

Workgroup: lpwan Working Group
Internet-Draft:
draft-ietf-lpwan-schc-compound-ack-02
Published: 10 December 2021
Intended Status: Standards Track
Expires: 13 June 2022
Authors: JC. Zuniga C. Gomez
 SIGFOX Universitat Politècnica de Catalunya
 S. Aguilar
 Universitat Politècnica de Catalunya
 L. Toutain S. Céspedes
 IMT-Atlantique NIC Labs, Universidad de Chile
 D. Wistuba
 NIC Labs, Universidad de Chile
SCHC Compound ACK

Abstract

The present document describes an extension to the SCHC (Static Header Compression and Fragmentation) protocol [RFC8724]. It defines a SCHC Compound ACK message format, which is intended to reduce the number of downlink transmissions (i.e., SCHC ACKs) by accumulating bitmaps of several windows in a single SCHC message (i.e., the SCHC Compound ACK).

The message format is generic, and can be used, for instance, by any of the four LWPAN technologies defined in [RFC8376], being Sigfox, LoRaWAN, NB-IoT and IEEE 802.15.4w.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 June 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Terminology](#)
- [3. SCHC Compound ACK](#)
 - [3.1. SCHC Compound ACK Message Format](#)
 - [3.2. SCHC Compound ACK Behaviour](#)
 - [3.2.1. Sender Behaviour](#)
 - [3.2.2. Receiver Behaviour](#)
 - [3.3. SCHC Compound ACK Examples](#)
 - [3.4. SCHC Compound ACK YANG Data Model](#)
 - [3.4.1. SCHC YANG Data Model Extension](#)
 - [3.4.2. SCHC YANG Tree Extension](#)
- [4. SCHC Compound ACK Parameters](#)
- [5. Security considerations](#)
- [6. Acknowledgements](#)
- [7. Normative References](#)
- [Authors' Addresses](#)

1. Introduction

The Generic Framework for Static Context Header Compression and Fragmentation (SCHC) specification [[RFC8724](#)] describes two mechanisms: i) a protocol header compression scheme, and ii) a frame fragmentation and loss recovery functionality. Either can be used on top of radio technologies such as the four LPWAN defined in [[RFC8376](#)], being Sigfox, LoRaWAN, NB-IoT and IEEE 802.15.4w. These LPWANs have similar characteristics such as star-oriented topologies, network architecture, connected devices with built-in applications, etc.

SCHC offers a great level of flexibility to accommodate all these LPWAN technologies. Even though there are a great number of similarities between them, some differences exist with respect to

the transmission characteristics, payload sizes, etc. Hence, there are optimal parameters and modes of operation that can be used when SCHC is used on top of a specific LPWAN technology.

The present document describes an extension to the SCHC protocol. It defines a SCHC Compound ACK format, which is intended to reduce the number of downlink transmissions (i.e., SCHC ACKs) in the ACK-on-Error mode of SCHC. The SCHC Compound ACK extends the SCHC ACK message format so that it can contain several bitmaps, each bitmap being identified by its corresponding window number.

The SCHC Compound ACK:

- *provides feedback only for windows with fragment losses,
- *has a variable size that depends on the number of windows with fragment losses being reported in the single Compound SCHC ACK,
- *includes the window number (i.e., W) of each bitmap,
- *has the same SCHC ACK format defined in [[RFC8724](#)] when only one window with losses is reported,
- *might not cover all windows with fragment losses of a SCHC Packet,
- *and is distinguishable from the SCHC Receiver-Abort.

2. Terminology

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

3. SCHC Compound ACK

The SCHC Compound ACK is a SCHC ACK message that can contain several bitmaps, each bitmap being identified by its corresponding window number.

The SCHC Compound ACK groups the window number (W) with its corresponding bitmap. Windows do not need to be contiguous. However, the window numbers and corresponding bitmaps included in the SCHC Compound ACK message MUST be ordered from the lowest-numbered to the highest-numbered window.

3.1. SCHC Compound ACK Message Format

[Figure 1](#) shows the regular SCHC ACK format when all fragments have been correctly received (C=1), as defined in [[RFC8724](#)].

```

|- SCHC ACK Header --|
+-----+---+-----+-----+
| RuleID | W | C=b'1 | b'0-pad(opt) |
+-----+---+-----+-----+

```

Figure 1: SCHC Success ACK message format, as defined in RFC8724

In case SCHC fragment losses are found in any of the windows of the SCHC Packet, the SCHC Compound ACK MAY be used. The SCHC Compound ACK message format is shown in [Figure 2](#).

The window numbered 00, if present in the SCHC Compound ACK, MUST be placed between the Rule ID and the C bit to avoid confusion with padding bits.

```

|--SCHC ACK Header--| W = w1 |...|      W = wi      |
+-----+-----+---+-----+-----+-----+
|RuleID|W=b'w1|C=b'0| Bitmap |...| W=b'wi | Bitmap |b'0-pad(opt)|
+-----+-----+-----+-----+---+-----+-----+

```

Losses are found in windows $W = w1, \dots, wi$; where $w1 < w2 < \dots < wi$

Figure 2: SCHC Compound ACK message format

The SCHC Compound ACK MUST NOT use the Compressed Bitmap format for intermediate windows/bitmaps (i.e., bitmaps that are not the last one), and therefore intermediate bitmaps fields MUST be of size WINDOW_SIZE. Hence, the SCHC Compound ACK MAY use a Compressed Bitmap format only for the last bitmap. The optional usage of this Compressed Bitmap for the last bitmap MUST be specified by the SCHC technology-specific profile.

If a SCHC sender gets a SCHC Compound ACK with invalid W's, such as duplicate W values or W values not sent yet, it MUST discard the whole SCHC Compound ACK message.

Each different SCHC LPWAN technology profile MUST specify how the SCHC Compound ACK is different from the Receiver-Abort message as per [\[RFC8724\]](#), e.g., the Receiver-Abort message is padded with 1s with an extra byte appended at the end, while the SCHC Compound ACK is 0-padded.

3.2. SCHC Compound ACK Behaviour

Besides the ACK-on-Error behaviour described in section 8.4.3 of [[RFC8724](#)], when implementing the SCHC Compound ACK the following applies:

3.2.1. Sender Behaviour

On receiving a SCHC Compound ACK:

- *The fragment sender MUST resend SCHC Fragment messages containing all the tiles of all the windows that are reported missing in the SCHC Compound ACK.

- *If the last of these SCHC Fragment messages reported missing is not an All-1 SCHC Fragment, then the fragment sender MAY either, send in addition a SCHC ACK REQ with the W field corresponding to the last window, continue the transmission of the remaining fragments to be transmitted, or repeat the All-1 fragment to confirm that all fragments have been correctly received.

3.2.2. Receiver Behaviour

On receiving a SCHC ACK REQ:

- *If the receiver knows of any windows with missing tiles for the packet being reassembled, it MAY return a SCHC Compound ACK for the missing fragments, starting from the lowest-numbered window.

On receiving an All-1 SCHC Fragment:

- *If the receiver knows of any windows with missing tiles for the packet being reassembled, it MUST return a SCHC Compound ACK for the missing fragments, starting from the lowest-numbered window.

3.3. SCHC Compound ACK Examples

[Figure 3](#) shows an example transmission of a SCHC Packet in ACK-on-Error mode using the SCHC Compound ACK. In the example, the SCHC Packet is fragmented in 14 tiles, with N=3, WINDOW_SIZE=7, M=2 and two lost SCHC fragments. Only 1 compound SCHC ACK is generated.

```

Sender                      Receiver
|-----W=0, FCN=6 ----->|
|-----W=0, FCN=5 ----->|
|-----W=0, FCN=4 ----->|
|-----W=0, FCN=3 ----->|
|-----W=0, FCN=2 --X-->|
|-----W=0, FCN=1 ----->|
|-----W=0, FCN=0 ----->| Bitmap: 1111011
(no ACK)
|-----W=1, FCN=6 ----->|
|-----W=1, FCN=5 ----->|
|-----W=1, FCN=4 ----->|
|-----W=1, FCN=3 ----->|
|-----W=1, FCN=2 ----->|
|-----W=1, FCN=1 --X-->|
|-- W=1, FCN=7 + RSC ->| Integrity check: failure
|<--- Compound ACK ----| [C=0, W=0 - Bitmap:1111011,
|-----W=0, FCN=2 ----->|           W=1 - Bitmap:1111101]
|-----W=1, FCN=1 ----->| Integrity check: success
|<--- ACK, W=1, C=1 ---| C=1
(End)

```

Figure 3: SCHC Compound ACK message sequence example

```

|-- SCHC ACK Header ---|- W=00 --|----- W=01 -----|
+ -----+-----+-----+-----+-----+
| RuleID | W=00  | C=0  | 1111011 | W=01 | 1111011 | b'0-pad (opt) |
+ -----+-----+-- +-----+-----+-----+-----+

```

Figure 4: SCHC Compound ACK message format example: Losses are found in windows 00 and 01

3.4. SCHC Compound ACK YANG Data Model

The present document also extends the SCHC YANG data model defined in [[I-D.ietf-lpwan-schc-yang-data-model](#)] by including a new leaf in the Ack-on-Error fragmentation mode to describe both the option to use the SCHC Compound ACK, as well as its bitmap format.

3.4.1. SCHC YANG Data Model Extension

<CODE BEGINS> file "ietf-compound-ack@2021-12-10.yang"

```
module ietf-schc-compound-ack {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-schc-compound-ack";
  prefix schc-compound-ack;

  import ietf-schc {
    prefix schc;
  }

  organization
    "IETF IPv6 over Low Power Wide-Area Networks (lpwan) working group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/lpwan/about/>
    WG List:  <mailto:lp-wan@ietf.org>
    Editor:   Laurent Toutain
              <mailto:laurent.toutain@imt-atlantique.fr>
    Editor:   Juan Carlos Zuniga
              <mailto:j.c.zuniga@ieee.org>";
  description
    "
    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.
    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).
    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices.
    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 (RFC 2119) (RFC 8174) when, and only when,
    they appear in all capitals, as shown here.

    *****

    This module extends the ietf-schc module to include the
    Compound ACK behavior for ACK-on-Error as defined in RFC YYYY.
    It introduces a new leaf for ACK-on-Error defining the format
    of the SCHC Compound ACK, adding the possibility to send
    several bitmaps in a single SCHC ACK message.";

  revision 2021-12-08 {
```

```

    description
        "Initial version for RFC YYYY ";
    reference
        "RFC YYYY: SCHC Compound ACK";
}

identity bitmap-format-base-type {
    description
        "Define how the bitmap is formed in ACK messages.";
}

identity bitmap-RFC8724 {
    base bitmap-format-base-type;
    description
        "Bitmap by default as defined in RFC8724.";
}

identity bitmap-compound-ack {
    base bitmap-format-base-type;
    description
        "Compound ACK.";
}

typedef bitmap-format-type {
    type identityref {
        base bitmap-format-base-type;
    }
    description
        "type used in rules";
}

augment "/schc:schc/schc:rule/schc:nature/schc:fragmentation/schc:mode/schc:ack-on-error" {
    leaf bitmap-format {
        when "derived-from(../schc:fragmentation-mode, 'schc:fragmentation-mode-ack-on-error')";
        type schc-compound-ack:bitmap-format-type;
        default "schc-compound-ack:bitmap-RFC8724";
        description
            "How the bitmaps are included in the Ack message.";
    }
    description
        "added to SCHC rules";
}
}

<CODE ENDS>

```

Figure 5: SCHC YANG Data Model - Compound ACK extension

3.4.2. SCHC YANG Tree Extension

```
augment /schc:schc/schc:rule/schc:nature/schc:fragmentation/schc:mode/  
+--rw bitmap-format?    schc-compound-ack:bitmap-format-type
```

Figure 6: SCHC YANG Tree - Compound ACK extension

4. SCHC Compound ACK Parameters

This section lists the parameters related to the SCHC Compound ACK usage that need to be defined in the Profile, in addition to the ones listed in Annex D of [\[RFC8724\]](#).

*Usage or not of the SCHC Compound ACK message.

*Usage or not of the compressed bitmap format in the last window of the SCHC Compound ACK message.

*Differentiation between SCHC Receiver-Abort and SCHC Compound ACK message, e.g., Receiver-Abort message padded with 1s with an extra byte appended at the end, while the SCHC Compound ACK is 0-padded.

5. Security considerations

The current document specifies a message format extension for SCHC. Hence, the same Security Considerations defined in [\[RFC8724\]](#) apply.

6. Acknowledgements

Carles Gomez has been funded in part by the Spanish Government through the Jose Castillejo CAS15/00336 grant, the TEC2016-79988-P grant, and the PID2019-106808RA-I00 grant, 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.

Sandra Cespedes has been funded in part by the ANID Chile Project FONDECYT Regular 1201893 and Basal Project FB0008.

Diego Wistuba has been funded by the ANID Chile Project FONDECYT Regular 1201893.

The authors would like to thank Rafael Vidal, Julien Boite, Renaud Marty, Antonis Platis, Dominique Barthel and Pascal Thubert for their very useful comments, reviews and implementation design considerations.

7. Normative References

- [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-04, February 2021, <<http://www.ietf.org/internet-drafts/draft-ietf-lpwan-schc-yang-data-model-04.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>>.

Authors' Addresses

Juan Carlos Zúñiga
SIGFOX
Montreal QC
Canada

Email: JuanCarlos.Zuniga@sigfox.com
URI: <http://www.sigfox.com/>

Carles Gomez
Universitat Politècnica de Catalunya
C/Esteve Terradas, 7
08860 Castelldefels
Spain

Email: carlesgo@entel.upc.edu

Sergio Aguilar
Universitat Politècnica de Catalunya
C/Esteve Terradas, 7
08860 Castelldefels
Spain

Email: sergio.aguilar.romero@upc.edu

Laurent Toutain
IMT-Atlantique
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France

Email: Laurent.Toutain@imt-atlantique.fr

Sandra Céspedes
NIC Labs, Universidad de Chile
Av. Almte. Blanco Encalada 1975
Santiago
Chile

Email: scesperdes@niclabs.cl

Diego Wistuba
NIC Labs, Universidad de Chile
Av. Almte. Blanco Encalada 1975
Santiago
Chile

Email: wistuba@niclabs.cl