Functions for Computing FEC Overhead

Lior Khermosh
Jeff Mandin
PMC-Sierra

802.3av Munich – May 2008

Summary

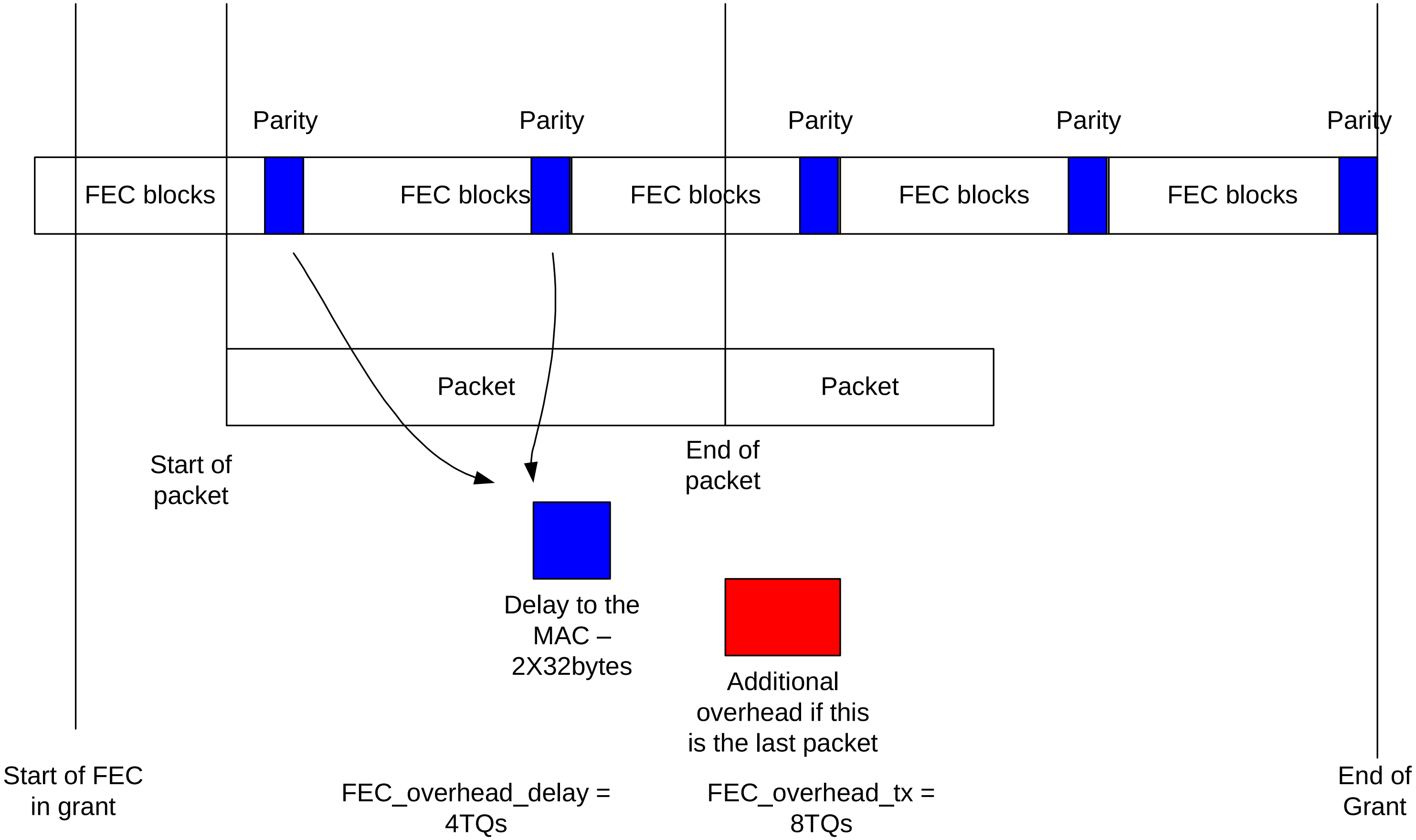
1. Simple functions for FEC overhead calculation for use in MPCP layer:
   - FEC_overhead_delay for between-frame backoff
   - FEC_overhead_tx to compute whether frame fits into grant
   - Detailed descriptions of parameters in the comment database

2. Optimum use of bandwidth
   - Keep the overhead between frames to the minimum required (ie. just enough for parity and IPG)

   - Keep the overhead at end of upstream burst to the minimum required (ie. just enough for IPG, and FEC codeword completion)

3. No delay of frames in the MAC sublayer
   - satisfy the requirements on delay variability from **93.3.2.4**

Concept


Parity
Parity
Parity
Parity
Parity
FEC blocks
FEC blocks
FEC blocks
FEC blocks
FEC blocks
Packet
Packet
Start of packet
End of packet
Delay to the MAC – 2X32bytes
Additional overhead if this is the last packet
Start of FEC in grant
FEC_overhead_delay = 4TQs
FEC_overhead_tx = 8TQs
End of Grant

FEC_Overhead_delay() function

1. Calculates the delay required after a packet due to insertion of FEC overhead

2. For each frame, required delay is calculated before invocation of TransmitFrame() :

   payloadBalance = ((localTime – beginTime) * 20) % 248
   + FrameLength + IPG

   FEC_overhead_delay =
   round_up(32/20 * round_down(payloadBalance / 216)) [TQ]

<u>Or:</u>

FEC_overhead_delay =
32 * round_down(payloadBalance / 216)   [Bytes]

FEC_Overhead_delay() function

Notes:

- Avg value for IPG (ie. 12) is used
- the packet size (plus the "balance" remaining from the previous packet) is rounded down to the nearest FEC codeword.
- If the FEC codeword is not full then there is no delay added for parity at the MPCP layer

- Inaccuracies in estimation are due to:
  - TQ granularity (ie. up to one byte less than a TQ might not be counted)
  - Uncertainty in Deficit IDLE Count (ie. up to 3 bytes might not be counted)

- Inaccuracies in estimation compensate across frames. Inaccuracies in estimation *do not accumulate.*

FEC_Overhead_tx() function

1. This function calculates the size of additional overhead, to be added by the FEC encoder, The function is used to check if the packet fits the grant.

2. For each frame:

```
payloadBalance = ((localTime – beginTime)*20) % 248
                  + length + IPG



FEC_overhead_tx =

    [round_up(((248) * round_up( payloadBalance / 216)]
                          - payloadBalance)
   --------------------------------------------------------------
                                20                                   [TQ]
```

FEC_Overhead_tx() function

Notes:

- Use worst-case assumptions about FEC parity requirements for the frame:
  - Max value for IPG (ie. 15) is used
  - the packet size (plus the "balance" remaining from the previous packet) is rounded up to the nearest FEC codeword.
- Nevertheless, the function can still determine if a frame can fit into a codeword (without requiring a full codeword for the final frame in the burst).

Latency Variation Requirement

93.3.2.4 Delay requirements

The MPCP protocol relies on strict timing based on distribution of timestamps. A compliant implementation needs to guarantee a constant delay through the MAC and PHY in order to maintain the correctness of the timestamping mechanism. The actual delay is implementation dependent, however, a complying implementation shall maintain a delay variation of no more than 1 time_quantum through the implemented MAC stack.

What happens if calculation does not use payloadBalance?

1. **Frames get delayed in the MAC for longer than 1 TQ, leading to violation of limit to delay variation**

2. **There are occasions where a frame remains untransmitted even though it fits into the final FEC codeword.**