Reporting at 10Gb/s

Eric Lynskey

IEEE 802.3av Interim Meeting May 13-15, 2008

Munich, Germany

Review of Tokyo straw polls and motion

- When calculating REPORT

– Include FEC overhead	_12_
– **Do not include overhead**	**_14_**
– Don’t care	_21_

- Report in units of

– FEC codewords	_0_
– **time_quanta**	**_23_**
– Bytes	_0_
– 66-bit blocks	_2_
– Don’t care	_20_

- When calculating REPORT round up to integer number of FEC codewords (nearest time_quantum)

– Yes	_0_
– **No**	**_24_**
– Don’t care	_26_

- REPORT messages shall report queue length in units of time_quanta without rounding up to integer number of FEC codewords.
  - **Yes: Passed by voice without opposition**
  - No:
  - Abstain:

How does ONU report?

- From 93.3.6.3, “[t]he reported length shall be adjusted to account for the necessary inter–frame spacing and FEC parity data overhead, if FEC is enabled.”

- ONU must begin burst at start of FEC codeword boundary and must end burst at end of full FEC codeword.

- It makes most sense for ONU to continue to request integer number time_quantua.

- With the existing FEC and definitions of time_quanta, it makes sense for the ONU to not report overhead.

Propose that REPORT is calculated without FEC overhead.

Errors in reporting

- ONU can only request in units of time_quanta.
  - Need to round up to next time_quantum
  - Possible error of up to 19 bytes
- ONU does not know exact IPG
  - Need to round up for deficit idle count
  - Possible error of up to 3 bytes
- Single priority versus multiple priority reporting
  - How is overhead calculated?
  - Is error accumulated per priority?

TEKNOVUS®

How does OLT grant?

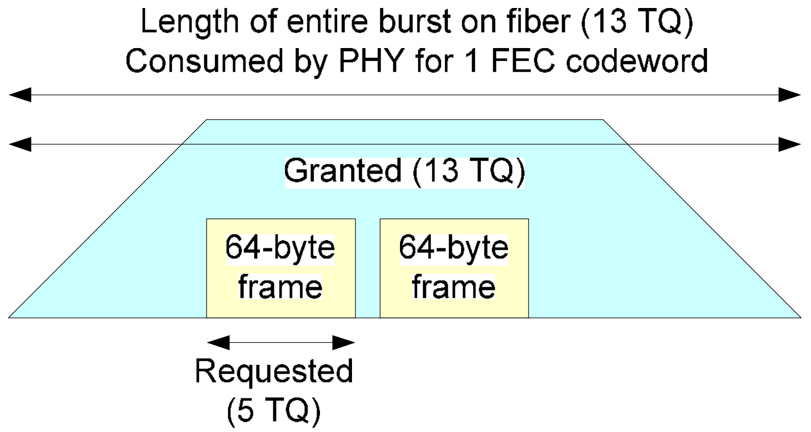
- The OLT is free to grant whatever bandwidth it wants to any ONU it wants. OLT DBA does not have to align grants to FEC code words.

- Since ONU must end burst after a complete FEC code word, OLT scheduler must schedule in FEC code words.

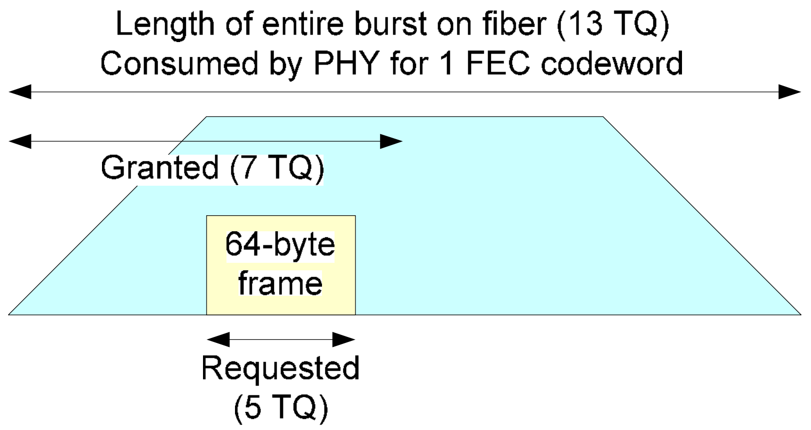
- Example
  - ONU requests single 64-byte frame
  - OLT schedules 1 FEC code word
  - OLT could grant 1 FEC code word or potentially only enough time_quanta for single 64-byte frame plus overhead.

Grant alignment to FEC code word

- If OLT grants whole code word, ONU could send more frames then requested, and you lose some control over bandwidth allocation.


Length of entire burst on fiber (13 TQ)
Consumed by PHY for 1 FEC codeword
Granted (13 TQ)
64-byte frame
64-byte frame
Requested (5 TQ)

- If OLT grants just enough for requested data and overhead, ONU can only send a single frame.


Length of entire burst on fiber (13 TQ)
Consumed by PHY for 1 FEC codeword
Granted (7 TQ)
64-byte frame
Requested (5 TQ)

NOTE: DBA is out of scope, granting is implementation choice.

Report for single priority with NO overhead

- Data = Length1 + Length2 + … + LengthN
- Preamble = N*8
- IPG = N*12 + 3
- Payload = Data + Preamble + IPG
- Report = Ceiling(Payload / 20)

- 64-byte request = 5 time_quanta
- DBA grant = 7 time_quanta (1 frame + overheads)
- Scheduler = 13 time_quanta (1 FEC codeword)

- 8x64-byte request = 34 time_quanta
- DBA grant = 41 time_quanta (8 frames + overhead)
- Scheduler = 50 time_quanta (4 FEC codewords)

Report for multiple priorities with NO overhead

- Calculate per priority:

```
Pri0 64-byte frame request  =   5 TQ
Pri1 64-byte frame request  =   5 TQ
           …
Pri7 64-byte frame request  =   5 TQ
Total time requested        =  40 TQ
OLT could grant             =  47 TQ
OLT would schedule          =  50 TQ
```

- In this example, no extra FEC code words necessary for REPORT frame with multiple priorities vs. single priority.

- Some combinations could require additional code word.

OLT calculation of grant and burst length

- **Variables**
  - report: reported time_quanta from ONU
  - laserOn: laser on time in time_quanta
  - laserOff: laser off time in time_quanta
  - syncTime: sync time in time_quanta (includes 0x5555… and BURST_DELIMITER)
- **Constants**

– tqSize: number of bytes per time_quantum,	value = 20
– idles: bytes of idle to be added at start of burst,	value = 16
– codeWordSize: number of bytes per FEC code word,	value = 216
– paritySize: number of parity bytes per FEC code word,	value = 32
– vectorSize: number of time_quanta per FEC code word,	value = 12.4

$$GrantLength = \frac{\left\lceil report \times tqSize + \left(\left\lceil \frac{report \times tqSize + idles}{codeWordSize} \right\rceil \times paritySize \right) \right\rceil}{tqSize}$$

$$BurstLength = \left\lceil \left\lceil \frac{report \times tqSize + idles}{codeWordSize} \right\rceil \times vectorSize \right\rceil + laserOn + laserOff + syncTime$$

ONU calculation of grant length

- ONU processes new grant
  - stopTime = currentGrant.start + currentGrant.length – laserOn – laserOff – SyncTime
  - effectiveLength = stopTime – localTime
  - transmitAllowed = TRUE
- ONU looks at the size of current frame to see if it will fit in grant

$$nextTxTime = \frac{(sizeof(data\_tx) + tailGuard) + FEC\_Overhead(sizeof(data\_tx) + tailGuard)}{tqSize}$$

- In previous example, ONU receives grant length of 7 time_quanta.
- ONU determines that only a single 64-byte frame can fit in current grant.

- FEC_Overhead function needs to be fixed, but that is covered by separate proposal.
- No changes proposed for how ONU processes grants.

- One possible area of improvement is to remove laserOnTime, laserOffTime, and syncTime from the grants.
  - After registration, these values are fixed.
  - OLT adds this fixed value to grant, and ONU subtracts this fixed value.
  - No need to perform this redundant step.
  - Minor changes to some definitions and state diagrams required.

Conclusion

- REPORT format stays the same.
- REPORT should include data + ipg + preamble and be rounded up to the nearest time_quantum.
- REPORT should be calculated without FEC overhead included.
- Each priority and queue set amount is calculated independently (error can accumulate).
- Should investigate whether or not to include overheads in grant.