SRv6 Operations
Internet-Draft
Intended status: Informational
Expires: 22 January 2026

C. Filsfils
Cisco Systems
C. Martin
Oracle Cloud
K. Pillai
IBM
P. Camarillo, Ed.
A. Abdelsalam
Cisco Systems
21 July 2025

SRv6 for Deterministic Path Placement in AI Backends
draft-filsfils-srv6ops-srv6-ai-backend-01

Abstract

This document describes the use of SRv6 to enable deterministic path placement in AI backends, optimizing load balancing and congestion control for predictable GPU workloads.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 January 2026.

Table of Contents

1. Introduction

Hyperscale AI training clusters rely on massive GPU-to-GPU data exchanges, where synchronization delays caused due to congestion delays and packet loss directly impact model convergence time and operational costs.

These workloads generate *large, predictable flows* that require ultra-low latency, high bandwidth, and precise congestion control to maintain efficiency. Traditional networking approaches, such as ECMP-based per-flow load balancing, suffer from poor entropy due to the limited number of RoCEv2 flows, leading to fabric hotspots, congestion, and slow reconvergence after failures.

SRv6 uSID (NEXT-CSID) provides the ability to steer in the fabric, allowing the NIC (i.e., SmartNIC, DPU) to perform *deterministic path placement of ROCEv2 traffic through the fabric. This ensures predictable performance, fine-grained traffic control, and real-time adaptation to congestion in a stateless manner.*

Future revisions of this draft will cover additional use-cases (multi-path transport, virtual rail topologies, stateless interaction between AI/LLM leasing a cluster infra and the operator managing the cluster, etc).

The document draft-filsfils-srv6ops-srv6-end-to-end-dc-frontend-wan-00 explains how SRv6 uSID (NEXT-CSID) is applied to an end-to-end DC Frontend and WAN fabric.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

SRv6 Segment Routing over IPv6 [RFC8986].

uSID Micro-segment. Formally defined as NEXT-CSID in [RFC9800].

> The term _uSID (micro SID)_ predates the formal naming and has been widely adopted across the industry - including operators with large-scale deployments, vendors, open-source implementations, and used consistently in multi-vendor interoperability reports.
>
> To maintain alignment with the formal specification while also acknowledging the widespread and practical use of the term, this document uses uSID and NEXT-CSID interchangeably.

ECMP Equal-Cost Multi-Path

uN The uN is a short notation for the End behavior with NEXT-CSID, PSP, and USD flavors as defined in [RFC9800].

uA The uA local behavior is a short notation for the End.X behavior with NEXT-CSID, PSP, and USD flavors [RFC9800].

ROCEv2 RDMA over Converged Ethernet version 2 [IBTA-ROCEv2].

NIC Network Interface Card, a hardware component that connects a computer to a network.

SmartNIC A Network Interface Card with embedded processing capabilities, designed to offload network and storage tasks from the host CPU.

DPU Data Processing Unit, a specialized processor designed to offload and accelerate data-centric tasks, often used in network and storage functions.

GPU Graphics Processing Unit, a processor designed for rendering graphics and performing parallel computation tasks, commonly used for AI and machine learning workloads.

3. AI Traffic Characteristics and Challenges

AI workloads exhibit highly structured traffic patterns:

* *Predictable Elephant Flows*: Collectives' communications require multiple GPUs to exchange data in a structured manner that is known in advance. Flows between GPUs are large, long-lived, high throughput and predictable.

* *Synchronized Bursts*: Model synchronization causes periodic, coordinated traffic spikes.

* *Low ECMP Entropy*: Data exchange between GPUs relies on a small number of flows (ROCEv2 Queue Pairs), leading to poor performance of traditional load-balancing solutions. A 5-tuple based ECMP load-balancing results in non-homogenous utilization across the fabric, leading to congestion.

* *Resilience*: Network failures prolong training time and increase significantly operational costs. Therefore, it is imperative for the network to provide high resiliency, and fast reaction to congestion.

4. SRv6 for Deterministic Path Placement

SRv6 enables the NIC to directly control the AI workload traffic journey through the fabric by encoding an ordered list of segments in the packet header.

* *AI Scheduler*: Upon AI job orchestration, the collectives' communications are defined (i.e., the GPU Topology). The AI scheduler determines the optimal fabric routed paths based on all the running jobs in the fabric, and the GPU topology for each one of them.

  - The encoding of a path as an SRv6 Network Program *does not require any per-path communication between the AI Scheduler and the fabric*.

  - At fabric bring up, the controller managing the fabric communicates the overall topology together with SRv6 uSID (NEXT-CSID) explicit instructions for each link (uA). These instructions are statically configured are thus independent of any routing protocol dynamic state. The AI Scheduler build any path through the fabric without any further control-plane interaction with the routers.

* *NIC*: The NIC, before sending the ROCEv2 traffic, encapsulates with an outer IPv6 header and encodes in the packet header the sequence of instructions to enforce the precomputed path through the fabric.

  - Note that an outer IPv6 header allows to encode 6 uSIDs in the Destination Address. This implies that even upon presence of a super-spine in a 3-tier Clos fabric, the entire path can be encoded without the need of any additional Segment Routing extension Header (SRH).

* *Highly Scalable Stateless Fabric*: The routers in the fabric enforce the path by following the sequence of SRv6 instructions in the packet header. There is *no per-flow state in the network* (unlike MPLS RSVP-TE which would require the instantiation of states in the fabric on a per GPU-to-GPU deterministic path basis).

* *Congestion Feedback Loop*: The NICs react in real time to congestion notifications (ECN, inband latency measurement, Packet Trimming, inband packet loss). These mechanisms are preserved and leveraged by the solution to optimize traffic steering and prevent congestion hotspots. At any time (without any fabric signaling or dependency), within a few nanoseconds, the NIC can change the deterministic path through the fabric by simply changing the outer IPv6 Destination Address. The change is only at the source NIC. There is no change required at any of the intermediate devices in the fabric.

5. Illustration

The following figure depicts a typical 2-tier Clos topology.

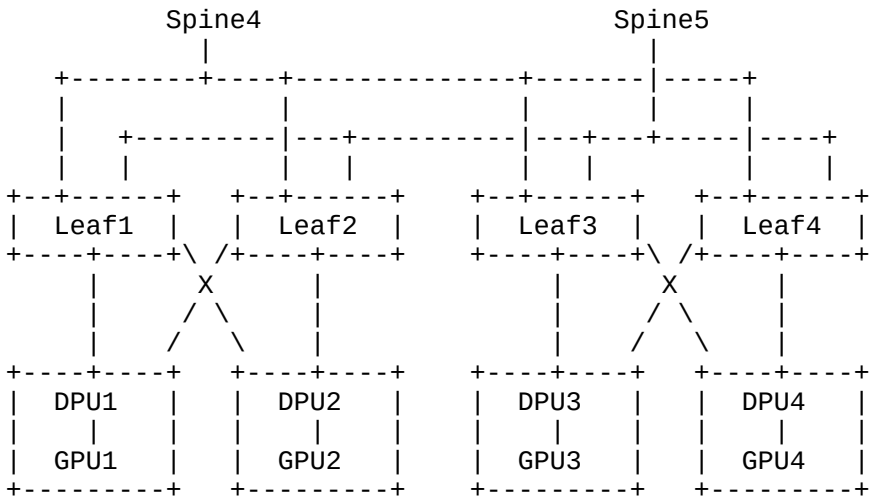

Figure 1: Reference Topology

The topology consists of two Spine devices. Each of the Spines is connected to four Leaf devices.

There are 4 NICs, which are connected through the host interface (e.g., PCIe) to a GPU. In this example each NIC is dual-homed to two Leaf devices.

5.1. SRv6 Fabric Provisioning

At a day0 cluster build-up (fabric bring-up), the topology is provisioned with SRv6 SIDs on the Spine and Leafs devices. These SIDs are statically configured and thus independent of any routing protocol dynamic state. The following is provisioned:

* SRv6 SID Space in the fabric 5f00:0::/32

* Leaf*1* instantiates the SID 5f00:0:0*1*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

* Leaf*2* instantiates the SID 5f00:0:0*2*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

* Leaf*3* instantiates the SID 5f00:0:0*3*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

* Leaf*4* instantiates the SID 5f00:0:0*4*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

* Spine*5* instantiates the SID 5f00:0:0*5*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

* Spine*6* instantiates the SID 5f00:0:0*6*00::/48 associated with the uN instruction (End with NEXT-CSID, PSP & USD)

5.2. SRv6-Based Deterministic Path Selection

In the fabric there is an AI job being orchestrated. As a result of the AI orchestration and the collectives' communication, it results that the GPU1 and GPU2 must send traffic periodically to GPU3.

The AI orchestration, based on the network topology, computes the paths which achieve homogenous utilization in the fabric to avoid congestion:

* GPU1->GPU3: via Leaf1, Spine5, Leaf3

* GPU2->GPU3: via Leaf2, Spine6, Leaf4

Upon AI job computation (at GPU synchronization time):

* NIC1: creates a ROCEv2 packet that must be sent to NIC3. NIC1 encapsulates the ROCEv2 packet with an outer IPv6 Header (H.Encaps.Red behavior).

  - IPv6 DA: 5f00:0:0100:0500:0300::

  - The packet has no SRH.

* Leaf1:

  - Packet in: (IPv6. DA=5f00:0:0100:0500:0300::)(ROCEv2)

  - Leaf1 has the SID 5f00:0:0100::/48 instantiated with the End with NEXT-CSID, PSP & USD behavior. As a result, it shifts, lookup, and forwards the packet.

  - Packet out: (IPv6. DA=5f00:0:0500:0300::)(ROCEv2)

* Spine5:

  - Packet in: (IPv6. DA=5f00:0:0500:0300::)(ROCEv2)

  - Spine5 has the SID 5f00:0:0500::/48 instantiated with the End with NEXT-CSID, PSP & USD behavior. As a result, it shifts, lookup, and forwards the packet.

- Packet out: (IPv6. DA=5f00:0:0500::)(ROCEv2)

* Leaf3:

  - Packet in: (IPv6. DA=5f00:0:0300::)(ROCEv2)

  - Leaf3 has the SID 5f00:0:0400::/48 instantiated with the End with NEXT-CSID, PSP & USD behavior. As a result it removes the outer IPv6 header and forward the inner packet.

  - Packet out: (ROCEv2)

* NIC3: receives the ROCEv2 packet, process it, and passes data to the GPU3.

Note that Leaf1, Spine5, and Leaf3 do not hold any state for this specific flow. It is a single uSID instruction per node instantiated upon cluster build-up and reused by all flows.

The flow for the traffic from GPU2 to GPU3 leverages the path Leaf2, Spine6, Leaf4. It does so by using the uSID Network Program 5f00:0:0200:0600:0400:: .

While in this example we have used the uN instruction, it can also be encoded using uA instructions specifying the sequence of interfaces.

5.3. Adaptive Routing with congestion feedback

At any time, during the execution of the AI job, Spine5 experiences congestion. NIC1 learns about the congestion of Spine5.

Within usecs, without any fabric signaling or new state at intermediate devices, NIC1 steers the traffic into a different path through the fabric. NIC1 switches the path from <Leaf1, Spine5, Leaf3> to <Leaf1, Spine6, Leaf3>. This is done simply by encapsulating any new traffic of the flow GPU1->GPU3 with the IPv6 DA 5f00:0:0100:0600:0300:: .

Note that the change of path is instantaneous. There is no routing protocol or control plane notification to the network devices to change the path. The fabric is entirely stateless, and the packet path is encoded into the IPv6 header built by the source NIC. This is essential as AI workloads cannot be exposed to slow reconvergence.

6. Benefits

* *Deterministic Path Placement*: SRv6 allows the NIC to control the path of each flow through the fabric.

* *Minimum-MTU*: A plain outer IPv6 encapsulation allows to encode 6 uSIDs in the outer DA. This implies that without the need of additional extension headers, only with 40Bytes of IPv6 encapsulation, we can encode up to 6 intermediate waypoints allowing to enforce a path in a 3-tier Clos network. This is sufficient to control a path hop-by-hop (link by link) through a leaf, spine, super-spine, spine, leaf.

* *Congestion Feedback Loop*: Instant rerouting at the source based on ECN, in-band measured One-Way and Two-Way latency, Packet Trimming feedback and in-band packet loss, without any dependency of routing protocols. There is neither any control-plane signaling involved between the GPU and the fabric, nor between the AI orchestrator and the fabric devices.

* *Standardization*: Open, vendor-agnostic implementation

* *Ease of operation*: as opposed to black-box proprietary solution which packs opaque layer-2 optimization, the SRv6 solution is minimalistic, IP based, fully standardized and a rich ecosystem (vendor, merchant and open source). The deterministic and open nature of the solution simplifies troubleshooting.

7. Hyperscale

AI workloads are deployed across thousands of GPUs in multi-tier Clos networks, requiring a networking architecture that scales efficiently. SRv6 uSID (NEXT-CSID) ensures deterministic path placement while maintaining scalability through the following mechanisms:

* *Stateless Fabric*: Unlike RSVP-TE or MPLS-TE, which require per-flow state on network devices, SRv6 enforces paths by including all the instructions in the packet header. This eliminates state explosion as the number of GPUs increases.

* *uSID Encapsulation*: The SRv6 uSID (NEXT-CSID) encoding allows paths to be efficiently encoded even in multi-tier topologies, reducing encapsulation overhead while supporting large deployments. If more than 6 instructions are required, a simple IPv6 Segment Routing Extension Header can be used to encode additional instructions.

* *Cross-Datacenter Extension*: The same SRv6-based mechanism can extend beyond a single cluster to multi-datacenter AI fabrics (i.e., inter-DC AI training), where deterministic path placement ensures efficient inter-cluster data transfers.

* *Overlay Tenant Separation*: SRv6 can provide per-tenant network segmentation, ensuring AI workloads from different tenants or jobs are isolated while sharing the same physical infrastructure. By adding into the network program VPN Service SIDs; traffic steering and resource allocation can be enforced at the network level without requiring additional overlay encapsulations.

8. Security Considerations

The deployment model described in this document is secured leveraging the mechanisms defined in [RFC8986].

9. Acknowledgements

The authors would like to recognize the work of Lihua Yuan, Guohan Lu, Rita Hui, and Riff Jiang at Microsoft.

Rita Hui presented this use-case at MPLS & SRv6 World Congress in March 2025. A recording is available here: https://www.segment-routing.net/conferences/Paris25-Microsoft-Rita-Hui/

The authors would like to acknowledge the work of the developers who have enabled this use-case in the open-source [SONiC] implementation. In particular: Carmine Scarpitta, Abhishek Dosi, Changrong Wu, Kumaresh Perumal, Eddie Ruan, and Yuqing Zhao.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9800] Cheng, W., Ed., Filsfils, C., Li, Z., Decraene, B., and F. Clad, Ed., "Compressed SRv6 Segment List Encoding", RFC 9800, DOI 10.17487/RFC9800, June 2025, <https://www.rfc-editor.org/info/rfc9800>.

11. Informative References

[IBTA-ROCEv2]
InfiniBand Trade Association, "InfiniBand Architecture Specification Volume 1, Release 1.2.1, Annex A17: ROCEv2", 2 September 2014, <https://web.archive.org/web/20200917012109/https://cw.infinibandta.org/document/dl/7781>.

[SONiC] Linux Foundation, "SONiC", <https://sonicfoundation.dev/>.

Authors' Addresses

Clarence Filsfils
Cisco Systems
Belgium
Email: cf@cisco.com

Chris Martin
Oracle Cloud
United States of America
Email: christian.j.martin@oracle.com

Kiran Pillai
IBM
United States of America
Email: Kiran.Pillai@ibm.com

Pablo Camarillo Garvia (editor)
Cisco Systems
Spain
Email: pcamaril@cisco.com

Ahmed Abdelsalam
Cisco Systems
Italy
Email: ahabdels@cisco.com