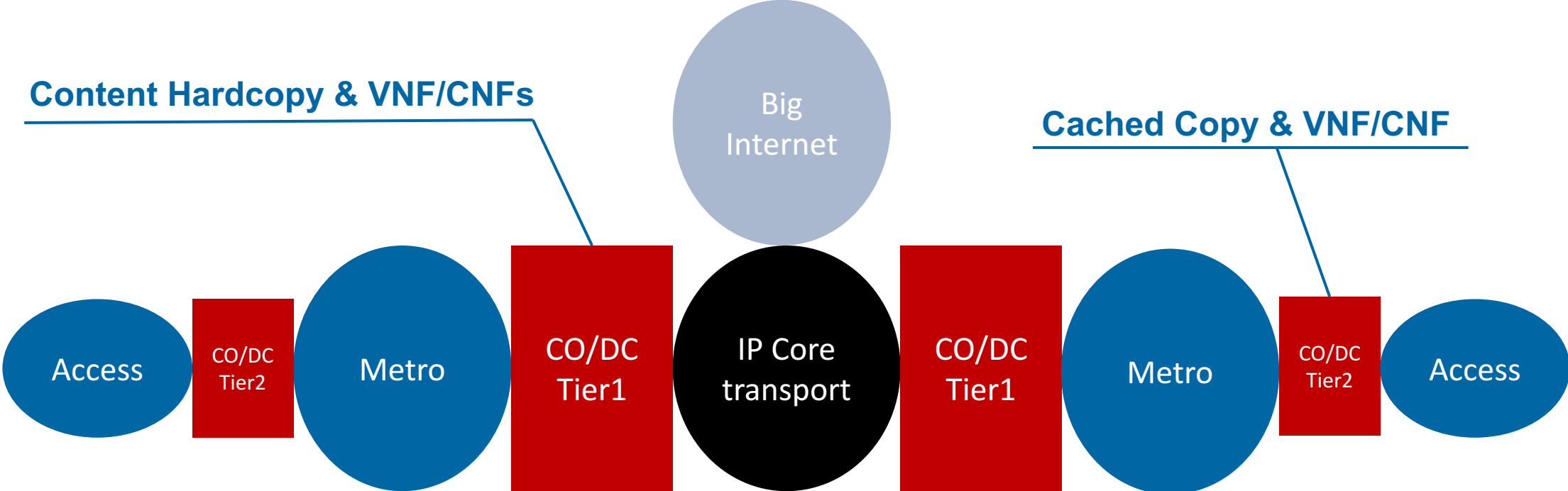
Next Generation Requirements

- Needs to be an industry standard ratified by global standards organizations
- Reusable in the core/WAN, possibly as the glue to bring all the networks together
- Software-programmable
- Leverage new CO/DC greenfield opportunity to try something new
- Provides solutions for both transition and end state
- Interoperability with the brownfield and greenfield
- Implicit ECMP handling

Before - Traditional View



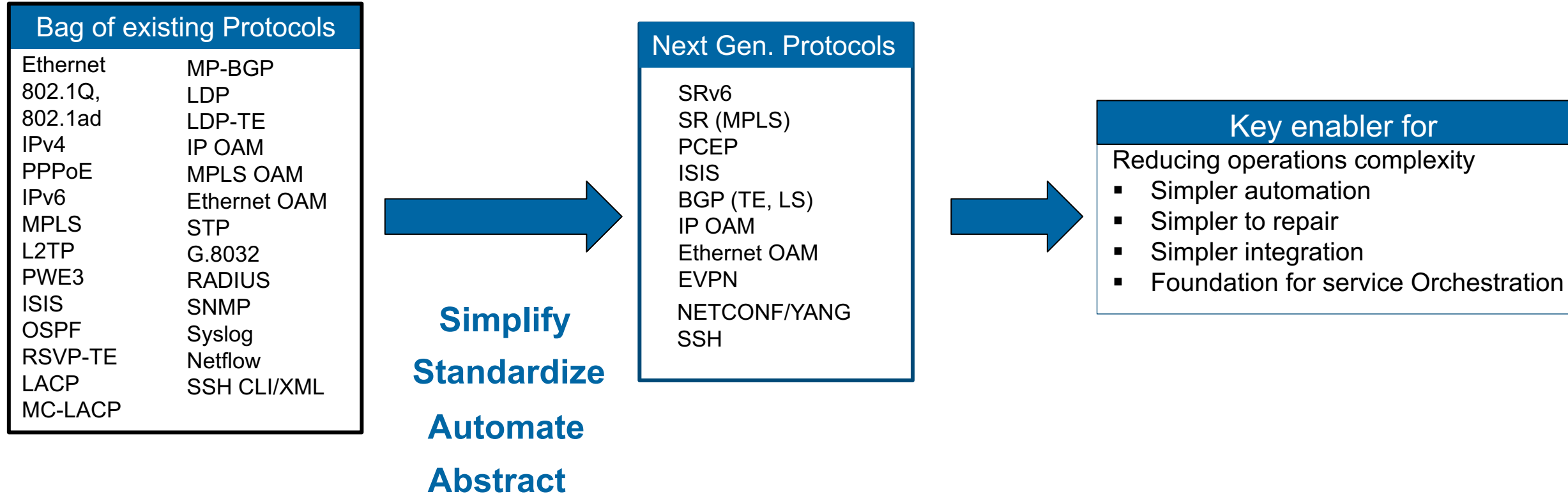
After – Network Transformation



Architecture Central Offices Re-Design for network operators virtualization use cases



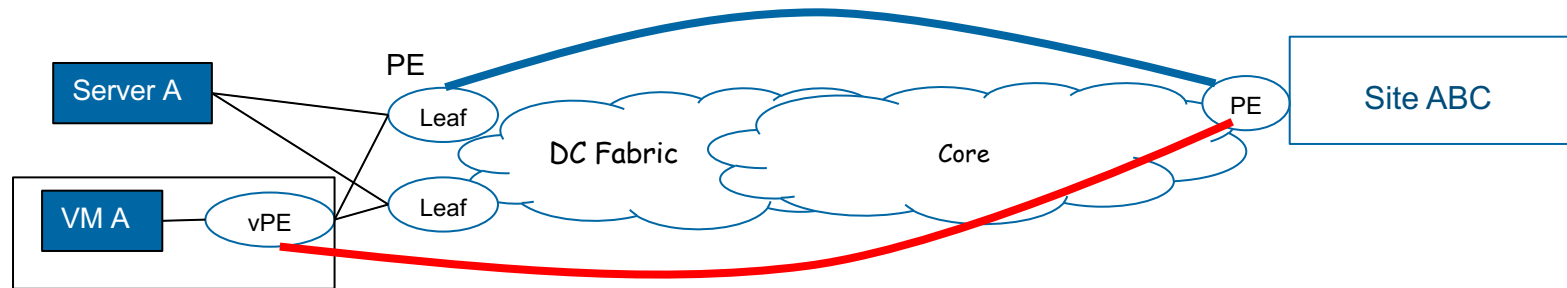
Architecture Change - Drastic Network Protocols Reduction @ Bell



Guiding principle required to accelerate execution and ensure evolution towards Agility

CO/DC – Fabric PE to Core PE – high level

The goal for network transformation is to move the complexity from core transport to the CO/DC and virtualize network components



Leverage existing Data Plane – MPLS E2E

Simplify Control Plane - SR fabric in DC - good starting point

The DC Fabric and Core Network seen as a common IP Network

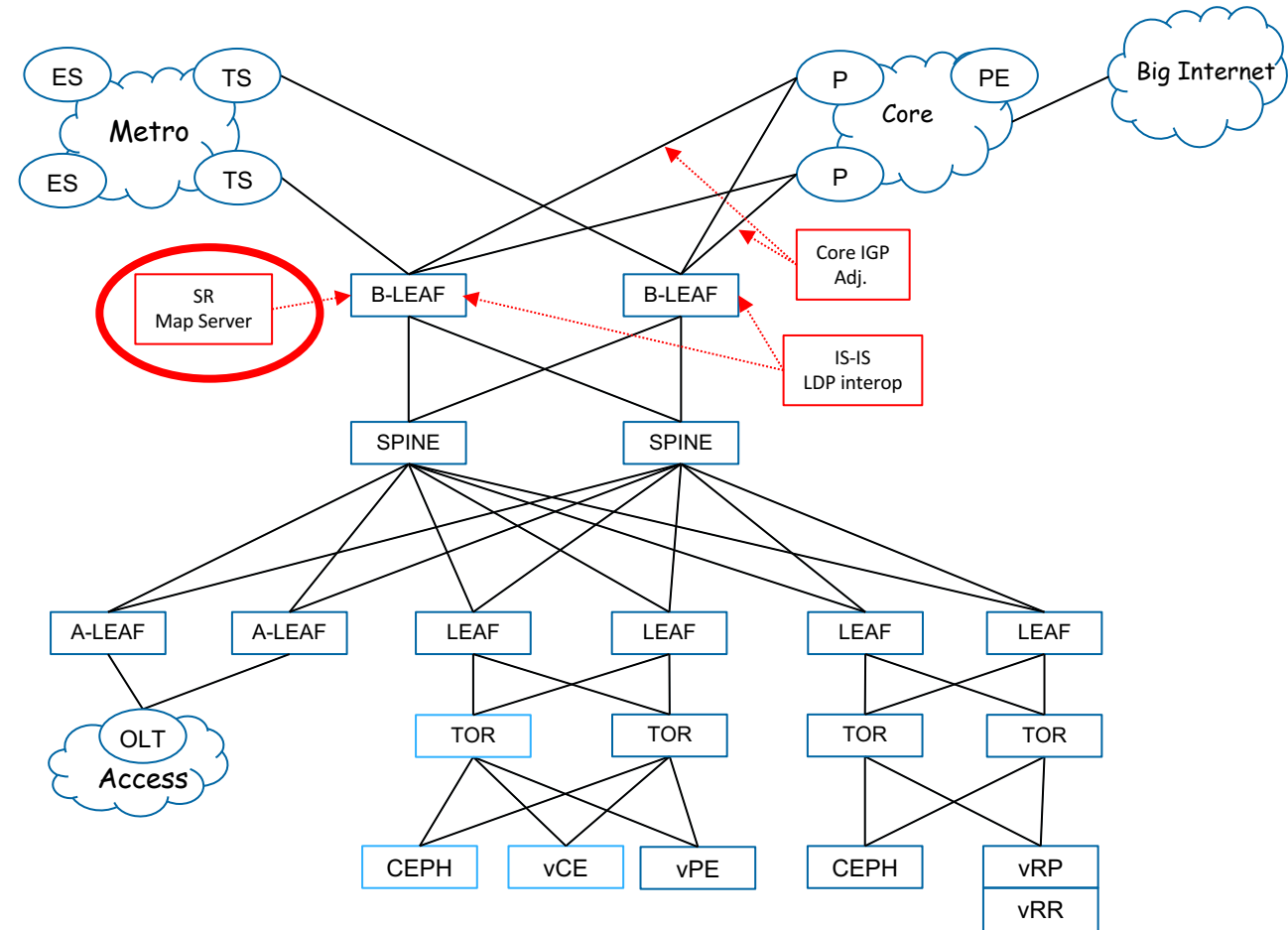
CO/DC – Architecture Overview

Key SR Points

- Fabric underlay is ISIS and SR = SIMPLE
- ECMP & SR for traffic engineering = FLEXIBLE
- SRTE with IP Core network = AGILE
- EVPN Overlay – L2/L3 services = AGILE & SIMPLE

CO/DC Challenges Solved by SR

- Classic DCI overlay is wrong for CO/DC, we need better integration to leverage network assets
- SR Solutions:
 - Map server for interop w/ brownfield LDP PE
 - Dual Stacking of LDP & SR



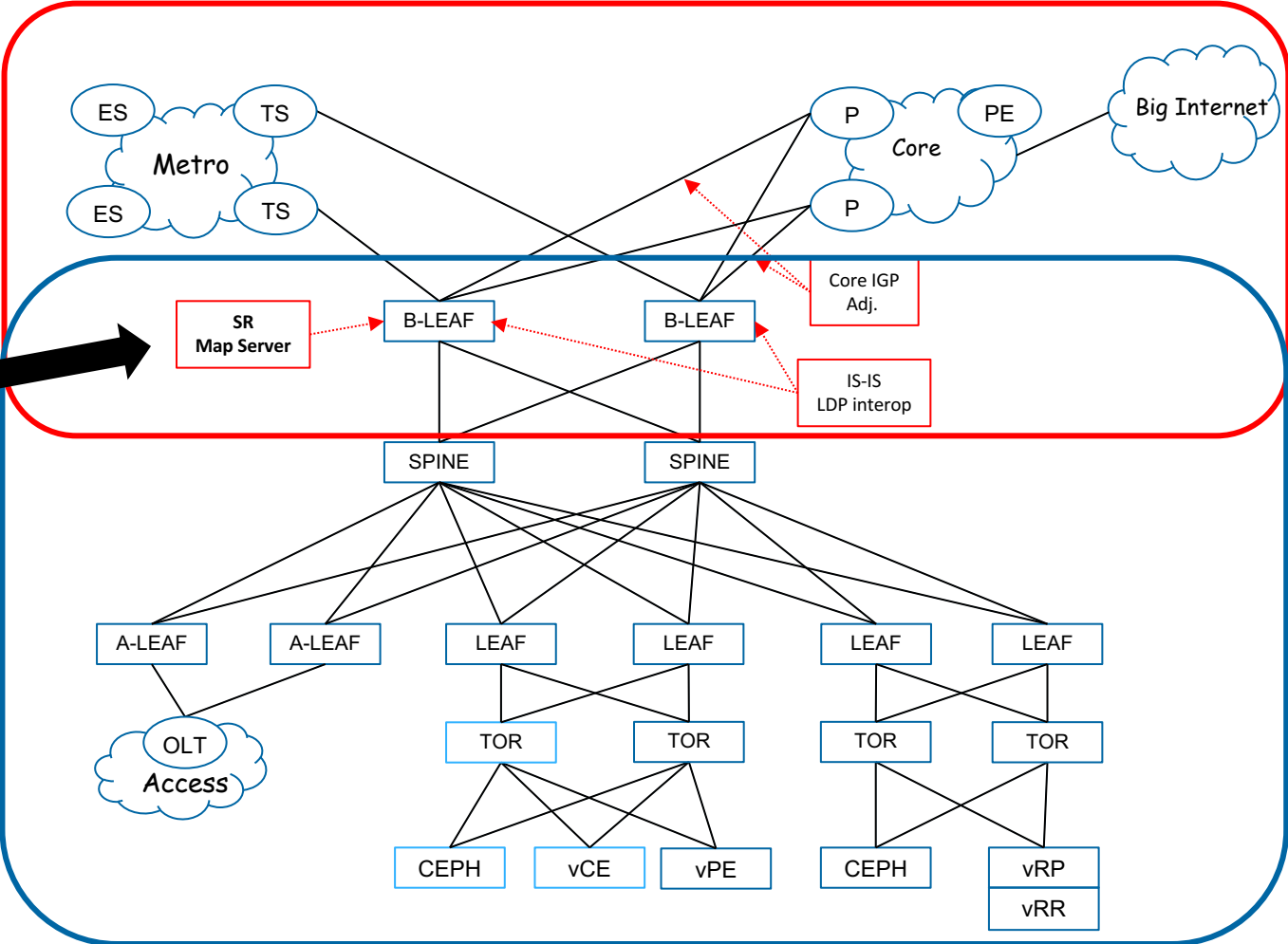
Segment Routing Mapping Server is important for brownfield interaction

CO/DC – SR & LDP intermediate state

Classic MPLS Domain (LDP)

SR – LDP interop using SRMS

SR MPLS Domain



SR Absolute Label Algorithm

- Segment Routing Mapping Servers & SR Allocation
 - SRMS is critical to any SR deployment with brownfield interop
 - Reuse of the SRMS algorithm to assign Label in the SR Domain
 - SR Label are then assigned with the IP loopback processes
- Plan the SR domains per label range
 - Use of full SRGB block: 65k
 - IP Core 8k block
 - CO/DC – Tier 1: 4k block
 - CO/DC – Tier 2: 2k Block

To ensure SR uniqueness across all domains, we came up with the following SR Absolute Label Algorithm

SRGB_Base* + (first-SID-Index [Infra_underlay|IPVPN|Internet] + loopback-last-octet) = SR absolute Label

*The SRGB_Base start at 16k

Learnings – SR Allocation example with absolute label

For a given CO/DC_A with the following loopback address, 209.71.196.15/32

Absolute Label

SRGB_Base (16000) + first-SID-index (10512) + loopback-last-octet (15) = SR Absolute Label (26527)

```
RP/0/RSP0/CPU0:router#show mpls forwarding prefix 209.71.196.15/32
Local   Outgoing   Prefix           Outgoing   Next Hop        Bytes
Label   Label      or ID            Interface  Next Hop        Switched
-----  -
26527   26527      SR Pfx (idx 10527) Hu0/3/0/2   10.55.65.41    0
26527   26527      SR Pfx (idx 10527) Hu0/6/0/2   10.55.65.45    0
```

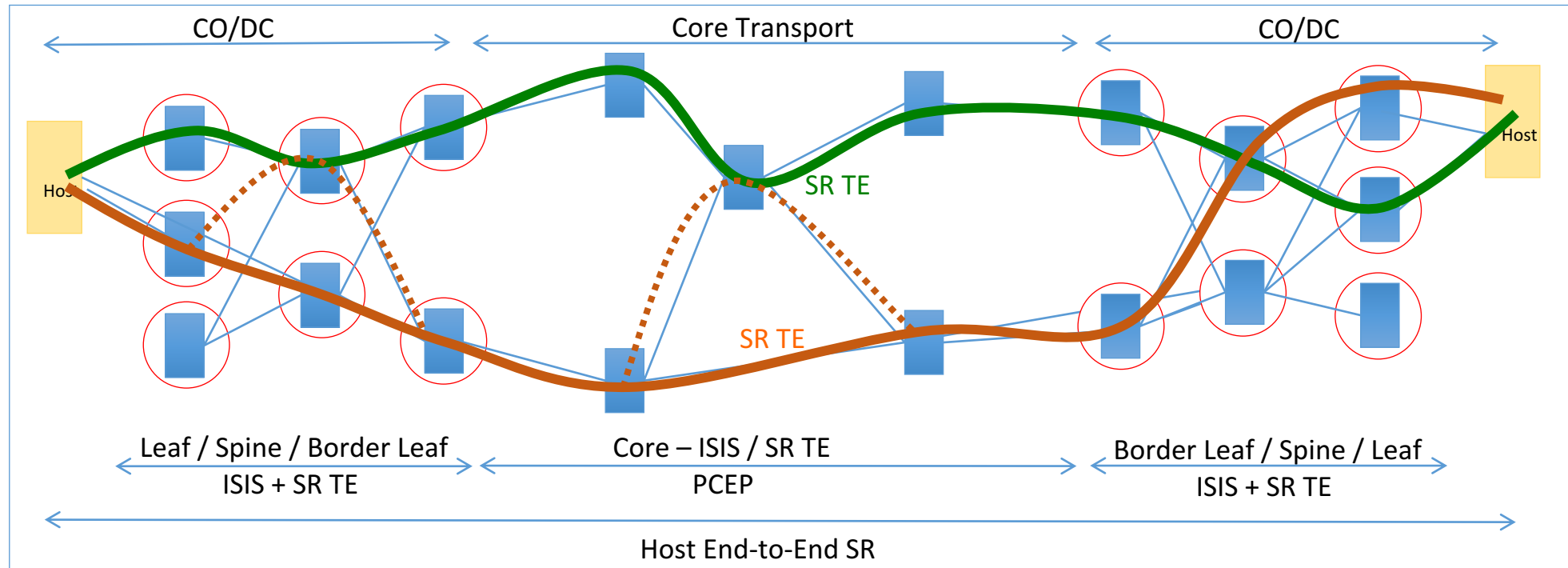
Prefix-SID only

first-SID-index (10512) + loopback-last-octet (15) = SR Prefix-SID Index (10527)

```
router isis CO/DC_A
interface Loopback0
address-family ipv4 unicast
prefix-sid index 10527
```

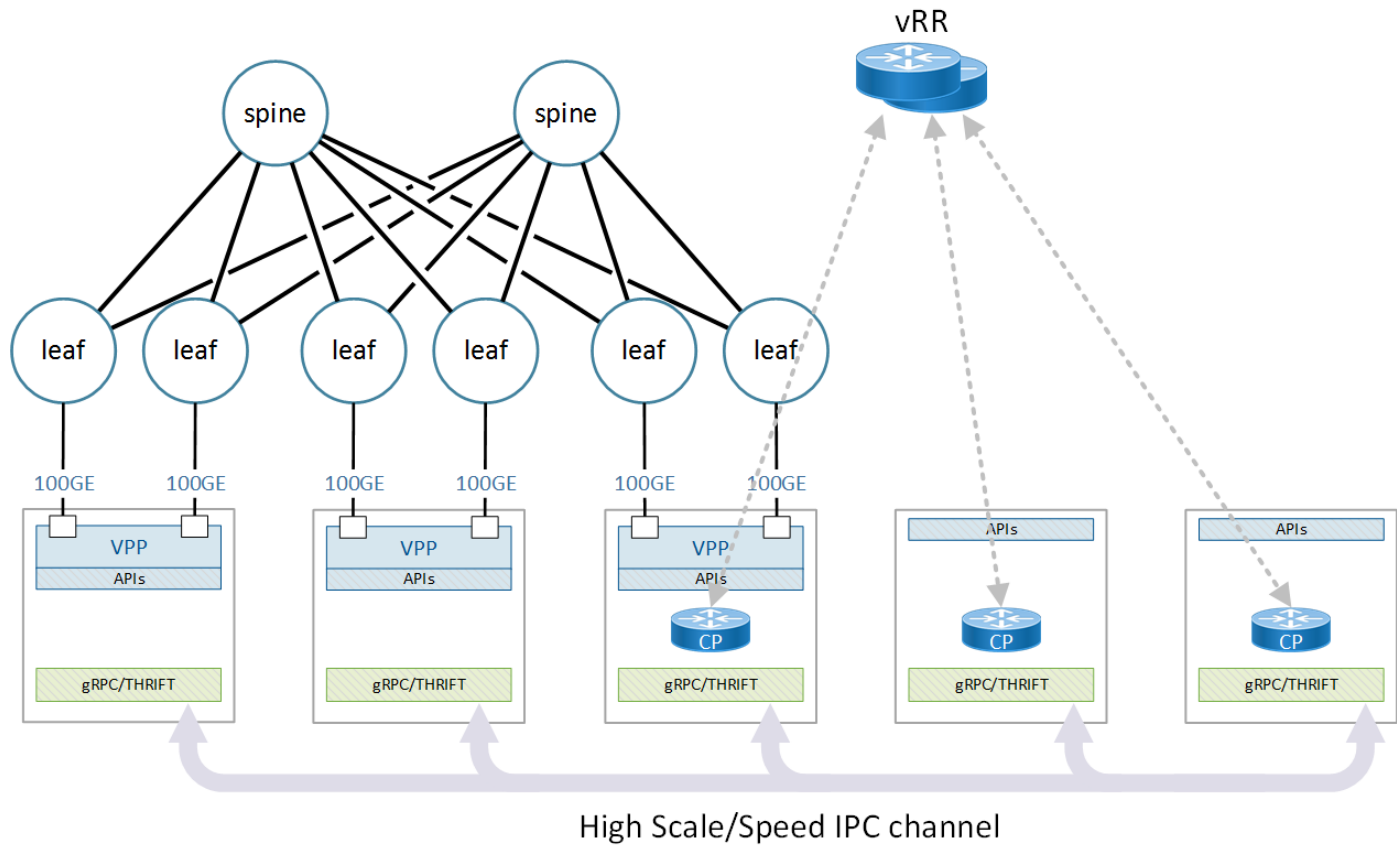
Next Steps

- Segment Routing from the host (HV, vPE, kernel, etc.)
 - Expand controller: SR-TE in DC, On-Demand Next-Hop
- SR _between_ DC's (core transformation to SR)



Application-responsive networking

Use-Case: Scaled-Out vPE



Build a highly-distributed vPE function that scales linearly with the CO/DC fabric.

- Limited East-West Traffic
 - Hypervisor is the new *edge*
 - Same protocol stack
 - Avoid the “Big Fat VNF”
-
- Follow *network disaggregation* principles and build using open, modular, replaceable components.
 - Same design principles can be applied to other high-throughput VNFs

Learnings

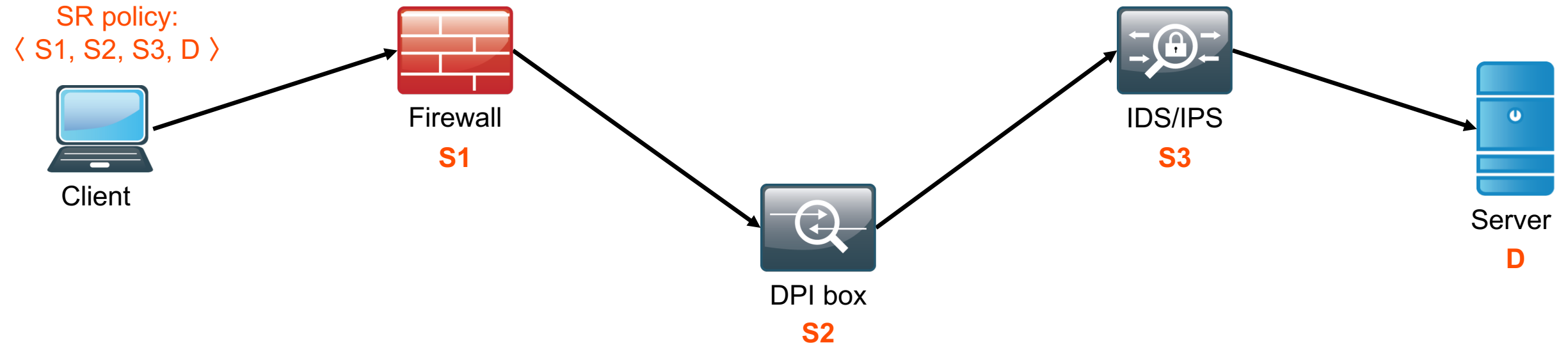
- **Simplicity wins everywhere**
 - Capture latent network value – leverage existing physical assets with efficient on-demand TE
 - Reduction CAPEX/OPEX and increase Agility
 - A lot of legacy protocols can be removed (LDP, RSVP-TE, etc.)
 - Make Engineering/Ops happy (deterministic labels, label reuse)
- **Start small, find a greenfield island to introduce new technologies**
- **The hard part is the brownfield transition, be careful**
 - SR vs Non-SR Nodes Interop/Integration
 - SR has lots of options!
- **SRGB planning is important**
 - In our case, we chose to allocate 64K labels to SR instead of default 8K (LOTS of VM's!)
- **Work with industry standards**
 - Keep the vendors honest

Do a lot more, with a lot less

Use Cases – SRv6

SRv6 - NFV - Service Chaining

Packets from are steered through a sequence of services on their way to the server.

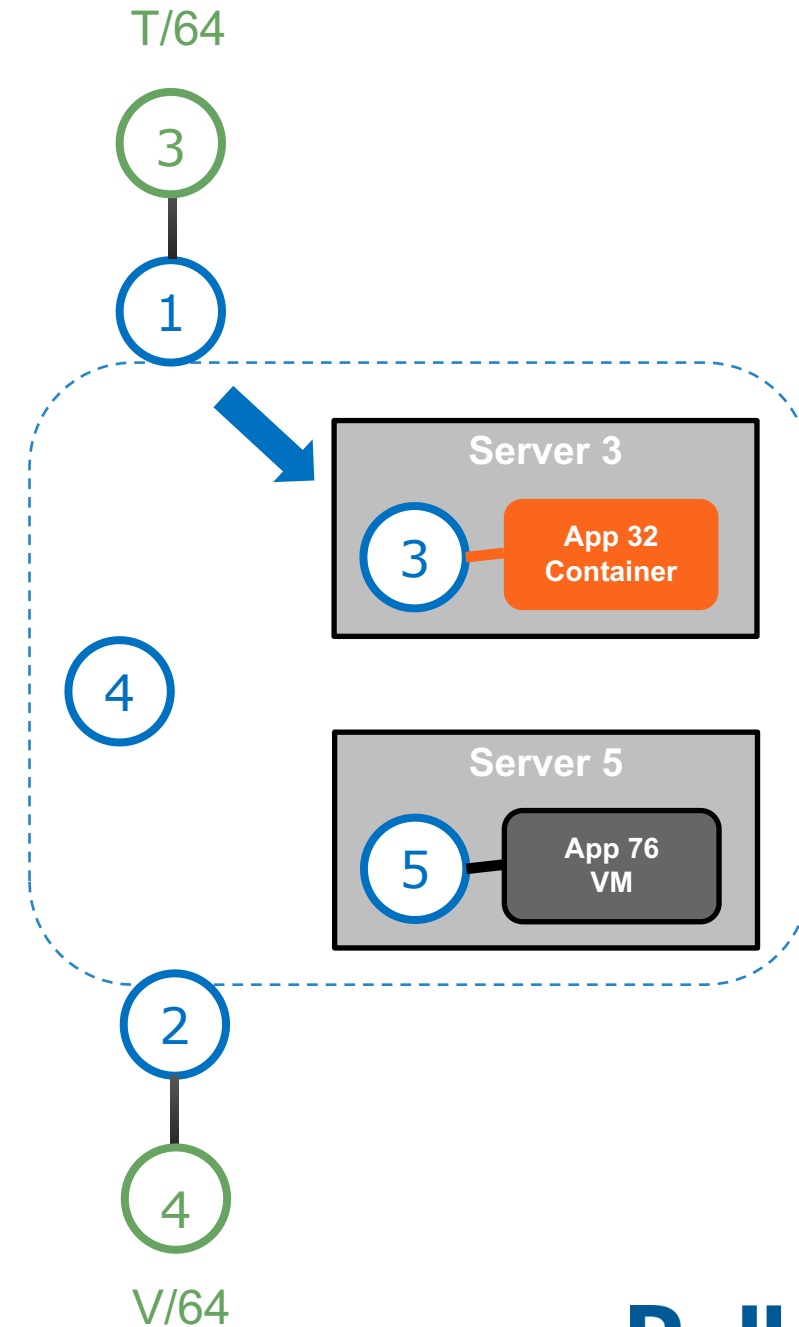
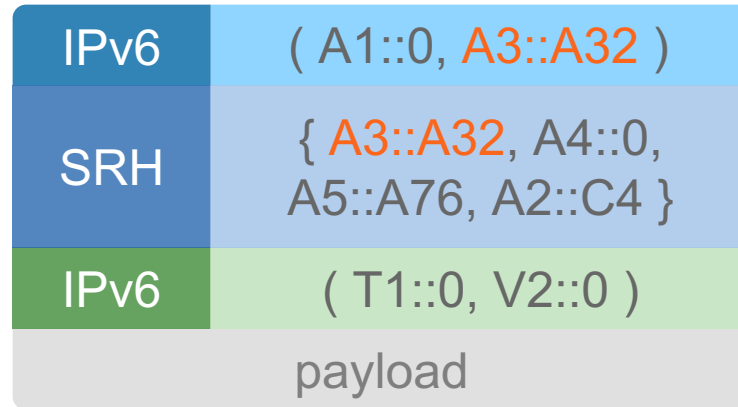
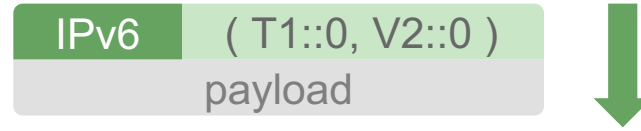


- **Segment Routing service chaining:** services are expressed with segments

- Flexible
- Scalable
- Stateless

SRv6 - Integrated NFV

- A3::A32 means
 - App in Container 32
 - @ node A3::/64
- Stateless
 - NSH creates per-chain state in the fabric
 - SR does not
- App is SR aware or not

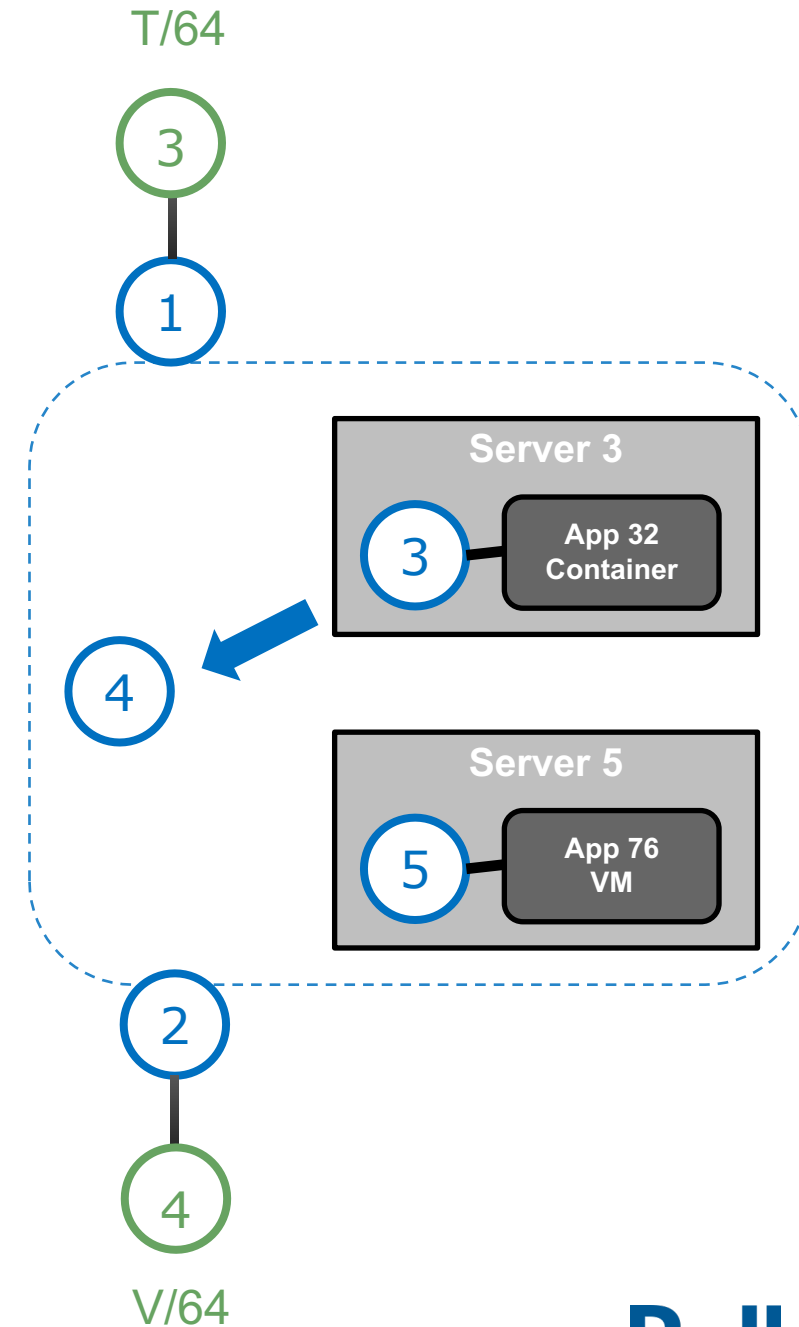


Inner header could also be IPv4 instead of IPv6

SRv6 - Integrated NFV

- Integrated with underlay SLA

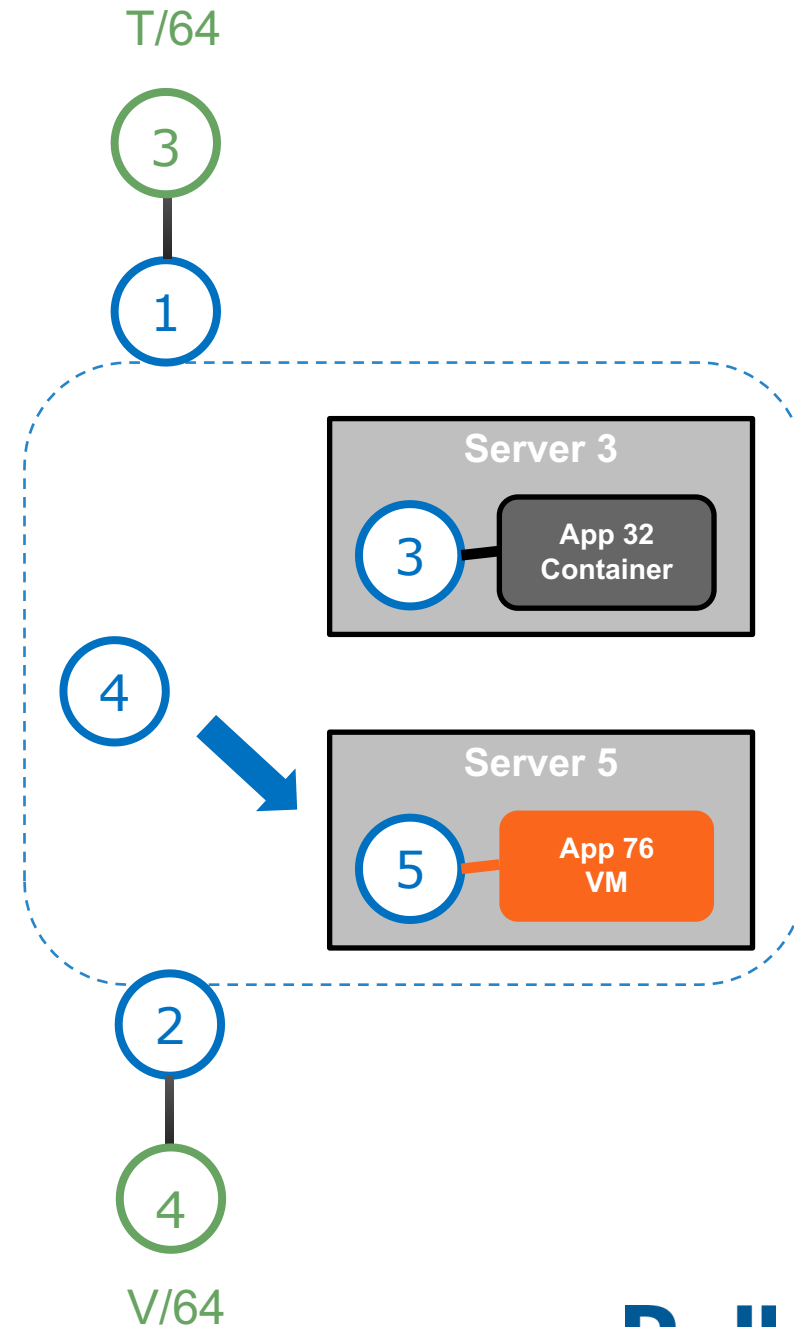
IPv6	(A1::0, A4::0)
SRH	{ A3::A32, A4::0, A5::A76, A2::C4 }
IPv6	(T1::0, V2::0)
payload	



SRv6 - Integrated NFV

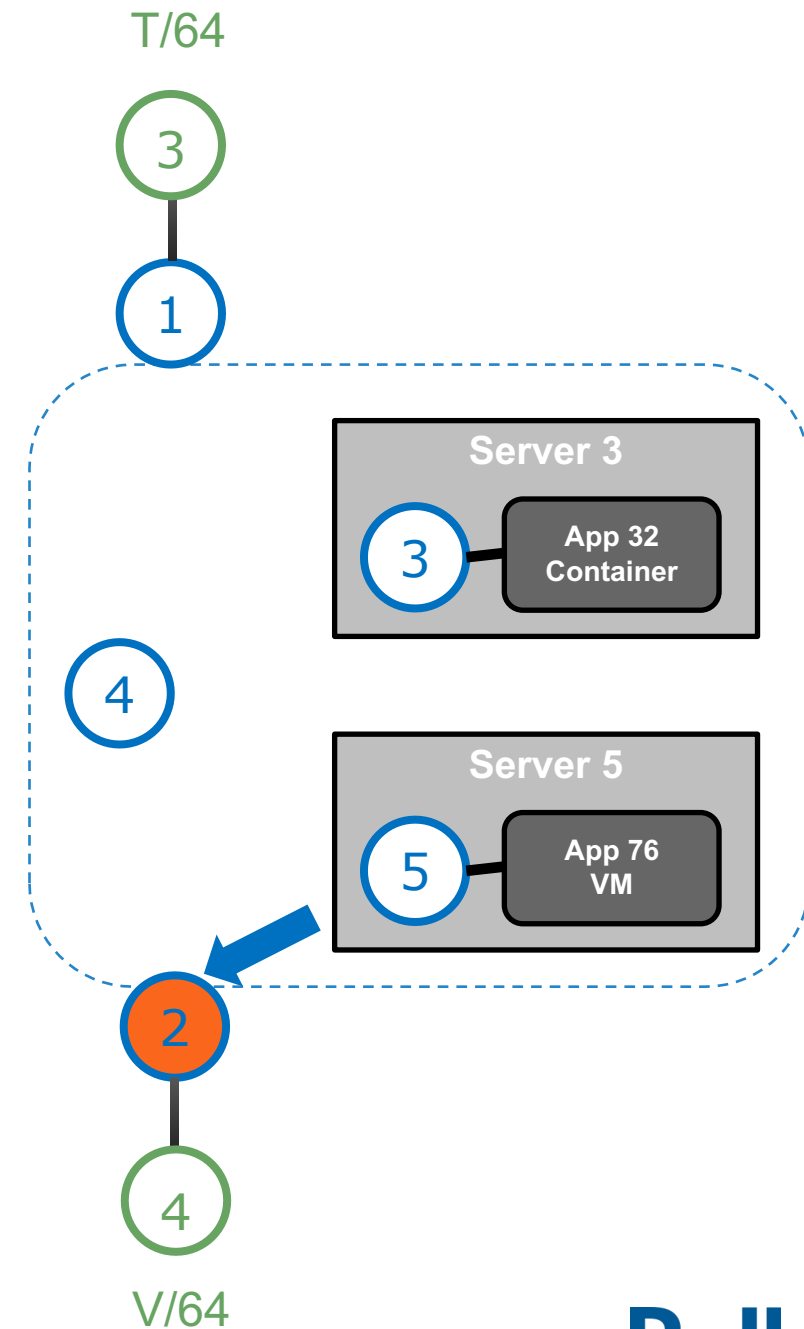
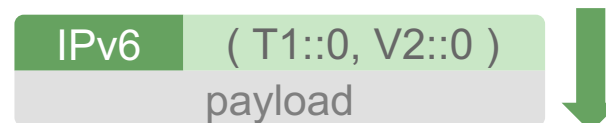
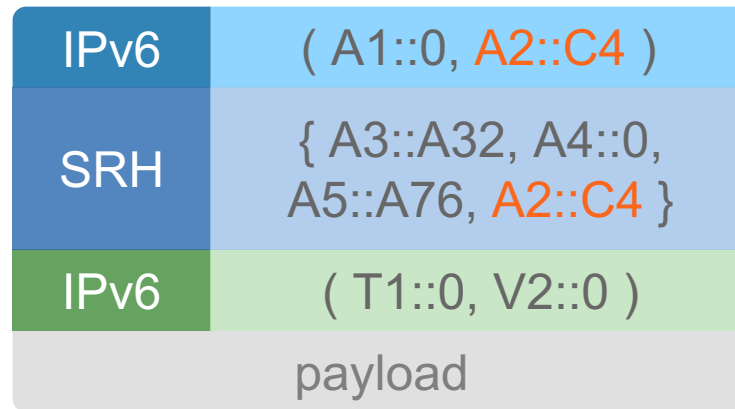
- A5::A76 means
 - App in VM 76
 - @ node A5::/64
- Stateless
 - NSH creates per-chain state in the fabric
 - SR does not
- App is SR aware or not

IPv6	(A1::0, A5::A76)
SRH	{ A3::A32, A4::0, A5::A76, A2::C4 }
IPv6	(T1::0, V2::0)
payload	



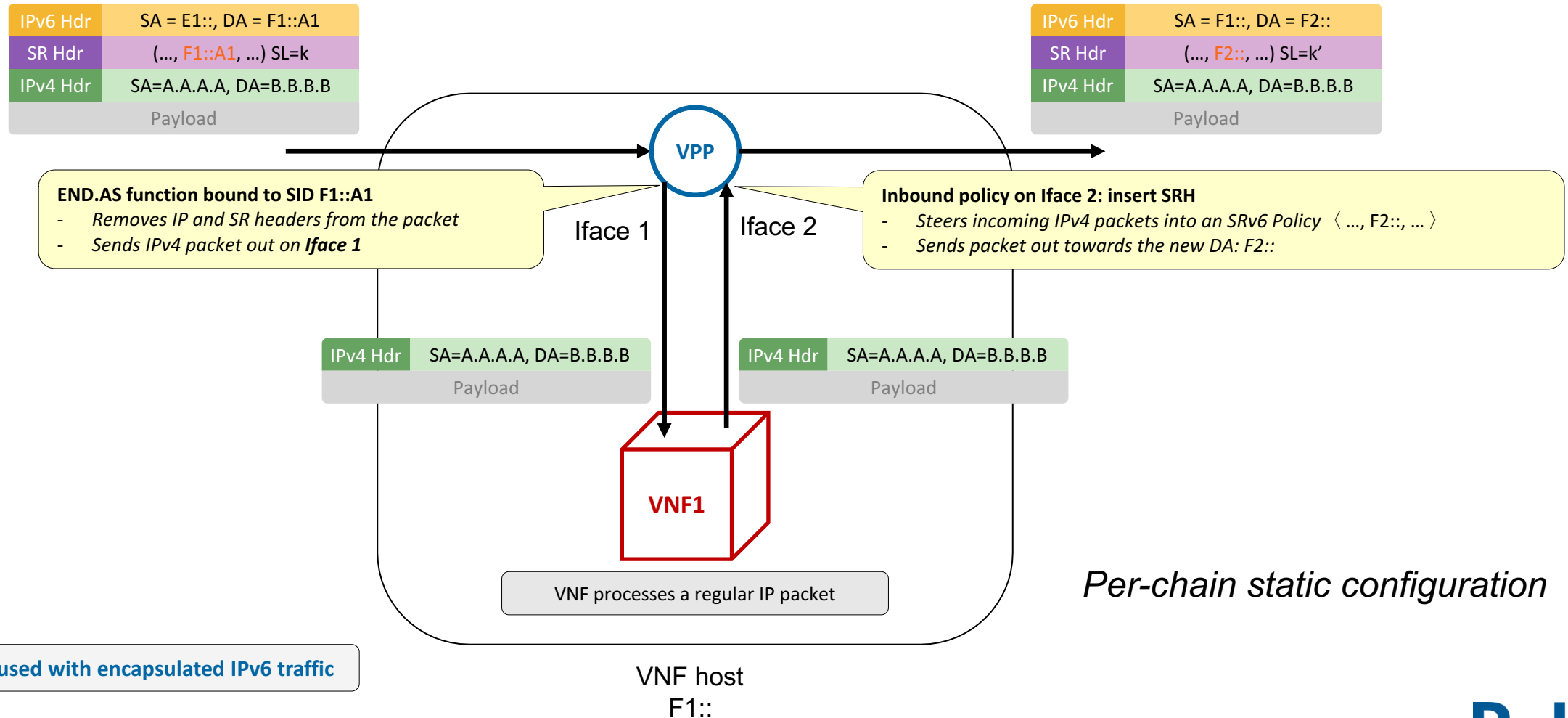
SRv6 - Integrated NFV

- Integrated with Overlay



NFV – SRv6 Function - END.AS – Static proxy

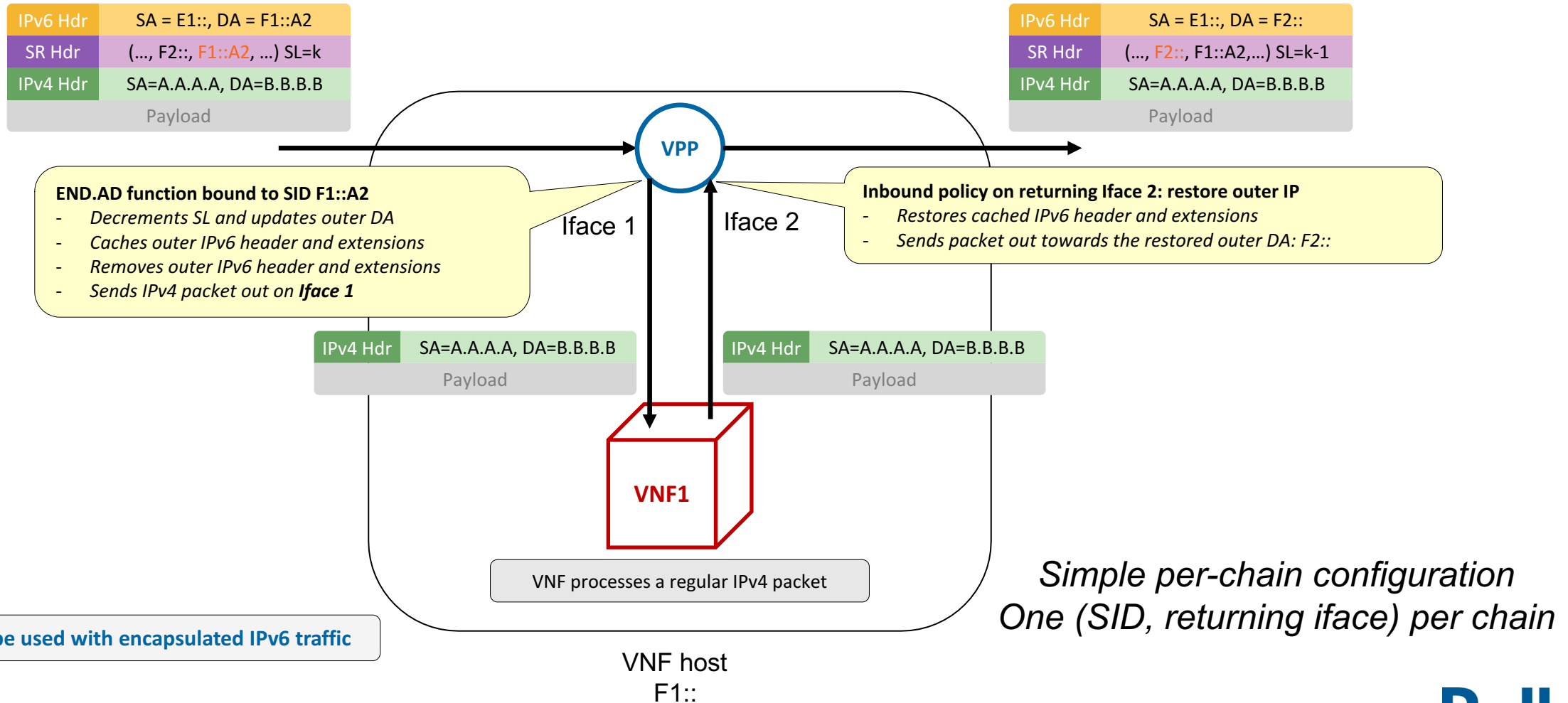
Endpoint to SR-unaware APP via static proxy



END.AS can also be used with encapsulated IPv6 traffic

NFV SRv6 Function - END.AD – Dynamic proxy

Endpoint to SR-unaware APP via dynamic proxy



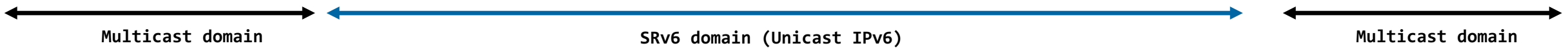
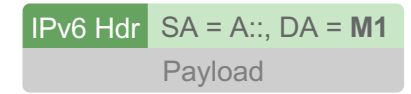
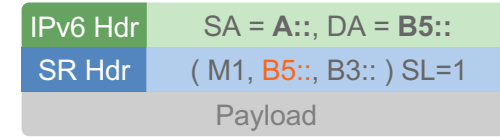
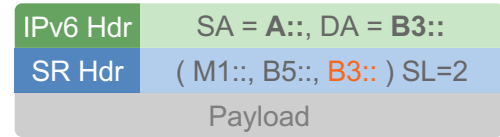
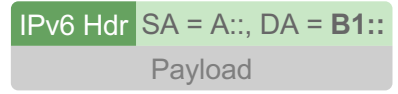
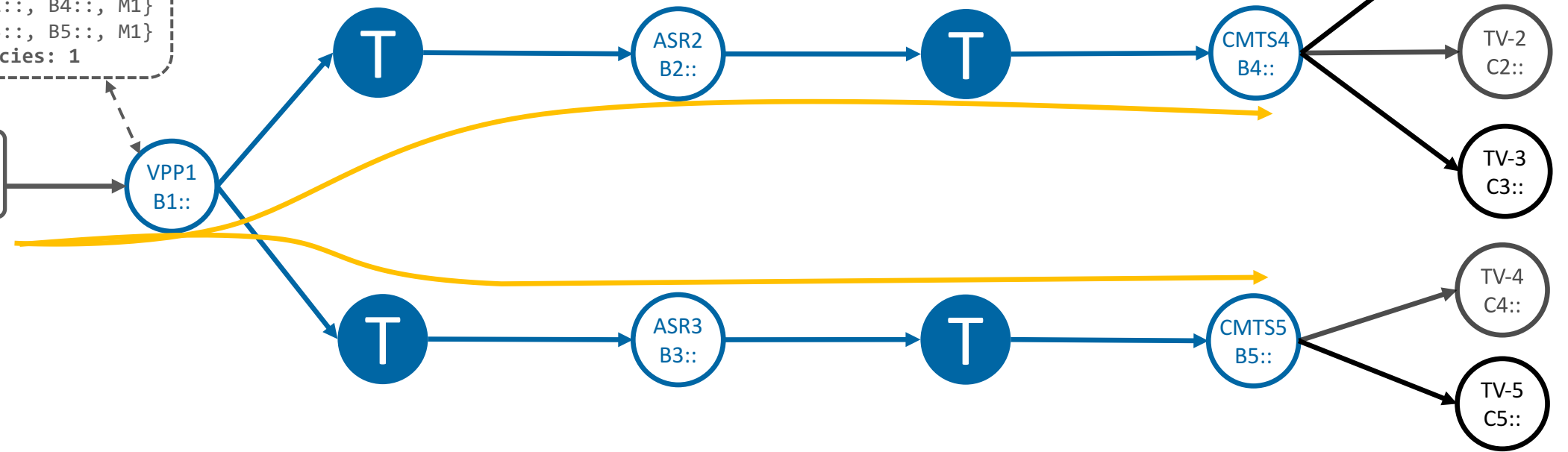
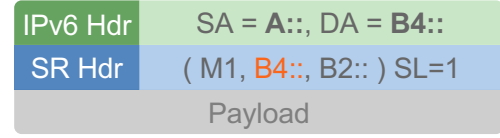
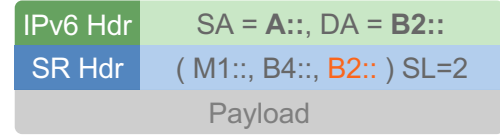
SRv6 - Spray use-case

TV2 and TV4 are **not** subscribed to this channel (M1) and do **not** receive the content

```
>VPP1: show sr spray policies
In_iface      SR Spray Policy
GE0/5/0       {B2::, B4::, M1}
               {B3::, B5::, M1}
Total SR spray policies: 1
```

Content App
A::

VPP1
B1::



Flexible, SLA-enabled and Efficient content injection without multicast core



See the Demo

Thank You