6lo
Internet-Draft
Updates: 6282 (if approved)
Intended status: Standards Track
Expires: April 27, 2015

P. Thubert, Ed.
Cisco
C. Bormann
TZI

October 24, 2014

A compression mechanism for the RPL option
draft-thubert-6lo-rpl-nhc-02

Abstract

This specification defines the RPL Packet Information (RPI) NHC compression, a method to compress RPL Option (RFC6553) information within 6LoWPAN-style ("6lo") adaptation layers. This extends 6LoWPAN Header Compression (RFC6282), saving up to 48 bits in each frame compared to the uncompressed form in RFC 6553.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which is the most constrained resource of all. The other constraints, such as the memory capacity and the duty cycling of the LLN devices, derive from that primary concern. Energy is typically available from batteries that are expected to last for years, or scavenged from the environment in very limited quantities. Any protocol that is intended for use in LLNs must be designed with the primary concern of saving energy as a strict requirement.

Controlling the amount of data transmission is one possible venue to save energy. In a number of LLN standards, the frame size is limited to much smaller values than the IPv6 maximum transmission unit (MTU) of 1280 bytes. In particular, an LLN that relies on the classical Physical Layer (PHY) of IEEE 802.14.5 [IEEE802154] is limited to 127 bytes per frame. The need to compress IPv6 packets over IEEE 802.14.5 led to the 6LoWPAN Header Compression [RFC6282] work (6LoWPAN-HC).

The Routing Protocol for Low Power and Lossy Networks [RFC6550] (RPL) is designed to optimize the routing operations in constrained LLNs.

As part of this optimization, RPL requires the addition of RPL Packet Information (RPI) in every packet, as defined in Section 11.2 of [RFC6550].

```
           ------+---------                            ^
                 |          Internet                   |
                 |                                     | Native IPv6
              +-----+                                  |
              |     | Border Router (RPL Root)    ^    |    ^
              |     |                             |    |    |
              +-----+                             |    |    | IPv6 in
                 |                                |    |    | IPv6 + RPI
           o    o   o    o                        |    |    |
       o o   o  o   o  o  o o   o                 |    |    |
      o  o o  o o    o   o   o  o  o              |    |    |
      o   o    o  o     o  o    o  o  o           |    |    |
     o  o   o  o   o         o   o o              v    v    v
     o          o             o     o
                       LLN
```

Figure 1: IP-in-IP Encapsulation within the LLN

The RPI is used to tag the packet with the RPL Instance ID and other information that RPL requires for its operation within the RPL domain. In particular, the SenderRank, which is the scalar metric computed by an specialized Objective Function such as [RFC6552], indicates the Rank of the sender and is modified at each hop. The SenderRank allows to validate that the packet progresses in the expected direction, either upwards or downwards, along the DODAG.

RPL defines the RPL Option for Carrying RPL Information in Data-Plane Datagrams [RFC6553] to transport the RPI, which is carried in an IPv6 Hop-by-Hop Options Header [RFC2460], typically consuming eight bytes per packet.

6TiSCH [I-D.ietf-6tisch-architecture] specifies the operation of IPv6 over the TimeSlotted Channel Hopping [I-D.ietf-6tisch-tsch] (TSCH) mode of operation of IEEE 802.14.5. The architecture requires the use of both 6LoWPAN HC and RPL over IEEE 802.14.5. Because it inherits the constraints on the frame size from the MAC layer, 6TiSCH cannot afford to spend 8 bytes per packet on the RPI. Hence the requirement for a 6LoWPAN header compression of the RPI.

This specification extends the 6lo adaptation layer framework ([RFC4944], [RFC6282]) to carry the same information in a 6LoWPAN RPL Packet Information (RPI) NHC Next-header compression (NHC) header, usually eliminating the Hop-by-Hop Options Header saving up to six bytes per packet.

2. Terminology

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 [RFC2119].

The Terminology used in this document is consistent with and incorporates that described in `Terminology in Low power And Lossy Networks' [RFC7102] and [RFC6550].

The term "byte" is used in its now customary sense as a synonym for "octet".

3. Updating RFC 6282

This specification proposes a new 6LoWPAN Next Header Compression (NHC) for the RPL option [RFC6553], called RPI_NHC, to be placed in a packet that is compressed per [RFC6282].

It updates [RFC6282] in that the "necessary property of encoding headers using LOWPAN_NHC" becomes that "the immediately preceding header must be encoded using either LOWPAN_IPHC, RPI_NHC or LOWPAN_NHC".

Additionally, the necessary property of encoding headers using RPI_NHC is that the immediately preceding header must be encoded using either LOWPAN_IPHC or LOWPAN_NHC.

(Discuss: Is this really an update of RFC 6282 or a straightforward addition to it?)

4. The RPL Packet Information NHC

[RFC6550], Section 11.2, specifies the RPL Packet Information (RPI) as a set of fields that are to be added to the IP packets for the purpose of Instance Identification, as well as Loop Avoidance and Detection.

[RFC6553] defines an encoding for the RPI as a RPL option located in the IPv6 Hop-by-hop Option Header. The present NHC compression mechanism compresses IPv6 Hop-by-hop Headers that contain only that RPL option.

The fields in the RPI include an 'O', an 'R', and an 'F' bit, an 8-bit RPLInstanceID (with some internal structure), and a 16-bit SenderRank.

This section defines the format of the RPL Packet Information NHC (RPI_NHC) that is used to compress the RPI in 6LoWPAN networks.

4.1. Compressing the RPLInstanceId

RPL Instances are discussed in [RFC6550], Section 5. A number of simple use cases will not require more than one instance, and in such a case, the instance is expected to be the global Instance 0. A global RPLInstanceID is encoded in a RPLInstanceID field as follows:

```
          0 1 2 3 4 5 6 7
         +-+-+-+-+-+-+-+-+
         |0|     ID      |  Global RPLInstanceID in 0..127
         +-+-+-+-+-+-+-+-+
```

Figure 2: RPLInstanceID Field Format for Global Instances

For the particular case of the global Instance 0, the RPLInstanceID field is all zeroes. This specification allows to elide a RPLInstanceID field that is all zeroes, and defines a 'I' flag that, when set, signals that the field is elided.

4.2. Compressing the SenderRank

The SenderRank is the result of the DAGRank operation on the rank of the sender; here the DAGRank operation is defined in [RFC6550], Section 3.5.1, as:

```
DAGRank(rank) = floor(rank/MinHopRankIncrease)
```

If MinHopRankIncrease is set to a multiple of 256, the least significant 8 bits of the SenderRank will be all zeroes; by eliding those, the SenderRank can be compressed into a single byte. This idea is used in [RFC6550] by defining DEFAULT_MIN_HOP_RANK_INCREASE as 256 and in [RFC6552] that defaults MinHopRankIncrease to DEFAULT_MIN_HOP_RANK_INCREASE.

This specification allows to encode the SenderRank as either one or two bytes, and defines a 'K' flag that, when set, signals that a single byte is used.

4.3. The RPI_NHC encoding

[RFC6553] defines an encoding for the RPL information as a RPL Option located in an IPv6 Hop-by-Hop Option Header. The RPI_NHC provides a compressed form for that information and is constructed as follows:

The RPI_NHC is immediately followed by the RPLInstanceID, unless that is elided (I=1), and then the SenderRank, which is either compressed into one byte (K=1) or fully inlined as the whole 2 bytes (K=0). Bits in the RPI_NHC indicate whether the RPLInstanceID is elided and/ or the SenderRank is compressed:

O, R, and F bits: The O, R, and F bits as defined in [RFC6550], Section 11.2.

NH: 1-bit flag. The Next Header (NH) bit is defined in [RFC6282], Section 4.2, and it is set to indicate that the next header is encoded using LOWPAN_NHC

I: 1-bit flag. If it is set, the Instance ID is elided and the RPLInstanceID is the Global RPLInstanceID 0. If it is not set, the byte immediately following the RPI_NHC contains the RPLInstanceID as specified in [RFC6550], Section 5.1.

K: 1-bit flag. If it is set, the SenderRank is compressed into one byte, and the least significant byte is elided. If it is not set, the SenderRank is fully inlined as 2 bytes.

In Figure 3, the RPLInstanceID is the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256 so the least significant byte is all zeroes and can be elided:

```
     0                   1
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | NHC: I=1, K=1 |  SenderRank   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The most compressed RPI_NHC

In Figure 4, the RPLInstanceID is the Global RPLInstanceID 0, but both bytes of the SenderRank are significant so it can not be compressed:

```
     0                   1                   2
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | NHC: I=1, K=0 |      SenderRank               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Eliding the RPLInstanceID

In Figure 5, the RPLInstanceID is not the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256:

```
       0                   1                   2
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | NHC: I=0, K=1 | RPLInstanceID |  SenderRank   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Compressing SenderRank

In Figure 6, the RPLInstanceID is not the Global RPLInstanceID 0, and both bytes of the SenderRank are significant:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | NHC: I=0, K=0 | RPLInstanceID |      SenderRank               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Least compressed form of RPI_NHC

The next sections provide alternatives for format of the RPI_NHC.

4.3.1. The Greedy Approach

The RPI_NHC is constructed as follows:

```
        0   1   2   3   4   5   6   7
      +---+---+---+---+---+---+---+---+
      | 1 | 0 | I | K | O | R | F |NH |
      +---+---+---+---+---+---+---+---+
```

Figure 7: The RPI_NHC, Greedy Version

Depending on the RPLInstanceID and the MinHopRankIncrease, the proposed format thus squeezes the RPL information into 16 to 32 bits, which compares to 64 bits when using a Hop-by-hop option with the RPL option as specified in [RFC6553].

(This is called the "greedy" approach as it consumes 1/4 of the NHC space just for the RPI compression.)

4.3.2. The Conservative Approach

In this approach, the encoding of the RPL Packet Information takes two bytes: one byte to indicate the NH type, and then one byte to signal the compressed information.

The NH type is indicated in an extension to existing LOWPAN_NHC encodings. Section 4.2 of [RFC6553] defines LOWPAN_NHC encodings for IPv6 Extension Headers as in Figure 8:

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

Figure 8: IPv6 Extension Header Encoding

Values 5 and 6 of the IPv6 Extension Header ID (EID) are still reserved. This specification uses EID of 5 to indicate that the next byte is a RPI_NHC. The RPI_NHC is constructed as shown in Figure 9:

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| I | K | O | R | F | reserved  |
+---+---+---+---+---+---+---+---+
```

Figure 9: The RPI_NHC, Conservative Version

The bits 5 to 7 of the RPI_NHC are reserved for future use and MUST be sent as zero.

(This is called the "conservative" approach as it consumes only 1/256 of the NHC space.)

4.3.3. The Efficient Approach

4.3.3.1. The NHC escape mechanism

The NHC space of [RFC6282] is limited to 256 code points. For the case some infrequent bit combinations do not fit into the 256 code points, this specification assigns four code points:

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 0 | 0 | 1 | X | Y |
+---+---+---+---+---+---+---+---+
```

Figure 10: NHC Escape Codes

Each NHC escape code is followed by a further NHC code point. The latter MUST be a code point for which special semantics for a preceding escape code are defined, i.e., an escape code MUST NOT be used in front of an NHC code point that does not define special semantics for this escape code.

An escape code followed by another escape code supplies additional semantics; again, a sequence of such escape codes MUST NOT be used unless the final NHC code following this sequence defines the semantics for the specific sequence.

4.3.3.2. RPI_NHC Encoding

The RPI_NHC provides a compressed form for the RPI and is constructed as follows:

```
          0   1   2   3   4   5   6   7
        +---+---+---+---+---+---+---+---+
        | 1 | 0 | 0 | 0 | O | I | K |NH |
        +---+---+---+---+---+---+---+---+
```

Figure 11: RPI NHC, efficient version

The R and F bits, as defined in [RFC6550], Section 11.2, are represented as follows:

If R=0 and F=0, the NHC code is used as defined above. If either is non-zero, a single escape code with X=R and Y=F is prepended in front of the NHC code. (An escape code with X=0 and Y=0 MUST NOT be used with RPI_NHC. A sequence of two or more escape codes MUST NOT be used with RPI_NHC.)

Depending on the RPLInstanceID and the MinHopRankIncrease, the proposed format thus squeezes the RPI into 16 to 40 bits, which compares to 64 bits when using a Hop-by-hop option with the RPL option as specified in [RFC6553].

(This is called the "efficient" approach as it consumes only 1/16 of the NHC space, but, depending on the frequency of set R or set F flags, is almost as efficient as the greedy approach.)

4.3.3.3. Operation

A 6lo compressor that is about to create either an RFC 6282 IPHC header [RFC6282] or a Frag1 header [RFC4944] and finds a Hop-by-Hop Options header [RFC2460] with an RPL Option [RFC6553] in it, performs the following checks:

1. Does the compression scheme apply? I.e.:

    A. is no sub-tlv present in the RPL Option?

    B. is the RPL Option the only option in the Hop-by-Hop Options header?

2. Does the additional compression for I=1 apply? I.e.: is RPLInstanceID == 0?

3. Does the additional compression for K=1 apply? I.e.: is SenderRank < 256?

4. Is both R=0 and F=0, or do we need an escape code?

If check 1 succeeds, the compressor removes the Hop-by-Hop Options header (replacing the zero-valued next header field in the IPv6 header with the value of the next header field of the Hop-by-Hop Options header), and, depending on the outcome of check 2 and 3, generates an RPI_NHC Header with I and K set from the payload information in the RPL Option. If one or both of R and F are non-zero (check 4), it precedes the first byte in the RPI_NHC header with an escape code with X=R and Y=F. It then continues generating the RFC 6282 IPHC or RFC 4944 Frag1 header, filling in the continuation of the RPL Information header as defined in Section 4.3.3.2.

A 6lo decompressor that encounters a RPL Information header reverses this process, creating a Hop-by-Hop Options header with a single RPL Option carrying the information in the RPL Information header.

5. Security Considerations

The security considerations of [RFC4944], [RFC6282], and [RFC6553] apply.

Using a compressed format as opposed to the full inline RPL option is logically equivalent and does not create an opening for a new threat when compared to [RFC6553].

6. IANA Considerations

(greedy variant:)

This document updates IANA registry for the LOWPAN_NHC defined in [RFC6282] and assigns the previously unassigned:

```
10IKORFN: RPL Information [RFCthis]
```

Capital letters in bit positions represent class-specific bit assignments. IKORF represents variables specific to RPL Information compression defined in Section 4. N indicates whether or not additional LOWPAN_NHC encodings follow, as defined in Section 4.2 of [RFC6553].

(efficient variant:)

This draft requests IANA to assign the following LOWPAN_NHC types in the "IPv6 Low Power Personal Area Network Parameters" registry:

010001XY: Escape X=0/Y=0 to X=1/Y=1 [RFCthis]

1000IOKN: RPL Information [RFCthis]

7. Acknowledgements

The author wishes to thank Laurent Toutain for suggesting this work and and Martin Turon for his constructive contributions. Ralph Droms supplied a number of helpful comments on the -00 draft of [I-D.bormann-6lo-rpl-mesh], which was superseded by the present document, turning into the "efficient approach". The discussion in the 6man and roll working groups also was helpful.

8. References

8.1. Normative References

[IEEE802154]
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

[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.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.

8.2. Informative References

[I-D.bormann-6lo-rpl-mesh]
Bormann, C., "NHC compression for RPL Packet Information", draft-bormann-6lo-rpl-mesh-02 (work in progress), October 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-03 (work in progress), July 2014.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-02 (work in progress), October 2014.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.

Authors' Addresses

Pascal Thubert (editor)
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

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org