6lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2017

C. Gomez
J. Paradells
UPC/i2CAT
J. Crowcroft
University of Cambridge
October 23, 2016

Optimized 6LoWPAN Fragmentation Header
draft-gomez-6lo-optimized-fragmentation-header-00

Abstract

RFC 4944 specifies 6LoWPAN fragmentation, in order to support the IPv6 MTU requirement over IEEE 802.15.4-2003 networks. The 6LoWPAN fragmentation header format comprises a 4-byte format for the first fragment, and a 5-byte format for subsequent fragments. This specification defines a more efficient 3-byte, optimized 6LoWPAN fragmentation header for all fragments.

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 April 26, 2017.

Copyright Notice

Copyright (c) 2016 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

IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) was originally designed as an adaptation layer intended to enable IPv6 over IEEE 802.15.4- 2003 networks [RFC4944]. One of the 6LoWPAN protocol suite components is fragmentation, which fulfills the IPv6 MTU requirement of 1280 bytes [RFC2460] over a radio interface with a layer two (L2) payload size around 100 bytes (in the best case) and without fragmentation support [RFC4944].

RFC 4944 defines the 6LoWPAN fragmentation header format, which comprises a 4-byte format for the first fragment, and a 5-byte format for subsequent fragments. This specification defines a more efficient 3-byte, optimized 6LoWPAN Fragmentation Header (6LoFH). The benefits of using 6LoFH are the following:

-- Reduced overhead for transporting an IPv6 packet that requires fragmentation (see Annex A). This decreases consumption of energy and bandwidth, which are typically limited resources in the scenarios where 6LoWPAN fragmentation is used.

-- Because the datagram offset can be expressed in increments of a single octet, 6LoFH 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. Note that RFC 4944 fragmentation can only be used over L2 technologies with a maximum L2 payload size of at least 13 bytes.

In comparison with the 6LoWPAN fragmentation header, parsing of the 6loFH format is also simplified, as the format has a constant size, and a 'symmetric' shape for both the first fragment and subsequent fragments. However, receiver buffer management will involve greater complexity as explained in Section 3.

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. 6LoFH 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.

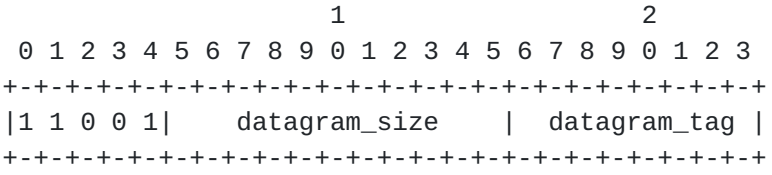

Figure 1: First Fragment

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.

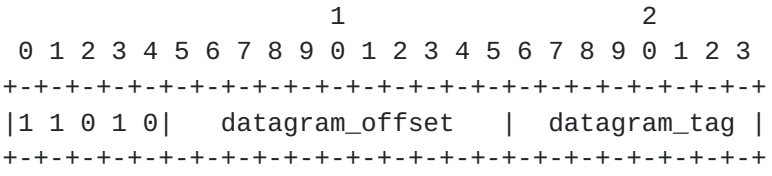

Figure 2: Subsequent Fragments

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. Nevertheless, the fragmentation format defined in this document supports reordering, at the expense of additional complexity in this regard.

-- The datagram tag size is reduced from 2 bytes to 1 byte. Rationale: Given the low bit rate, as well as the relatively low message rate in IEEE 802.15.4 scenarios, ambiguities due to datagram tag wrapping events are unlikely 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

6LoWPAN fragmentation attacks have been analyzed in the literature. Countermeasures to these have been proposed as well [HHWH].

A node can perform a buffer reservation attack by sending a first fragment to a target. Then, the receiver will reserve buffer space for the whole packet on the basis of the datagram size announced in that first fragment. Other incoming fragmented packets will be dropped while the reassembly buffer is occupied during the reassembly timeout. Once that timeout expires, the attacker can repeat the same

procedure, and iterate, thus creating a denial of service attack. The (low) cost to mount this attack is linear with the number of buffers at the target node. However, the cost for an attacker can be increased if individual fragments of multiple packets can be stored in the reassembly buffer. To further increase the attack cost, the reassembly buffer can be split into fragment-sized buffer slots. Once a packet is complete, it is processed normally. If buffer overload occurs, a receiver can discard packets based on the sender behavior, which may help identify which fragments have been sent by an attacker.

In another type of attack, the malicious node is required to have overhearing capabilities. If an attacker can overhear a fragment, it can send a spoofed duplicate (e.g. with random payload) to the destination. A receiver cannot distinguish legitimate from spoofed fragments. Therefore, the original IPv6 packet will be considered corrupt and will be dropped. To protect resource-constrained nodes from this attack, it has been proposed to establish a binding among the fragments to be transmitted by a node, by applying content-chaining to the different fragments, based on cryptographic hash functionality. The aim of this technique is to allow a receiver to identify illegitimate fragments.

Further attacks may involve sending overlapped fragments (i.e. comprising some overlapping parts of the original datagram) or announcing a datagram size in the first fragment that does not reflect the actual amount of data carried by the fragments. Implementers should make sure that correct operation is not affected by such events.

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.

The authors would like to thank Carsten Bormann, Tom Phinney, Ana Minaburo and Laurent Toutain for valuable comments that helped improve the document.

Carles Gomez has been funded in part by the Spanish Government (Ministerio de Educacion, Cultura y Deporte) through the Jose Castillejo grant CAS15/00336. Part of 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 performance comparison of RFC 4944 fragmentation header with 6LoFH

	IPv6 datagram size (bytes)							
	40		100		640		1280	
L2 payload (bytes)	4944	6LoFH	4944	6LoFH	4944	6LoFH	4944	6LoFH
10	----	18	----	45	----	276	----	549
20	19	9	59	18	394	114	794	228
40	0	0	19	9	99	54	199	105
60	0	0	9	6	69	36	134	69
80	0	0	9	6	44	27	89	51
100	0	0	0	0	39	21	74	42

Figure 3: Adaptation layer fragmentation overhead (in bytes) required to transport an IPv6 datagram

Note 1: while IEEE 802.15.4-2003 allows a maximum L2 payload size between 81 and 102 bytes, a range of L2 payload size between 10 and 100 bytes is considered in the study to illustrate the performance of 6LoFH also for other potential L2 technologies with short payload size and without fragmentation support.

Note 2: 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.

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

[HHWH] Hummen et al, R., "6LoWPAN fragmentation attacks and mitigation mechanisms", 2013.

[I-D.minaburo-lpwan-gap-analysis] Minaburo, A., Gomez, C., Toutain, L., Paradells, J., and J. Crowcroft, "LPWAN Survey and GAP Analysis", draft-minaburo-lpwan-gap-analysis-02 (work in progress), October 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