Segment Routing – фундамент для построения SDN

Дмитрий Дементьев, системный инженер, SP Russia

ddementi@cisco.com

23 мая, 2017

О чем будем сегодня говорить

- Архитектура Segment Routing
  - Введение
  - Сценарии использования
  - Внедрение на сети оператора
- Обзор SR Traffic-Engineering и новых возможностей
- Введение в SRv6

MPLS – простой или сложный?

Простой Data Plane

Label/Label stack + 3 операции (push/pop/swap)

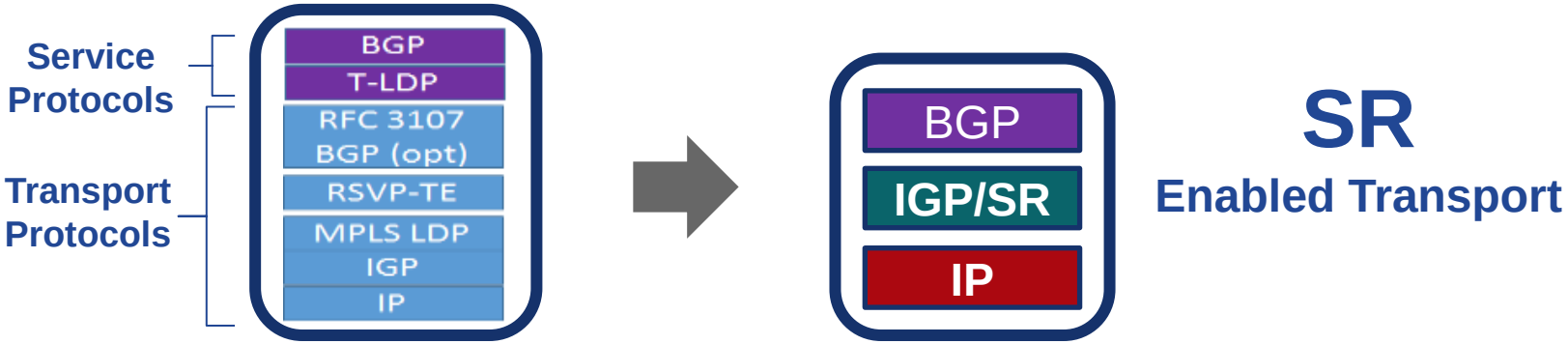
Сложный Control Plane

IGP + LDP + RSVP + Service Plane (LDP/BGP)

Требуется синхронизация протоколов
Легко сделать ошибку
Сложно отлаживать
Большая нагрузка на Control Plane

Что хочется сделать?

- Сохранить и использовать MPLS Data Plane
- Сохранить и использовать все MPLS-сервисы
  - VPLS, VPNv4/v6, VPLS, FRR, L2VPN и другие
- Создать более удобный Control Plane для forwarding
  - Меньше протоколов
  - Меньше настроек
  - Меньше нагрузки на CPU
- Сохранить возможность совместной работы с LDP и RSVP



Что такое Segment routing?

- **Source Routing** – возможность задать на источнике (Ingress PE) путь прохождения пакетов по сети, с помощью последовательности сегментов в заголовке самих пакетов

Сегмент = Инструкция (например, «доставить трафик до узла N кратчайшим путем»)

MPLS forwarding plane Сегмент = Label ← Начнем с простого

IPv6 forwarding plane Сегмент = routing extension header **(see 4.4 of RFC2460)**

The state is no longer in the network but in the packet

cisco

Segment Routing - стандартизация

Strong commitment for standardization and multi-vendor support

SPRING Working-Group (started Nov 2013)

All key documents are WG-status

Over 25 drafts maintained by SR team

- Over 50% are WG status
- Over 75% have a Cisco implementation

Several interop reports are available

First RFC document - RFC 7855 (May 2016)

Technology and Problem Statement

- Architecture (draft)
- Problem Statement
  - Generic (draft)
  - Resiliency (draft)
  - IPv6 (draft)
  - OAM (draft)
- Applicability
  - SR illustration to problem statement (draft)
  - Centralized Egress Peer Engineering (draft)

Protocol Extension

- ISIS extension for SR (draft)
- OSPF extension for SR (draft)
- OSPFv3 extension for SR (draft)
- BGP-LS extension for SR (draft)
- BGP-LS extension for SR EPE use-case (draft)
- PCEP extension for SR (SR ext, setup method)

FRR

- Topology-Independent LFA FRR with SR (draft)

MPLS instantiation of Segment Routing

- MPLS support for SR (draft)
- SR/LDP interaction and interworking (draft)

IPv6 instantiation of Segment Routing

- IPv6 SR routing extension header (draft)
- IPv6 use-cases (draft)

OAM

- SR/LSP Ping (draft)
- OAM (draft)

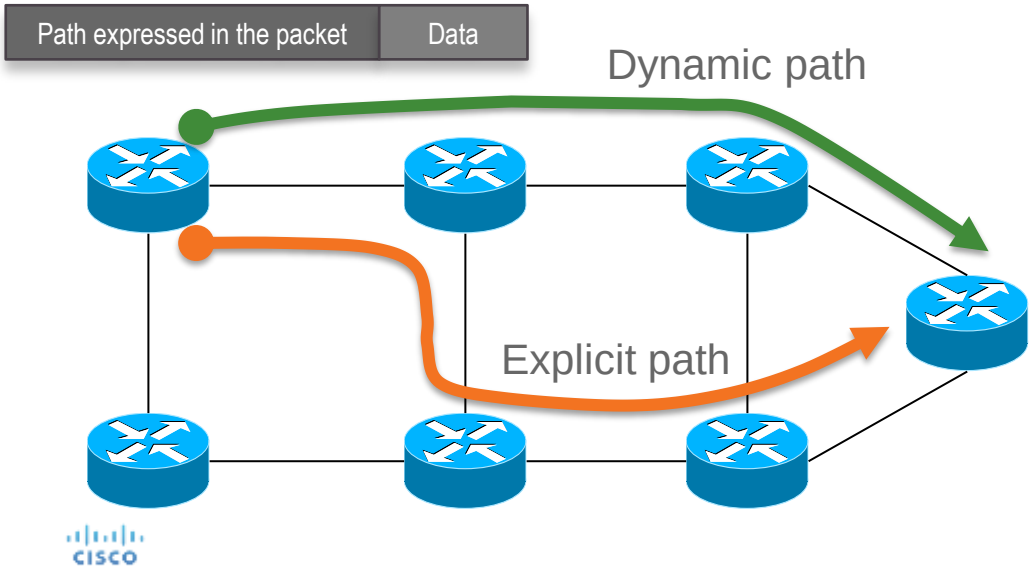
Segment Routing

Упрощая MPLS

Как это работает?

Нет LDP, нет RSVP-TE

Источник определяет маршрут и программирует его используя заголовок пакета с помощью сегментов, которые необходимо использовать (сегмент - MPLS метка или IPv6 адрес). Сеть передает пакет используя маршрут, закодированный с помощью сегментов.



Paths options

- Dynamic (Headend computation)
- Explicit (Operator / Controller)

Control Plane

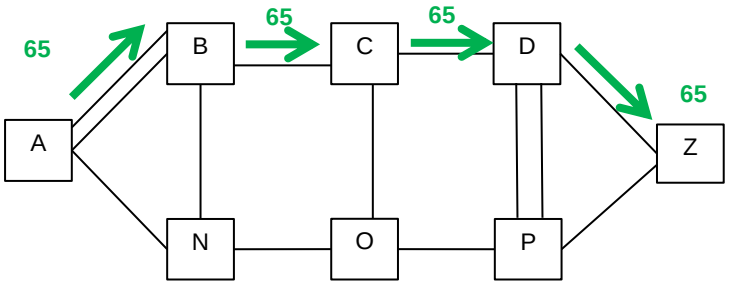
- Routing protocols with extensions (IS-IS,OSPF, BGP)
- SDN controller

Data Plane

- MPLS (segment labels)
- IPv6 (+SR header)

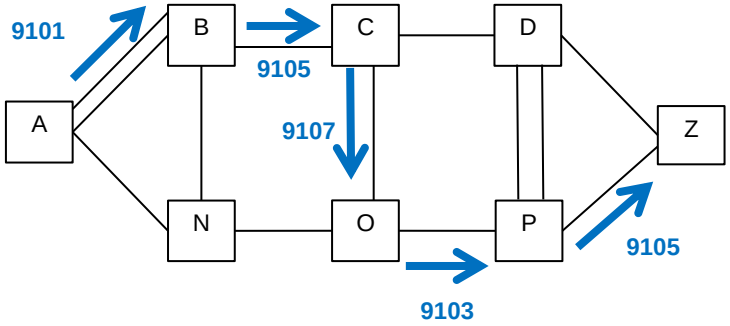
Идентификаторы IGP Segment

Prefix/Node SID



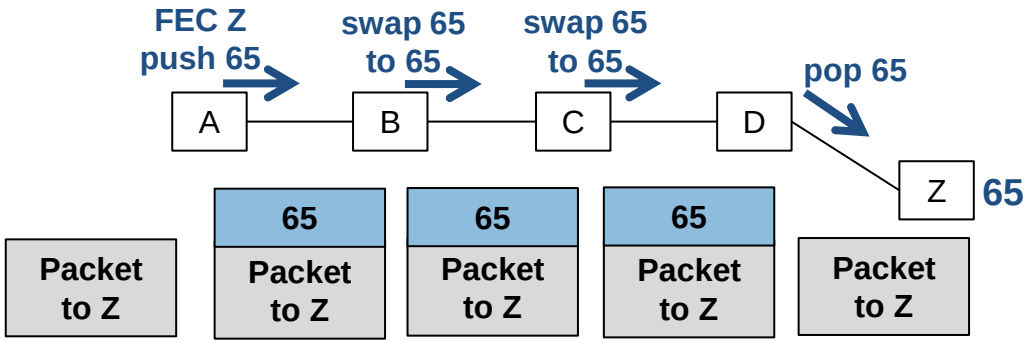
- Имеет глобальное значение внутри домена SR
- Передача трафика с использованием shortest-path tree
- Может быть задан как абсолютная величина, так и как индекс
- Используется зарезервированный блок меток (SR Global Block или SRGB)

Adjacency SID



- Имеет локальное значение внутри node
- Передача трафика на основе adjacency
- Задается как абсолютная величина

Node Segment (пример)

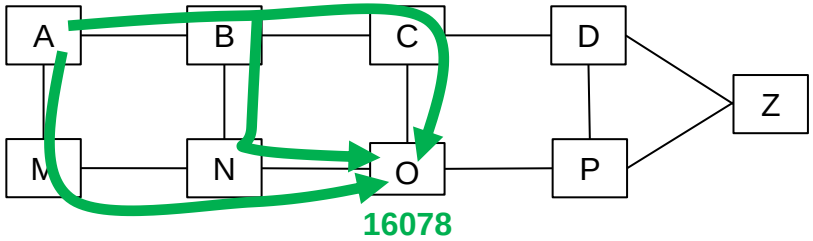


Пакет с label 65 передается к узлу “Z” по кратчайшему пути

- Узел Z анонсирует node-SID 65
  - IGP sub-TLV extension
- **Все узлы** инсталлируют node-SID в MPLS Data Plane

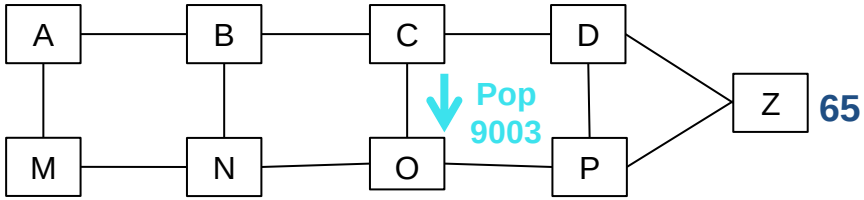
cisco

Автоматическая балансировка трафика в случае Node/Prefix SID



- ECMP
  - Трафик предназначенный для узла O и имеющий SID 16078 автоматически разбалансируется по всем доступным ECMP путям
  - Не требует дополнительных настроек

Adjacency Segment (пример)



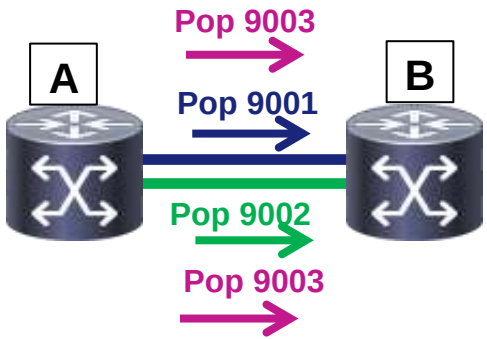
На узле “C” пакет с меткой 9003 должен быть передан по каналу “C-O”

- Узел “C” анонсирует adj-SID по IGP
- **Только узел “C”** инсталлирует adj-SID в MPLS Data Plane

cisco

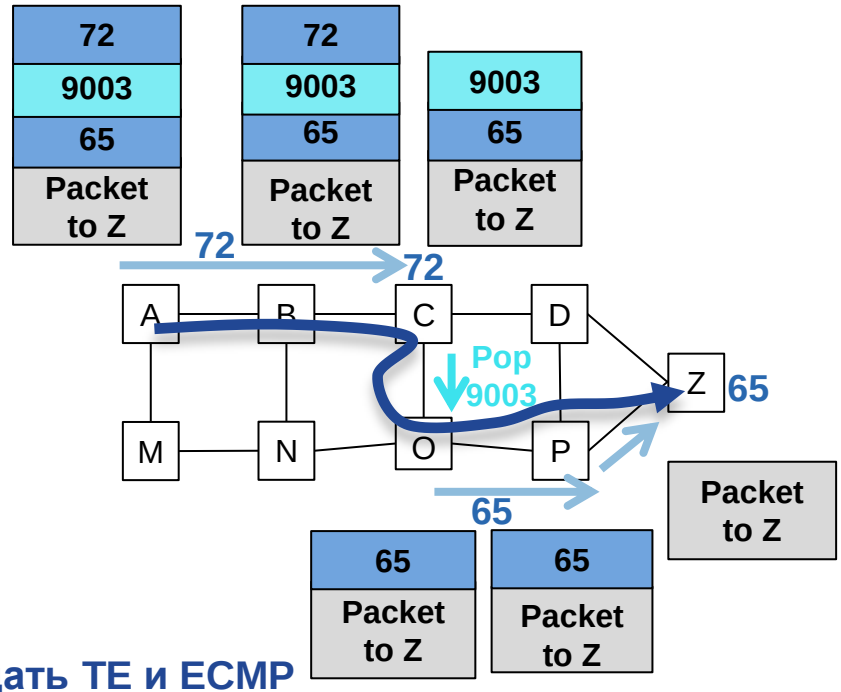
Балансировка трафика для Adjacency Segment

(Anycast Adjacency segment)



- 9001 передать через 1ый интерфейс
- 9002 передать через 2ой интерфейс
- 9003 балансировать по группе интерфейсов

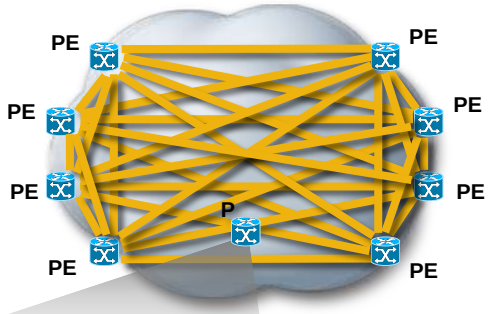
Комбинируем сегменты



- Возможность совмещать TE и ECMP
- Всего один протокол (IGP + extension)

Таблица LFIB с Segment Routing

- LFIB наполняется при помощи IGP (ISIS / OSPF)
- Таблица коммутации постоянна (Nodes + Adjacencies) вне зависимости от числа возможных путей
- Другие протоколы (LDP, RSVP, BGP) по прежнему могут программировать LFIB



In Label	Out Label	Out Interface
L1	L1	Intf1
L2	L2	Intf1
…	…	…
L8	L8	Intf4
L9	Pop	Intf2
L10	Pop	Intf2
…	…	…
Ln	Pop	Intf5

идентификаторы Node Segment

идентификаторы Adjacency Segment

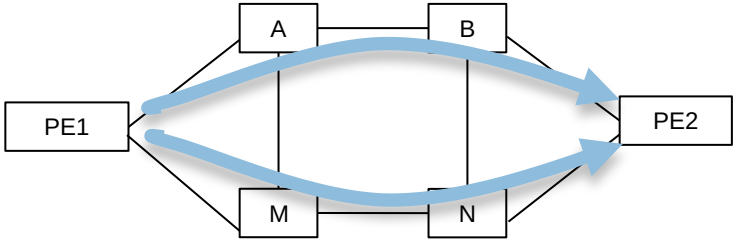
Таблица коммутации остается неизменной

draft-previdi-isis-segment-routing-extensions
draft-psenak-ospf-segment-routing-extensions

cisco

Segment Routing и сервисы MPLS

- **Эффективное использование преимущества пакетных сетей (ecmp-aware shortest-path) :**
  - ❑ **node segment!**



- **Упрощение работы сети:**
  - ❑ **одним работающим протоколом меньше**
  - ❑ **отсутствует LDP/ISIS синхронизация**

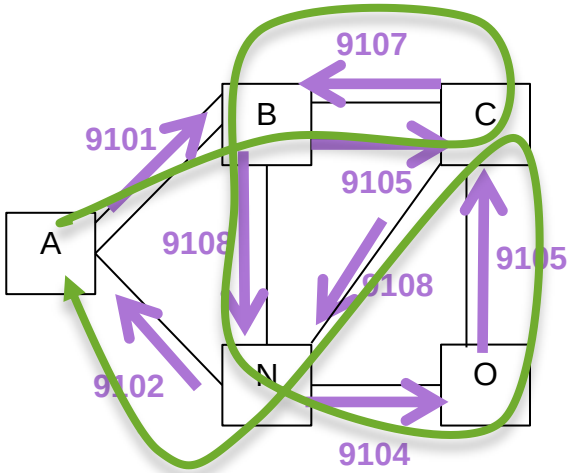
Все сущестующие VPN сервисы возможно реализовать поверх node segment для PE2

cisco

Сценарии применения технологии

MPLS dataplane monitoring

9101
9105
9107
9108
9104
9105
9108
9102
OAM



draft-geib-spring-oam-usecase-06

Google

Localizing packet loss

In a large complex network

Nicolas Guilbaud nguilbaud@google.com
Ross Cartlidge rossc@google.com

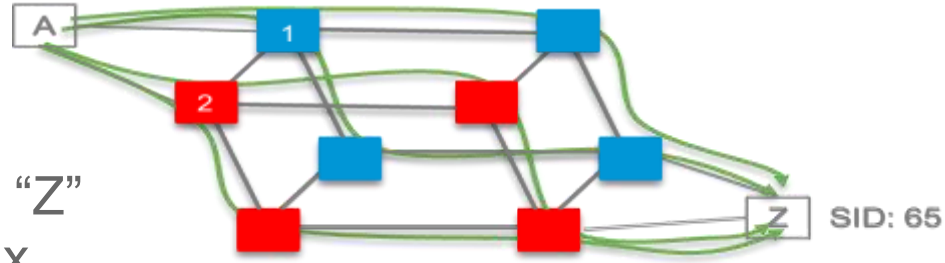
Nanog57, Feb 2013

cisco

Anycast segment для Dual Core

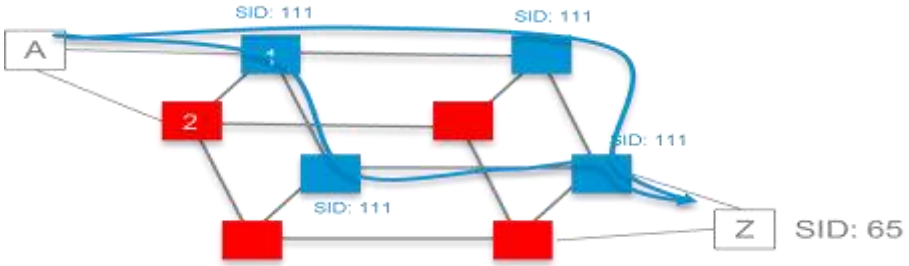
- Используем Node-SID [65]

Передаем трафик до узла “Z” используя ECMP в обеих плоскостях



- Используем Anycast-SID + Node-SID [111, 65]

Передаем трафик до узла “Z” используя ECMP только в одной плоскости



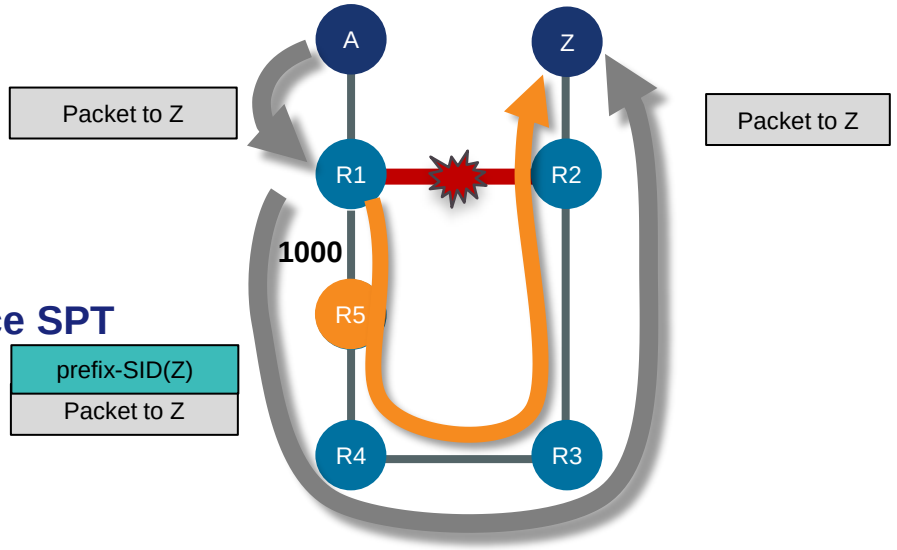
ECMP-awareness!

Topology Independent LFA FRR

Пример TI-LFA – Zero-Segment

Segment Routing позволяет гарантировать LFA FRR в любой топологии:

- **TI-LFA защита для линка R1R2 на R1**
- **Расчитаем LFA(s)**
- **Расчитаем post-convergence SPT**
- **Определим LFA узел для post-convergence SPT**



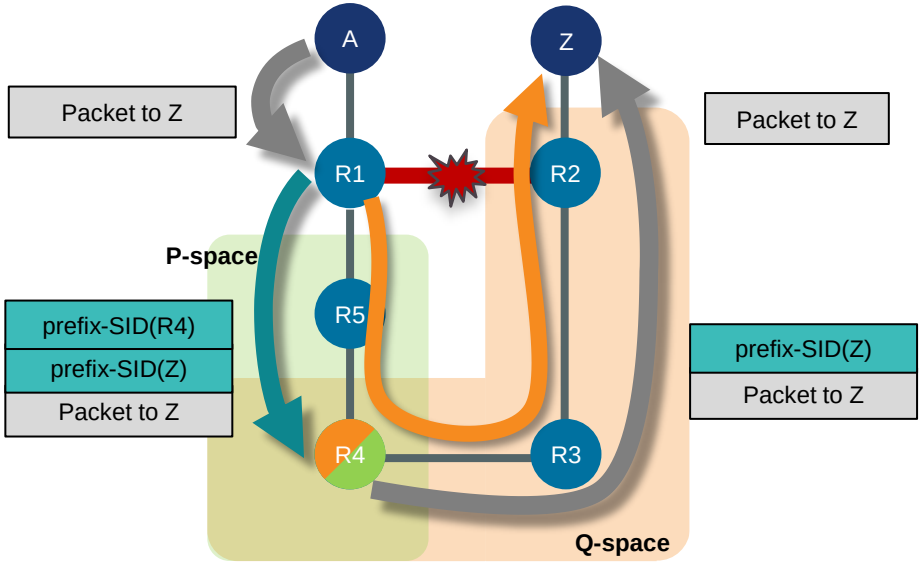
Default metric:10

Topology Independent LFA FRR

Пример TI-LFA – Single-Segment

Segment Routing позволяет гарантировать LFA FRR в любой топологии:

- TI-LFA защита для линка R1R2 на R1
- Расчитаем P и Q области
  - Они пересекаются в нашем случае
- Расчитаем post-convergence SPT
- Определим PQ узел для post-convergence SPT
- R1 добавит prefix-SID R4 для создания backup path



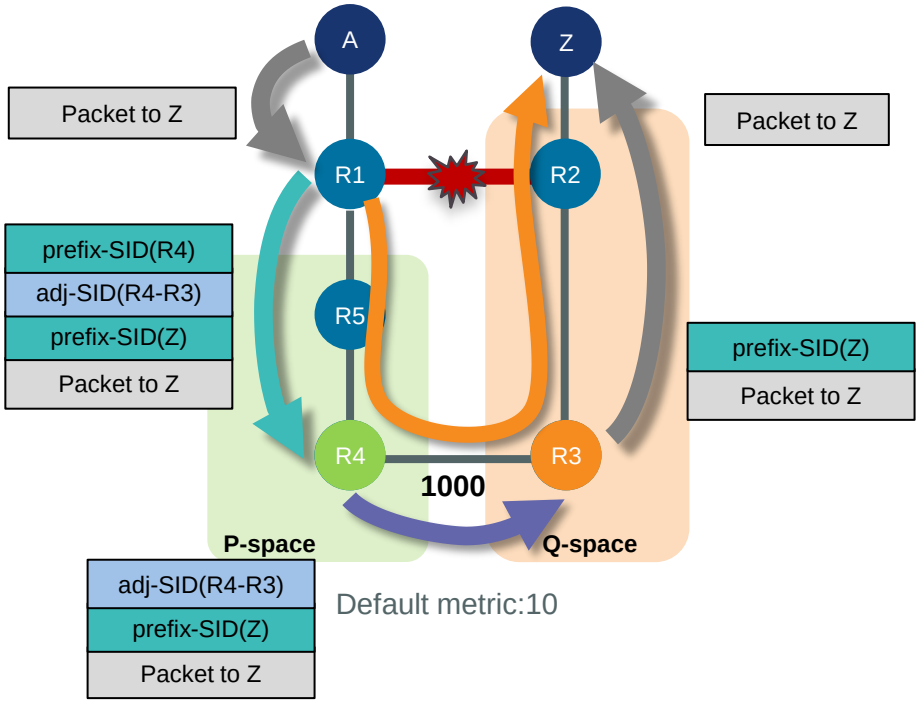
Default metric:10

Topology Independent LFA FRR

Пример TI-LFA – Double-Segment

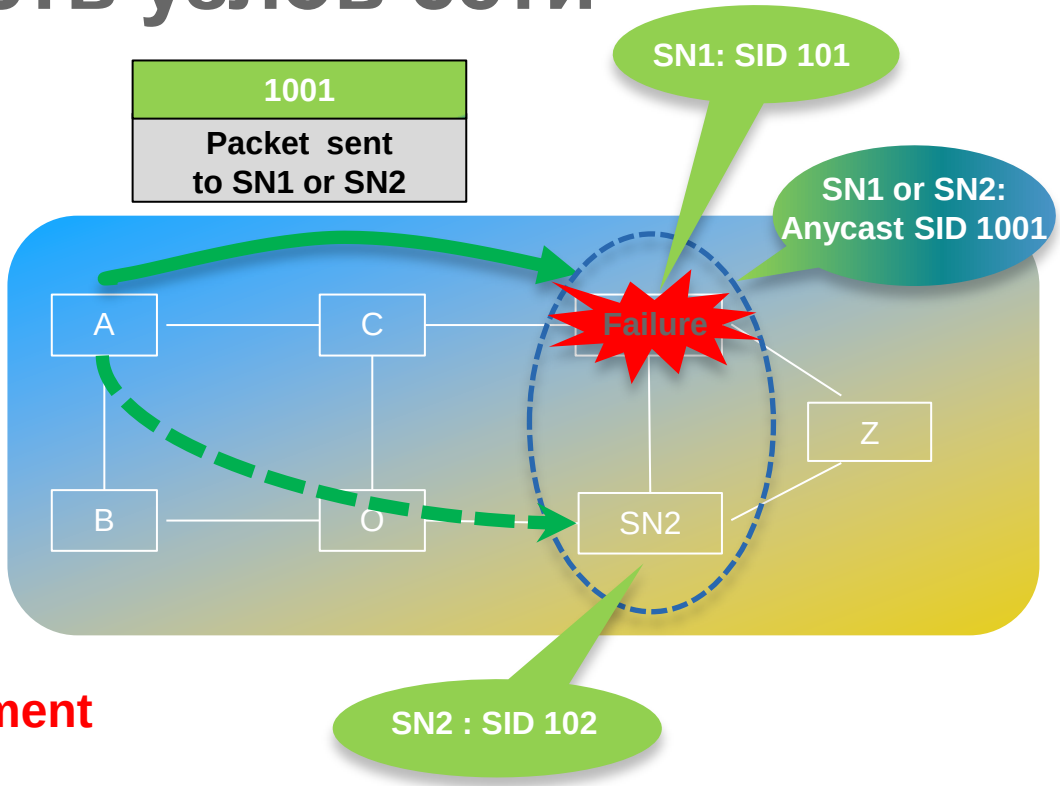
Segment Routing позволяет гарантировать LFA FRR в любой топологии:

- TI-LFA защита для линка R1R2 на R1
- Расчитаем P и Q области
- Расчитаем post-convergence SPT
- Определим Q узел и соседний с ним P узел для post-convergence SPT
- R1 добавит prefix-SID R4 и adj-SID R3-R4 линка для создания backup path



Отказоустойчивость узлов сети

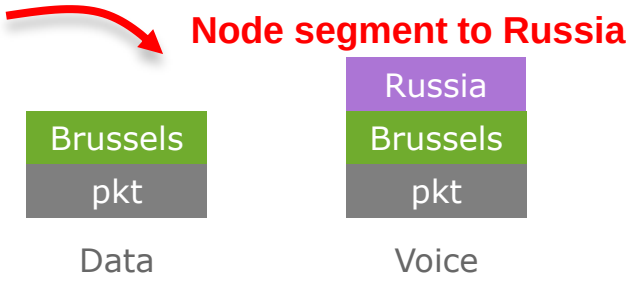
- Несколько устройств анонсируют одинаковый Segment Identifier (anycast segment) в дополнении к их SID
- Трафик передается к ближайшему устройству на основе IGP best path
- Если основное устройство сломается, то трафик перенаправится к другому устройству с тем же anycast segment



- Эффективный механизм отказоустойчивости на транспортном уровне, не требует дополнительных технологий

Latency TE Service

- «Данные» из Токио в Брюссель
  - IGP shortest-path через США, широкие и дешевые каналы
  - PrefixSID of Brussels
- «Голос» из Токио в Брюссель
  - SRTE политика добавляет один дополнительный сегмент “Russia Anycast”
  - Low-latency path
- Приемущества
  - ECMP
  - Отказоустойчивость при использовании anycast segment
  - Отсутствие hop-by-hop сигнализации load и delay
  - Отказ от midpoint state



Применение Segment Routing в SDN

Нет per-tunnel state на mid-point → можно перейти к tunnel per-application
ECMP + Explicit routing → позволяет уменьшить количество TE tunnels
Не нужно программировать mid-point → проще контролировать сеть

Сеть проста, программируема и способна реагировать на возникающие события

Еще один важный слайд

MPLS-метки лишены собственной семантики

Сегмент может выражать **любую** инструкцию

- **Service**
- **Context**
- **IGP-based forwarding construct**
- **BGP-based forwarding construct**
- **Locator**

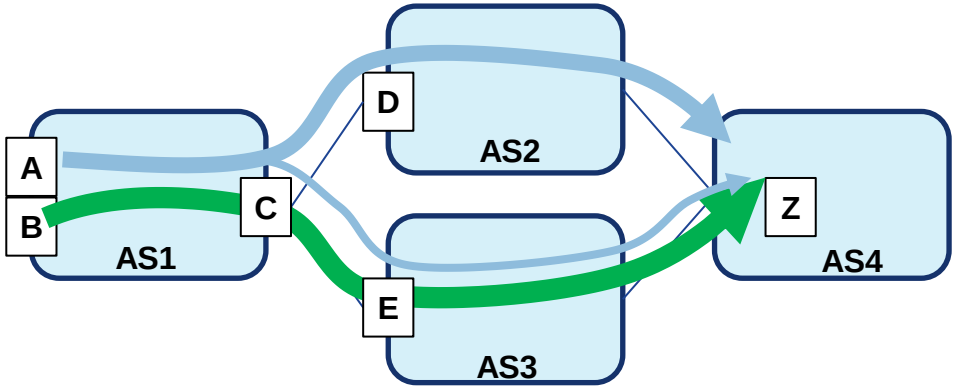
SR и управление внешней связностью

(Egress peering engineering)

http://tools.ietf.org/html/draft-filsfils-spring-segment-routing-central-epe-01

Cisco, Facebook, Yandex

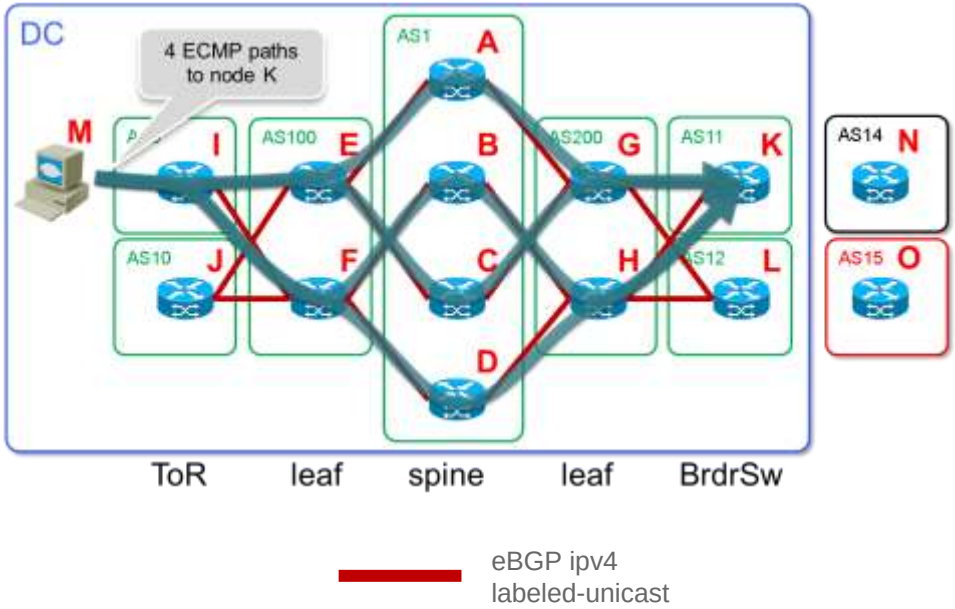
Определяем точку выхода из AS на ingress router



- PeerNode SID - Передать пакет через заданный пир
- PeerAdj SID - Передать пакет через заданный интерфейс
- PeerSet SID - Балансировать трафик по группе пиров

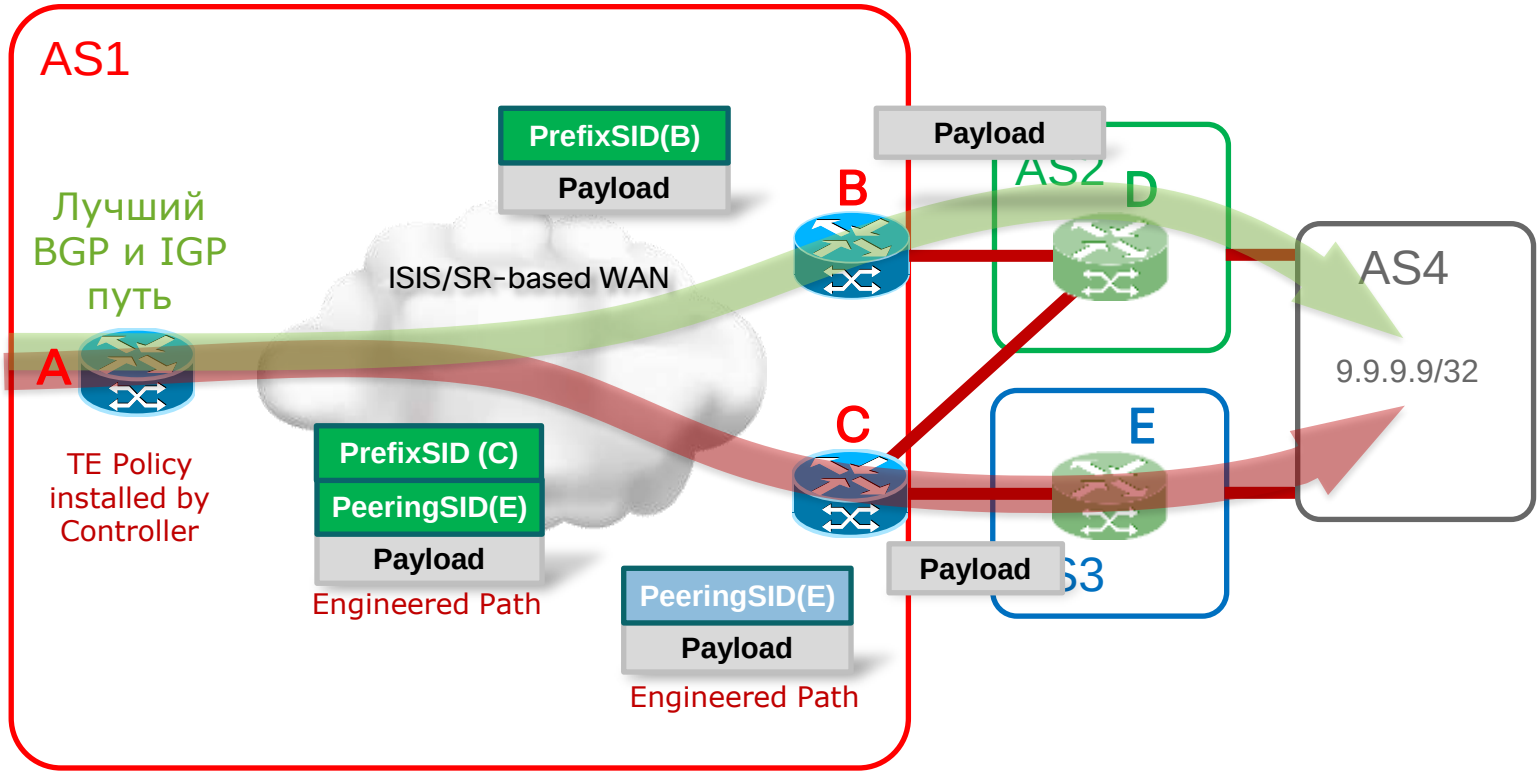
BGP Prefix SID - SR-based MSDC

- MPLS dataplane
- BGP control-plane (нет LDP, нет RSVP)
- Аналогичные преимущества IGP Prefix SID
  - ECMP
  - Automated FRR (BGP PIC)
    - отсутствие ручной конфигурации
  - Необходимые элементы для Traffic Engineering
  - Единый SRGB на каждом коммутаторе

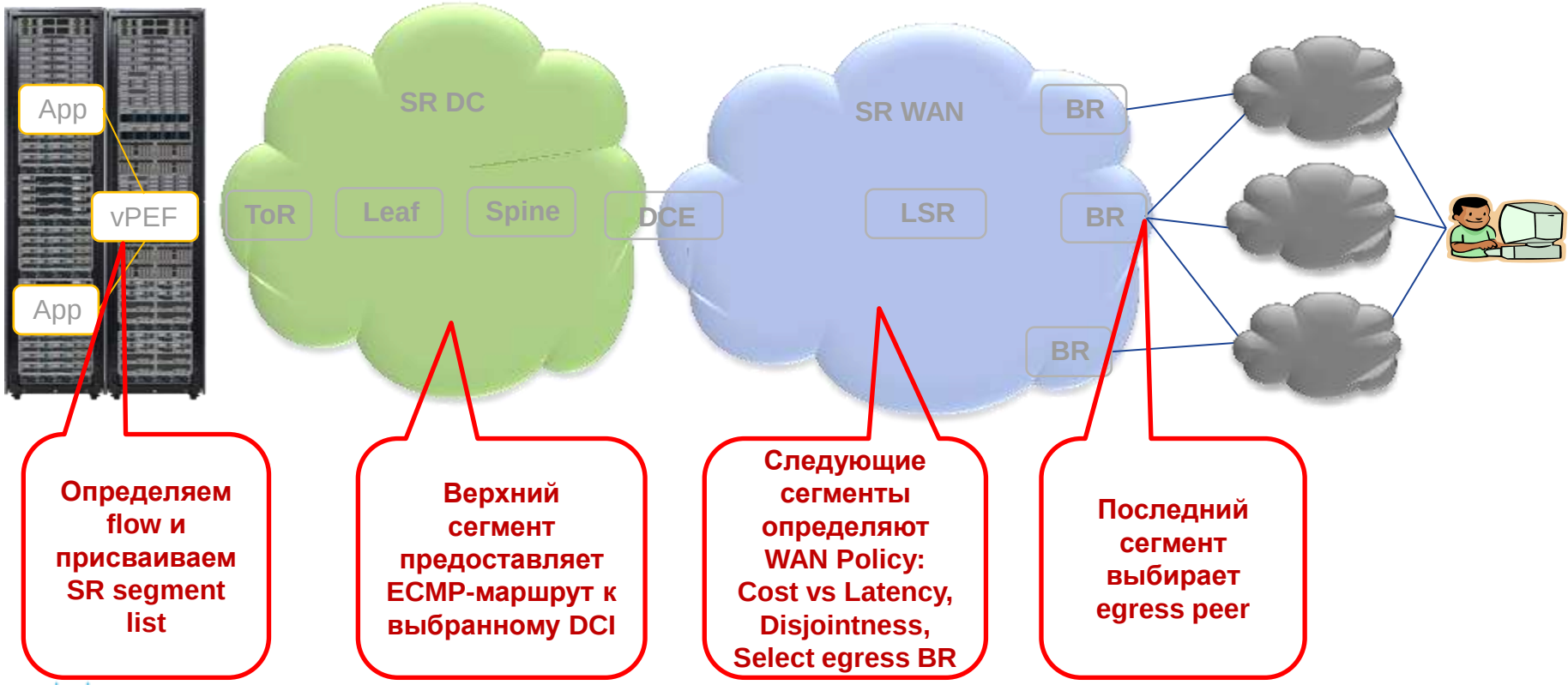


https://www.nanog.org/meetings/nanog55/presentations/Monday/Lapukhov.pdf
https://www.nanog.org/sites/default/files/wed.general.brainslug.lapukhov.20.pdf

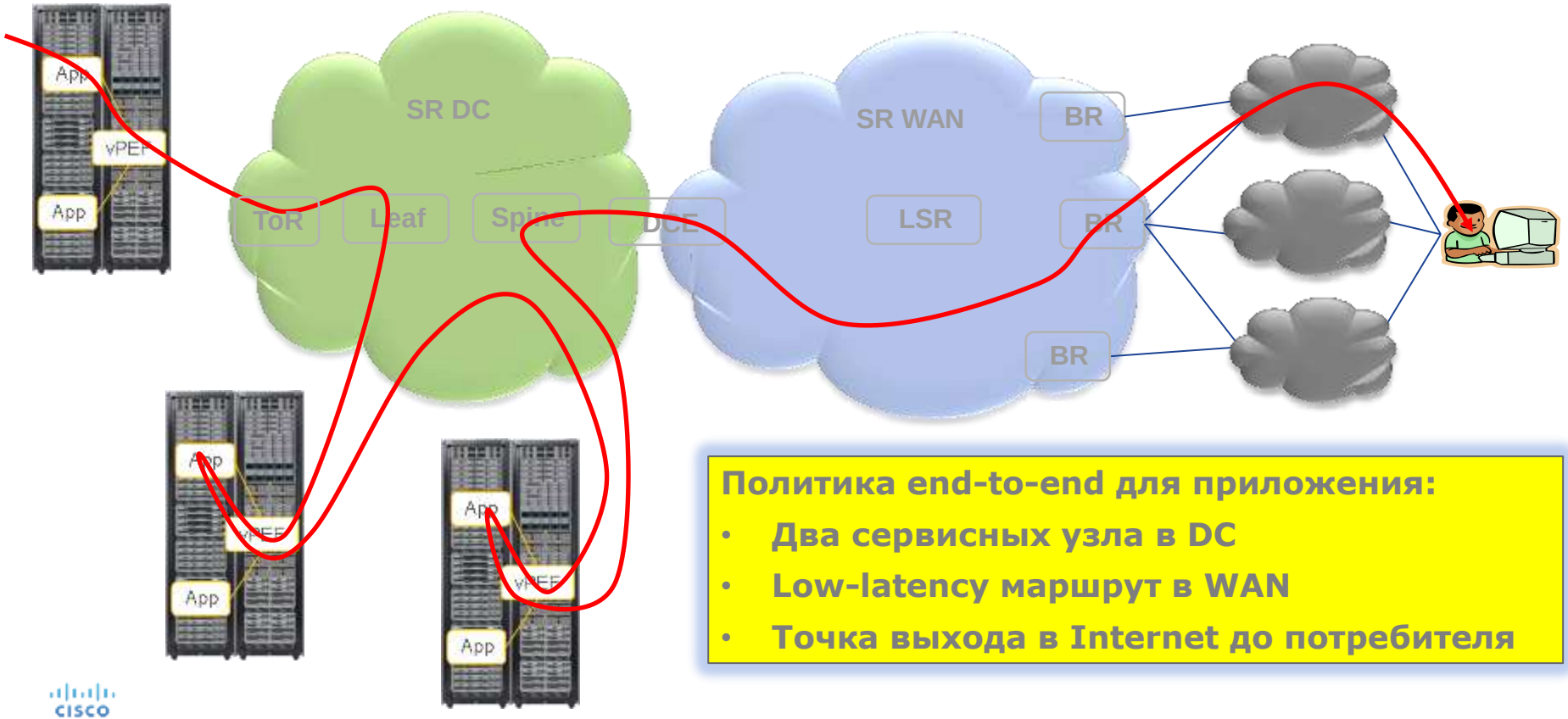
SR и управление внешней связностью AS до внешнего пира



Единая политика передачи трафика



Единая политика передачи трафика



Политика end-to-end для приложения:

- **Два сервисных узла в DC**
- **Low-latency маршрут в WAN**
- **Точка выхода в Internet до потребителя**

Критика Segment Routing

Проблема: HW ограничения глубины стека

В большинстве случаев для TE достаточно 2-3 сегмента

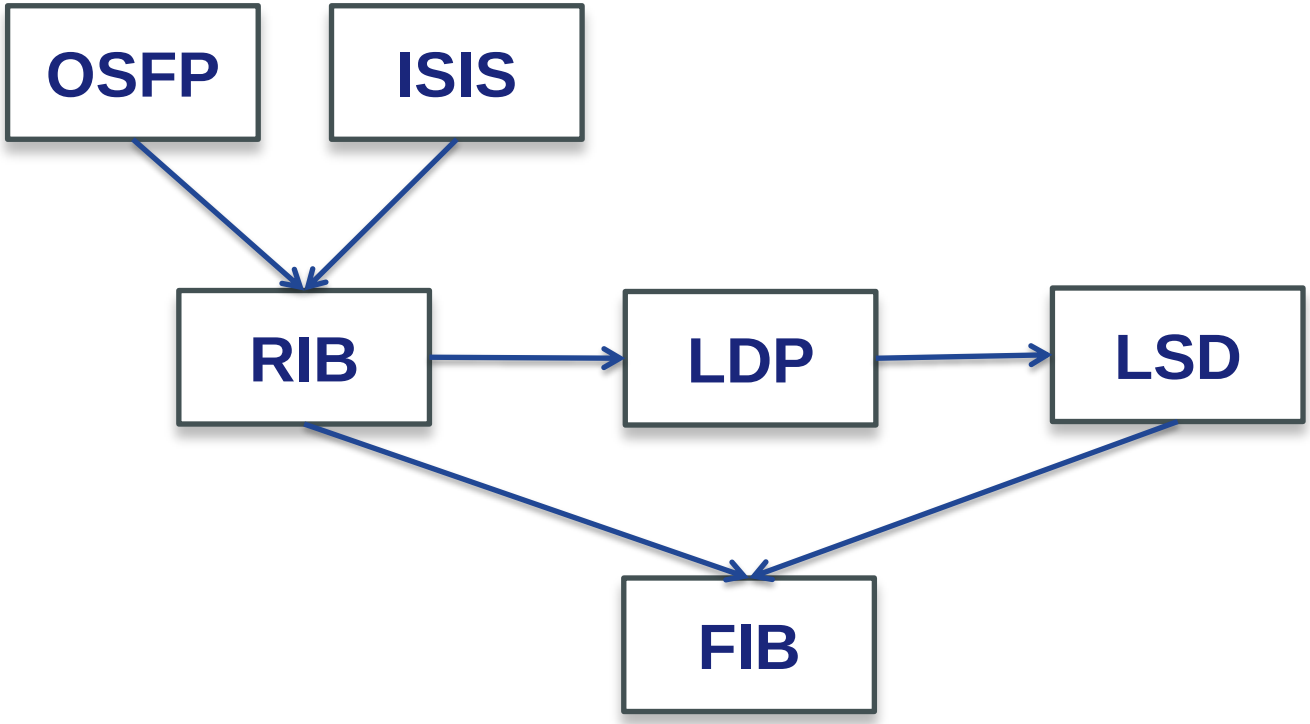
Для NG NPU глубина стека >10 лейблов

Проблема: Segment routing TE не учитывает ресурсы

Это так, но для этого есть контроллер

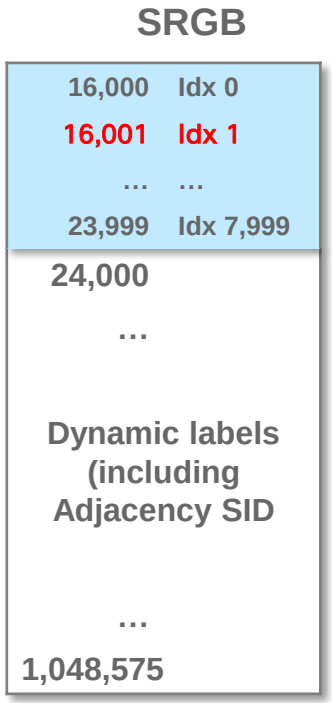
Внедрение SR в сети оператора связи

Как программируются метки на оборудовании?

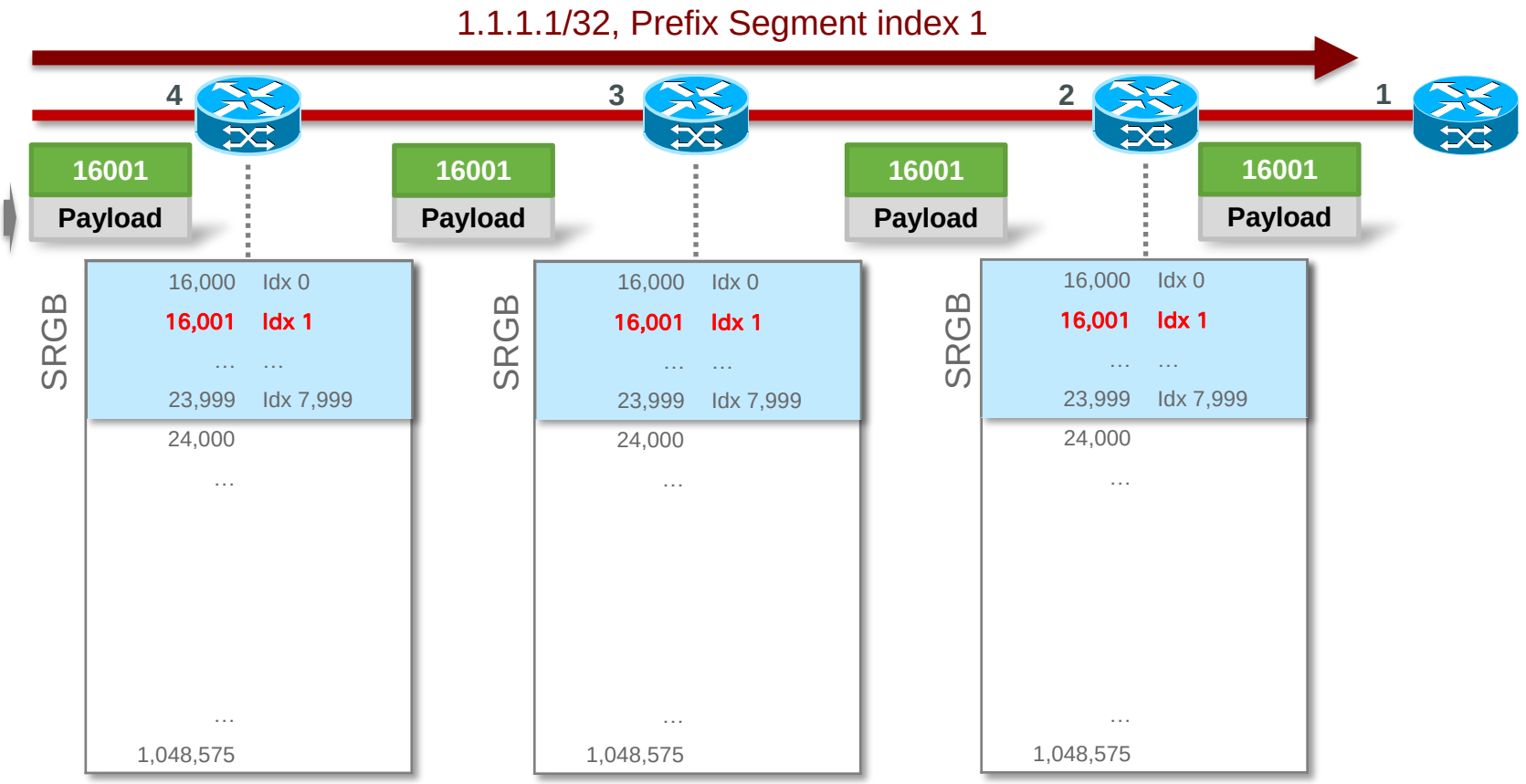


Segment Routing Global Block (SRGB)

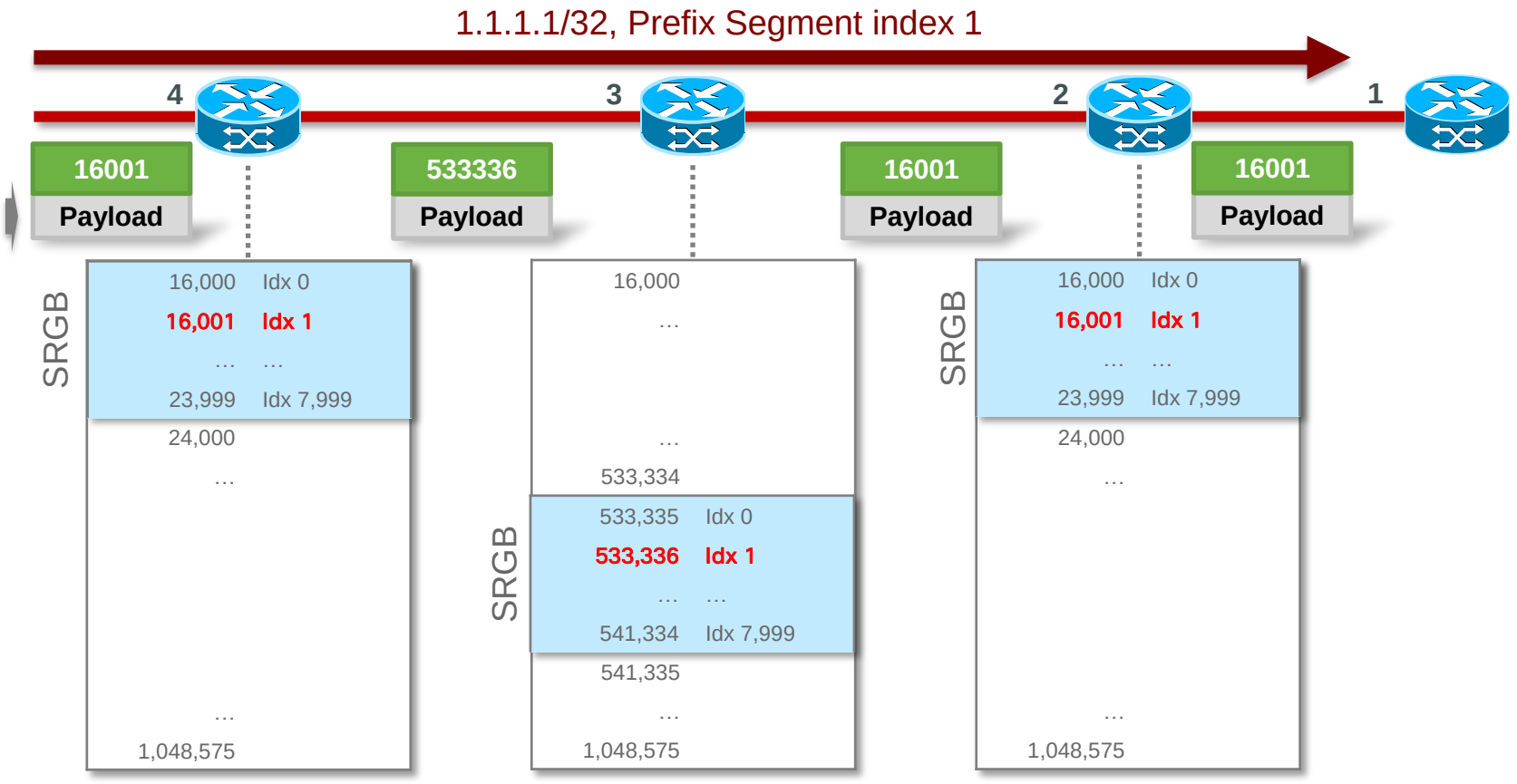
- Используйте **единый SRGB на всех устройствах**
  - Быстрое и простое внедрение технлогии
  - Global Segment == Global Label value
  - Использование разных SRGBs возможно, но сопряжено со сложностями настройки и interoperability
- Нестандартный SRGB может быть назначен из диапазона от 16,000 до 1,048,575



Рекомендованная модель применения SRGB



Возможный сценарий с разными SRGB



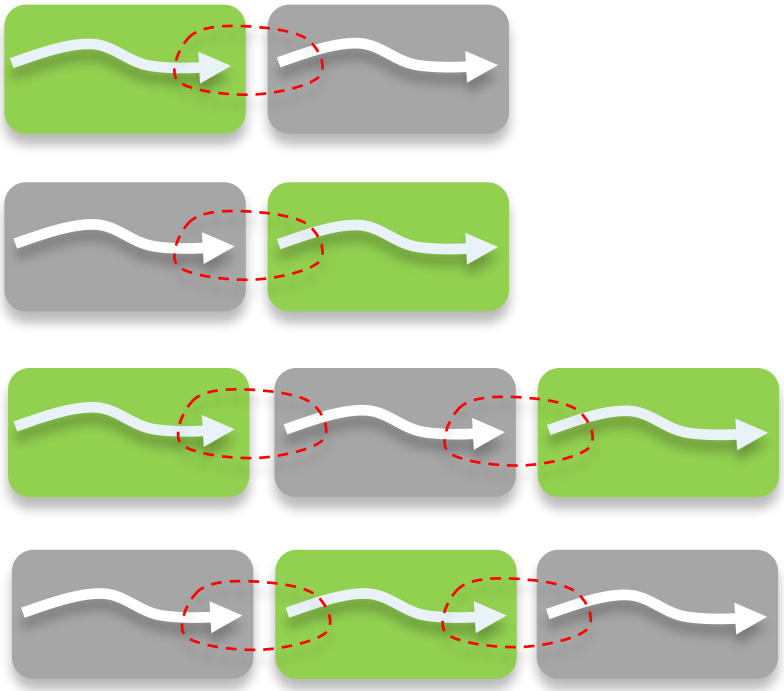
Модели взаимодействия SR и LDP

SR to LDP

LDP to SR

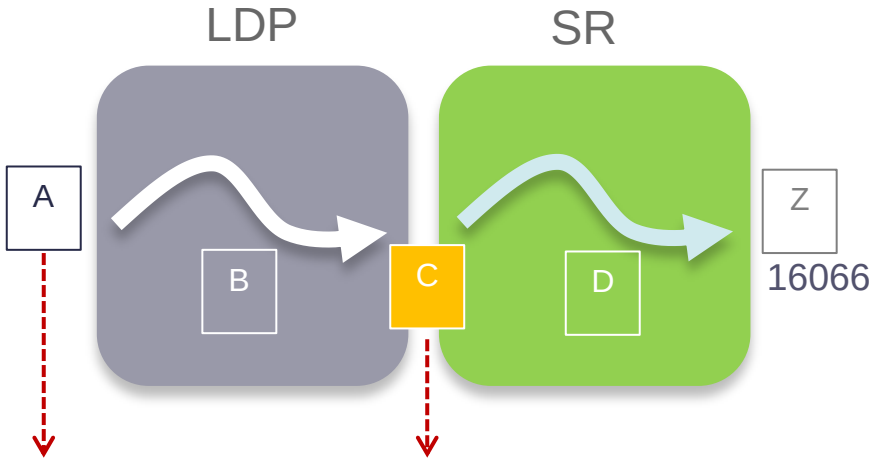
SR over LDP

LDP over SR



От LDP к SR – простой путь

- **Узел A хочет передать трафик на узел Z, но узел Z и часть промежуточных узлов не является LDP capable.**
  - **отсутствует LDP outgoing label**
- **В этом случае LDP LSP подключается к prefix segment LSP**
- **Узел C инсталирует следущую запись LDP-to-SR FIB:**
  - **incoming label: label назначенный LDP для FEC Z**
  - **outgoing label: prefix segment для достижения узла Z**
  - **outgoing interface: интерфейс в сторону узла D**



Prefix	Out Label (LDP), Interface
Z	16, 0

Input Label (LDP)	Out Label (SID), Interface
32	16066, 1

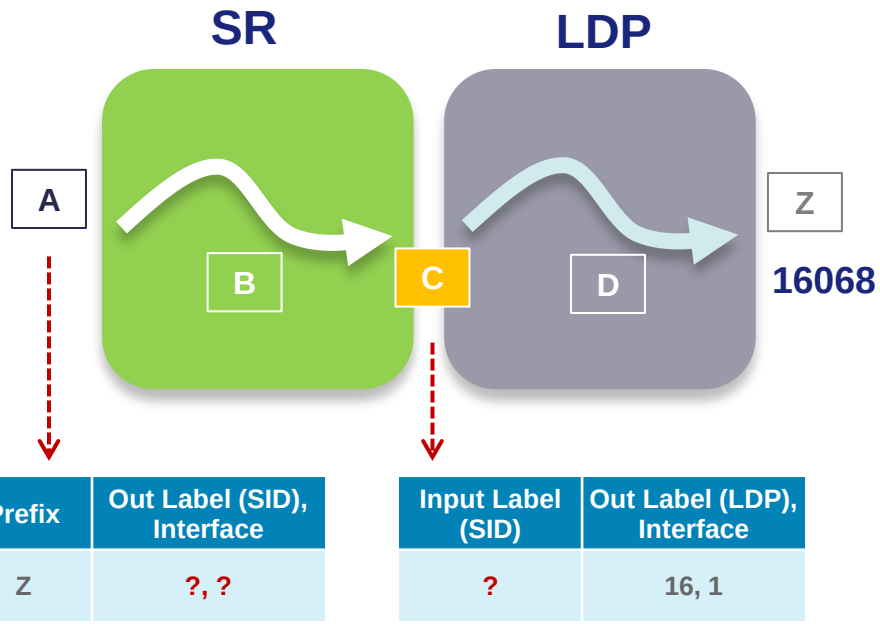
Связность двух различных доменов происходит автоматически и не требует ручной настройки

От SR к LDP – требуется уточнение

- Узлу A требуется передать трафик для узла Z, но узел Z и часть промежуточных узлов не поддерживают SR:
  - Отсутствует SR outgoing label, Z не поддежривает SR, Z не может объявить никакого prefix-SID
  - → Какой label следует использовать узлу A для передачи трафика?

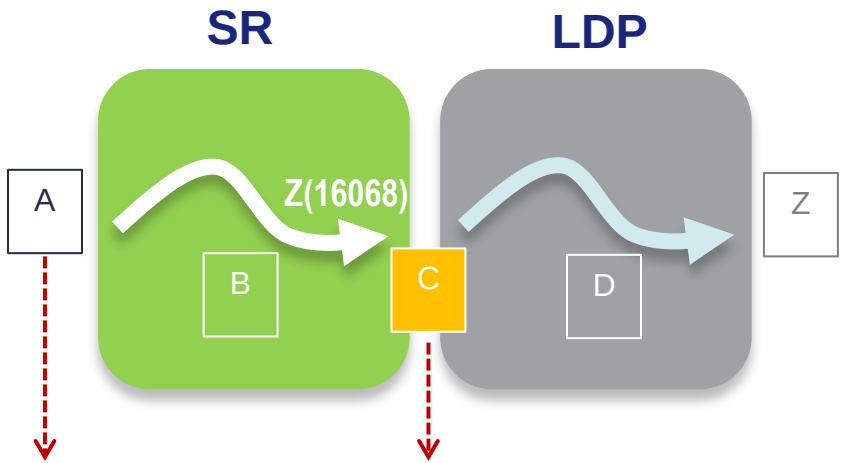
В этом случае необходимо подключить prefix segment LSP к LDP LSP вручную

Любой пограничный узел с поддержкой SR/LDP должен создать запись SR-to-LDP FIB



От SR к LDP – функционал Mapping Server

- Функционал Mapping Server используется для объявления SID mappings для узлов не поддерживающих SR
  - Например, узел C объявляет что узел Z имеет SID 16068
- Узлы A и B инсталлируют нормальный SR prefix segment 16068 для узла Z
- C понимает что его next hop для SPT в направлении Z не поддерживает SR, поэтому C добаляет запись SR-to-LDP FIB
  - incoming label: prefix-SID bound to Z (16068)
  - outgoing label: LDP binding from D for FEC Z
- Узел A посылает трафик для узла Z с single label: 16068



Prefix	Out Label (SID), Interface
Z	16068, 0

Input Label (SID)	Out Label (LDP), Interface
16068	16, 1

Поддержка в текущих продуктах Cisco

NCS6000

CRS-3 / CRS-X

ASR9000

NCS5500

(NCS4000)

CSR1000v

XRV-9000

ASR900

ASR1000 / ISR4000 / (*cBR8*)

(NEXUS 7000)
NEXUS 9000

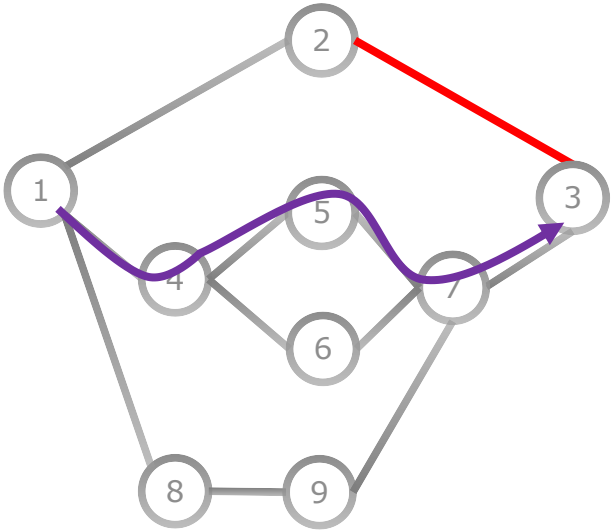
FD.io
(Docker)
(Linux Kernel)

IOS XR IOS XE NexOS Linux

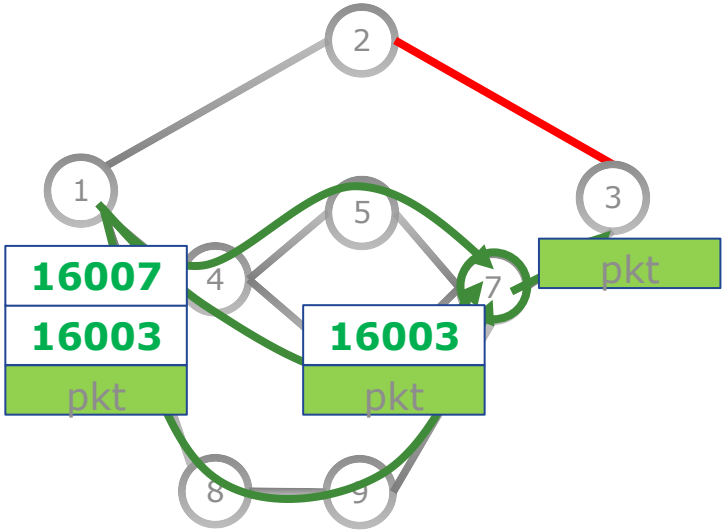
() roadmap

Segment Routing Traffic Engineering

Circuit Optimization vs SR Optimization

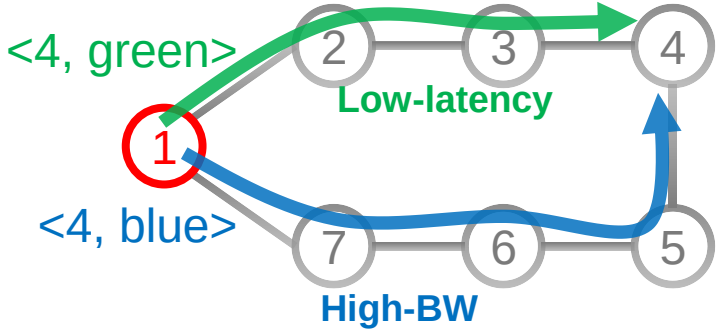


Классический алгоритм не эффективен
Требуется определить список всех транзитных узлов: {4, 5, 7, 3}
Нет ECMP,
Старый алгоритм и технология, ATM optimized



Используется SR-оптимизированный алгоритм
!No more circuit!
Recognized Innovation - Sigcomm 2015
SID List: {7, 3}
ECMP, minimized SID list, IP-optimized

SR Policy



- SR Policy определяется при помощи трех составляющих:
  - The **head-end** (точка примения политики)
  - The **endpoint** (точка назначения)
  - The **color** (обязательно числовое значение)
- Непосредственно на head-end, SR Policy однозначно определяется связкой **<color, endpoint>**
- В качестве **endpoint** может быть указан как IPv4 так и IPv6 address

cisco

SR Policy - пример

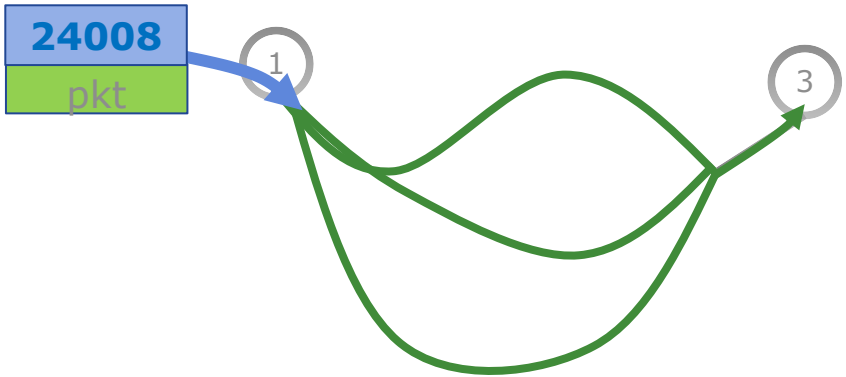
```
segment-routing
 traffic-eng
  policy FOO
   end-point ipv4 1.1.1.4 color 20
   binding-sid mpls 1000
   path
    preference 100
     explicit SIDLIST1
    preference 200
      dynamic mpls
      metric
       type latency
      affinity
       exclude-any red

explicit-path name SIDLIST1
   index 10 mpls label 16002
   index 20 mpls label 30203
   index 30 mpls label 16004
```

```
SR policy (1.1.1.4, 20)

Path received via BGP signaling
  preference 300
  binding-sid mpls 1000
  weight 1, SID list <16002, 16005>
  weight 2, SID list <16004, 16008>

Path received via PCEP signaling
  preference 400
  binding-sid mpls 1000
  SID list <16002, 16005>

Path received via NETCONF signaling
  preference 500
  binding-sid mpls 1000
  SID list <16002, 16005>
```

FIB @ headend
Incoming label: 1000
Action: pop and push <16002, 30203, 14004>

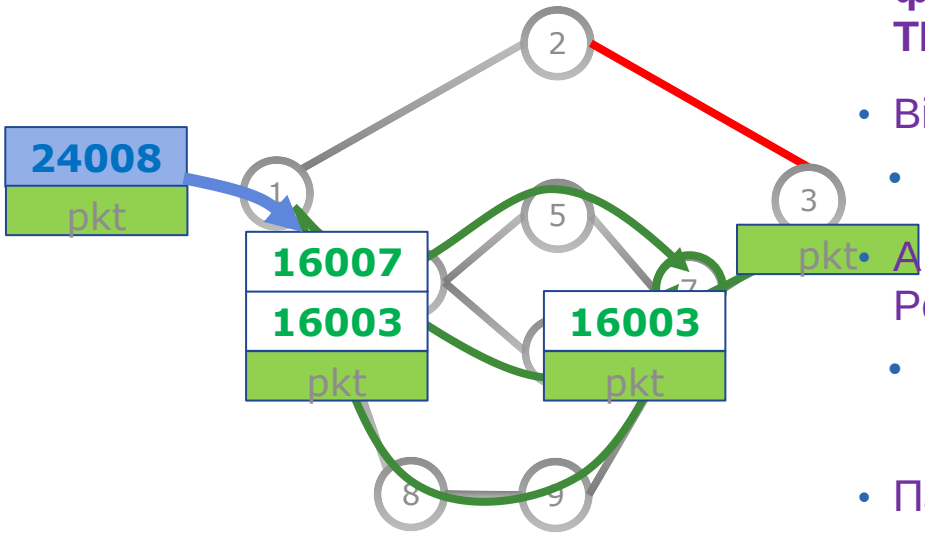
cisco

Binding SID



- **Binding Segment является фундаментальным блоком для работы SR-TE**
- Binding Segment - это локальный сегмент
  - Имеет локальное значение
- A Binding-Segment ID ссылается на SRTE Policy
  - Каждая SRTE Policy связана 1-к-1 с Binding-SID
- Пакеты, полученные с меткой Binding-SID в качестве top label, автоматически обрабатываются SRTE Policy, связанной с Binding-SID
  - Binding-SID label is popped, SRTE Policy's SID list is pushed

Binding SID

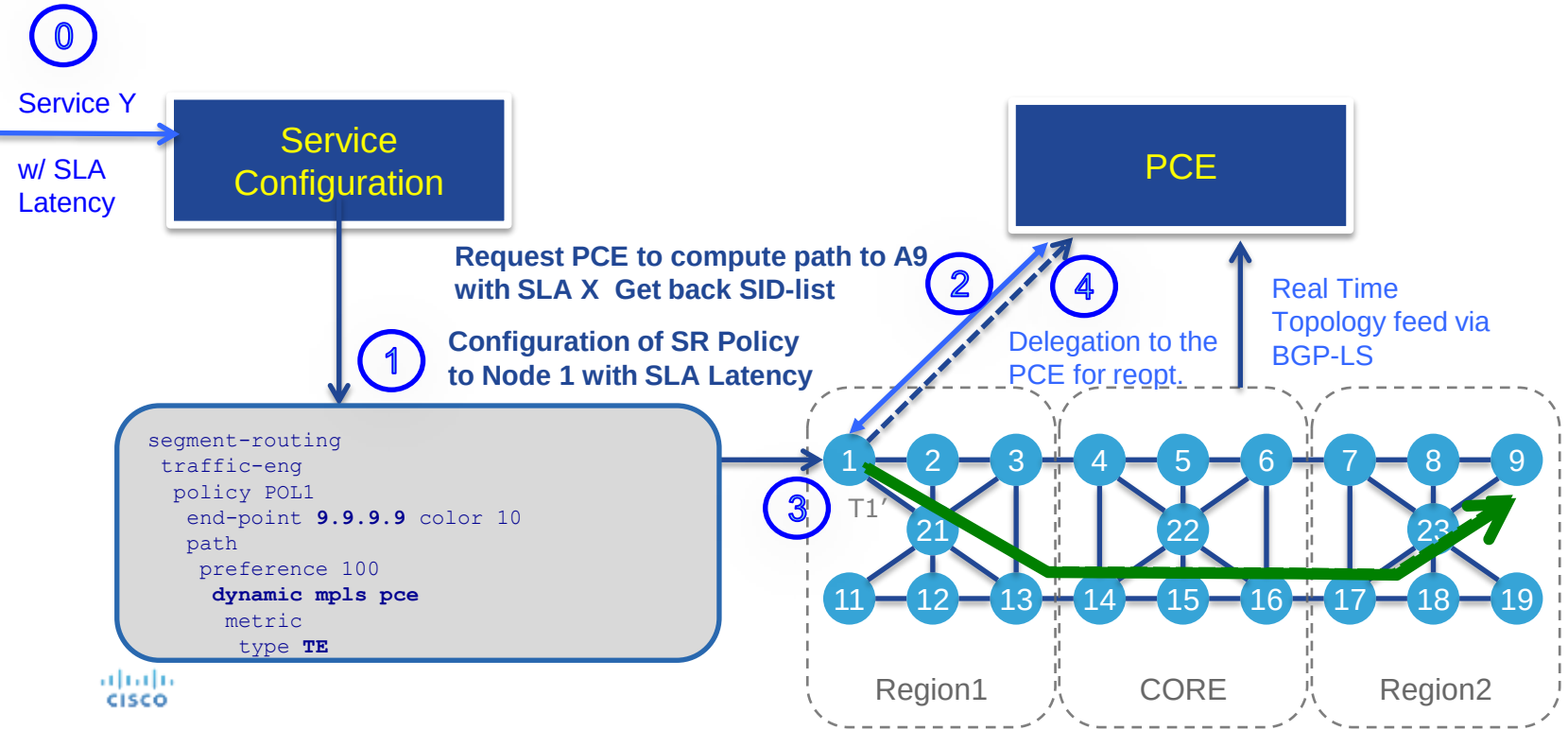


- **Binding Segment является фундаментальным блоком для работы SR-TE**
- Binding Segment - это локальный сегмент
  - Имеет локальное значение
  - A Binding-Segment ID ссылается на SRTE Policy
    - Каждая SRTE Policy связана 1-к-1 с Binding-SID
- Пакеты, полученные с меткой Binding-SID в качестве top label, автоматически обрабатываются SRTE Policy, связанной с Binding-SID
  - Binding-SID label is popped, SRTE Policy's SID list is pushed

cisco

Inter Area Path Computation with SLA

Ask: Provide latency optimized path across multiple AS's from a source to a destination

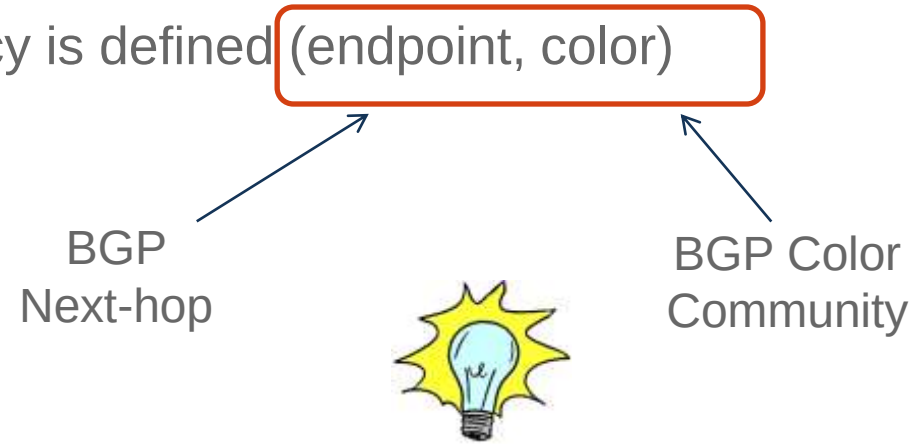


Контроллер SDN - XR Transport Controller (XTC)

- Централизация сбора данных о топологии
- Сбор данных о топологии с помощью стандартных протоколов **BGP-LS ISIS/OSPF**
- Программирование пути стандартным протоколом PCEP
- Поддержка Segment Routing
- Мультивендорная поддержка

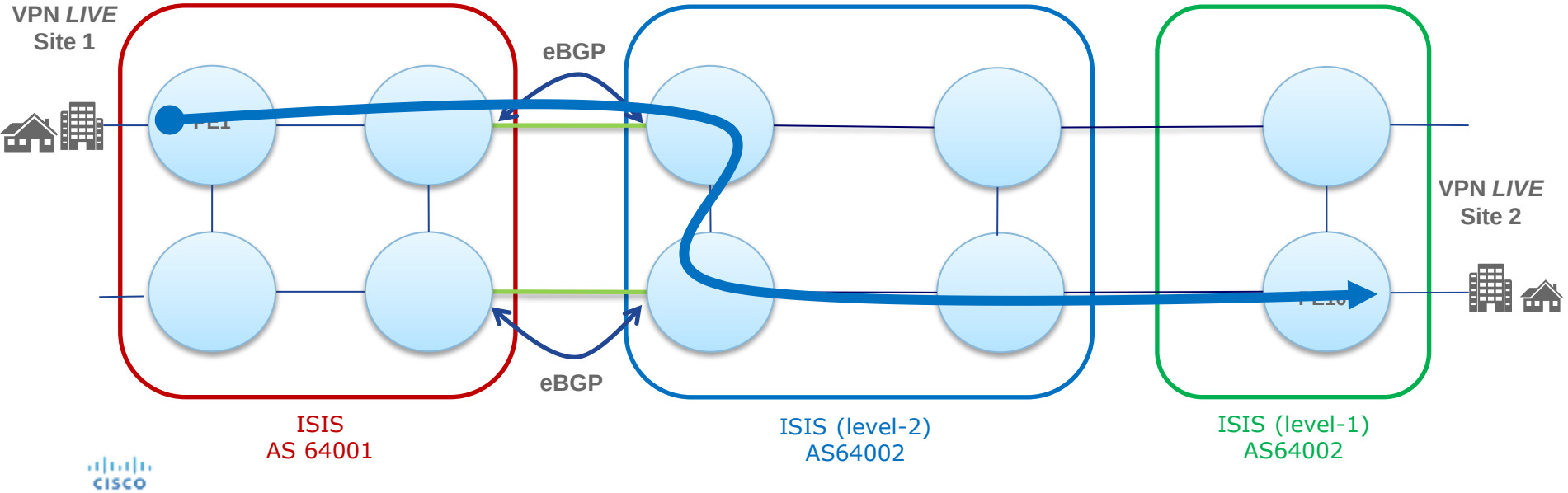
On-Demand SR Policy

- A service head-end automatically instantiates an SR Policy to a BGP nhop when required (on-demand), automatically steering the BGP traffic into this SR Policy
- Color community is used as SLA indicator
- Reminder: an SR policy is defined (endpoint, color)

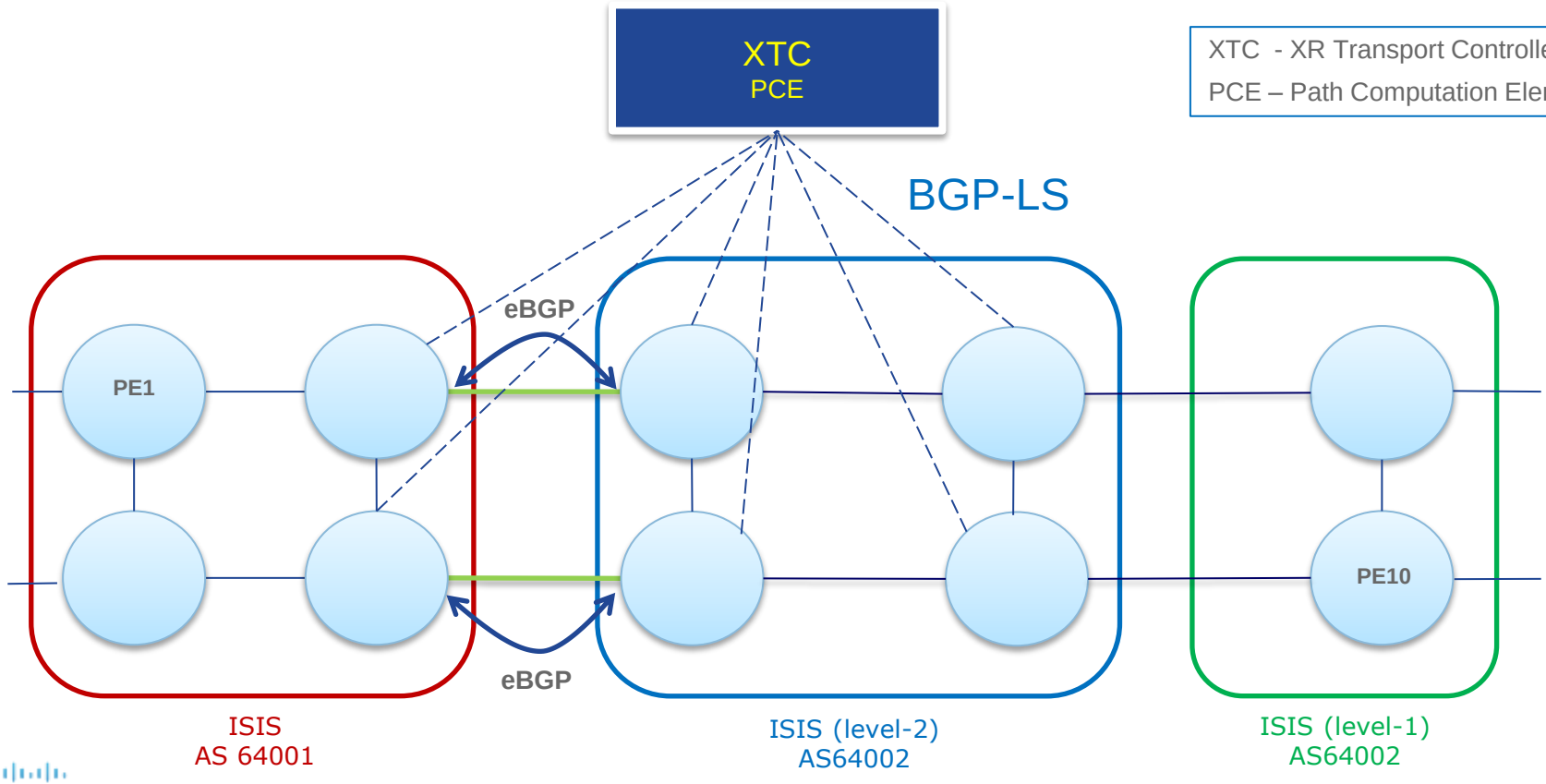


What is ODN

Intra and inter-area/AS On Demand Connectivity with SLA for L2 and L3 VPN

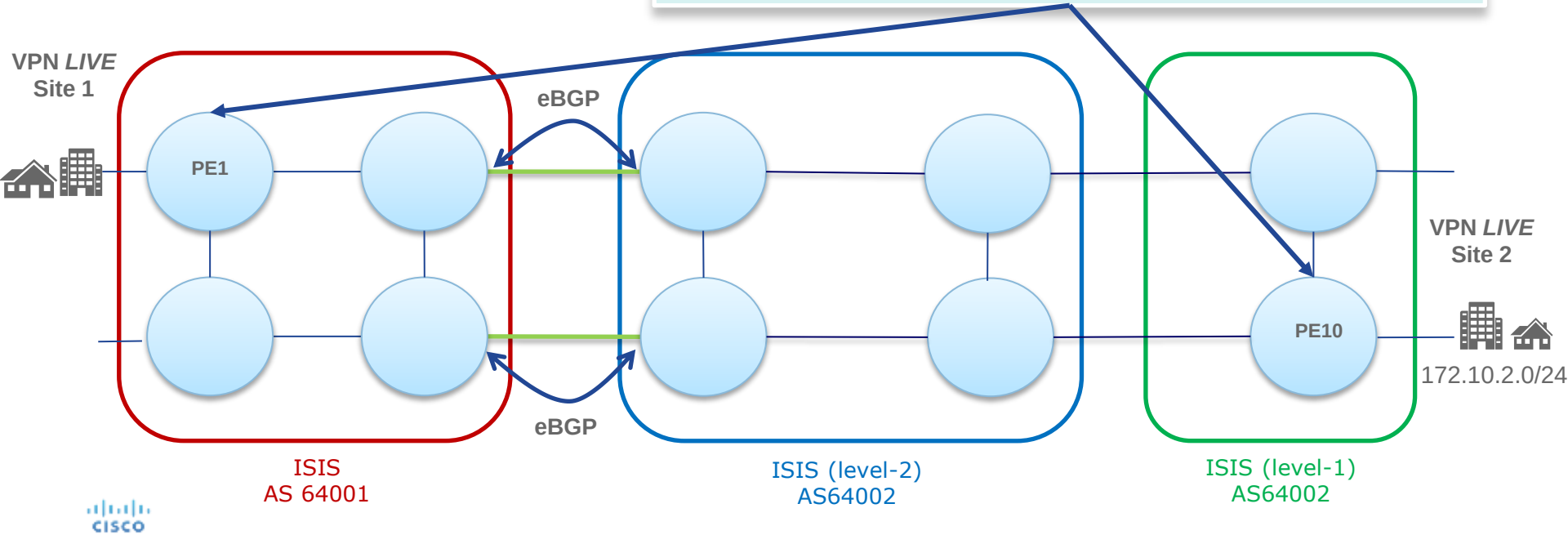


Real Time Topology Feed

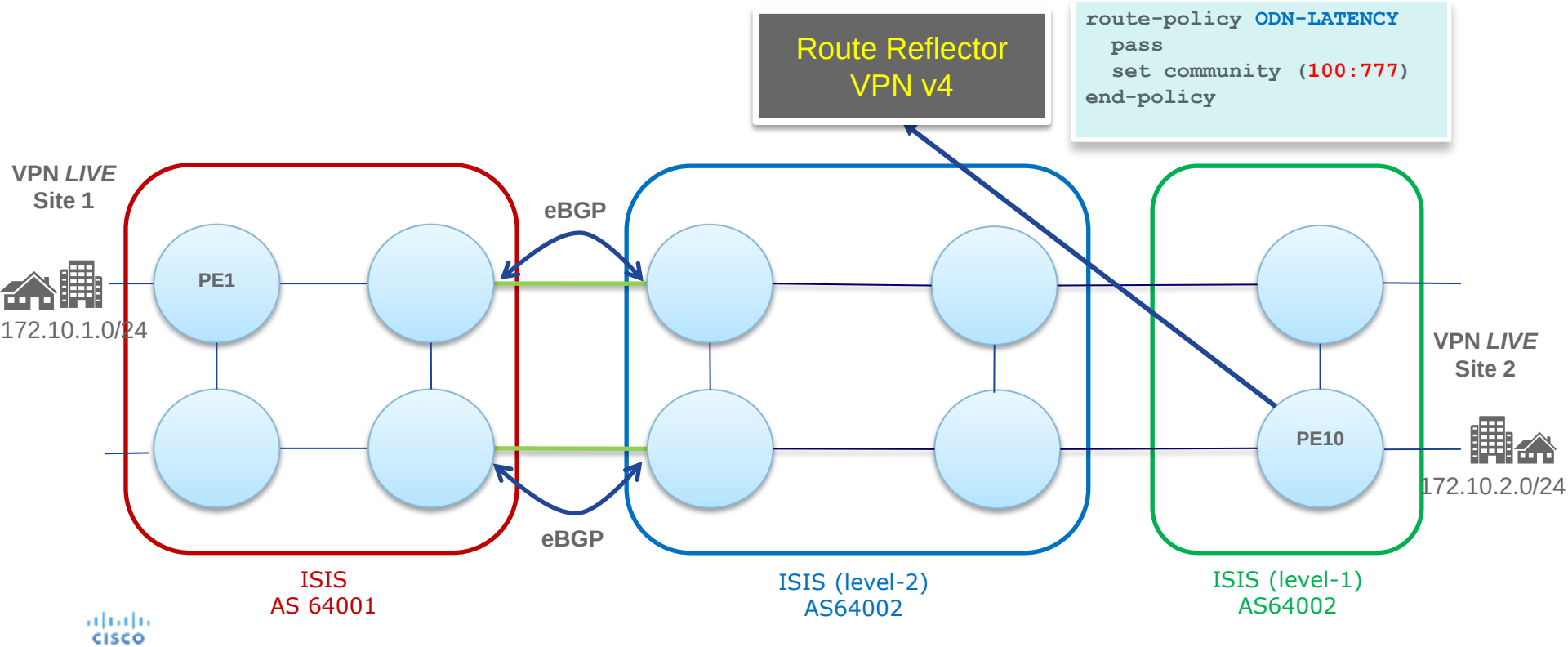


VPN provisioning

```
vrf LIVE
      address-family ipv4 unicast
      import route-target   1:303
      export route-target   1:303

router bgp 64002
   vrf LIVE
      rd auto  address-family ipv4 unicast
      redistribute connected route-policy ODN-LATENCY
```

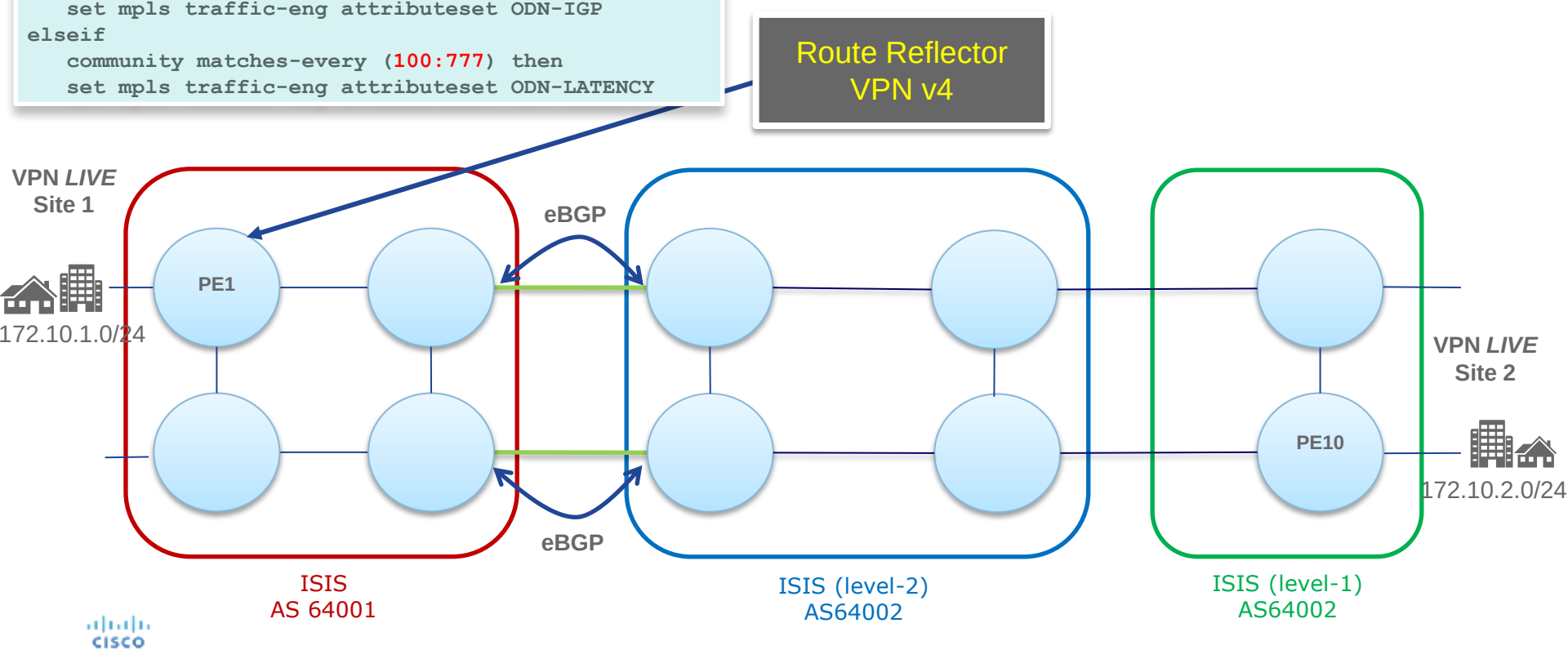


BGP VPN routes distribution via Route Reflector



PE1 process BGP VPNv4 172.10.2.0/24 nh 10.10.10.10

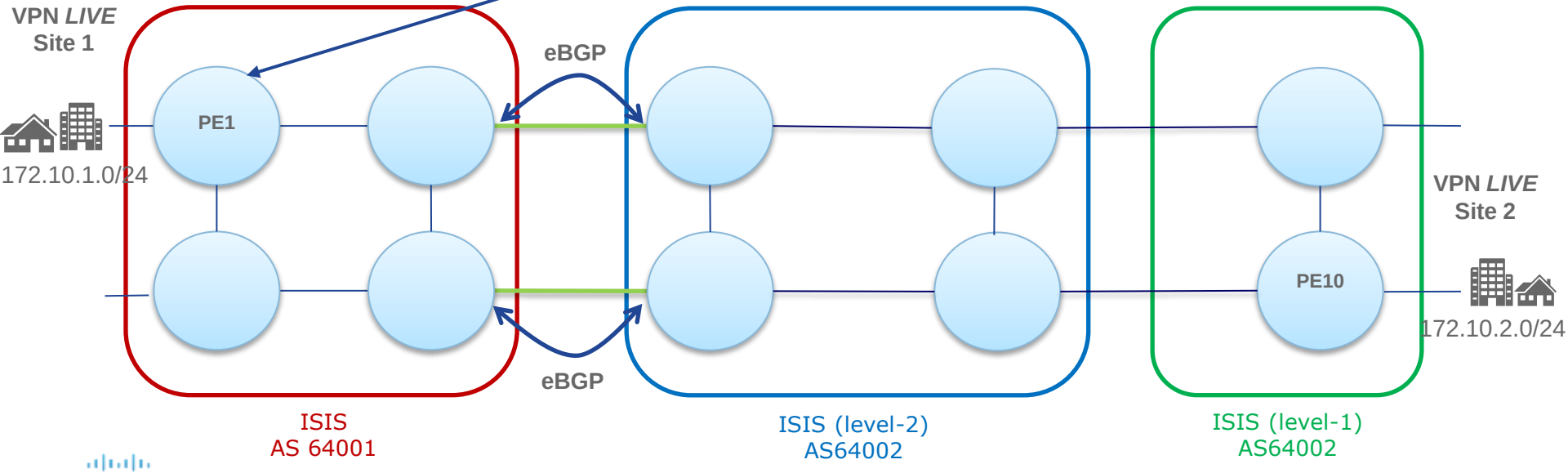
```
if community matches-every (100:666) then
   set mpls traffic-eng attributeset ODN-IGP
elseif
   community matches-every (100:777) then
   set mpls traffic-eng attributeset ODN-LATENCY
```



PE1 process BGP VPNv4 172.10.2.0/24 nh

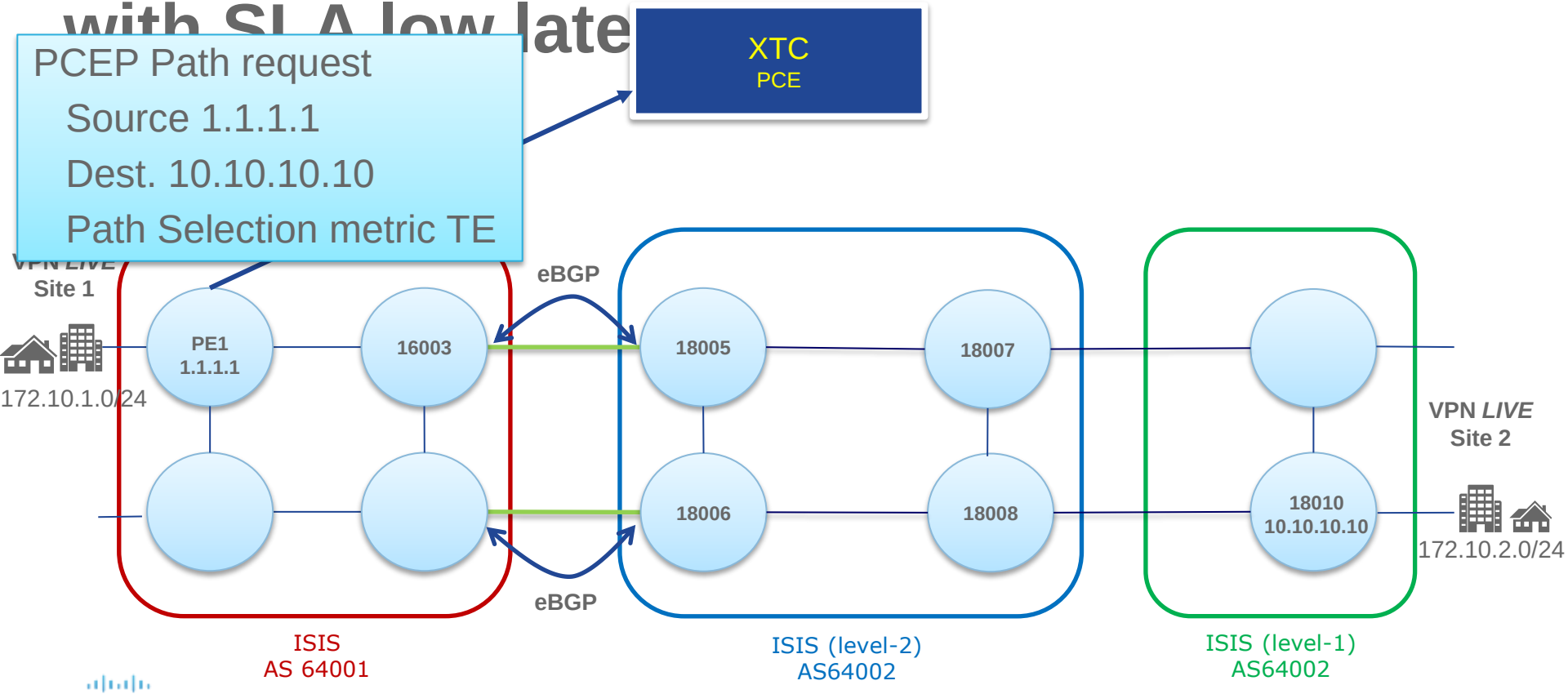
```
if community matches-every (100:666) then
   set mpls traffic-eng attributeset ODN-IGP
elseif
   community matches-every (100:777) then
   set mpls traffic-eng attributeset ODN-LATENCY
```

```
segment-routing
 traffic-eng
  attribute-set ODN-LATENCY
   pce
   metric  type te
```

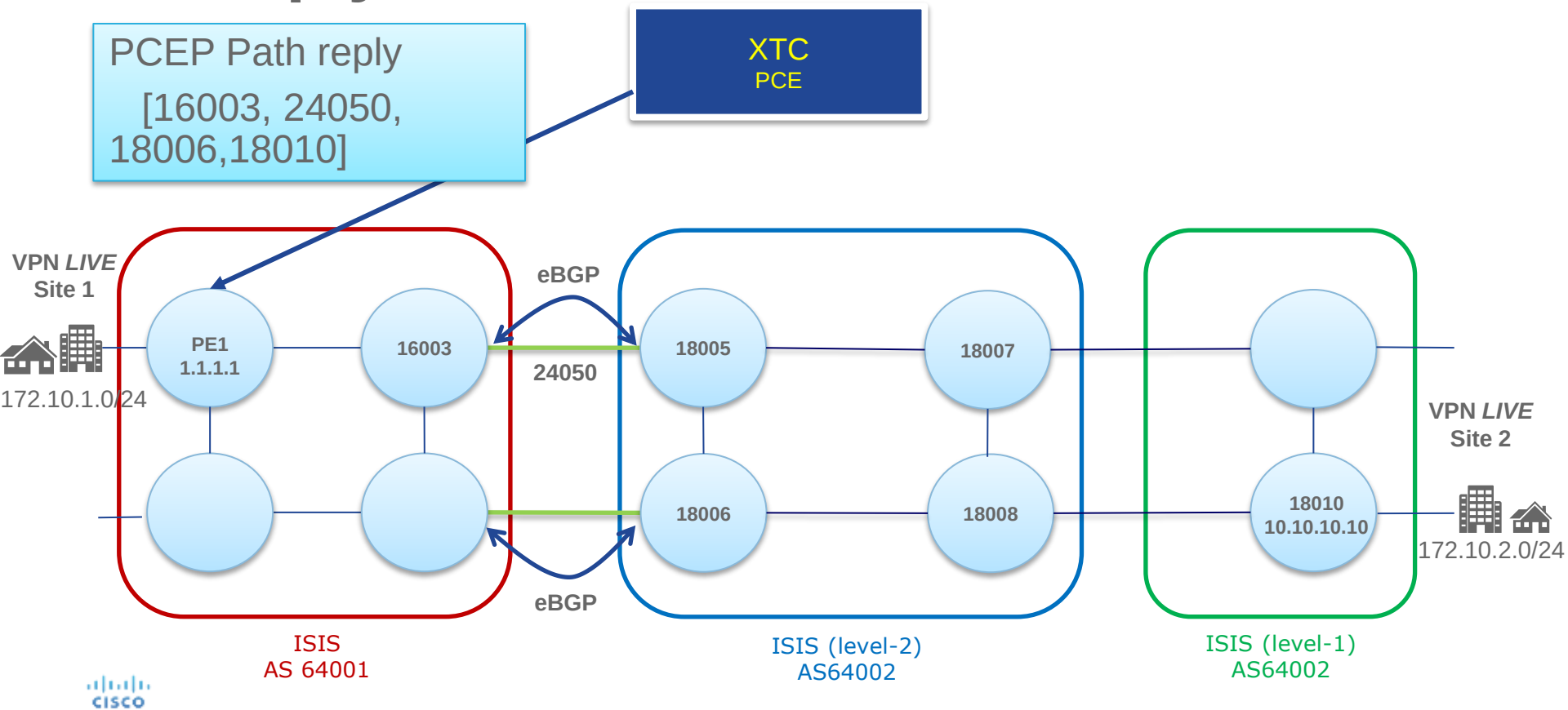


PE1 ask PCE to resolve 10.10.10.10 with SLA low late

PCEP Path request
- Source 1.1.1.1
- Dest. 10.10.10.10
- Path Selection metric TE



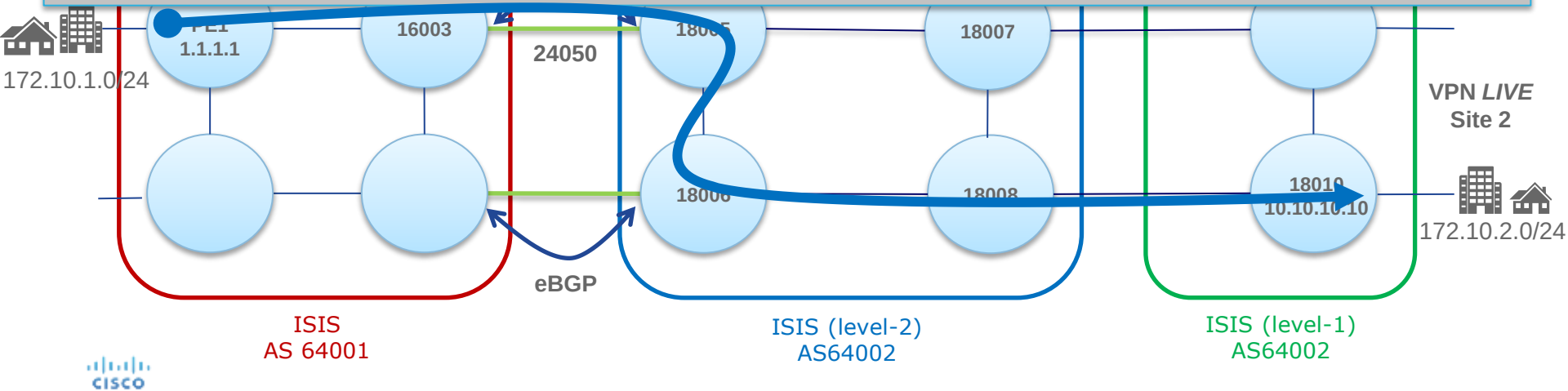
PCE reply with the SID list



PE1 install the SR Policy

and assign a Binding SID the SR policy

```
RP/0/0/CPU0:PE-1#show segment-routing traffic-eng policy |
incl "Name|Admin|Bind  "
Name: auto_sr_policy_1  End-Point: 10.10.10.10  Color: 20
    Admin: up  Operational: up for 00:00:50 (since Fri Feb 24
12:02:06 UTC 2017)
    Binding SID: 24008
```

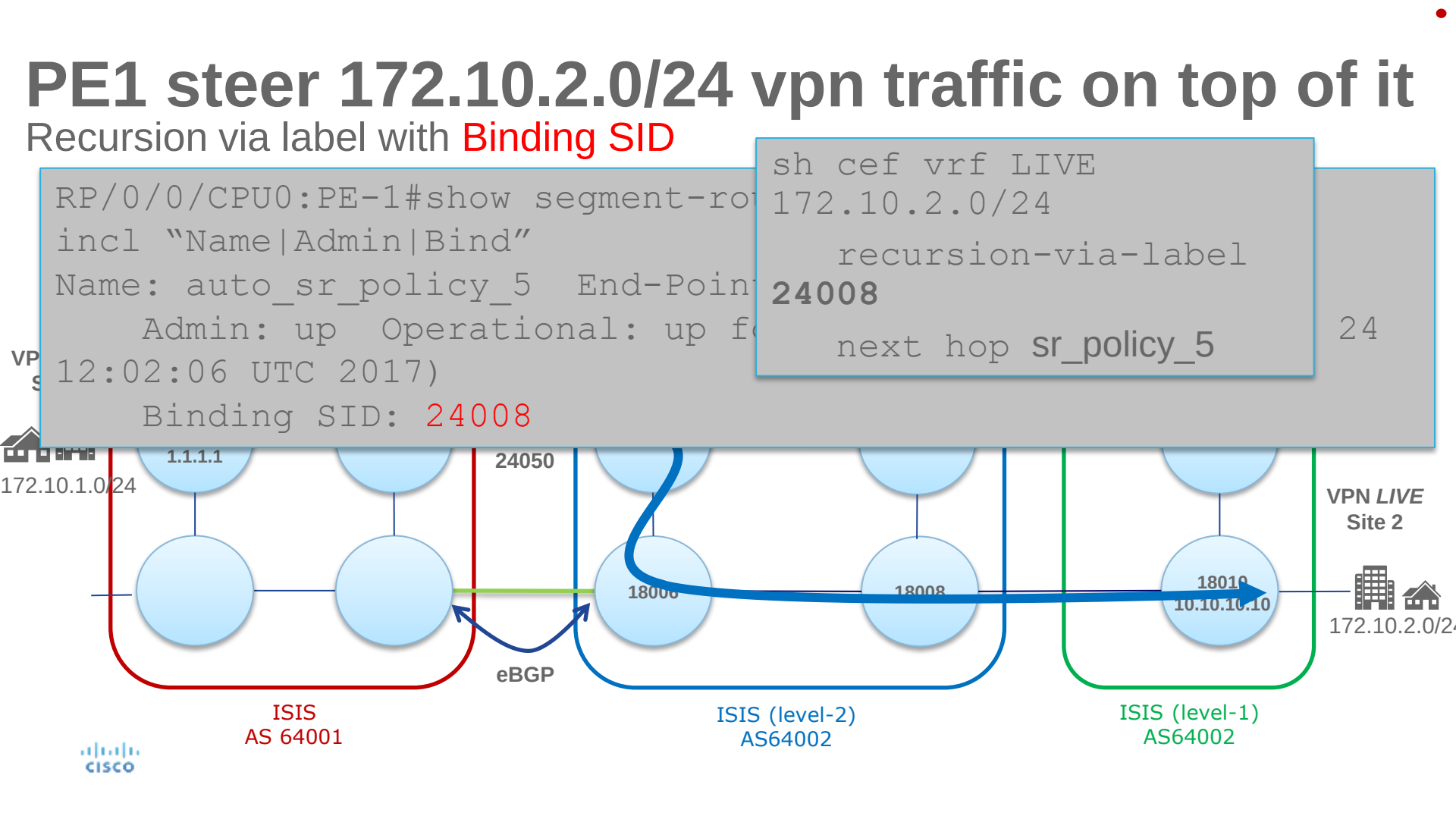


PE1 steer 172.10.2.0/24 vpn traffic on top of it

Recursion via label with Binding SID

```
RP/0/0/CPU0:PE-1#show segment-ro
incl "Name|Admin|Binding"
Name: auto_sr_policy_5  End-Poin
    Admin: up  Operational: up f
12:02:06 UTC 2017)
    Binding SID: 24008
```

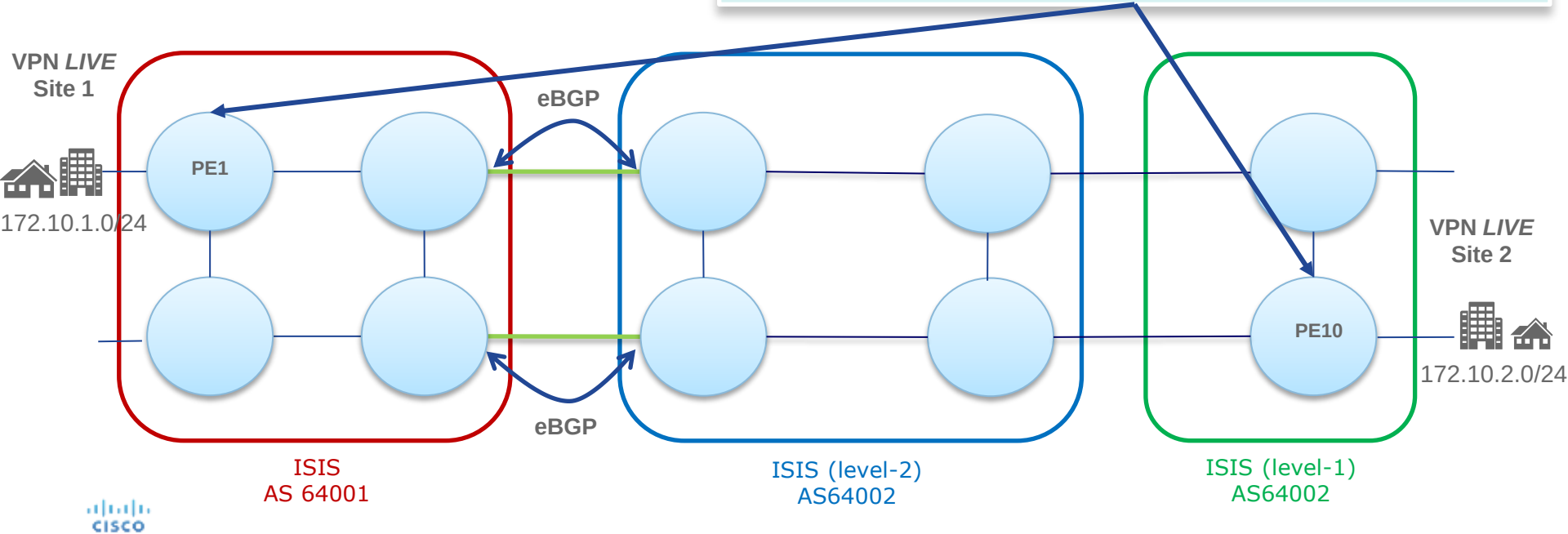
```
sh cef vrf LIVE
172.10.2.0/24
   recursion-via-label
24008
   next hop sr_policy_5
```



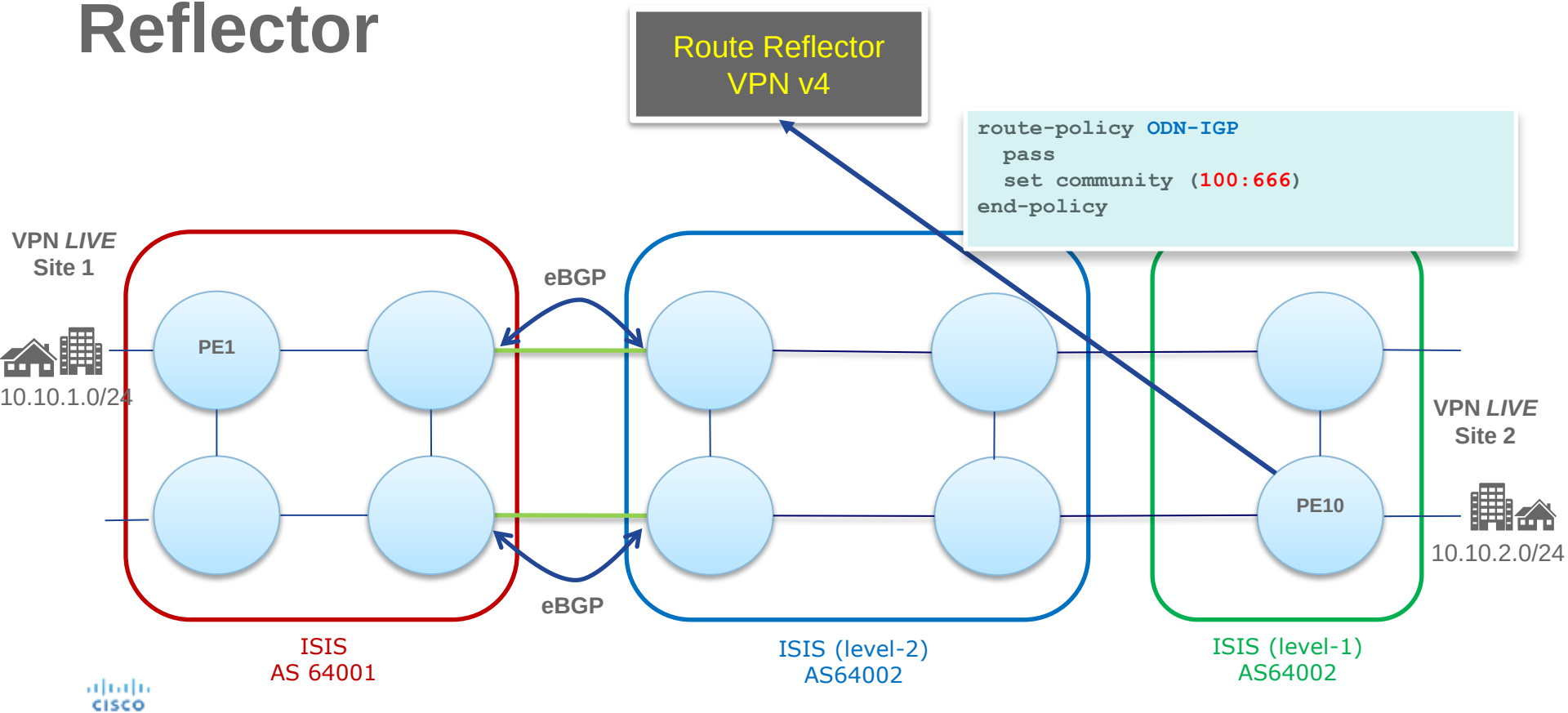
what if we change SLA?

VPN provisioning

```
vrf LIVE
      address-family ipv4 unicast
       import route-target    1:303
       export route-target    1:303

router bgp 64002
   vrf LIVE
      rd auto  address-family ipv4 unicast
      redistribute connected route-policy ODN-IGP
```



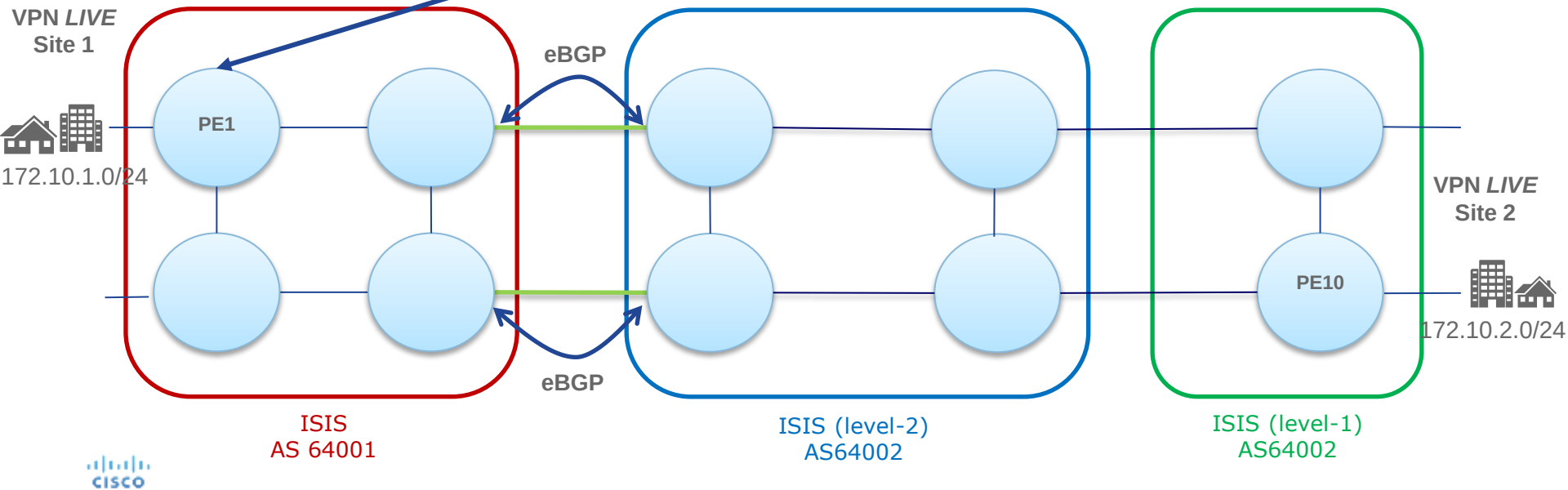
BGP VPN routes distribution via Route Reflector



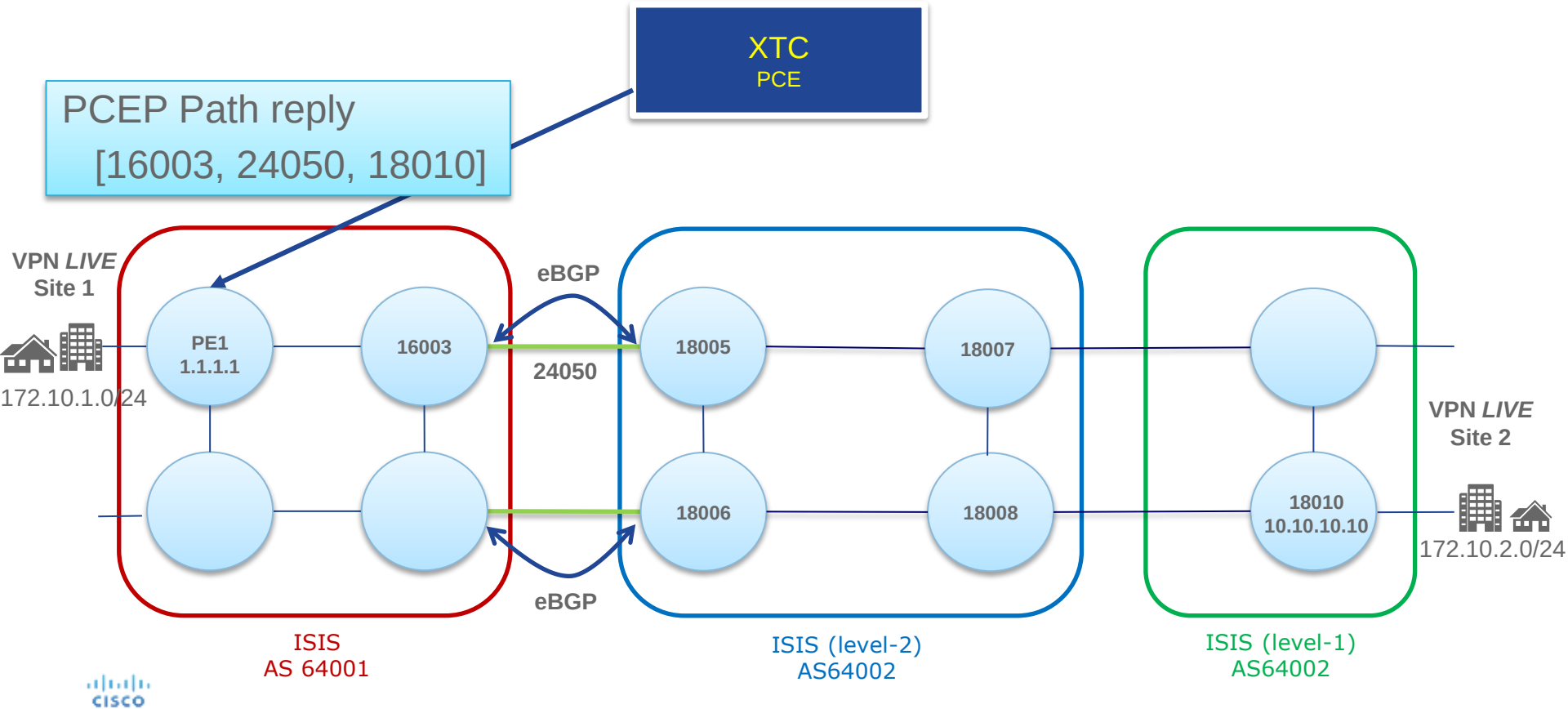
PE1 process BGP VPNv4 172.10.2.0/24 nh 10.10.10.10

```
if community matches-every (100:666) then
   set mpls traffic-eng attributeset ODN-IGP
elseif
   community matches-every (100:777) then
   set mpls traffic-eng attributeset ODN-LATENCY
```

```
segment-routing
 traffic-eng
  attribute-set ODN-IGP
   pce
   metric type igp
```



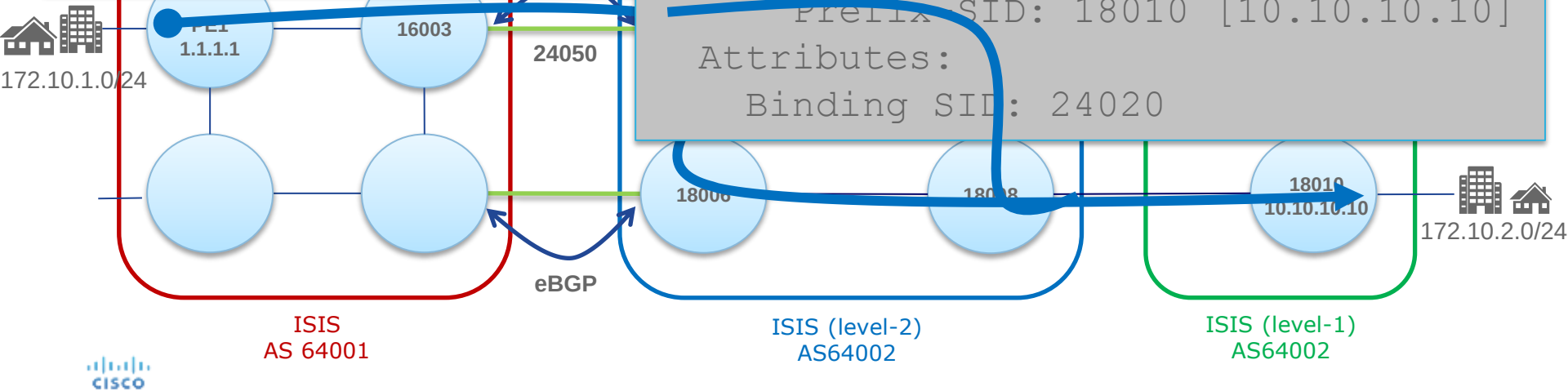
PCE reply with the SID list



XTC always optimize the path
in term of label stack and ECMP

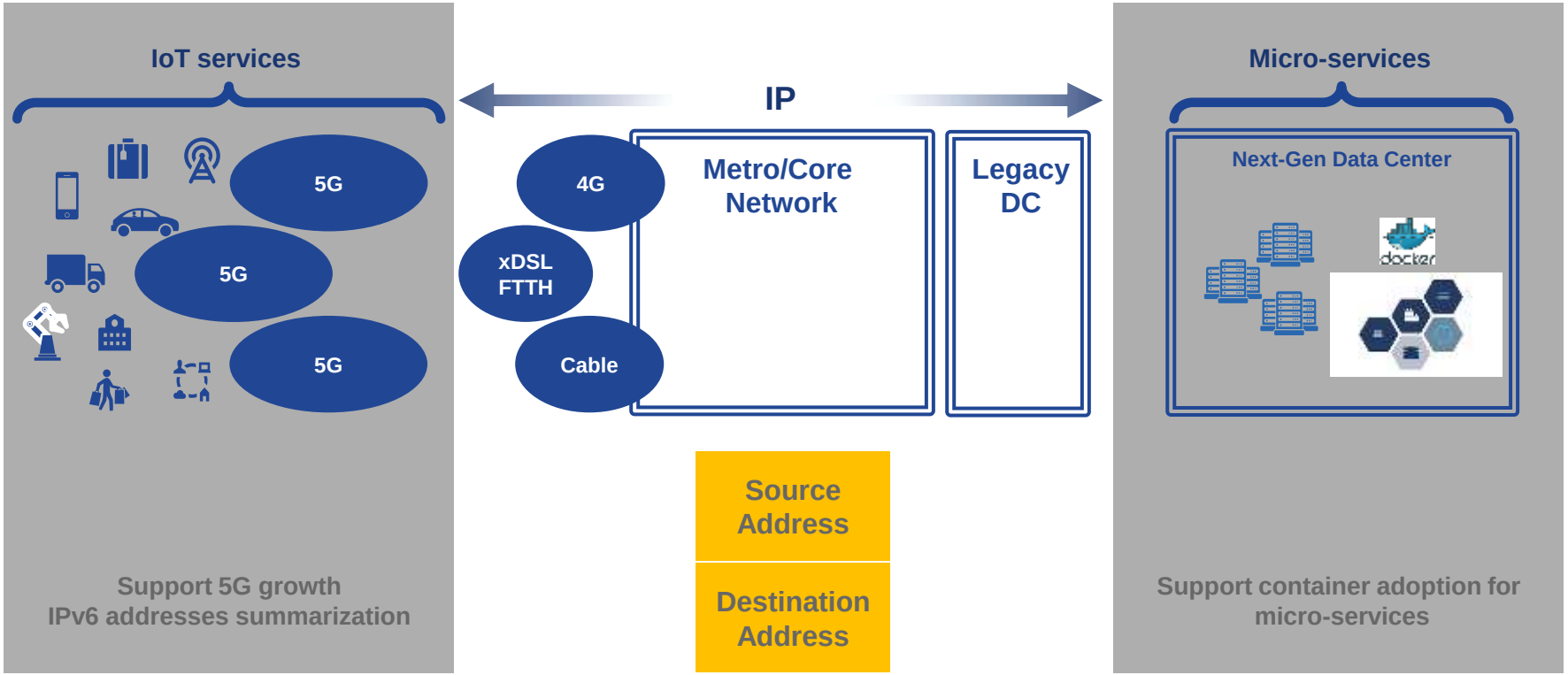
```
RP/0/0/CPU0:PE-1#show seg
incl "Name|Admin"
Name: auto_sr_policy_1  B
    Admin: up  Operationa
12:02:06 UTC 2017)
```

```
Paths:
    Preference 100 (dynamic pce)
(active)
      Prefix-SID: 16003 [3.3.3.3]
      Adjacency-SID: 24050 [99.3.5.3
- 99.3.5.3]
      Prefix-SID: 18010 [10.10.10.10]
  Attributes:
    Binding SID: 24020
```



Введение в SRv6

Рост IPv6-трафика



SRv6 – Segment Routing + IPv6

SRv6 for anything else

IPv6 for reach

- Simplicity
  - Protocol elimination
- SLA
  - FRR and TE
- Overlay
- NFV
- SDN
  - SR is de-facto SDN architecture
- 5G Slicing

cisco

SR для всех возможных задач
Сеть как компьютер

Сетевая инструкция

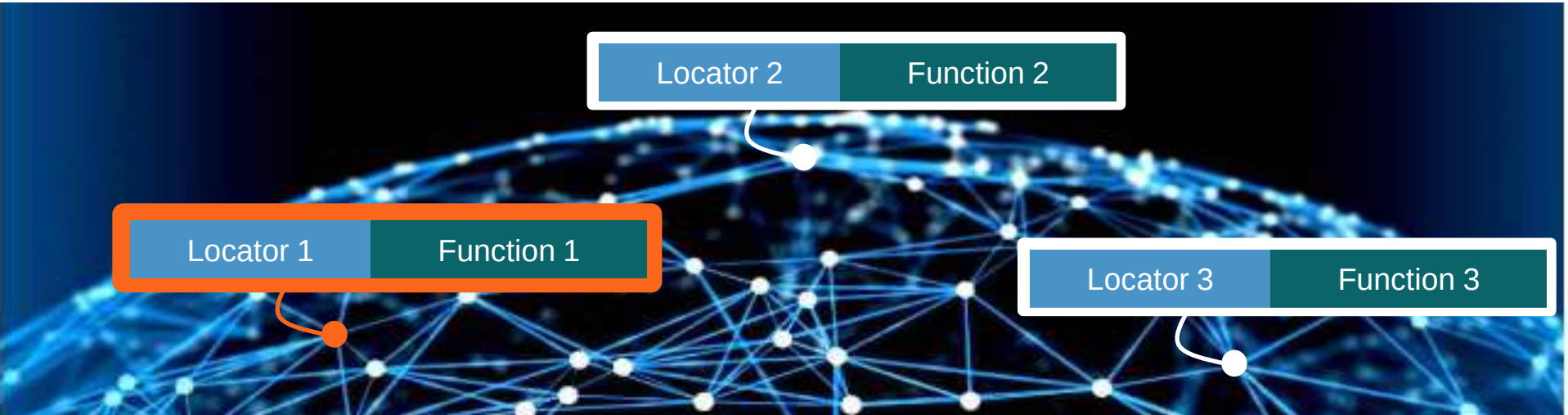
- 128-bit SRv6 SID
  - Locator: определяет путь у узлу, выполняющему функцию
  - Function: любая функция (с опциональной передачей аргументов)

    either local to NPU or app in VM/Container
  - Гибкое определние bit-length границы

Сетевая программа

Следующий сегмент

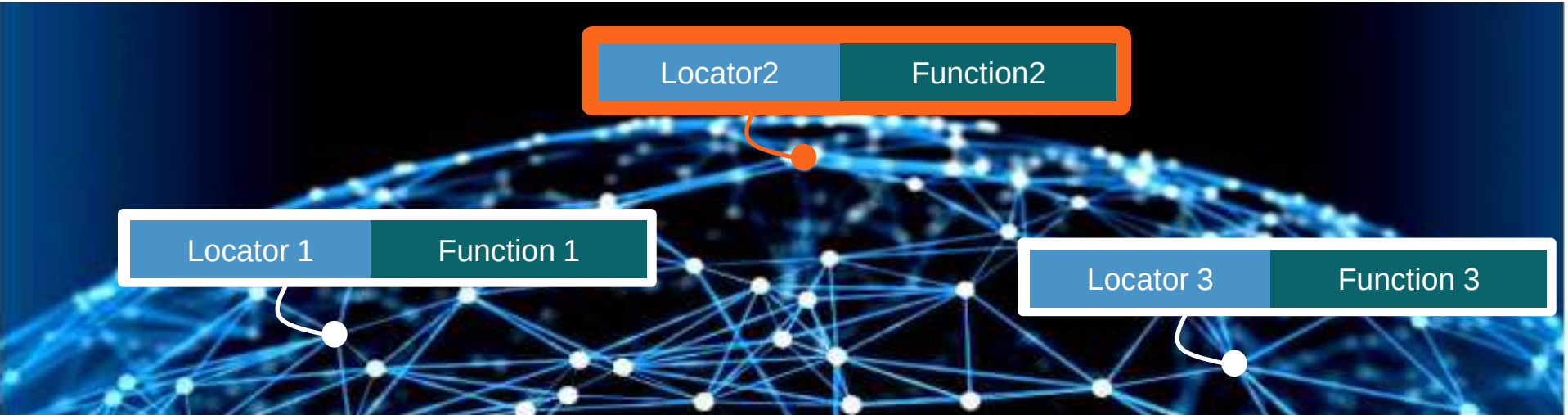
Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3



Сетевая программа

Следующий сегмент

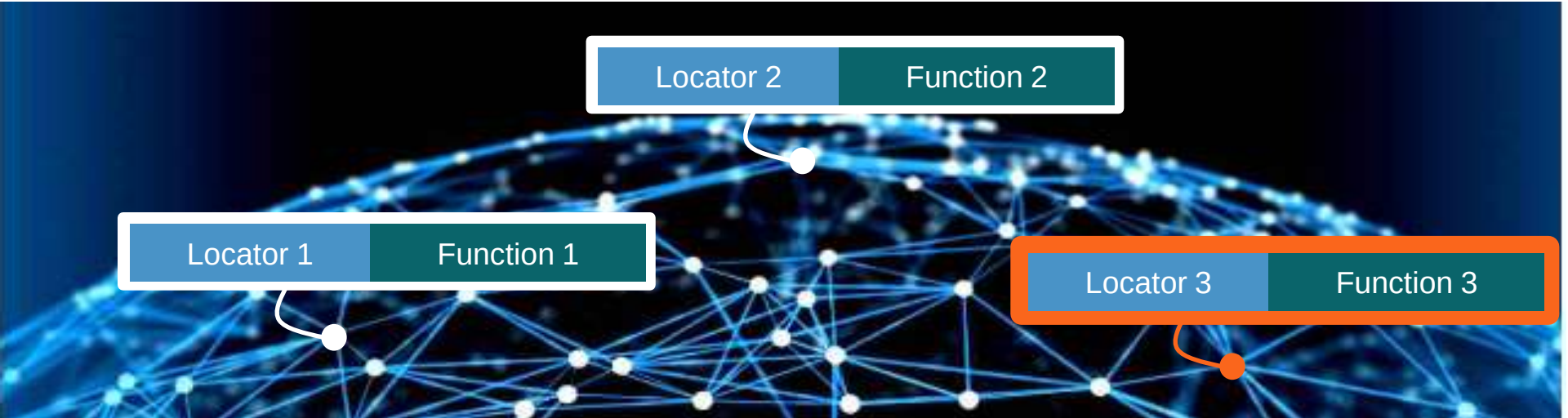
Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3



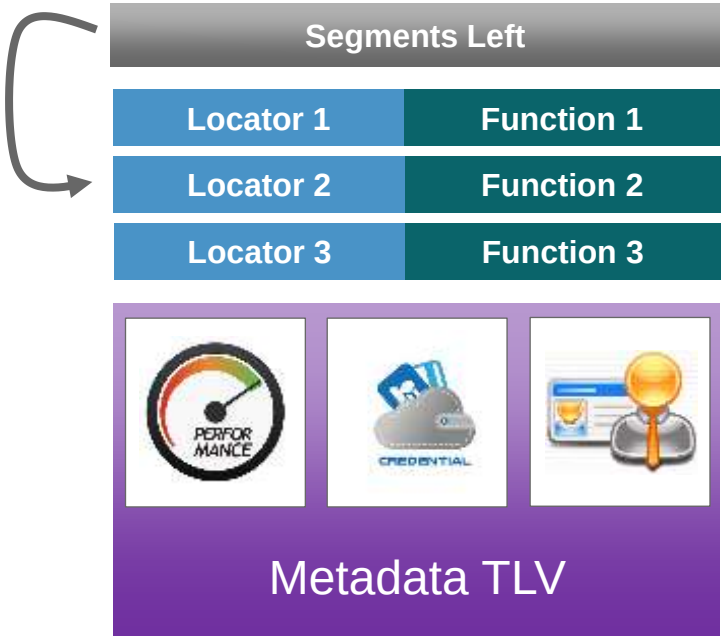
Сетевая программа

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

Следующий сегмент

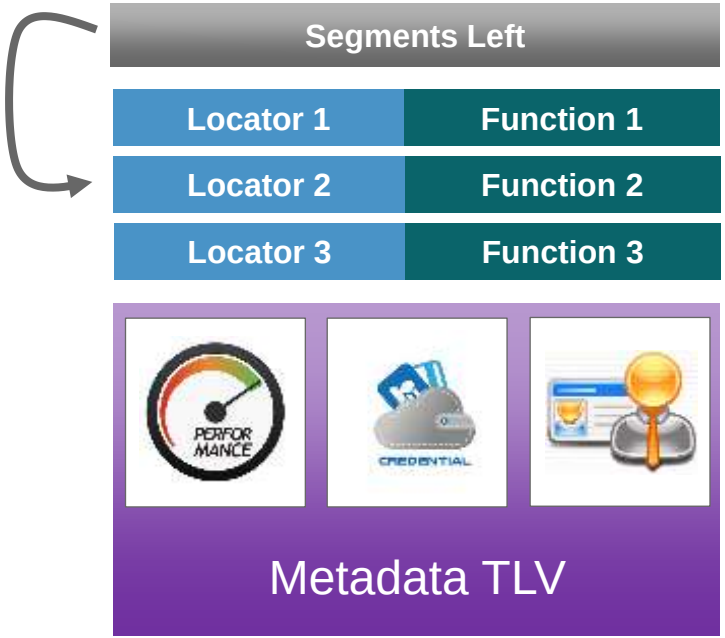


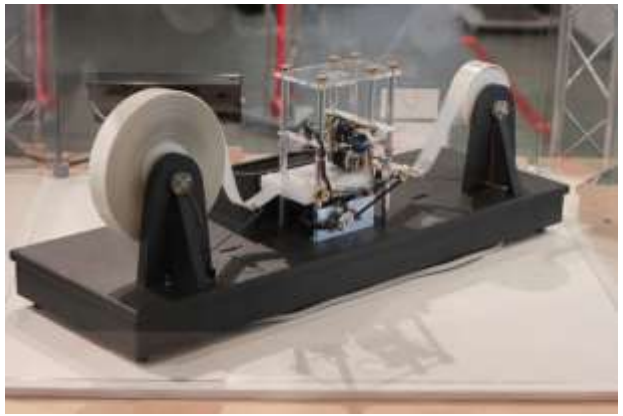
SR заголовок




```
Internet-Draft      IPv6 Segment Routing Header (SRH)     September 2016


    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | First Segment |             Flags             |   RESERVED    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |            Segment List[0] (128 bits IPv6 address)            |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
                               ...
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |            Segment List[n] (128 bits IPv6 address)            |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                                                             //
   //         Optional Type Length Value objects (variable)       //
   //                                                             //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

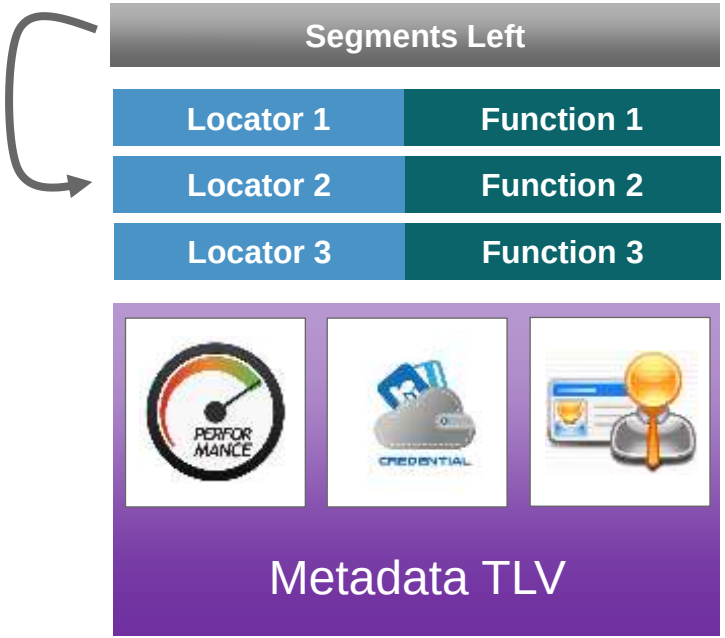
SRv6 для всех возможных задач





Машина Тьюринга

SRv6 для всех возможных задач

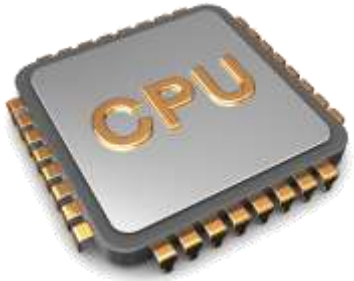


Optimized for HW processing

e.g. Underlay & Tenant use-cases

Optimized for SW processing

e.g. NFV, Container, Micro-Service



cisco

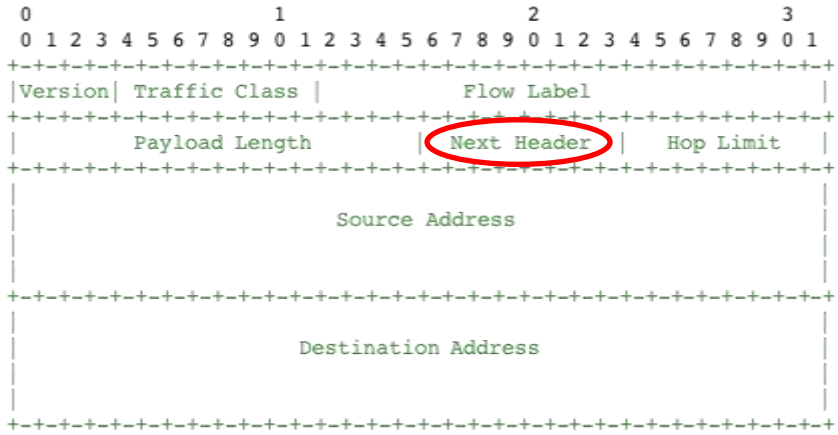
SR Header

SR-IPv6

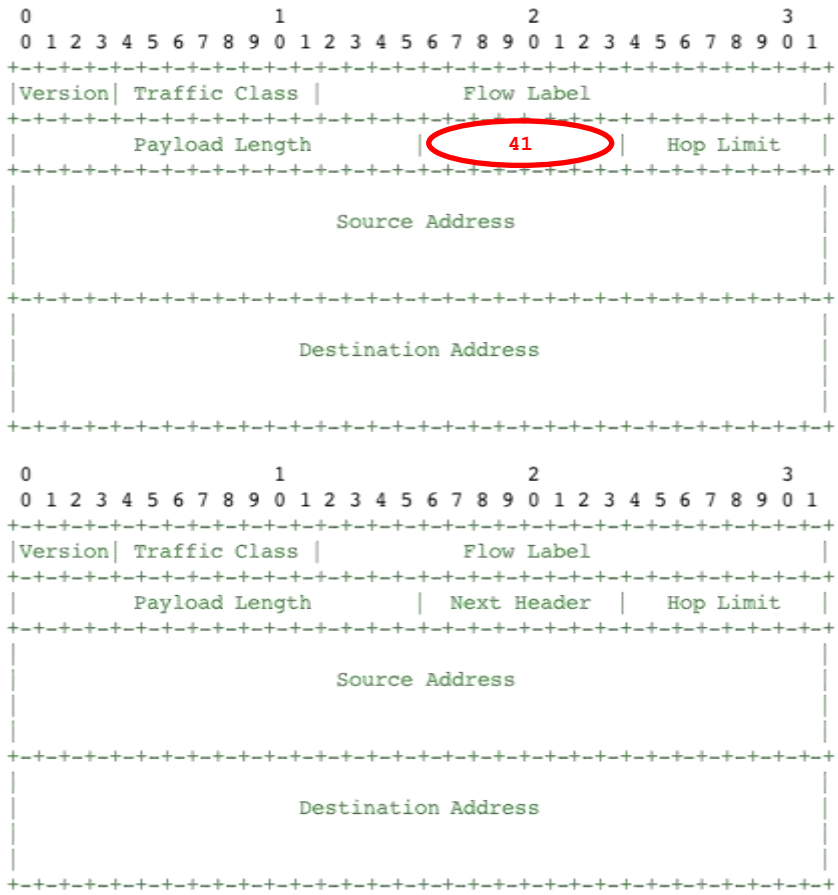
- SR-IPv6: список сегментов хранящихся в новом (безопасном) заголовке Routing Header
  - Segment Routing Header (SRH)
- Существуют две опции использования Segment Routing в v6 сетях
  - IPv6 control plane with a MPLS dataplane
  - IPv6 control plane with a IPv6 dataplane

IPv6 Header

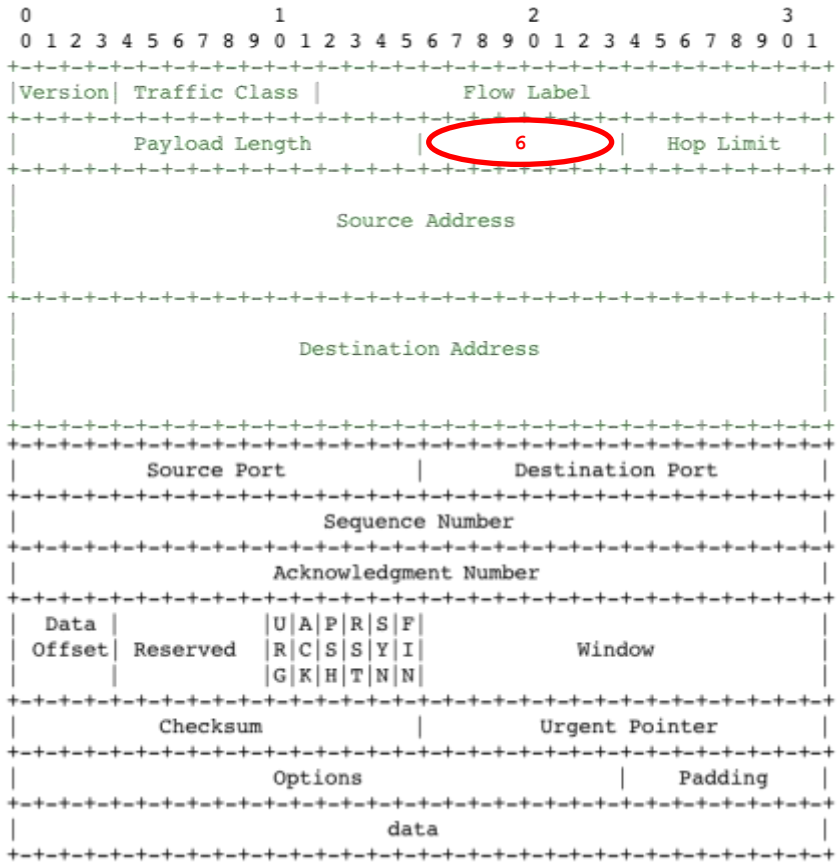
- **Next Header (NH)**
  - **Определяет последующие заголовки**



NH = IPv6

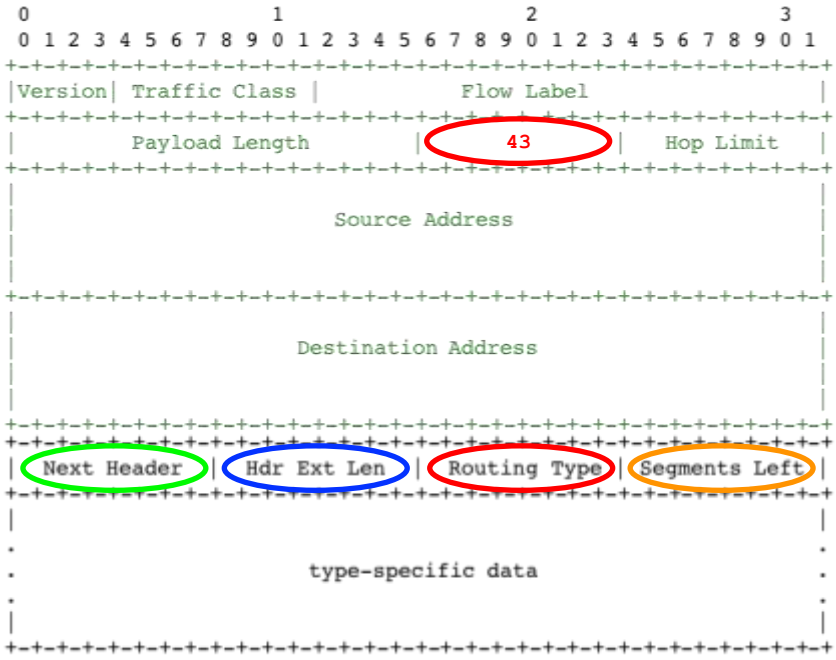

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version
Traffic Class
Flow Label
Payload Length
41
Hop Limit
Source Address
Destination Address
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address

NH = TCP


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version
Traffic Class
Flow Label
Payload Length
6
Hop Limit
Source Address
Destination Address
Source Port
Destination Port
Sequence Number
Acknowledgment Number
Data Offset
Reserved
U A P R S F
R C S S Y I
G K H T N N
Window
Checksum
Urgent Pointer
Options
Padding
data

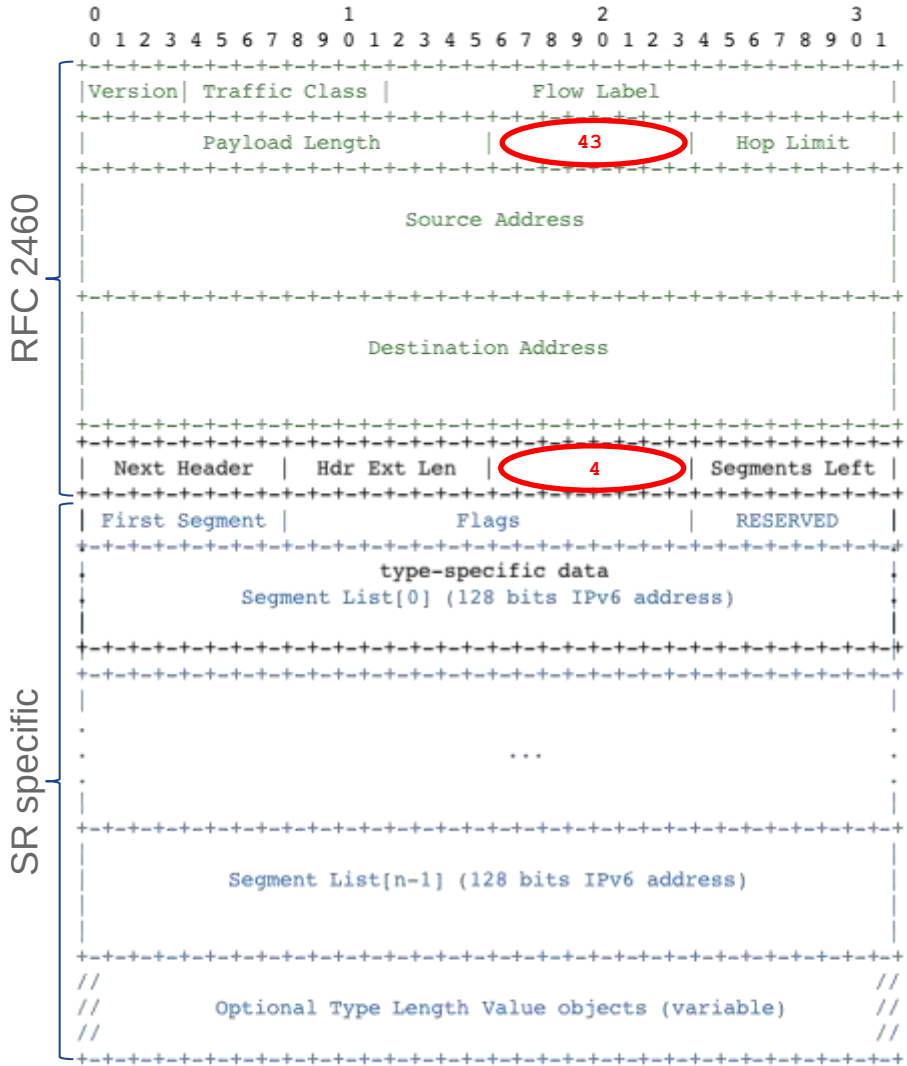
NH = Routing Extension

- Generic routing extension header
  - Defined in RFC 2460
  - Next Header: UDP, TCP, IPv6…
  - Hdr Ext Len: Any IPv6 device can skip this header
  - Segments Left: Ignore extension header if equal to 0
- Routing Type field:
  - 0 Source Route (deprecated since 2007)
  - 1 Nimrod (deprecated since 2009)
  - 2 Mobility (RFC 6275)
  - 3 RPL Source Route (RFC 6554)
  - 4 Segment Routing



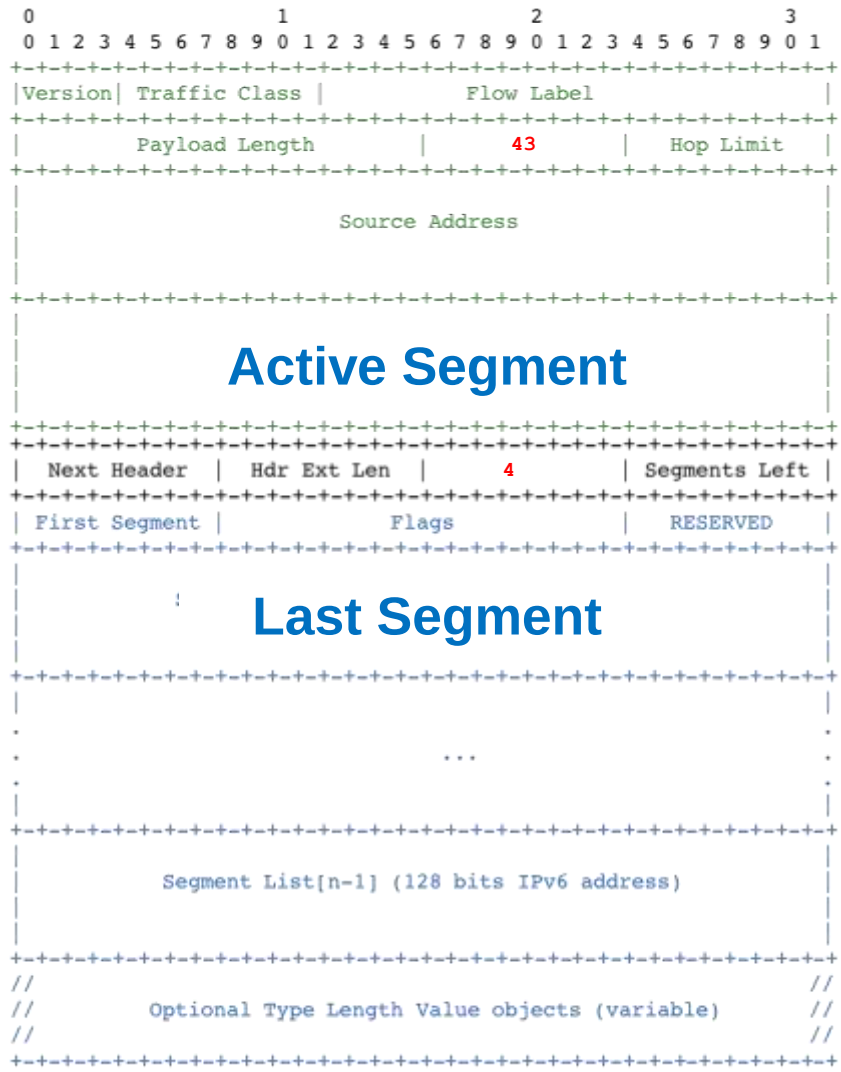
NH = SRv6

- NH = 43, Type = 4



SRH

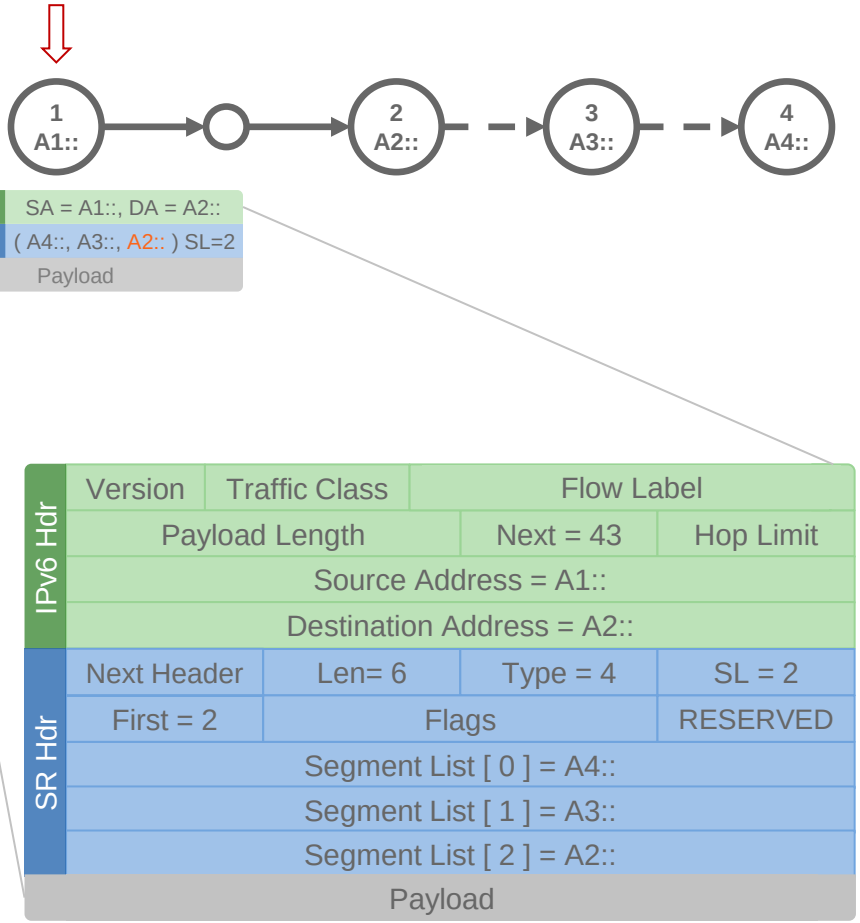
- SRH contains
  - the list of segments
  - Segments left (SL)
  - Flags
  - TLV
- Active segment is in the IPv6 DA
- Next segment is at index SL-1
- The last segment is at index 0
  - Reversed order



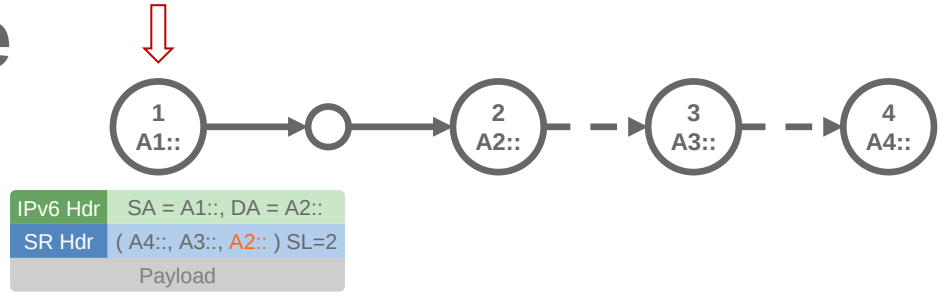
SRH Processing

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



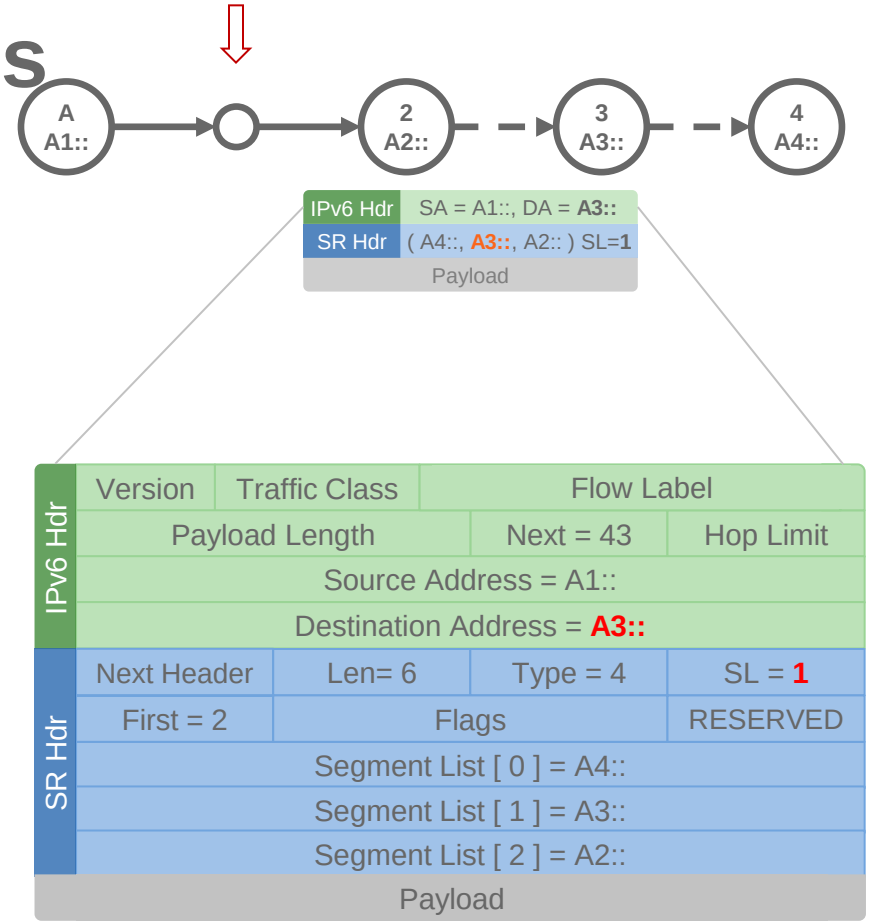
Non-SR Transit Node



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

SR Segment Endpoints

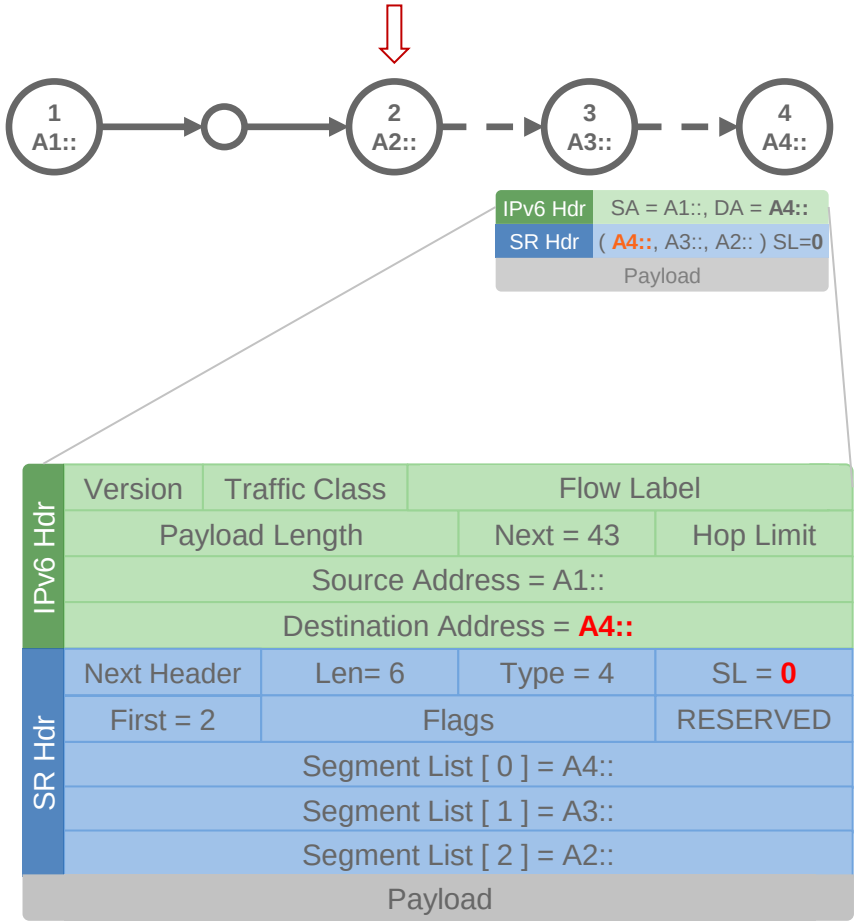
- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA



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
      - …

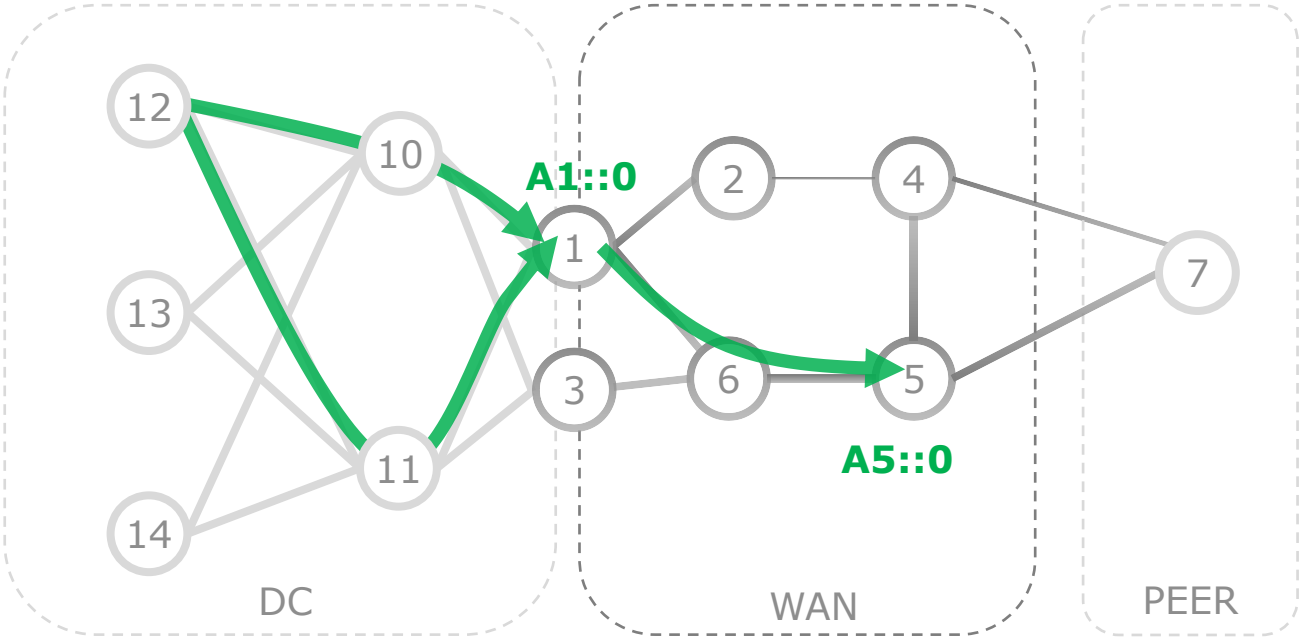
Standard IPv6 processing
The final destination does not have to be SR-capable.



SRv6 Use-Cases

Endpoint

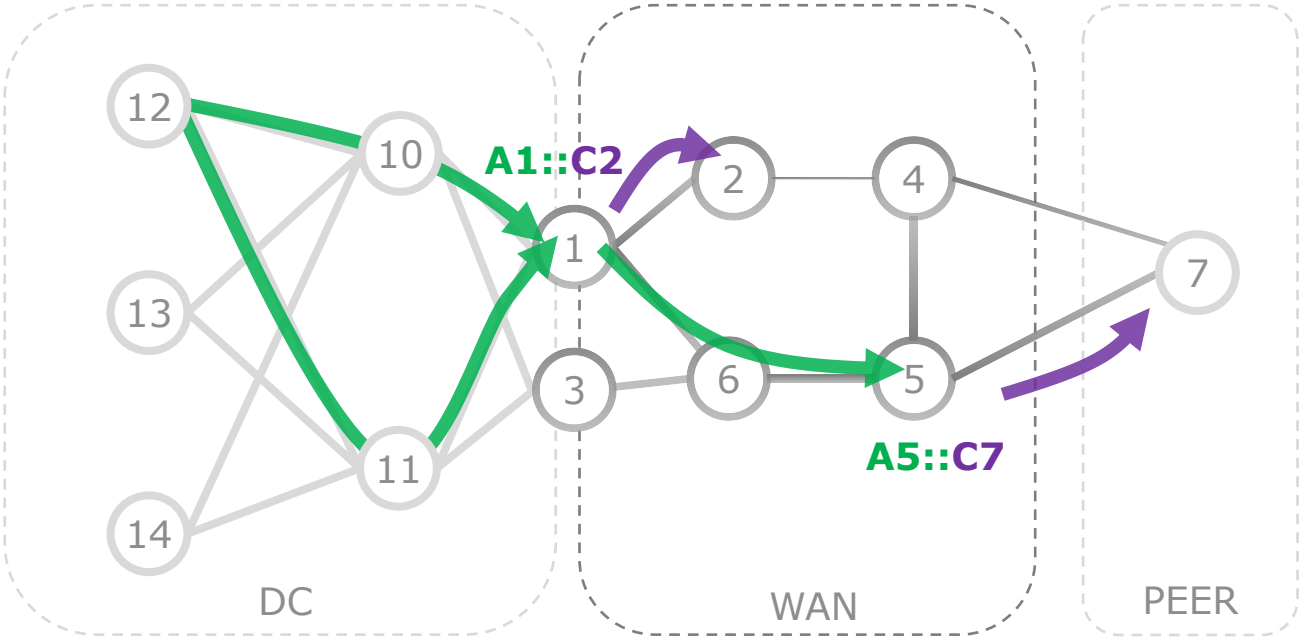
- For simplicity
- Function 0 denotes the most basic function
- Shortest-path to the Node



Endpoint then xconnect to neighbor

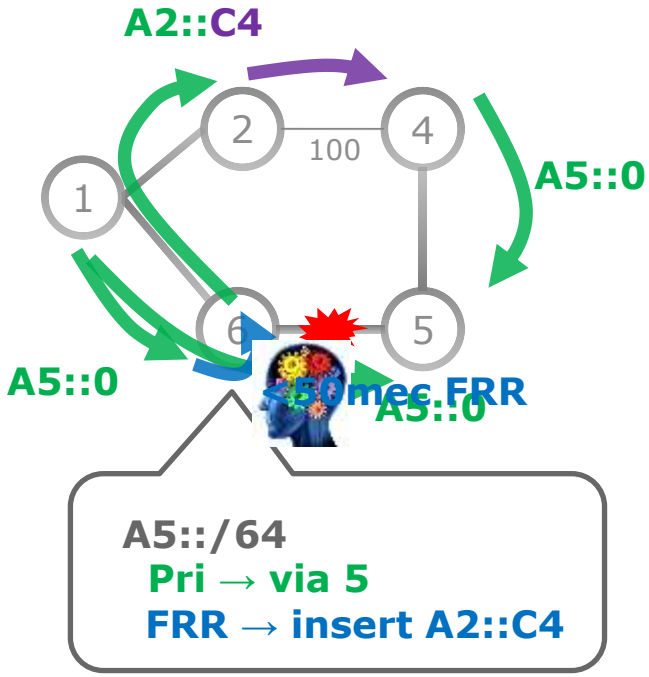
- For simplicity
- AK::CJ denotes

Shortest-path to the Node K and then x-connect (function C) to the neighbor J



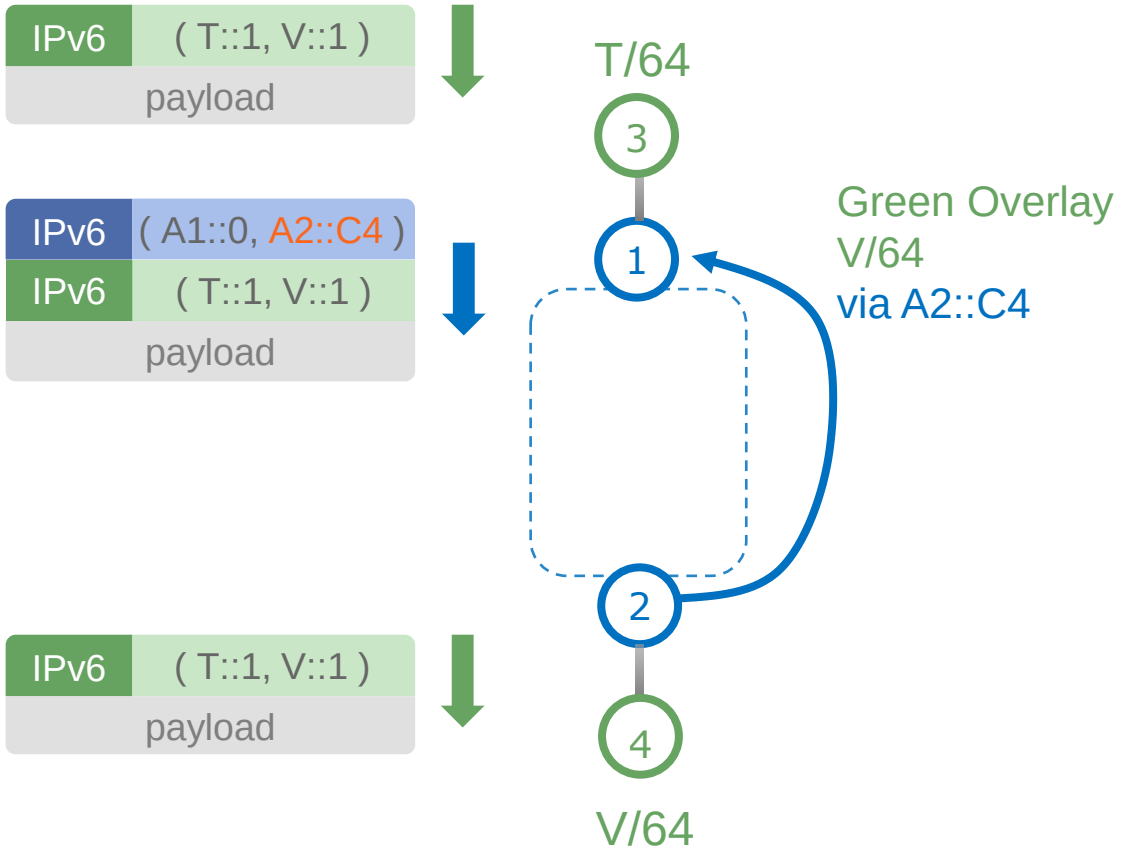
TILFA

- 50msec Protection upon local link, node or SRLG failure
- Simple to operate and understand
  - automatically computed by the router's IGP process
  - 100% coverage across any topology
  - predictable (backup = postconvergence)
- Optimum backup path
  - leverages the post-convergence path, planned to carry the traffic
  - avoid any intermediate flap via alternate path
- Incremental deployment
- Distributed and Automated Intelligence



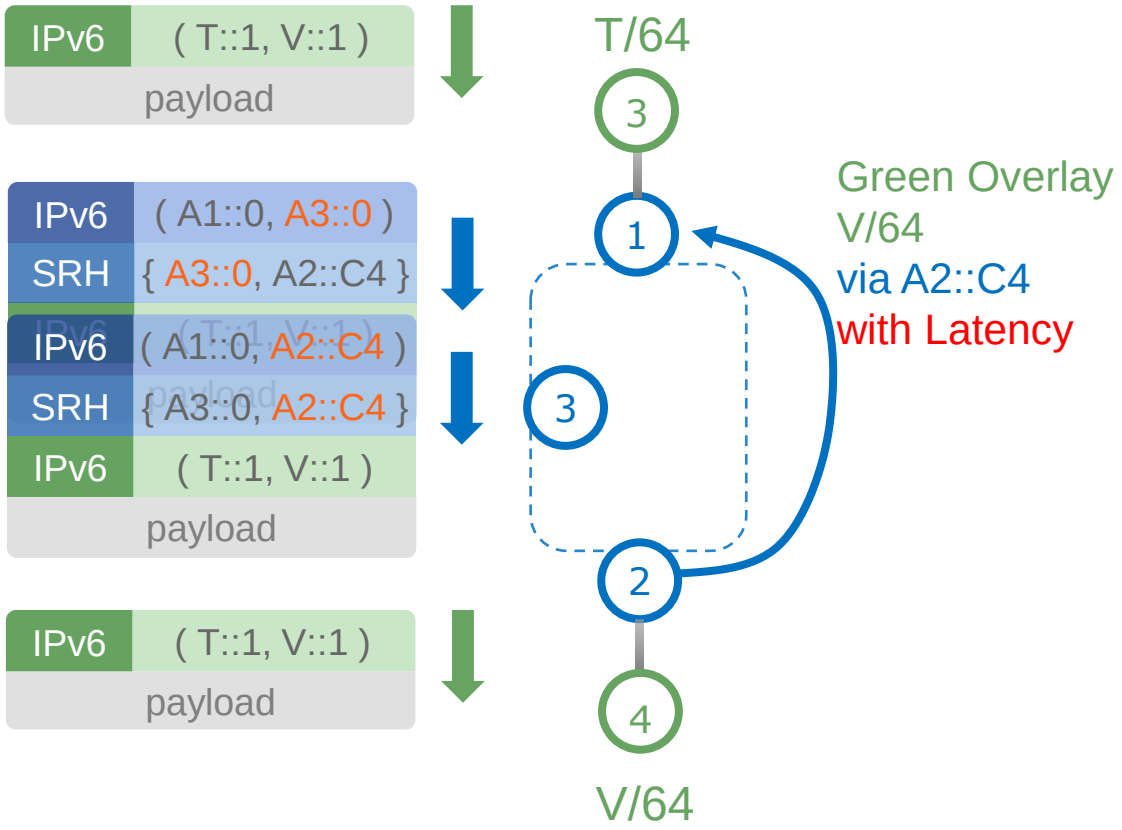
Overlay

- Automated
  - No tunnel to configure
- Simple
  - Protocol elimination
- Efficient
  - SRv6 for everything



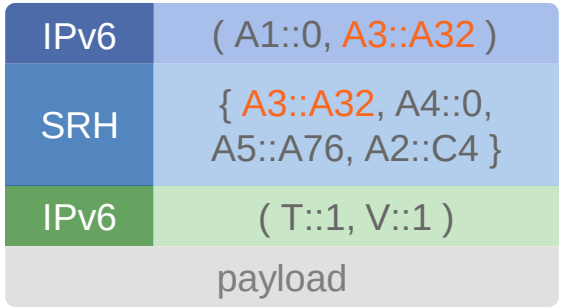
Overlay with Underlay Control

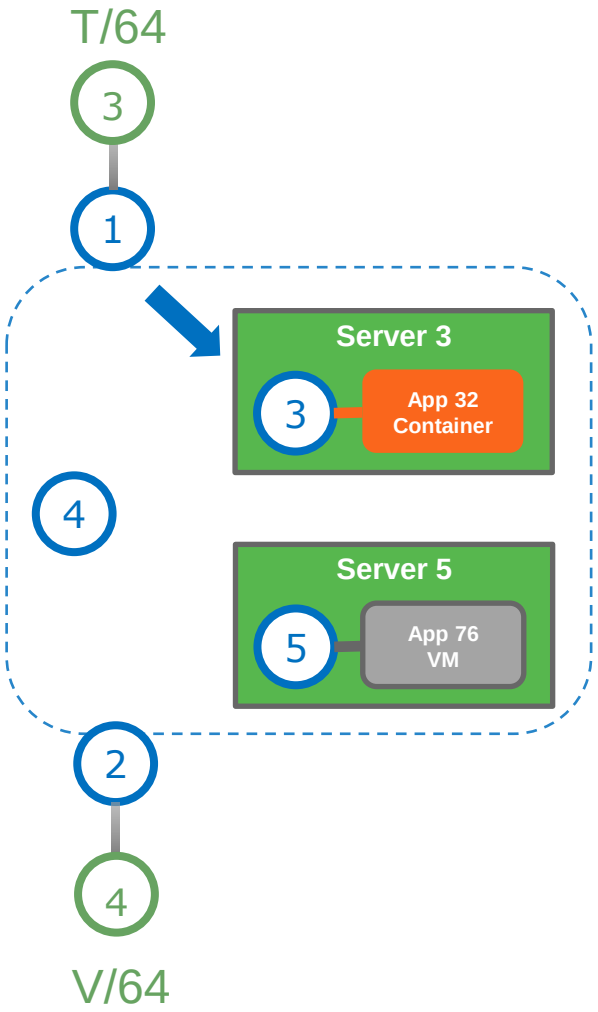
- SRv6 does not only eliminate unneeded overlay protocols
- SRv6 solves problems that these protocols cannot solve



Integrated NFV

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

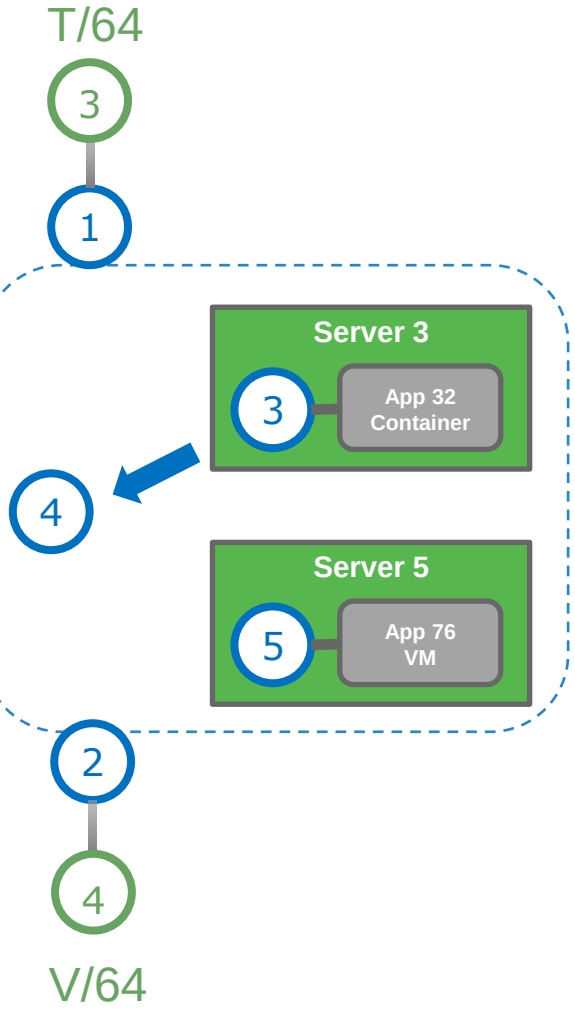




Integrated NFV

- Integrated with underlay SLA

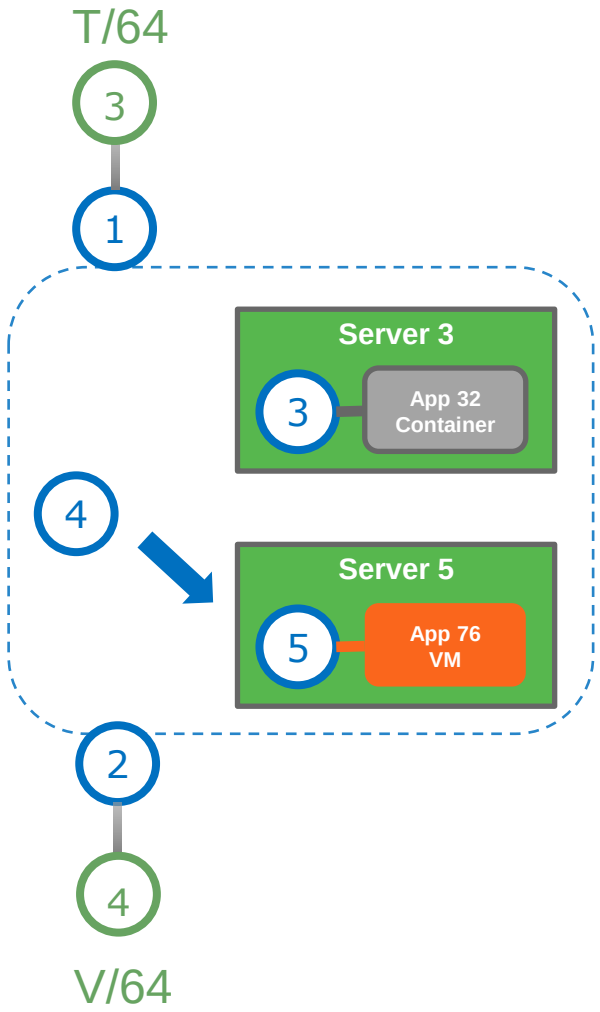
IPv6	(A1::0, A4::0)
SRH	{ A3::A32, A4::0, A5::A76, A2::C4 }
IPv6	(T::1, V::1)
payload	



Integrated NFV

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

IPv6	(A1::0, A5::A76)
SRH	{ A3::A32, A4::0, A5::A76, A2::C4 }
IPv6	(T::1, V::1)
payload	

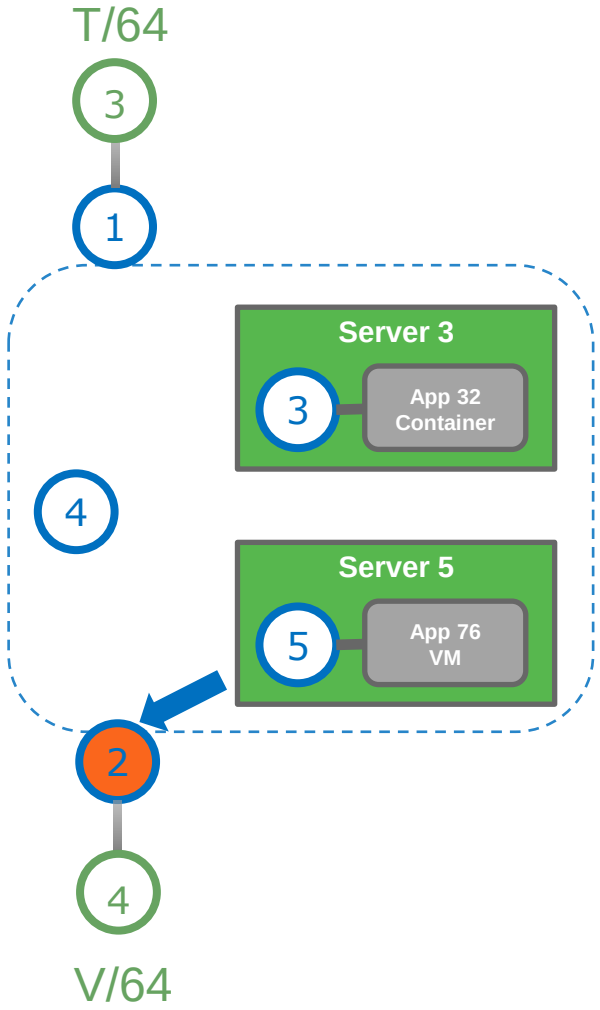


Integrated NFV

- Integrated with Overlay

IPv6	(A1::0, A2::C4)
SRH	{ A3::A32, A4::0, A5::A76, A2::C4 }
IPv6	(T::1, V::1)
payload	

IPv6	(T::1, V::1)
payload	



SRv6 status

- Cisco HW
  - ASR9k - XR
  - ASR1k – XE
- Open-Source
  - Linux 4.10
  - FD.IO

Подводя итоги – нам уже 4 года!!!

- Активная работа в IETF
  - Работа в рамках SPRING WG
  - 25 IETF drafts released
    - Over 50% are WG status
    - Over 75% have a Cisco implementation
  - **First RFC document - RFC 7855 (May 2016)**

Orange, Bell Canada, Deutche Telecom, British Telecom, Comcast, Google, Facebook, Microsoft, Yandex, Alcatel-Lucent, Ericsson, Juniper

www.segment-routing.net

Полный перечень материалов

Вопросы?

Пишите ddementi@cisco.com

TOMORROW starts here.