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 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

Minute takers, jabber scribes

- Minutes
  - Etherpad: https://etherpad.tools.ietf.org/p/lpwan
  - Minute takers volunteers?

- Mailing list: lp-wan@ietf.org
  - To subscribe: https://www.ietf.org/mailman/listinfo/lp-wan

Agenda bashing

16:05 Opening, agenda bashing (Chairs) 5mn
- Note-Well, Scribes, Agenda Bashing
- Purpose of the meeting

16:10 SCHC compression (Dominique Barthel) 50mn

IETF

- Open for participation
  - Discussions happen on open mailing list
- In person meetings (3x per year)
  - Europe, North America, Asia
    - Dates and locations fixed several years ahead
    - Remote participation for all meetings
- Rough consensus and running code
  - 30 years of open standardization process, with proven track record of timely, high-quality technology standards creation
- WG Creation happens during a BOF session
  - A WG Charter defines what is the scope of work
  - Upon creation/recharteing, liaison statements are sent to other SDOs

IETF LPWAN WG

- First BOF (launching the group)
  - April, 2016 (Buenos Aires)

- Creation of group
  - July, 2016 (Berlin)
    - Lora Alliance: Geoff Mulligan, Alper Yegin
    - Sigfox: Juan-Carlos Zunig, Benoit Ponsard
    - 3GPP: Antti Atailainen
    - WI-SUN: Bob Heile

IETF LPWAN WG

- Since then
  - 200+ engineers have participated in LPWAN WG
    - Experience in 6LoWPAN, ROHC, IP, LoRaWAN, SIGFOX, 3GPP, etc.
  - 9 in-person meetings (during IETF)
  - 27 interim video meetings
  - Several side- and design-team meetings
  - 5 Hackathons, several implementations, several academic publications
  - RFC 8376 – LPWAN Overview – published

What next?

- Founding document – SCHC for IP/UDP
  - https://tools.ietf.org/id/draft-ietf-lpwan-ipv6-static-context-hc-17.txt
  - Workgroup Last Call (WGLC) (July 2018) - **Done** for **Compression**
  - WGLC on **Fragmentation**
    - Started November 11th
    - **Ending November 27th**
- If no major questions – sent to IESG for comments
  - If no major problems, standard published March-Apr
- But the work is essentially over!
- SCHC-over-*
  - SCHC-over-LoRaWAN
  - SCHC-over-Sigfox
  - SCHC-over-NB-IOT
  - SCHC-over-802.15.4w

IETF LPWAN fragmentation

Authors:
Laurent Toutain <Laurent.Toutain@imt-atlantique.fr>
Carles Gomez <carlesgo@entel.upc.edu>
Ana Minaburo <ana@ackl.io>
Dominique Barthel <dominique.barthel@orange.com>
Juan Carlos Zuniga <JuanCarlos.Zuniga@sigfox.com>

Presentation agenda

- LPWAN Working Group at IETF
- Header Compression and Fragmentation
- ACK-on-Error fragmentation mode

Work at IETF on LPWAN

IETF LPWAN WG

- IETF LPWAN Working Group
  - Established July 2016
- Main goal: develop CoAP/UDP/IPv6 support for LPWAN technologies
  - Header compression
  - Fragmentation
- Target technologies
  - LoRaWAN
  - Sigfox
  - NB-IoT
  - WiSUN

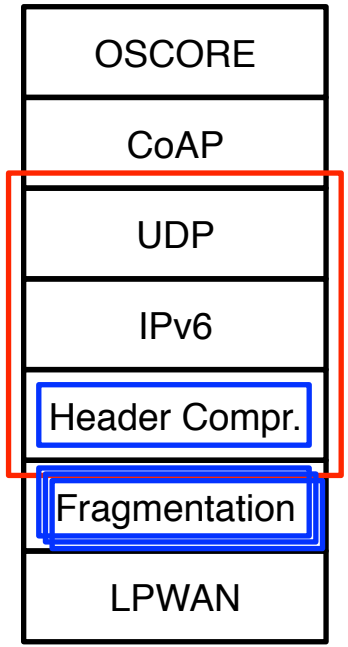
Open Source code in progress

- Hackathons at IETF meetings
- GitHub project
  - https://github.com/openschc
  - Python3/uPython
  - Arch, interfaces, code
  - Test scenarios

Header Compression and Fragmentation

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

- draft-ietf-lpwan-ipv6-static-context-hc-17
- Specification of a Header Compression engine
  - Generic engine, using Static Context (-> SCHC)
  - Stable since H1 2018
- Specification of UDP/IPv6 compression
  - Using the generic Header Compression engine
  - Stable since H1 2018
- Specification of a fragmentation protocol
  - A set of generic tools
  - 3 different "modes", that address different requirements
  - One mode extensively reworked over the summer

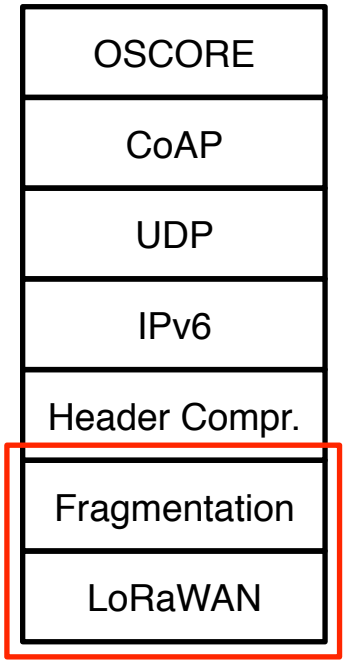


Fragmentation spec.

- Tools
- Message formats
- Algorithms (a.k.a. "modes")
  - No-ACK
  - ACK-Always
  - ACK-on-Error (reworked)

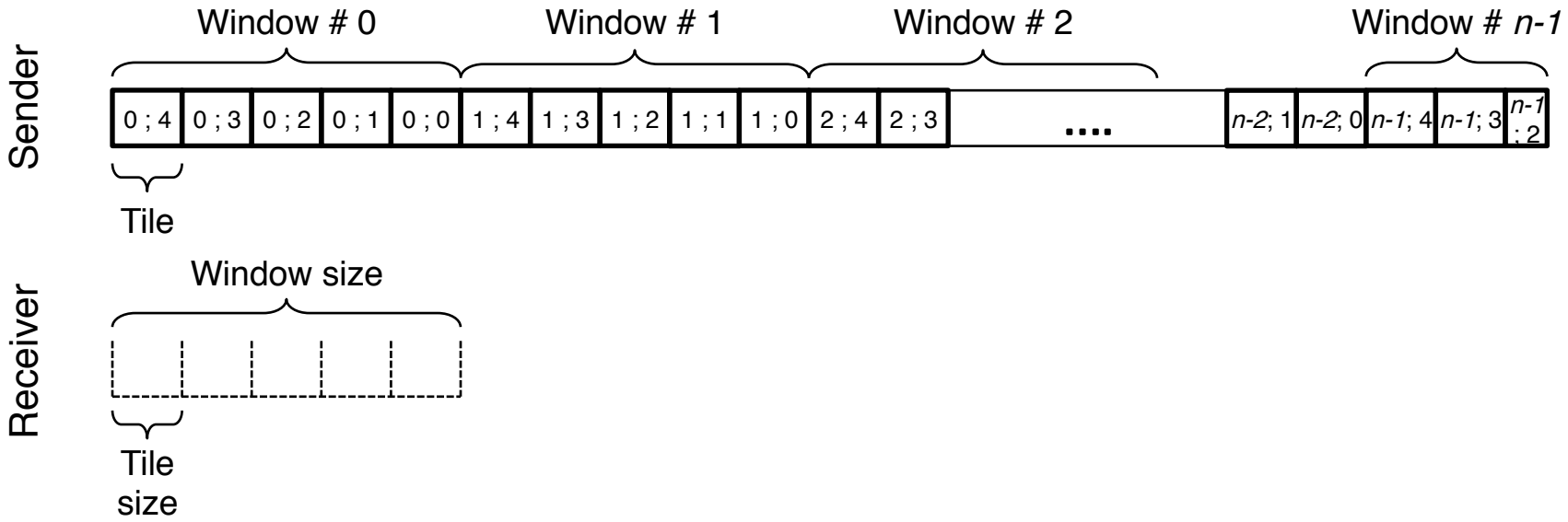
draft-petrov-lpwan-ipv6-schc-over-lorawan

- draft-petrov-lpwan-ipv6-schc-over-lorawan-02
  - N. Sornin, M. Coracin (Semtech), I. Petrov (Acklio), A. Yegin (Actility), J. Catalano (Kerlink), V. Audebert (EDF R&D)
- Application of generic engine to LoRaWAN
  - Parameters values
  - Selection of options
  - Alignment
- Work in progress

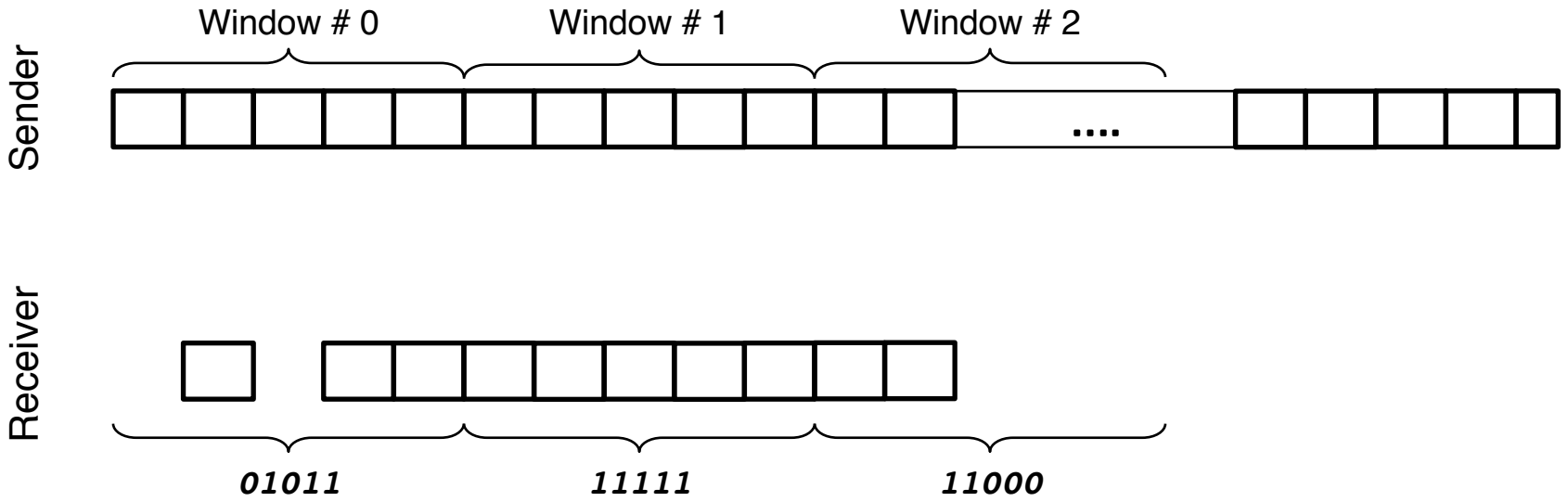


ACK-on-Error fragmentation mode

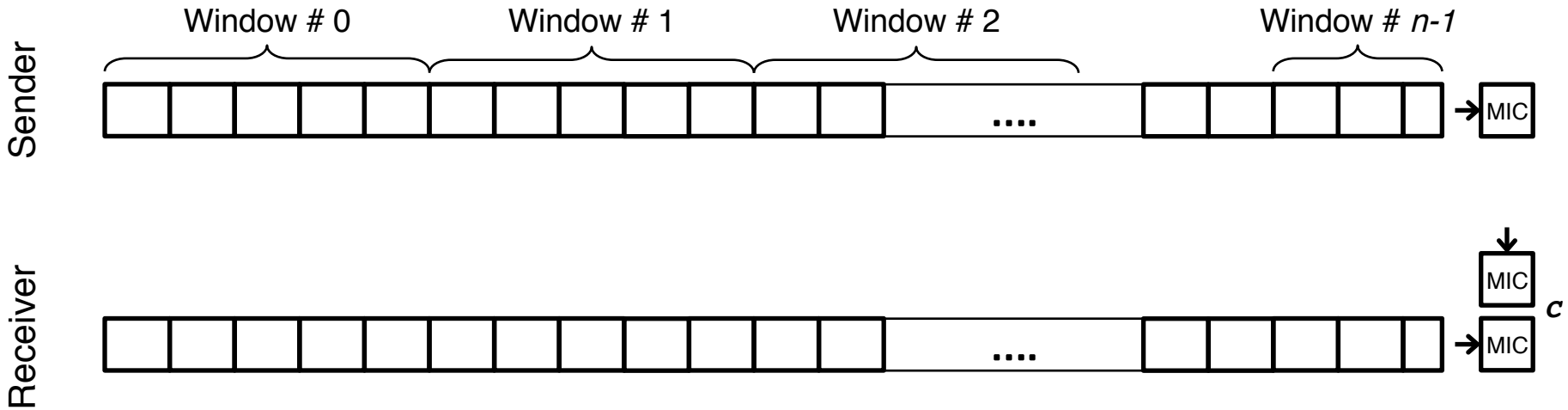
Tiles, windows of tiles



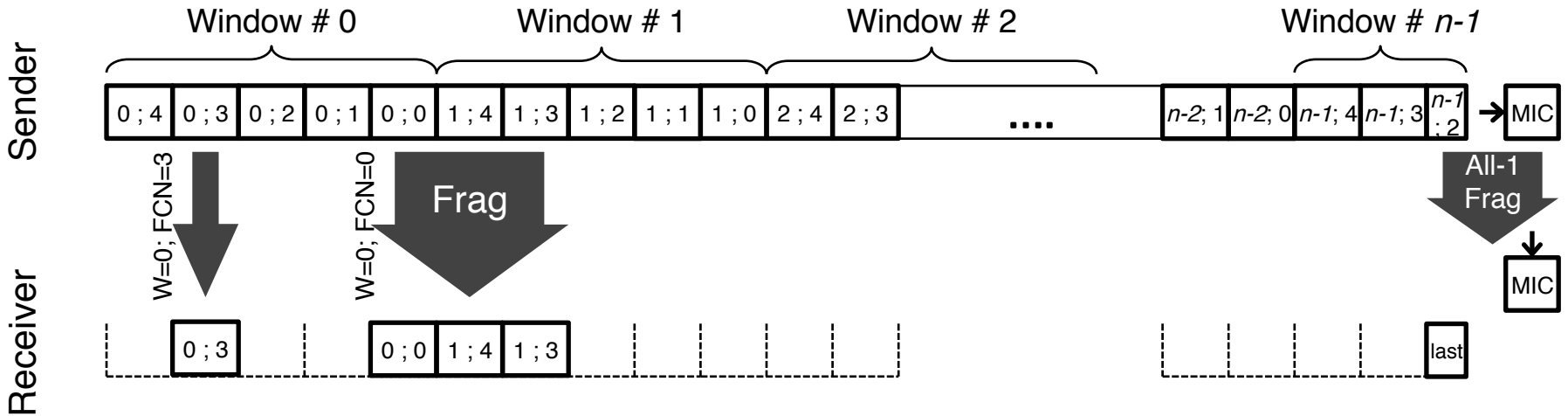
Bitmaps


Window # 0
Window # 1
Window # 2
Sender
....
Receiver
01011
11111
11000

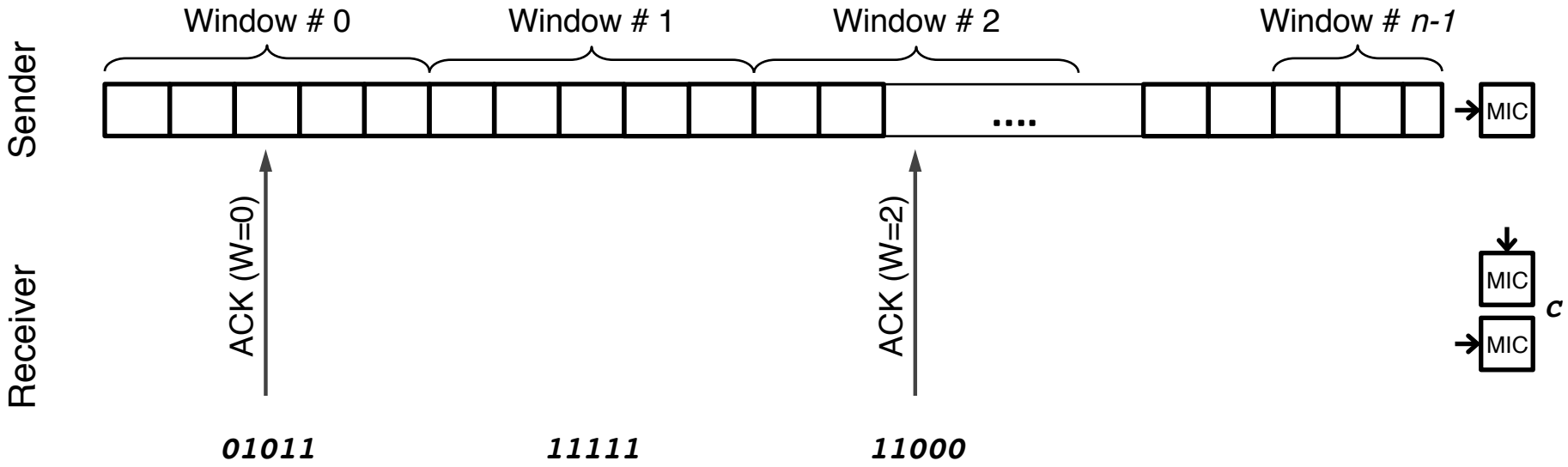
Message Integrity Check (MIC)



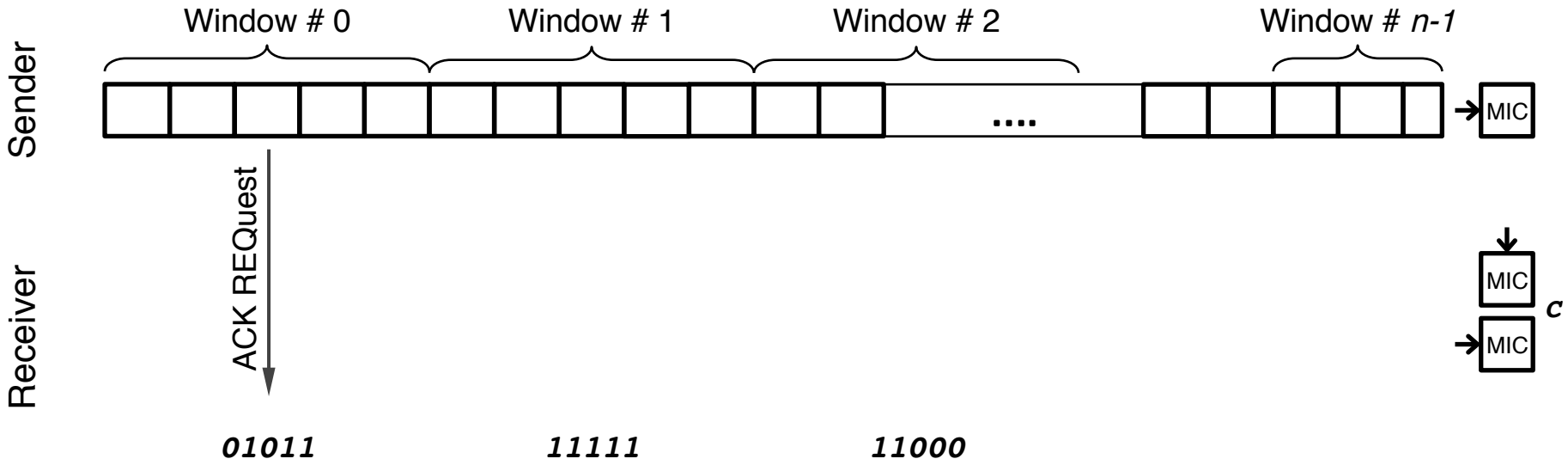
Fragment messages



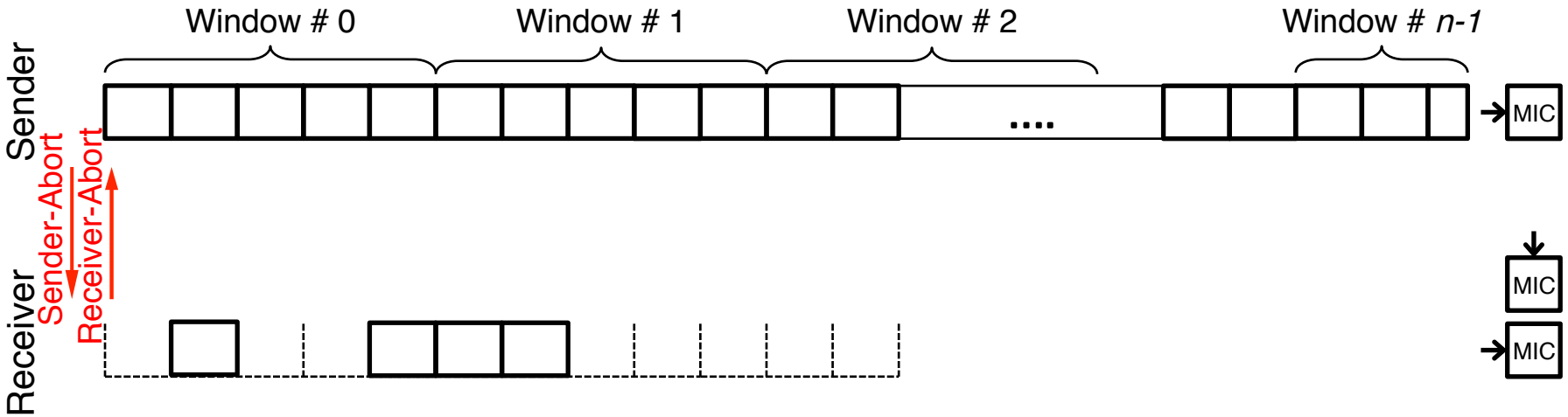
ACK messages



ACK REQ messages



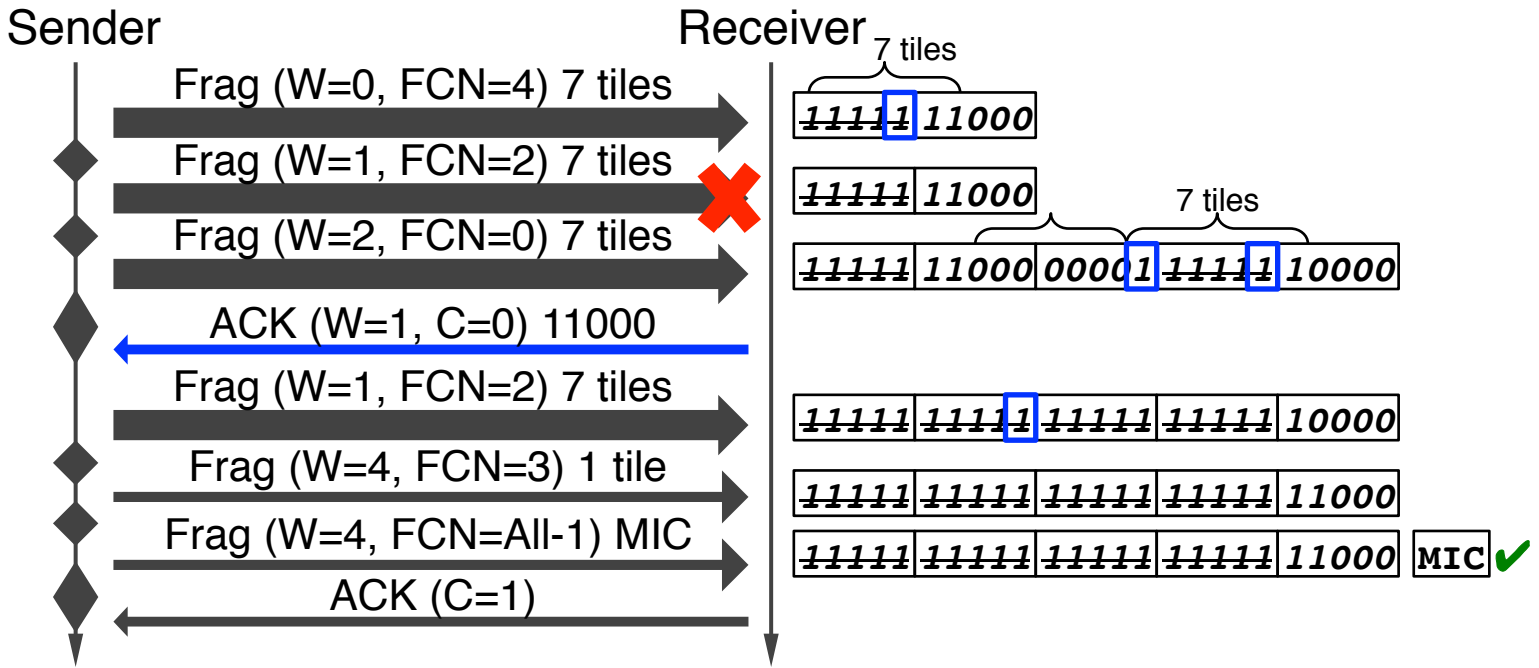
Abort messages



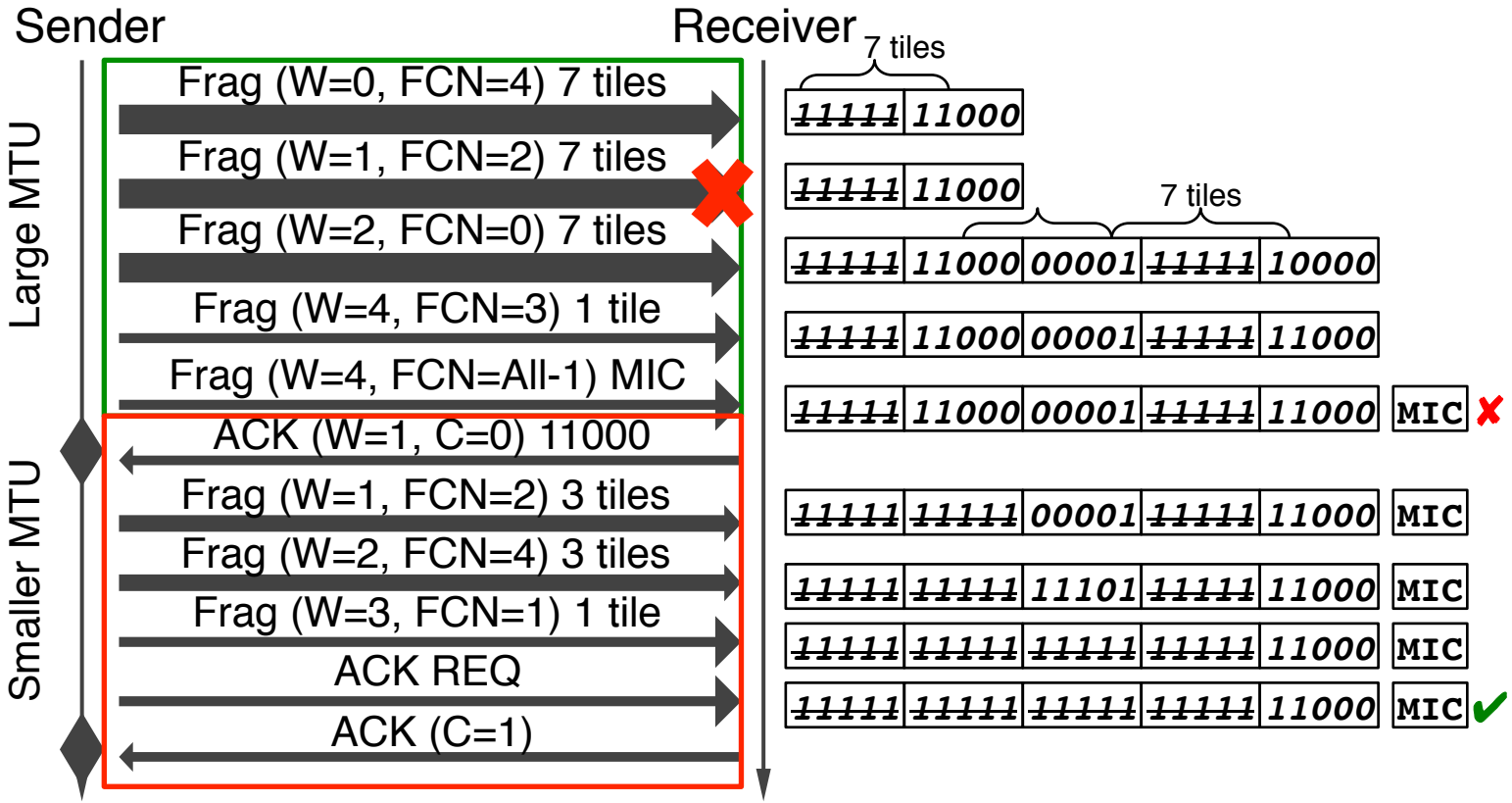
ACK-on-Error algorithm (simplified)

- Sender sends all tiles, expects ACK after sending All-1 Frag. Resends tiles reported missing by ACK, sends ACK REQ. Iterates until ACK reports MIC matches.
- Receiver assembles all tiles received. On receiving All-1 Frag or ACK REQ, returns ACK for lowest-numbered window with missing tile, expects more tiles. Iterates until MIC matches.
- Intermediate ACKs strategy defined in a Profile.

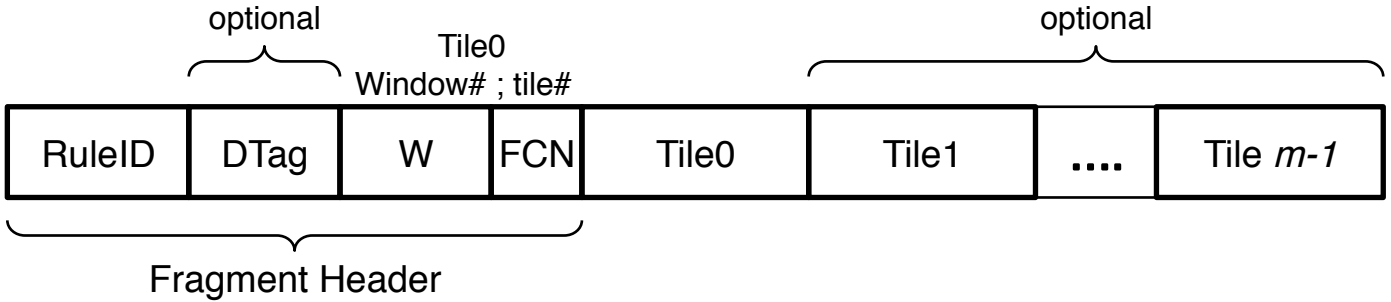
Example 1


Sender
Receiver
Frag (W=0, FCN=4) 7 tiles
Frag (W=1, FCN=2) 7 tiles
Frag (W=2, FCN=0) 7 tiles
ACK (W=1, C=0) 11000
Frag (W=1, FCN=2) 7 tiles
Frag (W=4, FCN=3) 1 tile
Frag (W=4, FCN=All-1) MIC
ACK (C=1)
7 tiles
11111 11000
11111 11000
7 tiles
11111 11000 00001 11111 10000
11111 11111 11111 11111 10000
11111 11111 11111 11111 11000
11111 11111 11111 11111 11000
MIC

Example 2


Sender
Receiver
Large MTU
Smaller MTU
Frag (W=0, FCN=4) 7 tiles
Frag (W=1, FCN=2) 7 tiles
Frag (W=2, FCN=0) 7 tiles
Frag (W=4, FCN=3) 1 tile
Frag (W=4, FCN=All-1) MIC
ACK (W=1, C=0) 11000
Frag (W=1, FCN=2) 3 tiles
Frag (W=2, FCN=4) 3 tiles
Frag (W=3, FCN=1) 1 tile
ACK REQ
ACK (C=1)
7 tiles
11000
7 tiles
11000 00001 10000
11000 00001 11000
11000 00001 11000 MIC
11111 11000 00001 11111 11000 MIC
11111 11111 11101 11111 11000 MIC
11111 11111 11111 11111 11000 MIC
11111 11111 11111 11111 11000 MIC

Tile size? Window size?



- Tile should just fit in Fragment message when MTU is lowest
  - With LoRaWAN US MTUs of 11, 53, 125 and 242 bytes → 8 bytes tiles
- Window_size adjusts the trade-off between ACK size and number of ACK messages
- The W field must be large enough to hold the full window number
  - Number of windows = ⌈max_packet_size / (window_size * tile_size)⌉
  - IPv6 packet = 1280 bytes, window_size = 7, tile_size = 6 bytes: → 31 windows → 5 bits
  - DLMS datagram = 400 bytes, window_size = 31, tile_size = 8 bytes: → 2 windows → 1 bit
  - FUOTA file image = 50KiB, window_size = 63, tile_size = 8 bytes: → 102 windows → 7 bits

Conclusions (1/3)

- Reliable fragmentation mode
  - Uses ACKs, retransmissions of missing tiles and MIC
- Supports variable MTU
- Supports Out-of-Order delivery
- ACKs used sparingly
  - Only sent for windows with missing tiles (ACK-on-Error)
  - No transmission error → no ACK (except one final ACK)
    - Valuable for uplink fragmented packet transmission
  - Exact number of ACKs dependent on error rate and distribution

Conclusions (2/3)

- Sender/receiver loosely coupled
  - simple State Machines
- Small receiver state
  - One bit per tile
  - Bitmaps for full windows can be freed
- Profiles define at what time ACKs are expected/transmitted
  - At the end of each window of tiles?
  - At the end of the packet?
  - At L2-scheduled times?

Conclusions (3/3)

- This presentation was about ACK-on-Error fragmentation mode
- Two other fragmentation modes are available
  - No-ACK
  - ACK-Always
- Integrates nicely with Header Compression!

Thank you for your attention

Open discussion