

Network Working Group
Internet-Draft
Updates: [4944](#) (if approved)
Intended status: Standards Track
Expires: February 24, 2011

J. Hui, Ed.
Arch Rock Corporation
P. Thubert
Cisco
August 23, 2010

Compression Format for IPv6 Datagrams in 6LoWPAN Networks
draft-ietf-6lowpan-hc-09

Abstract

This document specifies an IPv6 header compression format for IPv6 packet delivery in 6LoWPAN networks. The compression format relies on shared context to allow compression of arbitrary prefixes. How the information is maintained in that shared context is out of scope. This document specifies compression of multicast addresses and a framework for compressing next headers. UDP header compression is specified within this framework.

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 <http://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 February 24, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	4
2.	Specific Updates to RFC 4944	4
3.	IPv6 Header Compression	5
3.1.	LOWPAN_IPHC Encoding Format	6
3.1.1.	Base Format	6
3.1.2.	Context Identifier Extension	9
3.2.	IPv6 Header Encoding	9
3.2.1.	Traffic Class and Flow Label Compression	10
3.2.2.	Deriving IIDs from the Encapsulating Header	11
3.2.3.	Stateless Multicast Address Compression	12
3.2.4.	Stateful Multicast Address Compression	13
4.	IPv6 Next Header Compression	13
4.1.	LOWPAN_NHC Format	14
4.2.	IPv6 Extension Header Compression	14
4.3.	UDP Header Compression	16
4.3.1.	Compressing UDP ports	16
4.3.2.	Compressing UDP checksum	16
4.3.3.	UDP LOWPAN_NHC Format	17
5.	IANA Considerations	18
6.	Security Considerations	19
7.	Acknowledgements	19
8.	Changes	19
9.	References	21
9.1.	Normative References	21
9.2.	Informative References	22
	Authors' Addresses	22

1. Introduction

The [IEEE 802.15.4] standard specifies an MTU of 127 bytes, yielding about 80 octets of actual MAC payload with security enabled, on a wireless link with a link throughput of 250 kbps or less. The 6LoWPAN adaptation format [RFC4944] was specified to carry IPv6 datagrams over such constrained links, taking into account limited bandwidth, memory, or energy resources that are expected in applications such as wireless sensor networks. [RFC4944] defines a Mesh Addressing header to support sub-IP forwarding, a Fragmentation header to support the IPv6 minimum MTU requirement [RFC2460], and stateless header compression for IPv6 datagrams (LOWPAN_HC1 and LOWPAN_HC2) to reduce the relatively large IPv6 and UDP headers down to (in the best case) several bytes.

LOWPAN_HC1 and LOWPAN_HC2 are insufficient for most practical uses of 6LoWPAN networks. LOWPAN_HC1 is most effective for link-local unicast communication, where IPv6 addresses carry the link-local prefix and an Interface Identifier (IID) directly derived from IEEE 802.15.4 addresses. In this case, both addresses may be completely elided. However, though link-local addresses are commonly used for local protocol interactions such as IPv6 ND [RFC4861], DHCPv6 [RFC3315] or routing protocols, they are usually not used for application-layer data traffic, so the actual value of this compression mechanism is limited.

Routable addresses must be used when communicating with devices external to the LoWPAN or in a route-over configuration where IP forwarding occurs within the LoWPAN. For routable addresses, LOWPAN_HC1 requires both IPv6 source and destination addresses to carry the prefix in-line. In cases where the Mesh Addressing header is not used, the IID of a routable address must be carried in-line. However, LOWPAN_HC1 requires 64-bits for the IID when carried in-line and cannot be shortened even when it is derived from the IEEE 802.15.4 16-bit short address. When the destination is an IPv6 multicast address, LOWPAN_HC1 requires the full 128-bit address to be carried in-line.

As a result, this document defines an encoding format, LOWPAN_IPHC, for effective compression of Unique Local, Global, and multicast IPv6 Addresses based on shared state within contexts. In addition, this document also introduces a number of additional improvements over the header compression format defined in [RFC4944].

LOWPAN_IPHC allows for compression of some commonly-used IPv6 Hop Limit values. If the LoWPAN is a mesh-under stub, a Hop Limit of 1 for inbound and a default value such as 64 for outbound are usually enough for application layer data traffic. Additionally, a hop-limit

value of 255 is often used to verify that a communication occurs over a single-hop. This specification enables compression of the IPv6 Hop Limit field in those common cases, whereas LOWPAN_HC1 does not.

This document also defines LOWPAN_NHC, an encoding format for arbitrary next headers. LOWPAN_IPHC indicates whether the following header is encoded using LOWPAN_NHC. If so, the bits immediately following the compressed IPv6 header start the LOWPAN_NHC encoding. In contrast, LOWPAN_HC1 could be extended to support compression of next headers using LOWPAN_HC2, but only for UDP, TCP, and ICMPv6. Furthermore, the LOWPAN_HC2 octet sits between the LOWPAN_HC1 octet and uncompressed IPv6 header fields. This specification moves the next header encoding bits to follow all IPv6-related bits, allowing for a properly layered structure and direct support for IPv6 extension headers.

Using LOWPAN_NHC, this document defines a compression mechanism for UDP. While [\[RFC4944\]](#) defines a compression mechanism for UDP, that mechanism does not enable checksum compression when rendered possible by additional upper layer mechanisms such as upper layer Message Integrity Check (MIC). This specification adds the capability to elide the UDP checksum over the LoWPAN, which enables saving of a further two octets.

Also using LOWPAN_NHC, this document defines encoding formats for IPv6-in-IPv6 encapsulation as well as IPv6 Extension Headers. With LOWPAN_HC1 and LOWPAN_HC2, chains of next headers cannot be encoded efficiently.

[1.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [\[RFC2119\]](#).

[2.](#) Specific Updates to [RFC 4944](#)

This document specifies a header compression format that is intended to replace that defined in [Section 10 of \[RFC4944\]](#). Implementation of [Section 10 of \[RFC4944\]](#) is now NOT RECOMMENDED. New implementations MAY implement compression according to [Section 10 of \[RFC4944\]](#), but SHOULD NOT send packets compressed according to [Section 10 of \[RFC4944\]](#).

A compliant implementation of [\[RFC4944\]](#) as updated by this document MUST be able to properly process a packet received that makes use of the provisions of this document. A compliant implementation MAY

implement additional LOWPAN_NHC types ([Section 4](#)) that may be registered ([Section 5](#)) in the future. It is out of scope of this document how a compressor learns that a decompressor has additional capabilities.

[Section 5.3 of \[RFC4944\]](#) also defines how to fragment compressed IPv6 datagrams that do not fit within a single link frame. [Section 5.3 of \[RFC4944\]](#) defines the fragment header's datagram_size and datagram_offset values as the size and offset of the IPv6 datagram before compression. As a result, all fragment payload outside the first fragment must carry their respective portions of the IPv6 datagram before compression. This document does not change that requirement. When using the fragmentation mechanism described in [Section 5.3 of \[RFC4944\]](#), any header that cannot fit within the first fragment MUST NOT be compressed.

The header compression format defined in this document preempts the ESC dispatch value defined in [Section 5.1 of \[RFC4944\]](#). Instead, the value of 01 000000 is reserved as a replacement value for ESC, to be finally assigned with the first assignment of extension bytes.

3. IPv6 Header Compression

In this section, we define the LOWPAN_IPHC encoding format for compressing the IPv6 header. To enable effective compression LOWPAN_IPHC relies on information pertaining to the entire 6LoWPAN network. LOWPAN_IPHC assumes the following will be the common case for 6LoWPAN communication: Version is 6; Traffic Class and Flow Label are both zero; Payload Length can be inferred from lower layers from either the 6LoWPAN Fragmentation header or the IEEE 802.15.4 header; Hop Limit will be set to a well-known value by the source; addresses assigned to 6LoWPAN interfaces will be formed using the link-local prefix or a small set of routable prefixes assigned to the entire 6LoWPAN network; addresses assigned to 6LoWPAN interfaces are formed with an IID derived directly from either the 64-bit extended or 16-bit short IEEE 802.15.4 addresses.

```
+-----+-----+
| Dispatch + LOWPAN_IPHC (2-3 octets) | In-line IPv6 Header Fields
+-----+-----+
```

Figure 1: LOWPAN_IPHC Header

The LOWPAN_IPHC encoding utilizes 13 bits, 5 of which are taken from the rightmost bit of the dispatch type. The encoding may be extended

by another octet to support additional contexts. Any information from the uncompressed IPv6 header fields carried in-line follow the LOWPAN_IPHC encoding, as shown in Figure 1. In the best case, the LOWPAN_IPHC can compress the IPv6 header down to two octets (the dispatch octet and the LOWPAN_IPHC encoding) with link-local communication.

When routing over multiple IP hops, LOWPAN_IPHC can compress the IPv6 header down to 7 octets (1-octet dispatch, 1-octet LOWPAN_IPHC, 1-octet Hop Limit, 2-octet Source Address, and 2-octet Destination Address). The Hop Limit may not be compressed because it needs to be decremented at each hop and may take any value. Stateful address compression must be applied to the source and destination IPv6 addresses because they do not statelessly match the source and destination link layer addresses on intermediate hops.

3.1. LOWPAN_IPHC Encoding Format

This section specifies the format of the LOWPAN_IPHC encoding that describes how an IPv6 header is compressed. The encoding can be 2 octets long for the base encoding or 3 octets long when an additional context encoding is present. The IPv6 header fields that are not fully elided are placed immediately after the LOWPAN_IPHC, either in a compressed form if the field is partially elided, or literally.

3.1.1. Base Format

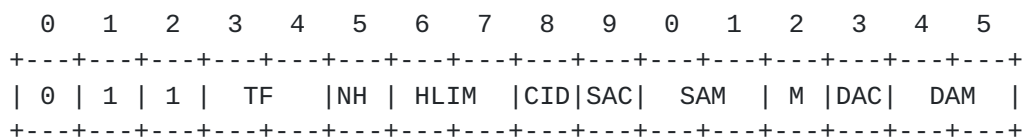


Figure 2: LOWPAN_IPHC base Encoding

TF: Traffic Class, Flow Label:

- 00: ECN + DSCP + 4-bit Pad + Flow Label (4 bytes)
- 01: ECN + 2-bit Pad + Flow Label (3 bytes), DSCP is elided
- 10: ECN + DSCP (1 byte), Flow Label is elided
- 11: Traffic Class and Flow Label are elided.

NH: Next Header:

- 0: Full 8 bits for Next Header are carried in-line.

- 1: The Next Header field is compressed and the next header is encoded using LOWPAN_NHC, which is discussed in [Section 4](#).

HLIM: Hop Limit:

- 00: The Hop Limit field is carried in-line.
- 01: The Hop Limit field is compressed and the hop limit is 1.
- 10: The Hop Limit field is compressed and the hop limit is 64.
- 11: The Hop Limit field is compressed and the hop limit is 255.

CID: Context Identifier Extension:

- 0: No additional 8-bit Context Identifier Extension is used. If context-based compression is specified in either SAC or DAC, context 0 is used.
- 1: An additional 8-bit Context Identifier Extension field immediately follows the DAM field.

SAC: Source Address Compression

- 0: Source address compression uses stateless compression.
- 1: Source address compression uses stateful, context-based compression.

SAM: Source Address Mode:

If SAC=0:

- 00: 128 bits. The full address is carried in-line.
- 01: 64 bits. The first 64-bits of the address are elided. The value of those bits is the link-local prefix padded with zeros. The remaining 64 bits are carried in-line.
- 10: 16 bits. The first 112 bits of the address are elided. The value of those bits is the link-local prefix padded with zeros. The remaining 16 bits are carried in-line.
- 11: 0 bits. The address is fully elided. The first 64 bits of the address are the link-local prefix padded with zeros. The remaining 64 bits are computed from the encapsulating header (e.g. 802.15.4 or IPv6 source address) as specified in [Section 3.2.2](#).

If SAC=1:

- 00: The UNSPECIFIED address, ::
- 01: 64 bits. The address is derived using context information and the 64 bits carried in-line.
- 10: 16 bits. The address is derived using context information and the 16 bits carried in-line.
- 11: 0 bits. The address is fully elided. The prefix is derived using context information. Any of the remaining 64 bits not covered by the context information are computed from the encapsulating header (e.g. 802.15.4 or IPv6 source address) as specified in [Section 3.2.2](#).

M: Multicast Compression

0: Destination address is not a multicast address.

1: Destination address is a multicast address.

DAC: Destination Address Compression

0: Destination address compression uses stateless compression.

1: Destination address compression uses stateful, context-based compression.

DAM: Destination Address Mode:

If M=0 and DAC=0 This case matches SAC=0 but for the destination address:

00: 128 bits. The full address is carried in-line.

01: 64 bits. The first 64-bits of the address are elided.

The value of those bits is the link-local prefix padded with zeros. The remaining 64 bits are carried in-line.

10: 16 bits. The first 112 bits of the address are elided.

The value of those bits is the link-local prefix padded with zeros. The remaining 16 bits are carried in-line.

11: 0 bits. The address is fully elided. The first 64 bits of the address are the link-local prefix padded with zeros. The remaining 64 bits are computed from the encapsulating header (e.g. 802.15.4 or IPv6 destination address) as specified in [Section 3.2.2](#).

If M=0 and DAC=1:

00: Reserved.

01: 64 bits. The address is derived using context information and the 64 bits carried in-line.

10: 16 bits. The address is derived using context information and the 16 bits carried in-line.

11: 0 bits. The address is fully elided. The prefix is derived using context information. Any of the remaining 64 bits not covered by the context information are computed from the encapsulating header (e.g. 802.15.4 or IPv6 destination address) as specified in [Section 3.2.2](#).

If M=1 and DAC=0:

00: 128 bits. The full address is carried in-line.

01: 48 bits. The address takes the form FFXX::00XX:XXXX:XXXX.

10: 32 bits. The address takes the form FFXX::00XX:XXXX.

11: 8 bits. The address takes the form FF02::00XX.

If M=1 and DAC=1:

00: 48 bits. This format is designed to match Unicast-Prefix-based IPv6 Multicast Addresses as defined in [[RFC3306](#)] and [[RFC3956](#)]. The multicast address takes the form FFXX:XXLL:PPPP:PPPP:PPPP:PPPP:XXXX:XXXX. where the X are the nibbles that are carried in-line, in the order in which they appear in this format. P denotes nibbles used to encode the prefix

itself. L denotes nibbles used to encode the prefix length. The prefix information P and L is taken from the specified context.

01: reserved
10: reserved
11: reserved

3.1.2. Context Identifier Extension

This specification expects that a conceptual context is shared between the node that compresses a packet and the node(s) that need to expand it. How the contexts are shared and maintained is out of scope. What information is contained within a context information is out of scope. Actions in response to unknown and/or invalid contexts are out of scope. The specification enables a node to use up to 16 contexts. The context used to encode the source address does not have to be the same as the context used to encode the destination address.

If the CID field is set to '1' in the LOWPAN_IPHC encoding, then an additional octet extends the LOWPAN_IPHC encoding following the DAM bits but before the IPv6 header fields that are carried in-line. The additional octet identifies the pair of contexts to be used when the IPv6 source and/or destination address is compressed. The context identifier is 4 bits for each address, supporting up to 16 contexts. Context 0 is the default context. The encoding is shown in Figure 3.

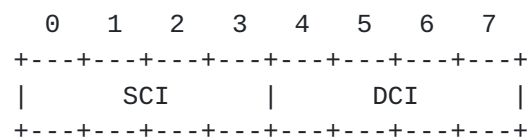


Figure 3: LOWPAN_IPHC Encoding

SCI: Source Context Identifier Identifies the prefix that is used when the IPv6 source address is statefully compressed.

DCI: Destination Context Identifier Identifies the prefix that is used when the IPv6 destination address is statefully compressed.

3.2. IPv6 Header Encoding

Fields carried in-line (in part or in whole) appear in the same order as they do in the IPv6 header format [[RFC2460](#)]. The Version field is always elided. Unicast IPv6 addresses may be compressed to 64 or 16 bits or completely elided. Multicast IPv6 addresses may be compressed to 8, 32, or 48 bits. The IPv6 Payload Length field MUST

TF = 01: Flow Label carried in-line.


```

  0 1 2 3 4 5 6 7
+-+--+--+--+--+--+
|ECN|  DSCP  |
+-+--+--+--+--+--+

```

TF = 10: Traffic Class carried in-line.

3.2.2. Deriving IIDs from the Encapsulating Header

LOWPAN_IPHC elides the IIDs of source or destination addresses when SAM = 3 or DAM = 3, respectively. In this mode, the IID is derived from the encapsulating header. When the encapsulating header carries IPv6 addresses, bits for the source and destination addresses are copied verbatim from the source and destination addresses of the encapsulating IPv6 header.

The remainder of this section defines the mapping from IEEE 802.15.4 link-layer addresses to IIDs for both short and extended IEEE 802.15.4 addresses. IID bits not covered by the context information MAY be elided if they match the link-layer address mapping and MUST NOT be elided if they do not.

An extended IEEE 802.15.4 address takes the form of an IEEE EUI-64 address. Generating an IID from an extended address is identical to that defined in [Appendix A of \[RFC4291\]](#). The only change needed to transform an IEEE EUI-64 identifier to an interface identifier is to invert the universal/local bit.

A short IEEE 802.15.4 address is 16 bits in length. Short addresses are mapped into the restricted space of IEEE EUI-64 addresses by setting the middle 16 bits to 0xffff, the bottom 16 bits to the short address, and all other bits to zero. As a result, an IID generated from a short address has the form:

```
0000:00ff:fe00:XXXX
```

where XXXX carries the short address. The universal/local bit is zero to indicate local scope.

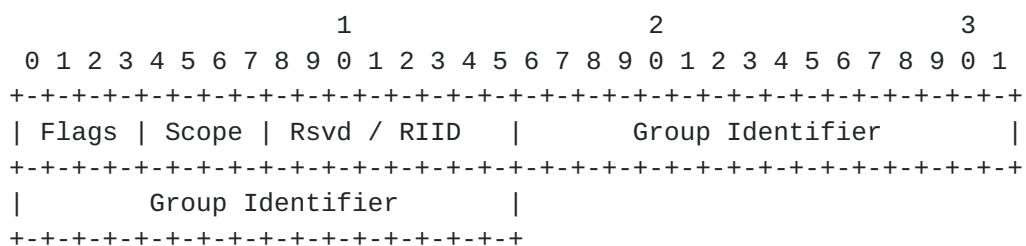
This mapping for non-EUI-64 identifiers differs from that presented in [Appendix A of \[RFC4291\]](#). Using the restricted space ensures no overlap with IIDs generated from unrestricted IEEE EUI-64 addresses. Also, including 0xffff in the middle of the IID helps avoid overlap with other locally managed IIDs.

This mapping from a short IEEE 802.15.4 address to 64-bit IIDs is also used to reconstruct any part of an IID not covered by context

DAM = 11. 8-bit Compressed Multicast Address (FF02::gg).

3.2.4. Stateful Multicast Address Compression

LOWPAN_IPHC supports stateful compression of multicast addresses when $M = 1$ and $DAC = 1$. This document currently defines $DAM = 00$: context-based compression of Unicast-Prefix-based IPv6 Multicast Addresses [RFC3306][RFC3956]. In particular, the Prefix Length and Network Prefix can be taken from a context. As a result, LOWPAN_IPHC can compress a Unicast-Prefix-based IPv6 Multicast Address down to 6 octets by only carrying the 4-bit Flags, 4-bit Scope, 8-bit RIID, and 32-bit Group Identifier in-line.



$DAM = 01$. Unicast-Prefix-based IPv6 Multicast Address Compression

Note that the Reserved field MUST carry the reserved bits from the multicast address format as described in [RFC3306]. When a Rendezvous Point is encoded in the multicast address as described in [RFC3956], the Reserved field carries the RIID bits in-line.

4. IPv6 Next Header Compression

LOWPAN_IPHC elides the IPv6 Next Header field when the NH bit is set to 1. This also indicates the use of 6LoWPAN next header compression, LOWPAN_NHC. The value of IPv6 Next Header is recovered from the first bits in the LOWPAN_NHC encoding. The following bits are specific to the IPv6 Next Header value. Figure 4 shows the structure of an IPv6 datagram compressed using LOWPAN_IPHC and LOWPAN_NHC.

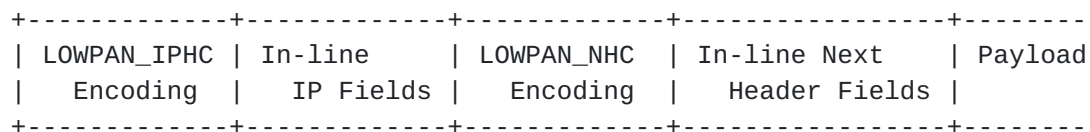


Figure 4: Typical LOWPAN_IPHC/LOWPAN_NHC Header Configuration

4.1. LOWPAN_NHC Format

Compression formats for different next headers are identified by a variable-length bit-pattern immediately following the LOWPAN_IPHC compressed header. When defining a next header compression format, the number of bits used SHOULD be determined by the perceived frequency of using that format. However, the number of bits and any remaining encoding bits SHOULD respect octet alignment. The following bits are specific to the next header compression format. This document defines a compression format for IPv6 Extension and UDP headers.

```
+-----+-----+
| var-len NHC ID | compressed next header...
+-----+-----+
```

Figure 5: LOWPAN_NHC Encoding

4.2. IPv6 Extension Header Compression

A necessary property of encoding headers using LOWPAN_NHC is that the immediately preceding header must either be encoded using LOWPAN_IPHC or LOWPAN_NHC. In other words, all headers encoded using the 6LoWPAN encoding format defined in this document must be contiguous. As a result, this document defines a set of LOWPAN_NHC encodings for selected IPv6 Extension Headers such that the UDP Header Compression defined in [Section 4.3](#) may be used in the presence of those extension headers.

The LOWPAN_NHC encodings for IPv6 Extension Headers are composed of a single LOWPAN_NHC octet followed by the IPv6 Extension Header. The format of the LOWPAN_NHC octet is shown in Figure 6. The first 7 bits serve as an identifier for the IPv6 Extension Header immediately following the LOWPAN_NHC octet. The remaining bit indicates whether or not the following header utilizes LOWPAN_NHC encoding.

```

0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+
| 1 | 1 | 1 | 0 |   EID   |NH|
+---+---+---+---+---+---+---+
```

Figure 6: IPv6 Extension Header Encoding

EID: IPv6 Extension Header ID:

- 0: IPv6 Hop-by-Hop Options Header[RFC2460]
- 1: IPv6 Routing Header[RFC2460]
- 2: IPv6 Fragment Header[RFC2460]
- 3: IPv6 Destination Options Header[RFC2460]
- 4: IPv6 Mobility Header [[RFC3775](#)]
- 5: Reserved
- 6: Reserved
- 7: IPv6 Header

NH: Next Header:

- 0: Full 8 bits for Next Header are carried in-line.
- 1: The Next Header field is elided and the next header is encoded using LOWPAN_NHC, which is discussed in [Section 4](#).

For the most part, the IPv6 Extension Header is carried verbatim in the bytes immediately following the LOWPAN_NHC octet, with two important exceptions: Length Field and Next Header Field.

The Next Header Field contained in IPv6 Extension Headers is elided when the NH bit is set in the LOWPAN_NHC encoding octet. Note that doing so allows LOWPAN_NHC to utilize no more overhead than the non-encoded IPv6 Extension Header.

The Length Field contained in a compressed IPv6 Extension Header indicates the number of octets that pertain to the (compressed) extension header following the Length Field. Note that this changes the Length Field definition in [[RFC2460](#)] from indicating the header size in 8-octet units, not including the first 8 octets. Changing the Length Field to be in units of octets removes wasteful internal fragmentation.

IPv6 Hop-by-Hop and Destination Options Headers may use a trailing Pad1 or PadN to achieve 8-octet alignment. When there is a single trailing Pad1 or PadN option of 7 octets or less and the containing header is a multiple of 8 octets, the trailing Pad1 or PadN option MAY be elided by the compressor. A decompressor MUST ensure that the containing header is padded out to a multiple of 8 octets in length, using a Pad1 or PadN option if necessary. Note that Pad1 and PadN options that appear in locations other than the end MUST be carried in-line as they are used to align subsequent options.

Note that specifying units in octets means that LOWPAN_NHC MUST NOT be used to encode IPv6 Extension Headers that have more than 255 octets following the Length Field after compression.

When the identified next header is an IPv6 Header (EID=7), the NH bit of the LOWPAN_NHC encoding is unused and MUST be set to zero. The

following bytes MUST be encoded using LOWPAN_IPHC as defined in [Section 3](#).

4.3. UDP Header Compression

This document defines a compression format for UDP headers using LOWPAN_NHC. The UDP compression format is shown in Figure 7. Bits 0 through 4 represent the NHC ID and '11110' indicates the specific UDP header compression encoding defined in this section.

4.3.1. Compressing UDP ports

This specification introduces a range of well-known ports (0xF0Bx) that can be compressed to 4 bits. Considering that this represents only 16 contiguous ports, it can be expected that many incompatible applications will use the same port numbers for their own end-to-end needs.

The overloading of the 0xF0Bx ports increases the risk of getting the wrong type of payload and misinterpreting the content compared to ports that are reserved at IANA. As a result, it is recommended that the use of those ports be associated with a mechanism such as a Transport Layer Security (TLS) Message Integrity Check (MIC) that validates that the content is expected and checked for integrity.

4.3.2. Compressing UDP checksum

The UDP checksum operation is mandatory with IPv6 [[RFC2460](#)] for all packets. For that reason [[RFC4944](#)] disallows the compression of the UDP checksum.

With this specification, a compressor in the source transport endpoint MAY elide the UDP checksum if it is authorized by the Upper Layer. The compressor SHOULD NOT set the C bit unless it has received such authorization. The Upper Layer SHOULD only provide the authorization in the following cases:

Tunneling: In this case, 6LoWPAN is deployed as a wireless pseudo-fieldbus by tunneling existing field protocols over UDP. If the tunneled PDU possesses its own addressing, security and integrity check, the tunneling mechanism MAY authorize to elide the UDP checksum in order to save on the encapsulation overhead.

Upper Layer Message Integrity Check: In this case, there is some other form of integrity check in the UDP payload that covers at least the same information as the UDP checksum (pseudo-header, data) and has at least the same strength.

A forwarding node MAY imply authorization from an incoming packet if

the C bit is set. A forwarding node that cannot unambiguously derive such authorization SHOULD NOT elide the UDP checksum when performing 6LoWPAN compression. The forwarding node that expands a 6LoWPAN packet with the C bit on MUST compute the UDP checksum on behalf of the source node and place that checksum in the restored UDP header as specified in the incumbent standards [[RFC0768](#)], [[RFC2460](#)].

If a 6LoWPAN termination is also the transport endpoint and it receives a compressed packet with the C bit set, then it is entitled to ignore the UDP checksum process completely. If the C bit is not set, the packet might have been forwarded by an edge router, so this is not an indication that the MIC is not present. If the terminating node knows that the message integrity will be validated by the upper layer by some state associated to the Service Access Point, it is entitled to ignore the checksum operation as if the C bit was set.

[4.3.3](#). UDP LOWPAN_NHC Format

```

      0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| 1 | 1 | 1 | 1 | 0 | C |   P   |
+---+---+---+---+---+---+---+

```

Figure 7: UDP Header Encoding

C: Checksum:

- 0: All 16 bits of Checksum are carried in-line.
- 1: All 16 bits of Checksum are elided. The Checksum is recovered by recomputing it on the 6LoWPAN termination point.

P: Ports:

- 00: All 16 bits for both Source Port and Destination Port are carried in-line.
- 01: All 16 bits for Source Port are carried in-line. First 8 bits of Destination Port is 0xF0 and elided. The remaining 8 bits of Destination Port are carried in-line.
- 10: First 8 bits of Source Port are 0xF0 and elided. The remaining 8 bits of Source Port are carried in-line. All 16 bits for Destination Port are carried in-line.
- 11: First 12 bits of both Source Port and Destination Port are 0xF0B and elided. The remaining 4 bits for each are carried in-line.

Fields carried in-line (in part or in whole) appear in the same order as they do in the UDP header format [[RFC0768](#)]. The UDP Length field MUST always be elided and is inferred from lower layers using the

6LoWPAN Fragmentation header or the IEEE 802.15.4 header.

5. IANA Considerations

This document defines a new IPv6 header compression format for 6LoWPAN networks. The document allocates the following 32 Dispatch type field values for LOWPAN_IPHC:

```
01 100000
    through
01 111111
```

This assignment preempts the assignment of 01 111111 for ESC [[RFC4944](#)], which is possible as no extension bytes have been allocated yet that would enable the use of ESC. Instead, the value:

```
01 000000
```

is reserved as a replacement value for ESC, to be finally assigned with the first assignment of extension bytes.

This document also creates a new IANA registry for the LOWPAN_NHC header type, with the following initial content:

```
00000000 to 11011111: (unassigned)
1110000N: IPv6 Hop-by-Hop Options Header      [RFCthis]
1110001N: IPv6 Routing Header                  [RFCthis]
1110010N: IPv6 Fragment Header                 [RFCthis]
1110011N: IPv6 Destination Options Header     [RFCthis]
1110100N: IPv6 Mobility Header                 [RFCthis]
1110101N: (Reserved for future extension headers)
1110110N: (Reserved for future extension headers)
1110111N: IPv6 Header                          [RFCthis]
11110CPP: UDP Header                          [RFCthis]
11111000 to 11111110: (unassigned)
11111111: (unassigned, reserved for extensions)
```

Capital letters in bit positions represent class-specific bit assignments. N indicates whether or not additional LOWPAN_NHC encodings follow, as defined in [Section 4.2](#). CPP represents variables specific to UDP header compression defined in [Section 4.3](#).

The policy for this registry [[RFC5226](#)] is IETF Review. In this process, new values SHOULD be assigned in a way that preserves the NHC ID abstraction of [Section 4](#) (i.e., k one-bits followed by one zero-bit identify the general class of NHC, followed by class-specific bit assignments).

6. Security Considerations

The definition of LOWPAN_IPHC permits the compression of header information on communication that could take place in its absence, albeit in a less efficient form. It recognizes that a IEEE 802.15.4 PAN may have associated with it a number of prefixes through shared context. How the shared context is assigned and managed is beyond the scope of this document.

The overloading of the 0xF0Bx ports increases the risk of getting the wrong type of payload and misinterpreting the content compared to ports that reserved at IANA. It is thus recommended that the use of those ports be associated with a mechanism such as a Transport Layer Security (TLS) Message Integrity Check (MIC) that validates that the content is expected and checked for integrity.

7. Acknowledgements

Thanks to Julien Abeille, Robert Assimiti, Dominique Barthel, Carsten Bormann, Robert Cragie, Stephen Dawson-Haggerty, Mathilde Durvy, Erik Nordmark, Christos Polyzois, Shoichi Sakane, Zach Shelby, Tony Viscardi, and Jay Werb for useful design consideration and implementation feedback.

8. Changes

(This section to be removed by the RFC editor.)

Draft 09:

- Indicate that a mechanism to learn decompressor's capabilities to decode additional (future) NHCs is out of scope.
- Clarify how to derive IID bits not covered by the context when only 16 bits are carried inline.
- Clarify the value of the Length field for compressed extension headers.
- Added an IANA registry for LOWPAN_NHC types.

Draft 08:

- Clarified that the lower bits of an IPv6 address may be derived from an IPv6 header, not just an 802.15.4 header. Change text from "derived from link-layer header" to "derived from encapsulating header".

Draft 07:

- Added section on mapping link-layer addresses to IIDs.
- Added text on restricting compressed headers to first fragment when using fragment headers defined in [Section 5.3 of \[RFC4944\]](#).
- Minor editorial edits.

Draft 06:

- Reworked introduction.
- Added section on updates to [\[RFC4944\]](#).
- Fixed description of number of bits used for IPHC encoding.
- Specify M=0 only for non-multicast addresses and M=1 only for multicast addresses.
- Move 128-bit multicast encoding to DAC=0.
- Redefined ESC dispatch value to 01 000000.
- Many detailed edits.

Draft 05:

- Added LOWPAN_NHC encodings for IPv6 Extension Headers.
- Specify use of context 0 when CID is 0.
- Indicate that first 64-bits is link-local prefix padded with zeros when link-local prefix is elided.
- Made prefix-based multicast encoding format more explicit for clarity.
- Changed wording around stateful compression to allow for using the in-line bits as an additional index to identify the compressed address.
- Removed support for compressing unspecified address.
- Full 128-bit addr in-line only in stateless encoding.

Draft 04:

- Fixed typos leftover from the changes in 03.
- Gave more details on UDP checksum compression.
- Clarify that the context information is out of scope.
- Added security concern on 0xF0Bx port overloading.

Draft 03:

- Decoupled meaning of SAM bits from the destination address.
- Have separate bit to indicate multicast address compression.
- More extensive support for multicast address compression, including Unicast-Prefix-based Multicast Addresses.

Draft 02:

- Updated wording with compression mode to clarify that a compression mode does not enforce what kind of destination address is being used. Specifically changed Destination Dependent Field to Compression Mode.

- Specify that the configuration and management of contexts is out of scope of this document.

Draft 01:

- HC back to 1 byte by default by stealing a few bits from the dispatch field.
- Added better support for multicast address compression.
- Fixed alignment for UDP port compression.
- Better support for Traffic Class and Flow Label compression.
- Pascal joined as an author.

9. References

9.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [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](#), December 1998.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", [RFC 4944](#), September 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.

9.2. Informative References

- [IEEE 802.15.4]
IEEE Computer Society, "IEEE Std. 802.15.4-2006",
October 2006.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6
Multicast Addresses", [RFC 3306](#), August 2002.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C.,
and M. Carney, "Dynamic Host Configuration Protocol for
IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous
Point (RP) Address in an IPv6 Multicast Address",
[RFC 3956](#), November 2004.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
"Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#),
September 2007.

Authors' Addresses

Jonathan W. Hui (editor)
Arch Rock Corporation
501 2nd St. Ste. 410
San Francisco, California 94107
USA

Phone: +415 692 0828
Email: jhui@archrock.com

Pascal Thubert
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

