Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 11 October 2026

H. Song, Ed.
Futurewei Technologies
K. Zhu
Huawei Technologies
J. Song
China Mobile
9 April 2026

# Network Header Compression for Converged AI Network
draft-song-cain-header-00

## Abstract

We envision the scale-up, scale-out, and scale-across networks for AI computing would eventually converged. The draft describes a scheme for L3 packet header compression in converged AI networks where IPv6 are assumed to be the L3 protocol, and a unified fabric supports all kinds of traffic. The header size can be reduced to 8 octets for packets transferred with a single super-node, representing 80% overhead saving. The document discusses the motivation, requirements, benefits, and feasibility in addition to the header format proposal.

## 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 11 October 2026.

Table of Contents



## 1. Introduction

The AI scale-up network is shifting from proprietary solutions to standard Ethernet-based, driven by several forces including vendor lock-in breaking, cost structure, and operational simplicity. Although in the mainstream the scale-up network and the scale-out network remain physically and semantically separated, there is not a fundamental barrier preventing the two from being bridged together (i.e., allowing direct packet forwarding between the two domains), or sharing the physical interfaces (i.e., mixing the traffic). The boundary is becoming blurry. Recent research [hot25] has proposed that, to support more flexible routing and load balancing, it is preferred to unify the scale-up domain and the scale-out domain. There are industry practices on the horizon as well. For example, Intel's Gaudi 3 [gaudi] only provides 24 unified RoCEv2 ports, removing the separation of the two domains altogether; Huawei's UBMesh [ub] uses unified bus to provide hierarchical interconnections extendable to multiple levels without distinguishing the two domains.

Meanwhile, scale-across network is becoming the third pillar of AI infrastructure which extend the scale-out network across multiple AI data centers. AI infrastructure is undergoing a paradigm shifts from super-node as a computer to datacenter as a computer to multi-datacenter as a computer. In the converged AI network, packets can move between any two AI accelerator nodes regardless of their locations. It is desirable to have a common L3 protocol for the unified routing and forwarding functions within and among the domains.

On the other hand, the accelerator affinity in conventional scale-up domain allows data transactions with more efficient memory semantics (i.e., the nodes in the same domain can share the unified memory space), while the scale-out domain typically resorts to message semantics for data move (e.g., RDMA). The two domains can use very different protocol stacks. For example, the scale-up domain uses L2 switching only but the scale-out domain requires L3 routing; even with the unified Ethernet-based L2, the L4 transport protocol diverge again. To unify the two domains, and further extend to the scale-across domain in the future, we need to introduce a unified L3 network protocol, based on the already unified Ethernet-based L2 link protocol, with the coexistence of potentially multiple L4+ protocols. This is critical for enabling a unified AI fabric with the benefits of open ecosystem, low cost, and simplified operation.

While IPv6 provides enough scalability and extensibility to support the converged AI network, its header overhead is too big for certain communication scenarios. For example, memory-semantic traffic （i.e., LD/ST) usually has the minimum sized payload; a large number of packets for signaling (e.g., ACK, CNP, barrier, trimmed packets) and for network control/management plane are also small. The base header of IPv6 is 40 bytes. When extension header is needed (e.g., SRv6), the size would be even greater. The L3 header poses a significant overhead to such packets. Given the bandwidth of AI network is always a precious resource and performance bottleneck, it is critical to reduce the network header overhead yet maintain the benefits of scalability and extensibility. Therefore, we need an effective header compression scheme which is suitable for the converged AI network, and retain the compatibility with standard IPv6 at the scale-across domain which shares the public WAN.

This document describes the Converged AI Network (CAIN) L3 header format. It is an IPv6 header compression scheme based on Short Hierarchical IP Address (SHIP) [I-D.song-ship-edge]. Within an AI DCN, it supports multiple hierarchical levels. The simplest two-level form distinguish the scale-up and scale-out domains. It can also support more levels as described in UBMesh [ub], and other hierarchical topologies (e.g., rack, pod, super-pod, etc.). To support scale-across at the DCN gateway, the CAIN header are translated into standard IPv6 header format for WAN compatibility.

### 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. Related Work

The related works and their limitations are summarized as follows.

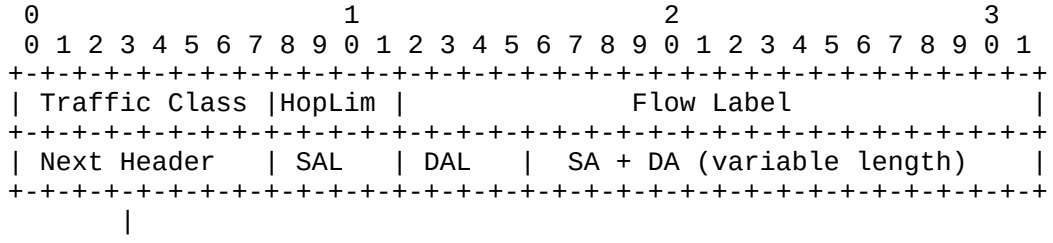
1. AFH: Broadcom’s scale-up Ethernet framework specifies a compact AI Fabric Header (AFH) [afh]. However, it encodes the node address information in MAC header and only works in L2 scale-up domain, unsuitable to be used as the CAIN header.

2. SUNH: The Internet draft [I-D.herbert-sunh] proposes an L3-based scale-up network header which supports L3 routing. However, it is designed with fixed address size and for scale-up network only. Therefore, the flexibility and extensibility are limited.

3. IPHC and SCHC: IPv6 header compression schemes have been specified for some particular low power IoT networks such as 6loWPAN [RFC6282] and LPWAN [RFC8724]. These networks feature low data rate and are insensitive to latency. However, due to the low power constraint, they are extremely sensitive to bandwidth efficiency. Therefore, they adopt the context-based compression schemes which, while needing extra storage and computation, can reduce the header overhead to the utmost extend. In contrast, AI networks requires high bandwidth, low latency, and low processing complexity which render these schemes unsuitable.

## 3. CAIN Header Format

The proposed CAIN Header format is as follows.




The traffic class, flow label, and next header fields are inherited from IPv6 without any change. The hop limit field is reduced to 4 bits to support up to 15 hops, which is enough because the number of hops in AI network is typically small (e.g., a 3-layer CLOS network has 5 hops at most).

In the CAIN header, no Version field is included; the protocol is identified by the EtherType value at the L2 layer. No Payload Length field is included; the payload length is derived from the L2 frame length minus the CAIN header length. The header length is deterministically computed as:

```
header_length = ceil4(6 + SAL_bytes + DAL_bytes)

where ceil4(x) = (x + 3) AND NOT(3)
      SAL_bytes = (SAL == 0) ? 16 : SAL
      DAL_bytes = (DAL == 0) ? 16 : DAL
```

4-bit SAL and DAL indicate the source address (SA) and the destination address (DA)’s length in 8-bit steps. For example, "0001" stands for 8, and "0010" stands for 16. Specifically, "0000" stands for 128, which means the corresponding address is a 128-bit IPv6 address. Such an address allocation scheme allows the lowest-level scale-up network to have up to 256 accelerator nodes, well aligned with the current and future network scales. In such case, the CAIN header is only 8 bytes. (Note: a none-linear code-to-length mapping table can be specified to provide more flexible address length hierarchy. TBD.)

The routing, forwarding, and other control plane provisions based on CAIN header is described in [I-D.song-ship-edge]. When accelerator nodes in the same scale-up network communicates, they always use the shortest addresses to keep the header overhead minimum. When a packet crosses the level boundary, the router is responsible to augment or prune prefix to or from the addresses in the packet. At any location, the packet only carries the minimum address bits to allow unique source and destination identification. Specifically, if a node sends a packet to another data center, at the data center boundary, the packet will be translated into a standard IPv6 packet

without any information loss. Such a design matches the network architecture well where the header overhead is small when the packet size is small.

### 3.1. CAIN Traffic and Header Overhead

In CAIN fabrics where Ethernet carries both scale-up (load/store memory semantics) and scale-out (RDMA message semantics) traffic, the CAIN header provides significant bandwidth efficiency gains for fine-grained memory access operations.

Load/store operations access data at cache-line granularity (typically 64 bytes). With a standard IPv6 + UDP + BTH (RoCEv2) header stack of 60 bytes, the protocol overhead for a 64-byte payload is approximately 48%. The CAIN header with SAL=1 and DAL=1 (intra-rack scale-up domain) reduces the header to 8 bytes, yielding a protocol overhead of 12.5% -- a reduction factor of approximately 4x.

### 3.2. Hierarchy Mapping to Network Topology

The SHIP hierarchy maps naturally to the physical topology of CAINs:

| SHIP Level | Fabric Tier | Address Length | Typical Scale | Dominant Traffic Type |
|---|---|---|---|---|
| L2 (leaf) | Intra-node scale-up | 1 byte | 8-72 GPUs | LD/ST (memory semantics) |
| L1 (mid) | Intra-pod | 2-3 byte | 100s-1000s | Mixed LD/ST and RDMA |
| L0 (root) | Cross-pod scale-out | 4+ byte | 10K+ GPUs | RDMA (message semantics) |
| External | Internet | 16 byte | global | IPv6 |

This mapping has a desirable property: the traffic type most sensitive to header overhead (LD/ST with small payloads) operates in the lowest hierarchy level where addresses are shortest. As traffic traverses higher levels of the hierarchy, payload sizes increase (RDMA bulk transfers for gradient synchronization), and the relative overhead of longer addresses diminishes.

The following table illustrates the total header size for representative deployment scenarios. The baseline for comparison is the 40-byte IPv6 fixed header.

| Scenario | SAL | DAL | Raw (B) | Padded (B) | Savings vs IPv6 |
|---|---|---|---|---|---|
| Intra-rack LD/ST | 1 | 1 | 8 | 8 | 80% |
| Intra-pod | 2 | 2 | 10 | 12 | 70% |
| Cross-pod | 3 | 3 | 12 | 12 | 70% |
| Cross-cluster | 4 | 4 | 14 | 16 | 60% |
| Edge-to-IPv6 (SA=4) | 4 | 0 | 26 | 28 | 30% |
| Full IPv6 (both) | 0 | 0 | 38 | 40 | 0% |

## 4. Implementation Considerations

CAIN header-based packet forwarding needs new functions on L3 switches. The cost analysis is given in appendix A which shows that the hardware cost is low, the throughput and latency performance is on par with the traditional L3 switch, and the benefit is high. Specifically, the power and memory efficiency is even better than the conventional L3 switch due to the simplified table lookups.

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. Security Considerations

TBD

## 7. References

### 7.1. 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>.

### 7.2. Informative References

[I-D.herbert-sunh]
Herbert, T., "Scale-Up Network Header (SUNH)", Work in Progress, Internet-Draft, draft-herbert-sunh-01, 16 January 2026, <https://datatracker.ietf.org/doc/html/draft-herbert-sunh-01>.

[I-D.song-ship-edge]
Song, H., "Short Hierarchical IP Addresses for Edge Networks", Work in Progress, Internet-Draft, draft-song-ship-edge-05, 13 April 2023, <https://datatracker.ietf.org/doc/html/draft-song-ship-edge-05>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/info/rfc8724>.

[hot25] Joshi et al., R., "Your network doesn’t end at the NIC: A case for unifying the inter-host and intra-host networks in (AI) datacenters", 24th ACM Workshop on Hot Topics in Networks, 2025, <https://dl.acm.org/doi/epdf/10.1145/3772356.3772415>.

[ub] Liao et al., H., "UB-Mesh: A Hierarchically Localized nD-FullMesh Data Center Network Architecture", IEEE Micro, 2025, <https://www.computer.org/csdl/magazine/mi/2025/05/11150738/29JWPYIYbIc>.

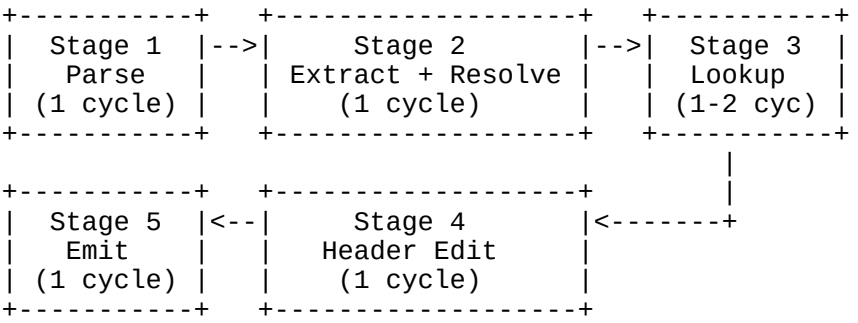
[afh] Broadcom, "Scale-Up Ethernet Framework Specification", 2025, <https://docs.broadcom.com/doc/scale-up-ethernet-framework>.

[gaudi] Intel, "Intel Gaudi 3 AI Accelerator White Paper", 2025, <https://www.intel.com/content/www/us/en/content-details/817486/intel-gaudi-3-ai-accelerator-white-paper.html>.

## Appendix A. Appendix A. Hardware Cost Analysis

### A.1. LGR Hardware Processing Pipeline

This appendix describes a reference hardware pipeline architecture for a level-gateway switch (i.e., LGR in [I-D.song-ship-edge]) processing the CAIN header. The pipeline achieves line-rate forwarding with address augmentation and pruning in 5-6 clock cycles, comparable to standard IPv6 L3 switch pipelines.




Total: 5-6 cycles at 1 GHz core clock = 5-6 ns latency

### A.2. Comparison with Standard IPv6 Pipeline

The following table compares the SHIP LGR pipeline with a standard IPv6 L3 switch pipeline across key implementation parameters.

| Parameter | Standard IPv6 L3 Switch | SHIP LGR (4B-aligned) |
|---|---|---|
| Parse stages | 1 cycle | 1 cycle |
| Direction/classify | 1 cycle | 1 cycle |
| Forwarding lookup | 1-2 cycles | 1-2 cycles |
| Header edit | 1 cycle | 1 cycle |
| Emit | 1 cycle | 1 cycle |
| Total pipeline depth | 5-6 cycles | 5-6 cycles |
| Lookup key width | 128-bit (fixed) | 8-128 bit (var) |
| Lookup engine | TCAM (LPM) | SRAM (hash) |
| Lookup power (relative) | ~10x | ~1x |

The SHIP LGR pipeline is the same as the standard IPv6 pipeline. The forwarding lookup is substantially more power-efficient because it uses SRAM-based hash tables instead of TCAM-based Longest Prefix Matching. In the most common intra-level forwarding case (SAL ==

DAL), the lookup key is only 1-4 bytes rather than the full 128-bit IPv6 address, further reducing hash computation cost and SRAM access energy.

### A.3. Latency Considerations

The 5-6 ns LGR pipeline latency is within the same order of magnitude as current Ethernet switch ASICs. For intra-level forwarding (the common case for LD/ST traffic), no address modification is performed, and the pipeline reduces to a simple hash-lookup-and-forward path.

LGR address augmentation and pruning add no additional latency beyond the base pipeline, as these operations execute within the existing header edit stage. The latency impact is felt only at hierarchy boundaries (LGR hops), which coincide with the topology boundaries where additional switch hops would exist regardless of the addressing scheme.

## Authors' Addresses

Haoyu Song (editor)
Futurewei Technologies
United States of America
Email: hsong@futurewei.com

Keyi Zhu
Huawei Technologies
China
Email: zhukeyi@huawei.com

Jian Song
China Mobile
China
Email: songjianyjy@chinamobile.com