EPOC

EPOC ISSUES IN CL 102

tqSize

- tqSize is used to generate OctetsRemaining via the formula on pg 95 ln 14 : (((stopTime – localTime) x tqSize) – tqOffset)
  - Note if tqSize < 1 the grant will effectively be terminated
- In 64.2.2.1 “Constants” - tqSize is defined as: “This constant represents time_quantum in octet transmission times” with a value of 2 Octets/TQ
- In 77.2.2.1 & 102.2.2.1 “Constants” - tqSize is defined with a value of 20 Octets/TQ
- tqSize is defined as an integer

What about EPoC?

- Integers < 1 would need to be rounded down (=0)

EPoC tqSize							
		Bit Loading (b/Hz) / QAM					
RF CH (Mhz)	Num SC	7	8	9	10	11	12
		128	256	512	1024	2048	4096
24	480	0.34	0.38	0.43	0.48	0.53	0.58
30	600	0.42	0.48	0.54	0.60	0.66	0.72
36	720	0.50	0.58	0.65	0.72	0.79	0.86
42	840	0.59	0.67	0.76	0.84	0.92	1.01
48	960	0.67	0.77	0.86	0.96	1.06	1.15
54	1080	0.76	0.86	0.97	1.08	1.19	1.30
60	1200	0.84	0.96	1.08	1.20	1.32	1.44
66	1320	0.92	1.06	1.19	1.32	1.45	1.58
72	1440	1.01	1.15	1.30	1.44	1.58	1.73
78	1560	1.09	1.25	1.40	1.56	1.72	1.87
84	1680	1.18	1.34	1.51	1.68	1.85	2.02
90	1800	1.26	1.44	1.62	1.80	1.98	2.16
96	1920	1.34	1.54	1.73	1.92	2.11	2.30

tqOffset (fyi)

- tqOffset is defined in 102.2.2.3 “Variables” pg 96 ln 5 as a variable that “denotes the offset (in octet times) of the current actual time from the localTime variable (which maintains the current time in units of time_quanta)”. It is an 8 bit unsigned variable. Not explicitly set in any SD.
- Only referenced in OctetsRemaining formula in 102.2.2.3 “Variables” on pg 95 ln 14 : (((stopTime – localTime) x tqSize) – tqOffset)
- The term “actual time” only appears in the definition above.

fecOffset

- Definition in 102.2.2.3 “Variables” pg 94 ln 22:
  - “A variable that advances by PHY_DATA_SIZE after every 8×(PHY_DATA_SIZE + PHY_OVERHEAD_SIZE) bit times (see EPoC de-rating equation 102-1). After reaching the value of FEC_CODEWORD_SIZE, this variable is reset to zero. In the CLT, this variable is initialized to 0 at system initialization. In the CNU, this variable is assigned in the GATE Processing CNU Activation state diagram (see Figure 102–14)”
  - In Figure 102–14 this is set to magic number16 (inherited from Fig 77-14 ONU Ctrl Mux SD represents 2 x 66b symbols in EPON for optical sync and RS alignment)
- Definition in 77.2.2.3 “Variables”:
  - “A variable that advances by 1 after every 8 bit times. After reaching the value of FEC_CODEWORD_SIZE, this variable is reset to zero. In the OLT, this variable is initialized to 0 at system initialization. In the ONU, this variable is assigned in the GATE Processing ONU Activation state diagram (see Figure 77–14)”
- fecOffset is a 32 bit unsigned variable use in numerous calculations & SDs (CheckGrantSize, P*MD_Overhead, Figure 102–13 and, Figure 102–14 CLT & CNU Ctrl Mux SDs resp))*

Eq 102-1

$$\beta = \frac{XGMII\_Rate}{PCS\_Rate} = \frac{PHY\_DATA\_SIZE + PHY\_OVERHEAD}{PHY\_DATA\_SIZE}$$

PHY_DATA_SIZE & PHY_OVERHEAD_SIZE

- Defined in 102.2.2.3 “Variables” pg 96 ln 35
- PHY_DATA_SIZE defined as an integer with “The number of octets constituting the denominator in the EPoC de-rating Equation (102–1). To normalize the effective data rate, the MPCP control multiplexer waits PHY_OVERHEAD_SIZE octets per every PHY_DATA_SIZE octets transmitted”
- PHY_OVERHEAD_SIZE TYPE defined as an integer with “The number of octets constituting (together with PHY_DATA_SIZE) the numerator in the EPoC de-rating Equation (102–1). To normalize the effective data rate, the MPCP control multiplexer waits PHY_OVERHEAD_SIZE octets per every PHY_DATA_SIZE octets transmitted”

Eq 102-1

$$\beta = \frac{XGMII\_Rate}{PCS\_Rate} = \frac{PHY\_DATA\_SIZE + PHY\_OVERHEAD}{PHY\_DATA\_SIZE}$$

What about this β?

- “Defined” in Eq 102-1 as ratio of XGMII rate to PCS Rate
- Used to calculate PMD_Overhead

PMD_Overhead (fyi)

- Analogous to FEC_Overhead in 2012 STD (which also appears in *Figure 102–13 CLT Ctrl Mux SDs*)
- Defined in 102.2.2.4 “Functions” pg 97 ln 22 as “This function calculates the additional amount of time (in octet times) that the MPCP control multiplexer waits following transmission of a frame of size ‘length’ by the MAC. The additional time is added to allow the insertion of parity data into the frame by the PHY layer and to adjust the data rate to the effective data rate supported by the PCS and PMD. PMD_Overhead(). returns the number of octets that the PHY inserts 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.” see formula below
- PMD_Overhead is currently only referenced in Figure 102–14 CNU Ctrl Mux SD where it is used to set packet_initiate_delay
  packet_initiate_delay <= PMD_Overhead(sizeof(data_tx) + tailGuard)
- Probably should also be used in Figure 102-13 for the same calculation (instead undefined param FEC_Overhead is used here)

$$PMD\_Overhead((length),\beta) =$$

$$12+\left\lceil (\beta-1)\times length+\beta\times\left(FEC\_PARITY\_SIZE\times\left\lfloor\frac{fecOffset+length}{FEC\_PAYLOAD\_SIZE}\right\rfloor\right)\right\rceil$$

packet_initiate_delay

- Defined in 102.2.2.3 “Variables” pg 95 ln 32 is a 16 bit unsigned value defined as:
  “This variable is used to set the time?ut interval for packet_initiate_timer defined in 102.2.2.5. The packet_initiate_delay value is represented in units of octets”
  - 16 bits is too small as the function PMD_Overhead will yield a value of 117,179 at min RF channel (24 MHz) min Mod Order (256 QAM)
- packet_initiate_timer is defined in 102.2.2.5 “Timers” pg 98 ln 27. It is defined as “This timer is used to delay frame transmission from MAC Control to avoid variable MAC delay while MAC enforces IPG after a previous frame. In addition, this timer increases interframe spacing just enough to accommodate the extra parity data to be added by the FEC encoder”

tailGuard (fyi)

- Defined in 102.2.2.1 “Constants” pg 93 ln 36. as an integer “This constant holds the value used to reserve space at the end of the upstream transmission at the CNU in addition to the size of last MAC service data unit (m_sdu) in units of octets. Space is reserved for the MAC overheads including: preamble, SFD, DA, SA, Length/Type, FCS, and minimum interpacket gap. The sizes of the above listed MAC overhead items are described in 3.1.1. The size of the minimum IPG is described in 4A.4.2.”

CheckGrantSize

- Defined in 102.2.2.4 “Functions” pg 97 ln 12 as “This function calculates the future time at which the transmission of the current frame (including the FEC parity overhead) is completed“
- The function is called in *Figure 102–13 and, Figure 102–14 CLT & CNU Ctrl Mux SDs resp. in both cases it is used to set OctetsRequired.*
  - OctetsRequired <= CheckGrantSize(sizeof(data_tx) + tailGuard))
- Does not account for
  - multiple FEC sizes
  - FEC foreshortening
- Eq 102-2

$$eckGrantSize(length) = \left\lceil \frac{fecOffset + length}{FECPAYLOADSIZE} \right\rceil \times FEC\_CODEWORD\_SIZE - fecOffse$$