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 IETF 102 - Dominque	10mn
17:20	ACK-on-Error - Juan Carlos, Carles, Edgar	30mn
17:50	Discussion	10mn
18:00	AOB	QS

Reminder since IETF 102

- WGLC on Compression Section – success
- Fragmentation section needs to be improved
  - Text normative (confirm on ML?) – have the State Machine in the appendix

Fragmentation meeting @IETF102

- What to do with ACK-on-Error?
  - Keep it as it is (no fundamental problem with it)
  - Provide minor modifications
    - Need to understand the consequences (a modification can be an improvement for some cases and can be a degradation of for others)
  - Move to a separate document (for major modifications)

draft-ietf-lpwan-ipv6-static-context-hc-16
Draft rework, status post IETF102

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 Well

- This presentation is about **editorial** changes
- This presentation is ***not*** about **protocol** changes
- « Don't shoot the pianist »

- All changes can be checked out at https://github.com/lp-wan/ip-compression/commits/master
  - Itemized commits, explicit commit messages, on-line diffs

Review by Charlie Perkins

- Thanks again, Charlie, for this thorough review!
- I applied all suggested edits, except
  - Section 8 Fragmentation/Reassembly
    - This section was to be reworked from the ground up
  - 7 points that I questioned or requested explanation for
    - Mail sent out Aug 29th

Few other changes

- A few edits on July 19th-20th, while still at IETF102
  - Reworked « L2 Data Unit » with Edgar
  - Gave structure to Appendix D, because fragmentation optional (Charlie)
  - ECN bits elision in IPv6 SCHC compression (Lars)
  - MIC computation fix-up (Soichi)

Major rework of F/R section

- Before:
  - Algorithmic specification spread out/replicated in many sections
  - Hard to read, to update and keep consistent, hard to add a new mode
- After:
  - Description of « tools » is independant of algorithm
  - Modes specification can be updated/added more easily

New organization of F/R section

- Sections 8.1, 8.2 and 8.3 are a mode-agnostic substrate for 8.4
  - No algorithmics, just description
- Full algorithmic description of F/R modes in 8.4
  - Currently empty
  - To be filled with Juan Carlos'es and Carles'es contributions
- What to do with 8.5, 8.6 ?

Next steps (1/2)

- Fill-up No-Ack, Ack-Always sections
  - Thanks again to Juan Carlos and Carles for providing text!
  - I will adjust it to the new « substrate » provided in 8.1-8.3
  - I will check for possible gaps, overlaps, mismatches and report on them
- Ack-on-Error
  - As draft secretary, I will implement text when WG has agreed on the desired behavior
  - As an individual, will contribute to the discussion

Next steps (2/2)

- Resolve 7 points under discussion with Charlie
- Check for inadvertently-dropped comments
- Awaiting WG decision on other pending decisions
  - State Machines in Appendix or in normative section
  - Other pending issues? (see discussion slot in this meeting)
- I'm committed to swift publication of -17
  - mid-Sept, end of Sept, Oct 22nd (IETF103 cut-off)?

Thank you!

ACK-on-Error discussion

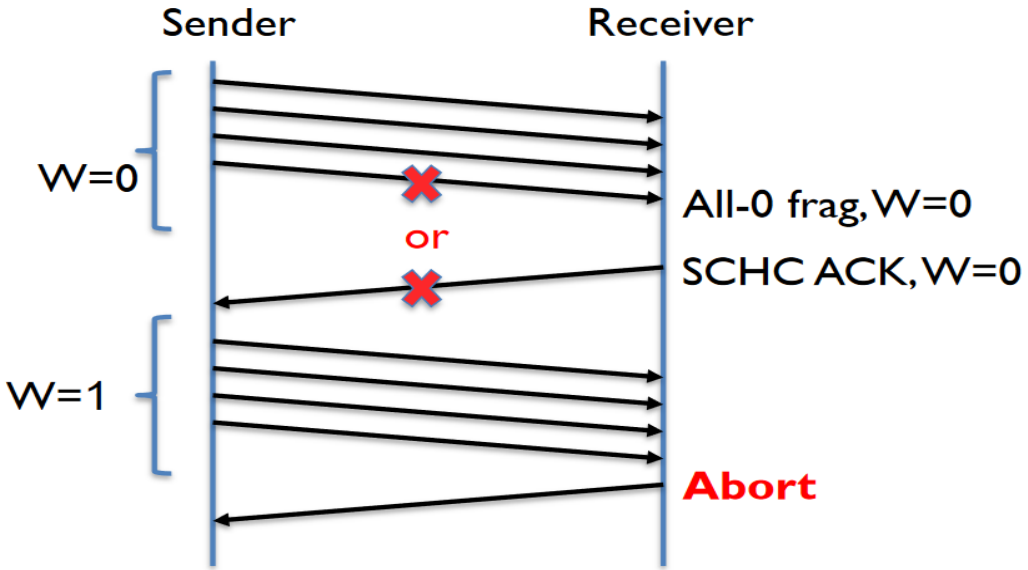
Juan Carlos Zúñiga <juancarlos.zuniga@sigfox.com>
Carles Gomez <carlesgo@entel.upc.edu>
Edgar Ramos <edgar.ramos@ericsson.com>

Probability of "Issue" in ACK-on-Error

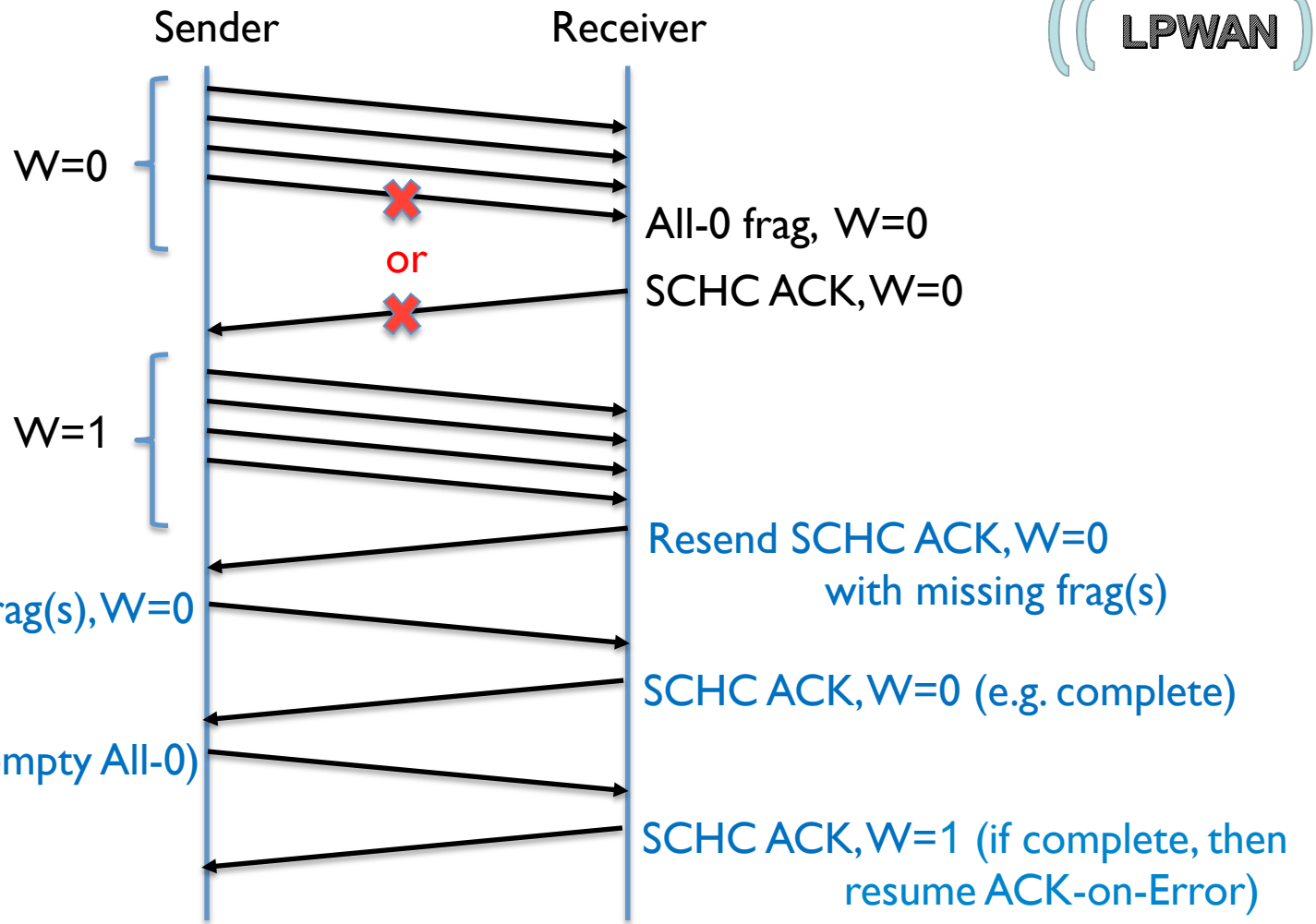
Numeric Analysis

The Issue

- Any of the two "red crosses" in JCZ's IETF 102 slide 5, leading to **packet transmission Abort (Issue)**



JCZ's proposal (slide 6, IETF 102) ACK-on-Error Proposed

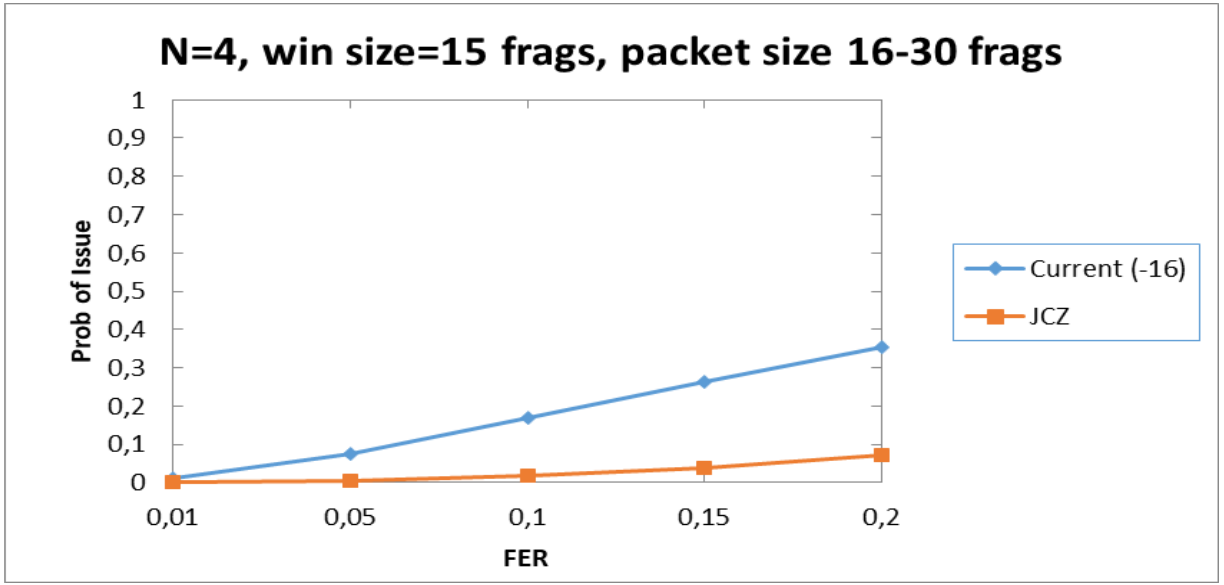


Goal and Assumptions

- Goal
  - To illustrate the **probability of Issue** for the current ACK-on-Error (-16) and JCZ's proposal
  - Simple mathematical analysis
    - To get a hint on how severe the problem may be
- Assumptions
  - Fragment losses are uncorrelated
  - Same error rate in uplink and downlink
  - Range of Fragment Error Rate (FER) values
    - Up to 0.2

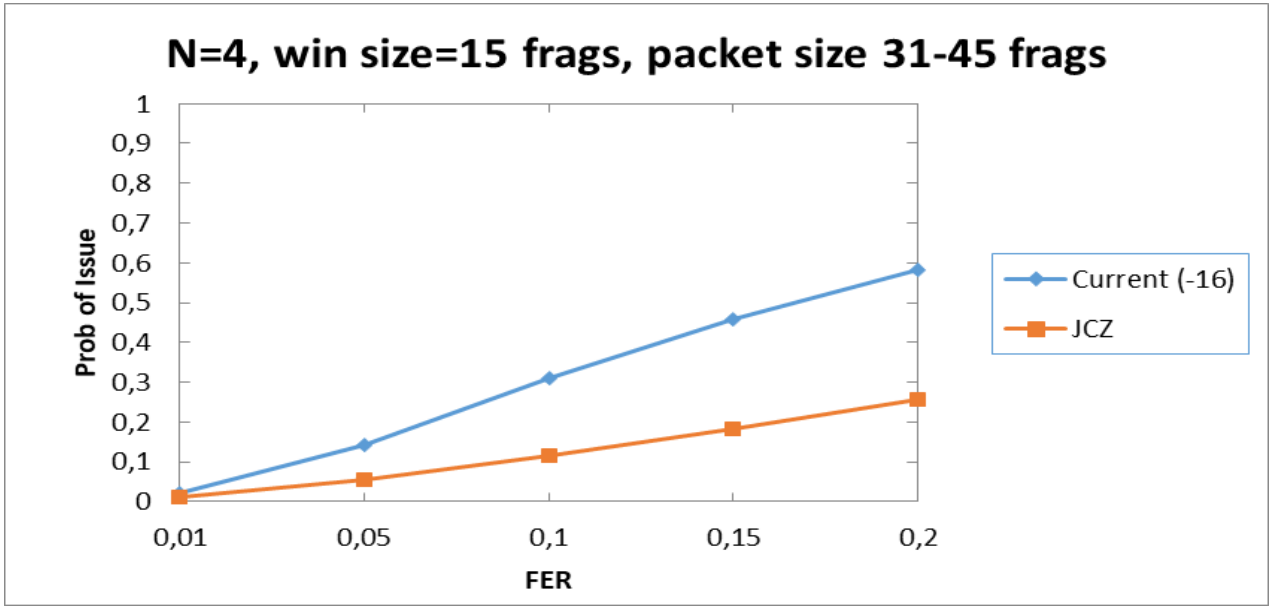
Results (I/IV)

- Prob of Issue = 0 for packets that fit a single window
- For a packet size between 1 and 2 windows:



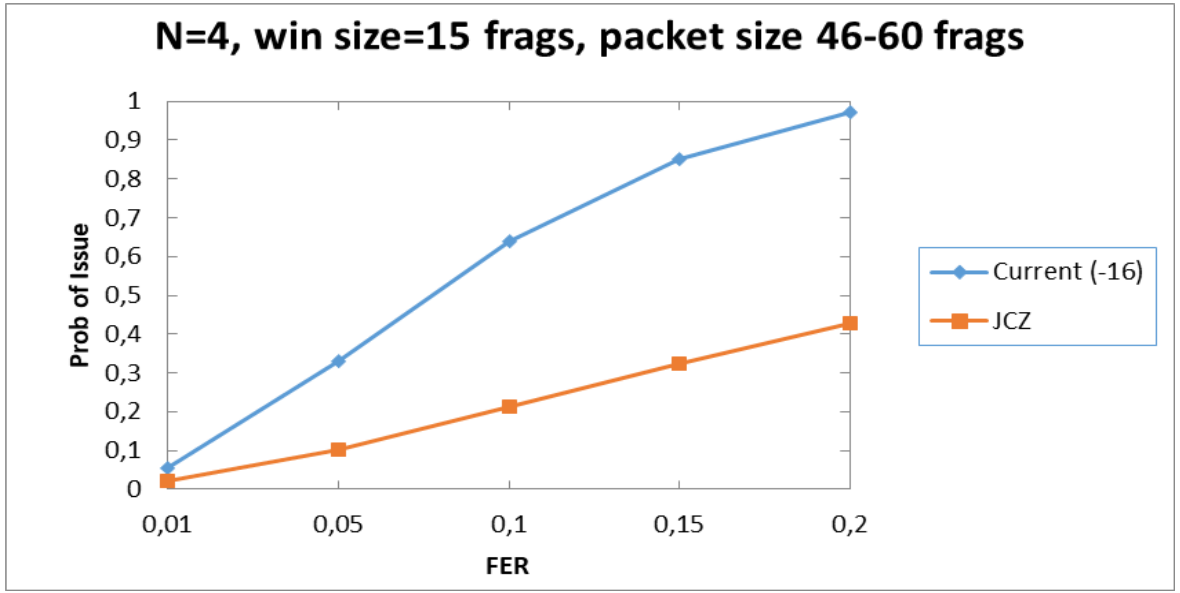
Results (II/IV)

- For a packet size between 2 and 3 windows



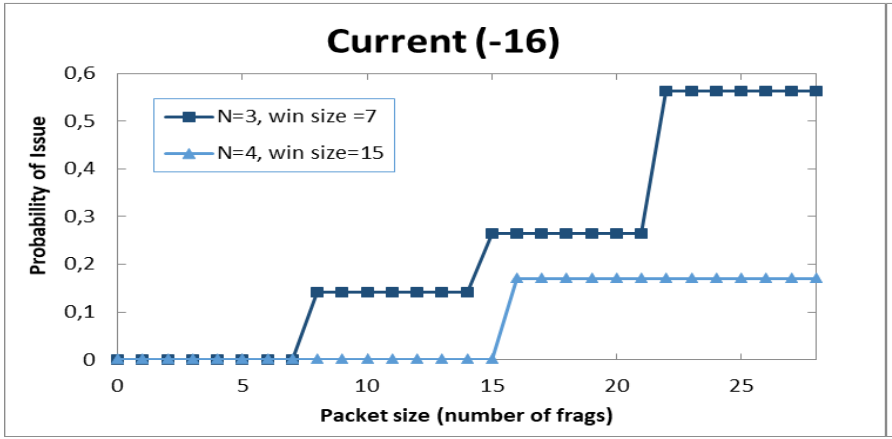
Results (III/IV)

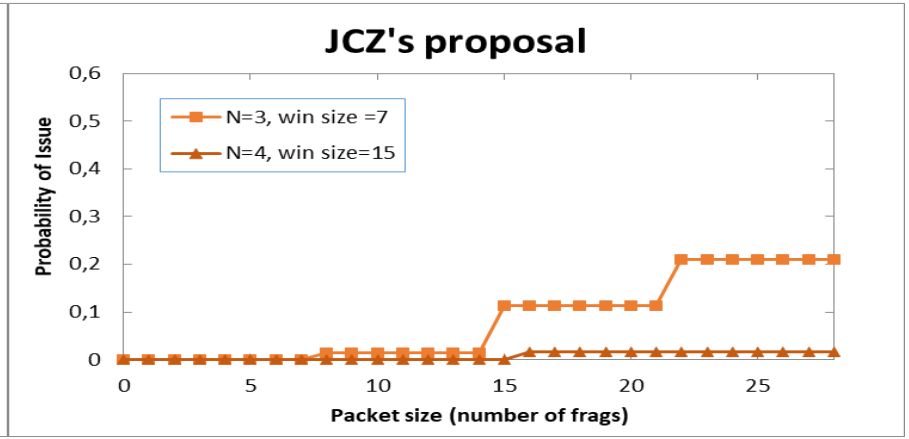
- For a packet size between 3 and 4 windows



Results (IV/IV)

- Impact of window size
  - FER=0.1 (i.e. 10%)





  - The probability of a real out-of-sync with the W bit can be reduced with a relatively larger window size, e.g. 15 (N=4)

State Machine

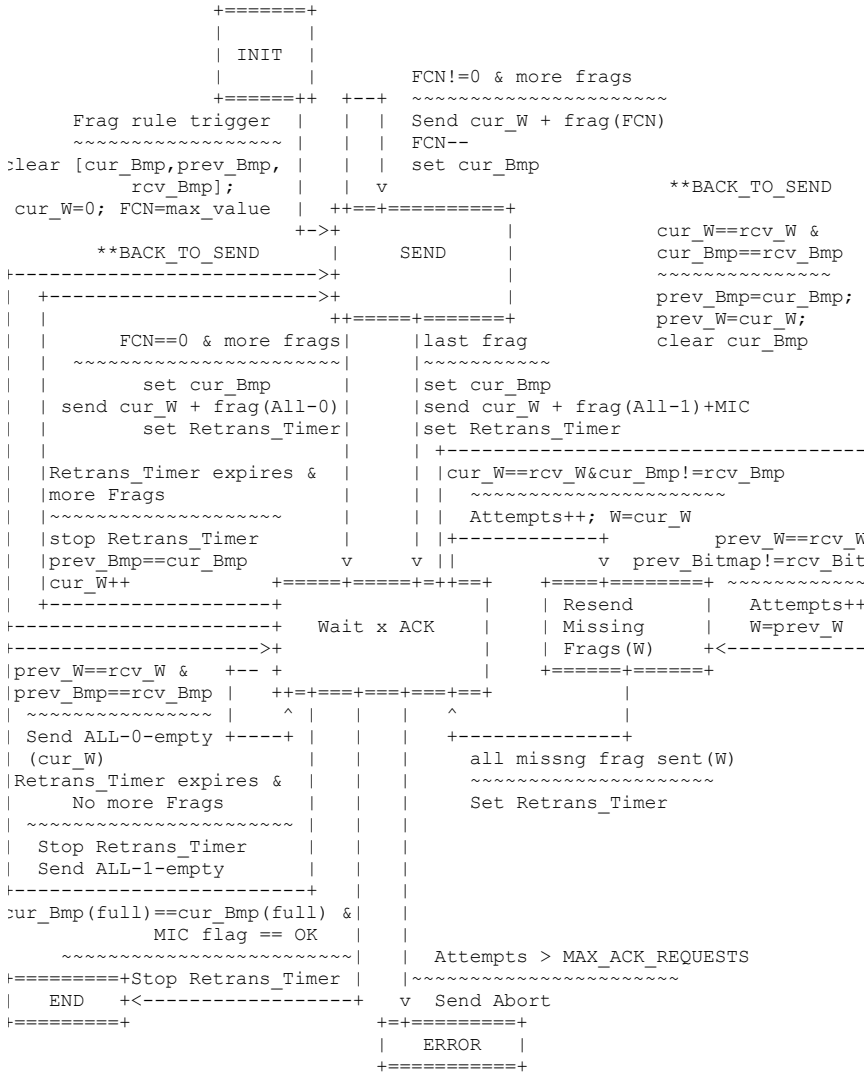
ACK-on-Error

As proposed at IETF 102

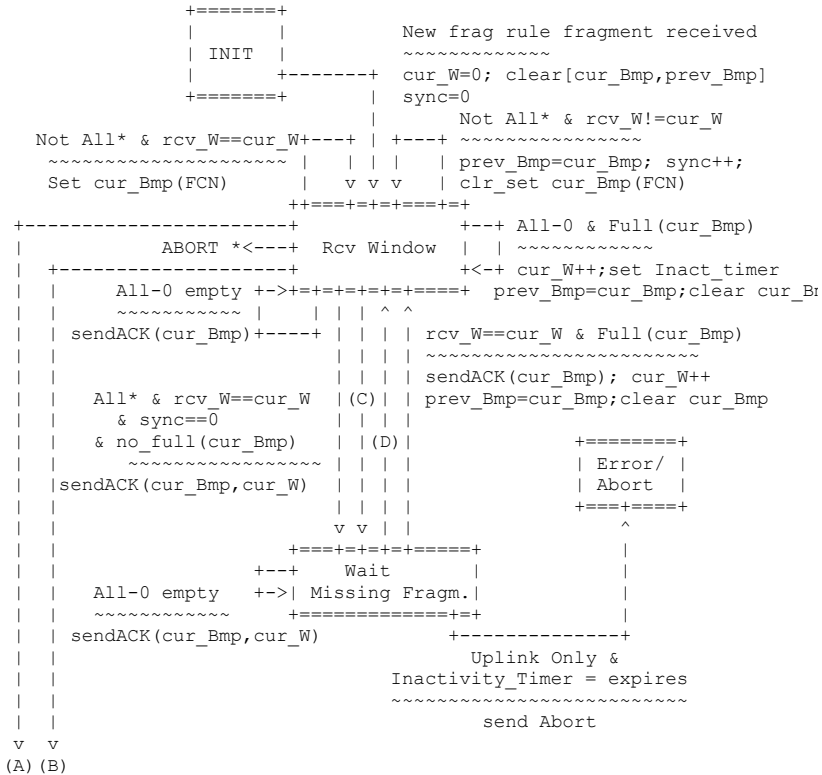
Modified State Machine

- Objectives
  - Implement the ACK-on-Error proposal as a minimal modification to the current State Machine (draft SCHC-16)
  - Changes implemented as per JCZ's proposal and text distributed on the mailing list
  - Maintain the same number of States

Sender


INIT
FCN!=0 & more frags
Frag rule trigger
Send cur_W + frag(FCN)
FCN--
set cur_Bmp
clear [cur_Bmp,prev_Bmp,
rcv_Bmp];
cur_W=0; FCN=max_value
**BACK_TO_SEND
SEND
**BACK_TO_SEND
cur_W==rcv_W &
cur_Bmp==rcv_Bmp
prev_Bmp=cur_Bmp;
prev_W=cur_W;
clear cur_Bmp
FCN==0 & more frags
last frag
set cur_Bmp
set cur_Bmp
send cur_W + frag(All-0)
set Retrans_Timer
send cur_W + frag(All-1)+MIC
set Retrans_Timer
Retrans_Timer expires &
more Frags
stop Retrans_Timer
prev_Bmp==cur_Bmp
cur_W++
cur_W==rcv_W&cur_Bmp!=rcv_Bmp
Attempts++; W=cur_W
prev_W==rcv_W
prev_Bitmap!=rcv_Bit
Wait x ACK
Resend
Missing
Frags(W)
Attempts++
W=prev_W
prev_W==rcv_W &
prev_Bmp==rcv_Bmp
Send ALL-0-empty
(cur_W)
all missng frag sent(W)
Retrans_Timer expires &
No more Frags
Set Retrans_Timer
Stop Retrans_Timer
Send ALL-1-empty
cur_Bmp(full)==cur_Bmp(full) &
MIC flag == OK
Attempts > MAX_ACK_REQUESTS
Stop Retrans_Timer
END
Send Abort
ERROR

Receiver (I/II)



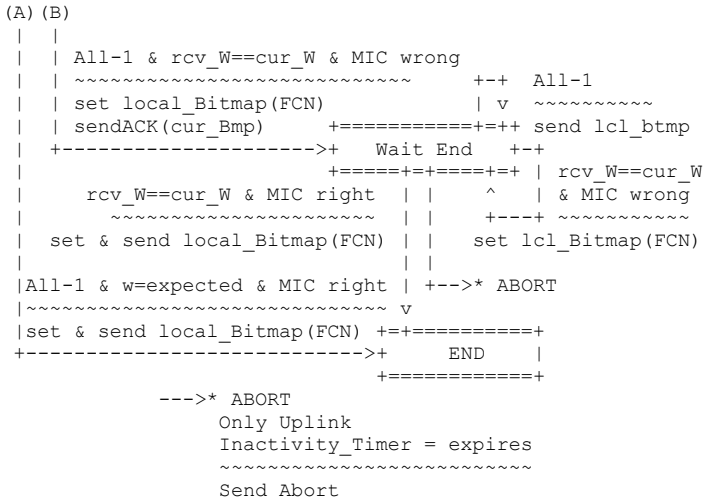
(C1) All* & rcv_W==cur_W
& sync==1
sendACK(prev_Bmp+prev_W)

(C2) All*||last_miss_frag &
& rcv_W!=cur_W
& sync==1 & Full(prev_Bmp)
sendACK(prev_Bmp+prev_W);
sync--

(D) rcv_W!=cur_W & sync==0
prev_Bmp=cur_Bmp; sync++;
clr_set cur_Bmp(FCN)

ABORT --> sync > 1
Send Abort

Receiver (II/II)



Alternative proposal

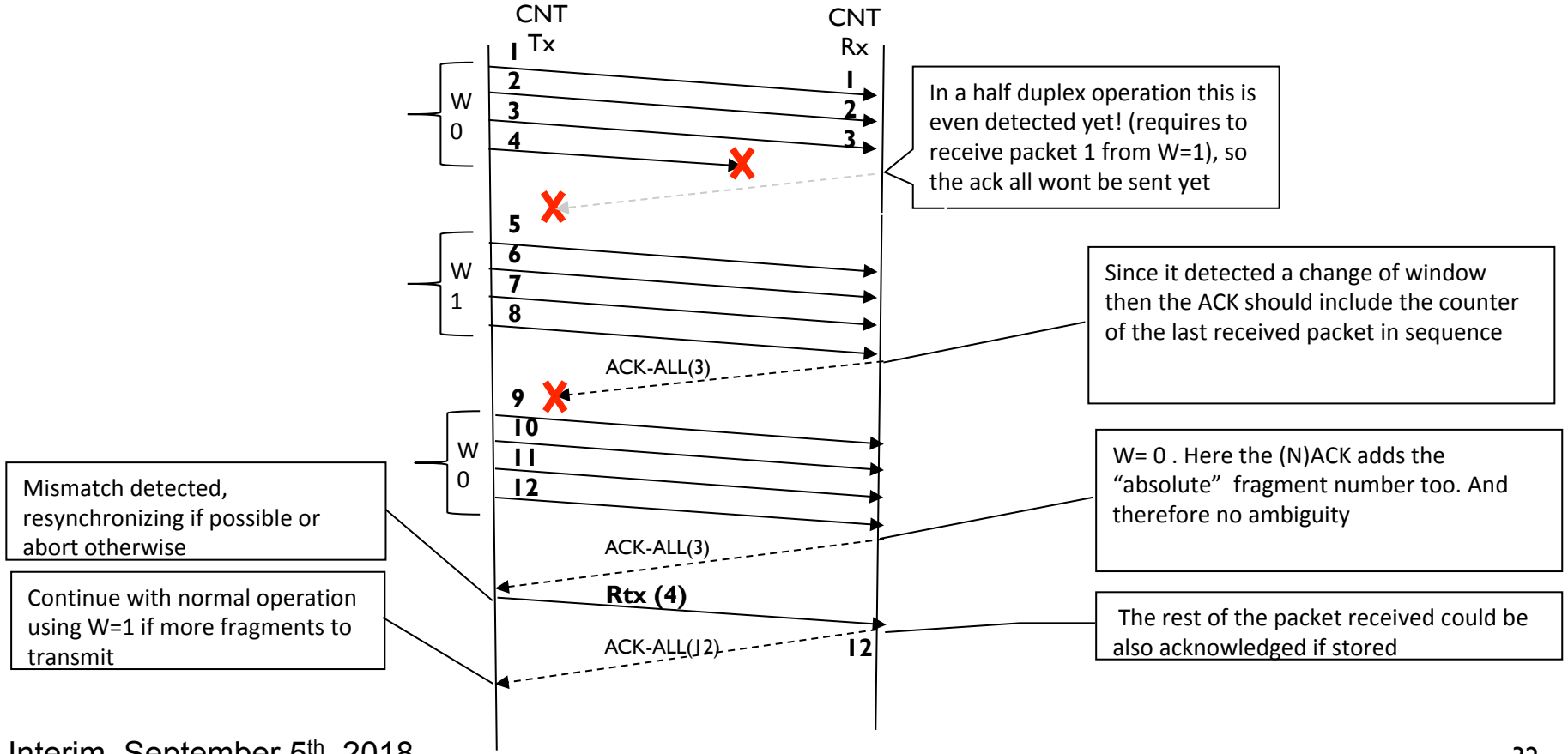
ACK-on-Error

Resync by fragment counting

Absolute fragment number

- Absolute count of fragments per IP packet
- Reset when new IP packet is processed
- Internal counter in Tx and Rx
- Transmitted in ACK only when window ambiguity (a packet span multiple windows)
  - The Rx transmits the counter of the last packet received in sequence
- The Tx keep fragments of at least a window otherwise left to implementation how many
  - Tx aborts if fragments are not available anymore

Example of operation with fragment counter



Fragment Counter - Considerations

- Memory requirements and how good recovery capability left to implementation
- Requires additional fields to signal the counter in the ACK
  - Only sent when danger of ambiguity and after recovery
  - Maximum number of fragments is not really determined.
    - Requires either fix size fields (maximum number of fragments is known)
    - Or a variable length indication

Conclusion

- Existing ACK-on-Error mode issue significantly decreases the performance
- Proposed text and State Machine changes based on draft-SCHC-16 can considerably mitigate the issue
- Other alternatives (e.g. absolute fragment number) would require changes to the protocol messages and to the state machine

AOB ?