SDN on ISP w/ SR

MARIO ROSI

Senior System Engineer
MPLS SR/ISP, VXLAN & CISCO ACI Data Center and SD-WAN expert

www.linkedin.com/in/mariorosi71

Agenda

IT complexity and Automation
Transition from Hardware-Centric to SDN
Software Defined Networking (SDN)
SDN on ISPs
Segment Routing
Disjoint paths
Disjoint paths + PCE
Solution SID list, adj-SID and color
Multi-domain topology
Orchestrator via North-bound Interface use

IT complexity and Automation

The IT industry is constantly changing and evolving

*«…every network, to some degree, has inherent **complexity**»*

A **rapid** and **scalable** deployment of network services became a MUST!

The IT operations process needs to be **faster** and **simplify** all the configurations that traditionally gone into networking

*«**Automation** is something that many in the industry are striving for»*

*«A Cisco Technical Assistance Center (TAC) survey taken in 2016, declared that **95%** of Cisco customers are performing configuration and deployment tasks manually.»*

*«The survey stated that **70%** of TAC cases created are related to misconfigurations»*

The necessary improvements, require to the whole model/process to be re-build:

Hardware-Centric networks are more difficult to support due to the box-by-box configurations approach

Transition from Hardware-Centric to SDN

It is important to shift from a **connectivity-centric** architecture to an **application** or **service-centric** infrastructure that focuses on user experience and simplicity

The shift is from hardware and device-centric options to an open, extensible, **software-driven**, **programmable** solution by leveraging the **controller** concept and splitting data plane from control/mgmt plane

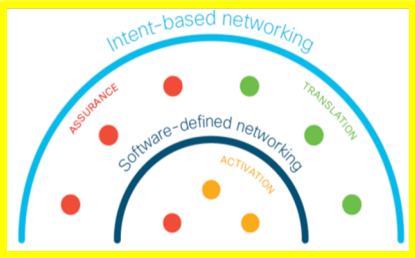
The ***«Intent-based Networking»*** (IBN) became the new *mantra:*

> *"Intent-based networking"* binds both **business purpose** and **network context** through **abstractions**, which are then translated (relying on **automation**) to achieve the desired outcome that satisfy a **business intent**

IBN

> *"...capturing and translating the intent your organization has for your network, then automating the intent, enforcing the intent, and assuring that your network is operating as intended."*

SDN is focused on instantiating changes in network functions



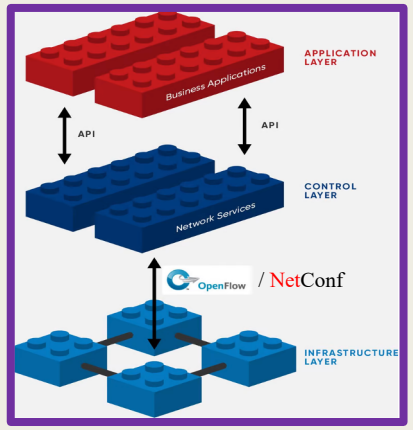
SDN

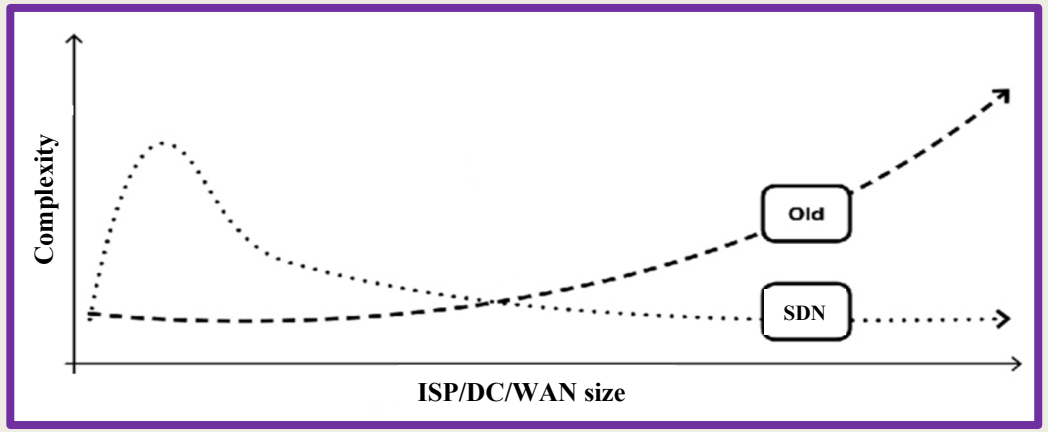
In SDN architecture, **control** and **data planes** are **decoupled**

With SDN, **intelligence** and **state** are **logically centralized** on the **controller**

What is SDN?

> *It's an enabling technology where **physical network layer** is **abstracted** from the **application layer***
>
> *A new method to interact with devices via **centralised controller***
>
> *Enables **high-scale**, **rapid** network and service **provisioning** & **mgmt***



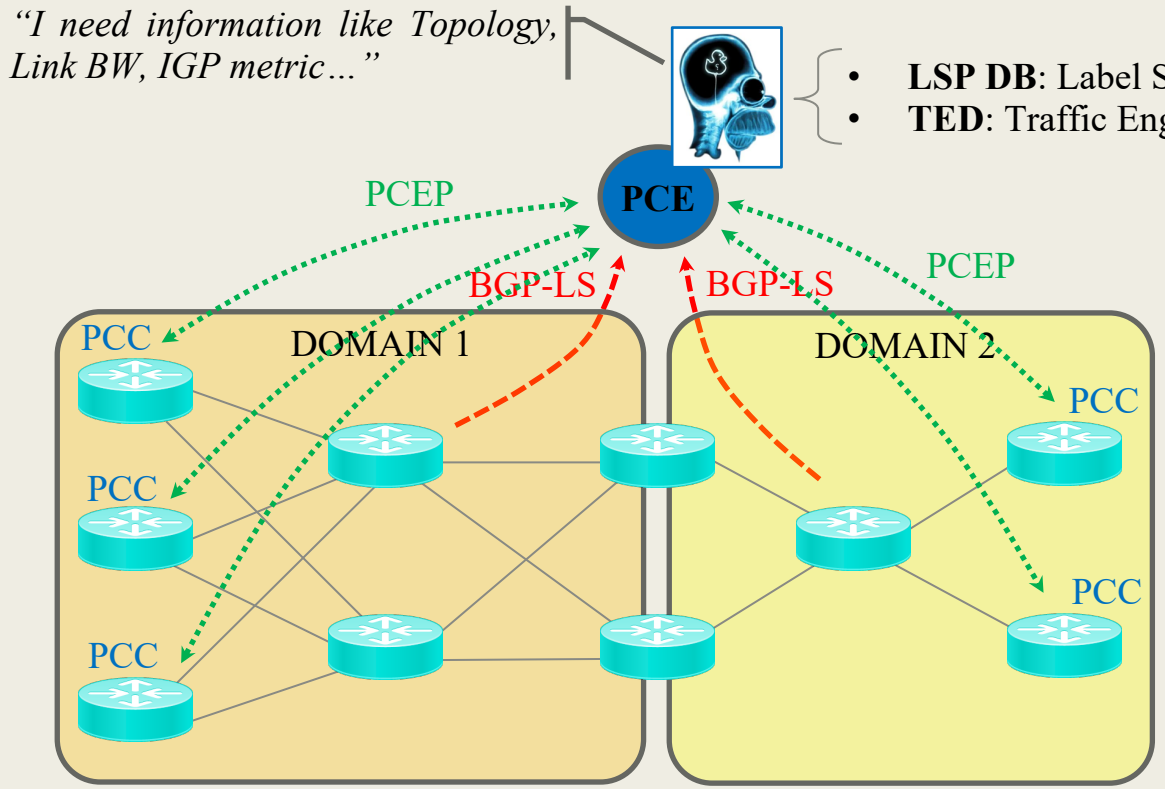


SDN on ISPs

Use cases:

- ✓ **Disjoint paths** with *distinct headends* on MPLS SR environment
- ✓ **Multi-domain topology** (paths traversing *different IGP areas/domains*)

✓ Both cases require a a **centralized solution**, a **centralised controller (PCE)**



- **LSP DB**: Label Switch Path DataBase
- **TED**: Traffic Engineering DataBase

- The *path Computation Element Protocol* (PCEP) is the protocol used by *Path Computation Client* (PCC), the *Head End* router, to delegate to *Path Computation Engine* PCE (SDN controller) the control and definition of head-end label switched paths (LSPs) starting from PCC
- The PCE programs the PCC's LSP defining the SR label stack ("*SR-TE intent*")
- The PCE, as orchestrator, applies *computational constraints,* to get the engineered path basing on requirements (the intent)
- The PCE, uses *Traffic Engineering Databse* (TED) information to do its job

Segment Routing

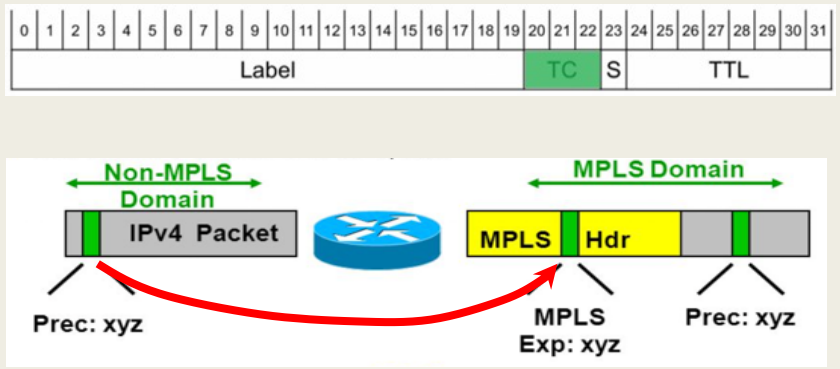
Global Segment: label locally assigned by each router and announced to the other ones (it's unique)

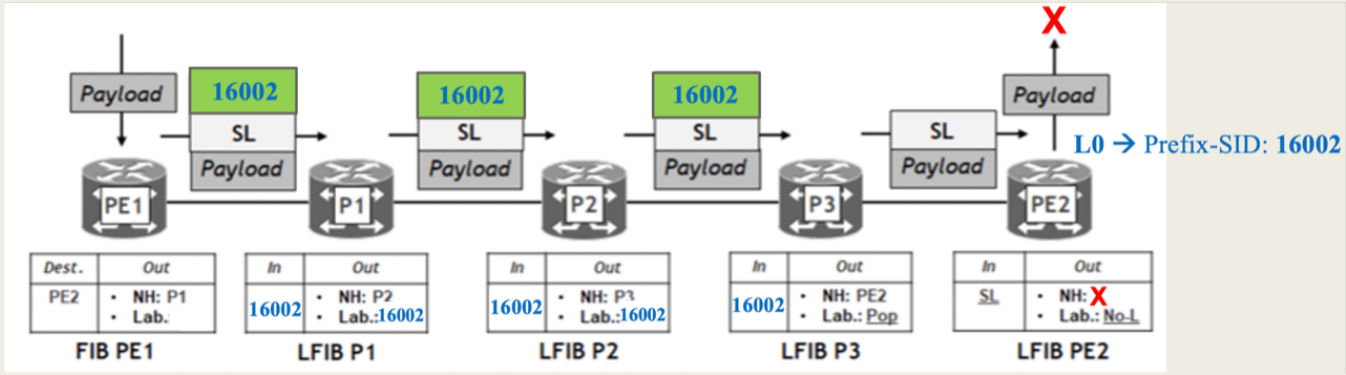
*i.e.: **Prefix-SID** (/32 for IPv4 or /128 for IPv6)* chosen within **Segment Routing Global Block** (SRGB): [base=16.000, size=23.999]) → *"base+Node-SID"*

Local Segment: label locally and dinamically assigned by each router and announced to the others; the router that generates it is the ONLY one that uses it); (can be NOT unique)

*i.e.: **Adj-SID*** (adjacency IGP unidirectional) chosen within the range [24.000-1.048.575]) → *"Adj-SID"*

Traffic forwarding in the SR paradigm use **always** the **same global SID** announced by the destination router (no label swapping as for LDP)





Disjoint paths

The SR is associated to **source routing** concept:

*the source of LSP MPLS path, the head-end (PE), decides the path end-to-end creating the **SR segments list** (the SIDs stack)*

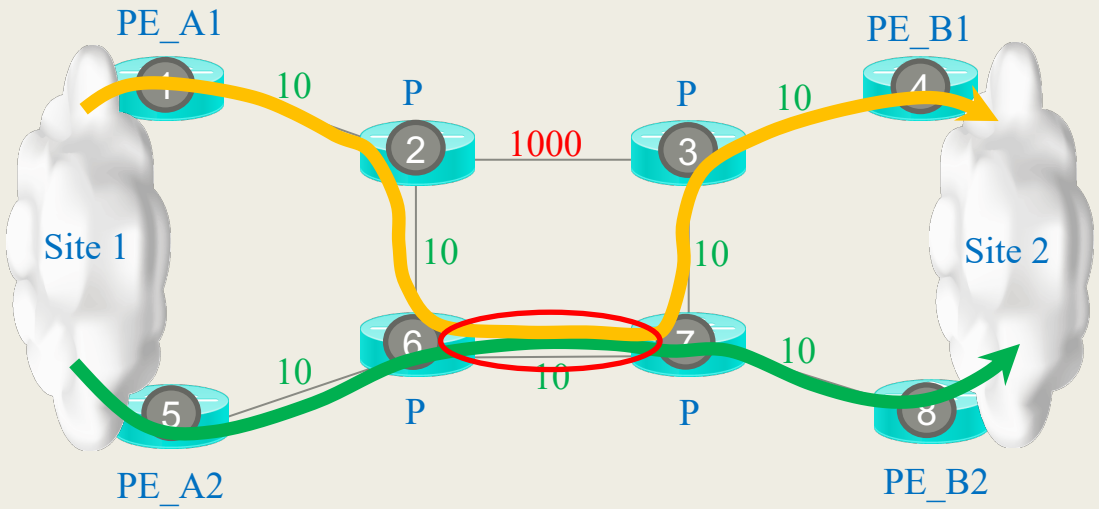
Distinct head-ends, cannot compute the paths on their own ☹:

They only know about their own SR Policy paths

They are unaware of the other head-end's SR Policy paths

It requires a centralized computation engine that is aware of both paths in order to provide disjoint paths

10, 1000: IGP metric



Disjoint paths + PCE

Disjoint group: Paths with the same *disjoint group-id* are disjoint from each other

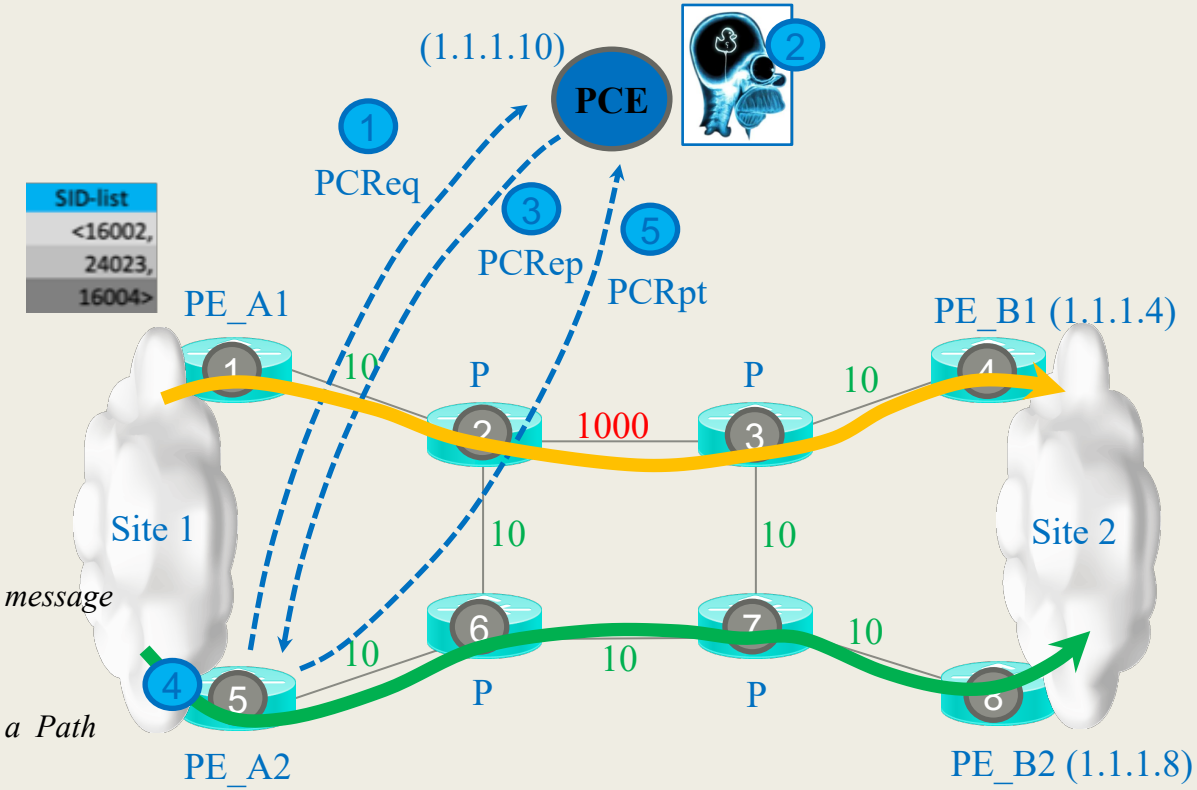
The IT operator indicates which paths must be assigned to the same disjoint group-id. The PCE acquires and enforces these constraints

1. *Path Computation Request (PCReq)*

```
segment-routing
traffic-eng
 policy POLICY1
  color 20 end-point 1.1.1.8
  candidate-paths
   preference 100
    dynamic
     pcep
     metric
      type igp
     constraints
      association-group
       type disjoint node identifier 1
 !
pcc
 pce address ipv4 1.1.1.10
```

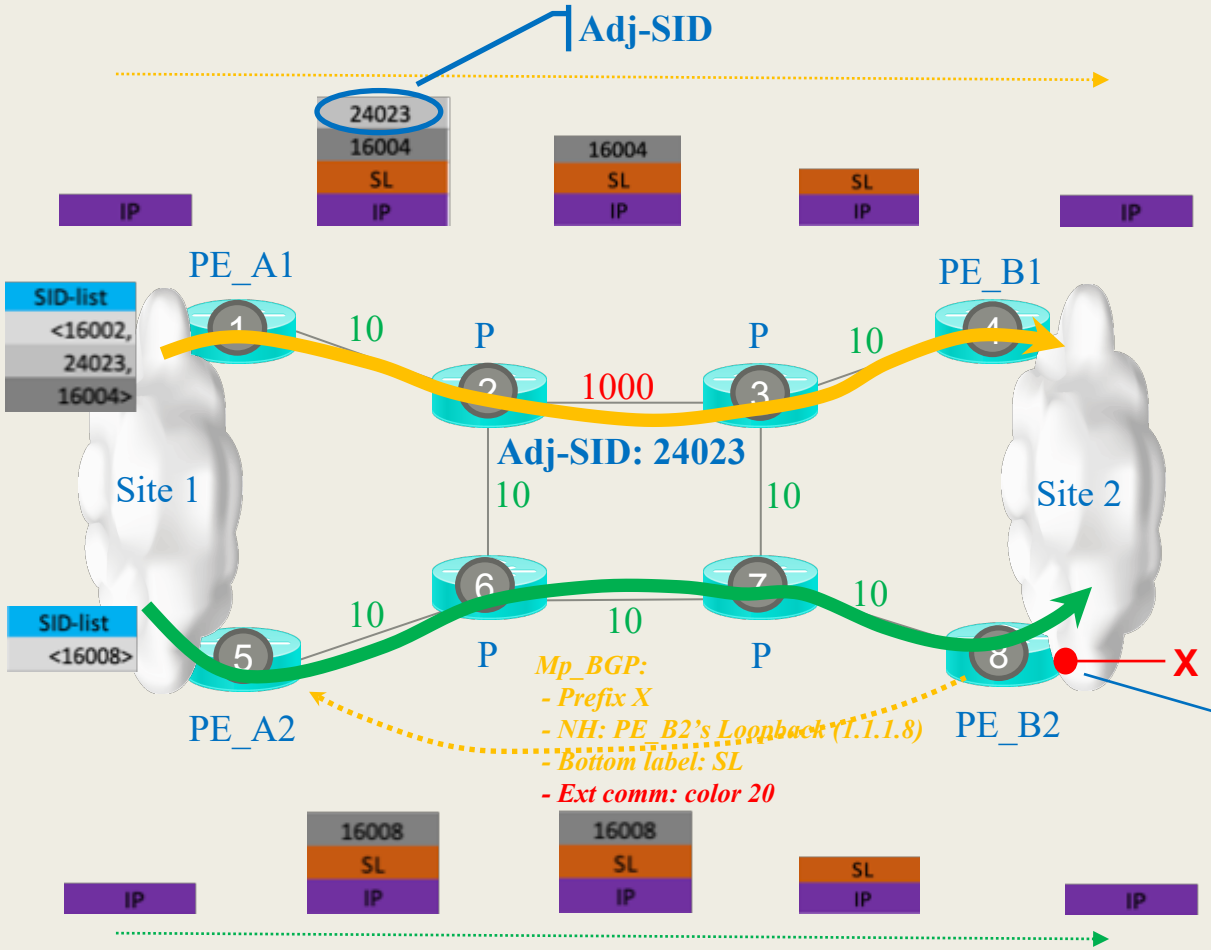
```
segment-routing
 traffic-eng
  policy POLICY2
   color 20 end-point 1.1.1.4
   candidate-paths
    preference 100
     dynamic
      pcep
      metric
       type igp
      constraints
       association-group
        type disjoint node identifier 1
  !
 pcc
  pce address ipv4 1.1.1.10
```

2. *PCE computes the unconstrained I… 2*
3. *PCE send the **solution SID list** <16008> in Path Computation Reply (PCRep) message*
4. *PCC (PE_A2) instantiates the path for the SR Policy to PE_B2*
5. *PE_A2 provides all details about the state of the path to the PCE using a Path Computation Report (PCRpt) message*



Solution SID list, adj-SID and color

SR PCE **autonomously** and **proactively** re-computes the paths and updates them (if required to maintain disjoint paths) for any topology changes.



color: represents an intent (low-latency, high bandwidth…)

```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point 1.1.1.8
   candidate-paths
    preference 100
     dynamic
      pcep
      metric
       type igp
     constraints
      association-group
       type disjoint node identifier 1
 !
 pcc
  pce address ipv4 1.1.1.10
```

Setting in ingress of color

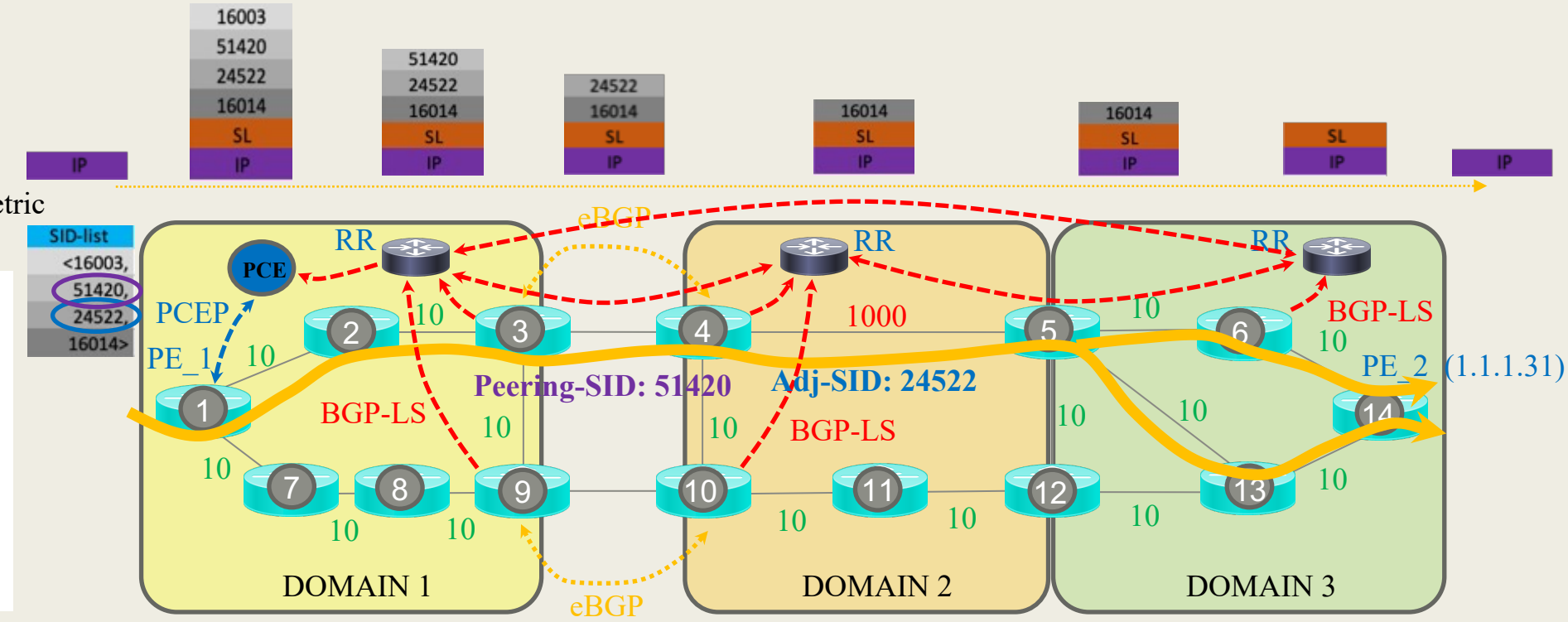
Multi-domain topology

Computing paths crossing different IGP areas/domains requires **knowledge** about **all** these **IGP areas/domains**

An head-end node cannot compute inter-area and interdomain paths

It has ONLY knowledge about its local IGP area/domain

It requires a **centralized computation engine** that is aware of inter-area/domain topology




```
segment-routing
 traffic-eng
  policy POLICY1
   color 20 end-point 1.1.1.31
   candidate-paths
    preference 100
     dynamic
      pcep
      metric
       type delay
  !
  pcc
   pce address ipv4 1.1.1.10
```

Orchestrator via North-bound Interface use

PCE, provides a unified interface to the network and the real-time topology enabling the third-party applications to **manage SR Policy status** via **REST API** instantiating, updating, and deleting SR Policy candidate paths

Once the application has collected the network information, it can compute the required SR Policy paths and deploy them using the PCE's *north-bound* interface. The PCE then initiates these paths via its *south-bound* PCEP interface.

1) PCE receives the path creation request from the application via API
2) PCE sends a PCEP *Initiate* message with the Create flag (C-flag) set
3) Head-end Node1 installs the SR Policy path in the forwarding table
4) Head-end Node1 sends a status report to PCE in the PCEP Report (PCRpt) message (it confirms that the path has been installed as instructed and delegates control to the PCE)
5) PCE stores the status information in its database and feeds the path information to the application via its North-bound interface