1 Another blog about Segment Routing

Written by **Diptanshu Singh**, working link:

https://packetpushers.net/yet-another-blog-about-segment-routing-part-1/

(there's many other well written posts at the above link)

Chapter 2 has been entirely written by me (Riccardo Andreetta)

In this blog post we will be taking a deeper look at Node/Prefix SID and SR/LDP Interworking. If you work or have an interest in the Routing area, then by this time you may have already heard about Segment Routing (SR) and I am assuming that you already have some awareness with the basic concepts of Segment Routing. There is plenty of material available over the Internet, which gives a good overview of Segment Routing and its use cases. For some reason if you just came out of the cave and weren't aware about it, I recommend you starting from:

http://www.segment-routing.net/

So we know that Segment routing (SR) is a source routing concept in which an ingress node selects a path and encodes that into the packet header as an ordered list of segments. From a control plane perspective IGP's (ISIS or OSPF), BGP, PCE-P and BGP-LS are being extended to support Segment Routing. There is no need to have LDP or RSVP to distribute the labels anymore in a pure SR domain. From a data plane perspective, we can use MPLS labels where MPLS labels represent a Segment ID (SID) or IPv6 (with option segment routing header). In this post we will be focusing on ISIS for control plane and MPLS from a data plane perspective.

1.1 Acronyms:

SR: Segment Routing

SID: Segment ID

SRGB: Segment Routing Global Block

1.2 Node/Prefix-SID and SRGB

The first major type of SID is Node-SID or Prefix-SID. A Node-SID represents a Node and has a **global significance**, something like a loopback of a router. Like an operator assigns a loopback's to their routers, it's expected that the Node-SID value will be assigned to every node. The assigned value can be an absolute or Index value and must be globally unique.

All the nodes, switch packets towards Node-SID via the shortest path. From **global significance** we mean here that every node in the SR domain will install an MPLS label in the forwarding plane representing Node-SID. Okay, so how is this different then every router installing a label for /32 loopback's in the case of MPLS+LDP domain? Let's wait till the end of this section to see if there is any difference.

In the Fig.1 , each Router has been assigned a unique Node-SID value. IGP metrics are highlighted in Blue. The idea here is pretty simple, if R1 wants to send a packet to R6, it will slap the Node label (16006) of R6 on the top of the packet and every router on the shortest path will swap the labels till it reaches R6. If you want to send a packet through specific Node before it reaches its destination like R1 wants to send data to R6 via R4 then it can slap Node-SID label of 16006 and 16004,16004 being at the top of the stack. Once the packet reaches R4, R4 sees the label value of 16006 sends the packet towards R6.

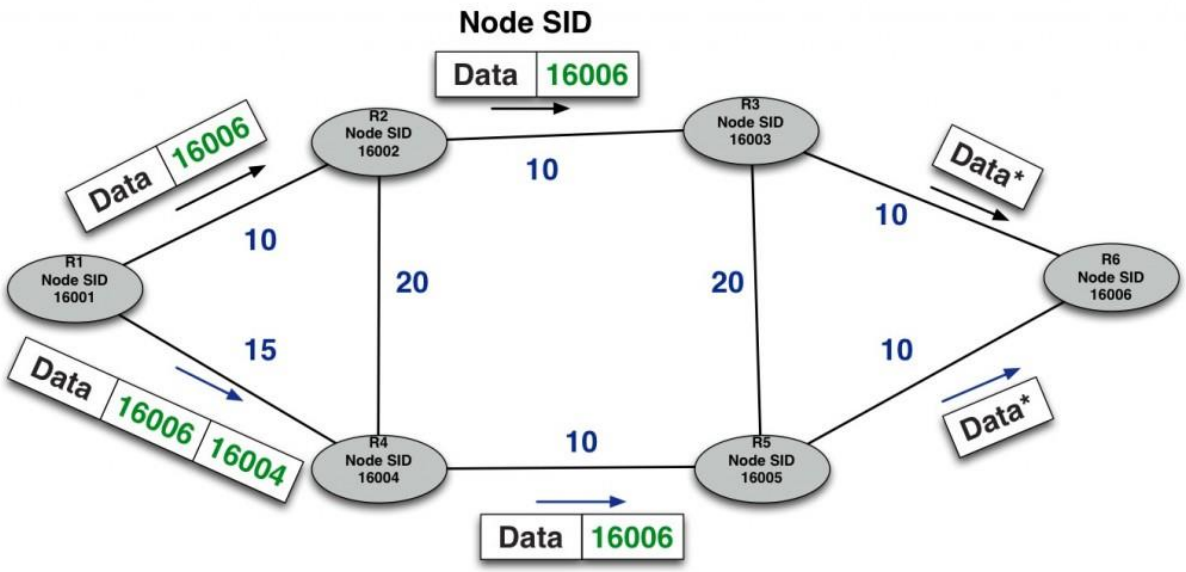


Figure 1 Example of label swapping in a Segment Routing environment.

Pretty simple, isn't it. But how does every router install the same label value in the forwarding plane? Or is it always the case?

1.3 Segment Routing Global Block

Segment Routing Global Block (SRGB) is the range of labels reserved for segment routing. SRGB is a local property of a segment routing node. In MPLS, architecture, SRGB is the set of local and global labels reserved for assigning labels to global segments like Node-SIDs originated by a router. In IOS-XR default SRGB range is from 16000-23999 and Dynamic label range is 24000-1048575. Every SR node advertises its SRGB as label base and label range. So in the case of IOS-XR it will be Base=16000 and range = 8000 via IGP. It is possible that different nodes can have different SRGB values either because the operator configured so or due to vendor implementation differences and hence **the absolute Node-SID** value associated with an IGP Prefix Segment **can change from node to node**.

So far we have established that there is a reserved SR (SRGB) block at every node and allocations of labels for Prefix/Node-SID value happens from that block.

In an SR domain if SRGB range of all the nodes is same then all the SR nodes will install the same SID label value for a node or prefix. So for instance, in the below Fig.2, I can assign an absolute value of 104 to R4 and R4 advertises that via an IGP, When R3 receives this information, It reserves 104 from its SRGB block for R4. This information gets further relayed to R2 via IGP.

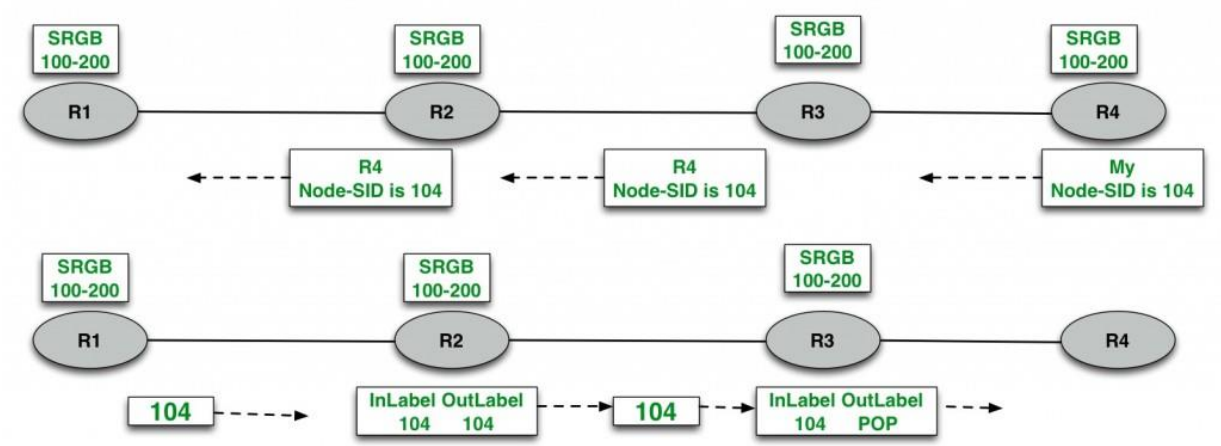


Figure 2 Label distribution through ISIS and label swapping.

But what if the SRGB are different for different nodes. In the below fig.3, If I assign an absolute value of 104 as Node-SID to R4 and when R4 advertises this via an IGP, R2 can't not assign 104 as it's outside of its SRGB range. So how do we solve this situation? In order to avoid a situation like this we have an option to assign a Unique Index value than an Absolute value to a Node-SID.

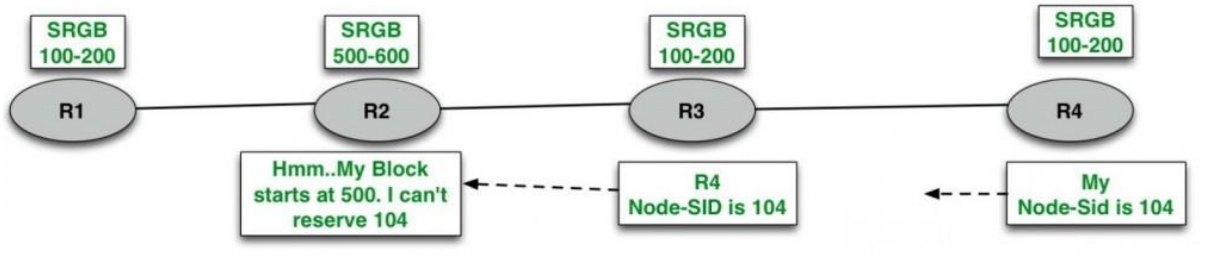


Figure 3 R2 can't assign label 104 to reach R4, since its SRGB range is 500-600.

In the below Fig.4, The operator has configured Unique Index values to each SR Node. When R4 advertises its Index Value to R3, then R3 programs 104 as a local label. R3 further advertises Index value of R4 to R2, R2 assigns its local label 504 (500+4) based on the below formula

Local Label (Prefix SID) = Start label + SID Index

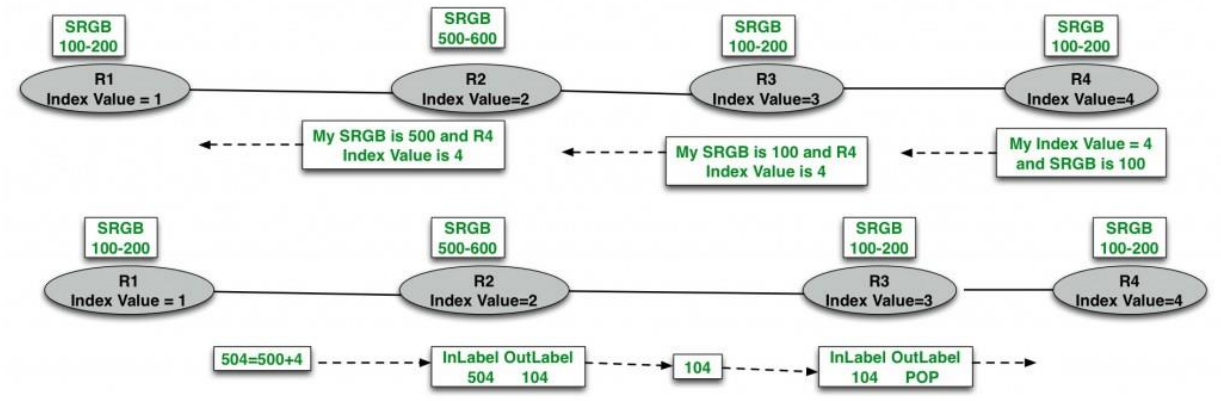


Figure 4 Routers advertise through ISIS their base SID and relative SID.

Now, as you can see that the label operation looks very similar to LDP when operating in the independent label distribution mode with the difference that the label value used to forward a packet to each downstream router is computed by the upstream router based on the advertised prefix SID index using the formula Local Label (Prefix SID) = Start label + SID Index.

1.4 Adjacency-SID

The other most important SID is the Adjacency SID. They usually represent a link and have a local significance. What I mean by local significance is that the remote SR node, unlike for Node-SID doesn't install a state in the forwarding plane for an Adjacency SID. Only the directly connected Nodes program the forwarding plane for Adjacency SIDs. They are absolute values (Not Index) and allocated dynamically by the router from the dynamic label range, which is outside of SRGB block. So in the case of IOS-XR they will be allocated from the Dynamic label range i.e. 24000-1048575. Since Adjacency SID's has local significance they don't have to be unique in the SR domain. For instance, in the below Fig. 5, 24001 Adjacency SID is present between R1- R4 and R3-R5.

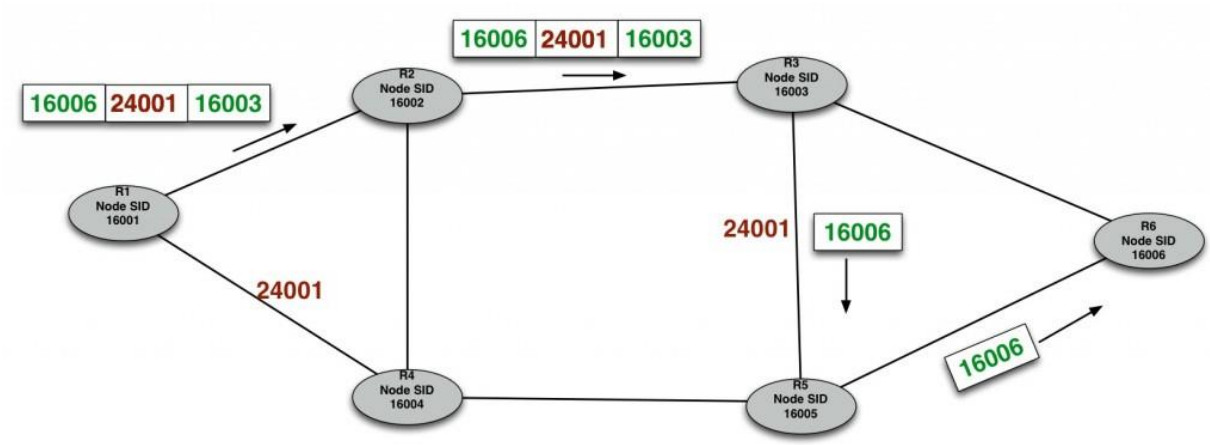


Figure 5 R1 wants to follow the path R3-R5-R6, thus uses adjacency label 24001.

1.5 SR/LDP Interworking

One other thing, which I wanted to cover in this post, is about SR/LDP interworking. For a technology to be successful, it's very important that it doesn't require a forklift upgrade and provide a migration/interoperability with current systems. So If an operator is already running an LDP network and wants to implement SR but not every node is SR capable which could be due to various reasons like those nodes are from different vendors and don't support SR yet, or they may not be running the SR capable operating system. SR/LDP interworking could be an option for him.

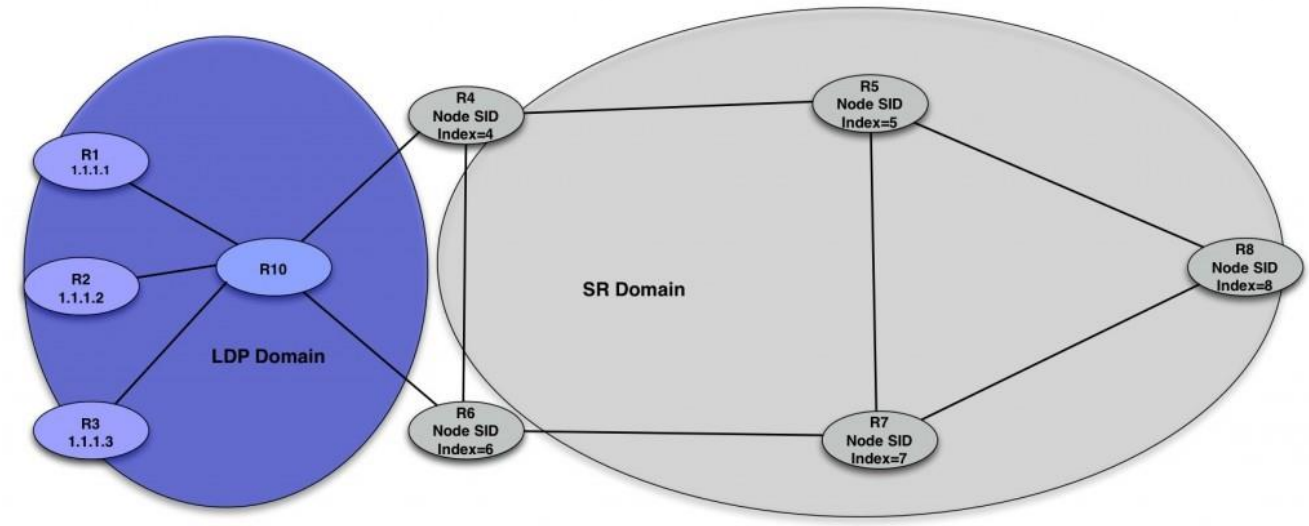


Figure 6 Mapping agents to pass from LDP domain to SR domain.

Let's look at an example. In the below Fig.6, R1, R2, R3 and R10 (In Blue) are only running LDP and R4-to-R8 Routers are SR capable Routers. R4 and R6 will be running both SR and LDP as they are on the border between SR and LDP. The whole network is running single IGP let's say IS-IS.

From LDP to SR Domain

So if a router from LDP only domain wants to communicate with SR domain, it's pretty straightforward. Let's say R1 wants to send a packet to R8.Since R4 and R6 are running LDP, they will allocate labels corresponding to the SR Nodes including R8 and advertise it to other LDP neighbors. When R1 sends the packet with label advertised by the LDP to R10, R10 swaps to the label towards either R4 or R6 (based on shortest path and in the example it's R4). R4 upon receiving the LDP label swaps to the SR label corresponding to the R8.So this part was easy.

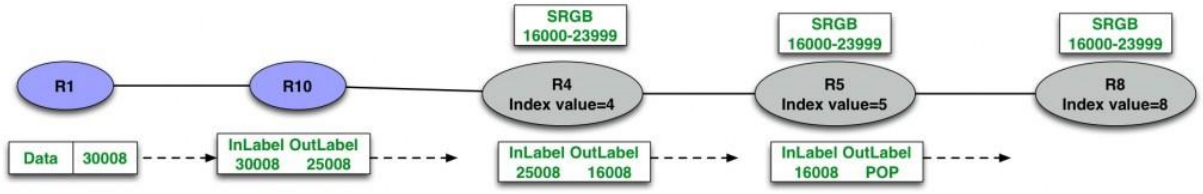
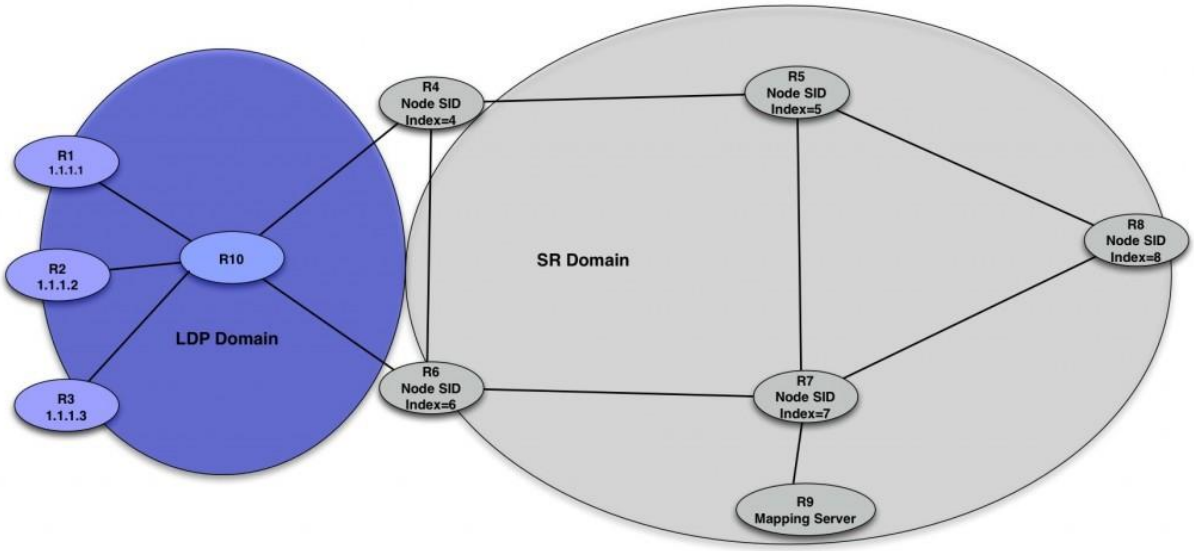


Figure 7 Switching from label 'ldp' domain to SR domain.

From SR to LDP Domain

Now In Fig.6, if R8 which is SR only node, wants to send a packet to LDP only nodes then we have a problem. Since R8 doesn't know the Node-SID corresponds to R1 it can't send a packet to R1 (or other LDP nodes). In order to solve this problem, Mapping server functionality was created.

In the below Fig.8, I am showing a separate mapping server node for one reason that it doesn't have to be in line as long as its participating in the IGP domain, otherwise any SR node can be a mapping server and doesn't have to have a separate box. But obviously it will be a crucial box as it failure means the failure of communication between SR and LDP domain.



On the **mapping server**, we define the Node-SID and their corresponding label value mapping something like this:

Prefix	Index Value	Range
1.1.1.1/32	2001	3

Where 1.1.1.1 /32 is the starting address of the loopback, 2001 is the starting label Index Value and range is 3.Mapping server will advertise this mapping to all the SR Nodes via IGP.Assuming SRGB range is 16000-23999 then Node-SID values derived by SR Nodes for LDP routers will be

1.1.1.1 /32 Node-SID label value is = 18001 (SRGB Base 16000+2001)

1.1.1.2 /32 Node-SID label value is = 18002 (SRGB Base 16000+2002)

1.1.1.3 /32 Node-SID label value is = 18003 (SRGB Base 16000+2003)

This way now SR Nodes will know what Node-SID value to use if they want to reach an LDP only Node.

In the below fig.9, Once the packet reaches at R4 which is on the border of SR and LDP, it will swap the label from the SR-to-LDP Label which was advertised by R10 for R1.

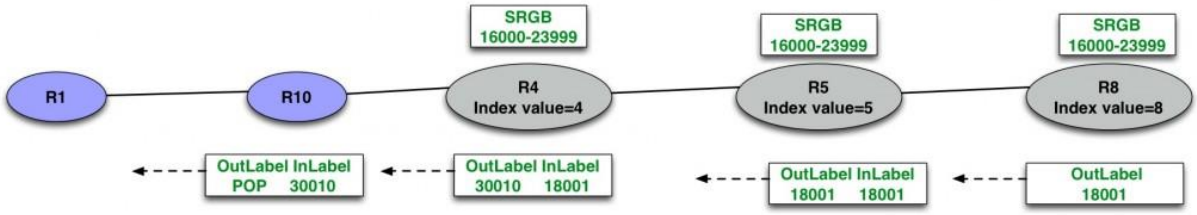


Figure 8 From SR domain to LDP domain.

2 Protection mechanisms

In service provider environments convergence times required by voice applications are usually quite low. Convergence times below 50ms are not easy to be obtained for every link/node failure, and even if they were, this came at the expense of manually complex configuretions, difficult to be maintained during time, with consequences on troubleshooting, number of traffic-engineering tunnels, signaling states in the core. In many cases all you did was link protection through FRR (fast reroute) to have a good balance between good convergence times and configuretion complexity.
Let's also consider that through simple optimizations and fine tuning of the IGP protocol, convergence times were usually always kept below 300ms even without using FRR or traffic-eng tunnels. For this reason, as far as I know there are even BIG telcos who decided that such a time was fine, and didn't configure any kind of 'fast protection' mechanism in their core network (transporting also VOICE services).

2.1 Convergence times

To achieve better convergence times without configuring traffic engineering tunnels, in the beginning LFA was standardized (RFC 6571). Later on came 'Remote LFA' with RFC 7490. Understanding them is important to better understand how Segment Routing solves such problems for any possible topology (thus the name **TI-LFA** or Topology Indipendent LFA).

2.1.1 LFA

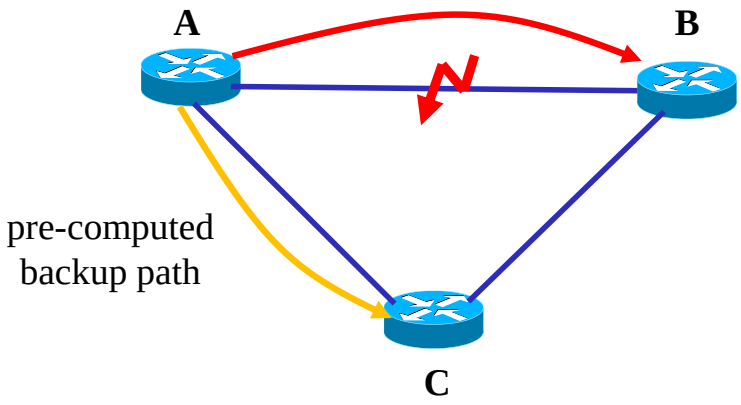


Figure 9 Router 'A' pre-computes the backup path in case of A->B link failure.

As you can see in Figure 9, LFA consists in having an already prepared backup path in case of a failure, using it immediately after the failure itself, without having to wait for the IGP convergence time in the whole network. The problem of such an approach, is that it doesn't work with any network topology, this is why they gave it such a name: it's 'loop free' because you can't always find a loop free alternate path.
In Figure 10 we have a square topology, in normal conditions 'A' routes traffic for 'B' through the direct link. Router 'C' could balance the traffic toward router 'B' in case all links have the same cost, or could even route all traffic for 'B' toward 'A' in case of links with different costs. To solve such a problem and cover a higher number of network topologies (say from 40% - 50% coverage with LFA to 80% - 90% coverage with remote lfa) 'remote LFA' kicks in.

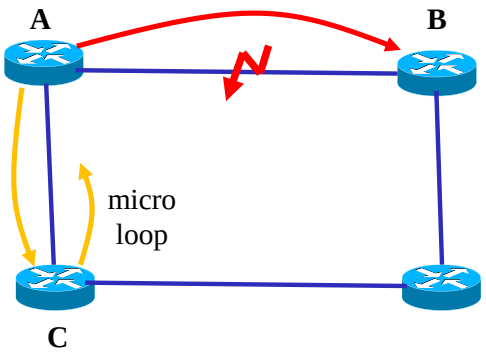


Figure 10 Router 'A' pre-computes the backup path in case of A->B link failure, but a loop occurs.

2.1.2 Remote LFA

In case of the failure depicted in Figure 11, R1 needs to reach R6. There was a link failure, and R1 can't simply send traffic to R2 because a loop would occur. Remember that we're always talking about **the few milliseconds after the link failure**, to easily achieve convergence times of less than 50ms. If you wait 300ms, all the nodes in the network will recalculate the path and there would be no more traffic loss.
So R1 establishes a multihop ldp session with R3, so that R1 knows:

- which label R3 assigned to its own loopback (for example 25)
- which label R3 has associated to R6 loopback (for example 75)

When the failure occurs, R1 puts labels 25 and 75 over the packets, and sends them to R2. R2 know that label 25 is associated to R3, so it removes the label (PHP or penultimate hop popping) and sends the packet to R3. R3 receives a packet with top label equal to 75, and performs a standard operation: it swaps the label and forwards it to R4, because it's the next-hop to reach R6.

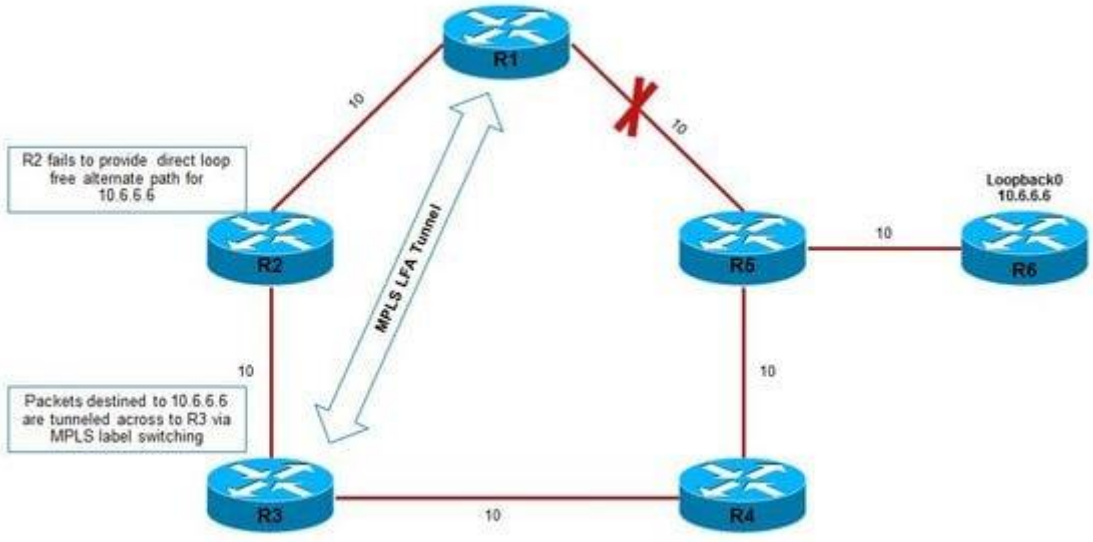


Figure 11 Router 'R1' pre-computes the backup path in case of R1->R5 link failure.

This is just an example to make you understand how 'remote LFA' works, it's not a general approach to describe an algorithm. This can be found in RFC7490 and goes through the definition of different spaces of nodes, respect to a specific link to be protected:

- a ‘P’ space of nodes, reachable from the local repairing node without using the broken link
- ‘Q’ space of nodes, routers reaching the destination without using the broken link
- a ‘PQ’ space of nodes that belong to BOTH the above spaces. PQ is the space of nodes toward multihop ldp sessions can be established to protect the broken link. This space could sometimes be EMPTY, this is why this solution doesn’t cover 100% of the network topologies

Let’s consider the following example, also to better understand the spaces definition of the nodes above. Let’s suppose that R2-R1 is the link to be protected with remote LFA, then the list of nodes belonging to the ‘P’ space, is the list of nodes that R2 can reach WITHOUT flowing through the link to be protected (when the link is up and running), thus without flowing through R1. The list is R3, R4 and R5. The term ‘Extended’ includes also R2 direct neighbors, since they can be reached by R2 despite of the link’s cost.

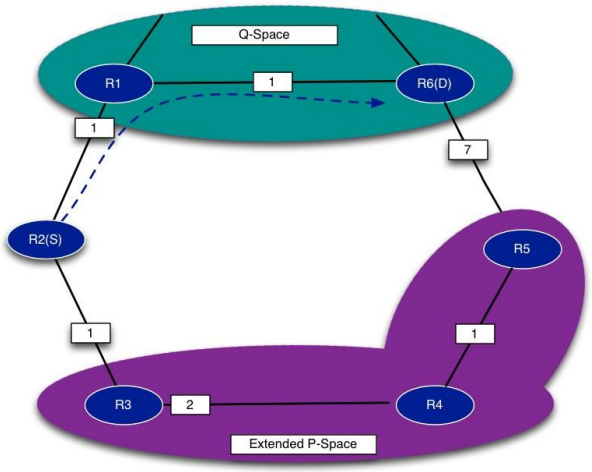


Figure 12 ‘P’ and ‘Q’ space of nodes.

To the ‘Q’ space of nodes, belong the nodes that reaches R1 without flowing through the link to be protected, thus to this space belong only R1 and R6. Since the ‘PQ’ space is empty, this specific topology can’t be covered with ‘remote LFA’, because R2 has no way to force R5 to use the high cost R5-R6 link to avoid micro loops during IGP convergence time.

Statistically it looks like ‘remote LFA’ covers around 90-95% of core network topologies, let’s say that the more a network is redundant and interconnected, the more it is covered. Squares are often used on ISPs as a good trade-off between costs and redundancy.

2.1.3 TI-LFA

TI-LFA seems still to be under standardization:

https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-06

With this algorithm that leverages Segment Routing, is possible to cover 100% of network topologies. As depicted in Figure 13 it is possible to cover link/node failures and **SRLG** links (‘Shared Risk Link Group’). In the latter case, we’re talking about multiple layer 3 links that share the same layer 1 paths. Backup paths are all pre-calculated and respect the imposed costraints. Since the ‘ingress’ router or node locally repairing the failure can decide the full-path to be chosen by the packet, all the micro-loops that has been previously described can be avoided. In the example of Figure 12, in case of a

failure of the link R2-R1, R2 will push a prefix-SIS label for R5 and the adjacency label to use link R5-R6, despite of the high cost of the link that would prevent R5 from using it in normal conditions.

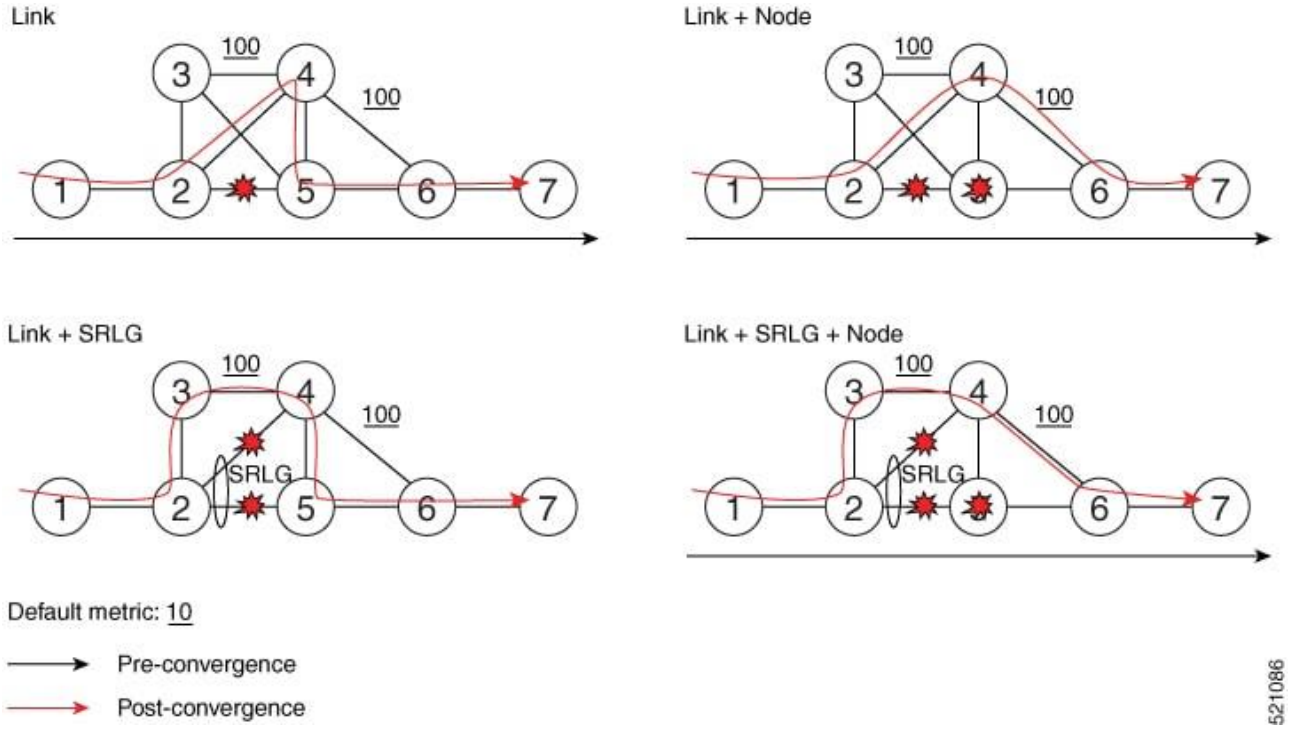


Figure 13 Different protection types with TI-LFA.

One other parameter is that of having the backup path using a different line-card of the router, we could think of it as a different scenario but similar to that of the SRLG. Some example configurations are the following:

```
router isis
 fast-reroute per-prefix level-2 all
 fast-reroute ti-lfa level-2
 fast-reroute tie-break level-2 linecard-disjoint 100
 fast-reroute tie-break level-2 node-protecting 200
```

We talk very much about automation in these days, this is not usually considered automation, but in my opinion it's the BEST possible automation. With a simple command you do a lot of complex stuff, that was previously achieved through a lot of complex configurations.

2.1.4 Micro loop avoidance

This feature can be used to further improve the convergence time of the network, even in case TI-LFA is configured. Consider the example of Figure 14, in case of the depicted link failure scenario, R2 can protect the traffic using SR, if the traffic still reaches this node. But convergence time is a matter of milliseconds, just like the spreading of the information regarding the failure scenario, thus node S could recalculate the 'shortest path first' and re-route the traffic toward the new path BEFORE this is done on R3 and R4. For this reason, traffic could loop for a few milliseconds on the path that would be reached after the failure, when ALL nodes in the network has converged.
By configuring the above command, S can use a similar approach to that of R2 to safely forward the traffic to R3: push a couple of labels, prefix-sid for R4 and adjacency link label to use R4-D link. This will be done for a configureable temporal period, like for example 5 seconds, after which we can be sure that all the network has converged and we can go back to normal forwarding.

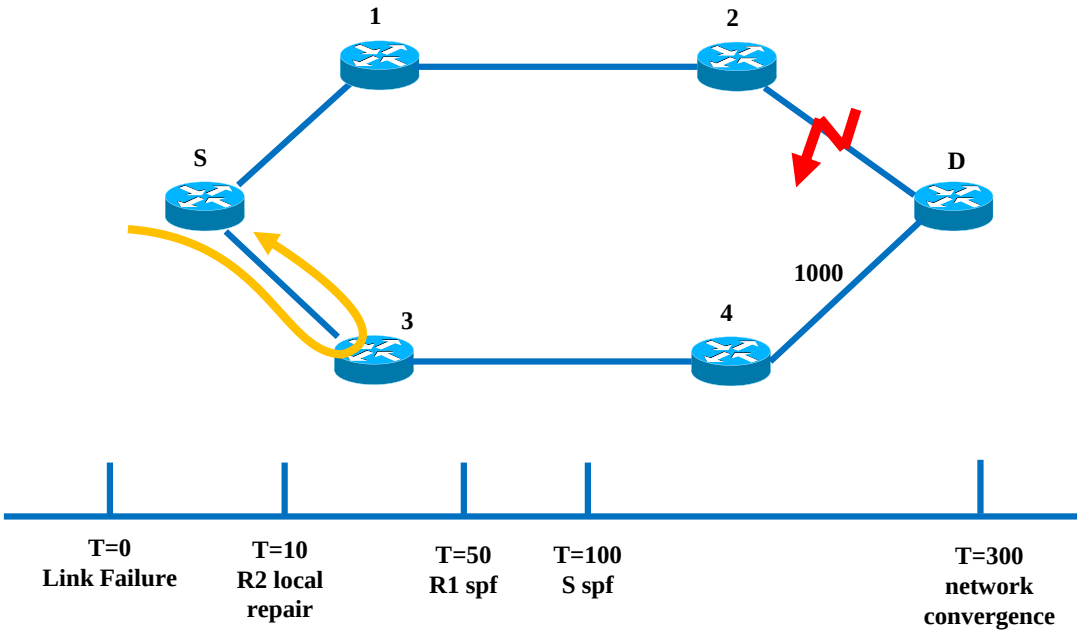


Figure 14 Microloops could occur even if TI-LFA is configured.

2.2 SDN

Segment routing is good to cooperate with an external element of type PCE ('Path Computation Element' RFC 4655), that has the full vision of the network even in case there are DIFFERENT domains and IGP protocols involved. A PCE can be a router or a physical or virtual server, the client and the server communicate through the PCEP (Protocol being the last 'P'). The architecture in this case is based on:

- a Traffic Engineering Database (TED): a PCE to execute his job requires all the information about the network topology, bandwidth, cost of the links, existent LSP. These information are usually collected through BGP-LS (Link State), which a bgp address-family that has been specifically designed to transport network topology information
- a PCC can ask to the PCE to find out a path that obeys to specific constraints, thus offloading potentially complex problems to an external and trusted device, specifically designed with this purpose

2.2.1 PCE devices types:

`Stateless PCE`

A stateless PCE has no knowledge about LSP previously configured in the network, and doesn't store a database of all the paths that has been calculated during time. This type of PCE doesn't have much sense in general, I personally wonder why it is listed at all.

Stateful PCE

A stateful PCE has an LSP database with ALL the used Label Switched Paths in the network, with all the occupied and available resources in terms of bandwidth. This database is kept up to date and the information can also be used in conjunction with telemetry data to perform even more complex

computations. For example, in case of congestion some traffic could be moved away from congested link and be routed somewhere else. Such things were managed through the auto-bandwidth command and traffic-engineering with rsvp, but results were not good enough since every node was independently taking its own decision, thus leading to an unstable network with tunnels fluctuating just as the congestion.

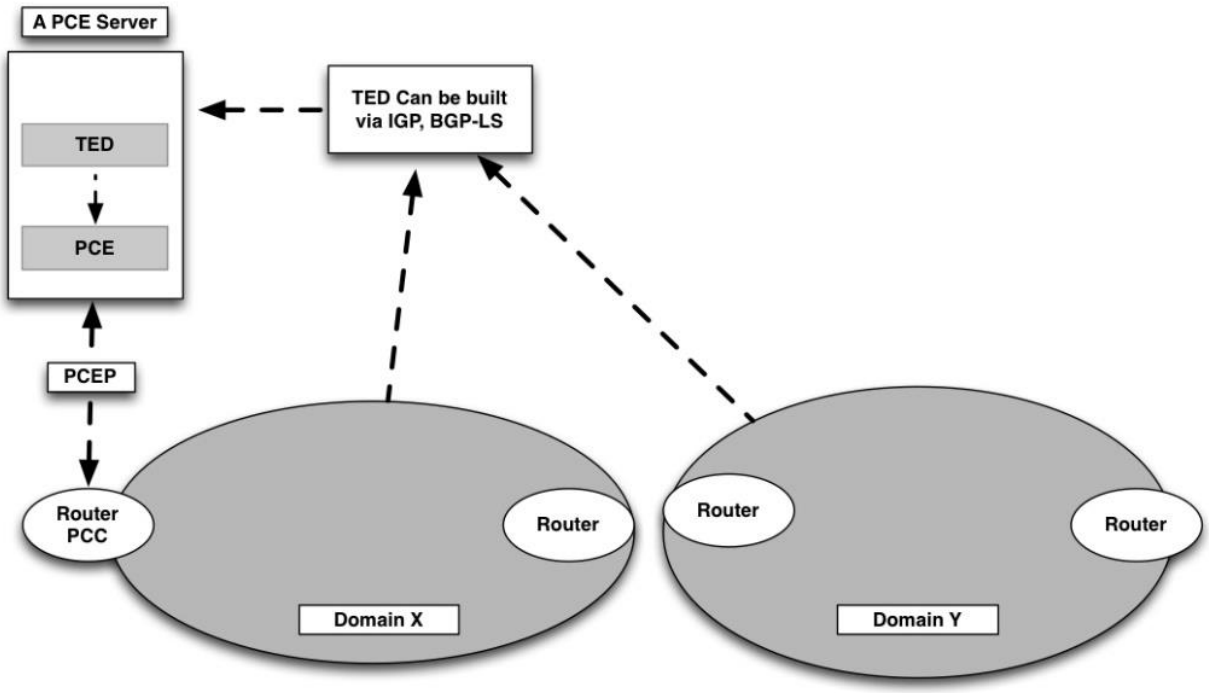


Figure 15 Elements of an SDN architecture.

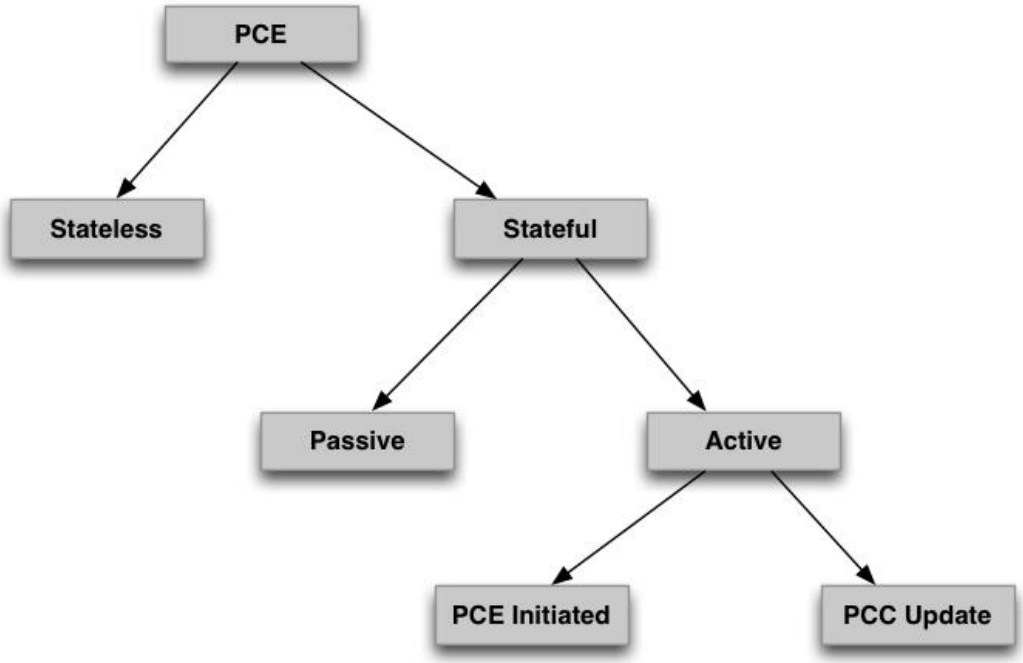


Figure 16 Differente PCE types.

2.2.2 Passive Stateful PCE

In case of a passive stateful PCE, the client decides to ask for help to the PCE, which considers the path request and constraints, and tries to fulfil the request.

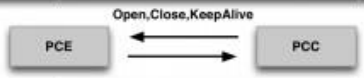
2.2.3 Active Stateful PCE

In this case, PCC creates LSP that have been created and calculated by the PCE, which can create them autonomously if certain specific events happen in the network. PCE is also responsible for removing or changing the LSP that it has previously instantiated.

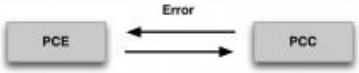
2.2.4 PCEP

It's the Path Computation Element Protocol (**PCEP**, RFC 8231) built on tcp port 4189. Some examples of messages are reported hereafter.

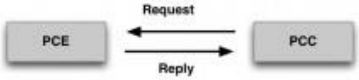
Messages	Purpose	PCE Type
Open, Close, Keep-alive	These are basically housekeeping messages. Open Message is also use for capability negotiation between PCE and PCC.	Stateless and Stateful



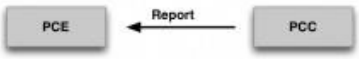
Messages	Purpose	PCE Type
Error Messages	Used for Error handling between PCE and PCC in case something goes wrong.	Stateless and Stateful



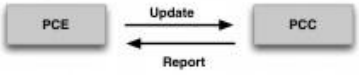
Messages	Purpose	PCE Type
Path Computation Request and Reply	PCE Request is sent by PCC to PCE asking for Path computation. PCE Reply is sent by PCE to PCC in response of PCE Request.	Stateless and Stateful



Messages	Purpose	PCE Type
PCC Report	PCC Report messages from PCC are used for multiple functions: 1) State Synchronization After the session between PCC and a stateful PCE is initialized, PCE must synchronize the state of a PCC's LSPs before it performs path computations or update LSP attributes in a PCC. 2) LSP State Report A PCC sends an LSP state report to a PCE whenever the state of an LSP changes. 3) LSP Control Delegation A PCC grants to a PCE the right to update LSP attributes on one or more LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect; the PCC may withdraw the delegation or the PCE may give up the delegation.(active stateful PCE only)	Stateful PCE



Messages	Purpose	PCE Type
PCC Update	A PCE requests updates of attributes on a LSP. (active stateful PCE only). PCC reports back with the status report messages to the PCE.	Stateful PCE



Messages	Purpose	PCE Type
PCE Initiate	Used by PCE to Initiate LSPs on PCC. A PCC reports back on the status to the PCE.	Stateful PCE

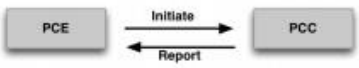


Figure 17 Communication between PCE and PCC.

2.2.5 Active Stateful PCE with SR

PCE is the 'thinkin brain' for traffic engineering applications also related to multiple IGP and routing domains, but interconnected each other.

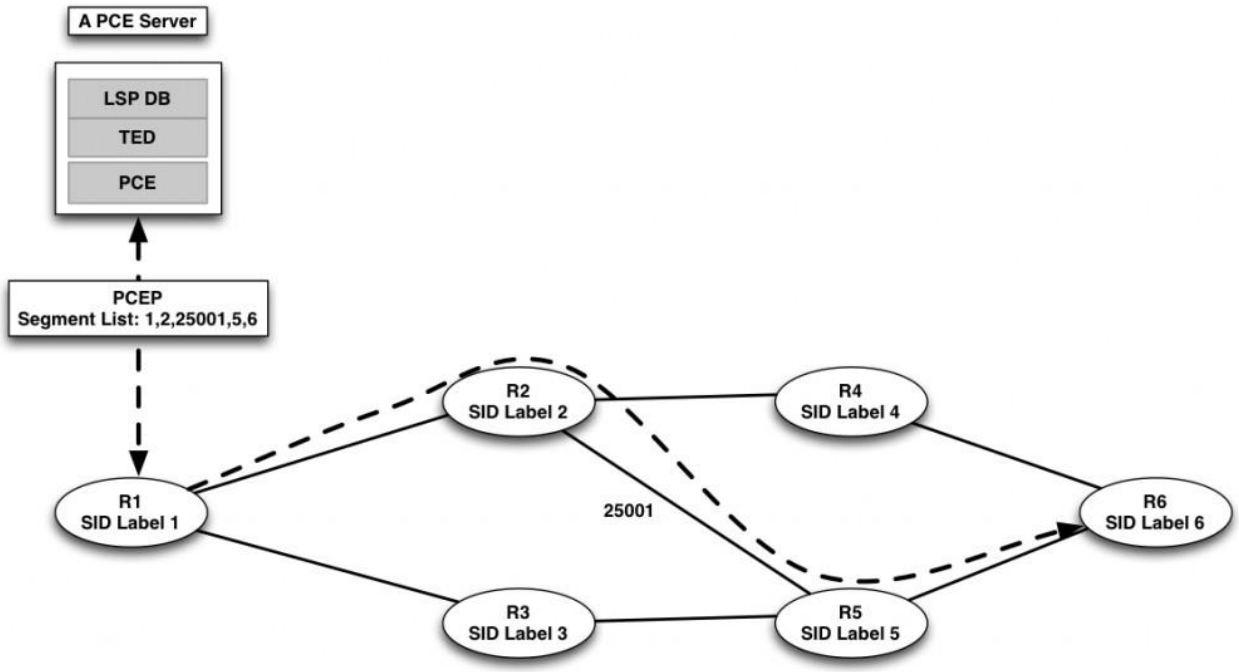


Figure 18 PCE calculated path, constraints provided by the PCC, PCE answers with the list.

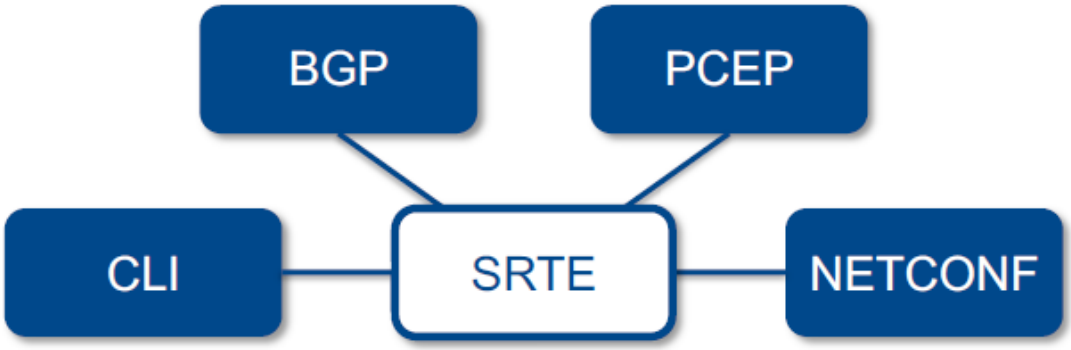
3 Traffic engineering

Traffic engineering in Service Providers environments didn't have such a great success, except for link-protection applications (FRR). We will provide here SOME of the available choices on a Segment Routing network, as provided by Cisco. More updated documentation can be for sure found on their sites.

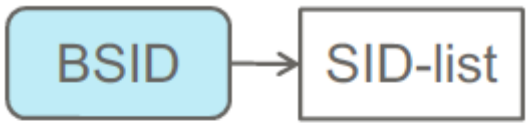
Every TE (Traffic Engineering) policy is determined by the following triple:

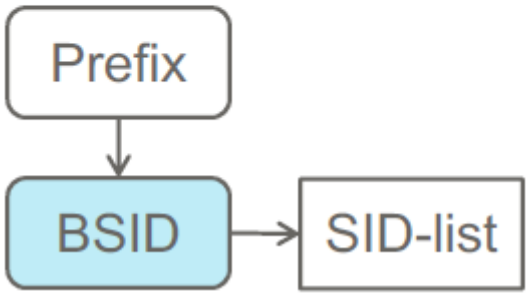
(source, color, endpoint)

With source/destination we are referring to the 'SID path' or LSP path (with the old terminology we could talk about the tunnel head-end or tail-end). Using Segment Routing, we have no more pre-signaled tunnels that need to be maintained and continuously refreshed in the network, at the expense of router's memory. With **MSD** (Maximum SID Depth) we refer to the maximum number of labels managed by the hardware (for example 10 labels on ASR9k and 5 on NCS). When traffic is diverted from the ingress PE into a specific SID path, we refer to this action as '**steering**'. Segment Routing policies can be installed in many different ways, as represented in the following picture.



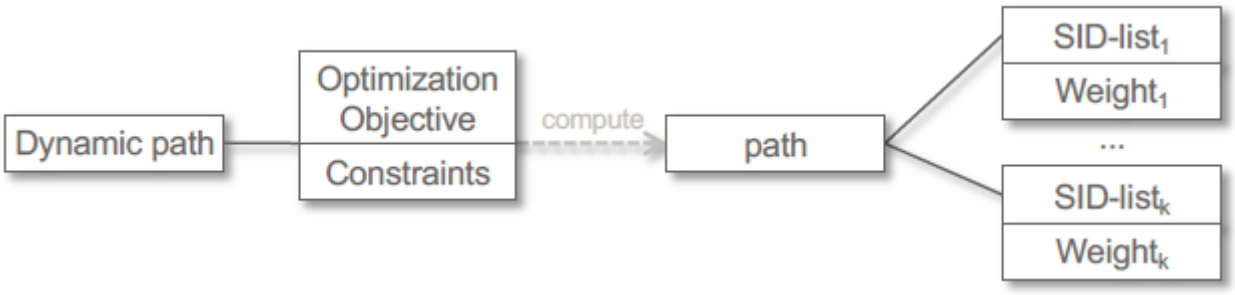
For every configured TE policy, is assigned a local identifier that uniquely identifies it and is called ‘binding SID’. If the router receives a packet with the ‘binding SID’ as the top label, the label is removed and the packet is pushed into the ‘SID path’.





A dynamic SR policy defines a certain goal in terms of constraints like for example:

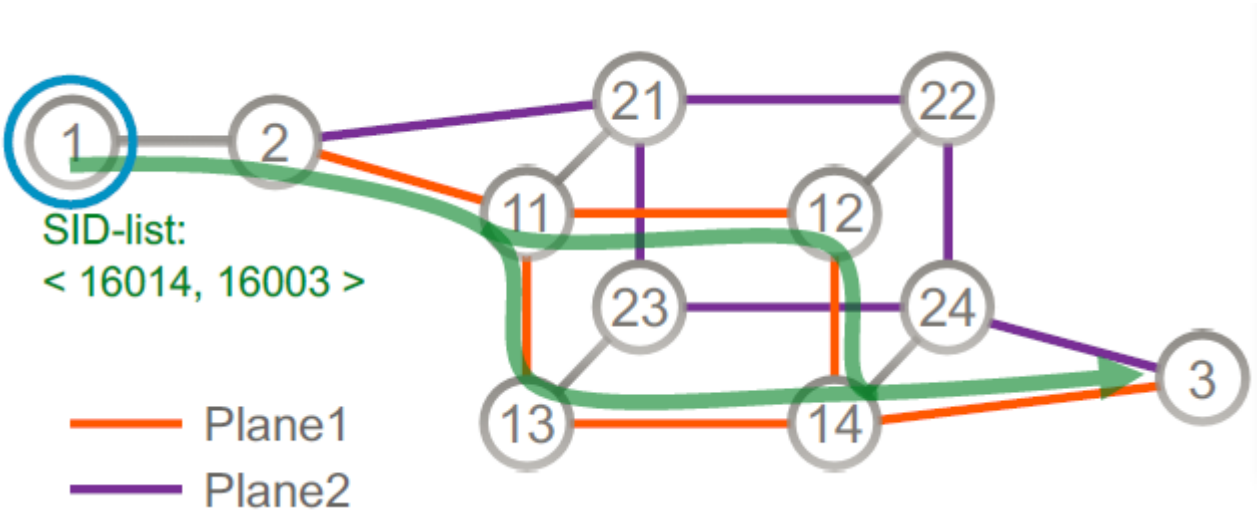
- Lower **IGP** cost (default)
- Lower **TE** cost (as we were used to, a link can have an IGP cost and a TE cost)
- lower **delay** (if you have ptp configured in your network, you can also measre the delay on every link and distribute the measured value through ISIS to share it with all the other nodes)



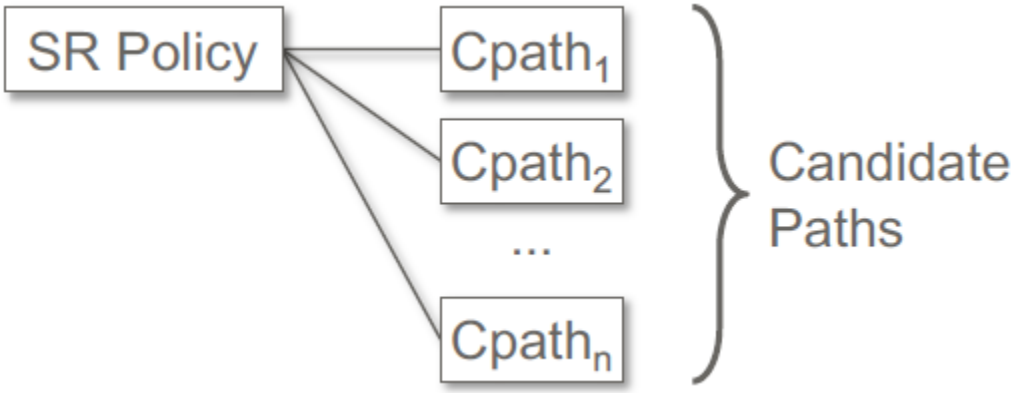
You can even define many constraints that a specific path or TE policy needs to respect:

- go through a certain list of nodes/links/ip address manually configureable

- maximum '**SID depth**' value
- include/exclude e certain group of links belonging to the same **SRLG**
- **maximum path cost** based on the calculation parameter (IGP cost, TE cost, delay, ...) possibly providing a range
- definition of a link **affinity**, define a color for every link and put a constraint of type 'do never/always (include-any/exclude-any) go through links with the same affinity'
- configure two paths as belonging to the same 'disjoint group', providing a link, node or SRLG constraint. This means that the two paths MUST NOT share the same links, nodes or SRLG group going from the head end to the tail. This is very useful on service providers when you have to separate in the core SCTP signaling voice traffic



Every SR policy identifies N potential solutions or paths, each of them could be used based on the configured preference, and a weight parameter (to proportionally balance the traffic over the available paths, more or less like in Qos weighted fair queing)



Recently has been added the 'flexible algorithm' feature, you can build different 'parallel' topologies, excluding some nodes or link in the network. In this way you can also obtain that TI-LFA is also calculated on the same topology, something you can't do that easily with normal approaches based on link coloring techniques.

Bgp steering is a feature active by default (it can be manually disabled), and uses the colors associated to SR policies. In particular it is possible to pre-configure on all PEs e certain number of SR TE policies, for example optimizing the delay, or the igp cost, or flowing only through red or blue links.

Then on REMOTE PEs you can color the routes with and extended community called '**opaque**', this number refers to the tunnel IDs on the routers headend of the policy. This is an easy, flexible and scalable way to achieve the desired result without reconfiguring every time the whole network. As an example take the following Cisco conFiguretion:

```
segment-routing
 traffic-eng
  policy POL10
   color 10 end-point ipv4 1.1.1.3      ← routes with color 10, next-hop 1.1.1.3
   candidate-paths
    preference 100
     explicit segment-list SIDLIST1
  !
  policy POL20
   color 20 end-point ipv4 1.1.1.3      ← routes with color 10, next-hop 1.1.1.3
   candidate-paths
    preference 100
     explicit segment-list SIDLIST2
  !
```

On every PE you can configure the following:

```
extcommunity-set opaque color10-te
  10
end-set
!
extcommunity-set opaque color20-igp
  20
end-set
!
extcommunity-set opaque color30-delay
  30
end-set

route-policy SET_COLOR_LOW_LATENCY_TE
  set extcommunity color color10-te
  pass
end-policy
!
route-policy SET_COLOR_HI_BW
  set extcommunity color color20-igp
  pass
end-policy
!
route-policy SET_COLOR_LOW_LATENCY
  set extcommunity color color30-delay
  pass
end-policy
!
```

Than on the PE to which the following network is attached, you color it to achieve the desired routing behaviour ... on all the other PEs in the network:

```
prefix-set sample-set
  88.1.0.0/24
end-set

route-policy SET_COLOR_GLOBAL
  if destination in sample-set then
    set extcommunity color color10-te
  else
```

```
    pass
  endif
end-policy
```

On voice networks you can set the opaque community number 30, optimizing the delay metric.