LPWAN Working Group
Internet-Draft
Intended status: Informational
Expires: February 27, 2020

C. Gomez
UPC
J. Crowcroft
University of Cambridge

August 26, 2019

Fragmentation in LPWAN considerations: CoAP Block vs SCHC fragmentation

draft-gomez-frag-lpwan-considerations-00

Abstract

The SCHC adaptation layer provides header compression and fragmentation functionality between IPv6 and an underlying LPWAN technology. SCHC fragmentation has been specifically designed for the characteristics of LPWANs. However, when CoAP is used at the application layer, there exists an alternative approach for fragmentation, which is using the CoAP Block option. This document aims at illustrating the advantages and limitations of each approach for transferring larger payloads.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Static Context Header Compression and fragmentation (SCHC) framework provides an adaptation layer that has been specifically designed to enable support of IPv6 over Low Power Wide Area Network (LPWAN) technologies [I-D.ietf-lpwan-ipv6-static-context-hc]. SCHC comprises header compression and fragmentation functionality. The latter is needed when SCHC is used over technologies such as LoRaWAN or Sigfox [RFC8376], and might be needed over further LPWAN technologies.

On the other hand, considering the significant resource constraints in many LPWAN scenarios, the Constrained Application Protocol (CoAP) is a suitable candidate application-layer protocol for use in LPWAN. CoAP has been specified over both UDP and TCP [RFC7252][RFC8323]. For CoAP over UDP, the Block option can be used in order to perform application-layer fragmentation [RFC7959]. In this document, CoAP over UDP is assumed.

Therefore, when CoAP and SCHC are used in LPWAN, there exist two possible approaches for fragmentation: SCHC-level fragmentation and CoAP-level fragmentation. This document aims at systematically analyzing the characteristics, advantages and limitations of both approaches.

2. 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].

3. Header overhead

The SCHC specification defines a fragmentation header that is prepended to each fragment. The SCHC fragmentation header has been defined in a flexible way, yet with the aim to enable very short fragmentation header sizes. Fragmentation header sizes are then actually specified per LPWAN technology in corresponding SCHC-over-foo documents. Fragmentation header sizes of 1-2 bytes are currently being used over technologies such as Sigfox or LoRaWAN. When SCHC fragmentation is used, each fragment will carry a SCHC fragmentation header. The first fragment will also carry the SCHC compressed IPv6/UDP header, which may have a size of 1 byte.

When using CoAP blockwise transfers, each block will carry a CoAP header comprising the base header (4 bytes, although perhaps it can be reduced with SCHC CoAP header compression), the option format (2 bytes), the Block option value (1-3 bytes) and the payload marker (1 byte). In addition, each block will be encapsulated in a different IPv6 datagram. A SCHC compressed IPv6/UDP header may have a size of 1 byte.

4. L2 MTU supported

Assuming that the L2 word [I-D.ietf-lpwan-ipv6-static-context-hc] of the underlying LPWAN technology is 1 byte, and a SCHC fragmentation header of 1 byte, SCHC fragmentation can support underlying LPWAN technologies with a maximum link layer data unit payload of even only 2 bytes.

The smallest CoAP layer payload size that can be supported with Blockwise transfers is 16 bytes. Note that the whole header overhead, comprising IPv6, UDP and CoAP headers, even when SCHC compression is applied, is added to the CoAP layer payload. While some LPWAN technologies can, at least in some cases, carry a CoAP Block, note that it is not possible with RFC 7959 to transport a CoAP Block over Sigfox (with uplink and downlink L2 MTU values of 12 and 8 bytes, respectively) or over LoRaWAN DR0 in the US band (with an L2 MTU of 11 bytes). Assuming that SCHC header compression is used, CoAP Blockwise transfers can only be supported over technologies with an L2 MTU of at least ~25 bytes.

5. Reliability modes

SCHC offers a gradation of reliability modes: No-ACK, ACK-Always, and ACK-on-Error. In No-ACK there is no feedback from the receiver to the sender, and there is no retransmission of fragments. In ACK-Always, the receiver inconditionally generates an ACK after a window of fragments. In ACK-on-Error, the receiver only generates an ACK after a window of fragments if at least one of the fragments of the window has been lost (an exception to this behavior is the last window, for which an ACK-Always behavior is used). Note that each mode involves a different message overhead.

CoAP blockwise transfers follow a stop-and-wait behavior by default. Note that congestion control defined in the basic CoAP specification applies, and NON messages are not very useful in blockwise transfers, as they increase the probability of non-delivery [RFC 7959].

6. CoAP RTO calculation

CoAP CON messages require an Acknowledgment (ACK) by the receiving endpoint. After sending a CON message, a sender waits for such ACK for Retransmission TimeOut (RTO) time. Upon RTO expiration, the CoAP sender retransmits the message. The CoAP main specification does not define a mechanism for adaptively calculating the RTO based on the Round Trip Time (RTT). However, it is expected that future specifications will do so [I-D.ietf-core-cocoa]. The RTT comprises the time since the CoAP CON message is passed to its lower layer until an ACK is received.

When SCHC-level fragmentation is used, the RTT seen by CoAP depends significantly on the corresponding IPv6 datagram size. CoAP has no visibility of how many fragments are needed to carry the datagram. Therefore, simple RTO schemes may be inaccurate if CON messages have a variable size. Such inaccuracy may lead to either too long RTO values (involving unnecessarily large delay) or too short ones (leading to spurious retries, which may consume scarce transmission resources).

In contrast, CoAP-level blockwise transfers may exploit the per-block RTT samples, as in fact, each block is carried by a CoAP message, and retries are carried out message-wise. Therefore CoAP blockwise transfers will result in accurate RTT estimation (as long as an adaptive RTO scheme based on RTT samples is used).

Whether the better RTO accuracy of CoAP blockwise transfers may compensate the advantages of SCHC fragmentation (i.e. lower header overhead, better support for payload size constrained L2 technologies, richer reliability approaches) needs to be determined

by means of further study. Note that this open question does not apply for CoAP NON messages.

7. Security Considerations

TBD

8. Acknowledgments

Carles Gomez has been funded in part by the Spanish Government (Ministerio de Ciencia, Innovacion y Universidades) through the Jose Castillejo grant CAS18/00170 and by European Regional Development Fund (ERDF) and the Spanish Government through project TEC2016-79988-P, AEI/FEDER, UE. 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.

9. References

9.1. Normative References

[I-D.ietf-lpwan-ipv6-static-context-hc]
Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and J. Zuniga, "Static Context Header Compression (SCHC) and fragmentation for LPWAN, application to UDP/IPv6", draft-ietf-lpwan-ipv6-static-context-hc-21 (work in progress), July 2019.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[I-D.ietf-core-cocoa]
Bormann, C., Betzler, A., Gomez, C., and I. Demirkol, "CoAP Simple Congestion Control/Advanced", draft-ietf-core-cocoa-03 (work in progress), February 2018.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

[RFC8323] Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, Ed., "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", RFC 8323, DOI 10.17487/RFC8323, February 2018, <https://www.rfc-editor.org/info/rfc8323>.

[RFC8376] Farrell, S., Ed., "Low-Power Wide Area Network (LPWAN) Overview", RFC 8376, DOI 10.17487/RFC8376, May 2018, <https://www.rfc-editor.org/info/rfc8376>.

Authors' Addresses

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

Email: carlesgo@entel.upc.edu

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

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