[add variables to 77.2.2.3:]

byteTime

TYPE: 32 bit unsigned

A clock that advances by 1 after every 8 bit times. The variable is periodically adjusted to avoid rollover.

In the OLT, this variable initialized to 0 at system initialization.

In the ONU, this variable is assigned in the GATE Processing ONU Activation state diagram.

[Replace definition of FEC_Overhead_min() (and associated table 77-1) in 77.2.2.4 with the following:]

FEC_Overhead_min(length)

This function calculates the amount of time (in octet times) that the MPCP control multiplexer must wait following transmission of a frame of size *length* so as to allow the insertion of parity data into the frame by the PHY layer.

As described in Clause @@92.2.3@@, FEC encoder adds 32 parity octets for each block of 216 data or control octets. *FEC_Overhead_delay*() returns the number of octets that the PHY will insert during transmission of a particular packet and its subsequent IPG.

Parameter *length* represents the size of an entire frame including preamble, SFD, DA, SA, Length/Type, FCS, and IPG.

The following formula is used to calculate the overhead:

```
FEC_Overhead_min(length)
{
   byte_time =  byteTime  Mod FEC_CODEWORD_SIZE

   return (FEC_PARITY_SIZE *   |__ ((byte_time + length) / FEC_PAYLOAD_SIZE))__|

}
```

NOTE–The notation |_ _| represents a *floor* function, which returns the value of its argument *x* rounded down to the nearest integer.

[Replace definition of FEC_Overhead_max() in 77.2.2.4 with the following:]

FEC_Overhead_Max(length)

This function calculates the size of additional overhead to be added by the FEC encoder while encoding a frame of size length as the final frame of a burst. As described in @@Subclause 76.2.3@@, FEC encoder adds 32 parity octets for each block of 216 data or control octets. This is equivalent to an overhead of 4 time_quanta for every 27 time_quanta transmitted. The following formula is used to calculate the overhead:

```
FEC_Overhead_Max(length)
{
   byte_time  =  byteTime  Mod FEC_CODEWORD_SIZE


  return(
        FEC_PAYLOAD_SIZE – ((byteTime+ length) mod  FEC_PAYLOAD_SIZE)    +

             FEC_PARITY_SIZE *   ⌈((byte_time + length) / FEC_PAYLOAD_SIZE))⌉)
    )

}
```

NOTE–The notation | | represents a *ceiling* function, which returns the value of its argument *x* rounded up to the nearest integer.

[Make the following changes to the OLT control multiplexer state diagram (77-12):]

- Change the first line in the Start_Packet_Initiate_timer state to this:

```
Packet_initiate_delay =  sizeof(data_tx) + tailGuard +
                        FEC_Overhead_Min(sizeof(data_tx) + tailGuard)
```

[Make the following changes to the ONU control multiplexer state diagram (77-13):]

- Change the first line in the Start_Packet_Initiate_timer state to this:

```
Packet_initiate_delay =  sizeof(data_tx) + tailGuard +
                        FEC_Overhead_Min(sizeof(data_tx) + tailGuard)
```

[Make the following changes to diagram 77-29:]

- Add the following line to the beginning of the START_TX state

```
ByteTime <=  [ ((laserOnTime + syncTime) *20) + 8) * -1]
```