LPWAN WG

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

AD: Suresh Krishnan
<suresh@kaloom.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)

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

17:05	Opening, agenda bashing (Chairs) • Note-Well, Scribes, Agenda Bashing • Status of drafts	5mn
17:10	SCHC Updates since last Interim - Dominique	15mn
17:25	Open window Full proposal - Laurent	15mn
17:40	Open discussion on Fragmentation	15mn
17:55	AOB	QS

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.

lpwan	2:00	80	Pascal Thubert	Suresh Krishnan	detnet core **lpwan** rift roll **6lo intarea** **lwig** netconf bier cbor **6man** anima Pascal Thubert, Suresh Krishnan, Alexander Pelov

Open Full Window

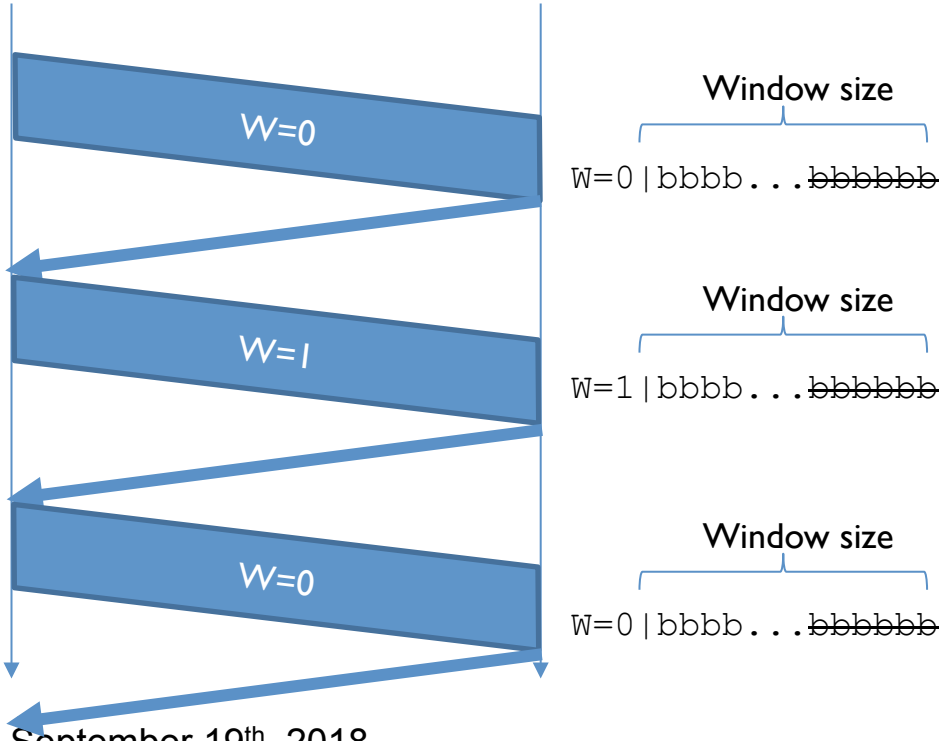
Laurent and Ana

Changes to the draft

Dominique

Interim, September 19th, 2018

Current state: Ack Always



Window size

W=0|bbbb...~~bbbbbb~~

Window size

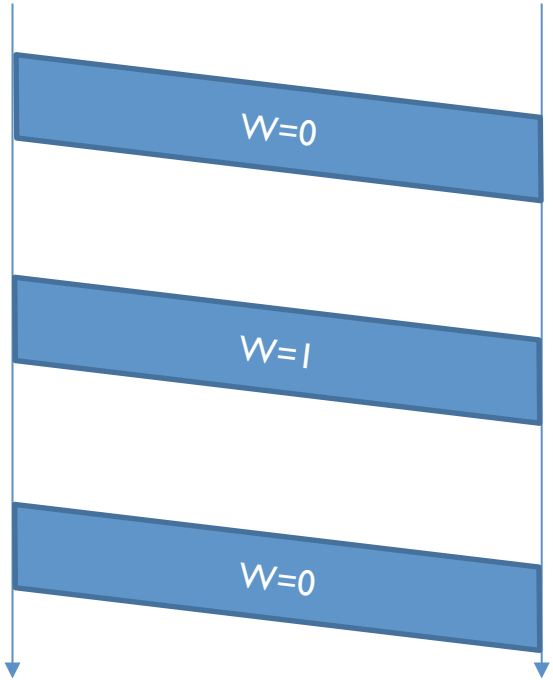
W=1|bbbb...~~bbbbbb~~

Window size

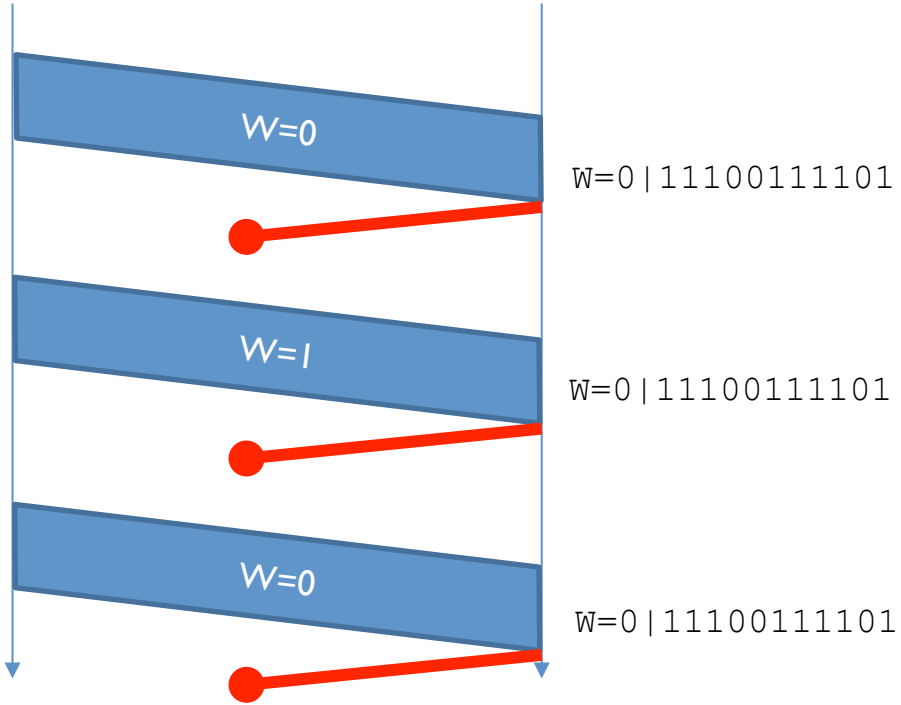
W=0|bbbb...~~bbbbbb~~

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

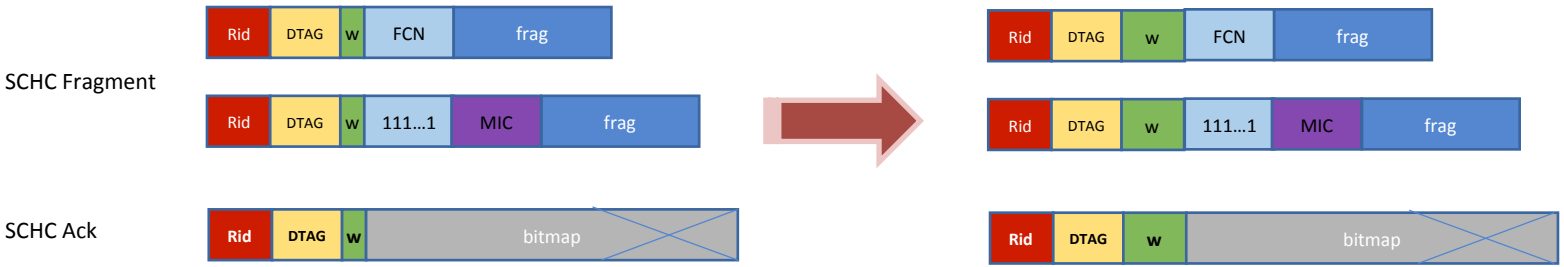


Send Ack only when error is detected:
- FCN out of sequence
- Repeat it if window is increased

Ambiguity since the window number rolls back

Open the window

- Allow larger size than 1 for windows



Which size ?

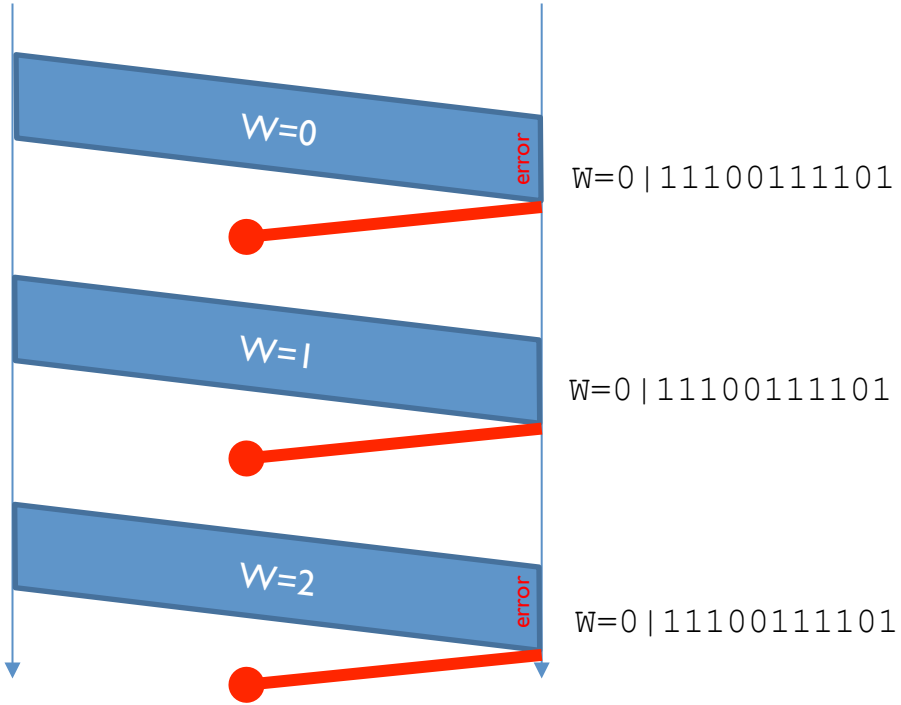
- Largest packet: 1500 bytes
- Fragment size: 8 Bytes
- Fragment number: 188
- FCN on 3 bits, 7 fragments by window: 27 windows
- 5 bits to code the window number
  - Window + FCN size : 1 byte

Other vision

Largest packet: 1500 bytes
Fragment size: 8 Bytes
Fragment number: 188

- 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



W=0|11100111101

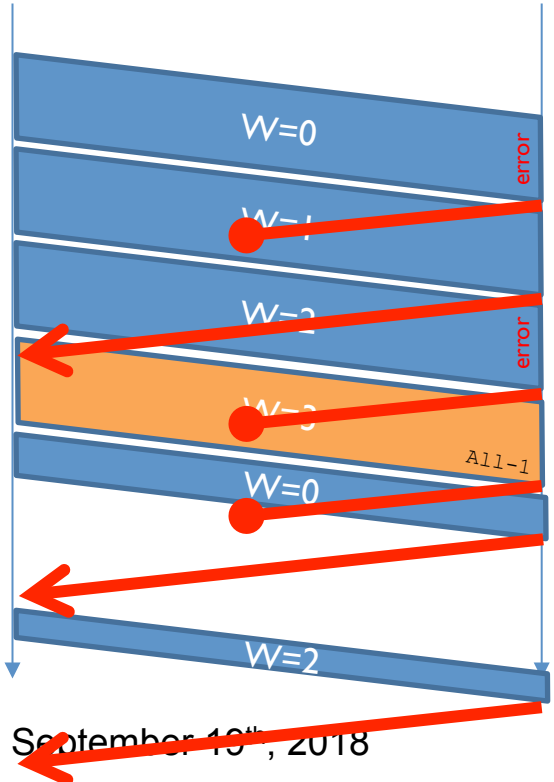
W=0|11100111101

W=0|11100111101

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



W=0|11100111101

W=0|11100111101

W=0|11100111101

W=0|11100111101

W=2|11101001101

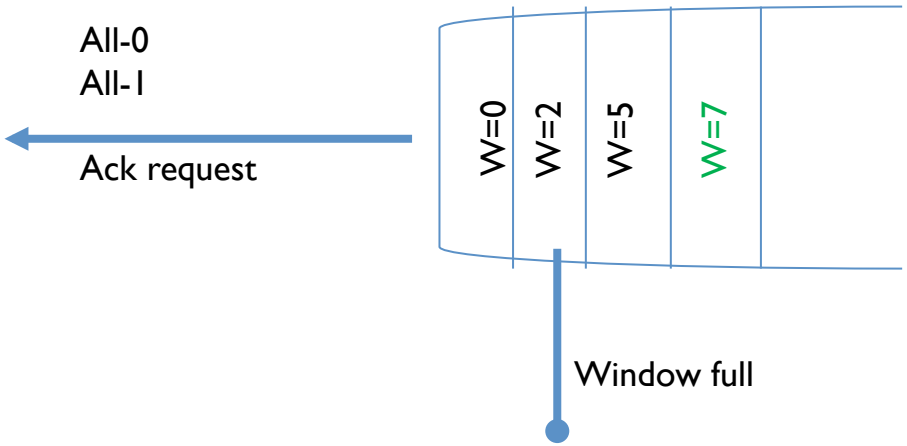
W=3|MIC OK

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



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:
  - Simplier 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>

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 1111101~~

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 identifiaction
  - 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
    - 11111101 – NOT-IMPLEMENTED

AOB ?