lpwan Working Group Internet-Draft Intended status: Informational Expires: May 7, 2020 O. Gimenez, Ed. Semtech I. Petrov, Ed. Acklio November 04, 2019

## Static Context Header Compression (SCHC) over LoRaWAN draft-ietf-lpwan-schc-over-lorawan-04

### 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. This is called a profile.

#### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of  $\underline{BCP 78}$  and  $\underline{BCP 79}$ .

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 <a href="https://datatracker.ietf.org/drafts/current/">https://datatracker.ietf.org/drafts/current/</a>.

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 May 7, 2020.

#### Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>https://trustee.ietf.org/license-info</u>) in effect on the date of publication of this document. Please review these documents

Gimenez & Petrov

Expires May 7, 2020

[Page 1]

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

2	$\underline{1}$ . Introduction	. <u>2</u>
1	<u>2</u> . Terminology	. <u>3</u>
	<u>3</u> . Static Context Header Compression Overview	. <u>3</u>
4	<u>4</u> . LoRaWAN Architecture	. <u>5</u>
	<u>4.1</u> . End-Device classes (A, B, C) and interactions	. <u>6</u>
	4.2. End-Device addressing	. <u>7</u>
	<u>4.3</u> . General Message Types	· <u>7</u>
	<u>4.4</u> . LoRaWAN MAC Frames	. <u>8</u>
	<u>4.5</u> . Unicast and multicast technology	. <u>8</u>
1	<u>5</u> . SCHC-over-LoRaWAN	. <u>8</u>
	<u>5.1</u> . LoRaWAN FPort	. <u>8</u>
	5.2. Rule ID management	. <u>9</u>
	<u>5.3</u> . IID computation	. <u>10</u>
	<u>5.4</u> . Padding	. <u>10</u>
	<u>5.5</u> . Compression	. <u>10</u>
	5.6. Fragmentation	. <u>10</u>
	<u>5.6.1</u> . DTag	. <u>11</u>
	<u>5.6.2</u> . Uplink fragmentation: From device to SCHC gateway .	. <u>11</u>
	5.6.3. Downlink fragmentation: From SCHC gateway to a devic	e 13
	<u>6</u> . Security considerations	. <u>17</u>
1	Acknowledgements	. <u>17</u>
(	Contributors	. <u>17</u>
9	<u>9</u> . References	. <u>18</u>
	<u>9.1</u> . Normative References	. <u>18</u>
	<u>9.2</u> . Informative References	. <u>19</u>
	Appendix A. Examples	. <u>19</u>
	A.1. Uplink - Compression example - No fragmentation	. <u>19</u>
	A.2. Uplink - Compression and fragmentation example	. <u>20</u>
	<u>A.3</u> . Downlink	. <u>22</u>
1	Appendix B. Note	. <u>23</u>
	Authors' Addresses	. 23

# **1**. Introduction

The Static Context Header Compression (SCHC) specification [<u>I-D.ietf-lpwan-ipv6-static-context-hc</u>] describes generic header compression and fragmentation techniques that can be used on all LPWAN (Low Power Wide Area Networks) technologies defined in [<u>RFC8376</u>]. Even though those technologies share a great number of common features like star-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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in <u>BCP</u> 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 end-device during the procedure while joining the network (Join Procedure)

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

o RCS: Reassembly Check Sequence. Used to verify the integrity of the fragmentation-reassembly process

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, 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 context MUST be stored in both ends, and it can either be learned by a provisioning protocol or by out-of-

Internet-Draft

band means or it can be pre-provisioned, etc. The way the context is learned on both sides is outside the scope of this document.

Dev					Арр	
App1 App2 App3				App1	App2	App3
UDP				UDP	UDP	UDP
IPv6				IPv6	IPv6	IPv6
SCHC C/D and F/R						
++   ++ +~  RGW  === ++	++  NGW   == ++	++  SCHC  ==  F/R   ++	++  SCHC   C/D   ++	Ir		++ 

#### Figure 1: Architecture

Figure 1 represents the architecture for compression/decompression, it is based on [RFC8376] terminology. The Device is sending applications flows using IPv6 or IPv6/UDP protocols. These flow might be compressed by an Static Context Header Compression Compressor/Decompressor (SCHC C/D) to reduce headers size and fragmented (SCHC F/R). The resulting information is sent on a layer two (L2) frame to an LPWAN Radio Gateway (RGW) which forwards the frame to a Network Gateway (NGW). The NGW sends the data to a SCHC F/R for defragmentation, if required, then C/D for decompression which shares the same rules with the device. The SCHC F/R and 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 F/R, then SCHC F/R and 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 F/R and 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 is an Application Server. It can be provided by the Network Server or any third party software. Figure 1 can be mapped in LoRaWAN terminology to:



Figure 2: SCHC Architecture mapped to LoRaWAN

### 4. LoRaWAN Architecture

An overview of LoRaWAN [lora-alliance-spec] protocol and architecture is described in [RFC8376]. The 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 (LoRaWAN gateway). This entity maps to the LoRaWAN End-Device.

o The Radio Gateway (RGW), which is the endpoint 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. In that case, the application server will be the SCHC gateway, doing C/D and F/R.





SCHC C/D (Compressor/Decompressor) and SCHC F/R (Fragmentation/ Reassembly) are performed on the LoRaWAN End-Device and the Application Server (called SCHC gateway). 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 (SCHC gateway) acts as the first hop IP router for the End-Device. 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. End-Device classes (A, B, C) and interactions

The LoRaWAN MAC layer supports 3 classes of end-devices named A, B and C. All end-devices implement the Class A, some end-devices may implement Class B or Class C. Class B and Class C are mutually exclusive.

- o Class A: The Class A is the simplest class of end-devices. The end-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 end-device to get a downlink opportunity. The Class A is the lowest power end-device class.
- o Class B: Class B end-devices implement all the functionalities of Class A end-devices, but also schedule periodic listen windows. Therefore, opposed to the Class A end-devices, Class B end-devices can receive downlinks that are initiated by the network and not following an uplink. There is a trade-off between the periodicity of those scheduled Class B listen windows and the power consumption of the end-device. The lower the downlink latency, the higher the power consumption.
- o Class C: Class C end-devices implement all the functionalities of Class A end-devices, but keep their receiver open whenever they

are not transmitting. Class C end-devices can receive downlinks at any time at the expense of a higher power consumption. Battery-powered end-devices can only operate in Class C for a limited amount of time (for example for a firmware upgrade overthe-air). Most of the Class C end-devices are grid powered (for example Smart Plugs).

## <u>4.2</u>. End-Device addressing

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

++		+-		+		+		+		+-	+
End -	<====>		Network		<===>		SCHC		<=====>		Internet
Device	devAddr		Server		devEUI		Gateway		IPv6/UDP		
++		+-		+		+		+		+-	+

## Figure 4: LoRaWAN addresses

### <u>4.3</u>. General Message Types

- Confirmed messages: The sender asks the receiver to acknowledge the message.
- Unconfirmed messages: The sender does not ask the receiver to acknowledge the message.

As SCHC defines its own acknowledgment mechanisms, SCHC does not require to use confirmed messages.

### 4.4. LoRaWAN MAC Frames

- o JoinRequest: This message is used by an end-device to join a network. It contains the end-device's unique identifier devEUI and a random nonce that will be used for session key derivation.
- o JoinAccept: To on-board an end-device, the Network Server responds to the JoinRequest end-device's message with a JoinAccept message. That message is encrypted with the end-device's AppKey and contains (amongst other fields) the major network's settings and a network random nonce used to derive the session keys.
- o Data

## **4.5.** Unicast and multicast technology

LoRaWAN technology supports unicast downlinks, but also multicast: a packet send over LoRaWAN radio link can be received by several devices. It is useful to address many end-devices with same content, either a large binary file (firmware upgrade), or same command (e.g. lighting control). As IPv6 is also a multicast technology this feature MAY be used to address a group of devices.

Note 1 : IPv6 multicast addresses must be defined as per [RFC4291]. LoRaWAN multicast group definition in a network server and the relation between those groups and IPv6 groupID are out of scope of this document.

Note 2 : LoRa Alliance defined [lora-alliance-remote-multicast-set] as RECOMMENDED way to setup multicast groups on devices and create a synchronized reception window.

## 5. SCHC-over-LoRaWAN

## 5.1. LoRaWAN FPort

The LoRaWAN MAC layer features a frame port field in all frames. This field (FPort) is 8 bits long and the values from 1 to 223 can be used. It allows LoRaWAN networks and applications to identify data.

The FPort field is part of the SCHC Packet or the SCHC Fragment, as shown in Figure 5. The SCHC C/D and the SCHC F/R SHALL concatenate the FPort field with the LoRaWAN payload to retrieve their payload as it is used as a part of the ruleId field.

Figure 5: SCHC payload in LoRaWAN

A fragmentation session with application payload transferred from device to server, is called uplink fragmentation session. It uses an FPort for data uplink and its associated SCHC control downlinks, named FPortUp in this document. The other way, a fragmentation session with application payload transferred from server to device, is called downlink fragmentation session. It uses another FPort for data downlink and its associated SCHC control uplinks, named FPortDown in this document.

FPorts can use arbitrary values inside the allowed FPort range and MUST be shared by the end-device, the Network Server and SCHC gateway prior to the communication. The uplink and downlink fragmentation FPorts MUST be different.

# 5.2. Rule ID management

RuleID MUST be 8 bits, encoded in the LoRaWAN FPort as described in <u>Section 5.1</u>. LoRaWAN supports up to 223 application FPorts in the range [1;223] as defined in section 4.3.2 of [<u>lora-alliance-spec</u>], it implies that RuleID MSB SHOULD be inside this range. An application MAY reserve some FPort values for other needs as long as they don't conflict with FPorts used for SCHC C/D and SCHC F/R.

In order to improve interoperability RECOMMENDED fragmentation RuleID values are:

- o RuleID = 20 (8-bit) for uplink fragmentation, named FPortUp
- o RuleID = 21 (8-bit) for downlink fragmentation, named FPortDown
- o RuleID = 22 (8-bit) for which SCHC compression was not possible (no matching rule was found)

The remaining RuleIDs are available for compression. RuleIDs are shared between uplink and downlink sessions. A RuleID different from FPortUp or FPortDown means that the fragmentation is not used, thus the packet SHOULD be sent to C/D layer.

The only uplink messages using the FPortDown port are the fragmentation SCHC control messages of a downlink fragmentation session (ex ACKs). Similarly, the only downlink messages using the

FPortUp port are the fragmentation SCHC control messages of an uplink fragmentation session.

An application can have multiple fragmentation sessions between a device and one or several SCHC gateways. A set of FPort values is REQUIRED for each SCHC gateway instance the device is required to communicate with.

The mechanism for sharing those RuleID values is outside the scope of this document.

## **5.3**. IID computation

As LoRaWAN network uses unique EUI-64 per end-device, the Interface IDentifier is the LoRaWAN DevEUI. It is compliant with [<u>RFC4291</u>] and IID starting with binary 000 must enforce the 64-bit rule.

TODO: Derive IID from DevEUI with privacy constraints ? Ask working group ?

## 5.4. Padding

All padding bits MUST be 0.

## <u>5.5</u>. Compression

SCHC C/D MUST concatenate FPort and LoRaWAN payload to retrieve the SCHC packet as per <u>Section 5.1</u>.

RuleIDs matching FPortUp and FPortDown are reserved for SCHC Fragmentation.

# **<u>5.6</u>**. Fragmentation

The L2 word size used by LoRaWAN is 1 byte (8 bits). The SCHC fragmentation over LoRaWAN uses the ACK-on-Error for uplink fragmentation and Ack-Always for downlink fragmentation. A LoRaWAN end-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 end-device at a given moment.

The fragmentation parameters are different for uplink and downlink fragmentation sessions and are successively described in the next sections.

## 5.6.1. DTag

A LoRaWAN device cannot interleave several fragmented SCHC datagrams on the same FPort. This field is not used and its size is 0.

Note: The device can still have several parallel fragmentation sessions with one or more SCHC gateway(s) thanks to distinct sets of FPorts, cf Section 5.2

## **<u>5.6.2</u>**. Uplink fragmentation: From device to SCHC gateway

In that case the device is the fragmentation transmitter, and the SCHC gateway the fragmentation receiver. A single fragmentation rule is defined. SCHC F/R MUST concatenate FPort and LoRaWAN payload to retrieve the SCHC fragment as per Section 5.1.

- SCHC header size is two bytes (the FPort byte + 1 additional byte).
- o RuleID: 8 bits stored in LoRaWAN FPort.
- o SCHC fragmentation reliability mode: "ACK-on-Error"
- o DTag: Size is 0 bit, not used
- o FCN: The FCN field is encoded on N = 6 bits, so WINDOW\_SIZE = 63 tiles are allowed in a window
- o Window index: encoded on W = 2 bits. So 4 windows are available.
- o RCS calculation algorithm: CRC32 using 0xEDB88320 (i.e. the reverse representation of the polynomial used e.g. in the Ethernet standard [<u>RFC3385</u>]) as suggested in [<u>I-D.ietf-lpwan-ipv6-static-context-hc</u>].
- o MAX ACK REQUESTS: 8
- o Tile: size is 10 bytes
- o Retransmission and inactivity timers: LoRaWAN end-devices do not implement a "retransmission timer". At the end of a window or a fragmentation session, corresponding ACK(s) is (are) transmitted by the network gateway (LoRaWAN application server) in the RX1 or RX2 receive slot of end-device. If this ACK is not received by the end-device at the end of its RX windows, it sends an all-0 (or an all-1) fragment with no payload to request an SCHC 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 SCHC gateway implements an "inactivity timer". The default RECOMMENDED duration of this timer is 12 hours. This value is mainly driven by application requirements and MAY be changed by the application.

o Last tile: The last tile can be carried in the All-1 fragment.

With this set of parameters, the SCHC fragment header is 16 bits, including FPort; payload overhead will be 8 bits as FPort is already a part of LoRaWAN payload. MTU is: 4 windows \* 63 tiles \* 10 bytes per tile = 2520 bytes

# **<u>5.6.2.1</u>**. Regular fragments

	FPort		LoRaW	٩N	payload	t		
T I	RuloTD	Ť		1	FCN	1	Pavload	т Т
1	Nucerb	1	vv	 _		1		1
i	8 bits	i	2 bits	i	6 bits	i		i

Figure 6: All fragments except the last one. SCHC header size is 16 bits, including LoRaWAN FPort.

## 5.6.2.2. Last fragment (All-1)

	FPort		LoRaWAN	l payload					
+	RuleID	+	W	FCN=All-1	RCS		Payload	- ·	
Ť	8 bits	Ť	2 bits	6 bits	32 bits	Ť	Last tile, if any	- ا /	

Figure 7: All-1 fragment detailed format for the last fragment.

## 5.6.2.3. SCHC ACK

	FPort		LoRaWA	١N	payloa	d		
+		+						+
I	RuleID		W	l	С		Encoded bitmap (if C = $0$ )	
+		+		+		+		+
	8 bits		2 bit		1 bit		0 to 127 bits	Ι

Figure 8: SCHC formats, failed RCS check.

## 5.6.2.4. Receiver-Abort

	FPort		LoRaWAN payload	
+		+	++	
ļ	RuleID	ļ	W = b'11   C = 1   b'11111   0xFF (all 1's)	
+	8 bits	+	2 bits   1 bit   5 bits   8 bits   next L2 Word boundary ->  < L2 Word>	•

Figure 9: Receiver-Abort format.

## 5.6.2.5. SCHC acknowledge request

	FPort		LoRaWAN payload	
+	RuleID	+	W   FCN = b'0000	+ 00
- 	8 bits		2 bits   6 bits	+

Figure 10: SCHC ACK REQ format.

## 5.6.3. Downlink fragmentation: From SCHC gateway to a 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. SCHC F/R MUST concatenate FPort and LoRaWAN payload to retrieve the SCHC fragment as described in <u>Section 5.1</u>.

- o SCHC fragmentation reliability mode:
  - \* Unicast downlinks: ACK-Always.
  - \* Multicast downlinks: No-ACK, reliability has be be ensured by the upper layer. This feature is OPTIONAL and may not be implemented by SCHC gateway.
- o RuleID: 8 bits stored in LoRaWAN FPort.
- o Window index (unicast only): encoded on W=1 bit, as per [I-D.ietf-lpwan-ipv6-static-context-hc].
- o DTag: Size is 0 bit, not used
- o FCN: The FCN field is encoded on N=1 bit, so WINDOW\_SIZE = 1 tile
  (FCN=All-1 is reserved for SCHC).

- o RCS calculation algorithm: CRC32 using 0xEDB88320 (i.e. the reverse representation of the polynomial used e.g. in the Ethernet standard [<u>RFC3385</u>]), as per [I-D.ietf-lpwan-ipv6-static-context-hc].
- o MAX ACK REQUESTS: 8

As only 1 tile is used, its size can change for each downlink, and will be maximum available MTU.

Note : The Fpending bit included in LoRaWAN protocol SHOULD NOT be used for SCHC-over-LoRaWAN protocol. It might be set by the Network Server for other purposes but not SCHC needs.

## **5.6.3.1**. Regular fragments

	FPort		Lo	RaWA	N	payload			
+		+							+
	RuleID		W			FCN = b'0		Payload	
+		+			+		+		+
	8 bits	I	1	bit	I	1 bit		X bytes	

Figure 11: All fragments but the last one. Header size 10 bits, including LoraWAN FPort.

## 5.6.3.2. Last fragment (All-1)

	FPort		LoRaWA	١N	payload					
Ī	RuleID	Ì	W		FCN = b'1		RCS	l	Payload	ļ
+	8 bits	+	1 bit	+	1 bit	+	32 bits	+	Last tile, if any	+

Figure 12: All-1 SCHC ACK detailed format for the last fragment.

## 5.6.3.3. SCHC acknowledge

	FPort		LoRaWA	Ν	payload				
+		+							+
	RuleID	I	W	I	C = b'1	I	Padding	b'000000	
+		+		+		+			+
	8 bits		1 bit		1 bit	Ι	6 bits		

Figure 13: SCHC ACK format, RCS is correct.

Gimenez & Petrov Expires May 7, 2020

### 5.6.3.4. Receiver-Abort

Figure 14: Receiver-Abort packet (following an all-1 packet with incorrect RCS).

Class A and Class B or Class C end-devices do not manage retransmissions and timers in the same way.

## 5.6.3.5. Class A end-devices

Class A end-devices can only receive in an RX slot following the transmission of an uplink. Therefore there cannot be a concept of "retransmission timer" for an SCHC gateway. The SCHC gateway cannot initiate communication to a Class A end-device.

The device replies with an ACK message to every single fragment received from the SCHC 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, but the end of a window), the device MUST transmit the SCHC ACK fragment until it receives the fragment of the next window. The device SHALL transmit up to MAX\_ACK\_REQUESTS ACK messages before aborting. The device should transmit those ACK as soon as possible while taking into consideration potential 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 an FCN=All-1 fragment (the last fragment of a datagram) and if the RCS is correct, the device SHALL transmit the ACK with the "RCS is correct" indicator bit set (C=1). This message might be lost therefore the SCHC 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, on SCHC FPortDown from the SCHC gateway different from an ACK-request: it indicates that the SCHC gateway has received the ACK message.

Following the reception of a FCN=All-1 fragment (the last fragment of a datagram), if all fragments have been received and the RCS is not correct, the device SHALL transmit a Receiver-Abort fragment. The

Gimenez & Petrov Expires May 7, 2020 [Page 15]

device SHALL keep this Abort message in memory until it receives a downlink, on SCHC FPortDown, from the SCHC gateway different from an ACK-request indicating that the SCHC gateway has received the Abort message. The fragmentation receiver (device) does not implement retransmission timer and inactivity timer.

The fragmentation sender (the SCHC gateway) implements an inactivity timer with a default duration of 12 hours. Once a fragmentation session is started, if the SCHC gateway has not received any ACK or Receiver-Abort message 12 hours after the last message from the device was received, the SCHC 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.

# 5.6.3.6. Class B or Class C end-devices

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

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

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

Following the reception of an FCN=All-1 fragment (the last fragment of a datagram), if all fragments have been received and if the RCS is NOT correct, the device SHALL transmit a Receiver-Abort fragment. The retransmission timer is used by the SCHC gateway (the sender), the optimal value is very much application specific but here are some recommended default values. For Class B end-devices, this timer

trigger is a function of the periodicity of the Class B ping slots. The RECOMMENDED value is equal to 3 times the Class B ping slot periodicity. For Class C end-devices which are nearly constantly receiving, the RECOMMENDED value is 30 seconds. This means that the end-device shall try to transmit the ACK within 30 seconds of the reception of each fragment. The inactivity timer is implemented by the end-device to flush the context in-case it receives nothing from the SCHC gateway over an extended period of time. The RECOMMENDED value is 12 hours for both Class B and Class C end-devices.

## **<u>6</u>**. Security considerations

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

### Acknowledgements

Thanks to all those listed in the Contributors section for the excellent text, insightful discussions, reviews and suggestions.

### Contributors

Contributors ordered by family name.

- o ins: V. Audebert name: Vincent AUDEBERT org: EDF R&D street: 7 bd Gaspard Monge city: 91120 PALAISEAU country: FRANCE email: vincent.audebert@edf.fr
- o ins: J. Catalano name: Julien Catalano org: Kerlink street: 1 rue Jacqueline Auriol city: 35235 Thorigne-Fouillard country: France email: j.catalano@kerlink.fr
- o ins: M. Coracin name: Michael Coracin org: Semtech street: 14 Chemin des Clos city: Meylan country: France email: mcoracin@semtech.com
- o ins: M. Le Gourrierec name: Marc Le Gourrierec org: SagemCom street: 250 Route de l'Empereur city: 92500 Rueil Malmaison country: FRANCE email: marc.legourrierec@sagemcom.com
- o ins: N. Sornin name: Nicolas Sornin org: Semtech street: 14 Chemin des Clos city: Meylan country: France email: nsornin@semtech.com

Gimenez & Petrov Expires May 7, 2020 [Page 17]

o ins: A. Yegin name: Alper Yegin org: Actility street: . city: Paris, Paris country: France email: alper.yegin@actility.com

### 9. References

# <u>9.1</u>. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, DOI 10.17487/RFC2119, March 1997, <<u>https://www.rfc-editor.org/info/rfc2119</u>>.
- [RFC3385] Sheinwald, D., Satran, J., Thaler, P., and V. Cavanna, "Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC)/Checksum Considerations", <u>RFC 3385</u>, DOI 10.17487/RFC3385, September 2002, <https://www.rfc-editor.org/info/rfc3385>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", <u>RFC 4291</u>, DOI 10.17487/RFC4291, February 2006, <<u>https://www.rfc-editor.org/info/rfc4291</u>>.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", <u>RFC 4944</u>, DOI 10.17487/RFC4944, September 2007, <<u>https://www.rfc-editor.org/info/rfc4944</u>>.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", <u>RFC 5795</u>, DOI 10.17487/RFC5795, March 2010, <<u>https://www.rfc-editor.org/info/rfc5795</u>>.
- [RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", <u>RFC 7136</u>, DOI 10.17487/RFC7136, February 2014, <<u>https://www.rfc-editor.org/info/rfc7136</u>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in <u>RFC</u> 2119 Key Words", <u>BCP 14</u>, <u>RFC 8174</u>, DOI 10.17487/RFC8174, May 2017, <<u>https://www.rfc-editor.org/info/rfc8174</u>>.

# <u>9.2</u>. Informative 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", <u>draftietf-lpwan-ipv6-static-context-hc-22</u> (work in progress), October 2019.

[lora-alliance-remote-multicast-set] Alliance, L., "LoRaWAN Remote Multicast Setup Specification Version 1.0.0", <<u>https://lora-alliance.org/sites/default/files/2018-09/remote\_multicast\_setup\_v1.0.0.pdf</u>>.

[lora-alliance-spec]

Alliance, L., "LoRaWAN Specification Version V1.0.3", <<u>https://lora-alliance.org/sites/default/files/2018-07/</u> lorawan1.0.3.pdf>.

# <u>Appendix A</u>. Examples

## A.1. Uplink - Compression example - No fragmentation

Figure 15 is representing an applicative payload going through SCHC, no fragmentation required

An applicative payload of 78 bytes is passed to SCHC compression layer using rule 1, allowing to compress it to 40 bytes and 5 bits: 1 byte ruleID, 21 bits residue + 37 bytes payload.

| RuleID | Compression residue | Payload | Padding=b'000 | | 1 | 21 bits | 38 bytes | 3 bits |

The current LoRaWAN MTU is 51 bytes, although 2 bytes FOpts are used by LoRaWAN protocol: 49 bytes are available for SCHC payload; no need for fragmentation. The payload will be transmitted through FPort = 1

	LoRaWAN	Header				LoRaWAN payl	Loa	ad (40 byte	es)		
		FOpts		RuleID=1		Compression residue		Payload		Padding=b'000	
1	XXXX   2	2 bytes	- 	1 byte	Ť	21 bits	- 	37 bytes	т 	3 bits	Ť

Figure 15: Uplink example: compression without fragmentation

#### A.2. Uplink - Compression and fragmentation example

Figure 16 is representing an applicative payload going through SCHC, with fragmentation.

An applicative payload of 478 bytes is passed to SCHC compression layer using rule 1, allowing to compress it to 282 bytes and 5 bits: 1 byte ruleID, 21 bits residue + 279 bytes payload.

| RuleID | Compression residue | Payload | + ----- + ----- + ----- + ------ + | 1 | 21 bits | 279 bytes |

The current LoRaWAN MTU is 11 bytes, 0 bytes FOpts are used by LoRaWAN protocol: 11 bytes are available for SCHC payload + 1 byte FPort field. SCHC header is 2 bytes (including FPort) so 1 tile is sent in first fragment.

| LoRaWAN Header | LoRaWAN payload (11 bytes) | | RuleID=20 | W | FCN | 1 tile | 

Internet-Draft SCHC-over-LoRaWAN

XXXX | 1 byte | 0 0 | 62 | 10 bytes |

Content of the tile is: | RuleID | Compression residue | Payload | + ----- + ------- + ------- + | 1 | 21 bits | 6 byte + 3 bits |

Next transmission MTU is 11 bytes, although 2 bytes FOpts are used by LoRaWAN protocol: 9 bytes are available for SCHC payload + 1 byte FPort field, a tile does not fit inside so LoRaWAN stack will send only FOpts.

Next transmission MTU is 242 bytes, 4 bytes FOpts. 23 tiles are transmitted:

	LoRaWAN Header						LoR	aWAN	payload	b	(231 bytes)	
-   -			F0pts		RuleID=20		W		FCN	   _	23 tiles	
1	XXXX	T	4 bytes	T	1 byte	Ţ	0	0	61		230 bytes	

Next transmission MTU is 242 bytes, no FOpts. All 5 remaining tiles are transmitted, the last tile is only 2 bytes + 5 bits. Padding is added for the remaining 3 bits.

LoRaWAN Header	LoRaWAN payload (44 bytes)	
RuleID=20	W   FCN   5 tiles	Padding=b'000
XXXX   1 byte	0 0   38   42 bytes + 5 bits	3 bits

All packets have been received by the SCHC gateway, computed RCS is correct so the following ACK is sent to the device:

I	LoRaWAN Header			I	LoRa	aWA	N	ра	۱y٦	08	ad	
+		+		+								+
			RuleID=20		W			С		Pa	adding	
+		+		+			+	-	+			+
	XXXX	I	1 byte	Ι	0	0		1		5	bits	Ι

Figure 16: Uplink example: compression and fragmentation

Internet-Draft

SCHC-over-LoRaWAN

### A.3. Downlink

An applicative payload of 443 bytes is passed to SCHC compression layer using rule 1, allowing to compress it to 130 bytes and 5 bits: 1 byte ruleId, 21 bits residue + 127 bytes payload.

| RuleID | Compression residue | Payload | + ----- + ------ + ------ + | 1 | 21 bits | 127 bytes |

The current LoRaWAN MTU is 51 bytes, no FOpts are used by LoRaWAN protocol: 48 bytes are available for SCHC payload + FPort field => it has to be fragmented.

Content of the tile is: | RuleID | Compression residue | Payload | + ----- + ----- + ------ + | 1 | 21 bits | 46 bytes + 3 bits |

The receiver answers with an SCHC ACK

	FPortDown		LoRaWAN	bayload		
+		+				+
I	RuleID	I	W = 0	C = b'1   P	adding=b'000000	
+		+	+	· + -		+
	1 byte		1 bit	Lbit	6 bits	

The second downlink is sent, two FOpts:

	LoRaWAN	Header				Lo	Ra	WAN	pa	ayload (49	) k	bytes)	
-   +		FOpts		RuleID=21	-   +	W		FCN		1 tile	Ĭ	Padding=b'000000	
1	XXXX   2	bytes	т   1	1 byte	-	1	- 	0	Ţ	48 bytes	+	6 bits	1

Gimenez & Petrov

Internet-Draft

The receiver answers with an SCHC ACK

ļ	FPortDown	ļ	LoRaWAN payload	
+	RuleID	+	W = 1   C = b'1   Padding=b'000000	-   -
+	1 byte	+	1 bit   1 bit   6 bits	

The last downlink is sent, no FOpts:

	LoRaWAN Header		LoR	aW/	AN pay	/load	(33 bytes)			
+   +	RuleID=21	-   -	W		FCN	1	tile		Padding=b'0	-   +
T	XXXX   1 byte	- 	0		1	32	bytes + 5 bits		1 bit	T

The receiver answers with an SCHC ACK

	FPortDown		LoRaWAN	pa	ayload			
т		т						т
Ι	RuleID	Ι	W = 0	С	= b'1		Padding=b'000000	
+		+	+			+		+
Ì	1 byte	Ì	1 bit	1	bit	Ì	6 bits	Ì

Figure 17: Downlink example: compression and fragmentation

## Appendix B. Note

Authors' Addresses

Olivier Gimenez (editor) Semtech 14 Chemin des Clos Meylan France

Email: ogimenez@semtech.com

Ivaylo Petrov (editor) Acklio 1137A Avenue des Champs Blancs 35510 Cesson-Sevigne Cedex France

Email: ivaylo@ackl.io