LPWAN WG

WG Chairs:
Alexander Pelov <a@ackl.io>
Pascal Thubert <pthubert@cisco.com>

AD: Suresh Krishnan
<suresh@kaloom.com>
Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- Any Birds of a Feather (BOF) session
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

All IETF Contributions are subject to the rules of RFC 5378 and RFC 8179.

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 8179 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.
Reminder:

Minutes are taken *
This meeting is recorded **
Presence is logged ***

* Scribe; please contribute online to the minutes at: https://etherpad.tools.ietf.org/p/lpwan
** Recordings and Minutes are public and may be subject to discovery in the event of litigation.
*** From the Webex login
Agenda bashing

17:00  Opening, agenda bashing (Chairs)  
       • Note-Well, Scribes, Agenda Bashing, Approval minutes from last meeting  
       • Review todo  
       • Status of drafts (WGLC / forthcoming WGLC)  

17:10  LPWAN Overview - WGLC status and updates  

17:15  Milestone updates and new items (Chairs)  

17:20  SCHC - Adding a length field for rules (Arun)  

17:30  SCHC – LPWAN Fragmentation (Carles)  

17:40  SCHC - Fragmentation optimization + FSM (Laurent)  

17:xx  AOB  

Interim, September 12th, 2017
Last meeting Action items

- Start WGLC Process for LPWAN overview
- CG: Default fragmentation mode (Window mode)
- DB, DD: Review SCHC IP/UDP fragmentation
## Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 2017</td>
<td>Submit CoAP compression mechanism to the IESG for publication as a Proposed Standard</td>
</tr>
<tr>
<td>May 2017</td>
<td>Submit IP/UDP compression and fragmentation mechanism to the IESG for publication as a Proposed Standard</td>
</tr>
<tr>
<td>Apr 2017</td>
<td>Submit LPWAN specification to the IESG for publication as an Informational Document</td>
</tr>
<tr>
<td><strong>Done</strong></td>
<td>Adopt CoAP compression mechanism as a WG item</td>
</tr>
<tr>
<td><strong>Done</strong></td>
<td>Adopt IP/UDP compression and fragmentation mechanism as a WG item</td>
</tr>
<tr>
<td><strong>Done</strong></td>
<td>Adopt LPWAN specifications as WG item</td>
</tr>
</tbody>
</table>
## Charter – Milestone updates

### Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 2017</td>
<td>Adopt CoAP compression mechanism as a WG item</td>
</tr>
<tr>
<td>Nov 2017</td>
<td>Adopt IP/UDP compression and fragmentation mechanism as a WG item</td>
</tr>
<tr>
<td>Jan 2018</td>
<td>Submit CoAP compression mechanism to the IESG for publication as a Proposed Standard</td>
</tr>
<tr>
<td></td>
<td>Submit IP/UDP compression and fragmentation mechanism to the IESG for publication as a Proposed Standard</td>
</tr>
<tr>
<td></td>
<td>Submit LPWAN specification to the IESG for publication as an Informational Document</td>
</tr>
<tr>
<td>Done</td>
<td>Adopts CoAP compression mechanism as a WG item</td>
</tr>
<tr>
<td>Done</td>
<td>Adopt IP/UDP compression and fragmentation mechanism as a WG item</td>
</tr>
<tr>
<td>Done</td>
<td>Adopt LPWAN specifications as WG item</td>
</tr>
</tbody>
</table>
WG Rechartering

- ICMP Compression (draft from Diego)
- SCHC over specific LPWAN technologies
  - Profile Technologies.
  - Develop the different document for each technology in order to define the different parameters.
- Security, all the security solution for the LPWAN
- Rules-ID’s management – Use to identify some specific cases:
  - Fragmentation
  - Format Values
  - Rule-IDs dedicated for some specific cases as CBOR structure representation, fragmentation, etc
  - The way to use the Rule-Id and their configuration
- YANG data modeling for SCHC

(Ana Minaburo’s slide presented at IETF99)
IETF 100

• 2Hours? More?
• ~70 people?
• Exclusions?
LPWAN Overview

Editor: Stephen Farrell
(many contributors)
WGLC status and updates

Some minor WGLC comments
But nothing that ought delay IETF LC
Stephen is ready for pushing out a new rev at the end of IETF LC with whatever's sent to the list(s) since the current rev went out.
Then shoot in the pub-req and let's move it along.
SCHC

Authors:
Ana Minaburo <ana@ackl.io>
Laurent Toutain <laurent.toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
SCHC Compression

– Dominique Barthel input request
– Carles modification on fragmentation part
– Document reordering text:
  1. Introduction ......................... 3
  2. LPWAN Architecture .................. 4
  3. Terminology .......................... 5
  4. Static Context Header Compression .... 6
  5. Fragmentation ........................ 14
  6. SCHC Compression for IPv6 and UDP headers .......... 25
  7. Security considerations .................. 29
  8. Acknowledgements ..................... 30
  9. References ............................ 30
     Appendix A. SCHC Compression Examples ............ 31
     Appendix B. Fragmentation Examples ............. 33
     Appendix C. Allocation of Rule IDs for fragmentation ...... 37
Representing Length field in the Rule
draft-ietf-lpwan-ipv6-static-context-hc-05

Authors:
Arunprabhu Kandasamy <arun@ackl.io>
Why we need it?

"The receiver must be able to find the size of each compressed field which can be given by the rule or may be sent with the compressed header" [Section:6, pg: 11]

- Compressor/Decompressor shall use the length from rule for CDA's

- makes life easier for interop
  - we spent considerable amount of time in rewriting the rules during interop @ Prague
How can it be done?

- A new column "length" in the context table.
- represent the length in bits for each fields.
- use '0' to represent variable length fields.
LPWAN SCHC Fragmentation

Authors:
Ana Minaburo <ana@ackl.io>
Laurent Toutain <laurent.toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
Outline

• Status

• Outstanding issues/details reported on the list
  – 1. Window mode – ACK “always”
    • Retries in the last window
  – 2. Window mode – ACK “always”
    • MAX_Frag_RETRIES and MAX_ACK_REQUESTS
  – 3. Window mode – ACK on error
    • MAX_Frag_RETRIES
Status

• Last version published is -05

• Updated working version already available on GitHub:
Item 1 (I/III)

• Window mode – ACK “always”
  – Retries in the last window
  – Problem: not clear for a receiver when to check the MIC and send an ACK after fragment retries in the last window
    • E.g. the last retransmitted fragment might not be $\text{FCN}=2^N-1$
Item 1 (II/III)

- **Solution (based on Laurent’s email)**
  - After frag FCN=$2^N-1$ received, check MIC after each retransmitted frag received
  - If reassembly OK, frag receiver sends the ACK

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----W=0, CFN=6-------</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=5-------</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=4--X---&gt;</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=3--X---&gt;</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=2--X---&gt;</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=1--X---&gt;</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=0--X---&gt;</td>
<td></td>
</tr>
<tr>
<td>-----W=0, CFN=7-------</td>
<td>MIC checked</td>
</tr>
<tr>
<td>&lt;-----ACK, W=0--------</td>
<td>bitmap:11000001</td>
</tr>
<tr>
<td>-----W=0, CFN=4-------</td>
<td>MIC checked: wrong</td>
</tr>
<tr>
<td>-----W=0, CFN=3-------</td>
<td>MIC checked: wrong</td>
</tr>
<tr>
<td>-----W=0, CFN=2-------</td>
<td>MIC checked: right</td>
</tr>
<tr>
<td>&lt;-----ACK, W=0--------</td>
<td>no bitmap</td>
</tr>
</tbody>
</table>
Item 1 (III/III)

- If no ACK arrives, ACK "always" timer expires
- Frag FCN=$2^N-1$ is resent

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w=0$, $CFN=6$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=5$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=4-X$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=3-X$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=2-X$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=7$</td>
<td>MIC checked</td>
</tr>
<tr>
<td>--&lt; ACK, $w=0$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=4$</td>
<td>MIC checked: wrong</td>
</tr>
<tr>
<td>$w=0$, $CFN=3$</td>
<td>MIC checked: wrong</td>
</tr>
<tr>
<td>$w=0$, $CFN=2$</td>
<td>MIC checked: right</td>
</tr>
<tr>
<td>$X$-ACK, $w=0$</td>
<td>$</td>
</tr>
<tr>
<td>timeout</td>
<td></td>
</tr>
<tr>
<td>$w=0$, $CFN=7$</td>
<td>$</td>
</tr>
<tr>
<td>--&lt; ACK, $w=0$</td>
<td>$</td>
</tr>
<tr>
<td>$w=0$, $CFN=2$</td>
<td>MIC checked: right</td>
</tr>
<tr>
<td>--&lt; ACK, $w=0$</td>
<td>$</td>
</tr>
</tbody>
</table>

End
Item 2

- **Window mode – ACK “always”**
  - **MAX_FRAG_RETRIES**
    - Maximum number of retries for a specific fragment reported in an ACK to be lost
  - **MAX_ACK_REQUESTS**
    - Maximum number of requests for a specific ACK
    - Might be e.g. a greater value than MAX_FRAG_RETRIES (e.g. for very large windows)

- **Proposal: simplify into a single parameter**
  - Actually, only MAX_ACK_REQUESTS applies
Item 3

• MAX_FRAG_RETRIES
  – Currently only defined for ACK “always”
  – ACK on error:
    • Currently, the receiver sends an ACK and expects frag retries
    • Might also be a way to deplete the battery of an end-device?
  – Some options:
    • a) Define MAX_FRAG_RETRIES also in ACK on error?
      – If yes, allow up to “infinite” retries and defer to SCHC over foo documents?
    • b) Handle this only at an implementation level?

No feedback from the list so far…
Thanks!

Comments?

Authors:
Ana Minaburo <ana@ackl.io>
Laurent Toutain <laurent.toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
SCHC

Fragmentation optimization

Authors:
Ana Minaburo <ana@ackl.io>
Laurent Toutain <laurent.toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
Fragmentation optimization

Notation:

- All-0 frag: all the bits of FCN are set to 0
- All-1 frag: all the bits of FCN are set to 1
- All-x frag: either All-0 or All-1 frag
Padding issue

- LPWAN is aligned on bytes
- SCHC ruleid/Dtag/W/LCN may break this alignment.

Sigfox downlink (padding > 8 bits): must be solve in SCHC or bitmap frame
Problem with Abort message

The entity (either the fragment sender or the fragment receiver) that triggers abortion transmits to the other endpoint a format that only comprises a Rule ID (of size R bits), which signals abortion of all on-going fragmented IPv6 packet transmissions.

SCHC: RRDD  WNNN
Abort: RRPP  PPPP

• Possible Confusion
• Proposal: use empty All-1 frag
  – Regular All-1 frag carries the MIC
  – Length $<$ MIC length $=>$ Abort
  – Do not abort all the Dtag transmissions
  – Work only in sender to receiver direction
    • Is it enough, can receiver trigger an Abort?
All-x frag optimization

• All-x frag are used to request ack from receiver
  – In normal fragmentation process All-x should contain:
    • 1 byte for All-0 frag
    • MIC-size + 1 byte for All-1 frag
  – Sending empty ack for bitmap transmission:
    • All-0 frag (SCHC+pad [no data]) or
    • All-1 frag (SCHC+MIC+pad [no data])
      – Empty All-1 frag (SCHC+pad) is abort message.
Ack Optimization

• Bitmap structure:

<------------------------ 2^N -1 bits ------------------------>

• Bitmap optimization:

Does not work if error are on left bitmap part, but full received window can be compressed

Impose always 1 Byte of bitmap : 0 Byte of bitmap => abort ?
Interim, September 12th, 2017

Receiver FSM

init

W=0
F=0
MIC=false

Wait fragment

recvW != W
------
ignore

If MIC is incorrect, some frag are lost (and resent in the last window).

recvW == W &
recvFCN != all-X
------
FN = frag_number(recvFCN, F)
set_bitmap(PCN)

Recieved of frag (except the window last one)

No MIC recv
Or
MIC incorrect

Check MIC

MIC recv
and
MIC correct

Abort msg

W changed, the sender accept the bitmap

End of the packet. Extract MIC from data and check integrity.

The only way for the receiver to know that the full window has been received in the W change. The sender is not obliged to send all the frag.

For the last window only MIC integrity can help.

Wait frag new window ?

recvW != W
------
W = recvW

W says the same, retransmission of lost frag

Send bitmap

recvW == W &
recvFCN == all-0
------
FN = frag_number(recvFCN, F)
If not optimized set_bitmap(PCN)
Send bitmap

Process fragment

recvW == W &
recvFCN == all-1
------
FN = frag_number(recvFCN, F)
Send bitmap

recvW == W &
recvFCN != all
------
X

End of the packet. Extract MIC from data and check integrity.

The only way for the receiver to know that the full window has been received in the W change. The sender is not obliged to send all the frag.

For the last window only MIC integrity can help.

Send bitmap

End of the packet. Extract MIC from data and check integrity.

The only way for the receiver to know that the full window has been received in the W change. The sender is not obliged to send all the frag.

For the last window only MIC integrity can help.

Send bitmap

End of the packet. Extract MIC from data and check integrity.

The only way for the receiver to know that the full window has been received in the W change. The sender is not obliged to send all the frag.

For the last window only MIC integrity can help.

Send bitmap

End of the packet. Extract MIC from data and check integrity.

The only way for the receiver to know that the full window has been received in the W change. The sender is not obliged to send all the frag.

For the last window only MIC integrity can help.

Send bitmap
Sender FSM

init

FCN = start value
W = 0

fragment to send

FCN == 0
----
Set local_bitmap
Send fragment
Trigger Timer

No more frag
----

Last packet fragment
----
Set local_bitmap
Send fragment
Trigger Timer

Other fragment
----
Send fragment
Set local_bitmap
FCN = FCN - 1

End

Wait bitmap

Local_bitmap == recv_bitmap
-----
Stop timer

Local_bitmap != recv_bitmap
-----
Stop timer
W = 1-W
Local_bitmap = 0...0

Send missing frag

Send (optimized) All-0 or All-1 frag

Compute retrans

Attemp <= MAX_FRAG_RETRY
-----

Attemp > MAX_FRAG_RETRY
-----
Send abort

End

Interim, September 12th, 2017
AOB ?