Virtual reassembly buffers in 6LoWPAN - draft-ietf-lwig-6lowpan-virtual-reassembly-00

Carsten Bormann, Thomas Watteyne

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

C. Bormann
Universitaet Bremen TZI
T. Watteyne
Analog Devices

July 02, 2018

Virtual reassembly buffers in 6LoWPAN
draft-ietf-lwig-6lowpan-virtual-reassembly-00

Abstract

When employing adaptation layer fragmentation in 6LoWPAN, it may be beneficial for a forwarder not to have to reassemble each packet in its entirety before forwarding it.

This has been always possible with the original fragmentation design of RFC 4944. Apart from a brief mention of the way to do this in Section 2.5.2 of the 6LoWPAN book, this has not been extensively described in the literature. The present document attempts to fill that gap.

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 January 3, 2019.

Table of Contents

1. Introduction

6LoWPAN [RFC4944] is the seminal standard for the transmission of IPv6 packets over IEEE 802.15.4 networks and has served as a blueprint for a number of related standards addressing low-power radios and other IoT connectivity solutions (the "6Lo suite").

One of the problems that need to be solved to enable sending IPv6 packets over low-power radios is that some of these (including IEEE 802.15.4) do not support frames that are large enough to hold IPv6 packets of the minimum MTU (Maximum Transmission Unit) defined for IPv6, 1280 bytes. This necessitates providing a fragmentation or segmentation scheme in the IP adaptation layer for the radio.

When employing adaptation layer fragmentation on constrained-node networks [RFC7228], it may be beneficial for a forwarder not to have to reassemble each packet in its entirety before forwarding it.

This has been always possible with the original fragmentation design of RFC 4944. Apart from a brief mention of the way to do this in Section 2.5.2 of the 6LoWPAN book [BOOK], this has not been extensively described in the literature. The present document attempts to fill that gap.

[I-D.watteyne-6lo-minimal-fragment] provides additional context and discussion about handling fragment forwarding in the 6Lo standards suite.

2. Reassembly buffers

An adaptation layer implementation for 6LoWPAN needs to perform reassembly of every fragmented packet received in order to be able to forward the packet (re-fragmenting it in the process).

A reassembly buffer for 6LoWPAN contains:

- datagram_size,
- datagram_tag and L2 sender and receiver addresses (to which the datagram_tag is local),
- actual packet data from the fragments received so far, in a form that makes it possible to detect when the whole packet has been received and can be processed or forwarded,
- a timer that allows discarding the partial packet after a timeout.

This requires a reassembly buffer for each fragmented packet the reception of which is in progress. Since the forwarder may be receiving fragments for multiple packets concurrently (e.g., from different senders), this means that multiple reassembly buffers are needed, easily dominating the memory requirements in a 6LoWPAN implementation. Worse, as this space may still be limited, any lack of reassembly buffers may lead to an increased loss rate for fragmented packets (which already have to cope with a higher compound loss rate).

3. Virtual reassembly

To reduce the memory requirement for reassembly buffers, the implementation may opt to not keep the actual packet data in the reassembly buffer. Instead, it may attempt to send out the data for a fragment in the form of a forwarded fragment, as soon as all necessary information for that is available. Obviously, all fragments need to be sent with the same outgoing address (otherwise a full reassembly implementation would discard the fragments) and the same datagram_tag.

To this end, the reassembly buffer now also stores, as soon as enough of the packet is available to make a forwarding decision (i.e., as soon as the first fragment has been received):

- L2 destination address used for forwarding,
- outgoing datagram_tag chosen for this packet.

A simple implementation may do away with any attempt to keep packet data in the virtual reassembly buffer. It then has to discard all non-first fragments for which a reassembly buffer is not already available (penalizing reordering, which however may be rare).

Note that the decision to do local processing of a packet needs to be taken with the first fragment - such packets of course do need to be fully reassembled (unless transport and application also can cope with fragments, which they rarely can in the presence of security).

4. Header compression

[RFC6282] defines the header compression format for 6LoWPAN. One important impact of header compression is that the header is no longer of a fixed length. In particular, changes made by a forwarder may gain or lose the ability to use a more highly compressed variant, changing the length of the header in the packet. If the change increases the size, the maximum frame size may be exceeded, leading to the need to re-fragment in the forwarder. This is less of a problem with full reassembly, but with virtual reassembly can lead to the need for sending an additional frame for each packet.

The well-known approach to minimize the probability of this need is for the original sender to put all slack in the frame sizes into the _first_ packet, making this the smallest fragment and not the last one as would be done in a naive implementation. (This also has other consequences related to delivery probability, which are not discussed here.) This makes sure an additional fragment only needs to be sent if the header expansion during forwarding would have created an additional fragment with full reassembly as well.

5. IANA Considerations

This document makes no requests of IANA.

6. Security considerations

TBD

7. References

7.1. Normative References

[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, <https://www.rfc-editor.org/info/rfc4944>.

7.2. Informative References

[BOOK] Shelby, Z. and C. Bormann, "6LoWPAN", John Wiley & Sons, Ltd monograph, DOI 10.1002/9780470686218, November 2009.

[I-D.watteyne-6lo-minimal-fragment] Watteyne, T., Bormann, C., and P. Thubert, "LLN Minimal Fragment Forwarding", draft-watteyne-6lo-minimal-fragment-01 (work in progress), March 2018.

[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, <https://www.rfc-editor.org/info/rfc6282>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

Acknowledgements

Many people have mentioned that it would be good to have a description of virtual reassembly in 6LoWPAN. Finally, Thomas Watteyne assembled a design team that intends to work on 6Lo fragmentation. Writing up the present document has been motivated by that work.

Authors' Addresses

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

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

Thomas Watteyne
Analog Devices
32990 Alvarado-Niles Road, Suite 910
Union City, CA 94587
USA

Email: thomas.watteyne@analog.com