



# SRv6 Fundamentals and Key Concepts

In today's dynamic networking environment, the demand for scalable, efficient, and flexible routing solutions continues to grow. Traditional approaches, such as MPLS, often involve complex protocols and lack native IPv6 support, making them less suitable for modern networks. Segment Routing over IPv6 (SRv6) addresses these challenges by leveraging the native IPv6 protocol to implement a simplified, programmable, and scalable routing framework.

- [Segment Routing over IPv6, on page 1](#)
- [Key concepts of SRv6, on page 3](#)
- [Supported and unsupported features for SRv6, on page 12](#)
- [Techniques for effective SID management, on page 13](#)

## Segment Routing over IPv6

Segment Routing over IPv6 (SRv6) is a network routing framework that

- embeds routing and processing instructions directly into IPv6 packets
- uses the Segment Routing Header (SRH) to encode these instructions
- uses Segment Identifiers (SIDs), which are 128-bit values, to define routing and processing behaviors, and
- enables a network programming model to allow packets to carry instructions for advanced traffic engineering and automated service chaining.

The components of SRv6 include Segment Routing Header and Segment Identifiers. See [Key concepts of SRv6, on page 3](#).

### Key difference between SR-MPLS and SRv6

The table compares the key benefits and features of SR-MPLS and SRv6 to highlight their differences and advantages:

Table 1: SR-MPLS and SRv6 comparison

| Feature                     | SR-MPLS (Segment Routing with MPLS)                           | SRv6 (Segment Routing with IPv6)  |
|-----------------------------|---|---|
| Underlying technology       | Uses MPLS labels to encode segment routing paths.             | Uses IPv6 addresses (Segment Identifiers or SIDs) in IPv6 header.                     |
| Packet Header               | MPLS label stack steers packets through the network.          | IPv6 extension headers carry the segment list for source routing.                     |
| Scalability and flexibility | Limited by MPLS label space and MPLS forwarding capabilities. | Larger address space and more flexible network programming due to IPv6 extensibility. |
| Network requirements        | Requires MPLS-enabled infrastructure.                         | Requires IPv6-enabled infrastructure and SRv6 processing support.                     |

### Benefits of SRv6

Modern networks require solutions that support growing scale, advanced services, and seamless operations. Traditional routing technologies often face limitations such as complex architectures, limited scalability, and rigid traffic management. Segment Routing over IPv6 (SRv6) addresses these challenges. It provides a highly programmable, scalable, and efficient solution by leveraging native IPv6 capabilities to simplify operations and enhance service delivery.

SRv6 delivers these key advantages:

- **Simplified network architecture:** SRv6 consolidates routing functions into a single IPv6-based data plane. This eliminates multiple legacy protocols and streamlines inter-domain routing with basic IP functions and prefix summarization.
- **Native IPv6 integration and enhanced scalability:** SRv6 fully leverages IPv6's capabilities and address space, ensuring long-term scalability and future readiness. Its flexible, hierarchical Segment Identifier (SID) structure efficiently scales networks, overcoming traditional label limitations and maximizing Forwarding Information Base (FIB) utilization across diverse segments.
- **Flexible and intent-based traffic engineering:** SRv6 offers precise and flexible path selection through its built-in traffic engineering capabilities. This enables network programming with SIDs, supports intent-based routing via IGP Flex-Algo, and facilitates complex service chaining and Network Function Virtualization (NFV) integration.
- **Optimal load balancing:** SRv6 efficiently distributes traffic across available paths. It leverages native IPv6 flow label capabilities for Equal-Cost Multi-Path (ECMP) load balancing, ensuring optimal traffic distribution based on inner packet entropy without additional protocols.
- **Unified service delivery and cloud integration:** SRv6 provides a single, unified IPv6-based data plane for various services, including Virtual Private Networks (VPNs), micro-services, and cloud connectivity. It offers native support for host and cloud environments without protocol translation.
- **Seamless deployment and migration:** SRv6 integrates smoothly into existing brownfield networks. It supports granular, phased adoption strategies, facilitating a seamless transition to modern network infrastructures.

- Improved reliability and performance: SRv6 enhances network resilience with features like Fast Reroute (FRR). It also enables better Quality of Service (QoS) management through granular control over traffic paths and functions, ensuring robust and high-performing networks.

### Use cases for SRv6

SRv6 is a technology that enhances various network applications through several use cases, such as:

- Traffic engineering (TE): SRv6 optimizes network paths to enhance performance, meet SLAs, routing requirements, as well as specific needs such as security or data sovereignty.
- Network segmentation: SRv6 enables the creation of logical network overlays, including L3VPNs and L2VPNs, over a common SRv6 underlay.
- Simplified OPEX: By providing a common end-to-end data plane, SRv6 simplifies operations, making it easier than managing multiple technologies.

## Key concepts of SRv6

The key concepts of SRv6 are described in this table.

**Table 2: SRv6 Concepts**

| SRv6 Concept   | Description  |
|--|--|
| <a href="#">Network program in the packet header</a> | The network programming model refers to the approach of encoding network operations and services directly into IPv6 headers using the concept of segments. |
| <b>Segment Routing Header (SRH):</b>                 | Encodes an ordered list of instructions (segments) within each packet.   |
| <a href="#">Segment Identifiers (SIDs)</a>           | 28-bit values that represent routing instructions, functions, or paths.  |
| <a href="#">SRv6 node roles, on page 6</a>           | Nodes in an SRv6 network perform specific functions as source, transit, or endpoint nodes based on their position and SID processing requirements.         |

## Network program in the packet header

The network programming model in SRv6 refers to the approach of encoding network operations and services directly into IPv6 headers using the concept of segments. SRv6 leverages the native IPv6 protocol to implement segment routing functionalities, allowing for flexible and scalable traffic engineering, and network programmability.

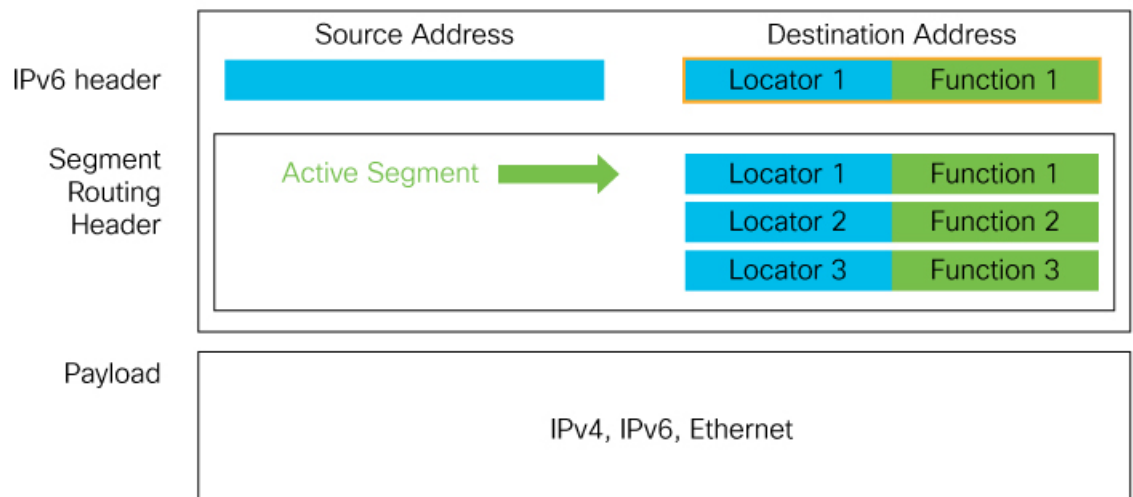
In SRv6, an IPv6 address represents an instruction. SRv6 uses a new type of IPv6 Routing Extension Header, called the Segment Routing Header (SRH), in order to encode an ordered list of instructions. The active segment is indicated by the destination address of the packet, and the next segment is indicated by a pointer in the SRH.

An packet header includes these components:

- IPv6 header: This is the standard IPv6 header, which includes the source address of the packet and destination address.
- Destination address: In a standard IPv6 header, the destination address contains a locator and a function.
- Segment Routing Header (SRH): SRH contains an ordered list of segments that the packet must traverse and header is inserted between the IPv6 header and the payload. SRH includes these components:
  - Active segment: Indicates the current segment being processed.
  - Locator and Function: Each segment consists of a locator and a function. The locator identifies a network node or a set of nodes, while the function specifies the action to be performed at that node.
- Payload: This section contains the actual data being transmitted. It can encapsulate various protocols, such as IPv4, IPv6, or Ethernet.

This figure illustrates the structure of a network program within the SRv6 packet header. In the example, there are multiple segments, each with its locator (Locator 1, Locator 2, Locator 3) and function (Function 1, Function 2, Function 3). The number of segments is determined by the Segments Left field in the SRH.

**Figure 1: Packet Header**



## Segment Routing Header

Segment Routing Header (SRH) is an IPv6 routing extension header that

- embeds an ordered list of instructions, called segments, within each packet
- facilitates efficient routing, and network programmability, and
- encodes explicit paths using a list of IPv6 addresses.

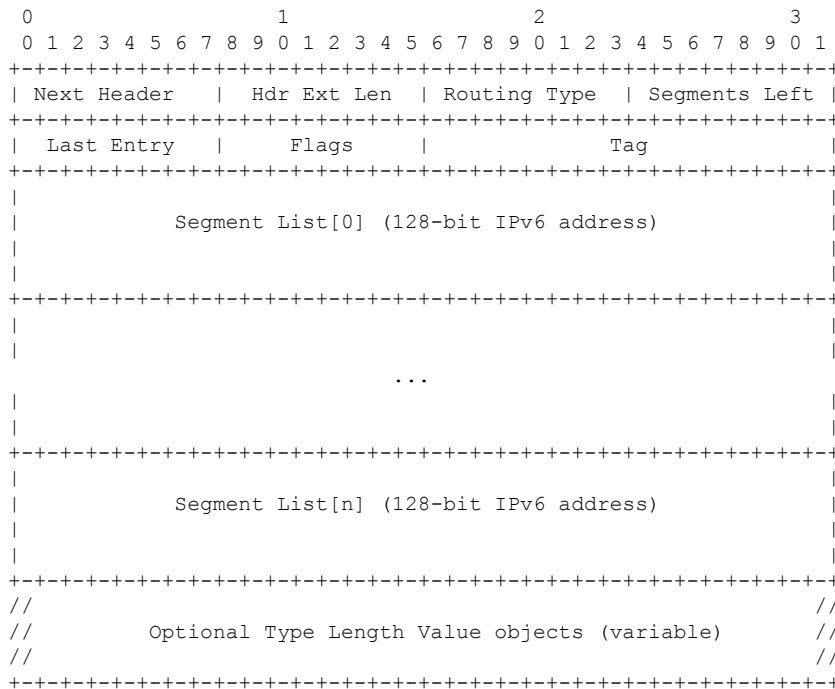
Segments are instructions represented as IPv6 addresses, that specify the sequence of nodes or functions a packet should traverse through a network using segment routing. For more information about SRH, see IETF RFC [IPv6 Segment Routing Header \(SRH\)](#).

**Fields in the SRH**

The SRH enables source routing in IPv6 by inserting a sequence of segments directly into packets. The active segment specifies the current destination address, while a pointer in the SRH identifies the next segment to be processed.

| Field                | Description  |
|----------------------|--|
| <b>Next Header</b>   | Identifies the type of header immediately following the SRH.   |
| <b>Hdr Ext Len</b>   | The length of the SRH in 8-octet units, not including the first 8 octets.  |
| <b>Segments Left</b> | Specifies the number of route segments remaining, i.e., the number of explicitly listed intermediate nodes still to be visited before reaching the final destination.  |
| <b>Last Entry</b>    | Contains the index (zero-based) of the last element of the segment list.   |
| <b>Flags</b>         | Contains 8 bits of flags.  |
| <b>Tag</b>           | Tags a packet as part of a class or group of packets, such as packets sharing the same set of properties.  |
| <b>Segment List</b>  | A list of 128-bit IPv6 addresses representing the nth segment in the segment list. The encoding starts from the last segment of the SR policy: Segment List [0] contains the last segment, Segment List [1] contains the penultimate segment, and so on. |

**Format of SRH**



## Segment Identifiers

A Segment Identifier (SID) is a 128-bit value that identifies segments in an SRv6 network and consists of these three parts:

- **Locator (LOC):** This is the first part of the SID with most significant bits and represents an address of a specific SRv6 node.
  - **SID Block (B):** The network designator, representing the SRv6 domain or a known address space.
  - **Node ID (N):** The specific identifier for the node within the SRv6 network.
- **Function (FUNCT):** The portion of the SID that specifies a local behavior, or network instruction, executed by the node identified by the Locator. The Function field is opaque and bound to the SID's behavior.
- **Arguments (ARG):** An optional field containing additional information needed for SID processing. ARG values should remain constant within a flow to maintain consistent Equal-Cost Multi-Path (ECMP) hashing and avoid packet reordering.

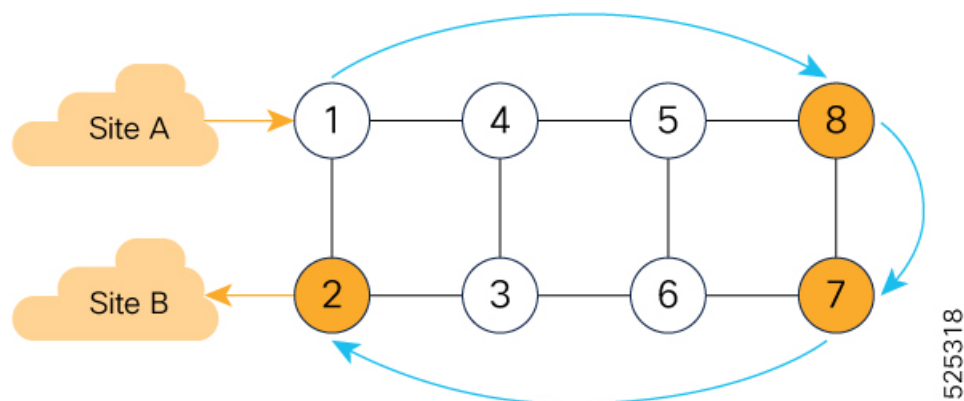
## SRv6 node roles

In an SRv6 network, nodes along the packet path perform specific roles based on their position in the network and the SID processing requirements. Each node along the SRv6 packet path has a different function.

The figure visually represents the different node roles in an SRv6 network.

- Node 1 is the headend or the source node.
- Node 4 and 5 are the transit node.
- Node 8 is the endpoint node.

**Figure 2: SRv6 Nodes**



This table provides an overview of the roles and descriptions of various node types, highlighting their specific functions within an SRv6 network.

Table 3: SRv6 node roles

| SRv6 node                          | Description  |
|------------------------------------|--|
| <b>Source node or headend node</b> | A node that can generate an IPv6 packet with an SRH, or an ingress node that can impose an SRH on an IPv6 packet. Also known as headend node.<br><br>For more information, see <a href="#">SRv6 Head-End Behaviors</a> .   |
| <b>Transit node</b>                | A node along the path of the SRv6 packet (IPv6 packet and SRH).<br><br>The transit node does not inspect the SRH. The destination address of the IPv6 packet does not correspond to the transit node.  |
| <b>Endpoint node</b>               | A node in the SRv6 domain where the SRv6 segment is terminated. The destination address of the IPv6 packet with an SRH corresponds to the end point node. The segment endpoint node executes the function bound to the SID. A node where the SRv6 segment is terminated.<br><br>For more information, see <a href="#">SRv6 endpoint behaviors, on page 9</a> . |

## SRv6 headend behaviors

A segment routing headend behavior is a set of actions and mechanisms performed by a head-end router (or source node) that:

- implements segment routing policies to manage and direct traffic through a network
- initiates and enforces SR policies by defining the path traffic should take, and
- ensures efficient traffic forwarding by leveraging SRv6.

Table 4: Feature History Table

| Feature Name                 | Release Information | Feature Description  |
|------------------------------|---------------------|--|
| H.Encap.Red headend behavior | Release 25.1.1      | <p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])</p> <p>The H.Encap.Red is a headend behavior that <b>encapsulates</b> the original packet into a new IPv6 packet with a Segment Routing Header (SRH). By encapsulating with the SRH, you can control and manage the path the data packets take through a network.</p> <p>This feature is now supported on:</p> <ul style="list-style-type: none"> <li>• 8712-MOD-M</li> <li>• 8011-4G24Y4H-I</li> </ul> |

| Feature Name  | Release Information | Feature Description   |
|---|---------------------|---|
| H.Insert.Red Headend Behavior for SRv6 on Cisco Silicon One P100-based Line Cards | Release 24.3.1      | <p>Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*)</p> <p>With H.Insert.Red head-end behavior, you can effectively steer traffic into an SR policy, allowing for fast rerouting, traffic optimization, and simplified path management without additional encapsulation.</p> <p>The H.Insert.Red head-end behavior enables the router to insert a Segment Routing Header (SRH) directly into an existing IPv6 packet.</p> <p>The feature is supported only on Cisco Silicon One P100-based line cards in Cisco 8800 modular routers operating in P100 compatibility mode.</p> <p>* This feature is supported on:</p> <ul style="list-style-type: none"> <li>• 88-LC1-36EH</li> <li>• 88-LC1-12TH24FH-E</li> <li>• 88-LC1-52Y8H-EM</li> </ul> |

### SRv6 headend behavior variants

SRv6 supports these head-end behaviors:

- H.Encaps.Red—H.Encaps with reduced encapsulation
- H.Insert.Red—H.Insert with reduced insertion. Starting from Cisco IOS XR Release 24.3.1, the H.Insert.Red is supported.

### Comparison between H.Encaps.Red and H.Insert.Red headend behaviors

*Table 5: H.Encaps.Red versus H.Insert.Red headend behaviors*

| Headend behaviors   | H.Encaps.Red  | H.Insert.Red  |
|---------------------|---|---|
| Definition          | <p>The H.Encaps.Red is a headend behavior that <b>encapsulates</b> the original packet into a new IPv6 packet with an SRH.</p> <p><b>Note</b><br/>We support Starting from Cisco IOS XR Release 24.3.1, the H.Insert.Red.</p> | <p>The H.insert.Red is a headend behavior that <b>inserts</b> an SRH into the original IPv6 packet without encapsulating it into a new IPv6 packet.</p> |
| Header manipulation | <p>A new IPv6 header with an SRH is added to the packet, encapsulating the original packet.</p>   | <p>The SRH is inserted into the packet by modifying the existing IPv6 header.</p>   |

| Headend behaviors                | H.Encaps.Red   | H.Insert.Red  |
|----------------------------------|--|---|
| Packet size                      | The packet size increases as it includes both the SRH and an additional IPv6 header.   | The packet size is smaller compared to H.encaps headend behavior. There is no extra IPv6 header and that helps maintain the packet size.  |
| Processing at intermediate nodes | Intermediate nodes process the packet by examining the outer IPv6 header's destination address and the SRH.  | Intermediate nodes process the packet by examining the inserted SRH and forwarding the packet based on the active segment.  |
| Termination process              | Ultimate Segment Pop<br><br>The termination process involves decapsulation, where the outer IPv6 header and SRH are removed to reveal the original packet. | Penultimate Segment Pop (PSP)<br><br>The termination process involves the removal of the SRH when the packet reaches the end of the SR Policy.  |
| References                       | The SR Headend with Encapsulation behaviors are documented in the <a href="#">IETF RFC 8986 SRv6 Network Programming</a> .                                 | The SR Headend with Insertion head-end behaviors are documented in the following IETF draft:<br><a href="https://datatracker.ietf.org/doc/draft-filsfils-spring-srv6-net-pgm-insertion/">https://datatracker.ietf.org/doc/draft-filsfils-spring-srv6-net-pgm-insertion/</a> |

## SRv6 endpoint behaviors

The SRv6 endpoint behaviors are a set of actions and mechanisms that:

- processes an SRv6 segment matching the network device's own address
- terminates the SRv6 segment at a node within the SRv6 domain, and
- executes the function bound to the SID.

The SRv6 endpoint behaviors are documented in the [IETF RFC 8986 SRv6 Network Programming](#).

### List of SRv6 endpoint behaviors

These are a subset of defined SRv6 endpoint behaviors that can be associated with a SID.

- End—Endpoint function. The SRv6 instantiation of a Prefix SID [[RFC8402](#)].
- End.X—Endpoint with Layer-3 cross-connect. The SRv6 instantiation of an Adj SID [[RFC8402](#)].
- End.DX6—Endpoint with decapsulation and IPv6 cross-connect (IPv6-L3VPN - equivalent to per-CE VPN label).
- End.DX4—Endpoint with decapsulation and IPv4 cross-connect (IPv4-L3VPN - equivalent to per-CE VPN label).
- End.DT6—Endpoint with decapsulation and IPv6 table lookup (IPv6-L3VPN - equivalent to per-VRF VPN label).

- End.DT4—Endpoint with decapsulation and IPv4 table lookup (IPv4-L3VPN - equivalent to per-VRF VPN label).
- End.DT46—Endpoint with decapsulation and specific IP table lookup (IP-L3VPN - equivalent to per-VRF VPN label). Starting from IOS XR Release 7.5.3, the End.DT46 endpoint is supported.
- End.DX2—Endpoint with decapsulation and L2 cross-connect (L2VPN use-case).
- End.B6.Encaps—Endpoint bound to an SRv6 policy with encapsulation. SRv6 instantiation of a Binding SID.
- End.B6.Encaps.RED—End.B6.Encaps with reduced SRH. SRv6 instantiation of a Binding SID.

## SRv6 endpoint behavior variants

Segment Routing over IPv6 (SRv6) defines several endpoint behavior variants for the `End` and `End.X` behaviors, depending on how the Segment Routing Header (SRH) is processed. These variants can be supported individually or in combinations.

**Table 6: Comparison of endpoint behavior variants**

|                       | <b>Penultimate Segment Pop (PSP)</b>   | <b>Ultimate Segment Pop (USP)</b>   | <b>Ultimate Segment Decapsulation (USD)</b>  |
|-----------------------|--|---|--|
| <b>Definition</b>     | Penultimate Segment Pop is a mechanism where the penultimate SR endpoint node processes the SRH by modifying the destination address and segments list.                  | Ultimate Segment Pop (USP) is a mechanism where the ultimate (last) SR Segment Endpoint Node removes the SRH from the IPv6 header when the <code>Segments Left</code> field equals 0.   | Ultimate Segment Decapsulation (USD) is a mechanism that processes upper-layer headers by removing the outer IPv6 header and forwarding the inner packet to its destination. |
| <b>Key operations</b> | as part of the SID processing, copies the last SID from the SRH into the IPv6 Destination Address and decrements the <code>segmentsl</code> list value from one to zero. | If <code>Segments Left</code> is 0, then: <ul style="list-style-type: none"> <li>• Updates the <code>Next Header</code> field in the preceding header to the next header value of the SRH.</li> <li>• Decreases the IPv6 header Payload Length by <math>8 * (\text{Hdr Ext Len} + 1)</math></li> <li>• Removes the SRH from the IPv6 extension header chain</li> <li>• Proceeds to process the next header in the packet</li> </ul> | See <a href="#">SRv6 endpoint behavior variants, on page 10</a> and <a href="#">SRv6 endpoint behavior variants, on page 10</a>  |

|                        | <b>Penultimate Segment Pop (PSP)</b>  | <b>Ultimate Segment Pop (USP)</b>  | <b>Ultimate Segment Decapsulation (USD)</b>  |
|------------------------|---|--|--|
| <b>Where performed</b> | The PSP operation takes place only at a penultimate SR Segment Endpoint Node and does not happen at non-penultimate endpoint nodes. When a SID of PSP-flavor is processed at a non-penultimate SR Segment Endpoint Node, the PSP behavior is not performed since Segments Left would not be zero. | USP is performed only at the ultimate SR Segment Endpoint Node. triggered when the Segments Left field equals 0.   | USD is performed at the last SR Segment Endpoint Node in the repair path.  |
| <b>Applications</b>    | PSP is used when the Source SR Node instructs the penultimate SR Segment Endpoint Node to remove the SRH from the IPv6 header. The Source SR Node achieves this by using a PSP-flavored SID.  | One of the applications of the USP flavor is when a packet with an SRH is destined to an application on hosts with smartNICs implementing SRv6. The USP flavor is used to remove the consumed SRH from the extension header chain before sending the packet to the host. | One of the applications of the USD flavor is the case of TI-LFA in P routers with encapsulation with H.Encaps. The USD flavor allows the last Segment Endpoint Node in the repair path list to decapsulate the IPv6 header added at the TI-LFA Point of Local Repair and forward the inner packet. |

Table 7: Key operation for the end behavior of USD

| <b>Upper-layer header type is ...</b> | <b>then ...</b>  |
|---------------------------------------|--|
| <b>41 (IPv6)</b>                      | <ol style="list-style-type: none"> <li>1. Removes the outer IPv6 Header with all its extension headers.</li> <li>2. Submits the packet to the egress IPv6 FIB lookup and transmission to the new destination.</li> </ol> |
| <b>4 (IPv4)</b>                       | <ol style="list-style-type: none"> <li>1. Removes the outer IPv6 Header with all its extension headers.</li> <li>2. Submits the packet to the egress IPv4 FIB lookup and transmission to the new destination.</li> </ol> |
| <b>else</b>                           | process as per Section 4.1.1 (Upper-Layer Header) of <a href="#">IETF RFC 8986 SRv6 Network Programming</a>  |

Table 8: Key operation for the End.X behavior of USD

| Upper-layer header type is ... | then ...   |
|--------------------------------|--|
| 41 (IPv6) or 4 (IPv4)          | <ol style="list-style-type: none"> <li>1. Removes the outer IPv6 header with all its extension headers.</li> <li>2. Forwards the exposed IP packet to the L3 adjacency J.</li> </ol> |
| else                           | process as per Section 4.1.1 (Upper-Layer Header) of <a href="#">IETF RFC 8986 SRv6 Network Programming</a>  |

## Supported and unsupported features for SRv6

### Supported hardware PIDs for SRv6

- SRv6 is supported on these Cisco 8000 series Q200-based line cards and fixed-port routers:
  - Cisco 8800 with 88-LC0-36FH-M, 88-LC0-36FH, 88-LC0-34H14FH line cards
  - Cisco 8201-32FH
  - Cisco 8102-64H
  - Cisco 8101-32-FH

### Supported SRv6 features

- IGP redistribution/leaking between levels
- Prefix Summarization on ABR routers
- IS-IS TI-LFA
- Microloop Avoidance
- Flex-algo
- OAM: Ping and traceroute are supported.

### Unsupported features for SRv6

- SRv6 is not supported on Q100-based line cards and fixed-port routers.
- SRv6 is supported on Cisco 8000 series routers used as core routers (P-role).
- SRv6 is not supported on Cisco 8000 series routers used as edge routers (PE-role). Services over SRv6 are not supported.
- SRv6 over GRE interface is not supported
- SRv6 over BVI interface is not supported
- Egress marking on the outer header during SRv6 encapsulation operations (TI-LFA) is not supported.

# Techniques for effective SID management

SIDs are a fundamental building block of SRv6, enabling flexible and programmable routing by encoding instructions directly into IPv6 headers. As networks grow in complexity, efficient management of SIDs becomes critical to ensure scalability, reduce overhead, and optimize network operations.

SRv6 employs specific techniques to effectively use the 128-bit SID for optimized network operations. The main techniques are:

- **Micro-SID (uSID):** This technique divides a single 128-bit SID into smaller sub-SIDs, referred to as micro-segments. It allows multiple operations to be represented within a single SID, which results in reduced header overhead and improved scalability, particularly useful in environments with limited resources. For more information, see [SRv6 Micro-segment \(uSID\) Overview](#).
- **Full-Length SID:** This technique utilizes a full 128-bit IPv6 address to represent each segment, function, or instruction. This method ensures simplicity and aligns with standard IPv6 routing practices.

## Difference between full-length SID and uSID

*Table 9: Comparison of full-length SID and uSID*

| Feature            | Full-Length SID   | uSID  |
|--------------------|---|---|
| <b>Definition</b>  | A full 128-bit IPv6 address used to represent a single segment, function, or instruction.             | A single 128-bit SID divided into smaller sub-SIDs (micro-segments) to represent multiple operations. |
| <b>Size</b>        | Each SID uses 128-bits in the SRH.  | Multiple smaller sub-SIDs are packed into a single 128-bit SID.                                       |
| <b>Header size</b> | Larger SRH due to 128-bits per SID.   | Smaller SRH by encoding multiple operations into fewer SIDs.  |
| <b>Examples</b>    | Each SID is a full 128-bit IPv6 address.<br>SRH = [2001:db8:1::1,<br>2001:db8:2::1,<br>2001:db8:3::1] | Multiple sub-SIDs are encoded in one 128-bit SID.<br>SRH = [2001:db8:1::1]                            |
| <b>Reference</b>   | <a href="#">SRv6 Micro-segment (uSID) Overview</a>  | <a href="#">Configure Segment Routing over IPv6 (SRv6) with Full-Length SIDs</a>                      |

