Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 July 2026

T. Herbert
XDPnet

16 January 2026

Scale-Up Network Header (SUNH)
draft-herbert-sunh-01

Abstract

This document specifies the Scale-Up Network Header that is a concise and efficient network layer header. The primary use case is high performance networking in limited domains, and in particular in scale-up networks for AI where even modest packet overhead packet may be detrimental to overall performance.

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 20 July 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 exhibit a majority of small packets, like for KVcache in AI, where packet sizes may commonly be 256 bytes or less. For traditional network layer protocols, including IPv4 and IPv6, network headers represent 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 a table lookup. Long addresses require CAMs, TCAMs, or tree mechanisms that increase latency and power consumption in routers.

1.2. 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 as well as Network Layer fields.

The fundamental problem with the AI Header Formats in the SUE approach is that it’s not compatible with commodity 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 bits of the Destination address are valid so the normal CAM lookup will fail. The 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 concise and efficient network layer protocol header. SUNH defines a common four byte SUNH header prefix that is followed by a source and destination address of some size. It's a "scale-up protocol" since an intended use case is in scale-up networks for AI, however the protocol is generic and may have broad use cases in datacenter networks.

The SUNH network header provides four primary benefits:

* It reduces the amount of on-the-wire protocol overhead

* It simplifies route lookup to be more efficient with lower latency

* It retains the properties and benefits of a Network Layer header including compatibility with deployed Ethernet switches

* It obviates the need to use a UDP shim header before the encapsulated transport protocol. An encapsulated transport protocol could be directly encapsulated in SUNH using it's own protocol number and the UDP header is not needed for compatibility with switches.

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 as only the Ethernet addresses are considered when switching and they're not modified by SUNH. 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 header, perform route lookups on the SUNH destination address, and then forward the packet. The SUNH header includes Hop Limit, Traffic Class, and Flow Label fields that can be used with the same semantics as for routing IP packets.

We can compare the header overhead of SUNH to IPv4 and IPv6, and can compare SUNH to cases where a shim UDP header is used:

Protocol header	# bytes
SUNH with 8-bit addresses	6 bytes
SUNH with 16-bit bit addresses	8 bytes
SUNH with 24-bit bit addresses	10 bytes
SUNH with 32-bit bit addresses	12 bytes
IPV4	20 bytes
IPV4 + UDP header	28 bytes
IPV6	40 bytes
IPV6 + UDP header	48 bytes

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

A SUNH header is comprised of a four byte common prefix followed by a source and destination address. The common prefix contains a Next Protocol number, Hop Limit, Traffic Class, and Flow Label. The source and destination addresses can have sizes of eight, sixteen, twenty-four, or thirty-two bits (one, two, three, or four bytes).

The addresses may be omitted to create an "address-less" network header. ENdpoint addresses would be inferred from Layer 2 addresses (e.g. Ethernet addresses).

2.1. SUNH domains

SUNH only works within a limited domain called a "SUNH domain". THe addresses of SUNH are effectively private addresses that are not routable on the Internet.

2.2. SUNH EtherType

SUNH is distinguished by its own EtherType when encapsulated in Ethernet. Different EtherTypes may indicate the size of the SUNH addresses in packets, or one size may be configured in the SUNH domain. For other Layer 2 technologies, an equivalent identifier may indicate SUNH is encapsulated. Using an EtherType specific to SUNH facilitated filtering of SUNH at boundaries of a limited domain.

2.3. Interpretation of a SUNH header

Within a limited domain a SUNH header may be interpreted as a the header for a standalone network layer protocol, or as a compressed IPv4 or IPv6 header. Note that any interpretation of SUNH is local to the SUNH domain. There are no fields in the SUNH header that indicate which interpretation is applicable in the domain. It is expected that all the nodes in the domain are configured with the same interpretation.

2.3.1. SUNH as a standalone network layer header

SUNH header may be interpreted as its own network layer protocol in a limited domain. Effectively, SUNH would be a variant of IP with addresses of some length. Host stacks could be adapted to support SUNH as another network layer protocol.

2.3.2. SUNH as a compressed IP header

The SUNH header may be interpreted as a compressed IPv4 or IPv6 header. In this case, a "SUNH prefix" is specified for a domain. The prefix serves as the implied network prefix for SUNH addresses. A SUNH address is thus the host part (i.e. low order bits) of a fully qualified IPv4 or IPv6 address. THe SUNH prefix is either for IPv4 or IPv6. A SUNH address can be expanded to an fully qualified IPv4 or IPv6 address by prepending the SUNH prefix to the SUNH address. A SUNH header can be expanded to an IPv4 or IPv6 header by expanding the SUNH addresses and copying the common fields from the SUNH header to the IPv4 or IPv6 header.

All participating hosts in a SUNH domain are configured with the common SUNH prefix and whether SUNH headers sent in the domain are compressed IPv4 or IPv6 headers. 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.

2.4. 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 SUNH Destination Address, and there are no multicast or broadcast addresses.

This specification allows four sizes of SUNH addresses. This allows use an address size that scales well for the domain. For instance, there were at most 1000 nodes in some network then 16-bit SUNH addresses could be used and the nodes could be assigned addresses in in the range 0 to 1023. 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.

The size of SUNH addresses in a packet can either be inferred from the EtherTYpe or from configuration in the SUNH domain. In the case of using EtherTypes, different EtherType would be defined for different size SUNH addresses. In the case of configuration, all the nodes in the SUNH would assume that size of SUNH addresses in the domain is the configured value.

2.5. SUNH Domains and VLANs

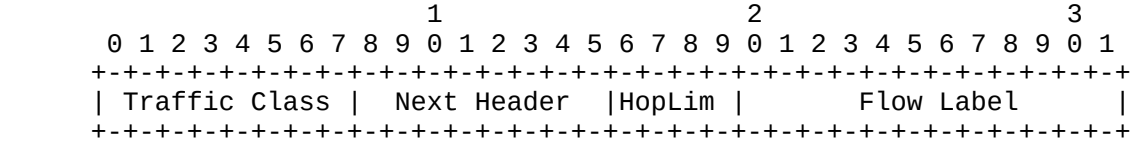
Multiple SUNH domains may be defined in a network using VLANs. Each VLAN could be associated with its own SUNH domain. A host may have SUNH addresses in different domains that are associated with VLANs and Layer 3 switches would have forwarding tables for each VLAN that defines a SUNH domain.

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].

3.1. Common prefix

The common SUNH prefix is shown below:



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. SUNH defines its own number space that is distinct from IP Protocol numbers [IANA-PN], however there may be some overlap for convenience.

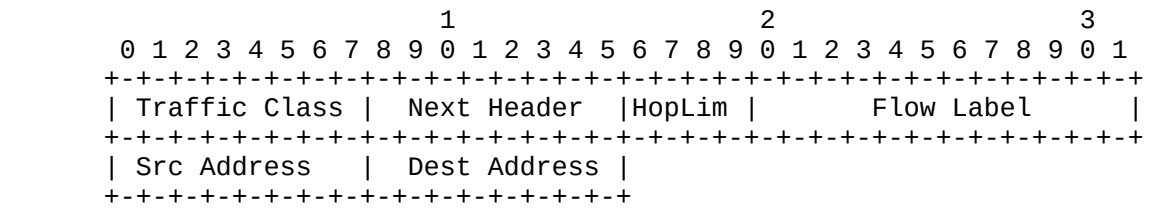
HopLim
: 4-bit integer Hop Limit field. This is equivalent to the IPv6 Hop Limit field except that the SUNH Hop Limit is four bits adn the handling is slightly differ. The field value is decremented by 1 by each node that forwards the packet. When forwarding, the packet is discarded if Hop Limit was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

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

Note that the SUNH header does not have a Total Length field or Payload Length field as IPv4 and IPv6 has. 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. Since the SUNH header does not have a length, care must be taken to ensure that the real length of a packet can be unambiguously interpreted especially with regards to meeting the minimum Ethernet packet size requirement (see Section 4.4).

3.2. SUNH with 8-bit addresses

The format of the SUNH header with 8-bit address is show below



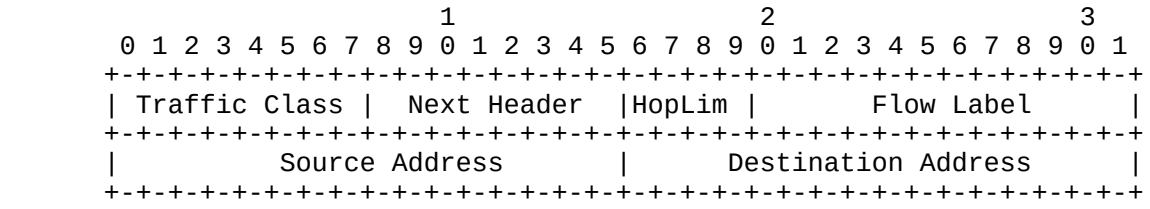
Addresses:

Source Address
8-bit SUNH address of the originator of the packet. This maybe the host part of a fully qualified IPv4 (24-bit prefix) or IPv6 address (120-bit prefix).

Destination Address
8-bit SUNH address of the intended recipient. This may be the host part of a fully qualified IPv4 (24-bit prefix) or IPv6 address (120-bit prefix).

3.3. SUNH with 16-bit addresses

The format of the SUNH header with 16-bit addresses is show below



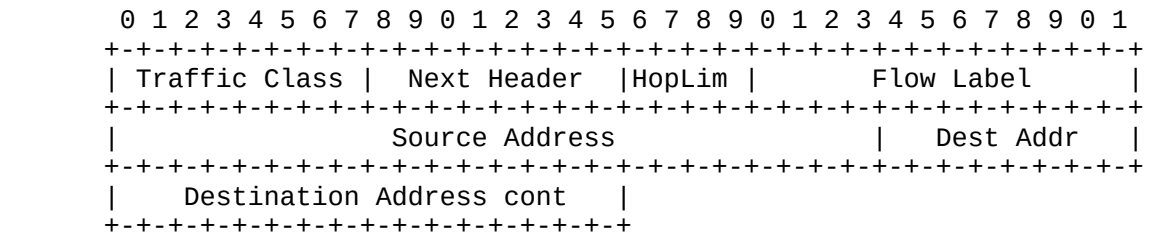
Addresses:

Source Address
16-bit SUNH address of the originator of the packet. This may be the host part of a fully qualified IPv4 (16-bit prefix) or IPv6 address (112-bit prefix).

Destination Address
16-bit SUNH address of the intended recipient. This maybe the host part of a fully qualified IPv4 (16-bit prefix) or IPv6 address (112-bit prefix).

3.4. SUNH with 24-bit addresses

The format of the SUNH header with 24-bit addresses is show below



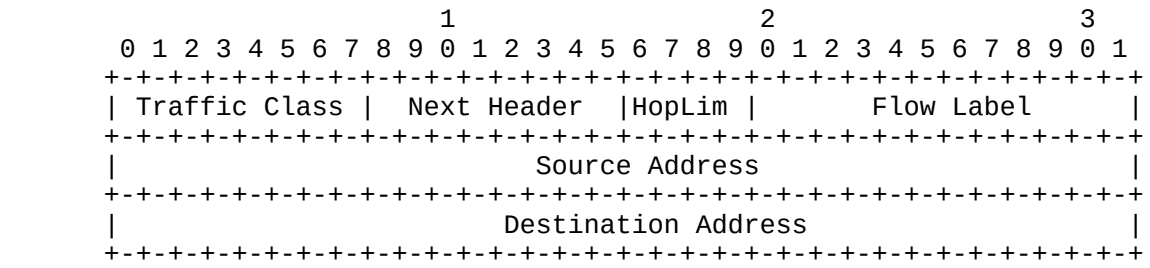
Addresses:

Source Address
: 24-bit SUNH address of the originator of the packet. This may be the host part of a fully qualified IPv4 (8-bit prefix) or IPv6 address (104-bit prefix).

Destination Address
: 24-bit SUNH address of the intended recipient (note that the Destination Address is split over two rows in the diagram). This may be the host part of a fully qualified IPv4 (16-bit prefix) or IPv6 address (104-bit prefix).

3.5. SUNH with 32-bit addresses

The format of the SUNH header with 32-bit addresses is show below



Addresses:

Source Address
: 32-bit SUNH address of the originator of the packet. This may be a fully qualified IPv4 address or the host part of an IPv6 address (96-bit prefix).

Destination Address
: 32-bit SUNH address of the intended recipient of the packet. This may be a fully qualified IPv4 address or the host part of an IPv6 address (96-bit prefix).

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 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

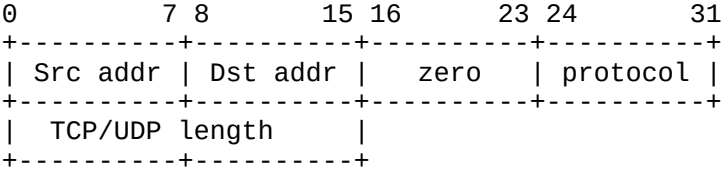
The SUNH Next Protocol MAY used with any IP protocol that is independent of the IP address format in the network header of the packet. In other cases the protocols may be used with adaption or prohibited as described below.

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).

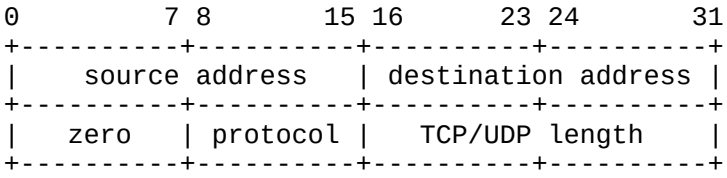
4.3. TCP UDP in SUNH

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.

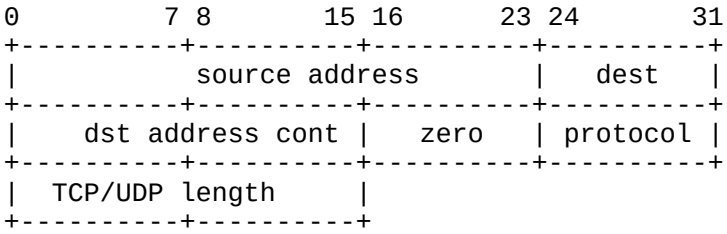
The pseudo header with 8-bit addresses is:



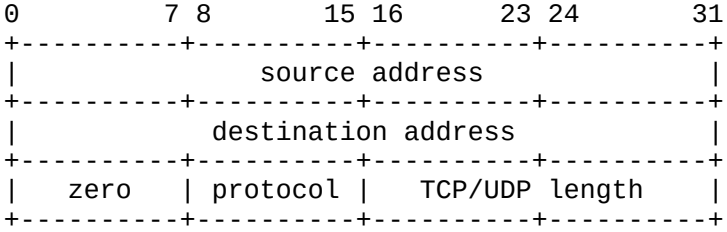
The pseudo header with 16-bit addresses is:



The pseudo header with 24-bit addresses is:



The pseudo header with 32-bit addresses 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. 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 and the real length of a packet can be unambiguously determined.

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

4.4.1. SUNH header and SUNH payload is greater than forty-six bytes

If the length of the SUNH header including addresses plus the length of SUNH payload is greater than or equal to forty-six 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 header plus the payload length is greater than or equal to forty-six bytes.

4.4.2. The SUNH payload has self-describing length

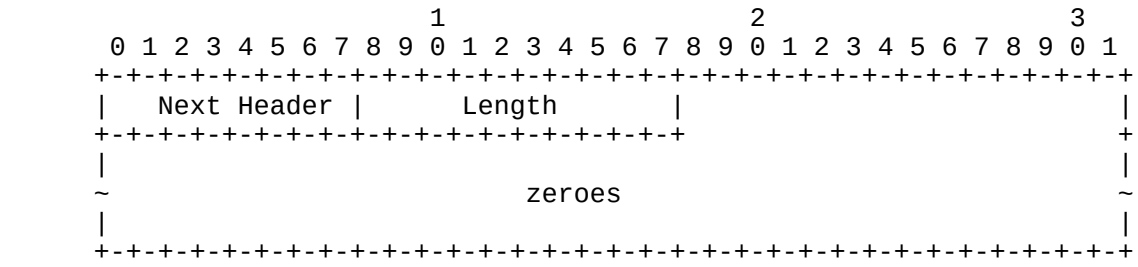
If the SUNH payload has self-describing length then the minimum length requirement is satisfied by padding the packet with zeroes up to be at least sixty-four bytes in size.

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 SUNW payload. For instance, if a UDP packet with minimum length of eight is encapsulated in a SUNH header with sixteen bit addresses then the total length of the Ethernet payload is sixteen bytes and the packet is sent by padding it with thirty bytes (zeroes after the UDP header).

4.4.3. Explicit padding

If the SUNH header plus payload is less than forty-six bytes length the payload length cannot be deduced from the protocol headers in the SUNH payload, then explicit padding can be used. The idea is to insert a "SUNH Padding Header" after the SUNH header.

The format of the padding header is:



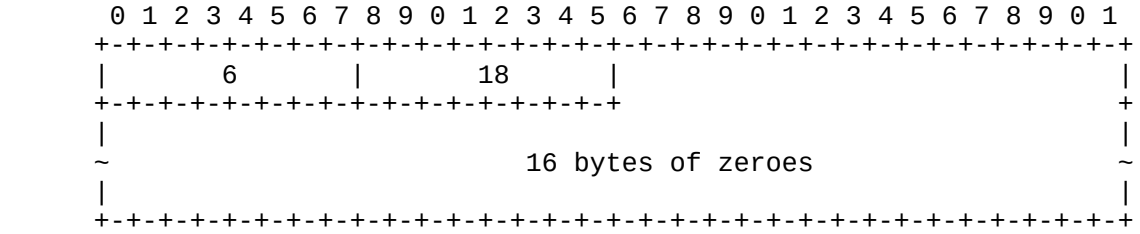
Fields:

Next Header
: The same semantics as Next Header in the SUNH header (see Section 3.1).

Length
Length of the padding. This includes two bytes for the Next Header and Length and the number of following zeroes.

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 with sixteen bit addresses 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 using the SUNH Padding Header with Length of eighteen.

For our example, the SUNH Padding Header would be:



The next protocol field in the SUNH would be set to 252 to indicate the Padding Header follows the SUNH header. The Next Header field in the Padding Header is set to 6 to indicate TCP.

4.5. 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 into 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 standalone network protocol 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 is specific to each SUNH domain.

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 the router is performing ECMP routing 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 forward the packet then the non-zero hop limit is decremented.

6. IANA Considerations

IANA is requested to create a new registry named "Assigned SUNH Protocol Numbers". These are 8-bit numbers. The initial assignments are listed below:

Decimal	Keyword	Protocol	Reference
6	TCP	Transmission control	RFC9293
17	UDP	User datagram	RFC768
196-250		User defined	[This document]
252	SUNHPAD	SUNH padding and testing	[This document]
253		Experimentation and testing	RFC3692
254		Experimentation and testing	RFC3692
255	Reserved		RFC3692

7. Security Considerations

SUNH is a new network protocol explicitly designed for use in limited domains. Border routers of limited domains can discard packets with SUMH headers based on their EtherType. Within the limited domain it is expected that nodes are trusted. Otherwise, SUNH does not introduce any new security concerns.

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>.

[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