Chapter 2

What Is Segment Routing over MPLS (SR-MPLS)?

We took a brief look at MPLS and its shortcomings in Chapter 1, "MPLS in a Nutshell." Now it is time to build a solid understanding of segment routing (SR) so you will be ready for the upcoming chapters, which cover high-level design, configuration, and verification of various transport- and service-related aspects of SR-enabled networks. This chapter introduces basic segment routing concepts by using an analogy and then goes into the theory behind the MPLS data plane encapsulation implementation. This chapter covers Segment Routing for MPLS (SR-MPLS), and Chapter 3, "What Is Segment Routing over IPv6 (SRv6)?" covers IPv6 (SRv6) data plane encapsulations. The terms, abbreviations, and acronyms introduced in this chapter are consistently used throughout the remainder of this book.

Before delving into the more technical specifications of segment routing, let's consider a simplified high-level analogy that serves as an example to explain underlying key concepts. The central processing unit (CPU) installed in an everyday device, such as a mobile phone, smart TV, laptop, or router is the brain of the system that controls other components, such as memory, hard disk, and a network interface card. The main task of the CPU is to execute program instructions in the form of machine code. Machine code is platform-specific binary code consisting of zeros and ones that is not human readable. Machine code for a given program is not portable between processor architectures; for example, ARM64 architecture-based machine code is not compatible with and cannot be run on x64 architecture-based devices and vice versa. You could think of it as two gingerbread recipes, one written in English and one in Bahasa Indonesian, each providing a list of instructions. While the Indonesian alphabet uses the same 26 letters as the English alphabet, a native English speaker will not be able to read or follow the recipe written in Bahasa Indonesian.

High-level programming languages such as Python, Java, C++, and Go allow programmers to write code that is independent of the underlying hardware architecture and human readable and that provides an abstraction layer to hide low-level hardware details. For instance, Example 2-1 shows a simple computer program that allocates a few variables, stores the sum of a + b in a variable, and sends the result to the standard output (that is, the user's screen in the terminal).

Example 2-1 *High-Level C++ Source Code*

```
#include <stdio.h>

int main(void){
    int a,b,c;
    a=1;
    b=2;
    c=a+b;
    printf("%d + %d = %d\n",a,b,c);
 }
```

A compiler is a special program that translates high-level programming language source code into machine code that can be executed on a CPU. As an intermediate step, a compiler creates assembler code, which is one step away from machine code. Unlike machine code, assembler code is human readable and nicely shows the order of instructions that must be executed by the CPU to achieve the specified outcome of the high-level source code. Example 2-2 shows the same program from Example 2-1 but in assembler code.

Example 2-2 *Low-Level Assembler Source Code*

```
.LC0:
 .string "%d + %d = %d\n"
 main:
 push rbp
 mov rbp, rsp
 sub rsp, 16
 mov DWORD PTR [rbp-4], 1
 mov DWORD PTR [rbp-8], 2
 mov edx, DWORD PTR [rbp-4]
 mov eax, DWORD PTR [rbp-8]
 add eax, edx
 mov DWORD PTR [rbp-12], eax
 mov ecx, DWORD PTR [rbp-12]
 mov edx, DWORD PTR [rbp-8]
 mov eax, DWORD PTR [rbp-4]
 mov esi, eax
 mov edi, OFFSET FLAT:.LC0
 mov eax, 0
 call printf
 mov eax, 0
 leave
 ret
```

The assembler program consists of a list of instructions whose machine code counterparts will be executed one by one by the CPU at runtime. A special CPU register, generally referred to as the program counter, stores the memory address of the current instruction. Upon completion, the program counter is incremented, and the next instruction is fetched from the updated memory address to be executed. In other words, the program counter keeps track of where the CPU is in the program execution—that is, where it is in the sequence of instructions.

Don't worry if you don't understand the assembler program. The details are not relevant. What is relevant is the fact that there are different instructions, such as **push**, **mov**, **sub**, and **add**, that seem to accept one or more parameters. The supported instructions vary between hardware architectures and CPU models. The instruction set architecture (ISA) defines which instructions can be used by a software program to control the CPU. Reading such a manual reveals that instructions have the following format:

```
label: mnemonic argument1, argument2, argument3
```

where:

- *label* is an identifier (not related to MPLS labels).
- *mnemonic* is a name for a class of instructions that have the same function.
- Arguments are mandatory or optional, depending on the mnemonic.

Example 2-2 shows a label called **main**, followed by **push** (*mnemonic*) and **rbp** (*argument1*). This instruction tells the CPU to store a special register on the stack, whereas the **add eax, edx** instruction takes two arguments to perform the addition of a + b in the source code. This simple program uses common instructions, but applications in the field of artificial intelligence (AI) and machine learning (ML) use more complex and specialized instructions. In principle, there are no limits on what kind of instructions a CPU can execute, as long as it is implemented in hardware and there is a practical benefit of implementing it. The length of an instruction may vary within an ISA, depending on the underlying hardware architecture.

Finally, executing the binary yields the output shown in Example 2-3.

Example 2-3 *Output of Program Execution*

```
cisco@ubuntu-server:~/Code$ ./program
1 + 2 = 3
cisco@ubuntu-server:~/Code$
```

At this point, you might wonder about the relevance of CPU instructions, program counters, and instruction formats in a segment routing book. The coming paragraphs shed light on the analogy and emphasize similarities between computer and segment routing architectures.

Segment routing (RFC 8402) leverages source routing, which allows the source node (ingress PE node) to steer a packet flow through the SR domain. This ability is a key difference from traditional MPLS-based networks, where ingress PE nodes lack such fine-grained control over the traffic path through the network when relying on LDP labels. Traffic engineering (TE) techniques enable the optimization of traffic distribution in MPLS networks at the cost of additional protocols such as Resource Reservation Protocol (RSVP) and network state information (TE tunnels) in the network, which is challenging to operate and negatively impacts the overall network scale. Segment routing significantly simplifies the network protocol stack by superseding signaling protocols like LDP or RSVP-TE.

Instead, SR extensions elevate the underlying link-state routing protocol, providing a comprehensive view of the network topology across the entire domain, to provide the same functionality that relied on multiple protocols in the past. The interior gateway protocol (IGP) advertises *segments*, which are essentially network instructions, throughout the network, which guarantees that every node within the domain has the same view. The flooding of segments enables the IGP to replace the previously mentioned signaling protocols and facilitates moving any tunnel state information from the network to the packet headers. A segment can have global significance within the network, such as instructing nodes in the SR domain to steer traffic to a specific node, or local significance, such as instructing a specific node to steer traffic across a specific interface.

Figure 2-1 shows the two supported data planes of the segment routing architecture. SR-MPLS reuses the MPLS data plane, whereas SR IPv6 (SRv6) relies on the IPv6 data plane.

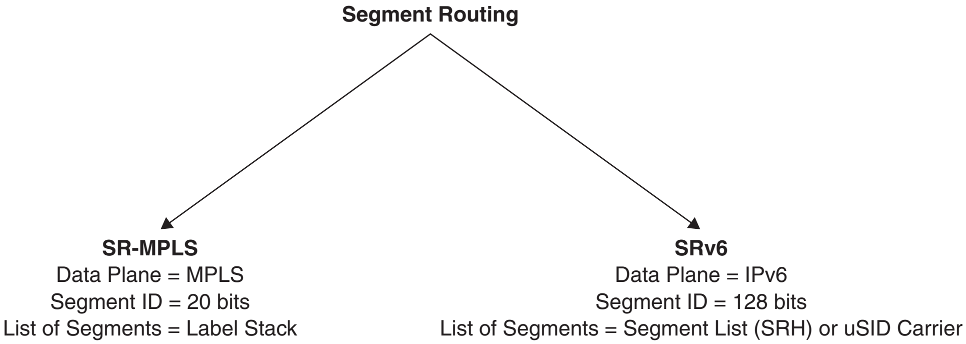


Figure 2-1 *Segment Routing Data Planes*

As previously mentioned, a segment represents a single instruction identified by a segment identifier (SID). The length of a SID depends on the underlying data plane. For SR-MPLS, the SID is 20 bits long and is written in the Label field of the MPLS header. In contrast, an SRv6 SID is a 128-bit identifier in the Destination Address field of the IPv6 header. As with the assembler program shown in Example 2-2, multiple ordered instructions can be expressed as a list of segments. A list of segments can be realized using multiple SIDs, which in the MPLS data plane results in a label stack. In the SRv6 data plane, a list of segments may be encoded using the segment routing header (SRH), a micro-SID (uSID) carrier, or a combination of both, depending on the SRv6 flavor and the number of segments. The fundamental terminology of segment routing is agnostic to the underlying data plane; the concept of a segment, SID, and list of segments applies to both encapsulation types.

Note SRv6 terms and concepts, such as SRH and uSID, are explained in Chapter 3. The different segments in SR-MPLS are presented in more detail later in this chapter, in the section "Segment Routing for MPLS (SR-MPLS)."

You may have come across the term *network as a computer* in the context of segment routing, in reference to the network as a large distributed system where several devices work together to execute a network program consisting of a list of instructions or segments. All nodes within an SR domain must speak a common language to be able to interpret the segments correctly. SR can be applied to both the MPLS and IPv6 architectures, which means that nodes within an SR domain are not limited to networking devices if they understand the underlying data plane. This is especially true for IPv6, which is widely supported across a range of different networking nodes from the Internet of Things (IoT) in the industry to containers in the data center. Figure 2-2 shows an imaginary local weather station with sensors in three different locations and some services running in a data center (DC).

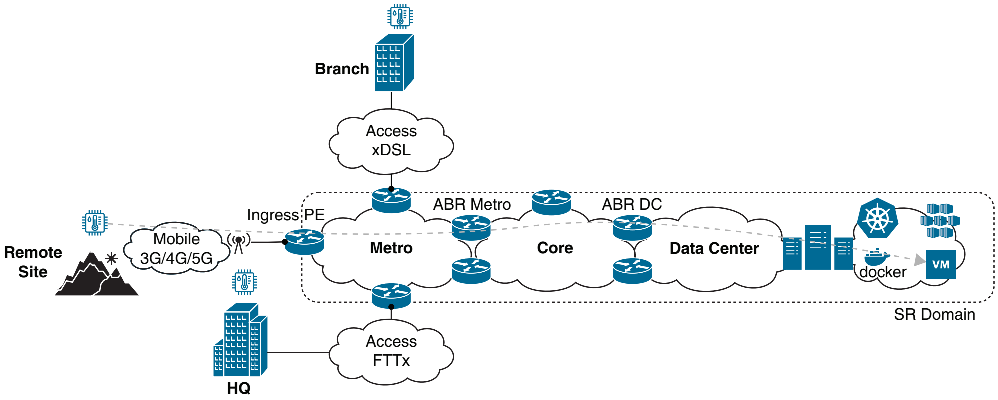


Figure 2-2 *Weather Station Network Topology*

The sensors connect to a local service provider (SP) using different access technologies. Each sensor measures temperature, humidity, and barometric pressure on a regular basis and transmits the data to a microservice hosted in a remote data center (on the right-hand side of the figure). The collected data is processed, stored, and evaluated every 24 hours to provide the weather forecast for the next seven days. It goes without saying that meteorology is far more complex than presented here, but this illustration will suffice for our example.

The SR domain in our example includes metro, core, and data center, up to and including the virtual machine or container, which means that segments could be executed by any of the nodes belonging to the SR domain. Within an SR domain, different roles can be distinguished:

- **Source/ingress node:** Handles the traffic as it enters the SR domain.
- **Transit node:** Handles the forwarding of traffic within the SR domain.
- **Endpoint/egress node:** Handles the traffic as it leaves the SR domain.

In traditional Layer 3 virtual private network (VPN) services, service provider and customer networks are isolated logically using virtual routing and forwarding (VRF) instances or access lists on the service edge to protect the SP infrastructure. Consequently, VRF instances or access lists are used to enforce the demarcation point of the SR domain. In our example, all three weather station sensors are isolated from the SP through VRF instances on the PE node, which means a network program can only be initiated by the ingress PE device receiving customer traffic.

Unlike in traditional software development, with segment routing there are no high-level network programming languages available. Instead, a network program is defined as an ordered list of segments, also known as an *SR policy*, that steers a packet flow along a desired path in the network. SR policies are source-routed policies identified through the tuple, such as headend, color, or endpoint. Headend and endpoint should be self-explanatory; the 32-bit color value identifies the intent or objective of the policy. The endpoint and color are used as identifiers to steer traffic into the corresponding SR policy. Examples of such an intent are low latency or MACsec encrypted paths from the headend to the endpoint. The source routing is crucial in moving the traffic engineering tunnel state from intermediate routers to the packet headers imposed by the ingress node through an SR policy.

Complementary information on how to implement such traffic engineering capabilities using the IGP is provided in the section “IGP Flexible Algorithm (Flex Algo) (RFC 9350),” later in this chapter. Example 2-4 shows an imaginary SR policy that defines a loose path from the ingress PE node (source node) to the container (endpoint node) hosting the weather application via two transit nodes. Note that there are two area border routers (ABRs) in the metro and the data center, which may result in equal-cost multipath routing (ECMP). If desired, a more restrictive path could be defined, such as using a specific ABR or only traversing the core over MACsec-encrypted links.

Example 2-4 *Network Program Pseudocode*

```
policy weather-app-policy
 1 goto ABR Metro
 2 goto ABR DC
 3 goto container weather-app
```

Figure 2-3 shows the ordered list of segments expressed in this pseudocode.

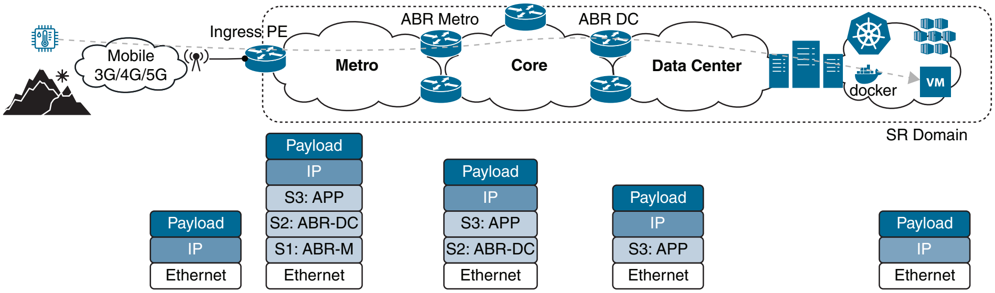


Figure 2-3 *Network Program Segment Routing Policy (SR-MPLS)*

The ingress PE node imposes one or more additional headers to encapsulate the original customer packet. Note that the exact headers depend on the underlying data plane, as discussed in detail in the section "Segment Routing for MPLS (SR-MPLS)," later in this chapter, and in Chapter 3. The additional encapsulation overhead is negligible in most cases and justified by the significant scalability gains in the backbone network achieved by transferring the tunnel state information from the network to the packet. In the case of SR-MPLS, the length of the list of segments decreases as the network program is executed. The first segment is executed by one of the ABR metro nodes. The metro ABR pops its own instruction from the stack and forwards the packet toward an ABR DC, which pops its own instruction and forwards the packet toward the weather-app container. Eventually, the packet reaches its destination, which in our example is the SR-aware weather-app container that decapsulates and processes the inner IP packet. Note that this example excludes a few details, such as penultimate hop popping (PHP) and the BGP service label for simplicity.

It should be becoming clear now that the execution of segments in a segment routing domain and the execution of instructions in computer architectures share several fundamental principles. In fact, those similarities are even more prominent with SRv6, as you will see in Chapter 3, which covers the Segments Left field of the SRH and the SRv6 SID format that are comparable to the program counter and instruction format, respectively.

The segment routing ecosystem encompasses a wide variety of Internet Engineering Task Force (IETF) standards and drafts across numerous working groups. The standardization process for segment routing has been progressing at an impressive pace, and most key drafts have become proposed standards. One exception worth highlighting here is the SRv6 compression drafts that are in the later stages of the standardization process. The successful mass-scale rollout of SR lead operators shows that there is no reason to delay the SR adoption.

Figure 2-4 displays a selection of the most important building blocks that make up segment routing (RFC 8402) and the segment routing policy architecture (RFC 9256).

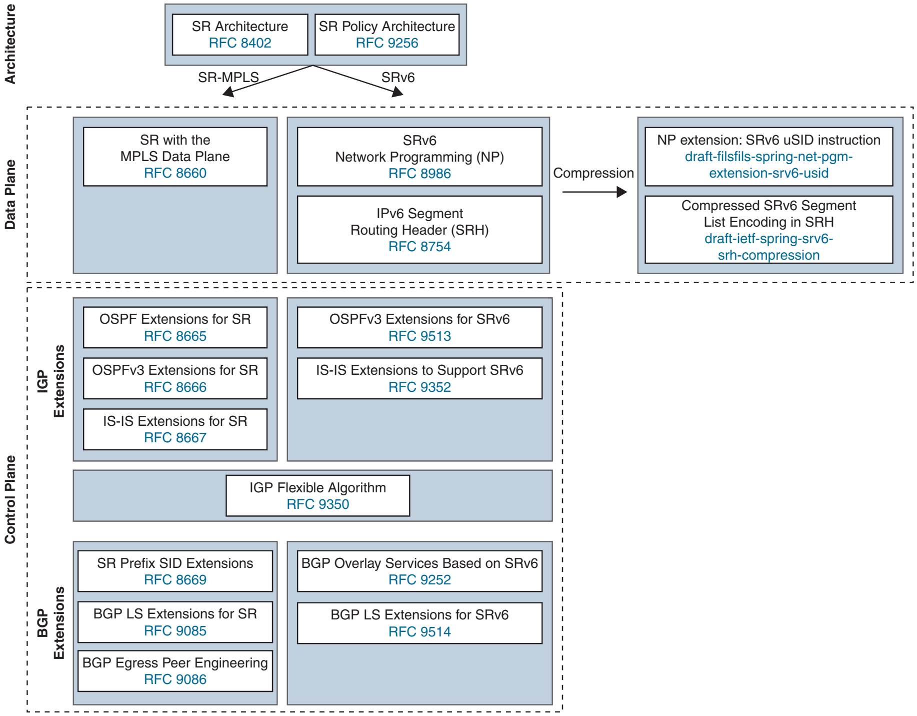

Architecture
SR Architecture
RFC 8402
SR Policy Architecture
RFC 9256
SR-MPLS
SRv6
Data Plane
SR with the MPLS Data Plane
RFC 8660
SRv6 Network Programming (NP)
RFC 8986
IPv6 Segment Routing Header (SRH)
RFC 8754
Compression
NP extension: SRv6 uSID instruction
draft-filsfils-spring-net-pgm-extension-srv6-usid
Compressed SRv6 Segment List Encoding in SRH
draft-ietf-spring-srv6-srh-compression
Control Plane
IGP Extensions
OSPF Extensions for SR
RFC 8665
OSPFv3 Extensions for SR
RFC 8666
IS-IS Extensions for SR
RFC 8667
OSPFv3 Extensions for SRv6
RFC 9513
IS-IS Extensions to Support SRv6
RFC 9352
IGP Flexible Algorithm
RFC 9350
BGP Extensions
SR Prefix SID Extensions
RFC 8669
BGP LS Extensions for SR
RFC 9085
BGP Egress Peer Engineering
RFC 9086
BGP Overlay Services Based on SRv6
RFC 9252
BGP LS Extensions for SRv6
RFC 9514

Figure 2-4 *Segment Routing Architecture*

> **Note** Some of the official standards and draft titles have been shortened or modified in Figure 2-4 for better readability. This book covers many of the drafts and proposed standards in more detail, accompanying the somewhat dry theory with visual illustrations and packet captures.

Problem Description and Requirements

Before we delve into the details of segment routing, it is helpful to recall the problems and requirements segment routing aims to address and what it does not address. RFC 7855 takes into account many of the shortcomings of MPLS described in Chapter 1 and proposes a new network architecture based on source routing. These are the key take-aways in the RFC:

- The SPRING architecture must be backward compatible. That is, SPRING-capable and non-SPRING-capable nodes must be able to interoperate for both MPLS and IPv6 data planes.
- Existing MPLS VPN services must be deployable using the SPRING architecture without the need for additional signaling protocols.

Chapter 3

What Is Segment Routing over IPv6 (SRv6)?

Introduction

This chapter covers the theory behind the Segment Routing over IPv6 (SRv6) data plane encapsulation implementation. It provides valuable decision-making process inputs and outlines potential pitfalls when evaluating the network architecture evolution to Segment Routing over IPv6. In addition, the section "SR-Powered Network Evolution" describes the network evolution journey that began with the introduction of Segment Routing for MPLS (SR-MPLS) and ends with a converged SDN transport network based on SRv6.

Segment Routing over IPv6 (SRv6)

This section introduces SRv6, which shares many fundamental concepts with SR-MPLS. Although some operators might view SR-MPLS as a transitional step toward SRv6, this chapter shows that SRv6 is a superior solution that effectively addresses the challenges associated with MPLS discussed in the section "Challenges and Shortcomings of MPLS," in Chapter 1, "MPLS in a Nutshell." While SR-MPLS is already well established, a select number of compression-related SRv6 extensions are still in the process of being standardized as of this writing. The IETF has been advancing at an impressive pace, and all the major key drafts have been successfully standardized, achieving RFC status. This standardization marks a significant milestone in the evolution of SRv6, showcasing its readiness for widespread deployment and the promise of enhanced network efficiency.

Since SRv6 relies on the IPv6 data plane, it is crucial to have a solid understanding of IPv6 encapsulation and the IPv6 header. In fact, as you will see in this chapter, the vast majority of SRv6 use cases rely on IPv6 routing using an IPv6 header without any extension headers.

IPv6 for SRv6 Recap

Figure 3-1 shows the format of the IPv6 header, as specified in RFC 8200.

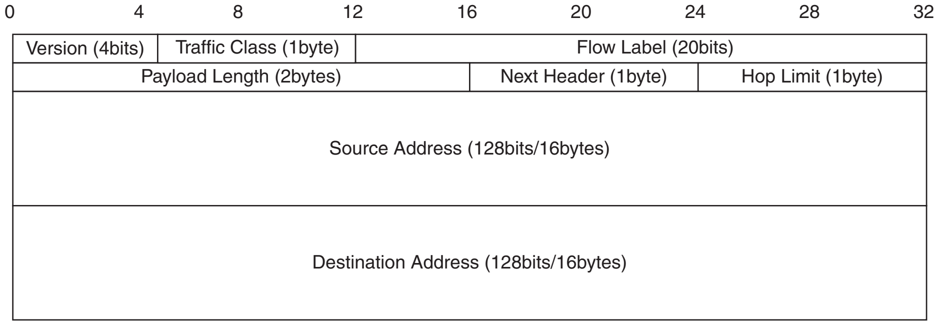


Figure 3-1 *IPv6 Header*

The IPv6 header consists of the following fields:

- **Version (4 bits):** Specifies the version of IP; set to 6 for IPv6.
- **Traffic Class (8 bits):** Used for traffic management (QoS) based on DSCP (6-bit) and ECN (2-bit).
- **Flow Label (20 bits):** Used for encoding entropy of the payload and subsequent flow hashing (load balancing).
- **Payload Length (16 bits):** Specifies the length of the payload following the IPv6 header.
- **Next Header (8 bits):** Identifies the header following the IPv6 header (for example, IPv4, IPv6, Ethernet, ICMP, TCP).
- **Hop Limit (8 bits):** Equivalent to the Time to Live (TTL) field of the IPv4 header.
- **Source Address (128 bits):** Identifies the source of the packet.
- **Destination Address (128 bits):** Identifies the destination of the packet.

Most of the fields are self-explanatory or easy to grasp. However, special attention should be paid to the Traffic Class, Flow Label, and Next Header fields.

The Traffic Class field is used for quality of service (QoS) marking, which involves Differentiated Services Codepoint (DSCP) and Explicit Congestion Notification (ECN). The 6-bit value of DSCP covers the decimal range from 0 to 63, which makes it possible to distinguish more than eight traffic classes. The 3-bit value of MPLS EXP in SR-MPLS is a significant limitation with SR-MPLS.

The Flow Label field facilitates efficient flow classification in combination with other IPv6 header fields, such as the Source Address and Destination Address fields. A sequence of packets belonging to the same Layer 3 flow are generally classified based on the 5-tuple of network addresses, transport protocol, and transport ports. Note that not all of those identifiers may exist in a flow, depending on the payload (for example ICMP), or they may be unavailable due to encryption or fragmentation. Layer 2 traffic flows usually take into account data link layer information for classification and may or may not include some of the higher-layer protocol information. Often, flow classification is not only vendor dependent but also platform dependent, with some devices supporting 7 or more-tuple flow classification, taking into account one or more MPLS labels.

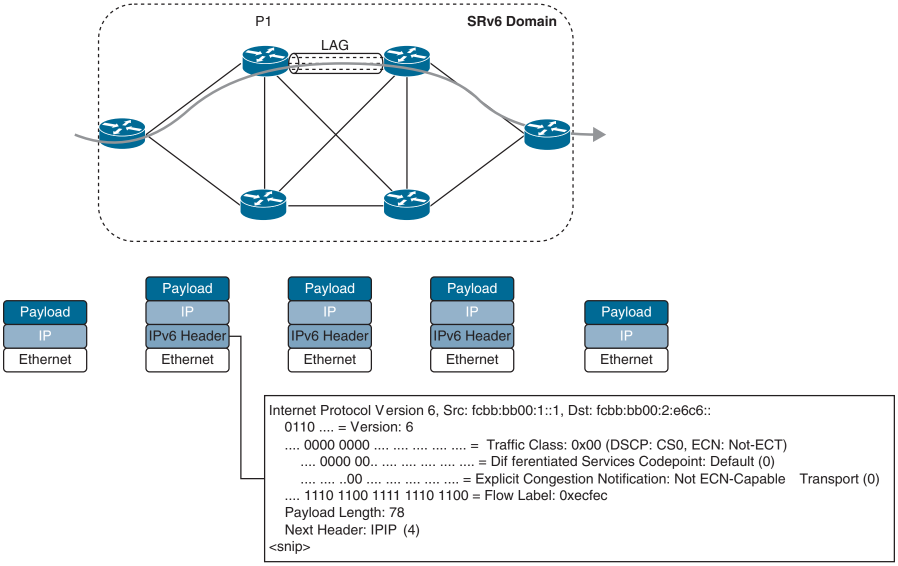
The Flow Label field is a radical simplification for IPv6 compared to IPv4 or MPLS. Instead of cumbersomely inspecting a packet and trying to figure out where the relevant fields are located within the packet to extract the 5+-tuple, IPv6 uses a 3-tuple consisting of Source Address, Destination Address and Flow Label fields. Having all those fields at fixed positions within the IPv6 header simplifies the extraction of flow identifiers and consequently the hardware implementation of this process.

The Flow Label value is computed by the source node and not changed by transit nodes along the path. The source node in an SRv6 domain is usually an ingress PE device, which encapsulates the received packet coming from the edge into an outer IPv6 header with an optional segment routing header (SRH) extension header. The exact algorithm to compute the hash for the Flow Label value is implementation specific and may differ between vendors or platforms. However, depending on the type of service, different fields are considered, such as the following common identifiers:

- **Layer 2 VPN service:** Source and destination MAC addresses and source and destination IP addresses (IP payload only)
- **Layer 3 VPN service:** Source and destination IP addresses, transport protocol, and source and destination ports

This list is an example, and different implementations may consider additional fields. It is important to understand that the Flow Label value is computed only once in the network, at the source node, which is service aware; that is, Layer 2 or Layer 3 VPN services can be easily distinguished, and the tuple used for hashing can be extracted before additional encapsulation takes place. After the hash has been computed, it is written to the Flow Label field, which is, in turn, used by all transit nodes. In essence, the hard work of computing a proper hash needs to be performed only once by the source node, and all other nodes along the path can take advantage of this hash, which greatly reduces the complexity of flow classification to achieve proper load balancing for ECMP routing or LAG hashing.

Figure 3-2 shows an example of a traffic flow entering an SRv6 domain. The network is highly symmetric, with one core link relying on a link aggregation group (LAG). The ingress PE device encapsulates the received packet from the edge into an IPv6 header and populates the Flow Label field with the computed 20-bit hash (0xecfec). The packet entering the SRv6 domain is an IPv4 packet, as you can see from the Next Header field of the IPv6 header.



```
Internet Protocol Version 6, Src: fcbb:bb00:1::1, Dst: fcbb:bb00:2:e6c6::
   0110 .... = Version: 6
   .... 0000 0000 .... .... .... .... .... =  Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
      .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
      .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification: Not ECN-Capable   Transport (0)
   .... 1110 1100 1111 1110 1100 = Flow Label: 0xecfec
   Payload Length: 78
   Next Header: IPIP  (4)
<snip>
```

Figure 3-2 *IPv6 Flow Label and Capture*

Note Many of the network figures related to MPLS/SR-MPLS in this book include an inner IP header with payload to emphasize the fact that MPLS sits between Layer 2 (the data link layer) and Layer 3 (the network layer) in the OSI model, which is why it is sometimes referred to as a Layer 2.5 networking protocol. SRv6 goes back to the roots of the OSI model and no longer relies on this shim layer. SRv6 network figures in this book are generally drawn with an inner payload only, unless the context asks for more detailed inner header information, such as IPv4/IPv6 or Ethernet.

The flow label must not change en route, and P1 classifies the flow based on the IPv6 source address, destination address, and flow label. P1 chooses the ECMP path and physical link of the LAG based on the hash of the IPv6 header, as shown in Figure 3-2.

The Next Header field identifies the upper layer protocol, which follows immediately after the IPv6 header. A key difference between IPv4 and IPv6 is the flexible support for extensions and options in IPv6, where *extension headers* are placed between the IPv6 header and upper layer protocols (for example, TCP or UDP).

Note IP protocol numbers and IPv6 extension headers registered with IANA are listed at https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml and https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#extension-header.

The routing header for an IPv6 extension is a central puzzle piece of the SRv6 solution as it allows the insertion of an optional segment routing header (SRH) after the IPv6 header.

The details of the SRH are introduced later in this chapter, in the section "IPv6 Segment Routing Header (SRH) (RFC 8754)."

SRv6 Network Programming (RFC 8986)

RFC 8986 lays the foundation of the segment routing architecture in the IPv6 data plane. Network instructions have to be encoded into the IPv6 header, which differs fundamentally from MPLS, where each instruction is represented by a label. Many fundamental concepts covered in Chapter 2, "What Is Segment Routing over MPLS (SR-MPLS)?" are the same for SRv6, though. An SRv6 SID is still associated with a segment, but instead of using an MPLS label, it is now represented as an IPv6 address. The IPv6 destination address in the outer IPv6 header is set to the SRv6 SID, which represents a network program, including a single instruction or an SR policy with a single segment. SRv6-unaware transit nodes forward an SRv6 SID based on the longest-prefix-match lookup on the IPv6 destination address. An IPv6 address associated with an SRv6 SID has a special format.

SRv6 Segment Identifier (SID)

SRv6 SIDs are 128-bit long IPv6 addresses that follow the format shown in Figure 3-3:

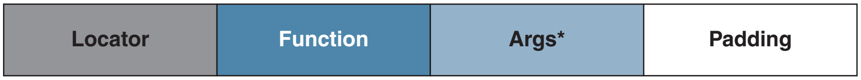


Figure 3-3 *SRv6 SID Format*

- Locator:
    - Most significant bits
    - Routable part, which points to the parent node that instantiated the SID
    - Advertised through a link-state IGP (IS-IS or OSPF)
    - Should be unique within the SRv6 domain, except for SRv6 anycast locators
- Function:
    - Identifies a locally significant behavior of the parent node
- Arguments:
    - Least significant bits
    - One or more input arguments to the function (for example, service or flow information)
    - Optional

The length of the Locator (L), Function (F), and Arguments (A) fields are flexible as long as the total length is less than or equal to 128 bits. If the total length is less than 128 bits, the SID should be padded to 128 bits with zeros. As mentioned previously, the Arguments field is optional.

As shown in Figure 3-4, the Locator field may be expressed as two different fields: SID block (B) and Node ID (N). A common format for early SRv6 deployments was to allocate B::/48 for the SID block and B:N::/64 for the locator. In Cisco documentation, this is commonly referred to as *base format*. The section "SRv6 Locator Addressing Scheme," later in this chapter, discusses alternative locator assignments suited for large-scale deployments. For now, this basic split into SID Block, Node ID, and Function fields will suffice as an introduction to SRv6 SIDs.

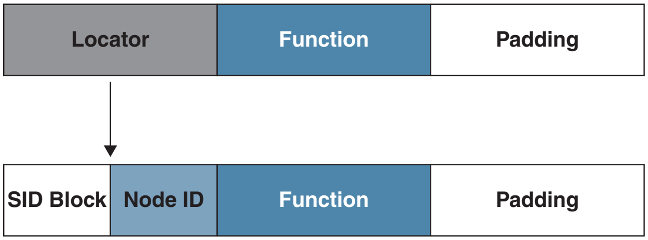


Figure 3-4 *SRv6 SID Format (Simplified)*

Let us look at the example presented in Figure 3-5:

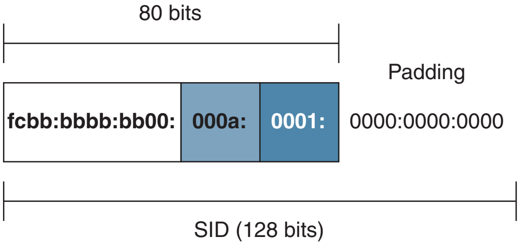


Figure 3-5 *SRv6 SID Format (Example)*

- A service provider allocates the SRv6 SID block—for example fcbb:bbbb:bb00::/48—from the unique local address (ULA) space for SRv6 locators in the network shared by all SRv6 nodes.
- Router 10 would be assigned the locator fcbb:bbbb:bb00:000a::/64 (L = 64 bits).
- Router 10 locally allocates function 0x0001 (F = 16 bits) without any arguments (A = 0 bits) for its SRv6 SID. The sum of L + F + A equals 80, which means that the remaining 48 bits must be padded with zero since SRv6 SIDs are 128-bit addresses.
- The resulting SRv6 SID associated with the segment to Router 10 equals fcbb:bbbb:bb00:a:1::.

Note IPv6 addresses are expressed in hexadecimal, where 10 translates to 0xA and not 0x10, which would be 16.

For reachability and backward compatibility between SRv6-capable and IPv6-capable nodes, SRv6 nodes advertise the locator (for example, /64) as an IPv6 prefix in the link-state IGP, as shown in Figure 3-6. The locator prefixes act as aggregate prefixes for source and transit nodes to perform longest-prefix-match lookups on SRv6 SIDs and forward the packets accordingly.

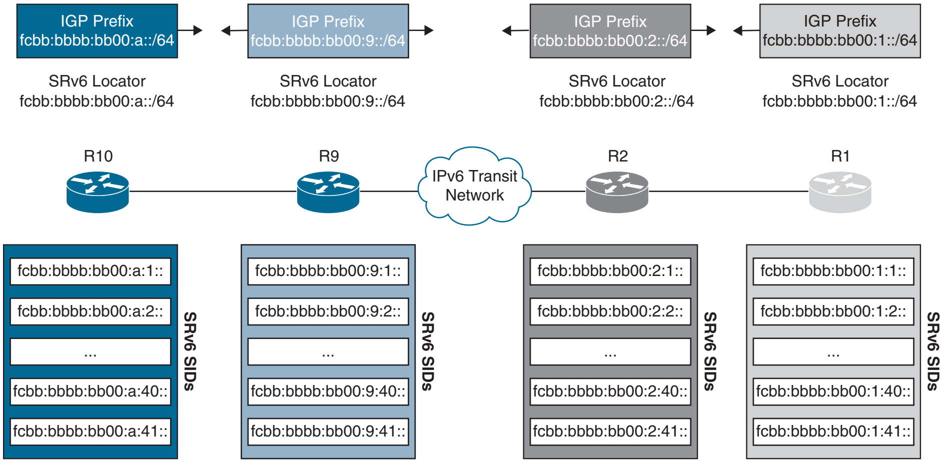


Figure 3-6 *SRv6 Locator and IGP Prefix*

Therefore, it is possible to provision SRv6 services over a native IPv6 network as long as the service edge is SRv6 capable. Obviously, certain SRv6 functionality, such as Topology Independent Loop-Free Alternate (TI-LFA) or traffic engineering, will not be available on native IPv6 transit nodes. This is discussed in more detail later in this chapter, in the section "IPv6 Segment Routing Header (SRH) (RFC 8754)." At this point, it may still be confusing how SRv6 SIDs can be used to provision services. A simplified analogy using familiar concepts from SR-MPLS will hopefully shed some light. Figure 3-7 shows the data plane encapsulation for both SR-MPLS and SRv6. It should be clear by now that SRv6 does not use labels, so how are transport and service identifiers encoded using a single IPv6 header?

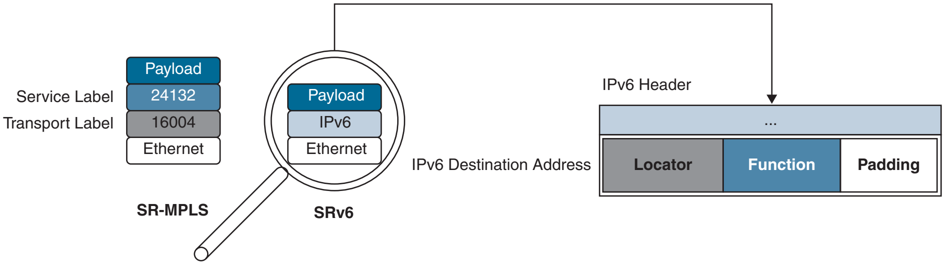


Figure 3-7 *SR-MPLS and SRv6 SID Analogy*

Chapter 5

Migrating to Segment Routing

Network operators face the ongoing challenge of staying competitive, which often requires proactively adopting new technologies. This brings us to the concept of *migration*, which involves shifting to newer technological landscapes. But what does migration really entail? It is a concept that is deceptively simple to state yet complex to execute.

At its core, migration involves moving from an existing network configuration to an enhanced, desired state. This journey unfolds a tapestry of complexity as it demands adaptations in hardware, software, configurations, IT systems, personnel, and business processes to accommodate the new technology. For network operators, especially those managing large-scale or diverse networks, the transformation is multifaceted and challenging. It can range from deploying entirely new infrastructures and overhauling networks to replacing outdated equipment, upgrading existing hardware, implementing new software solutions, and embracing novel protocols.

With the introduction of two segment routing (SR) technologies and their advantages in the previous chapters, this chapter provides a practical roadmap for implementing SR, whether you're starting afresh with a greenfield approach, or using a brownfield method, which involves integration with existing systems. This chapter presents simple steps to mitigate the risks associated with transitioning services from MPLS to SR.

This chapter covers the following topics:

- Deployment models and strategies for achieving connectivity between MPLS and SR networks during the migration phase, which is a critical period during which a network incrementally adopts SR
- Specifics of migrating from an LDP network to SR-MPLS and details of both deployment models and MPLS and SR interconnectivity options
- Three distinct strategies for migrating from MPLS to SRv6
- A roadmap for migration to an SRv6 network for four different MPLS networks

Note In this chapter, *MPLS* refers to both LDP and SR-MPLS, and *SR* denotes SR-MPLS and SRv6.

The migration process typically concerns the transition of customer edge (CE) or provider edge (PE) devices from one network environment to another. Although this chapter uses IPv4 to illustrate service migrations, the concepts are equally applicable to IPv6 services.

Note This chapter does not cover the decommissioning of legacy networks after a migration, although that is an important step.

Deployment Models

Greenfield network deployment refers to the installation of a network where previously there was none. This term is derived from the construction industry, where new development on previously undeveloped land is termed a *greenfield development*. An important advantage of greenfield deployment is the opportunity to implement cutting-edge technology solutions from the ground up, free from the constraints or dependencies of existing infrastructure, software, biases, or business processes. In the context of SR, a greenfield deployment refers to building a separate SR network and then migrating the services from the MPLS network to the SR network. During the migration phase, services may connect with both the SR and MPLS networks, and so interworking is essential to maintain seamless service connectivity across the networks. Implementing interworking might necessitate additional hardware and software, which can be phased out after the completion of the service migration.

In contrast to a greenfield deployment, a *brownfield deployment* involves an upgrade or expansion of an existing network. This type of deployment involves installation and configuration of new hardware or network technology that is designed to coexist with the legacy network. One benefit of brownfield development is the ability to enhance existing technology solutions within established business processes; in addition, an organization can avoid extra capital expenditure on new infrastructure by undertaking a brownfield development. However, brownfield projects also come with their own challenges, including the need for a comprehensive and accurate understanding of the existing network's limitations and issues with legacy infrastructure that can potentially slow the development process and inflate overall costs.

A brownfield deployment involves activating SR in the MPLS network. This process can be conducted in phases during several maintenance windows. Throughout these periods, SR and MPLS coexist within the network, which introduces increased complexity in terms of features and configurations. However, it also reduces the effort and mitigates risks typically associated with maintenance windows.

Migration Strategies

Transitioning the services to SR-MPLS or SRv6 in a network can be achieved through a multitude of viable methodologies, and this section illustrates several strategies. When embarking on a greenfield deployment and establishing a new SR core alongside metro and access layers, there are two principal strategies that stand out for migrating services from MPLS to SR network. The first strategy relies on the ability of the PE routers to simultaneously support MPLS and SR during the migration phase, and the second strategy depends on interworking.

Figure 5-1 illustrates the first strategy, which leverages an interworking gateway to sequentially transition the MPLS PE devices to the SR network in multiple maintenance windows. This strategy assumes that all the MPLS PE devices support SR.

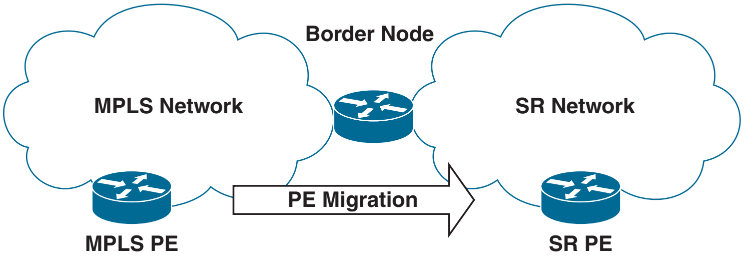


Figure 5-1 *Interworking Greenfield Strategy: Migrating PE Devices to a New SR Network*

In cases where the MPLS PE devices do not support SR, the CE devices can be migrated from the MPLS PE devices to the SR PE devices, as shown in Figure 5-2.

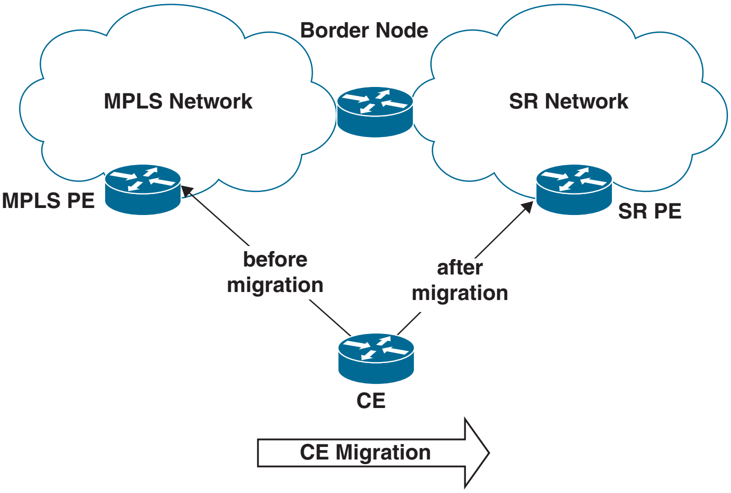


Figure 5-2 *Interworking Greenfield Strategy: Migrating CE Devices to the New SR Network*

Later sections of this chapter covering the interworking strategy involve the migration of PE devices or CE devices:

- The section “Building a New SR-MPLS Network” covers migration from LDP to SR-MPLS, where a border node is the gateway between the LDP and SR-MPLS networks
- The section “Building a New SRv6 Network Using an SRv6 IWG” covers migration from MPLS to SRv6 where an SRv6 interworking gateway (IWG) acts as the IWG between the MPLS and SRv6 networks, and the section “Building a New SRv6 Network Using Inter-AS Option A" presents a solution using ASBRs as IWGs between the two networks. The information on the migration of CE devices is also valid for the migration of PE devices.

Figure 5-3 shows the second strategy, where PE devices can be concurrently connected to both the MPLS and SR networks, thus removing the necessity for an interworking gateway between the two networks.

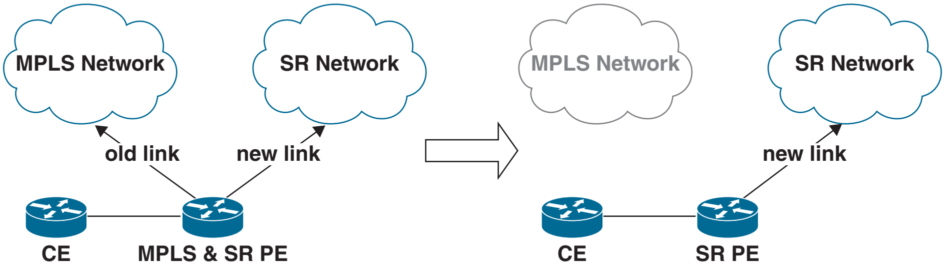


Figure 5-3 *Dual-Homed Greenfield Strategy : PE Devices Connected to the MPLS and SR Networks*

The PE devices must have sufficient port capacity, CPU, and memory to connect to the MPLS and SR networks simultaneously. The dual-homed strategy for migration from MPLS to SR is described in the section "SRv6 Network Using Dual-Connected PE Devices." The MPLS network can be decommissioned once all the MPLS PE devices have been migrated to the SR network.

A brownfield deployment uses a coexistence strategy, incorporating SR into the existing network as illustrated in Figure 5-4. This strategy entails activating SR in specific parts of the MPLS network and progressively expanding its scope in multiple migration windows.

For this approach to work, the existing network must be able to support SR. Throughout the transition phase, PE and P devices are configured to support MPLS and SR concurrently. When all the devices in the network are migrated to SR, the MPLS-related configuration can be removed from the network.

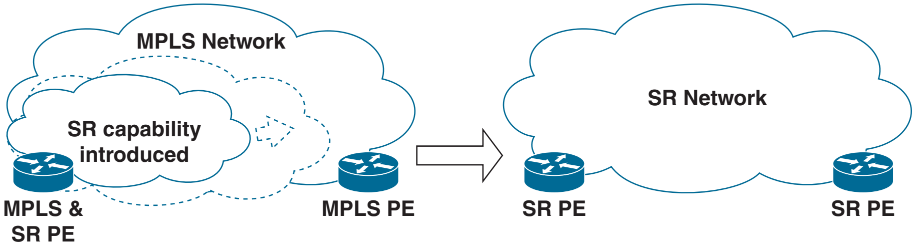


Figure 5-4 *Coexistence Brownfield Strategy: Enabling SR in an MPLS Network*

The coexistence strategy is discussed in the following sections:

- LDP to SR-MPLS migration is presented in the section "Enabling SR-MPLS in an Existing Network (Coexistence)."
- MPLS to SRv6 migration is explained in the section "SRv6 Network Using Dual-Connected PE Devices." Although this section shows the migration strategy for a greenfield deployment, it is also valid for a brownfield deployment.

Figure 5-5 shows another SRv6-specific brownfield strategy, which involves expanding the coverage of SRv6-based network services between PE devices connected to an existing IPv6 network.

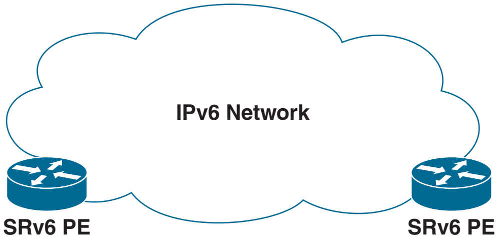


Figure 5-5 *IPv6 Backhaul Brownfield Strategy: SRv6-Based Services over an IPv6 Network*

Since the data packets from an SRv6 PE device are IPv6 packets with optional extension headers, they can be transported across any IPv6 network to facilitate SRv6-based service connectivity. However, this strategy comes with certain limitations, like the lack of TI-LFA and SRv6 traffic engineering in the native IPv6 network.

Note Migration of SRv6-based services using an existing IPv6 network is not covered in this chapter.

Chapter 6

L2VPN Service Deployment: Configuration and Verification Techniques

In this chapter, we explore L2VPN overlay services established across the SRv6 transport underlay frameworks we've already examined. This chapter outlines fundamental approaches for configuration and methods for confirming the integrity of L2VPN service structures. Keep in mind that the content provided here does not delve extensively into L2VPN services. Instead, this chapter serves as a primer on the transition and deployment of well-established L2VPN technologies within SRv6 transport infrastructures.

If there is a requirement for Layer 2 connectivity, a service provider must set up an L2VPN service using technologies such as Virtual Private LAN Service (VPLS), Virtual Private WAN Service (VPWS), or the more recent advancement in L2VPN technology, Ethernet VPN (EVPN). These overlay services are then transported across a unified core transport network. Transport networks capable of handling L2VPN can offer Ethernet LAN (E-LAN), Ethernet Tree (E-Tree), or Ethernet Line (E-Line) services. In this setup, the service provider maps the incoming customer Layer 2 traffic into bridge domains or establishes point-to-point circuits. These bridge domains or point-to-point circuits are then interconnected across the transport network with other customer site Layer 2 bridge domains or point-to-point circuits. In this way, L2VPN services facilitate the extension of subnets from one end to the other, enabling the provision of managed services such as point-to-point, Internet connectivity, intranet, and extranet services to end customers. Figure 6-1 provides a high-level illustration of how L2VPN overlay services are transported over an SRv6 core network, which will be the primary focus of this chapter."

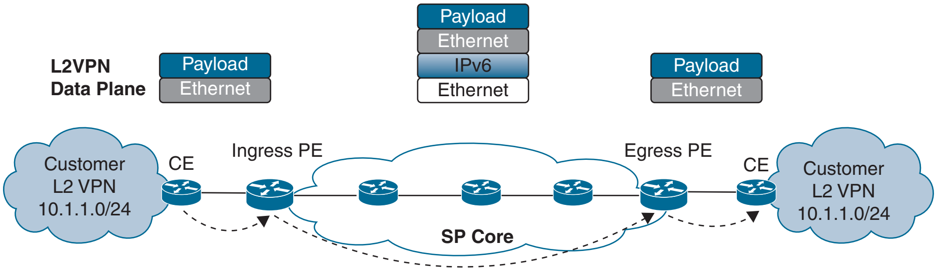


Figure 6-1 *L2VPN Connectivity Across an SRv6 Core Network*

L2VPN (EVPN)

The introduction of Multiprotocol Label Switching (MPLS) in RFC 3031, with its highly efficient and flexible data transportation capabilities through the use of MPLS labels, allowed for fast and efficient forwarding decisions without the need for complex IP lookups at each hop. MPLS is protocol agnostic, meaning it can transport packets of various network protocols, such as IP and Ethernet packets. This flexibility is a key part of MPLS's utility in creating sophisticated and scalable network services, including Layer 2 virtual private networks (L2VPNs).

MPLS L2VPN services provide a transport mechanism for Layer 2 frames between multiple customer sites across an MPLS backbone. These services essentially allow the extension of customer Layer 2 networks across geographically dispersed locations, making it possible to create a VPN that emulates a single LAN segment to the customer. L2VPNs are built using two main architectures: Virtual Private LAN Service (VPLS) and Virtual Private Wire Service (VPWS). VPLS provides a multipoint-to-multipoint service, emulating an Ethernet LAN, while VPWS offers point-to-point connectivity, similar to a traditional leased line. Both VPLS and VPWS are essentially transported across a core network through the use of MPLS pseudowires (PW). A pseudowire is simply an emulated point-to-point connection established across a packet-switched network (PSN) that uses Label Distribution Protocol (LDP) for setting up the pseudowire circuits.

Over the past 15 years, Ethernet VPN (EVPN) has begun to gain traction with service providers and large enterprises, and it is now seen as the logical evolution from the VPLS architecture for Layer 2 provisioning. Limitations that are inherent to VPLS lack of multipathing and multihoming capabilities, lack of multicast optimization and redundancy, among others are resolved through EVPN.

EVPN uses Border Gateway Protocol (BGP) to address these limitations via the control plane, whereas VPLS is inherently a data plane learning and forwarding solution. BGP as its control plane protocol provides more scalability than the flooding and learning mechanism used in VPLS. EVPN can handle a larger number of endpoints without suffering from the same level of complexity and resource consumption that VPLS might encounter due to network expansion. EVPN is able to multicast traffic more efficiently through the use of inclusive multicast Ethernet routes, eliminating the requirement to flood multicast traffic to all endpoints. VPLS, in contrast, typically floods multicast traffic across the entire Layer 2 domain. The multihoming capabilities inherent with EVPN enable single customer edge (CE) routers to connect to multiple provider edge (PE) devices for increased redundancy and load balancing. More granular traffic isolation through unique route targets (RTs) and route distinguishers (RDs) for different services is an additional benefit of EVPN. This is an improvement over VPLS, which typically relies on a single broadcast domain for all connected sites. EVPN uses BGP to advertise MAC addresses, leading to more optimal forwarding paths and faster convergence in the event of network failures. VPLS, on the other hand, relies on traditional MAC learning, which can be slower to converge. EVPN supports both Layer 2 VPN and Layer 3 VPN services, allowing for integrated routing and bridging in the same service instance. This provides greater flexibility in network design compared to VPLS. EVPN is therefore the superior option for Layer 2 VPN services, offering enhanced robustness, scalability, and flexibility over VPLS.

Metro Ethernet Forum (MEF) is an industry consortium that defines standards for carrier Ethernet services. Within the Metro Ethernet Forum 3.0 (MEF 3.0) umbrella standard, several subscriber and operator services standards are defined. One of them, MEF 6.3, is a specification that outlines various Ethernet services and attributes from the perspective of the subscriber. It defines the characteristics and types of Ethernet services that service providers can offer to their subscribers. Figure 6-2 provides a diagrammatic overview of these subscriber services.

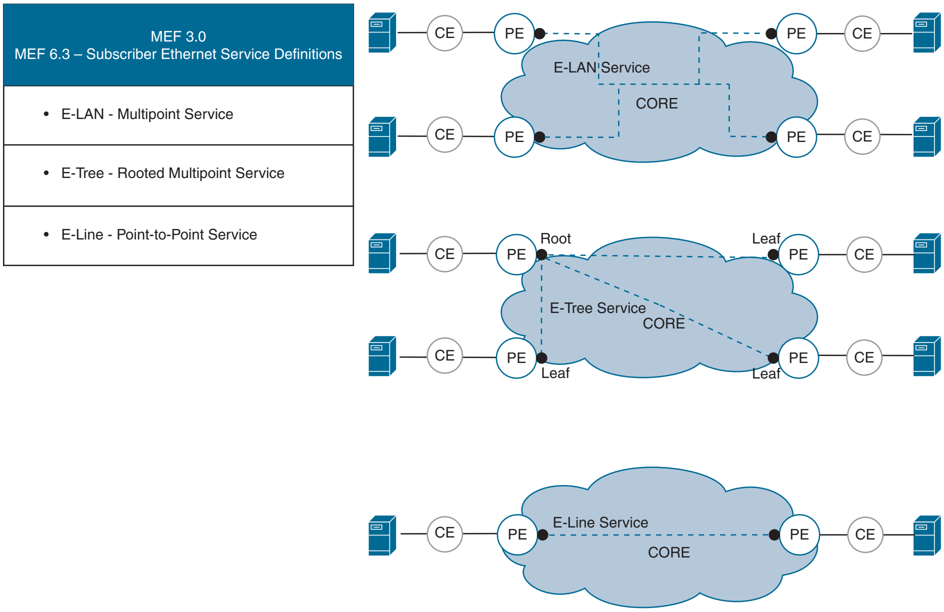

MEF 3.0
MEF 6.3 – Subscriber Ethernet Service Definitions
• E-LAN - Multipoint Service
• E-Tree - Rooted Multipoint Service
• E-Line - Point-to-Point Service
CE
PE
E-LAN Service
CORE
PE
CE
Root
Leaf
E-Tree Service
CORE
Leaf
Leaf
E-Line Service
CORE

Figure 6-2 *MEF 6.3: Subscriber Ethernet Service Definitions*

Figure 6-3 illustrates how EVPN services map to the MEF 6.3 subscriber service definitions:

- **E-LAN service (multipoint-to-multipoint connection):** EVPN E-LAN interconnects multiple endpoints in a multipoint-to-multipoint fashion that allows for any-to-any connectivity between customer sites, similar to traditional Ethernet LAN services.
- **E-Tree service (rooted multipoint connection):** EVPN E-tree involves designating certain sites as “root” or “hub” sites and others as “leaf” sites, with traffic flowing from leaf to root and from root to leaf but not directly between leaf sites.
- **E-Line service (point-to-point connection):** EVPN VPWS can be used to create a virtual point-to-point Ethernet service. This is typically achieved by setting up an EVPN instance with only two endpoints to provide a dedicated Ethernet connection between two customer sites

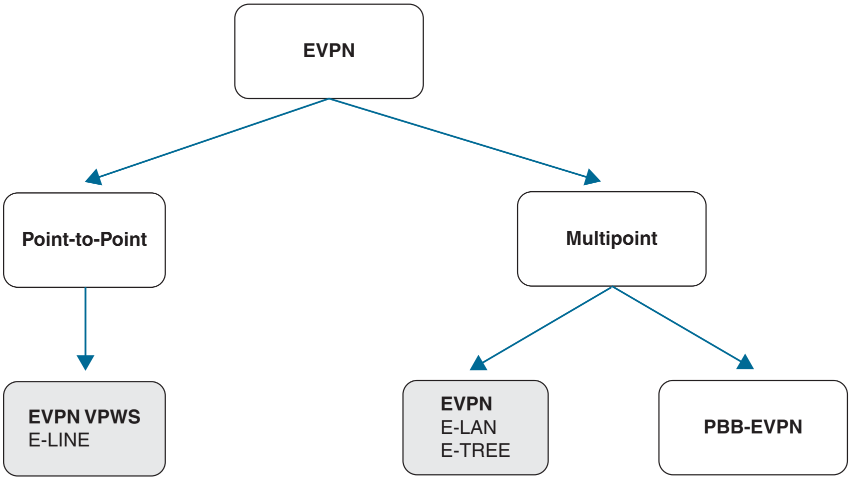


Figure 6-3 *EVPN Family: Next-Generation Ethernet Service Solutions*

EVPN in Detail

Ethernet VPN (EVPN) is a technology that provides Layer 2 VPN services over IP/MPLS transport networks and supports both unicast and multicast traffic, using BGP to distribute MAC/IP address reachability information. EVPN provides extensible and flexible multihoming VPN solutions for intra-subnet connectivity among tenant systems and end devices, which may be physical or virtual. EVPN provides E-LAN services and also allows for the provisioning of E-LINE services, with either port-active, single-active, or all-active multihoming with flow-based load balancing. EVPN VPWS simplifies pseudowire (PW) signaling and provides fast convergence upon node or link failure. The RFCs that define standards for EVPN include RFC 7432, RFC 8214, RFC 8365, and RFC 9135

Chapter 10

Business Opportunities

While the previous chapters provide detailed technical information about segment routing (SR), this chapter explores the business opportunities that emerge from adopting this innovative technology. No company will invest in a technology if its business cannot benefit in some way. This chapter discusses various topics to help build a bridge between engineering, operations, marketing, finance, and product management. It can help stakeholders and leadership identify relevant benefits in their areas, effectively communicate, and justify the investment in a transformation where silos are torn down and barriers are identified and addressed in a timely manner for a successful transition to SR.

Networks and associated IT systems are implemented differently across network service providers and large enterprises, tailored to offer the desired services to internal and external customers. Likewise, network service providers adhere to different standards and frameworks, and each organization's history and decisions influence to what extent standards and frameworks are adopted. We appreciate that organizational structures, processes, terminologies, services, and other aspects vary significantly across network service providers, and this chapter attempts to be descriptive without using any specific standard, framework, or terminology.

Echoing the insights from earlier technical chapters—such as Chapter 2, "What Is Segment Routing over MPLS (SR-MPLS)?" and Chapter 3, "What Is Segment Routing over IPv6 (SRv6)?" —this chapter highlights substantial opportunities and advantages of SRv6 compared to SR-MPLS.

Note Given that not all organizations will prioritize SRv6 transformation work items or perceive the benefits the same way, there is no universal blueprint that fits all possible scenarios. Consequently, the topics in this chapter are presented with different perspectives in mind.

The introduction of SRv6 technology presents a unique opportunity for network service providers. By harnessing its full potential, providers can gain a significant market edge or at least ensure that they remain competitive with their network service offerings over the coming decade. The improved performance, greater scale, network simplification, and new service options associated with SRv6 allow for the convergence of multiple networks into one. This consolidation ideally results in just one network to purchase, build, operate, support, power, cool, and host, thereby leading to substantial reductions in capital expenditures (CapEx) and operational expenditures (OpEx). The benefits extend to potential organizational optimizations, avoided costs for redundancy and scale-related spare capacity in multiple networks, simpler and improved service-level management (SLM), fewer integration points, and the convergence of operation support systems (OSS), business support systems (BSS), and IT systems in general. These factors can multiply the business opportunity related to the introduction of SRv6.

However, the advantages of SRv6 are not limited to consolidation and optimization. You may recall the transition from Asynchronous Transfer Mode (ATM)–based services to IP technology and its profound market impact two to three decades ago. Similarly, SR protocol options enable the transportation of new services over IP. Leased line and optical point-to-point services can now be offered over IP, reducing the need for optical network-based services to be exposed to customers. This shift simplifies the optical network stack, reducing the requirements for its OSS, BSS, and related IT systems to implement, test, support, and maintain. Instead, the IP services stack can incorporate these as additional service flavors, potentially reducing the cost of offering traditional leased line or optical services and enabling providers to offer these services at more competitive prices. Network service providers currently relying on third-party leased line or optical point-to-point services may even consider offering such services themselves, using their SR network. SRv6 allows for simplified chaining of network connectivity services with additional services such as Network Address Translation (NAT), firewalls, deep packet inspection (DPI), intrusion prevention systems (IPSs), and services offered on virtual machines or containers within data centers or clouds, reducing complexity and costs.

Network service providers operating single IP networks and requiring relatively simple IP or VPN connectivity services can also benefit from the introduction of SRv6. Although traditional MPLS VPN transport networks have been around for two decades, the networking industry is less likely to invest in significant developments or address known limitations due to its maturity, and so the transition to SRv6 is the next logical step. With device generations increasingly focusing their feature support around SRv6 and reducing legacy features to remove complexity and costs, it's worth evaluating investment in SRv6 for the coming decade.

The introduction of SRv6 represents a tremendous opportunity for network service providers. Those who thoroughly analyze and leverage its full potential can undergo a true transformation, benefiting their business and customers for years to come. A detailed analysis may even reveal the business benefits associated with SRv6 as justification for an early network lifecycle. The following sections delve into how simplification, convergence, and standardization can lead to new business and enrich existing services.

Technological Opportunities and Benefits

Before we dive into CapEx and OpEx opportunities, this section provides background information on selected opportunities. Some of these opportunities are not directly related to SR as a technology but rather to the fact that a new network may be built, or a new technology may be introduced.

Fewer Protocols

Figure 10-1 shows network technology–specific protocol stacks (though it omits protocols that are in common across all technologies).

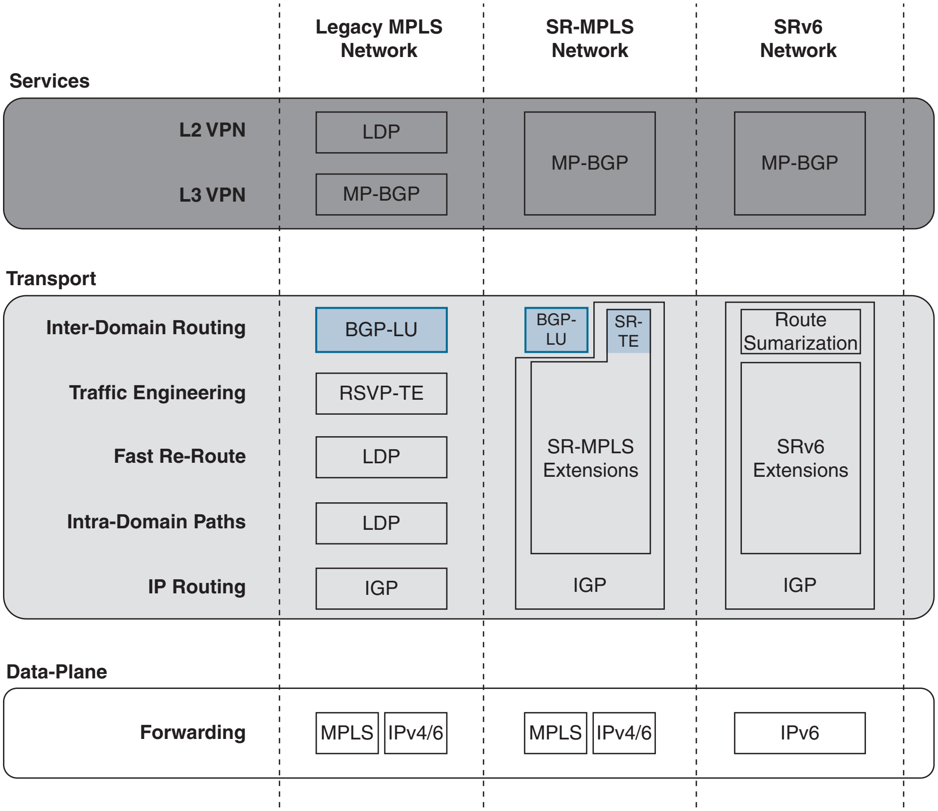


Figure 10-1 *Comparison of MPLS, SR-MPLS, and SRv6 Protocols*

Legacy MPLS networks require around six (6) specific protocols to provide VPN services. Label Distribution Protocol (LDP) is used to exchange labels for endpoints participating in L2VPN services, and Multi-Protocol BGP (MP-BGP) handles the exchange

Chapter 11

Organizational Considerations

After reviewing the promising business case for segment routing (SR) in Chapter 10, “Business Opportunities,” you are now ready to evaluate how the introduction of SR will affect your organization. The potential impacts can vary greatly, depending on the number of networks converging and migrating to a new SR-based IP transport network, as well as the complexity of traditional non-IP services that may migrate to SR using new capabilities such as private line emulation (PLE). Just as the transition from Time-Division Multiplexing (TDM)–based services to IP/MPLS networks posed challenges in the mid-2000s, SR may impact more than just network engineering or operation teams. Departments that take care of marketing, sales, customer relationship management, and product and service portfolio management, along with any business partners and resellers will need to adapt to varying degrees.

Network service providers adhere to a variety of standards and frameworks, each shaped by an organization’s unique history and set of decisions. This chapter serves as a guide, outlining important considerations and potential pitfalls for those leading the transformation to a programmable SR network and offering strategies to circumvent those pitfalls before they hinder progress. Given the considerable diversity in organizational structures, processes, terminologies, services, and other aspects across network service providers, this chapter is descriptive and deliberately avoids adherence to any specific standard, framework, or terminology, allowing for broad applicability and flexibility.

Throughout this book, the term *domain* refers to a segment of a network. This chapter expands on that concept, discussing how forming an SR domain can affect various areas, such as personnel, network infrastructure, IT frameworks, processes, service offerings, and development activities.

Although each network service provider follows its own unique path to SR, this chapter categorizes the various paths into the two scenarios, shown in Figure 11-1, to examine impacts and assist in navigating potential challenges.

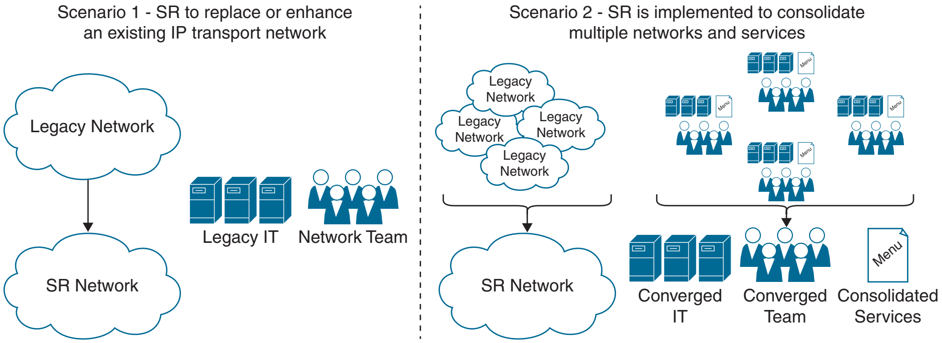


Figure 11-1 *Two Scenarios for Implementing SR*

The following areas are relevant in Scenario 1, where SR is introduced as a new technology to replace or enhance an existing IP transport network:

- **Knowledge:** The network architecture, engineering, and operation teams need to familiarize themselves with the SR technology.
- **Migration strategy:** A review of the current infrastructure for feature support, scalability, and anticipated remaining lifetime supports the choice between migrating to SR-MPLS versus SRv6.
- **IT evolution and gap awareness:** Applications involved in automating network resource and service configurations, along with monitoring and assurance systems, need adjustments to handle SR technology specifics. An assessment can provide an overview on the adaptions required to manage a new SR network. In the absence of viable options, establishing a new SR IT stack could be seen as a strategic move to modernize and phase out legacy systems.

Each of these points is discussed in greater detail later in this chapter.

When implementing SR to merge various networks and services, as in Scenario 2, the impact is potentially even greater. It's not just about combining networks; it's about fusing teams, processes, IT systems, and operational domains into a single entity. Alongside the still-relevant focus areas just listed for the first scenario, the more complex Scenario 2 calls for thoughtful evaluation of several additional considerations.

- **IT evolution and gap awareness (extension):** In Scenario 2, IT systems from various departments need to be consolidated. Separate workflow automation, fulfillment, inventory, IP address management, backup, and other systems must be converged to maximize simplicity, efficiency, and business benefits.
- **Domain definitions:** Consolidating teams and their operational domains demands a strategic approach to guarantee that the newly formed entity overseeing the SR network domain operates efficiently and effectively. This process includes evaluating and, if needed, redefining domains, roles, authorities, and responsibilities, as well as

potentially consolidating physical network locations. Thorough preparation paves the way for a smoother transition for all affected teams and domains.

- **Team organization and transformation:** The merging of teams involves not only the blending of different skill sets but also the integration of diverse backgrounds and varying approaches to work, communication, and decision making. It also introduces significant uncertainty, raising questions about team composition. Developing a clear strategy for the team's evolution and maintaining open communication are essential to preserve motivation and ensure continuity throughout the transformation.
- **Existing and new processes:** Reviewing existing processes to pinpoint those affected by the transition to the new SR network domain is crucial for defining the overall transition scope. By simultaneously capturing the efficiency and effectiveness of existing processes, it is possible to identify potential templates for any new processes required in the SR domain.
- **Network services portfolio consolidation:** Merging network service portfolios is pivotal in consolidating multiple networks. Services often vary widely across networks, with some potentially offering numerous manual configuration options. Developing a service model that consolidates all services from the affected legacy networks demands considerable effort and will help determine which variants should be phased out to establish a standardized service definition. Although unlikely, it may be possible for the service modeling process to reveal a comprehensive model that encapsulates all service options from the merging networks. Regardless of the details, harmonizing the service portfolio is essential in order to streamline automation, assurance, testing, migration, and operations of the converged SR network.
- **Development and release methodology:** Individuals forming the new SR domain team will bring a variety of experiences from their previous roles, where they might have used Agile, Waterfall, DevOps, or a blend of these and other methodologies. They are likely comfortable with a variety of practices, artifacts, lab environments, processes, and tools. The integration of network services does more than just merge these different professional experiences; it also consolidates engineering and operational responsibilities, risks, and accountability within a unified domain. To effectively navigate the complexities of the SR domain and ensure both superior quality and efficient operations, a well-defined and robust development and release methodology is crucial.
- **Change management across domains:** When converging organizations, there is a need for a comprehensive change management strategy that addresses all levels of the affected domains in the organization. This strategy should include communication plans to keep all stakeholders informed, a common overall roadmap to align domains, training programs to upskill employees where necessary, and feedback mechanisms to address any concerns and challenges that may arise. Such a central change management strategy is critical to help domains and their employees transition to new ways of working, to foster acceptance of the new organizational structure, and to ensure that the combined entity can achieve its desired synergies and performance objectives.

The subsequent sections of this chapter delve into all these aspects in detail.

Scenario 1: Replacing or Enhancing a Legacy Network with SR

Every journey to SR needs to consider at least the three areas shown in Figure 11-2.

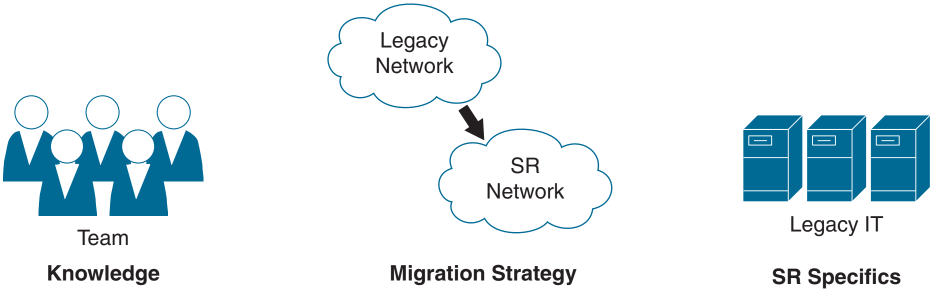


Figure 11-2 *Scenario 1: Replacing or Enhancing a Legacy Network with SR*

The following subsections explore these areas in greater detail and offer ideas to simplify the transition to SR.

Knowledge

What type of SR knowledge is essential for various roles within the organization? Who is tasked with making investment decisions? Who will fill which roles in the upcoming months, and who must possess substantial SR knowledge? This section aims to address these questions by identifying the key actors responsible for introducing SR within a service provider's network and discussing potential sources of knowledge. The actors and the sequence in which they require SR knowledge have been greatly simplified to serve as an introductory guide. To prevent redundancy, Scenario 2, which involves a more comprehensive transition to SR, is visually distinguished in Figure 11-3 by gray highlighting on the Scenario 2–relevant stakeholders and key tasks.

Let's now look at the reasoning for the required knowledge and the type of expertise needed by the actors at each chronological phase:

- **Network architects:** Faced with lifecycle challenges such as software or hardware nearing end-of-life or scaling issues, these professionals are tasked with finding suitable alternatives or successor technologies. They evaluate SR capabilities, identify network elements that lack SR support (such as load balancers and NAT), assess the impact on existing infrastructure, explore new service opportunities or enhancements using SR, and develop strategies for seamless SR migration to maintain service continuity. All these elements are then integrated into a target architecture, which serves as a baseline to evaluate the transition's impact on interfacing networks and existing IT systems.

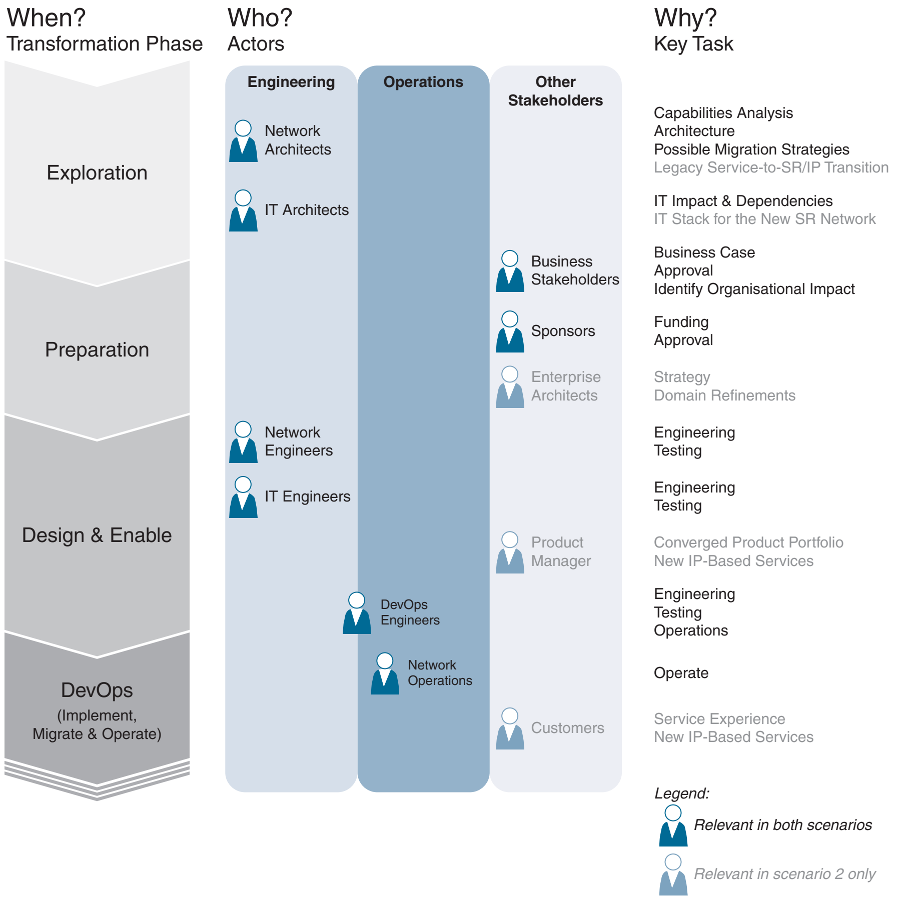


Figure 11-3 *SR Knowledge: Who Needs It, When, and Why*

- **IT architects:** IT specialists are engaged to assess the impact of introducing SR on the overall IT infrastructure. In Scenario 2, these experts must reach consensus on selecting the most suitable IT applications from the existing suite to oversee the new SR network. If current options prove inadequate, the IT architects may need to agree on the implementation of new IT systems, aiming to both modernize the framework and systematically retire outdated platforms.

- **Business stakeholders:** Business stakeholders must, at a high level, assess the various options—along with their benefits and impacts—identified by the architects. Collaboratively, they will craft a business case, refine the preferred solution, and define the commitments required from all affected parties. It is crucial that business stakeholders and architects reach and document consensus on the selected solution before proceeding with the transformation; their documentation serves as a reference