

# Improving Frame FEC Efficiency Using Frame Bursts

Ariel Maislos – Passave

Lior Khermosh – Passave

# Methodology

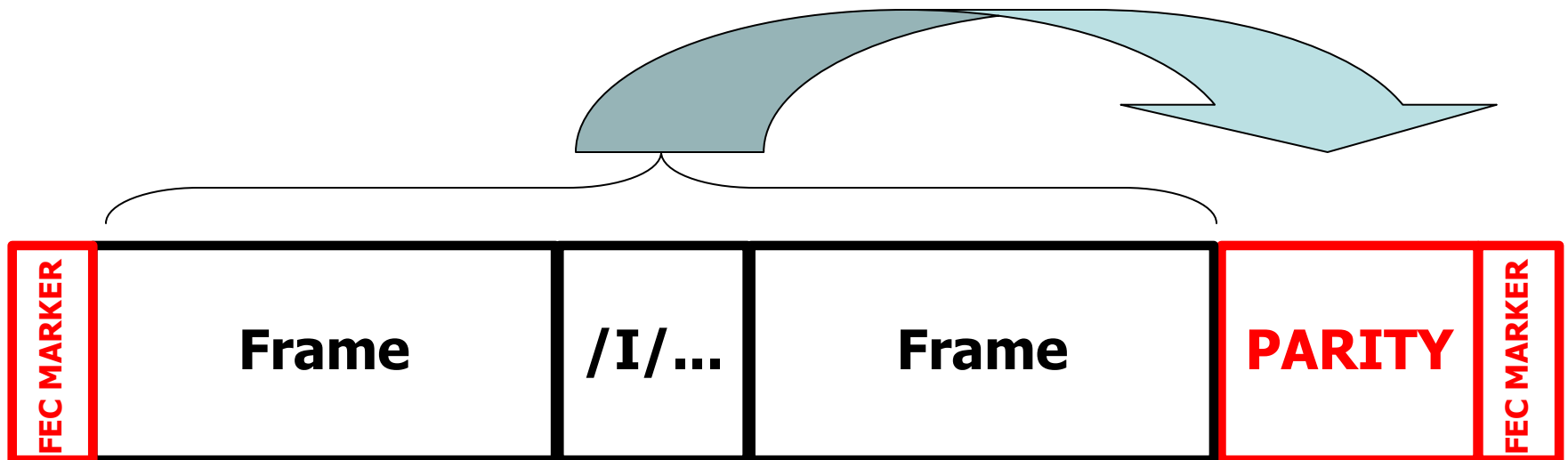
---

- FEC overhead was function of frame size
  - FEC overhead vulnerable to frame length distribution
  - Efficiency when transmitting small frames required improvement
- Solution: joint coding of a burst of frames
  - FEC layer exhibits efficiency as if transmitting long frames
  - Framing overhead reduced
  - Frame-FEC shows efficiency higher than Stream-FEC
- Frames forwarded to FEC sublayer from MAC are coded jointly
- Two methods proposed to signal FEC sublayer of frame burst
- Achieved efficiency is higher than Stream FEC proposal

# What is a burst

---

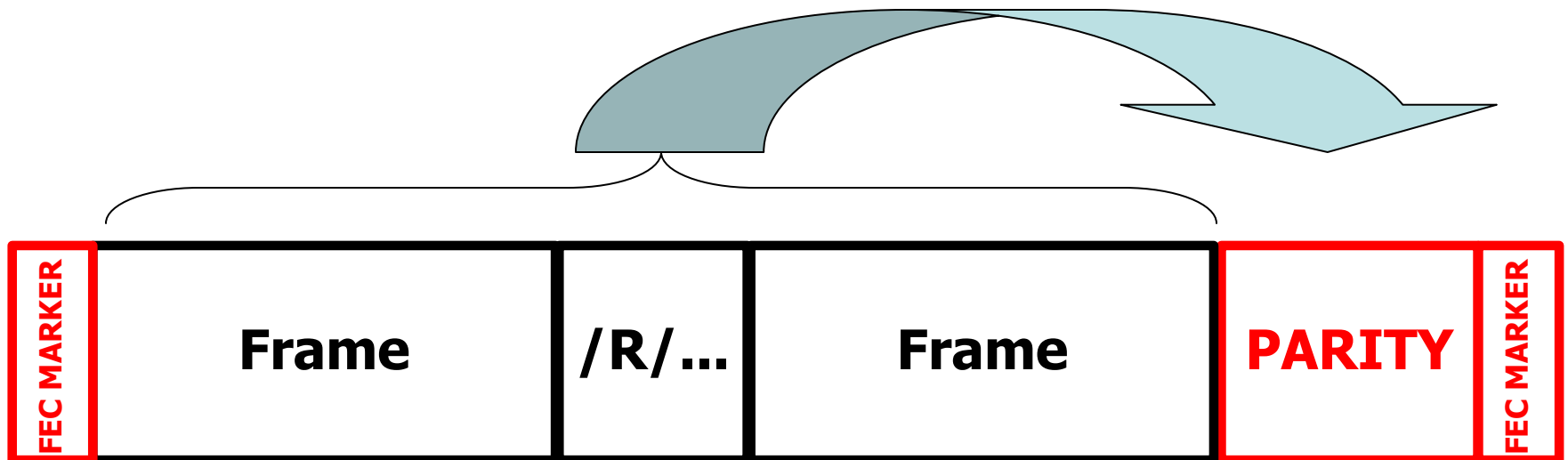
- FEC markers signal start of burst and end of burst
- Parity corrects multiple frames inside burst
- 1:1 relation between parity size and marker distance



# Carrier Extend Method

---

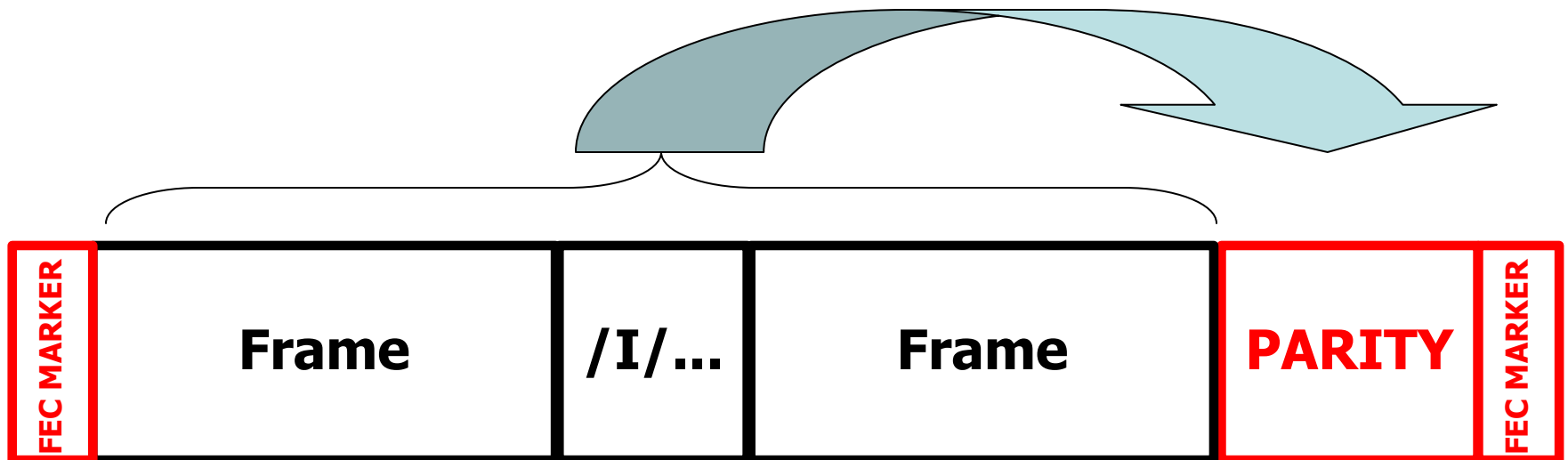
- MAC signals burst of frames using Carrier Extend interface
- FEC sees /R/ symbols between frames in same burst



# Idle Method

---

- MAC signals burst of frames by not stretching IPG
- FEC sees /I/ symbols between frames in same burst without stretched IPG – requires 12 byte memory



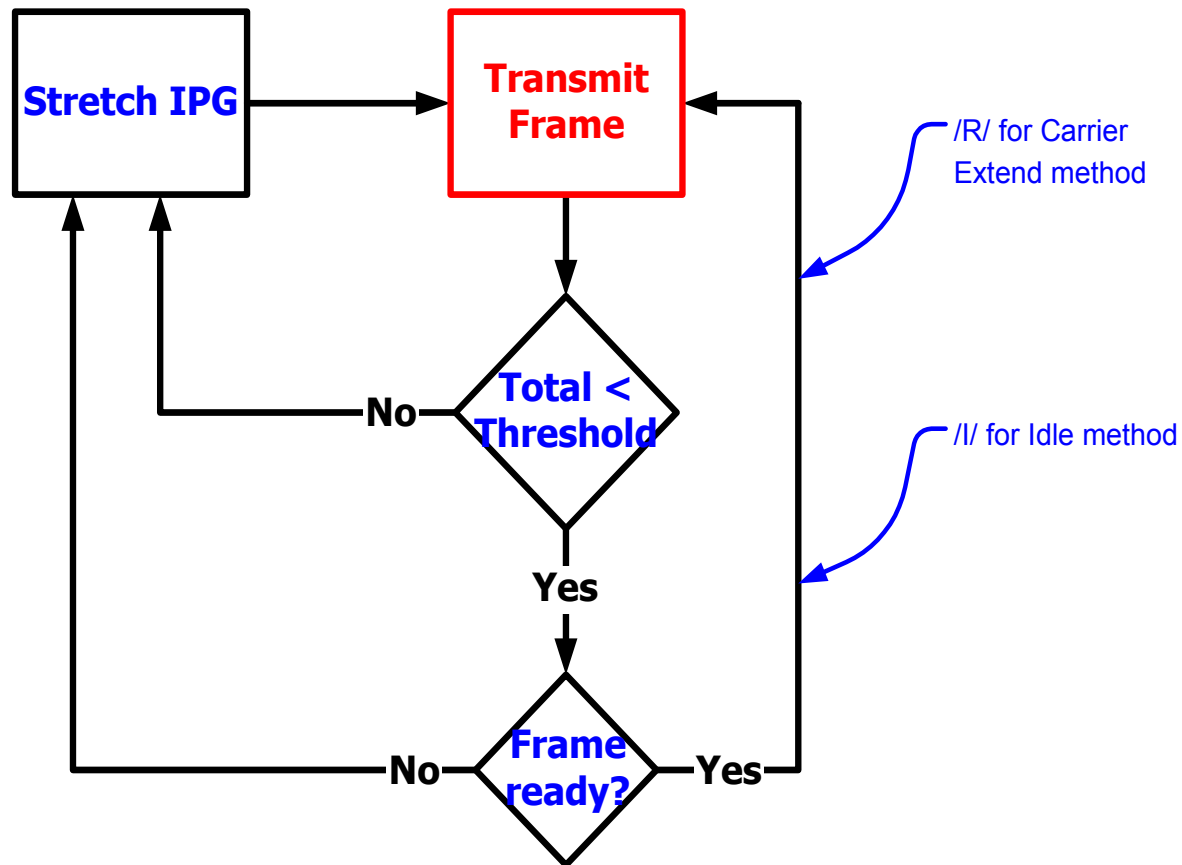
# FEC TX Implementation

---

- Receives a burst of packets
- Encode packets, gathering 1518 to 3035 bytes
- Add start symbols at start of burst
- Add parity bytes after the last frame in the sequence
- Add stop symbols following parity

# Burst Generation MAC Logic

---



# FEC RX Implementation

---

- Detect start of burst
- Receive burst of frames gather 1518 to 3035 bytes
- Detect end of burst
- Decode FEC
- Correct errors and replace parity with /I/
- Receiver has constant delay of 3035 bytes



# Frame FEC Efficiency When Bursting

---

- Frame FEC overhead: 7.3%
- Assuming
  - Carrier Extension for frame concatenation worse case scenario
  - Frame Size 1518
  - Ethernet Framing 20
  - FEC Data 123
- Stream FEC overhead: 8.2%
- Assuming:
  - Frame size 191
  - FEC Data 17

# Conclusions

---

- FEC proposal using Ethernet framing proposed
- Reduction in framing overhead achieved without increase in complexity
- Backward compatibility with Ethernet maintained
- FEC overhead is 7.3% - lower than any proposal with same performance
  - F-FEC code of (255,239) inherently more efficient than (208,191) used by S-FEC