# "MPCP+"
# Downstream Channel Bonding State Diagrams

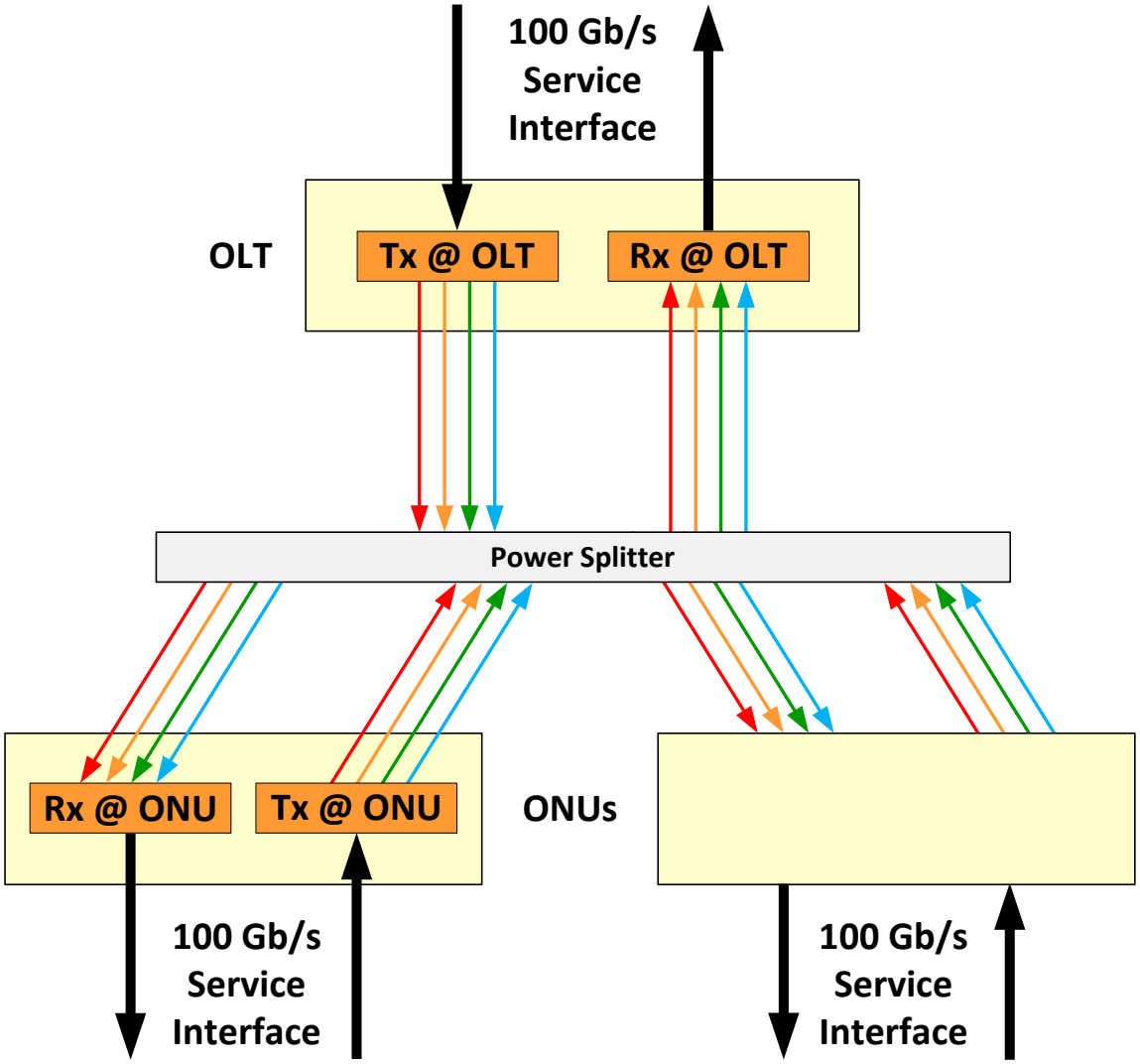
Glen Kramer, glen.kramer@broadcom.com

# Scope of this proposal

This presentation proposes state diagrams for downstream channel bonding:

1. **Tx @ OLT**
2. **Rx @ ONU**

- Upstream channel bonding is subject to additional discussion

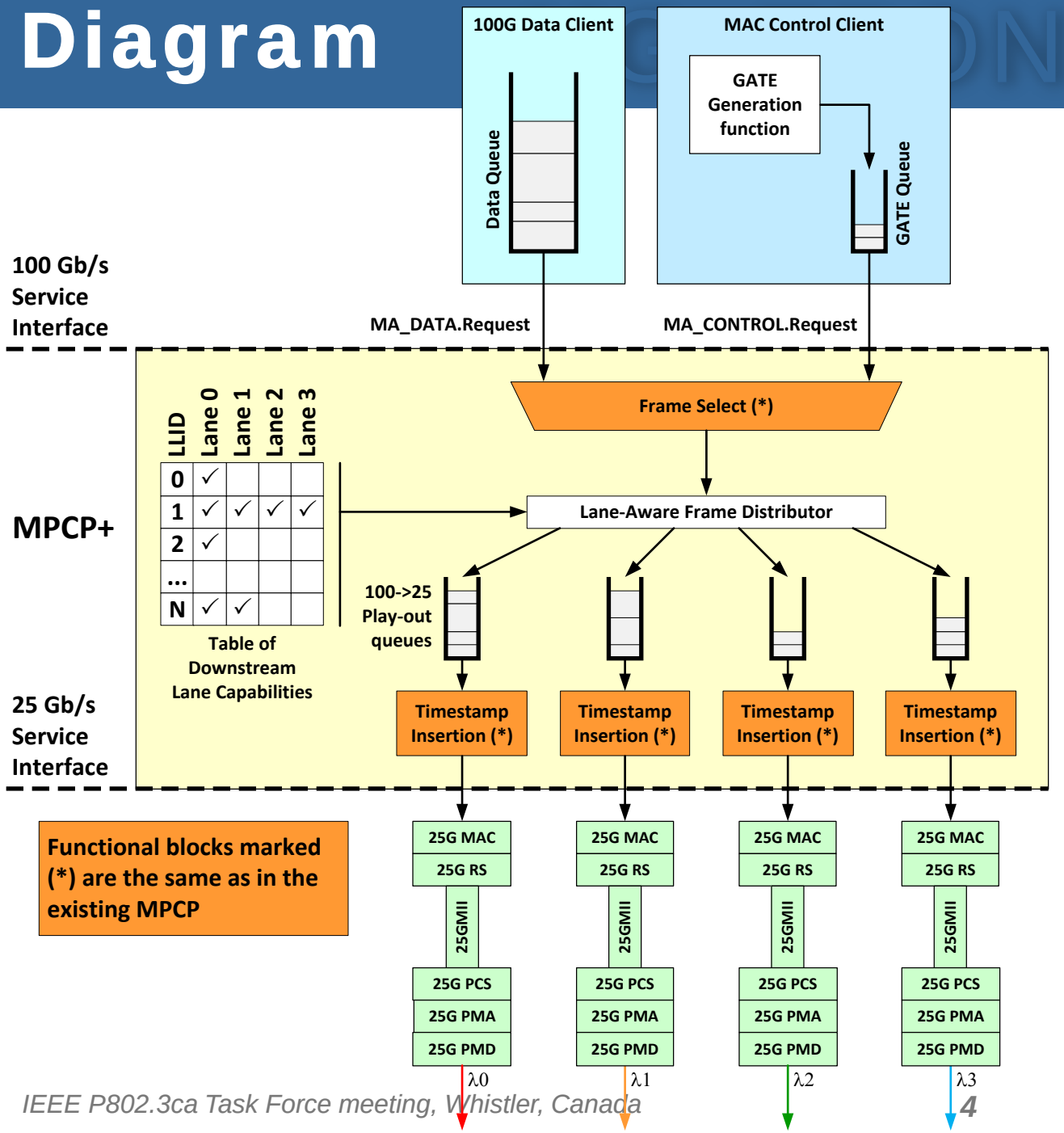


# Part 1: Transmission Process at the OLT

# OLT Block Diagram

Key additions:

- ***Table of Downstream Lane Capabilities***
- ***Lane-Aware Frame Distributor***
- ***4 downstream play-out queues*** (100G → 25G)



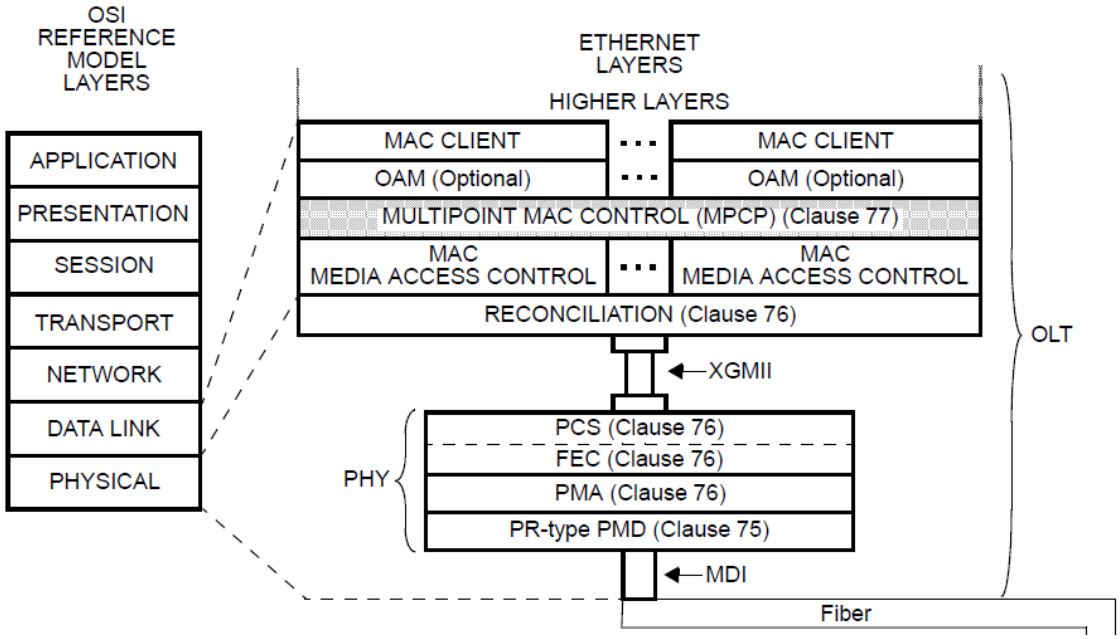
# Lane-Aware Frame Distributor (LAFD)

- **Downstream Lane Capabilities (DLC)** table reflects ONU capabilities, but can also be updated by management to turn certain lanes on and off.
- **Lane-Aware Frame Distributor (LAFD)** uses a very simple algorithm:
  1) For each DS frame ***F***, consult the DLC table and determine a set ***S*** of available DS lanes
  2) From the set ***S***, select the lane ***N*** with the earliest availability time
     - In case of a tie, select the lane with the highest index
  3) Place the frame ***F*** into the play-out queue for the selected lane ***N***
  4) Increment the availability time $T_{avail}$ for the selected lane ***N***

$$T_{avail}(N) = T_{avail}(N) + TxDuration(F)$$

# 10G-EPON OLT Layering Diagram

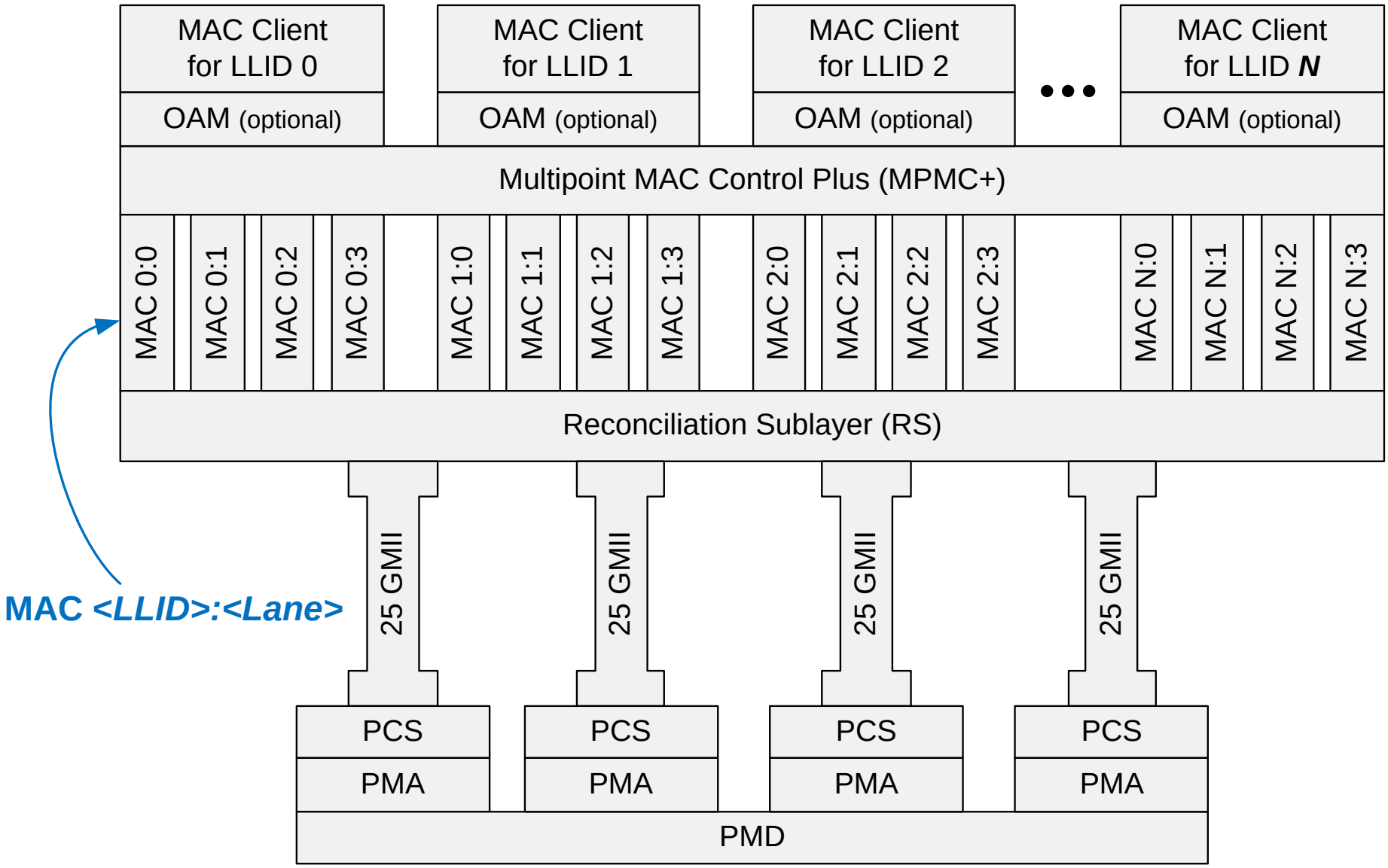
- 10G-EPON OLT has a separate MAC and a separate MAC Client for each LLID.



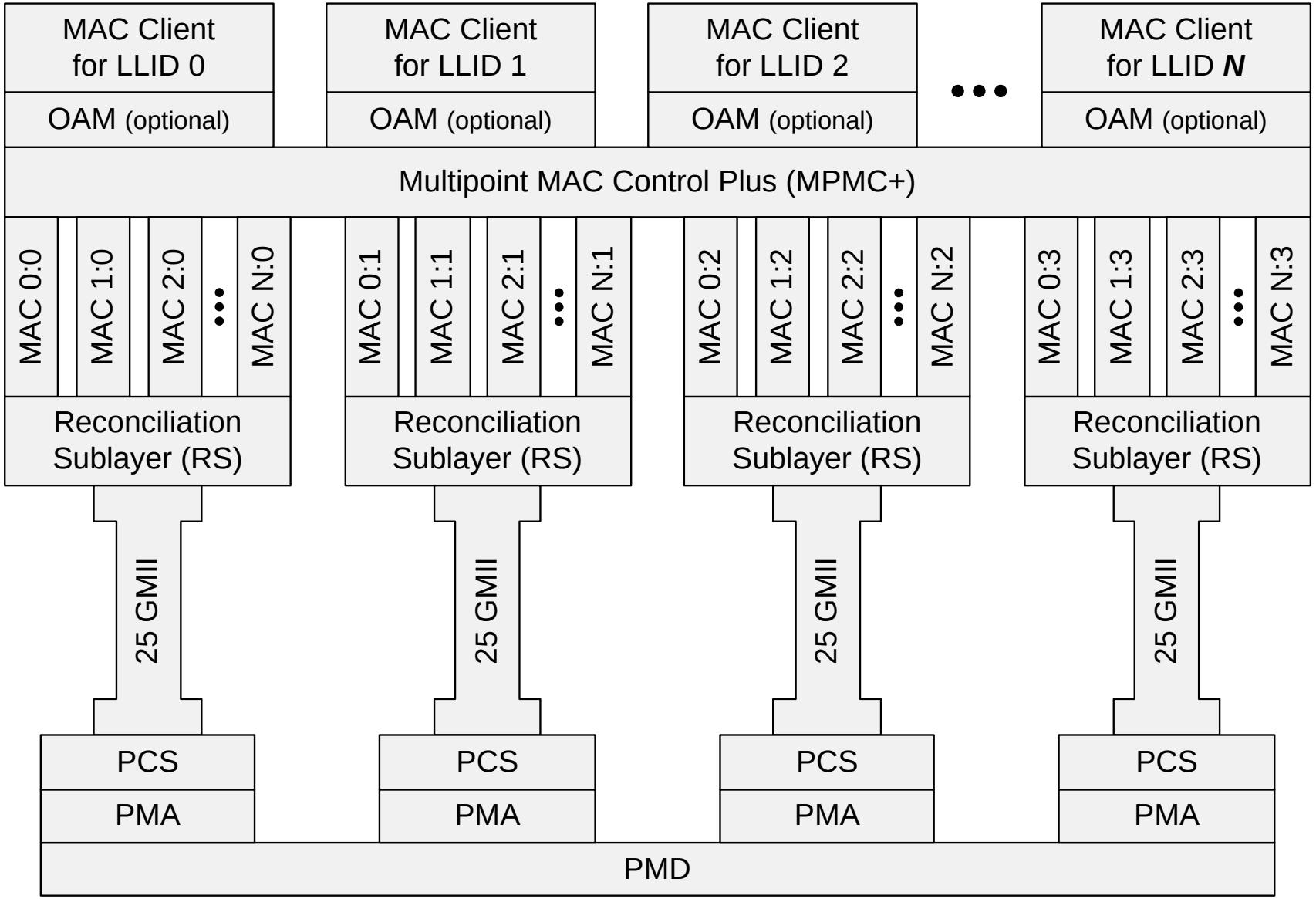

- 100G-EPON OLT needs to have a separate MAC for each LLID and also for each lane.

- Just like in 10G-EPON implementations, these are virtual MACs.

# 100G-EPON OLT Layering Diagram v.1

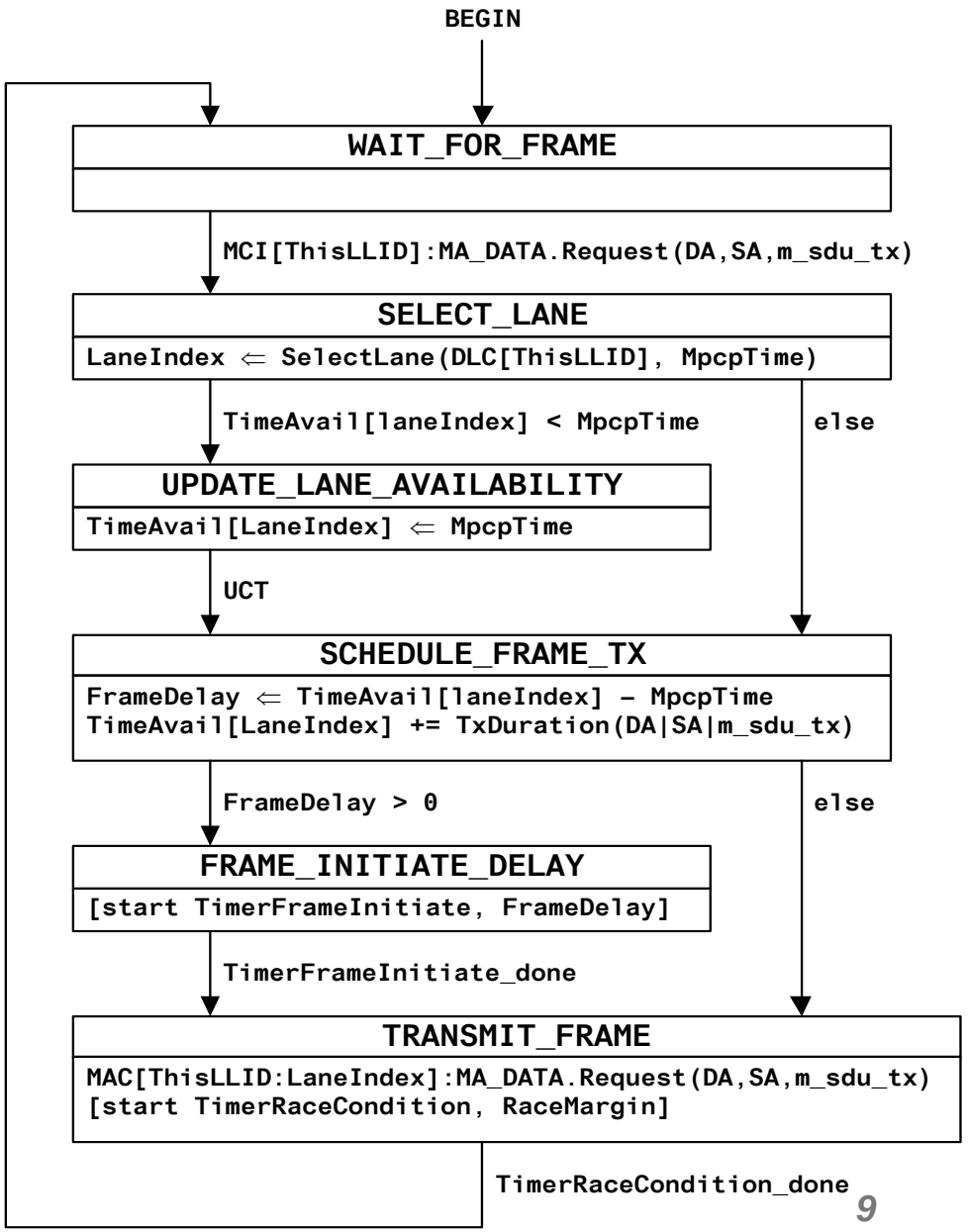


# 100G-EPON OLT Layering Diagram v.2



# LAFD State Diagram

- A separate instance of this state diagram is created for each LLID
- Constant `ThisLLID` is implicit for each state diagram




See 802.3 subclause 21.5 for state diagrams conventions

# SelectLane(…) function

**SelectLane(LaneStatus[4], currentMpcpTime)**

This function is invoked to select a transmission lane when the next frame becomes available at the MPCP Data Client. If multiple lanes are available at the time when a given frame is selected for transmission, then a lane with a higher index is selected. If no lanes are currently available, then a lane with an earliest future availability time is selected.

– Arguments:

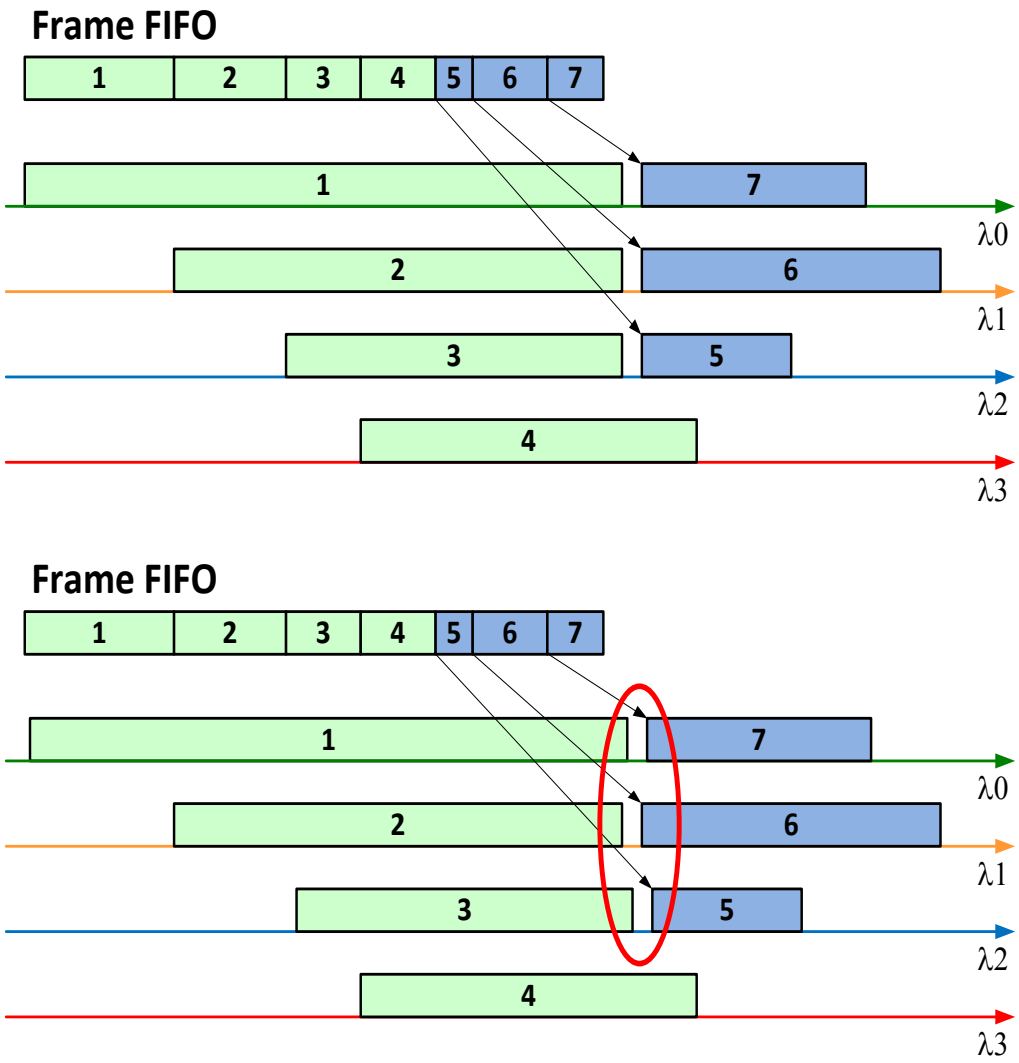
- **LaneStatus[4]** – an array indicating whether each lane is active or inactive for the given LLID. Lane status is determined at the ONU registration time and can further be modified by the NMS.
- **MpcpTime** – a value of MPCP clock at the time the next frame selected for transmission.
  - » Need to discuss the units. 16ns (50 bytes) will waste too much capacity.

# SelectLane(…) function

```
SelectLane( LaneStatus[4], currentMpcpTime )
{
    nextLane = NOT_SET;
    for( laneIndex = 3; laneIndex >= 0; laneIndex-- )
    {
        if( LaneStatus[laneIndex] == ACTIVE )
        {
            if( TimeAvail[laneIndex] <= currentMpcpTime )
                return laneIndex;
            else if( nextLane == NOT_SET OR
                    TimeAvail[nextLane] < TimeAvail[laneIndex])
                nextLane = laneIndex;
        }
    }
    return nextLane;
}
```

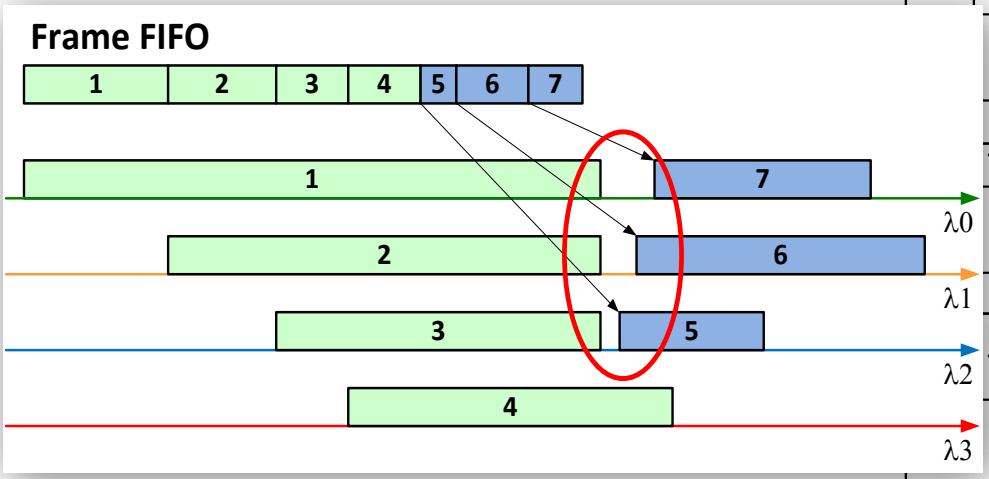
# Potential Race Condition

- Different lanes may have slightly different propagation delays.
- If transmission of two frames started close enough in time, their SFD reception order may get reversed due to a race condition.
- We need to worry about the frame transmission order only within an LLID. Relative order of frames on different LLIDs does not matter.

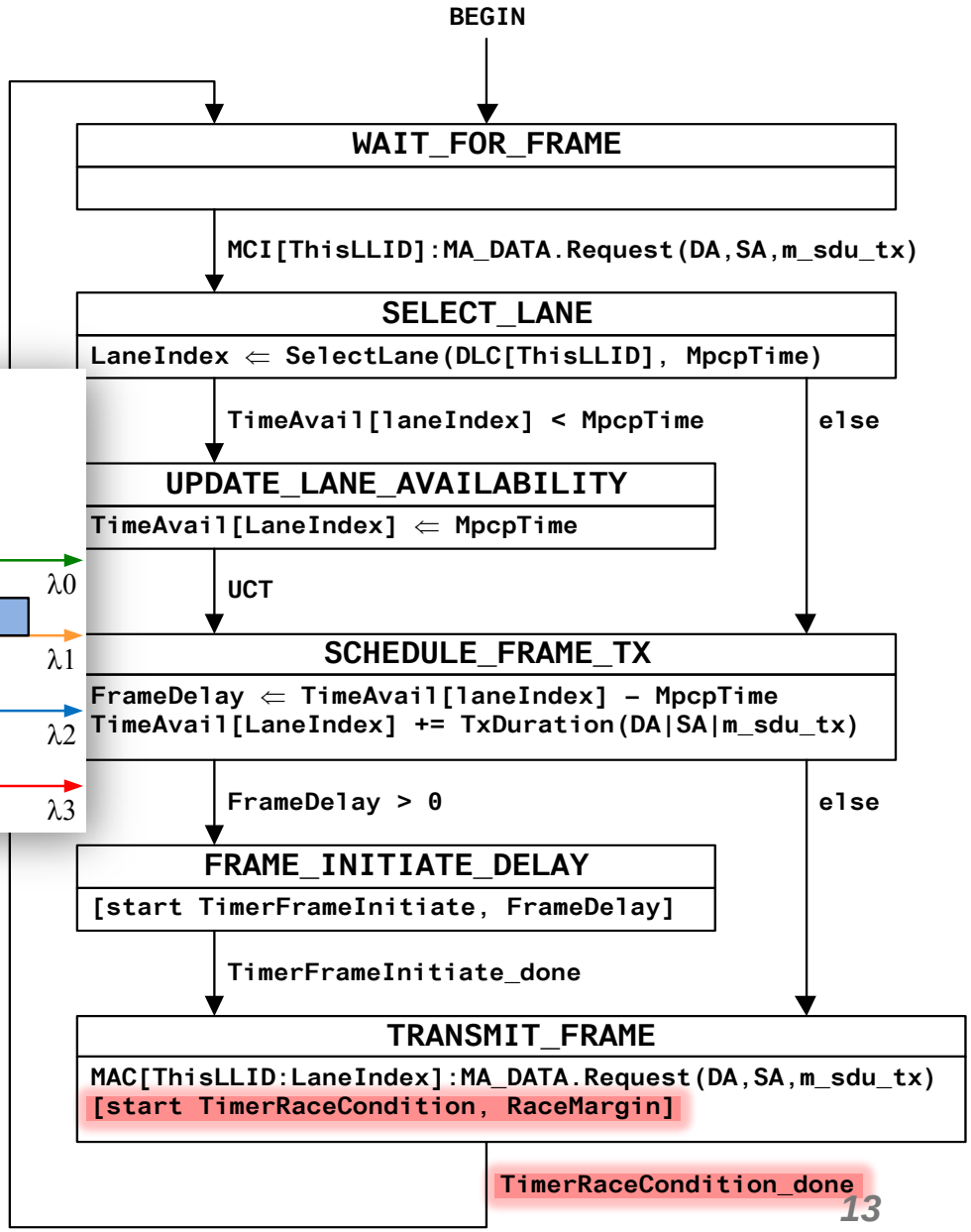


# Avoiding Race Condition

- To ensure that frame order is preserved, enforce an interval between SFDs of the frames with the same LLID.




- The interval **RaceMargin** shall exceed the variability of the propagation delay on each lane.

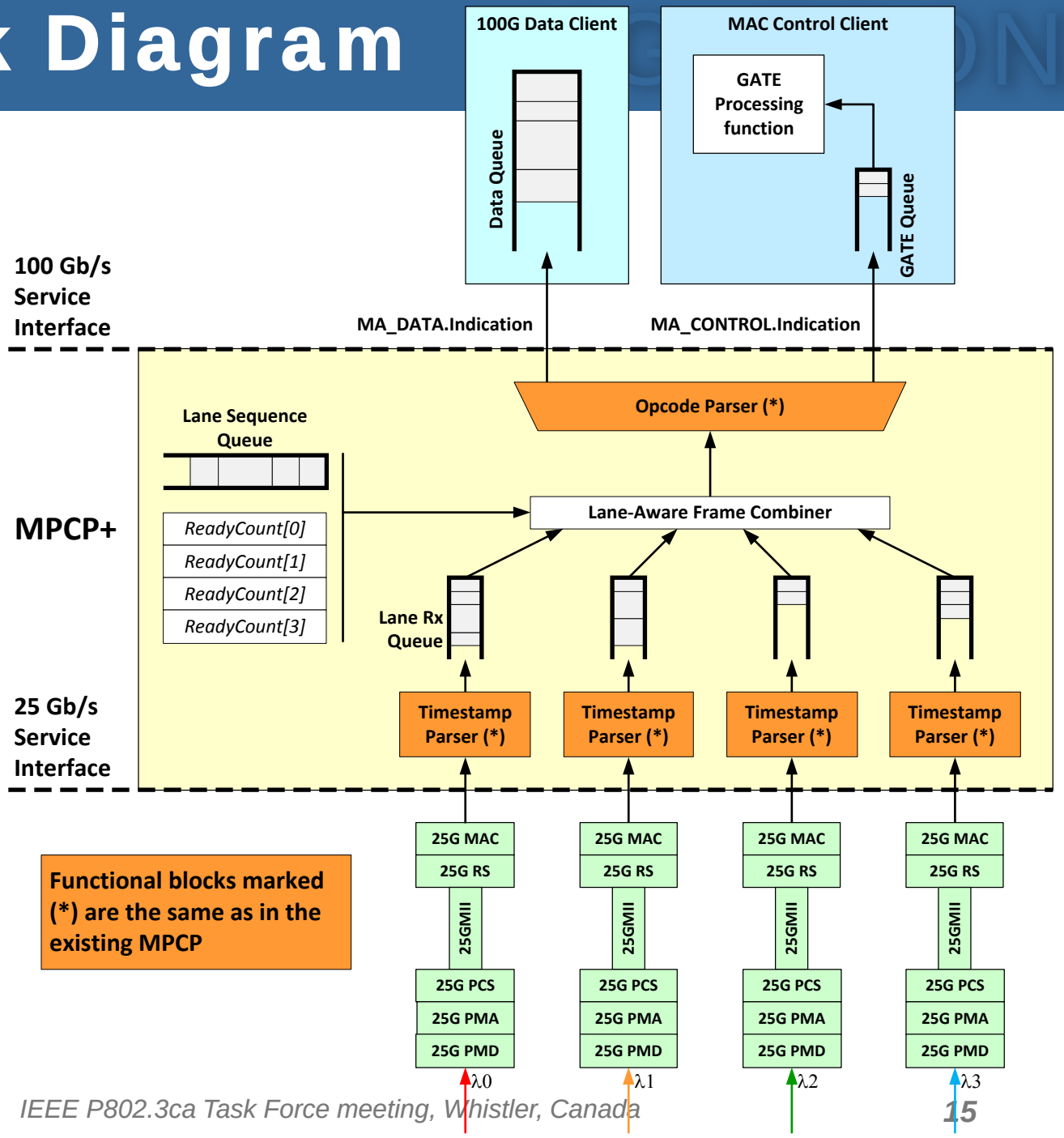


# Part 2: Reception Process at the ONU

# ONU Block Diagram

Key additions:

- ***Lane Sequence Queue***
- ***Lane-Aware Frame Combiner***
- ***4 Lane Rx Queues***
- ***4 counters ReadyCount*** *(one per lane)*



# Lane-Aware Frame Combiner (LAFC)

- **Lane-Aware Frame Combiner (LAFC)** algorithm latches the lane index into a *Lane Sequence Queue* (FIFO) at the time the first bit of the frame arrives (SFD detected).

- Each lane's queue also counts the number of frames ready to be sent (*ReadyCounter[n]*). While one queue is receiving a large frame, other queues may receive several smaller frames, which will have to wait for their transmission turn.

- If frames are taken from each lane queue in order of lane indices recorded in the *Lane Sequence Queue*, the original sequence of packets will be restored.

# LAFC Algorithm

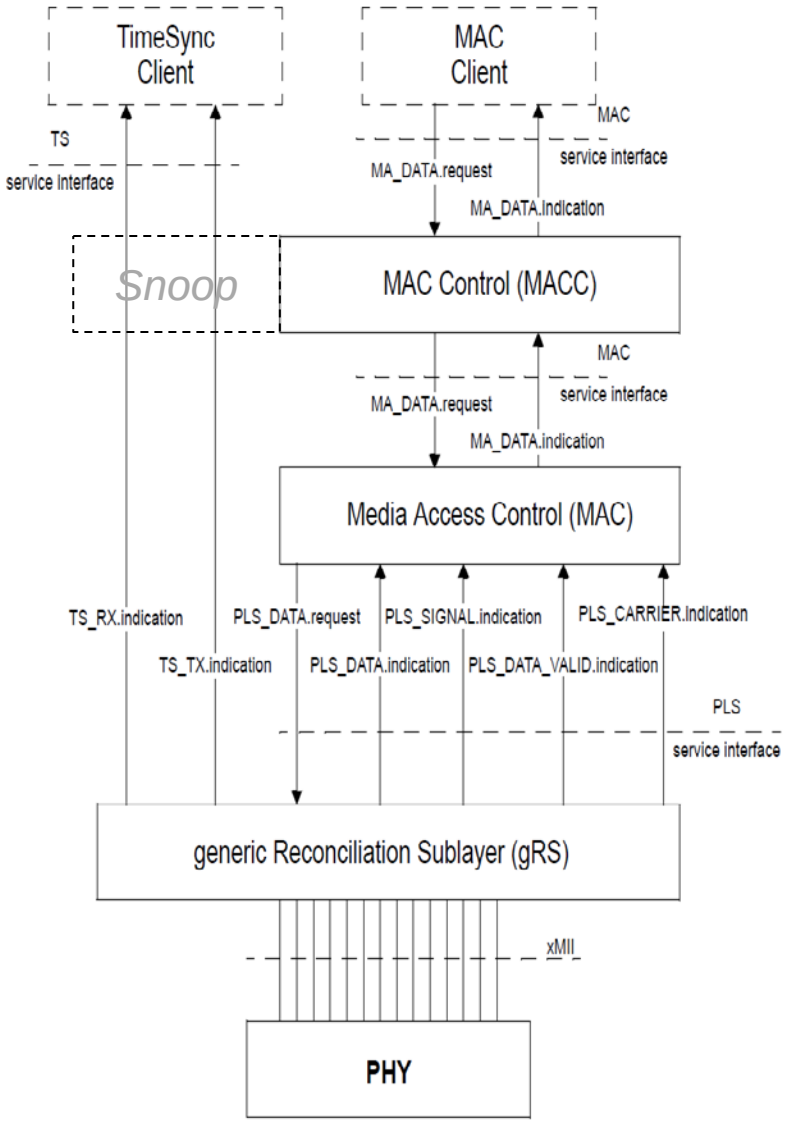


- **Input Process (Runs in parallel for each lane):**
  1) Wait for Start of Frame Delimiter (SFD) detection on lane ***N***.
  2) When SFD is detected, append value ***N*** to *Lane Sequence Queue*.
     - If two SFDs are detected simultaneously, higher lane index takes priority (see LAFD)
  3) Wait for End of Frame Delimiter (EFD) detection on lane ***N***.
  4) When EFD is detected, increment the ***ReadyCounter[N]*** by 1
  5) Go to 1.

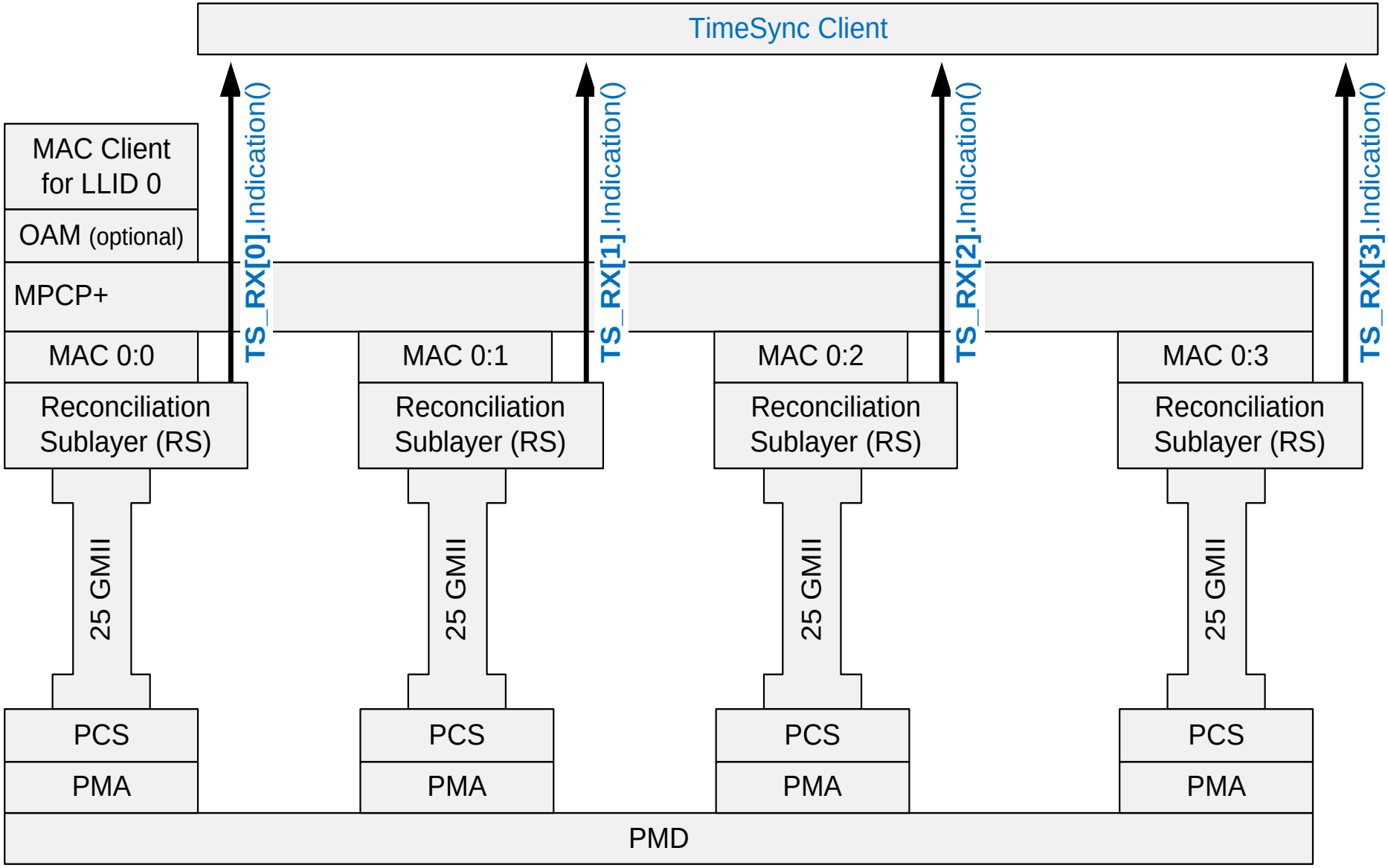
- **Output Process:**
  1) Read the value at the head of *Lane Sequence Queue* (denote the read value ***M***)
  2) Wait for ***ReadyCounter[M]*** to be > 0
  3) Transmit the complete packet from lane ***M*** queue.
  4) Decrement ***ReadyCounter[M]*** by 1
  5) Remove value ***M*** from the head of *Lane Sequence Queue*
  6) Go to 1.

# SFD detection issue

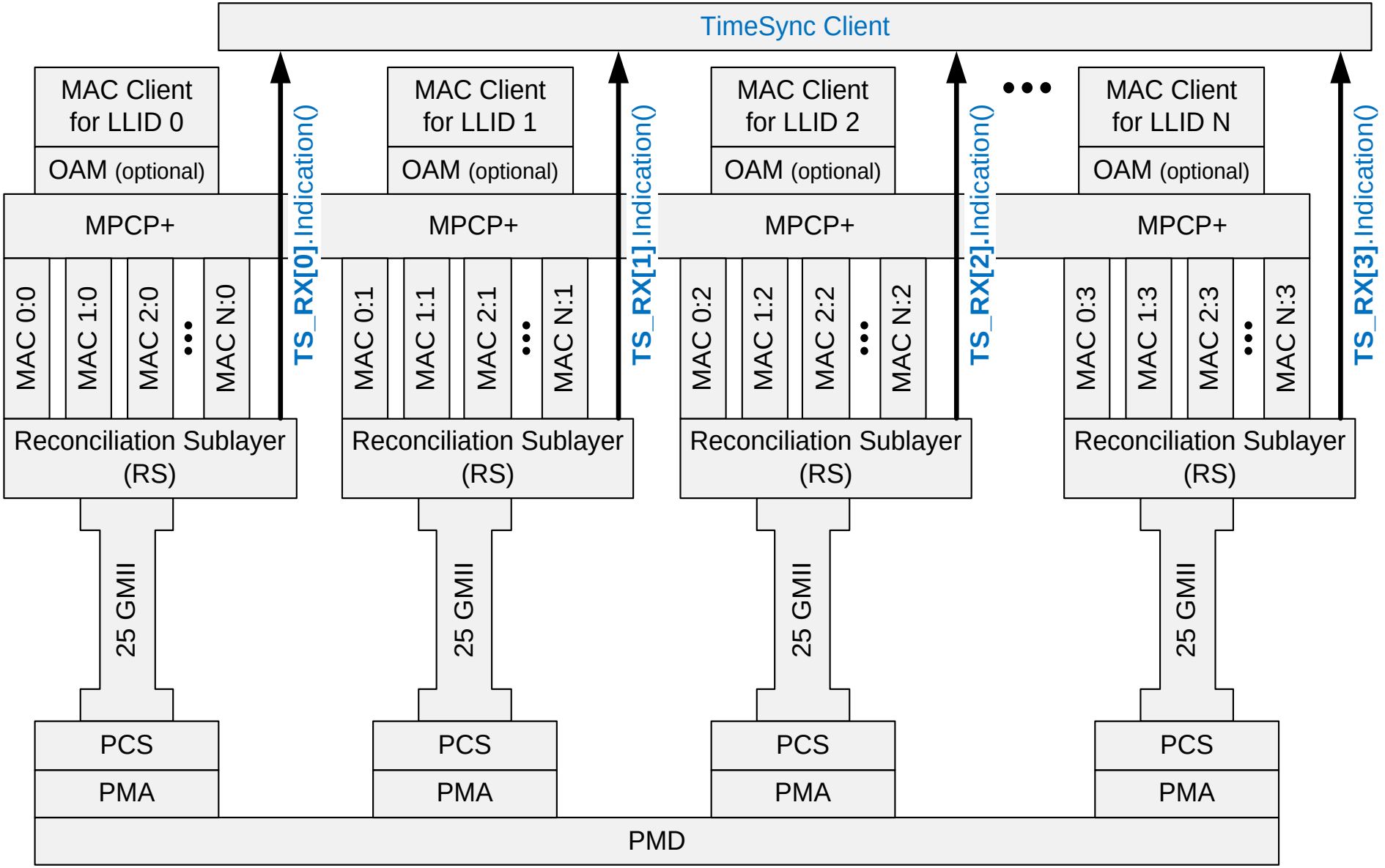
- The instantaneous transmission of `MA_DATA.Indication()` primitive is aligned with EFD detection. But the LAFC input process needs to know the time (or order) of SFD arrivals.
- **Proposal:** Utilize the TimeSync service interface (`TS_RX`)
- Normally, `TS_RX.Indication()` is generated by gRS and is received by the TimeSync client.
- MPCP+ can snoop on `TS_RX.Indication()` primitives to know when an SFD is detected



# 100G-EPON ONU with single LLID

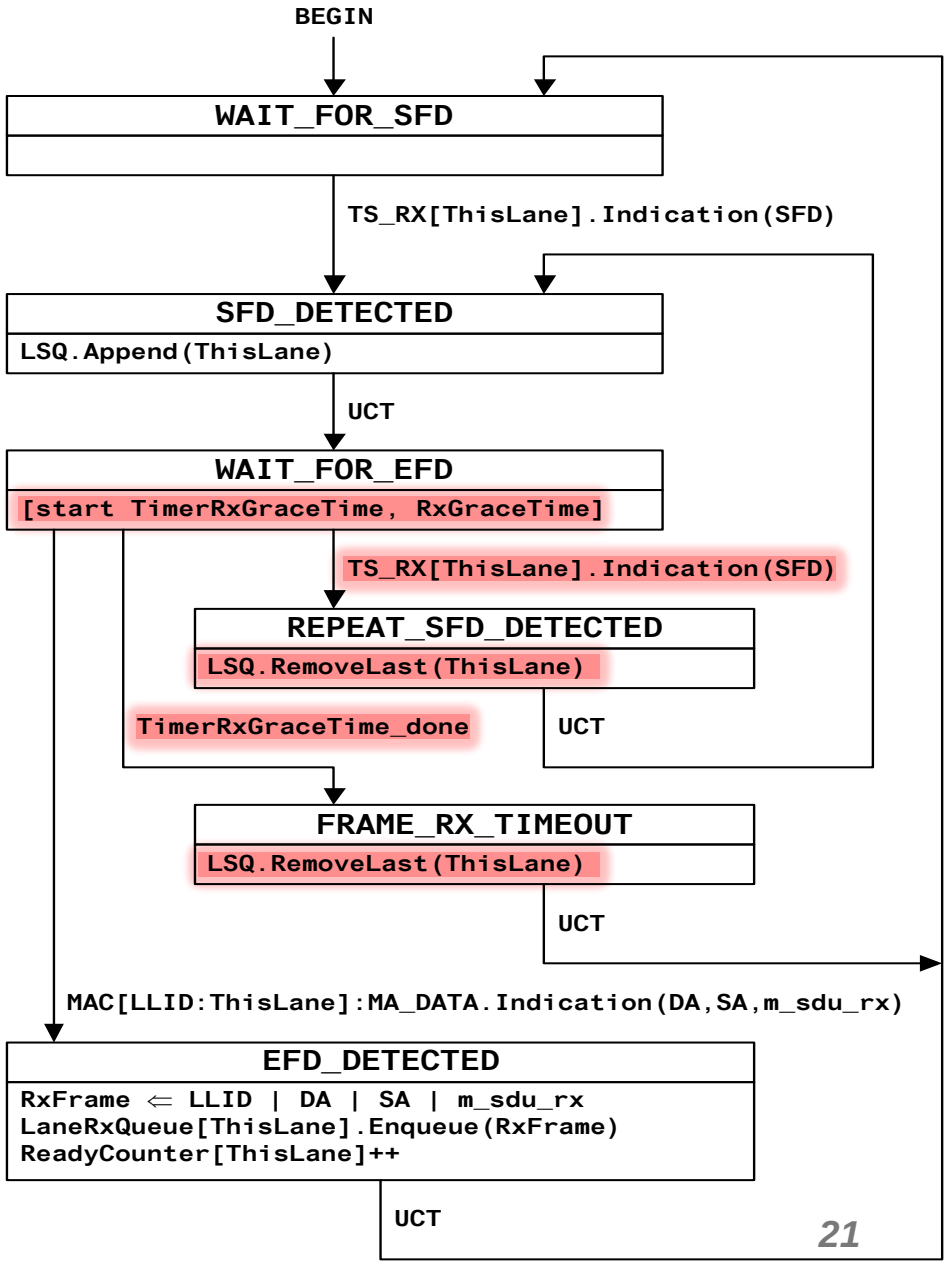


# 100G-EPON ONU with N LLIDs



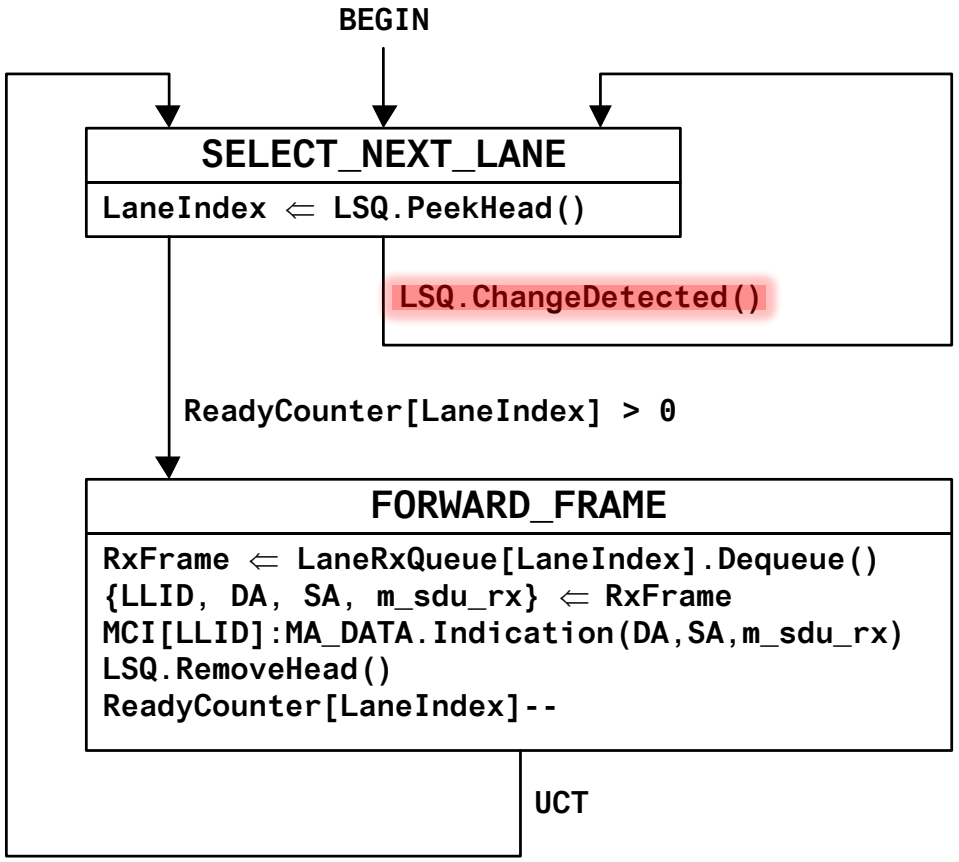
# LAFC Input Process

- A separate instance of this state diagram is created for each enabled downstream lane
- Constant **ThisLane** is implicit for each state diagram
- After reception of normal SFD, the rest of the frame may be missing or corrupted. This condition is easy to detect:
  - **Case 1**: Before an EFD, another SFD is detected on the same lane
  - **Case 2**: Timeout after the SFD reception
- The interval **RxGraceTime** shall be equal or greater than the reception time of the maximum size frame.



# LAFC Output Process

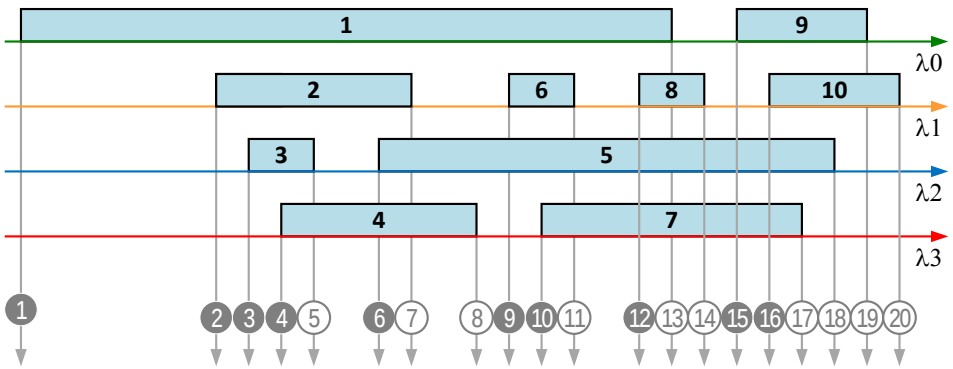
- LAFC Output Process restores the original order of frames
- When the LAFC Input Process recovers from a reception error, it removes an errored entry from the LSQ.
  When this happens (i.e., `LSQ.ChangeDetected()` returns true), the Output Process needs to re-read the first entry from the LSQ.



# Thank You

# Backup

# LAFC example




- Example of incoming data streams on 4 lanes

- Corresponding transformations of *Lane Sequence Queue* and *ReadyCount[0…3]* counters

| # | Event | Lane Sequence Queue | ReadyCount[0..3] |
|---|---|---|---|
| 1 | SoP on lane 0 | 0 | 0 0 0 0 |
| 2 | SoP on lane 1 | 0 1 | 0 0 0 0 |
| 3 | SoP on lane 2 | 0 1 2 | 0 0 0 0 |
| 4 | SoP on lane 3 | 0 1 2 3 | 0 0 0 0 |
| 5 | EoP on lane 2 | 0 1 2 3 | 0 0 1 0 |
| 6 | SoP on lane 2 | 0 1 2 3 2 | 0 0 1 0 |
| 7 | EoP on lane 1 | 0 1 2 3 2 | 0 1 1 0 |
| 8 | EoP on lane 3 | 0 1 2 3 2 | 0 1 1 1 |
| 9 | SoP on lane 1 | 0 1 2 3 2 1 | 0 1 1 1 |
| 10 | SoP on lane 3 | 0 1 2 3 2 1 3 | 0 1 1 1 |
| 11 | EoP on lane 1 | 0 1 2 3 2 1 3 | 0 2 1 1 |
| 12 | SoP on lane 1 | 0 1 2 3 2 1 3 1 | 0 2 1 1 |
| 13 | EoP on lane 0 | 0 1 2 3 2 1 3 1 | 1 2 1 1 |
| a | Tx from lane 0 | 1 2 3 2 1 3 1 | 0 2 1 1 |
| b | Tx from lane 1 | 2 3 2 1 3 1 | 0 1 1 1 |
| c | Tx from lane 2 | 3 2 1 3 1 | 0 1 0 1 |
| d | Tx from lane 3 | 2 1 3 1 | 0 1 0 0 |
| 14 | EoP on lane 1 | 2 1 3 1 | 0 2 0 0 |
| 15 | SoP on lane 0 | 2 1 3 1 0 | 0 2 0 0 |
| 16 | SoP on lane 1 | 2 1 3 1 0 1 | 0 2 0 0 |
| 17 | EoP on lane 3 | 2 1 3 1 0 1 | 0 2 0 1 |
| 18 | EoP on lane 2 | 2 1 3 1 0 1 | 0 2 1 1 |
| e | Tx from lane 2 | 1 3 1 0 1 | 0 2 0 1 |
| f | Tx from lane 1 | 3 1 0 1 | 0 1 0 1 |
| g | Tx from lane 3 | 1 0 1 | 0 1 0 0 |
| h | Tx from lane 1 | 0 1 | 0 0 0 0 |
| 19 | EoP on lane 0 | 0 1 | 1 0 0 0 |
| j | Tx from lane 0 | 1 | 0 0 0 0 |
| 20 | EoP on lane 1 | 1 | 0 1 0 0 |
| i | Tx from lane 0 | | 0 0 0 0 |