LPWAN WG

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

AD: Suresh Krishnan
<suresh@kaloom.com>
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)
Reminder:

Minutes are taken *
This meeting might be 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

<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:05</td>
<td>Opening, agenda bashing (Chairs)</td>
<td>5mn</td>
</tr>
<tr>
<td></td>
<td>• Note-Well, Scribes, Agenda Bashing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Status of drafts</td>
<td></td>
</tr>
<tr>
<td>17:10</td>
<td>SCHC Updates since last Interim - Dominique</td>
<td>15mn</td>
</tr>
<tr>
<td>17:25</td>
<td>Open window Full proposal - Laurent</td>
<td>15mn</td>
</tr>
<tr>
<td>17:40</td>
<td>Open discussion on Fragmentation</td>
<td>15mn</td>
</tr>
<tr>
<td>17:55</td>
<td>AOB</td>
<td>QS</td>
</tr>
</tbody>
</table>
Upcoming IETF 103

**2018-09-17 (Monday):** Early Bird registration and payment cut-off at UTC 23:59.

**2018-09-21 (Friday):** Cut-off date for requests to schedule Working Group Meetings at UTC 23:59.
Changes to the draft

Dominique
draft-ietf-lpwan-ipv6-static-context-hc-16

Draft status

Authors:
Laurent Toutain <Laurent.Toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
Ana Minaburo <ana@acklio.io>
Dominique Barthel <dominique.barthel@orange.com>
Note

• As always, all changes can be checked out at https://github.com/lp-wan/ip-compression/commits/master
  – Itemized commits
  – (hopefully) explicit commit messages
  – on-line diffs available
Few changes since last interim

• Removed F/R terminology from Section 4
  – Was largely redundant with 8.2 SCHC F/R tools
• Made W field M bits wide
  – Anticipation for work by Laurent
• Rewrote No-Ack section
  – Comments by Carles, Juan Carlos, Laurent
No-Ack

• From the toolbox, which tools are used and which are not?

• Educational reading [https://www.ietf.org/blog/how-read-rfc/](https://www.ietf.org/blog/how-read-rfc/)
  – Specify protocol from sender and receiver behavior

• Sender-Abort: MAY be sent, MAY be processed

• Receiver-Abort? Is No-Ack strictly unidirectional?
Next steps

• MIC optional: 2 supports (Juan Carlos, Pascal). Decision? Ticket?
  – Finish implementing change in draft
• Implement text for Ack-Always?
  – Still needed if new Ack-on-Error is adopted?
• Implement text for Ack-on-Error
• Resolve 7 points under discussion with Charlie
• Awaiting WG decision on other pending decisions
  – State Machines in Appendix or in normative section: chairs?
• Housekeeping chores
  – Remove 5.2 SCHC F/R message formats
  – Check for inadvertently-dropped comments
• Oct 22nd is IETF103 draft publication cut-off date
Thank you!
Fragmentation: Open the window

Ana Minaburo
Laurent Toutain

Interim, September 19th, 2018
Current state: Ack Always

1 bit is enough to avoid ambiguities between windows:
- retransmission: window remains the same
- Next window: (window number + 1) % 2
Current state: Ack on Error
Current state: Ack on Error

Send Ack only when error is detected:
- FCN out of sequence
- Repeat it if window is increased

Ambiguity since the window number rolls back
Open the window

- Allow larger size than 1 for windows
Which size?

- Largest packet: 1500 bytes
- Fragment size: 8 Bytes
- Fragment number: 188
- FCN on 3 bits, 7 fragments by window: 27 windows
- 5 bits to code the window number
  - Window + FCN size : 1 byte
Other vision

• One big window of 188 fragments
• Bitmap is divided into 32 parts coded in:
  – Log + unary
• Limit the bitmap size in ack message
  – Bitmap encoding can still be used
• Trade-off between:
  – Transmission: larger windows are better
  – Ack: size limitation
• Asynchronicity between sending and acknowledgment.
Current state: Ack on Error

Send Ack only when error is detected:
- FCN out of sequence
- Repeat it if window is increased

No Ambiguity: Window number is unique
Current state: Ack on Error

Send Ack only when error is detected:
- FCN out of sequence
- Repeat it if window is increased

No Ambiguity: Window number is unique

All-1 window is always ACKed
Ack queue

All-0
All-1
Ack request

Window full

W=0
W=2
W=5
W=7
Which window size?

- Limit the window size:
  - A window number with all bit set to 1 has the same behavior as a All-1 window.
  - Must be acked before roll back
  - Ack Always differs:
    - Ack for window 0 can be lost,
    - Ack for window 1 will never be lost

- Don’t impose limits:
  - Simpler state machine
  - Have a rule for big packet (not commonly used)
  - Have a rule for smaller packet (more efficient)
    - 400 bytes with 8 byte long fragment and FCN size of 3 => 3 bits
    - 1500 bytes with 50 byte long fragment and FCN size of 3 => 3 bits
SCHC NOT-IMPLEMENTED RULE

Alexander Pelov <a@ackl.io>

LPWAN interim, 5th of September 2018
Interoperability and Extensibility of SCHC

• What happens when tomorrow we have SCHC-over-FOO and SCHC-over-BAR, and they try to talk to each-other?
  – Need to know what flavor of SCHC each supports
  – At least one of them needs to support two SCHC flavors

• More pragmatically – differentiating between SCHC-Baseline and SCHC-Extended
  – We seem to be trying to improve parts of the Fragmentation
  – Stable, working versions are there:
    • NO-ACK
    • ACK-ALWAYS

• Ideally, this is solved with having the full device context
  – We don’t have this one yet..
  – And we need extension mechanism for Fragmentation
Proposal

• Add a paragraph in the SCHC IP/UDP that:
  – Every technology-specific document MUST specify a Rule ID which indicates to the sender that a message it has issued does not match any SCHC Rule ID on the receiver, or that the receiver doesn’t implement that feature (« NOT-IMPLEMENTED »)

• Dream extension one:
  – Have a SCHC-Minimal document, which will provide several basic Rule IDs and will be endorsed by all baseline technologies (thus, a generic SCHC Compressor/Decompressor can talk to any LoRaWAN/Sigfox/NB-IoT/Wi-SUN device without prior knowledge)

• Dream extension two:
  – Specify that the NOT-IMPLEMENTED Rule ID MUST BE 11111101
Stop seeking Fragmentation perfection!

- Need to provide mechanism to solve interop issues

- Very interesting Fragmentation proposals
  - Changing window size
  - Advanced ACK-ON-ERR
  - ...
  - Which may or may not be available accross different technologies

- Ship today and take time to do them correctly
BONUS
(don’t read beyond this point if you are faint-hearted)

Alexander Pelov <a@ackl.io>
NOT-IMPLEMENTED format

- Frame format
  - NOT-IMPLEMENTED Rule ID + 16 bits
    - 8 bits header
      - 3 bits - reserved (must be 000)
      - 5 bits - length of Rule ID snippet in bits (31 bits max)
    - 0...31 leading bits of offending SCHC Packet (RuleID+...)

- The goal is for the sender to be able to match a NOT-IMPLEMENTED message to the Rule ID that provoked it

- Alternatively, the NOT-IMPLEMENTED RuleID can be named ERROR RuleID, and the 3 bits be used for identification
  - 000 - Not supported / not-implemented
  - 001 - Error in compression/decompression
  - 010 - Error in fragmentation/defragmentation
Bonus

• What could get into SCHC-Minimal?
  – Rule ID:
    • 1000 – Raw (No compression / No fragmentation)
    • 1001 – Simple IPv6 compression
    • 1010 – Simple CoAP
    • 1011 – Simple CoAP POST
    • 1100 – NO-ACK Fragmentation, Fragment Size 7 bytes
    • 1110 – ACK-ALWAYS Fragmentation, Fragment Size 7 bytes
    • 1101 – Reserved
    • 1111101 – NOT-IMPLEMENTED
AOB ?