

# LPWAN WG

WG Chairs:

Alexander Pelov <a@ackl.io>

Pascal Thubert <pthubert@cisco.com>

AD: Eric Vyncke

<evyncke@cisco.com>

# Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPS. For advice, please talk to WG chairs or ADs:

[BCP 9](#) (Internet Standards Process)

[BCP 25](#) (Working Group processes)

[BCP 25](#) (Anti-Harassment Procedures)

[BCP 54](#) (Code of Conduct)

[BCP 78](#) (Copyright)

[BCP 79](#) (Patents, Participation)

<https://www.ietf.org/privacy-policy/> (Privacy Policy)



I E T F

## Reminder:

Minutes are taken \*

This meeting might be recorded \*\*

Presence is logged \*\*\*

\* Please contribute to the minutes at: <https://codimd.ietf.org/notes-ietf-interim-2021-lpwan-10-lpwan>

\*\* Recordings and Minutes are public and may be subject to discovery in the event of litigation.

\*\*\* From the Webex login

# Agenda bashing

[16:05] Administrivia

[5min]

- o Note-Well, Scribes, Agenda Bashing
- o WG Status

[16:10] SCHC for CoAP

[10min]

[16:20] SCHC for SigFox

[20min]

[16:40] Data Model for SCHC

[15min]

# WG Status

## Date ◆ Milestone

- |          |  |
|----------|--|
| Feb 2022 | Produce a Standards Track document for SCHC over NB-IoT<br><a href="#">draft-ietf-lpwan-schc-over-nbiot</a>  |
| Oct 2021 | Produce a Standards Track document for SCHC over SigFox<br><a href="#">draft-ietf-lpwan-schc-over-sigfox</a>   |
| Jul 2021 | Produce a Standards Track document to enable operations, administration and maintenance (OAM) to the LPWAN device, including support for delayed or proxied liveness verification (Ping) |
| Feb 2021 | Produce a Standards Track document to define the generic data models to formalize the compression and fragmentation contexts for LPWANs  |
| Dec 2020 | Produce Standard Track documents to apply SCHC IPv6/UDP over the baseline technologies   |
| May 2020 | Perform SCHC Maintenance, including enabling SCHC mechanisms for Upper layer Protocols   |

# Document advancement



Document	Date	Status	IPR	AD / Shepherd
<b>Active Internet-Drafts (5 hits)</b>				
<a href="#">draft-ietf-lpwan-architecture-00</a> LPWAN Static Context Header Compression (SCHC) Architecture	2021-05-18 10 pages	I-D Exists WG Document: Informational		
<a href="#">draft-ietf-lpwan-coap-static-context-hc-19</a> LPWAN Static Context Header Compression (SCHC) for CoAP	2021-05-08 34 pages	RFC Ed Queue: AUTH48 Submitted to IESG for Publication: Proposed Standard Reviews: genart, iotdir, opsdir, secdir, tsvalt	<b>AUTH48 for 99 days</b>	Éric Vyncke Pascal Thubert
<a href="#">draft-ietf-lpwan-schc-over-nbiot-04</a> SCHC over NB-IoT	2021-01-19 22 pages	I-D Exists WG Document Feb 2022		Éric Vyncke
<a href="#">draft-ietf-lpwan-schc-over-sigfox-06</a> SCHC over Sigfox LPWAN	2021-06-11 32 pages	I-D Exists <b>New</b> WG Document Oct 2021		Éric Vyncke
<a href="#">draft-ietf-lpwan-schc-yang-data-model-04</a> Data Model for Static Context Header Compression (SCHC)	2021-02-02 42 pages	I-D Exists WG Document Reviews: yangdoctors		Éric Vyncke
<b>RFCs (3 hits)</b>				
<a href="#">RFC 8576 (was draft-ietf-lpwan-overview)</a> Low-Power Wide Area Network (LPWAN) Overview	2018-05 43 pages	Informational RFC		Suresh Krishnan Alexander Pelov
<a href="#">RFC 8724 (was draft-ietf-lpwan-ipv6-static-context-hc)</a> SCHC: Generic Framework for Static Context Header Compression and Fragmentation	2020-04 71 pages	Proposed Standard RFC		Suresh Krishnan Pascal Thubert
<a href="#">RFC 9011 (was draft-ietf-lpwan-schc-over-lorawan)</a> Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN	2021-04 26 pages	Proposed Standard RFC	<b>1</b>	Éric Vyncke Dominique Barthel

# Action items

- (All) Consider RFC 5856 and RFC 6920
- (Carsten) Kick off OSCORE discussion on passing SCHC Rules with security context

# IETF 111

- Meetings will be in the middle of the CEST night
  - On days that are not in the middle of the night
- We have interim meetings
  - 5 Interims
  - Should we do it now and then?
- Should we ask for an official meeting?

interim-2021-lpwan-1	Tue	2021-05-18
interim-2021-lpwan-2	Tue	2021-06-01
interim-2021-lpwan-3	Tue	2021-06-15
interim-2021-lpwan-4	Tue	2021-06-29
interim-2021-lpwan-5	Tue	2021-07-13

# LPWAN-ing



# draft-ietf-lpwan-coap-static-context- hc-19 AUTH-48

Ana Minaburo (ana@ackl.io)

Laurent Toutain ([laurent@imt-atlantique.fr](mailto:laurent@imt-atlantique.fr))

Ricardo Andreasen (randreasen@fi.uba.ar)

# Title

- Abbreviated SCHC for CoAP
- Extended

1. Static Context Header Compression (SCHC) for Constrained Application Protocol (CoAP)”
2. SCHC compression for CoAP
3. Compression of CoAP using the SCHC framework

# draft-ietf-lpwan-schc-over-sigfox-06

Juan Carlos Zúñiga (Sigfox), Carles Gómez, Sergio Aguilar (UPC),  
Laurent Toutain (IMT-Atlantique),  
Sandra Céspedes, Diego Wistuba (U Chile)

# Draft updates

- rev 05 -> 06
  - SCHC Compound ACK message definition and examples
  - Frame Formats
  - Changed draft status to Standards Track
  - Added more message sequence examples to explain different SCHC/Sigfox scenarios

# SCHC Compound ACK – Definition

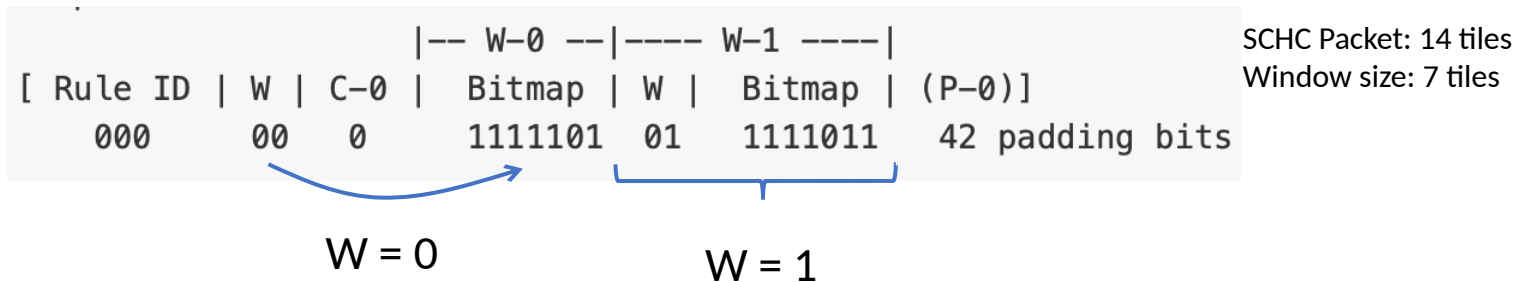
- The SCHC Compound ACK:
  - Only reports windows with fragment losses
  - Includes W field for each bitmap
  - May not fit all bitmaps of all windows for a SCHC packet
  - Has variable size
  - Is compatible with the SCHC Receiver Abort and ACK Failure message formats (RFC8724)
- ACK messages reduction when using SCHC Compound ACK:
  - SCHC Compound ACK messages = Regular SCHC ACKs - (# of windows - 1)

SCHC Packet: 14 tiles  
 Window size: 7 tiles  
 2 SCHC ACK Messages

## Example

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 - no DL Enable)	
-----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-->	
DL Enable  -----W=1, FCN=7----->	Bitmap: 1111101
<--- Compound ACK ---	W=0,1111011 - W=1,1111101
-----W=0, FCN=2----->	W=0 completed
-----W=1, FCN=1----->	W=1 completed
DL Enable  -----W=1, FCN=7----->	
<--- ACK, C=1, W=1 ---	
(End)	

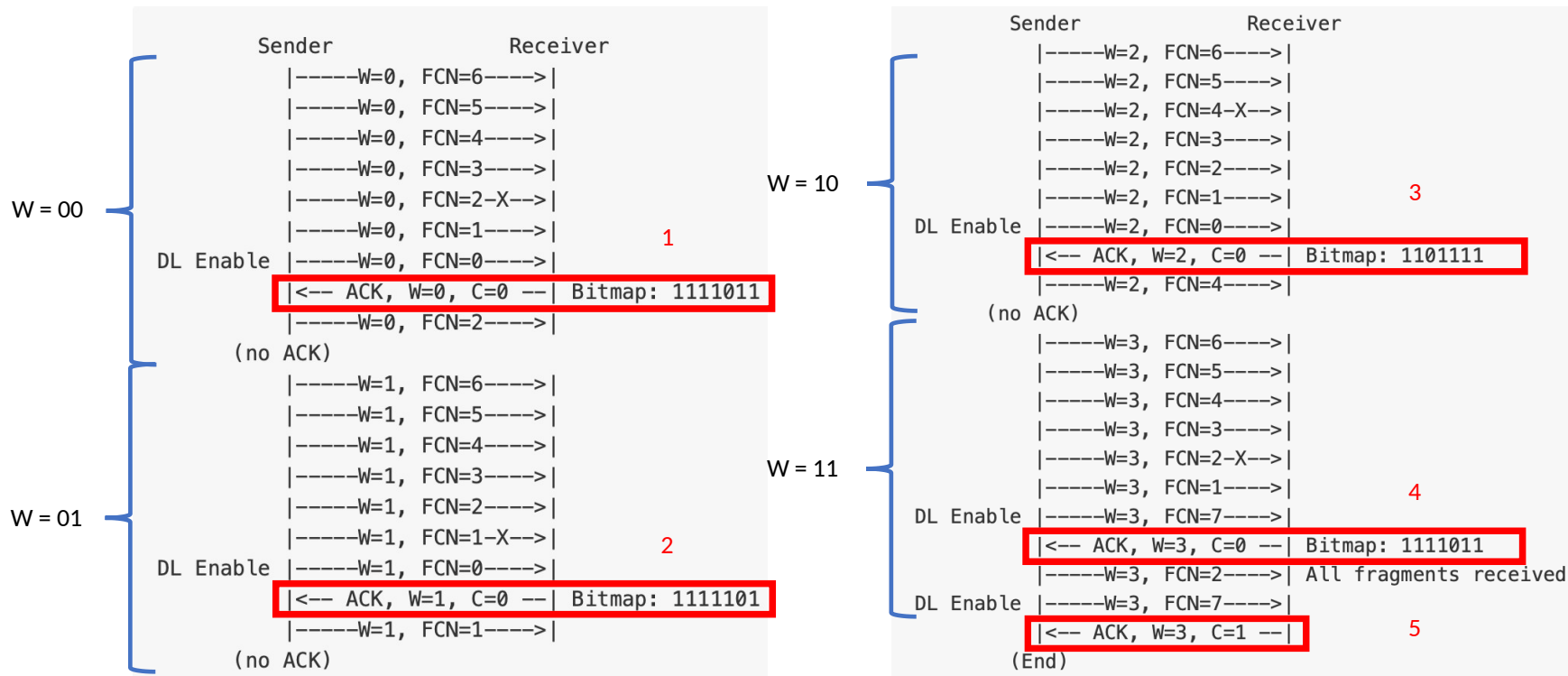
# SCHC Compound ACK – Message Format



- When ACK-on-Error mode is used for UL fragmentation, SCHC Compound ACKs MUST be used in the downlink responses
- $W + \text{Bitmap}$  groups MUST be ordered from the smallest window number to the largest

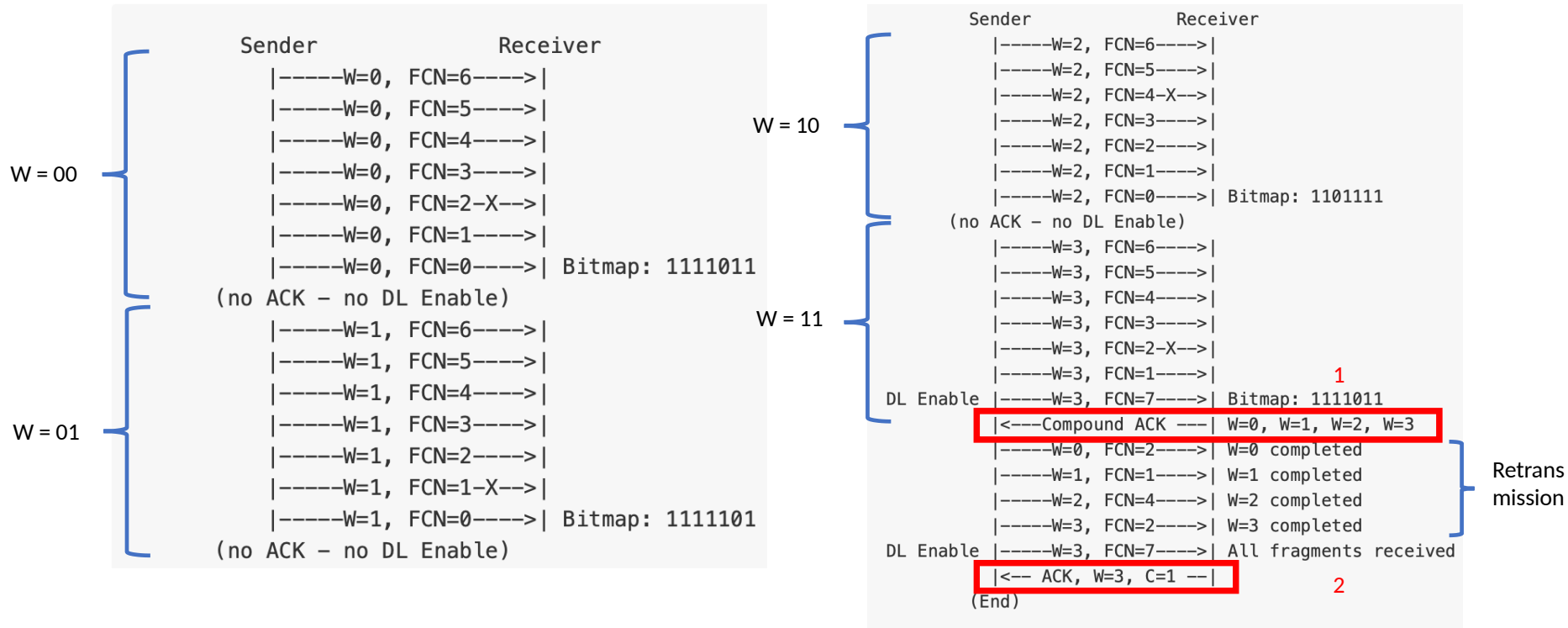
# Example – SCHC Packet 28 tiles – Normal SCHC ACK

5 SCHC ACK Messages



# Example – SCHC Packet 28 tiles – Compound ACK

2 SCHC ACK Messages





# draft-ietf-lpwan-schc-yang-data-model-04 featuring ... "feature"

Laurent Toutain (laurent.toutain@imt-atlantique.fr)

Ana Minaburo (ana@ackl.io)

```
feature fragmentation {  
    .. description "Fragmentation is usually required only at the lowest level."  
}
```

```
.. container schc {  
    .. leaf version {  
        .. type uint64;  
        .. description "used as an indication for versioning";  
    }  
    .. list rule {  
        .. key "rule-id-value rule-id-length";  
        .. uses rule-id-type;  
        .. choice nature {  
            .. case fragmentation {  
                .. if-feature "fragmentation";  
                .. uses fragmentation-content;  
            }  
            .. case compression {  
                .. uses compression-content;  
            }  
            .. description "A rule is either for compression or fragmentation";  
        }  
        .. description "Set of rules compression or fragmentation rules identified by  
            their rule-id";  
    }  
    .. description "a SCHC set of rule is composed of a list of rule which are either  
        compression or fragmentation";  
}
```

```

module: ietf-schc
+--rw schc
  +--rw version?  uint64
  +--rw rule* [rule-id-value rule-id-length]
    +--rw rule-id-value      uint32
    +--rw rule-id-length    uint8
  +--rw (nature)?
    +--:(fragmentation) {fragmentation}?
      +--rw direction          schc:direction-indicator-type
      +--rw dtagsize?         uint8
      +--rw wsize?           uint8
      +--rw fcsize           uint8
      +--rw RCS-algorithm?   RCS-algorithm-type
      +--rw maximum-window-size? uint16
      +--rw retransmission-timer? uint64
      +--rw inactivity-timer?  uint64
      +--rw max-ack-requests?  uint8
      +--rw maximum-packet-size? uint16
      +--rw fragmentation-mode schc:fragmentation-mode-type
      +--rw (mode)?
        +--:(no-ack)
        +--:(ack-always)
        +--:(ack-on-error)
        +--rw tile-size?      uint8
        +--rw tile-in-All1?   schc:all1-data-type
        +--rw ack-behavior?   schc:ack-behavior-type
    +--:(compression)
      +--rw entry* [field-id field-position direction-indicator]
        +--rw field-id          schc:field-id-type
        +--rw field-length     schc:field-length-type
        +--rw field-position    uint8
        +--rw direction-indicator schc:direction-indicator-type
        +--rw target-values* [position]
          | +--rw value?      union
          | +--rw position    uint16
        +--rw matching-operator schc:matching-operator-type
        +--rw matching-operator-value* [position]
          | +--rw value?      union
          | +--rw position    uint16
        +--rw comp-decomp-action schc:comp-decomp-action-type
        +--rw comp-decomp-action-value* [position]
          +--rw value?      union
          +--rw position    uint16

```

**AOB ?**