SRv6 Network Programming Introduction

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 dataplane instantiations

Segment Routing

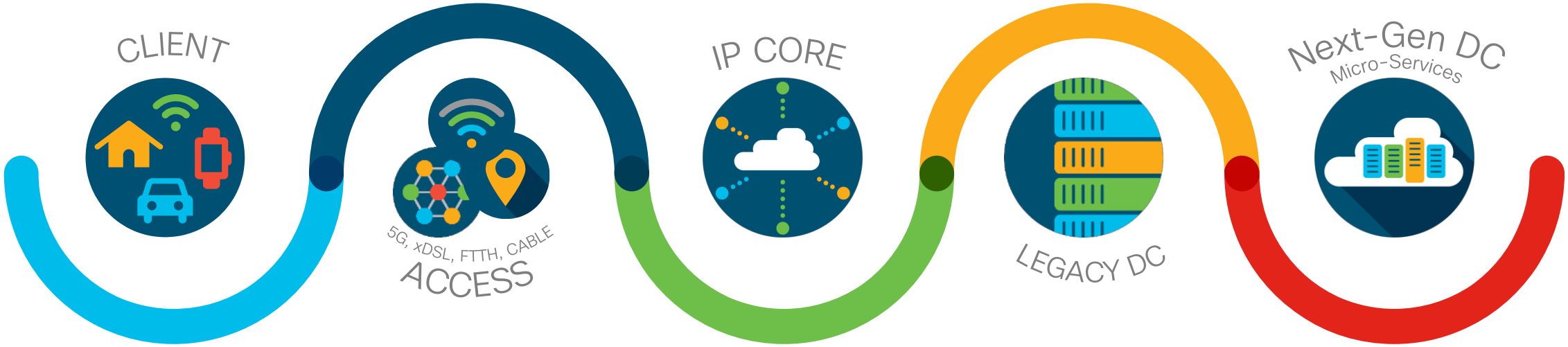
MPLS

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

IPv6

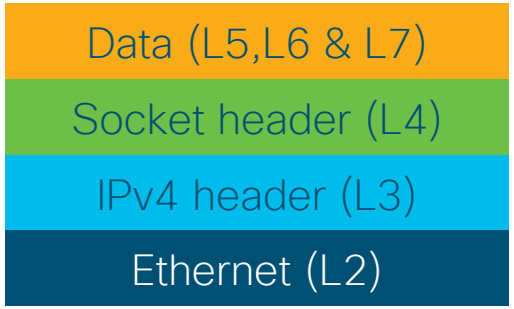
- leverages RFC8200 provision for source routing extension header
- 1 segment = 1 address
- a segment list = an address list in the SRH

IPv6 provides reachability



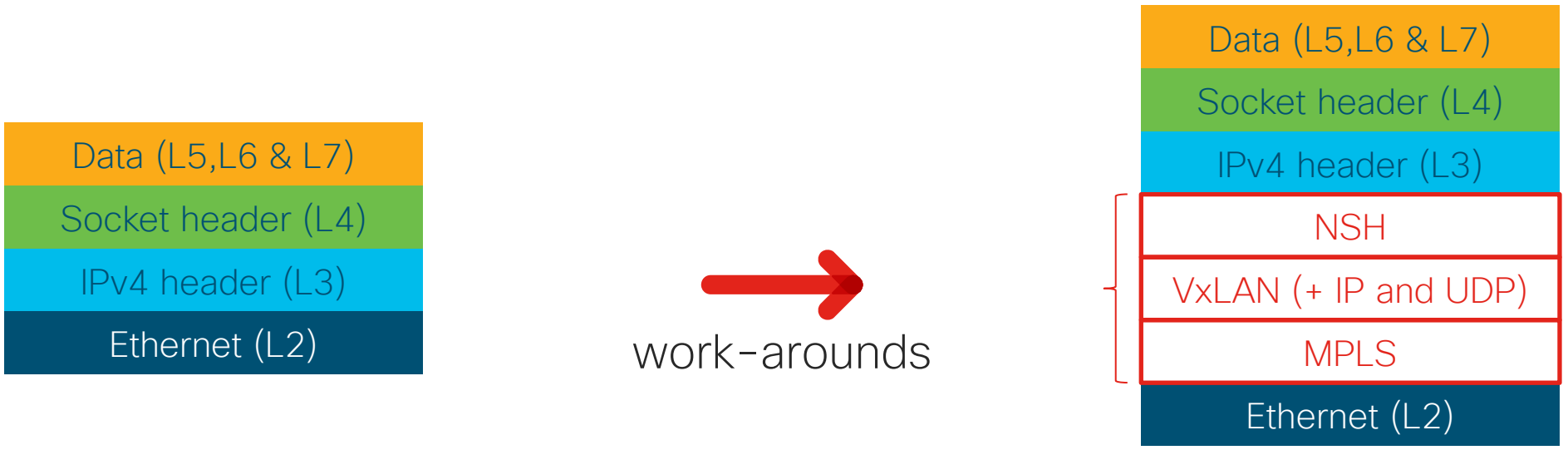
IPv4 limitations & work-arounds

- × Limited address space
- × No engineered Load Balancing
- × No VPN
- × No Traffic Engineering
- × No Service Chaining



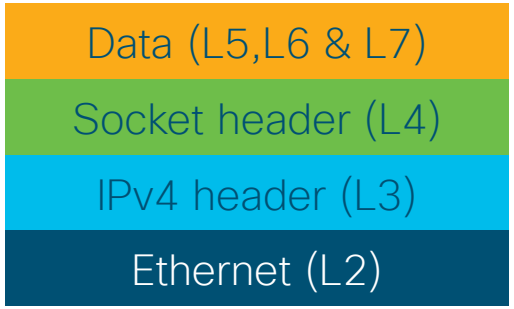
IPv4 limitations & work-arounds

- × Limited address space → NAT
- × No engineered Load Balancing → MPLS Entropy Label, VxLAN UDP
- × No VPN → MPLS VPN's, VxLAN
- × No Traffic Engineering → RSVP-TE, SR-TE MPLS
- × No Service Chaining → NSH

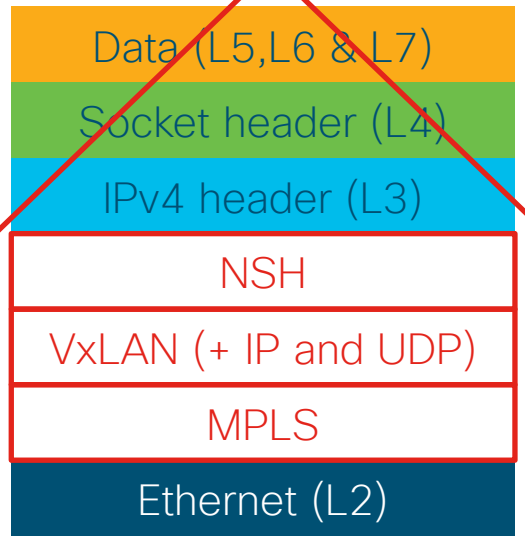


IPv4 limitations & work-arounds

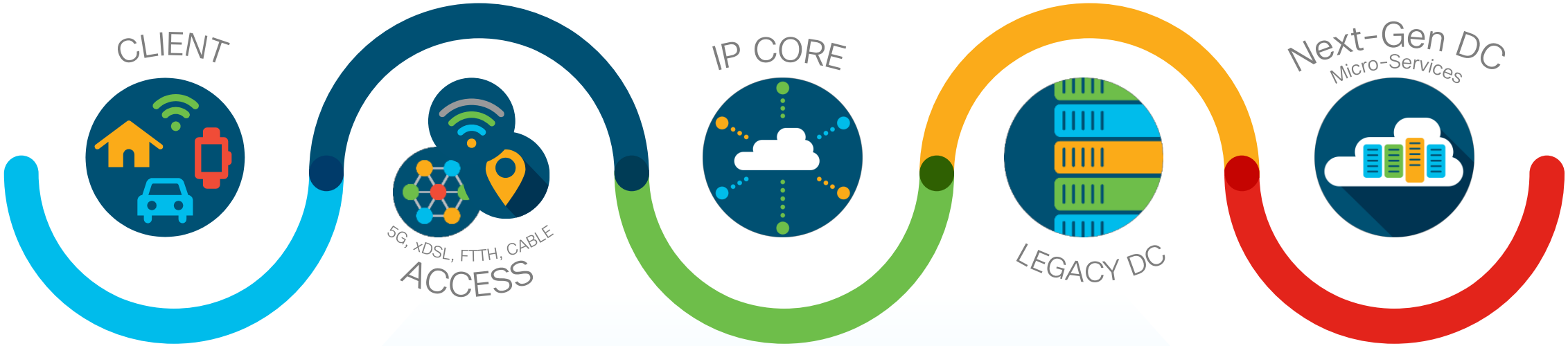
- × Limited address space → NAT
- × No engineered Load Balancing → MPLS Entropy Label, VxLAN UDP
- × No VPN → MPLS VPN's, VxLAN
- × No Traffic Engineering → RSVP-TE, SR-TE MPLS
- × No Service Chaining → NSH



Simplicity (back to the OSI model)



SRv6 unleashes IPv6 potential



TE FRR VPN NFV Scalability Automation Single protocol

SR for anything:
Network as a Computer

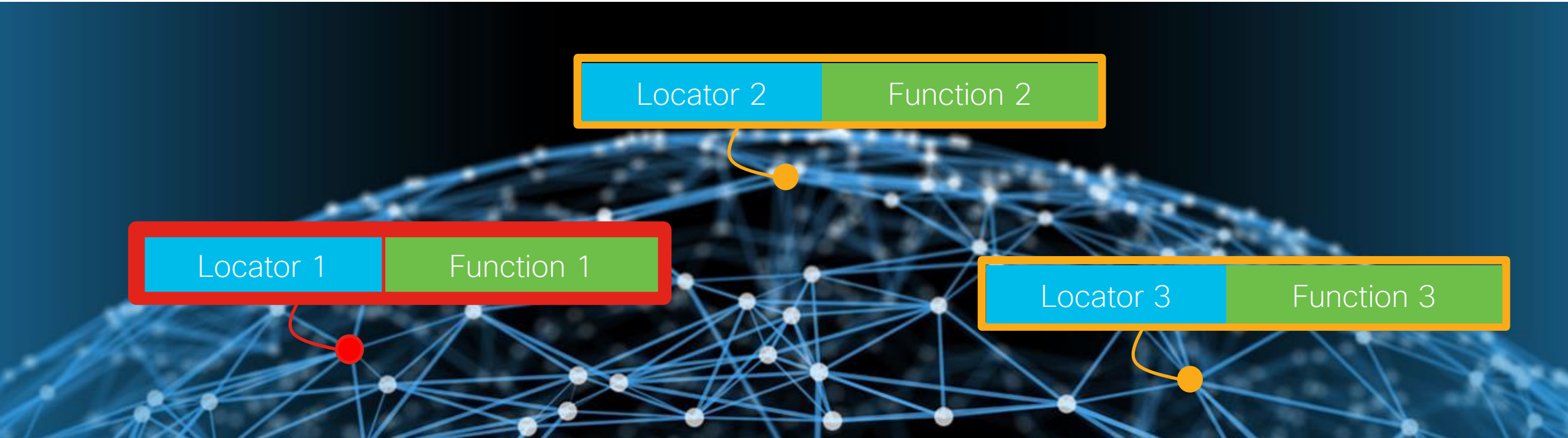
Network instruction

- 128-bit SRv6 SID
  - Locator: routed to the node performing the function
  - Function: any possible function
    either local to NPU or app in VM/Container
  - Flexible bit-length selection

Network Program

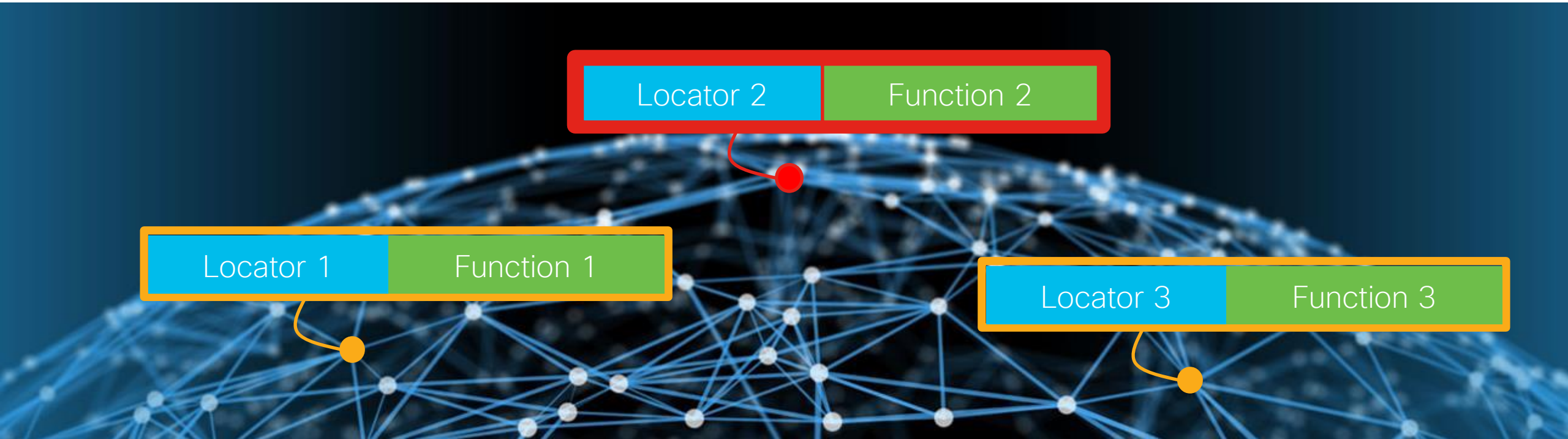
Next Segment

Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3



Network Program

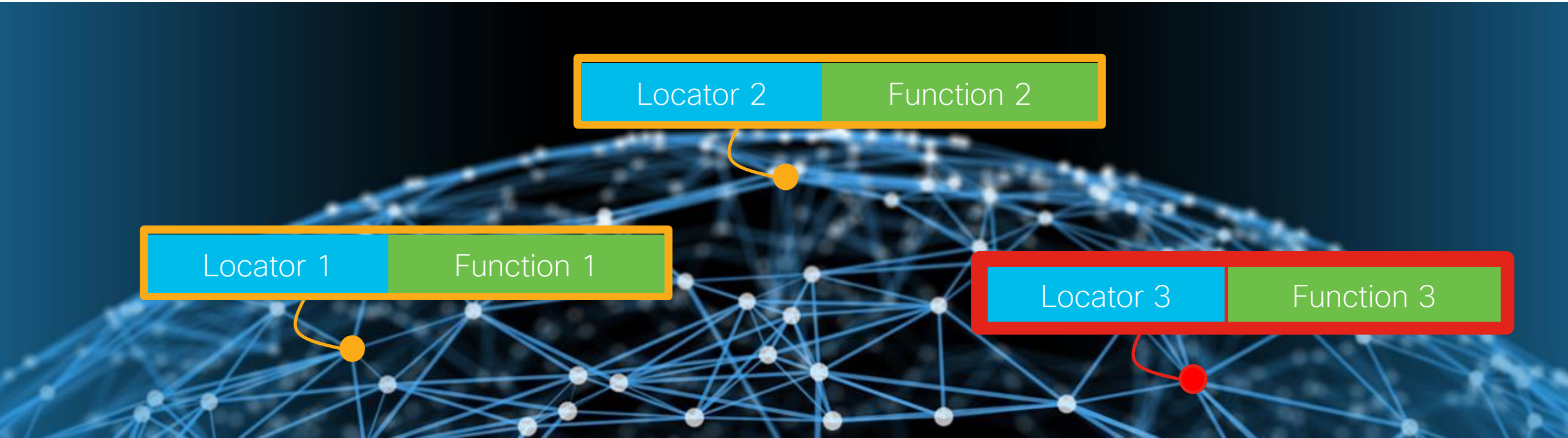
Next Segment



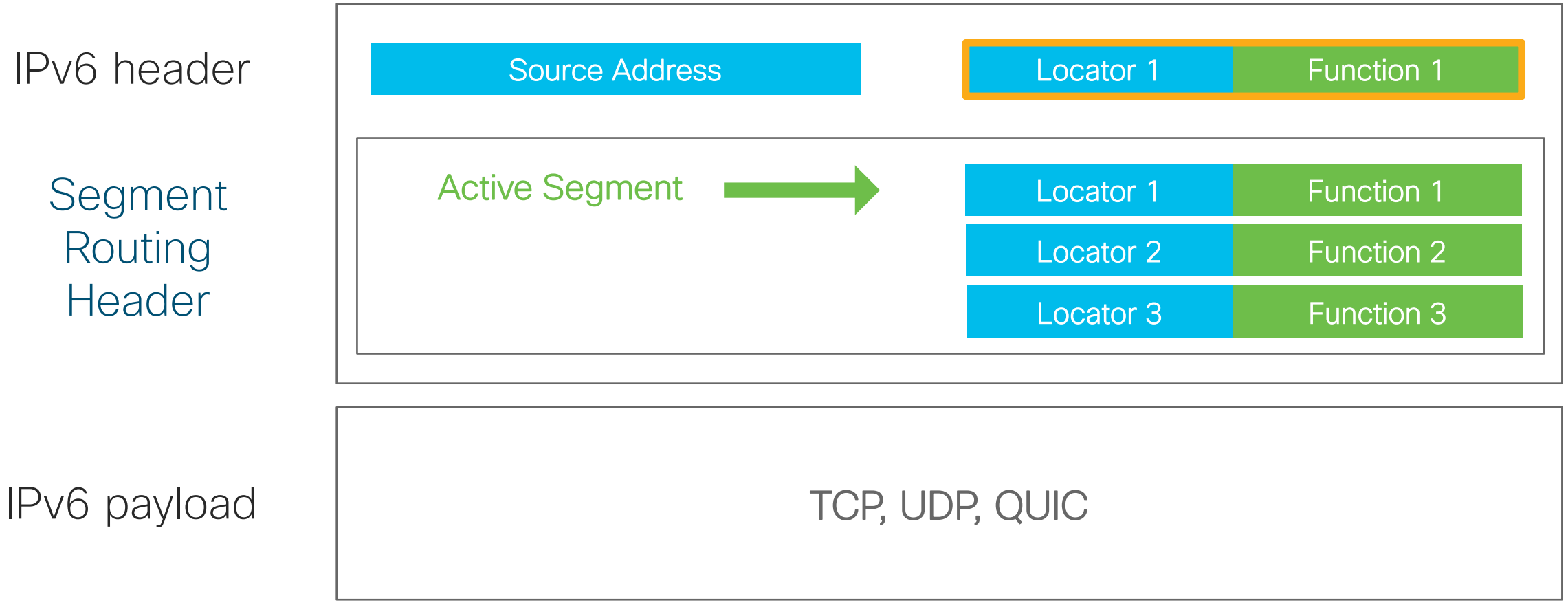
Network Program

Next Segment

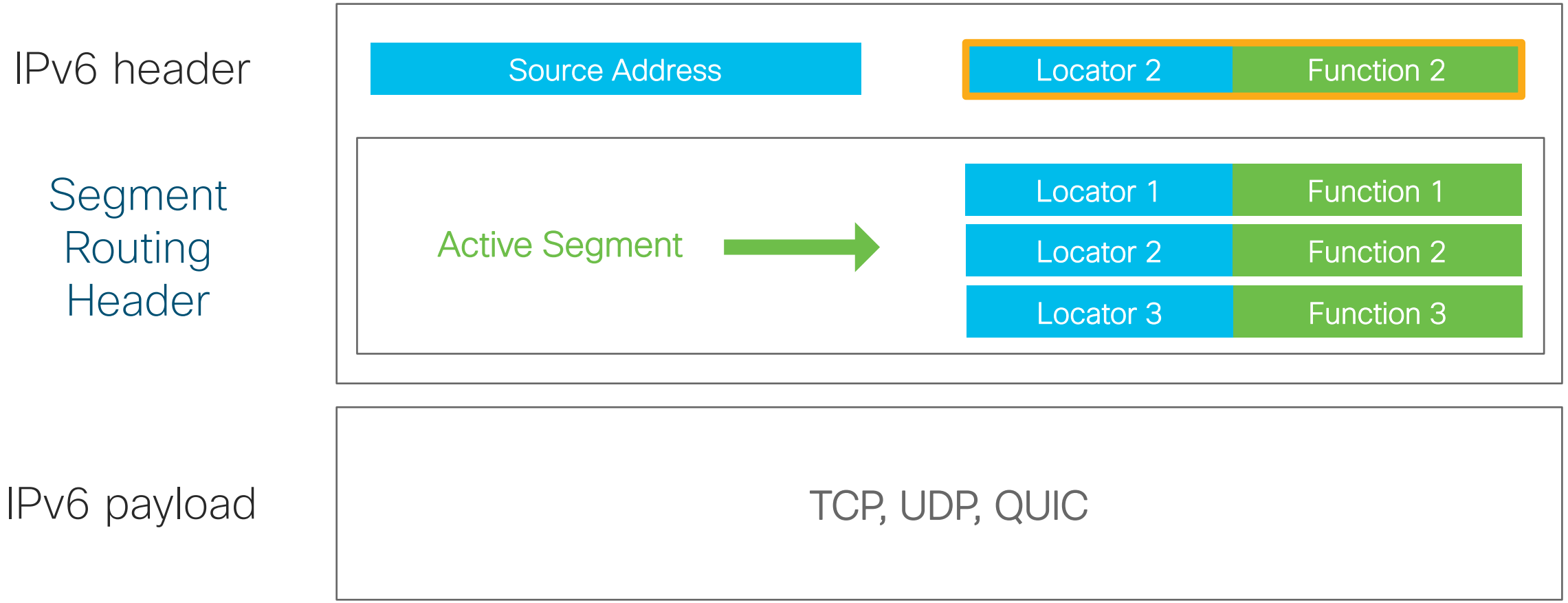
Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3



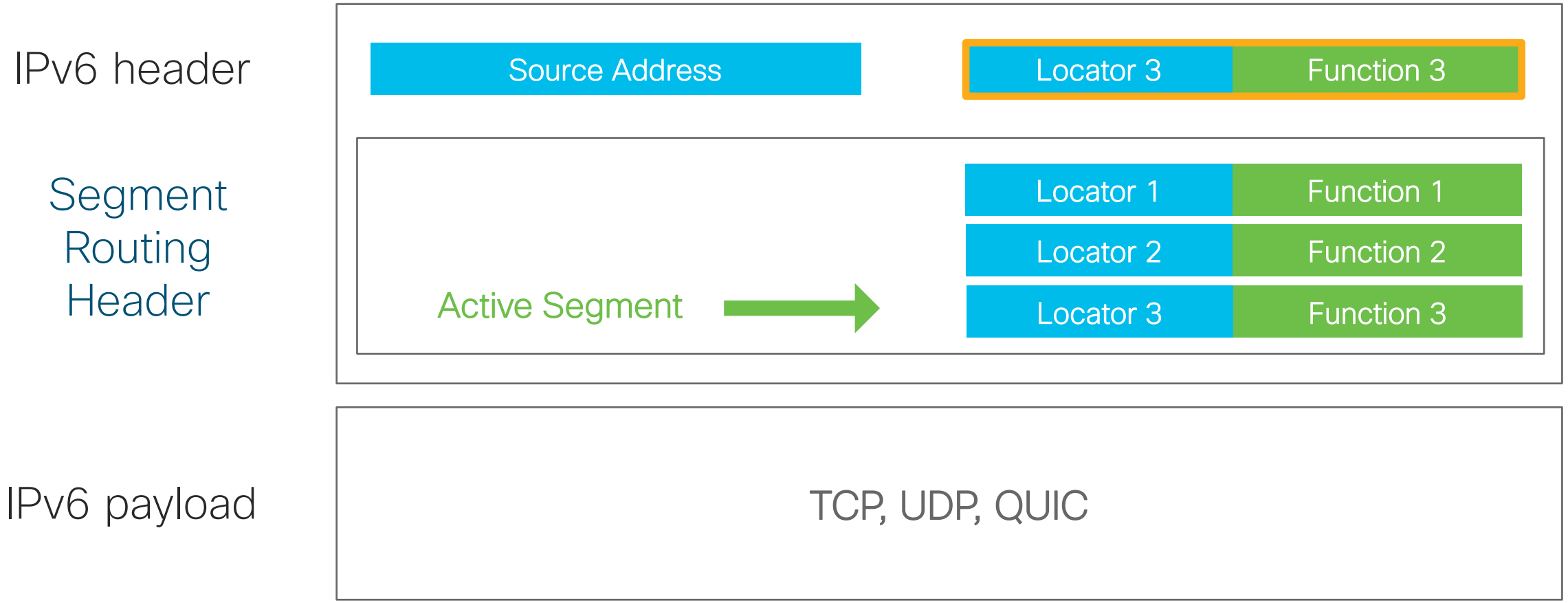
Network Program in the Packet Header



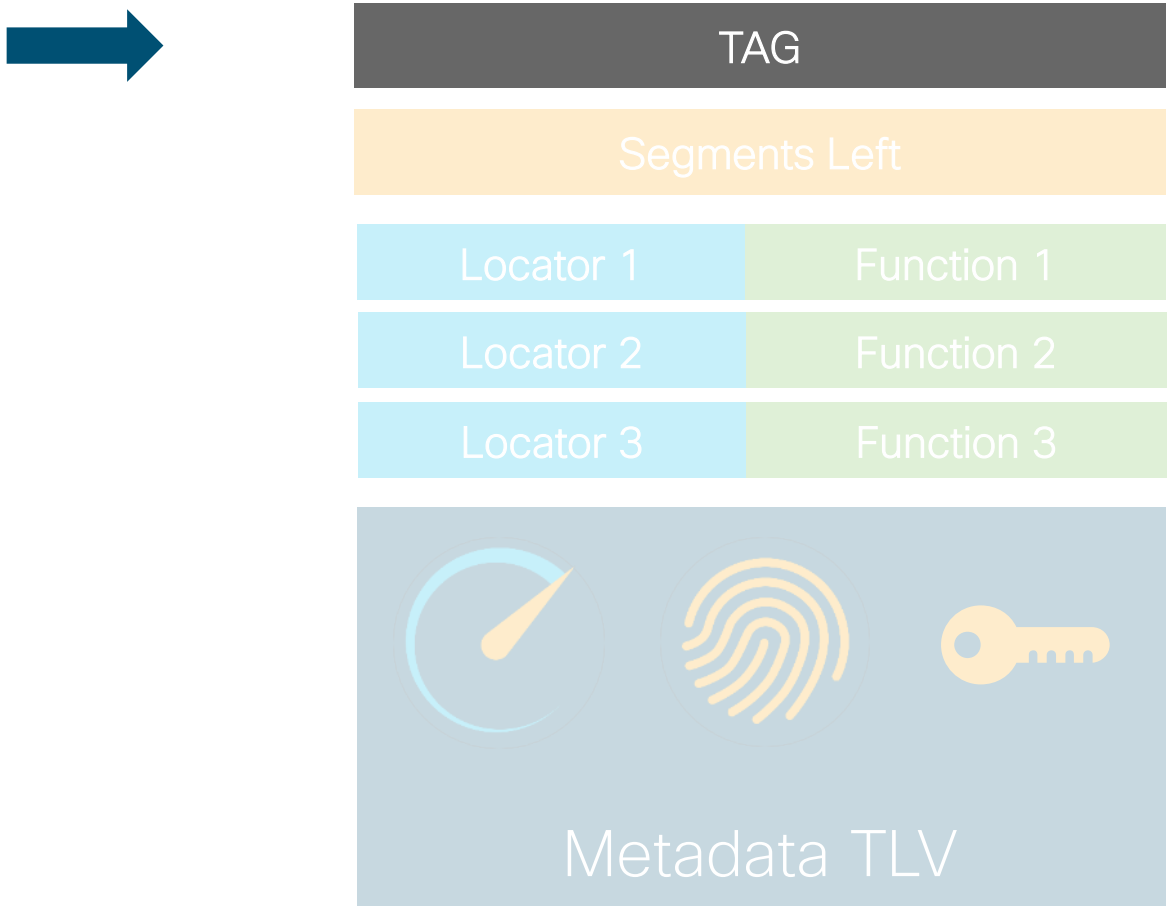
Network Program in the Packet Header



Network Program in the Packet Header

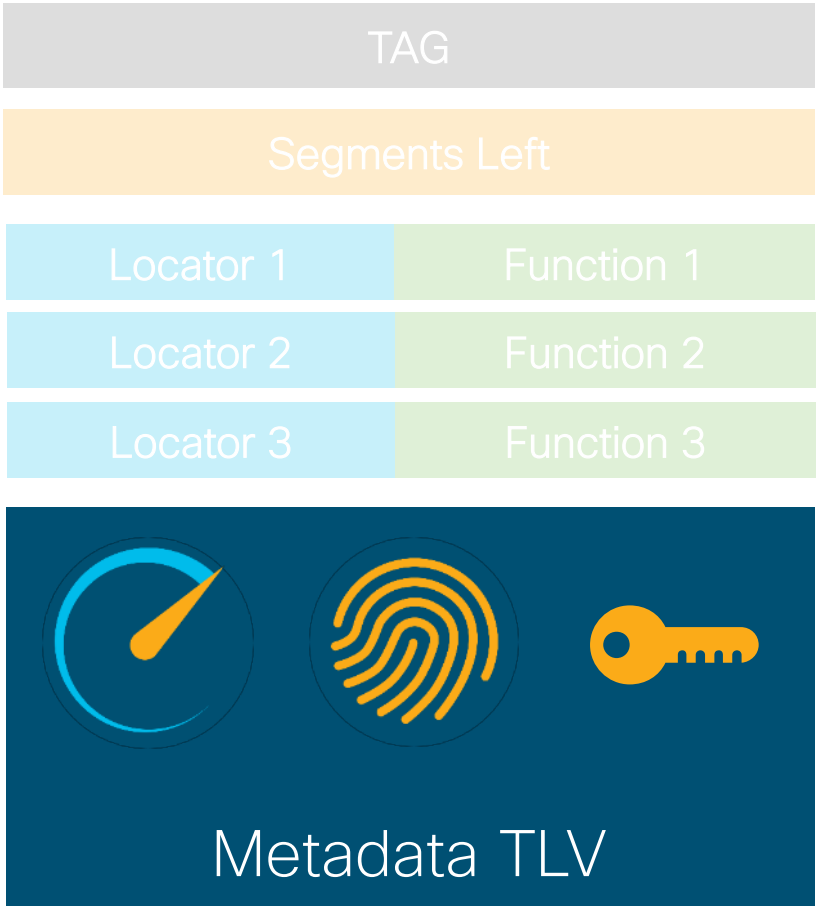


Group-Based Policy

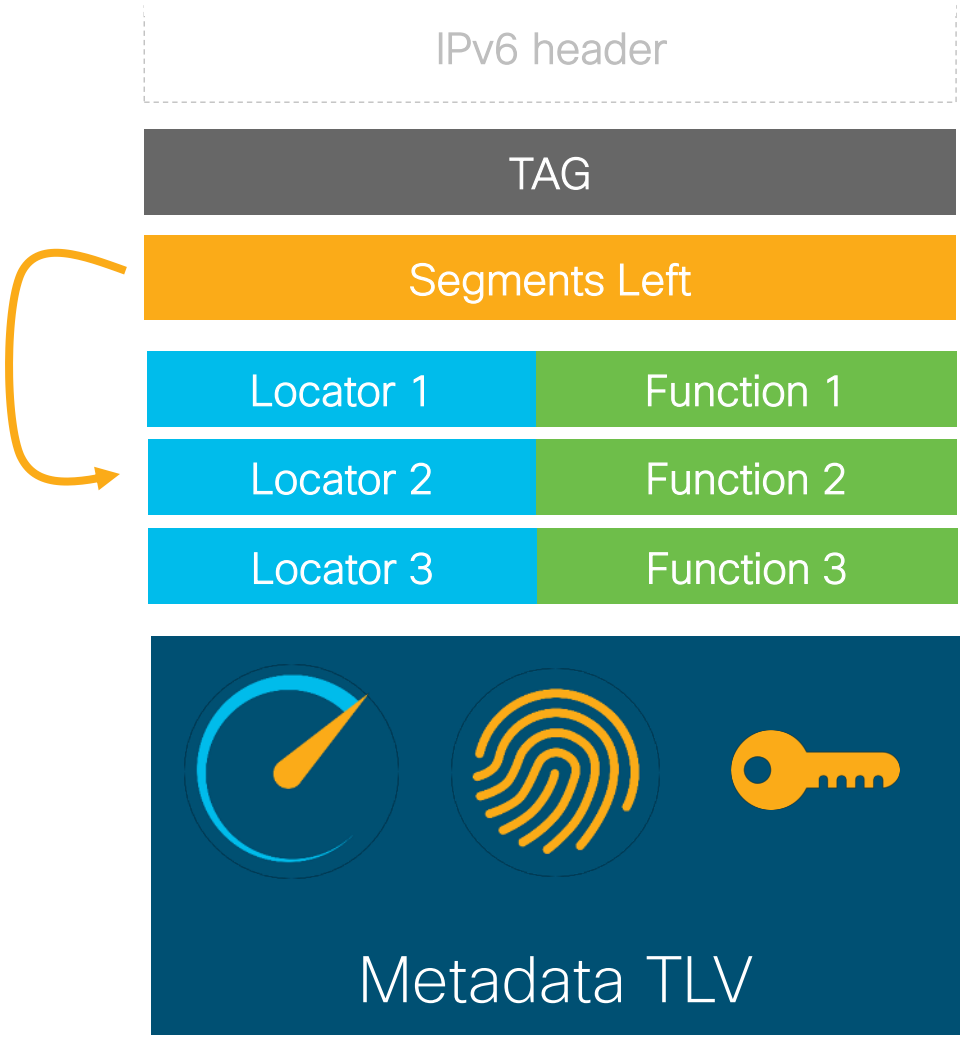


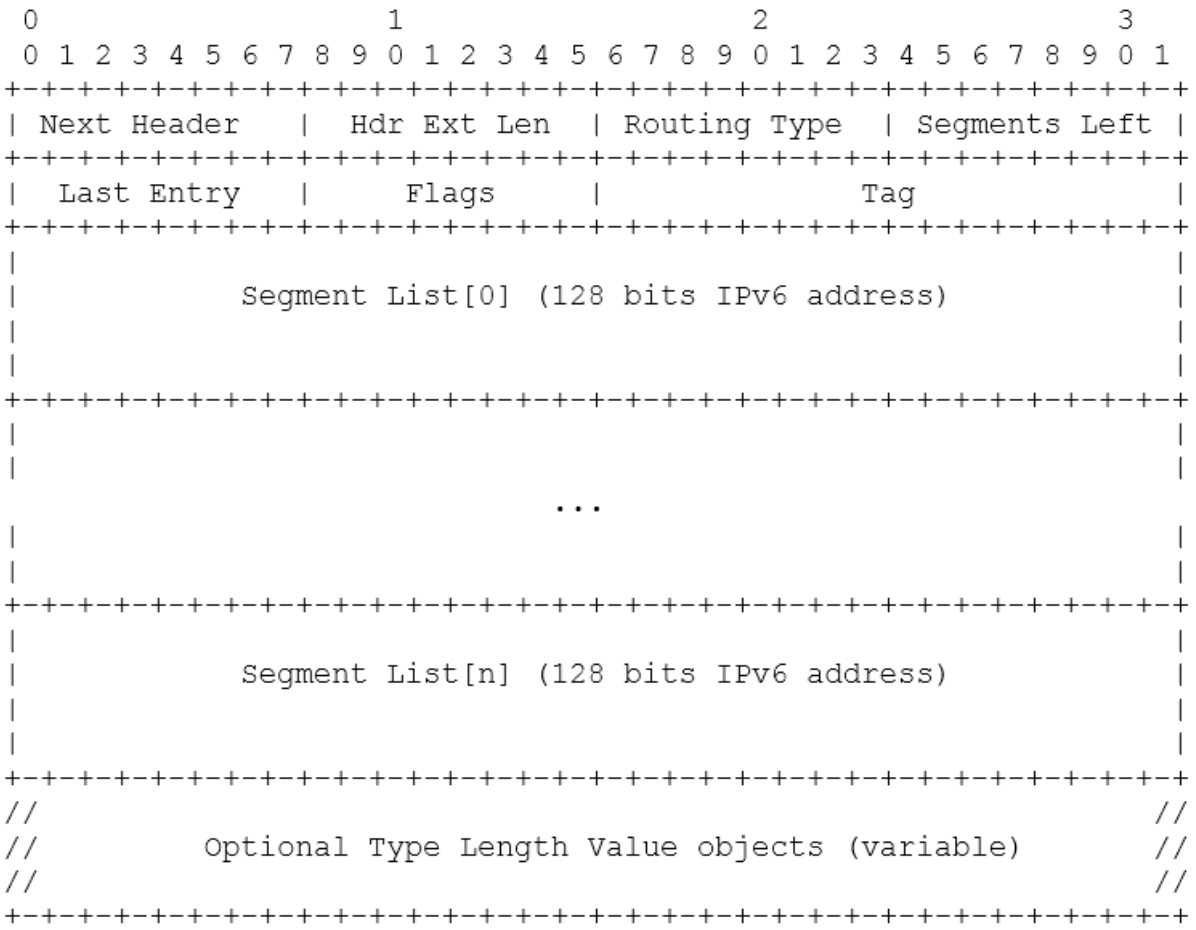
Argument shared between functions

“Global” Argument

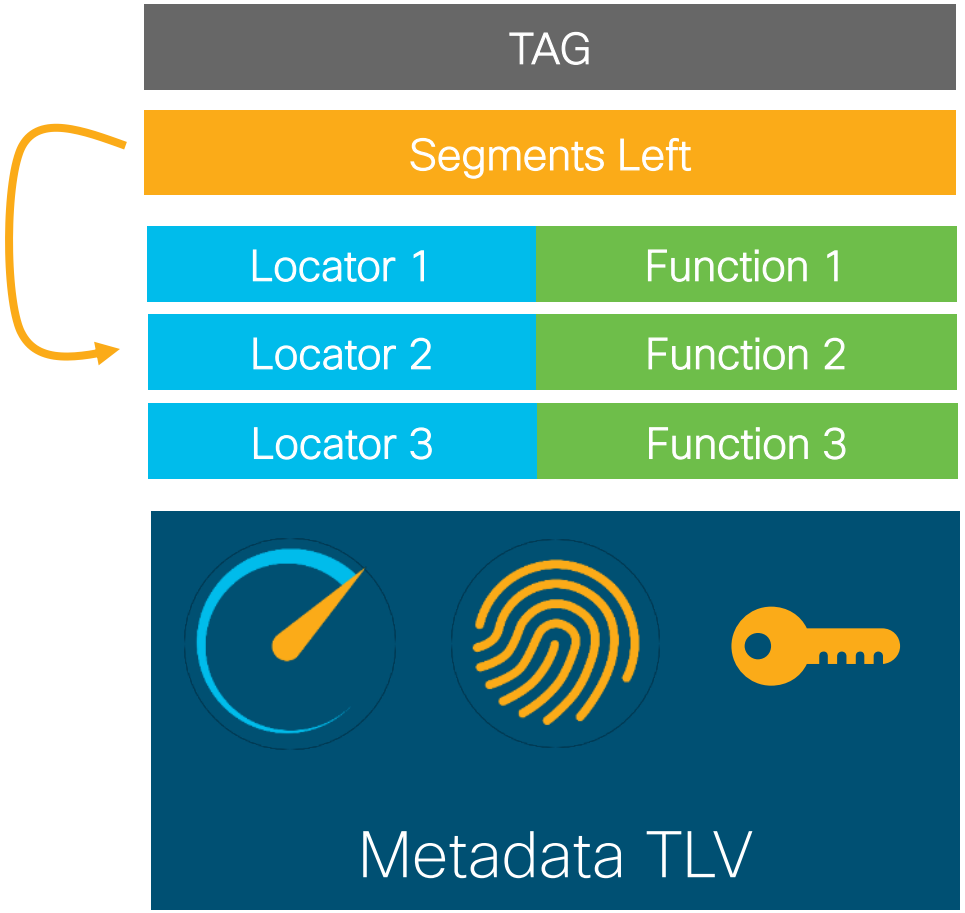


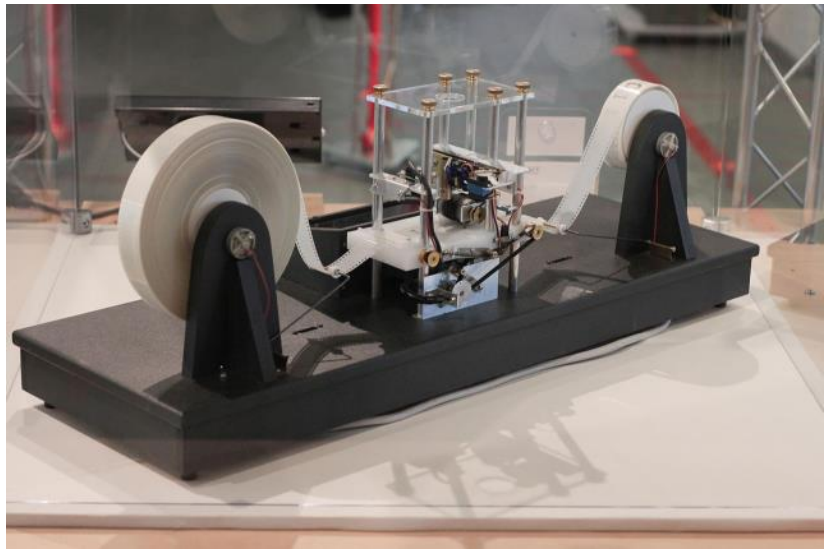
SRv6 Header





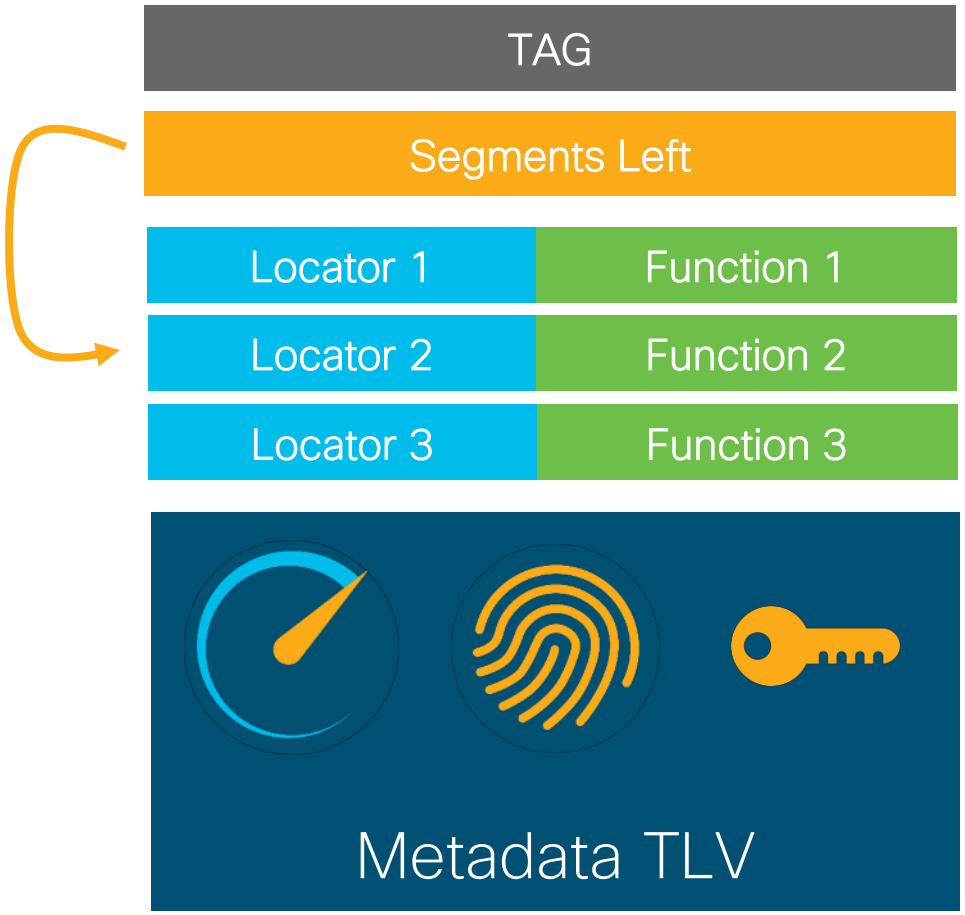
SRv6 for anything





Turing

SRv6 for anything



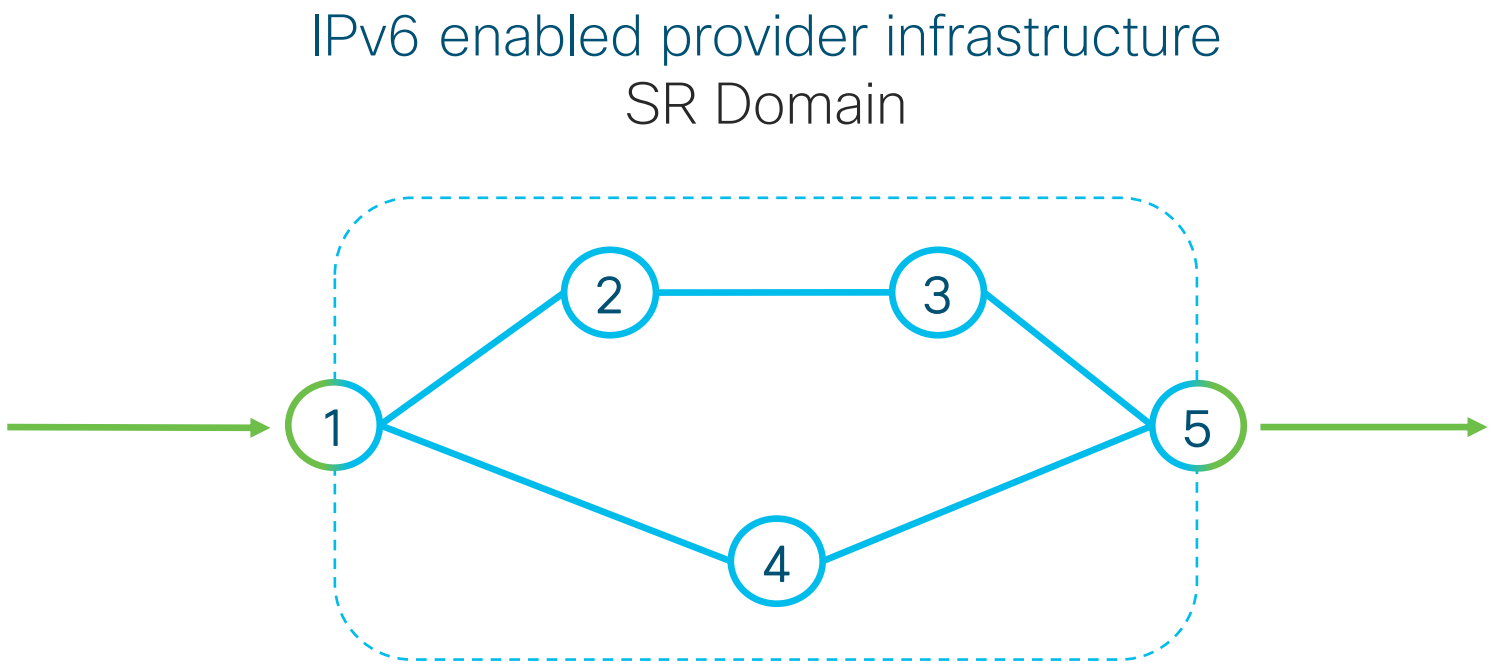
Optimized for HW processing

e.g. Underlay & Tenant use-cases

Optimized for SW processing

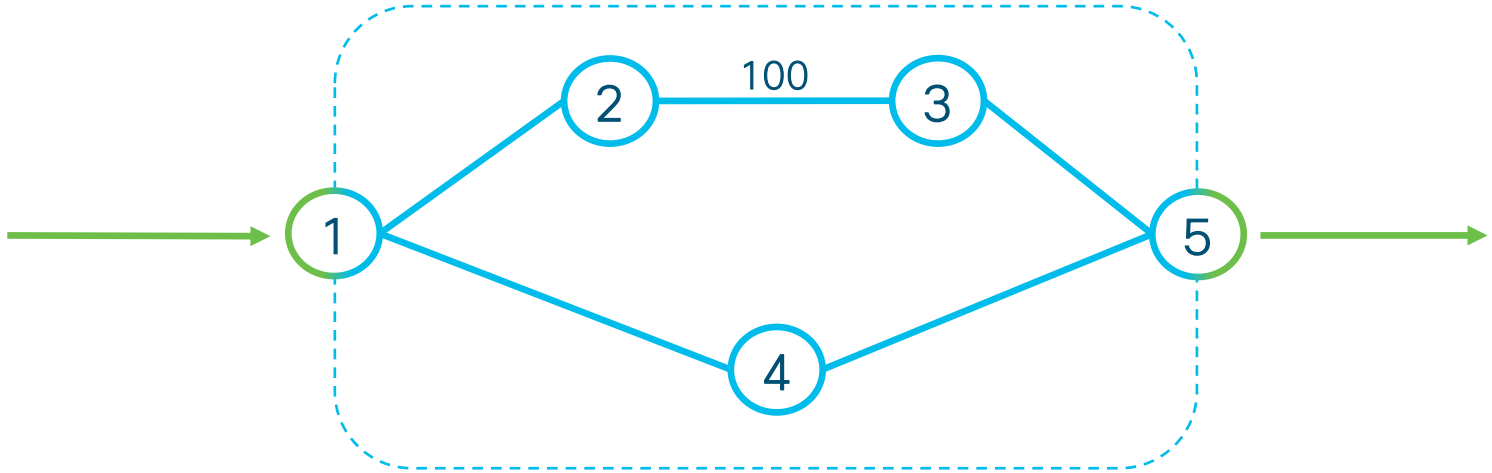
e.g. NFV, Container, Micro-Service

SRv6 Domain



Encapsulation at the Domain ingress

- IPv4, IPv6 or L2 frame is encapsulated within the SR Domain
- Outer IPv6 header includes an SRH with the list of segments

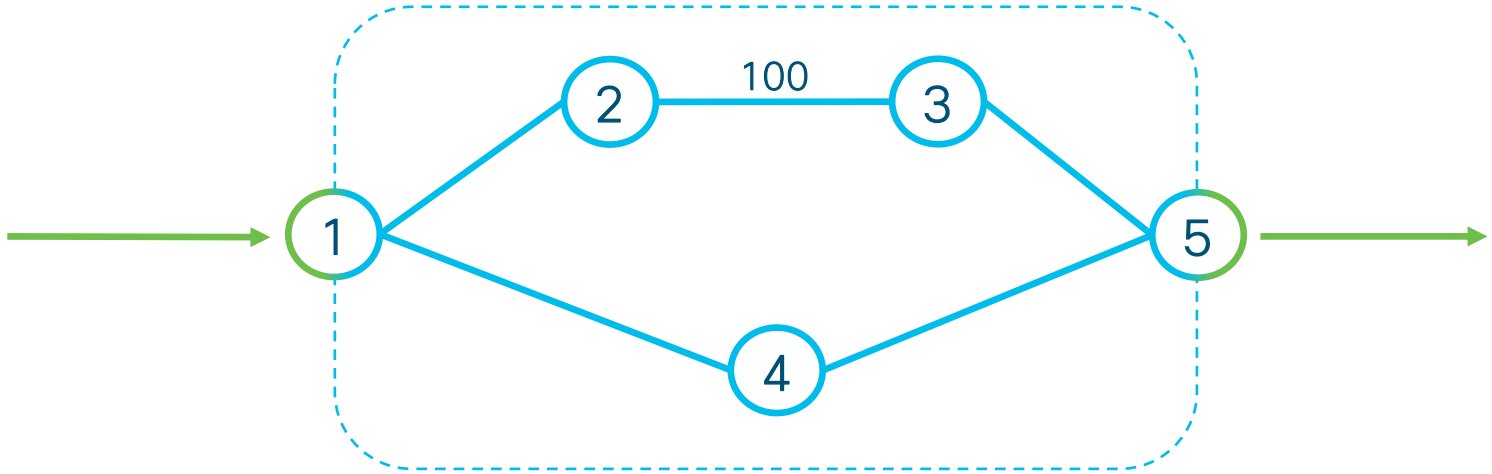


IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:2:C3::
SR	(B:0:0:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

SRH of the outer IPv6 encapsulation

- Domain acts as a giant computer
- The network program in the outer SRH is executed



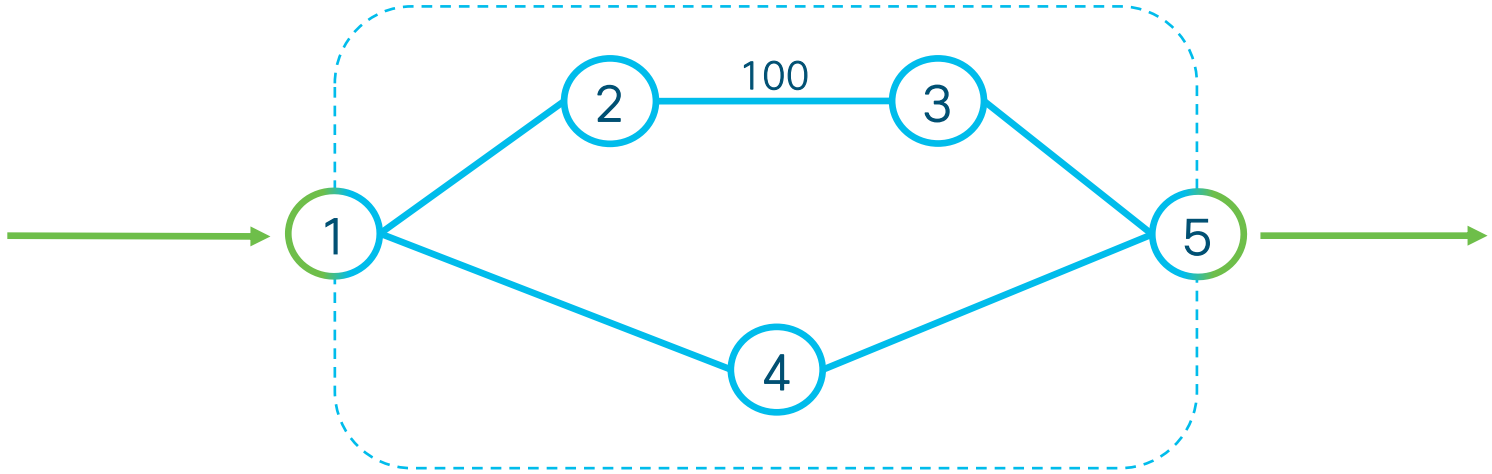
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:2:C3::
SR	(B:0:0:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:5:DB::
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

Decapsulation at Domain Egress

- Egress PE removes the outer IPv6 header as the packet leaves the SR domain



IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

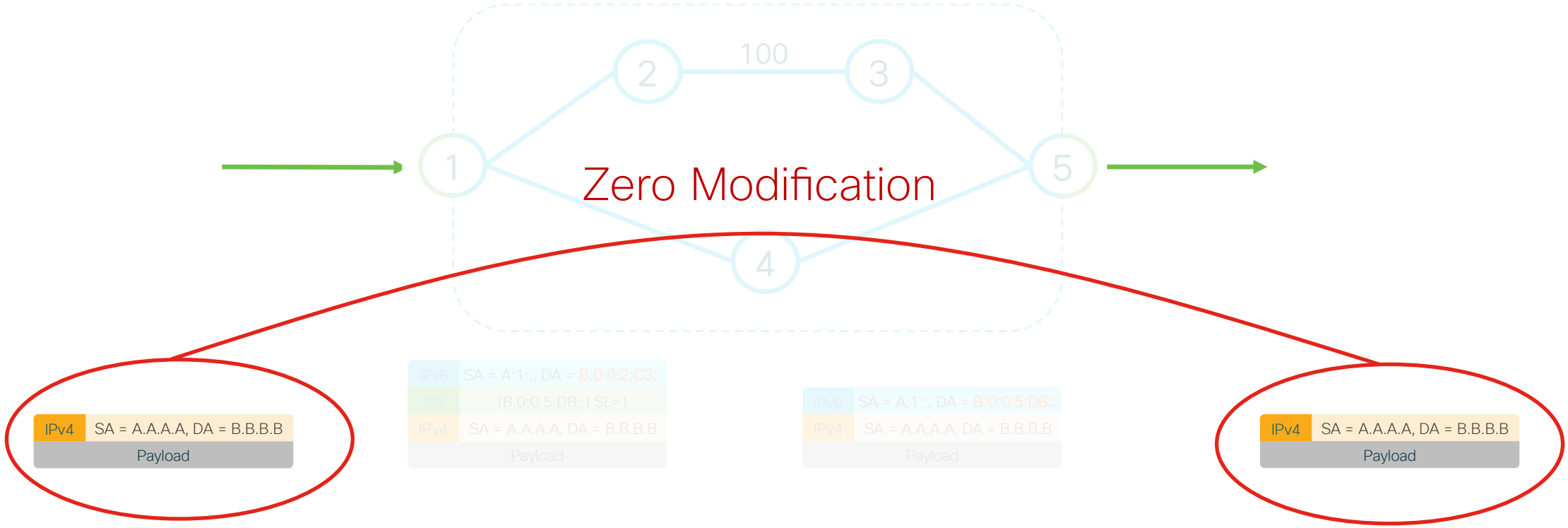
IPv6	SA = A:1::, DA = B:0:0:2:C3::
SR	(B:0:0:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:5:DB::
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

End-to-End Integrity

- End-to-end integrity principle is strictly guaranteed
  - Inner packet is unmodified
  - Same as SR-MPLS (MPLS stack is replaced by IPv6 outer header and SRH)



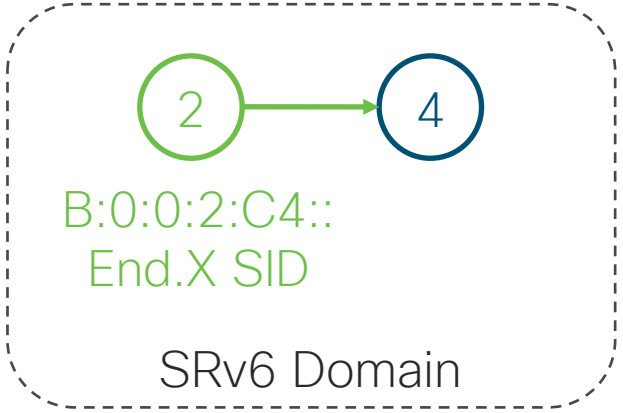
End and End.X SID behaviors

Illustration convention:

- IPv6 address of node k is A:<k>::
- SRv6 SID of node k is B:0:0:<k>:<function>::

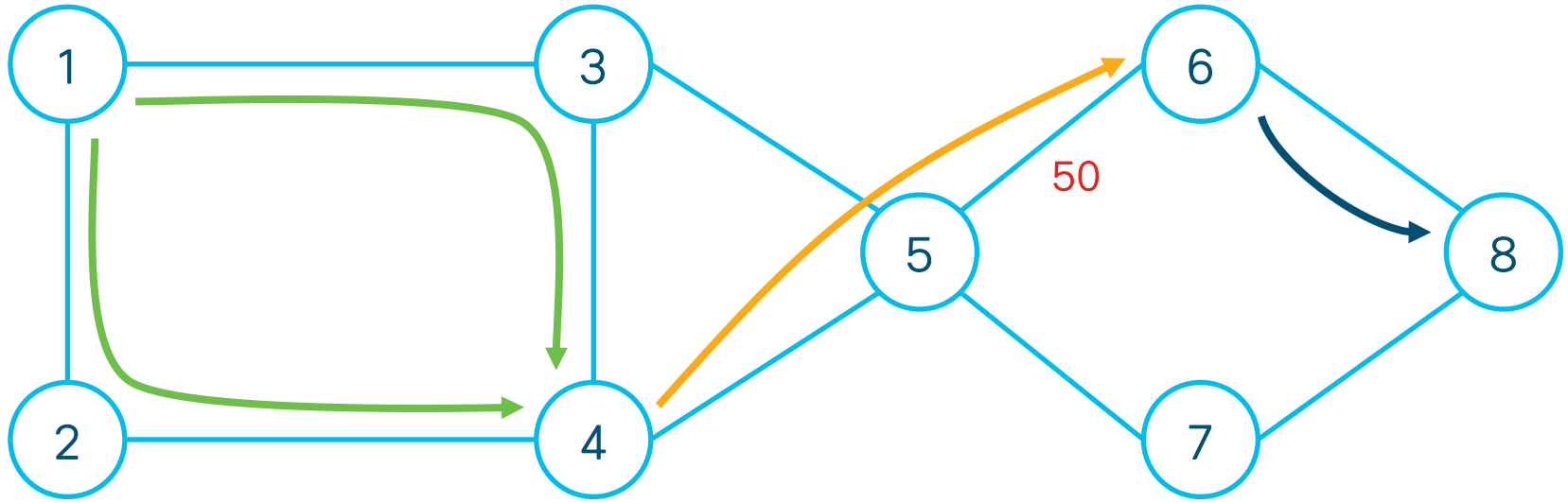
- End – Default endpoint behavior
  - shortest-path to the SID's endpoint
  - endpoint updates DA with next SID
  - endpoint forwards according to updated DA

- End.X – Endpoint with cross-connect
  - shortest-path to SID's endpoint
  - endpoint updates DA with next SID
  - endpoint forwards to interface associated with SID



Endpoint behaviors illustration

SR: ⟨ B:0:0:4:1::, B:0:0:5:C6::, A:8:: ⟩



Default metric 10

- B:0:0:4:1:: shortest path to node 4
- B:0:0:5:C6:: shortest path to node 5, then cross-connect towards 6
- A:8:: regular IPv6 address of node 8