You make **possible**

Towards Transport SDN (T-SDN)

Network Architecture & Operational Evolution, Simplification and Transformation for Next Generation Services

Kashif Islam, Solutions Architect, Cisco CX
Syed Hassan, Solutions Architect, Cisco CX

TECSPG-2735

Barcelona | January 27-31, 2020

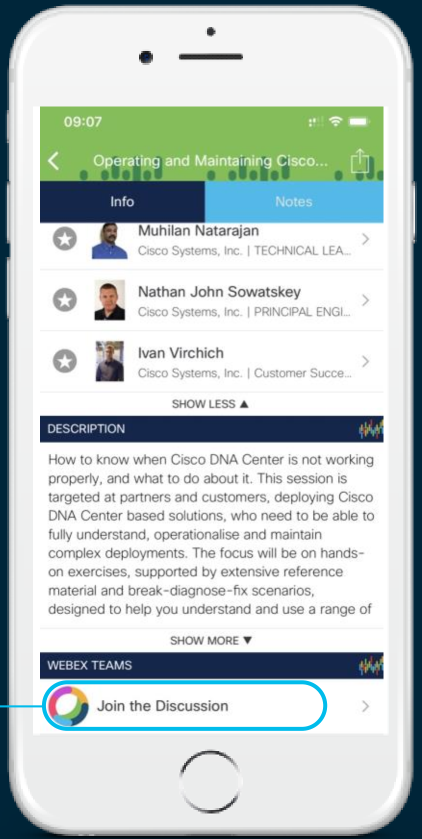
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



“Give me six hours to chop down a tree and I will spend the first four sharpening the axe”

Abraham Lincoln

16th U.S President

Growth Driving Network Transformation

Architecture and Operations Need Transformation Too!!

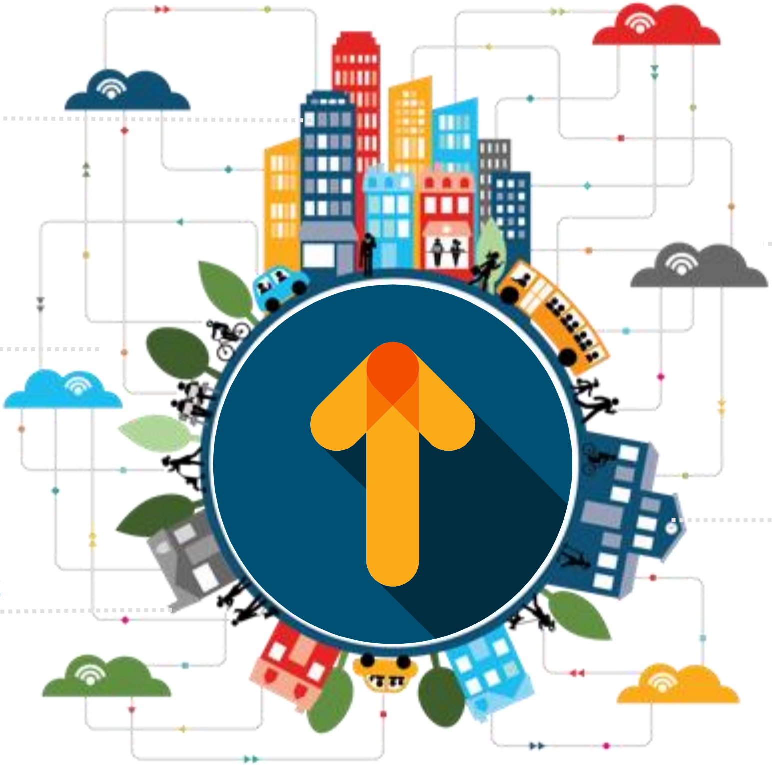
14.6B M2M

Connections by 2022

(51% of total)

4.8B Internet Users

3.6x more devices than people



28.5B

Connected Devices

3.6 devices/person

85gbps pp/mo

More !!!!

Avg Broadband Speed to Double!!

5.7 billion mobile users in 2022**

3X global traffic increase**

**Cisco VNI 2019

Agenda

Today's transport networks have organically grown to support a multitude of technologies, features and protocols. More than 2 decades of technology innovations have built layers upon layers on protocols, architectures and services. A lot of this existing, but quickly becoming legacy technologies, are a hinderance to extending Software Defined Networking capabilities all the way down to Access, Aggregation and Core networks.

This session will explore technological transformation taking place on transport network level and how that ties into application driven, software defined networks.

First half of the session will explore Segment Routing as the driving force behind this transformation and incorporating Segment Routing Path Compute Element (SR-PCE) to enhance functionality and simplify network architectures. This part will also discuss strategies and scenarios for network simplification, removing unwanted protocols such as BGP Labeled Unicast while still being able to provide an end-to-end Inter and Intra-AS Service path through strategic placement of SR-PCE and use of various features and functionalities supported by Segment Routing (such as Egress Peer Engineering, BGP-LS, PCEP and others). This part of the session will also cover co-existence of legacy MPLS-LDP and Segment Routing technologies while planning a network transformation to a pure End-to-End Programmable network infrastructure

Second half of this session will solely focus on utilizing the underlying "Programmable" Infrastructure to create a true Software Defined Transport Network, where various application may be used to provide service orchestration, monitoring, path visualization, intent-definition, remediation and on-demand path calculation between service nodes on the transport network. This part of the session will solely focus on creating an "Application Driven" network that utilizes the tools discussed in first half, but solely uses higher layers application to abstract the networking layer while establishing, deploying, monitoring and refining service "intent" as needed, right from the application layer.

Agenda

- Architectural Transformation Landscape
- Preparing an SDN Ready, Programming Transport
- Ensuring Services and Architectural Parity
- Introduce Intent Based Path Control
- Routing and Architectural Simplification Examples
- New Features, New Services
- Operational Efficiency Through Orchestration and Automation
- Software Driven Usecases
- Summary

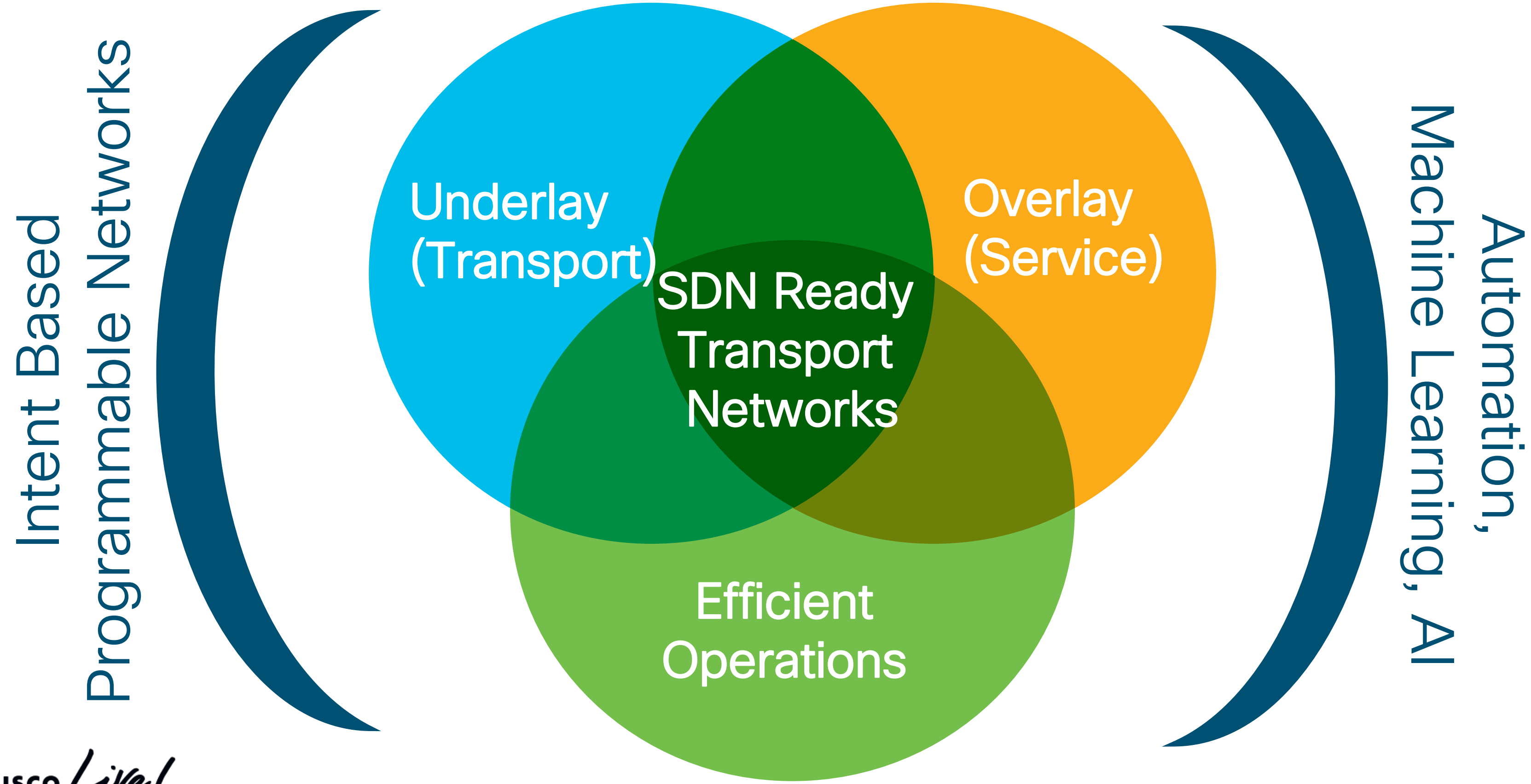
Architectural Transformation Landscape

Happening Now: Network Transformation

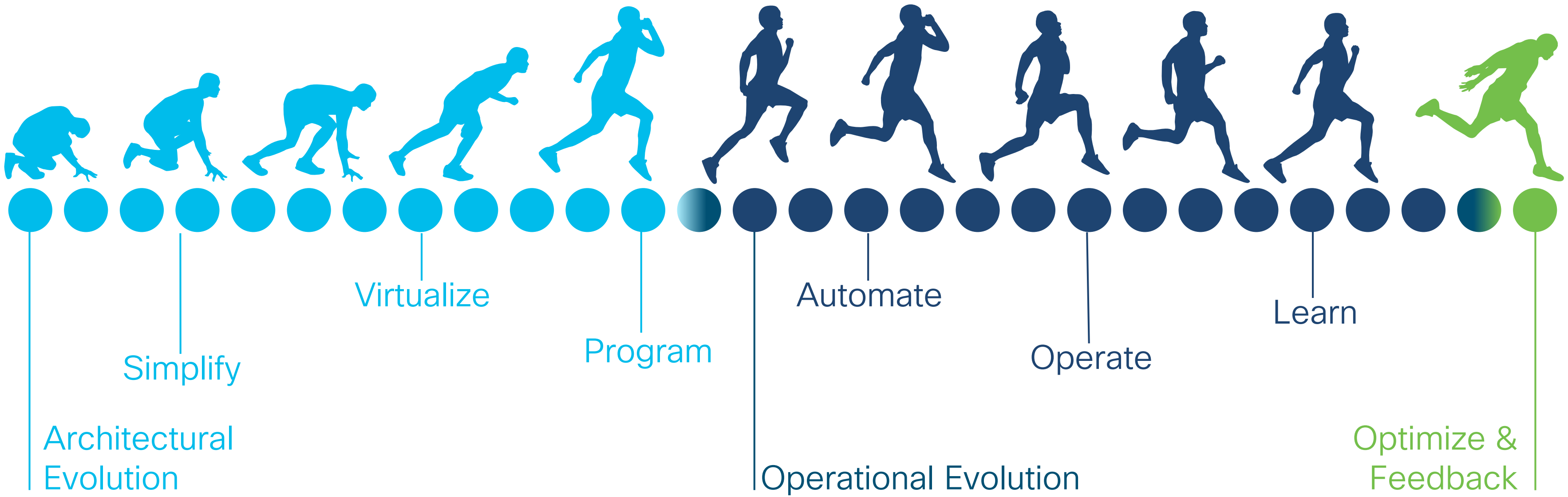
Architecture xFormation → Intent Based Programmable Infrastructure | Embrace Mobile | NG Products | Increase Revenue

Operations xFormation → Efficient Operations | Data and Analytics | Improve Experience | Reduce Cost

Wholistic Network View – Summary



Get Your (Transformation) Priorities Straight



It all starts with Simplification …. at all layers of the Network

Network Evolution and Simplification Journey

	Single Domain		
Technology Arch.	IP/MPLS		
Provisioning			
Programmability			
Services (L2/L3 VPN)	LDP, BGP		
Scaling Mechanism			
TE, FRR	RSVP		
Overlay Protocol	LDP		
Connectivity Protocol	IGP		

Network Evolution and Simplification Journey

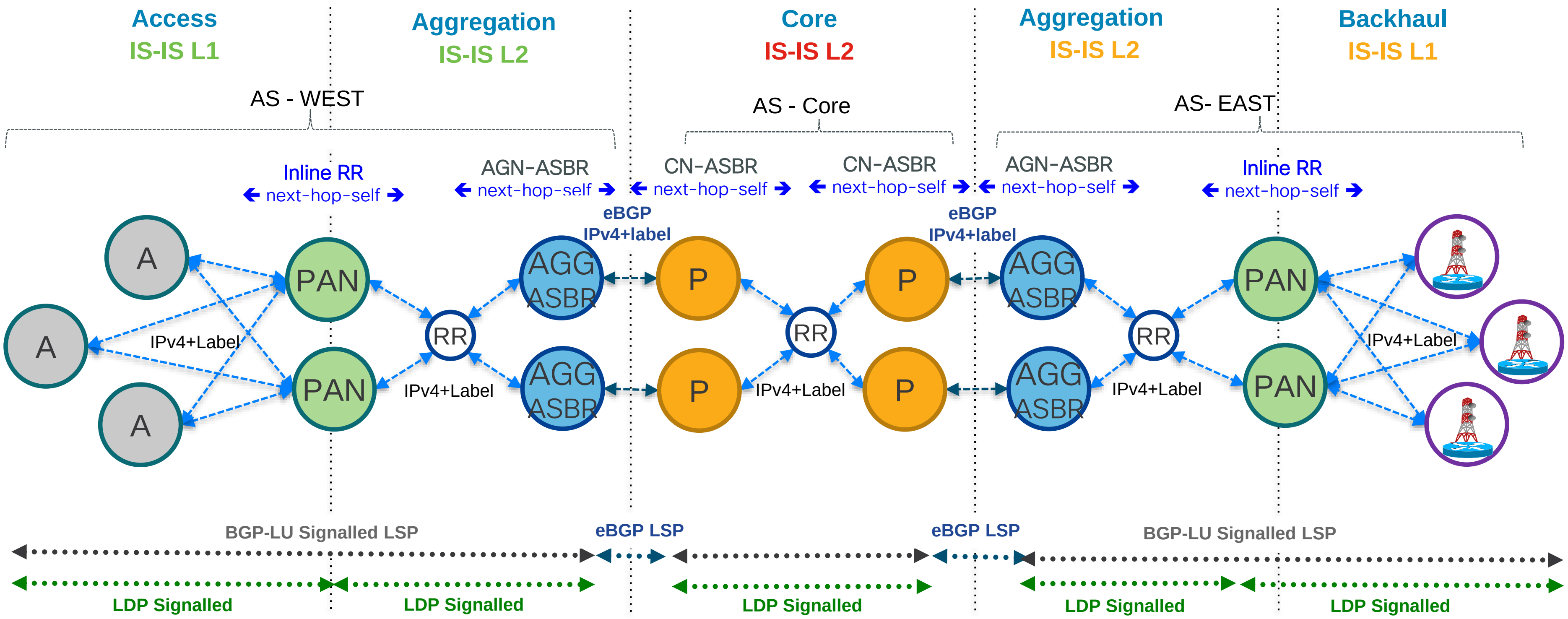
	Single Domain	Multi-Domain	Programmable
Technology Arch.	IP/MPLS	Unified MPLS	
Provisioning			
Programmability			
Services (L2/L3 VPN)	LDP BGP	LDP BGP	
Scaling Mechanism		BGP-LU	
TE, FRR	RSVP	RSVP	
Overlay Protocol	LDP	LDP	
Connectivity Protocol	IGP	IGP	

Network Evolution and Simplification Journey

	Single Domain	Multi-Domain	Programmable
Technology Arch.	IP/MPLS	Unified MPLS	Segment Routing
Provisioning			NETCONF, YANG
Programmability			Path Control Element (PCE)
Services (L2/L3 VPN)	LDP, BGP	LDP, BGP	BGP
Scaling Mechanism		BGP-LU	Segment Routing w/ IGP
TE, FRR	RSVP	RSVP	
Overlay Protocol	LDP	LDP	
Connectivity Protocol	IGP	IGP	

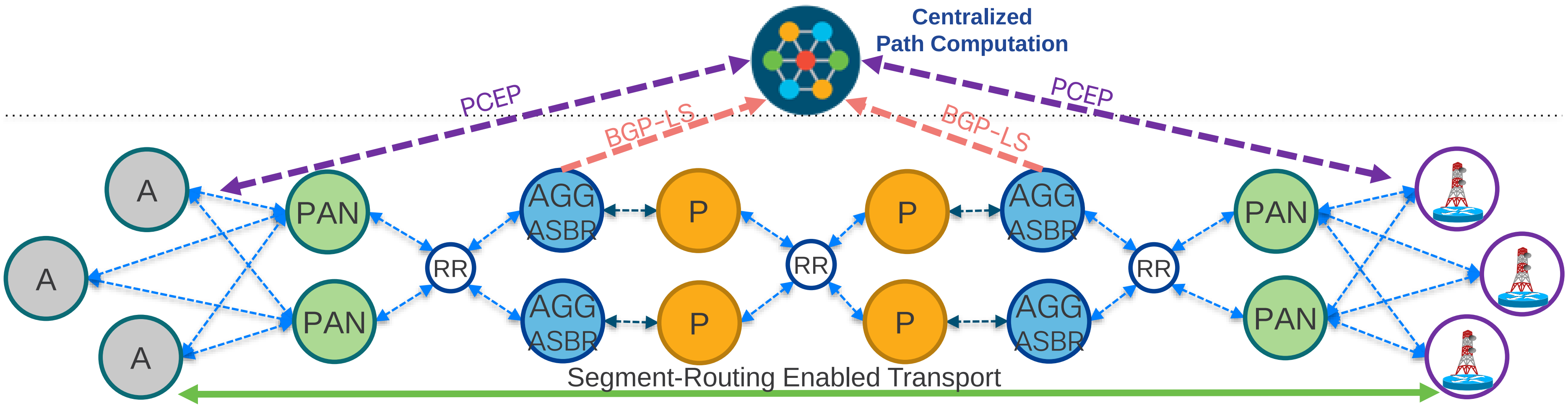
Where We Are ...

Current Unified MPLS Baseline



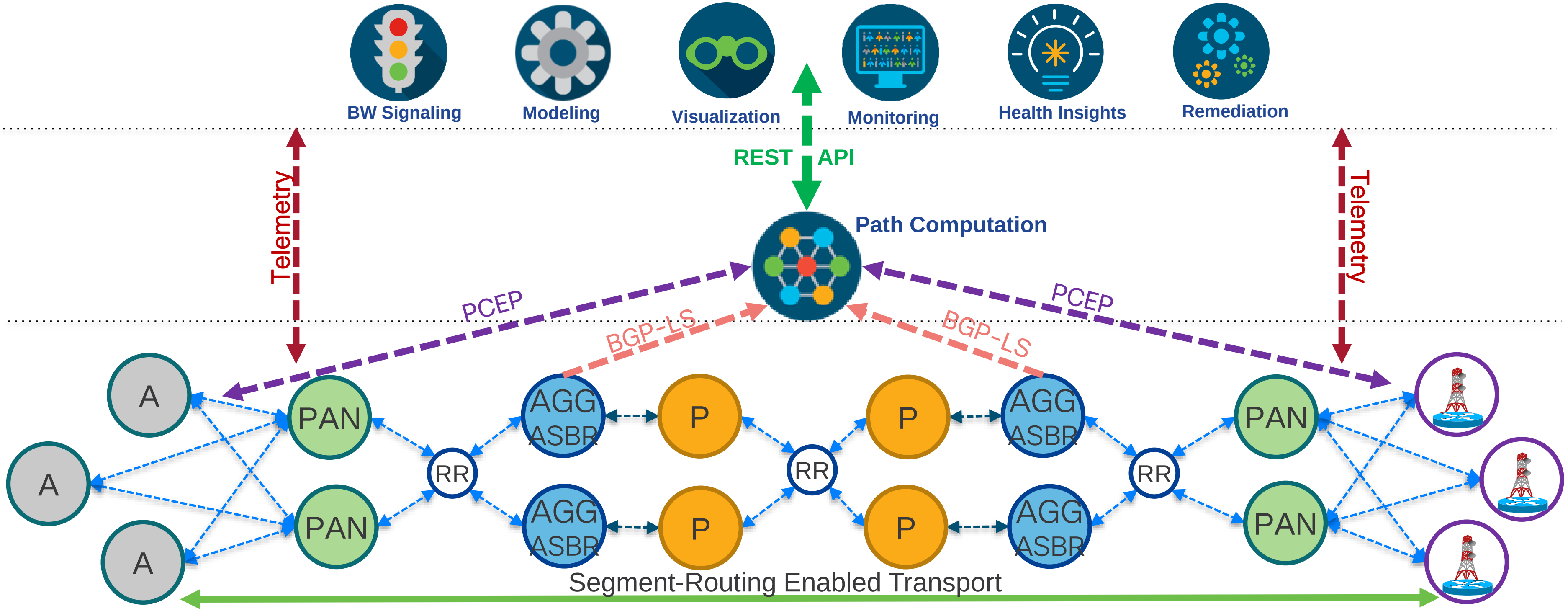
Where Do We Want to Go...

Intent Based Software Defined Transport Network (SDTN)



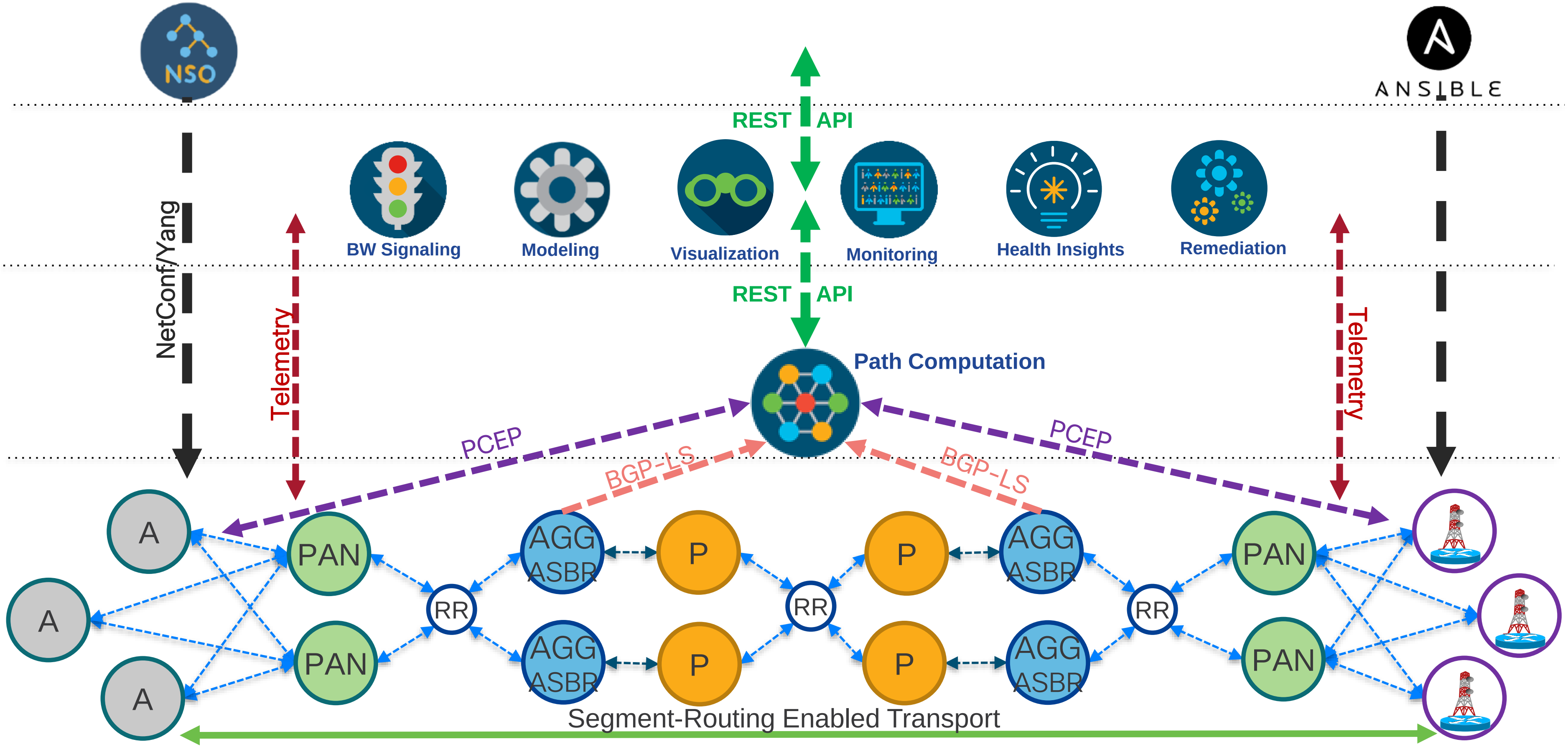
Where Do We Want to Go...

Intent Based Software Defined Transport Network (SDTN)



Where Do We Want to Go...

Intent Based Software Defined Transport Network (SDTN)



How Do We Get There?

Multi-Step Network Evolution

Pave the Path to Simplification
(Road to simplification is complex)

Simplify

Enhance

Monetize

How Do We Get There?

Multi-Step Network Evolution

Pave the Path to Simplification

1. Architectural and Service Parity: Programmable Transport Underlay
2. Introduce Central Intent Based Path Control

Simplify

3. Simplify Routing Design

Enhance & Monitize

4. New Features to Support Value Add Services
5. Ready for Software Driven Usecases

Preparing an SDN Ready, Programmable Transport Underlay

Programmable Network Infrastructure – How?

Network Forwarding – Today	• Routed based on Next Hop Forwarding i.e No route control once packet leaves the source Optimal path (mostly) but not suited for “Network Programmability”

Programmable Network Infrastructure – How?

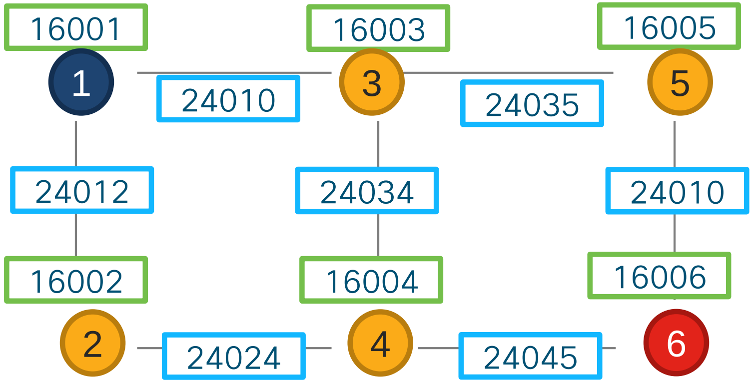
Network Forwarding – Today	• Routed based on Next Hop Forwarding i.e No route control once packet leaves the source Optimal path (mostly) but not suited for "Network Programmability"
Network Forwarding – Intent Based and Programmable	• Ability to define "Intent" for traffic • Source Influences Exact Traffic Path • Requires Source Routing for MPLS

Segment Routing – Technology Overview

- Segment ID (SID) is used as label in MPLS-SR
- Globally unique Prefix-SID identifies the router
- Locally unique Adjacency-SID identifies link on a router
- Simple extension to IS-IS or OSPF to propagate SIDs through the network
- Builds & Maintains "Segment"

Segment Routing – Technology Overview

Prefix SID (Node SID)	Adjacency SID
Globally Significant	Locally Significant Unidirectional



Segment = **Instructions** such as
"go to node N using the shortest path"

Segment Routing – Configuration Concepts

- Configured under IGP Routing Protocol
- Requires: Enabling SR & Configuring Prefix-SID
  - Configure “Absolute Value” or “Index”
- Optional: Configure SR-Global-Block (SRGB).
  - Default 16000 - 23999
- SRGB & Index advertised using IGP

Segment Routing Configuration Example

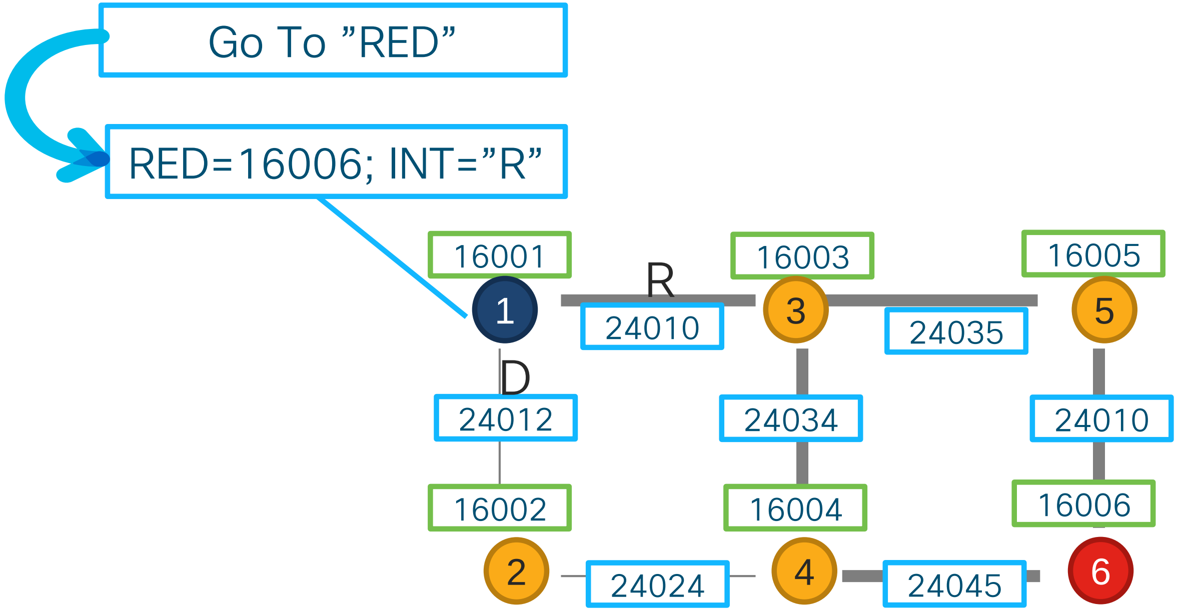
```
router isis 1
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls [sr-prefer]
 !
interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid index 1
!
```

Wide Metrics

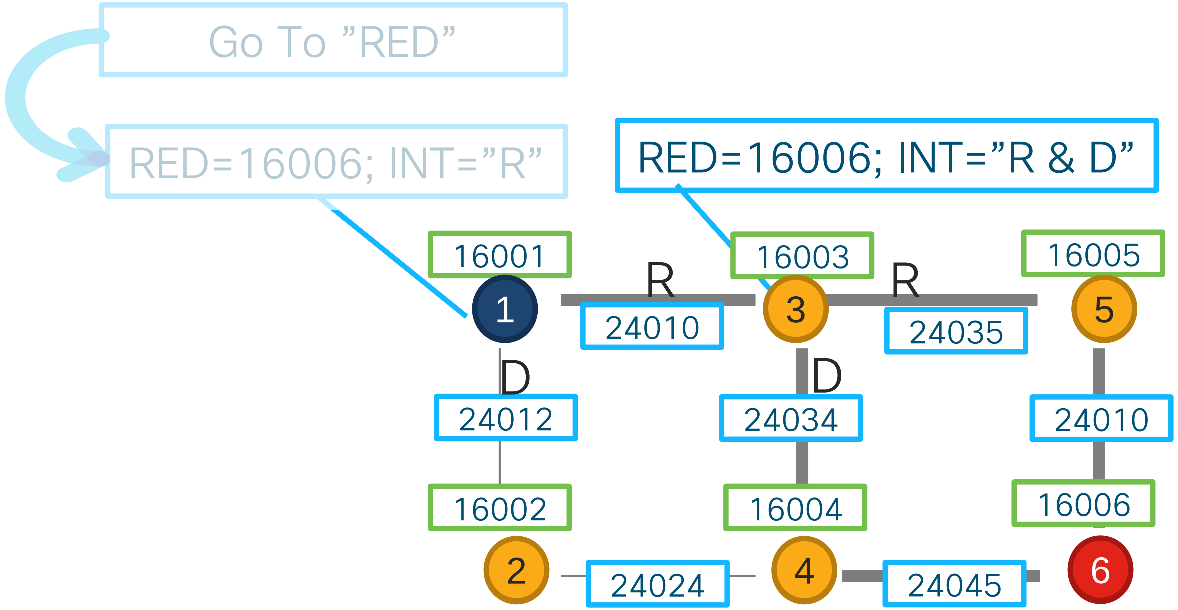
enable SR IPv4 control plane and SR MPLS data plane on all ipv4 interfaces in this IS-IS instance

Ipv4 Prefix-SID value for loopback0 (Index translate to 16001 absolute vlaue)

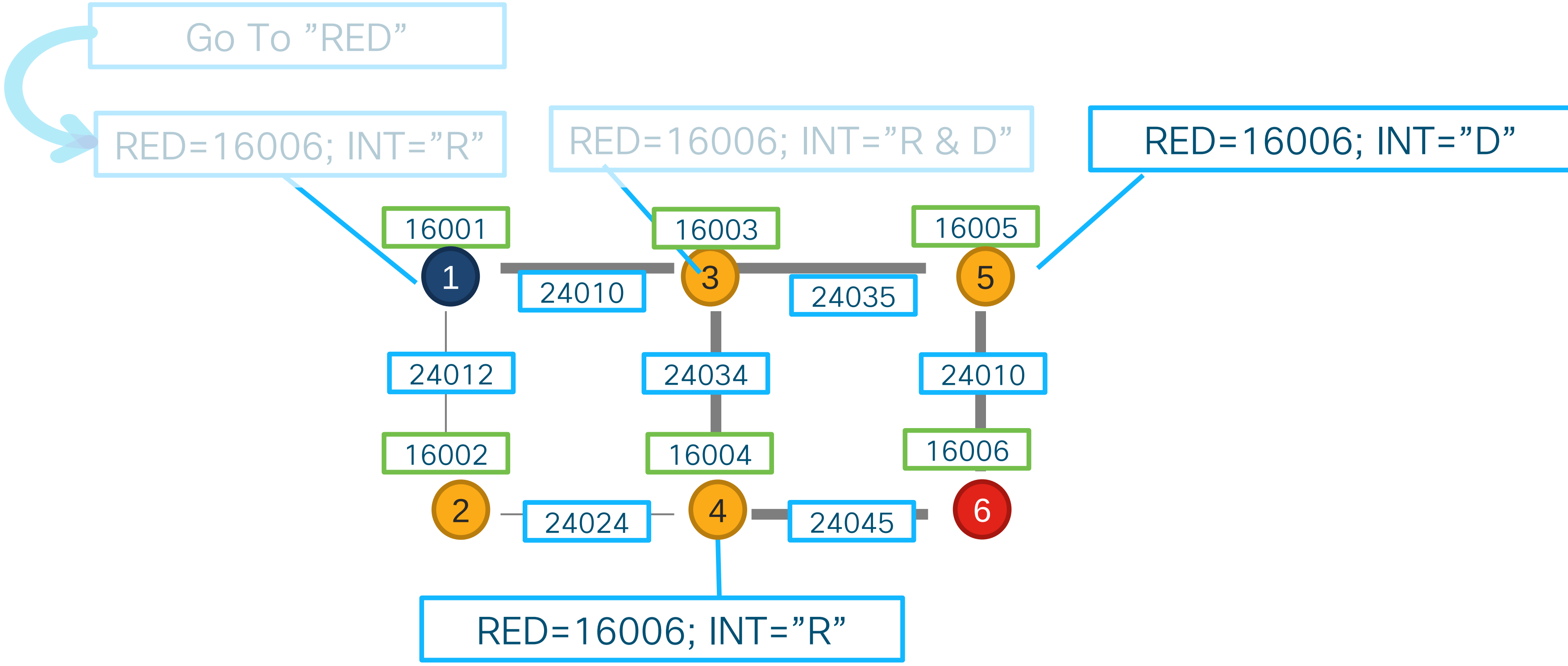
Segment Routing – Technology Overview



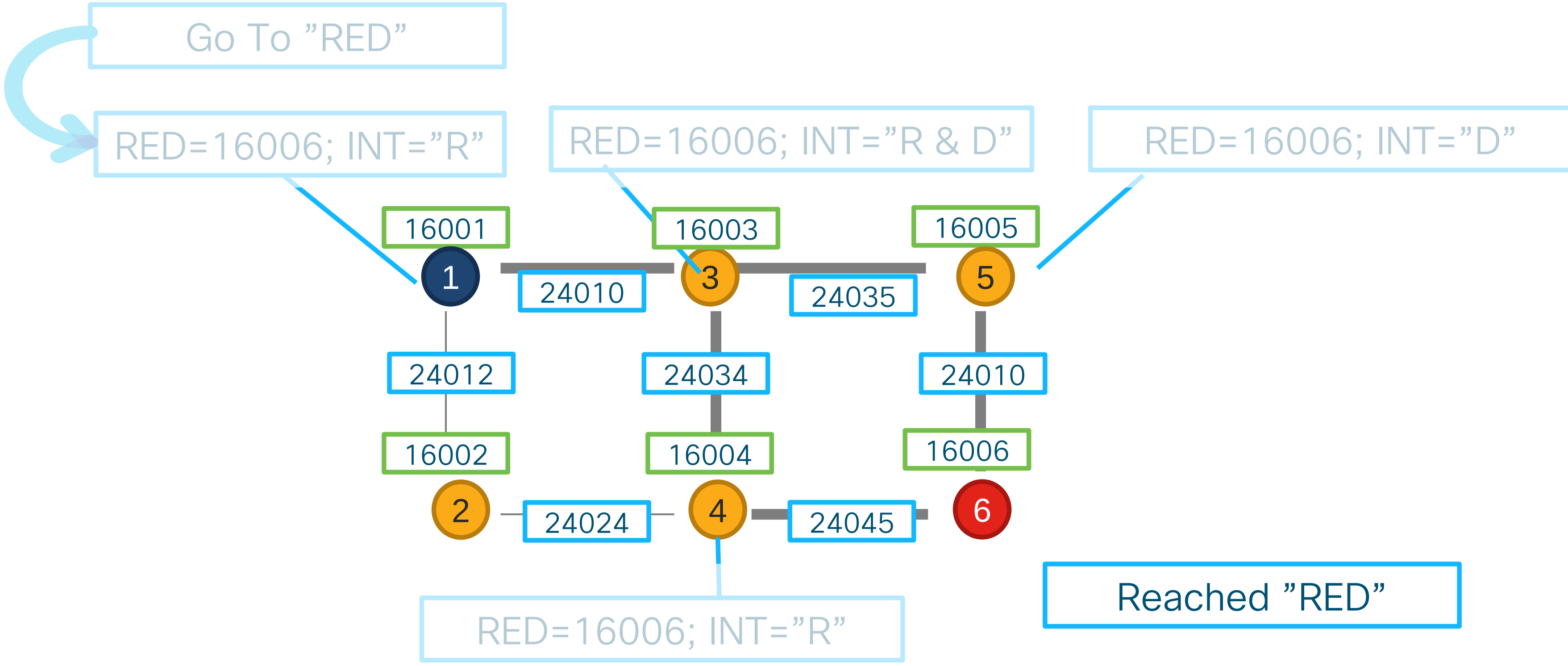
Segment Routing – Technology Overview



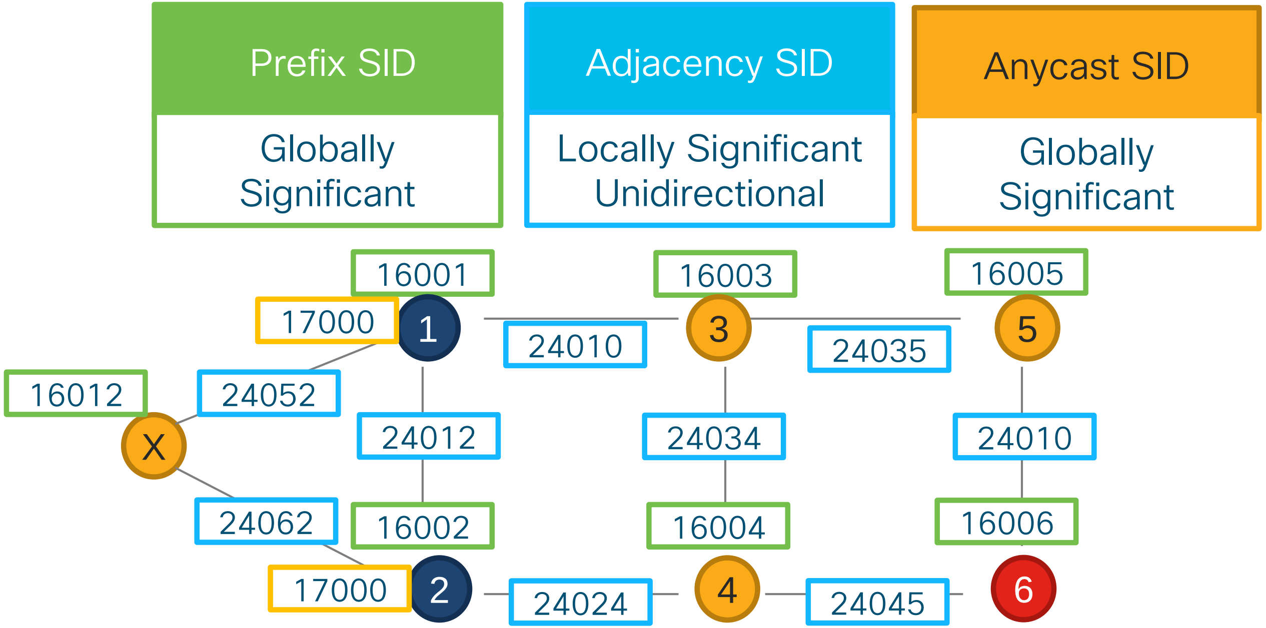
Segment Routing – Technology Overview



Segment Routing – Technology Overview

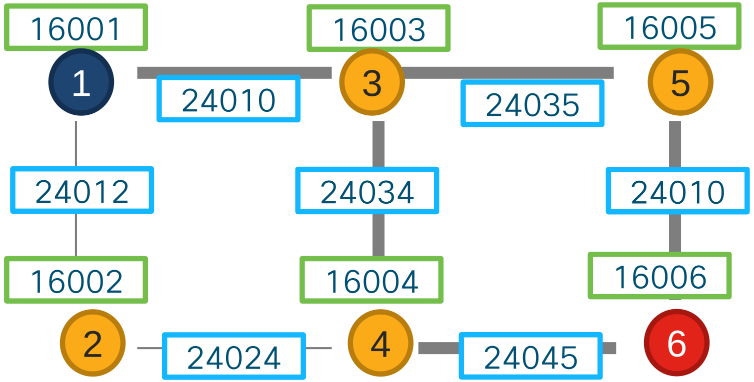


Segment Routing – Technology Overview



Segment Routing – Programming The Path

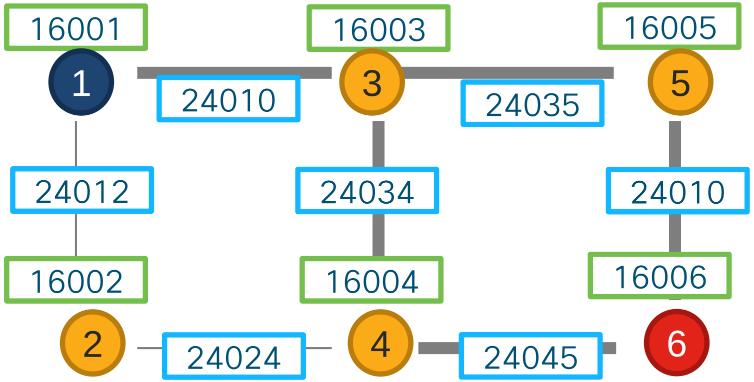
Go To "RED"; Desired Path 1→2→4→3→5→6



Segment Routing – Programming The Path

Go To "RED"; Desired Path 1→2→4→3→5→6

SID-List:
24012
24024
16003
16005
16006

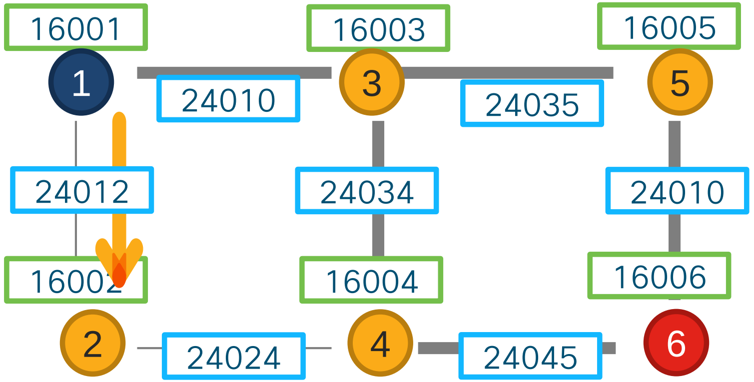


Segment Routing – Programming The Path

Go To "RED"; Desired Path 1→2→4→3→5→6

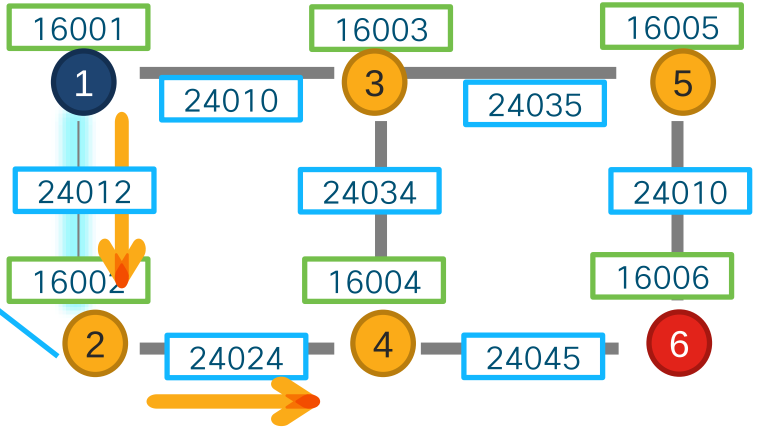
SID-List:
24012
24024
16003
16005
16006

24012; INT="D"

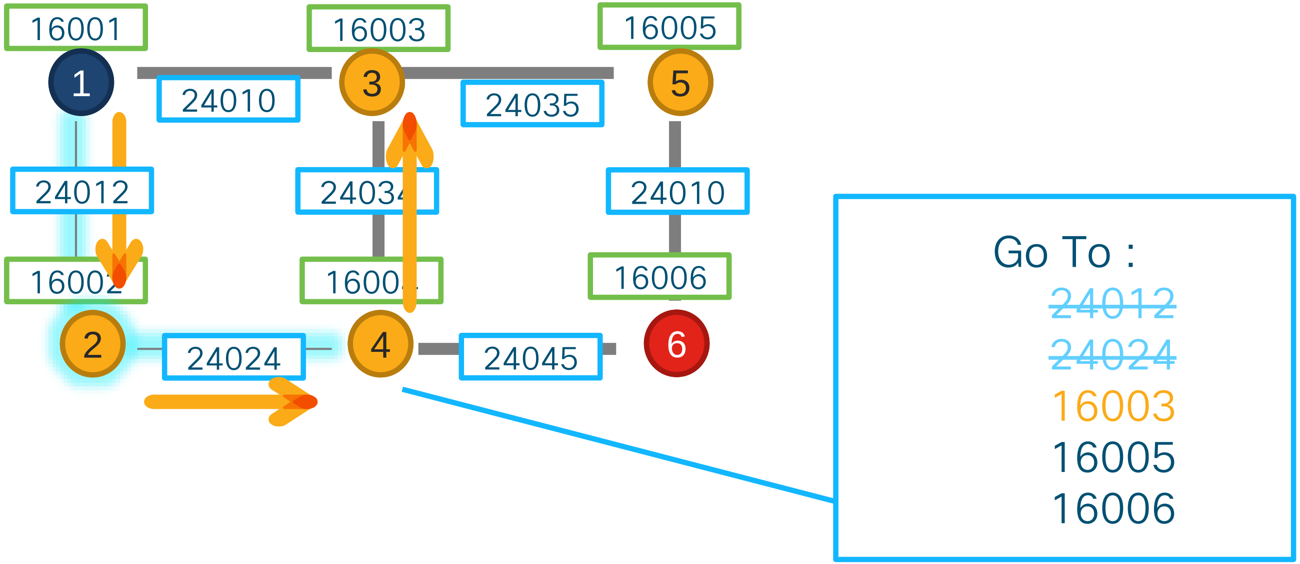


Segment Routing – Programming The Path

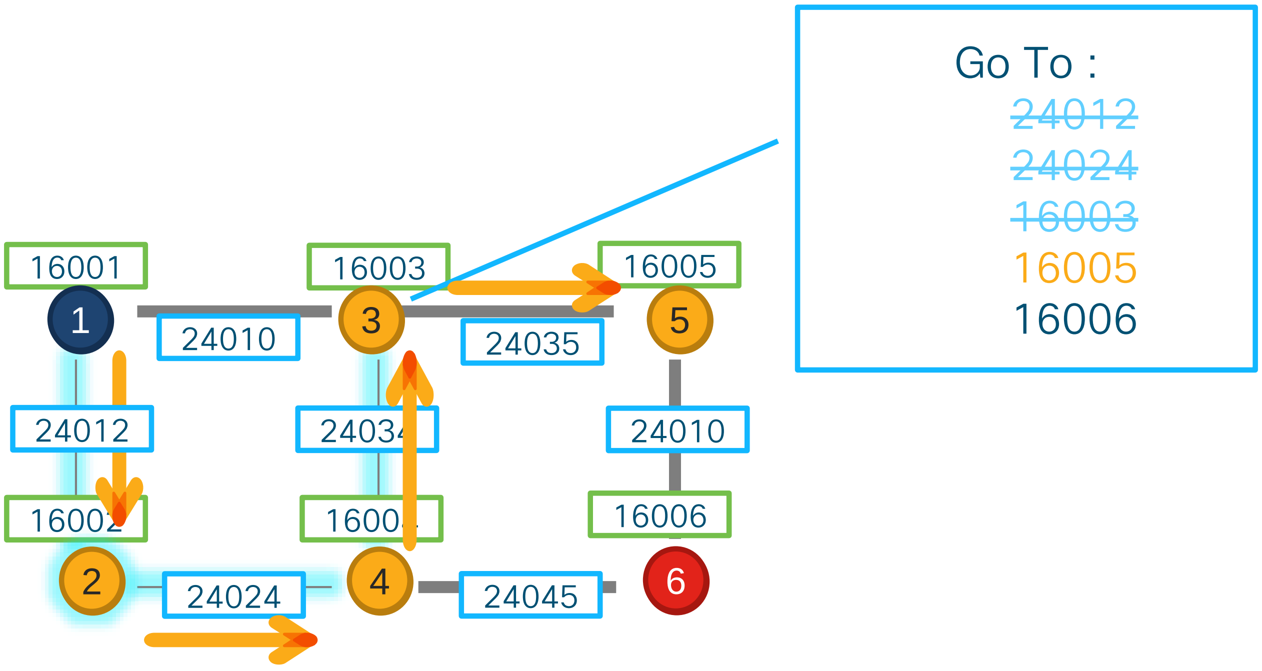
SID-List:
~~24012~~
24024
16003
16005
16006



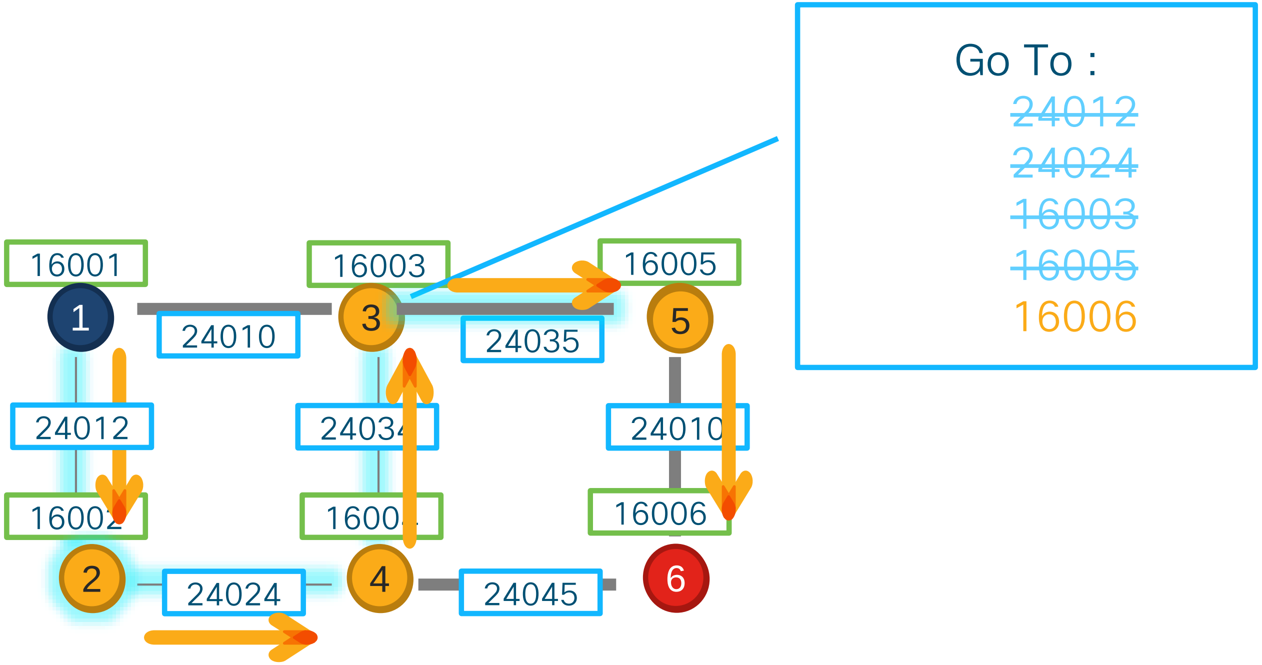
Segment Routing – Programming The Path



Segment Routing – Programming The Path

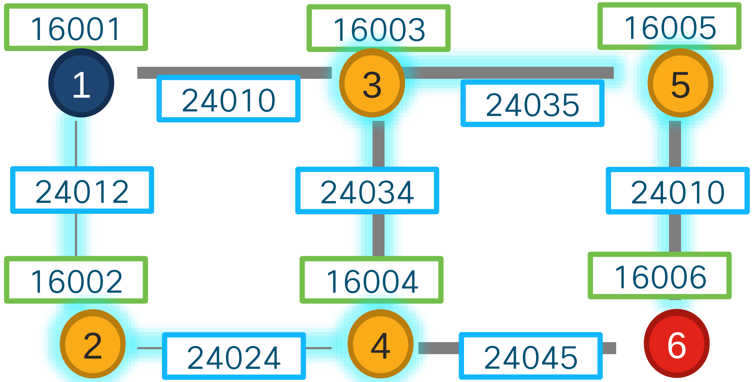


Segment Routing – Programming The Path



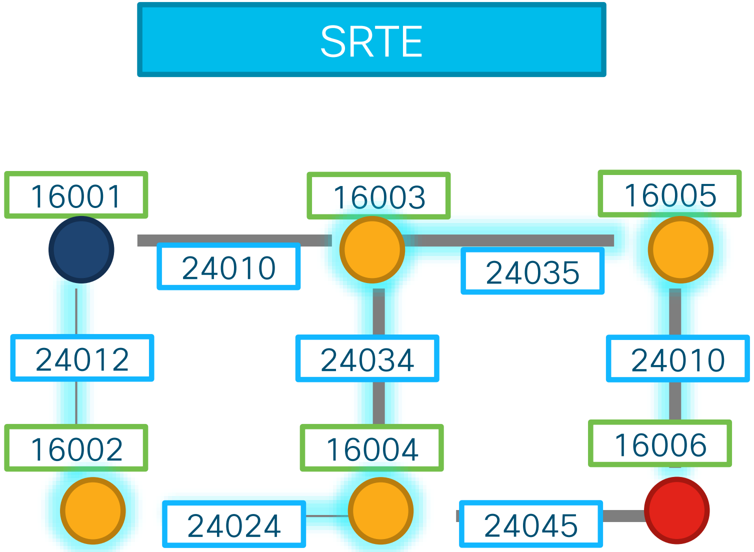
Segment Routing – Programming The Path

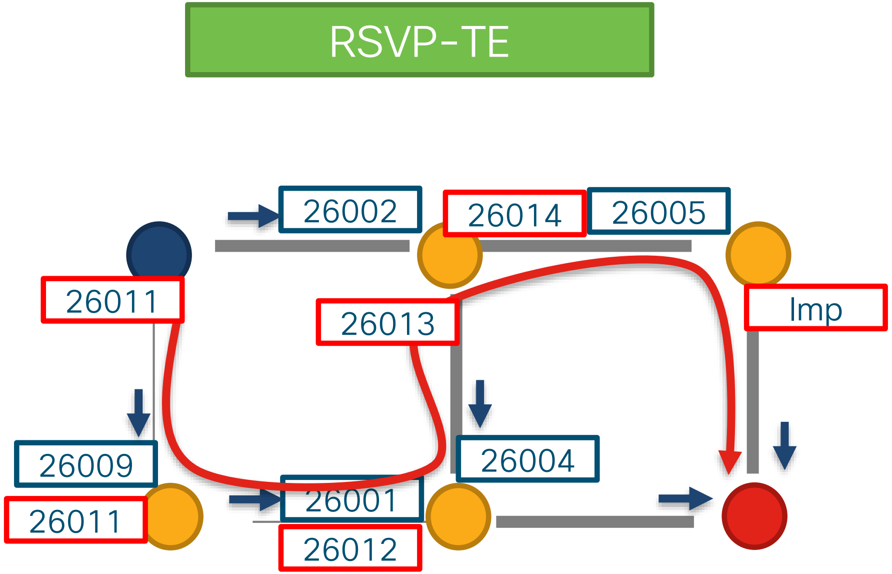
SRTE



Reached "RED"

Programming The Path – SRTE vs RSVP-TE





	SR-TE	RSVP-TE
TE state only at head-end	✔	✘
Engineered for SDN	✔	✔ / ✘
ECMP-capability for TE	✔	✘

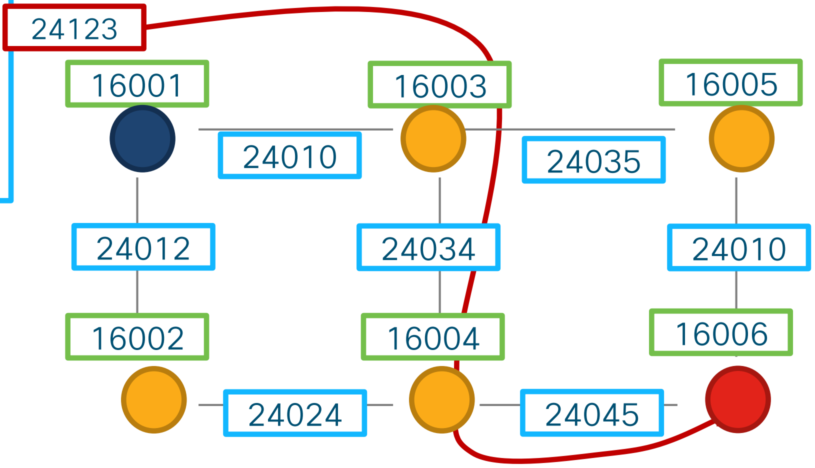
SRTE – Binding SID

Go To "RED"; Desired Path 1→2→4→3→5→6

SID-List:

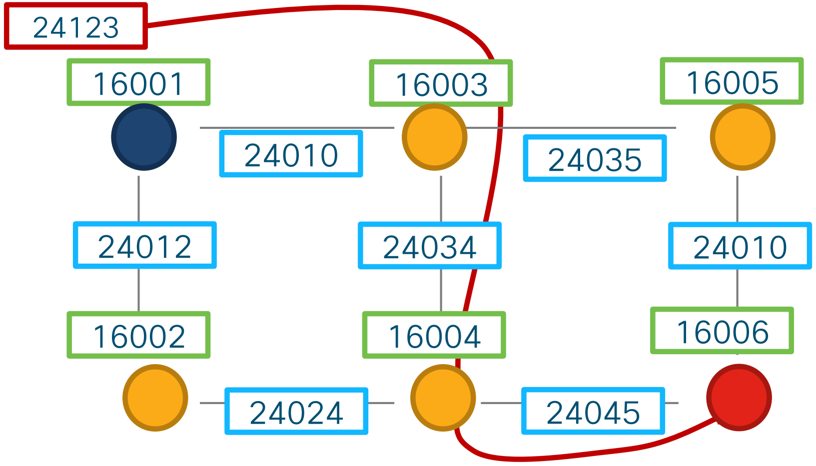
24123 → { 24012, 24024, 16003, 16005, 16006 }

- 24012
- 24024
- 16003
- 16005
- 16006



Segment Routing – Technology Overview

Node SID (Prefix SID)	Adjacency SID	Anycast SID	Binding SID
Globally Significant	Locally Significant Unidirectional	Globally Significant	Globally or Locally Significant



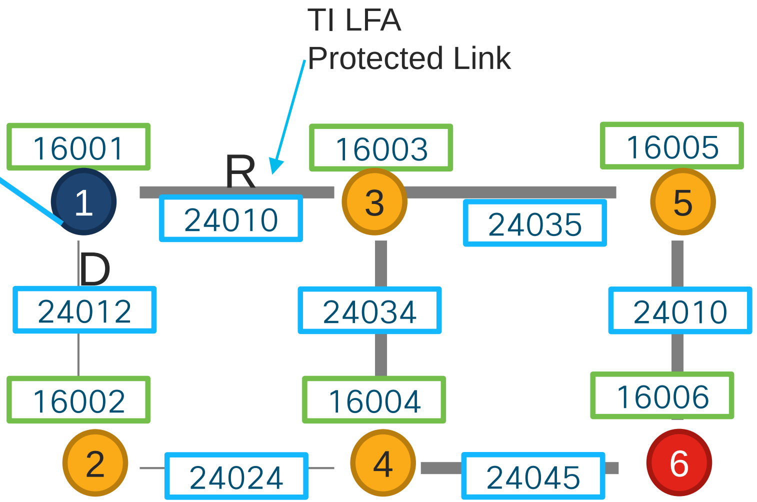
Fast Re-route with Segment Routing : TI-LFA

Topology Independent Loop Free Alternate Fast Re-route

Go To "RED"

RED=16006; INT="R"

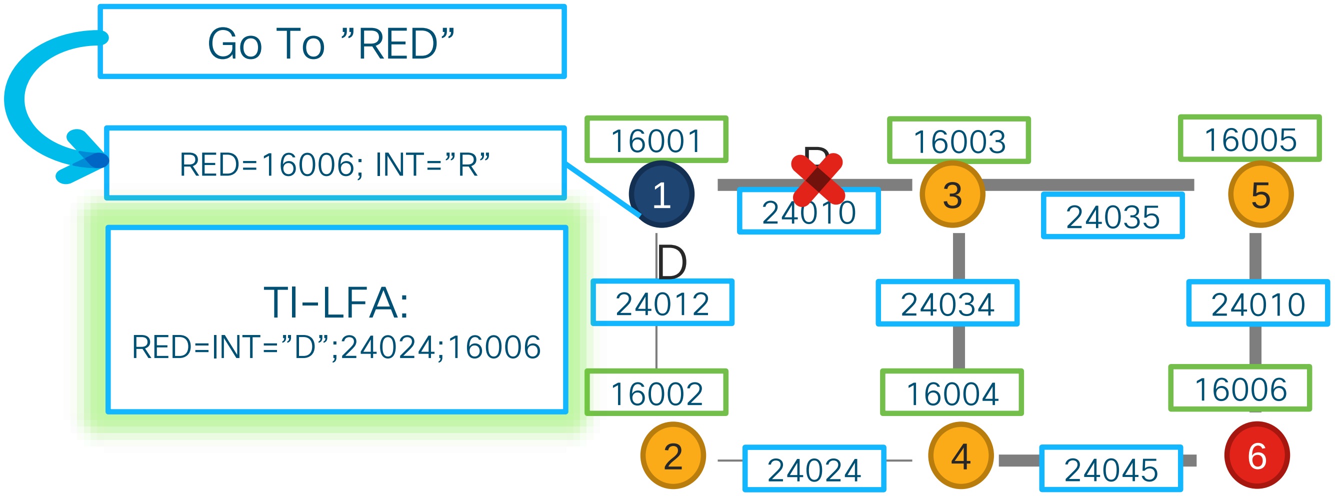
TI-LFA:
RED=INT-"D";24024;16006



https://www.ietf.org/id/draft-ietf-rtgwg-segment-routing-ti-lfa-01.txt

Fast Re-route with Segment Routing : TI-LFA

Topology Independent Loop Free Alternate Fast Re-route

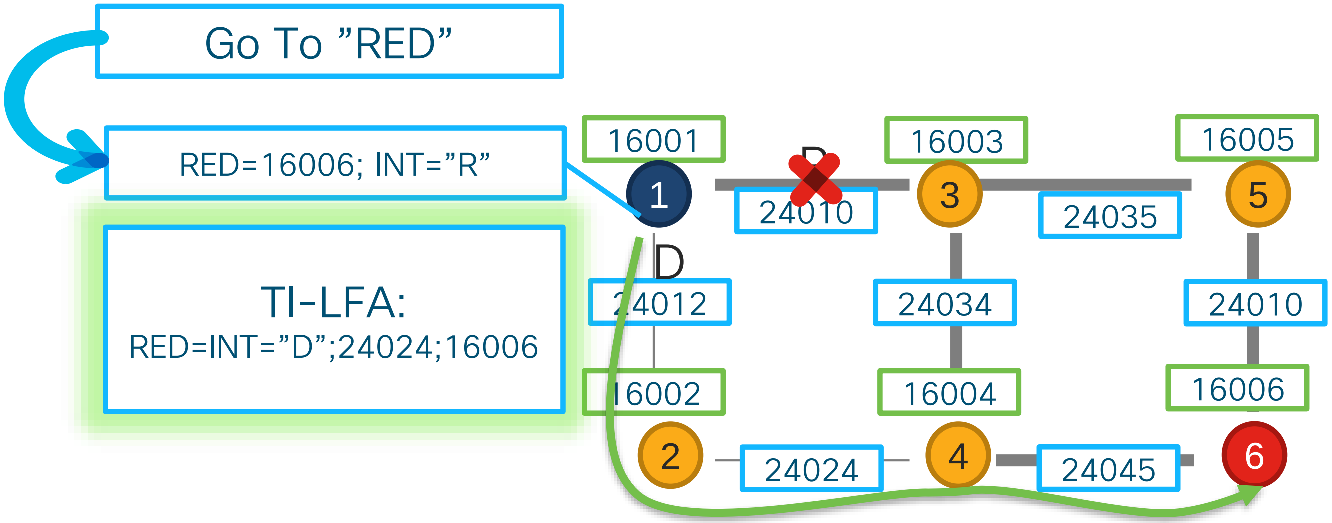


https://www.ietf.org/id/draft-ietf-rtgwg-segment-routing-ti-lfa-01.txt

cisco Live!

Fast Re-route with Segment Routing : TI-LFA

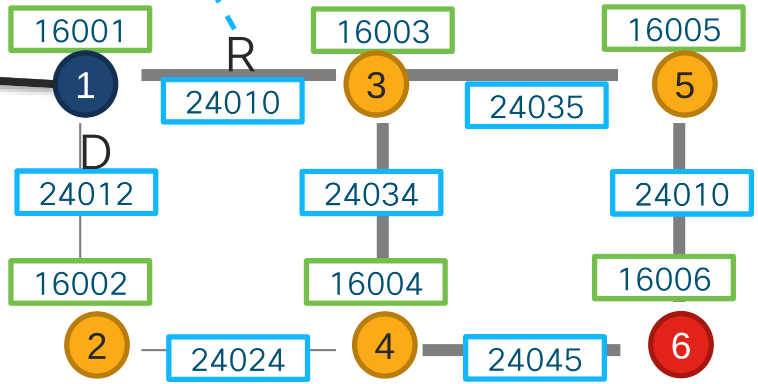
Topology Independent Loop Free Alternate Fast Re-route



https://www.ietf.org/id/draft-ietf-rtgwg-segment-routing-ti-lfa-01.txt

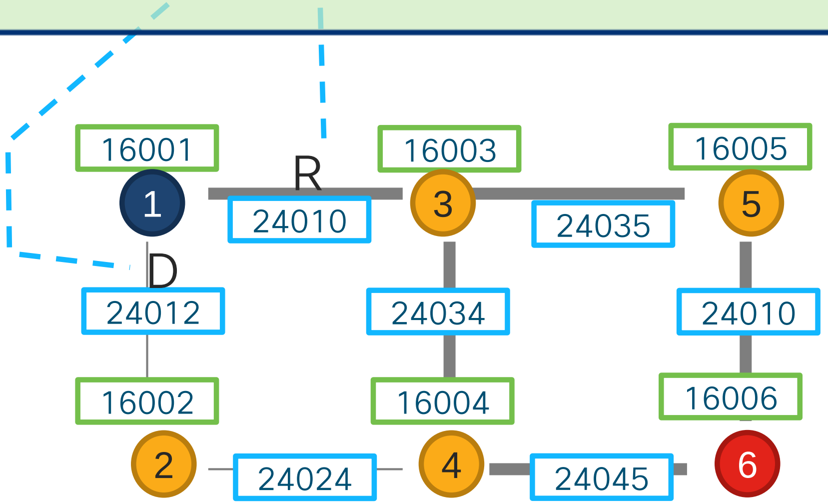
SRTE Protection using TI-LFA

```
router isis 100
 interface GigabitEthernet0/0/0/0
  address-family ipv4 unicast
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa
  !
 !
```



SRTE Protection using TI-LFA

```
RP/0/RP0/CPU0:R1#show mpls forwarding labels 16003
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16003  16003       SR Pfx (idx 5)     Gi0/0/0/0    192.1.3.3       640
       16003       SR Pfx (idx 5)     Gi0/0/0/1    192.1.2.2       0             (!)
```



SRTE Protection using TI-LFA

```
RP/0/RP0/CPU0:R1#show mpls forwarding labels 16003 detail
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16003  16003       SR Pfx (idx 5)     Gi0/0/0/0    192.1.3.3       2720
     Updated: Jun  6 21:12:49.488
     Path Flags: 0x400 [  BKUP-IDX:1 (0xee64350) ]
     Version: 88, Priority: 1
     Label Stack (Top -> Bottom): { 16003 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 1, Weight: 0
     MAC/Encaps: 4/8, MTU: 1500
     Outgoing Interface: GigabitEthernet0/0/0/0 (ifhandle 0x01000018)
     Packets Switched: 68


       16003       SR Pfx (idx 5)     Gi0/0/0/1    192.1.2.2       0            (!)
     Updated: Jun  6 21:12:49.488
     Path Flags: 0x300 [  IDX:1 BKUP, NoFwd ]
     Version: 88, Priority: 1
     Label Stack (Top -> Bottom): { Imp-Null 24024 16003 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0
     MAC/Encaps: 4/8, MTU: 1500
     Outgoing Interface: GigabitEthernet0/0/0/1 (ifhandle 0x01000020)
     Packets Switched: 0
     (!): FRR pure backup

  Traffic-Matrix Packets/Bytes Switched: 0/0
```

Notice that FRR backup path that is calculated and ready in case of failure

Introducing Intent Based Path Control With SRTE

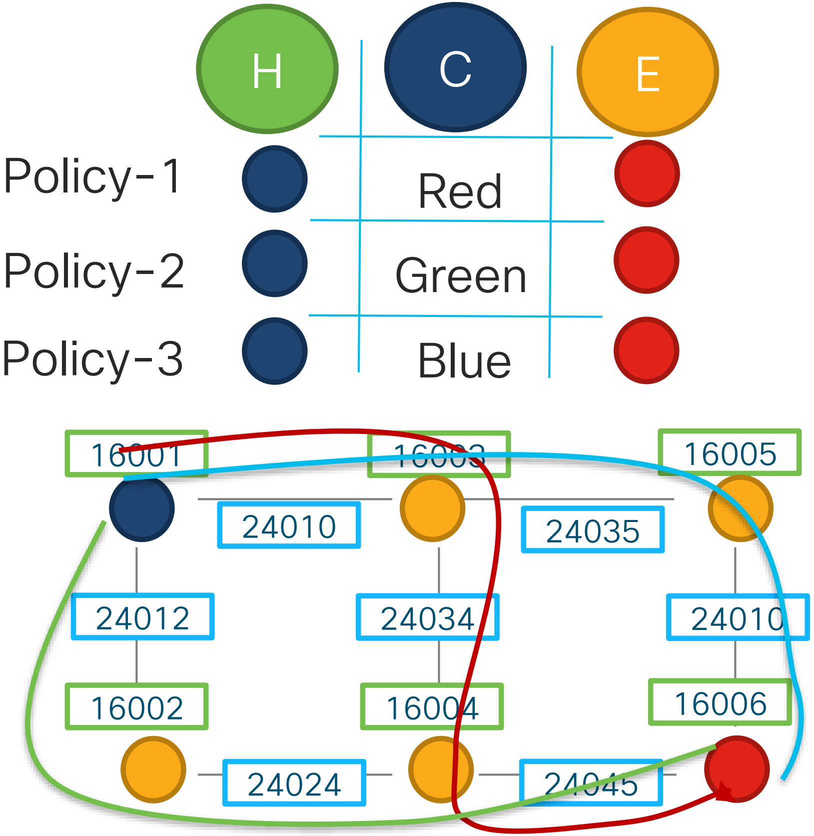
SRTE Policy

SRTE policy : defines a routing intent based on constraints

Policy is uniquely identified by a 3-tuple

Head End	Where the SR Policy is instantiated (*implemented*)	H
Color	Numeric value to differentiate multiple SRTE Policies between the same pair of nodes	C
Endpoint	Destination of the SR Policy	E

SRTE Policy Identification

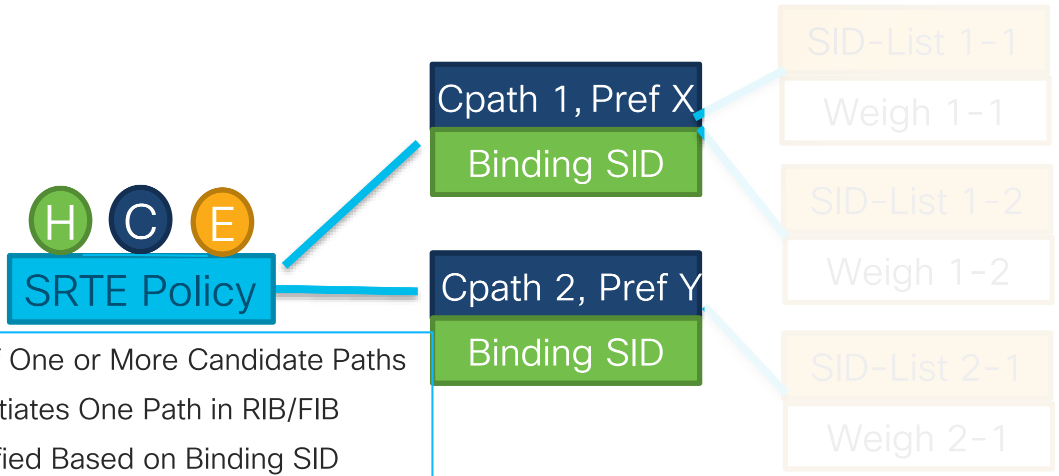


SRTE Policy – Candidate Paths

Explicit

Dynamic - Local

Dynamic - Remote



Consists of One or More Candidate Paths

Instantiates One Path in RIB/FIB

Identified Based on Binding SID

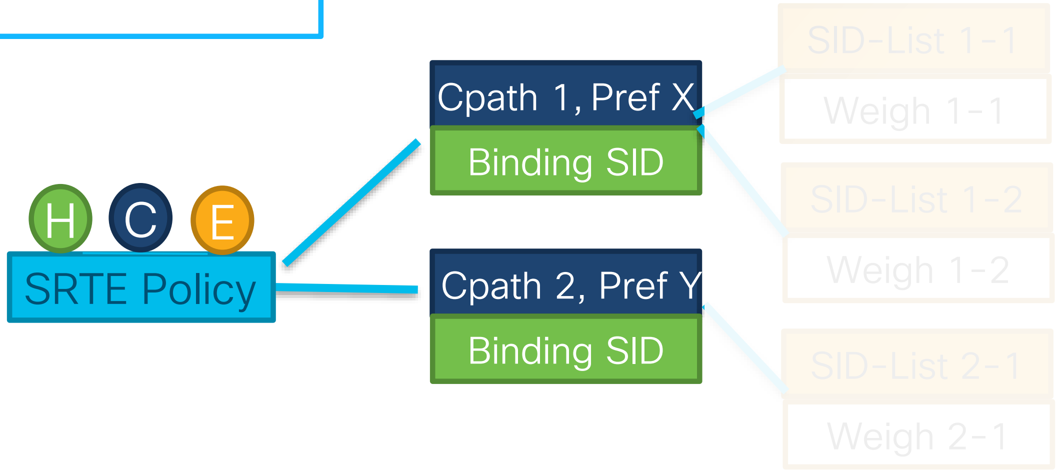
SRTE Policy – Candidate Paths

Explicit

Dynamic - Local

Dynamic - Remote

Explicitly Defined using:
- SID
- Prefix



SRTE Policy – Candidate Paths

Explicit

Explicitly Defined using:
- SID
- Prefix

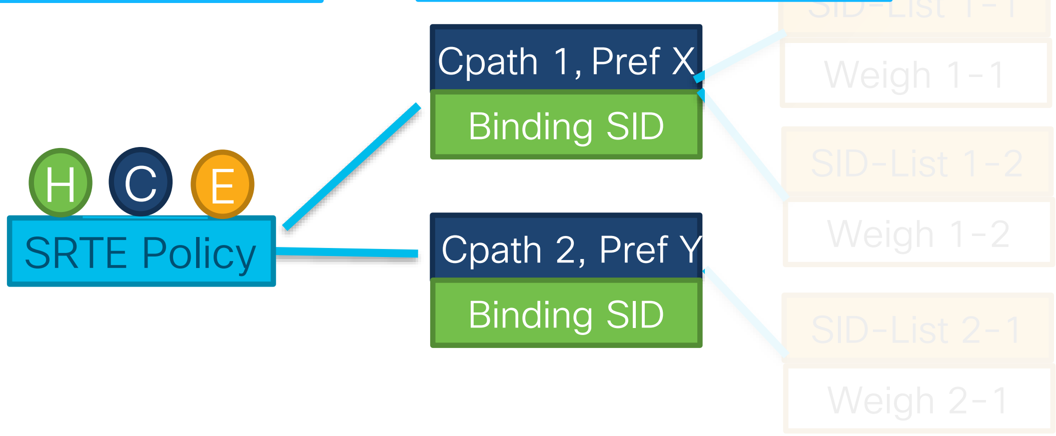
Dynamic - Local

Dynamically Computed by Headed

Based on Constraints

Recomputed if Network changes

Dynamic - Remote



SRTE Policy – Candidate Paths

Explicit

Explicitly Defined using:
- SID
- Prefix

Dynamic - Local

Dynamically Computed by Headed

Based on Constraints

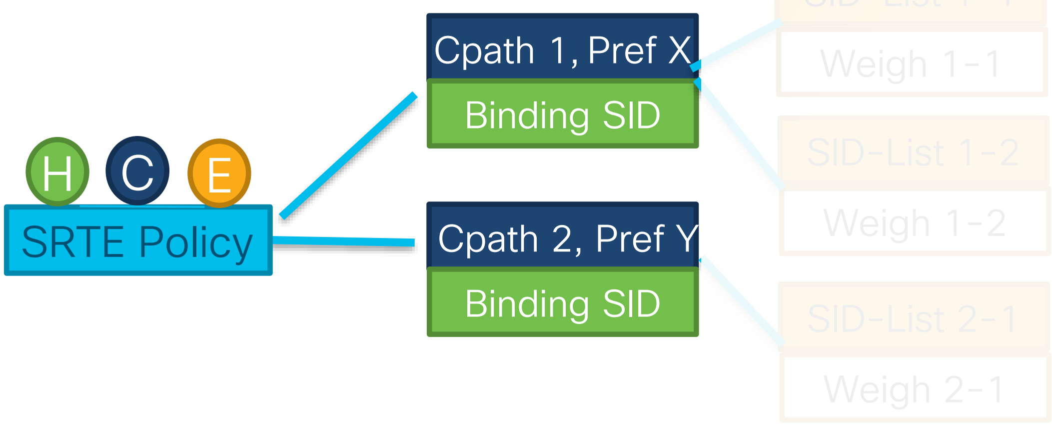
Recomputed if Network changes

Dynamic - Remote

Dynamically Computed by Controller

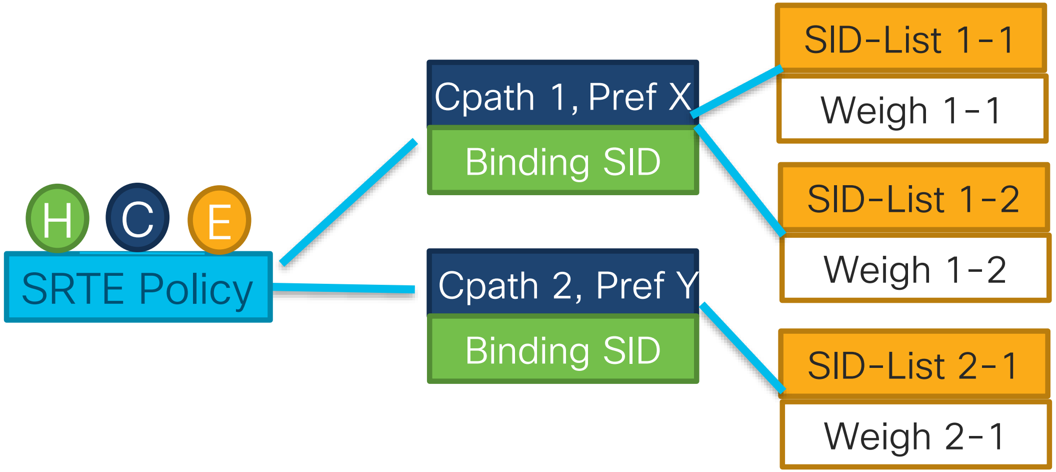
Based on Constraints

Recomputed if Network changes

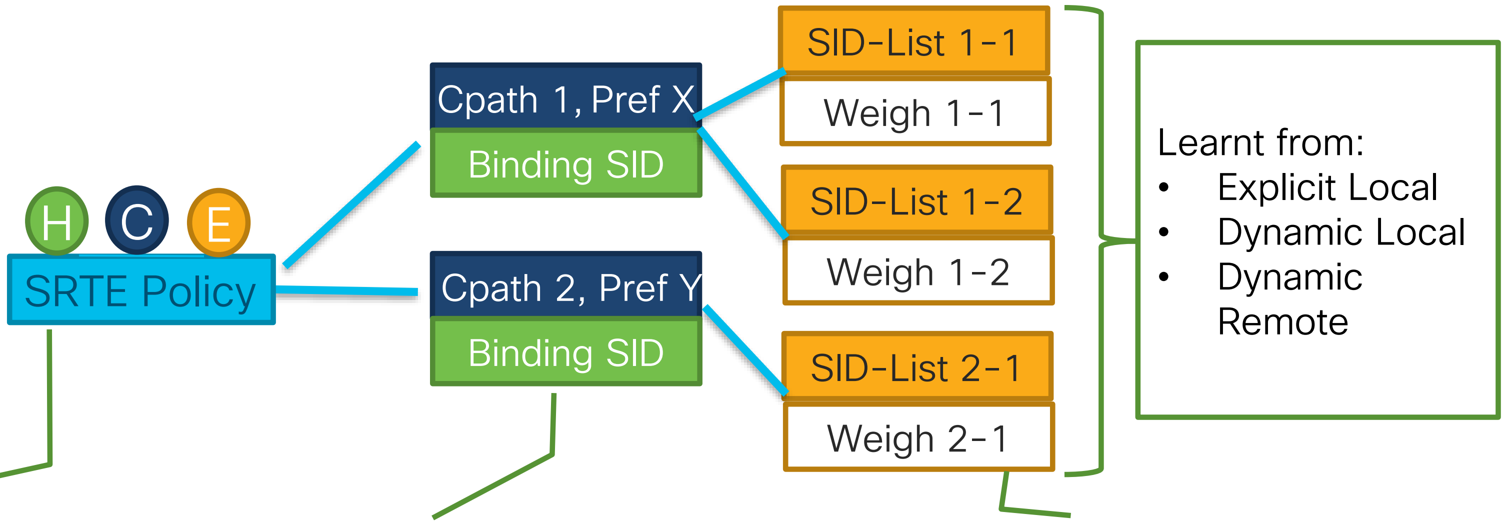


SRTE Policy – SID-Lists

- Each SRTE Candidate Path…
  - Single SID Lists, or Weighed SID-Lists
  - Traffic on Candidate Path is Load Balanced
  - Weight defines Load-Balance Ratio



SRTE Policy - Summary

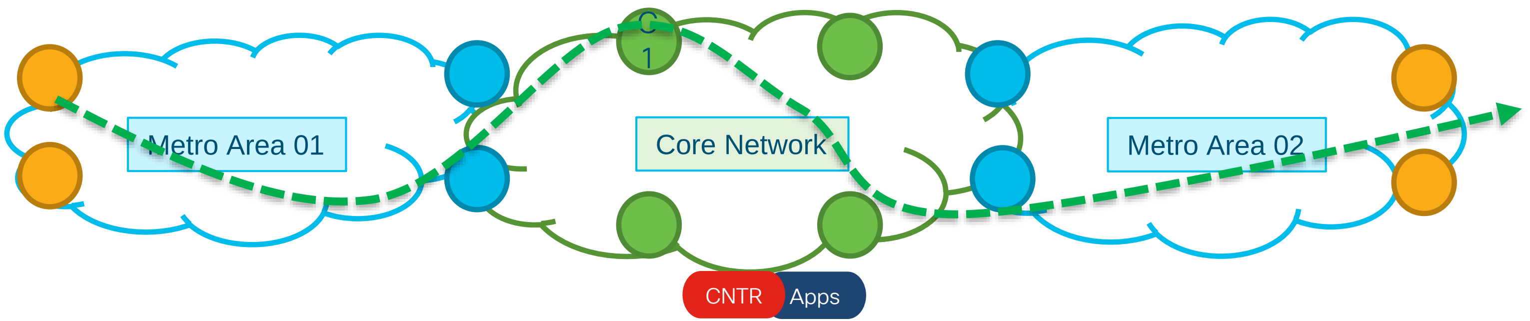


- Identified by Unique: (Head-End + Color + End-Point)

- Chosen based on Preference
- Valid if: Any of the SID-list is valid
- Identified by: Binding SID (Auto)

- Programmed in FIB simultaneously
- Load-balanced based on
- Weight

SRTE – Path Compute & Configuration

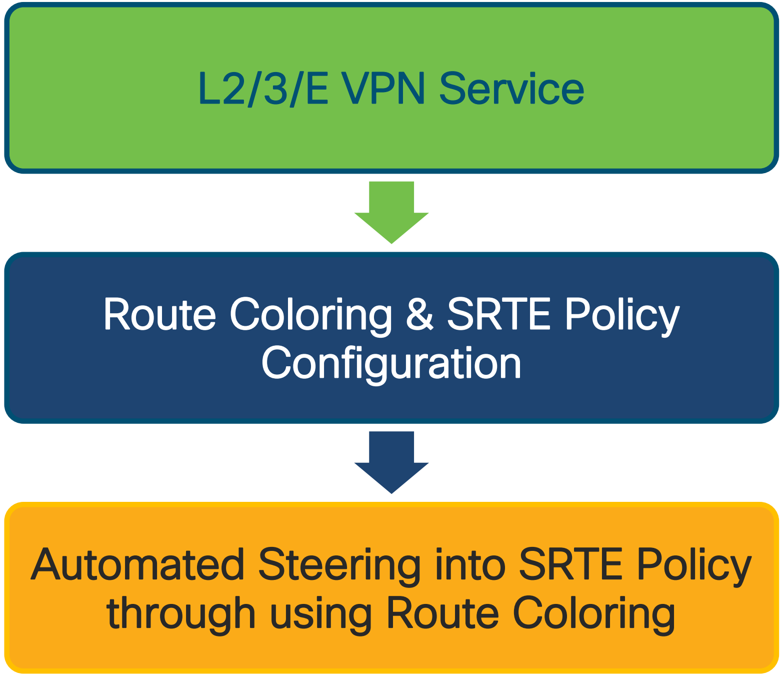


Computed : Locally Configured : Locally	Computed : Controller Configured : Locally	Computed : Controller Configured : Controller

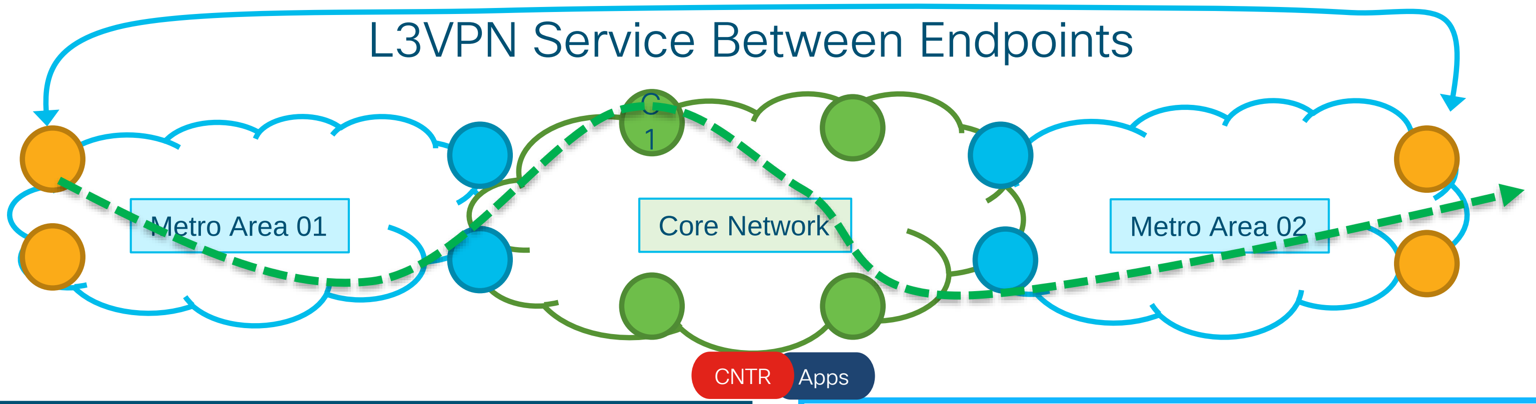
• Explicit Path
• Dynamic Path
• Based on constraints
IGP, BW, Delay, SRLG, Affinity …
	• Requested by Headend
	  • Configured
	  • On-Demand
	• Computed by Controller using Network View
		• Application Triggered
		• Configured on Controller
		• Pushed down to Headend

Automated Steering

- Configure a service for Network Slice (e.g. L3VPN)
- "Color" the service routes with BGP Ext Community
- Configure SRTE policy for the Network Service
- BGP will automatically steer traffic into an SR Policy based on BGP next-hop and color of a route



SRTE – Route Coloring Configuration Example



Setting Color using Ext Community

```
extcommunity-set opaque BLUE
  20
end-set

route-policy SET_COLOR_BLUE
 if destination in (10.100.1.1/32) then
    set extcommunity color BLUE
  endif
end-policy
```

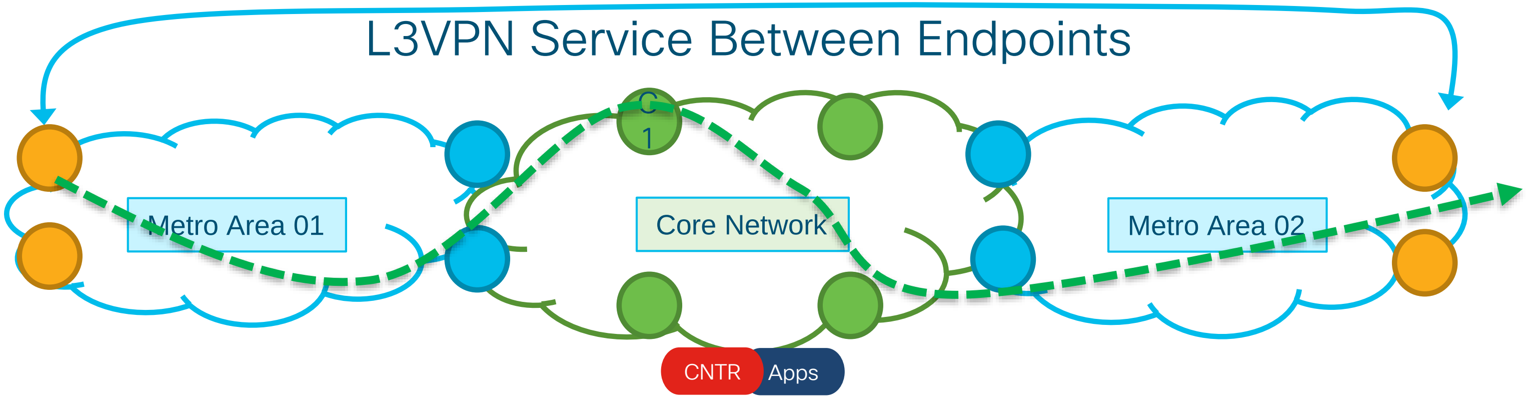
Coloring Route for Automated Steering

```
router bgp 65000
 neighbor 6.1.1.x
  address-family vpnv4 unicast
   route-policy SET_COLOR_BLUE in
     !could be outbound policy as well
```

OR

```
vrf C-Blue
  address-family ipv4 unicast
   export route-policy SET_COLOR_BLUE
```

SRTE – Route Coloring Verification



Verify Route Coloring

```
RP/0/0/CPU0:PE25#sh bgp vpnv4 unicast vrf L3VPN-1  10.100.1.0
BGP routing table entry for 10.100.1.0/24, Route Distinguisher: 65000:1
<snip>
Extended community: Color:20 RT:65000:1
      Originator: 192.168.0.15, Cluster list: 192.168.0.5
      Source AFI: VPNv4 Unicast, Source VRF: L3VPN-1, Source Route Distinguisher: 65000:1
<snip>
```

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Enable SRTE

Local Configured SRTE Policy

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

- Enable SRTE
- Local Configured SRTE Policy
- Color (C) & End-Point (E)
  Color Used for Automated Steering
- Binding SID for Selected C-Path
- Candidate Path List
- Candidate Path Preference

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Enable SRTE

Local Configured SRTE Policy

Color (C) & End-Point (E)
Color Used for Automated Steering

Binding SID for Selected C-Path

Candidate Path List

Candidate Path Preference

Dynamic – Local Path

Optimize: TE-Metric
Constraint: Affinity

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Enable SRTE

Local Configured SRTE Policy

Color (C) & End-Point (E)

Second Candidate Path; Lower Preference

Using Explicit SID-List

SID-List

Dynamic – Local Path

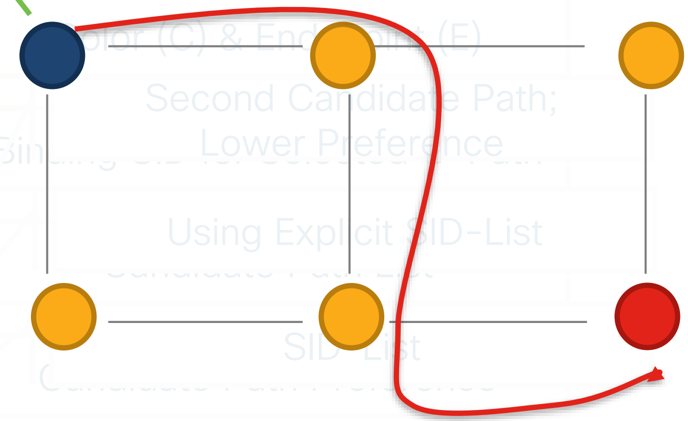
Optimize: TE-Metric
Constraint: Affinity

SRTE - Configuration Example [Local Policy]

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point ipv4 1.1.1.4
   binding-sid mpls 1000
   candidate-paths
    preference 100
     dynamic mpls
      metric
       type te
      affinity
       exclude-any red
    !
    preference 50
     explicit segment-list SIDLIST1
  !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

Selected

FIB Programmed:
POLICY1 → Dynamic Computed
Incoming Label 1000 → POLICY1
Color=20, End-Point=1.1.1.4 → POLICY1

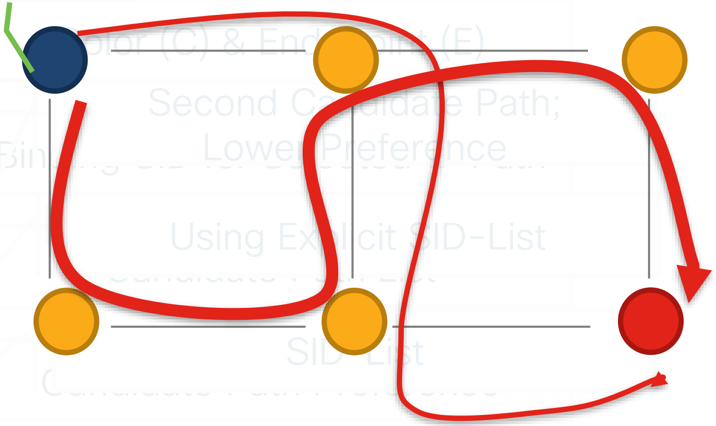


Segment Routing Configuration Example Weights

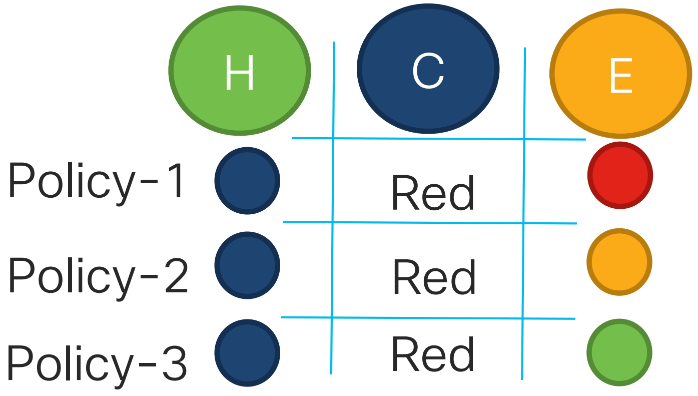
```
segment-routing
 traffic-eng
  policy POLICY2
   color 30 end-point ipv4 1.1.1.4
   binding-sid mpls 2000
   candidate-paths
    preference 200
     explicit segment-list SIDLI
      weight 1
     !
     explicit segment-list SIDLI
      weight 4
     !
  segment-list name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 16003
   index 30 mpls label 16004
  !
  segment-list name SIDLIST2
   index 10 address ipv4 1.1.1.4
```

Selected

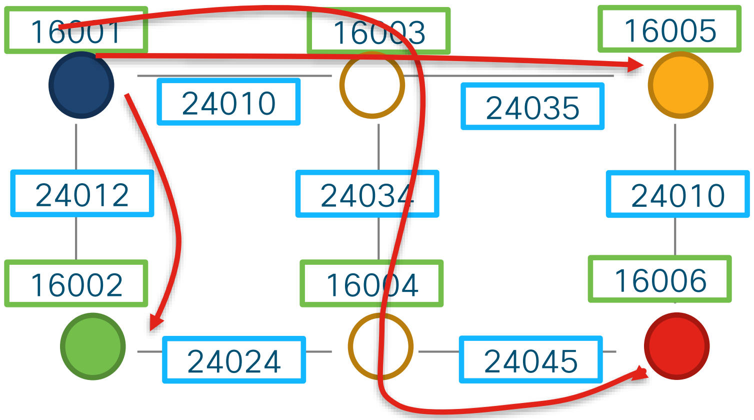
FIB Programmed:
Incoming Label 2000 → POLICY2
Color=30, End-Point=1.1.1.4 → POLICY2
POLICY2 → Explicit



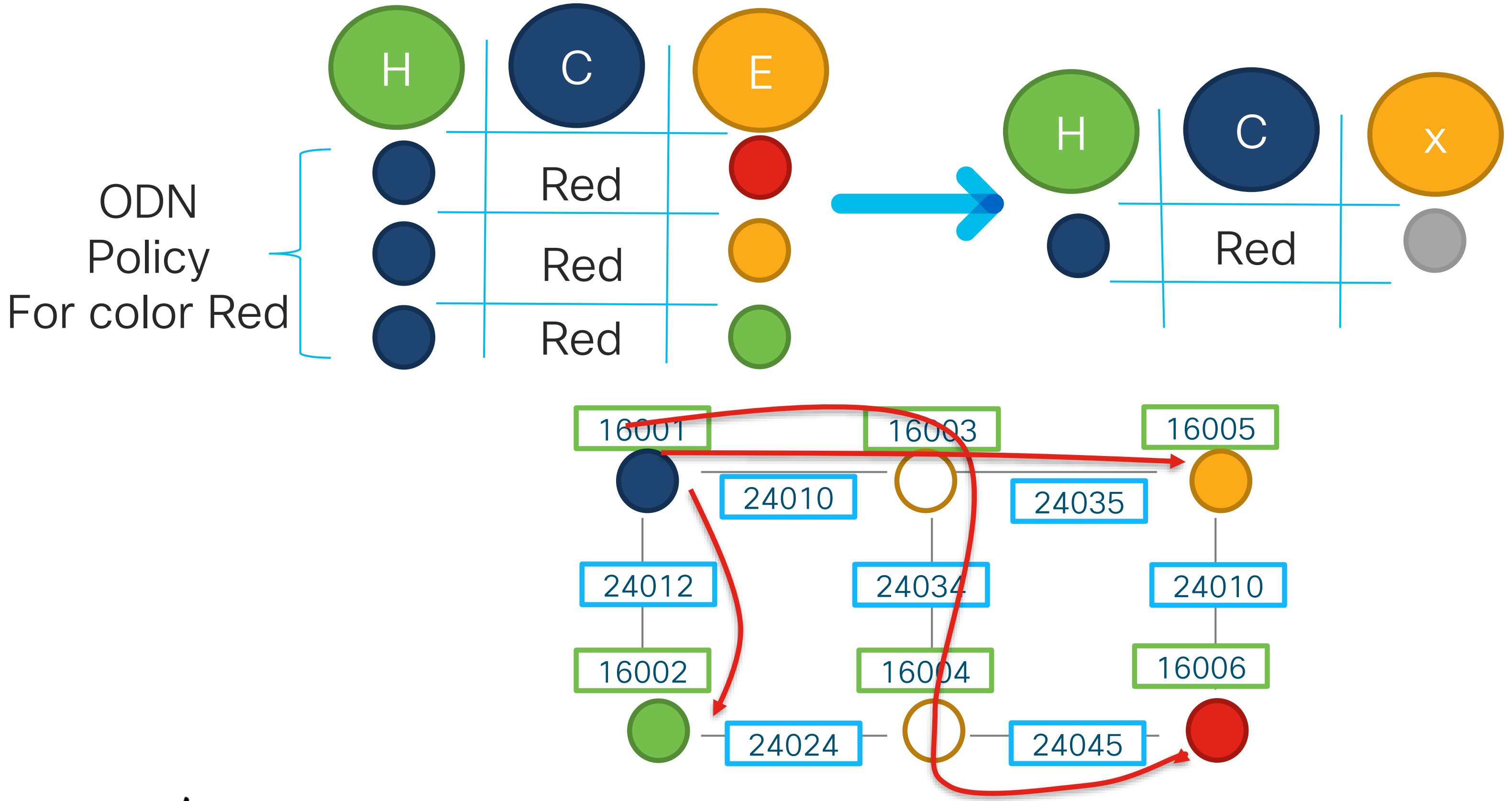
SRTE - On Demand NextHop (ODN) Policy



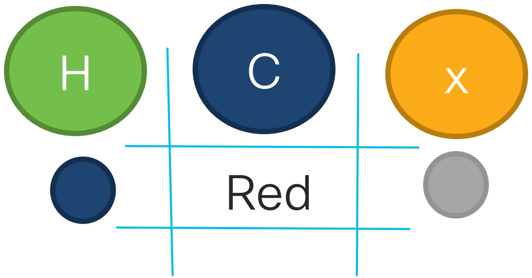
Policy-1, Policy-2 & Policy-3
Have same constraints but
different end points
(e.g. L3VPN with multiple endpoints)



SRTE - On Demand NextHop (ODN) Policy



SRTE - Configuration Example [ODN]




```
segment-routing
   traffic-eng
    on-demand color 20
     dynamic
       metric
         type latency
```

- Enable SRTE
- Define Color for ODN
- Dynamic policy using latency as metric

Flex Algo & Segment Routing TE

Segment Routing (SR):
Use Default IGP Metric to forward traffic (**Default Algo**)
Ability to define a SID-List at the source for traffic forwarding

Segment Routing Traffic Engineering (SRTE):
Intent based forwarding that goes beyond Best Path forwarding
Uses SID List to influence forwarding path

SRTE with Flex-Algo
Intent based forwarding that **uses specific best paths** based on flexible definitions of best path
Uses SID matching the algorithm

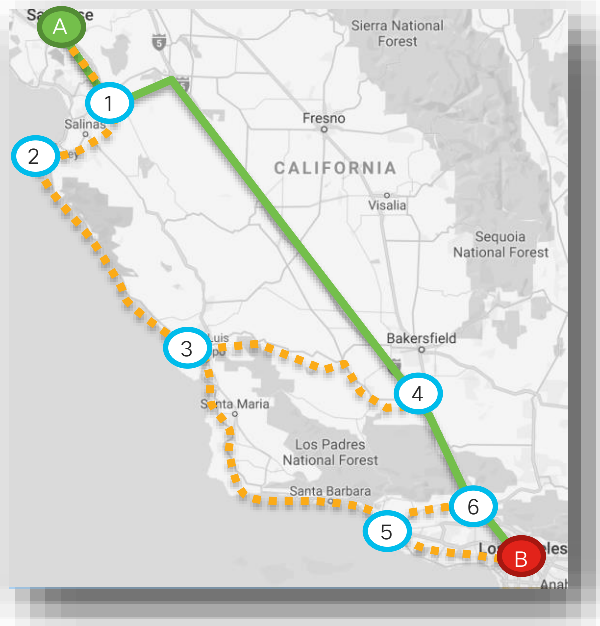
Flex-Algo Leverages all SRTE benefits and simplicity – TI-LFA, ODN, Auto Steering, Coloring, etc.

Flex Algo - Introduction

- Many Possible Routes from SJC to SFO
- Interior Routing Protocols will use Metrics
  - Metric:
    - OSPF Based on Bandwidth
    - ISIS Based on Hop

Flex Algo - Introduction

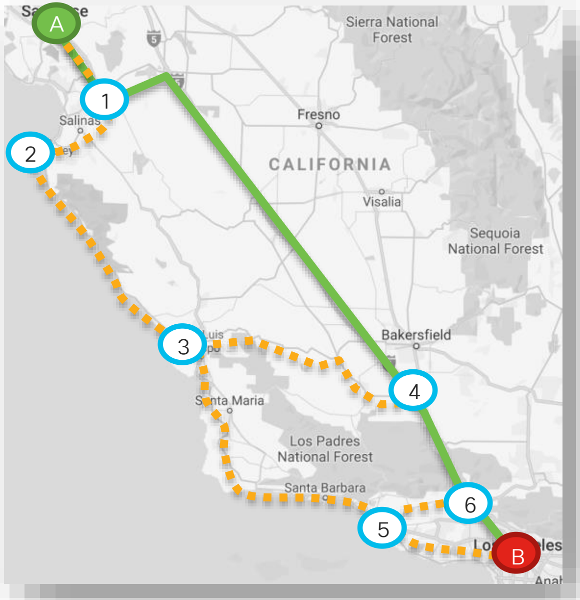
Intent == Shortest Path



- Many Possible Routes from SJC to SFO
- Interior Routing Protocols will use Metrics
  - Metric:
    - OSPF Based on Bandwidth
    - ISIS Based on Hop

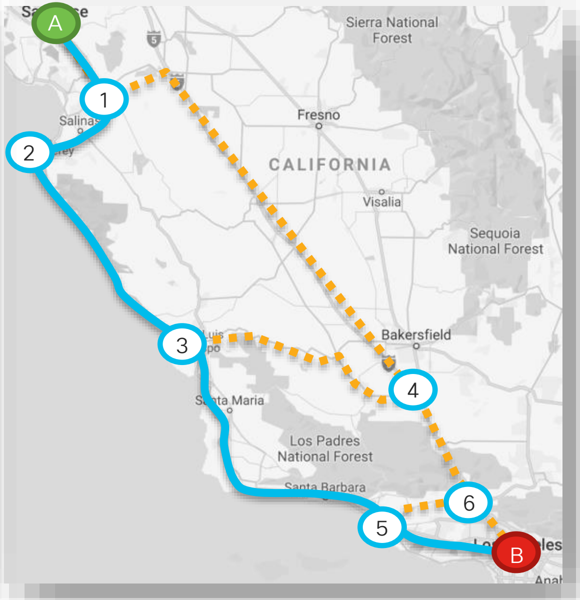
Flex Algo –Different Intent, Different Paths

Intent == Shortest Path



A→1→4→5→B

Intent == Scenic

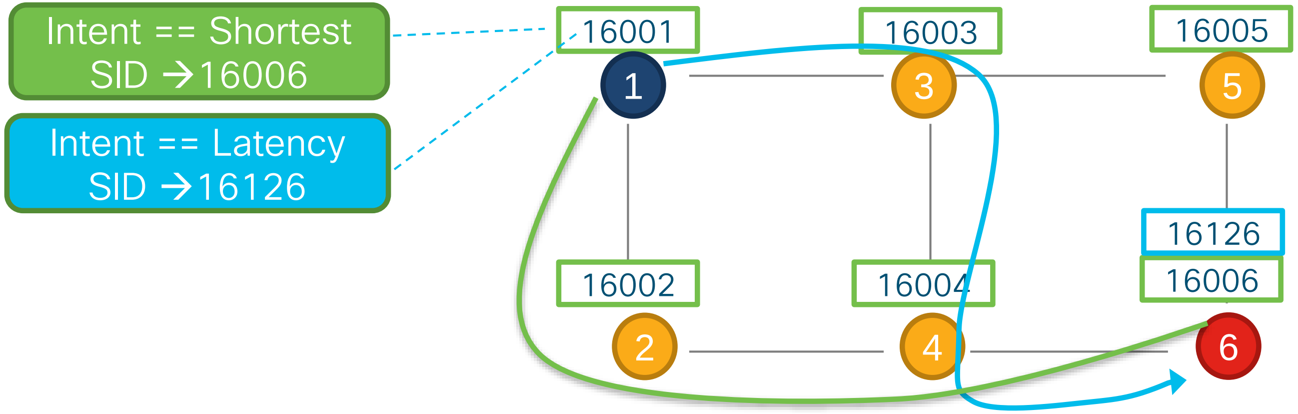


A→1→2→3→5→B

Different Intent may end up calculating different best paths, based upon constraints

Flex Algo –Segment Routing TE

- A new Algo is defined in IGP , with new constraints (Latency etc)
- A node may or may not participate in non default Algo(s)
- Each participating node must advertise Flex-Algo(s) that it is participating in
- Nodes participating in a Flex-Algo, also advertise a prefix SID for that Flex-Algo



Node 6 advertises

Prefix SID 16006 for ALGO 0

Prefix SID 16126 for ALGO 128

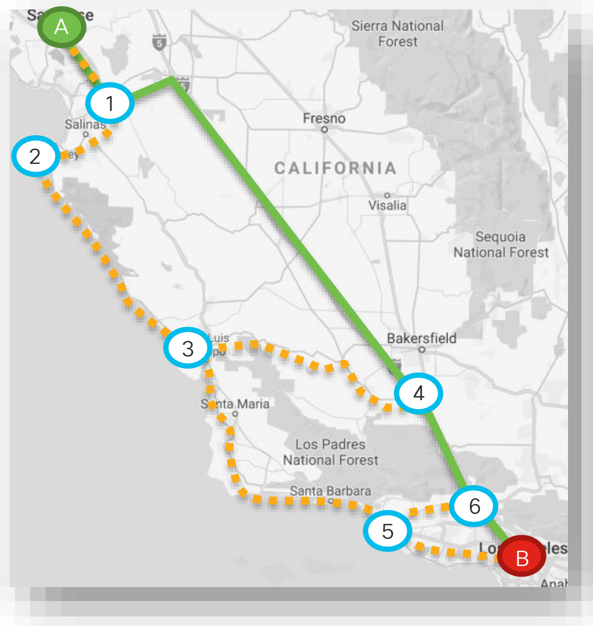
Every node participating in

Algo 0, 128

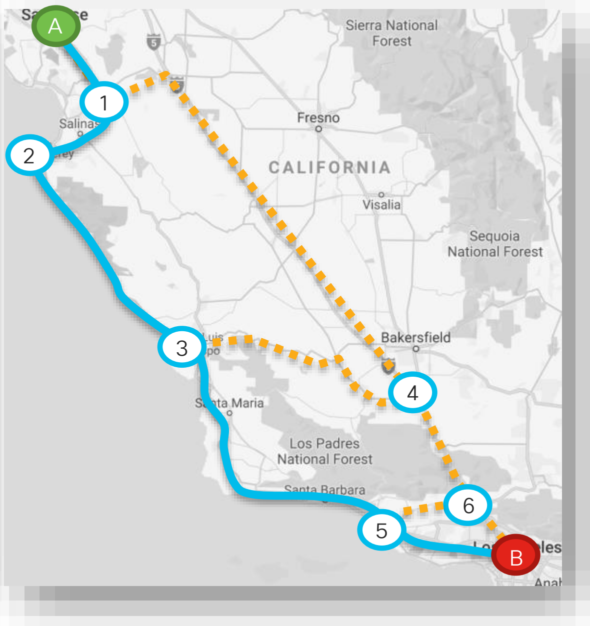
https://tools.ietf.org/html/draft-ietf-lsr-flex-algo-02

Flex Algo

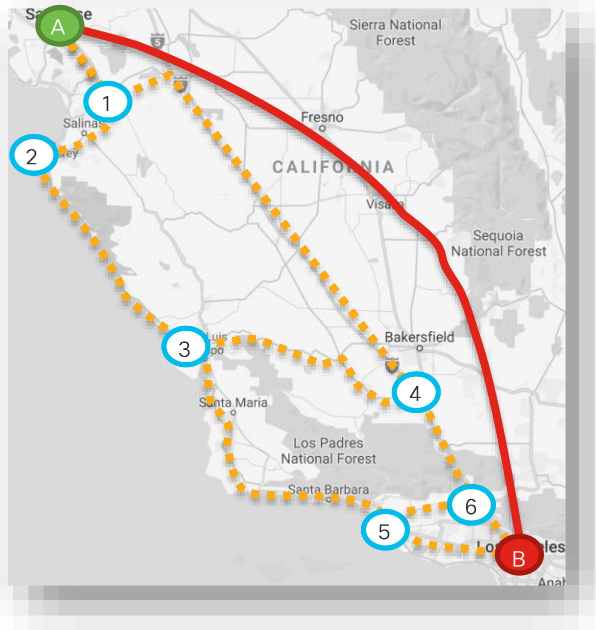
Intent == Shortest



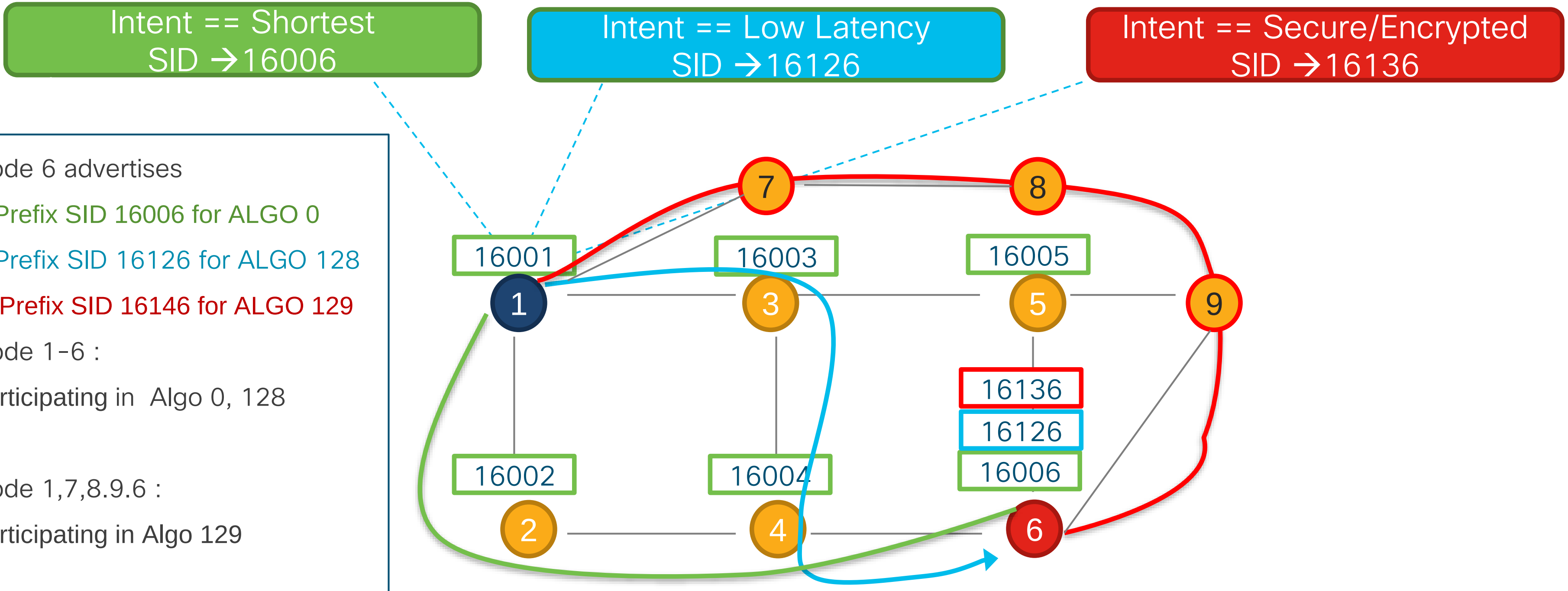
Intent == Scenic



Intent == Avoid Driving



Flex Algo –Segment Routing TE



Node 6 advertises

Prefix SID 16006 for ALGO 0

Prefix SID 16126 for ALGO 128

Prefix SID 16146 for ALGO 129

Node 1-6 :

participating in Algo 0, 128

Node 1,7,8.9.6 :

participating in Algo 129

Flex-Algo Configuration Example

```
router isis 1
net 49.0001.0000.0000.0002.00
 flex-algo 128
   metric-type latency
 !
 address-family ipv4 unicast
   router-id 6.1.1.9
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 2
   prefix-sid algorithm 128 absolute 16802

segment-routing
  traffic-eng
    on-demand color 100
       dynamic
          sid-algo 128
```

Flex Algo Definition. Multiple Flex-Algo's (128-255) could be defined

Defining Intent for Flex Algo. Default is IGP. Constraints could be defined here as well.

Prefix SID for Default Algo (IGP)

Additional Per-Algo Prefix SID is defined. Same rules as default Prefix SID (uses SRGB, can be defined as index or absolute, etc)

Flex-Algo Configuration Example

```
router isis 1
net 49.0001.0000.0000.0002.00
 flex-algo 128
   metric-type latency
 !
 address-family ipv4 unicast
   router-id 6.1.1.9
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 2
   prefix-sid algorithm 128 absolute 16802

segment-routing
  traffic-eng
    on-demand color 100
        dynamic
           sid-algo 128
```

Flex Algo Definition. Multiple Flex-Algo's (128-255) could be defined

Defining Intent for Flex Algo. Default is IGP. Constraints could be defined here as well.

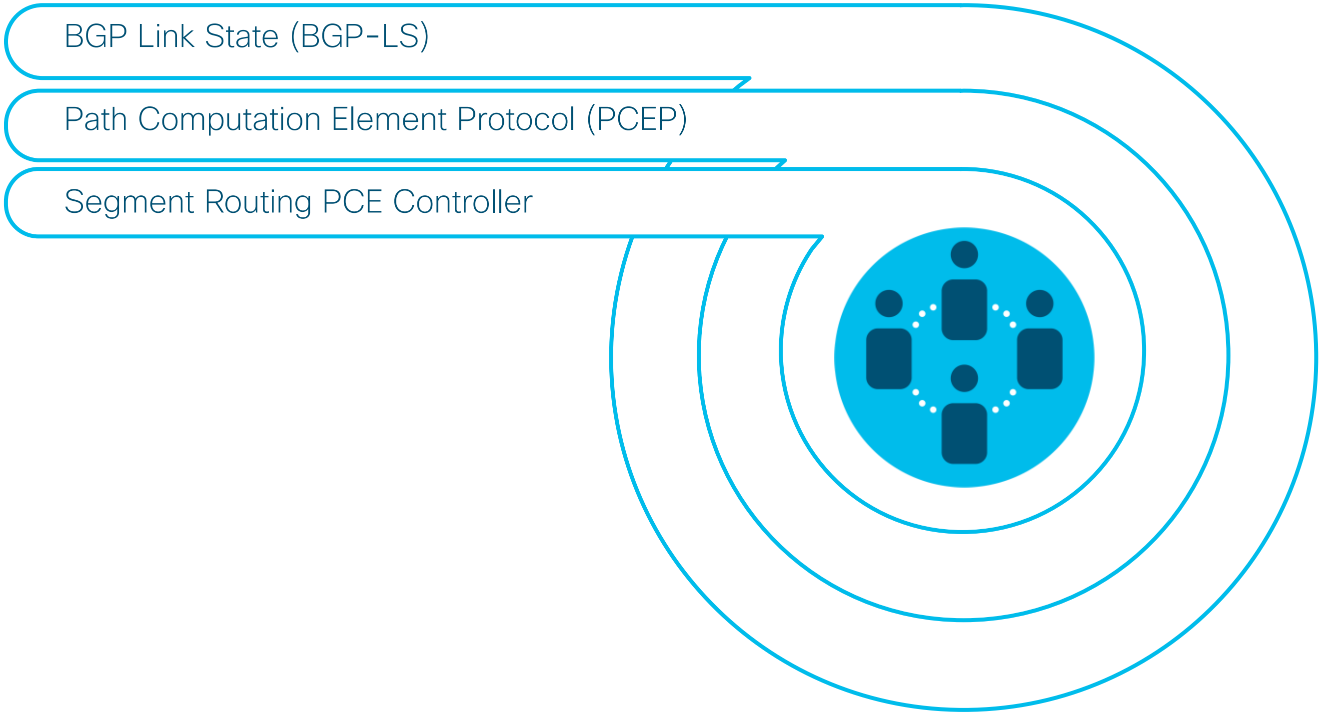
Prefix SID for Default Algo (IGP)

Additional Per-Algo Prefix SID is defined. Same rules as default Prefix SID (uses SRGB, can be defined as index or absolute, etc)

Uses Automated Steering using colors for Traffic Forwarding (Same as SRTE)

Application Based Path Control using SR-PCE

Centralized Control for SRTE – Building Blocks



Centralized Control for SRTE – Building Blocks

BGP Link State (BGP-LS)

Path Computation Element Protocol (PCEP)

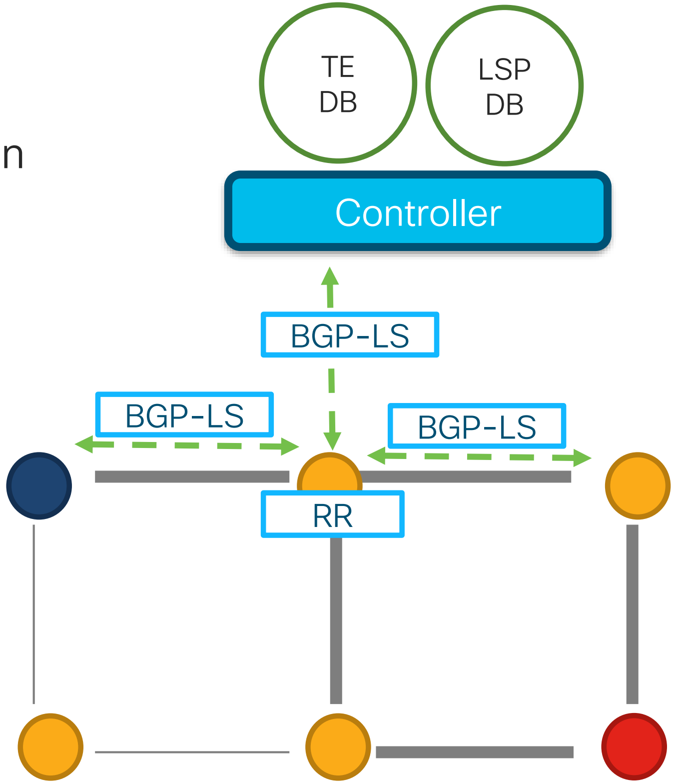
Segment Routing PCE Controller

Centralized Knowledge of IGP Database

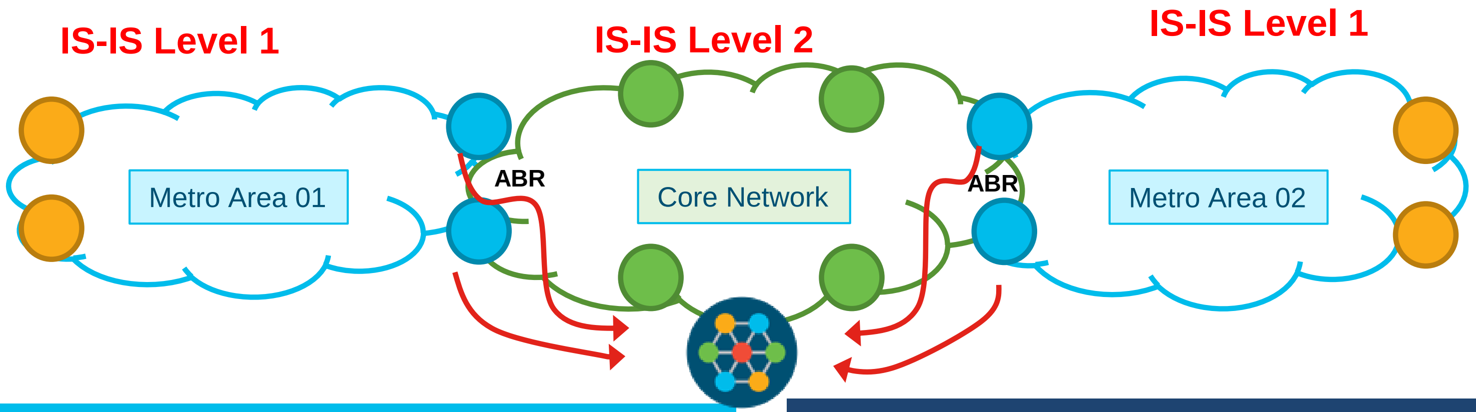
- IGP Database knowledge contained in IGP domains
- IGP DB is Distributed into new BGP NLRI
- BGP Carries the information to Central Controller

BGP Link State - Overview

- Build TE-DB for Multi-area Optimal Path Computation
- Scalable Solution is BGP, not IGP.
- BGP is less chatty
- Can carry multiple IGP domains
- BGP-LS is an address-family
  - afi=16388, safi=71
- Defined to carry IGP link-state database via BGP
  - Supports both IS-IS and OSPF
  - Delivers topology information to outside agents



BGP Link State Configuration Sample



Redistribute IGP Link State

```
router isis 100
 net 49.1921.5500.0004.00
 distribute link-state
```

Advertise via BGP-LS

```
router bgp 65000
 address-family link-state link-state
 neighbor 192.168.0.15
  remote-as 65000
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family link-state link-state
   route-reflector-client
```

Centralized Control for SRTE – Building Blocks

BGP Link State (BGP-LS)

Path Computation Element Protocol (PCEP)

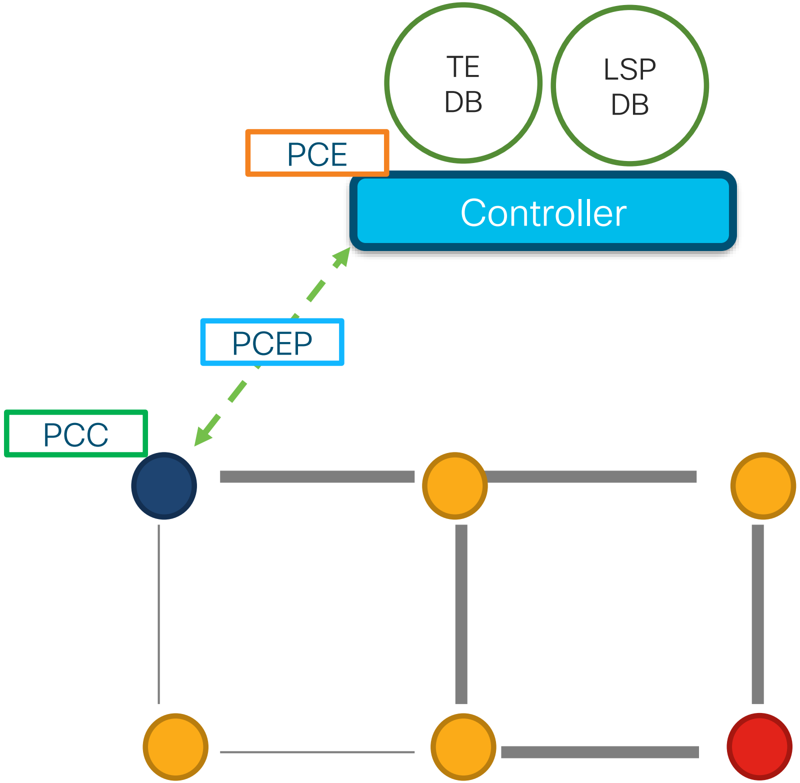
Segment Routing PCE Controller

Protocol for client/server relationship for Path Computation Communicate

- RFC4655
- Other options : CLI, NETCONF, new BGP NLRI for SRTE etc.

PCEP Architectural Introduction

- Path Compute Element (PCE):
  - Stores TE Topology Database
  - Computes Network Path based on constraints
  - May initiate Path Creation
- Path Compute Client (PCC):
  - Requests path computation by PCE
  - Send Path updates to PCE
- Path Compute Element Protocol (PCEP):
  - Protocol for PCE-PCC Communication



Centralized Control for SRTE – Building Blocks

BGP Link State (BGP-LS)

Path Computation Element Protocol (PCEP)

Segment Routing PCE Controller

Central controller with full LSDB view

- PCE relationship with HeadEnd nodes
- Computes/communicates path using constraints
- Northbound API for App control

SR PCE Functions & Building Blocks

Topology Collection

Learn Network Topology across domains

Maintain Multi-domain Network Topology

Uses: BGP-LS or IGP

Application Interface

REST API Based communication with External Applications

Path Computation

Compute TE path based on policy constraints

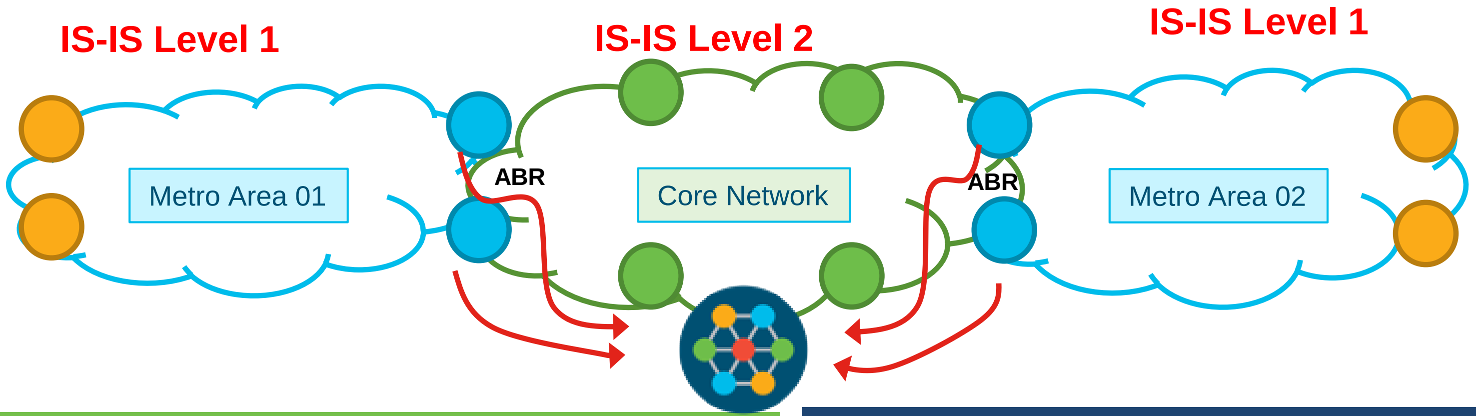
Uses PCEP for accepting path computation requests

Deploy TE Policy

Program & Instantiate TE policy on the Headend

Uses PCEP or BGP-SR

PCEP Client and Server Configuration



PCE Client Configuration

```
segment-routing
traffic-eng
 pcc
   source-address ipv4 6.1.1.1
   pce address ipv4 6.1.1.100 precedence 100
   pce address ipv4 6.1.1.101 precedence 101
               ! Higher precedence server preferred
```

PCE Server Configuration

```
pce
 address ipv4 6.1.1.100  → Enable PCE Server
 rest  → Option, Enable Application Access

 state-sync ipv4  6.1.1.101 → Sync between
                              SR-PCE pair

 peer ipv4 6.1.1.1 → Optional, for Remote SRTE
                     Policy Instantiation
```

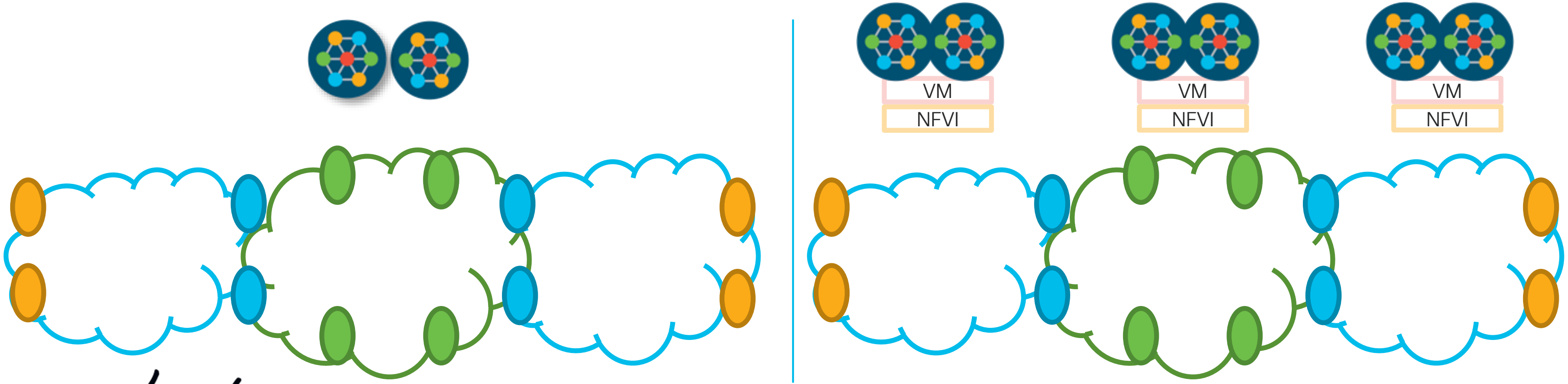

SR PCE Design Considerations

SR PCE runs as IOS XR feature

Physical vs Virtual

Centralized vs Distributed

Cost, Complexity, Scale



Simplifying Routing Design

How Do We Get There?

Multi-Step Network Evolution

Pave the Path to Simplification

1. Architectural and Service Parity: Programmable Transport Uderlay ✓
2. Introduce Central Intent Based Path Control ✓

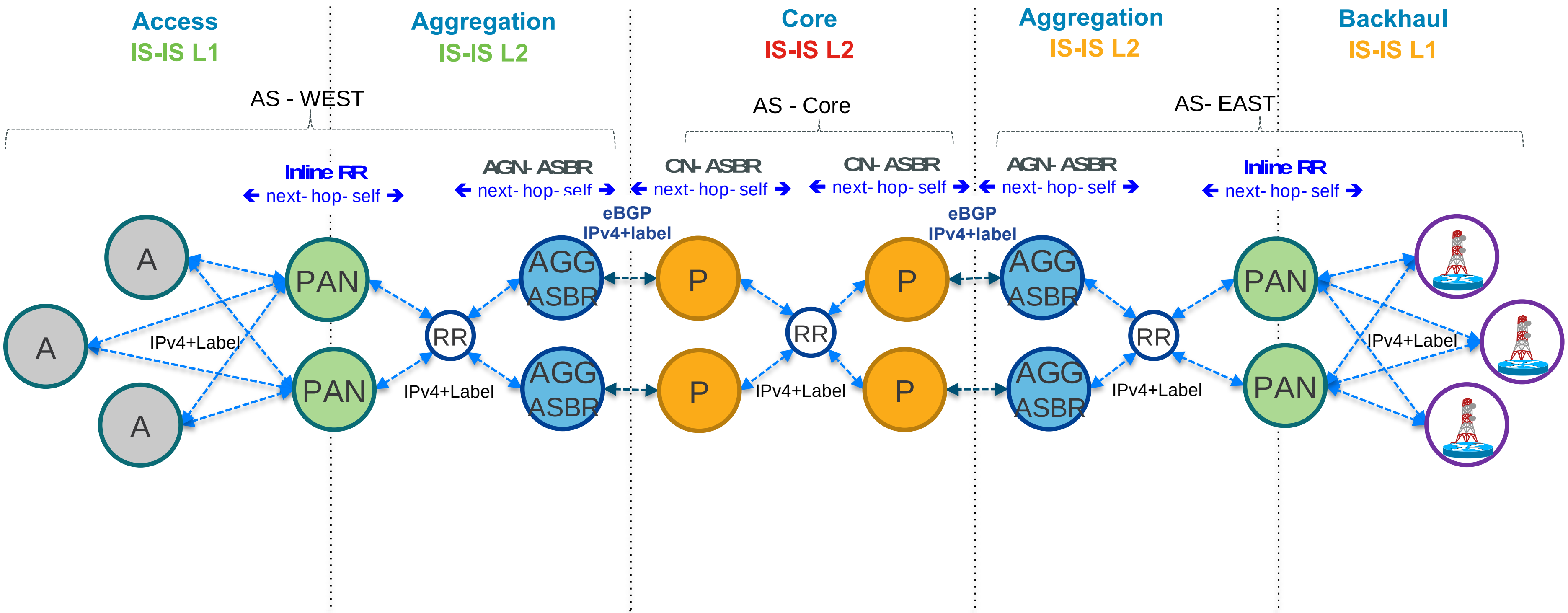
Simplify

3. Simplify Routing Design

Enhance & Monitize

4. New Features to Support Value Add Services
5. Ready for Software Driven Usecases

BGP- LU, with MPLS- LDP Design - Challenges



BGP-LU, with MPLS-LDP Design - Challenges

- Scalability Issues: Access Nodes need to have /32 route to each service destination
- Complicated route filtering between ISIS Areas/Process
- Community based BGP filtering to support typical low cost access platform limitations
- No programmability, value added services such as Low Latency, Secure network slice are hard to configure

On-Demand Next Hop (ODN)

Value Proposition

Simplification

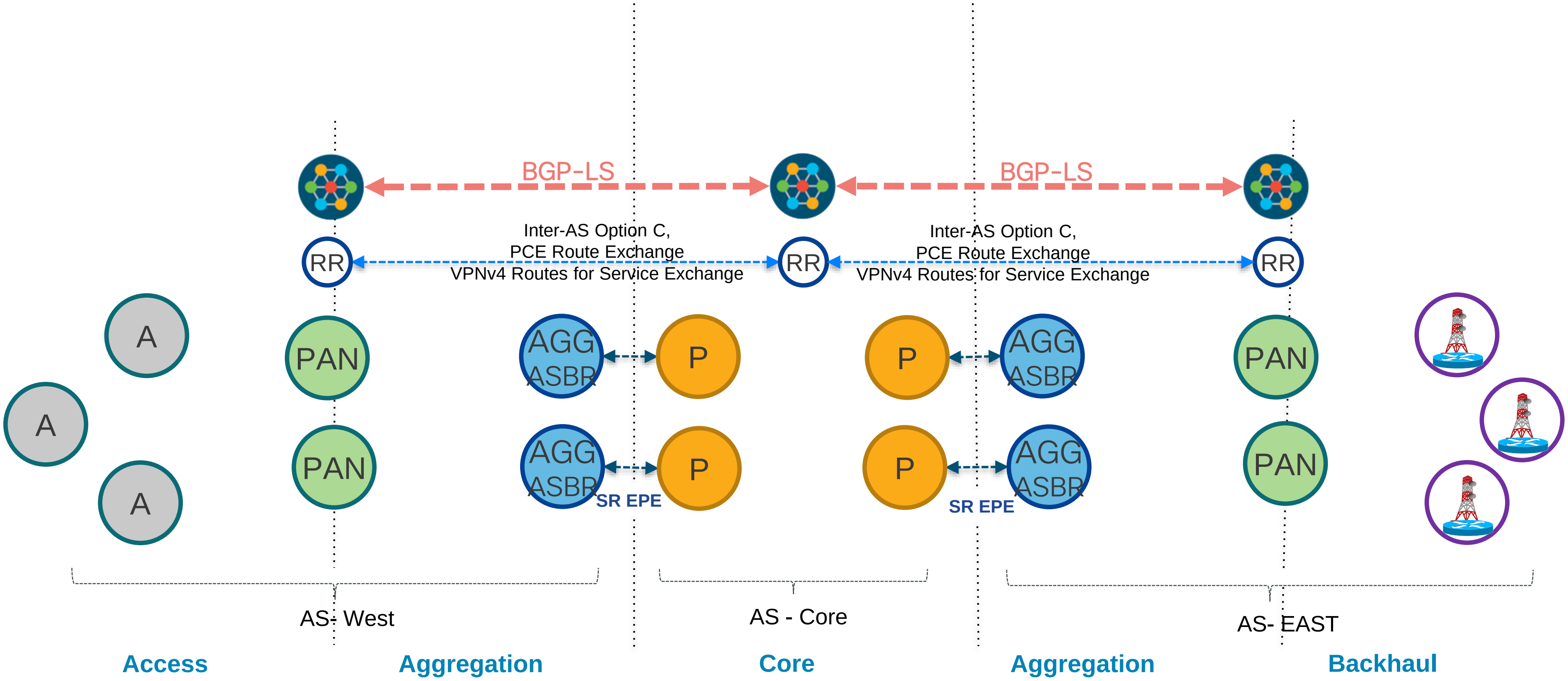
- No need for a per-destination intent policy
- ODN works as a “template”
- Specify only intent and color
- Intent applies to all Service routes/dest that matches the color
- No Need to have /32 route per service destination

OnDemand Policy Instantiation

- Intent can be pre-configured
- No policy is instantiated or programmed
- Policy only instantiated when a route is received for that Intent
- Policy de-programmed, once the route goes away, freeing up resources
- Very helpful for bursty, sporadic traffic .. Like IOT

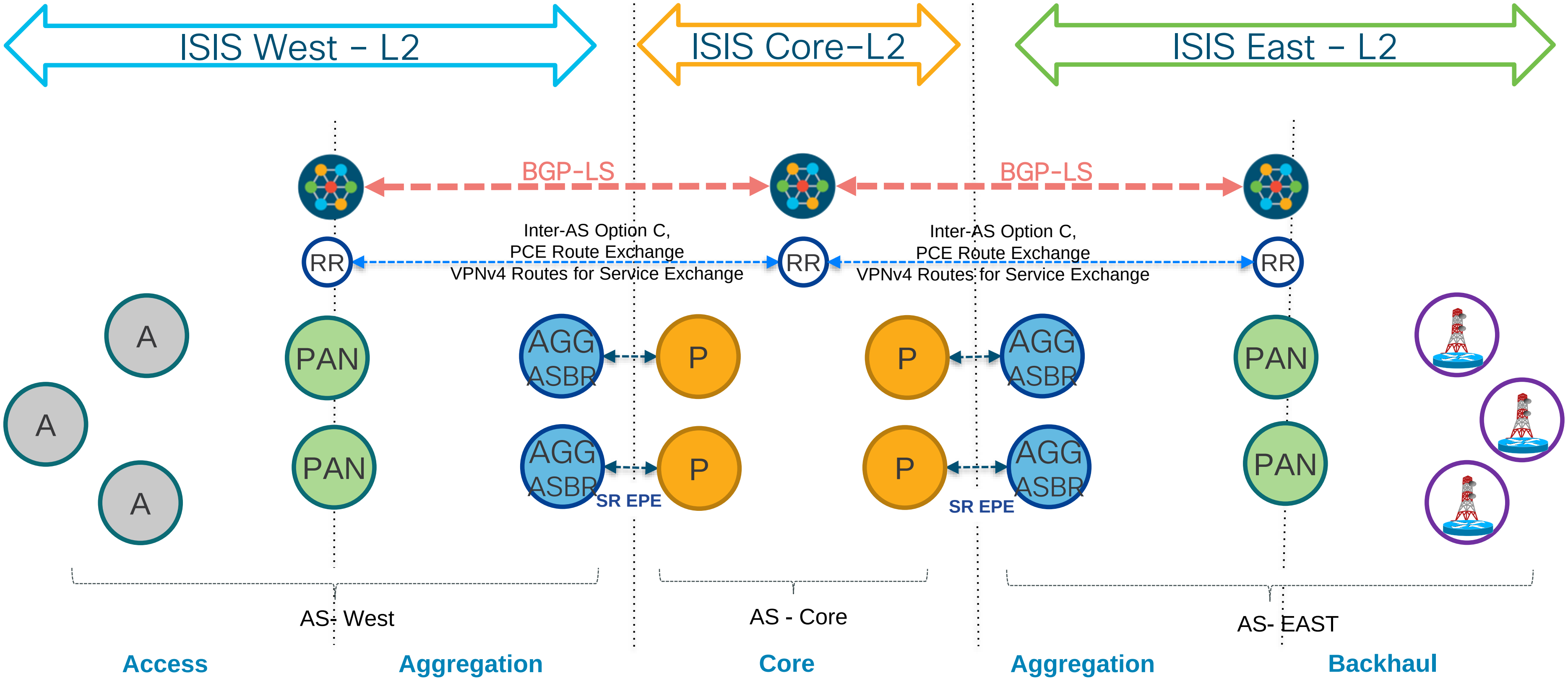
Simplify Routing Design

Using SR, SRTE, SR-PCE



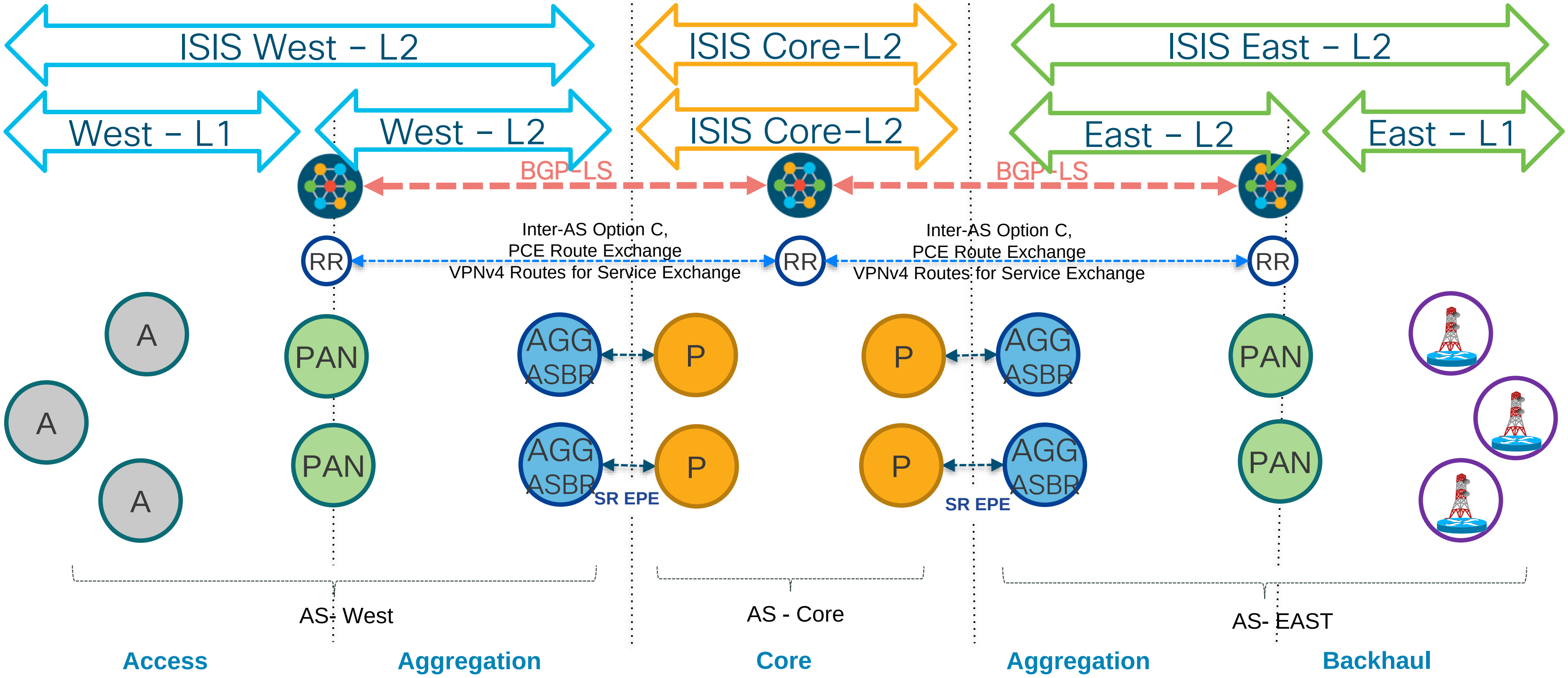
Simplify Routing Design

Using SR, SRTE, SR-PCE



Simplify Routing Design

Using SR, SRTE, SR-PCE



Simplify Routing Design

Using SR, SRTE, SR-PCE, ODN

- Scaleable: Access Nodes don't need BGP/IGP route to remote service nodes by using ODN
- Simple: No L1/L2 leaking, No BGP-LU routes
- Extensible : PCE could be part of Multiple L1/L2 areas and have reachability to Access Nodes
- Redundant SR-PCE's in different domain can communicate through IGP route leaking (PCE only) or through inter-AS option C through RR

How Do We Get There?

Multi-Step Network Evolution

Pave the Path to Simplification

1. Architectural and Service Parity: Programmable Transport Uderlay ✓
2. Introduce Central Intent Based Path Control ✓

Simplify

3. Simplify Routing Design ✓

Enhance & Monitize

4. New Features to Support Value Add Services
5. Ready for Software Driven Usecases

Using Low Latency Intent With SRTE

- SRTE can be used to provide “Low Latency” services as well
- A Low Latency Path Intent requirements:
    1. Per Link Delay Measurement and distribution
    2. Ability to use “Latency” as SRTE “Metric”
- “Per Link Delay Measurement” feature used to calculate delay
- Delay values are injected into IGP for simplicity and scalability

```
segment-routing
  traffic-eng
    policy Low_Latency_Intent
      color 20 end-point ipv4 6.1.1.10
      candidate-paths
        preference 100
          dynamic
            metric
              type latency
         preference 200
```

Latency Metric

Per-Link Delay Measurement Basics


```
performance-measurement
   interface GigE0/0/0/0
      delay-measurement
```

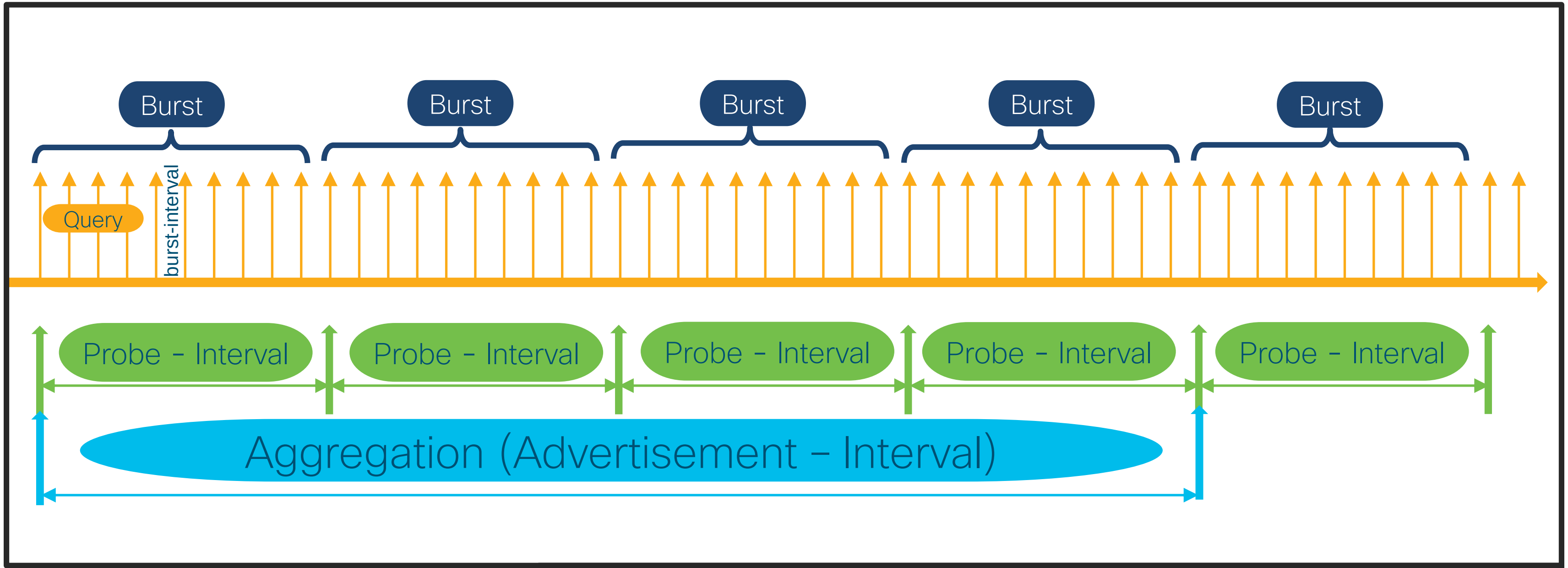
```
performance-measurement
   interface GigE0/0/0/0
      delay-measurement
```

- Simple 1-line configuration to enable Per-Link DM.
- Delay values (min, max, Avg) and jitter advertised via TLVs in IGP's LSP/LSA
- SRTE uses Min-Delay metric only
- Various configurable parameters for perLink DM configuration and advertisement

Per-Link DM Measurement: Common Terminology

Query	• One single delay measurement packet
Burst	• A series of queries sent at pre-defined burst intervals
Probe	• A burst of queries sent within a “prober interval) • Compute min, max, avg, var after each probe
Aggregation	• A series of 1 or more Probes • Periodic “conditional” advertisement sent per aggregation interval

Workflow Example @ Default Interval values



- Each query 3 sec apart (burst Interval)
- 10 Queries per Burst
- 30 sec probe interval (1 burst per probe)
- 120 sec aggregation interval (4 probes/aggregation)

DM Probe Configuration Parameters

Delay Measurement Configuration Options

```
performance-measurement
  delay-profile interfaces
    probe
     one-way
     interval < 30-3600 sec >
     burst
       count < 1-30 count >
       interval < 30-15000 msec >

     advertisement
       periodic
         disabled
         interval < 30-3600 sec >
         threshold < 0-100% >
         minimum-change <0-100000 usec>
       accelerated
         threshold < 0-100% >
         minimum-change <1-100000 usec>
```

Global Default Profile. Set Configurable parameters for all DM enabled interfaces

1-way or 2-way DM. Default is 2 way, 1-way requires clock sync (not covered here)

Default Probe Interval is 30 seconds

Defines number of Queries in a Burst – Default is 10 queries per burst

Time between each query, default is 3000 msec, lowest is 300msec (dependent on LC)

Burst count * Burst Interval CANNOT be more than Probe Interval (config check is enforced)

DM Periodic Advertisement Configuration Parameters

Delay Measurement Configuration Options

```
performance-measurement
  delay-profile interfaces
    probe
     one-way
     interval < 30-3600 sec >
     burst
       count < 1-30 count >
       interval < 30-15000 msec >

     advertisement
       periodic
         disabled
         interval < 30-3600 sec >
         threshold < 0-100% >
         minimum-change <0-100000 usec>
       accelerated
         threshold < 0-100% >
         minimum-change <1-100000 usec>
```

Periodic advertisements are enabled by default. Could be disabled by this CLI

Default 120 sec, rounded to next probe-interval multiple. Periodic adv sent at this interval if the min-change AND threshold values are met for the new “min” latency.

Default threshold is 10%, default min-change is 500usec (100km optical fiber delay).

Periodic Adv is sent ONLY if the new min latency value is more than old value by BOTH threshold AND min-change amount

If both values exceed in new min latency vs old min latency, then all DM values (min, max, avg, var) are flooded via IGP

DM Accelerated Advertisement Configuration Parameters

Delay Measurement Configuration Options

```
performance-measurement
  delay-profile interfaces
    probe
     one-way
     interval < 30-3600 sec >
     burst
       count < 1-30 count >
       interval < 30-15000 msec >
     advertisement
       periodic
         disabled
         interval < 30-3600 sec >
         threshold < 0-100% >
         minimum-change < 0-100000 usec >
       accelerated
         enabled
         threshold < 0-100% >
         minimum-change < 1-100000 usec >
```

When enabled, Accelerated advertisements will flood updated Latency values after every probe interval, but only if the Threshold and Min-Change parameters are met.

Accelerated advertisements will always be atleast 1-probe interval apart

Accelerated advertisements are DISABLED by default. Can be enabled through this CLI

Default threshold is 20%, default min-change is 500usec
DM Values are flooded if both values are exceed in the latest DM Probe when compared to the current min-latency value.

Static DM Configuration CLI

- Static delay can be configured on interfaces
  - When static delay is configured, advertisement is immediately triggered with following:

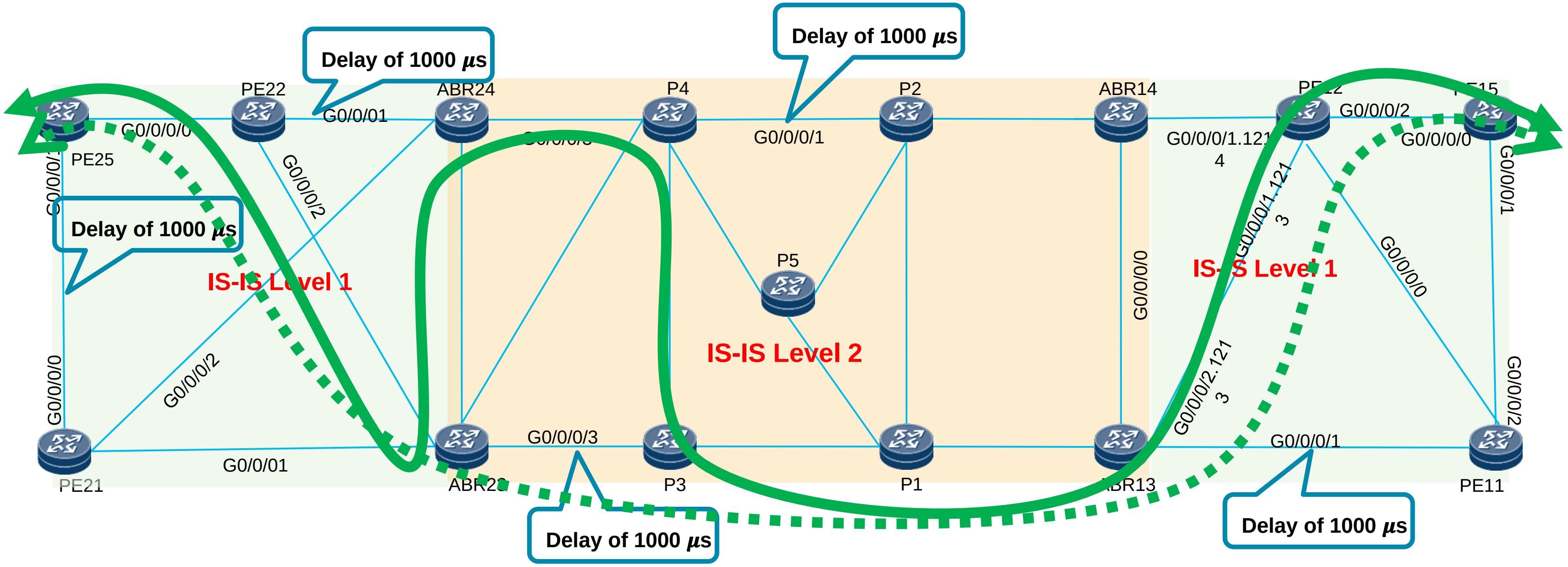
    min-delay = max-delay = avg-delay
    variance = 0

  - Probes are continued to be scheduled and delay metrics are aggregated, stored in the history buffers and streamed.
  - Adv. threshold checks are suppressed resulting in no flooding/advertisement of the currently measured delay values.
  - When the advertise-delay is un-configured, the next scheduled advertisement threshold check will update the advertised delay values, if required.

Static DM Configuration

```
performance-measurement
  interface <tengig 0/0/0/0>
    delay-measurement
     advertise-delay <value in uSec>
```

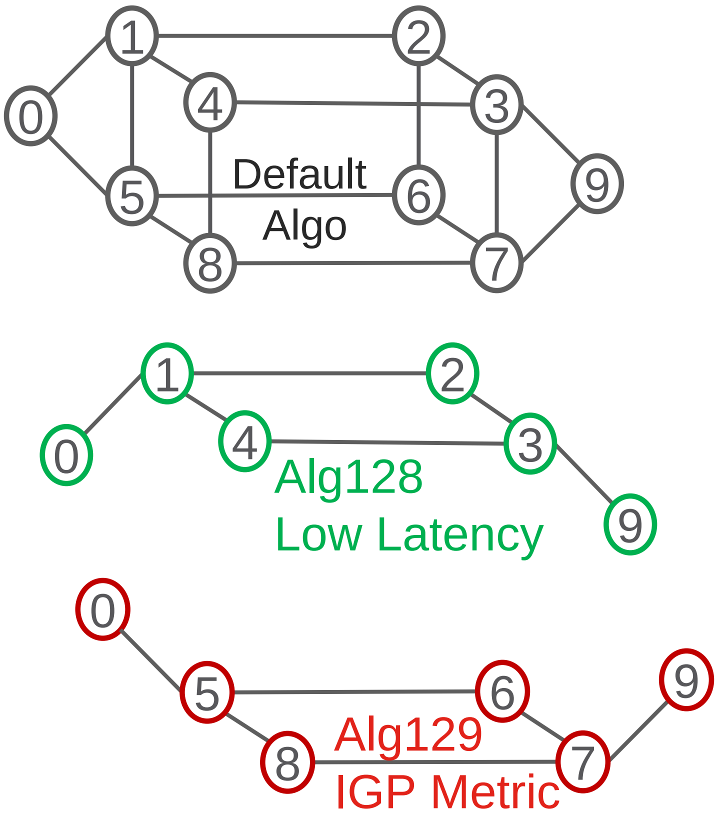

Low Latency Intent – E2E Traffic Sample



Best Low Latency Path
Best IGP Path

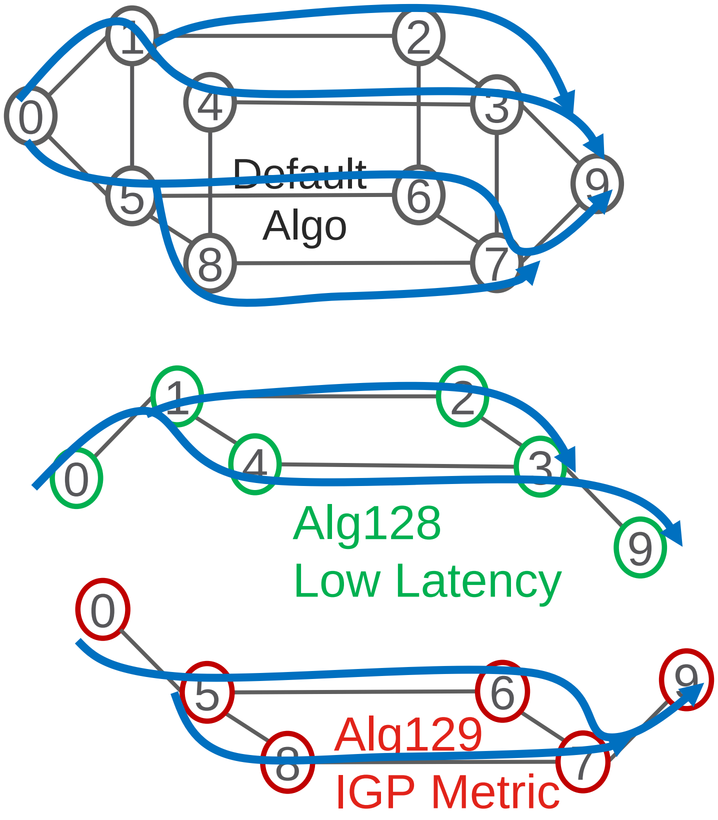
Network Slicing through Flex-Algo

```
router isis 1
net 49.0001.0000.0000.0002.00
 flex-algo 128
   metric-type latency
 !
 address-family ipv4 unicast
   router-id 6.1.1.9
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 2
   prefix-sid algorithm 128 absolute 16802

segment-routing
  traffic-eng
    on-demand color 100
        dynamic
           sid-algo 128
```



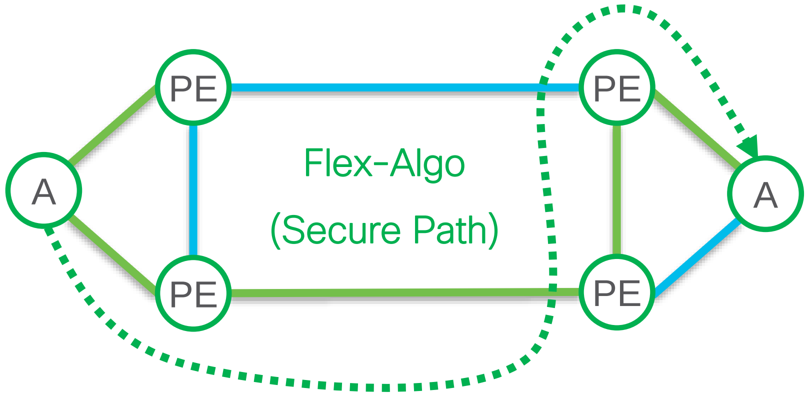
Network Slicing through Flex-Algo

```
router isis 1
net 49.0001.0000.0000.0002.00
 flex-algo 128
   metric-type latency
 !
 address-family ipv4 unicast
   router-id 6.1.1.9
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 2
   prefix-sid algorithm 128 absolute 16802

segment-routing
  traffic-eng
    on-demand color 100
       dynamic
          sid-algo 128
```



Intent Statement: Secure Slice Usecase

1. Financial customer asks for a secure path E2E
2. Requests link-level encryption for any of its traffic
3. Using Lowest Latency possible is still part of their "intent"
4. **Your solution:** You will create a "Secure Network slice" using Flex algo that would avoid non-encrypted links



Fulfilling New Services and Service Requirements

Bulk Update

High Bandwidth
SW Updates
Sporting Events

Intent Definition:
Bandwidth Signaling

IOT

M2M Non-Critcal
Low intensity Bursts
Smart Services

Intent Definition:
ODN

Entertainment

AR, VR, Gaming
Upsell Opportunities
User Experience

Intent Definition:
Latency Bound

Mission Critical

Ultra Reliable Low Latency
Public Health
Self Driving Cars

Intent Definition:
Low Latency

Private Network

Create your own Slice
Industry Verticals
Security, 5G

Intent Definition:
Flex-Algo
Constraints
Network Slicing

How Do We Get There?

Multi-Step Network Evolution

Pave the Path to Simplification

1. Architectural and Service Parity: Programmable Transport Underlay ✓
2. Introduce Central Intent Based Path Control ✓

Simplify

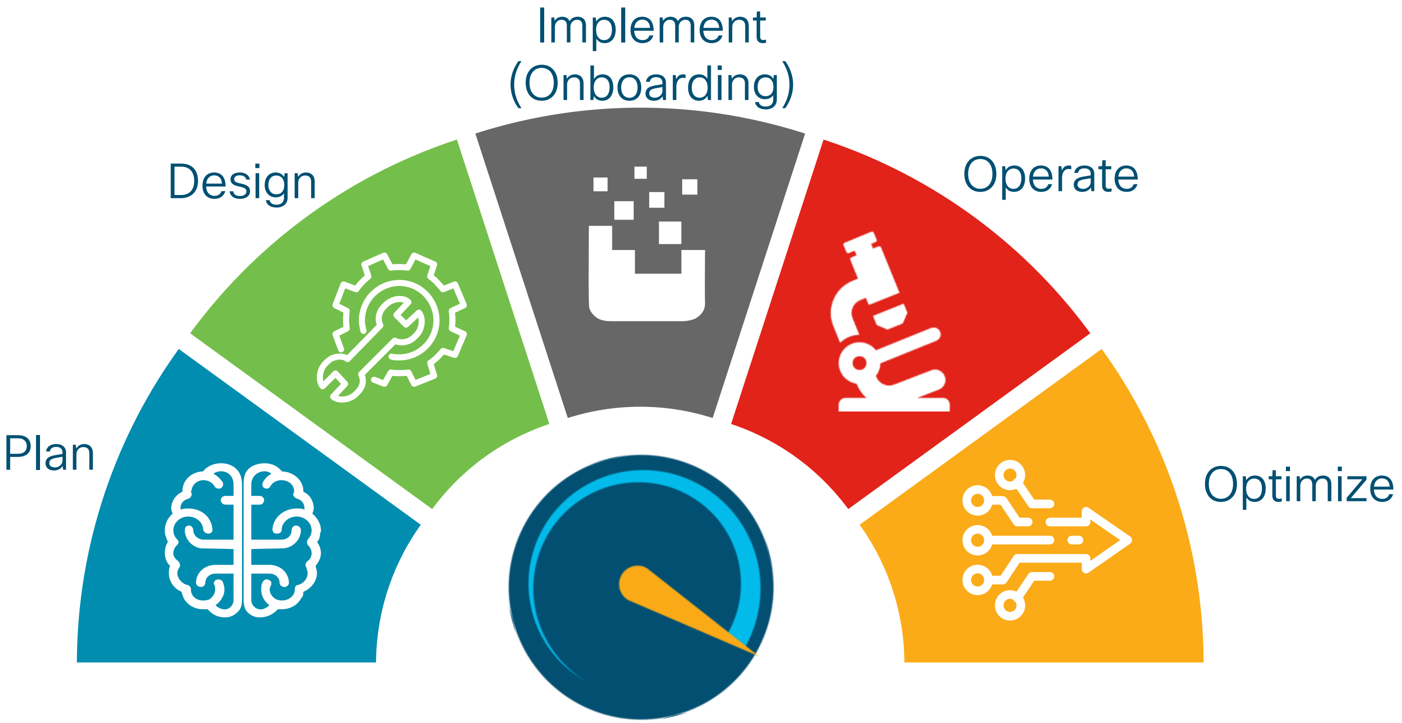
3. Simplify Routing Design ✓

Enhance & Monitize

4. New Features to Support Value Add Services ✓
5. Ready for Software Driven Usecases ↓

A Foundation for Operations Evolution

Architectural and Service Lifecycle



Modernizing Device Lifecycle

	Device Onboarding (Day 0)	Service Orchestration (Day 1)	Monitoring, Analytics, Operations (Day 2)
Yesterday	• Costly device Bring up • Skilled labor required on site • Manual, lengthy config process	• Scripts for service bring-up • High Maintenance, inflexible	• Non Scalable pull mechanism • Unstructured and periodic data bursts
Today	• Zero touch Deployment • Orders of Magnitude faster	• Automation friendly, flexible • Vendor neutral Model Driven Service bring up	• Near real time push mechanism • Consistent, scalable and machine readable

Orchestration, Data Analytics, Closed Loop Automation drive next generation SP Ops

Operational Efficiency: Automated Onboarding

Why Zero Touch Deployment

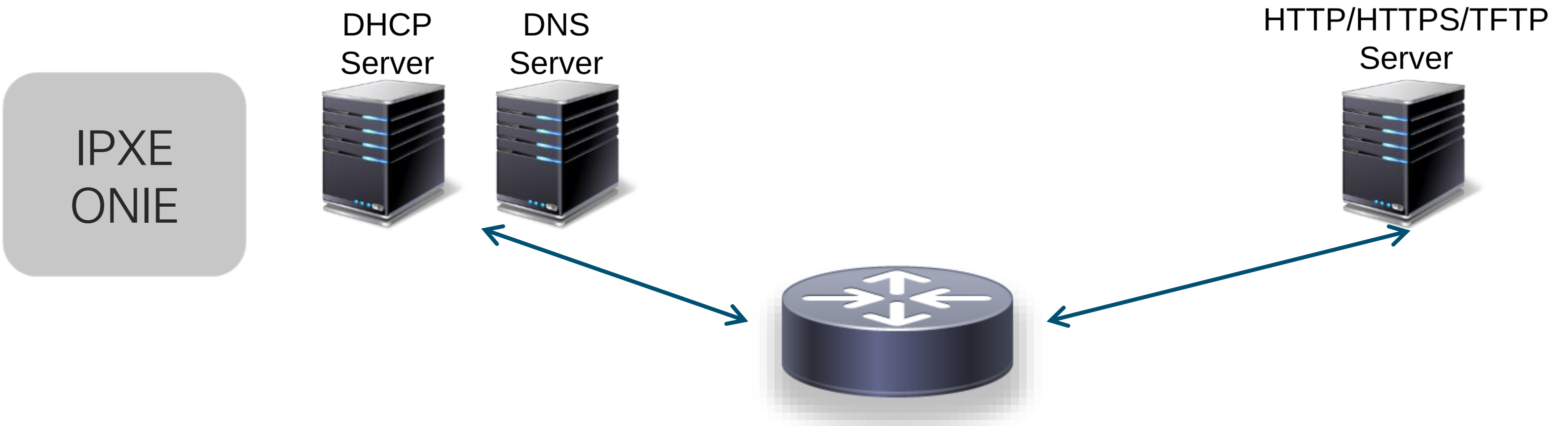
Agility and Speed of Deployment

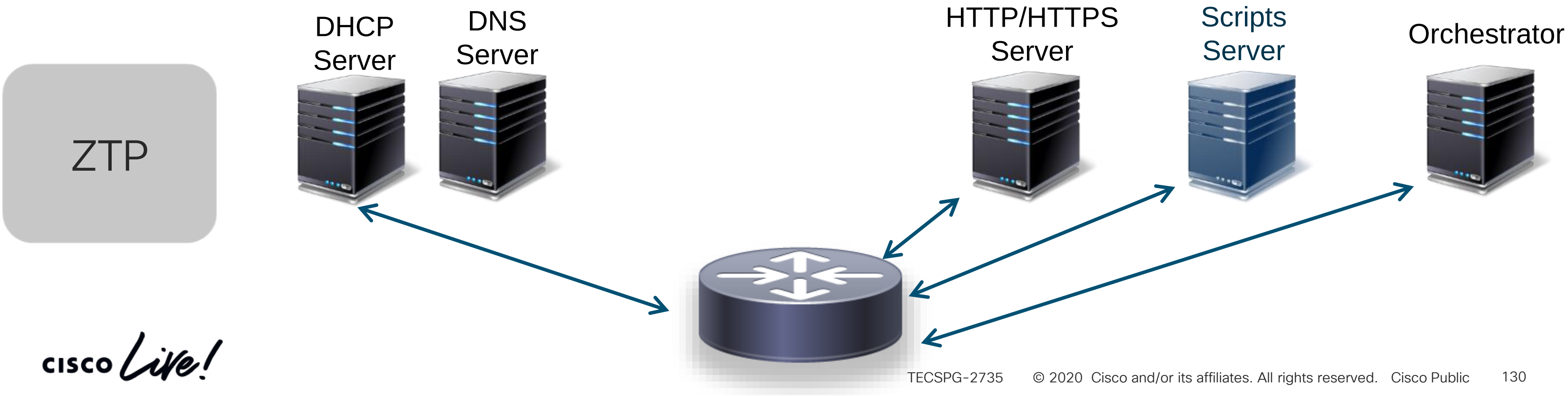
Automation for Repetitive Tasks

OpEx Savings by Minimizing Human Errors

TTM for growing number of Devices

ZTD Architectural Components

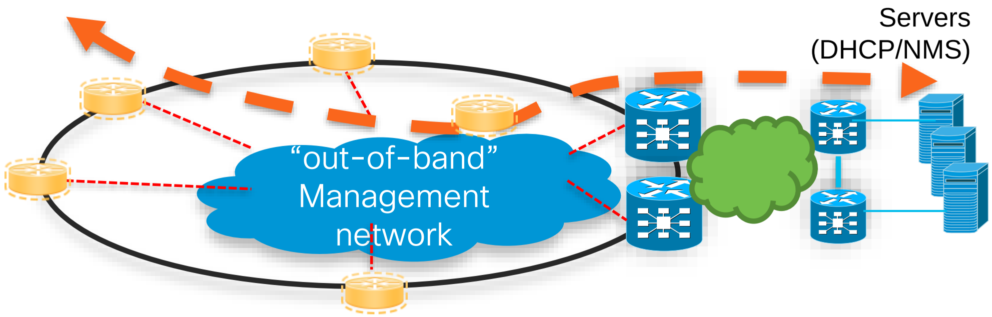




ZTP – Two Different Deployment Scenarios

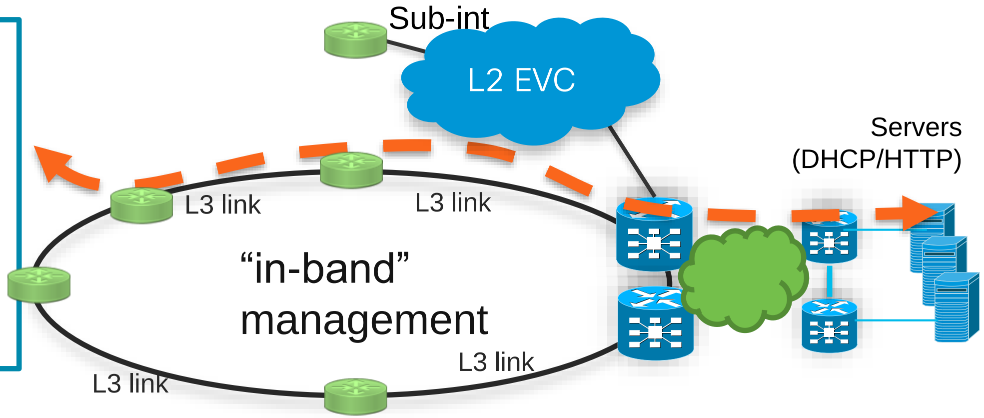
1

- Routers are connected to a management network via out-of-band management port
- Popular in Data Center, Enterprise, and Web customers

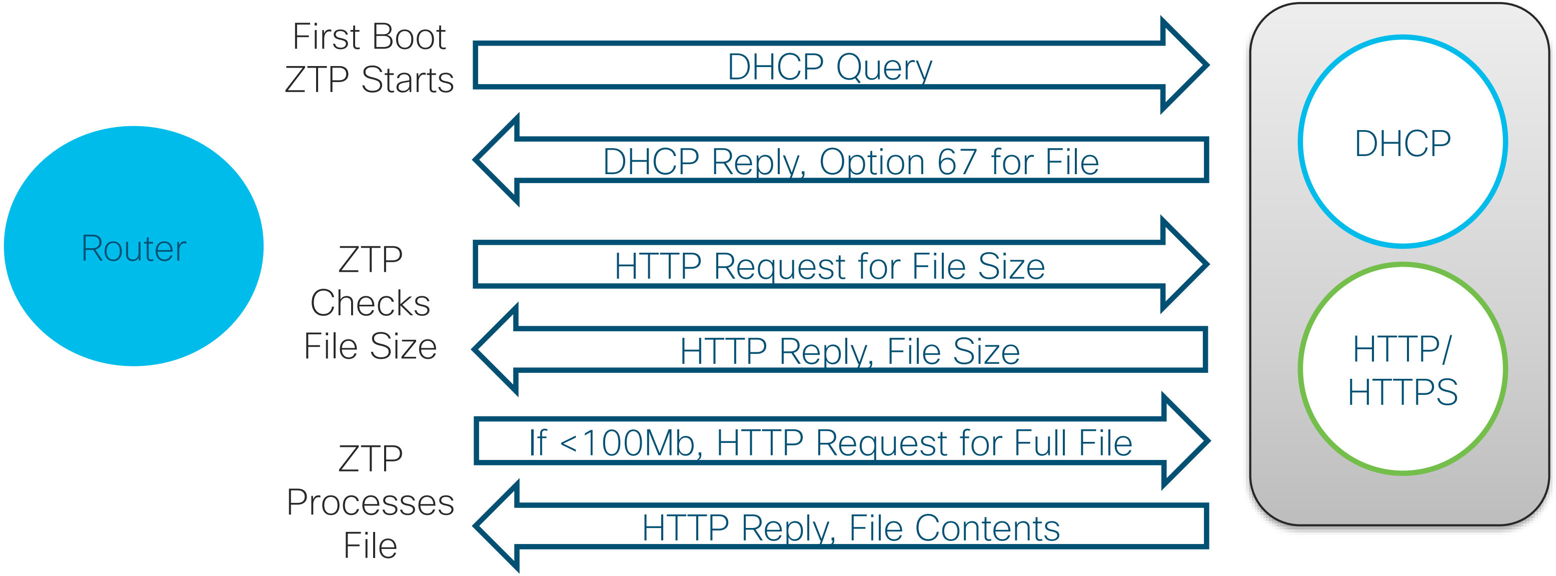


2

- There is no dedicated management network.
- Routers are managed via in-band, the same as user data network
- Typical deployment in the SP Access/Metro



Zero Touch Provisioning (ZTP) Eco-System

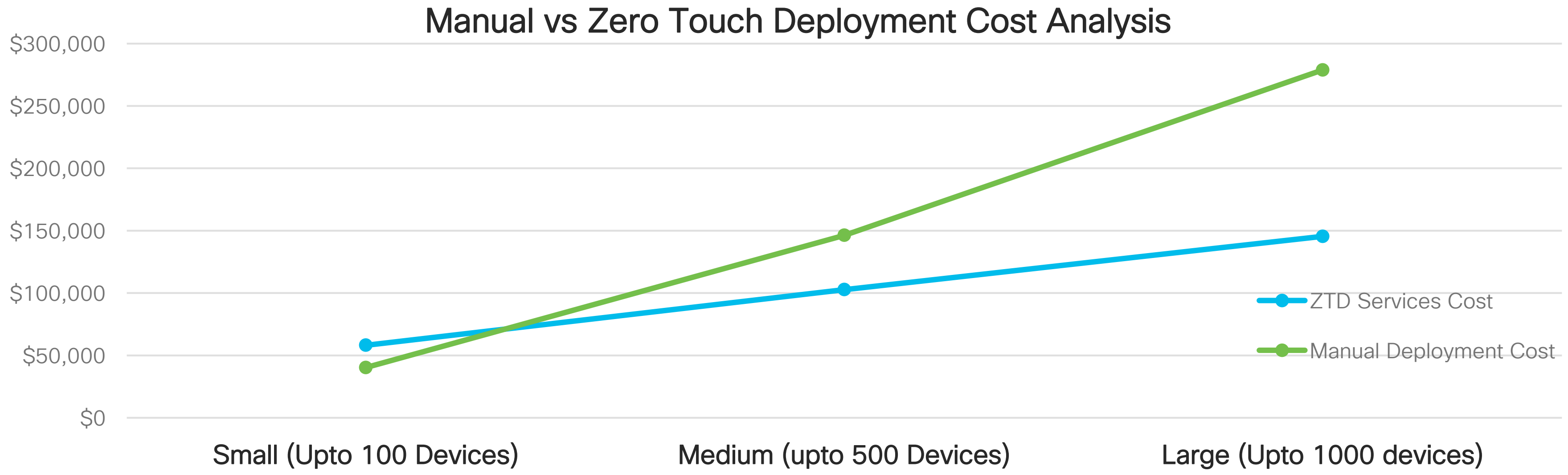


File Starting with !! IOS XR Treated as Configuration File	File Starting with #! /bin/bash #!/bin/sh #!/usr/bin/env python Processes as script

Cost Analysis*: Manual vs Zero Touch Deployment

- Higher startup cost for smaller deployments
- Cost savings beyond ~200 devices using ZTD

Deployment Type	ZTD Cost	Manual Deployment Cost
Small (Upto 100 Devices)	$58,186.85	$40,267.22
Medium (upto 500 Devices)	$102,760.00	$146,289.44
Large (Upto 1000 devices)	$145,437.78	$278,817.22



* Analysis based on a specific usecase. Other costs may vary

Case Study – ZTD at Tier 1

Solution Requirements and Components

ZTD Solution Requirements

- Use Cisco ASR-920 as Cell Site Router
- Automatic Configuration of Cell Site Router during Maintenance Window
- No Dedicated Management i.e ZTP over Data Ports
- Use pre-determined VLANs for connectivity

Solution Components

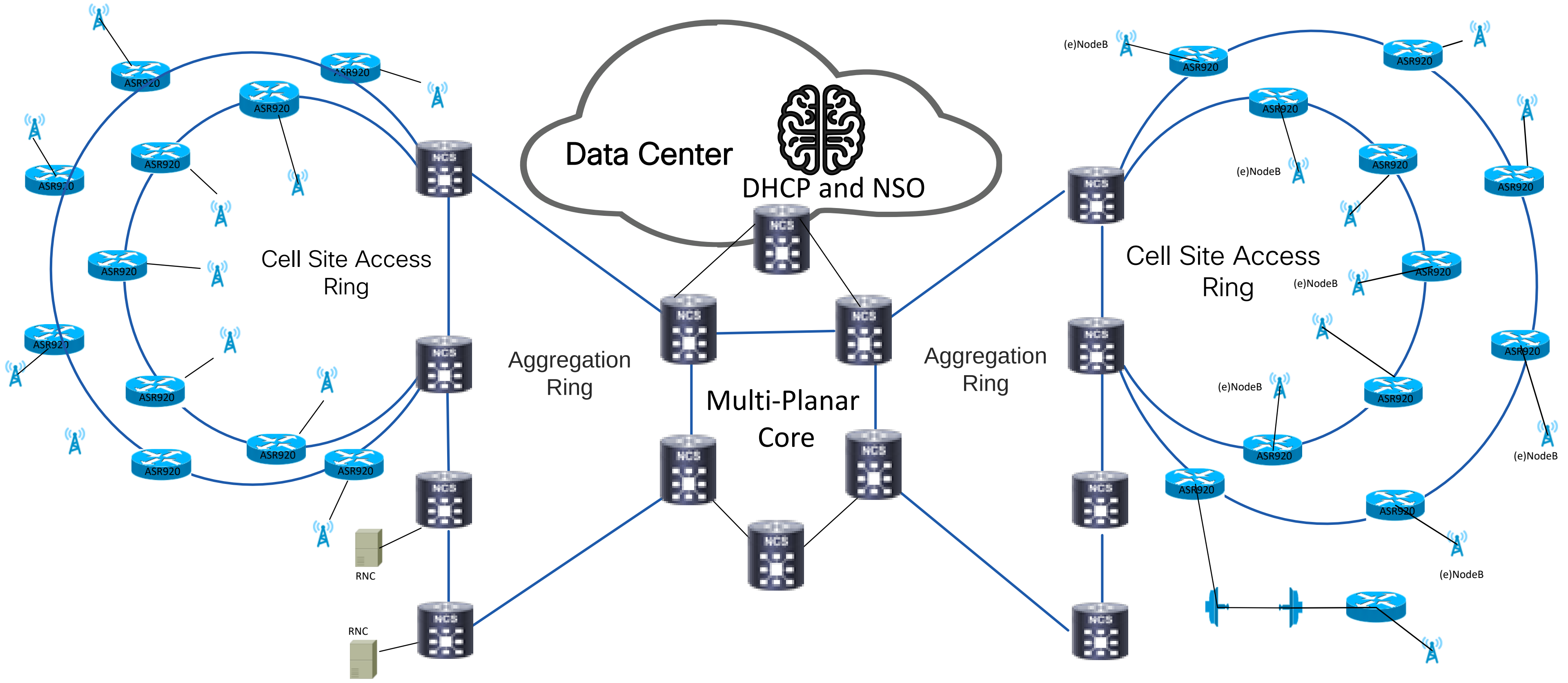
Orchestrator (Cisco NSO)

Cell Site Router

DHCP/TFTP Server

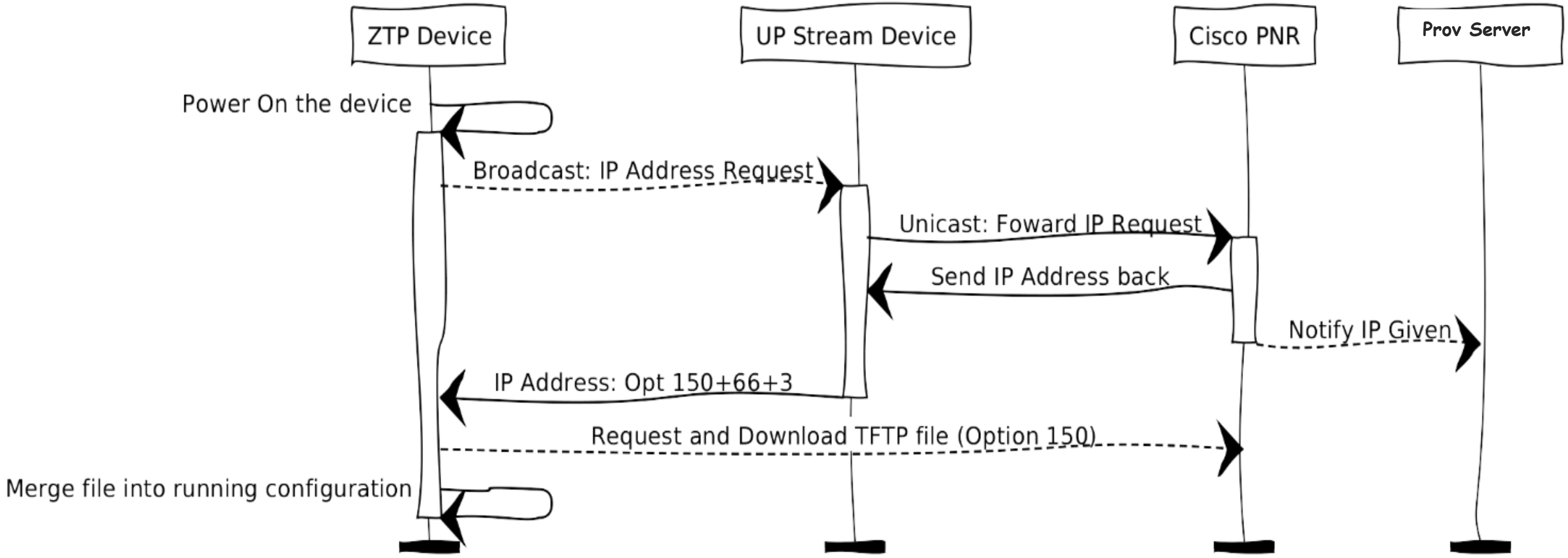
Lease Client and Import Scripts

Case Study – ZTD at Tier 1 SP

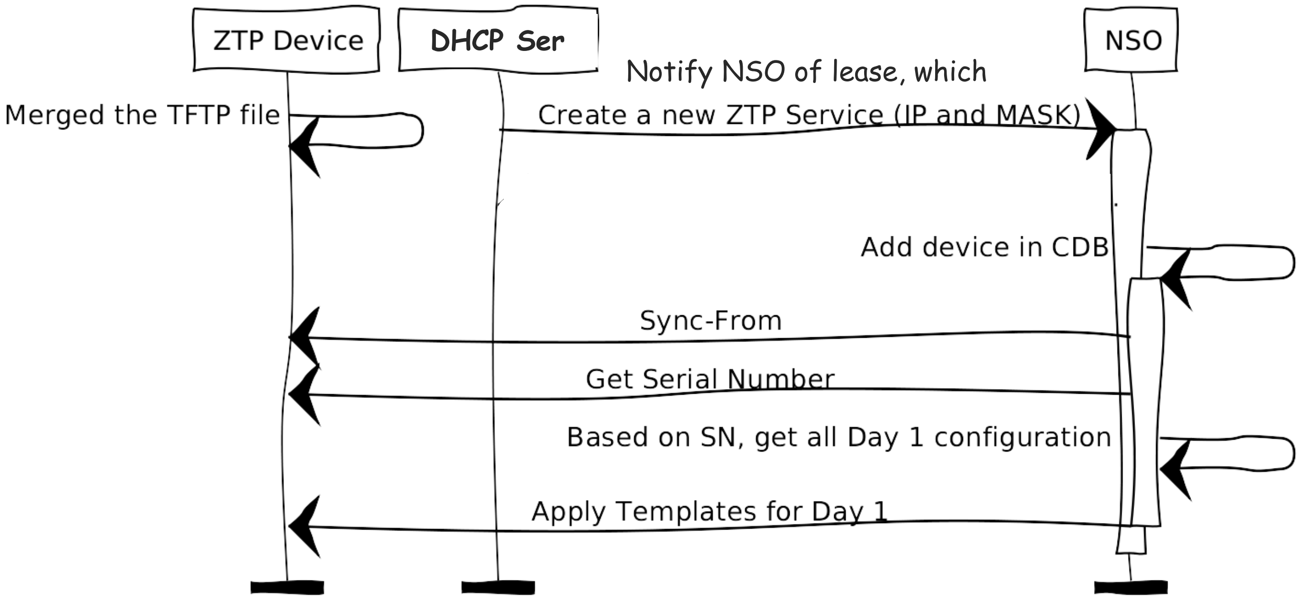


cisco Live!

ZTD Design and Workflow – Device Startup



ZTD Design and Workflow – Config Application

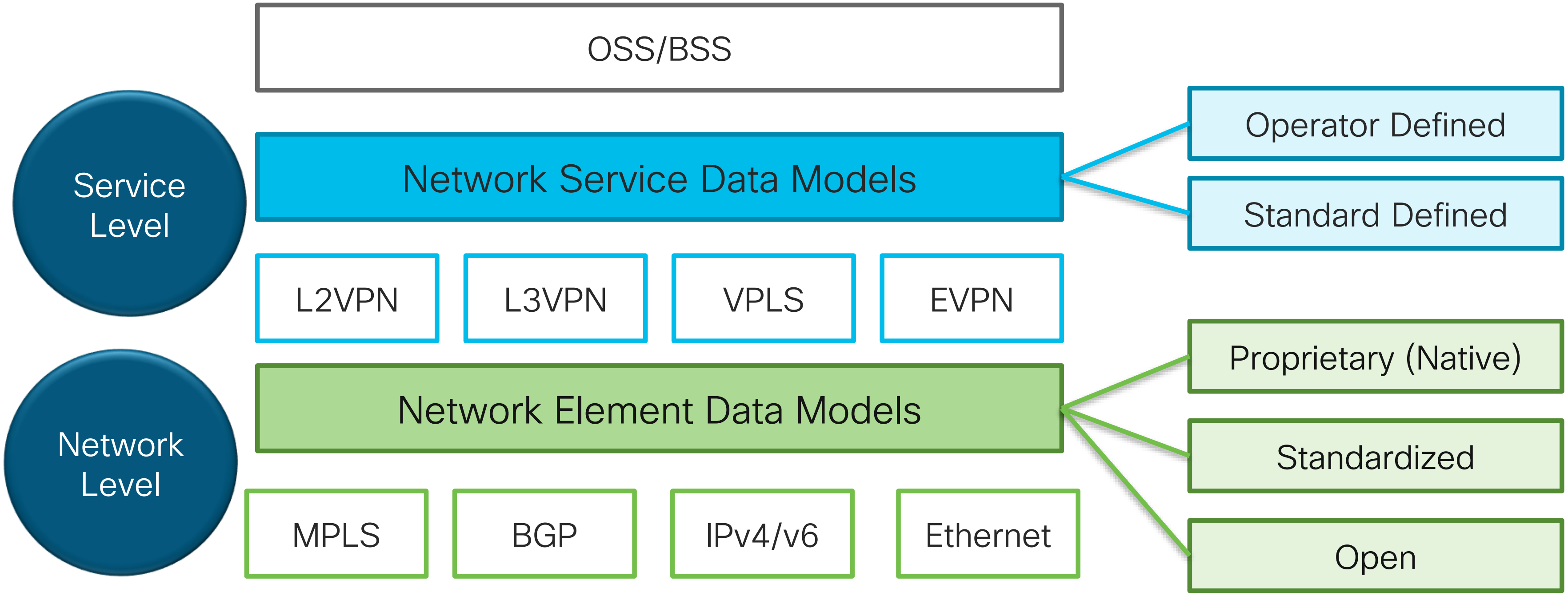


Zero Touch Deployment Challenges & Solutions

Username/Password Required for Telnet	Challenge: Require Authentication config before NSO could connect. Solution: Use TFTP with a common config file for all devices
Device to Site Correlation	Challenge: Serial Number to installation site mapping Solution: Do not pre-assign devices to sites, ask installation team to provide serial number to site mapping via Excel
Preparing Backend	Challenge: NSO to be pre-configured with devices' config templates Solution: Standardized port usage, XLS sheet w/ all Devices baseline parameters, Custom script to import XLS into NSO DB
Process Enhancements	Process enhancement and alignment between Deployment and Ops team

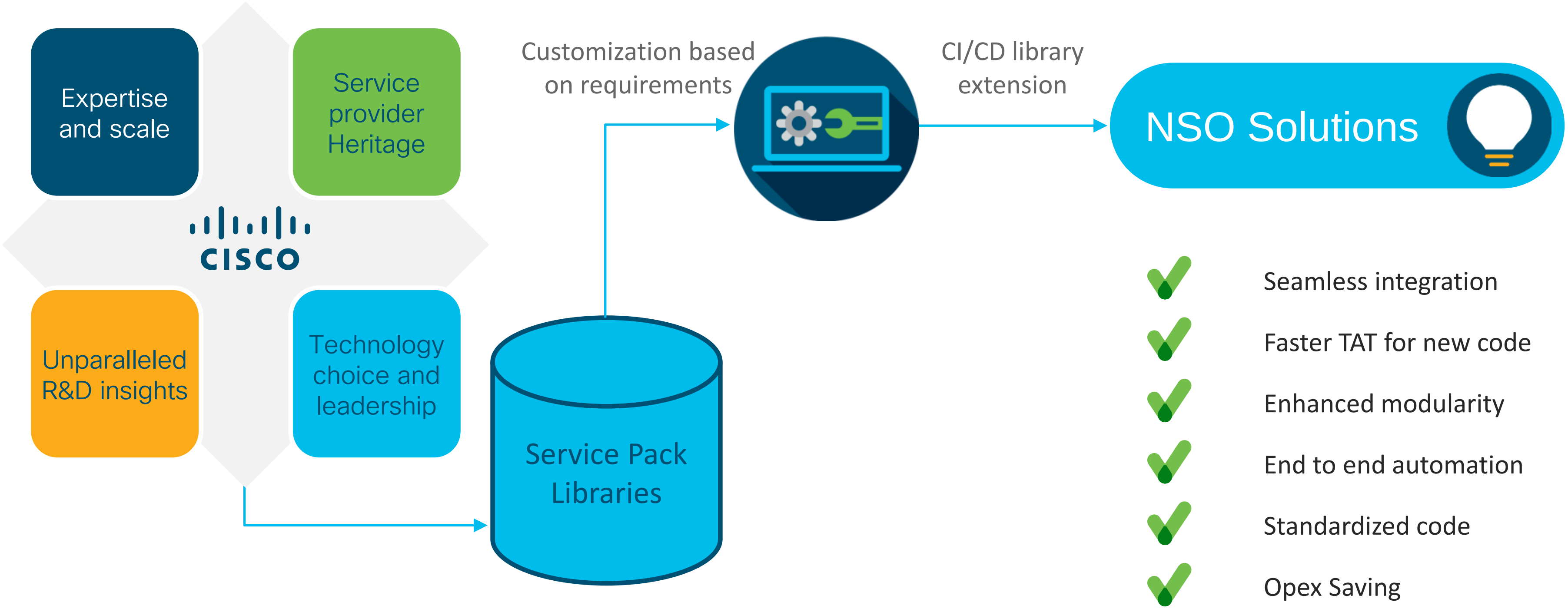
Operational Efficiency: Services Orchestration

Network and Services Orchestration



Refer to “BRKSPG-2303: Model-Driven Programmability for Cisco IOS XR” for more details

Automation Service Packs



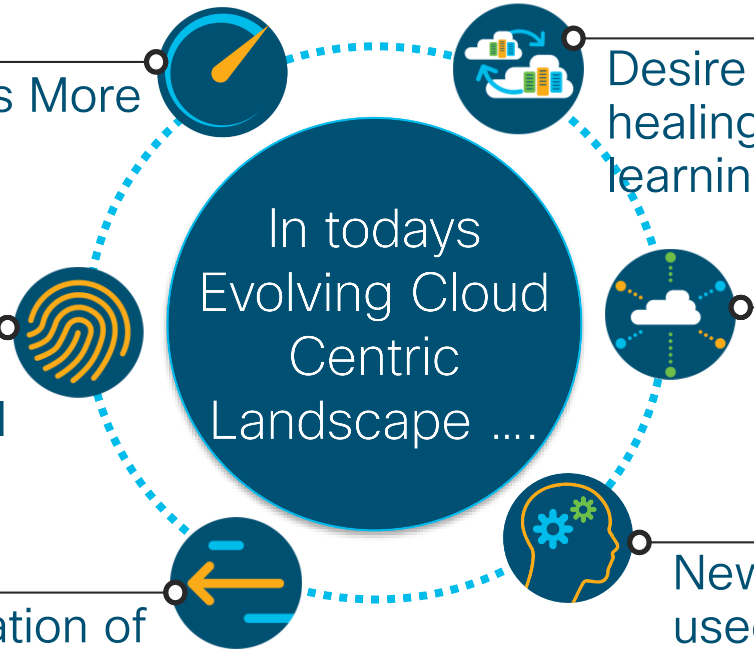
Automation Services Packages
Code libraries for easier insertion and expansion

Reference

Zero Touch Provisioning	Onboard new network devices with no human interaction.
Device OS Upgrade	Upgrade from source OS version to target OS version with pre and post checks
Device Port Turn up	Configure physical ports using configures attributes e.g. VLAN, MTU & Speed
Service Discovery Framework	User can define transformation logic for a particular service and SDF will discover and populate the service model.
Device Migration	Migrate configuration from device A to device B with pre and post checks
Metro E Services	(1) Ethernet Private Line (2) Ethernet Virtual Private Line (3) Ethernet Network Service
ACL Management	Manage firewall ACL in multi-firewall environment

Operational Efficiency: Monitoring and Analytics

Why Network Visibility Matter Today?



Speed Matters More than Size

Desire for self healing and self learning

Dynamic threat landscape requires real time response

Mass Scale Networking and IOT

Operationalization of Steaming Data

New Real time usecases

Pull vs Push Data Collection

Too slow

Not Adequate

For necessities only

SNMP

Syslogs

CLI Based Data Collection

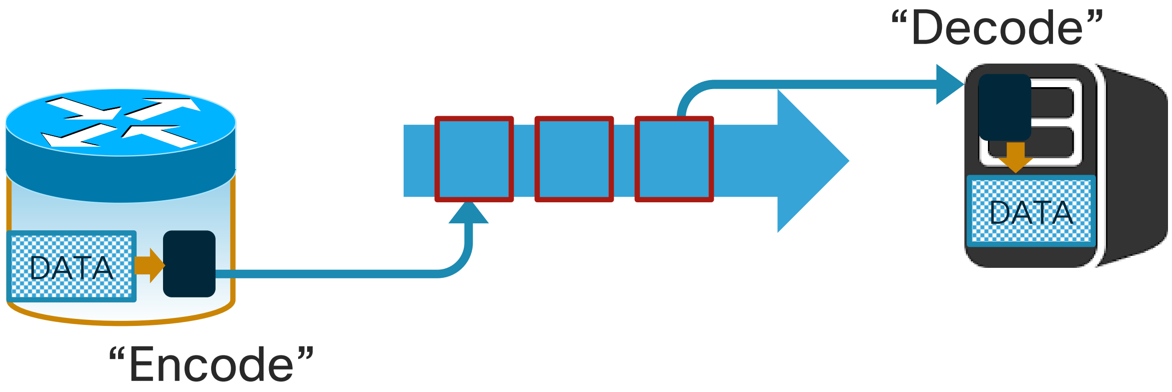
Always available

Fast and convenient

Potential of flooding !!

Telemetry

Telemetry Basic Concept: Encoding



Common Text-Based Encodings

- JSON
- XML

Encoding: XML vs Telemetry

Traditional Networking Approach: XML

```
<interface-name>GigabitEthernet0/0/0/0</interface-name>
  <packets-received>13560392</packets-received>
  <bytes-received>1903082966</bytes-received>
  <packets-sent>2887148</packets-sent>
  <bytes-sent>2482103559</bytes-sent>
  <multicast-packets-received>0</multicast-packets-received>
  <broadcast-packets-received>63445</broadcast-packets-received>
  ...
```

Telemetry: GPB

```
 1: GigabitEthernet0/0/0/0
50: 13560392
51: 1903082966
52: 2887148
53: 0
54: 63445
   ...
```

Other options: self-describing GPB, JSON

Telemetry Basic Concept: Encoding

GPB – Compact Encoding	GPB – Self Describing Encoding
1: GigabitEthernet0/0/0/0 50: 449825 51: 41624083 52: 360333 53: 29699362 54: 91299 <snip>	{InterfaceName: GigabitEthernet0/0/0/0 GenericCounters { PacketsSent: 449825 BytesSent: 41624083 PacketsReceived: 360333 BytesReceived: 29699362 MulticastPacketsReceived: 91299 <snip>

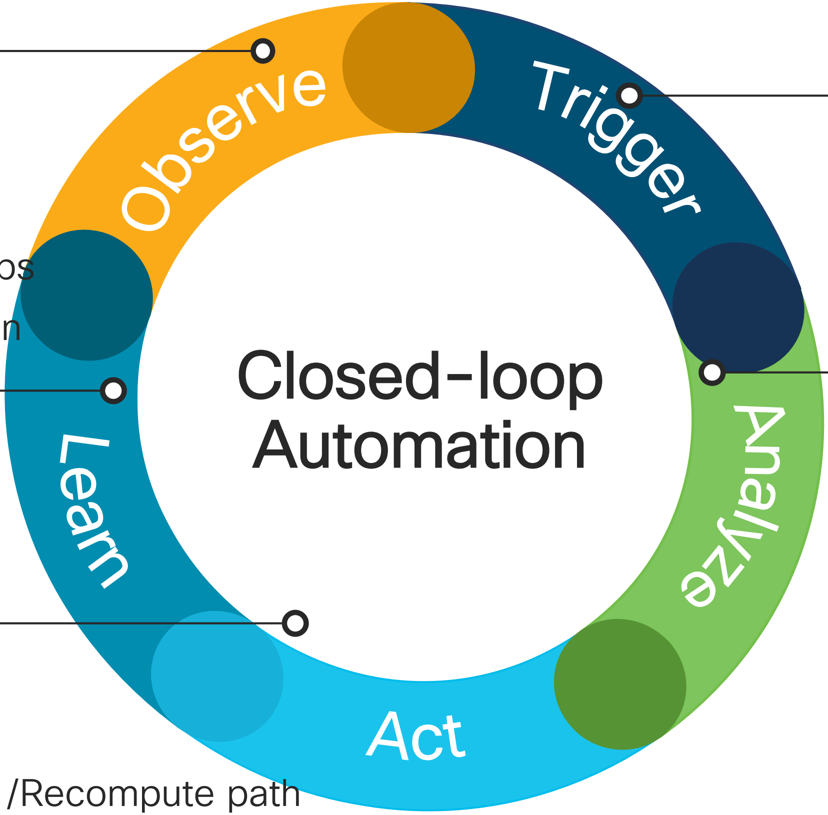
Operational Efficiency through Automation

Problem Resolution Life Cycle

Do you know upto 100% of this could be Automated? *

http://cs.co/BRKSPG-2810

Closed Loop Automation Basics



Observe

Anomaly Observed e.g.

- E2E Latency Change
- Excessive Interface flaps
- High resource utilization

Trigger and Notify

BGL-LS update

Streaming telemetry

Analyze

Corrective action need analysis

Path Precomputation

Interface Isolation

Add new resources

Take Action

Auto Remediation

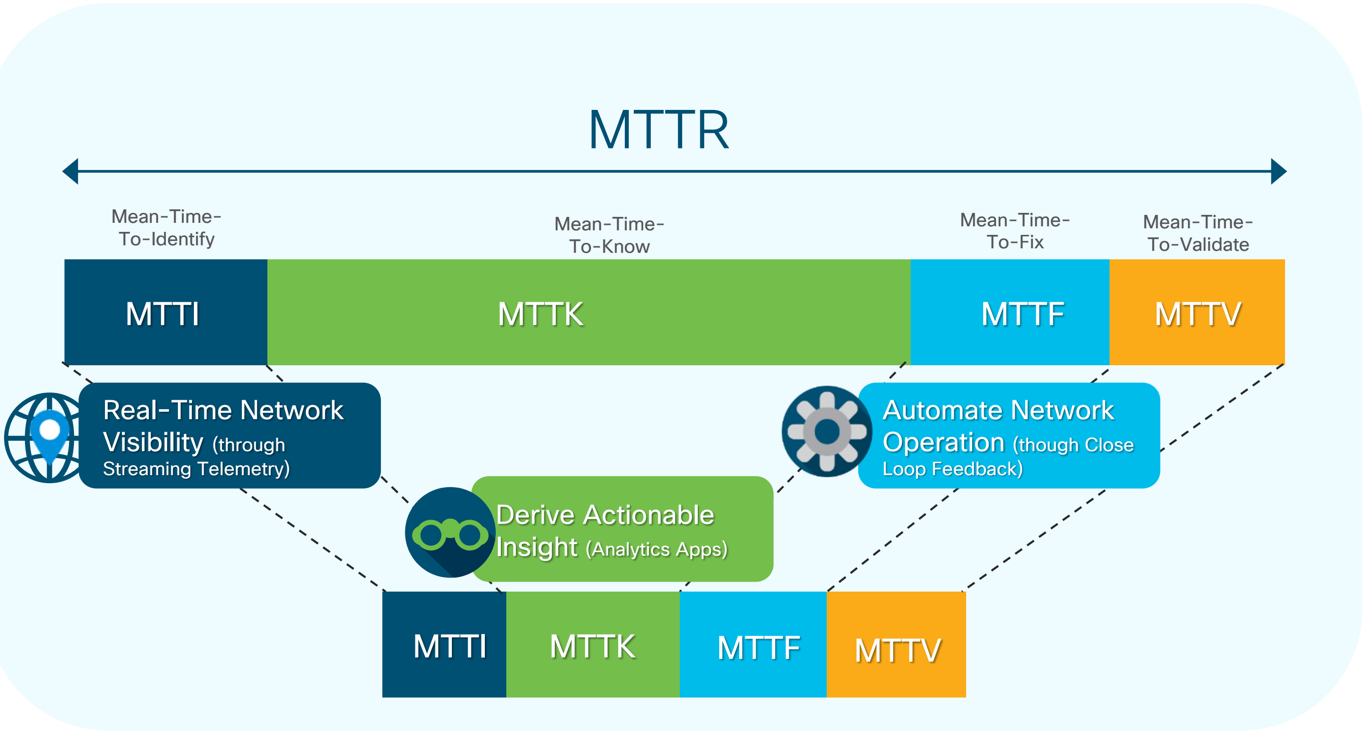
- Push New SR Policy /Recompute path
- Shutdown interface or Spin up new VM

Learn

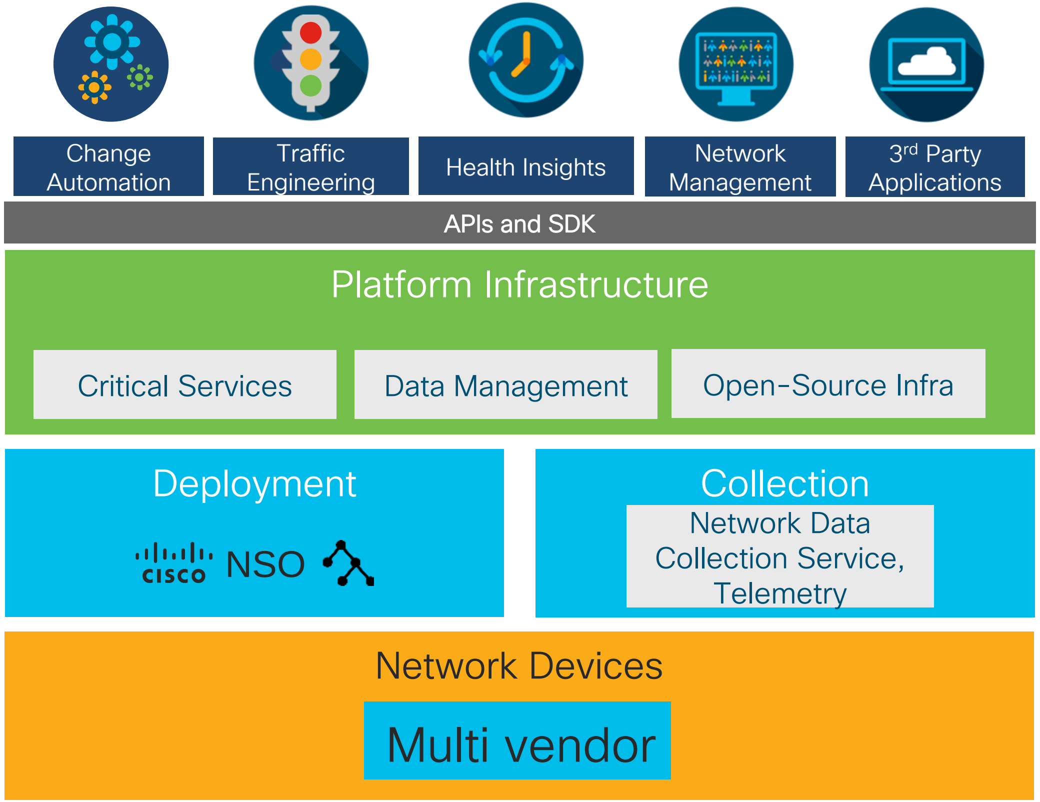
Update KPI's

Human or Machine Learning

Analytics and Automation enable MTTR Reduction



Building Blocks to Operational Automation

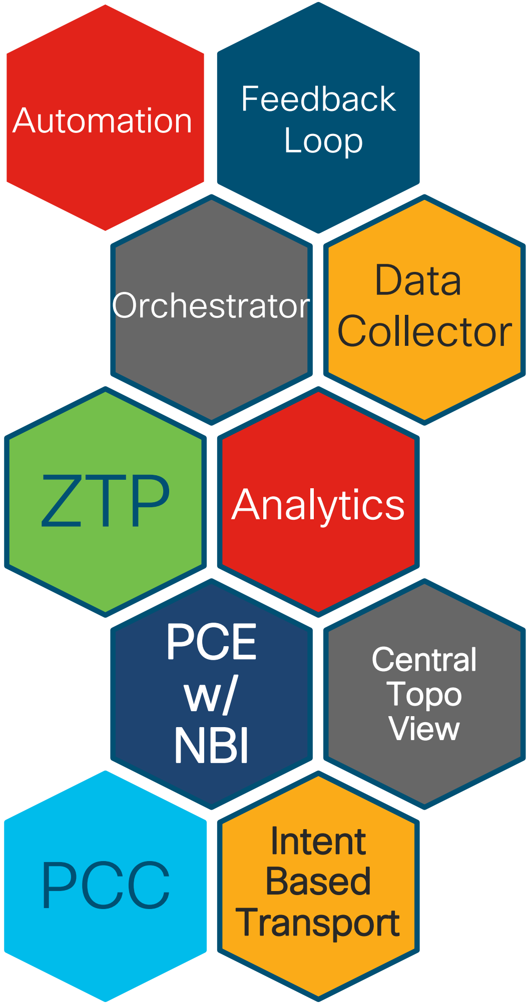


Cisco brings network expertise to provide solution integration and support

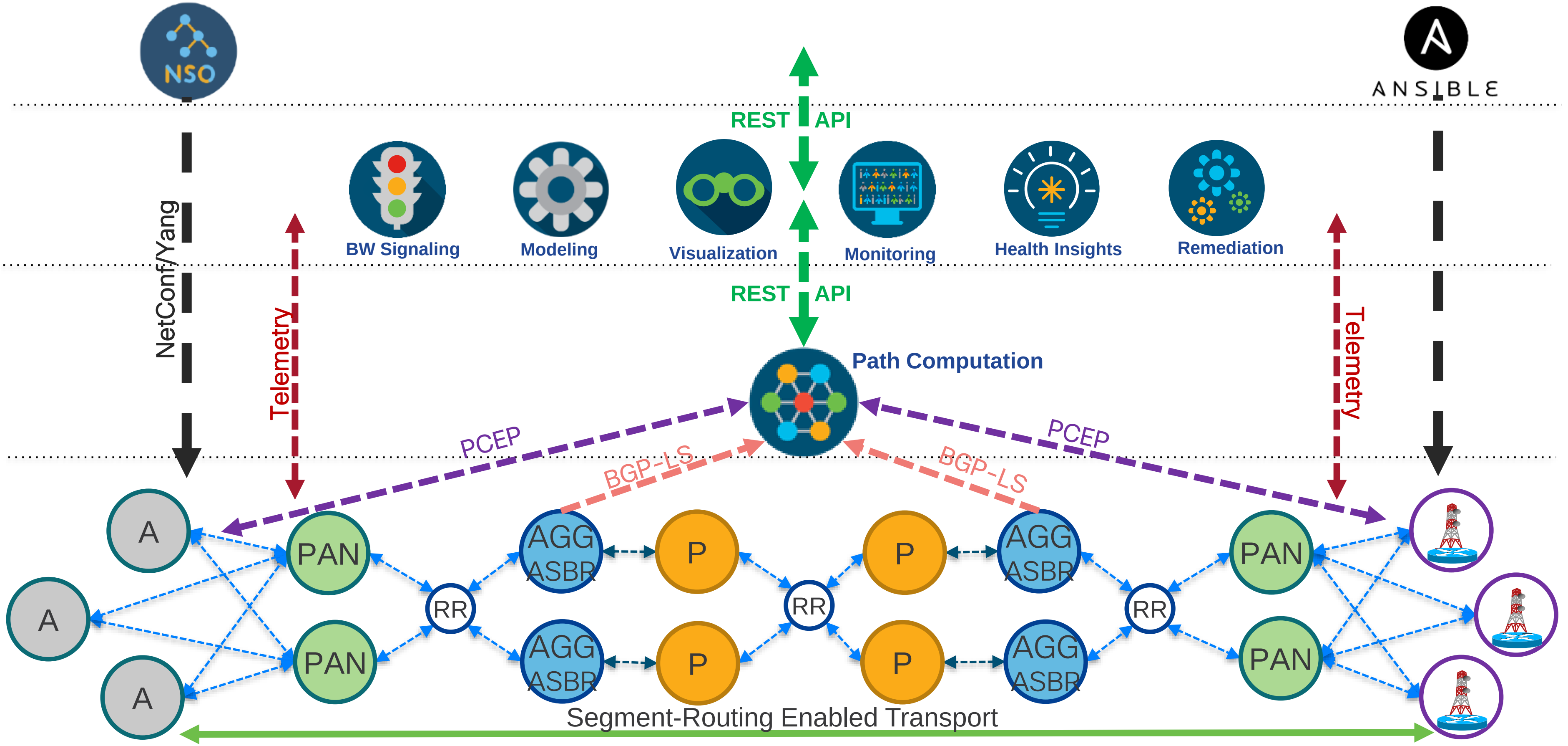
Tying it Together with Software Driven Usecases

A Recipe For Transport SDN

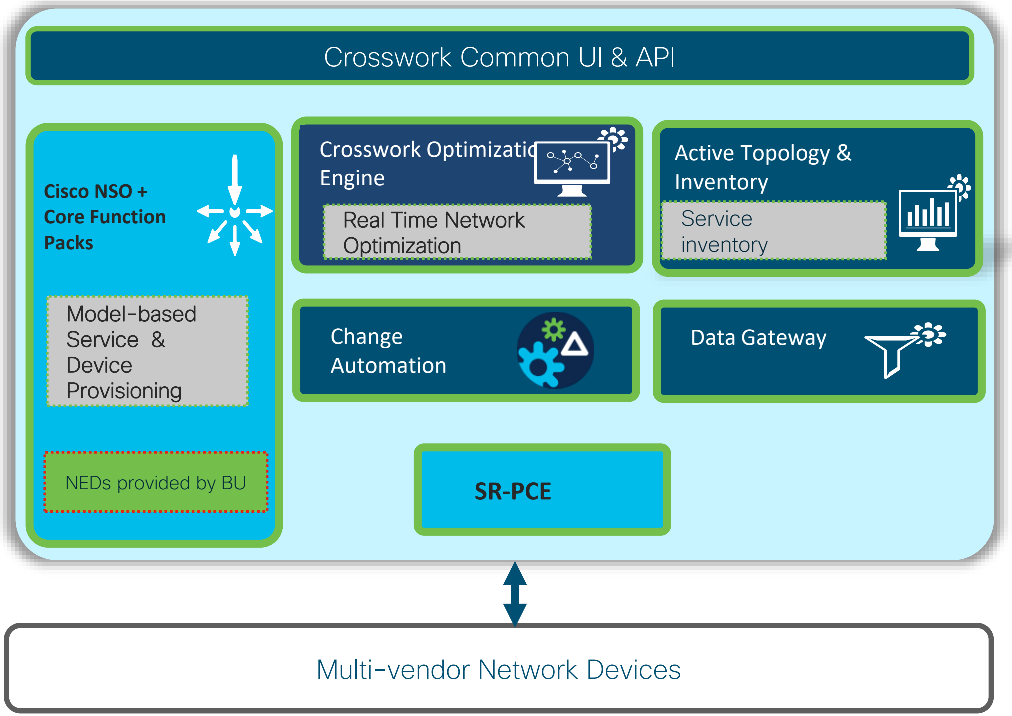
- Network Simplification and intent based transport paves the way
- Individual components for a “Transport SDN” architecture widely available
- Integration between various software components in key
- Applications interact with and actively drive Transport Network



Intent Based Software Defined Transport Network



Software Defined Transport Usecases

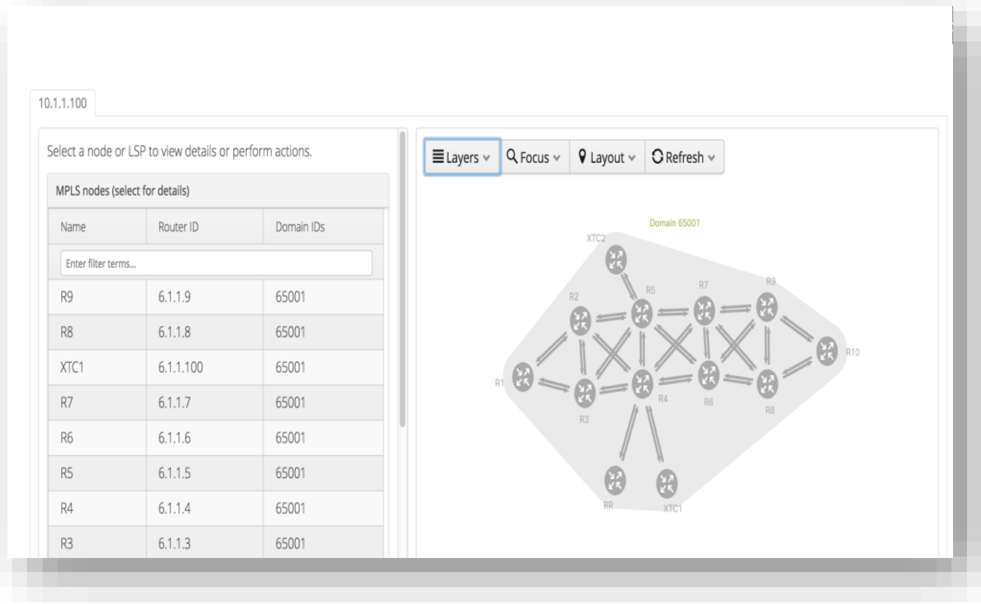


Use Case	Description
Service-Oriented Transport Provisioning	Provision segment routing traffic-engineering policies for services with SLAs.
Service Provisioning	Provision L2VPN & L3VPN services (sample)
Bandwidth Optimization	Tactically optimize the network during times of congestion
Real time network optimization	Collect real-time performance information and optimize the network as needed to maintain the SLA
Topology & Inventory	Collect and expose information on network and services
Closed Loop Automation	KPI/SLI based automation actions (open, configurable, multivendor)

Intent Based SDN Ready Transport Use Cases:

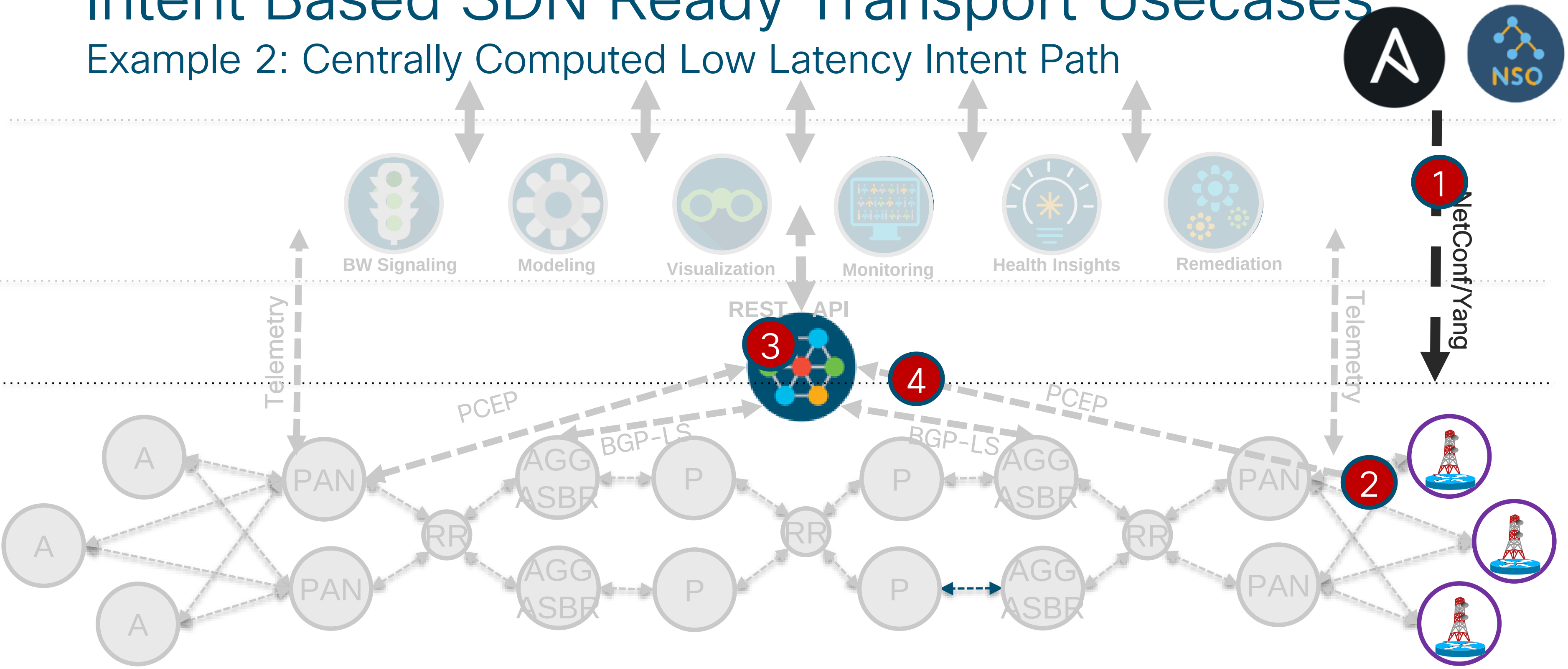
Example1: Centralized Control and Visualization for End-to-End Path

- SR-PCE enables REST API
- External Application gather Topology from SR-PCE
- Visualization includes:
  - Link/Node info
  - SID Allocation
  - Intent Based Path, if defined on nodes/PCE



Intent Based SDN Ready Transport Usecases:

Example 2: Centrally Computed Low Latency Intent Path



1. Provision Low Latency Service

2. Request LSP Computation

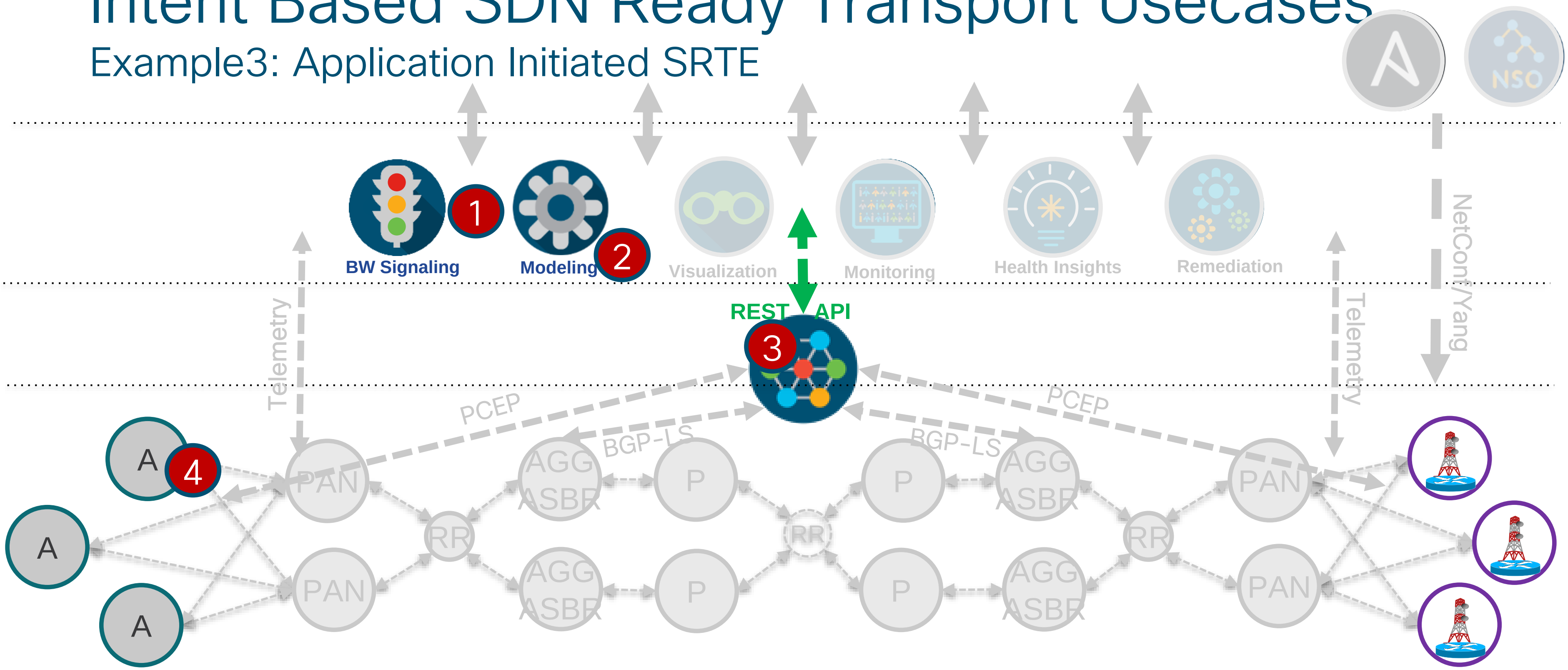
3. Perform Computation

4. Send Computed LSP

5. Applications Updated with new LSP

Intent Based SDN Ready Transport Usecases:

Example3: Application Initiated SRTE



1. Application Determines need for TE Path

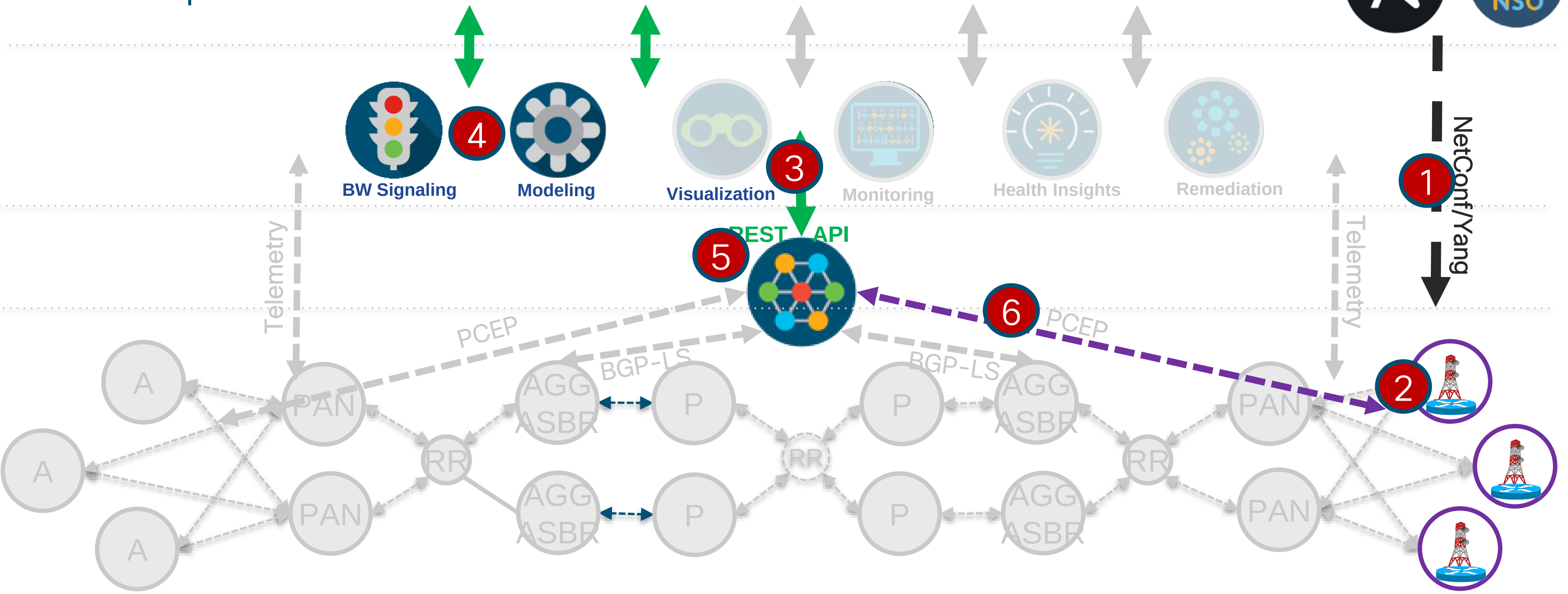
2. Application Computes new LSP

3. Application Instructs SR PCE of new LSP

4. Intent based LSP Path is instantiated on Headend

Intent Based SDN Ready Transport Usecases:

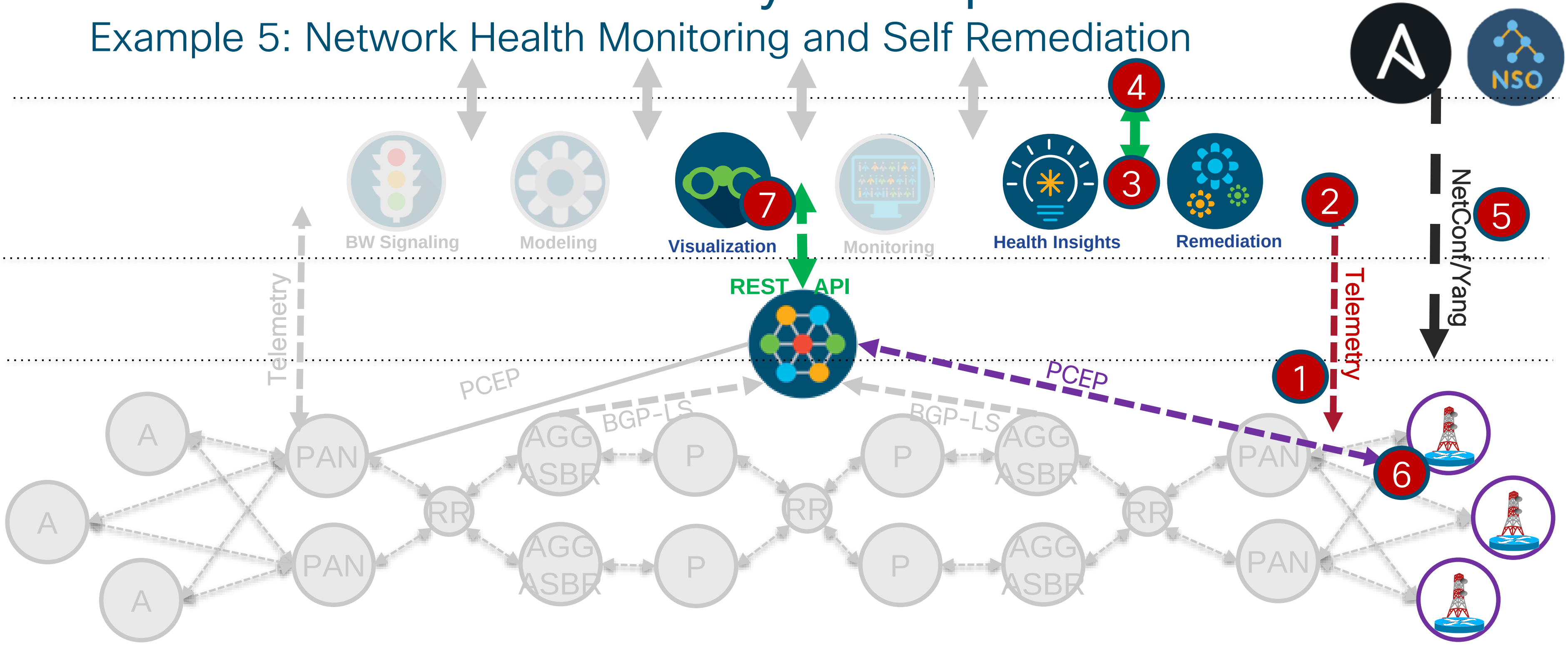
Example 4: Bandwidth Intent Service



1. New Service Requests BW	2. LSP Computation Requested	3. Computation Delegated to Application	4. LSP Path Computed	5. LSP Path sent to SR PCE	6. LSP Path with BW allocated sent to Headend

Intent Based SDN Ready Transport Usecases:

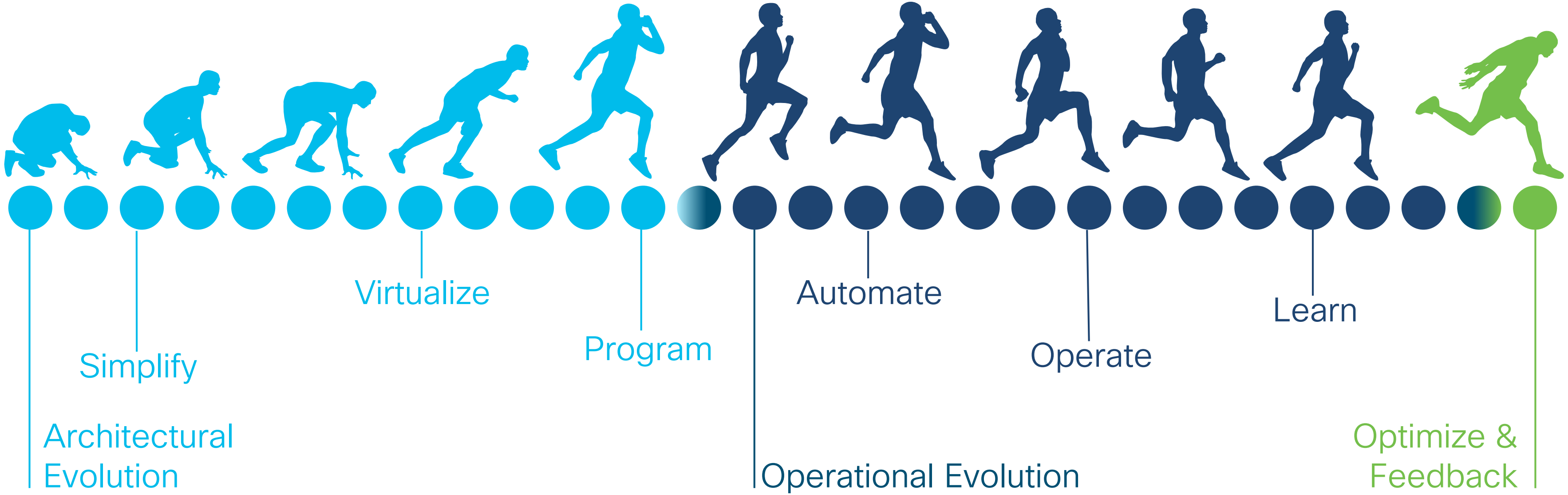
Example 5: Network Health Monitoring and Self Remediation



1. Network Telemetry Data (Model/Event)	2. Published on Data Bus for Subscribers	3. Applications Collect Data and Analyze	4. App Takes action, advices NSO	5. NSO makes n/w changes for continuity	6. PCC Requests new Path from PCE	7. PCE update Apps with new path

Summary

Start Your (Transformation) Journey



It all starts with Simplification .... at all layers of the Network

cisco Live!

"Give me the challenge to transform a legacy network into SDN ready and I will first simplify my transport network with Segment Routing, then use software to drive programmable transport

The Quasi-Experts at Cisco Live 2020

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

You make **possible**