Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 May 2026

T. Herbert
XDPnet

4 November 2025

Scale-Up Network Header (SUNH)
draft-herbert-sunh-00

Abstract

This document specifies a network header that is a type of compressed IPv4 or IPv6 header. The use case is high performance networking in limited domains, in particular in scale-up networks for AI where even modest packet overhead packet may be materially detrimental to overall performance. The SUNH network header provides three primary benefits: 1) It reduces the amount of on-the-wire protocol overhead, 2) It simplifies route lookup to be more efficient with lower latency, 3) It retains the properties and benefits of a Network Layer header including compatibility with deployed Ethernet switches and NICs.

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 8 May 2026.

Table of Contents

1. Introduction

1.1. Problem statement

AI and Machine Learning are pushing networking requirements to stratospheric levels of performance. This is especially true in so called "scale-up networks for AI" which consist of clusters of tightly coupled GPUs or generically XPUs. The performance requirements include multi-terabits of throughput and latencies measured in the tens of nanoseconds. Some traffic patterns may have a majority of small packets, like for KVcache in AI, where packet

sizes may commonly be 256 bytes or less. For traditional network layers, including IPv4 and IPv6, network headers are appreciable protocol overheard and can reduce throughput in high performance scenarios. For instance, the twenty byte header of IPv4 represents about 8% overhead in a 256 byte packet, and the forty bytes of IPv6 header would be about 16% overhead.

In addition to the problems of per packet overhead, there is also the cost of looking up IP addresses for routing or matching addresses in table lookups. Long addresses require CAMs, TCAMs, or tree mechanisms that increase latency and power consumption in routers.

1.2. Prior work: Scale-Up Ethernet approach

Scale-Up Ethernet, or SUE [SUE], is a proposal for a new Ethernet protocol for scale-up networks. SUE addresses the network layer overheads by redefining the Ethernet header based on the Structured Local Address Plan in IEEE 802.c-2017 [IEEE_802c-2017]. SUE defines two "AI Header Formats" for the Ethernet header: AHF Gen 1 and AHF Gen 2. These formats turn the Ethernet header into a quasi network layer header. AFH Gen 1 uses the standard Ethernet header format, however switching is performed only on the low order sixteen bits of the Ethernet addresses. AF Gen 1 also allows a four byte shim header to follow the Ethernet header that contains pertinent network layer fields like Hop Limit and Traffic Class. AFH Gen 2 repurposes the entire Ethernet header to contain sixteen bit addresses or thirty-two bit "XPU addresses" as well as Network Layer fields.

The fundamental problem with the AI Header Formats in the SUE approach is that they’re not compatible with deployed switches. In order to properly forward AFH packets, switches must be updated specifically with new functionality. The modifications are invasive since switches need to fundamentally reinterpret the Ethernet header.

In addition to the creating incompatibility with legacy switches, AFH headers are also incompatible with deployed NICs. When a NIC receives a packet it attempts to match the destination MAC address as being local. Typically, this is done by an exact match CAM lookup on the forty-eight bit Destination MAC address. For AFH, only a subset of bits in the forty-eight bit Destination address are valid so the normal CAM lookup will fail. The only work around would be to place the NIC in Promiscuous mode and then match the XPU Destination address in software.

1.3. Introducing Scale-Up Network Header

This document specifies the Scale-Up Network Header, or SUNH that is a compressed network header for IPv4 or IPv6. It's a "scale-up protocol" since intended use cases are limited domains, such as scale-up networks for AI, where reducing packet overhead and protocol header complexity benefits performance.

The primary difference between SUE's AFH format and SUNH is that SUNH is a proper Network Layer header. Other than a new EtherType, SUNH is transparent to the Ethernet Header. This means that SUNH is compatible with existing deployed Ethernet switches. Layer 2 switching "just works" with SUNH since only the Ethernet addresses are considered and they're not modified. Layer 3 switching would work the same way that switches can switch IP packets based on IP addresses. A Layer 3 switch could identify SUNH packets by EtherType, parse the SUNH header, perform route lookups on the SUNH destination address, and then forward the packet per Hop Limit, Traffic Class, and Flow Label fields in the SUNH header.

The SUNH header may be used in lieu of an IPv4 or IPv6 header. SUNH is an eight byte header so there is a substantial savings in protocol overhead compared to IPv4 or IPv6. Additionally, SUNH addresses are sixteen bits in size that affords the use of very efficient route lookup techniques.

1.4. 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. Design and Requirements

We define the Scale-Up Ethernet Header, or SUNH, as a compressed IP header. SUNH is an eight octet header. The source and destination addresses are each sixteen bits (two octets). The other four octets constitute a Next Header number, Hop Limit, Traffic Class, and a Flow Label. Note that a compressed IPv4 or IPv6 header share the same format in SUNH. Any interpretation of a SUNH header being a compressed IPv4 or IPv6 header is from domain local configuration and cannot be inferred from the SUNH header.

2.1. SUNH domains and EtherType

SUNH only works within a limited domain [RFC8799] called a "SUNH domain". Within the SUNH domain a "SUNH prefix" is defined. This is the implied IPv4 or IPv6 prefix for host unicast addresses in the SUNH format. A SUNH address is then the sixteen bit host part of a fully qualified IPv4 or IPv6 address. If the SUNH prefix is IPv4 then the prefix is sixteen bits or more, and for IPv6 the prefix is 112 bits or more. All participating nodes in a SUNH domain are configured with the common SUNH prefix and whether the SUNH header is a compressed IPv4 or IPv6 header. A SUNH can be expanded to an full IPv4 or IPv6 address by appending the SUNH prefix to the SUNH address.

SUNH is distinguished by its own EtherType when encapsulated in Ethernet. For other Layer 2 technologies, an equivalent identifier may indicate SUNH is encapsulated. Note that SUNH does not have an IP version number, any interpretation of a SUNH header as being a compressed IPv4 header or IPv6 header is local to nodes in a SUNH domain. Additionally, the SUNH network might be not even be interpreted as a compressed IP header, and instead it may be treated as a separate network protocol. As long as all nodes in the SUNH limited domain are agreement, such the interpretation of SUNH is consistent within the limited domain.

2.2. Switching and Routing

Other than defining a new EtherType, SUNH has no impact on the link layer. SUNH packets may be Layer 2 switched by commodity switches since Layer 2 switching is based on solely Ethernet addresses such that the EtherType and Ethernet payload are transparent to the forwarding operation.

A SUNH packet may be routed or Layer 3 switched. In this case the router would be modified to understand and route SUNH packets. The routing decision can be based entirely on the sixteen bit Destination Address, and there are no multicast or broadcast addresses.

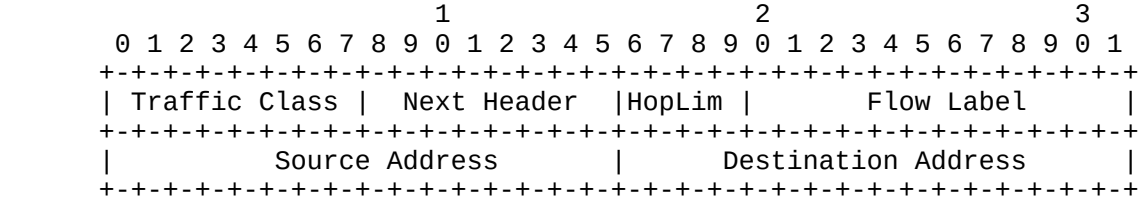
Because SUNH addresses are sixteen-bits, in a limited domain SUNH addresses could be assigned from a small range. This potentially simplifies route lookup on addresses. For instance, if all addresses were assigned from a ten bit host part, that is assigned SUNH addresses are in the range 0 to 1023, then a route lookup can be implemented with a simple array where the address is used to index the array. This array could easily be in SRAM, so that route lookups degenerate to very fast indexed memory loads.

2.3. SUNH Domains and VLANs

Multiple SUNH domains may be defined in a network using VLANs. Each VLAN could be associated with its own SUNH domain. A host may have SUNH addresses in different domains that are associated with different VLANs and Layer 3 switches would have forwarding tables for each VLAN that defines a SUNH domain.

3. SUNH Format

The SUNH header format is shown below, Note that definitions of the non-address fields are consistent with the analogous fields in the IPv6 header [RFC8200].



Fields:

Traffic Class
: 8-bit Traffic Class field equivalent to the IPv6 Traffic Class field and has the same semantics. The current use of the Traffic Class field for Differentiated Services and Explicit Congestion Notification is specified in [RFC2474] and [RFC3168].

Next Header
: 8-bit selector. Identifies the type of header immediately following the SUNH header. Uses the same values as the IPv4 Protocol field [IANA-PN].

HopLim
: 4-bit integer Hop Limit field. This is equivalent to the IPv6 Hop Limit field except that the SUNH Hop Limit is four bits. The field value is decremented by 1 by each node that forwards the packet. When forwarding, the packet is discarded if Hop Limit was zero when received or is decremented to zero. A node that is the destination of a packet SHOULD NOT discard a packet with Hop Limit equal to zero; it SHOULD process the packet normally.

Flow Label
: 12-bit flow label. This is a smaller version of the IPv6 Flow Label and is otherwise set and processed with the same semantics.

Source Address
: 16-bit SUNH address of the originator of the packet. Normally this is the host part of the uncompressed source IP address (IPv4 or IPv6).

Destination Address
: 16-bit SUNH destination address of the intended recipient. Normally this is the host part of the uncompressed destination IP address (IPv4 or IPv6).

Note that the SUNH header does not have a Total Length field or Payload Length field as IPv4 and IPv6 have. The total length of the packet is determined from the length of the Layer 2 frame. That is the last byte in a SUNH packet coincides with the last byte in an Ethernet packet for example.

4. Host requirements

The requirements for host with regard to SUNH are described below.

4.1. SUNH domains

A sender MUST only send a packet with an SUNH header if the SUNH domain prefix is properly configured and the destination is within the domain. The SUNH source and destination addresses MUST be host unicast addresses (not multicast or broadcast).

Nodes within a SUNH MAY use SUNH addresses as canonical endpoints. In this case SUNH addresses might not correspond to IP addresses, and all communications would be done only using SUNH as the network layer protocol. If this interpretation is valid within a SUNH domain then all nodes MUST be configured with the same interpretation.

4.2. Encapsulation in SUNH

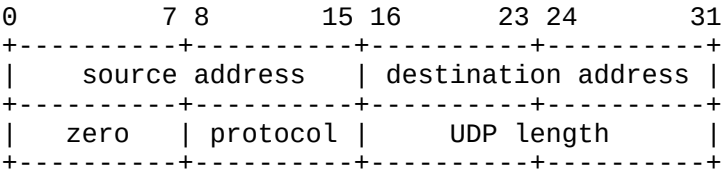
The SUNH Next Protocol MAY used with any IP protocol that is independent of the IP address format in the network header of the packet. In other cases, the protocols may be used with adaption or prohibited as described below.

A SUNH packet is constructed from an Ethernet header (or other Layer 2 header), followed by a SUNH header, followed by an IP protocol (e.g. TCP or UDP). The EtherType in the Ethernet header MUST be set to the EtherType for SUNH (or the equivalent next protocol identifier in another Layer 2 protocol is set to indicate SUNH).

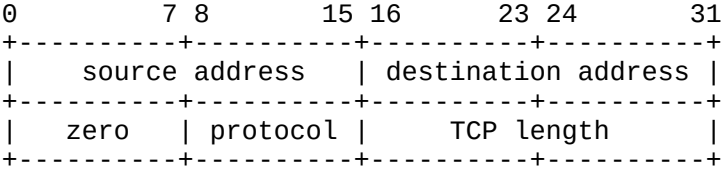
4.3. TCP and UDP in SUNH

A SUNH header MAY encapsulate TCP or UDP. The pseudo checksum for the TCP or UDP checksum includes the IPv4 or IPv6 addresses. When TCP or UDP is encapsulated in SUNH a pseudo header with SUNH addresses MUST be used as shown below.

The pseudo header for the UDP checksum with SUNH is:



The pseudo header for the TCP checksum with SUNH is:



Note that the TCP and UDP pseudo headers for SUNH are fewer bytes than that for IPv4 or IPv6, this does provide a minor performance benefit in calculating the pseudo header checksum.

4.4. IPv6 extension headers and SUNH

If the protocol of an uncompressed SUNH header is IPv6 then IPv6 extension headers MAY be used as the Next Protocol in a SUNH header. An exception is that the Routing Header MUST NOT be used since routers are not expected to have SUNH addresses and even if they did then the IPv6 addresses in the segment list of a Routing Header are not compatible to set as the destination address in the SUNH. In practice, not beng able to use Routing Headers with SUNH is no great loss, the major point of SUNH is to reduce protocol header overhead and the overhead of even a single Routing Header would nullify most of the benefits of SUNH.

4.5. Minimum packet length

The minimum Ethernet frame size is sixty-four bytes which translates to a minimum Ethernet payload size of forty-six bytes. Since SUNH does not have a total length or payload length like IPv4 and IPv6 headers do, care must be taken to ensure the minimum packet size is maintained.

There are three ways to ensure Ethernet frames have a proper minimum length. These are described below.

4.5.1. SUNH payload is greater than thirty-eight bytes

If the SUNH payload is greater than or equal to thirty-eight bytes in length then the minimum frame length requirement is automatically satisfied.

For this method a sender would just need to check that the SUNH payload is greater than or equal to thirty-eight bytes. If the payload length is greater than or equal to thirty-eight bytes then no further action needs to be taken and the packet can be transmitted.

4.5.2. The SUNH payload has self-describing length

If the SUNH payload is less than thirty-eight bytes in size but the length of the SUNH payload can be unambiguously deduced from protocol headers in the payload, then the minimum length requirement is satisfied by padding the packet up to the minimum sixty-four byte frame length.

An example of this case is a UDP packet encapsulated in SUNH (Next Protocol is 17). The UDP header contains a UDP Length field that unambiguously gives the length of the SUNH payload. For instance, if a UDP packet with minimum length of eight is encapsulated in a SUNH header then the total length of the Ethernet payload is sixteen bytes. The packet is sent by padding it with thirty bytes (zeroes after the UDP header).

4.5.3. Explicit padding

If the SUNH payload is less than thirty-eight bytes in size and the the length cannot be deduced from the protocol headers in the SUNH payload, then explicit padding can be used. The idea is to insert a Destination Option Extension Header after the SUNH header that contains a single PADN option [RFC8200].

An example of this case is TCP that unlike UDP does not have a payload length field in its header. For instance, if a pure ACK were sent without any TCP options that would be a twenty byte header with no TCP payload. When encapsulated in SUNH the length of the Ethernet payload is twenty-eight bytes which is eighteen bytes short of the minimum length. Zero bytes cannot be padded to the packet since they would be interpreted as bytes of TCP payload. In order to handle this case, explicit padding is inserted into the packet in the form of an IPv6 Destination option.

The Destination option is set in an Destination Options extension header with one PADN option. The number of bytes needed in the extension header is rounded up to be divisible by eight:

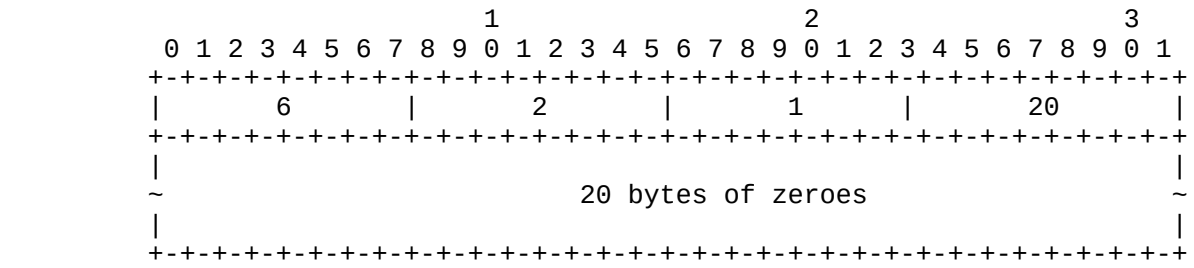
```
#bytes = 8 * ((bytes_needed + 7) / 8)
```

The extension header length and PADN option length are then set as:

```
Extension header length = (#bytes / 8) - 1

PADN option length = #bytes - 4
```

For our example, the Destination Options extension header for padding a TCP packet with no TCP options and no TCP payload in SUNH header would be:



The next protocol field in the SUNH would be set to 60 to indicate a Destination Options header. The Next Header field in the Destination Options header is set to 6 to indicate TCP.

Note that a SUNH packet may already have extension headers. In this case the Destination Options extension header can be inserted as another extension header after the SUNH header, unless a Hop-by-Hop Options is present then the Destination option must be inserted after the Hop-by-Hop options header.

4.6. Receiver processing

A configured host receiver SHOULD accept SUNH packets. Packets received with a SUNH EtherType are processed by the network layer. If the Destination Address matches the local SUNH per the applicable SUNH domain then the recipient is the intended receiver. If a properly formatted SUNH packet is received to the local address, the Next Protocol is used to parse and process the next protocol.

A receiver has two options for how to treat SUNH addresses internally:

* It may append the SUNH prefix and then perform internal processing as though the packet had been received to the uncompressed address (e.g. the TCP connection table uses plain IP addresses in connection tuples)

* The host MAY treat the packet as having different network layer state where protocol tuples include the SUNH addresses. The latter case might be appropriate for a scale-up protocol with packets that are only ever sent in SUNH.

The choice of option is specific to each SUNH domain.

4.7. ICMP errors

Hosts MAY send ICMP errors in response to a SUNH packet. The ICMP message MUST be encapsulated in SUNH with the source and destination addresses swapped from the original packet. The ICMP message format, ICMPv4 or ICMPv6, would correspond to the uncompressed SUNH protocol format (IPv4 or IPv6).

4.8. Flow hash

SUNH affords a very efficient mechanism to produce a flow hash for a packet that includes the addresses and Flow Label as input. The procedures are (assuming sixty-four bit registers):

1. Load the eight byte SUN header into a register

2. "Or" the Traffic Class field with all ones

3. Run a hash function over the register

The result of this process is a hash over the addresses, Flow Label, Next Header, and Hop Limit. The Next Header is reasonable input since it’s expected that a host would normally send all packets with for a Transport Layer flow with the same Next Header. The Hop Limit

is reasonable to include since we normally expect that packets of the same flow will have the same Hop Limit when received by some node on the network (although the Hop Limit seen for a flow could differ between nodes but that’s not a problem).

5. Router requirements

A router MAY support routing (or Layer 3 switching) of SUNH packets. The basic idea is that the router performs route lookup on the SUNH destination address to get next hop information and then forwards the packet accordingly.

A router forwards a SUNH packet solely base on the contents of the Ethernet and SUNH headers. The router SHOULD NOT parse the SUNH payload. If a router is performing ECMP routing then it SHOULD use the SUNH Flow Label for flow entropy input, and it SHOULD NOT parse beyond the SUNH header to extract information from the transport layer headers (i.e. a router should not use transport port numbers as flow entropy input).

A router processes the Hop Limit field in expected fashion. When a router receives a SUNH packet, the Hop Limit is checked. If the Hop Limit is zero or one then the packet is discarded and an "ICMP Time Exceeded" error MAY be sent in response. If the router forwards the packet then the non-zero hop limit is decremented.

6. IANA Considerations

There are no actions required for IANA defined in this document.

7. Security Considerations

SUNH is expressly a limited domain protocol meaning that packets with the SUNH protocol should be constrained to the networks of the SUNH domain. This can be accomplished via a firewall that blocks packets with the SUNH EtherType from crossing the domain boundary.

As a network layer protocol SUNH does not introduce any new security concerns. It would have similar concerns to those of IPv4 or IPv6.

8. References

8.1. 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>.

8.2. Informative References

[IANA-PN] "Assigned Internet Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

[IEEE_802c-2017] "IEEE 802c-2017: IEEE Standard for Local and Metropolitan Area Networks:Overview and Architecture--Amendment 2: Local Medium Access Control (MAC) Address Usage", June 2017, <https://standards.ieee.org/ieee/802c/6890/>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

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

Author’s Address

Tom Herbert
XDPnet
Los Gatos, CA,
United States of America
Email: tom@herbertland.com