

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.**



I E T F

Note Super Well

IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the IETF Guidelines for Conduct (RFC 7154), the IETF Anti-Harassment Policy, and the IETF Anti-Harassment Procedures (RFC 7776). If you have any concerns about observed behavior, please talk to the Ombudsteam, who are available if you need to confidentially raise concerns about harassment or other conduct in the IETF.

The IETF strives to create and maintain an environment in which people of many different backgrounds are treated with dignity, decency, and respect. Those who participate in the IETF are expected to behave according to professional standards and demonstrate appropriate workplace behavior.

IETF participants must not engage in harassment while at IETF meetings, virtual meetings, social events, or on mailing lists. Harassment is unwelcome hostile or intimidating behavior -- in particular, speech or behavior that is aggressive or intimidates.

If you believe you have been harassed, notice that someone else is being harassed, or have any other concerns, you are encouraged to raise your concern in confidence with one of the Ombudspersons.



I E T F

Note-Well Definitive information

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)

Reminder:

Minutes are taken *

This meeting is recorded **

Presence is logged ***

- * All: Please contribute online to the minutes at: <https://codimd.ietf.org/notes-ietf-112-lpwan>
- ** Recordings and Minutes are public and may be subject to discovery in the event of litigation
- *** Based on Meetecho Attendance Record. You can log once and only once to each session.

Minute takers, jabber scribes



- Minutes
 - CodiMD: <https://codimd.ietf.org/notes-ietf-112-lpwan>
 - Minute takers volunteers?
- Remote participation
 - Meetecho:
<https://meetings.conf.meetecho.com/ietf112/?group=lpwan&short=&item=1>
- Mailing list: lp-wan@ietf.org
 - To subscribe: <https://www.ietf.org/mailman/listinfo/lp-wan>
- Meeting materials: <https://datatracker.ietf.org/meeting/112/agenda/lpwan-drafts.pdf>

Agenda bashing

14:30 (UTC)	Opening, agenda bashing (Chairs) <ul style="list-style-type: none">• Note-Well, Scribes, Agenda Bashing• Status of drafts• Presenters: The Chairs	5mn
14:35	LPWAN Architecture and general newcomer presentation <ul style="list-style-type: none">• Presenter: Alexander Pelov• Associated drafts: draft-ietf-lpwan-architecture	5mn
14:40	Yang Data Model for SCHC <ul style="list-style-type: none">• Presenter: Laurent Toutain• Associated drafts: draft-ietf-lpwan-schc-yang-data-model	15mn
14:55	SCHC-over-SigFox <ul style="list-style-type: none">• Presenters: Juan Carlos Zuniga / Sergio Aguilar• Associated drafts: draft-ietf-lpwan-schc-over-Sigfox• draft-ietf-lpwan-schc-compound-ack	15mn

Agenda bashing (cont.)

15:10	SCHC-over-NB-IoT <ul style="list-style-type: none">• Presenter: Ana Minaburo• Associated drafts: draft-ietf-lpwan-schc-over-nbiot	10mn
15:20	SCHC adoption within the LoRa Alliance <ul style="list-style-type: none">• Presenters: Dominique Barthel	5mn
15:25	Future of SCHC-over-PPP <ul style="list-style-type: none">• Presenters: Pascal Thubert	5mn
15:30	Meetecho ends	

WG Status

Milestones

Date	↕ Milestone	↕
Feb 2022	Produce a Standards Track document for SCHC over NB-IoT draft-ietf-lpwan-schc-over-nbiot	
Feb 2022	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)	
Oct 2021	Produce a Standards Track document for SCHC over SigFox draft-ietf-lpwan-schc-over-sigfox	
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's advancement



Document	↕ Date	↕ Status	↕ IPR ↕ AD / Shepherd ↕
Active Internet-Drafts (5 hits)			
draft-ietf-lpwan-architecture-00 🗨️ LPWAN Static Context Header Compression (SCHC) Architecture	2021-05-18 10 pages Expires soon	I-D Exists WG Document: Informational	
draft-ietf-lpwan-schc-compound-ack-01 🗨️ SCHC Compound ACK	2021-10-18 9 pages	I-D Exists WG Document	
draft-ietf-lpwan-schc-over-nbiot-06 🗨️ SCHC over NB-IoT	2021-10-25 21 pages New	I-D Exists WG Document <i>Feb 2022</i>	Éric Vyncke ✉
draft-ietf-lpwan-schc-over-sigfox-08 🗨️ SCHC over Sigfox LPWAN	2021-10-24 33 pages	I-D Exists WG Document <i>Oct 2021</i>	Éric Vyncke ✉
draft-ietf-lpwan-schc-yang-data-model-05 🗨️ Data Model for Static Context Header Compression (SCHC)	2021-09-09 51 pages	I-D Exists WG Document Reviews: yangdoctors	Éric Vyncke ✉

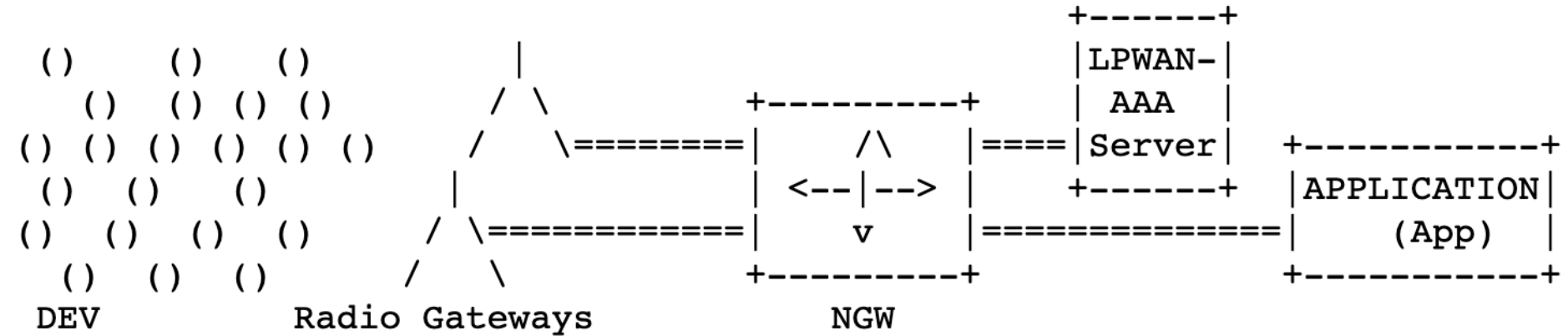
LPWAN Architecture and general newcomer presentation

Alexander Pelov (a@ackl.io)

Pascal Thubert (pthubert@cisco.com)

Ana Minaburo (ana@ackl.io)

Low-Power Wide-Area Networks



RFC 8376 : LPWAN Architecture

Function/ Technology	LoRaWAN	NB-IoT	Sigfox	Wi-SUN	IETF
Sensor, Actuator, device, object	End Device	User Equipment	End Point	Leaf Node	Device (DEV)
Transceiver Antenna	Gateway	Evolved Node B	Base Station	Router Node	Radio Gateway
Server	Network Server	PDN GW/ SCEF*	Service Center	Border Router	Network Gateway (NGW)
Security Server	Join Server	Home Subscriber Server	Registration Authority	Authent. Server	LPWAN- AAA Server
Application	Application Server	Application Server	Network Application	Appli- cation	Application (App)

Collisions

Data-over-NAS

In-band

Duty cycling

Acknowledgements

Guard-bands

License free

In licensed spectrum

No scheduling

Star topology

ALOHA

25 mW transmission power

15-50 km rural outdoor

Low-Power Wide-Area Networks

20 years on simple battery

2-3 km urban indoor

Device-initiated com

Huge densities

Low throughput

Asymmetric links

Throughput:
Hundreds bps

Payload Size:
Tens of Bytes

Uplink:
Few Messages

Downlink:
**Even fewer
messages**

SCHC Architecture

- Provide the reference architecture
 - Modes:
 - SCHC Device/SCHC Gateway
 - SCHC Peers
- RFC 8724 and Rules
 - Introduces Yang Data Model
 - Discusses rule creation and update
 - Discusses rule installation and discovery

SCHC Architecture

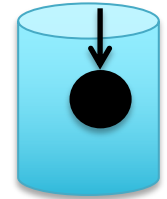


Device
Application



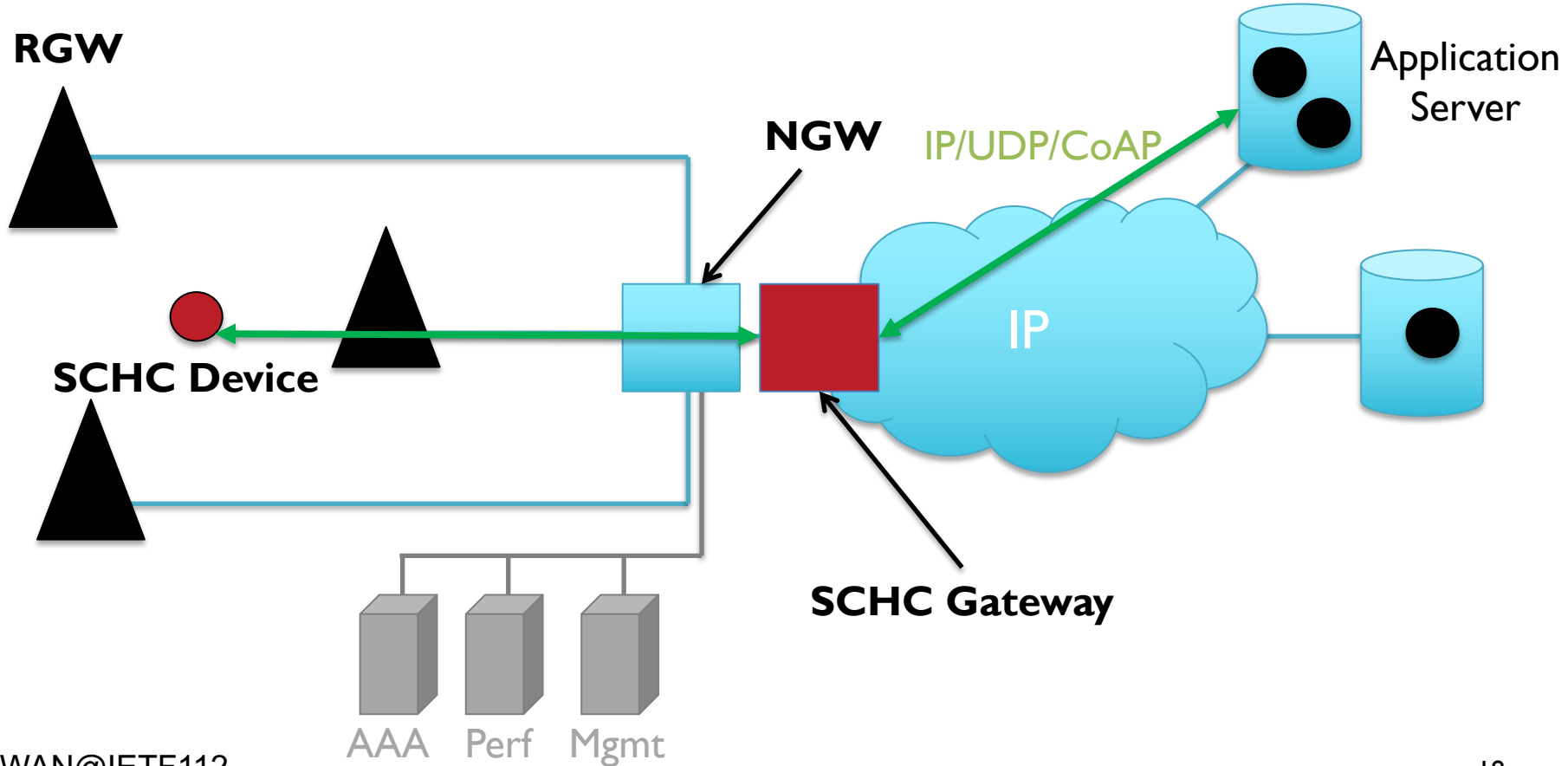
End-Device

Network
Application

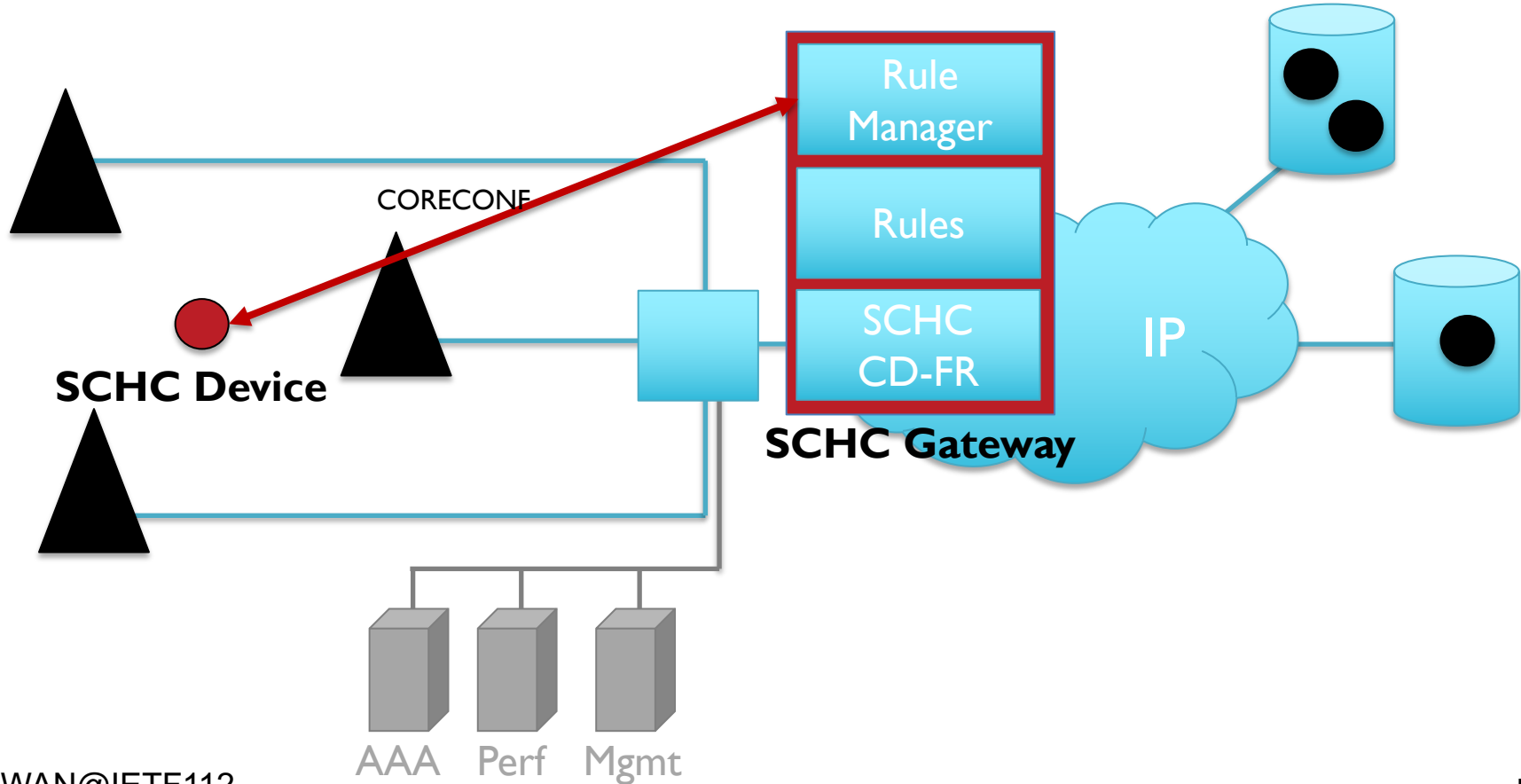


Application
Server

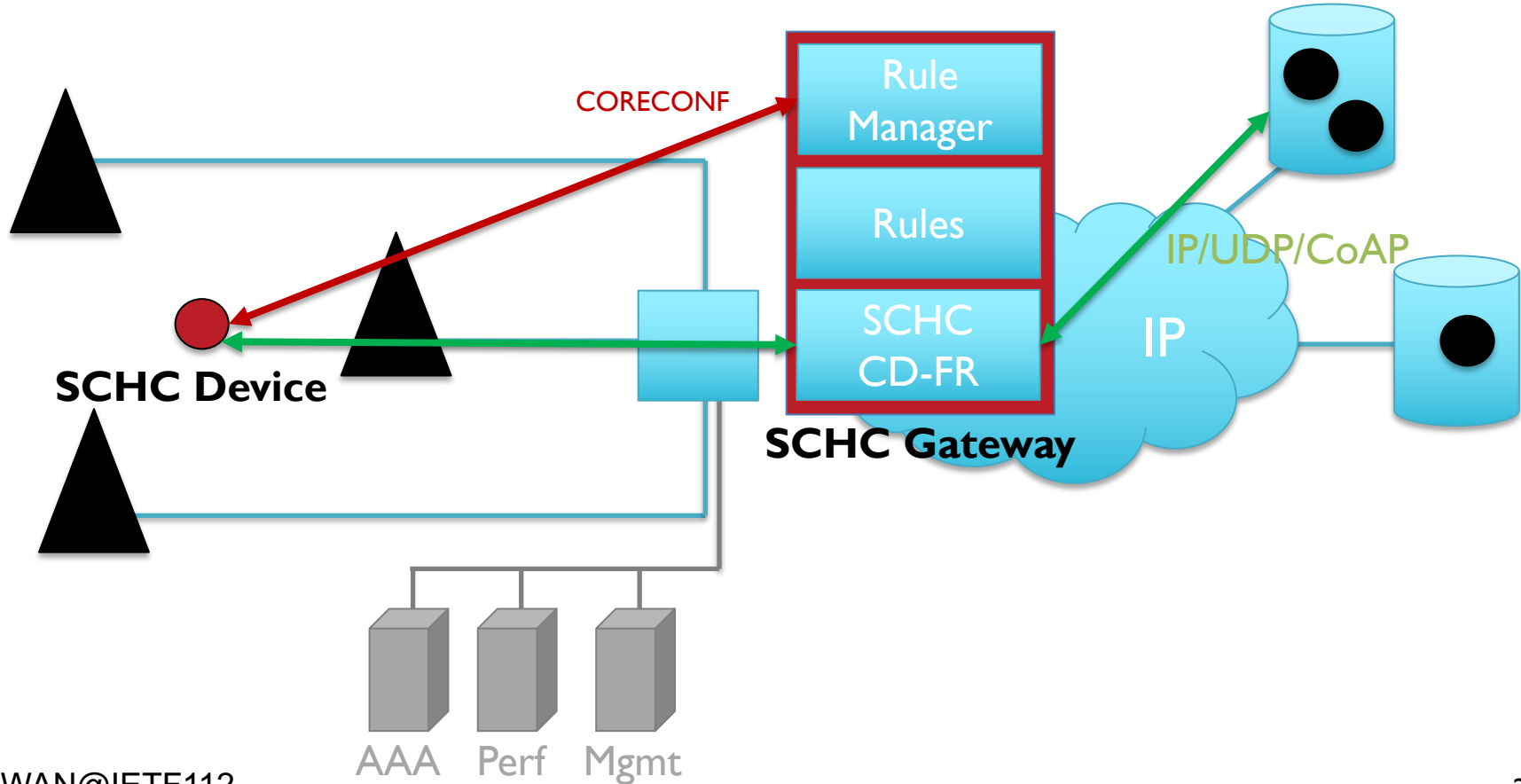
SCHC Architecture



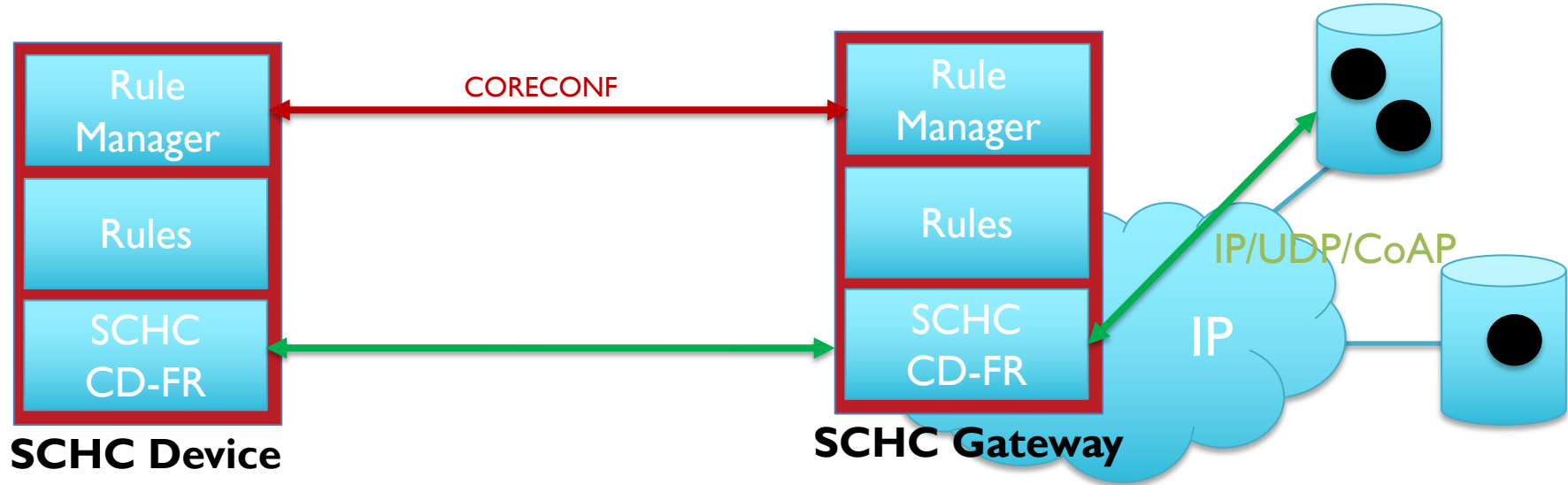
SCHC Architecture



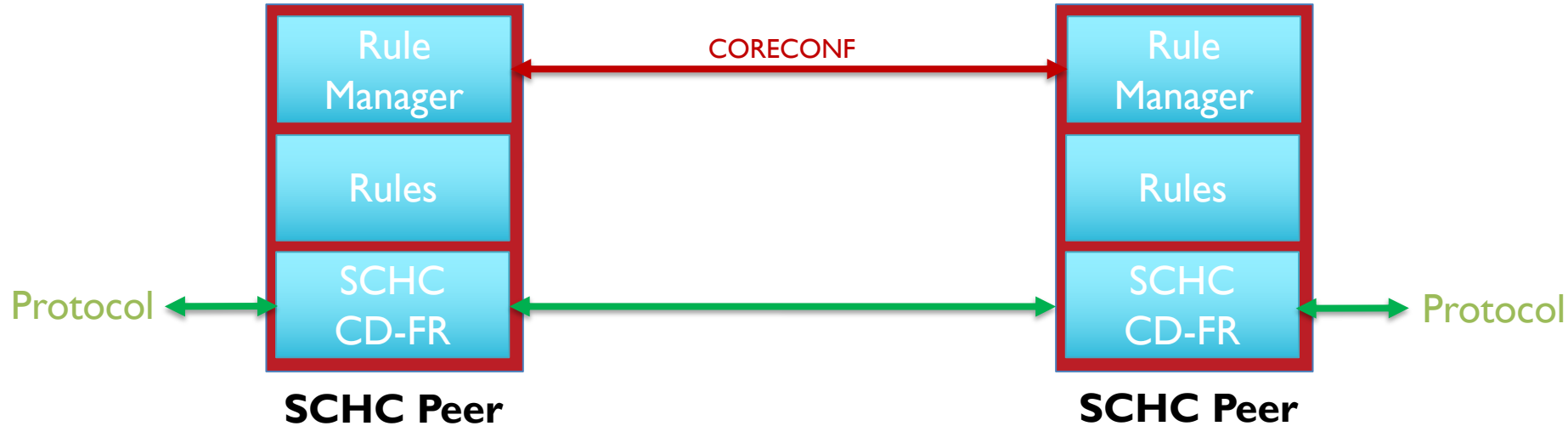
SCHC Architecture



SCHC Architecture



SCHC Architecture



Next Steps in Architecture

- Steps to Provision and Install a SCHC Device
- Steps to Generate and Publish Rules
- Steps to Provision Network for Device Type
- Steps to Identify Device Type and get Rule Set
- State maintenance in the Network
- Steps to Deprecate (forget) a Device Type

draft-ietf-lpwan-schc-yang-data-model

Version 20210817

```
revision 2021-08-17 {  
  description  
    "Initial version from RFC XXXX";  
  reference  
    "RFC XXX: Data Model for Static Context Header Compression (SCHC)";  
}
```

Authors:

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

Ana Minaburo (ana@ackl.io)



Cosmetic

Add a « - »

```
018     }
019     .. leaf dtag-size {
020     .. .. type uint8;
021     .. .. default "0";
022     .. .. description
023     .. .. "Size in bit of the DTag field";
024     .. }
025     .. leaf w-size {
026     .. .. when "not(dderived-from(../fragmentation-mode, 'fragmentation-mode-no-ack'))";
027     .. .. type uint8;
028     .. .. description
029     .. .. "Size in bit of the window field";
030     .. }
031     .. leaf fcn-size {
032     .. .. type uint8;
033     .. .. mandatory true;
034     .. .. description
035     .. .. "Size in bit of the FCN field";
036     .. }
```



Cosmetic

Rename types to use RFC8724 terminology and shorter IDs

```

...";
...
... leaf field-id {
...   type schc:field-id-type;
...   mandatory true;
...   description "Field ID, identify a field in the header with a YANG refe
... }
... leaf field-length {
...   type schc:field-length-type;
...   mandatory true;
...   description "Field Length in bit or through a function defined as a YA
... }
... leaf field-position {
...   type uint8;
...   mandatory true;
...   description "field position in the header is a integer. If the field i
...
... in the header the value is 1, and incremented for each repetition of t
... 0 means that the position is not important and order may change when d
... }
... leaf direction-indicator {
...   type schc:direction-indicator-type;
...   mandatory true;
...   description "Direction Indicator, a YANG reference id to say if the pac
814 ...";
815 ... leaf field-id {
816+ ...   type schc:fid-type;
817 ...   mandatory true;
818+ ...   description
819+ ...     "Field ID, identify a field in the header with a YANG
820 ... }
821 ... leaf field-length {
822+ ...   type schc:fl-type;
823 ...   mandatory true;
824+ ...   description
825+ ...     "Field Length in bit or through a function defined as a
826 ... }
827 ... leaf field-position {
828 ...   type uint8;
829 ...   mandatory true;
830+ ...   description
831+ ...     "Field position in the header is a integer. If the fie
832 ...     in the header the value is 1, and incremented for each
833 ...     0 means that the position is not important and order
834 ... }
835 ... leaf direction-indicator {
836+ ...   type schc:di-type;
837 ...   mandatory true;
838+ ...   description

```

Compression

- Simplify TV (remove union)

```
776  
777  ..//----- RULE ENTRY DEFINITION -----  
778  
779+  ..grouping tv-struct {  
780+  ...description  
781+  ...  "Define the target value element. Always a binary type, strings  
782+  ...  must be converted to binary. field-id allows the conversion to the appropriate  
783+  ...  type.";  
  
784  ... leaf value {  
785  ...   type binary;  
  
786  ... }
```



Compression

- Rename target-value

```
841+ ... }  
842+ ... list target-value {  
843+ ...   key "position";  
844+ ...   uses tv-struct;  
845+ ...   description  
846+ ...     "A list of value to compare with the header field value. If target value  
847 ...     is a singleton, position must be 0. For matching-list, should be consecutive po  
848 ...     values starting from 1.";  
849 ... }
```

— « rule » and « entry » are singular



No-Compression

- Add a new type of rule for no-compression
- Remove version field (not in RFC8724)

```
1036 container schc {  
1037     list rule {  
1038         key "rule-id-value rule-id-length";  
1039         uses rule-id-type;  
1040         choice nature {  
1041             case fragmentation {  
1042                 if-feature "fragmentation";  
1043                 uses fragmentation-content;  
1044             }  
1045             case compression {  
1046                 uses compression-content;  
1047             }  
1048+            case no-compression {  
1049+                description  
1050+                "RFC8724 allows a rule for uncompressed headers";  
1051+            }  
1051 }
```



Fragmentation

- Add l2-word-size

```
899  .. grouping fragmentation-content {
900+  .. description
901+  .. "This grouping defines the fragmentation parameters for
902  .. all the modes (No Ack, Ack Always and Ack on Error) specified in
903  .. RFC 8724.";
904+  .. leaf l2-word-size {
905+  ..   type uint8;
906+  ..   default "8";
907+  ..   description
908+  ..     "Size in bit of the layer 2 word";
909+  .. }
```

Added Compound Ack

```
751+
752+ identity bitmap-format-base-type {
753+   description
754+     "Define how the bitmap is defined in ACK messages.";
755+ }
756+
757+ identity bitmap-RFC8724 {
758+   base bitmap-format-base-type;
759+   description
760+     "Bitmap as defined in RFC8724.";
761+ }
762+
763+ identity bitmap-compound-ack {
764+   base bitmap-format-base-type;
765+   description
766+     "Compound Ack.";
767+ }
768+
769+ typedef bitmap-format-type {
770+   type identityref {
771+     base bitmap-format-base-type;
772+   }
773+   description
774+     "type used in rules";
775+ }
```

- See draft

Added Compound Ack

```
1002+ ..... leaf bitmap-format {  
1003+ .....   type schc:bitmap-format-type;  
1004+ .....   when "derived-from(..fragmentation-mode, 'fragmentation-mode-ack-on-error')";  
1005+ .....   default "schc:bitmap-RFC8724";  
1006+ .....   description  
1007+ .....     "How the bitmaps are included in the Ack message.";  
1008 ..... }
```



Is it useful ?

- Has disappear from RFC 8724

```
812     . . . . . leaf maximum-window-size {  
813     . . . . . | . . . . . type uint16;  
814     . . . . . | . . . . . description "by default 2^wsize - 1";  
815     . . . . . }  
816
```



relations between values

- add MUST statement in compression rules

```
leaf matching-operator {  
  type schc:mo-type;  
  must "../target-value or derived-from-or-self(., 'mo-ignore')"{  
    error-message "mo-equal, mo-msb and mo-match-mapping require target-value";  
    description  
    "target-value is not required for mo-ignore";  
  }  
  must "not (derived-from-or-self(., 'mo-msb')) or ../matching-operator-value"{  
    error-message "mo-msb requires length value";  
  }  
  mandatory true;  
  description  
  "MO: Matching Operator";  
}
```

– derived-from-**or-self** to match identityref



relations between values

- add WHEN statement in fragmentation rules

```
981     case ack-always,
982     case ack-on-error {
983     leaf tile-size {
984     type uint8;
985+     when "derived-from(..fragmentation-mode, 'fragmentation-mode-ack-on-error')";
986+     description
987+     "Size in bit of tiles, if not specified or set to 0: tile fills the fragment
988     }
989     leaf tile-in-All1 {
990     type schc:all1-data-type;
991+     when "derived-from(..fragmentation-mode, 'fragmentation-mode-ack-on-error')";
992+     description
993+     "When true, sender and receiver except a tile in All-1 frag";
994     }
995     leaf ack-behavior {
996     type schc:ack-behavior-type;
997+     when "derived-from(..fragmentation-mode, 'fragmentation-mode-ack-on-error')";
998+     description
999+     "Sender behavior to acknowledge, after All-0, All-1 or when the
1000     LPWAN allows it (Always);
1001     }
1002+     leaf bitmap-format {
1003+     type schc:bitmap-format-type;
1004+     when "derived-from(..fragmentation-mode, 'fragmentation-mode-ack-on-error')";
1005+     default "schc:bitmap-RFC8724";
1006+     description
```



Other relations between values ?

((LPWAN))

- Test if MSB arg is shorter than field-length ?
 - How to deal with length functions ?
- Test is LSB/Map-send CDA with MSB/M-M MO ?
 - Not in the spec
- Any other ???



Conclusion

- New version is on github lp-wan repository
 - Check against RFC 8724 for fragmentation
- Used yangson to check rules
 - Done during Hackathon
 - Transform a openSCHC JSON file to a JSON following YANG DM
 - For Compression, Fragmentation To Be Done
 - CORECONF TBD
 - Openshc version will be released soon.



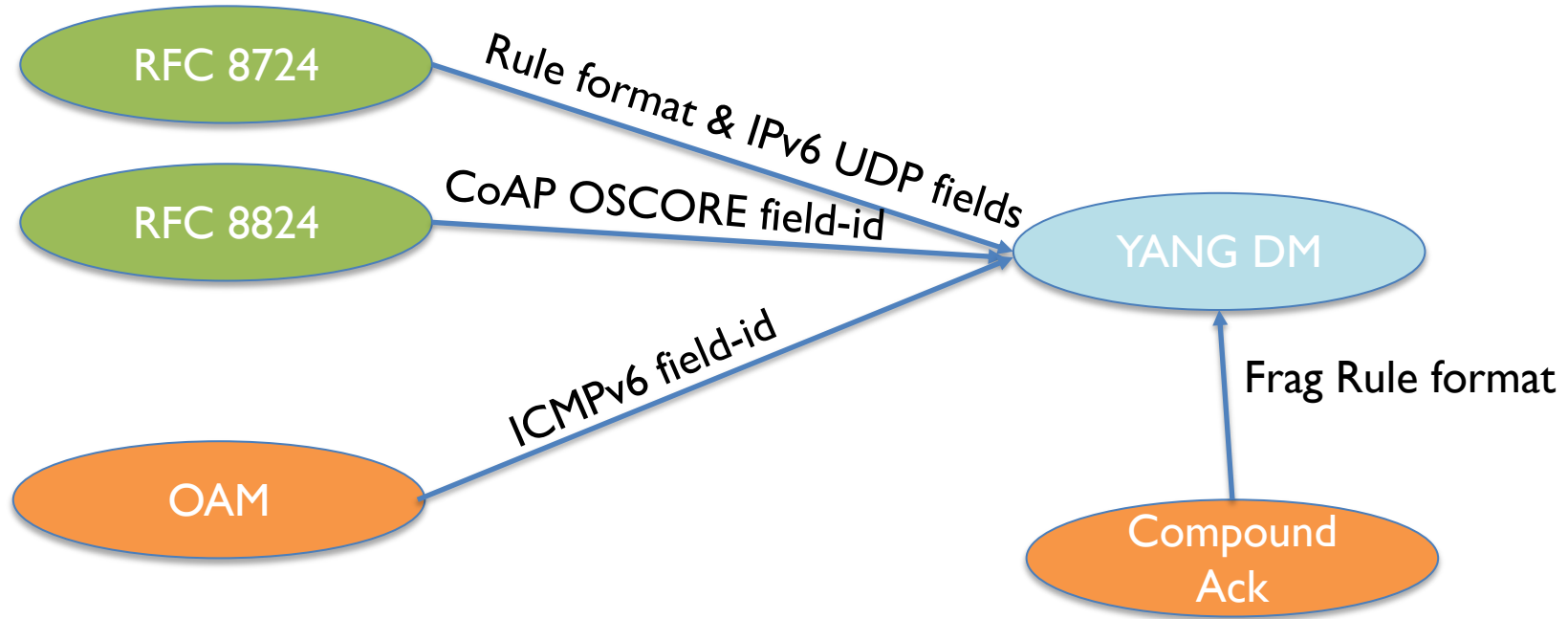
Example openSCHC

Rule 6/3		110	
IPV6.VER	4	1	BI
IPV6.TC	8	1	BI
IPV6.FL	20	1	BI
IPV6.LEN	16	1	BI
IPV6.NXT	8	1	BI
IPV6.HOP_LMT	8	1	BI
IPV6.DEV_PREFIX	64	1	BI
IPV6.DEV_IID	64	1	BI
IPV6.APP_PREFIX	64	1	BI
IPV6.APP_IID	64	1	BI
ICMPV6.TYPE	8	1	DW
ICMPV6.TYPE	8	1	UP
ICMPV6.CODE	8	1	BI
:	:	:	:
ICMPV6.CKSUM	16	1	BI
ICMPV6.IDENT	16	1	BI
ICMPV6.SEQNO	var	1	BI

6	EQUAL	NOT-SENT
0	EQUAL	NOT-SENT
0	IGNORE	NOT-SENT
	IGNORE	COMPUTE-LENGTH
58	EQUAL	NOT-SENT
255	IGNORE	NOT-SENT
200104701f2101d2	EQUAL	NOT-SENT
00000000000000001	EQUAL	NOT-SENT
	IGNORE	VALUE-SENT
	IGNORE	VALUE-SENT
128	EQUAL	NOT-SENT
129	EQUAL	NOT-SENT
0	MATCH-MAPPING	MAPPING-SENT
1:	:	:
0	IGNORE	COMPUTE-CHECKSUM
0	MSB(8)	LSB
0	IGNORE	VALUE-SENT



Dependancies



Tree



```
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 l2-word-size?   uint8
        +--rw direction      schc:direction-indicator-type
        +--rw dtag-size?     uint8
        +--rw w-size?        uint8
        +--rw fcn-size       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
            +--rw bitmap-format? schc:bitmap-format-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-value* [position]
              +--rw value?   binary
              +--rw position uint16
            +--rw matching-operator      schc:matching-operator-type
            +--rw matching-operator-value* [position]
              +--rw value?   binary
              +--rw position uint16
            +--rw comp-decomp-action      schc:comp-decomp-action-type
            +--rw comp-decomp-action-value* [position]
              +--rw value?   binary
              +--rw position uint16
          +--:(no-compression)
```



draft-ietf-lpwan-schc-over-sigfox-08

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

Draft updates since IETF 111

- rev 05 -> 06
 - SCHC Compound ACK mechanism added
- rev 06 -> 07
 - SCHC Compound ACK split as generic mechanism for all LPWANs
 - Yang model description added to SCHC Compound ACK
- rev 07 -> 08
 - Added normative reference to SCHC Compound ACK usage for SCHC/Sigfox, together with signaling examples
 - SCHC-Abort definitions added

Hackathon – SCHC/Sigfox

- Objectives:
 - Use SCHC over Sigfox draft in a "real case" scenario
 - Mini weather station: Measure temperature and humidity
 - Send JSON file with measured data from Sigfox Device
- Equipment:
 - LoPy4
 - Pysense
- Codebase
 - SCHC over Sigfox Project in github
 - url: <https://github.com/schc-over-sigfox>



W: 0

```
001 - , "light":
010 - ": 58.64736
011 - 61903, "hum
100 - "temp": 21.
101 - 4.872366,
110 - {"bat_vol":
111 - [4, 5]}
```

Case 2

W: 0

```
100 - p": 21.8871
101 - 54949, "tem
110 - {"hum": 57.
111 - 5}
```

Implementation and Results

- LoPy4 + Pysense
 - Two threads
 - One thread in charge of sensing
 - Second thread handles SCHC over Sigfox (i.e., fragmentation, transmissions, etc.)
 - Both threads are connected using a list
 - JSON file format:
 - 1- {"temp": XXX, "hum": YYY, "bat_vol": ZZZ, "light": ABC}
 - 2- {"temp": XXX, "hum": YYY}
 - SCHC Fragments per JSON
 - 1- 7 SCHC Fragments (1 window)
 - 2- 4 SCHC Fragments

Results

Case 2

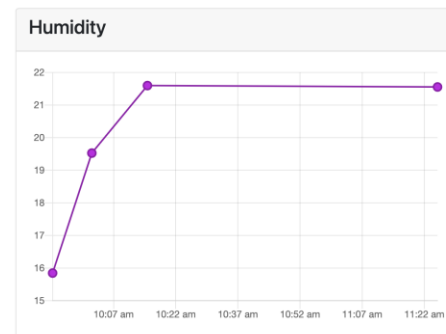
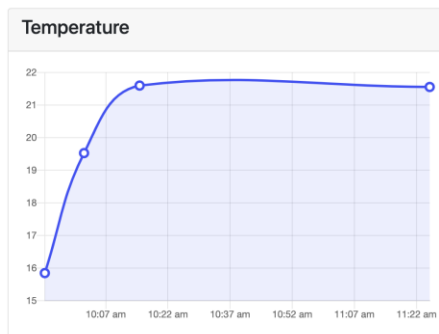
Payload

{"hum": 63.08279, "temp": 15.84892}

{"hum": 53.7601, "temp": 19.52763}

{"hum": 45.95958, "temp": 21.59757}

{"hum": 48.65317, "temp": 21.55467}



Next steps

- Verify latest details on implementation
- Test one more use case (e.g. asset tracking)

Thanks!

draft-ietf-lpwan-schc-compound-ack-01

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

Draft history

- SCHC Compound ACK message definition and examples added to draft-ietf-schc-over-Sigfox rev 06
- WG decision to make compound ACK mechanism generic for all LPWAN technologies on a standalone draft
- Implementation of SCHC Compound ACK tested over Sigfox
- draft-ietf-lpwan-schc-compound-ack-00 posted in July'21, and revised in Oct'21
- WG Last Call held in October-November 2021

Backup slides

- SCHC Compound ACK Definition
- Message format
- Message sequence examples

SCHC Compound ACK – Definition

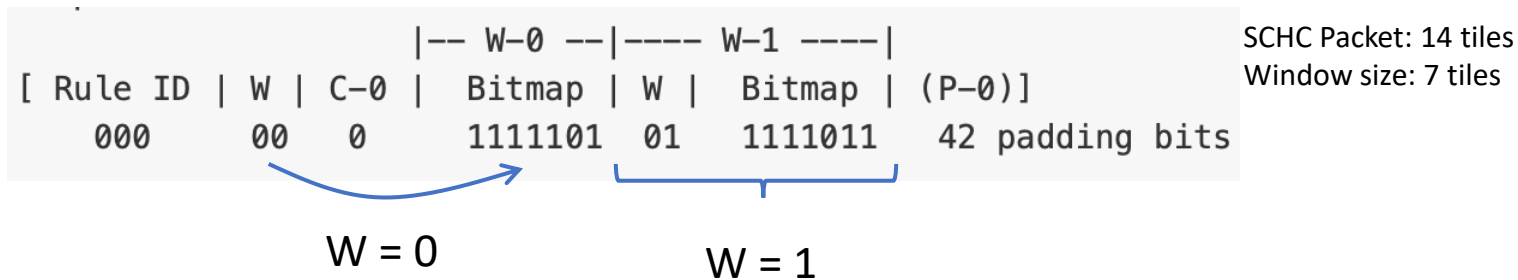
- The SCHC Compound ACK:
 - Only reports windows with fragment losses
 - Includes W field for each bitmap
 - May not fit all bitmaps of all windows for a SCHC packet
 - Has variable size
 - Is compatible with the SCHC Receiver Abort and ACK Failure message formats (RFC8724)
- ACK messages reduction when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

Example

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

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)	

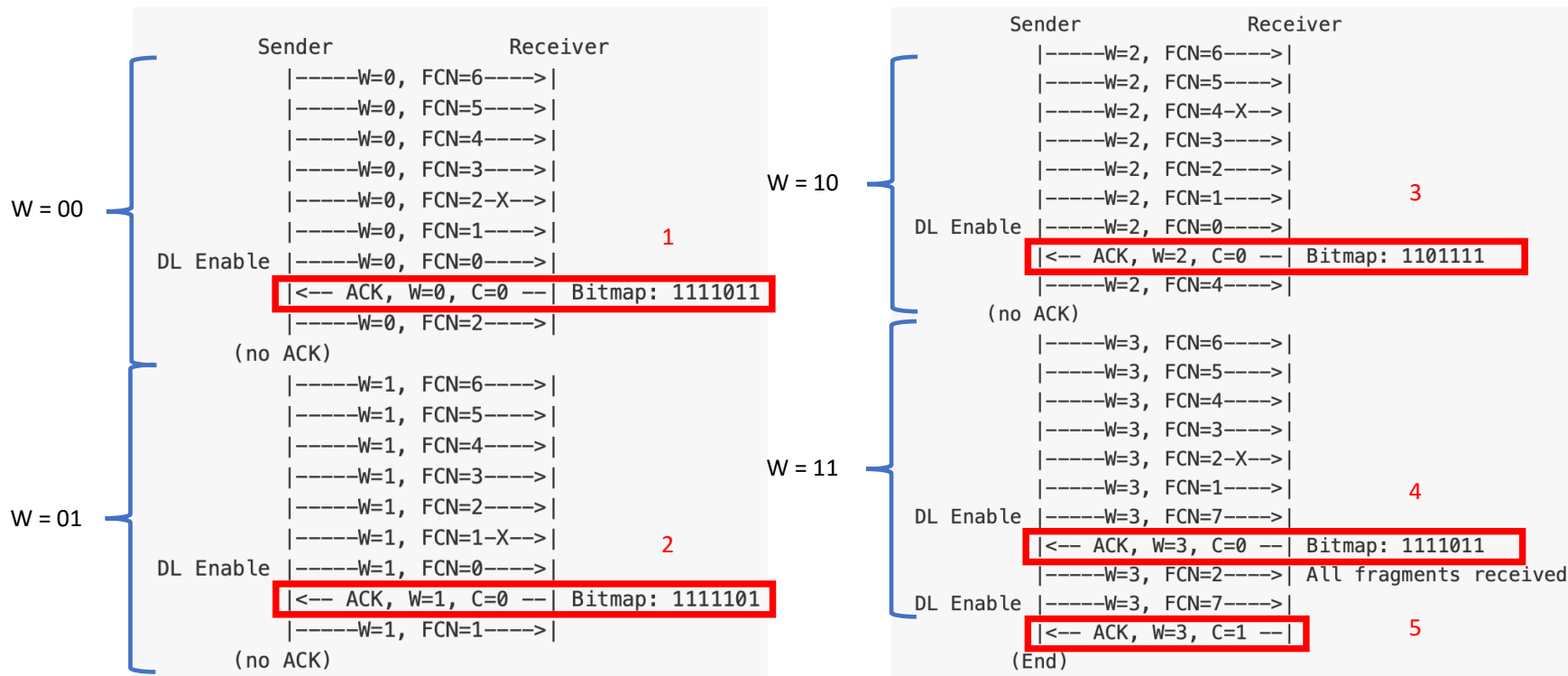
SCHC Compound ACK – Message Format



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

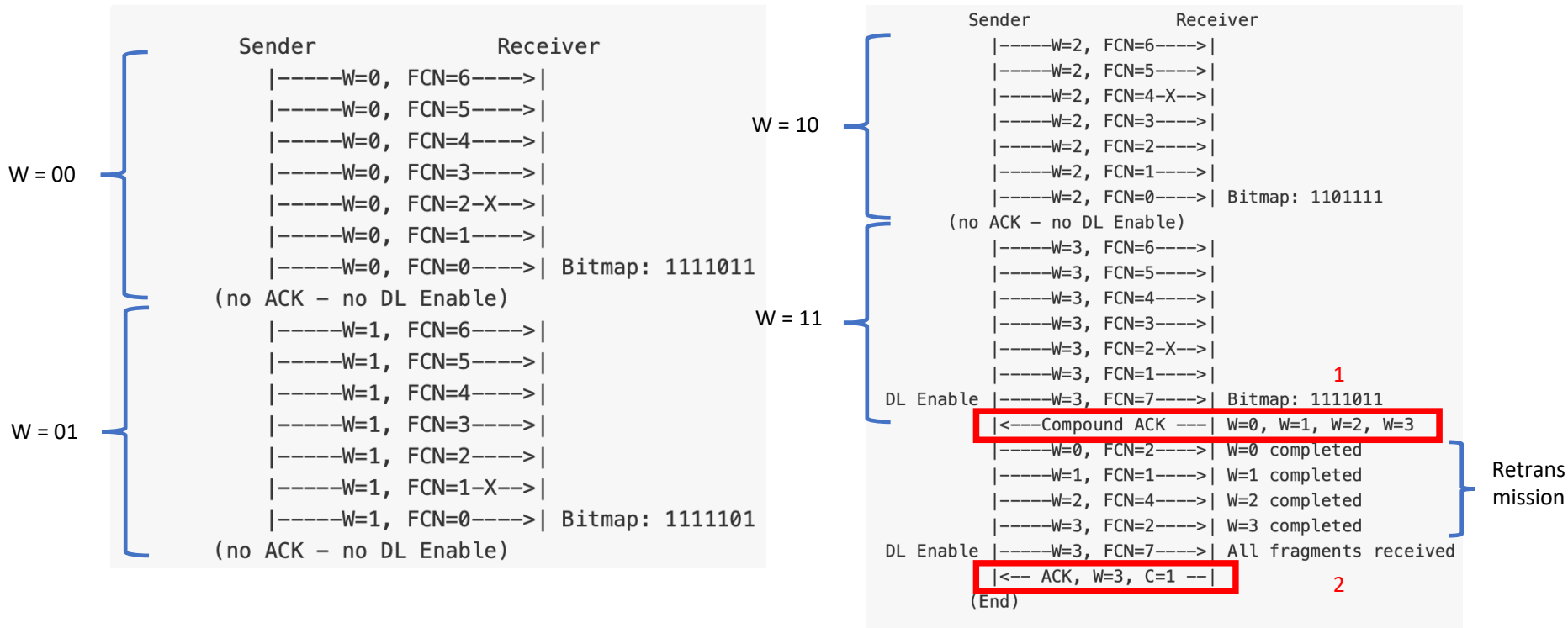
Example – SCHC Packet 28 tiles – Normal SCHC ACK

5 SCHC ACK Messages



Example – SCHC Packet 28 tiles – Compound ACK

2 SCHC ACK Messages



Thanks!

draft-ietf-lpwan-schc-over-nbiot-06

Authors:

Edgar Ramos

Ana Minaburo

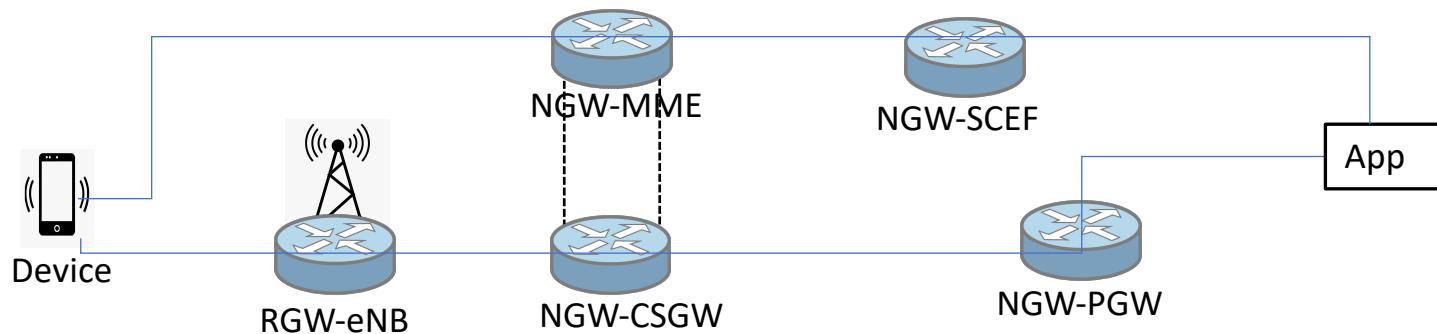
Reviews

- Reviews from ML
 - Thanks to Ivaylo Petrov & Laurent Toutain for your inputs

Principal changes

- Nits
- Terminology
 - Device
 - RGW-eNB
 - NGW-MME
 - NGW-SGSN
 - NGW-PGW
 - NGW-SCEF

NB-IoT Architecture



Thank you

- Last Call?

RFC 8724, 9011 adoption at LoRa Alliance

Authors:

Dominique Barthel <dominique.barthel@orange.com>

Olivier Gimenez <ogimenez@semtech.com>

IPv6 support at LoRa Alliance

- First use case is DLMS over LoRaWAN
 - DLMS is mainly energy metering application protocol
 - Reuse of existing DLMS/UDP/IPv6 stack, with SCHC/LoRAWAN underneath
- IPv6_over_LoRaWAN Technical Specification being completed
 - Editors: Acklio, Semtech, IMT Atlantique
 - Contributors: 10 companies/institutions
 - Mandates compliance with RFC8724, 9011 and
 - Recommends to pick compression rule with min. Compression Residue
 - Recommends shorter Inactivity and Retransmission Timers than RFC9011 defaults, based on application

Next

- Vote at the Technical Committee meeting Nov 16th
- Write DLMS/UDP/IPv6/LoRaWAN certification document
- Discussions with CSA (formerly Zigbee Alliance) on LoRaWAN in Connected Home over IP (Matter)

draft-thubert-intarea-schc-over-ppp

Authors:

Pascal Thubert <pthubert@cisco.com>

draft-thubert-intarea-schc-over-ppp

- SCHC over PPP (and then PPP over foo)
- Enables SCHC over
 - Serial, 3GPP
 - Ethernet with PPPoE, Wi-Fi with Ethernet
- Signals
 - A new compression for PPP (Updates RFC 5172)
 - The URL of the data model for the compression
 - Dependency on draft-ietf-lpwan-schc-yang-data-model

Draft status: Stable

- 03 expired
- No discussion at INT AREA
- Co Authors?
- Add applicability statement?
- Possible extensions?