November 8th, 2024

FEC for SCHC

draft-pelov-schc-fragmentation-fec-rule-format-00

Authors: Javier Alejandro Fernandez (IMT Atlantique), Quy Nguyen (Actility), Martin Heusse (Université Grenoble Alpes), Alexander Pelov (IMT Atlantique)

Presenter: Alexander Pelov

Agenda

1. What / Why FEC (context)
2. High-level idea
    1. Use-case: Smart Metering, Complement for No-Ack and others (fewer downlink messages)
    1. FEC Fragment Rule Format
    2. FEC-before-Fragment

Italgas rolling out 7,5 mln gas meters with SCHC in Italy and Greece.

What / Why Forward Error Correction?

1. Forward Error Correction
    1. On the PDU level, not on the symbols / bits level
2. In (very) asymmetric links, sending several uplinks to save a downlink can be (very) beneficial
    1. Reducing the number of total messages (uplink+downlink) can also be (very) beneficial
3. Two modes
    1. Ships in the night
    2. FEC before frag

“Ships in the night” – a.k.a. “SCHC FEC Rule Format”

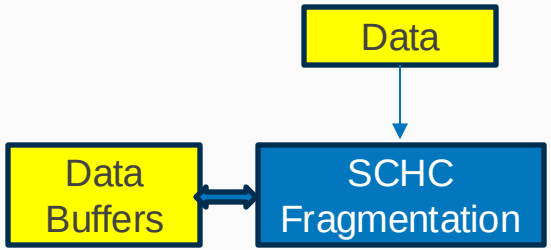
- New Rule type - “FEC Fragment Rule Format”
  - It is linked (bound) to a classical Fragmentation Rule
  - The Fragmentation Rule is oblivious to the FEC Fragments
- Send Fragments as usual, then build and send a FEC Fragment ever *n* Fragments
  - Upon reception, FEC Fragments are used to recover missing Fragments

```
RuleSet {
  10: { [compression] },
  20: { [fragmentation] },
  30: { [fec],
    FragRuleID: 20,
    FECAlgorithm: xor,
    [DTag, W, FCN],
    ...
}
```

draft-pelov-schc-fragmentation-fec-rule-format

“Ships in the night” – a.k.a. “SCHC FEC Rule Format”

- New Rule type - “FEC Fragment Rule Format”
  - It is linked (bound) to a classical Fragmentation Rule
  - The Fragmentation Rule is oblivious to the FEC Fragments
- Send Fragments as usual, then build and send a FEC Fragment ever *n* Fragments
  - Upon reception, FEC Fragments are used to recover missing Fragments



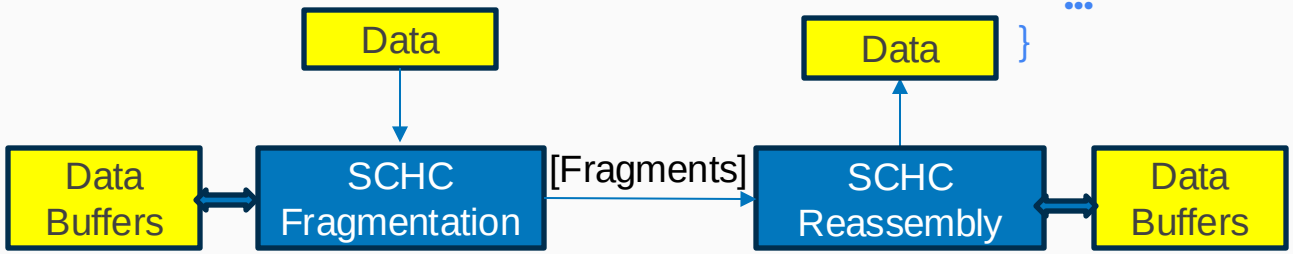

```
RuleSet {
  10: { [compression] },
  20: { [fragmentation] },
  30: { [fec],
    FragRuleID: 20,
    FECAlgorithm: xor,
    [DTag, W, FCN],
    ...
}
```

draft-pelov-schc-fragmentation-fec-rule-format

“Ships in the night” – a.k.a. “SCHC FEC Rule Format”

- New Rule type - “FEC Fragment Rule Format”
  - It is linked (bound) to a classical Fragmentation Rule
  - The Fragmentation Rule is oblivious to the FEC Fragments
- Send Fragments as usual, then build and send a FEC Fragment ever *n* Fragments
  - Upon reception, FEC Fragments are used to recover missing Fragments

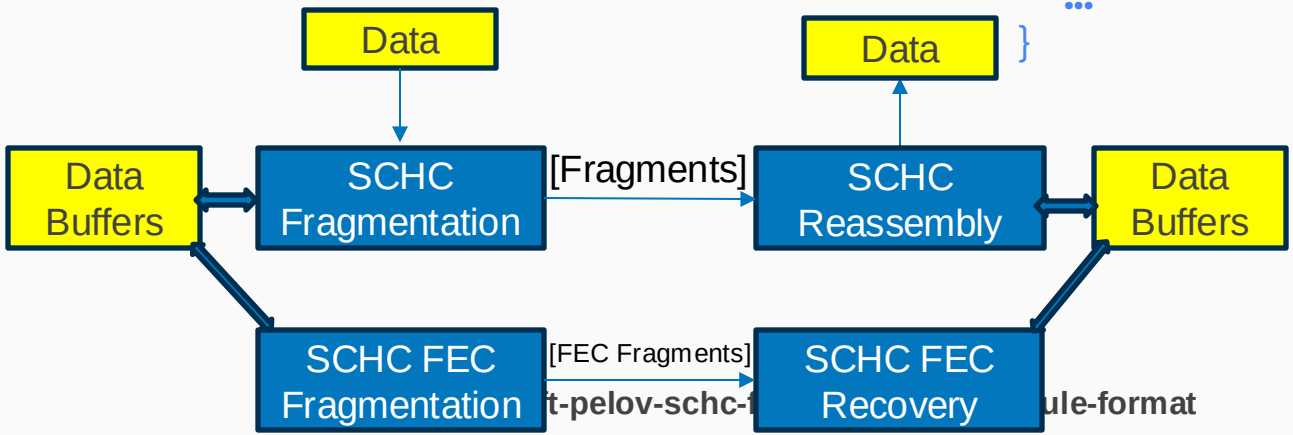
RuleSet {
10: { [compression] },
20: { [fragmentation] },
30: { [fec],
FragRuleID: 20,
FECAlgorithm: xor,
[DTag, W, FCN],
…
}



“Ships in the night” – a.k.a. “SCHC FEC Rule Format”

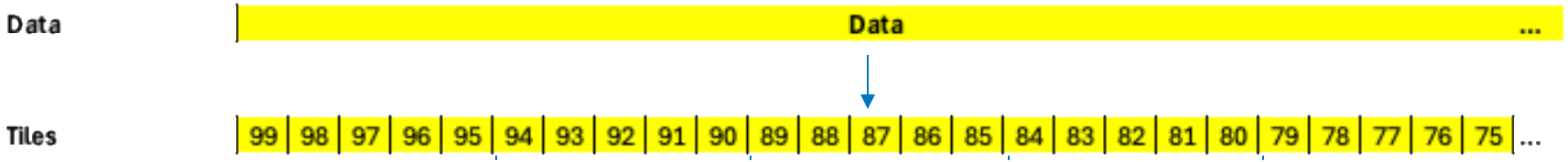
- New Rule type - “FEC Fragment Rule Format”
  - It is linked (bound) to a classical Fragmentation Rule
  - The Fragmentation Rule is oblivious to the FEC Fragments
- Send Fragments as usual, then build and send a FEC Fragment ever *n* Fragments
  - Upon reception, FEC Fragments are used to recover missing Fragments

RuleSet {
10: { [compression] },
20: { [fragmentation] },
30: { [fec],
FragRuleID: 20,
FECAlgorithm: xor,
[DTag, W, FCN],
…
}

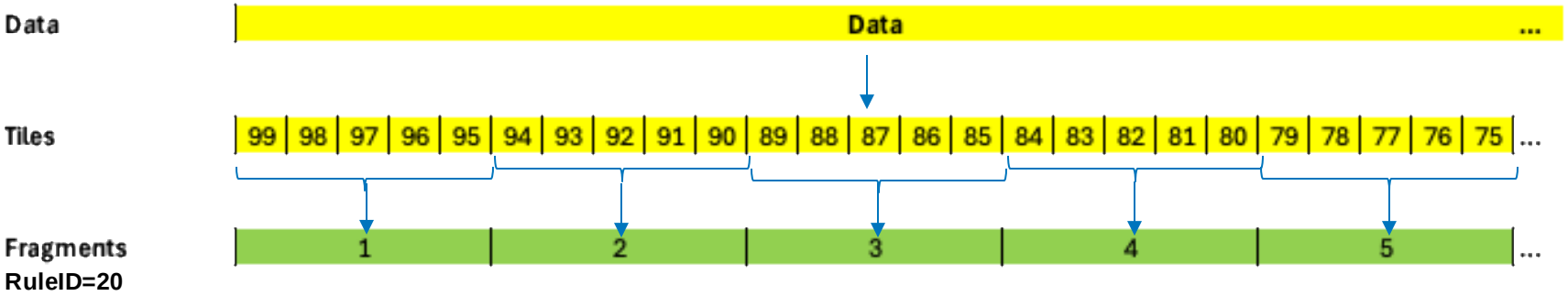


Fragmentation
MTU = 5 tiles

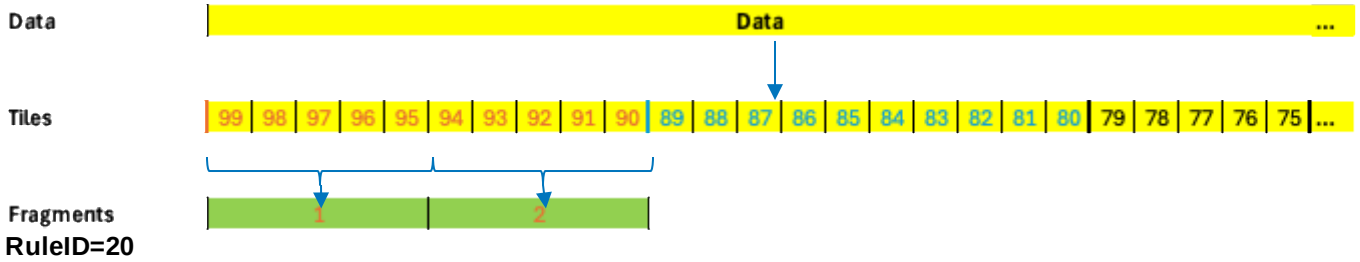
Fragmentation
MTU = 5 tiles



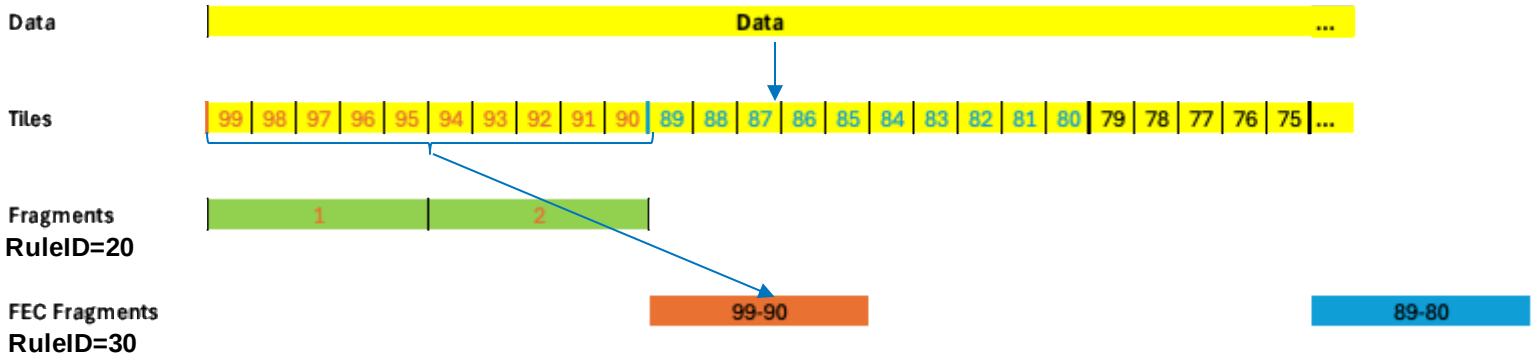
Fragmentation
MTU = 5 tiles



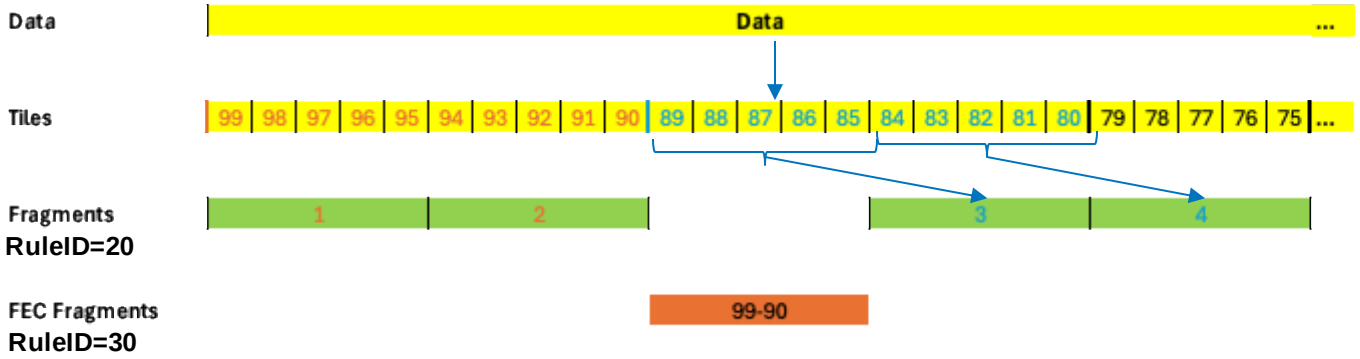
Fragmentation with FEC Fragments
MTU = 5 tiles, FEC Window = 10 tiles



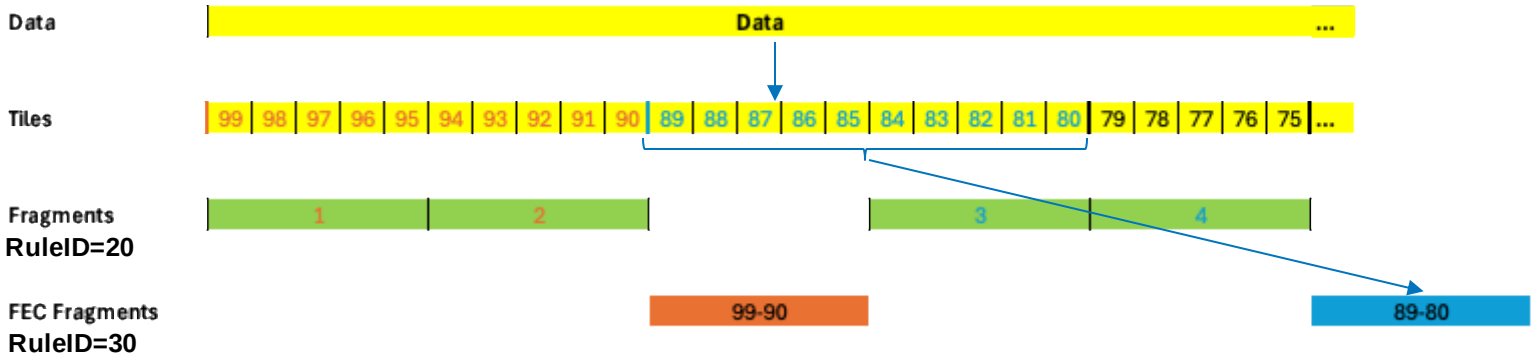
Fragmentation with FEC Fragments
MTU = 5 tiles, FEC Window = 10 tiles



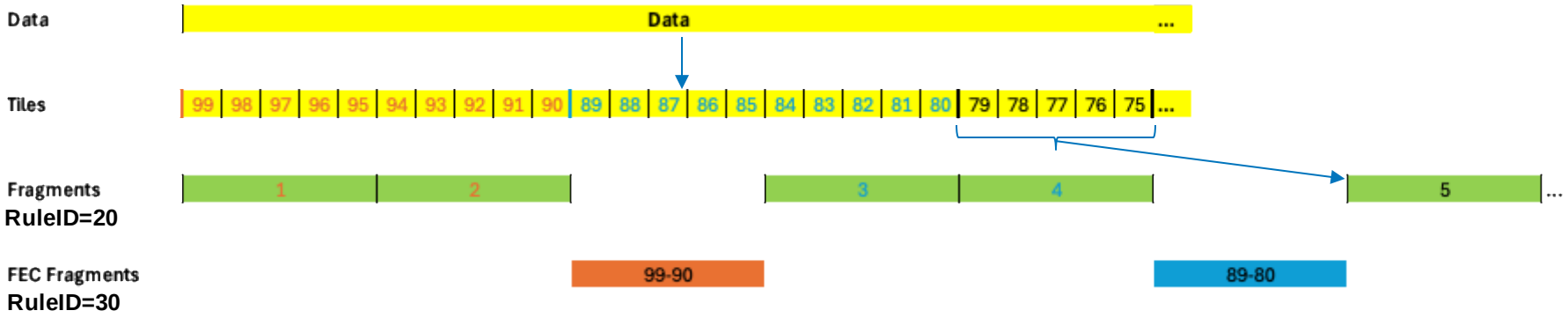
Fragmentation with FEC Fragments
MTU = 5 tiles, FEC Window = 10 tiles



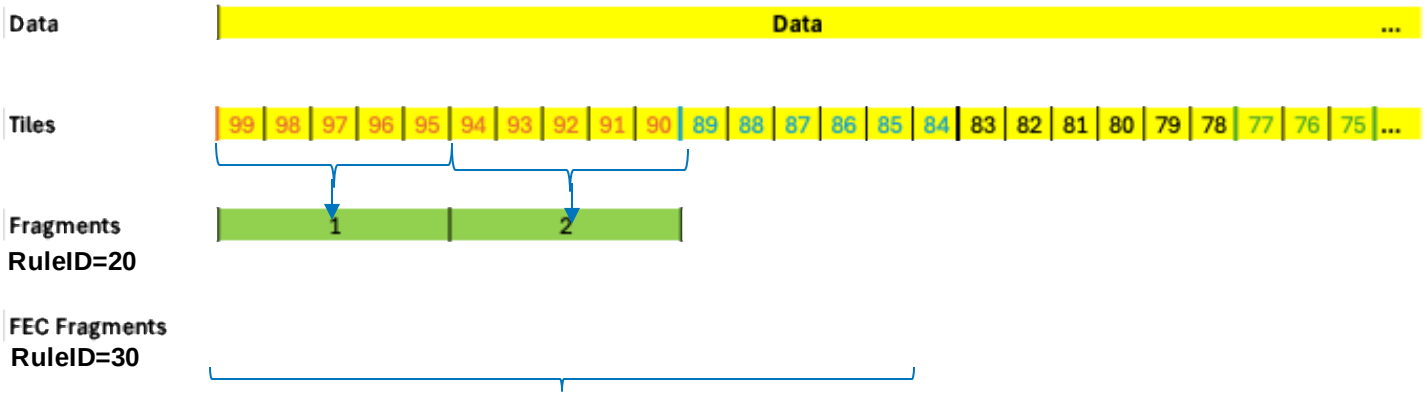
Fragmentation with FEC Fragments
MTU = 5 tiles, FEC Window = 10 tiles



Fragmentation with FEC Fragments
MTU = 5 tiles, FEC Window = 10 tiles

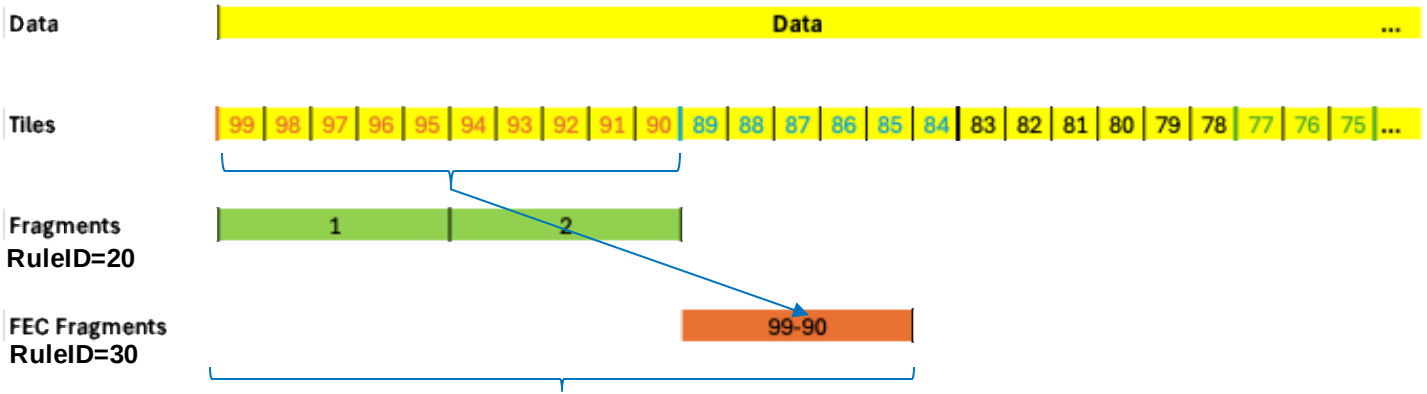


Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles



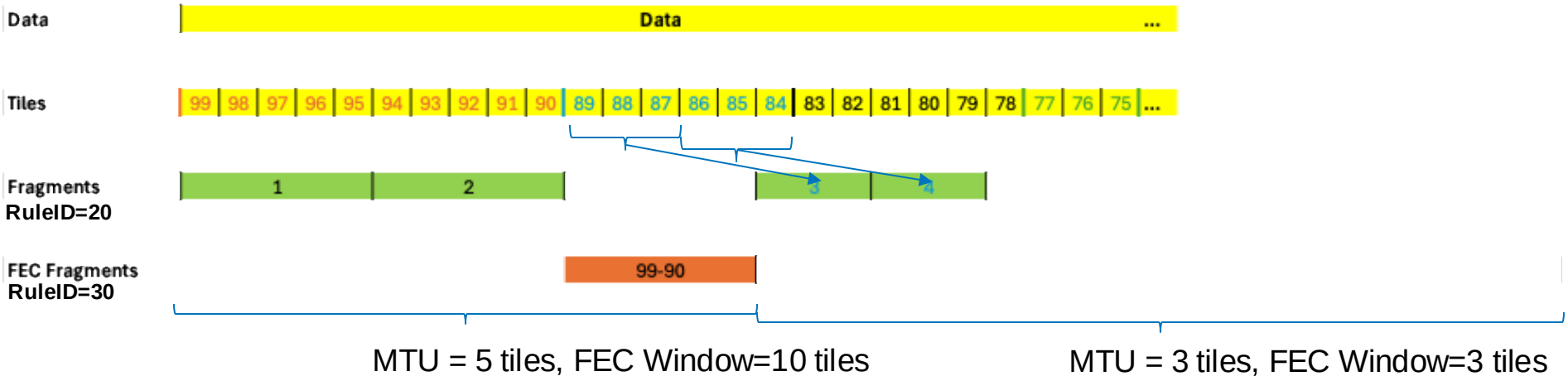
MTU = 5 tiles, FEC Window=10 tiles

Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles

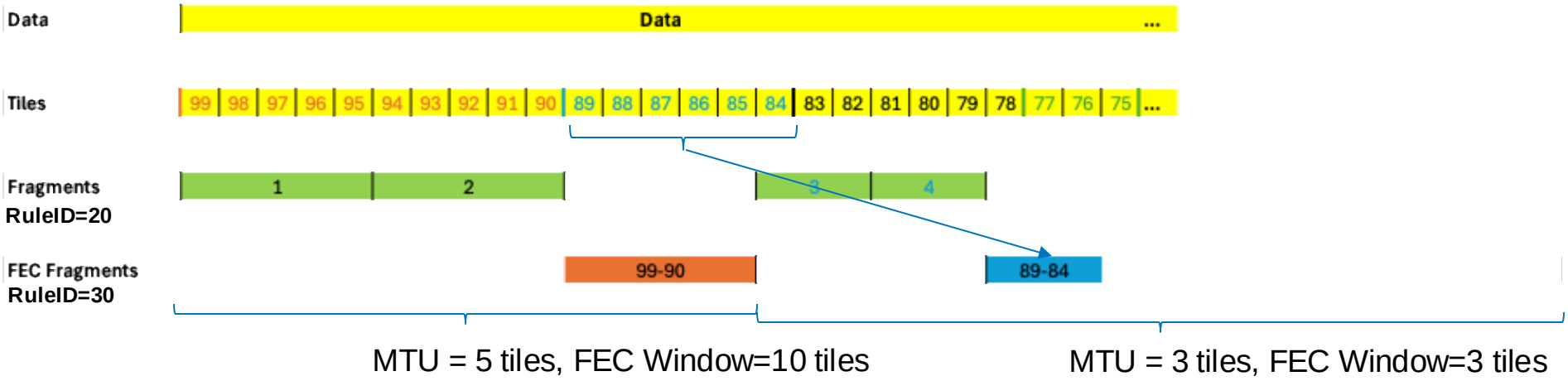


MTU = 5 tiles, FEC Window=10 tiles

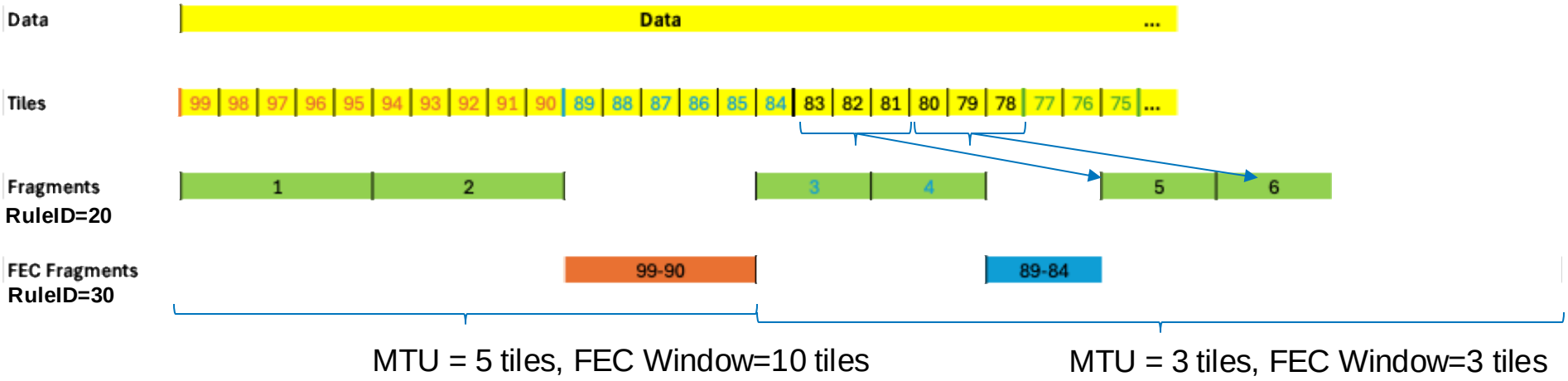
Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles



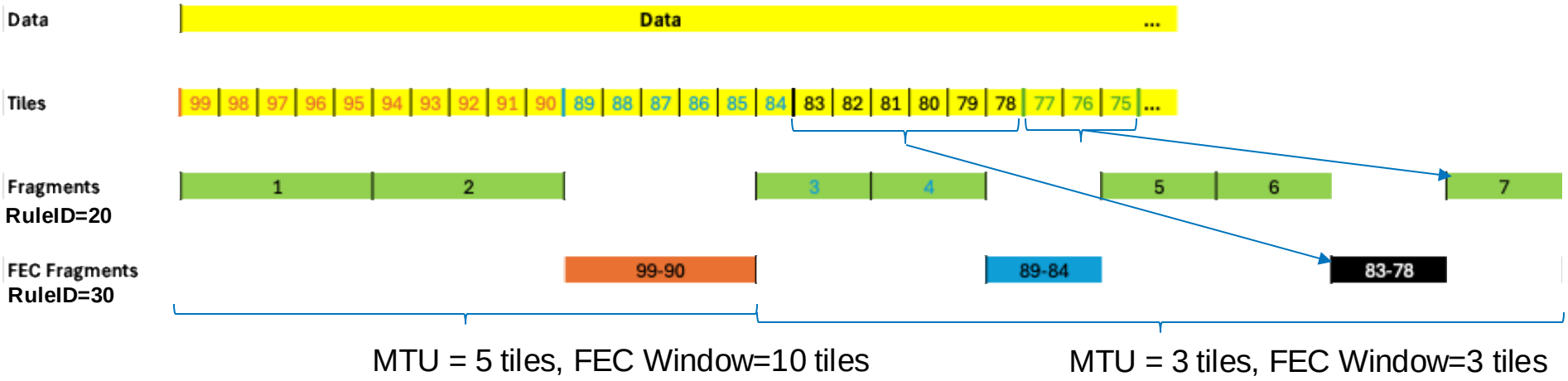
Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles



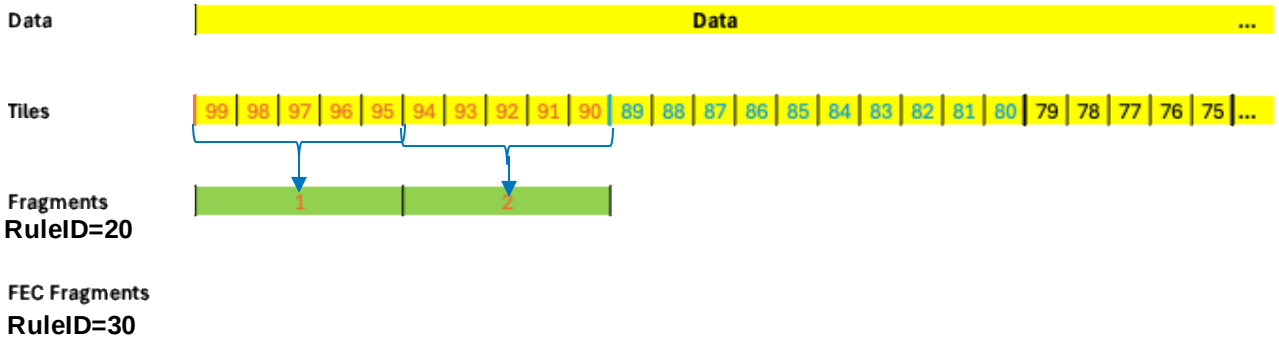
Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles



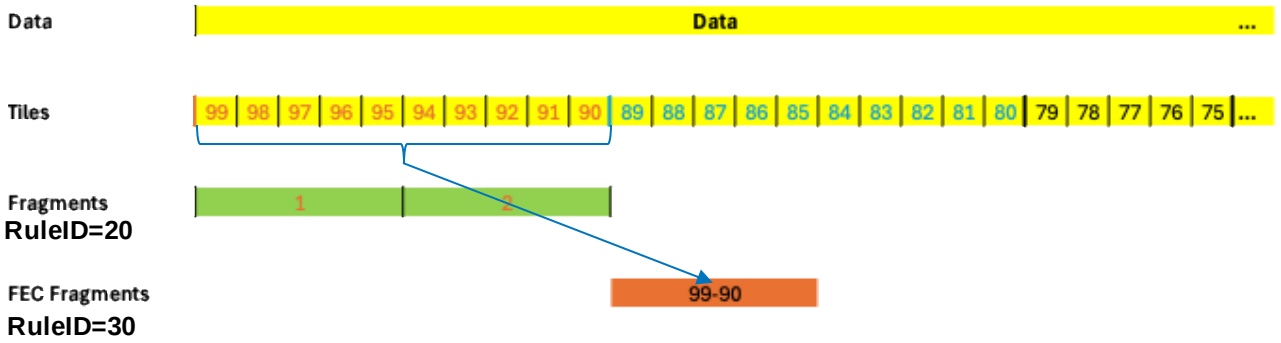
Fragmentation with FEC Fragments and MTU Change
FEC Window = 2xMTU tiles



Fragmentation with FEC Fragments
FEC Fragments can be sent at any time

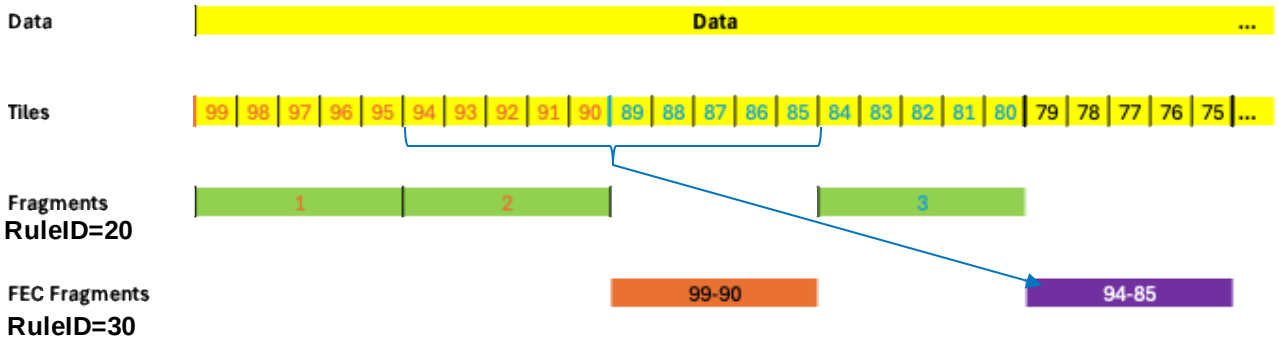


Fragmentation with FEC Fragments
FEC Fragments can be sent at any time

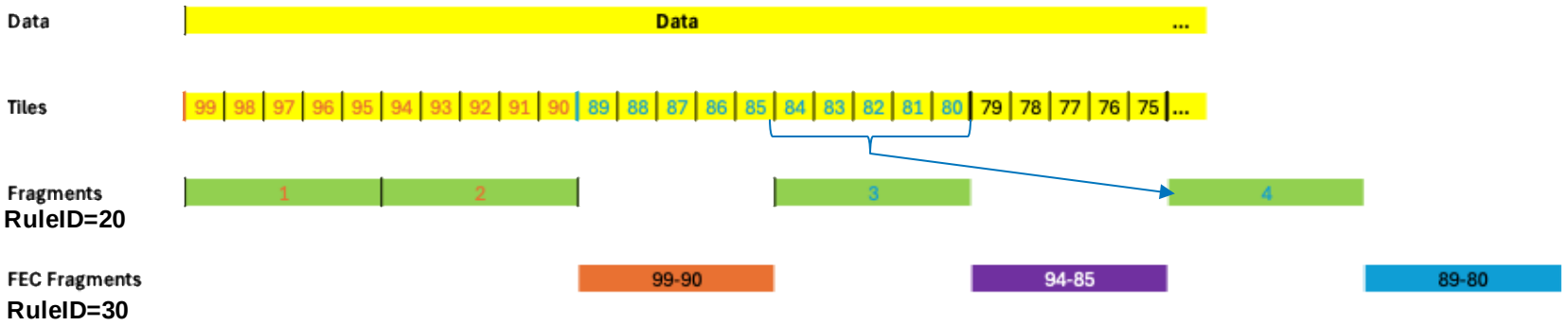


Fragmentation with FEC Fragments
FEC Fragments can be sent at any time

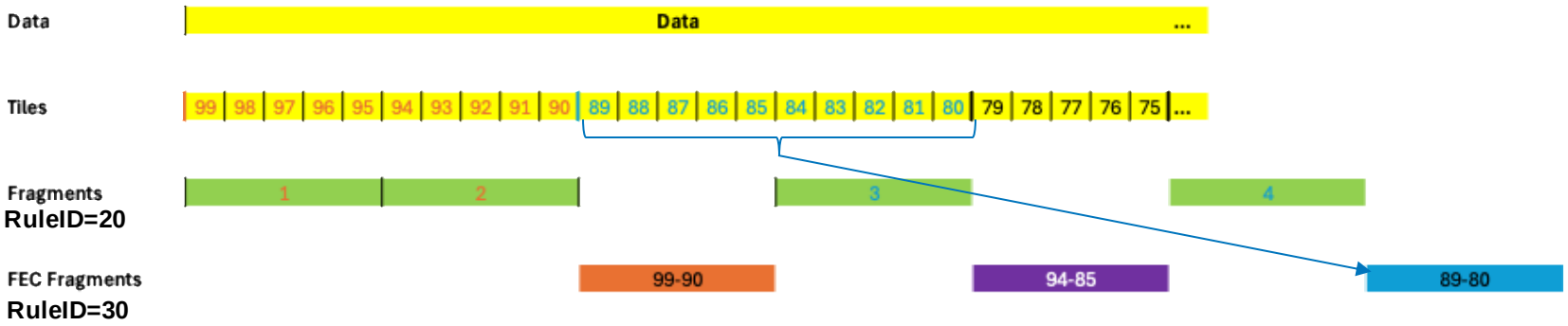
Fragmentation with FEC Fragments
FEC Fragments can be sent at any time



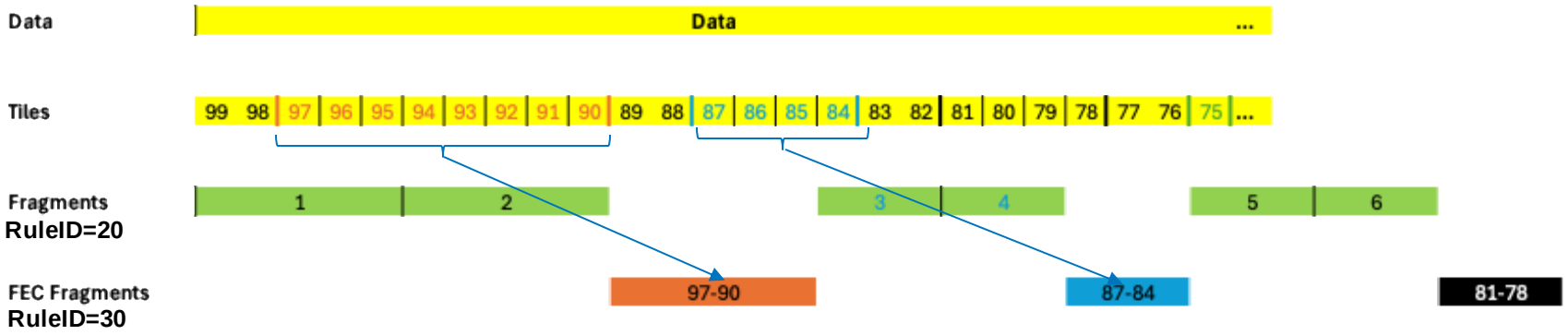
Fragmentation with FEC Fragments
FEC Fragments can be sent at any time



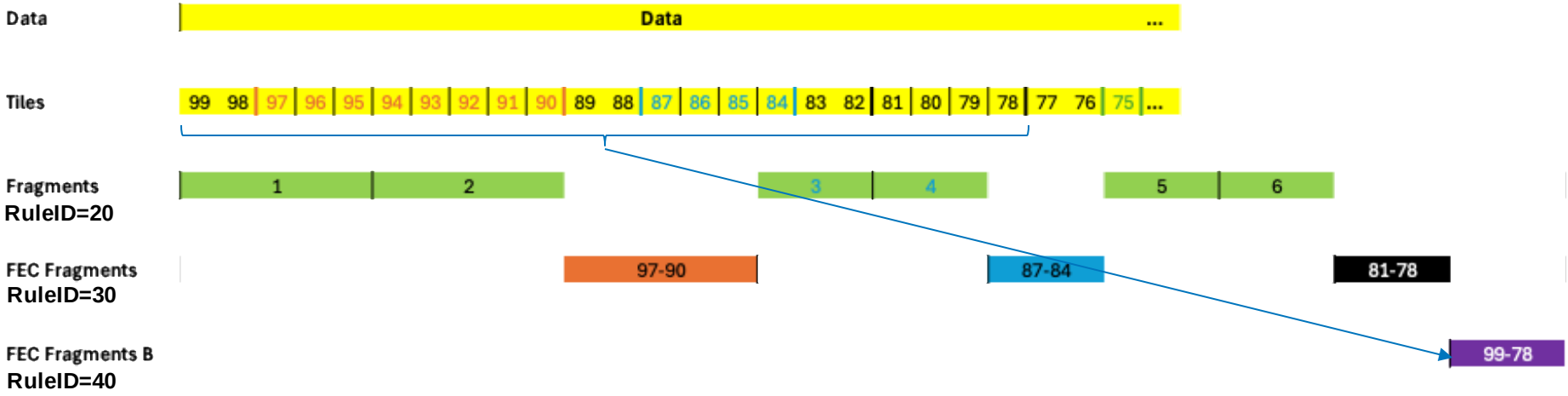
Fragmentation with FEC Fragments
FEC Fragments can be sent at any time



Fragmentation with FEC Fragments and MTU Change
FEC Window < 2xMTU tiles



Fragmentation with FEC Fragments and MTU Change
Adding Ships in the night – bound to the same base Fragmentation



“FEC before Frag”

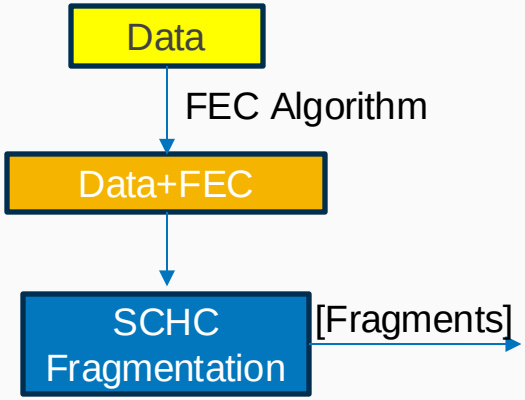
- Apply FEC algorithm to data, then use existing Fragmentation modes to send the data
  - Requires adding a parameter to existing fragmentations - % of tiles necessary to consider reception successful
  - If received fragments carry more than % of tiles, OK
- From architectural perspective – FEC can be treated as a CDA
  - But then we cannot do SCHC Compression -> FEC -> SCHC Fragmentation
  - Consider the possibility to “chain” SCHC operations
    - SCHC Compression -> SCHC Compression (FEC) -> SCHC Fragmentation
    - Others?

RuleSet {
10: { [compression] },
20: { [**fragmentation**], **fragmentation-mode**: fragmentation-mode-no-ack, **minimal-received-fragments-success**: 60%,
...
}

draft-pelov-schc-fragmentation-fec-rule-format

“FEC before Frag”

- Apply FEC algorithm to data, then use existing Fragmentation modes to send the data
  - Requires adding a parameter to existing fragmentations - % of tiles necessary to consider reception successful
  - If received fragments carry more than % of tiles, OK

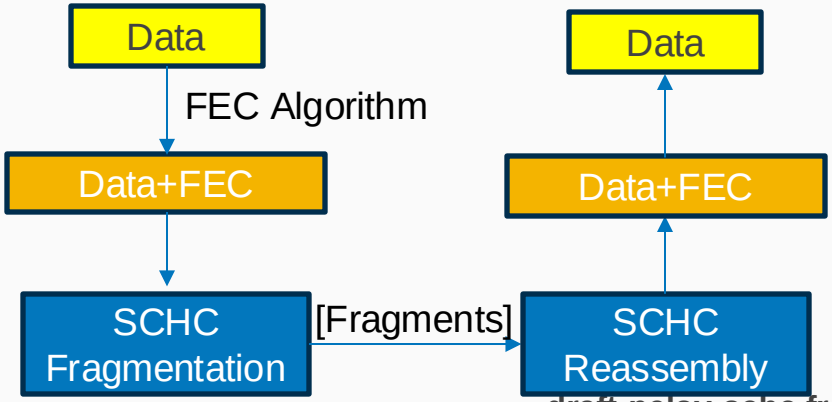


RuleSet {
10: { [compression] },
20: { [**fragmentation**], **fragmentation-mode**: fragmentation-mode-no-ack, **minimal-received-fragments-success**: 60%,
...
}

draft-pelov-schc-fragmentation-fec-rule-format

“FEC before Frag”

- Apply FEC algorithm to data, then use existing Fragmentation modes to send the data
  - Requires adding a parameter to existing fragmentations - % of tiles necessary to consider reception successful
  - If received fragments carry more than % of tiles, OK

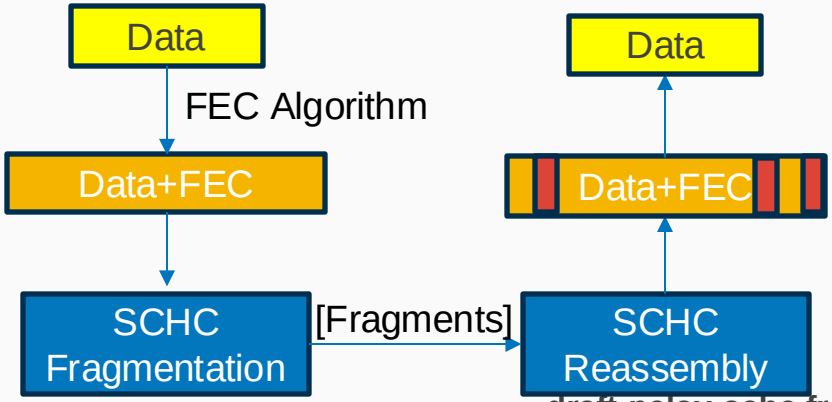


RuleSet {
10: { [compression] },
20: { [**fragmentation**], **fragmentation-mode**: fragmentation-mode-no-ack, **minimal-received-fragments-success**: 60%,
...
}

draft-pelov-schc-fragmentation-fec-rule-format

“FEC before Frag”

- Apply FEC algorithm to data, then use existing Fragmentation modes to send the data
  - Requires adding a parameter to existing fragmentations - % of tiles necessary to consider reception successful
  - If received fragments carry more than % of tiles, OK



RuleSet {
10: { [compression] },
20: { [**fragmentation**], **fragmentation-mode**: fragmentation-mode-no-ack, **minimal-received-fragments-success**: 60%,
...
}

draft-pelov-schc-fragmentation-fec-rule-format

“FEC before Frag”
Architecture considerations

- From architectural perspective – FEC can be treated as a CDA
  - But then we cannot do SCHC Compression -> FEC -> SCHC Fragmentation
- Consider the possibility to “chain” SCHC operations
  - SCHC Compression -> SCHC Compression (FEC) -> SCHC Fragmentation
- Alternatively, define new Fragmentation modes
    - E.g. Ack-On-Err-With-FEC
- Other approaches?

Ships in the night vs FEC before Frag

Ships in the night

Pros:

- ✓ Backwards compatible with RFC8724.
- ✓ Simple to implement – no change to fragmentation
- ✓ Less buffer memory on sender and receiver
- ✓ Flexible (can send FEC Fragments at any time)
- ✓ Can be used for reliability of small messages (TBD)

Cons:

- ✓ Need to specify FEC other than XOR
- ✓ Need to specify strategy when to send FEC Fragments (trivial exists, e.g. every 2 fragments)
- ✓ MTU change behavior suboptimal
- ✓ Accessing fragmentation buffer from FEC needs to be done with care

FEC before Frag

Pros:

- ✓ Simple to implement – minor changes to fragmentation state machine
- ✓ No need for FEC fragment emission strategy

Cons:

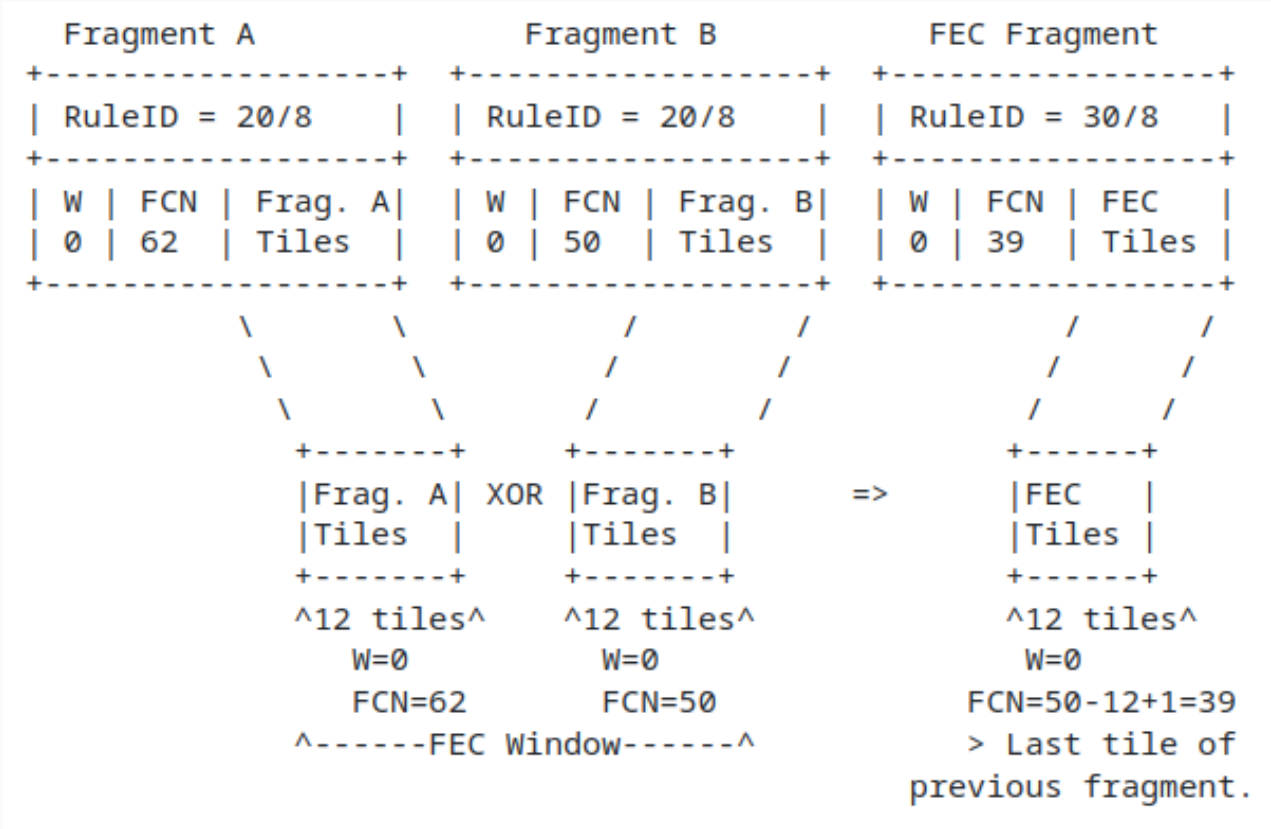
- ✓ Not backwards compatible with RFC8724.
- ✓ Need additional RAM for Data+FEC buffer
- ✓ Increased latency – necessary to receive all Data+FEC fragments before processing them

TODOs and Questions?

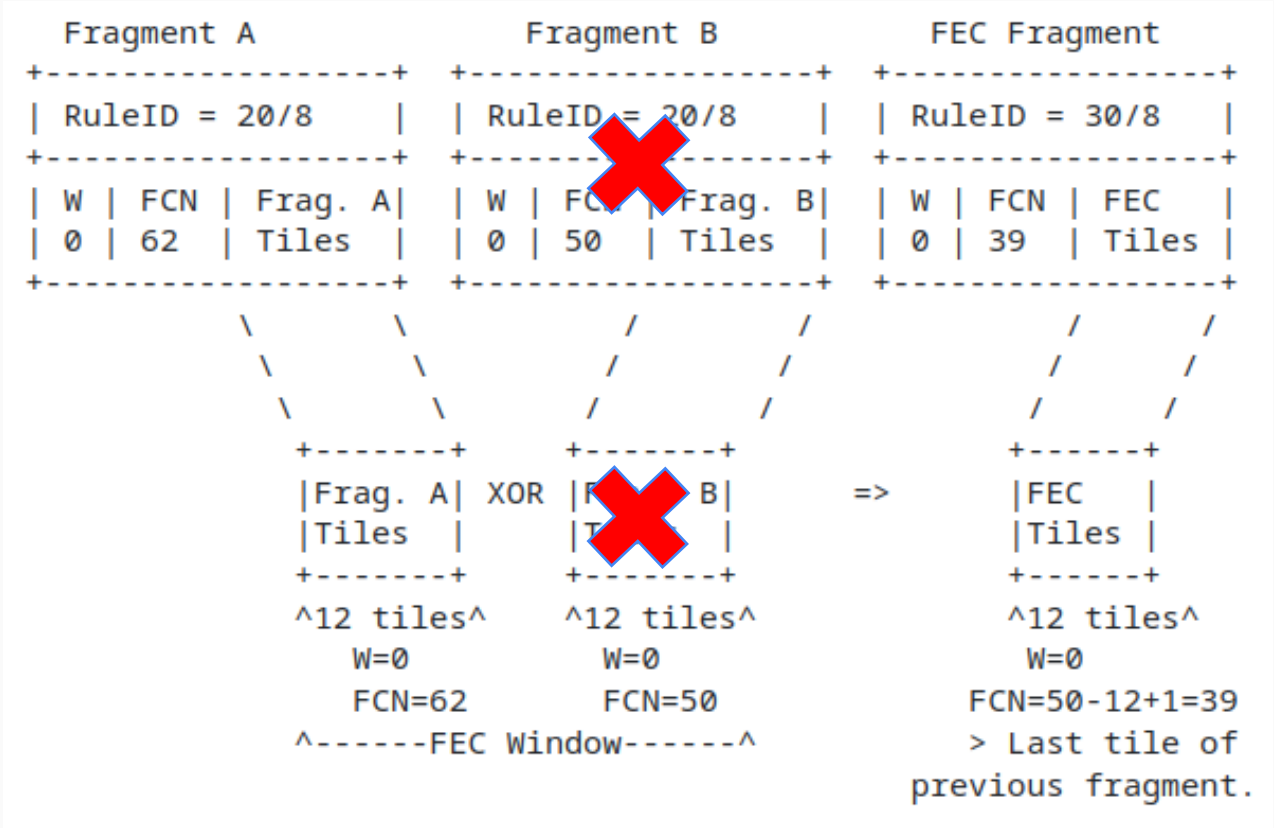
- Will change name of draft, as it has additional FEC mode
  - ”FEC for SCHC”
- Write YANG models for both modes
- Currently selecting FEC algorithm for “FEC before Frag”
- Specify use of “Ships in the night” for isolated small messages
  - Fragmentation used for Reliability
- Specification for NoAck, Ack-On-Error, Ack-Always
- Use-cases – need feedback
- Running code

- Questions ?

FEC Fragment Rule Format (draft)



FEC Fragment Rule Format



FEC Fragment Rule Format