Improving Frame FEC Efficiency Using Frame Bursts

Ariel Maislos, Passave
Lior Khermosh, Passave

Motivation: Efficiency Improvement

F-FEC

Pkt size	64	512	1518
Without FEC	86	534	1540
Efficiency	74.4%	95.9%	98.6%
With FEC	113	593	1663
Efficiency	56.6%	86.3%	91.3%

% distibution	56.3%	28.1%	15.6%
AVG without Fec	95.0%		
AVG with Fec	85.1%		
Overhead	9.9%		

% distibution	80.0%	0.0%	20.0%
AVG without Fec	94.2%		
AVG with Fec	83.9%		
Overhead	10.3%		

Mostly indifferent to traffic distribution

Improved F-FEC	Worse-Case Pkt size 1518
AVG without Fec	98.6%
AVG with Fec	91.3%
Overhead	7.3%

Overhead reduced

Methodology

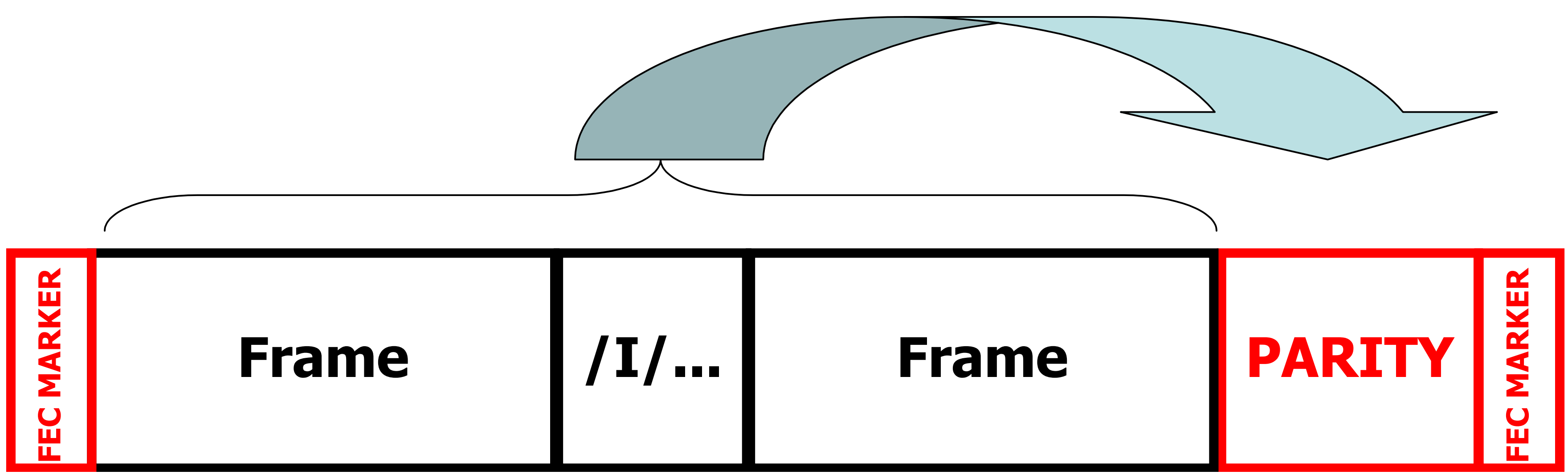
- FEC overhead was function of frame size
  - FEC overhead vulnerable to frame length distribution
  - Efficiency when transmitting small frames required improvement
- Solution: joint coding of a burst of frames
  - FEC layer exhibits efficiency as if transmitting long frames
  - Framing overhead reduced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

Stretching Concept

- FEC parity follows the protected frame, space for the parity’s transmission is made by stretching the IPG
- Instead of stretching IPG following every frame, stretching is delayed until ‘enough’ frames have passed
- As the wait for the next frame is not bound, a burst is ‘closed’ once there is no frame ready to transmit, i.e. the IPG is greater than 12
- This has the effect of not increasing overhead when traffic is sparse
- This also allows quick transmission of frames when required (for example OAM dying gasp)

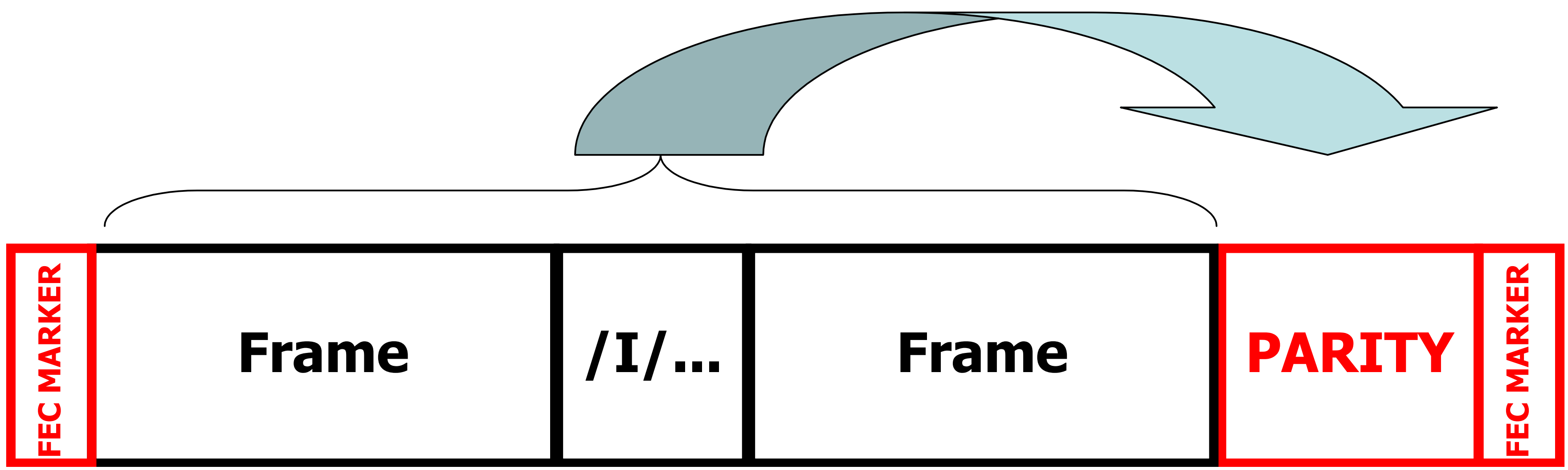
What is a burst

- 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



Burst Content

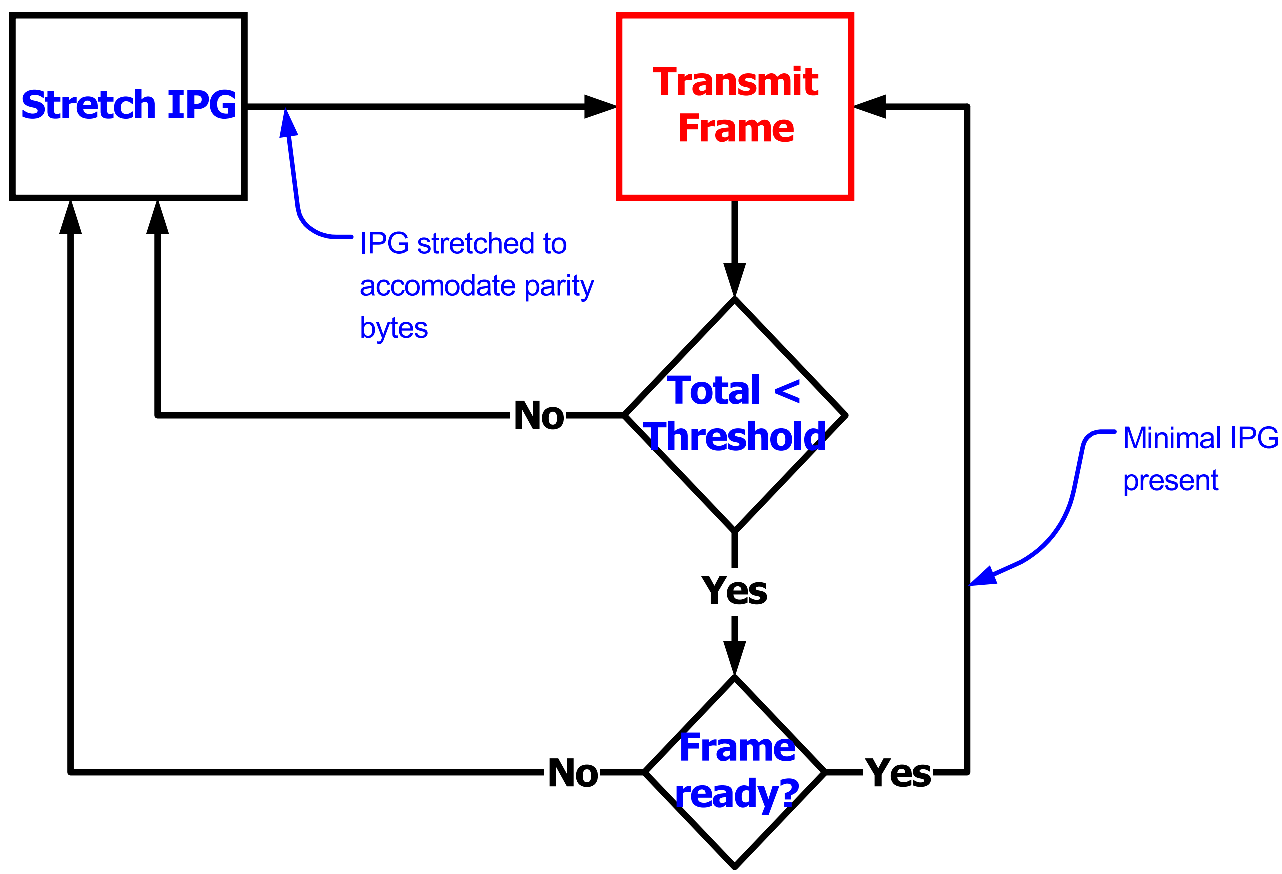
- MAC signals burst of frames by not stretching IPG
- FEC sees /I/ symbols between frames in same burst without stretched IPG – requires 12 byte memory



Summary of Technical Concepts

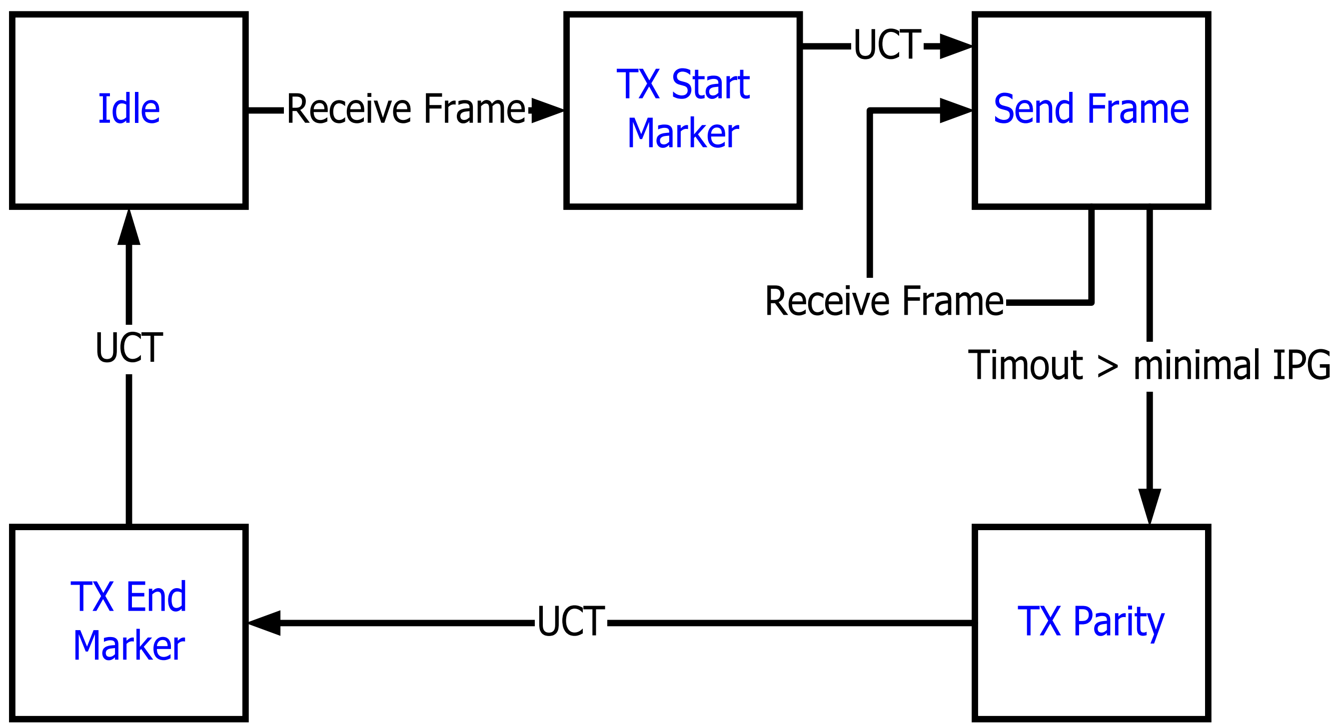
- IPG stretching behavior is modified:
  - Stretching to occur at tend of burst only
- FEC delimiters added only at start and end of burst
- Parity bytes protect entire burst as one unit
- Decoder decodes entire burst as one unit
- Correlator detects minimal sized IPG to reconstruct internal frame boundaries

Burst Generation MAC Logic



- Threshold is set to 1518

Burst Generation FEC Logic



- Shaping of burst performed by MAC

Burst Shape

- Decision threshold is set to MAX_FRAME_SIZE (1518)
- As frames are concatenated into a burst end of burst problem exists
  - What happens when the last frame to arrive when burst is at 1517 is of MAX_FRAME_SIZE?
- Gathering of frames causes burst to be sized between 1518 to 3035 bytes

MAC → FEC Synchronization

- MAC operates in open loop – making room for parity
- FEC recognizes presence of frame as start of burst
- FEC recognizes lack of frame as end of burst
  - Lack is defined as no frame arriving following minimal IPG
- FEC truncates parity generation, and inserts at first stretched IPG opportunity
- Acquisition – trivial
  - First frame received by FEC is considered start of burst
  - Worst case is wasted IPG space
- Compatibility:
  - MAC following non-burst rules of stretching + FEC with bursting results in non-burst transmitter

FEC → MAC Synchronization

- FEC replaces all parity bytes with /I/
- MAC is self-synchronous

FEC TX Implementation

- Add start symbols at start of burst
  - Burst start defined as first frame to arrive at not in burst state
  - Last frame in burst detected as IPG greater than minimal or when threshold is exceeded
- Add parity bytes after the last frame in the sequence
- Add stop symbols following parity

FEC RX Implementation

- Receiver detects same /FEC_S/ and /FEC_T/ symbols
  - Gathering 1518 to 3035 bytes until end of burst
- Correction of errors identical to FEC with no bursting
  - Just larger buffer
- Last step added is separation of frames composing burst using correlation of /I/ sequences

Frame Extraction

- Presented burst contains multiple frames
- Delimiter between frames in burst:
  - 6 X /I/
- Boundaries Separating between the frames using a correlation on the /T/R/I/I/I/I/I/S/ sequence (a few combinations exist)
- The burst is detected using the same burst delimiters
- Two options discussed:
  - Detection of boundaries before FEC correction (on 8b vectors)
  - Detection of boundaries after FEC correction (on 10b vectors)

Frame Extraction – Before Correction

- Correlating sequence has 4 options:
  - /T/R/I1/I2/I2/I2/I2/S/
  - /T/R/R/I1/I2/I2/I2/I2/S/
  - /T/R/I2/I2/I2/I2/I2/S/
  - /T/R/R/I2/I2/I2/I2/I2/S/
- Correlation sequence is long enough to be detected with very high probability
  - Scan the input symbol stream for a match sequence and say you have sync when the match has less than d/2 symbols errors

Frame Extraction – Before Correction

- P_{un_lock}=(P_{symbol_error})^(d/2)
- Correlation common to all /I/ options (need to check only in bursts)
- Error probability in 11 bytes: can have error in 6bytes – Pe=(1e-3)^6
- Probability for mis-detect:
- P_{false_lock}=(P_{bit_error})^(d/2) = (1e-4)^6 (1 in 3e7 years)
  - (only during bursts, assuming data always 1 bit different from Idles)
- A Shift is differed in 8bits per symbol – Pe<(1e-4)^8
- Error protection is sufficient

Frame Extraction – Following Correction

- sequence between frames:
  - /T/R/I1/I2/I2/I2/I2/S/
  - /T/R/R/I1/I2/I2/I2/I2/S/
  - /T/R/I2/I2/I2/I2/I2/S/
  - /T/R/R/I2/I2/I2/I2/I2/S/
- Protect with FEC as 8 bit words and add feedback to 8B/10B
- Reduces the error probability
  - Change In last page equations:
  - $P_{bit\_error}=(1e-12)*256$
  - $P_{symbol\_error}=(1e-11)*256$

Analysis of Results

- Longer burst includes more parity bytes.
  - Overall error performance remains same
- Same average BER
  - Same delimiter loss probability (same delimiter)
  - Larger penalty when loss occurs (more buts lost per event)
  - Smaller delimiter rate (frames are bunched together)

Conclusions

- Addition to FEC using bursting proposed
- Reduction in framing overhead achieved without significant increase in complexity
- Mean overhead reduced for traffic with variable frame size
- Frame structure for FEC and Ethernet framing maintained

Appendix - Compatibility

- EPONs require fixed delay through Ethernet stack for timestamp mechanism to function
  - EPON implementation must compensate by maintaining constant delay in decoder of 3035
  - Previously this was constant delay of 1518 w/o bursting
- Auto-negotiation special characters are not protected
  - Same as w/o bursting
- Small OAM frames are still protected
  - Same as w/o bursting