lpwan Working Group   O. Gimenez, Ed.
Internet-Draft   Semtech
Intended status: Standards Track   I. Petrov, Ed.
Expires: July 29, 2021   Acklio
                                                        January 25, 2021

Static Context Header Compression (SCHC) over LoRaWAN
draft-ietf-lpwan-schc-over-lorawan-14

Abstract

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

This document specifies a profile of RFC8724 to use SCHC in LoRaWAN(R) networks, and provides elements such as efficient parameterization and modes of operation.

Table of Contents

1. Introduction

SCHC specification [RFC8724] describes generic header compression and fragmentation techniques that can be used on all Low Power Wide Area Networks (LPWAN) technologies defined in [RFC8376]. 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 with respect to payload sizes, reactiveness, etc.

SCHC provides a generic framework that enables those devices to communicate on IP 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 parameters and modes of operation when SCHC is used over LoRaWAN networks. LoRaWAN protocol is specified by the LoRa Alliance(R) in [lora-alliance-spec]

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 BCP 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 [RFC8724].

o DevEUI: Device Extended Unique Identifier, an IEEE EUI-64 identifier used to identify the device during the procedure while joining the network (Join Procedure). It is assigned by the manufacturer or the device owner and provisioned on the Network Gateway.

o DevAddr: a 32-bit non-unique identifier assigned to a device either:

   * Statically: by the device manufacturer in _Activation by Personalization_ mode.

   * Dynamically: after a Join Procedure by the Network Gateway in _Over The Air Activation_ mode.

o Downlink: LoRaWAN term for a frame transmitted by the network and received by the device.

- EUI: Extended Unique Identifier

- LoRaWAN: LoRaWAN is a wireless technology based on Industrial, Scientific, and Medical (ISM) radio bands that is used for long-range, low-power, low-data-rate applications developed by the LoRa Alliance, a membership consortium: https://www.lora-alliance.org [1].

- FRMPayload: Application data in a LoRaWAN frame.

- MSB: Most Significant Byte

- OUI: Organisation Unique Identifier. IEEE assigned prefix for EUI.

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

- RX: Device's reception window.

- RX1/RX2: LoRaWAN class A devices open two RX windows following an uplink, called RX1 and RX2.

- SCHC gateway: The LoRaWAN Application Server that manages translation between IPv6 network and the Network Gateway (LoRaWAN Network Server).

- Tile: Piece of a fragmented packet as described in [RFC8724] section 8.2.2.1

- Uplink: LoRaWAN term for a frame transmitted by the device and received by the network.

3. Static Context Header Compression Overview

This section contains a short overview of SCHC. For a detailed description, refer to the full specification [RFC8724].

It defines:

1. Compression mechanisms to avoid transporting information known by both sender and receiver over the air. Known information is part of the "context". This component is called SCHC Compressor/ Decompressor (SCHC C/D).

2. Fragmentation mechanisms to allow SCHC Packet transportation on small, and potentially variable, MTU. This component is called SCHC Fragmentation/Reassembly (SCHC F/R).

Context exchange or pre-provisioning is out of scope of this document.

```
    Device                                                   App
 +----------------+                                  +----+ +----+ +----+
 | App1 App2 App3 |                                  |App1| |App2| |App3|
 |                |                                  |    | |    | |    |
 |       UDP      |                                  |UDP | |UDP | |UDP |
 |      IPv6      |                                  |IPv6| |IPv6| |IPv6|
 |                |                                  |    | |    | |    |
 |SCHC C/D and F/R|                                  |    | |    | |    |
 +--------+-------+                                  +----+ +----+ +----+
          |  +---+     +----+    +----+    +----+     .      .      .
          +~ |RGW| === |NGW | == |SCHC| == |SCHC|...... Internet ....
             +---+     +----+    |F/R |    |C/D |
                                 +----+    +----+
 |<- - - - LoRaWAN - - ->|
```

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 flows might be compressed by a Static Context Header Compression Compressor/Decompressor (SCHC C/D) to reduce headers size and fragmented by the SCHC Fragmentation/Reassembly (SCHC F/R). The resulting information is sent on a layer two (L2) frame to an LPWAN Radio Gateway (RGW) that forwards the frame to a Network Gateway (NGW). The NGW sends the data to a SCHC F/R for reassembly, if required, then to SCHC C/D for decompression. The SCHC C/D shares the same rules with the device. The SCHC C/D and F/R can be located on the Network Gateway (NGW) or in another place as long as a communication is established between the NGW and the SCHC F/R, then SCHC F/R and C/D. The SCHC C/D and F/R in the device and the SCHC gateway 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 and F/R process is bidirectional, so the same principles can be applied to the other direction.

In a LoRaWAN network, the RGW is called a Gateway, the NGW is Network Server, and the SCHC C/D and F/R are an Application Server. It can be provided by the Network Gateway or any third party software. Figure 1 can be mapped in LoRaWAN terminology to:

```
   End Device                                                App
+--------------+                                  +----+ +----+ +----+
|App1 App2 App3|                                  |App1| |App2| |App3|
|              |                                  |    | |    | |    |
|     UDP      |                                  |UDP | |UDP | |UDP |
|    IPv6      |                                  |IPv6| |IPv6| |IPv6|
|              |                                  |    | |    | |    |
|SCHC C/D & F/R|                                  |    | |    | |    |
+-------+------+                                  +----+ +----+ +----+
        |  +-------+     +-------+    +-----------+    .      .      .
        +~ |Gateway| === |Network| == |Application|..... Internet ....
           +-------+     |server |    |server     |
                         +-------+    | F/R - C/D |
                                      +-----------+
|<- - - - - LoRaWAN - - - ->|
```

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 [RFC8724] and the ones in [lora-alliance-spec] is as follows:

o Devices are LoRaWAN End Devices (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 SCHC gateway (LoRaWAN Application server). This entity maps to the LoRaWAN Network Server.

o SCHC C/D and F/R are handled by LoRaWAN Application Server; ie the LoRaWAN application server will do the SCHC C/D and F/R.

o The LPWAN-AAA Server is the LoRaWAN Join Server. Its role is to manage and deliver security keys in a secure way, so that the devices root key is never exposed.

```
                                         (LPWAN-AAA Server)
    ()   ()   ()       |                      +------+
     ()  () () ()     / \       +---------+   | Join |
    () () () () ()   /   \======|    ^    |===|Server|  +-----------+
     () ()  ()      |           | <--|--> |   +------+  |Application|
    () ()  ()  ()  / \==========|    v    |=============|  Server   |
     ()  ()  ()   /   \         +---------+             +-----------+
    End-devices  Gateways     Network Server         (SCHC C/D and F/R)
     (devices)    (RGW)            (NGW)
```

Figure 3: LPWAN Architecture

Note: Figure 3 terms are from LoRaWAN, with [RFC8376] terminology in brackets.

SCHC Compressor/Decompressor (SCHC C/D) and SCHC Fragmentation/ Reassembly (SCHC F/R) are performed on the LoRaWAN end-device and the Application Server (called SCHC gateway). While the point-to-point link between the device and the Application Server constitutes a 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 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. 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 Class A, some devices may implement Class B or Class C. Class B and Class C are mutually exclusive.

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

- Class B: Class B devices implement all the functionalities of Class A devices, but also schedule periodic listen windows. Therefore, opposed to the Class A devices, Class B devices can receive downlinks that are initiated by the Network Gateway 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 device: if the periodicity is high

downlinks from the NGW will be sent faster, but the device wakes up more often: it will have higher power consumption.

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

4.2. Device addressing

LoRaWAN end-devices use a 32-bit network address (devAddr) to communicate with the Network Gateway over-the-air, this address might not be unique in a LoRaWAN network. Devices using the same devAddr are distinguished by the Network Gateway based on the cryptographic signature appended to every LoRaWAN frame.

To communicate with the SCHC gateway, the Network Gateway MUST identify the devices by a unique 64-bit device identifier called the DevEUI.

The DevEUI is assigned to the device during the manufacturing process by the 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 Gateway translates the devAddr into a DevEUI in the uplink direction and reciprocally on the downlink direction.

```
+--------+         +---------+        +---------+          +----------+
| Device | <=====> | Network | <====> | SCHC    | <======> | Internet |
|        | devAddr | Gateway | DevEUI | Gateway | IPv6/UDP |          |
+--------+         +---------+        +---------+          +----------+
```

Figure 4: LoRaWAN addresses

4.3. General Frame Types

LoRaWAN implements the possibility to send confirmed or unconfirmed frames:

- o Confirmed frame: The sender asks the receiver to acknowledge the frame.

- o Unconfirmed frame: The sender does not ask the receiver to acknowledge the frame.

As SCHC defines its own acknowledgment mechanisms, SCHC does not require the use of LoRaWAN Confirmed frames (MType=0b100 as per [lora-alliance-spec])

4.4. LoRaWAN MAC Frames

In addition to regular data frames, LoRaWAN implements JoinRequest and JoinAccept frame types, which are used by a device to join a network:

- o JoinRequest: This frame is used by a device to join a network. It contains the device's unique identifier DevEUI and a random nonce that will be used for session key derivation.

- o JoinAccept: To on-board a device, the Network Gateway responds to the JoinRequest issued by a device with a JoinAccept frame. That frame is encrypted with the device's AppKey and contains (amongst other fields) the network's major settings and a random nonce used to derive the session keys.

- o Data: MAC and application data. Application data are protected with AES-128 encryption. MAC related data are AES-128 encrypted with another key.

4.5. 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.

4.6. LoRaWAN empty frame

A LoRaWAN empty frame is a LoRaWAN frame without FPort (cf Section 5.1) and FRMPayload.

4.7. Unicast and multicast technology

LoRaWAN technology supports unicast downlinks, but also multicast: a packet sent over LoRaWAN radio link can be received by several devices. It is useful to address many 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 can 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 Gateway 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 the RECOMMENDED way to setup multicast groups on devices and create a synchronized reception window.

5. SCHC-over-LoRaWAN

5.1. LoRaWAN FPort and RuleID

The FPort field is part of the SCHC Message, as shown in Figure 5. The SCHC C/D and the SCHC F/R SHALL concatenate the FPort field with the LoRaWAN payload to recompose the SCHC Message.

```
| FPort | LoRaWAN payload  |
+ ------------------------ +
|       SCHC Message       |
```

Figure 5: SCHC Message in LoRaWAN

Note: SCHC Message is any datagram sent by SCHC C/D or F/R layers.

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

All RuleID can use arbitrary values inside the FPort range allowed by LoRaWAN specification and MUST be shared by the device and SCHC gateway prior to the communication with the selected rule. 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 Section 5.1. LoRaWAN supports up to 223 application FPorts in the range [1;223] as defined in section 4.3.2 of [lora-alliance-spec], it implies that RuleID MSB SHOULD be inside this range. An application can send non SCHC traffic by using FPort values different from the ones used for SCHC.

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 (i.e., no matching compression Rule was found), as described in [RFC8724] section 6.

FPortUp value MUST be different from FPortDown. The remaining RuleIDs are available for compression. RuleIDs are shared between uplink and downlink sessions. A RuleID not in the set(s) of FPortUp or FPortDown means that the fragmentation is not used, thus, on reception, the SCHC Message MUST be sent to the SCHC C/D layer.

The only uplink frames using the FPortDown port are the fragmentation SCHC control messages of a downlink fragmented datagram (for example, SCHC ACKs). Similarly, the only downlink frames using the FPortUp port are the fragmentation SCHC control messages of an uplink fragmented datagram.

An application can have multiple fragmented datagrams 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 application can use additional uplinks or downlink fragmented parameters but SHALL implement at least the parameters defined in this document.

The mechanism for context distribution across devices and gateways is outside the scope of this document.

5.3. Interface IDentifier (IID) computation

In order to mitigate the risks described in [RFC8064] and [RFC8065], implementation MUST implement the following algorithm and SHOULD use it.

1. key = LoRaWAN AppSKey
2. cmac = aes128_cmac(key, DevEUI)
3. IID = cmac[0..7]

aes128_cmac algorithm is described in [RFC4493]. It has been chosen as it is already used by devices for LoRaWAN protocol.

As AppSKey is renewed each time a device joins or rejoins a LoRaWAN network, the IID will change over time; this mitigates privacy, location tracking and correlation over time risks. Join periodicity is defined at the application level.

Address scan risk is mitigated thanks to AES-128, which provides enough entropy bits of the IID.

Using this algorithm will also ensure that there is no correlation between the hardware identifier (IEEE-64 DevEUI) and the IID, so an attacker cannot use manufacturer OUI to target devices.

Example with:

- DevEUI: 0x1122334455667788

- appSKey: 0x00AABBCCDDEEFF00AABBCCDDEEFFAABB

```
1. key: 0x00AABBCCDDEEFF00AABBCCDDEEFFAABB
2. cmac: 0xBA59F4B196C6C3432D9383C145AD412A
3. IID: 0xBA59F4B196C6C343
```

Figure 6: Example of IID computation.

There is a small probability of IID collision in a LoRaWAN network. If this occurs, the IID can be changed by rekeying the device at L2 level (ie: trigger a LoRaWAN join). The way the device is rekeyed is out of scope of this document and left to the implementation.

Note: Implementation also using another IID source MUST ensure that the same IID is shared between the device and the SCHC gateway in the compression and decompression of the IPv6 address of the device.

5.4. Padding

All padding bits MUST be 0.

5.5. Decompression

SCHC C/D MUST concatenate FPort and LoRaWAN payload to retrieve the SCHC Packet as per Section 5.1.

RuleIDs matching FPortUp and FPortDown are reserved for SCHC Fragmentation.

5.6. Fragmentation

The L2 Word Size used by LoRaWAN is 1 byte (8 bits). The SCHC fragmentation over LoRaWAN uses the ACK-on-Error mode for uplink fragmentation and Ack-Always mode for downlink fragmentation. A LoRaWAN device cannot support simultaneous interleaved fragmented datagrams in the same direction (uplink or downlink).

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

5.6.1. DTag

[RFC8724] section 8.2.4 describes the possibility to interleave several fragmented SCHC datagrams for the same RuleID. This is not used in SCHC over LoRaWAN profile. A 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 fragmented datagrams with more than one SCHC gateway thanks to distinct sets of FPorts, cf Section 5.2.

5.6.2. Uplink fragmentation: From device to SCHC gateway

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

- o SCHC fragmentation reliability mode: "ACK-on-Error".
- o SCHC header size is two bytes (the FPort byte + 1 additional byte).
- o RuleID: 8 bits stored in LoRaWAN FPort. cf Section 5.2
- o DTag: Size T=0 bit, not used. cf Section 5.6.1
- o Window index: 4 windows are used, encoded on M = 2 bits
- o FCN: The FCN field is encoded on N = 6 bits, so WINDOW_SIZE = 63 tiles are allowed in a window.
- o Last tile: it can be carried in a Regular SCHC Fragment, alone in an All-1 SCHC Fragment or with any of these two methods. Implementation must ensure that:

   * The sender MUST ascertain that the receiver will not receive the last tile through both a Regular SCHC Fragment and an All-1 SCHC Fragment during the same session.

   * If the last tile is in All-1 SCHC message: current L2 MTU MUST be big enough to fit the All-1 header and the last tile.

o Penultimate tile MUST be equal to the regular size.

o RCS: Use recommended calculation algorithm in [RFC8724] (S.8.2.3. Integrity Checking).

o Tile: size is 10 bytes.

o Retransmission timer: Set by the implementation depending on the application requirements. 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 Inactivity timer: 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 MAX_ACK_REQUESTS: 8. 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_

In addition to the per-rule context parameters specified in [RFC8724], for uplink rules, an additional context parameter is added: whether or not to ack after each window.
For battery powered devices, it is RECOMMENDED to use the ACK mechanism at the end of each window instead of waiting until the end of all windows:

o The SCHC receiver SHOULD send a SCHC ACK after every window even if there is no missing tile.

o The SCHC sender SHOULD wait for the SCHC ACK from the SCHC receiver before sending tiles from the next window. If the SCHC ACK is not received, it SHOULD send a SCHC ACK REQ up to MAX_ACK_REQUESTS times, as described previously.

This will avoid useless uplinks if the device has lost network coverage.

For non-battery powered devices, the SCHC receiver MAY also choose to send a SCHC ACK only at the end of all windows. This will reduce downlink load on the LoRaWAN network, by reducing the number of downlinks.

SCHC implementations MUST be compatible with both behaviors, and this selection is part of the rule context.

5.6.2.1. Regular fragments

```
| FPort  |  LoRaWAN payload          |
+ ------ + ------------------------- +
| RuleID |   W    | FCN    | Payload |
+ ------ + ------ + ------ + ------- +
| 8 bits | 2 bits | 6 bits |         |
```

Figure 7: 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 payload             |
+ ------ + --------------------------- +
| RuleID |   W    | FCN=All-1 |  RCS    |
+ ------ + ------ + --------- + ------- +
| 8 bits | 2 bits | 6 bits    | 32 bits |
```

Figure 8: All-1 SCHC Message: the last fragment without last tile.

```
| FPort  | LoRaWAN payload                                          |
+ ------ + -------------------------------------------------------- +
| RuleID |   W    | FCN=All-1 |  RCS    |  Last tile   | Opt. padding |
+ ------ + ------ + --------- + ------- + ------------ + ------------ +
| 8 bits | 2 bits |  6 bits   | 32 bits | 1 to 80 bits | 0 to 7 bits  |
```

Figure 9: All-1 SCHC Message: the last fragment with last tile.

5.6.2.3. SCHC ACK

```
| FPort  | LoRaWAN payload           |
+ ------ + -------------------------+
| RuleID |   W   | C = 1 |  padding  |
|        |       |       | (b'00000) |
+ ------ + ----- + ----- + --------- +
| 8 bits | 2 bit | 1 bit |  5 bits   |
```

Figure 10: SCHC ACK format, correct RCS check.

```
| FPort  | LoRaWAN payload                                       |
+ ------ + ------------------------------- + ---------------- +
| RuleID |   W   | C = 0 | Compressed bitmap | Optional padding |
|        |       |       |      (C = 0)      |    (b'0...0)     |
+ ------ + ----- + ----- + ----------------- + ---------------- +
| 8 bits | 2 bit | 1 bit |    5 to 63 bits   |  0, 6 or 7 bits  |
```

Figure 11: SCHC ACK format, failed RCS check.

Note: Because of the bitmap compression mechanism and L2 byte alignment, only the following discrete values are possible for the compressed bitmap size: 5, 13, 21, 29, 37, 45, 53, 61, 62 and 63. Bitmaps of 63 bits will require 6 bits of padding.

5.6.2.4. Receiver-Abort

```
| FPort  | LoRaWAN payload                                     |
+ ------ + ------------------------------------------- +
| RuleID | 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 12: Receiver-Abort format.

5.6.2.5. SCHC acknowledge request

```
| FPort  | LoRaWAN payload          |
+------- +------------------------ +
| RuleID | W      | FCN = b'000000  |
+ ------ + ------ + --------------- +
| 8 bits | 2 bits | 6 bits          |
```

Figure 13: SCHC ACK REQ format.

5.6.3. Downlink fragmentation: From SCHC gateway to device

In this 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 Packet as described in Section 5.1.

- SCHC fragmentation reliability mode:
  - Unicast downlinks: ACK-Always.
  - Multicast downlinks: No-ACK, reliability has to be ensured by the upper layer. This feature is OPTIONAL and may not be implemented by SCHC gateway.
- RuleID: 8 bits stored in LoRaWAN FPort. cf Section 5.2
- DTag: Size T=0 bit, not used. cf Section 5.6.1
- FCN: The FCN field is encoded on N=1 bit, so WINDOW_SIZE = 1 tile.
- RCS: Use recommended calculation algorithm in [RFC8724] (S.8.2.3. Integrity Checking).
- 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.

The following parameters apply to ACK-Always (Unicast) only:

- Retransmission timer: See Section 5.6.3.5.
- MAX_ACK_REQUESTS: 8.
- Window index (unicast only): encoded on M=1 bit, as per [RFC8724].

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

Class A devices can only receive during an RX slot, following the transmission of an uplink. Therefore the SCHC gateway cannot initiate communication (e.g., start a new SCHC session). In order to create a downlink opportunity it is RECOMMENDED for Class A devices to send an uplink every 24 hours when no SCHC session is started, this is application specific and can be disabled. The RECOMMENDED uplink is a LoRaWAN empty frame as defined Section 4.6. As this uplink is to open an RX window, any LoRaWAN uplink frame from the device MAY reset this counter.

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

5.6.3.1. Regular fragments

```
| FPort  | LoRaWAN payload                    |
+ ------ + ---------------------------------- +
| RuleID | W     | FCN = b'0 | Payload          |
+ ------ + ----- + --------- + ---------------- +
| 8 bits | 1 bit | 1 bit     | X bytes + 6 bits |
```

Figure 14: All fragments but the last one. Header size 10 bits, including LoRaWAN FPort.

5.6.3.2. Last fragment (All-1)

```
| FPort  | LoRaWAN payload                                       |
+ ------ + -------------------------- + ------------------------ +
| RuleID | W     | FCN = b'1 |   RCS   |   Payload   | Opt padding |
+ ------ + ----- + --------- + ------- + ----------- + ----------- +
| 8 bits | 1 bit | 1 bit     | 32 bits | 6 to X bits | 0 to 7 bits |
```

Figure 15: All-1 SCHC Message: the last fragment.

5.6.3.3. SCHC ACK

```
| FPort  | LoRaWAN payload                  |
+ ------ + -------------------------------- +
| RuleID | W     | C = b'1 | Padding b'000000 |
+ ------ + ----- + ------- + ---------------- +
| 8 bits | 1 bit | 1 bit   | 6 bits           |
```

Figure 16: SCHC ACK format, RCS is correct.

```
| FPort  | LoRaWAN payload                                  |
+ ------ + ------------------------------------------------ +
| RuleID | W     | C = b'0 | Bitmap = b'1 | Padding b'000000 |
+ ------ + ----- + ------- + ------------ + ---------------- +
| 8 bits | 1 bit | 1 bit   |    1 bit     |      5 bits      |
```

Figure 17: SCHC ACK format, RCS is incorrect.

5.6.3.4. Receiver-Abort

```
| FPort  | LoRaWAN payload                                      |
+ ------ + ---------------------------------------------- +
| RuleID | W = b'1 | C = b'1 | b'111111 | 0xFF (all 1's)  |
+ ------ + ------- + ------- + -------- + --------------- +
| 8 bits | 1 bit   | 1 bits  | 6 bits   | 8 bits          |
                next L2 Word boundary ->| <-- L2 Word --> |
```

Figure 18: Receiver-Abort packet (following an All-1 SCHC Fragment with incorrect RCS).

5.6.3.5. Downlink retransmission timer

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

5.6.3.5.1. Class A devices

Class A devices can only receive in an RX slot following the transmission of an uplink.

The SCHC gateway implements an inactivity timer with a RECOMMENDED duration of 36 hours. For devices with very low transmission rates (example 1 packet a day in normal operation), that duration may be extended: it is application specific.

RETRANSMISSION_TIMER is application specific and its RECOMMENDED value is INACTIVITY_TIMER/(MAX_ACK_REQUESTS + 1).

SCHC All-0 (FCN=0)

All fragments but the last have an FCN=0 (because window size is 1). Following an All-0 SCHC Fragment, the device MUST transmit the SCHC ACK message. It MUST transmit up to MAX_ACK_REQUESTS SCHC ACK messages before aborting. In order to progress the fragmented datagram, the SCHC layer should immediately queue for transmission those SCHC ACK if no SCHC downlink have been received during RX1 and RX2 window. LoRaWAN layer will respect the applicable local spectrum regulation.

Note: The ACK bitmap is 1 bit long and is always 1.

SCHC All-1 (FCN=1)

SCHC All-1 is the last fragment of a datagram, the corresponding SCHC ACK message might be lost; therefore the SCHC gateway MUST request a retransmission of this ACK when the retransmission timer expires. To open a downlink opportunity the device MUST transmit an uplink every RETRANSMISSION_TIMER/(MAX_ACK_REQUESTS * SCHC_ACK_REQ_DN_OPPORTUNITY). The format of this uplink is application specific. It is RECOMMENDED for a device to send an empty frame (see Section 4.6) but it is application specific and will be used by the NGW to transmit a potential SCHC ACK REQ. SCHC_ACK_REQ_DN_OPPORTUNITY is application specific and its recommended value is 2. It MUST be greater than 1. This allows to open a downlink opportunity to any downlink with higher priority than the SCHC ACK REQ message.

Note: The device MUST keep this SCHC ACK message in memory until it receives a downlink SCHC Fragmentation Message (with FPort == FPortDown) that is not a SCHC ACK REQ: it indicates that the SCHC gateway has received the SCHC ACK message.

5.6.3.6. Class B or Class C devices

Class B devices can receive in scheduled RX slots or in RX slots following the transmission of an uplink. Class C devices are almost in constant reception.

RECOMMENDED retransmission timer value:

o Class B: 3 times the ping slot periodicity.

o Class C: 30 seconds.

The RECOMMENDED inactivity timer value is 12 hours for both Class B and Class C devices.

5.7. SCHC Fragment Format

5.7.1. All-0 SCHC fragment

Uplink fragmentation (Ack-On-Error):

All-0 is distinguishable from a SCHC ACK REQ as [RFC8724] states _This condition is also met if the SCHC Fragment Header is a multiple of L2 Words_; this condition met: SCHC header is 2 bytes.

Downlink fragmentation (Ack-always):

As per [RFC8724] the SCHC All-1 MUST contain the last tile, implementation must ensure that SCHC All-0 message Payload will be at least the size of an L2 Word.

5.7.2. All-1 SCHC fragment

All-1 is distinguishable from a SCHC Sender-Abort as [RFC8724] states _This condition is met if the RCS is present and is at least the size of an L2 Word_; this condition met: RCS is 4 bytes.

5.7.3. Delay after each LoRaWAN frame to respect local regulation

This profile does not define a delay to be added after each LoRaWAN frame, local regulation compliance is expected to be enforced by LoRaWAN stack.

6. Security Considerations

This document is only providing parameters that are expected to be best suited for LoRaWAN networks for [RFC8724]. IID security is discussed in Section 5.3. As such, this document does not contribute to any new security issues beyond those already identified in [RFC8724]. Moreover, SCHC data (LoRaWAN payload) are protected at the LoRaWAN level by an AES-128 encryption with a session key shared by the device and the SCHC gateway. These session keys are renewed at each LoRaWAN session (ie: each join or rejoin to the LoRaWAN network)

7. IANA Considerations

This document has no IANA actions.

Acknowledgements

Thanks to all those listed in the Contributors section for the excellent text, insightful discussions, reviews and suggestions, and also to (in alphabetical order) Dominique Barthel, Arunprabhu Kandasamy, Rodrigo Munoz, Alexander Pelov, Pascal Thubert, Laurent Toutain for useful design considerations, reviews and comments.

Contributors

Contributors ordered by family name.

Vincent Audebert
EDF R&D
Email: vincent.audebert@edf.fr

Julien Catalano
Kerlink
Email: j.catalano@kerlink.fr

Michael Coracin
Semtech
Email: mcoracin@semtech.com

Marc Le Gourrierec
Sagemcom
Email: marc.legourrierec@sagemcom.com

Nicolas Sornin
Semtech
Email: nsornin@semtech.com

Alper Yegin
Actility
Email: alper.yegin@actility.com

10. References

10.1. Normative References

[lora-alliance-spec]
Alliance, L., "LoRaWAN Specification Version V1.0.4", <https://lora-alliance.org/resource_hub/lorawan-104-specification-package/>.

[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>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, DOI 10.17487/RFC4493, June 2006, <https://www.rfc-editor.org/info/rfc4493>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/info/rfc8724>.

10.2. Informative References

[lora-alliance-remote-multicast-set] Alliance, L., "LoRaWAN Remote Multicast Setup Specification Version 1.0.0", <https://lora-alliance.org/sites/default/files/2018-09/remote_multicast_setup_v1.0.0.pdf>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <https://www.rfc-editor.org/info/rfc8065>.

[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>.

10.3. URIs

[1] https://www.lora-alliance.org

Appendix A. Examples

In following examples "applicative data" refers to the IPv6 payload sent by the application to the SCHC layer.

A.1. Uplink - Compression example - No fragmentation

This example represents an applicative data going through SCHC over LoRaWAN, no fragmentation required

An applicative data of 78 bytes is passed to SCHC compression layer. Rule 1 is used by SCHC C/D layer, 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	37 bytes	3 bits

Figure 19: Uplink example: SCHC Message

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 payload (40 bytes)		
	FOpts	RuleID=1	Compression residue	Payload	Padding=b'000
XXXX	2 bytes	1 byte	21 bits	37 bytes	3 bits

Figure 20: Uplink example: LoRaWAN packet

A.2. Uplink - Compression and fragmentation example

This example represents an applicative data going through SCHC, with fragmentation.

An applicative data of 300 bytes is passed to SCHC compression layer. Rule 1 is used by SCHC C/D layer, 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

Figure 21: Uplink example: SCHC Message

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
XXXX	1 byte	0 0	62	10 bytes

Figure 22: Uplink example: LoRaWAN packet 1

Content of the tile is:

RuleID	Compression residue	Payload
1	21 bits	6 bytes + 3 bits

Figure 23: Uplink example: LoRaWAN packet 1 - Tile content

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			LoRaWAN payload (231 bytes)		
	FOpts	RuleID=20	W	FCN	23 tiles
XXXX	4 bytes	1 byte	0 0	61	230 bytes

Figure 24: Uplink example: LoRaWAN packet 2

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

Figure 25: Uplink example: LoRaWAN packet 3

Then All-1 message can be transmitted:

LoRaWAN Header		LoRaWAN payload (44 bytes)		
	RuleID=20	W	FCN	RCS
XXXX	1 byte	0 0	63	4 bytes

Figure 26: Uplink example: LoRaWAN packet 4 - All-1 SCHC message

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

```
| LoRaWAN Header             | LoRaWAN payload     |
+ -------------- + --------- + ------------------- +
|                | RuleID=20 |   W    | C | Padding |
+ -------------- + --------- + ----- + - + ------- +
|       XXXX     | 1 byte    | 0   0 | 1 | 5 bits  |
```

Figure 27: Uplink example: LoRaWAN packet 5 - SCHC ACK

A.3. Downlink

An applicative data of 155 bytes is passed to SCHC compression layer. Rule 1 is used by SCHC C/D layer, 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 |
```

Figure 28: Downlink example: SCHC Message

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

```
| LoRaWAN Header    | LoRaWAN payload (51 bytes)            |
+ ---- + ---------- + ------------------------------------- +
|      | RuleID=21  |  W = 0 | FCN = 0 |       1 tile       |
+ ---- + ---------- + ------ + ------- + ------------------ +
| XXXX | 1 byte     |  1 bit |  1 bit  | 50 bytes and 6 bits |
```

Figure 29: Downlink example: LoRaWAN packet 1 - SCHC Fragment 1

Content of the tile is:

```
| RuleID | Compression residue |       Payload     |
+ ------ + ------------------- + ----------------- +
|   1    |       21 bits       | 48 bytes and 1 bit |
```

Figure 30: Downlink example: LoRaWAN packet 1: Tile content

The receiver answers with a SCHC ACK:

```
| LoRaWAN Header   | LoRaWAN payload                   |
+ ---- + --------- + --------------------------------- +
|      | RuleID=21 | W = 0 | C = 1 | Padding=b'000000 |
+ ---- + --------- + ----- + ----- + ---------------- +
| XXXX |  1 byte   | 1 bit | 1 bit |     6 bits       |
```

Figure 31: Downlink example: LoRaWAN packet 2 - SCHC ACK

The second downlink is sent, two FOpts:

```
| LoRaWAN Header              |  LoRaWAN payload (49 bytes)            |
+ --------------------------- + -------------------------------------- +
|      |  FOpts  | RuleID=21  | W = 1 | FCN = 0 |        1 tile        |
+ ---- + ------- + ---------- + ----- + ------- + -------------------- +
| XXXX | 2 bytes | 1 byte     | 1 bit |  1 bit  | 48 bytes and 6 bits  |
```

Figure 32: Downlink example: LoRaWAN packet 3 - SCHC Fragment 2

The receiver answers with an SCHC ACK:

```
| LoRaWAN Header   | LoRaWAN payload                   |
+ ---- + --------- + --------------------------------- +
|      | RuleID=21 | W = 1 | C = 1 | Padding=b'000000 |
+ ---- + --------- + ----- + ----- + ---------------- +
| XXXX |  1 byte   | 1 bit | 1 bit |     6 bits       |
```

Figure 33: Downlink example: LoRaWAN packet 4 - SCHC ACK

The last downlink is sent, no FOpts:

```
| LoRaWAN Header | LoRaWAN payload (37 bytes)                           |
+ ---- + ------- + ---------------------------------------------------- +
|      | RuleID  |   W   |  FCN  |   RCS   |      1 tile     | Padding |
|      |   21    |   0   |   1   |         |                 | b'00000 |
+ ---- + ------- + ----- + ----- + ------- + --------------- + ------- +
| XXXX | 1 byte  | 1 bit | 1 bit | 4 bytes | 31 bytes+1 bits | 5 bits  |
```

Figure 34: Downlink example: LoRaWAN packet 5 - All-1 SCHC message

The receiver answers to the sender with an SCHC ACK:

```
| LoRaWAN Header   | LoRaWAN payload                   |
+ ---- + --------- + --------------------------------- +
|      | RuleID=21 | W = 0 | C = 1 | Padding=b'000000 |
+ ---- + --------- + ----- + ----- + ---------------- +
| XXXX |  1 byte   | 1 bit | 1 bit |     6 bits       |
```

Figure 35: Downlink example: LoRaWAN packet 6 - SCHC ACK

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