SRv6 Operations
Internet-Draft
Intended status: Informational
Expires: 5 December 2026

C. Filsfils
P. Camarillo, Ed.
Cisco Systems
G. Lu
Microsoft
J. Brar
D. Becker
A. Jouhari
Oracle
K. Pillai
IBM
A. Abdelsalam
Cisco Systems
J. Tantsura
NVIDIA
K. Patel
Arrcus, Inc.
3 June 2026

SRv6 for Deterministic Path Placement in AI Backends
draft-filsfils-srv6ops-srv6-ai-backend-04

Abstract

This document describes how SRv6 uSID (NEXT-CSID) enables deterministic path placement in AI backend fabrics through L3-L4 integration: the transport stack on the NIC encodes each path as an ordered list of segments (a uSID network program) in the packet header, while the fabric forwards statelessly. It explains operational benefits including deterministic probing and alignment with hyperscale production deployments.

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 5 December 2026.

Table of Contents

1. Introduction

Hyperscale AI training clusters rely on massive GPU-to-GPU data exchanges, where training-step synchronization delays from congestion and packet loss directly degrade training performance and operational cost.

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) enables *L3-L4 integration* in AI backend fabrics: the transport stack on the NIC (i.e., SmartNIC, DPU) controls which path each packet follows by encoding an ordered list of segments in the outer IPv6 header, while switches perform simple, static forwarding without per-flow state. This ensures predictable performance, fine-grained traffic control, and rapid reaction to congestion without fabric reconvergence.

This model is deployed at hyperscale. OpenAI, Microsoft, and Oracle Cloud Infrastructure operate training clusters using Multipath Reliable Connection (MRC) over static SRv6 source routing. MRC extends RoCEv2 with multipath packet spraying and transport-layer path selection; SRv6 provides a deterministic mapping from each path identifier to a unique physical path through the fabric. Section 10 summarizes these deployments and provides references.

[SRv6-E2E-Frontend-WAN] 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.

L3-L4 integration Coordination between the network layer (static SRv6 forwarding in the fabric) and the transport layer (NIC-controlled path selection, congestion response, and probing), without switch-based dynamic routing or per-flow network state.

Deterministic path placement Encoding by the source transport of the path as an SRv6 uSID network program (an ordered list of segments in the packet) so each packet or spray round follows a fixed physical path through the fabric. Distinct path identifiers map to disjoint segment programs over the multi-plane topology. Paths are not assigned by a centralized flow scheduler or traffic-engineering controller; the network holds no per-flow state and does not pre-install paths.

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*: The fabric must minimize avoidable disruption and support fast, predictable recovery. Even brief hotspots or reconvergence delays can amplify tail latency across a synchronized job. Designs that provide multipath spraying over disjoint encoded segment programs, transport-controlled rerouting, and accurate probing reduce the risk that the network becomes the limiting factor during long training runs.

At hyperscale, faults are routine rather than exceptional. In a 54-day Llama 3 405B pre-training run on 16,384 GPUs, Meta reported 419 unexpected interruptions (about 78% hardware-related; 8.4% network switches and cables) [Llama3-Herd]. Synchronized training makes fabric congestion, loss, or degraded paths costly for jobs that run for weeks. Meta’s large-scale RoCE backend design is described in [Meta-RoCE-SIGCOMM24].

4. SRv6 for Deterministic Path Placement

The source encodes each path as a uSID network program in the packet header; transports spray packets across disjoint network paths, and choose a different path upon congestion. SRv6 enables L3-L4 integration: the transport stack on the NIC controls the AI workload traffic journey through the fabric by encoding an ordered list of segments in the packet header, while the network layer provides stateless static forwarding.

* *Control plane and orchestration*: At bring-up, the orchestrator discovers topology and the SRv6 uSIDs configured on each link. These uSID instructions are statically configured on the routers and are independent of any dynamic routing protocol state.

  - The orchestrator provides to the NICs with topology information, including the uSIDs available on each link in the fabric.

  - Based on that information, the NIC transport composes a path as a sequence of uSIDs and encodes the resulting network program in the outer IPv6 Destination Address of each packet. Encoding a path in this way *does not require any per-path communication between the orchestrator and the fabric*.

* *Transport stack on the NIC*: Before sending RoCEv2 traffic, the transport stack encapsulates the packet with an outer IPv6 header that carries the uSID program selected on the NIC for that packet or spray round.

  - An outer IPv6 header allows encoding 6 uSIDs in the Destination Address. This implies that even with a super-spine in a 3-tier Clos fabric, the entire path can be encoded without an additional Segment Routing Header (SRH).

* *Highly Scalable Stateless Fabric*: Routers enforce the path by following SRv6 instructions in the packet header. There is *no per-flow state in the network* (unlike MPLS RSVP-TE, which would require per-path state for each GPU-to-GPU deterministic path).

* *Congestion feedback loop*: The transport stack reacts in real time to congestion notifications (ECN, in-band latency measurement, Packet Trimming, in-band packet loss). At any time, without fabric wide signaling, the source NIC can change the path by updating the outer IPv6 Destination Address. Only the source changes; intermediate devices remain unchanged.

5. Statically Provisioned Fabric

In a dynamically routed fabric, protocols such as BGP maintain reachability across the Clos fabric. When a link fails or the topology changes, switches must reconverge: prefixes are withdrawn or re-advertised, each device updates its RIB, and forwarding entries are reprogrammed. A point-to-point link-down may be detected within milliseconds, but completing BGP reconvergence in a large datacenter fabric typically takes on the order of tens of milliseconds to a sub-second interval in optimized designs, and can be substantially longer when many paths or prefixes are affected [RFC7938]. At extreme scale, convergence may require many round-trip times across the fabric before forwarding is stable [MRC-SRv6-Paper].

In the statically provisioned SRv6 model in this document, GPU traffic paths do not depend on that reconvergence cycle. uSID instructions are configured at fabric bring-up and remain in place on the routers. When a link degrades or fails, the fabric does not wait for BGP or other dynamic routing to install a new path. The source NIC or transport stack detects the problem through loss, ECN,

latency, or probing, selects another uSID network program from the topology information it already holds, and encodes it in the outer IPv6 Destination Address of subsequent packets. That change is local to the sender, takes effect within microseconds, and does not require signaling to intermediate switches or coordination with routing protocol timers [Microsoft-Fairwater].

6. Deterministic Probing

Accurate visibility into fabric health is essential for AI backend operations: scheduling repairs, tuning performance, and correlating transport behavior with physical paths. Traditional approaches face limitations at scale.

With ECMP-based forwarding, probe and data packets may traverse different paths because hashing is sensitive to header fields. Mechanisms that send probes to remote nodes depend on remote availability and do not always localize faults precisely. ICMP probes to switches are often handled by the control plane, limiting probe frequency. Dynamic routing can change forwarding while probes are in flight, reducing the ground truth of measurements.

SRv6 uSID source routing enables *deterministic probe pinning*: a probe encoded with the same segment program as data traffic follows the identical physical path through the fabric. There is no ECMP ambiguity, no dependence on switch control-plane ICMP handling for path validation, and no interaction with dynamic routing reconvergence during measurement.

* *Path pinning*: Each probe is assigned a specific SRv6 network program, so operators and transport stacks know exactly which links and switches a measurement traverses.

* *Dataplane fidelity*: Probes are forwarded like data traffic in the dataplane, enabling high-frequency monitoring suitable for large clusters.

* *Self-probes and localization*: Agents on cluster nodes can source-route probes to a top-of-rack switch and back, or to aggregation switches and back, localizing NIC-to-fabric or fabric-internal faults without requiring a remote peer to be up.

* *Transport alignment*: When the transport stack selects paths using SRv6 programs, health probes and background path validation use the same encoding, so measurements reflect the paths data actually uses.

Deterministic probing simplifies denylist and spray/path-selection policies on the NIC, supports resurrection of paths after transient failures, and provides operations teams ground-truth telemetry independent of switch control-plane health.

7. Illustration

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

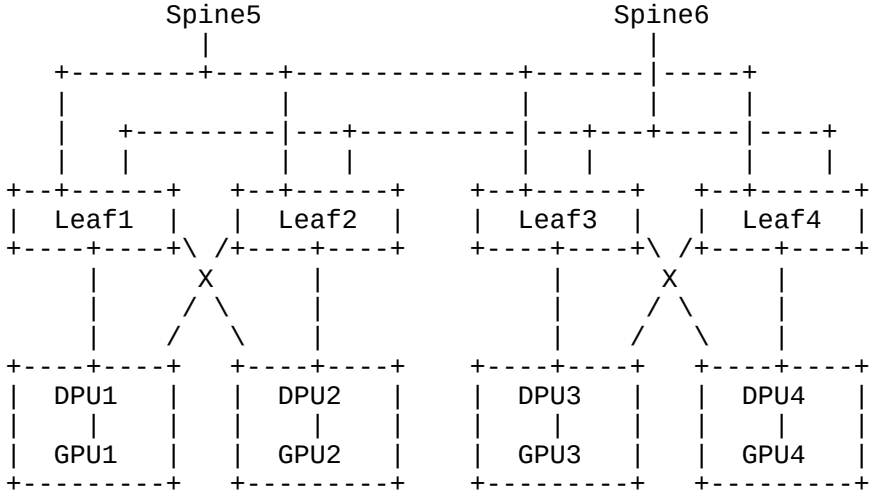

Figure 1: Reference Topology

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.

7.1. 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)

7.2. SRv6-Based Deterministic Path Selection

During a collective, GPU1 and GPU2 send traffic to GPU3. The transport on each NIC sprays packets across disjoint uSID network programs, each encoded as an ordered segment list in the outer IPv6 header. The orchestrator supplies topology and uSID information at bring-up; path choice and spraying are performed by the NIC transport (e.g., MRC) at send time. Example programs:

* GPU1->GPU3: via Leaf1, Spine5, Leaf3 (uSID program 5f00:0:0100:0500:0300::)

* GPU2->GPU3: via Leaf2, Spine6, Leaf4 (uSID program 5f00:0:0200:0600:0400::)

When sending RoCEv2 traffic from GPU1 to GPU3:

* 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=5f00:0:0300::)(ROCEv2)

* Leaf3:

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

  - Leaf3 has the SID 5f00:0:0300::/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 traffic using that program. GPU2->GPU3 uses the second example program; forwarding is the same stateless model on each hop.

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

7.3. Transport-Controlled Path Selection with Congestion Feedback

At any time during the execution of the AI job, Spine5 may experience congestion. The transport stack on NIC1 detects this via ECN, in-band latency, packet trimming, or loss feedback.

Within microseconds, without fabric signaling or new state at intermediate devices, the transport stack steers traffic to a different path. NIC1 switches the path from <Leaf1, Spine5, Leaf3> to <Leaf1, Spine6, Leaf3> by encapsulating new traffic for GPU1->GPU3 with IPv6 DA 5f00:0:0100:0600:0300::.

This is not switch adaptive routing (e.g., dynamic ECMP or BGP reconvergence). Path changes are made only at the source NIC; switches continue static SRv6 forwarding. The fabric is entirely stateless, and the packet path is encoded in the IPv6 header built at the source. Separating transport-controlled path selection from switch-based adaptive routing avoids unpredictable interactions at scale and is essential because AI workloads cannot tolerate slow reconvergence [Microsoft-Fairwater].

8. Benefits

* *Deterministic Path Placement*: SRv6 allows the NIC to encode, per packet or spray round, distinct uSID network programs that pin traffic to disjoint paths 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 solutions that pack opaque layer-2 optimizations, the SRv6 solution is minimalistic, IP-based, fully standardized, and supported by a rich ecosystem (vendor, merchant silicon, and open source). The deterministic and open nature of the solution simplifies troubleshooting.

* *Production validation*: Hyperscale AI training clusters operate static SRv6 source routing at scale; see Section 10.

9. Massive Scale

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 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 encode additional instructions.

* *Multi-plane topologies*: High-radix, multi-plane Clos designs spread NIC capacity across parallel network planes, improving physical redundancy and enabling clusters well beyond 100K GPUs in two-tier fabrics while keeping latency low.

* *Cross-Datacenter Extension*: The same SRv6-based mechanism can extend beyond a single cluster to multi-datacenter AI fabrics (inter-DC AI training), where deterministic path placement ensures efficient inter-cluster data transfers. SRv6 network programs can be extended to forward between clusters using the same path-encoding model.

* *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. Dedicated network resources can be assigned on a per-tenant basis in the fabric, providing resource isolation so that bandwidth, paths, and forwarding capacity for one tenant are not conflated with another. By adding VPN Service SIDs into the encoded network program, distinct path identifiers and network planes per tenant can be enforced at the network level without additional overlay encapsulations.

10. Deployment

Static SRv6 uSID (NEXT-CSID) source routing is deployed at hyperscale in production AI training clusters together with Multipath Reliable Connection (MRC), an RDMA transport developed collaboratively by OpenAI, Microsoft, NVIDIA, AMD, Intel, and Broadcom [MRC-SRv6-Paper] [OpenAI-MRC]. MRC extends RoCEv2 Reliable Connection with multipath packet spraying, selective retransmission, packet trimming for incast, and transport-layer path health management; it runs Ethernet in best-effort mode and relies on fast recovery at the transport layer rather than Priority Flow Control. In these fabrics, dynamic routing in switches is disabled: each packet's path is encoded in the outer IPv6 destination using uSID, path identifiers map algorithmically to SRv6 network programs, and the transport stack gains deterministic control while routers forward statelessly. This combination has been used to train frontier large language models on clusters exceeding 100,000 GPUs, with implementations on 400 and 800 Gb/s RDMA NICs and SRv6 forwarding across multiple switch platforms and NOS distributions [MRC-SRv6-Paper].

The same architecture spans OpenAI, Microsoft, and Oracle Cloud Infrastructure (OCI) training sites that operate as one coherent design rather than isolated experiments. OpenAI runs MRC over static SRv6 on its largest NVIDIA GB200 supercomputers [OpenAI-MRC] [MRC-SRv6-Paper]; Microsoft's Fairwater supercomputer applies the same model on a two-tier, multi-plane fabric, removing BGP and other dynamic routing from the scale-out network in favor of compact uSID source routing, with probe traffic following the same paths as data for ground-truth visibility [Microsoft-Fairwater]; and OCI's Oracle Acceleron multiplanar networking deployed MRC and SRv6 source-based routing at scale, including the Stargate datacenter in Abilene, Texas, with path intelligence at the NIC and simple static forwarding in the network [Oracle-Acceleron-MRC] [Oracle-Acceleron-Arch]. Across these environments, multipath spraying and SRv6 path pinning reduce flow collisions, improve utilization across network planes, and allow large synchronous training jobs to continue through link flaps and partial failures that previously caused restarts [OpenAI-MRC] [Microsoft-Fairwater]. Microsoft has additionally open-sourced MRC software interfaces and SONiC SRv6 enhancements for AI backend networks [Microsoft-Fairwater].

11. Security Considerations

This document is informational and does not define a new protocol. Security for the SRv6 data plane and segment programming is covered in [RFC8986]. The deployment model assumes a dedicated AI backend fabric in a single administrative domain: an orchestrator statically provisions uSIDs on routers and supplies topology to NICs, and transport stacks on GPU hosts encode segment programs in each packet while the fabric forwards without per-flow state. Security therefore depends on integrity of provisioning, access control over router configuration, and path-selection logic on each host.

Because the source encodes the full segment list, a compromised or misconfigured host could steer traffic along unintended paths. Operators SHOULD limit which workloads may send SRv6-encapsulated traffic and constrain hosts to programs authorized by the orchestrator. Backend fabrics are typically isolated from untrusted networks, and operators SHOULD maintain that separation. Security of RoCEv2 and Multipath Reliable Connection (MRC) transports is outside the scope of this document.

12. Contributors

The following person contributed significantly to this document:

Chris Martin (Cisco Systems), <martincj@cisco.com>

13. Acknowledgements

The authors thank the teams behind the MRC and SRv6 production deployments described in Section 10, including contributors at OpenAI, Microsoft, Oracle Cloud Infrastructure, NVIDIA, AMD, Intel, and Broadcom.

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

Pablo Camarillo and Rita Hui presented this use case at NANOG 96; a recording is available at https://www.segment-routing.net/conferences/2026-02-02-NANOG96-SRv6-AI-Backend-Microsoft. Clarence Filsfils and Guohan Lu presented related material at OCP EMEA 2026; a recording is available at https://www.segment-routing.net/conferences/2026-OCP-EMEA-summit-scalable-ai-protocol-stack.

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, Yuqing Zhao, Rajasekar Raja, and Vivek Venkatraman.

14. 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>.

15. Informative References

[SRv6-E2E-Frontend-WAN] Filsfils, C., Camarillo, P., Michielsen, K., and A. Gorovoy, "SRv6 End-to-End DC Frontend and WAN", Work in Progress, Internet-Draft, draft-filsfils-srv6ops-srv6-e2e-dc-frontend-wan-01, 2026, <https://datatracker.ietf.org/doc/html/draft-filsfils-srv6ops-srv6-e2e-dc-frontend-wan-01>.

[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/>.

[MRC-SRv6-Paper]
Araujo, J., Chow, A., Handley, M., Lu, G., and L. Yuan, "Resilient AI Supercomputer Networking using MRC and SRv6", May 2026, <https://arxiv.org/abs/2605.04333>.

[OpenAI-MRC]
OpenAI, "Supercomputer networking to accelerate large scale AI training", May 2026, <https://openai.com/index/mrc-supercomputer-networking/>.

[Microsoft-Fairwater]
Cutts, V. and J. Jose, "Building resilient networks for AI supercomputers", May 2026, <https://techcommunity.microsoft.com/blog/azurehighperformancecomputingblog/building-resilient-networks-for-ai-supercomputers/4516919>.

[Oracle-Acceleron-MRC]
Vincent, P., "First Principles: Unlocking Oracle Acceleron Multiplanar Fabric with Multipath Reliable Connection", May 2026, <https://blogs.oracle.com/cloud-infrastructure/first-principles-multipath-reliable-connection>.

[Oracle-Acceleron-Arch]
Vincent, P., "First Principles: Oracle Acceleron Multiplanar Networking Architecture", May 2026, <https://blogs.oracle.com/cloud-infrastructure/first-principles-acceleron-multiplanar-networking>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

[Llama3-Herd]
Dubey, A., Jauhri, A., and A. Pandey, "The Llama 3 Herd of Models", July 2024, <https://arxiv.org/abs/2407.21783>.

[Meta-RoCE-SIGCOMM24]
Gangidi, A., Miao, R., and S. Zheng, "RDMA over Ethernet for Distributed AI Training at Meta Scale", August 2024, <https://doi.org/10.1145/3651890.3672233>.

Authors' Addresses

Clarence Filsfils
Cisco Systems
Belgium

Email: cf@cisco.com

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

Guohan Lu
Microsoft
United States of America
Email: gulv@microsoft.com

Jag Brar
Oracle
United States of America
Email: jag.brar@oracle.com

David Becker
Oracle
United States of America
Email: david.b.becker@oracle.com

Abderrahman Jouhari
Oracle
United States of America
Email: abderrahman.jouhari@oracle.com

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

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

Jeff Tantsura
NVIDIA
United States of America

Email: jefftant.ietf@gmail.com

Keyur Patel
Arrcus, Inc.
United States of America
Email: keyur@arrcus.com