# Fragmentation

One option

www.huawei.com

Author/ Email: Duane Remein

Version: V1.0(20YYMMDD)

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

# Why fragmentation?

- **Previous proposals for channel bonding all have obvious flaws**
  - How to fill grants issues based on reports of packet boundaries (kramer_3ca_2b_0116)?
  - Bonding by Burst (kramer_3ca_2a_0316): huge buffers (several MB), limited visibility on data volume.
  - Bonding by Frame (effenberger_3ca_1_0516): large buffers (Max frame per lane), pre-sorting data.
- **Fragmentation is viewed as a mechanism to minimize buffer requirements and also eliminate need to have detailed information on data packet boundaries**
- **Lastly as Glen pointed out in Jan (see kramer_3ca_2a_0116):**

**"Everyone should drop what they are doing and work on the multi-lane scheduling issues. Seriously."**

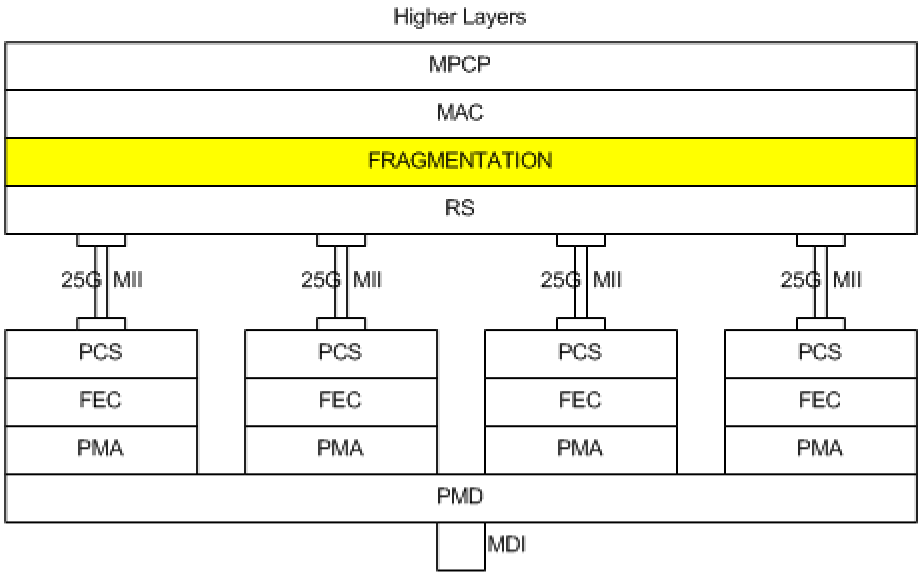
- **Ref: kramer_3ca_2b_0116.pdf, kramer_3ca_2a_0316.pdf, effenberger_3ca_1_0516.pdf, kramer_3ca_1a_0516.pdf**

# How do we accomplish fragmentation?

- **Need a method to signal start & terminate the fragment over the MII**
  - LLID is unknown to PHY
  - Lane usage by LLID is unknown to PHY
- **Need a method to signal that fragments in different lanes are associated with the same or different LLIDs**
  - One ONU should be able to send different LLIDs on different lanes simultaneously and fragment each frame in process.
- **Need a method to ensure fragments are aligned in the proper order at the receiver**
  - Don’t assume that fragments from different lanes are aligned at the transmitter
  - Cannot assume that fragments from different lanes are aligned at the receiver
- **Need to agree on the minimum fragment size**

# Proposal

- **Add Fragmentation Layer**
  - Terminates MACs
  - Handles frame distribution between 4 potential input MACs and up to 4 output PLS interfaces
  - Aligns fragments on receive side
- **RS**
  - Modified to recognize Fragment End and Start

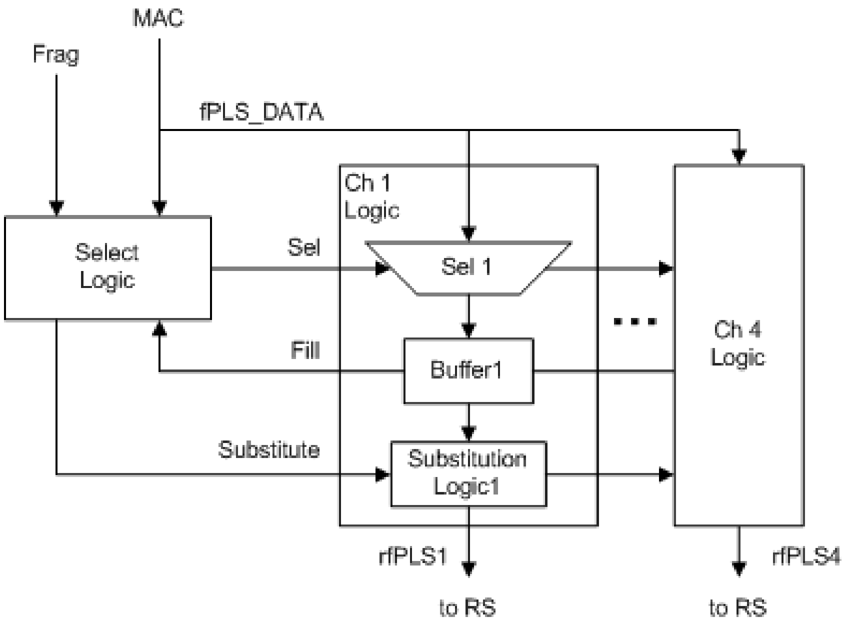


**Fragmentation & RS could be combined into a modified RS if desired**

# Fragmentation Layer Tx

- **Frag (per LLID)**
  - Indicates which lanes to use
  - Indicates when to insert markers
  - Like MM_CTRL.request in 802.3br
- **Select Logic**
  - Controls source of data to input to Buffer 1..n (data distribution based on 1st available allowed lane)
  - Controls Substitution Logic 1..n
  - Inputs fill level from Buffer 1..n
- **Sel 1..n**
  - When enabled takes data from MAC and transfers it to Buffer 1..n
- **Substitution Logic 1..n**
  - When Enabled by Select Logic inserts fragment markers

- **Buffer 1..n**
  - ~ 1 Fragment
  - Could be used to pre-fetch data to align fragments scheduled to leave at the same time



# Fragment Start

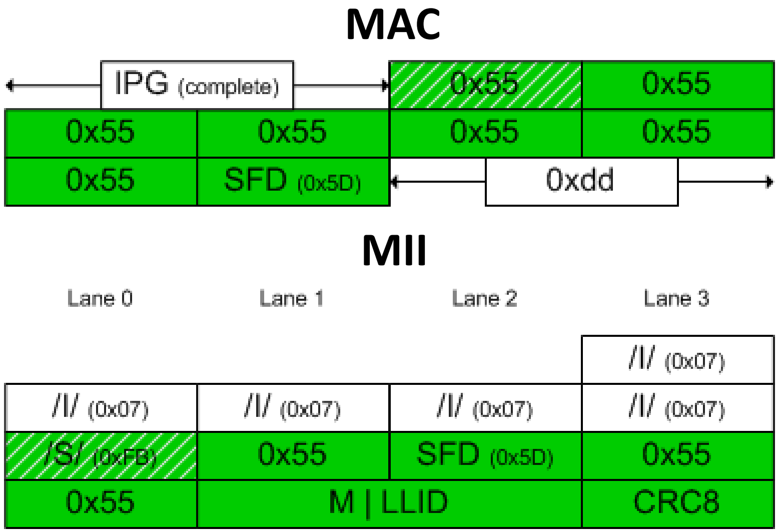
- **Variation on standard preamble**
  - SFM – Start Fragment Marker
    - **Indicates Fragment *f of transmission in n Lanes***
    - **Ex. for a transmission that occupies all 4 lanes where fragment start order by lane is lane 2, 4, 3, 1**

      **SFM Lane 1 = 4 of 4**

      **SFM Lane 2 = 1 or 4**

      **SFM Lane 3 = 3 of 4 and**

      **SFM Lane 4 = 2 of 4**
  - Need total of 16 values
- **Frag transition from 0 to 1**
  - indicates insert Fragment Start at the beginning of the next fragment



**MAC**

| | | | |
|---|---|---|---|
| ← IPG (complete) | → | ▨ 0x55 | 0x55 |
| 0x55 | 0x55 | 0x55 | 0x55 |
| 0x55 | SFD (0x5D) | ← 0xdd | → |

**MII**

| Lane 0 | Lane 1 | Lane 2 | Lane 3 |
|---|---|---|---|
| | | | /I/ (0x07) |
| /I/ (0x07) | /I/ (0x07) | /I/ (0x07) | /I/ (0x07) |
| ▨ /S/ (0xFB) | 0x55 | SFD (0x5D) | 0x55 |
| 0x55 | M \| LLID | | CRC8 |

**Fragment Start (MII)**

| | | | |
|---|---|---|---|
| /Fs/ (tbd) | 0x55 | SFM (tbd) | 0x55 |
| 0x55 | M \| LLID | | CRC8 |

May not need new control code

# Fragment Terminate

- **SoF – Size of Fragment**
  - Indicates size of the remaining fragment (optional, borderline prescient)
  - Would be valuable to DBA
- **Termination marker could be shortened to 4 bits if omitted**
- **Frag transition from 1 to 0**
  - indicates insert Fragment Terminate at the end of the next fragment

<table>
<tr><td>/Ft/ (tbd)</td><td>0x55</td><td>p | SoF</td><td>0x55</td></tr>
<tr><td>0x55</td><td colspan="2">M | LLID</td><td>CRC8</td></tr>
</table>

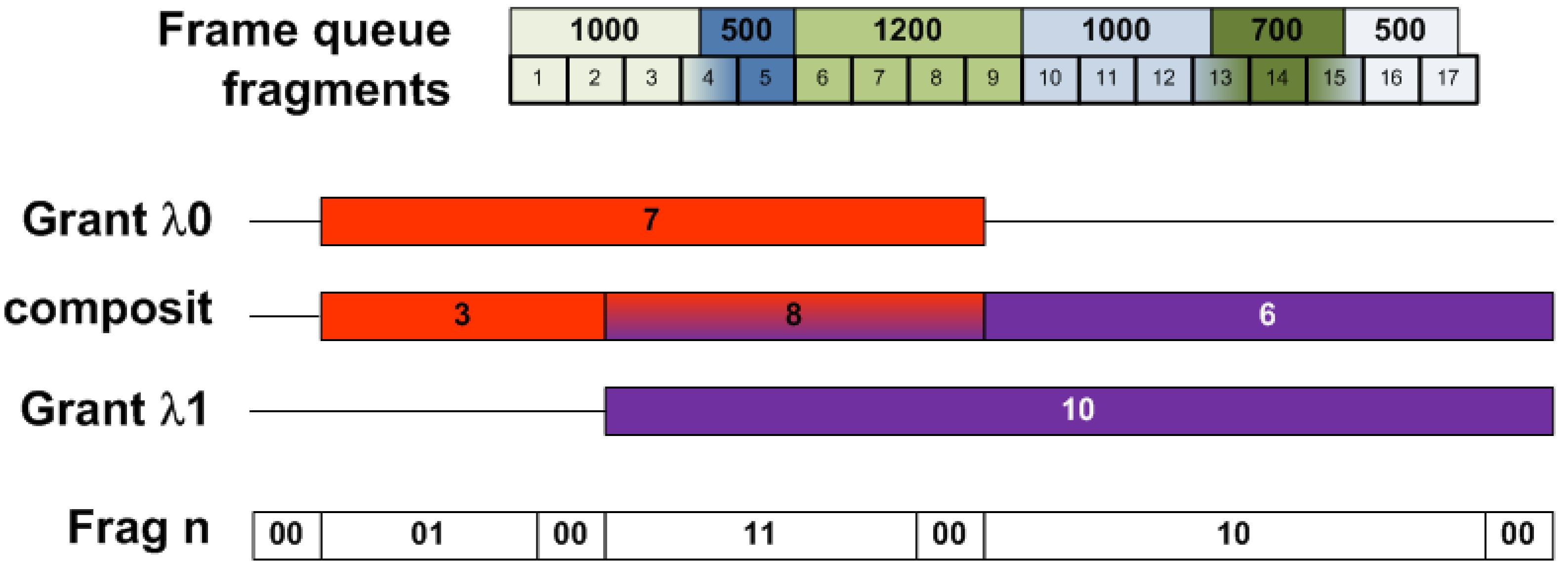
May not need new control code

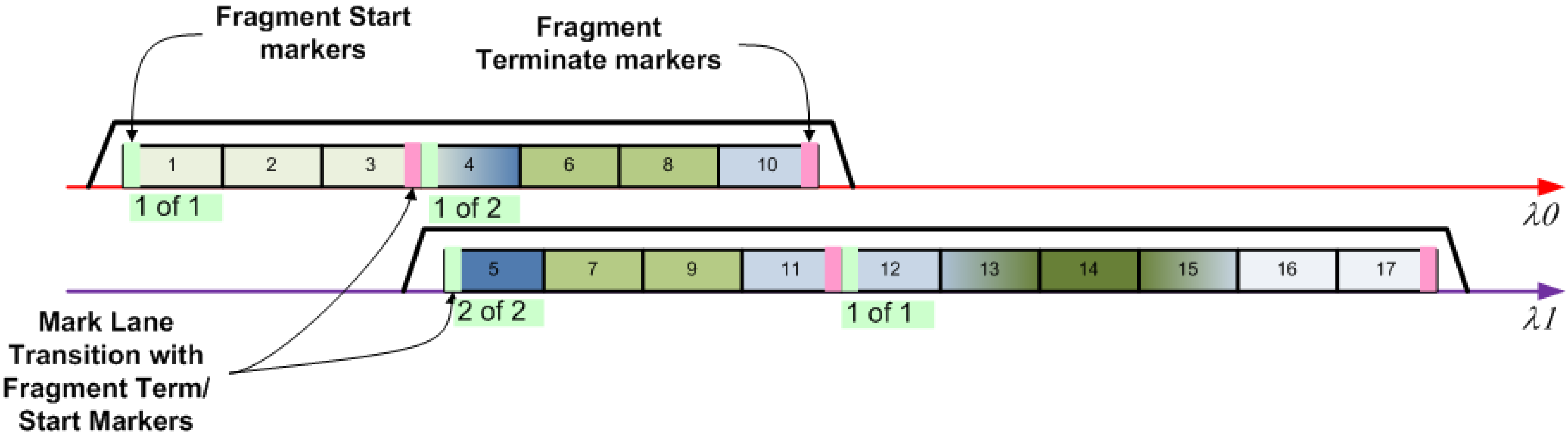
# How it works

(illustration on next slide)

1. **ONU has 6 packets totaling 4900B**
2. **ONU receives two grants;**
    1. Grant λ0 is for 7 fragments (Grants issued in whole fragments)
    2. Grant λ1 is for 10 fragments starting aprox 3 fragment times after Grant λ0.
3. **MPCP+ signals Tx on Lane 0 (adds ovhd to lane 0 calculation*)**
4. **RS sends Frag Start marker (1 of 1) and data for 2 fragments (discounted by Start/Term marker) to Lane 0**
5. **MPCP+ signals Term Fragment followed by Tx on Lane 0 & 1 (adds ovhd to Lane 0/1 calculations *)**
6. **RS sends fragment #3 with Term marker in Lane 0**
8. **RS sends Start marker (1 of 2) and fragment #4 in Lane 0**
9. **RS sends Start marker (2 of 2) and fragment #5 in Lane 1**
10. **RS Alternates sending fragments to Lane 0 and Lane 1 until Fragment 10 is sent to Lane 0**
11. **MPCP+ signals Term Fragment and Tx on Lane 1 (adds ovhd to lane 1 calculation*)**
12. **RS sends fragment #10 with Term marker in Lane 0**
13. **RS sends fragment #11 with Term marker in Lane 1**
14. **RS sends Start marker (1 of 1) to Lane 1**
15. **RS sends remaining fragments to Lane 1 with fragment #17 ending in a Term marker**

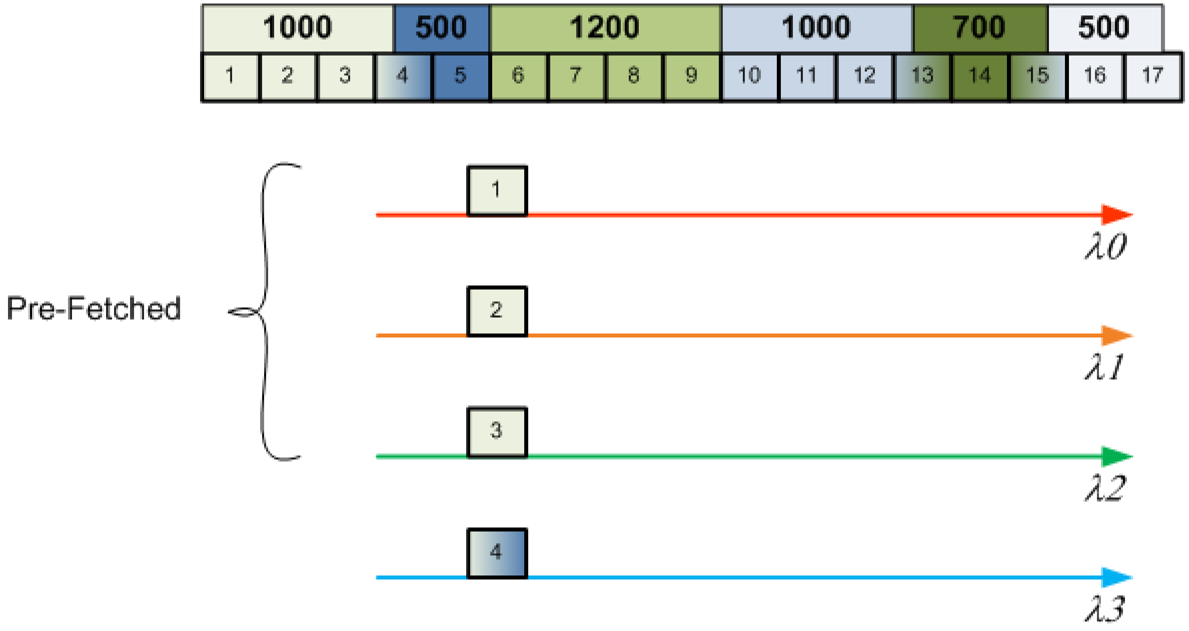
* Frag Markers would represent another overhead and would be calculated similar to packet_initiate_delay in TRANSMIT FRAME state Figure 77–14—ONU Control Multiplexer state diagram


Frame queue
fragments
1000
500
1200
1000
700
500
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Grant λ0
7
composit
3
8
6
Grant λ1
10
Frag n
00
01
00
11
00
10
00


Fragment Start
markers
Fragment
Terminate markers
1 2 3 4 6 8 10
1 of 1
1 of 2
λ0
5 7 9 11 12 13 14 15 16 17
2 of 2
1 of 1
λ1
Mark Lane
Transition with
Fragment Term/
Start Markers

# Buffering Rx

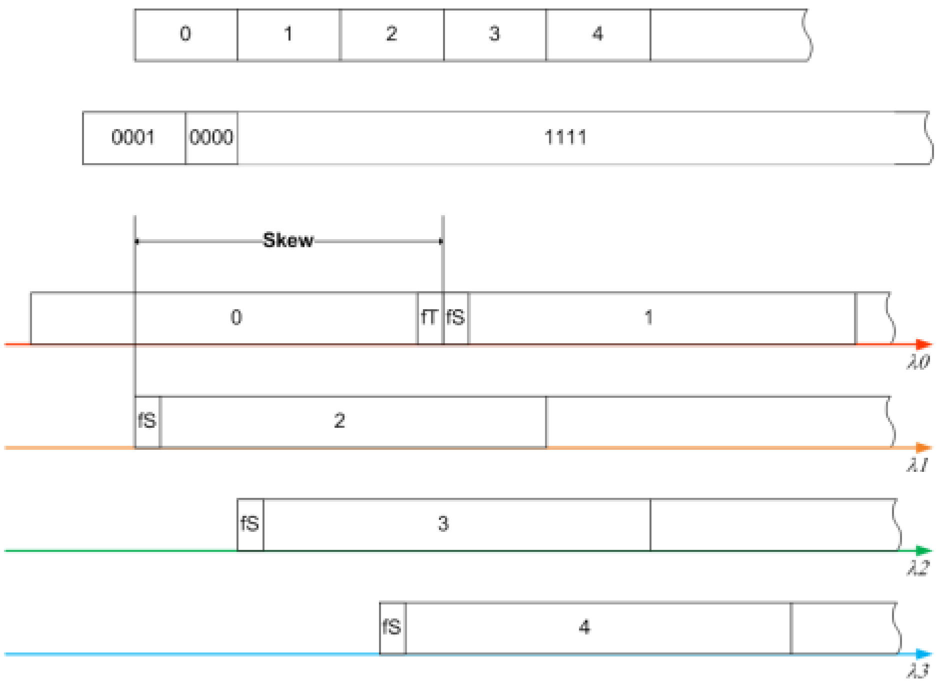
- In order to allow for alignment of coincident burst starts at the Tx a minimum of 3 fragments would need to be buffered



# Buffering Rx

- **Dependent on Skew at Receiver**
  - Δt at transmitter (imp. dep.)
    - **fix max in std.**
  - ΔtDelay at transmitter (imp. dep.)
    - **fix max in std.**
  - Δt due to wavelength transmission speed
    - **< 50 ns?, depends on wavelength plan**
  - ΔtDelay at receiver (imp. dep.)
    - **vendor specific no need to std.**
  - Transmission time for 1 FEC CW (RS256,223) is ~80 ns
  - Some methods to eliminate skew are under discussion (ex. yin_3ca_1_0716)

- May need to balance buffer requirements between ONU (once per ONU) vs OLT (once per MAC)



# Thank you

www.huawei.com

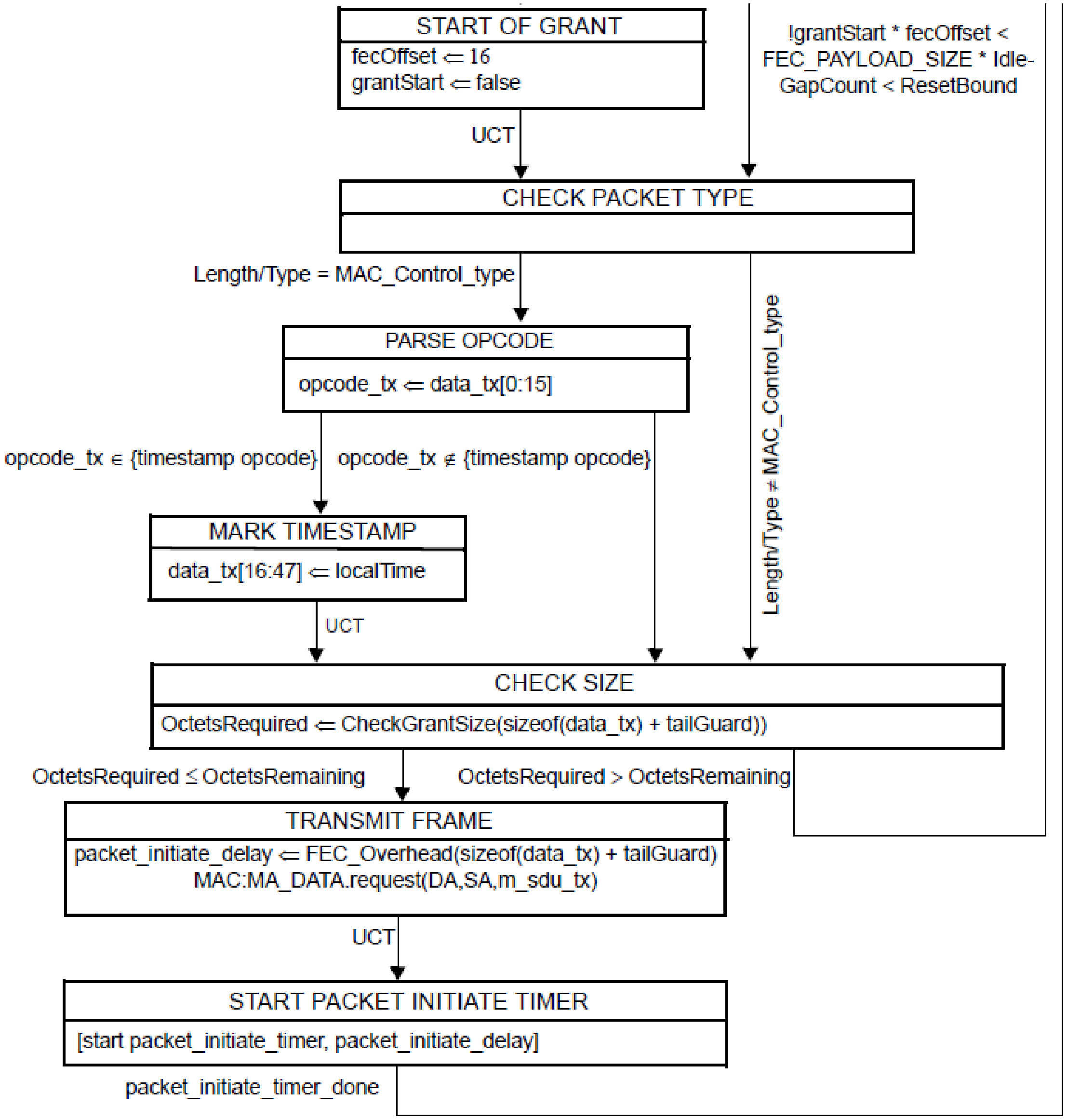
# Backup

www.huawei.com

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

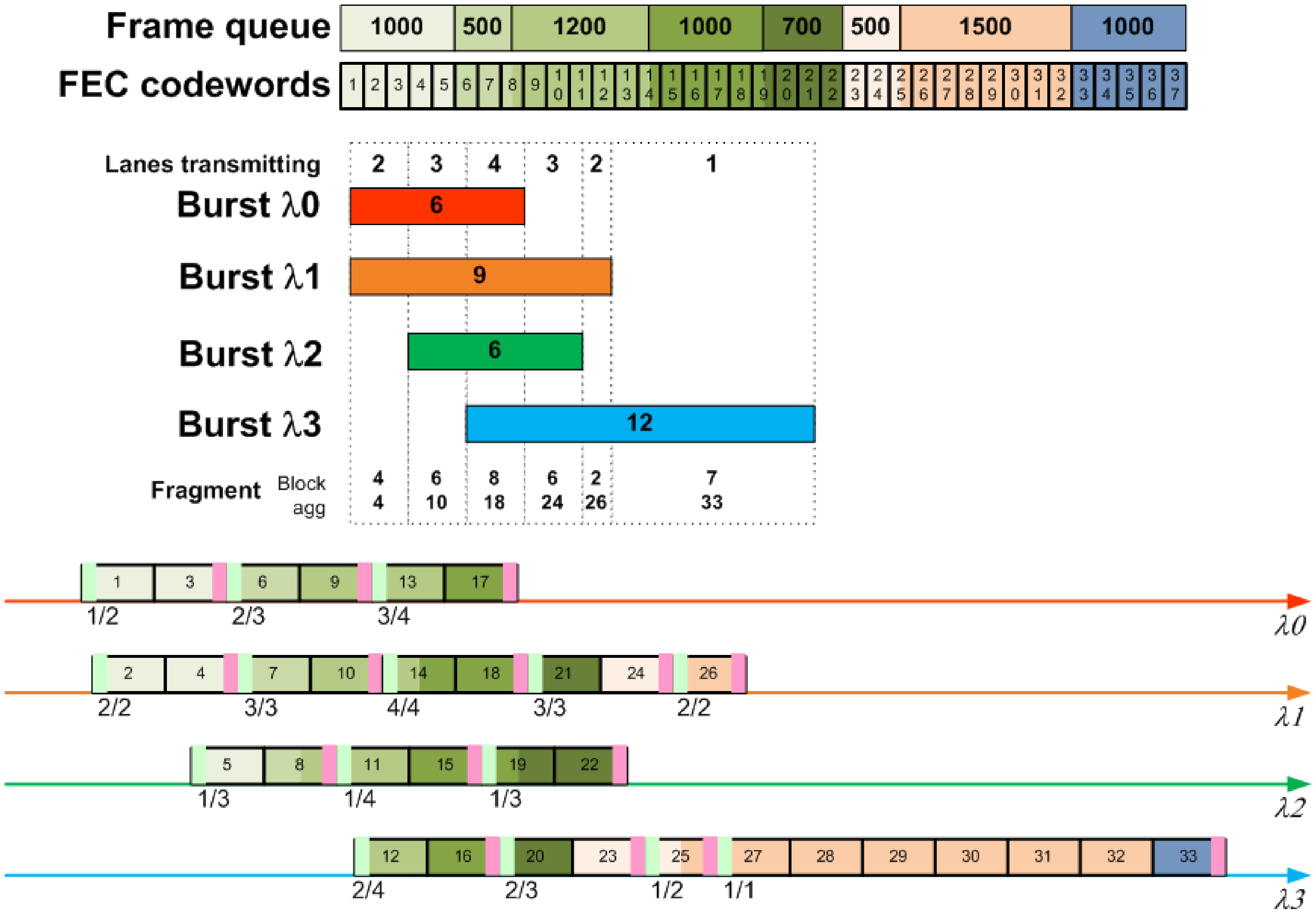
# From Fig 77-14


START OF GRANT
fecOffset ⇐ 16
grantStart ⇐ false
!grantStart * fecOffset < FEC_PAYLOAD_SIZE * Idle-GapCount < ResetBound
UCT
CHECK PACKET TYPE
Length/Type = MAC_Control_type
Length/Type ≠ MAC_Control_type
PARSE OPCODE
opcode_tx ⇐ data_tx[0:15]
opcode_tx ∈ {timestamp opcode}
opcode_tx ∉ {timestamp opcode}
MARK TIMESTAMP
data_tx[16:47] ⇐ localTime
UCT
CHECK SIZE
OctetsRequired ⇐ CheckGrantSize(sizeof(data_tx) + tailGuard))
OctetsRequired ≤ OctetsRemaining
OctetsRequired > OctetsRemaining
TRANSMIT FRAME
packet_initiate_delay ⇐ FEC_Overhead(sizeof(data_tx) + tailGuard)
MAC:MA_DATA.request(DA,SA,m_sdu_tx)
UCT
START PACKET INITIATE TIMER
[start packet_initiate_timer, packet_initiate_delay]
packet_initiate_timer_done

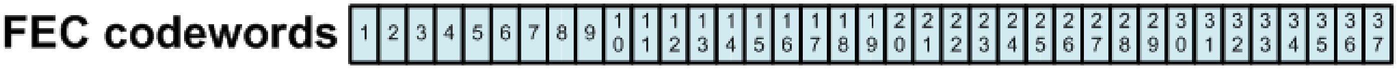
**Instances of MAC data service interface:**
MAC=interface to subordinate sublayer
MCI=interface to MAC Control multiplexer

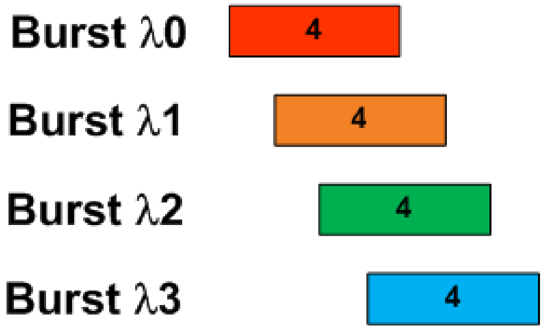
**Figure 77–14—ONU Control Multiplexer state diagram**

# Example of Fragment Start marking



# Worst case overhead



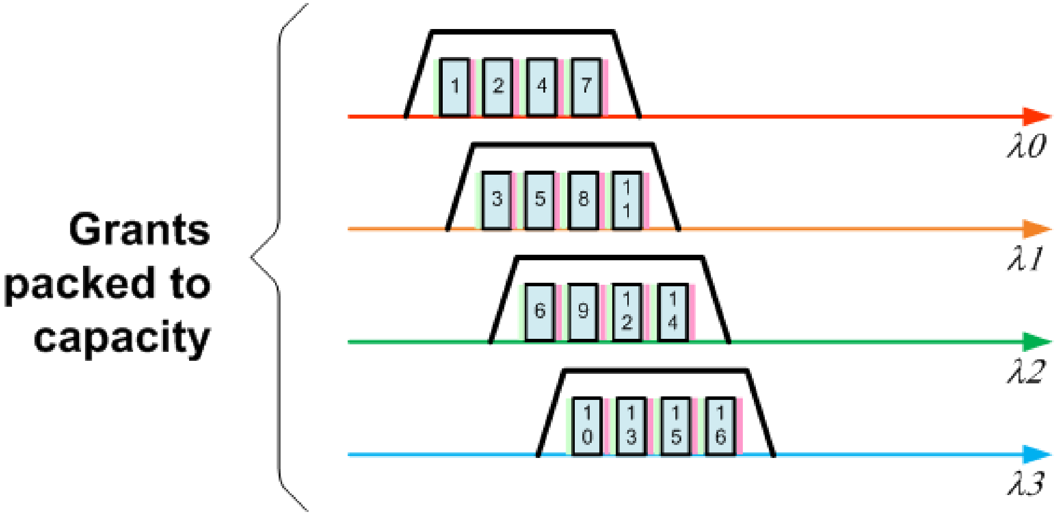


Worst Case illustration

fStart = 8B, fEnd = 8B

worst case is 16 fStart/fStop pairs

16 fStart/fStop pairs = 256B



# Keep in mind

- **Cl 46**

| | | | | | |
|---|---|---|---|---|---|
| PL11 | DATA_VALID_STATUS | 46.1.7.5.3 | Value of DATA_VALID on a lane 0 Start control character preceded by four idles or a Sequence ordered set | RS:M | Yes [ ]<br>N/A [ ] |

- **49.2.4.1 Notation conventions**
- **Control characters other than /O/, /S/ and /T/ are labeled C0 to C7. The control character for ordered set is labeled as O0 or O4 since it is only valid on the first octet of the XGMII. The control character for start is labeled as S0 or S4 for the same reason. The control character for terminate is labeled as T0 to T7.**
- **Also see 49.2.4.4 Control codes, Figure 49–7, Table 49–1.**