Forward Error Correction (FEC) for SCHC framework

draft-papadopoulos-schc-fec-00

G. Papadopoulos, A. Bruniaux

Motivation

- SCHC framework can be applied over LPWAN technologies:
  - fragments of a SCHC packet can be lost →
  - which may lead to the failure of the reception of the SCHC packet (**especially** in **No-ACK** mode).

- **FEC over SCHC** allows the receiver to increase the chances to recover the missing SCHC fragments *without fragment retransmissions*.

Forward Error Correction (FEC)

- **FEC Principle**: FEC is a method to control errors in packet transmission by embedding additional redundant information within transmitted fragments, thereby reducing the chances for the receiver to request retransmission of missing fragments.

- 2 FEC types:
  - **intra-frame**: recover lost bytes of a fragment from redundant bytes
  - **inter-frame**: recover lost fragments from redundant fragments

XORFEC

- XORFEC employs the Exclusive OR (XOR) operator ($\oplus$) to produce an additional fragment.
  - This extra fragment contains the redundant information.
  - This additional fragment is sent after the original SCHC fragments.
  - **If** the receiver **detects** a loss of an original fragment, it may recover it.

XORFEC (1/2)

- XORFEC employs the Exclusive OR (XOR) operator ($\oplus$) to produce an additional fragment.
  - This extra fragment contains the redundant information.
  - This additional fragment is sent after the original SCHC fragments.
  - **If** the receiver **detects** a loss of an original fragment, it may recover it.

- The **key property** of **XOR** in XORFEC for fragment recovery:
  - By applying XOR to **the result of an initial XOR** and one of its input values, the result will be the other input value!

  - If a SCHC packet is fragmented into two fragments A and B, the additional fragment C generated by the source node will be:

$$C = A \oplus B$$

  - The **receiver** can recover B fragment (*if B fragment was not received*) by applying the XOR operator to the successfully received fragments.

$$B = A \oplus C$$

XORFEC (2/2)

- Generalisation to SCHC packets that consists of more than two fragments.
  - **k** original fragments (F1, F2, ..., Fk), the extra fragment F_additional will be:

$$F\_additional = F1 \oplus F2 \oplus F3 \oplus ... \oplus Fk$$

- If the receiver receives all fragments except **Fi**:
  - it can recover the latter by applying the XOR operator:

$$Fi = (F1 \oplus ... \oplus Fi-1 \oplus Fi+1 \oplus ... \oplus Fm) \oplus F\_additional$$

XORFEC Operation Examples in LPWAN

No-ACK mode

The example from [RFC8724] of **No-ACK mode** of a SCHC Packet that requires 5 SCHC Fragments (FCN=1 bit).

Sender **Receiver**

|-----FCN=0 ----->| 1st Fragment (received)

|-----FCN=0 ----->| 2nd Fragment (received)

|-----FCN=0 ----->| 3rd Fragment (received)

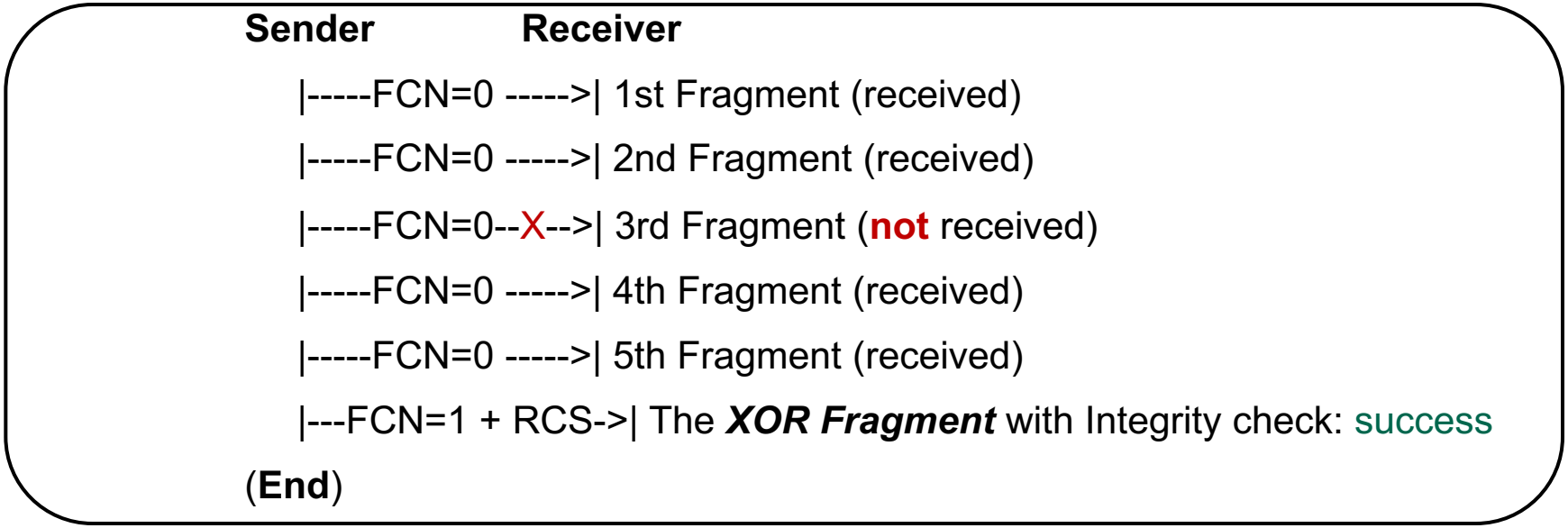
|-----FCN=0 ----->| 4th Fragment (received)

|-----FCN=1 + RCS --->| Integrity check: success

(**End**)

XORFEC over No-ACK mode

The example from [RFC8724] of **No-ACK mode** of a SCHC Packet that requires 5 SCHC Fragments (FCN=1 bit) ***is adapted*** when **XORFEC** is applied to all 5 SCHC Fragments.



XORFEC over No-ACK mode

The example from [RFC8724] of **No-ACK mode** of a SCHC Packet that requires 5 SCHC Fragments (FCN=1 bit) ***is adapted*** when **XORFEC** is applied to all 5 SCHC Fragments.

Sender **Receiver**

|-----FCN=0 ----->| 1st Fragment (received)

|-----FCN=0 ----->| 2nd Fragment (received)

|-----FCN=0--X-->| 3rd Fragment (**not** received)

|-----FCN=0 ----->| 4th Fragment (received)

|-----FCN=0 ----->| 5th Fragment (received)

|---FCN=1 + RCS->| The ***XOR Fragment*** with Integrity check: success

(**End**)

No-ACK mode → there is NO **feedback** from the **Receiver**:

- The Receiver may **successfully reassemble** the original SCHC Packet.
- The **network reliability** is increased.

XORFEC over ACK-on-Error mode

The example from [RFC8724] of **ACK-on-Error mode** of a SCHC Packet fragmented in 11 tiles ***is adapted*** when **XORFEC** is applied per Window. **One** additional (**XOR**) fragment is introduced **per Window**.

Sender Receiver
|-----W=0, FCN=6----->| 1st Tile/Fragment (received)
|-----W=0, FCN=5----->| 2nd Tile/Fragment (received)
|-----W=0, FCN=4----->| 3rd Tile/Fragment (received)
|-----W=0, FCN=3----->| 4th Tile/Fragment (received)
|-----W=0, FCN=2-X-->| 5th Tile/Fragment (**not** received)
|-----W=0, FCN=1----->| 6th Tile/Fragment (received)
|-----W=0, FCN=0----->| The **additional** (**XOR**) Fragment
(no ACK)
|-----W=1, FCN=6----->| 7th Tile/Fragment (received)
|-----W=1, FCN=5----->| 8th Tile/Fragment (received)
|-----W=1, FCN=4-X-->| 9th Tile/Fragment (**not** received)
|-----W=1, FCN=3----->| 10th Tile/Fragment (received)
|-----W=1, FCN=2----->| 11th Tile/Fragment (received)
|- W=1, FCN=7 + RCS ->| The **XOR Fragment** with Integrity check: **success**
|<-- ACK, W=1, C=1 ---| C=1
(End)

XORFEC over ACK-on-Error mode

The example from [RFC8724] of **ACK-on-Error mode** of a SCHC Packet fragmented in 11 tiles ***is adapted*** when **XORFEC** is applied per Window. **One** additional (**XOR**) fragment is introduced **per Window**.

```
Sender                    Receiver
    |-----W=0, FCN=6----->| 1st Tile/Fragment (received)
    |-----W=0, FCN=5----->| 2nd Tile/Fragment (received)
    |-----W=0, FCN=4----->| 3rd Tile/Fragment (received)
    |-----W=0, FCN=3----->| 4th Tile/Fragment (received)
    |-----W=0, FCN=2-X--->| 5th Tile/Fragment (not received)
    |-----W=0, FCN=1----->| 6th Tile/Fragment (received)
    |-----W=0, FCN=0----->| The additional (XOR) Fragment
(no ACK)
    |-----W=1, FCN=6----->| 7th Tile/Fragment (received)
    |-----W=1, FCN=5----->| 8th Tile/Fragment (received)
    |-----W=1, FCN=4-X--->| 9th Tile/Fragment (not received)
    |-----W=1, FCN=3----->| 10th Tile/Fragment (received)
    |-----W=1, FCN=2----->| 11th Tile/Fragment (received)
    |- W=1, FCN=7 + RCS ->| The XOR Fragment with Integrity check: success
    |<-- ACK, W=1, C=1 ---| C=1
 (End)
```

14 messages

- The two lost fragments (the 5th and 9th) were recovered thanks to the **additional XOR fragments**.
- The message exchanges is reduced by two transmissions (**16** in the original vs **14** with XORFEC).
  - The **ACKs** with the Bitmap of the missing fragments, and the **retransmissions** of the missing fragments.

ACK-on-Error mode

The example from [RFC8724] of **ACK-on-Error mode** of a SCHC Packet fragmented in 11 tiles.

```
Sender                    Receiver
    |-----W=0, FCN=6----->| 1st Tile/Fragment (received)
    |-----W=0, FCN=5----->| 2nd Tile/Fragment (received)
    |-----W=0, FCN=4----->| 3rd Tile/Fragment (received)
    |-----W=0, FCN=3----->| 4th Tile/Fragment (received)
    |-----W=0, FCN=2-X-->| 5th Tile/Fragment (not received)
    |-----W=0, FCN=1----->| 6th Tile/Fragment (received)
    |-----W=0, FCN=0----->| 7th Tile/Fragment (received)
    |<-- ACK, W=0, C=0 ---| Bitmap:1111011
    |-----W=0, FCN=2----->| 5th Tile/Fragment (received)
(no ACK)
    |-----W=1, FCN=6----->| 8th Tile/Fragment (received)
    |-----W=1, FCN=5----->| 9th Tile/Fragment (received)
    |-----W=1, FCN=4-X-->| 10th Tile/Fragment (not received)
    |- W=1, FCN=7 + RCS ->| 11th Tile/Fragment + Integrity check: failure
    |<-- ACK, W=1, C=0 ---| C=0, Bitmap:1100001
    |-----W=1, FCN=4----->| Integrity check: success
    |<-- ACK, W=1, C=1 ---| C=1
 (End)
```

16 messages

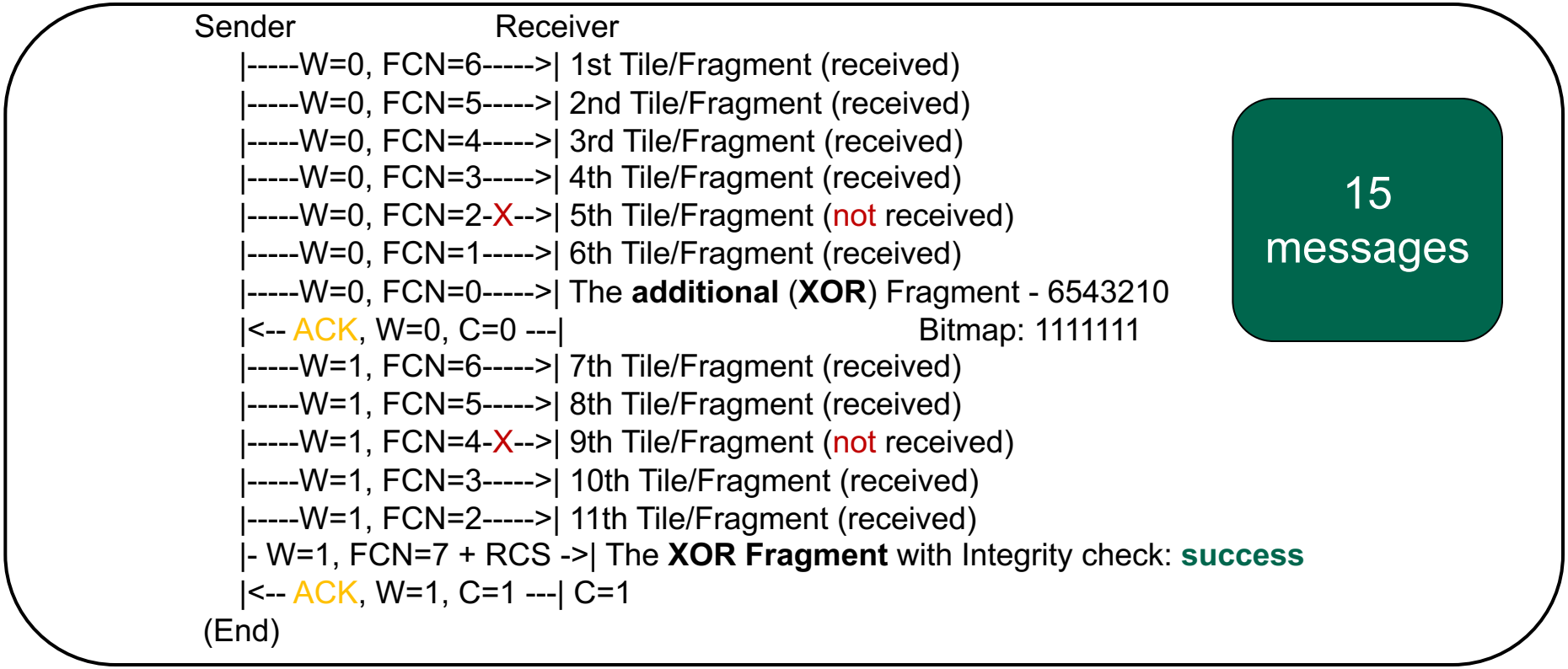
XORFEC over ACK-Always mode

The example from [RFC8724] of **ACK-Always mode** of a SCHC Packet fragmented in 11 tiles ***is adapted*** when **XORFEC** is applied per Window. **One** additional (**XOR**) fragment is introduced **per Window**.

Sender Receiver
|-----W=0, FCN=6----->| 1st Tile/Fragment (received)
|-----W=0, FCN=5----->| 2nd Tile/Fragment (received)
|-----W=0, FCN=4----->| 3rd Tile/Fragment (received)
|-----W=0, FCN=3----->| 4th Tile/Fragment (received)
|-----W=0, FCN=2-X-->| 5th Tile/Fragment (not received)
|-----W=0, FCN=1----->| 6th Tile/Fragment (received)
|-----W=0, FCN=0----->| The **additional** (**XOR**) Fragment - 6543210
|<-- ACK, W=0, C=0 ---| Bitmap: 1111111
|-----W=1, FCN=6----->| 7th Tile/Fragment (received)
|-----W=1, FCN=5----->| 8th Tile/Fragment (received)
|-----W=1, FCN=4-X-->| 9th Tile/Fragment (not received)
|-----W=1, FCN=3----->| 10th Tile/Fragment (received)
|-----W=1, FCN=2----->| 11th Tile/Fragment (received)
|- W=1, FCN=7 + RCS ->| The **XOR Fragment** with Integrity check: **success**
|<-- ACK, W=1, C=1 ---| C=1
(End)

XORFEC over ACK-Always mode

The example from [RFC8724] of **ACK-Always mode** of a SCHC Packet fragmented in 11 tiles ***is adapted*** when **XORFEC** is applied per Window. **One** additional (**XOR**) fragment is introduced **per Window**.



- The lost Fragments were recovered thanks to the **XOR fragments**.

ACK-Always mode

The example from [RFC8724] of **ACK-Always mode** of a SCHC Packet fragmented in 11 tiles ***is adapted*** when **XORFEC** is applied per Window. **One** additional (**XOR**) fragment is introduced **per Window**.

```
Sender                    Receiver
    |-----W=0, FCN=6----->| 1st Tile/Fragment (received)
    |-----W=0, FCN=5----->| 2nd Tile/Fragment (received)
    |-----W=0, FCN=4----->| 3rd Tile/Fragment (received)
    |-----W=0, FCN=3----->| 4th Tile/Fragment (received)
    |-----W=0, FCN=2-X-->| 5th Tile/Fragment (not received)
    |-----W=0, FCN=1----->| 6th Tile/Fragment (received)
    |-----W=0, FCN=0----->| 7th Tile/Fragment (received) - 6543210
    |<-- ACK, W=0, C=0 ---|                          Bitmap: 1111011
    |-----W=0, FCN=2----->| 5th Tile/Fragment (received)
    |<-- ACK, W=0, C=0 ---|                          Bitmap: 1111111
    |-----W=1, FCN=6----->| 8th Tile/Fragment (received)
    |-----W=1, FCN=5----->| 9th Tile/Fragment (received)
    |-----W=1, FCN=4-X-->| 10th Tile/Fragment (not received)
    |- W=1, FCN=7 + RCS ->| 11th Tile/Fragment + Integrity check: failure
    |<-- ACK, W=1, C=0 ---| C=0, Bitmap:1100001
    |-----W=1, FCN=4----->| Integrity check: success
    |<-- ACK, W=1, C=1 ---| C=1
(End)
```

17 messages

Key Takeaways & Road Forward

Key Takeaways:

- FEC makes sense especially in **No-ACK** (**no feedback** from the receiver).
  - Mitigating: "*the loss of one fragment leads to the entire packet being lost*".
- The main limitation of the XORFEC: the loss tolerance is one missing fragment.
- FEC increase network reliability, they also introduce additional costs (i.e., *additional fragments demands energy and bandwidth*).

Road Forward:

- Formalize how to signal that a data unit (i.e., XOR fragment) carries redundancy.