lpwan Working Group
Internet-Draft
Intended status: Informational
Expires: August 17, 2019

N. Sornin, Ed.
M. Coracin
Semtech
I. Petrov
Acklio
A. Yegin
Actility
J. Catalano
Kerlink
V. Audebert
EDF R&D

February 13, 2019

Static Context Header Compression (SCHC) over LoRaWAN
draft-petrov-lpwan-ipv6-schc-over-lorawan-03

Abstract

The Static Context Header Compression (SCHC) specification describes generic header compression and fragmentation techniques for LPWAN (Low Power Wide Area Networks) technologies. SCHC is a generic mechanism designed for great flexibility, so that it can be adapted for any of the LPWAN technologies.

This document provides the adaptation of SCHC for use in LoRaWAN networks, and provides elements such as efficient parameterization and modes of operation.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Static Context Header Compression (SCHC) specification [I-D.ietf-lpwan-ipv6-static-context-hc] describes generic header compression and fragmentation techniques that can be used on all LPWAN (Low Power Wide Area Networks) technologies defined in [I-D.ietf-lpwan-overview]. Even though those technologies share a

great number of common features like start-oriented topologies, network architecture, devices with mostly quite predictable communications, etc; they do have some slight differences in respect of payload sizes, reactiveness, etc.

SCHC gives a generic framework that enables those devices to communicate with other Internet networks. However, for efficient performance, some parameters and modes of operation need to be set appropriately for each of the LPWAN technologies.

This document describes the efficient parameters and modes of operation when SCHC is used over LoRaWAN networks.

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].

This section defines the terminology and acronyms used in this document. For all other definitions, please look up the SCHC specification [I-D.ietf-lpwan-ipv6-static-context-hc].

o DevEUI: an IEEE EUI-64 identifier used to identify the device during the procedure while joining the network (Join Procedure)

o DevAddr: a 32-bit non-unique identifier assigned to a device statically or dynamically after a Join Procedure (depending on the activation mode)

o TBD: all significant LoRaWAN-related terms.

3. Static Context Header Compression Overview

This section contains a short overview of Static Context Header Compression (SCHC). For a detailed description, refer to the full specification [I-D.ietf-lpwan-ipv6-static-context-hc].

Static Context Header Compression (SCHC) avoids context synchronization, which is the most bandwidth-consuming operation in other header compression mechanisms such as RoHC [RFC5795]. Based on the fact that the nature of data flows is highly predictable in LPWAN networks, some static contexts may be stored on the Device (Dev). The contexts must be stored in both ends, and it can either be learned by a provisioning protocol or by out of band means or it can be pre-provisioned, etc. The way the context is learned on both sides is out of the scope of this document.

```
         Dev                                                   App
   +--------------+                                    +--------------+
   |APP1 APP2 APP3|                                    |APP1 APP2 APP3|
   |              |                                    |              |
   |     UDP      |                                    |     UDP      |
   |    IPv6      |                                    |    IPv6      |
   |              |                                    |              |
   |   SCHC C/D   |                                    |              |
   |   (context)  |                                    |              |
   +-------+------+                                    +-------+------+
           |   +--+     +----+     +----------+               .
           +~~ |RG| === |NGW | === |SCHC C/D |... Internet ..
               +--+     +----+     |(context)|
                                   +----------+
```

Figure 1: Architecture

Figure 1 represents the architecture for compression/decompression, it is based on [I-D.ietf-lpwan-overview] terminology. The Device is sending applications flows using IPv6 or IPv6/UDP protocols. These flows are compressed by an Static Context Header Compression Compressor/Decompressor (SCHC C/D) to reduce headers size. Resulting information is sent on a layer two (L2) frame to a LPWAN Radio Network (RG) which forwards the frame to a Network Gateway (NGW). The NGW sends the data to a SCHC C/D for decompression which shares the same rules with the Dev. The SCHC C/D can be located on the Network Gateway (NGW) or in another place as long as a tunnel is established between the NGW and the SCHC C/D. The SCHC C/D in both sides must share the same set of Rules. After decompression, the packet can be sent on the Internet to one or several LPWAN Application Servers (App).

The SCHC C/D process is bidirectional, so the same principles can be applied in the other direction.

In a LoRaWAN network, the RG is called a Gateway, the NGW is Network Server, and the SCHC C/D can be embedded in different places, for example in the Network Server and/or the Application Server.

Next steps for this section: detailed overview of the LoRaWAN architecture and its mapping to the SCHC architecture.

4. LoRaWAN Architecture

An overview of LoRaWAN [lora-alliance-spec] protocol and architecture is described in [I-D.ietf-lpwan-overview]. Mapping between the LPWAN architecture entities as described in

[I-D.ietf-lpwan-ipv6-static-context-hc] and the ones in [lora-alliance-spec] is as follows:

o Devices (Dev) are the end-devices or hosts (e.g. sensors, actuators, etc.). There can be a very high density of devices per radio gateway. This entity maps to the LoRaWAN End-device.

o The Radio Gateway (RGW), which is the end point of the constrained link. This entity maps to the LoRaWAN Gateway.

o The Network Gateway (NGW) is the interconnection node between the Radio Gateway and the Internet. This entity maps to the LoRaWAN Network Server.

o LPWAN-AAA Server, which controls the user authentication and the applications. This entity maps to the LoRaWAN Join Server.

o Application Server (App). The same terminology is used in LoRaWAN.

```
    ()   ()   ()       |                      +------+
     ()  () () ()      / \       +---------+   | Join |
    () () () () ()   /   \======|    ^    |===|Server|  +-----------+
     () ()  ()      |           | <--|--> |   +------+  |Application|
    () ()  ()  ()  / \==========|    v    |=============|  Server   |
     ()  ()  ()    /   \         +---------+             +-----------+
    End-Devices  Gateways     Network Server
```

Figure 2: LPWAN Architecture

SCHC C/D (Compressor/Decompressor) and SCHC Fragmentation are performed on the LoRaWAN End-device and the Application Server. While the point-to-point link between the End-device and the Application Server constitutes single IP hop, the ultimate end-point of the IP communication may be an Internet node beyond the Application Server. In other words, the LoRaWAN Application Server acts as the first hop IP router for the End-device. Note that the Application Server and Network Server may be co-located, which effectively turns the Network/Application Server into the first hop IP router.

4.1. Device classes (A, B, C) and interactions

The LoRaWAN MAC layer supports 3 classes of devices named A,B and C. All devices implement the classA, some devices implement classA+B or class A+C. ClassB and classC are mutually exclusive.

- *ClassA*: The classA is the simplest class of devices. The device is allowed to transmit at any time, randomly selecting a communication channel. The network may reply with a downlink in one of the 2 receive windows immediately following the uplinks. Therefore, the network cannot initiate a downlink, it has to wait for the next uplink from the device to get a downlink opportunity. The classA is the lowest power device class.

- *ClassB*: classB devices implement all the functionalities of classA devices, but also schedule periodic listen windows. Therefore, as opposed the classA devices, classB devices can receive downlink that are initiated by the network and not following an uplink. There is a trade-off between the periodicity of those scheduled classB listen windows and the power consumption of the device. The lower the downlink latency, the higher the power consumption.

- *ClassC*: classC devices implement all the functionalities of classA devices, but keep their receiver open whenever they are not transmitting. ClassC devices can receive downlinks at any time at the expense of a higher power consumption. Battery powered devices can only operate in classC for a limited amount of time (for example for a firmware upgrade over the air). Most of the classC devices are main powered (for example Smart Plugs).

4.2. Device addressing

LoRaWAN devices use a 32bits network address (devAddr) to communicate with the network over the air. However that address might be reused several time on the same network at the same time for different devices. Devices using the same devAddr are distinguish by the network server based on the cryptographic signature appended to every single LoRaWAN MAC frame, as all devices use different security keys. To communicate with the SCHC gateway the network server MUST identify the devices by a unique 64bits device ID called the devEUI. Unlike devAddr, devEUI is guaranteed to be unique for every single device across all networks. The devEUI is assigned to the device during the manufacturing process by the device's manufacturer. The devEUI is built like an Ethernet MAC address by concatenating the manufacturer's IEEE 24bits OUI field with a 40bits serial number. The network server translates the devAddr into a devEUI in the uplink direction and reciprocally on the downlink direction.

```
+--------+         +---------------+       +--------------------+
| device | <=====> | Network Server| <====> | Application Server |
+--------+ devAddr +---------------+ devEUI +--------------------+
```

Figure 3: LoRaWAN addresses

4.3. General Message Types

- o Confirmed messages:

- o Unconfirmed messages:

4.4. LoRaWAN MAC Frames

- o JoinRequest

- o JoinAccept

- o Data

5. SCHC over LoRaWAN

5.1. Rule ID management

The LoRaWAN MAC layers features a port field in all frames. This port field (FPort) is 8bit long and the values from 1 to 220 can be used. SCHC over LoRaWAN uses 2 contiguous FPort value to separate the uplink SCHC traffic from the downlink and avoid any confusion. Those FPorts are called FPortUp and FPortDwn. Those FPorts can use arbitrary values inside the allowed Fport range but must be shared by the end-device and SCHC gateway.

SCHC over LoRAWAN SHOULD support encoding RuleID on 3 bits, there are therefore 8 possible RuleIds on both uplink and downlink direction.

The RuleID 0 is reserved for fragmentation in both directions. The 7 remaining RuleIDs are available for IPV6 header compression. Uplink (on FPortUp) and downlink (on FportDwn) RuleIDs are independent. The same RuleID may have different meanings on the uplink and downlink paths.

The only uplink messages using the FportDwn port are the fragmentation SCHC ACKs messages of a downlink fragmentation session. Similarly, the only downlink messages using the FportUp port are the fragmentation SCHC ACKs messages of an uplink fragmentation session

5.2. IID computation

TBD (To discuss with the SCHC authors).

5.3. No compression packets are sent using Rule ID 7.

5.4. Fragmentation

The L2 word size used by LoRaWAN is 1 octet (8 bits). The SCHC fragmentation over LoRaWAN exclusively uses the ACK-always mode. A LoRaWAN device cannot support simultaneous interleaved fragmentation sessions in the same direction (uplink or downlink). This means that only a single fragmented IPV6 datagram may be transmitted and/or received by the device at a given moment. The fragmentation parameters are different for uplink and downlink fragmentation sessions and are successively described in the next sections.

5.4.1. Uplink fragmentation: From device to gateway

In that case the device is the fragmentation transmitter, and the SCHC gateway the fragmentation receiver.

- o SCHC fragmentation reliability mode : "ACK_ALWAYS"

- o Window size: 8, the FCN field is encoded on 3 bits

- o DTag : 1bit. this field is used to clearly separate two consecutive fragmentation sessions. A LoRaWAN device cannot interleave several fragmented SCHC datagrams.

- o MIC calculation algorithm: CRC32 using 0xEDB88320 (i.e. the reverse representation of the polynomial used e.g. in the Ethernet standard [RFC3385])

- o Retransmission Timer and inactivity Timer: LoRaWAN devices do not implement a "retransmission timer". At the end of a window the ACK corresponding to this window is transmitted by the network gateway in the RX1 or RX2 receive slot of the device. If this ACK is not received the device sends an all-0 (or an all-1) fragment with no payload to request an ACK retransmission. The periodicity between retransmission of the all-0/all-1 fragments is device/ application specific and may be different for each device (not specified). The gateway implements an "inactivity timer". The default recommended duration of this timer is 12h. This value is mainly driven by application requirements and may be changed.

```
| RuleID | DTag  | W     | FCN    | Payload |
+ ------ + ----- + ----- | ------ + ------- +
| 3 bits | 1 bit | 1 bit | 3 bits |         |
```

Figure 4: All fragment except the last one. Header size is 8 bits.

```
| RuleID | DTag  | W     | FCN    | MIC     | Payload |
+ ------ + ----- + ----- | ------ + ------- + ------- +
| 3 bits | 1 bit | 1 bit | 3 bits | 32 bits |         |
```

Figure 5: All-1 fragment detailed format for the last fragment.
Header size is 8 bits.

The format of an all-0 or all-1 acknowledge is:

```
| RuleID | DTag  | W     | Encoded bitmap | Padding (0s) |
+ ------ + ----- + ----- | -------------- + ------------ +
| 3 bits | 1 bit | 1 bit | 3 or 8 bits    | 0 or 3 bits  |
```

Figure 6: ACK format for All-0 windows. Header size is 1 or 2 bytes.

```
| RuleID | DTag  | W     | C     | Encoded bitmap (if C = 0) | Padding (0s) |
+ ------ + ----- + ----- + ----- + ------------------------- + ------------ +
| 3 bits | 1 bit | 1 bit | 1 bit | 2 or 8 bits               | 0 or 2 bits  |
```

Figure 7: ACK format for All-1 windows. Header size is 1 or 2 bytes.

5.4.2. Downlinks: From gateway to device

In that case the device is the fragmentation receiver, and the SCHC gateway the fragmentation transmitter. The following fields are common to all devices.

- o SCHC fragmentation reliability mode : ACK_ALWAYS

- o Window size : 1 , The FCN field is encoded on 1 bits

- o DTag : 1bit. This field is used to clearly separate two consecutive fragmentation sessions. A LoRaWAN device cannot interleave several fragmented SCHC datagrams.

- o MIC calculation algorithm: CRC32 using 0xEDB88320 (i.e. the reverse representation of the polynomial used e.g. in the Ethernet standard [RFC3385])

o MAX_ACK_REQUESTS : 8

```
| RuleID | DTag  | W     | FCN    | Payload          |
+ ------ + ----- + ----- | ------ + ------- + ------- +
| 3 bits | 1 bit | 1 bit | 1 bits | X bytes + 2 bits  |
```

Figure 8: All fragments but the last one. Header size is 6 bits.

```
| RuleID | DTag  | W     | FCN    | MIC     | Payload | Padding (0s) |
+ ------ + ----- + ----- | ------ + ------- + ------- + ------------ +
| 3 bits | 1 bit | 1 bit | 1 bits | 32 bits | X bytes | 0 to 7 bits  |
```

Figure 9: All-1 Fragment Detailed Format for the Last Fragment. Header size is 6 bits.

The format of an all-0 or all-1 acknowledge is:

```
| RuleID | DTag  | W     | Encoded bitmap | Padding (0s) |
+ ------ + ----- + ----- | -------------- + ------------ +
| 3 bits | 1 bit | 1 bit | 1 bit          | 2 bits       |
```

Figure 10: ACK format for All-0 windows. Header size is 8 bits.

```
| RuleID | DTag  | W     | C = 1 | Padding (0s) |
+ ------ + ----- + ----- + ----- + ------------ +
| 3 bits | 1 bit | 1 bit | 1 bit | 2 bits       |
```

Figure 11: ACK format for All-1 windows, MIC is correct. Header size is 8 bits.

```
| RuleID | DTag  | W     | b'111  | 0xFF (all 1's) |
+ ------ + ----- + ----- + ------ + -------------- +
| 3 bits | 1 bit | 1 bit | 3 bits | 8 bits         |
```

Figure 12: Receiver ABORT packet (following an all-1 packet with incorrect MIC). Header size is 16 bits.

Class A and classB&C devices do not manage retransmissions and timers in the same way.

5.4.2.1. Class A devices

Class A devices can only receive in an RX slot following the transmission of an uplink. Therefore there cannot be a concept of "retransmission timer" for a gateway talking to classA devices for downlink fragmentation.

The device replies with an ACK fragment to every single fragment received from the gateway (because the window size is 1). Following the reception of a FCN=0 fragment (fragment that is not the last fragment of the packet or ACK-request), the device MUST transmit the ACK fragment until it receives the fragment of the next window. The device shall transmit up to MAX_ACK_REQUESTS ACK fragments before aborting. The device should transmit those ACK as soon as possible while taking into consideration eventual local radio regulation on duty-cycle, to progress the fragmentation session as quickly as possible. The ACK bitmap is 1 bit long and is always 1.

Following the reception of a FCN=1 fragment (the last fragment of a datagram) and if the MIC is correct, the device shall transmit the ACK with the "MIC is correct" indicator bit set. This message might be lost therefore the gateway may request a retransmission of this ACK in the next downlink. The device SHALL keep this ACK message in memory until it receives a downlink from the gateway different from an ACK-request indicating that the gateway has received the ACK message.

Following the reception of a FCN=1 fragment (the last fragment of a datagram) and if the MIC is NOT correct, the device shall transmit a receiver-ABORT fragment. The device SHALL keep this ABORT message in memory until it receives a downlink from the gateway different from an ACK-request indicating that the gateway has received the ABORT message. The fragmentation receiver (device) does not implement retransmission timer and inactivity timer.

The fragmentation sender (the gateway) implements an inactivity timer with default duration 12 hours. Once a fragmentation session is started, if the gateway has not received any ACK or receiver-ABORT message 12 hours after the last message from the device was received, the gateway may flush the fragmentation context. For devices with very low transmission rates (example 1 packet a day in normal operation) , that duration may be extended, but this is application specific.

5.4.2.2. Class B or C devices

Class B&C devices can receive in scheduled RX slots or in RX slots following the transmission of an uplink. The device replies with an ACK fragment to every single fragment received from the gateway (because the window size is 1). Following the reception of a FCN=0 fragment (fragment that is not the last fragment of the packet or ACK-request), the device MUST always transmit the corresponding ACK fragment even if that fragment has already been received. The ACK bitmap is 1 bit long and is always 1. If the gateway receives this ACK, it proceeds to send the next window fragment If the retransmission timer elapses and the gateway has not received the ACK of the current window it retransmits the last fragment. The gateway tries retransmitting up to MAX_ACK_REQUESTS times before aborting.

Following the reception of a FCN=1 fragment (the last fragment of a datagram) and if the MIC is correct, the device shall transmit the ACK with the "MIC is correct" indicator bit set. If the gateway receives this ACK, the current fragmentation session has succeeded and its context can be cleared.

If the retransmission timer elapses and the gateway has not received the all-1 ACK it retransmits the last fragment with the payload (not an ACK-request without payload). The gateway tries retransmitting up to MAX_ACK_REQUESTS times before aborting.

The device SHALL keep the all-1 ACK message in memory until it receives a downlink from the gateway different from the last (FCN=1) fragment indicating that the gateway has received the ACK message. Following the reception of a FCN=1 fragment (the last fragment of a datagram) and if the MIC is NOT correct, the device shall transmit a receiver-ABORT fragment. The retransmission timer is used by the gateway (the sender), the optimal value is very much application specific but here are some recommended default values. For classB devices, this timer trigger is a function of the periodicity of the classB ping slots. The recommended value is equal to 3 times the classB ping slot periodicity. For classC devices which are nearly constantly receiving, the recommended value is 30 seconds. This means that the device shall try to transmit the ACK within 30 seconds of the reception of each fragment. The inactivity timer is implemented by the device to flush the context in-case it receives nothing from the gateway over an extended period of time. The recommended value is 12 hours for both classB&C devices.

6. Security considerations

As this document is only providing parameters that are expected to be better suited for LoRaWAN networks for [I-D.ietf-lpwan-ipv6-static-context-hc]. As such, this parameters does not contribute to any new security issues in addition of those identified in [I-D.ietf-lpwan-ipv6-static-context-hc].

7. Acknowledgements

TBD

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

[RFC3385] Sheinwald, D., Satran, J., Thaler, P., and V. Cavanna, "Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC)/Checksum Considerations", RFC 3385, DOI 10.17487/RFC3385, September 2002, <https://www.rfc-editor.org/info/rfc3385>.

[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>.

[RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <https://www.rfc-editor.org/info/rfc5795>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <https://www.rfc-editor.org/info/rfc7136>.

8.2. Informative References

[I-D.ietf-lpwan-ipv6-static-context-hc]
Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and J. Zuniga, "LPWAN Static Context Header Compression (SCHC) and fragmentation for IPv6 and UDP", draft-ietf-lpwan-ipv6-static-context-hc-18 (work in progress), December 2018.

[I-D.ietf-lpwan-overview]
Farrell, S., "LPWAN Overview", draft-ietf-lpwan-overview-10 (work in progress), February 2018.

[lora-alliance-spec]
Alliance, L., "LoRaWAN Specification Version V1.0.2", <http://portal.lora-alliance.org/DesktopModules/Inventures_Document/FileDownload.aspx?ContentID=1398>.

Appendix A. Examples

Appendix B. Note

Authors' Addresses

Nicolas Sornin (editor)
Semtech
14 Chemin des Clos
Meylan
France

Email: nsornin@semtech.com

Michael Coracin
Semtech
14 Chemin des Clos
Meylan
France

Email: mcoracin@semtech.com

Ivaylo Petrov
Acklio
2bis rue de la Chataigneraie
35510 Cesson-Sevigne Cedex
France

Email: ivaylo@ackl.io

Alper Yegin
Actility
.
Paris, Paris
France

Email: alper.yegin@actility.com

Julien Catalano
Kerlink
1 rue Jacqueline Auriol
35235 Thorigne-Fouillard
France

Email: j.catalano@kerlink.fr

Vincent AUDEBERT
EDF R&D
7 bd Gaspard Monge
91120 PALAISEAU
FRANCE

Email: vincent.audebert@edf.fr