FEC Grouping Issues in SDP

draft-begen-mmusic-fec-grouping-issues-00

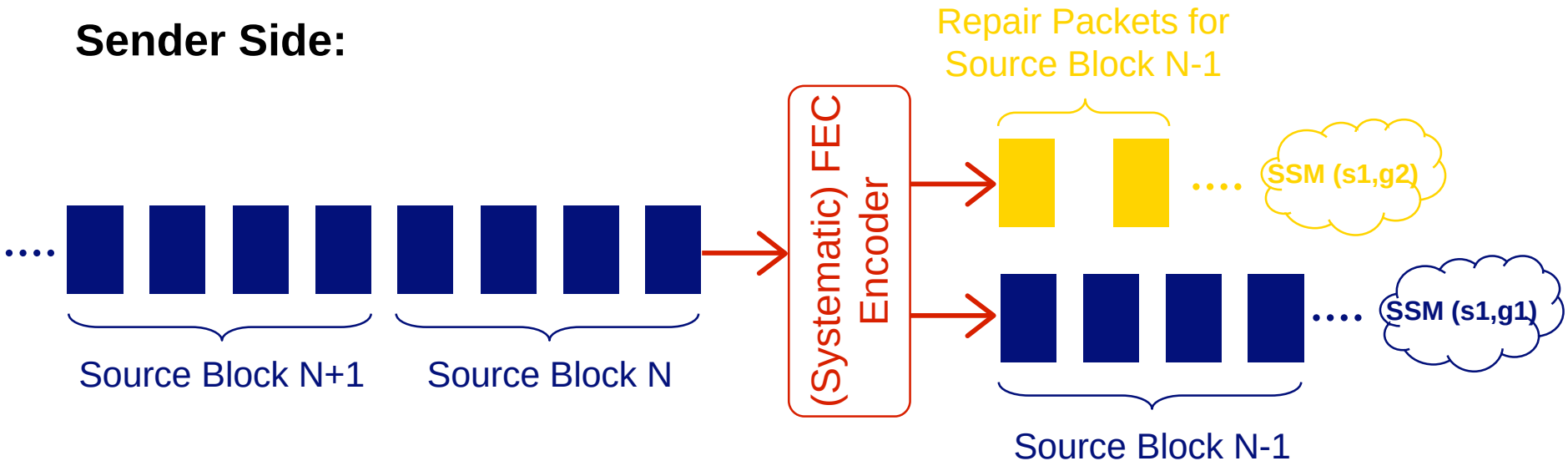
IETF 71 – March 2008

Ali C. Begen

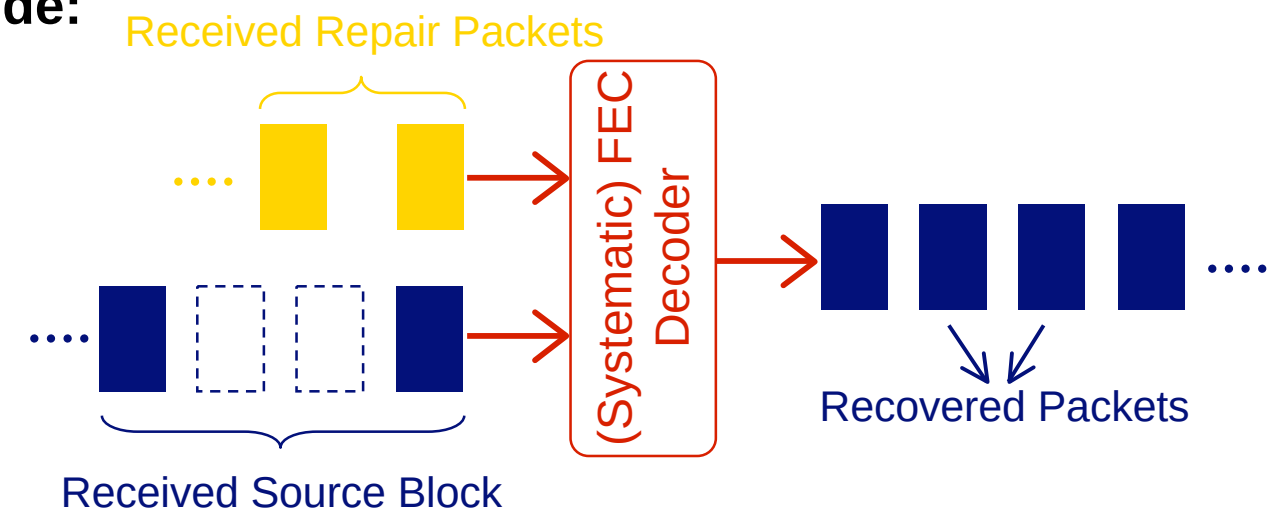
abegen@cisco.com

Forward Error Correction (FEC)

Sender Side:



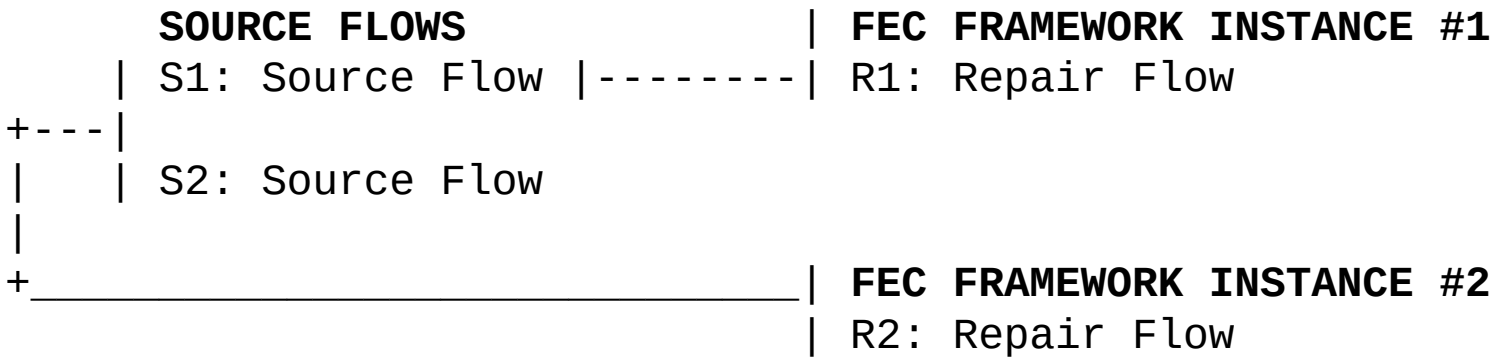
Receiver Side:



FEC Framework Flexibility

- Framework Requirements:
  - Source and repair flows are carried in different flows
  - Each FEC scheme requires a different FEC Framework instance
- We'd like to support flexible source/repair flow grouping
  - A source flow **MAY** be protected by multiple instances
  - Within an instance, multiple repair flows **MAY** exist
  - Source flows **MAY** be grouped (combined) prior to FEC protection
- If multiple repair flows are associated with a source flow, we'd like to support
  - Additive repair flows that may be decoded jointly to improve the recovery chances
  - Prioritization among the repair flows
- Can we support these features with existing tools?

Source and Repair Flow Association

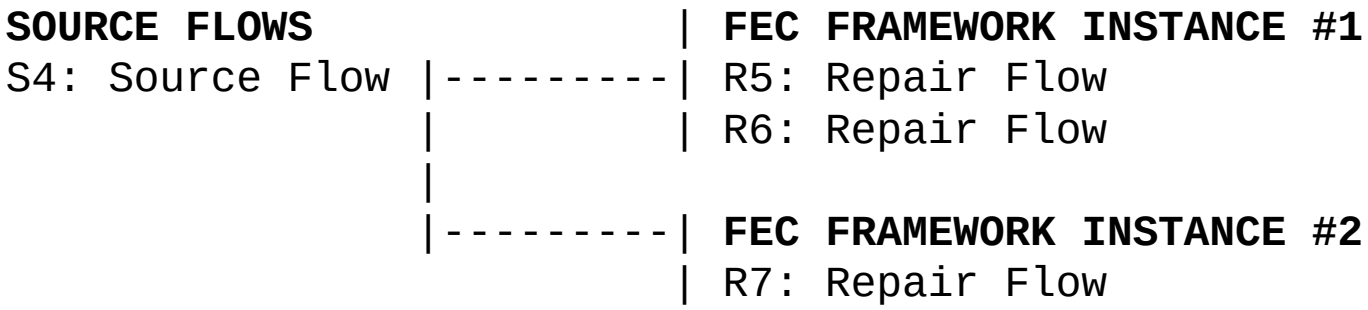


- RFC 3388: An “m” line identified by its ‘mid’ attribute **MUST NOT** appear in more than one “a=group” line using the same semantics
- RFC 4756 (based on RFC 3388) cannot handle the example above
- We could write as below, but it would not make any sense

  a=group:FEC S1 S2 R1 R2

  → No particular association

Support for Additivity / Prioritization



- Additivity
  - Multiple repair flows may be decoded jointly to improve the recovery chances
  - Additive repair flows can be generated by the same or different FEC schemes
- Prioritization
  - The sender uses prioritization to let the receivers know in which order they **MUST** receive/decode the repair flows
  - The repair flows that are assigned a priority may or may not be additive
- Currently, there is no SDP semantics for additivity/prioritization

Solution Approaches

- **New Grouping Attribute (One “a=gengroup” line per instance)**

  `a=gengroup:FEC S1 R1`

  `a=gengroup:FEC S1 S2 R2  → Associations are completely defined`

  – Additivity

  `a=gengroup:FEC S4 R5 R6 → Repair flows R5 and R6 are additive`

  `a=gengroup:FEC S4 R7    → Repair flow R7 is not additive`

  – Prioritization: Priority may be indicated by the order of the ‘mid’ values of the repair flows (e.g., p(R5) > p(R6) > p(R7) in the example above)

- **New Grouping Semantics**

  `a=group:genFEC S1 R1`

  `a=group:genFEC S1 S2 R2  → Associations are completely defined`

  – Additivity and prioritization are handled in the same way as above

- Both approaches are backward compatible

  – New grouping attribute is safer, though

Comments/Feedback

- Anybody else having issues with RFC 3388?
- Should we come up with a more general solution?
  - Obsolete RFC 3388?
  - Define something new (e.g., “a=gengroup”) that is still backward compatible with RFC 3388?
- Or, should we leave RFC 3388 as it is and propose an FEC-specific solution?