State of the Art Packet and
Optical Networking

Segment Routing in Packet Transport

White Paper

Introduction: How did Segment Routing START?

Networks have always been built to adapt to the nature of the traffic that it carries. Not long ago, when the bulk of the traffic was voice, TDM-based networks were a good fit for the characteristics of voice traffic – predictable, connection-oriented, and point-to-point. Since much of the revenue came from voice calls, TDM networks were designed to be reliable and robust. Everything would have been perfect, but then the World Wide Web (WWW) emerged. Data traffic grew, thanks to the quickly growing Internet. Traffic became unpredictable and more efficient on connectionless networks through flexible routing. Networks had to evolve to tame the surge of data traffic with two competing networking religions; ATM and IP. Both technologies tried to solve the growing problem, but failed because they were not efficient enough.

As networks tried to cope with the high capacity demands, a switching technique called MPLS (Multiprotocol Label Switching) was standardized with the original intent of speeding up packet forwarding. MPLS was also deployed, along with RSVP-TE, for its traffic engineering capabilities, making it possible to steer traffic to paths other than the shortest IGP path. This gave the network operator more control over the network. Implementation of MPLS in high-availability environments was also possible because of its recovery features, which were on par with SDH protection switching speeds. Apart from this, MPLS became a migration tool in transforming networks from circuit-based to packet-based. This was a necessity as global traffic became increasingly data-centric, rather than voice-centric. Network operators did not need to take a big leap towards packet-based networking because of MPLS' ability to carry different kinds of traffic (IP, Frame Relay, ATM, TDM and Ethernet). Voice and Data can coexist in the same network, so they do not need to be separated. MPLS enabled network operators to have an ultimate transport network carrying a variety of services and it quickly became a widespread industry standard.

The Need for Better Label Switching

The versatility of MPLS made it a common name in transport networks, but as networks grew, operations became more challenging. MPLS is dependent on IGP and label distribution or signaling protocols, such as MP-BGP, LDP, and RSVP-TE. These three signaling protocols, which operate on top of IGP in the control plane, are often present in networks delivering multiservices. The router has to work extra hard , as it must maintain dynamic IP routing while it takes care of the signaling protocol used to establish and maintain the state of the LSPs (Label Switched Paths), usually referred to as the outer label. Then, it has to establish and maintain another signaling session for the inner label to connect the services. Maintaining IGP convergence, parallel signaling sessions, traffic engineering, label switching, plus the task of data forwarding itself proved to be very taxing on routers in large-scale networks. From the operational perspective, many new services demanded traffic path diversion from the IGP's congested shortest path to better utilize network capacity. This led to the use of traffic engineering and explicit routing, which required complex configurations. This proved to be difficult to monitor and troubleshoot. It became evident that a new way of data forwarding had to be developed.

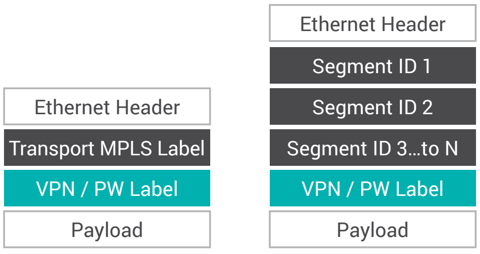
Segment Routing

Segment routing (SR) was conceptualized to reduce the number of signaling protocols in the network, to simplify traffic engineering and to enable network programmability. Implementing SR does not require a new hardware because it can operate on the existing MPLS forwarding plane, which makes migration from traditional MPLS to SR easier. SR can also interwork with RSVP-TE-based MPLS and LDP-based MPLS, thus avoiding islands in the network infrastructure. With new network capabilities introduced by SR, it is now possible to implement a transport network that is scalable, dynamic, flexible, and deterministic. This white paper describes SR's basic concepts and key components, based on an MPLS forwarding plane with a centralized controller in packet transport networks.

Basic Concepts

Segment Routing is a source routing technique, whereby the sender (ingress router) of the packet determines the path for reaching the destination. This path is not necessarily the shortest path and can be a traffic-engineered path. This differs from the classical routing technique, where the router looks at the destination address and uses it to decide where a packet should go next, using a routing table.

Segment Routing implements source routing by encoding the path in the packet header as a set of instructions on how a packet should traverse the network to reach its destination. This set of instructions is a stack of segments, represented by **segment identifiers (SIDs)**, and are very similar to stacked labels in MPLS. The packets carry the stack of segments as part of the packet header. The topmost segment of the stack instructs the route where and how to forward the packet. Forwarding may be according to the IGP route or according to an explicit route. The adjacent figure illustrates the similarity of how labels and segments are stacked in MPLS and SR.

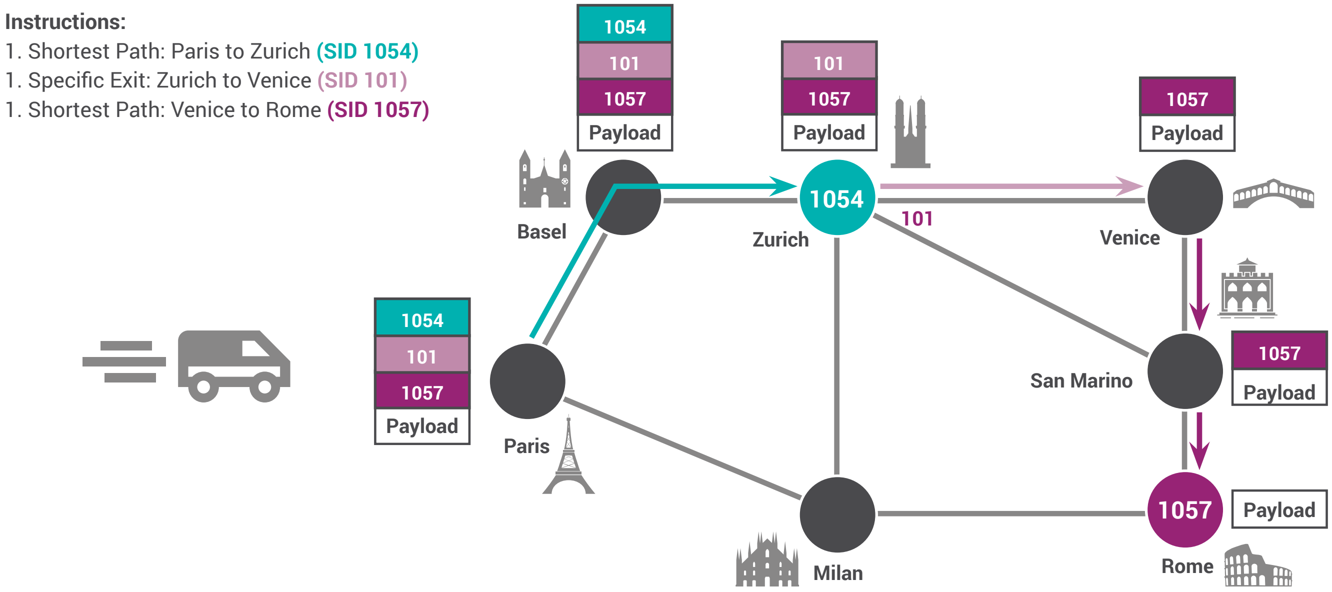


MPLS Label Stack

Segment Routing Ordered List of Segment IDs

There are different ways to implement SR. IETF's RFC 8402 describes two main implementations; SR over IPv6 **(SRv6)** and SR over MPLS **(SR-MPLS)**. While SRv6 presents both advantages and challenges, SR-MPLS is considered the preferred migration path to SR, using the same label switching principles on the existing MPLS forwarding plane. This means that implementation of SR-MPLS will not require a hardware change; just a revision to the embedded SW and a configuration change for switching from MPLS to SR-MPLS.

To illustrate how SR works, think of a packet as a truck delivering its payload from Paris to Rome, as shown in the illustration. The normal shortest route for the truck is to drive to Milan and then to Rome. If the shortest path is congested, we can instruct the driver to take a diversion to Zurich **(1054)**. Once in Zurich, we steer the truck to take a specific exit **(101)** towards Venice. From Venice, the truck will take the shortest route to Rome **(1057)**. The three instructions represented by segments **(1054, 101, 1057)** are executed sequentially, with the topmost instruction being carried out first and then deleted after completion, revealing the next instruction in the stack.

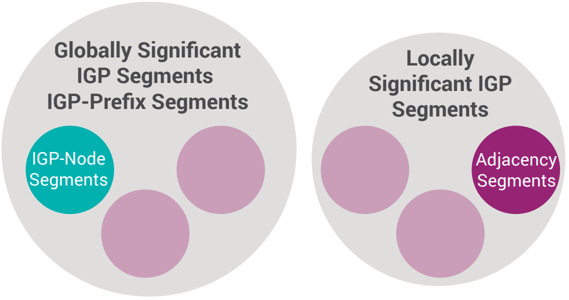


Instructions:

1. Shortest Path: Paris to Zurich **(SID 1054)**
1. Specific Exit: Zurich to Venice **(SID 101)**
1. Shortest Path: Venice to Rome **(SID 1057)**

IGP Segments

In networking terms, the instructions in the truck analogy are called **IGP Segments**, represented by SIDs. These are the essential type of segments used in packet transport networks. IGP Segments or Link-State IGP Segments correspond to the **prefixes** and **adjacencies** attached to an IGP Node being advertised within an SR domain. The advertisements are carried out by link-state IGPs, which have been extended to support SR; namely OSPF with SR extensions and IS-IS with SR extensions.



Segments are generally categorized as globally significant or locally significant. There are various types of segments within these categories, which can be used in different scenarios. For simplicity, we cover two very fundamental types; the IGP-Node Segments and the IGP Adjacency Segments.

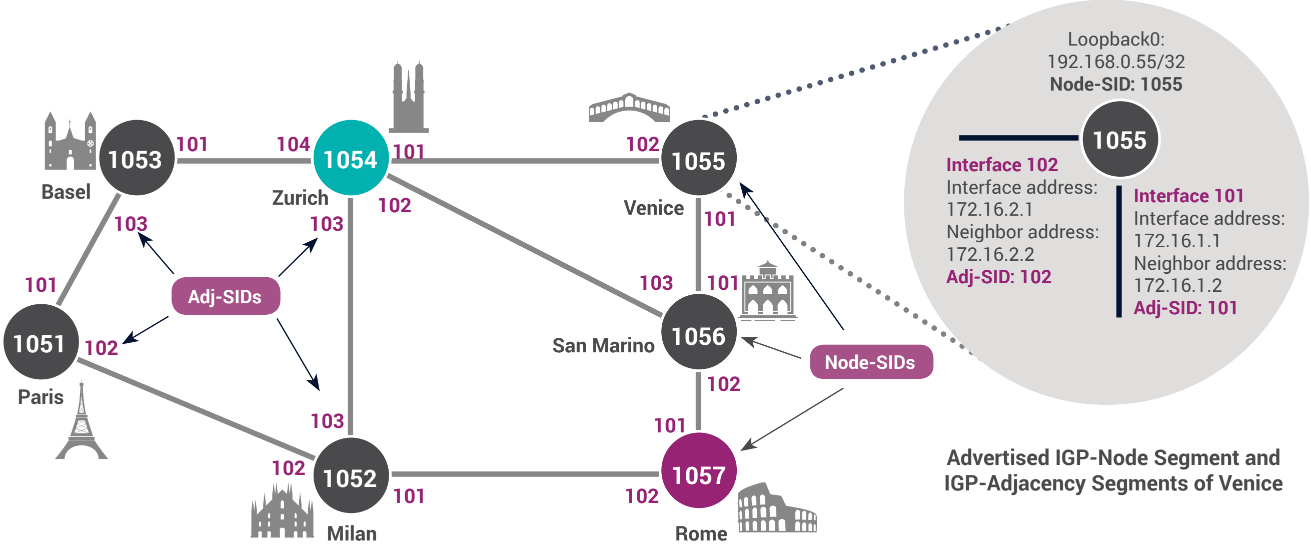
IGP Node Segments

- Identifies the IGP Node and normally correspond to its loopback address/prefix.
- Represented by **Node-SID**, is unique within the SR domain, and has global significance by default.
- It is an instruction to forward the packet using the shortest ECMP-aware IGP path towards the IGP Node configured with the Node-SID.

IGP Adjacency Segments

- Identifies the adjacency between an IGP Node (local node) and a remote node, usually adjacent to the local node.
- Represented by Adj-SID and is, by default, local to the IGP Node.
- It is an instruction to forward the packet from the IGP Node out, through a specific interface or set of interfaces configured with the Adj-SID.

In our truck example, we used two types of IGP segments, the **IGP-Node Segment** and the **IGP-Adjacency Segment**. **1054** and **1057** are Node-SIDs of Zurich and Rome, respectively, and **101** is an Adj-SID representing the adjacency of Zurich with Venice. When the topmost instruction is a Node-SID, like **1054** and **1057**, the processing IGP node will forward the packet towards the node configured with the Node-SID. We saw this in Paris, Basel, Venice and San Marino (as in the truck diagram on page 3). When the topmost instruction is an Adj-SID like 101, the node where **101** adjacency is attached, will forward the packet via this adjacency.

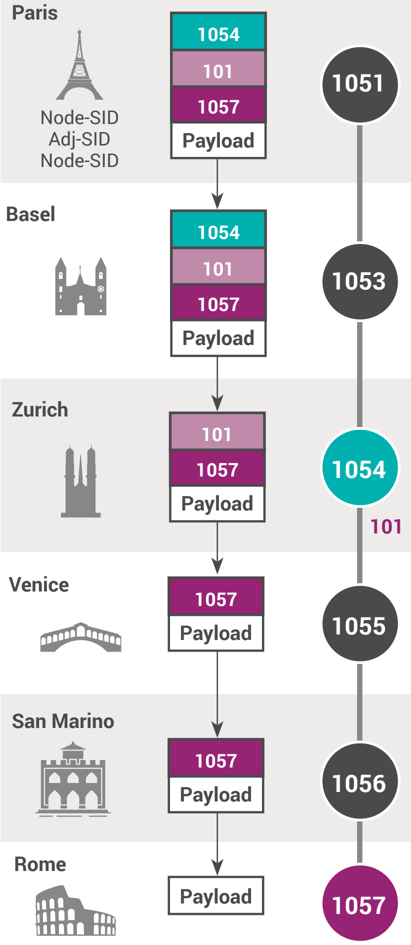


Advertised IGP-Node Segment and IGP-Adjacency Segments of Venice

If we configure our example network, it would look like the configuration shown on page 4. Each IGP Node will have a corresponding Node-SID, which is unique in the SR domain. IGP Adjacency segments are usually dynamically allocated with Adj-SIDs. These have local significance, which means that the same Adj-SID can be used by other IGP Nodes, as shown. Traffic can be steered explicitly by directing the packet to a specific interface or adjacency using the local Adj-SID.

If a path has only one segment consisting of a Node-SID, the behavior of the path will be similar to an LDP-based IP/MPLS. It will take the shortest path towards the advertising Node. To illustrate this, if a packet starts in Paris with only one segment being **1057** (Rome Node-SID), the packet will be forwarded towards Milan and then to Rome because this is the shortest path. To divert the packet from the shortest path, a combination of Node Segments and Adjacency Segments can be used. We illustrated this with the path using **1054**, **101** and **1057** SIDs, with **101** forcing the path to go towards Venice, regardless of the IGP cost. This is a form of explicit routing.

To illustrate this principle further, let's examine the Paris-to-Rome path, hop-by-hop. The packet forwarding process comprises the following steps:



Node, Node-SID and Segment list	MPLS operation	SR operation	Description
Paris 1054 101 1057 Payload	Push labels 1054, 101 & 1057	Push	Paris encodes the segment list on the packet and sends it towards Zurich using IGP shortest path.
Basel 1054 101 1057 Payload	Pop 1054 P-flag: PHP	Next	Basel receives the packet and knows that 1054 is a directly connected neighbour and pops-out 1054 before forwarding the packet.
Zurich 101 1057 Payload	Pop 101	Next	Zurich receives the packet and pops-out 101 before forwarding the packet out of interface 101.
Venice 1057 Payload	Swap 1057 with 1057	Continue	Venice receives the packet. Swaps 1057 with 1057 and forwards packet towards Rome using IGP shortest path.
San Marino 1057 Payload	Pop 1057 P-flag: PHP	Next	San Marino receives the packet and knows that 1057 is a directly connected neighbour. It pops out 1057 before forwarding the packet.
Rome Payload			The packet reaches the destination in Rome.

In essence, we see that the traffic is being steered via two Node-SIDs (**1054**, **1057**) and an Adj-SID (**101**). The Node-SID is used to forward the packet using the shortest IGP path, and the Adj-SID is used to divert the traffic to a specific interface or adjacency. As each of the three instructions are being executed, the SID is popped out from the stack, exposing the next instruction. The process is repeated until the packet arrives at its destination.

SR-MPLS

SR-MPLS uses the same MPLS label operations in the forwarding plane by instantiating and processing Segments as MPLS labels. The previous figure illustrates three SR equivalent MPLS operations on the MPLS forwarding plane, which can be summarized as follows:

- The PUSH operation is implemented as an MPLS PUSH operation.
- The CONTINUE operation is implemented as an MPLS SWAP operation.
- The NEXT operation is implemented as an MPLS POP operation.

When SIDs are represented as MPLS labels, they are allocated from two important blocks of MPLS labels on each of the nodes.

SR Global Block (SRGB)

In a MPLS forwarding plane, a set of local MPLS labels on an IGP Node is reserved for global segments such as IGP Prefix-SIDs. This set of labels is called SR Global Block or SRGB. The use of identical SRGBs on all nodes within the SR domain simplifies operations and troubleshooting.

SR Local Block (SRLB)

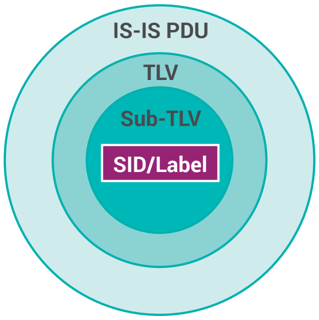
In a MPLS forwarding plane, a set of local labels on an IGP Node is reserved for local segments, such as Adj-SIDs.

The globally significant labels have the benefit of a unified and consistent view of the network, while locally significant labels are normally used to carry out instructions associated only at the node level. Aside from operational simplification, proper allocation and assignment of SIDs/labels, SRGB and SRLB are very important, as they do not only represent instructions.They also represent the type of segment. This becomes very relevant later on, when advanced algorithms are applied in traffic engineering.

IS-IS Extensions for SR: The role of TLVs and Sub-TLVs in SID/Label Advertisements

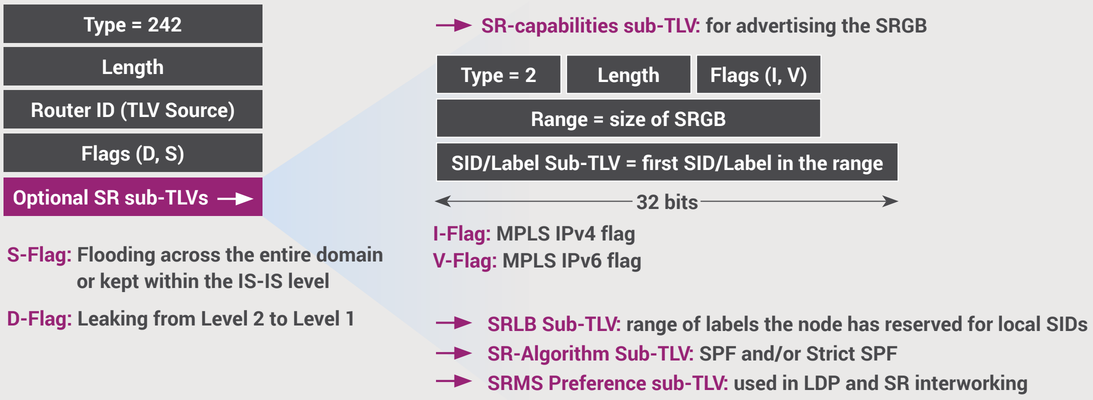
One of the advantages of SR is the elimination of label distribution protocols, sometimes called signaling protocols, such as LDP and/or RSVP-TE. Instead, SR uses extensions on existing IGPs to advertise segments or labels. Both OSPF and IS-IS were extended to carry segment information with various flag options to indicate certain characteristics, adding more flexibility in the processing of segments. SR capabilities in IGPs are facilitated by the use of TLVs and sub-TLVs. TLVs are Type-Length-Value codes, which enable the IGP to announce optional information or capabilities, like SR, together with related specific behaviors.

In this white paper, we take a closer look at how TLVs and sub-TLVs are used in an SR-MPLS environment, according to the **IS-IS SR extensions**. TLVs account for IS-IS' versatility and extendibility by allowing new fields of information to be added to the existing protocol. Therefore, IS-IS did not need to be reinvented to support SR. It only needed to be extended to be able to advertise SR information, using newly-defined TLVs and sub-TLVs. The adjacent figure shows how the SID or Label is carried by the IS-IS PDU as part of the Link State advertisement. On a high level, the IS-IS PDU contains the TLV, the TLV contains the sub-TLV, and the sub-TLV contains the SID/Label.



One of the TLVs that carries SR information is the **IS-IS Router Capability** TLV or TLV type 242. It has an S-Flag and a D-Flag, which can be set according to how a SR sub-TLV is propagated across IS-IS Level boundaries (Level 1 and Level 2). It can be for the entire domain, only for the respective level, or it can be directed to cross between levels.

IS-IS Router Capability TLV-242



Type = 242
Length
Router ID (TLV Source)
Flags (D, S)
Optional SR sub-TLVs →

S-Flag: Flooding across the entire domain or kept within the IS-IS level

D-Flag: Leaking from Level 2 to Level 1

→ SR-capabilities sub-TLV: for advertising the SRGB

Type = 2	Length	Flags (I, V)
Range = size of SRGB		
SID/Label Sub-TLV = first SID/Label in the range		

← 32 bits →

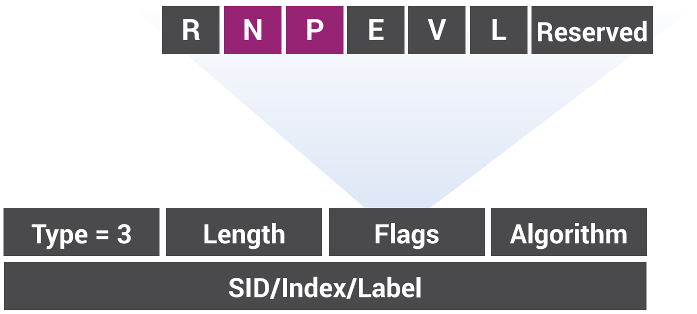
I-Flag: MPLS IPv4 flag
V-Flag: MPLS IPv6 flag

→ SRLB Sub-TLV: range of labels the node has reserved for local SIDs
→ SR-Algorithm Sub-TLV: SPF and/or Strict SPF
→ SRMS Preference sub-TLV: used in LDP and SR interworking

Like the TLVs, the inserted SR sub-TLVs can also have flags and other values. The most important SR sub-TLV of the TLV type 242 is the **SR Capabilities sub-TLV** or the sub-TLV type 2. Every IGP Node participating in SR will advertise this sub-TLV. It has an I-Flag and a V-Flag, indicating whether SR-MPLS encapsulated IPv4 packets or IPv6 packets are being processed. Sub-TLV type 2 also includes the SRGB range and the first label of that range. There are other defined sub-TLVs, which indicate the SRLB, SR-Algorithm, and the SRMS preference, which is used in LDP and SR interworking. Different networking scenarios will require some, if not all, of the information advertised by these sub-TLVs. TLV type 242 and its sub-TLVs are shown above.

Prefix-SID Sub-TLV

The Prefix-SID sub-TLV carries the SR IGP-Prefix-SID, which is unique, by default, within the IGP domain. The format is shown here:



Prefix-SID sub-TLV can be present in the following TLVs:

TLV-135 Extended IPv4 reachability
TLV-235 Multitopology IPv4 Reachability
TLV-236 IPv6 IP Reachability
TLV-237 Multitopology IPv6 IP Reachability
TLV-149 Binding-TLV
TLV-150 Multi-Topology Binding-TLV

R: Re-advertisement: set if the attached non-local prefix is propagated to another level or redistributed

N: Node-SID: Set if the prefix-SID is a node-SID

P: PHP: If not set then any upstream neighbor of the Prefix-SID originator MUST pop the Prefix-SID.

E: Explicit-Null: Set if penultimate hop must replace prefix-SID with Explicit-Null label

V: Value: Set if prefix-SID carries a value (not an index)

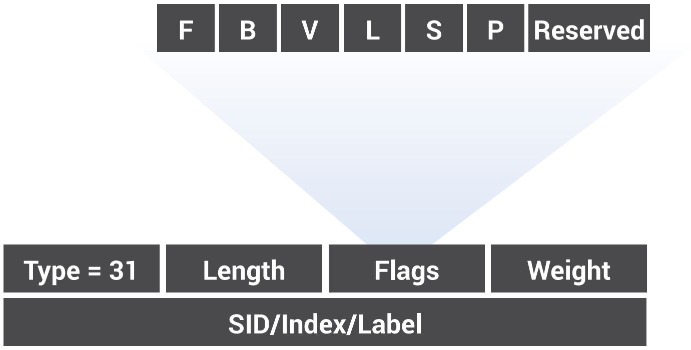
L: Local: Set if prefix-SID has local significance

The Prefix-SID sub-TLV can be present in different types of TLVs, each with a different purpose, depending on the context in which the prefix-SID is being advertised. The possible TLVs that can carry a Prefix-SID sub-TLV are listed above. To further illustrate the Sub-TLV flag function in our Paris-to-Rome example, the Zurich Node-SID 1054 is advertised as a prefix-SID sub-TLV with the N-Flag set to 1 and the P-Flag set to 0. The N-Flag=1 identifies the advertised SID as a Node-SID and the P-Flag=0 gives the instruction to pop the Node-SID 1054 from the segment list before delivering the packet to the node advertising the Node-SID 1054. We observed this process when Basel popped out the Node-SID 1054 before it forwarded the packet to Zurich, the node advertising Node-SID 1054. Here, we are also illustrating PHP behavior, which can be enabled or disabled using the P-flag, depending on the desired outcome.

The Prefix-SID sub-TLV also has an Algorithm field next to the Flag field, indicating how reachability is calculated for the prefix. Examples are Algorithm=0 for Shortest Path First and Algorithm=1 for Strict Shortest Path First.

Adj-SID Sub-TLV

The figure below shows the format of Adj-SID sub-TLV, which carries the Adj-SID. The Adj-SID sub-TLV is similar to the Prefix-SID sub-TLV. The associated flags are specific to adjacency characteristics and, therefore, are different from the flags of Prefix-SID sub-TLV. The Flag field is next to the Weight field, (instead of the Algorithm field). The Weight field represents the weight of the adjacency, which is useful for load balancing between parallel adjacencies.



F: Address-Family: Unset for IPv4 encapsulated traffic. Otherwise, it forwards IPv6 encapsulated traffic

B: Backup: Set, if eligible for protection

V: Value: Set, if the Adj-SID carries a value. Set by default.

L: Local: Set, if the value/index has local significance. Set by default.

S: Set: Set, if Adj-SID refers to a set of adjacencies

P: Persistent: Set, if the Adj-SID is persistently allocated

Adj-SID sub-TLV can be present in the following IS-Neighbor TLVs:

TLV-22 Extended IS reachability

TLV-222 Multitopology IS

TLV-23 IS Neighbor Attribute

TLV-223 Multitopology IS Neighbor Attribute

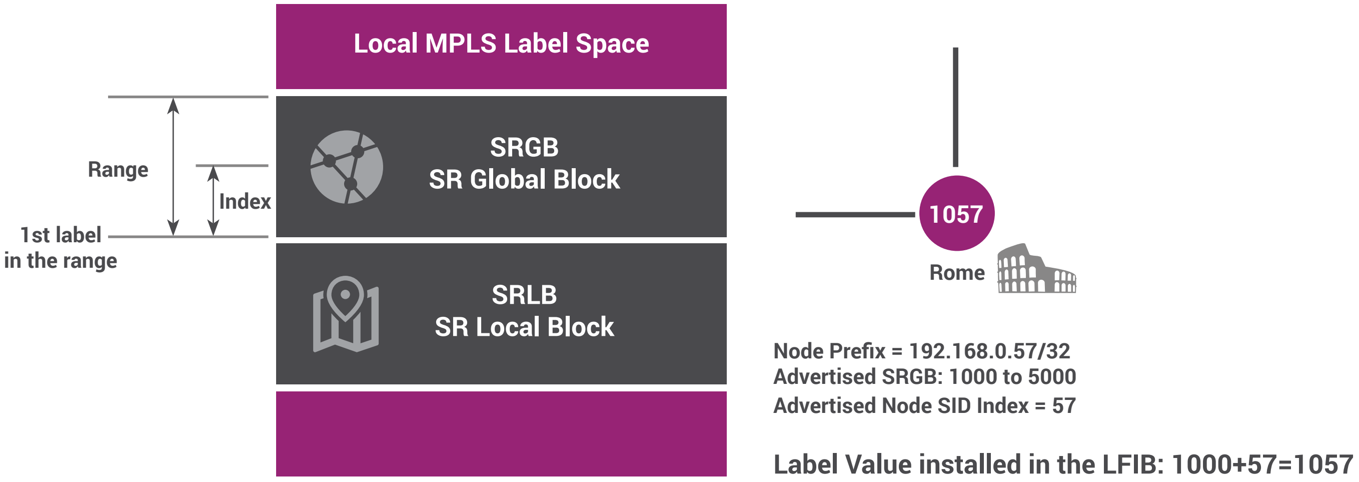
TLV-141 Inter-AS reachability information

It is noteworthy that, so far, we illustrated only two types of segments; namely, the Node and Adjacency segments. There are other types of segments with many combination possibilities in the segment list to steer the traffic in different ways, achieving network programmability and flexibility.

Representing the SID with an MPLS Label or an Index

In both types of sub-TLVs described, there is a SID/Label/Index field, which is the most important information in the sub-TLV. In SR-MPLS, this field contains either a label or an index, depending on the chosen implementation. If it is an index, the label value is generally derived, using the following formula:

Label Value = Index + Start of the SRGB Range


Local MPLS Label Space
Range
Index
1st label
in the range
SRGB
SR Global Block
SRLB
SR Local Block
1057
Rome
Node Prefix = 192.168.0.57/32
Advertised SRGB: 1000 to 5000
Advertised Node SID Index = 57
Label Value installed in the LFIB: 1000+57=1057

In this example, using an SRGB range for all IGP Nodes to be 1000 to 5000, the Rome Node could be configured with a SID index=57. When the node prefix is advertised with SID index=57 via IS-IS, all the IGP Nodes receiving the advertisement will install the Label Value of 1000+57=1057 into their LFIB for Rome's node prefix 192.168.0.57. When a packet is received with 1057 as the topmost label of the label stack by any of the IGP nodes, it will be processed as the incoming label. The label mapping on the LFIB will determine the outgoing label and will be processed, depending on the IGP reachability of the node 1057. If the node 1057 is a directly connected neighbour of the IGP Node, label 1057 will be popped-out (with P-flag=0) and the packet is forwarded accordingly.

For more meaningful representation of the segments configured in the network, label indices are used instead of exact label values. This simplifies network configuration and troubleshooting. For example, we can configure the IGP nodes to have label indices starting from 1, then 2, 3, 4, and so on. This is simpler than using an exact label from the SRGB that could start with 786543, for example. Nevertheless, implementation differs from one operator to another.

In the case of adjacency segments, the IGP Node normally and/or automatically assigns a label value from the SRLB and advertises this label value as Adj-SID. Adj-SIDs are visible to the other IGP Nodes but the corresponding label value is only installed in the LFIB of the originating IGP Node. The originating IGP Node will be the only one that executes the instruction and forwards the packet to the specific adjacency segment attached to it. Even though it has local significance, the Adj-SID still needs to be advertised, so that the path computation entity (an ingress node or a controller) will be aware of its existence and be able to consider it when computing a path.

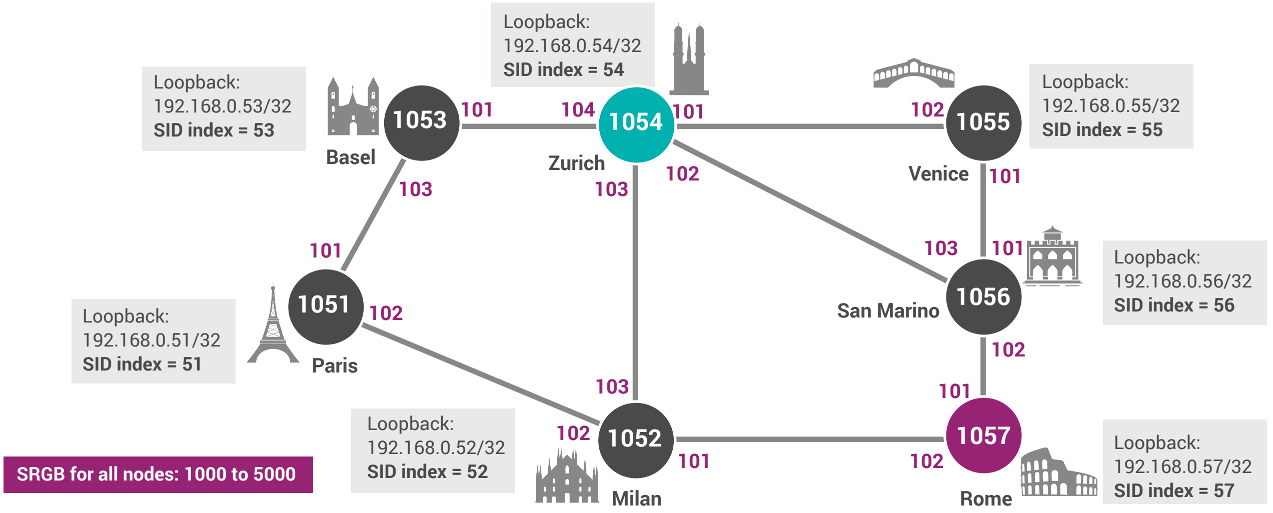
Label Mapping

Going back to our example, after the SIDs are advertised via IGP, each IGP Node will install the labels in the LFIB, accordingly. We normally find three types of entries:

1 **Node-SIDs** belonging to a directly connected IGP Node will have a pop operation, if PHP is enabled. The packet is forwarded to the neighbour as a normal non-labelled packet.

2 **Node-SIDs** belonging to a non-directly connected IGP Node will have a swap operation. The labelled packet is forwarded as per IGP Shortest Path.

3 **Adj-SIDs** are dynamically allocated and advertised but only installed in the LFIB of the originating IGP Node, being local SIDs. The packet is forwarded through the adjacency without consideration of the IGP cost.



1054 Zurich Node Label Map

Label in	Label out	Operation: MPLS/SR	Next Hop
1055	-	Pop/Next	To Node1055
1052	-	Pop/Next	To Node1052
1053	-	Pop/Next	To Node1053
1056	-	Pop/Next	To Node1056
1051	1051	Swap/Continue	To Node1051
1057	1057	Swap/Continue	To Node1057
101	-	Pop/Next	To Intf 101
102	-	Pop/Next	To Intf 102
103	-	Pop/Next	To Intf 103
104	-	Pop/Next	To Intf 104

1057 Rome Node Label Map

Label in	Label out	Operation: MPLS/SR	Next Hop
1056	-	Pop/Next	To Node1056
1052	-	Pop/Next	To Node1052
1051	1051	Swap/Continue	To Node1051
1053	1053	Swap/Continue	To Node1053
1054	1054	Swap/Continue	To Node1054
1055	1055	Swap/Continue	To Node1055
101	-	Pop/Next	To Intf 101
102	-	Pop/Next	To Intf 102

■
- Node-SIDs belonging to a directly connected IGP Node
- With Penultimate Hop Popping (PHP)
- Packets are forwarded to the neighbour as a normal non-labelled packet.

▨
- Node-SIDs belonging to a non-directly connected IGP Node.
- Forwarded as per IGP Shortest Path

□
- Adj-SIDs local to the IGP Node.
- Packets are forwarded through the adjacency without consideration of the IGP cost.

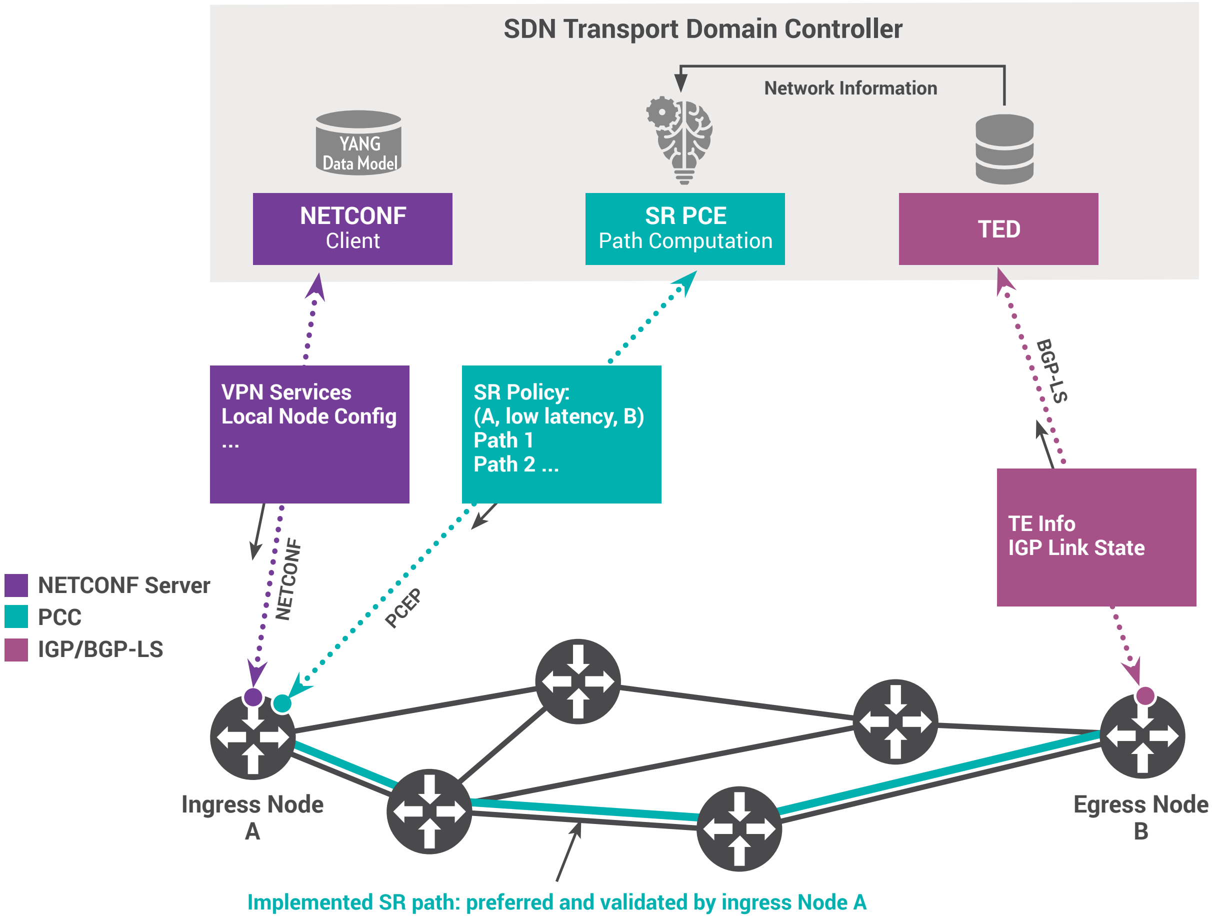
In the figure above, we see that the Zurich node has four Adj-SIDs in the LFIB corresponding to four adjacencies attached to it (101, 102,103, and 104), while the Rome Node has only two Adj-SIDs in the LFIB, corresponding to only two adjacencies attached to it (101 and 102). Notice that the label-in and label-out value is the same for the swap/continue operation. This is normally the case when all nodes have the same SRGB, like in the example above. In cases where different SRGBs are used, the label-out value is calculated according to the next hop's SRGB and, therefore, will be different from the label-in value.

Path Computation: The Roles of NETCONF/YANG, PCEP and BGP-LS

So far, we showed how SIDs are advertised via IGP and how SIDs are processed as labels in SR-MPLS. Once the IGP has converged, the labels advertised, and the SID/Label entries are in LFIB, the ingress node can then encode the segment list on the packet and send it through the SR domain. To steer the packet to the egress node, paths can be created and configured using multiple mechanisms:

CLI | EMS | NETCONF | PCEP

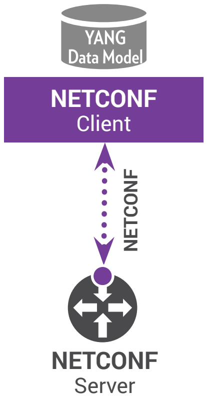
Best-effort paths or IGP shortest paths are normally computed by the ingress node and configured/enabled via management agents, such as CLI, EMS, or NETCONF. CLI and EMS management agents are the traditional way of configuring a network device. On the other hand, NETCONF has been gaining acceptance over the past few years, providing a simple and universal way to abstract the network infrastructure.



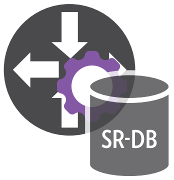
Note: NETCONF connections are normally present in all nodes. PCEP connections are required for Ingress Nodes which implement SR Policies. BGP-LS connections are at least one within a domain

NETCONF or Network Configuration Protocol. This is an IETF network management protocol, which was initially designed to replace the CLI-based programming interface on the network devices, providing an open standard and programmable architecture across a multivendor environment. NETCONF offers a flexible way of manipulating device configuration, while offering significant improvements over traditional SNMP. NETCONF uses the YANG (Yet Another Next Generation) data modelling language, enabling higher level abstraction for orchestration and automation. This makes NETCONF/YANG essential for Network Operators intending to migrate from a traditional NMS platform to a unified SDN platform.

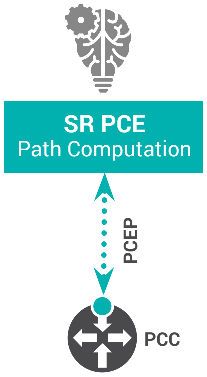
While NETCONF can be used to configure SR paths, it is mainly used to install, manipulate, and delete configuration on the network devices. It can co-exist with the traditional EMS and CLI, giving the operator flexibility to choose different ways of configuring SR.



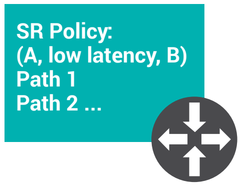
Before an SR path is configured and enabled on a network device, it has to be computed by an entity. This entity can be the ingress node or it can be an external controller. We call these two approaches as distributed path computation approach and the centralized path computation approach, respectively.

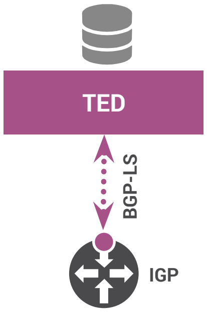


Distributed Path Computation. When an ingress node computes the SR path, the network is supposed to perform a distributed path computation. In principle, the ingress node can use the IGP link-state database and IGP TLVs to learn path variables, such as latencies and topologies, in order to compute the path and select the corresponding segment list. Path computation can be based on the node's SR Database (SR-DB), which contains information like IGP metrics, SRGB, SRLB, Prefix-SIDs, and Adj-SIDs. It also contains traffic engineering attributes, such as latency and loss, to name a few. The ingress nodes make independent decisions to compute and validate the paths. This approach is also called a distributed control plane, since there is no centralized controller involved.



Centralized Path Computation using a Path Computation Element (PCE). This is an approach that relieves the routers from performing complex tasks, such as advanced path computations and traffic engineering, by taking advantage of an SDN centralized controller. Centralized path computation with SR can be implemented using a PCE located within the SDN ecosystem. In packet transport, the PCE is part of an SDN controller, responsible for computing network paths based on network information. This information includes network topology, reachability, and resource information. The PCE communicates with the network devices via the Path Computation Element Communication Protocol (PCEP) as shown in the Path Computation map on the previous page. The network devices run a Path Computation Client (PCC) process that sends path computation requests to a PCE via PCEP. The PCE can also initiate the path setup by sending a request to the PCC via PCEP. In both cases, the PCE computes a candidate path or set of candidate paths, which satisfy a given requirement and sends it to the PCC of the network device. The set of candidate paths forms an SR Policy, which is associated with an intent for a given pair of ingress and egress nodes. The intent of the SR Policy can be a specific network requirement, like low latency or avoiding certain link properties, for example. The SR policy being sent to the ingress node can have multiple candidate paths and can arrive from other sources besides the PCE, like NETCONF or CLI. However, in simple use cases, there is usually only one candidate path in the SR policy. According to preference and validity, the ingress router selects a candidate path from the SR policy and installs it in the RIB/FIB. Then, the SID list of the chosen path is encoded as a header for the packet.

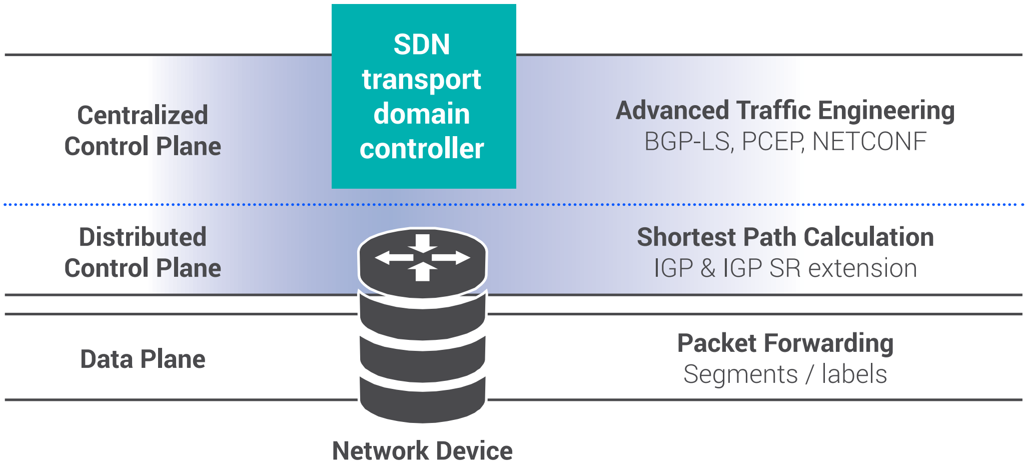




BGP-Link State (BGP-LS). The PCE computes the path, based on the network information derived from the network via BGP-LS. A BGP-LS connection between the transport domain controller and at least one of the network devices enables the mechanism to advertise link-state and traffic engineering information from the network infrastructure to an external component (controller) using the BGP routing protocol. BGP-LS takes the information from the IGP, filters only the PCE-related information, and relays them to the transport domain controller to form the Traffic Engineering Database (TED). The TED can also be fed with information other than what is available from the BGP-LS using other sources, such as a Network Management System or via direct data input; for example, related link information from an underlay transport. The transport domain controller builds and maintains the TED.

The Hybrid Control Plane Approach in Packet Transport

What has been described is called the hybrid control plane approach, wherein routing protocols are present in both controller and network devices. It is a combination of a centralized and a distributed control plane approach. This is a slight deviation from SDN's ideal case of total separation of the control and data planes. For packet transport networking, which inherently has a distributed control plane architecture, it makes more sense to keep some of the basic control plane functions in a distributed manner, like reachability information exchange and shortest path calculation. This makes it easier for SR migration and interoperation with traditional MPLS networks because there is no abrupt change in the way basic connectivity is established. To take advantage of SDN principles together with SR, the hybrid approach reduces the complexity in the network infrastructure, by assigning complex functions to a centralized controller where the PCE resides. This is achieved by having the ingress node compute best-effort paths or IGP shortest paths locally, while letting the controller/PCE compute advanced traffic engineered paths and other service-related controls, like VPNs.


SDN transport domain controller
Centralized Control Plane
Advanced Traffic Engineering
BGP-LS, PCEP, NETCONF
Distributed Control Plane
Shortest Path Calculation
IGP & IGP SR extension
Data Plane
Packet Forwarding
Segments / labels
Network Device

Conclusion

Segment Routing is a new way of forwarding data, which simplifies the process of traffic engineering. It was conceptualized during the time when network operators are finding new ways to make networks scalable, dynamic, flexible, and deterministic. While LDP and RSVP-TE based MPLS have served their purpose, networks had to evolve to carry new services with a huge variety of requirements.

MPLS has proven its versatility over the years. Therefore, instead of developing a new protocol, Segment Routing was designed to run using the MPLS forwarding plane. This eases the migration to the new technology while solving MPLS' complexity and scalability problems. We believe packet networks need to transition from MPLS to SR in order for them to support new services. This is driven by the need for more automation and orchestration, which can be facilitated by SR's hybrid control plane approach.

While we pointed out SR advantages, there is still a lot to consider in migrating the networks to the new ideology. The key components in the SR ecosystem are illustrated in this paper, as well as their interaction and interrelationship. SR is not only a method of forwarding packets; it defines an ecosystem that stretches beyond the forwarding plane. To gain the benefits, changes in the control plane and the management plane also have to take place: centralizing control for advanced traffic engineering; upgrading the network to support PCEP, BGP-LS, and IGP with SR extension; and moving from a pure NMS platform to an SDN platform for attaining full cycle automation. The huge change is the introduction of a controller and the associated applications, enabling intelligent traffic engineering and assured services. While the controller reduces complexity and network state, operations must evolve to the new way of transport networking by keeping the network infrastructure simple and bringing the complex tasks onto the SDN platform.

Nevertheless, SR-MPLS does not change the forwarding plane significantly and it still uses labels. In many existing network devices, SR-MPLS is just a matter of upgrading the embedded software on existing hardware and migrating the traffic simply by configuring the services to prefer the SR Tunnels instead of MPLS Tunnels. With this, migration plans can be incremental and not radical.

As part of this new direction in transport networking, considerable effort on research and collaboration tests in developing SR's ecosystem has been invested within the telecom industry to achieve multilayer transport networking and end-to-end orchestration/automation. IETF continues to generate enhancements to SR and to its related services covering various use cases. One important SR use case is transport network slicing, which applies both to 5G networks and fixed networks. SR's flexibility and programmability also enable application-driven networking, which is a way of generating new revenue for Service Providers. This will be impractical to implement in traditional networks. SR does not only solve network complexities. It is an inevitable change that needs to take place in packet transport as part of its evolution.

Contact Ribbon today to learn more about our Packet Transport solutions at rbbn.com

Abbreviations

ATM	Asynchronous transfer mode
AS	Autonomous System
BGP-LS	Border Gateway Protocol Link-State
CLI	Command-Line Interface
EMS	Element Management System
ECMP	Equal-cost multi-path
5G	Fifth Generation of Cellular Nework Technology
FIB	Forwarding Information Base
IGP	Interior Gateway Protocols
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	IP version 4
IPv6	IP version 6
LDP	Label Distribution Protocol
LFIB	Label Forwarding Information Base
LSP	Label-Switched Path
MP-BGP	Multiprotocol Extensions for BGP
MPLS	Multiprotocol Label Switching
NETCONF	Network Configuration Protocol
NMS	Network Management System
IS-IS	Intermediate System to Intermediate System
OSPF	Open Shortest Path First
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Communication Protocol
PDU	Protocal Data Unit
RFC	Request for Comments
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
RIB	Routing Information Base
SR-DB	Segment Routing Database
SID	Segment Identifier
SR	Segment Routing
SRMS	Segment Routing Mapping Server
SPF	Shortest Path First
SNMP	Simple Network Management Protocol
SRGB	SR Global Block
SRLB	SR Local Block
SRv6	SR over IPv6
SR-MPLS	SR over MPLS
SDH	Synchronous Digital Hierarchy
TDM	Time Division Multiplexing
TE	Traffic Engineering
TED	Traffic Engineering Database
TLV	Type-Length-Value
VPN	Virtual Private Network
WWW	World Wide Web
YANG	Yet Another Next Generation

About Ribbon

Ribbon Communications (Nasdaq: RBBN), which recently merged with ECI Telecom Group, delivers global communications software and network solutions to service providers, enterprises and critical infrastructure sectors. We engage deeply with our customers, helping them modernize their networks for improved competitive positioning and business outcomes in today's smart, always-on and data-hungry world. Our innovative, end-to-end solutions portfolio delivers unparalleled scale, performance, and agility, including core to edge IP solutions, UCaaS/CPaaS cloud offers, leading-edge software security and analytics tools, as well as packet and optical networking leveraging ECI's Elastic Network technology.