lpwan Working Group
Internet-Draft
Intended status: Standards Track
Expires: 26 April 2023

S. Aguilar
C. Gomez
R. Vidal
Universitat Politecnica de Catalunya

23 October 2022

SCHC Convergence Profile
draft-aguilar-lpwan-schc-convergence-00

Abstract

The present document defines a profile of Static Context Header Compression and fragmentation (SCHC) [RFC8724] for multi-radio devices or multi-network application. This profile can be used simultaneously over LoRaWAN, Sigfox, NB-IoT and any other technology that may use SCHC Fragmentation/Reassembly functionality.

Table of Contents

1. Introduction

The Static Context Header Compression and fragmentation (SCHC) specification [RFC8724] provides generic adaptation layer functionality, including Compression/Decompression (C/D) and Fragmentation and Reassembly (F/R) functionality. The latter offers three different modes, providing different features.

SCHC over LoRaWAN [RFC9011], SCHC over Sigfox [I-D.lpwan-schc-over-sigfox] and SCHC over NB-IoT [I-D.lpwan-schc-over-nbiot] are technology-specific SCHC profiles, which provide an optimal configuration of SCHC over the corresponding technologies. However, the F/R functionalities of these profiles are not compatible. Therefore, multi-radio devices (e.g., supporting

LoRaWAN, Sigfox and NB-IoT interfaces on the same device) require multiple implementations of the SCHC F/R sublayer, one for each technology.

Moreover, multi-network solutions, where the same application is deployed over different LPWAN technologies also require multiple implementations of the SCHC F/R sublayer, one for each deployment.

To reduce implementation complexity, and enable a single convergent F/R sublayer, this document provides the F/R details for a SCHC profile that can be used over all the LPWAN technologies overviewed in [RFC8376], leveraging the benefits of the Compound ACK. This profile can also be used over other technologies that may use SCHC Fragmentation/Reassembly functionality.

2. Terminology

It is assumed that the reader is familiar with the terms and mechanisms defined in [RFC8376] and in [RFC8724].

3. Motivation and Use Cases

3.1. Motivation

IoT applications running over LPWAN devices are tied up to the selected LPWAN technology. The LPWAN constrains influence the design of the IoT application itself. This presents problems when migrating to other LPWANs or networks, as it may imply redesigning the complete IoT application (from device code to backend code). The LPWAN, as a Layer 2 (L2), should be transparent to IoT application (and developers), as it is in the IP domain.

Current advances in the adoption of IPv6 over LPWAN achieved interoperability for application thanks to SCHC [RFC8724], and a single SCHC C/D sublayer. However, each LPWAN technology requires a different implementation of the SCHC F/R sublayer, with different (but actually very similar) F/R modes. Therefore, an IoT application using multiple LPWANs (multiple radios o multiple networks) will require multiple SCHC F/R implementation in device and backend code. This is not the case for the C/D sublayer.

To reduce code complexity and maintenance, and achieve a single convergent SCHC F/R sublayer, this document provides a SCHC Profile which considers the singularities of LoRaWAN, Sigfox and NB-IoT, while providing general F/R modes that work over all of these technologies simultaneously.

3.2. Use Cases

The SCHC over All profile has several use cases:

* Generic SCHC F/R Profile for implementation of SCHC to test over a new technology. SCHC out-of-the-box F/R modes.

* Multi-radio devices: Devices implementing more than one LPWAN radio.

* Multi-network applications: Applications deployed over more than one LPWAN.

* Network Redundancy:

  - Devices using another LPWAN as backup,

  - devices sending the same SCHC Fragment in different networks to increase the probability of successful fragmented packet reception.

* Increased device duty-cycle as more networks are available, e.g., if SCHC Packet transmission is not possible over LoRaWAN due to duty-cycle restriction, SCHC Packet transmission may be performed over Sigfox or NB-IoT. This applies also for SCHC Fragments.

* Devices sending SCHC Fragments over different LPWANs to check available coverage.

4. SCHC over All Profile

4.1. SCHC over All Architecture

[RFC8376] overviews the LoRaWAN, Sigfox, and NB-IoT protocols and their network architectures. More specifically, [RFC9011] maps the network architecture entities between LoRaWAN and LPWAN, as described in [RFC8724]. Similarly, [I-D.lpwan-schc-over-sigfox] and [I-D.lpwan-schc-over-nbiot] for Sigfox and NB-IoT performs the same mapping for Sigfox and NB-IoT, respectively.

Figure 1 shows the architecture when using several SCHC F/R implementations, one for each LPWAN technology. In this case, it is possible to send SCHC Packets over different LPWAN networks.

```
 ()   ()   ()       |  |
  ()  () () ()     / \/ \     +---------+   +---------+
 () () () () ()   / / \  \====| Network |===|SCHC over|===
  ()  () () ()                | Gateway |   |  NB-IoT | ||
 () () () () ()               | (NB-IoT)|   +---------+ ||
  ()  () () ()                +---------+               ||
  () () () ()                                           || +-----------+
 ()   ()   ()       |  |                                || |Application|
  ()  () () ()     / \/ \     +---------+   +---------+ || +-----------+
 () () () () ()   / / \  \====| Network |===|SCHC over|====|  SCHC C/D |
  ()  () () ()                |   Core  |   | Sigfox  | || +-----------+
 () () () () ()               | (Sigfox)|   +---------+ ||
  ()  () () ()                +---------+               ||
 () () () ()                                            ||
 ()   ()   ()       |  |      +---------+   +---------+ ||
  ()  () () ()     / \/ \     | Network |   |SCHC over| ||
 () () () ()      / / \  \====| Server  |===| LoRaWAN |===
  ()  () () ()                |(LoRaWAN)|   +---------+
 () () () ()                  +---------+
 End devices  Radio Gateways  Network Server SCHC C/D and F/R
   (devices)        (RGW)         (NGW)
```

Figure 1: Architecture when using several SCHC F/R implementations

Figure 2 presents the SCHC over All architecture, with a single SCHC C/D and F/R sublayer. This architecture provides a single implementation of the SCHC F/R sublayer.

```
   ()   ()   ()       |  |
    ()  () () ()     / \/ \     +---------+
   () () () () ()   / / \  \====| Network |==========
    ()  () () ()                | Gateway |        ||
   () () () () ()               | (NB-IoT)|        ||
    ()  () () ()                +---------+        ||
 () () () () ()                                +---------+
   ()   ()   ()       |  |                    |SCHC C/D |
    ()  () () ()     / \/ \     +---------+   |---------|  +-----------+
   () () () () ()   / / \  \====| Network |===|SCHC over|==|Application|
    ()  () () ()                |   Core  |   |   All   |  +-----------+
   () () () () ()               | (Sigfox)|   +---------+
    ()  () () ()                +---------+        ||
   () () () ()                                     ||
   ()   ()   ()       |  |                         ||
    ()  () () ()     / \/ \     +---------+        ||
   () () () () ()   / / \  \====| Network |==========
    ()  () () ()                |  Server |
   () () () () ()               |(LoRaWAN)|
    ()  () () ()                +---------+
   End devices   Radio Gateways  Network Server SCHC C/D and F/R
    (devices)        (RGW)         (NGW)            Server
```

Figure 2: SCHC over All architecture

In the SCHC over All Profile, as devices have a single SCHC F/R implementation, F/R RuleIDs are the same, independently of the LPWAN technology used, reducing the device memory and complexity requirements when compared to multiple SCHC F/R implementation.

4.2. Single SCHC ID

To simplify the access to RuleIDs and to converge the different device IDs provided by the networks involved, a device needs to have a new identifier called the single SCHC ID.

A device ID translation table maps the network device ID to single SCHC ID. Then, with the single Device ID, it is possible to look up the Rules set and identify the corresponding Rules for such device. This dissociates the network device ID form the Rules, allowing to use the same Rule set for the same device independently of the access network.

The network device IDs used by the LPWAN technologies included in this Profile are:

* LoRaWAN: DevID

* Sigfox: DeviceID

* NB-IoT: IMEI

Figure 3 presents a diagram of the SCHC over All architecture including the Single SCHC device ID translation table.

```
         +-----------+
         |   Table   |==========
         | Device ID |         ||
         |translation|         ||
         +-----------+         ||
                          +---------+
                          |SCHC C/D |
                          +---------+  +-----------+
  (from All LPWANs) ===|SCHC over|==|Application|
                          |   All   |  +-----------+
                          +---------+
```

Figure 3: Single SCHC device ID translation table diagram

4.3. Uplink Fragmentation

ACK-on-Error mode is RECOMMENDED for the transmission of Uplink SCHC Packets that require fragmentation and need to be sent reliably. ACK-on-Error mode is optimal, since it leads to a reduced number of ACKs in the lower capacity Downlink channel as Downlink messages can be sent asynchronously and opportunistically. Moreover, ACK-on-Error mode supports variable MTU (which is critical for changing from one LPWAN technology to another when sending SCHC Fragments spread across different LPWANs), and out-of-order delivery (in case SCHC Fragments are received out-of-order at the SCHC F/R receiver).

SCHC over LoRaWAN [RFC9011], SCHC over Sigfox [I-D.lpwan-schc-over-sigfox] and SCHC over NB-IoT [I-D.lpwan-schc-over-nbiot] provide uplink fragmentation SCHC profiles. At the SCHC Fragment level, these profiles are not compatible with one another. However, one of the SCHC over Sigfox uplink fragmentation modes (Two-bytes Option 2) has several similarities with the ACK-on-Error SCHC over LoRaWAN profile. Such similarities include:

* 2-byte SCHC Fragmentation Header size.

* 10-byte tile size.

* 2-byte Rule ID size.

* No DTag

Differences between the SCHC over LoRaWAN and SCHC over Sigfox (Two-byte Option 2) uplink fragmentation profiles include:

* WINDOW_SIZE (tiles per window).

* M size (maximum number of windows).

* N size (tiles per window).

* Different RCS size and algoritm.

SCHC over LoRaWAN ACK-on-Error includes a WINDOW_SIZE of 64 tiles. This allows feedback from receiver to sender with larger ACKs. Larger ACKs provide better performance in error-prone environments.

On the other hand, SCHC over Sigfox leverages the Compound ACK with a WINDOW_SIZE of 32, allowing more downlink opportunities, and enabling larger ACKs, notifying more than one window, in error-prone environments and smaller ACKs, notifying one window.

Therefore, the SCHC over All Profile uses smaller WINDOW_SIZE values than the ones proposed in SCHC over LoRaWAN [RFC9011], as it uses the Compound ACK to accomplish larger ACK size, while still having the option of smaller ACKs and more downlink opportunities.

In error-prone environments, larger ACKs pool more fragment error in a single ACK, reducing the total number of ACKs, compared to the increase in ACK size. Smaller ACKs performed better when error are scatter, as ACKs will be small and less frequent.

4.3.1. Uplink ACK-on-Error Mode: Two-byte SCHC Header

In order to take advance of the similarities of the different LPWAN profiles, the SCHC Uplink Fragmentation Header size is RECOMMENDED to have a size of 16 bits and be composed as follows:

* Rule ID size is: 8 bits

* DTag size (T) is: 0 bits

* Window index (W) size (M): 3 bits

* Fragment Compressed Number (FCN) size (N): 5 bits.

* MAX_ACK_REQUESTS: 5

* WINDOW_SIZE: 31 (with a maximum value of FCN=0b1011)

* Regular tile size: 10 bytes

* All-1 tile size: 1 to 10 bytes

* Retransmission Timer: Application-dependent. The RECOMMENDED value is 12 hours.

* Inactivity Timer: Application-dependent. The RECOMMENDED value is 12 hours.

* RCS size: 32 bits

4.3.2. Downlink Consideration in Uplink Fragmentation

When fragmentation is performed in the Uplink, the Compound ACK allows to optimally manage receiver acknowledgements, as the number of windows and the moment the Compound ACK is transferred can be freely selected, e.g., depending on network conditions or capacity. This advantage, compared with [RFC8724] and [RFC9011], benefits smaller windows sizes, as smaller windows sizes provide more downlink opportunities than a larger windows for the same number of tiles.

4.4. Rule Management

The RuleID MUST be 8 bits. In LoRaWAN it MUST be encoded in the LoRaWAN FPort.

4.5. SCHC over All F/R Message Formats

This section depicts the different formats of SCHC Fragment, SCHC ACK (including the SCHC Compound ACK defined in [I-D.ietf-lpwan-schc-compound-ack]), SCHC Aborts and ACK Request used in SCHC over All Uplink ACK-on-Error mode.

4.5.1. Regular SCHC Fragment

Figure 4 shows an example of a regular SCHC fragment for all fragments except the last one. The penultimate tile of a SCHC Packet is of the regular size.

RuleID	W	FCN	Payload
8 bits	3 bits	5 bits	Variable

Figure 4: Regular SCHC Fragment

4.5.2. All-1 SCHC Fragment

```
+ ------ + ---------------------------- +
| RuleID |   W    | FCN=All-1 |  RCS    |
+ ------ + ------ + --------- + ------- +
| 8 bits | 3 bits | 5 bits    | 32 bits |
```

Figure 5: All-1 SCHC Fragment (no tile)

```
+ ------ + ---------------------------------------------------------- +
| RuleID |   W    | FCN=All-1 |  RCS    |  Last tile   | Opt. padding |
+ ------ + ------ + --------- + ------- + ------------ + ------------ +
| 8 bits | 3 bits |  5 bits   | 32 bits | 1 to X bits  | 0 to 7 bits
```

Figure 6: All-1 SCHC Fragment (with tile)

4.5.3. SCHC ACK Format

```
+ ------ + --------------------------+
| RuleID |   W    | C = 1 |  padding |
+ ------ + ----- + ----- + --------- +
| 8 bits | 3 bit | 1 bit |  X bits   |
```

Figure 7: Successful SCHC ACK

```
| 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  |


|-- SCHC ACK Header --|- W=w1 -|...|---- W=wi -----|
+------+------+-------+--------+...+------+--------+------+-------+
|RuleID|W=b'w1| C=b'0 | Bitmap |...|W=b'wi| Bitmap | 000  |b'0-pad|
+------+------+-------+--------+...+------+--------+------+-------+
|8 bits|3 bits| 1 bit | 31 bits|...|3 bits| 31 bits|3 bits|
```

Losses are found in windows W = w1,...,wi; where w1<w2<...<wi

Figure 8: Failure SCHC ACK

4.5.4. SCHC Receiver-Abort Message

```
|-- Receiver-Abort Header -|
+---------------------------------+-----------------+---------+
| RuleID | W=b'111 | C=b'1 | b'1111 |  0xFF (all 1's) | b'0-pad |
+--------+---------+-------+--------+-----------------+---------+
| 8 bits |  3 bits | 1 bit | 4 bit  |  8 bit          |  X bits |
           next L2 Word boundary ->| <-- L2 Word -->  |
```

Figure 9: SCHC Receiver-Abort

4.5.5. SCHC Sender-Abort Messages

```
|---- Sender-Abort Header ----|
+-----------------------------+
| RuleID |    W    | FCN=ALL-1 |
+--------+---------+-----------+
| 8 bits | 3 bits |  5 bits    |
```

Figure 10: SCHC Sender-Abort

4.5.6. SCHC ACK Request

```
|------- ACK Request Header -------|
+------- +------------------------ +
| RuleID |    W    | FCN = b'00000  |
+ ------ + ------ + -------------- +
| 8 bits | 3 bits | 5 bits         |
```

Figure 11: SCHC ACK Request

5. Acknowledgements

Carles Gomez has been funded in part by the Spanish Government through the TEC2016-79988-P grant, and the PID2019-106808RA-I00 grant (funded by MCIN / AEI / 10.13039/501100011033), and by Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya 2017 through grant SGR 376.

Sergio Aguilar has been funded by the ERDF and the Spanish Government through project TEC2016-79988-P and project PID2019-106808RA-I00, AEI/FEDER, EU (funded by MCIN / AEI / 10.13039/501100011033).

6. Normative References

[I-D.ietf-lpwan-schc-compound-ack]
Zuniga, JC., Gomez, C., Aguilar, S., Toutain, L., Cespedes, S., and D. Wistuba, "SCHC Compound ACK", Work in Progress, Internet-Draft, draft-ietf-lpwan-schc-compound-ack-07, October 2022, <http://www.ietf.org/internet-drafts/draft-ietf-lpwan-schc-compound-ack-07.txt>.

[I-D.lpwan-schc-over-nbiot]
Ramos, E. and A. Minaburo, "SCHC over NBIoT", Work in Progress, Internet-Draft, draft-ietf-lpwan-schc-over-nbiot-12, October 2022, <http://www.ietf.org/internet-drafts/draft-ietf-lpwan-schc-over-nbiot-12.txt>.

[I-D.lpwan-schc-over-sigfox]
Zuniga, JC., Gomez, C., Aguilar, S., Toutain, L., Cespedes, S., Wistuba, D., and J. Boite, "SCHC over Sigfox LPWAN", Work in Progress, Internet-Draft, draft-ietf-lpwan-schc-over-sigfox-13, October 2022, <http://www.ietf.org/internet-drafts/draft-ietf-lpwan-schc-over-sigfox-13.txt>.

[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>.

[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>.

[RFC9011] Gimenez, O., Ed. and I. Petrov, Ed., "Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN", RFC 9011, DOI 10.17487/RFC9011, April 2021, <https://www.rfc-editor.org/info/rfc9011>.

Authors' Addresses

Sergio Aguilar
Universitat Politecnica de Catalunya
C/Esteve Terradas, 7
08860 Castelldefels
Spain
Email: sergio.aguilar.romero@upc.edu

Carles Gomez
Universitat Politecnica de Catalunya
C/Esteve Terradas, 7
08860 Castelldefels
Spain
Email: carles.gomez@upc.edu

Rafael Vidal
Universitat Politecnica de Catalunya
C/Esteve Terradas, 7
08860 Castelldefels
Spain
Email: rafael.vidal@upc.edu

LPWAN Working Group
Internet-Draft
Intended status: Informational
Expires: 1 January 2023

A. Pelov
Acklio
P. Thubert
Cisco Systems
A. Minaburo
Acklio

30 June 2022

LPWAN Static Context Header Compression (SCHC) Architecture
draft-ietf-lpwan-architecture-02

Abstract

This document defines the LPWAN SCHC architecture.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IETF LPWAN WG defined the necessary operations to enable IPv6 over selected Low-Power Wide Area Networking (LPWAN) radio technologies. [rfc8376] presents an overview of those technologies.

The Static Context Header Compression (SCHC) [rfc8724] technology is the core product of the IETF LPWAN working group. [rfc8724] defines a generic framework for header compression and fragmentation, based on a static context that is pre-installed on the SCHC endpoints.

This document details the constitutive elements of a SCHC-based solution, and how the solution can be deployed. It provides a general architecture for a SCHC deployment, positioning the required specifications, describing the possible deployment types, and indicating models whereby the rules can be distributed and installed to enable reliable and scalable operations.

2. LPWAN Technologies and Profiles

Because LPWAN technologies [rfc8376] have strict yet distinct constraints, e.g., in terms of maximum frame size, throughput, and/or directionality, a SCHC instance must be profiled to adapt to the specific necessities of the technology to which it is applied.

Appendix D. "SCHC Parameters" of [rfc8724] lists the information that an LPWAN technology-specific document must provide to profile SCHC for that technology.

As an example, [rfc9011] provides the SCHC profile for LoRaWAN networks.

3. The Static Context Header Compression

SCHC [rfc8724] specifies an extreme compression capability based on a state that must match on the compressor and decompressor side. This state comprises a set of Compression/Decompression (C/D) rules.

The SCHC Parser analyzes incoming packets and creates a list of fields that it matches against the compression rules. The rule that matches best is used to compress the packet, and the rule identifier (RuleID) is transmitted together with the compression residue to the decompressor. Based on the RuleID and the residue, the decompressor can rebuild the original packet and forward it in its uncompressed form over the Internet.

[rfc8724] also provides a Fragmentation/Reassembly (F/R) capability to cope with the maximum and/or variable frame size of a Link, which is extremely constrained in the case of an LPWAN network.

If a SCHC-compressed packet is too large to be sent in a single Link-Layer PDU, the SCHC fragmentation can be applied on the compressed packet. The process of SCHC fragmentation is similar to that of compression; the fragmentation rules that are programmed for this Device are checked to find the most appropriate one, regarding the SCHC packet size, the link error rate, and the reliability level required by the application.

The ruleID allows to determine if it is a compression or fragmentation rule.

4. SCHC Applicability

4.1. LPWAN Overview

4.2. Compressing Serial Streams

[rfc8724] was defined to compress IPv6 [rfc8200] and UDP; but SCHC really is a generic compression and fragmentation technology. As such, SCHC is agnostic to which protocol it compresses and at which layer it is operated. The C/D peers may be hosted by different entities for different layers, and the F/R operation may also be performed between different parties, or different sub-layers in the same stack, and/or managed by different organizations.

If a protocol or a layer requires additional capabilities, it is always possible to document more specifically how to use SCHC in that context, or to specify additional behaviours. For instance, [rfc8824] extends the compression to CoAP [RFC7252] and OSCORE [RFC8613].

4.3. Example: Goose and DLMS

5. SCHC Architecture

5.1. SCHC Endpoints

Section 3 of [rfc8724] depicts a typical network architecture for an LPWAN network, simplified from that shown in [rfc8376] and reproduced in Figure 1.

```
 ()   ()   ()       |
  ()  () () ()     / \       +---------+
() () () () () () /   \======|    ^    |             +-----------+
 ()  ()   ()     |           | <--|--> |             |Application|
()  ()  ()  ()  / \==========|    v    |=============|   Server  |
  ()  ()  ()   /   \         +---------+             +-----------+
 Dev           RGWs             NGW                     App
```

Figure 1: Typical LPWAN Network Architecture

Typically, an LPWAN network topology is star-oriented, which means that all packets between the same source-destination pair follow the same path from/to a central point. In that model, highly constrained Devices (Dev) exchange information with LPWAN Application Servers (App) through a central Network Gateway (NGW), which can be powered and is typically a lot less constrained than the Devices. Because Devices embed built-in applications, the traffic flows to be compressed are known in advance and the location of the C/D and F/R functions (e.g., at the Dev and NGW), and the associated rules, can be pre provisioned in the system before use.

The SCHC operation requires a shared sense of which SCHC Device is Uplink (Dev to App) and which is Downlink (App to Dev), see [rfc8376]. In a star deployment, the hub is always considered Uplink and the spokes are Downlink. The expectation is that the hub and spoke derive knowledge of their role from the network configuration and SCHC does not need to signal which is hub thus Uplink vs. which is spoke thus Downlink. In other words, the link direction is determined from extrinsic properties, and is not advertised in the protocol.

Nevertheless, SCHC is very generic and its applicability is not limited to star-oriented deployments and/or to use cases where applications are very static and the state provisioned in advance. In particular, a peer-to-peer (P2P) SCHC Instance (see Section 5.2) may be set up between peers of equivalent capabilities, and the link direction cannot be inferred, either from the network topology nor from the device capability.

In that case, by convention, the device that initiates the donnection that sustains the SCHC Instance is considered as being Downlink, IOW it plays the role of the Dev in [rfc8724].

This convention can be reversed, e.g., by configuration, but for proper SCHC operation, it is required that the method used ensures that both ends are aware of their role, and then again this determination is based on extrinsic properties.

5.2. SCHC Instances

[rfc8724] defines a protocol operation between a pair of peers. A session called a SCHC Instance is established and SCHC maintains a state and timers associated to that Instance.

When the SCHC Device is a highly constrained unit, there is typically only one Instance for that Device, and all the traffic from and to the device is exchanged with the same Network Gateway. All the traffic can thus be implicitly associated with the single Instance that the device supports, and the Device does not need to manipulate the concept. For that reason, SCHC avoids to signal explicitly the Instance identification in its data packets.

The Network Gateway, on the other hand, maintains multiple Instances, one per SCHC Device. The Instance is derived from the lower layer, typically the source of an incoming SCHC packet. The Instance is used in particular to select from the rule database the set of rules that apply to the SCHC Device, and the current state of their exchange, e.g., timers and previous fragments.

This architecture generalizes the model to any kind of peers. In the case of more capable devices, a SCHC Device may maintain more than one Instance with the same peer, or a set of different peers. Since SCHC does not signal the Instance in its packets, the information must be derived from a lower layer point to point information. For instance, the SCHC session can be associated one-to-one with a tunnel, a TLS session, or a TCP or a PPP connection.

For instance, [I-D.thubert-intarea-schc-over-ppp] describes a type of deployment where the C/D and/or F/R operations are performed between peers of equal capabilities over a PPP [rfc2516] connection. SCHC over PPP illustrates that with SCHC, the protocols that are compressed can be discovered dynamically and the rules can be fetched on-demand by both parties from the same Uniform Resource Name (URN) [rfc8141], ensuring that the peers use the exact same set of rules.

```
+----------+  Wi-Fi /   +----------+                ....
|    IP    |  Ethernet  |    IP    |            ..          )
|   Host   +-----/------+  Router  +----------(   Internet   )
| SCHC C/D |  Serial    | SCHC C/D |            (          )
+----------+            +----------+                 ...
            <-- SCHC -->
              over PPP
```

Figure 2: PPP-based SCHC Deployment

In that case, the SCHC Instance is derived from the PPP connection. This means that there can be only one Instance per PPP connection, and that all the flow and only the flow of that Instance is exchanged within the PPP connection.

5.3. Layering with SCHC Instances

[rfc8724] states that a SCHC instance needs the rules to process C/D and F/R before the session starts, and that rules cannot be modified during the session.

As represented figure Figure 3, the compression of the IP and UDP headers may be operated by a network SCHC instance whereas the end-to-end compression of the application payload happens between the Device and the application. The compression of the application payload may be split in two instances to deal with the encrypted portion of the application PDU. Fragmentation applies before LPWAN transportation layer.

```
       (Device)            (NGW)                              (App)

       +--------+                                           +--------+
 A S   |  CoAP  |                                           |  CoAP  |
 p C   |  inner |                                           |  inner |
 p H   +--------+                                           +--------+
 . C   |  SCHC  |                                           |  SCHC  |
       |  inner |   cryptographical boundary                |  inner |
-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._
 A S   |  CoAP  |                                           |  CoAP  |
 p C   |  outer |                                           |  outer |
 p H   +--------+                                           +--------+
 . C   |  SCHC  |                                           |  SCHC  |
       |  outer |   layer / functional boundary             |  outer |
-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._.-._
 N     .  UDP   .                                           .  UDP   .
 e     ..........     ..................                    ..........
 t     .  IPv6  .     .      IPv6      .                    .  IPv6  .
 w S   ..........     ..................                    ..........
 o C   .SCHC/L3 .     . SCHC/L3.       .                    .        .
 r H   ..........     ..........       .                    .        .
 k C   .  LPWAN .     . LPWAN  .       .                    .        .
       ..........     ..................                    ..........
           ((((LPWAN))))             ------   Internet  ------
```

Figure 3: Different SCHC instances in a global system

This document defines a generic architecture for SCHC that can be used at any of these levels. The goal of the architectural document is to orchestrate the different protocols and data model defined by the LPWAN working group to design an operational and interoperable framework for allowing IP application over contrained networks.

6. SCHC Data Model

A SCHC instance, summarized in the Figure 4, implies C/D and/or F/R present in both end and that both ends are provisioned with the same set of rules.

```
         (-------)                                (-------)
         ( Rules )                                ( Rules )
         (-------)                                (-------)
          . read                                   . read
          .                                        .
         +-------+                                +-------+
     <===| R & D |<===                        <===| C & F |<===
     ===>| C & F |===>                        ===>| R & D |===>
         +-------+                                +-------+
```

Figure 4: Summarized SCHC elements

A common rule representation that expresses the SCHC rules in an interoperable fashion is needed yo be able to provision end-points from different vendors To that effect, [I-D.ietf-lpwan-schc-yang-data-model] defines a rule representation using the YANG [rfc7950] formalism.

[I-D.ietf-lpwan-schc-yang-data-model] defines an YANG data model to represent the rules. This enables the use of several protocols for rule management, such as NETCONF[RFC6241], RESTCONF[RFC8040], and CORECONF[I-D.ietf-core-comi]. NETCONF uses SSH, RESTCONF uses HTTPS, and CORECONF uses CoAP(s) as their respective transport layer protocols. The data is represented in XML under NETCONF, in JSON[RFC8259] under RESTCONF and in CBOR[RFC8949] under CORECONF.

```
                   create
         (-------)  read   +=======+ *
         ( rules )<------->|Rule   |<--|-------->
         (-------)  update |Manager|   NETCONF, RESTCONF or CORECONF
            . read  delete +=======+   request
            .
         +-------+
     <===| R & D |<===
     ===>| C & F |===>
         +-------+
```

Figure 5: Summerized SCHC elements

The Rule Manager (RM) is in charge of handling data derived from the YANG Data Model and apply changes to the rules database Figure 5.

The RM is an Application using the Internet to exchange information, therefore:

* for the network-level SCHC, the communication does not require routing. Each of the end-points having an RM and both RMs can be viewed on the same link, therefore wellknown Link Local addresses

can be used to identify the Device and the core RM. L2 security MAY be deemed as sufficient, if it provides the necessary level of protection.

* for application-level SCHC, routing is involved and global IP addresses SHOULD be used. End-to-end encryption is RECOMMENDED.

Management messages can also be carried in the negotiation protocol as proposed in [I-D.thubert-intarea-schc-over-ppp]. The RM traffic may be itself compressed by SCHC: if CORECONF protocol is used, [rfc8824] can be applied.

7. SCHC Device Lifecycle

In the context of LPWANs, the expectation is that SCHC rules are associated with a physical device that is deployed in a network. This section describes the actions taken to enable an autimatic commissioning of the device in the network. SCHC

7.1. Device Development

The expectation for the development cycle is that message formats are documented as a data model that is used to generate rules. Several models are possible:

1. In the application model, an interface definition language and binary communication protocol such as Apache Thrift is used, and the serialization code includes the SCHC operation. This model imposes that both ends are compiled with the generated structures and linked with generated code that represents the rule operation.

2. In the device model, the rules are generated separately. Only the device-side code is linked with generated code. The Rules are published separately to be used by a generic SCHC engine that operates in a middle box such as a SCHC gateway.

3. In the protocol model, both endpoint generate a packet format that is imposed by a protocol. In that case, the protocol itself is the source to generate the Rules. Both ends of the SCHC compression are operated in middle boxes, and special attention must be taken to ensure that they operate on the compatible Rule sets, basically the same major version of the same Rule Set.

Depending on the deployment, the tools thar generate the Rules should provide knobs to optimize the Rule set, e.g., more rules vs. larger residue.

7.2. Rules Publication

In the device model and in the protocol model, at least one of the endpoints must obtain the rule set dynamically. The expectation is that the Rule Sets are published to a reachable repository and versionned (minor, major). Each rule set should have its own Uniform Resource Names (URN) [RFC8141] and a version.

The Rule Set should be authenticated to ensure that it is genuine, or obtained from a trusted app store. A corrupted Rule Set may be used for multiple forms of attacks, more in Section 8.

7.3. SCHC Device Deployment

The device and the network should mutually authenticate themselves. The autonomic approach [RFC8993] provides a model to achieve this at scale with zero touchn, in networks where enough bandwidth and compute are available. In highly constrained networks, one touch is usually necessary to program keys in the devices.

The initial handshake between the SCHC endpoints should comprise a capability exchange whereby URN and the version of the rule set are obtained or compared. SCHC may not be used if both ends can not agree on an URN and a major version. Manufacturer Usage Descriptions (MUD) [RFC8520] may be used for that purpose in the device model.

Upon the handshake, both ends can agree on a rule set, their role when the rules are asymmetrical, and fetch the rule set if necessary. Optionally, a node that fetwhed a rule set may inform the other end that it is reacy from transmission.

7.4. SCHC Device Maintenance

URN update without device update (bug fix) FUOTA => new URN => reprovisioning

7.5. SCHC Device Decommissionning

Signal from device/vendor/network admin

8. Security Considerations

SCHC is sensitive to the rules that could be abused to form arbitrary long messages or as a form of attack against the C/D and/or F/R functions, say to generate a buffer overflow and either modify the Device or crash it. It is thus critical to ensure that the rules are distributed in a fashion that is protected against tempering, e.g., encrypted and signed.

9. IANA Consideration

This document has no request to IANA

10. Acknowledgements

The authors would like to thank (in alphabetic order):

11. References

11.1. Normative References

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

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

[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>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/info/rfc8520>.

[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>.

[rfc8824] Minaburo, A., Toutain, L., and R. Andreasen, "Static Context Header Compression (SCHC) for the Constrained Application Protocol (CoAP)", RFC 8824, DOI 10.17487/RFC8824, June 2021, <https://www.rfc-editor.org/info/rfc8824>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <https://www.rfc-editor.org/info/rfc8993>.

[rfc9011] Gimenez, O., Ed. and I. Petrov, Ed., "Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN", RFC 9011, DOI 10.17487/RFC9011, April 2021, <https://www.rfc-editor.org/info/rfc9011>.

11.2. Informative References

[I-D.ietf-core-comi]
Veillette, M., Stok, P. V. D., Pelov, A., Bierman, A., and I. Petrov, "CoAP Management Interface (CORECONF)", Work in Progress, Internet-Draft, draft-ietf-core-comi-11, 17 January 2021, <https://www.ietf.org/archive/id/draft-ietf-core-comi-11.txt>.

[I-D.ietf-lpwan-schc-yang-data-model]
Minaburo, A. and L. Toutain, "Data Model for Static Context Header Compression (SCHC)", Work in Progress, Internet-Draft, draft-ietf-lpwan-schc-yang-data-model-12, 25 May 2022, <https://www.ietf.org/archive/id/draft-ietf-lpwan-schc-yang-data-model-12.txt>.

[I-D.thubert-intarea-schc-over-ppp]
Thubert, P., "SCHC over PPP", Work in Progress, Internet-Draft, draft-thubert-intarea-schc-over-ppp-03, 21 April 2021, <https://www.ietf.org/archive/id/draft-thubert-intarea-schc-over-ppp-03.txt>.

[rfc2516] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., and R. Wheeler, "A Method for Transmitting PPP Over Ethernet (PPPoE)", RFC 2516, DOI 10.17487/RFC2516, February 1999, <https://www.rfc-editor.org/info/rfc2516>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

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

[rfc7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[rfc8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

[rfc8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[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>.

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/info/rfc8613>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

Authors' Addresses

Alexander Pelov
Acklio
1137A avenue des Champs Blancs
35510 Cesson-Sevigne Cedex
France
Email: a@ackl.io

Pascal Thubert
Cisco Systems
45 Allee des Ormes - BP1200
06254 Mougins - Sophia Antipolis
France
Email: pthubert@cisco.com

Ana Minaburo
Acklio
1137A avenue des Champs Blancs
35510 Cesson-Sevigne Cedex
France
Email: ana@ackl.io