



# SRv6

## Record-Speed Standardization and Deployment

Ahmed Abdelsalam  
ahabdels@cisco.com

ITNOG6 - Bologna

# Thank you

- Lead operators
- EcoSystem Partners
- Academic Partners
- IETF Partners
- Cisco SR Team

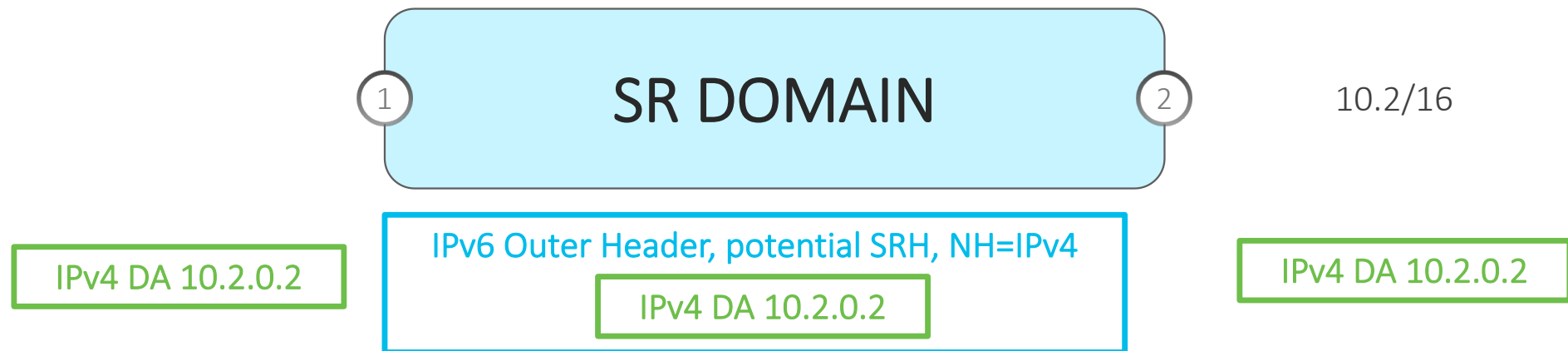


# segment-routing.net

- SRv6 uSID: [CKN](#), [MPLS WC](#), [DKNOG](#), [Training](#)
- SRv6 Ultra-Scale SR Policy: 26 uSID push at linerate: [demo](#)
- BGP PIC Edge with SRv6 Summarization: ISIS UPA: [demo](#)
- Path Tracing: [NANOG](#), [Tutorial](#)

# Introduction

# Transparent Service



- Customer packet is encapsulated from ingress to egress of the SR Domain
- SRv6 is applied to the OUTER header
- The inner packet is untouched

# SRv6 Network Programming - RFC8986

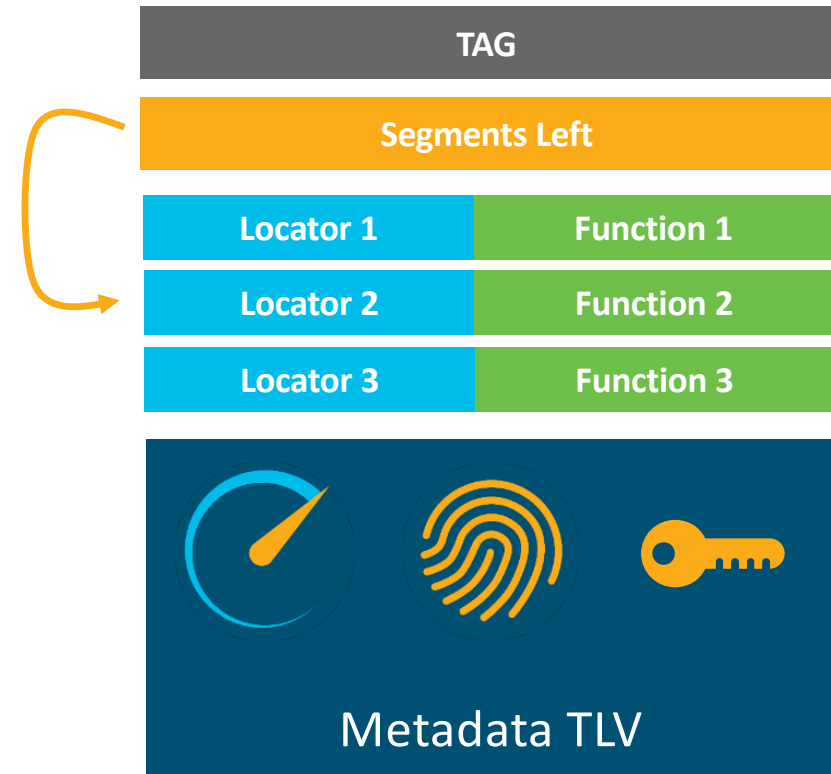
- The End-to-End Policy is encoded as a Network Program
  - The first instruction is in the outer DA
  - The remaining instructions are in the SRH
- An instruction (a SID) may be bound to any behavior
  - TILFA FRR and uLoop Avoidance
  - Traffic Engineering: internal to the domain and across peering links
  - L2/L3 VPN's
  - NFV
  - Any HW custom behavior: P4 on Silicon1
  - Any SW custom behavior: Container orchestrated by Kubernetes
- Powerful Service Creation
  - Any service can be encoded as an ordered list of instructions (Low-latency Slice, VPN)

# Stateless Fabric

- The state (network program) is in the header
- The state is not in the fabric

# SR Extension Header

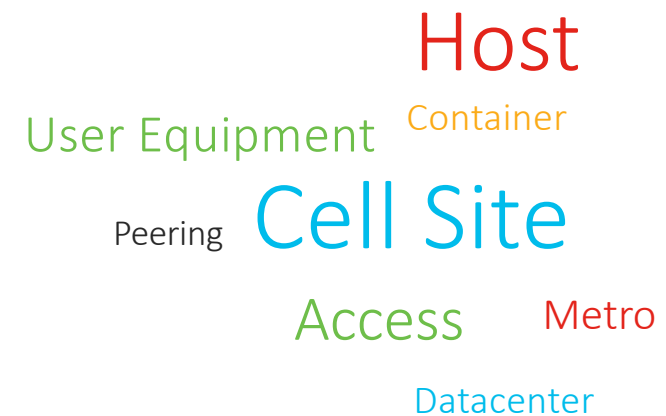
- SRv6 is a native extension of IPv6
  - RFC 8754
  - As foreseen 25 years ago by RFC2460
- SRH contains an ordered list of SID's



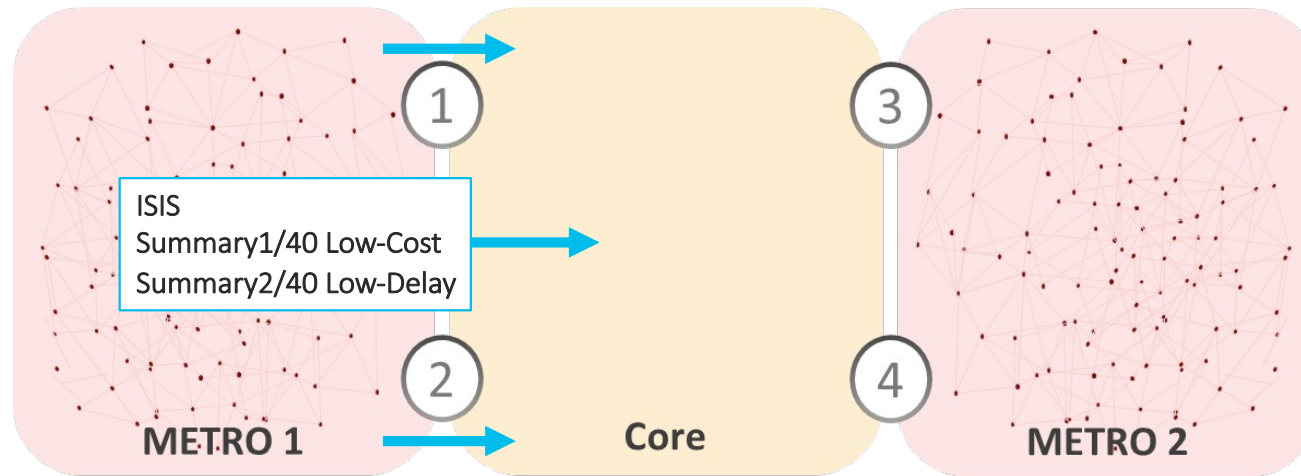


# Unified Data Plane And Seamless Deployment

- A **unified data plane** spanning from **Application socket to Internet peering** though Datacenter, Access, Metro, Core, Peering
- A **single data plane** natively supported by all nodes – **just IP!**
- Seamless forwarding through IPv6 transit nodes
- Most use-cases do not need an SRH
  - DA SID contains up to 6 micro-instructions (uSID's)



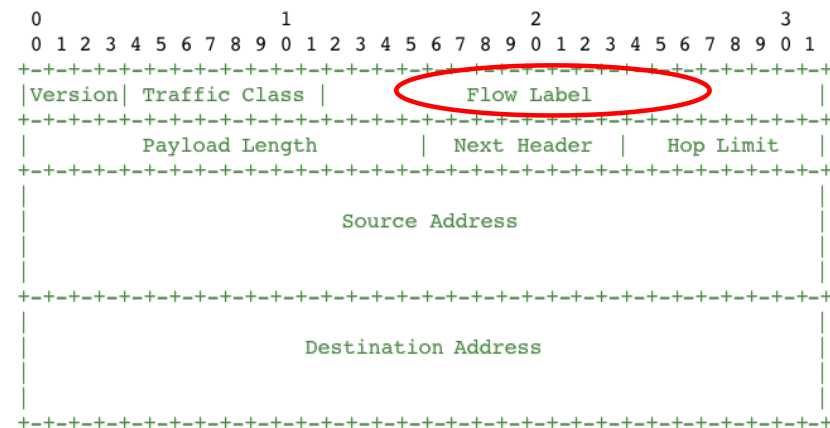
# Prefix Summarization



- 1000 times less IGP routes than with MPLS
- No BGP inter-AS Option A/B/C
- End-to-End Flex Algo Continuity

# Optimum Load Balancing

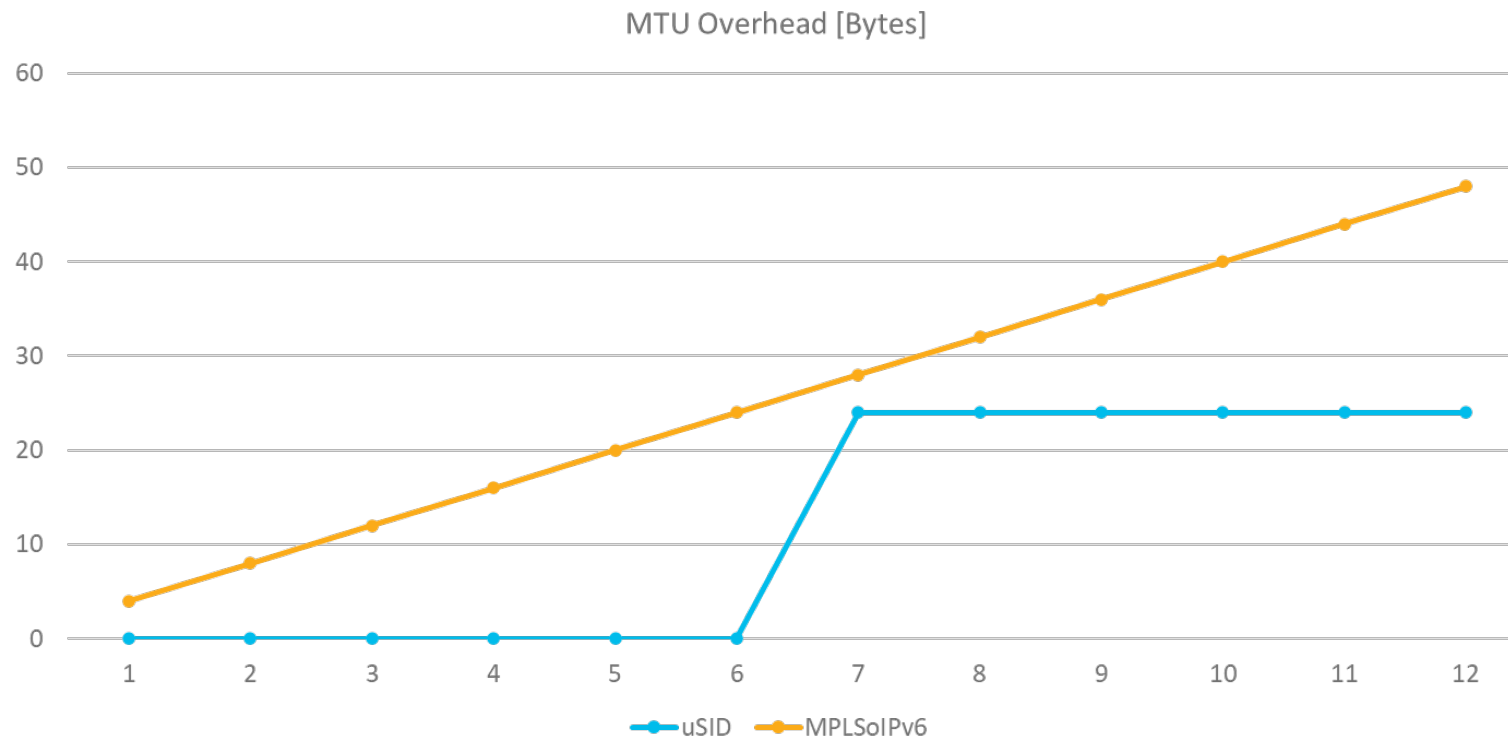
- The entropy of the inner packet is present in the outer Flow Label



# Less is More

- No MPLS dataplane
  - 66% less dataplane entries and counters
- No MPLS to the host
  - Bringing MPLS to the host is very complex: i.e. there is no label in a socket
- No VxLAN gateway at the DC edge
  - SRv6 end-to-end
  - Massive economical benefit
- No UDP trick for LB
- No RSVP-TE for TE/FRR
- No NSH for NFV

# Lowest MTU Overhead



One of the many benefits of SRv6 Micro Instructions

# Deployment Status

# Record-Speed Deployment

- 3 years of commercial deployment (2019-2022)
- ~100M SRv6 subscribers
- ~100 deployments, with ~14 public reports
- Across markets (Web, SP, Enterprise) and geographies (Asia, EU, US)

---

SoftBank

Rakuten

Indosat

MTN Uganda

Noia

Iliad

Alibaba

China  
Telecom

China Bank

Cernet2

Free

Bell Canada

China Unicom

Line

---

NEW

# Commitment to SR Lead-Operators



Standard-Based Technology



Vendor eco-system



Open-Source eco-system



# Mature Standardization

- Proposed Standard
  - RFC 8402 SR Architecture
  - RFC 8754 SRv6 DataPlane
  - RFC 8986 SRv6 Network Programming
  - RFC 9252 SRv6 BGP Extension
  - RFC 9256 SR Policy
  - RFC 9259 SRv6 OAM

Much faster standardization than usual  
Sign of the SRv6 Industry Endorsement

# Rich Eco-System

- 25 HW implementations
  - Cisco Systems, Nokia, Arrcus, Kaloom, NoviFlow, Huawei, ZTE, Juniper
  - Broadcom, Barefoot, Intel SmartNIC, Marvell, Mellanox,
  - Spirent, Ixia
  - Multiple Interop Reports
- 14 open-source platforms/ Applications
  - Linux, FD.io VPP, P4, iptables, nftables, snort, ExaBGP, GoBGP, GoBMP, Contiv-VPP, SONiC, SAI, FRR, SERA

# Many Successful Interops

- 2020/04: EANTC: SRv6 interop between Cisco, Huawei, Juniper, Arrcus, Ixia ([link](#))
  - Classic IPv6 nodes as SRv6 transit nodes
  - SRv6-L3VPN for IPv4 and IPv6 services
  - SRv6 TI-LFA FRR link protection with SRH insert
  - SRv6 EVPN for E-Line and EVPN L3VPN services
  - SRv6 TE SR Policy
- 2021/02: NetOne Systems ([link](#))
  - Cisco XR, Cisco NX, Juniper
- 2021/09: EANTC: SRv6 interop between Cisco, Huawei, Juniper, Nokia, Spirent ([link](#))
  - [SRv6-Based Global IPv4 and IPv6 services](#)
  - SRv6-L3VPN for IPv4 and IPv6 services
  - SRv6 TI-LFA FRR [local SRLG protection](#) with SRH insert
  - SRv6 EVPN for E-Line and EVPN L3VPN services
  - [IGP Flex- Algo using TWAMP-measured link delays](#)

SRv6 uSIDs

# SRv6 uSID Terminology

- Industry:
  - SRv6 Micro Segment
  - SRv6 uSID
  - Briefly: uSID
- IETF: NEXT-C-SID
  - Briefly: Next
  - IETF document: [draft-ietf-spring-srv6-srh-compression-01](#)
  - Training: [link](#)

# Container of 6 uSID's

- SRv6 Network Programming (RFC8986)
  - The source encodes any end-to-end program as an ordered list of instructions
  - The first instruction is in the outer DA
  - The remaining instructions are in the SRH
- An instruction is called a SID
- A Container SID may contain up to 6 micro-instructions called uSID's

FC00:0000:1111:2222:3333:4444:5555:6666

uSID Block

uSID1

uSID2

uSID3

uSID4

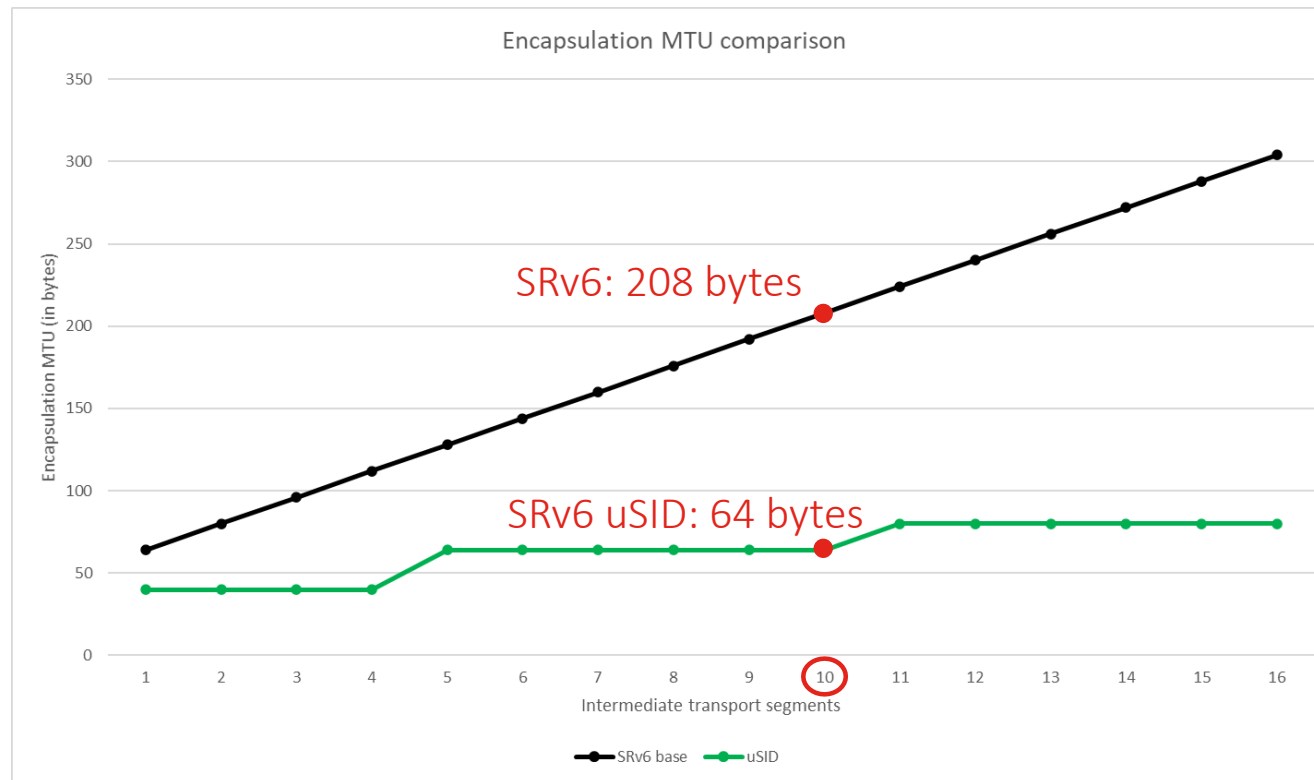
uSID5

uSID6

# Perfect SRv6 Integration

- uSID reuses SRH (RFC8754) without any change
- uSID strictly applies the SRv6 Network Programming (RFC8986)
- uSID can be bound to any Network Programming Instruction

# SRv6 uSID offers the best SRv6 Compression





# Rakuten – SRv6 uSID in Deployment



- 5G End-to-End Network Slicing based on SRv6 uSID Flex-Algo
  - SRv6 uSIDs are allocated from the ULA address range
  - SRv6 uSID ISIS Flex-Algo: Low-Cost vs Low-Delay
  - SRv6 uSID BGP services
  - SRv6 uSID TILFA
  - Cisco NCS5500 and NCS-540 series
- Innovation in partnership with Cisco
  - BGP PIC Core and Edge with SRv6 Summarization: ISIS UPA ([demo](#))
  - SR BW counters for deterministic and scalable capacity planning and BW guarantee

<https://www.segment-routing.net/conferences/MPLS-WC-2022-Amit-Dhamija/>



# Bell Canada - SRv6 uSID Deployment

- Bell promptly switched from SR-MPLS to SRv6 uSID
- Continued Simplification (remove MPLS dataplane)
- Better Routing Scale: Summarization
- Better HW Scale: linerate 26 uSID push for end-to-end SR Policy
- Seamless Deployment (6 uSID's in DA without SRH)
- End-to-End IP Unified Dataplane from socket to Internet Peering
  - SRv6-TE Policy: topological and service uSID's
- Service Programming
- Reduce network service costs by up to 90%
  - footprint by 75%
  - power consumption by as much as 66%

<https://www.segment-routing.net/conferences/MPLS-WC-2022-Daniel-Voyer/>

<https://www.segment-routing.net/conferences/MPLS-WC-2022-Daniel-Bernier-Jesper-Eriksson/>

# Seamless Deployment

- Many use-cases leverage classic IPinIP
- The outer DA already holds 6 uSIDs !! [D. Cai – Alibaba podcast - SRv6 uSID](#)
- SDN coherence
  - The SDN controller packs the uSID policy in the outer DA
  - The host/access node only sees an opaque DA
- VPN GW to interconnect with legacy VPN

# SRv6 uSID - Feature Parity with SR-MPLS

- TILFA & uLoop Avoidance
- Flex-Algo Low-Cost/Low-Delay with VPN Automated Steering
  - Performance Monitoring: Link Latency (for Low-Delay slice)
- SRv6-TE Policy: topological and service uSID's
- L3VPN (IPv4 and IPv6), IPv4 Internet, IPv6 Internet, PW, BD
- VPN GW to interconnect with legacy VPN
- Seamless Inter-Domain with Summarization
  - No need for complex BGP3107 mechanisms

# SRv6 uSID - Rich Eco-System

- Cisco, NoviFlow, Arrcus, Nokia, Ciena
- Merchant: Silicon One, Broadcom, Marvell, Barefoot
- Open Source: Linux, FD.io, P4, eBPF, Cilium, SAI, SONIC, FRR



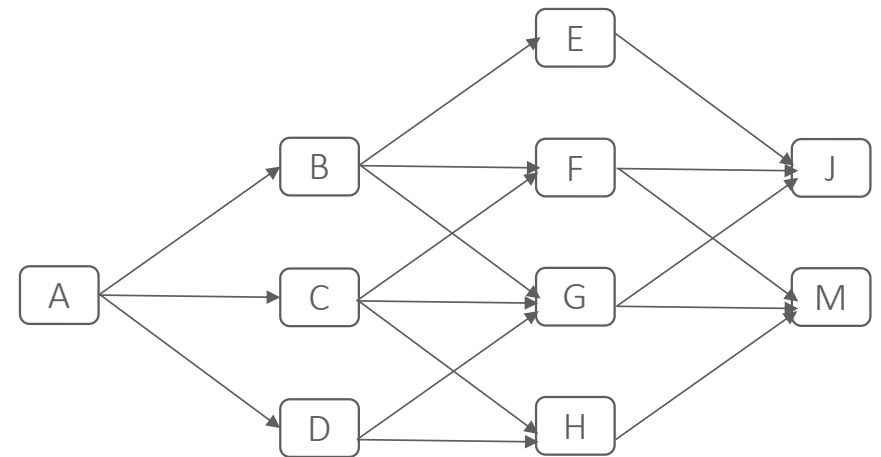
Linux



OPEN  
Compute Project®

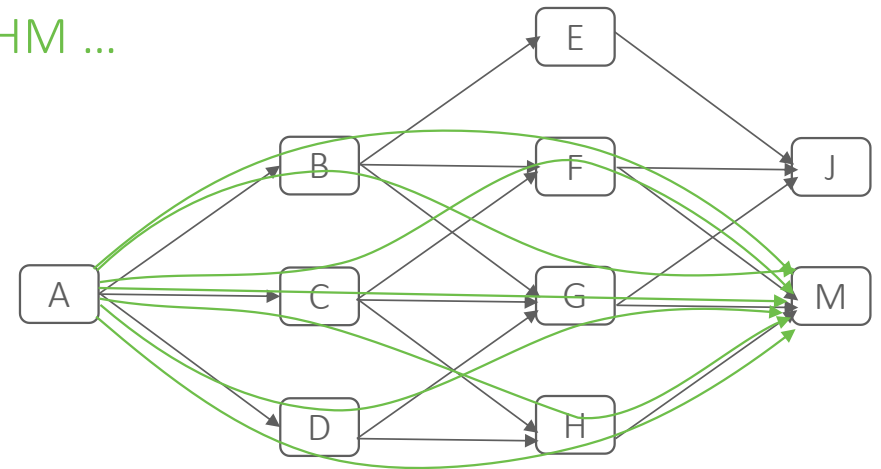
# Path Tracing

The exact path from A to M is not known



# The exact path from A to M is not known

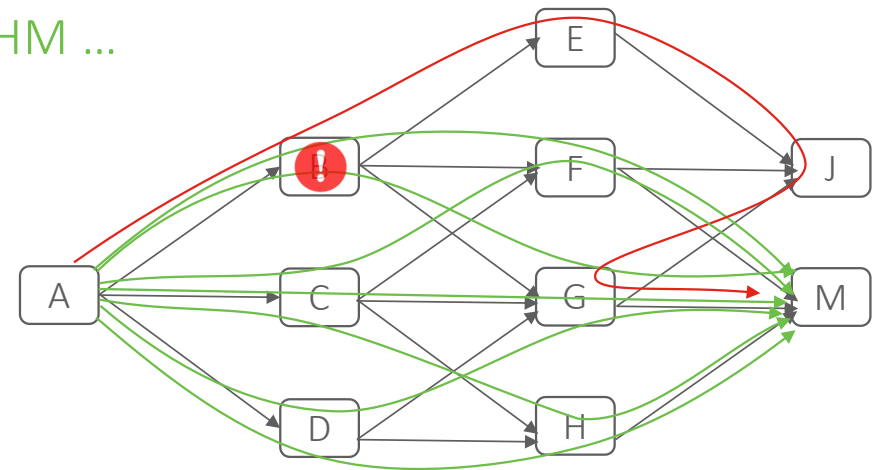
- 7 possible “valid” ECMP path
  - ABFM, ABGM, ACFM, ACGM, ACHM, ADGM, ADHM ...





# The exact path from A to M is not known

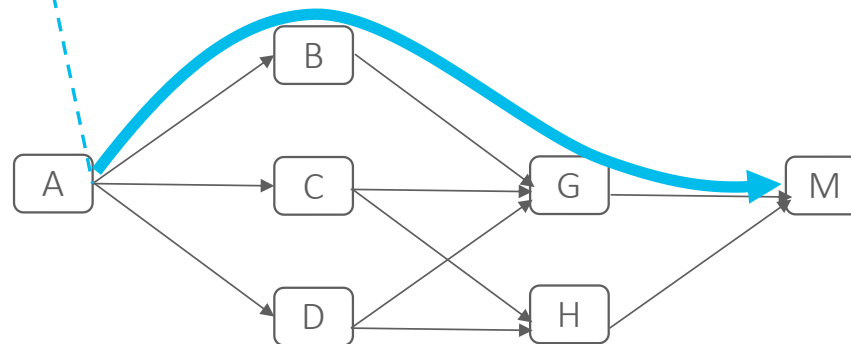
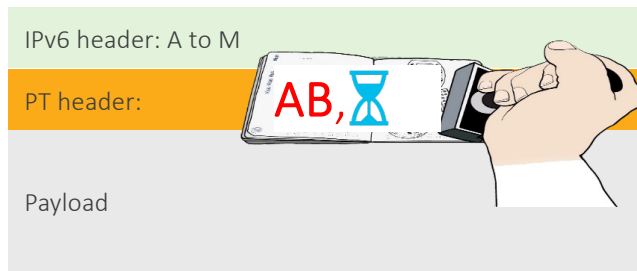
- 7 possible “valid” ECMP path
  - ABFM, ABGM, ACFM, ACGM, ACHM, ADGM, ADHM ...
- The path may be invalid
  - Routing or FIB corruption @ B
- Timestamp at each hop
- Interface Load at each hop



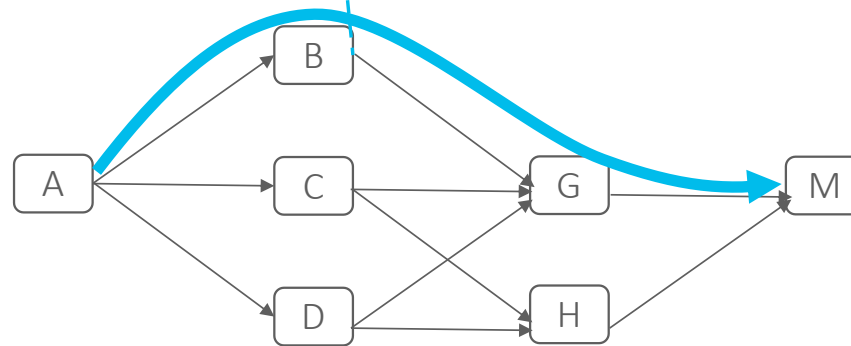
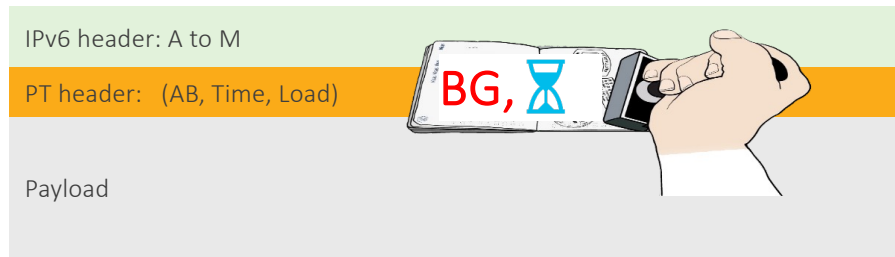
# The Path Tracing idea

- Stamping in the Packet Header
- Implemented at line rate in the most basic HW pipeline
  - Line rate for any packet
  - No punting to CPU, no offload to co-processors
- Ultra-MTU-efficient: only 3 bytes per hop!
  - 12-bit Interface, 8-bit Timestamp, 4-bit Load
- For IPv6, with or without SRH
  - MPLS solution also designed
- Interwork with legacy node

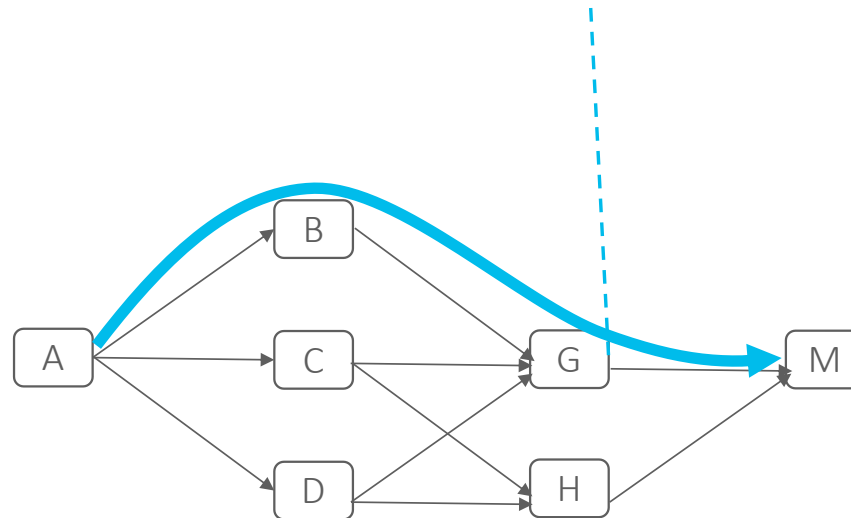
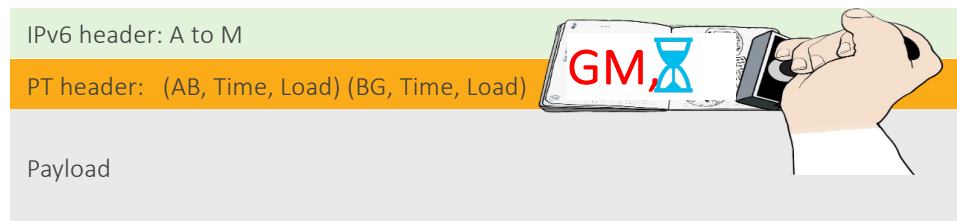
# Stamping Trajectory in PT Header



# Stamping Trajectory in PT Header



# Stamping Trajectory in PT Header

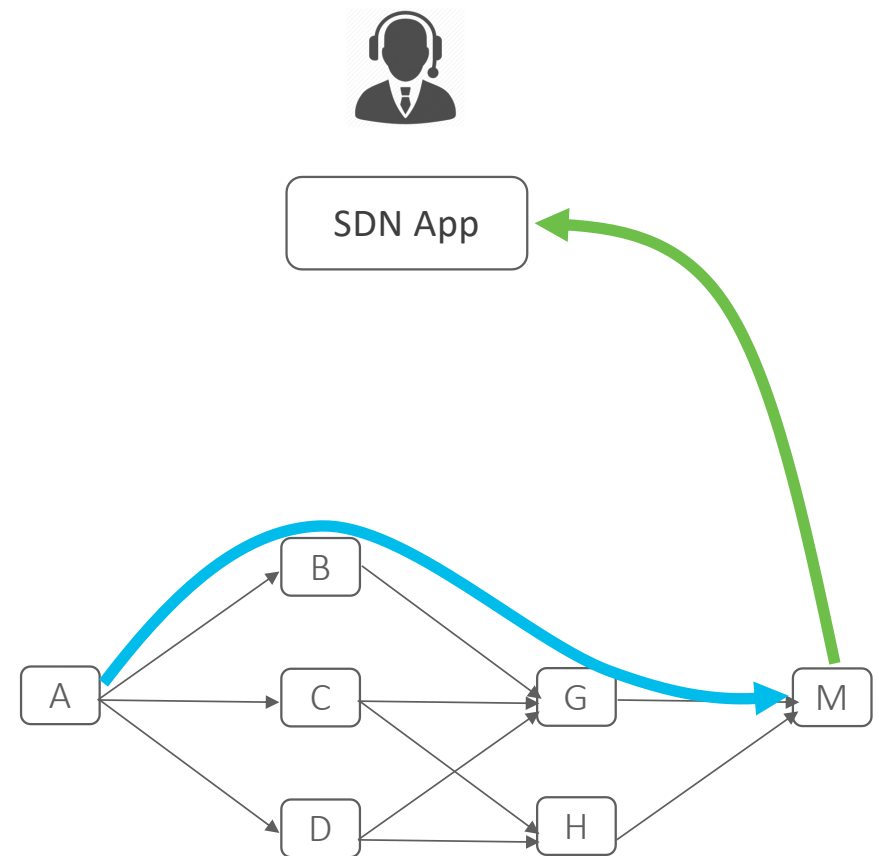


# Dataplane Encapsulation

- Minimize NPU parsing
- Minimize # of Read/Write
- Minimize depth of Read/Write
- Maximize Read/Write at fixed positions
- Avoid Header Insert/Resize
- Minimize MTU

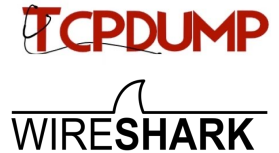
# Minimize HW complexity by leveraging SDN analytics

- Exact path validation (interface by interface)
  - detect blackholes
  - detect unexpected paths
  - detect missing paths
- Propagation latency & Optical stability
- Per node jitter measurement



# Product, Deployment & Ecosystem

- Cisco Shipping in CY22
  - C8000 (Q200), NCS5700 (J2), ASR9000 (LS)
- Rich Eco-System
  - Cisco, Broadcom, Marvell, +others
- Rich Open-Source
  - Linux, FD.io VPP, P4, Wireshark, TCPDUMP
- Path Tracing is being standardized at IETF
  - [Path Tracing in SRv6 networks \(ietf.org\)](https://ietf.org)
  - [Path Tracing in SR-MPLS networks \(ietf.org\)](https://ietf.org)





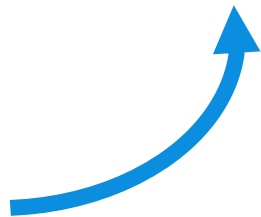
Conclusion

# Simplicity Always Prevails

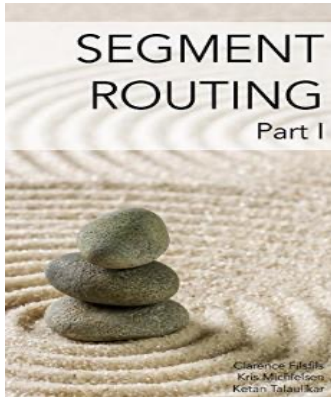


~~LDP~~  
~~RSVP-TE~~  
~~BGP 3107~~  
~~MPLS~~  
~~UDP/VxLAN~~  
~~NSH~~

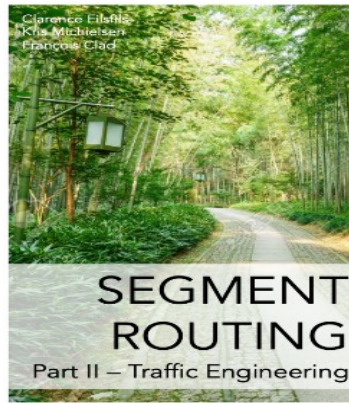
Furthermore with more scale and functionality



# Stay up-to-date



[amzn.com/B01I58LSUO](https://amzn.com/B01I58LSUO)



[amazon.com/dp/B07N13RDM9](https://amazon.com/dp/B07N13RDM9)

SRv6 Part III  
CY2023



[twitter.com/SegmentRouting](https://twitter.com/SegmentRouting)



[facebook.com/SegmentRouting/](https://facebook.com/SegmentRouting/)



[segment-routing.net](https://segment-routing.net)



[linkedin.com/groups/8266623](https://linkedin.com/groups/8266623)

[ask-segment-routing@cisco.com](mailto:ask-segment-routing@cisco.com)

