Network Working Group C. Gomez
Internet-Draft J. Paradells
Intended status: Standards Track UPC/i2CAT
Expires: September 21, 2016 J. Crowcroft
University of Cambridge
March 20, 2016

Optimized 6LoWPAN Fragmentation Header for LPWAN
draft-gomez-lpwan-fragmentation-header-01

Abstract

LPWAN technologies are characterized by a very limited data unit and/or payload size, often one order of magnitude below the one in IEEE 802.15.4. However, many such technologies do not support layer 2 fragmentation. The 6LoWPAN fragmentation header defined in RFC 4944 represents very high overhead for LPWAN technologies, and it even does not support transporting IPv6 datagrams that require fragmentation over layer 2 technologies of a payload size below 13 bytes. This specification defines an optimized 6LoWPAN fragmentation header for LPWAN.

Table of Contents

1. Introduction

Low Power Wide Area Network (LPWAN) technologies are characterized, among others, by a very reduced data unit and/or payload size [I-D.minaburo-lp-wan-gap-analysis]. However, many such technologies do not support layer two fragmentation, therefore the only option for these to support IPv6 (and, in particular, its MTU requirement of 1280 bytes [RFC2460]) is the use of a fragmentation header at the adaptation layer below IPv6, such as the 6LoWPAN fragmentation header defined in [RFC4944].

While the aforementioned 6LoWPAN fragmentation header is appropriate for IEEE 802.15.4-2003 (which has a frame payload size of 81 to 102 bytes), it is not suitable for several LPWAN technologies, many of which have a maximum payload size that is one order of magnitude below that of IEEE 802.15.4-2003. The overhead of the 6LoWPAN fragmentation header is high, considering the reduced payload size of LPWAN technologies and the limited energy availability of the devices using such technologies. Furthermore, its datagram offset field is expressed in increments of eight octets. In some LPWAN technologies, the 6LoWPAN fragmentation header plus eight octets from the original datagram exceeds the available space in the layer 2 (L2) payload.

This specification defines an optimized 6LoWPAN Fragmentation Header for LPWAN (6LoFHL). Nevertheless, other L2 technologies beyond the LPWAN category may benefit from using 6LoFHL.

It is expected that this specification will be used jointly with other 6Lo(WPAN) mechanisms such as [RFC6282] based header compression.

The benefits of using 6LoFHL are the following:

-- While the 6LoWPAN fragmentation header defined in RFC 4944 has a size of 4 bytes (first fragment) or 5 bytes (subsequent fragments), 6LoFHL has a size of 3 bytes (any fragment). This reduces significantly both the L2 overhead and the adaptation layer overhead for transporting an IPv6 packet that requires fragmentation (see Annex A).

-- Because the datagram offset can be expressed in increments of a single octet, 6LoFHL enables the transport of IPv6 packets over L2 data units with a maximum payload size as small as only 4 bytes in the most extreme case.

1.1. Conventions used in this document

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]

2. 6LoFHL rules and format

If an entire payload (e.g., IPv6) datagram fits within a single L2 data unit, it is unfragmented and a fragmentation header is not needed. If the datagram does not fit within a single L2 data unit, it SHALL be broken into fragments. The first fragment SHALL contain the first fragment header as defined in Figure 1.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 0 0 1|    datagram_size    |  datagram_tag |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: First Fragment

The second and subsequent fragments (up to and including the last) SHALL contain a fragmentation header that conforms to the format shown in Figure 2.

```
                         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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |1 1 0 1 0|   datagram_offset   |  datagram_tag |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Subsequent Fragments

datagram_size: This 11-bit field encodes the size of the entire IP packet before link-layer fragmentation (but after IP layer fragmentation). For IPv6, the datagram size SHALL be 40 octets (the size of the uncompressed IPv6 header) more than the value of Payload Length in the IPv6 header [RFC4944] of the packet. Note that this packet may already be fragmented by hosts involved in the communication, i.e., this field needs to encode a maximum length of 1280 octets (the required by IPv6).

datagram_tag: The value of datagram_tag (datagram tag) SHALL be the same for all fragments of a payload (e.g., IPv6) datagram. The sender SHALL increment datagram_tag for successive, fragmented datagrams. The incremented value of datagram_tag SHALL wrap from 255 back to zero. This field is 8 bits long, and its initial value is not defined.

datagram_offset: This field is present only in the second and subsequent fragments and SHALL specify the offset, in increments of 1 octet, of the fragment from the beginning of the payload datagram. The first octet of the datagram (e.g., the start of the IPv6 header) has an offset of zero; the implicit value of datagram_offset in the first fragment is zero. This field is 11 bits long.

The recipient of link fragments SHALL use (1) the sender's L2 source address, (2) the destination's L2 address, (3) datagram_size, and (4) datagram_tag to identify all the fragments that belong to a given datagram.

Upon receipt of a link fragment, the recipient starts constructing the original unfragmented packet whose size is datagram_size. It uses the datagram_offset field to determine the location of the individual fragments within the original unfragmented packet. For example, it may place the data payload (except the encapsulation header) within a payload datagram reassembly buffer at the location specified by datagram_offset. The size of the reassembly buffer SHALL be determined from datagram_size.

If a fragment recipient disassociates from its L2 network, the recipient MUST discard all link fragments of all partially

reassembled payload datagrams, and fragment senders MUST discard all not yet transmitted link fragments of all partially transmitted payload (e.g., IPv6) datagrams. Similarly, when a node first receives a fragment with a given datagram_tag, it starts a reassembly timer. When this time expires, if the entire packet has not been reassembled, the existing fragments MUST be discarded and the reassembly state MUST be flushed. The reassembly timeout MUST be set to a maximum of TBD seconds).

3. Changes from RFC 4944 fragmentation header and rationale

The main changes introduced in this specification to the fragmentation header format defined in RFC 4944 are listed below, together with their rationale:

-- The datagram size field is only included in the first fragment. Rationale: In the RFC 4944 fragmentation header, the datagram size was included in all fragments to ease the task of reassembly at the receiver, since in an IEEE 802.15.4 mesh network, the fragment that arrives earliest to a destination is not necessarily the first fragment transmitted by the source. However, in LPWAN, such reordering effects are not expected. LPWAN technologies typically define star topology networks, peripheral to peripheral communications are not expected, and the central device is not expected to perform priority queuing operations. Nevertheless, the fragmentation format defined in this document supports limited reordering.

-- The datagram tag size is reduced from 2 bytes to 1 byte. Rationale: Given the low bit rate, as well as the low message rate of LPWAN technologies, ambiguities due to datagram tag wrapping events are not expected to occur despite the reduced tag space.

-- The datagram offset size is increased from 8 bits to 11 bits. Rationale: This allows to express the datagram offset in single-octet increments.

4. IANA Considerations

This document allocates the following sixteen RFC 4944 Dispatch type values:

11001 000

through

11001 111

and

11010 000

through

11010 111

5. Security Considerations

TBD

6. Acknowledgments

In section 2, the authors have reused extensive parts of text available in section 5.3 of RFC 4944, and would like to thank the authors of RFC 4944.

Carles Gomez has been funded in part by the Spanish Government (Ministerio de Educacion, Cultura y Deporte) through the Jose Castillejo grant CAS15/00336. His contribution to this work has been carried out during his stay as a visiting scholar at the Computer Laboratory of the University of Cambridge.

7. **Annex A.** Quantitative comparison of RFC 4944 fragmentation header with 6LoFHL

	IPv6 datagram size (bytes)							
	11		40		100		1280	
L2 payload (bytes)	4944	6LoFHL	4944	6LoFHL	4944	6LoFHL	4944	6LoFHL
10	----	2	----	6	----	15	----	183
15	1	1	5	4	13	9	160	107
20	1	1	4	3	12	6	159	76
25	1	1	3	2	7	5	80	59
30	1	1	2	2	5	4	54	48

Figure 3: L2 overhead (in terms of L2 data units) required to transport an IPv6 datagram

	IPv6 datagram size (bytes)							
	11		40		100		1280	
L2 payload (bytes)	4944	6LoFHL	4944	6LoFHL	4944	6LoFHL	4944	6LoFHL
10	----	6	----	18	----	45	----	768
15	0	0	24	12	64	27	799	321
20	0	0	19	9	59	18	794	228
25	0	0	14	6	34	15	399	177
30	0	0	9	6	24	12	269	144

Figure 4: Adaptation layer fragmentation overhead (in bytes) required to transport an IPv6 datagram

Note 1: with the RFC 4944 fragmentation header it is not possible to transport IPv6 datagrams of the considered sizes over a 10-byte payload L2 technology.

Note 2: 11 bytes is the size of an IPv6 datagram with a 3-byte RFC 6282 compressed header (the shortest possible IPv6 header that uses global addresses), a 4-byte RFC 6282 UDP compressed header, and a CoAP message without header options and without payload.

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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

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

8.2. Informative References

[I-D.minaburo-lp-wan-gap-analysis]
Minaburo, A., Pelov, A., and L. Toutain, "LP-WAN GAP Analysis", draft-minaburo-lp-wan-gap-analysis-01 (work in progress), February 2016.

Authors' Addresses

Carles Gomez
UPC/i2CAT
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

Josep Paradells
UPC/i2CAT
C/Jordi Girona, 1-3
Barcelona 08034
Spain

Email: josep.paradells@entel.upc.edu

Jon Crowcroft
University of Cambridge
JJ Thomson Avenue
Cambridge, CB3 0FD
United Kingdom

Email: jon.crowcroft@cl.cam.ac.uk