RTGWG
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

P. Camarillo, Ed.
C. Filsfils
N. Chachmon
O. Iny
Cisco
Y. Su
R. Jiang
Alibaba

2 March 2026

Lightspeed Notification Protocol
draft-camarillo-rtgwg-lsn-00

Abstract

This document defines the Lightspeed Notification Protocol (LSN), a hardware-accelerated signaling mechanism designed for sub-100 microsecond network convergence in AI/ML data center fabrics. By operating entirely within the forwarding plane, LSN bypasses traditional CPU-based latencies to propagate link failures and congestion via a hardware-efficient encoding. It serves as a high-speed complement to routing protocols like BGP, providing an immediate hardware "veto" to prune congested/failed paths while maintaining control-plane stability for path recovery.

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 3 September 2026.

Table of Contents

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) workloads impose stringent demands on data center fabrics, characterized by high-bandwidth, synchronized and bursty collective communication patterns. As outlined in [draft-clad-rtgwg-ipfrr-aiml], these environments require network convergence times in the sub-100 microsecond range to avoid performance degradation caused by packet loss and jitter.

Traditional routing protocols and convergence mechanisms rely on control-plane intervention, where link state changes are processed by the CPU. This introduces activation latencies in the tens of milliseconds, which is orders of magnitude too slow for AI/ML requirements. Furthermore, existing mechanisms typically operate on a binary "up/down" model, failing to account for capacity degradation or congestion.

This document defines a Hardware-Accelerated Notification Protocol designed to operate entirely within the forwarding plane to handle both network failures and congestion events. This protocol serves as a complementary mechanism to traditional routing protocols (e.g., BGP). The interaction between the routing protocol and this notification mechanism is functionally equivalent to a boolean AND operation: a path is eligible for forwarding only if permitted by the routing protocol AND confirmed healthy by the hardware notification.

This enforces an asymmetric reaction: "bad news" (failure or congestion) acts as an immediate hardware veto, disabling the path in sub-microseconds. Conversely, "good news" (recovery) is gated by the routing protocol, ensuring the control plane has fully converged before traffic resumes. This approach combines sub-microsecond protection with routing coherence/stability.

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. Problem Statement

Current IP Fast Reroute (IPFRR) and resiliency mechanisms face fundamental limitations when applied to the extreme performance requirements of AI/ML fabrics.

Latency of CPU-Based Processing

Existing failure detection and propagation mechanisms generally trigger interrupts that must be processed by a line-card or system CPU. As noted in [draft-clad-rtgwg-ipfrr-aiml], this CPU-mediated path introduces an activation delay typically in the range of 10-50 milliseconds. For AI training workloads requiring barrier synchronization, this latency results in unacceptable flow disruption. To achieve the required sub-100 microsecond convergence, the propagation and processing of network state changes must occur with sub-microsecond latency, necessitating a solution implemented directly in hardware (NPU) logic without CPU intervention.

Inability to Handle Congestion and Partial Failures

Standard routing protocols and mechanisms like ECMP operate on a binary failure model-a link is either available or unavailable. However, AI fabrics frequently experience "brownout" scenarios, such

as partial capacity reduction (e.g., loss of a single lane in a port group) or acute congestion. Current mechanisms lack the granularity to signal these states, causing traffic to be blackholed or hashed onto congested paths.

Limited Visibility of Remote Network State

In multi-stage Clos topologies and irregular scale-across networks, a failure or congestion event often occurs multiple hops away from the ingress point. Local protection mechanisms (like classic LFA) implemented at the point of failure are often insufficient or lead to suboptimal "hairpin" routing that increases latency. Current routing protocols do not provide ingress nodes with the real-time visibility into remote link states or congestion levels required to prevent traffic from entering compromised paths or to optimize load balancing across the fabric.

3. Protocol Specification

3.1. Notification Packet Format

The packet header is the following:

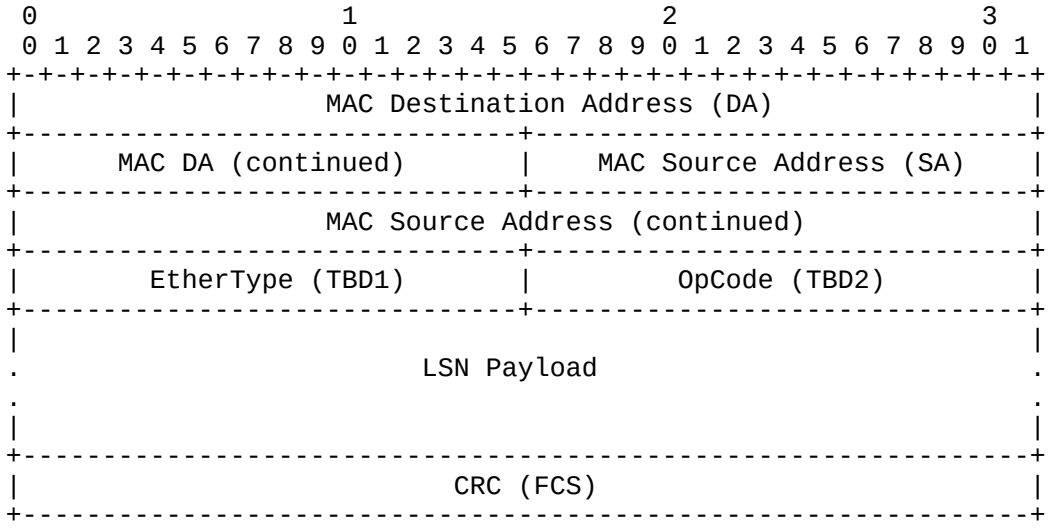

Figure 1

Notes:

* MAC Destination Address: 01-80-C2-00-00-01 (IEEE MAC-specific control protocols)

* Ethertype = TBD1 (Currently uses 0x8808 [IEEE802.3] MAC Control for interoperable running code)

* Opcode = TBD2 (Currently uses 0x5AA5 for interoperable running code)

The LSN Payload is defined as follows:

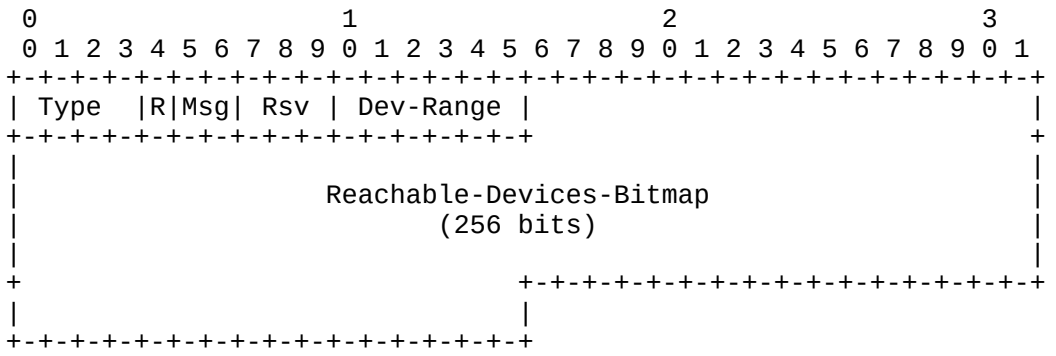

Figure 2

Field Descriptions:

* Header Type (4 bits): Value ’12’ - Identifies this as a topology reachability packet.

* R (1 bit): Reserved. It MUST be set to zero upon transmission and ignored upon receival.

* Msg-type (2 bits): Indicates the information conveyed by the bitmap. It may represent reachability or congestion status. The congestion status is defined as 3 different congestion levels, which are implementation specific. The following Msg-Types are defined:

  - 0x0: Reachable

  - 0x1: Reachable and not congested (Congestion Level 1)

  - 0x2: Reachable and not congested (Congestion Level 2)

  - 0x3: Reachable and not congested (Congestion Level 3)

* Reserved (3 bits): Reserved. It MUST be set to zero upon transmission and ignored upon receival.

* Reachable-Devices-Range (6 bits): Determines the range of the reachable devices bitmap. The range per packet is within [Reachable-Devices-Range*256, Reachable-Devices-Range *256 + 255].

  - '0': Devices 0-255

  - '1': Devices 256-511

  - '2': Devices 512-767

  - ... (Covers up to 16K devices).

* Reachable-Devices-Bitmap (256 bits): One bit per device that indicates if the device is reachable through the device sending this message.

3.2. Origination and Processing Procedures

This section defines the procedures for assigning node identifiers, generating notification messages, and processing received notifications in hardware.

3.2.1. Node Identification

To facilitate efficient bitmap-based signaling, every leaf node within the fabric is assigned a unique *Global Node ID*. This identifier maps the node to a specific bit position within the notification bitmap.

The allocation, synchronization, and lifecycle management of these Global Node IDs are performed by an external central controller or management plane. The specific mechanisms for ID allocation are outside the scope of this document.

3.2.2. Node Origination

Each node is responsible for periodically generating notification messages that convey the direct reachability or congestion state of potential destinations to the rest of the fabric.

Message Scope and Fragmentation

A single notification message supports a payload covering up to 256 destination devices.

* The message header MUST specify the Reachable-Devices-Range (or Base ID), indicating the starting Global Node ID for the bitmap payload.

* If the fabric scale (Radix) exceeds 256 nodes, the originating node MUST generate multiple distinct notification messages. Each message corresponds to a different, non-overlapping Reachable-Devices-Range to cover the full topology.

MSG-Type and State Encoding

The interpretation of the bitmap payload is determined by the MSG-Type field in the header. The specific assignment of MSG-Type values is implementation-specific.

* *Reachability Information:* When the MSG-Type indicates reachability, the bitmap encodes the binary status of the links:
  - *Bit value 1:* Indicates the destination node is directly reachable.
  - *Bit value 0:* Indicates the destination node is not reachable.
* *Congestion Information:* When the MSG-Type indicates congestion, the bitmap conveys the congestion status of the associated paths. The mapping of queue congestion levels to the various congestion values is implementation-specific, provided it supports the hardware logic required for path pruning or de-prioritization.

As mentioned earlier, the messages are periodically generated. However, if there is a significant change in the quality of one path (congestion level change, of interface status change), a notification message is also triggered.

3.2.3. Hardware Processing and Aggregation

Upon receipt of a notification message, the receiving node MUST process the packet directly in the forwarding hardware without punting to the control plane CPU.

State Aggregation

The receiving node maintains a global state table representing the health of the fabric. When a message is received:

1. The hardware identifies the segment of the global state corresponding to the Reachable-Devices-Range in the packet.
2. The received bitmap is overlaid onto the local state, updating the reachability or congestion status for the specific subset of nodes.

Forwarding Decision Logic

To derive the final forwarding decision, the hardware performs a bitwise operation between the standard routing table and the aggregated notification state.

* For reachability, this is functionally a *Bitwise AND* operation: A path is valid if and only if the Routing Protocol has installed it (Logical 1) *AND* the Notification Protocol reports it as reachable (Logical 1).

* If the result of this operation is 0 for a specific next-hop or path, the hardware immediately removes that path from the Equal-Cost Multi-Path (ECMP) set or activates a pre-programmed backup path.

4. Illustration: Usage in the DC

This section illustrates the operation of the Hardware-Accelerated Notification Protocol within a massive scale 2-tier Spine-Leaf Data Center topology. This example focuses exclusively on *reachability* signaling to demonstrate the bitwise logic used to prune invalid paths.

4.1. Topology and ID Assignment

Consider a fully connected Clos fabric consisting of *256 Spine switches* and *256 Leaf switches*.

* *Spine Naming Terminology:* The Spines are identified as SA, SB, SC, ..., SZ, SAA, SAB, ..., SIU, SIV.

* *Leaf Naming & Global IDs:* The Leaves are identified as L0, L1, L2, ... through L255. Each Leaf is assigned a Global Node ID corresponding to its index (e.g., Leaf_5 has ID 5).

* *Traffic Flow:* An ingress Leaf (L_Ingress) sends traffic to a destination Leaf_5.

* *Routing State:* Under normal operation, BGP advertises that Leaf_5 is reachable via all 256 Spines. The Ingress Leaf maintains an ECMP group for Leaf_5 containing 256 next-hops: {SA, SB, SC, ... SIV}.

4.2. Failure Scenario

A physical link failure occurs between Spine_A and the destination Leaf_5.

1. *Detection and Generation:* Spine_A detects the port connected to Leaf_5 is down. The hardware at Spine A immediately generates a notification packet to be transmitted to all connected leaves (including L_Ingress).

   * *MAC Destination Address:* 01-80-C2-00-00-01 (MAC Control)

   * *MAC Source Address:* Spine_A

   * *Reachable-Devices-Range (Base ID):* 0 (Covering Leaves 0-255).

   * *MSG-Type:* Reachability.

   * *Bitmap Payload:* A 256-bit string where the bit at index 5 is set to 0 (Unreachable), and all other connected leaves are set to 1 (Reachable).

   ```
   Bitmap from Spine A: [1, 1, 1, 1, 1, 0, 1, ... 1]
                                        ^
                                   Index 5 (L5) set to 0
   ```

   Figure 3

2. *Hardware Processing at Ingress:* The Ingress Leaf (L_Ingress) receives the notification from Spine_A. The forwarding engine performs a logical *AND* operation between the Routing Table state and the received Notification state specifically for the path via Spine_A.

   * *Path via Spine_A:*

     - Routing Table State for Leaf_5: 1 (Control plane has not yet converged; leaf_5 still considered reachable).

     - Notification State from Spine_A for Leaf_5: 0 (Hardware Notification reported Down).

     - *Result:* 1 AND 0 = 0 (Path Invalid).

   * *Path via Spine_B (and others):*

     - Routing Table State for Leaf_5: 1.

     - Notification State from Spine_B for Leaf_5: 1 (No failure reported by Spine_B).

     - *Result:* 1 AND 1 = 1 (Path Valid).

3. *Forwarding Result:* Because the result for Spine_A is 0, the forwarding hardware excludes Spine_A from the ECMP group for destination Leaf_5. Traffic to Leaf_5 is instantly rebalanced across the remaining 255 Spines (B...IV). Balancing of traffic to the other leaves is unchanged.

This mechanism ensures sub-microsecond protection. When the link Spine_A-Leaf_5 is eventually restored, Spine_A will send notification message with an updated bitmap where index 5 is set to 1. However, traffic will not resume via Spine_A until BGP also re-installs the route, satisfying the 1 AND 1 condition.

5. Illustration: Usage across DCs

Future revisions of this document will document how the mechanism defined here can be used for DCI/regional networks.

6. Security Considerations

The Lightspeed Notification Protocol (LSN) is designed to achieve with sub-100 microsecond convergence directly in the forwarding hardware. To achieve this performance, LSN messages do not include cryptographic authentication or integrity checks. Consequently, the protocol introduces several security considerations that must be mitigated by the deployment architecture.

Spoofing and Denial of Service (DoS): Because LSN messages are unauthenticated, an attacker capable of injecting forged LSN frames into the fabric could broadcast false congestion or failure notifications. This would cause receiving nodes to prune valid paths from their forwarding tables, potentially leading to forced congestion, suboptimal routing, or a complete denial of service.

Mitigation via Boundary Filtering: LSN is intended exclusively for internal use within highly controlled, single-domain AI/ML data center fabrics. To prevent spoofing, boundary nodes and leaf switches MUST implement strict port-level filtering. LSN notification packets MUST be dropped unconditionally on any port facing a server, host, or external network. LSN processing MUST only be enabled on trusted, switch-to-switch infrastructure links.

Fail-Safe Routing Logic: As defined in this document, LSN acts only as a hardware "veto" (a logical AND operation with the routing protocol). While an attacker can maliciously prune a path, they cannot force the network to forward traffic into a blackhole or an invalid topology segment, because reachability requires the routing protocol (e.g., BGP) to also authorize the path. The control plane remains the ultimate source of truth for topology validation.

7. IANA Considerations

This document requests the following allocations.

* *Ethertype:* A new Ethertype for the Lightspeed Notification Protocol (LSN).

* *Opcode:* A new Opcode for LSN Bitmap Reachability.

Upon assignment of these values, the RFC Editor is requested to replace all instances of "TBD" for the Ethertype and Opcode with the newly allocated hexadecimal values.

8. Acknowledgements

The authors would like to acknowledge the following people: Kris Michielsen.

9. 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>.

10. Informative References

[draft-clad-rtgwg-ipfrr-aiml] Clad, F., Filsfils, C., Jiang, R., and D. Cai, "IP Fast Reroute for AI/ML Fabrics", 2 March 2026, <https://datatracker.ietf.org/doc/draft-clad-rtgwg-ipfrr-aiml/>.

[IEEE802.3] IEEE computer Society, "IEEE Standard for Ethernet; IEEE Std 802.3-2022", 13 May 2022, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9844436>.

Authors' Addresses

Pablo Camarillo Garvia (editor)
Cisco
Spain

Email: pcamaril@cisco.com

Clarence Filsfils
Cisco
Belgium
Email: cf@cisco.com

Nadav Chachmon
Cisco
Israel
Email: nchachmo@cisco.com

Ofer Iny
Cisco
Israel
Email: oiny@cisco.com

Yuanchao Su
Alibaba
China
Email: yitai.syc@alibaba-inc.com

Roy Jiang
Alibaba
China
Email: royjiang@aliyun-inc.com