Internet Engineering Task Force (IETF)
Request for Comments: 9011
Category: Standards Track
ISSN: 2070-1721

O. Gimenez, Ed.
Semtech
I. Petrov, Ed.
Acklio

April 2021

Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN

Abstract

The Static Context Header Compression and fragmentation (SCHC) specification (RFC 8724) describes generic header compression and fragmentation techniques for Low-Power Wide Area Network (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 defines a profile of SCHC (RFC 8724) for use in LoRaWAN networks and provides elements such as efficient parameterization and modes of operation.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9011.

Table of Contents

1. Introduction

The SCHC specification [RFC8724] describes generic header compression and fragmentation techniques that can be used on all Low-Power Wide Area Network (LPWAN) technologies defined in [RFC8376]. Even though those technologies share a great number of common features like star-oriented topologies, network architecture, devices with communications that are mostly quite predictable, 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. The LoRaWAN protocol is specified by the LoRa Alliance in [LORAWAN-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 abbreviations used in this document. For all other definitions, please look up the SCHC specification [RFC8724].

> Note: The SCHC acronym is pronounced like "sheek" in English (or "chic" in French). Therefore, this document writes "a SCHC Packet" instead of "an SCHC Packet".

AppKey: Application Key. An AES-128 root key specific to each device.

AppSKey: Application Session Key. An AES-128 key derived from the AppKey for each new session. It is used to encrypt the payload field of a LoRaWAN applicative frame.

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

Statically: by the device manufacturer in "Activation-by-Personalization" mode, or

Dynamically: after a LoRaWAN "Join Procedure" by the Network Gateway in "Over-the-Air-Activation" mode.

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.

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

EUI: Extended Unique Identifier

FRMPayload: Application data in a LoRaWAN frame

IID: Interface 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>.

MSB: Most Significant Byte

NGW: Network Gateway

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

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

RGW: Radio Gateway

RX: A device’s reception window.

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

SCHC C/D: SCHC Compression/Decompression

SCHC F/R: SCHC Fragmentation/Reassembly

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

Tile: A piece of a fragmented packet as described in Section 8.2.2.1 of [RFC8724].

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

3. SCHC 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 the "SCHC Compression/Decompression" (SCHC C/D).

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

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

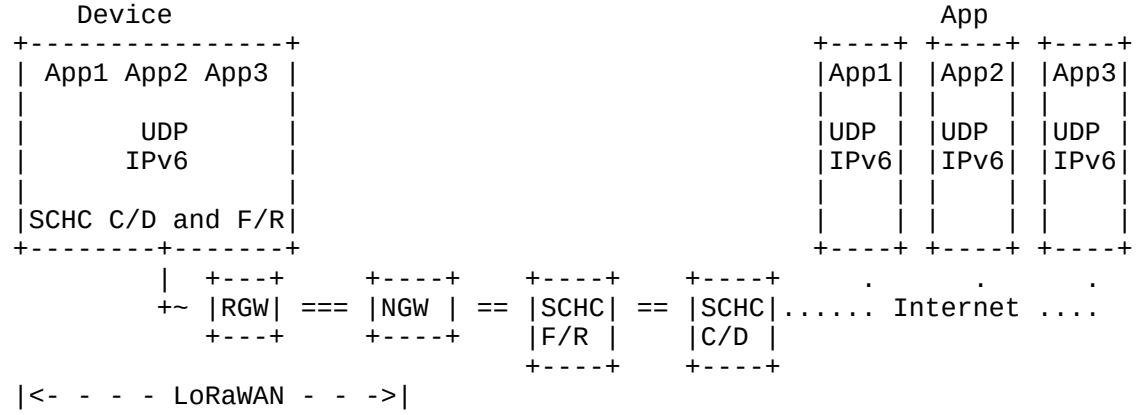


Figure 1: Architecture

Figure 1 represents the architecture for compression/decompression; it is based on the terminology from [RFC8376]. The device is sending application flows using IPv6 or IPv6/UDP protocols. These flows might be compressed by a SCHC C/D to reduce header size, and fragmented by the SCHC F/R. The resulting information is sent on a Layer 2 (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 a SCHC C/D for decompression. The SCHC C/D shares the same rules with the device. The SCHC C/D and SCHC F/R can be located on the 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 SCHC C/D. The SCHC C/D and SCHC 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 SCHC 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 a "Network Server", and the SCHC C/D and SCHC F/R are one or more "Application Servers". Application servers can be provided by the NGW or any third-party software. Figure 1 can be mapped in LoRaWAN terminology to:

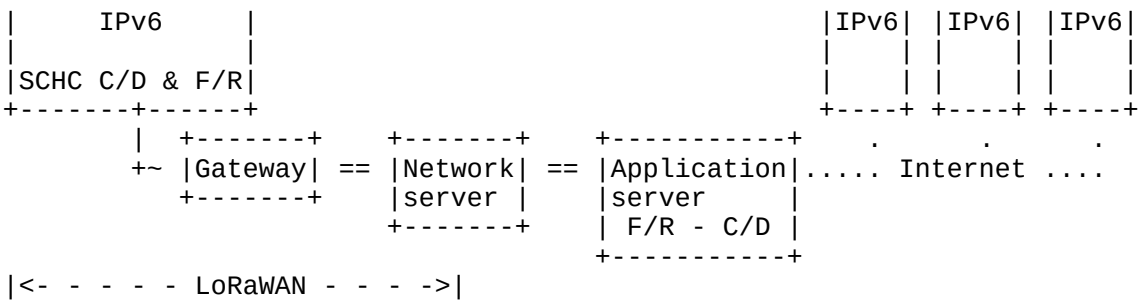


Figure 2: SCHC Architecture Mapped to LoRaWAN

4. LoRaWAN Architecture

An overview of the LoRaWAN protocol and architecture [LORAWAN-SPEC] is described in [RFC8376]. The mapping between the LPWAN architecture entities as described in [RFC8724] and the ones in [LORAWAN-SPEC] is as follows:

* 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.

* The RGW is the endpoint of the constrained link. This entity maps to the LoRaWAN Gateway.

* The NGW is the interconnection node between the Radio Gateway and the SCHC gateway (LoRaWAN Application Server). This entity maps to the LoRaWAN Network Server.

* The SCHC C/D and SCHC F/R are handled by the LoRaWAN Application Server.

* 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.

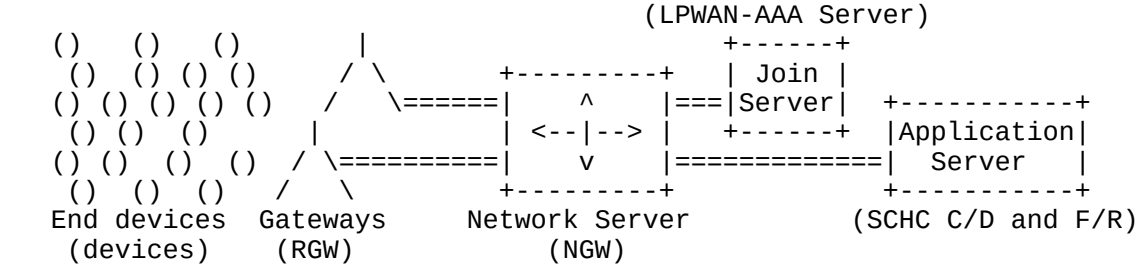


Figure 3: LPWAN Architecture

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

The SCHC C/D and SCHC F/R are performed on the LoRaWAN end device and the Application Server (called the SCHC gateway). While the point-to-point link between the device and the Application Server constitutes a single IP hop, the ultimate endpoint 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 Medium Access Control (MAC) layer supports three classes

of devices named A, B, and C. All devices implement Class A, and some devices may implement Class B or Class C. Class B and Class C are mutually exclusive.

Class A: 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 two 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. 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, as opposed to 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:

High periodicity: Downlinks from the NGW will be sent faster but the device wakes up more often and power consumption is increased.

Low periodicity: Downlinks from the NGW will have higher latency but lower power consumption.

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. For example, a 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.

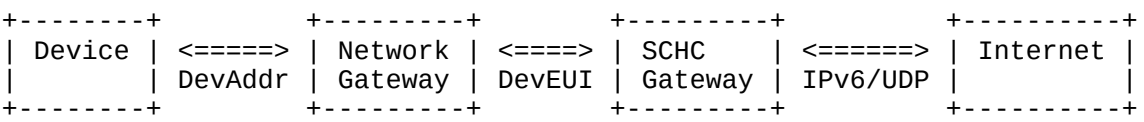


Figure 4: LoRaWAN Addresses

4.3. General Frame Types

LoRaWAN implements the possibility to send confirmed or unconfirmed

frames:

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

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 (FType = 0b100 as per [LORAWAN-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:

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.

JoinAccept: To onboard 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 (among other fields) the network's major settings and a random nonce used to derive the session keys.

Data: This refers to MAC and application data. Application data is protected with AES-128 encryption. MAC-related data is 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 multicast packet sent over a LoRaWAN radio link can be received by several devices. It is useful to address many devices with the same content: either a large binary file (firmware upgrade) or the 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]. The 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: The LoRa Alliance defined [LORAWAN-REMOTE-MULTICAST-SET] as the RECOMMENDED way to set up 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.

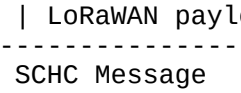


Figure 5: SCHC Message in LoRaWAN

| Note: The SCHC Message is any datagram sent by the 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 a "downlink-fragmented datagram". It uses another FPort for data downlink and its associated SCHC control uplinks, named "FPortDown" in this document.

All RuleIDs can use arbitrary values inside the FPort range allowed by the LoRaWAN specification [LORAWAN-SPEC] 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. RuleID Management

The RuleID MUST be 8 bits and 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 [LORAWAN-SPEC]; it implies that the 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:

* RuleID = 20 (8-bit) for uplink fragmentation, named FPortUp.

* RuleID = 21 (8-bit) for downlink fragmentation, named FPortDown.

* RuleID = 22 (8-bit) for which SCHC compression was not possible (i.e., no matching compression Rule was found), as described in Section 6 of [RFC8724].

The FPortUp value MUST be different from the FPortDown value. 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], implementations MUST implement the following algorithm and SHOULD use it.

1. key = LoRaWAN AppSKey

2. cmac = aes128_cmac(key, DevEUI)

3. IID = cmac[0..7]

The aes128_cmac algorithm is described in [RFC4493]. It has been chosen as it is already used by devices for the 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 concerns, for example, location tracking or correlation over time. Join periodicity is defined at the application level.

Address-scan risk is mitigated thanks to the entropy added to the IID by the inclusion of AppSKey.

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

Example with:

* DevEUI: 0x1122334455667788

* AppSKey: 0x00AABBCCDDEEFF00AABBCCDDEEFFAABB

1. key: 0x00AABBCCDDEEFF00AABBCCDDEEFFAABB
2. cmac: 0x4E822D9775B2649928F82066AF804FEC
3. IID: 0x4E822D9775B26499

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 the L2 level (i.e., triggering a LoRaWAN join). The way the device is rekeyed is out of scope of this document and left to the implementation.

> | Note: Implementations 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

The 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

Section 8.2.4 of [RFC8724] describes the possibility to interleave several fragmented SCHC datagrams for the same RuleID. This is not used in the 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 is the fragment receiver. A single fragmentation rule is defined. The SCHC F/R MUST concatenate FPort and LoRaWAN payload to retrieve the SCHC Packet, as per Section 5.1.

SCHC fragmentation reliability mode: "ACK-on-Error".

SCHC header size: 2 bytes (the FPort byte + 1 additional byte).

RuleID: 8 bits stored in the LoRaWAN FPort (cf. Section 5.2).

DTag: Size T = 0 bits, not used (cf. Section 5.6.1).

Window index: 4 windows are used, encoded on M = 2 bits.

FCN: The FCN field is encoded on N = 6 bits, so WINDOW_SIZE = 63 tiles are allowed in a window.

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. Implementations 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 an All-1 SCHC Message, the current L2 MTU MUST be big enough to fit the All-1 header and the last tile.

Penultimate tile: MUST be equal to the regular size.

RCS: Use the recommended calculation algorithm in Section 8.2.3 of [RFC8724], Integrity Checking.

Tile: Size is 10 bytes.

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.

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.

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:

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

* 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

Figure 7 is an example of a regular fragment for all fragments except the last one. SCHC Header Size is 16 Bits, including the LoRaWAN FPort.

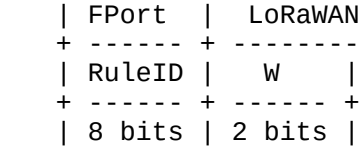


Figure 7: All Fragments Except the Last One.

5.6.2.2. Last Fragment (All-1)

Following figures are examples of All-1 messages. Figure 8 is without the last tile, Figure 9 is with the last tile.

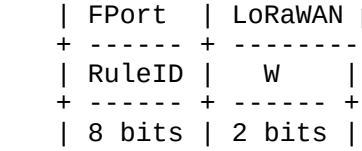
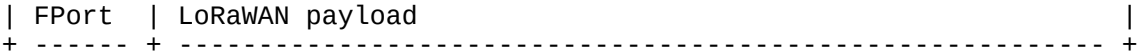


Figure 8: All-1 SCHC Message without Last Tile



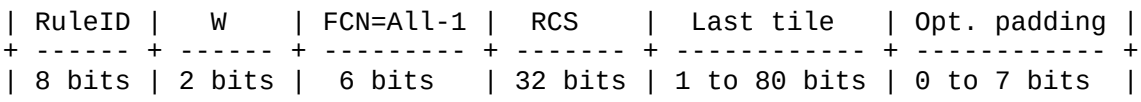


Figure 9: All-1 SCHC Message with Last Tile

5.6.2.3. SCHC ACK

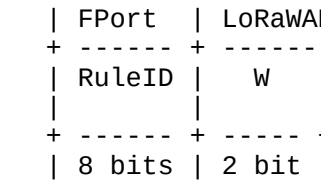


Figure 10: SCHC ACK Format - Correct RCS Check

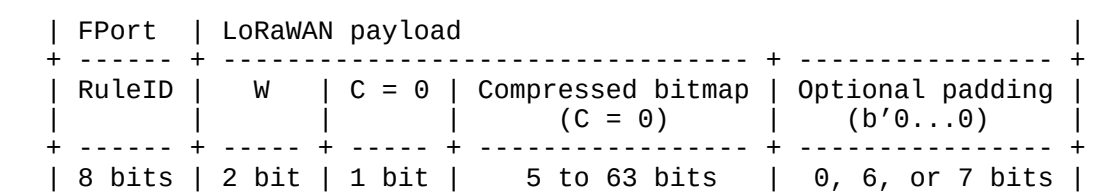


Figure 11: SCHC ACK Format - Incorrect RCS Check

> 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

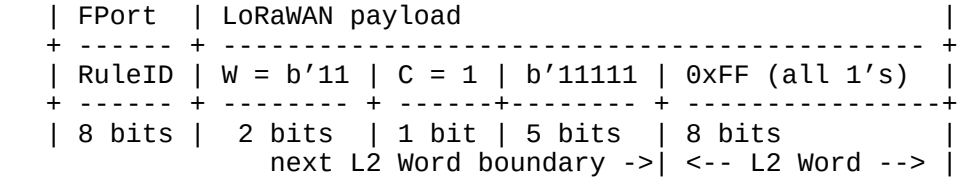


Figure 12: Receiver-Abort Format

5.6.2.5. SCHC Acknowledge Request

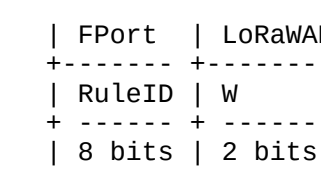


Figure 13: SCHC ACK REQ Format

5.6.3. Downlink Fragmentation: From SCHC Gateway to Device

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

RuleID: 8 bits stored in the 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 the recommended calculation algorithm in Section 8.2.3 of [RFC8724], 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 one 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 in Section 4.6. As this uplink is sent only to open an RX window, any LoRaWAN uplink frame from the device MAY reset this counter.

| Note: The FPending bit included in the LoRaWAN protocol SHOULD NOT be used for the 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

Figure 14 is an example of a regular fragment for all fragments except the last one. SCHC Header Size is 10 Bits, including the LoRaWAN FPort.

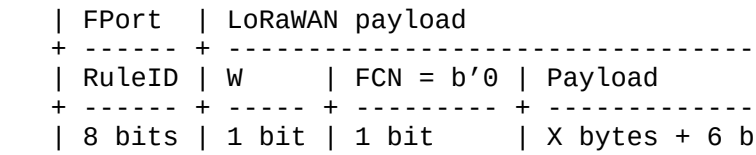


Figure 14: All Fragments but the Last One.

5.6.3.2. Last Fragment (All-1)

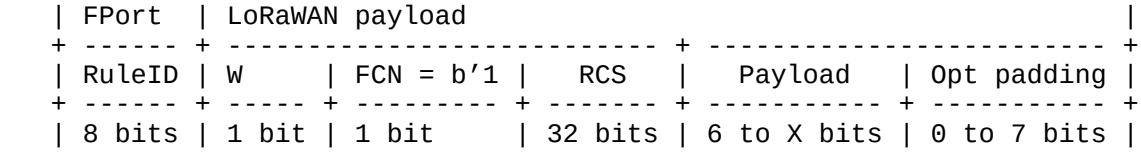


Figure 15: All-1 SCHC Message: The Last Fragment

5.6.3.3. SCHC ACK

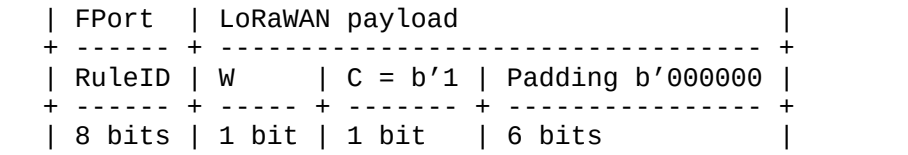


Figure 16: SCHC ACK Format - Correct RCS Check

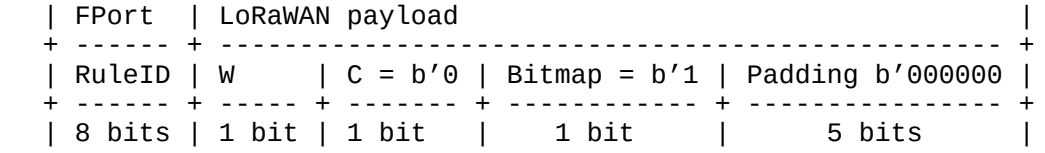


Figure 17: SCHC ACK Format - Incorrect RCS Check

5.6.3.4. Receiver-Abort

Figure 18 is an example of a Receiver-Abort packet, following an All-1 SCHC Fragment with incorrect RCS.

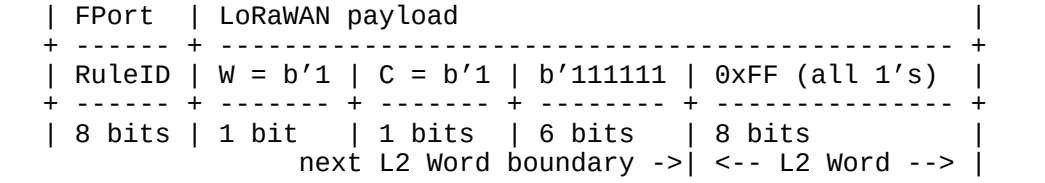


Figure 18: Receiver-Abort Packet

5.6.3.5. Downlink Retransmission Timer

Class A, Class B, and 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 (for 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 the 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 messages if no SCHC downlink has been received during the RX1 and RX2 windows. The 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, and 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 interval of 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 the opening of 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; this 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 values are:

Class B: 3 times the ping slot periodicity.

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", the following condition being met: SCHC header is 2 bytes.

Downlink fragmentation (ACK-Always):

As per [RFC8724], SCHC All-1 MUST contain the last tile, and implementations 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", the following condition being 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 the 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 (i.e., each join or rejoin to the LoRaWAN network).

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[LORAWAN-SPEC] LoRa Alliance, "LoRaWAN 1.0.4 Specification Package", <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. Ziga, "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>.

8.2. Informative References

[LORAWAN-REMOTE-MULTICAST-SET] LoRa Alliance, "LoRaWAN Remote Multicast Setup Specification v1.0.0", <https://lora-alliance.org/resource_hub/lorawan-remote-multicast-setup-specification-v1-0-0/>.

[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>.

Appendix A. Examples

In the 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 the SCHC compression layer. Rule 1 is used by the 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-byte FOpts are used by the 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 the SCHC compression layer. Rule 1 is used by the 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-byte FOpts are used by the LoRaWAN protocol: 11 bytes are available for SCHC payload + 1 byte FPort field. The SCHC header is 2 bytes (including FPort), so 1 tile is sent in the 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

The tile content is described in Figure 23

Content of the tile is:

RuleID	Compression residue	Payload

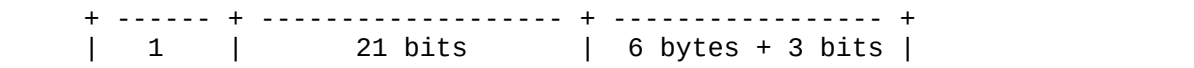


Figure 23: Uplink Example: First Tile Content

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

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

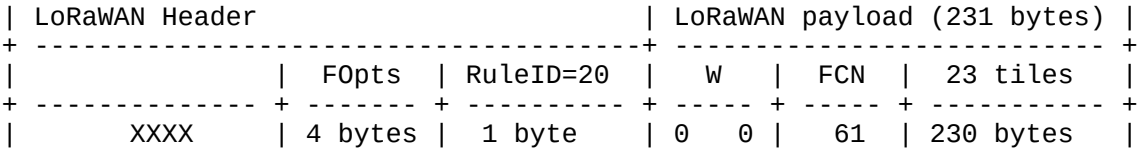


Figure 24: Uplink Example: LoRaWAN Packet 2

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.

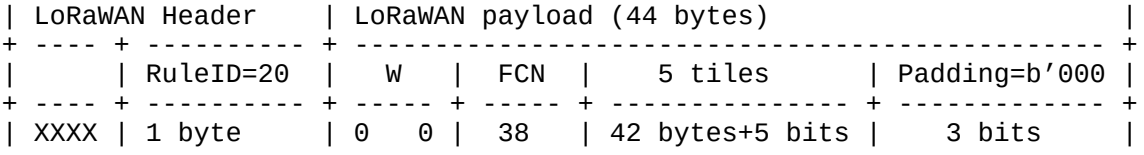


Figure 25: Uplink Example: LoRaWAN Packet 3

Then All-1 message can be transmitted:

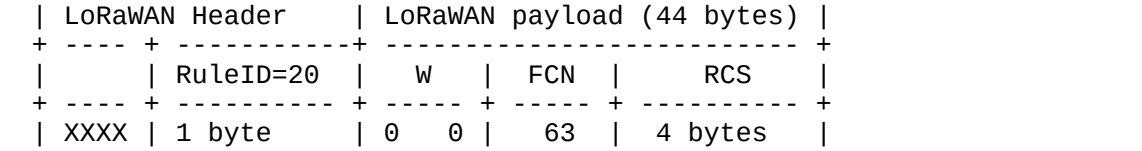


Figure 26: Uplink Example: LoRaWAN Packet 4 - All-1 SCHC Message

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:

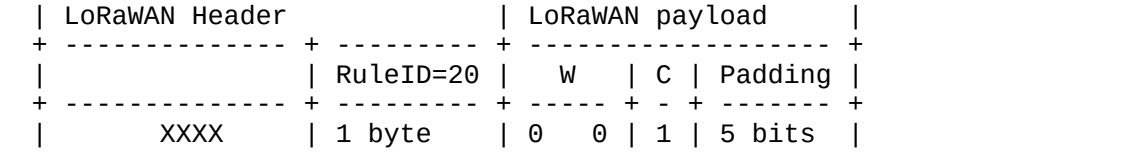


Figure 27: Uplink Example: LoRaWAN Packet 5 - SCHC ACK

A.3. Downlink

An applicative data of 155 bytes is passed to the SCHC compression layer. Rule 1 is used by the SCHC C/D layer, allowing to compress it to 130 bytes and 5 bits: 1 byte RuleID, 21 bits residue + 127 bytes payload.

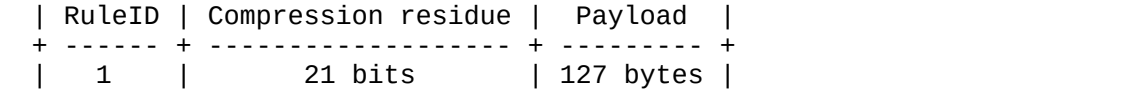


Figure 28: Downlink Example: SCHC Message

The current LoRaWAN MTU is 51 bytes; no FOpts are used by the LoRaWAN protocol: 51 bytes are available for SCHC payload + FPort field; the applicative data 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

The tile content is described in Figure 30

RuleID	Compression residue	Payload
1	21 bits	48 bytes and 1 bit

Figure 30: Downlink Example: First 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 a 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 bit	5 bits

Figure 34: Downlink Example: LoRaWAN Packet 5 - All-1 SCHC Message

The receiver answers to the sender with a SCHC ACK:

LoRaWAN Header	LoRaWAN payload			

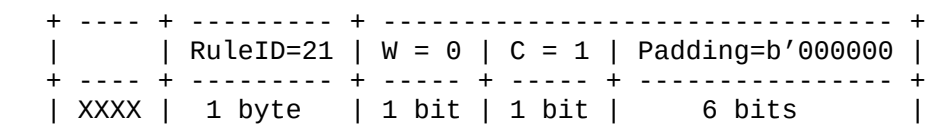


Figure 35: Downlink Example: LoRaWAN Packet 6 - SCHC ACK

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, and 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
Chirp Foundation

Email: nicolas.sornin@chirpfoundation.org

Alper Yegin
Actility

Email: alper.yegin@actility.com

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-Svign Cedex
France

Email: ivaylo@ackl.io