Introduction to SRv6 Technology

Network Simplicity, Use Case and Ecosystem

錢小山
首席技術顧問
思科大中華區數據中心架構事業部

二〇二〇年九月

Segment Routing

Source Routing

- The topological and service (NFV) path is encoded in packet header

Scalability

- The network fabric does not hold any per-flow state for TE or NFV

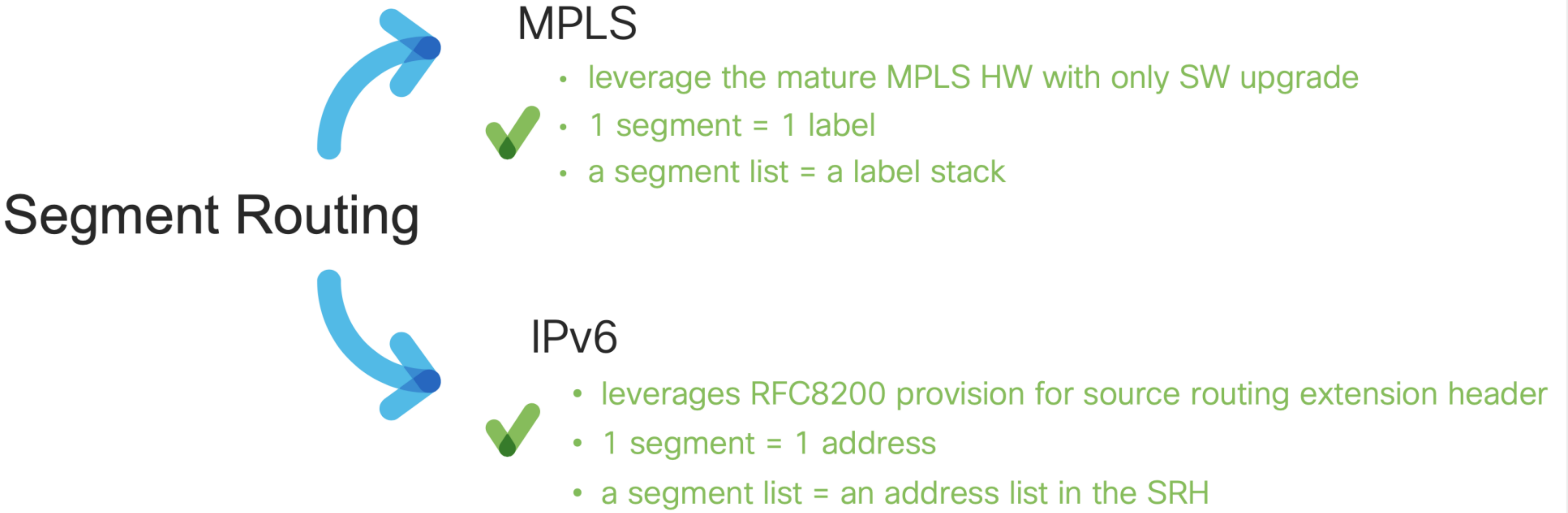
Simplicity

- Automation: TILFA sub-50msec FRR
- Protocol elimination: LDP, RSVP-TE, VxLAN, NSH, GTP, ...

End-to-End

- DC, Metro, WAN

Two Data Plane Instantiations



Segment Routing

MPLS

✓

- leverage the mature MPLS HW with only SW upgrade
- 1 segment = 1 label
- a segment list = a label stack

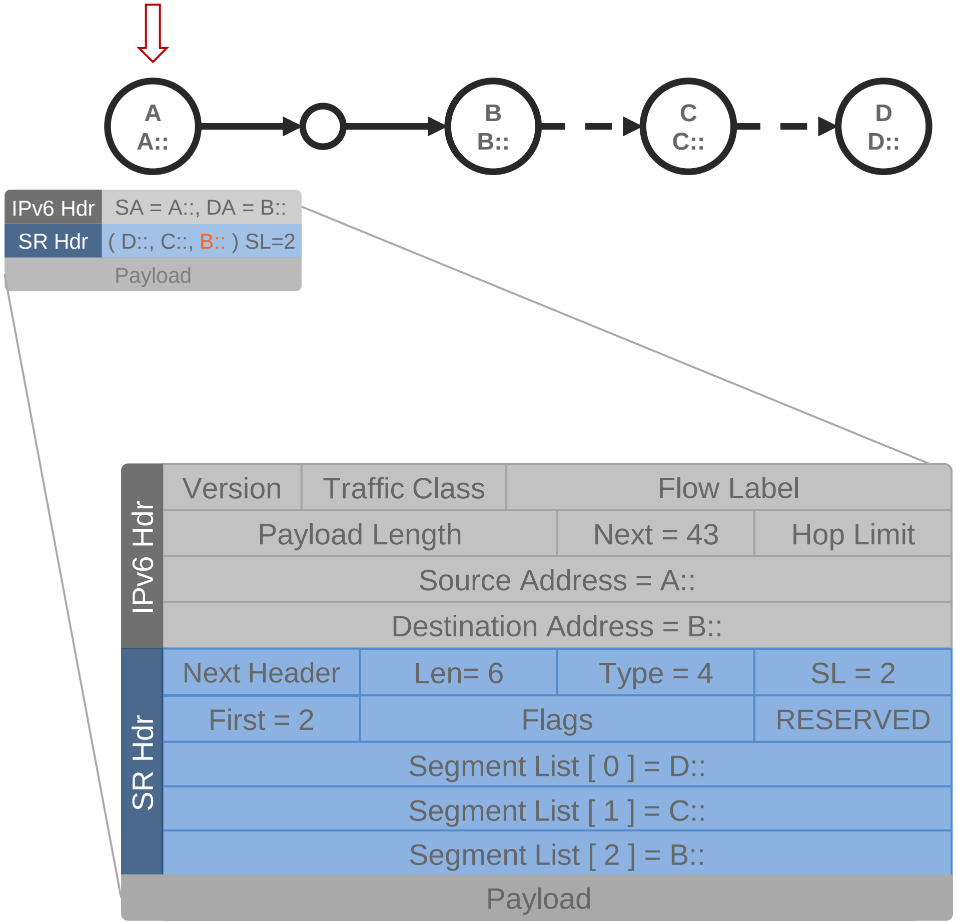
IPv6

✓

- leverages RFC8200 provision for source routing extension header
- 1 segment = 1 address
- a segment list = an address list in the SRH

Source Node

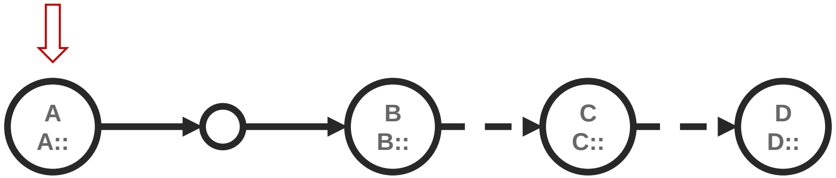
- Source node is SR-capable
- SR Header (SRH) is created with
  - Segment list in reversed order of the path
    - Segment List [0] is the LAST segment
    - Segment List [$n-1$] is the FIRST segment
  - Segments Left is set to $n-1$
  - First Segment is set to $n-1$
- IP DA is set to the first segment
- Packet is send according to the IP DA
  - Normal IPv6 forwarding



IPv6 Hdr	SA = A::, DA = B::
SR Hdr	(D::, C::, B::) SL=2
Payload	

<table>
<tr><td rowspan="4">IPv6 Hdr</td><td>Version</td><td colspan="2">Traffic Class</td><td colspan="3">Flow Label</td></tr>
<tr><td colspan="4">Payload Length</td><td>Next = 43</td><td>Hop Limit</td></tr>
<tr><td colspan="6">Source Address = A::</td></tr>
<tr><td colspan="6">Destination Address = B::</td></tr>
<tr><td rowspan="5">SR Hdr</td><td colspan="2">Next Header</td><td colspan="2">Len= 6</td><td>Type = 4</td><td>SL = 2</td></tr>
<tr><td colspan="2">First = 2</td><td colspan="3">Flags</td><td>RESERVED</td></tr>
<tr><td colspan="6">Segment List [0] = D::</td></tr>
<tr><td colspan="6">Segment List [1] = C::</td></tr>
<tr><td colspan="6">Segment List [2] = B::</td></tr>
<tr><td colspan="7">Payload</td></tr>
</table>

Non-SR Transit Node

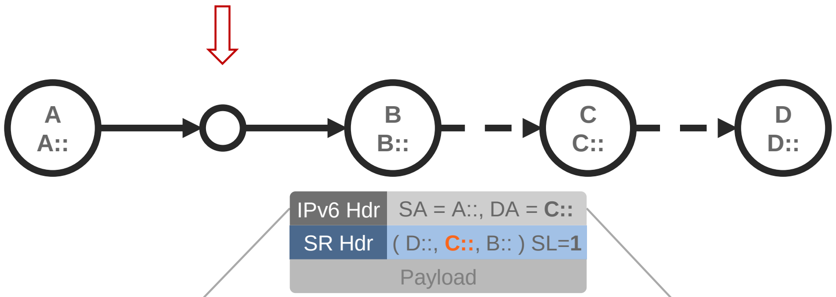


IPv6 Hdr	SA = A::, DA = B::
SR Hdr	(D::, C::, B::) SL=2
Payload	

- Plain IPv6 forwarding
- Solely based on IPv6 DA
- No SRH inspection or update

SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA



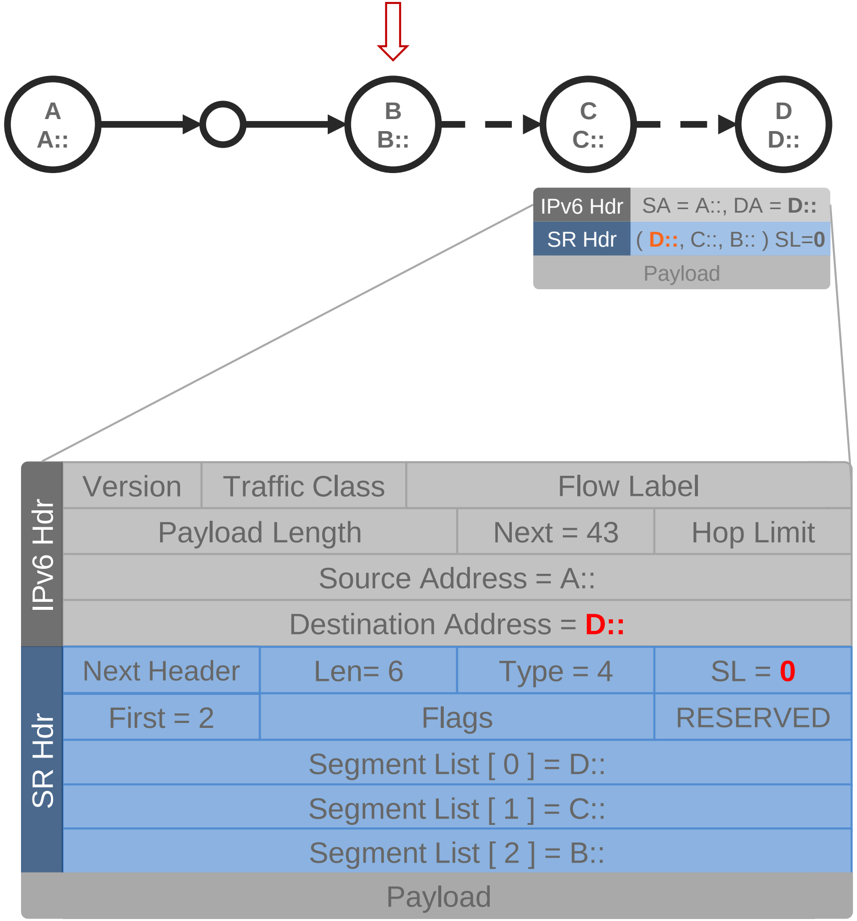
IPv6 Hdr	Version	Traffic Class	Flow Label	
	Payload Length		Next = 43	Hop Limit
	Source Address = A::			
	Destination Address = C::			
SR Hdr	Next Header	Len= 6	Type = 4	SL = 1
	First = 2	Flags		RESERVED
	Segment List [0] = D::			
	Segment List [1] = C::			
	Segment List [2] = B::			
Payload				

SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA
  - ELSE (Segments Left = 0)
    - Remove the IP and SR header
    - Process the payload:
      - Inner IP: Lookup DA and forward
      - TCP / UDP: Send to socket
      - …

Standard IPv6 processing

The final destination does not have to be SR-capable.



IPv6 Hdr | SA = A::, DA = D::

SR Hdr | (D::, C::, B::) SL=0

Payload

IPv6 Hdr

Version | Traffic Class | Flow Label

Payload Length | Next = 43 | Hop Limit

Source Address = A::

Destination Address = D::

SR Hdr

Next Header | Len= 6 | Type = 4 | SL = 0

First = 2 | Flags | RESERVED

Segment List [0] = D::

Segment List [1] = C::

Segment List [2] = B::

Payload

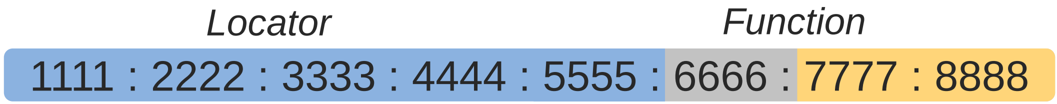
Introduction to SRv6 Technology

Network Simplicity, Use Case and Ecosystem

錢小山
首席技術顧問
思科大中華區數據中心架構事業部

二〇二〇年九月

Segment format



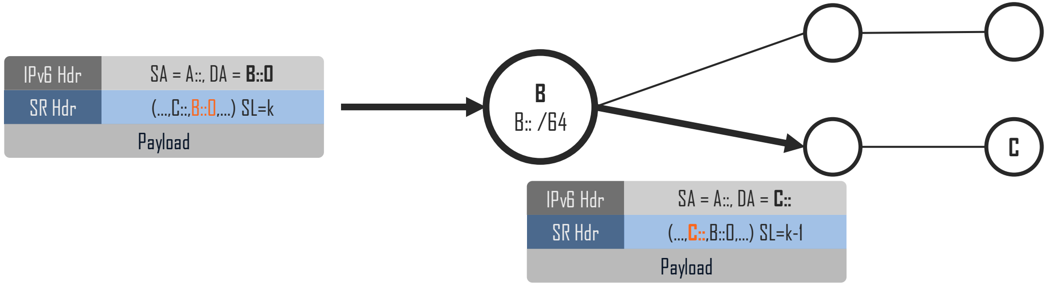
Locator *Function*

1111 : 2222 : 3333 : 4444 : 5555 : 6666 : 7777 : 8888

- SRv6 SIDs are 128-bit addresses
  - **Locator**: most significant bits are used to **route** the segment to its **parent node**
  - **Function**: least significant bits identify the **action** to be performed on the **parent node**
    - **Argument** [optional]: Last bits can be used as a local function argument
- Flexible bit-length allocation
  - Segment format is **local knowledge** on the parent node
- SIDs have to be **specifically enabled** as such on their parent node
  - A local address **is not** by default a local SID
  - A local SID does not have to be associated with an interface

END – Default endpoint

- *Default endpoint* behavior (node segment)
  - Decrement Segments Left, update DA
  - Forward according to new DA
- Node B advertises prefix B::/64 (B::/64 is the SID **locator**)
  - Packets are forwarded to B along the default routes (shortest path)
- On B, the *default endpoint* behavior is associated with ID 0 (0 is the **function**)
- The SID corresponding to the *default endpoint* behavior on node B is B::0

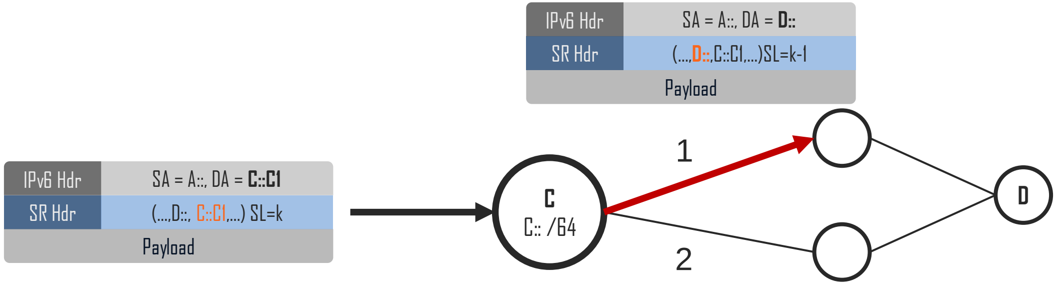


IPv6 Hdr	SA = A::, DA = **B::0**
SR Hdr	(....,C::,B::0,....) SL=k
Payload	

IPv6 Hdr	SA = A::, DA = **C::**
SR Hdr	(....,**C::**,B::0,....) SL=k-1
Payload	

END.X – Endpoint then Xconnect

- *Endpoint xconnect* behavior (adjacency segment)
  - Decrement Segments Left, update DA
  - **Forward on the interface associated with the Xconnect segment**
- Node C advertises prefix C::/64
  - Packets are forwarded to C along the default routes (shortest path)
- On C, the *endpoint xconnect* behavior for link (C, E) is associated with ID CE
- The SID corresponding to *endpoint xconnect-(C,E)* behavior on node C is **C::CE**

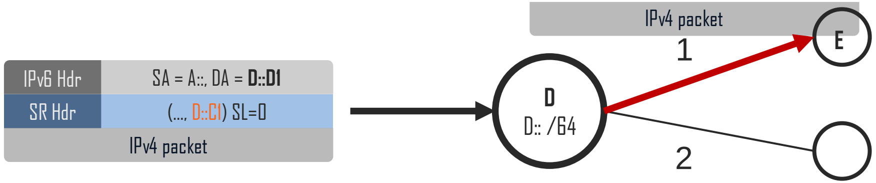


IPv6 Hdr	SA = A::, DA = **C::C1**
SR Hdr	(....,D::, C::C1,....) SL=k
Payload	

IPv6 Hdr	SA = A::, DA = **D::**
SR Hdr	(....,D::,C::C1,....)SL=k-1
Payload	

END.DX4 – Endpoint with Decapsulation and Xconnect (END.DX6, END.DX2)

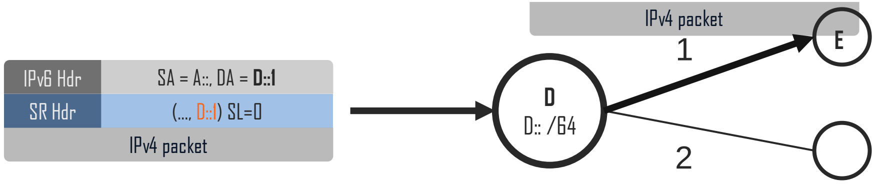
- *Endpoint xconnect* behavior (adjacency segment)
  - Segments Left must be 0
  - NH must be IPv4 (or IPv6 or L2)
  - Decapsulate inner packet
  - **Forward on the interface associated with the Xconnect**
- Node D advertises prefix C::/64
  - Packets are forwarded to D along the default routes (shortest path)
- On D, the *endpoint xconnect* behavior for link (D, E) is associated with ID DE
- It is like L3 VPN with per CE label allocation



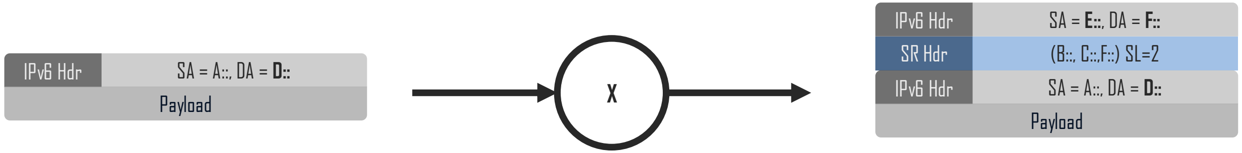
IPv6 Hdr	SA = A::, DA = **D::D1**
SR Hdr	(…, D::C1) SL=0
IPv4 packet	

END.DT4 – Endpoint with Decapsulation and Table lookup(END.DT6)

- *Endpoint xconnect* behavior (adjacency segment)
  - Segments Left must be 0
  - NH must be IPv4 (or IPv6)
  - Decapsulate inner packet
  - **Do the lookup for IPv4 destination of inner packet and forward accordingly**
- Node D advertises prefix D::/64
  - Packets are forwarded to D along the default routes (shortest path)
- On D, the *endpoint* behavior for link (D, E) is associated with ID DE
- It is like L3 VPN with per VRF label allocation



Transit behavior T.ENCAP and T.INSERT



IPv6 Hdr SA = A::, DA = **D::**

Payload

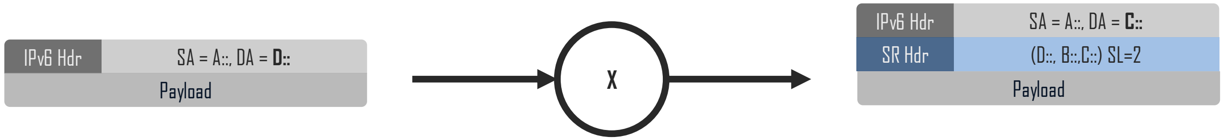
X

IPv6 Hdr SA = **E::**, DA = **F::**

SR Hdr (B::, C::,F::) SL=2

IPv6 Hdr SA = A::, DA = **D::**

Payload



IPv6 Hdr SA = A::, DA = **D::**

Payload

X

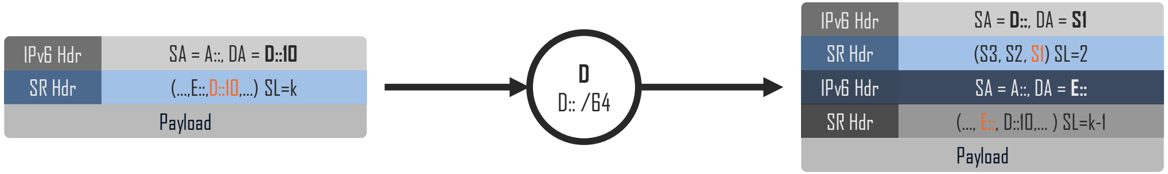
IPv6 Hdr SA = A::, DA = **C::**

SR Hdr (D::, B::,C::) SL=2

Payload

END.B6.ENCAPS – IPv6 Binding Segment (encap)

- *IPv6 binding segment*
  - Decrement Segments Left, update DA
  - **Push outer IP and SR headers associated with the binding segment**
  - Forward according to outer header DA (first segment of the new SRH)
- Node D advertises prefix D::/64
- The SR *encaps* policy (SA = D::, SL = 〈 S1, S2, S3 〉) is associated with ID 10
- The corresponding *binding SID* is **D::10**

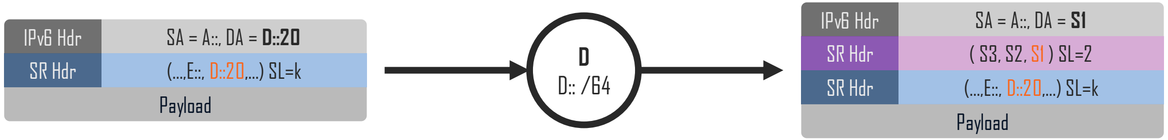


IPv6 Hdr	SA = A::, DA = **D::10**
SR Hdr	(…,E::,D::10,…) SL=k
Payload	

IPv6 Hdr	SA = **D::**, DA = **S1**
SR Hdr	(S3, S2, S1) SL=2
IPv6 Hdr	SA = A::, DA = **E::**
SR Hdr	(…, E::, D::10,…) SL=k-1
Payload	

END.B6 – IPv6 Binding Segment (insert)

- *IPv6 binding segment*
  - **Do not** decrement Segments Left
  - **Push outer SR header associated with the binding segment**
  - **Update DA with the first segment of the outer SR header**
  - Forward according to outer header DA (first segment of the new SRH)
- Node D advertises prefix D::/64
- On D, the SR *insert* policy 〈 S1, S2, S3 〉 is associated with ID 20
- The corresponding *binding SID* is **D::20**

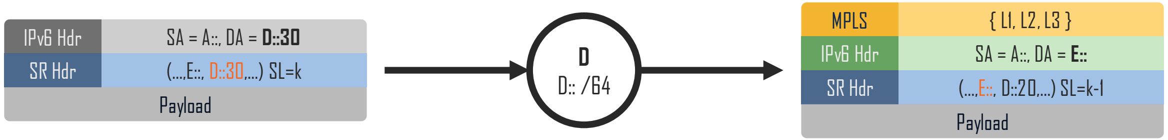


IPv6 Hdr	SA = A::, DA = **D::20**
SR Hdr	(...,E::, D::20,...) SL=k
Payload	

IPv6 Hdr	SA = A::, DA = **S1**
SR Hdr	(S3, S2, S1) SL=2
SR Hdr	(....,E::, D::20,...) SL=k
Payload	

END.BM – MPLS Binding Segment

- *MPLS binding segment*
  - Decrement Segments Left
  - **Push outer MPLS label stack associated with the binding segment**
  - Forward according to the top MPLS label
- Node D advertises prefix D::/64
- On D, the MPLS SR policy { L1, L2, L3 } is associated with ID 30
- The corresponding *binding SID* is **D::30**



IPv6 Hdr	SA = A::, DA = **D::30**
SR Hdr	(...,E::, D::30,...) SL=k
Payload	

D
D:: /64

MPLS	{ L1, L2, L3 }
IPv6 Hdr	SA = A::, DA = **E::**
SR Hdr	(...,E::, D::20,...) SL=k-1
Payload	

Functions might be signaled differently

Signalling	IGP	BGP-LS	BGP-IP/VPN
End	Yes	Yes	
End.X	Yes	Yes	
End.T	Yes	Yes	
End.DX4		Yes	Yes
End.DX6	Yes	Yes	Yes
End.DX2		Yes	Yes
END.DT4		Yes	Yes
End.DT6	Yes	Yes	Yes
End.B		Yes	

Signalling	IGP	BGP-LS	BGP-IP/VPN
T.insert		Yes	
T.Encap		Yes	

Locator – routing table

IETF

- **Segment Routing IPv6 for Mobile User Plane**
    - https://tools.ietf.org/html/draft-ietf-dmm-srv6-mobile-uplane-03
- **Segment Routing IPv6 for mobile user-plane PoCs**
    - https://tools.ietf.org/html/draft-camarillo-dmm-srv6-mobile-pocs-01

End.MAP -Endpoint function with SID mapping

End.M.GTP6.D -Endpoint function with IPv6/GTP decapsulation into SR policy

End.M.GTP6.E -Endpoint function with encapsulation for IPv6/GTP tunnel

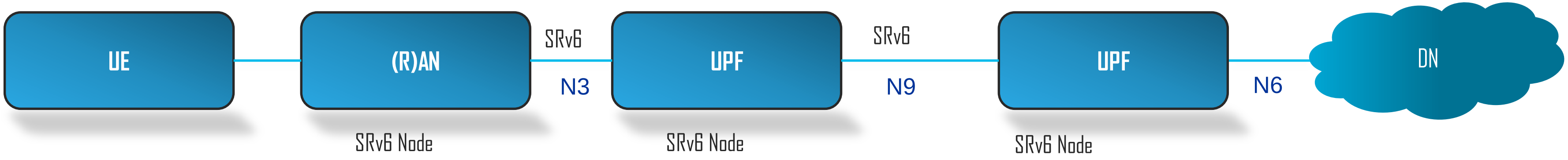
End.M.GTP4.E -Endpoint function with encapsulation for IPv4/GTP tunnel

T.M.Tmap -Transit with tunnel decapsulation and map to an SRv6 policy

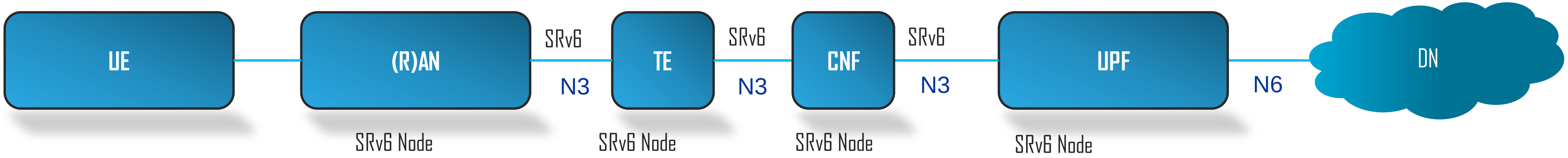
End.Limit -Rate Limiting Function

IETF -Modes

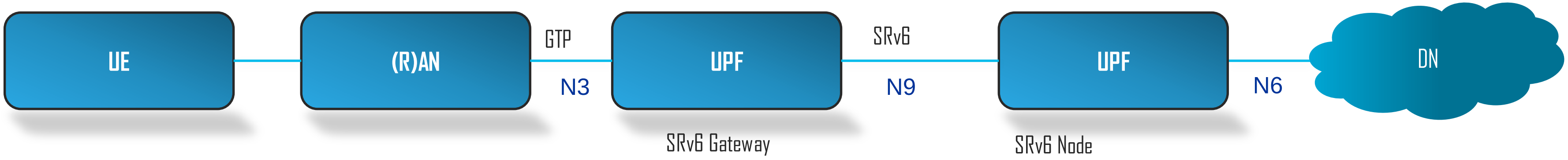
Traditional



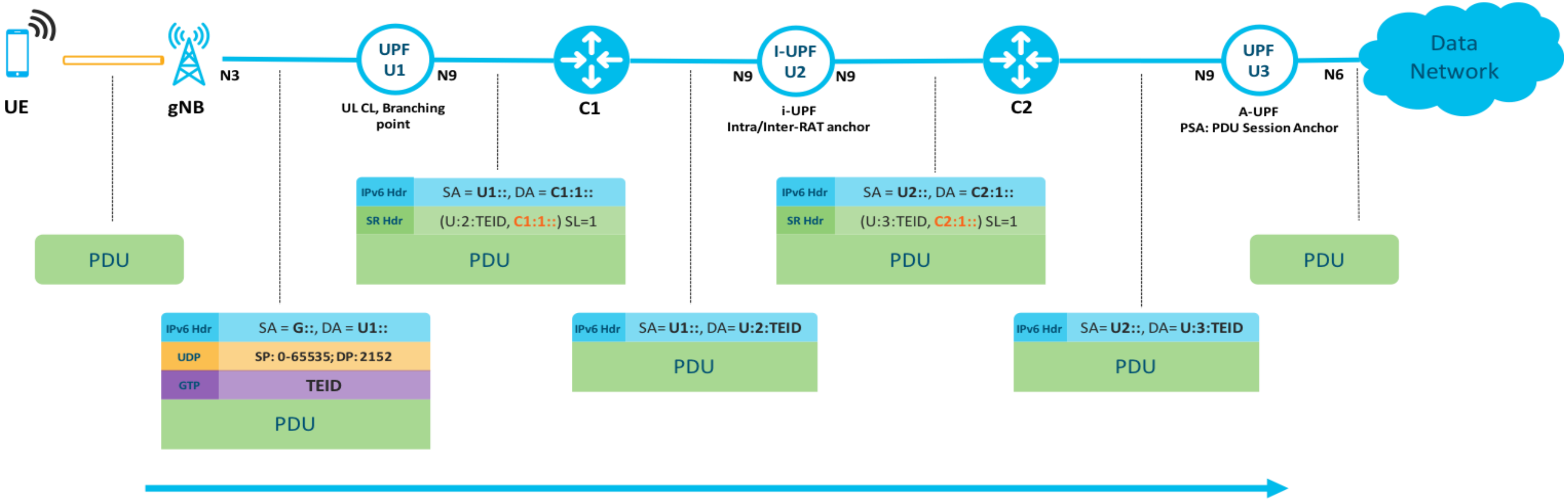
Enhanced (Traffic Engineering, Service Programming)



Enhanced with unchanged gNB (Interworking)



3GPP

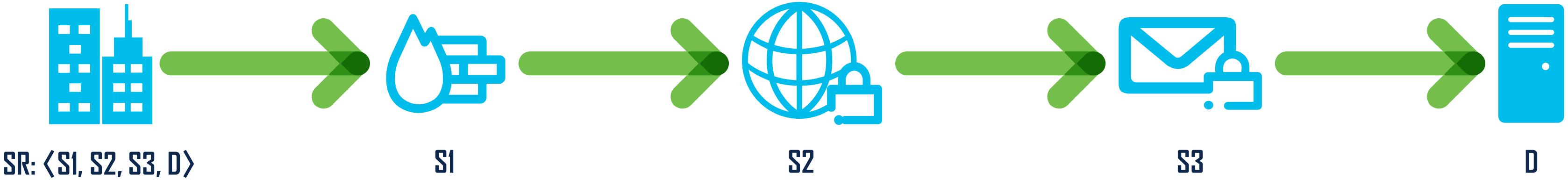

UE
gNB
N3
UPF
U1
N9
UL CL, Branching point
C1
N9
I-UPF
U2
N9
i-UPF
Intra/Inter-RAT anchor
C2
N9
UPF
U3
N6
A-UPF
PSA: PDU Session Anchor
Data Network
PDU
IPv6 Hdr
SA = G::, DA = U1::
UDP
SP: 0-65535; DP: 2152
GTP
TEID
PDU
IPv6 Hdr
SA = U1::, DA = C1:1::
SR Hdr
(U:2:TEID, C1:1::) SL=1
PDU
IPv6 Hdr
SA= U1::, DA= U:2:TEID
PDU
IPv6 Hdr
SA = U2::, DA = C2:1::
SR Hdr
(U:3:TEID, C2:1::) SL=1
PDU
IPv6 Hdr
SA= U2::, DA= U:3:TEID
PDU
PDU

GTP-U Conclusion

- GTP is legacy protocol
- GTP has significant overhead (especially for 5G with extension header)
- GTP is load balancing unfriendly (can be used by using IPv6 + flow label)
- We can encapsulate GTP into SRV6 ☺
- SRv6 can do the same as GTP
- SRv6 provides natural link between Mobile and Transport

Service Chaining with **SR**

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



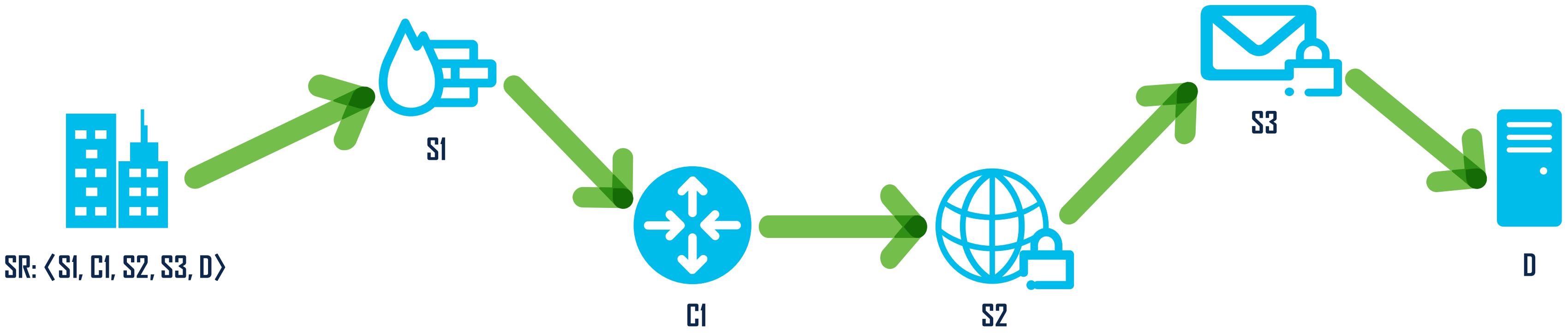
- **Services** are expressed with **segments**
  - Flexible
  - Scalable
  - Stateless

SRv6 service segments

- SID instantiated on an SRv6 router / host connected to the service
  - Send with SRH to SRv6-capable device
  - Use proxy function to remove / hide SRH before sending to SRv6 unaware service

- SID instantiated on an SRv6 aware service
  - Traffic processing depends on the SID
    e.g.
    - F1::10 → Firewall F1 with rule-set 10
    - F1::20 → Firewall F1 with rule-set 20

Service segments in SR architecture

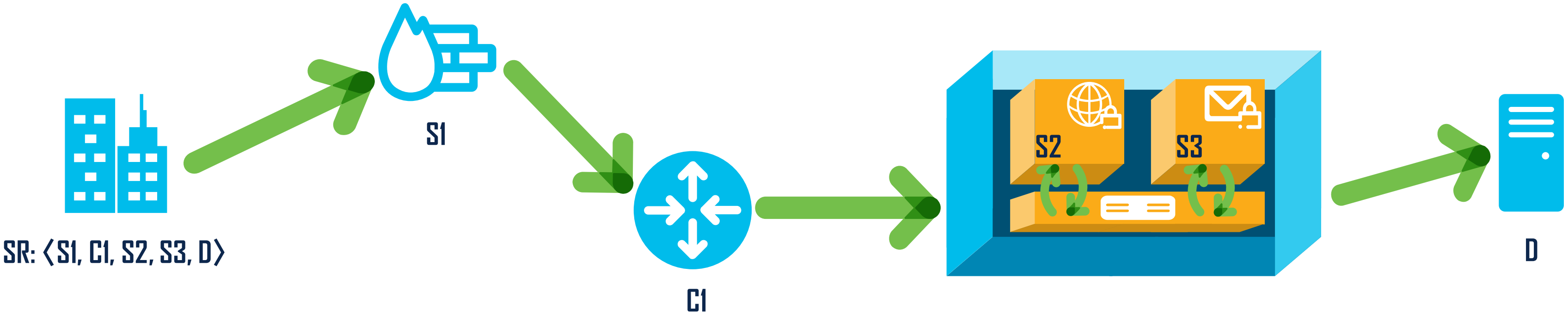


Just another type of segment

- Stateless in the fabric
- Seamless integration with VPN and/or TE
- Service is opaque to the head-end and intermediate nodes

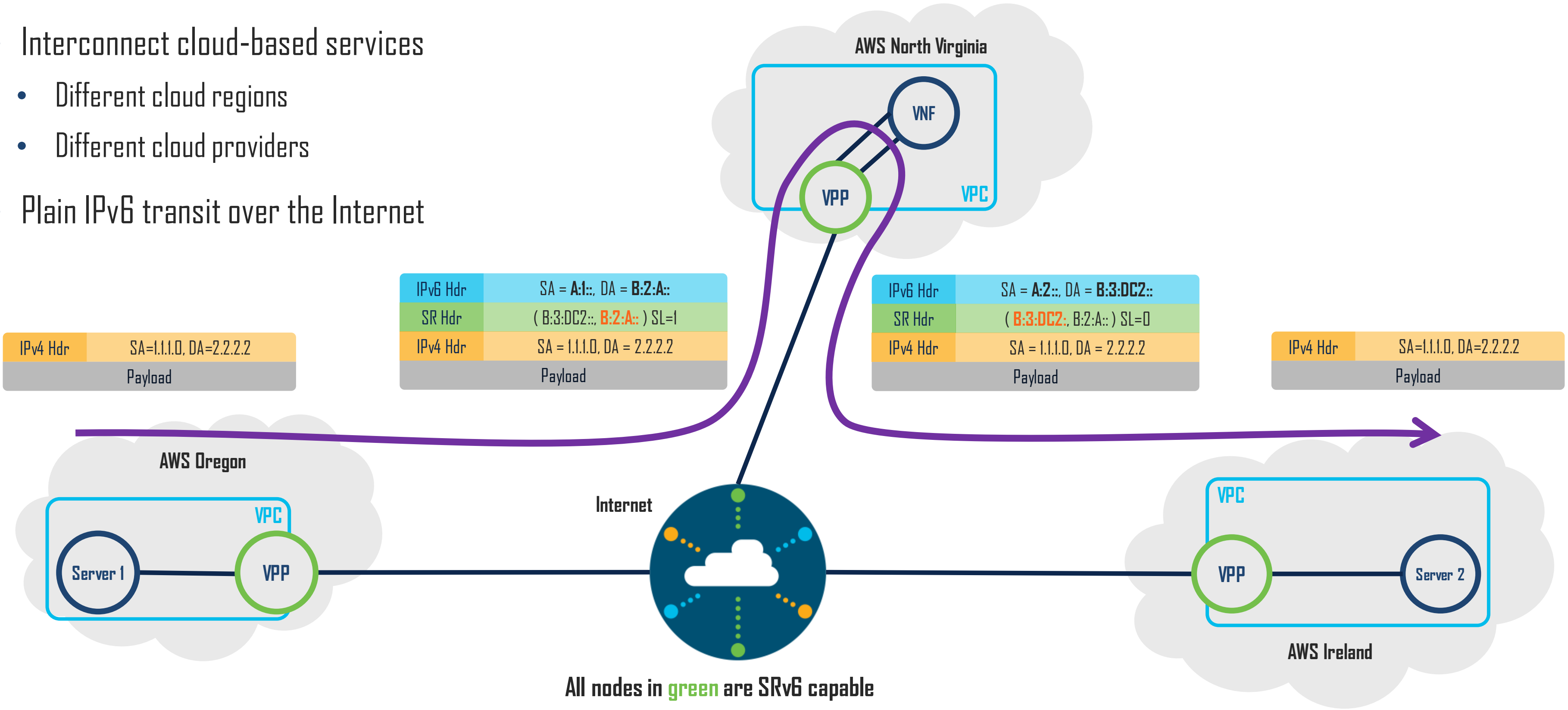
Services

- SR-Aware: Service is bound to an SR endpoint function
  - **Processes all the traffic** passing through the node or is **triggered by a specific endpoint function**
- SR-UnAware: Service is not able to process SR traffic
  - Requires an SR proxy to operate properly



Multi-cloud overlays

- Interconnect cloud-based services
  - Different cloud regions
  - Different cloud providers
- Plain IPv6 transit over the Internet



IPv4 Hdr	SA=1.1.1.0, DA=2.2.2.2
Payload	

IPv6 Hdr	SA = **A:1::**, DA = **B:2:A::**
SR Hdr	(B:3:DC2::, **B:2:A::**) SL=1
IPv4 Hdr	SA = 1.1.1.0, DA = 2.2.2.2
Payload	

IPv6 Hdr	SA = **A:2::**, DA = **B:3:DC2::**
SR Hdr	(**B:3:DC2:**, B:2:A::) SL=0
IPv4 Hdr	SA = 1.1.1.0, DA = 2.2.2.2
Payload	

IPv4 Hdr	SA=1.1.1.0, DA=2.2.2.2
Payload	

All nodes in green are SRv6 capable

Service Programming Conclusion

- Current approach is inefficient
- NSH is stateful hence not scalable
- SRv6 is stateless
- There are VNFs supporting SRv6 already
- For legacy VNF we will use proxy functionality
- TLVs allow to use metadata to carry additional information
- SRv6 concept is ready for microservices
- SRv6 excellent for cloud based application
- Any function can be anywhere

SRv6 Ecosystem

CISCO

2018

ISIS SRv6 w/ TILFA (NfV ; END.AS)
L3 Service (IPv4 L3VPN)
SRv6 OAM (Ping/Trace)

2019

SRv6 for Internet (v4/v6, VPNv6)
SRv6 Flex-Algo
Multi-plane, Delay optimized
L2 P2P Service (EVPN VPWS)
SRv6/MPLS Gateway

Linux / FD.io

FD.io VPP P4

- END.AS
- END.AD
- END.ASM
- END.AM

Open-Source Applications

WIRESHARK TCPDUMP Pyroute2 SERA IPTABLES SNORT

NfV Partners

radware TREND MICRO ENEA Qosmos Division

Smart NIC

intel Mellanox TECHNOLOGIES

Conclusion

- SRv6 is Here
- It is ready for greenfield
- It has wide opensource support
- Massive Scale!
- Stateless Service Chaining –Network Programming
- Can Replace GTP

Other Information

- https://dcloud-cms.cisco.com/demo/cisco-srv6-l3-vpn-with-flexible-algorithm-and-ti-lfa-v1
- https://telecoms.com/intelligence/ipv6-enhanced-innovation-embracing-the-ip-future-in-5g-cloud-era/
- https://www.brighttalk.com/webcast/12761/434617
- https://www.segment-routing.net/updates-20191029-srv6-state/

Thank you