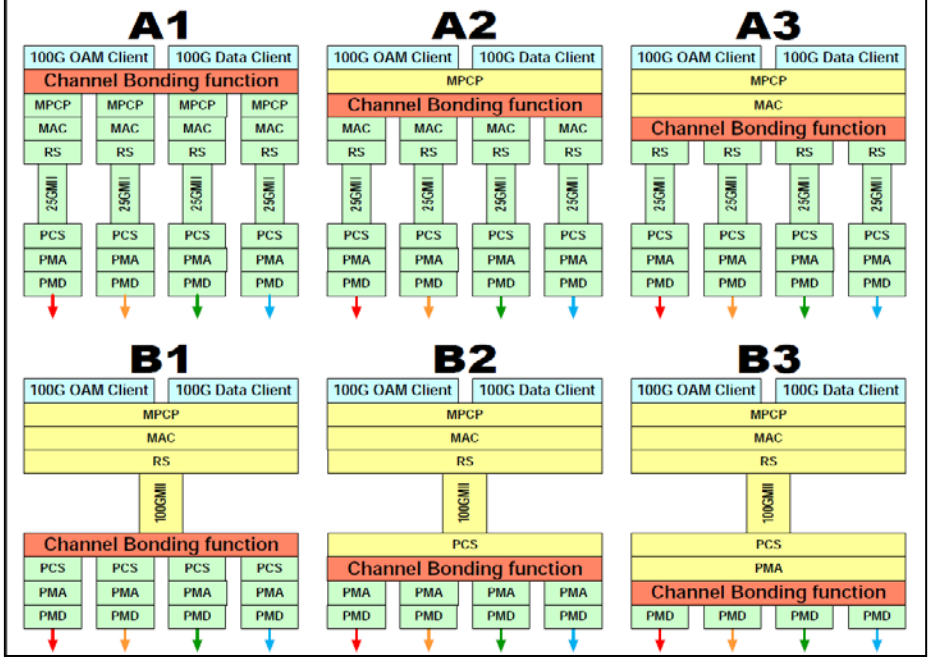
"MPCP+"
A proposal for Channel Bonding at MAC Control Sublayer

Glen Kramer, glen.kramer@broadcom.com

Where we left off

- Difficulties with placing the channel bonding sublayer were discussed in Atlanta
  - See *100G-EPON: Channel Bonding Placement Issues*, http://www.ieee802.org/3/ca/public/meeting_archive/2016/01/kramer_3ca_2b_0116.pdf

Where to put channel bonding?



No Place for Channel Bonding

Technical Issue		Bonding is above xMII			Bonding is below xMII		
		A1	A2	A3	B1	B2	B3
Issue #1	MAC runs at 100 Gb/s	No	No	Yes	Yes	Yes	Yes
Issue #2	MPCP time can be synchronized between the OLT and ONUs	Yes	No	No	No	No	No
Issue #3	ONU is able to control lasers independently for each lane	Yes	Yes	Yes	No	No	No
Issue #4	ONU is able to turn each laser on/off at correct times	Yes	No	No	No	No	No
Issue #5	ONU is able to insert frame sequence number in preamble	No	No	Yes	No	No	No
Issue #6	ONU is able to pack grants based on previously-reported packet boundaries	Yes	No	No	No	No	No

January 2016 IEEE P802.3ca Task Force meeting, Atlanta GA 14

A New Approach

- Previously, we tried to keep existing EPON sublayers as is and to fit the Channel Bonding sublayer anywhere in between. That didn't work.

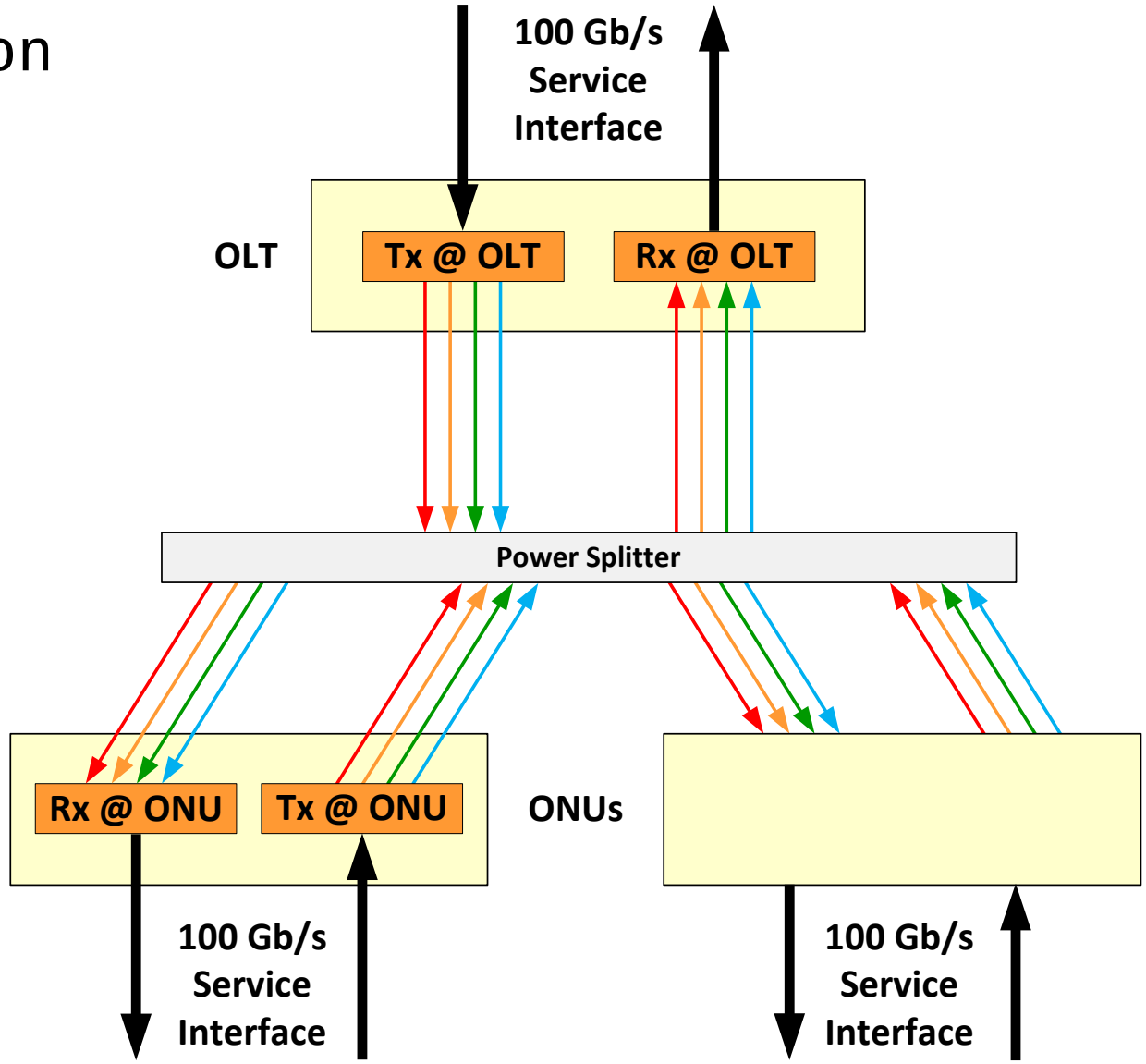
- Proposed new approach:
  - Break MPCP into key functional blocks
  - Channel Bonding is just another functional block
  - Some of existing MPCP blocks need to be above Channel Bonding block, some need to be below

- New MPCP → MPCP+

Scope of work

To define operation of MPCP+, all we need to do is to figure out these four parts:

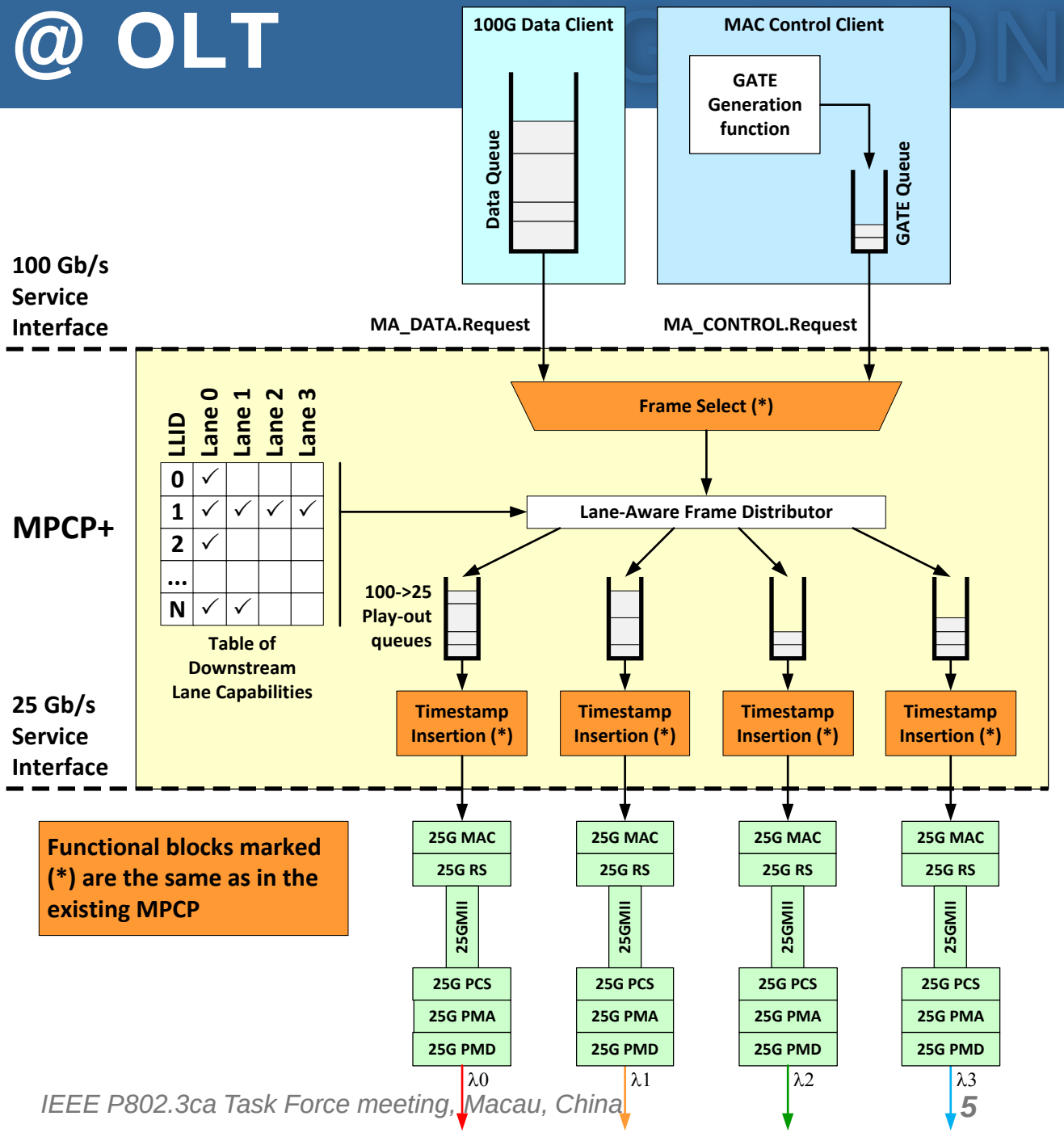
- Downstream:
  1. Tx @ OLT
  2. Rx @ ONU
- Upstream
  3. Tx @ ONU
  4. Rx @ OLT



Part 1: Tx @ OLT

Key additions:

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

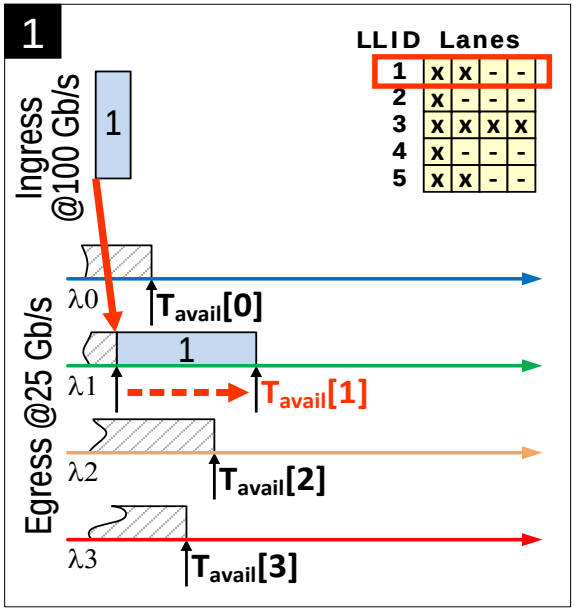


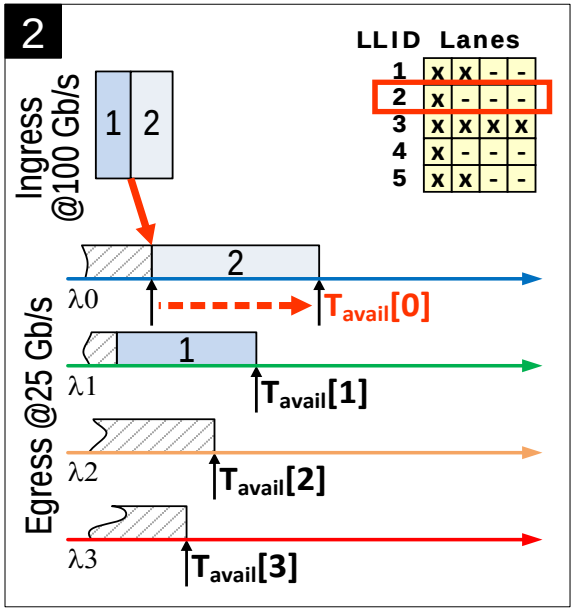
Lane-Aware Frame Distributor (LAFD)

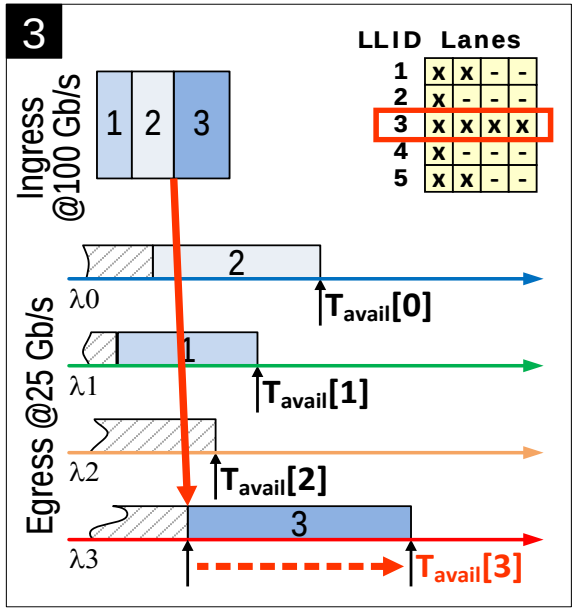
- **Downstream Lane Capabilities (DLC)** table reflects ONU capabilities, but can also be updated by management to turn certain lanes on and off.
- **Lane-Aware Frame Distributor (LAFD)** uses a very simple algorithm:
  1) For each DS frame ***F***, consult the DLC table and determine a set ***S*** of 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***

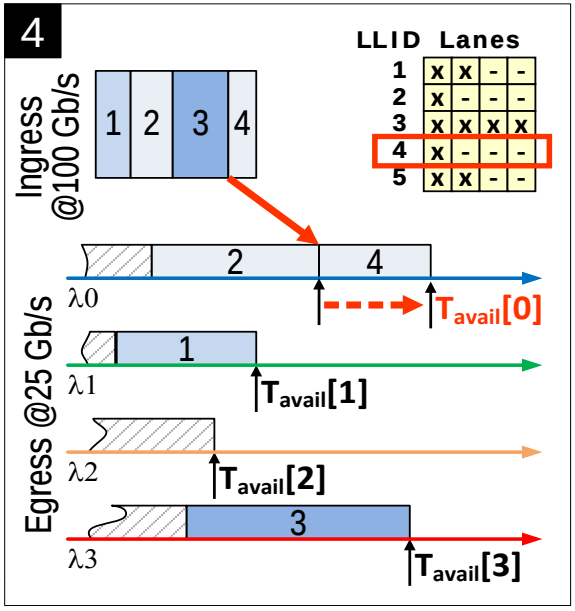
$$T_{avail}(N) = T_{avail}(N) + tx\_duration(F)$$

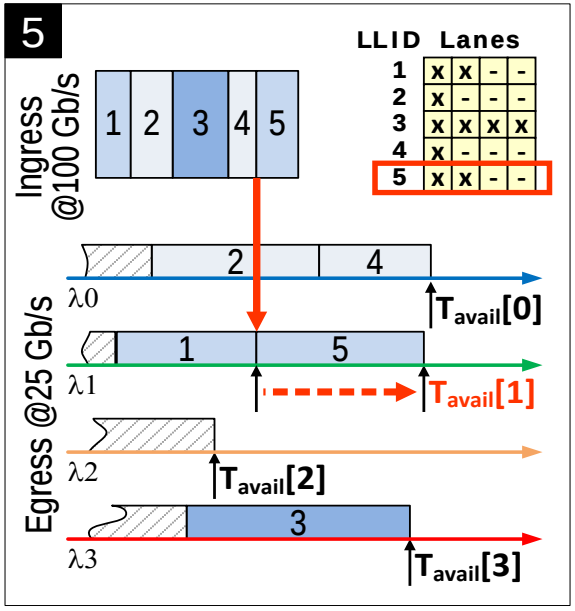
LAFD Illustration

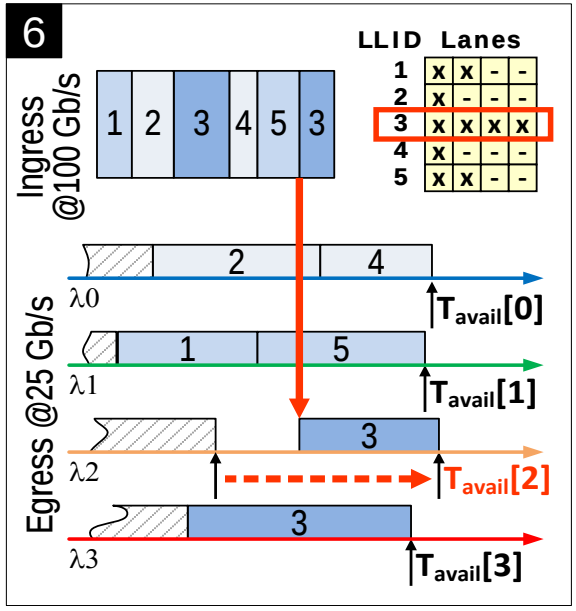




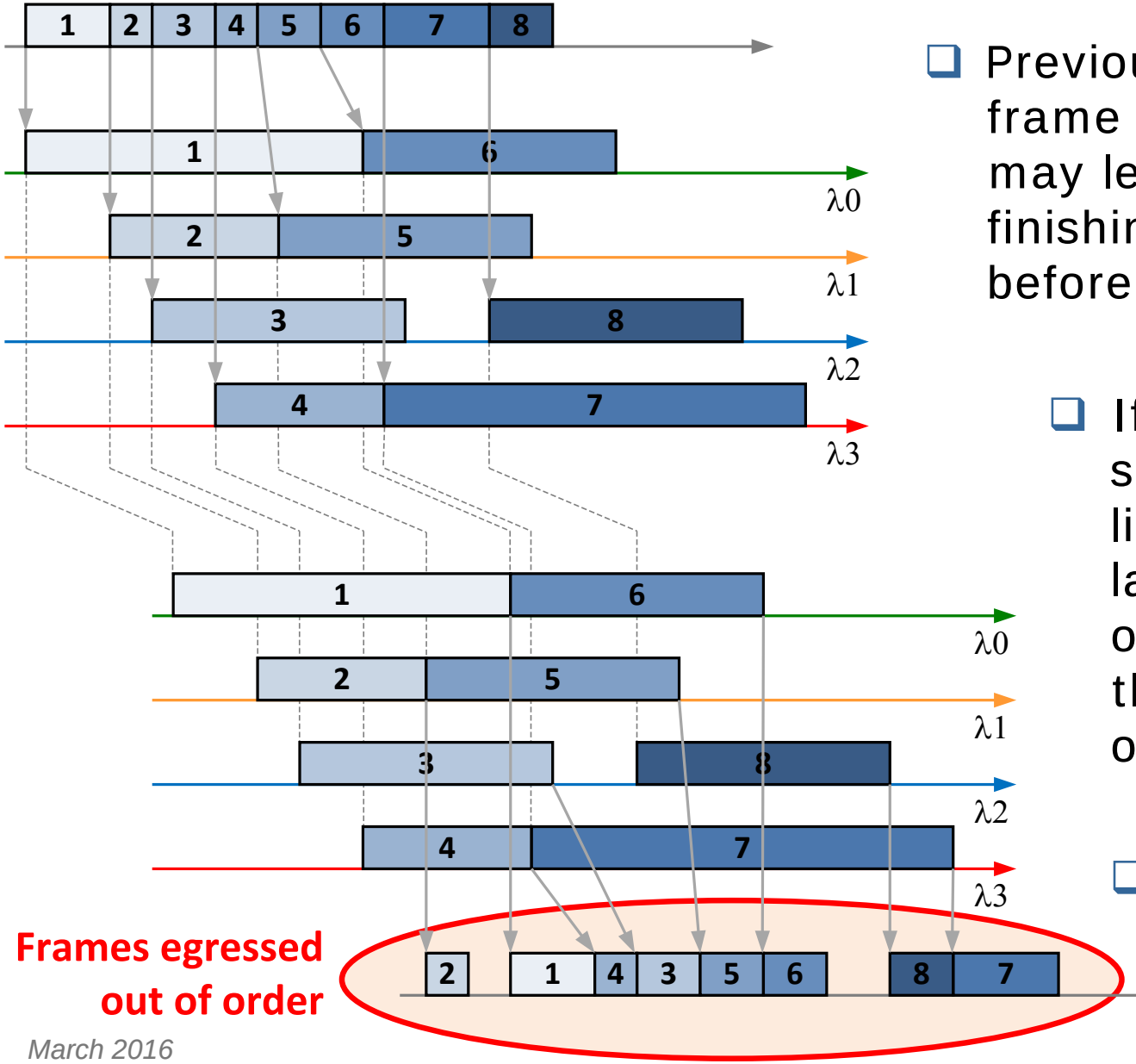








Frame reordering problem

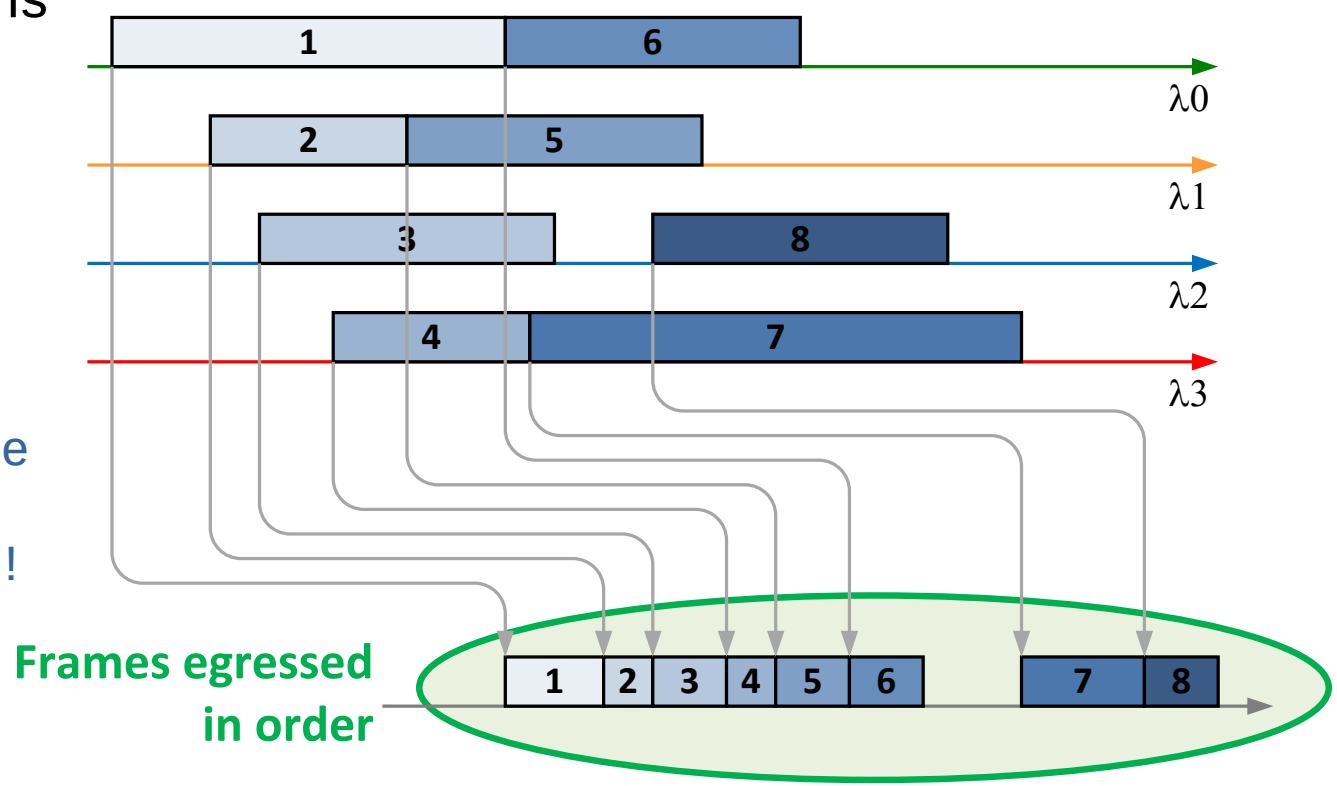


- Previously, we saw that frame demultiplexing may lead to later frames finishing transmission before earlier frames.
- If the frames are sent out on 100Gb/s link as soon as the last bit is received on a 25Gb/s link, they will be sent out of order.
- How to restore the original frame sequence?

How to recover frame sequence

- One method is to add a frame sequence number to each frame and use this number to reorder the frame at the receiving end.
  - Where to put the sequence number?
  - Preamble does not have enough unused bits

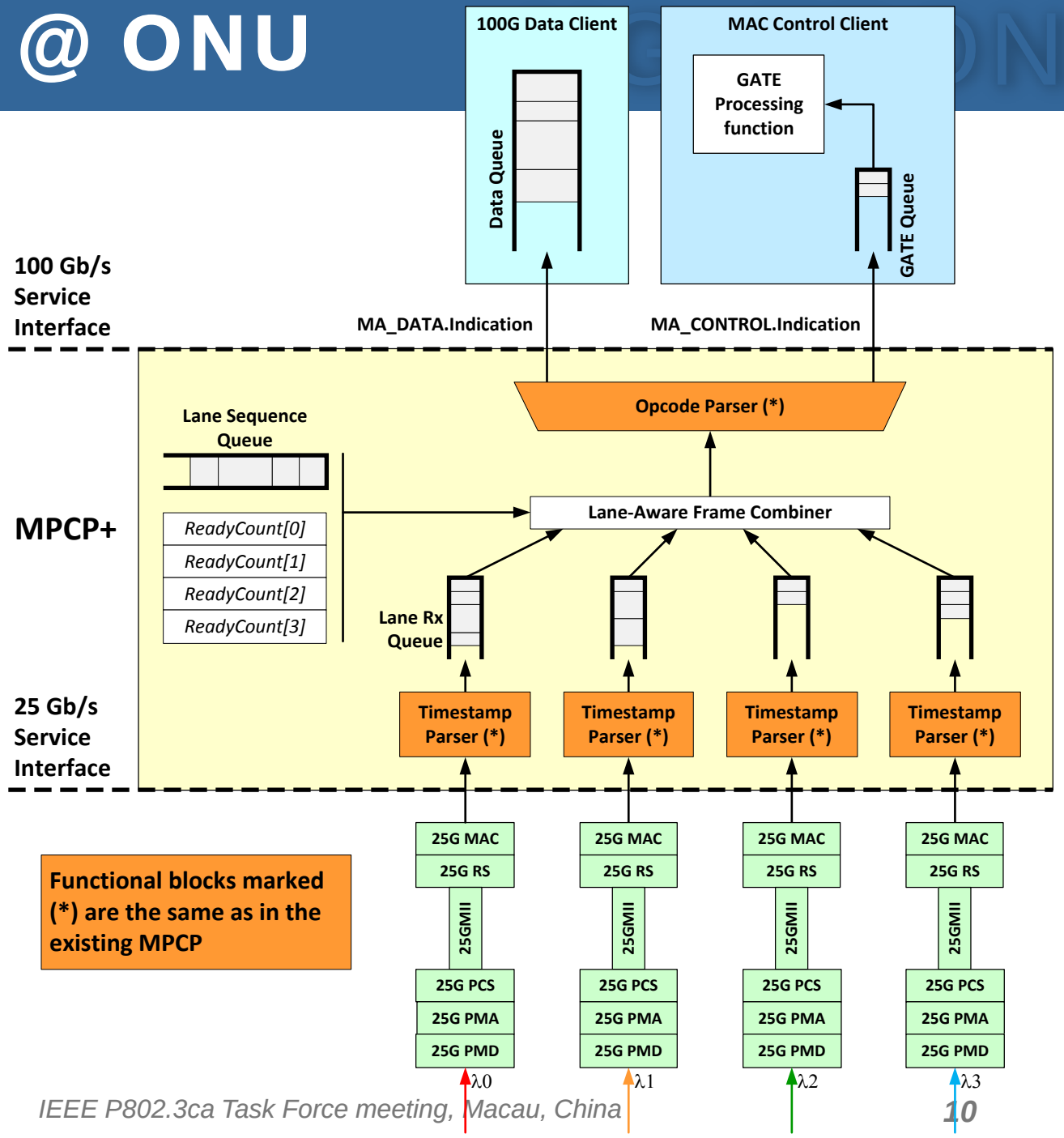
- A better method is to order frames not by their last bit Rx times, but by their <u>first bit Rx times.</u>
  - LAFD algorithm ensures that the first bits are always in order!



Part 2: Rx @ ONU

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 (Start-of-Packet 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-Packet (SoP) detection on lane ***N***.
  2) When SoP is detected, append value ***N*** to *Lane Sequence Queue*.
     - If two SoPs are detected simultaneously, higher lane index takes priority (see LAFD)
  3) Wait for End-of-Packet (EoP) detection on lane ***N***.
  4) When EoP is detected, increment the ***ReadyCounter[N]*** by 1
  5) Go to 1.

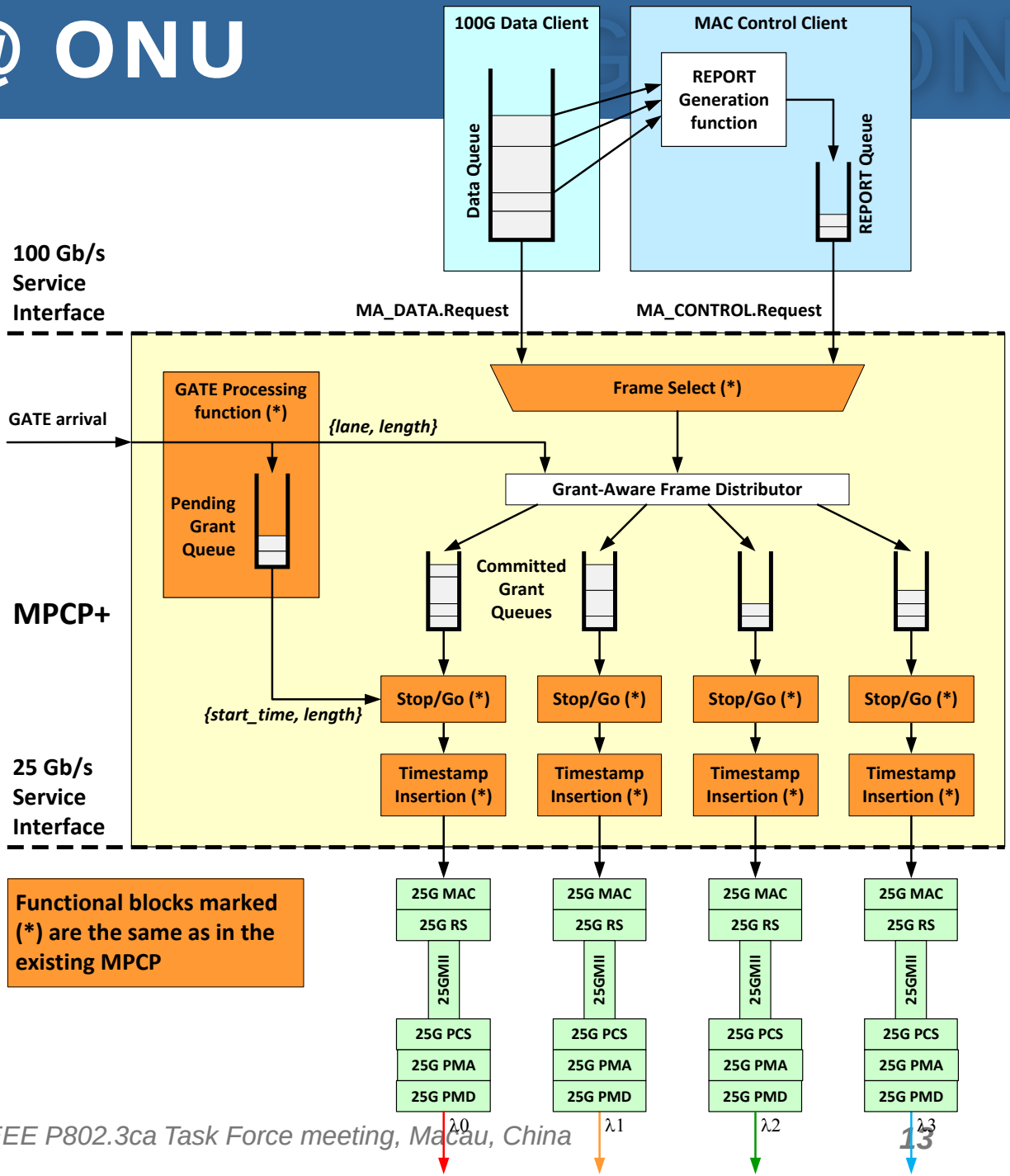
- **Output Process:**
  1) Read the value at the head of *Lane Sequence Queue* (denote the read value ***M***)
  2) Wait for ***ReadyCounter[M]*** to be > 0
  3) Transmit the complete packet from lane ***M*** queue.
  4) Decrement ***ReadyCounter[M]*** by 1
  5) Remove value ***M*** from the head of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

For simplicity, handling of error conditions, like missing SoP or EoP, is not shown here. But there is nothing magical about it.

Part 3: Tx @ ONU

Key additions:

- ***Grant-Aware Frame Distributor***
- ***4 upstream Committed Grant Queues (100G → 25G)***

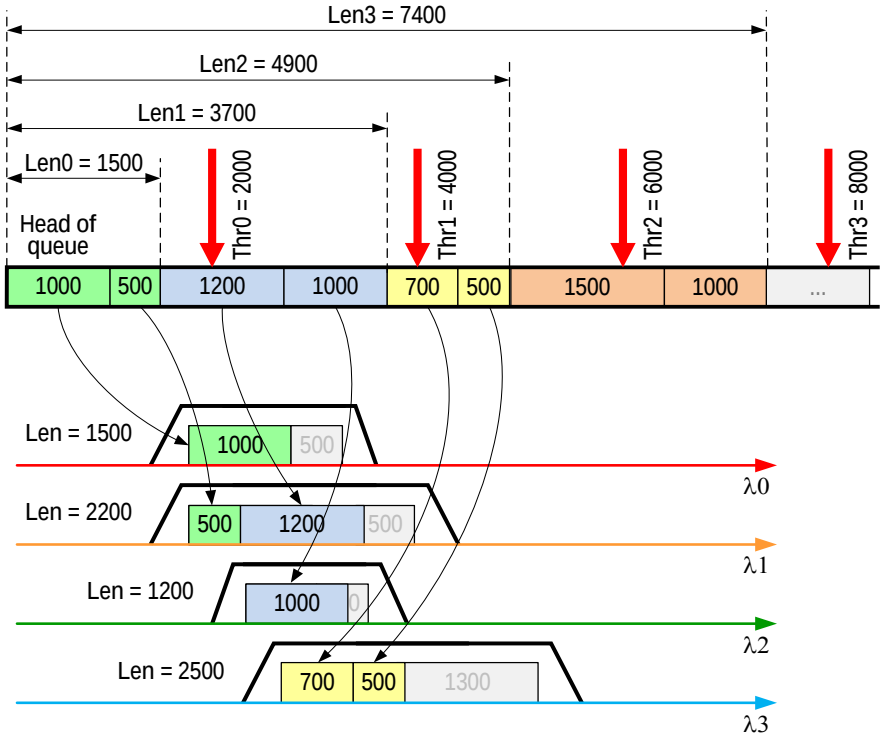


Grant-Aware Frame Distributor (GAFD)

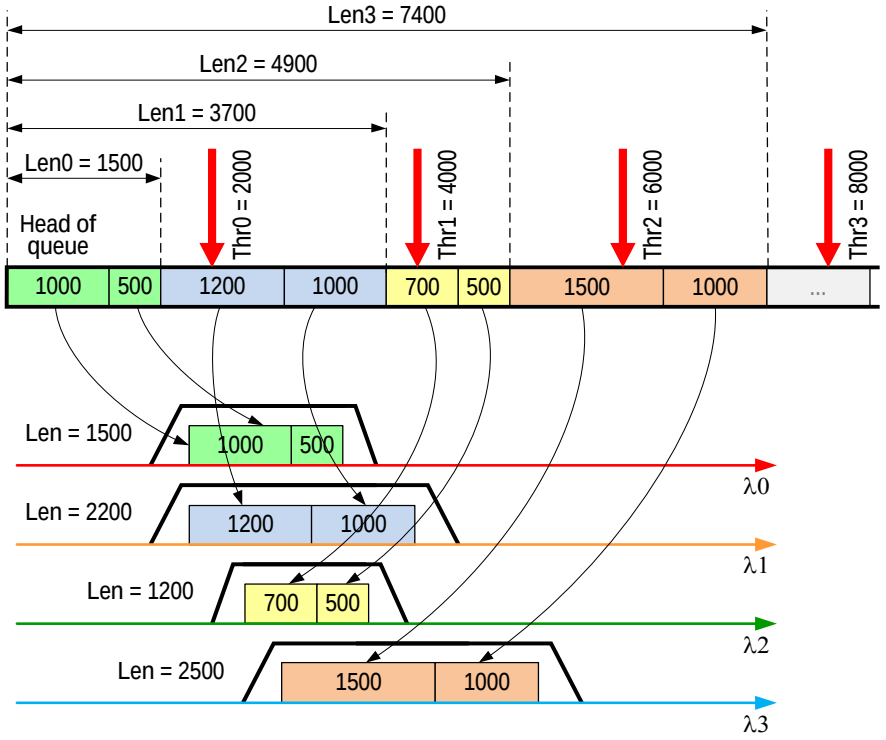
- **Grants have the following syntax:**
  - **Grant: <*LaneIndex*, *StartTime*, *Length*>**
- **Grant-Aware Frame Distributor (GAFD)** also uses a very simple algorithm:
  1) When a grant of length ***L*** for lane ***N*** arrives, admit waiting frames up to a total size of ***L*** bytes into the Committed Grant Queue for lane ***N***
  2) Do not process another grant (i.e., do not start filling another Committed Grant Queue) until either the current grant is full (i.e., it got ***L*** bytes of data queued) or there are no more waiting frames in MAC Client.
  3) Once the commitment for the next grant has started, do not come back to top of an earlier-committed grant.

Grant Commitment

- To achieve peak rates > 25Gb/s, the ONU must receive simultaneous (overlapping) grants on multiple wavelengths
- If the ONU sends the queued frames in FIFO order across all active grants, the reported frame boundaries are lost, grants are left under-filled, and significant amount of bandwidth is wasted.
- **Committing frames to grants solves this problem**



- **Frames are sent in strict FIFO order**
- **Grants are left under-utilized**

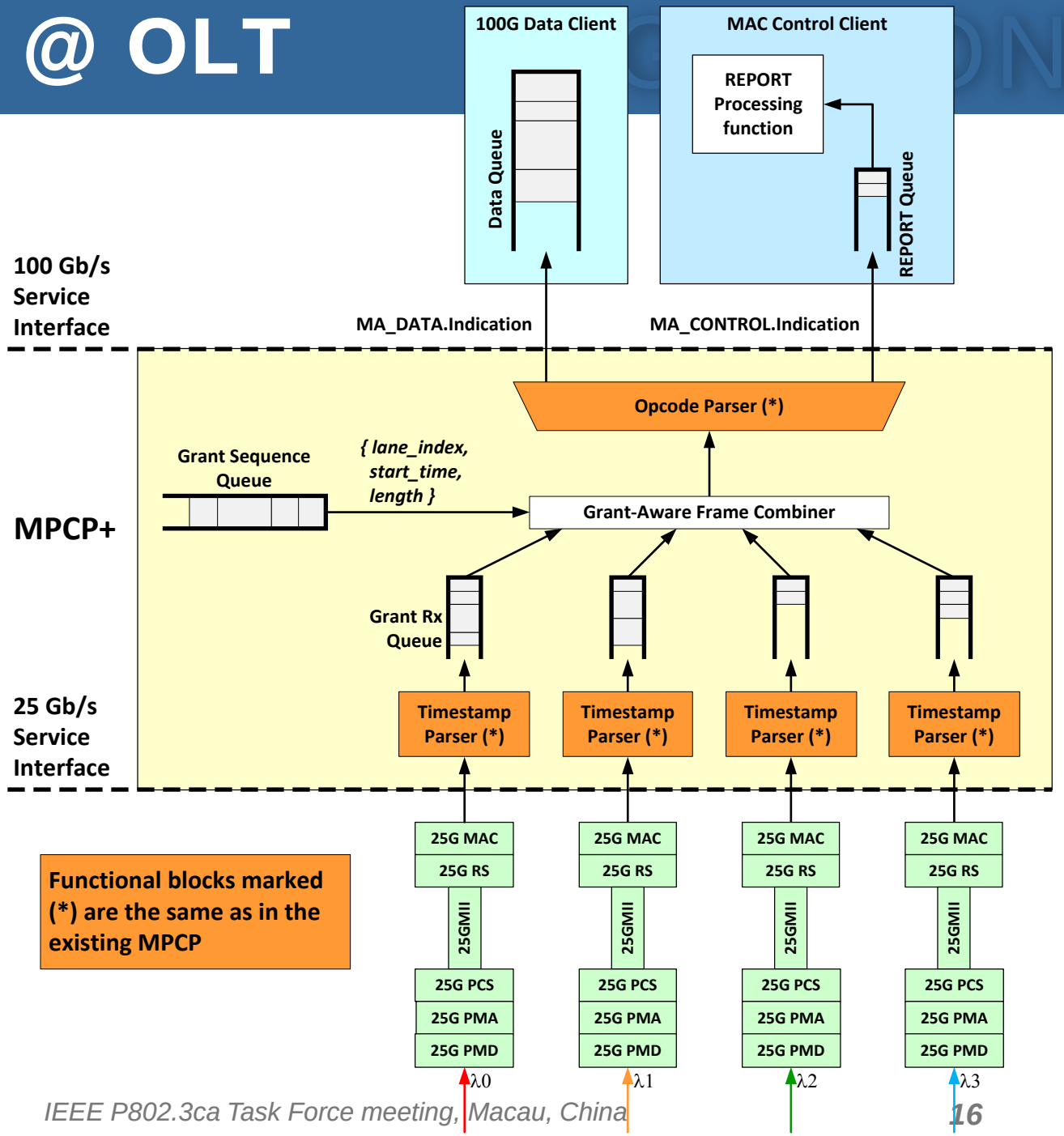


- **Frames are committed to proper grants**
- **Bandwidth loss is eliminated**

Part 4: Rx @ OLT

Key additions:

- ***Grant-Aware Frame Combiner***
- ***4 upstream Grant Rx Queues (25G → 100G)***
- ***Grant Sequence Queue***



Functional blocks marked (*) are the same as in the existing MPCP

Grant-Aware Frame Combiner (GAFC)

- Upon issuing grants to the ONU, the OLT also stores the grants in the ***Grant Sequence Queue***
- Frames arriving to the OLT are stored in Grant Rx Queues (separate queue per lane).
- Received frames are mapped to the grants stored in in the *Grant Sequence Queue*
- Operation of the ***Grant-Aware Frame Combiner (GAFC)*** algorithm:

1) Read the grant info at the head of *Grant Sequence Queue* (denote the read value ***G*** = <*Lane:**N***, *StartTime:**S***, *Length:**L***>)
2) If the frame at the head of *Grant Rx Queue* of lane ***N*** maps to grant ***G***,
   a) Dequeue the head-of-queue frame and pass it up to next MPCP block
   b) Go to 2.
3) Else //the frame maps to a different grant or no more frames are queued
   a) Remove element ***G*** from the head of *Grant Sequence Queue*
   b) Go to 1.

MPCP+ Score Card

Issue #1	MAC runs at 100 Gb/s	Each of four MACs runs at 25 Gb/s, but MPCP+ provides 100 Gb/s MAC Service Interface to higher layers.
Issue #2	MPCP time can be synchronized between the OLT and ONUs	Yes. In the OLT and ONUs, the timestamp insertion/parsing point is below the lane queues that introduce delay variability.
Issue #3	ONU is able to control lasers independently for each lane	Yes. Four independent 25GMIIs allow independent operation of four data detectors.
Issue #4	ONU is able to turn each laser on/off at correct times	Yes. MPCP+ starts and stops data flow for each lane at exact times.
Issue #5	ONU is able to insert frame sequence number in preamble	MPCP+ recovers the correct frame order without frame sequence number
Issue #6	ONU is able to pack grants based on previously-reported packet boundaries	Yes. Grants are packed without wastage if the order and lengths of grants matches the reported thresholds

Summary

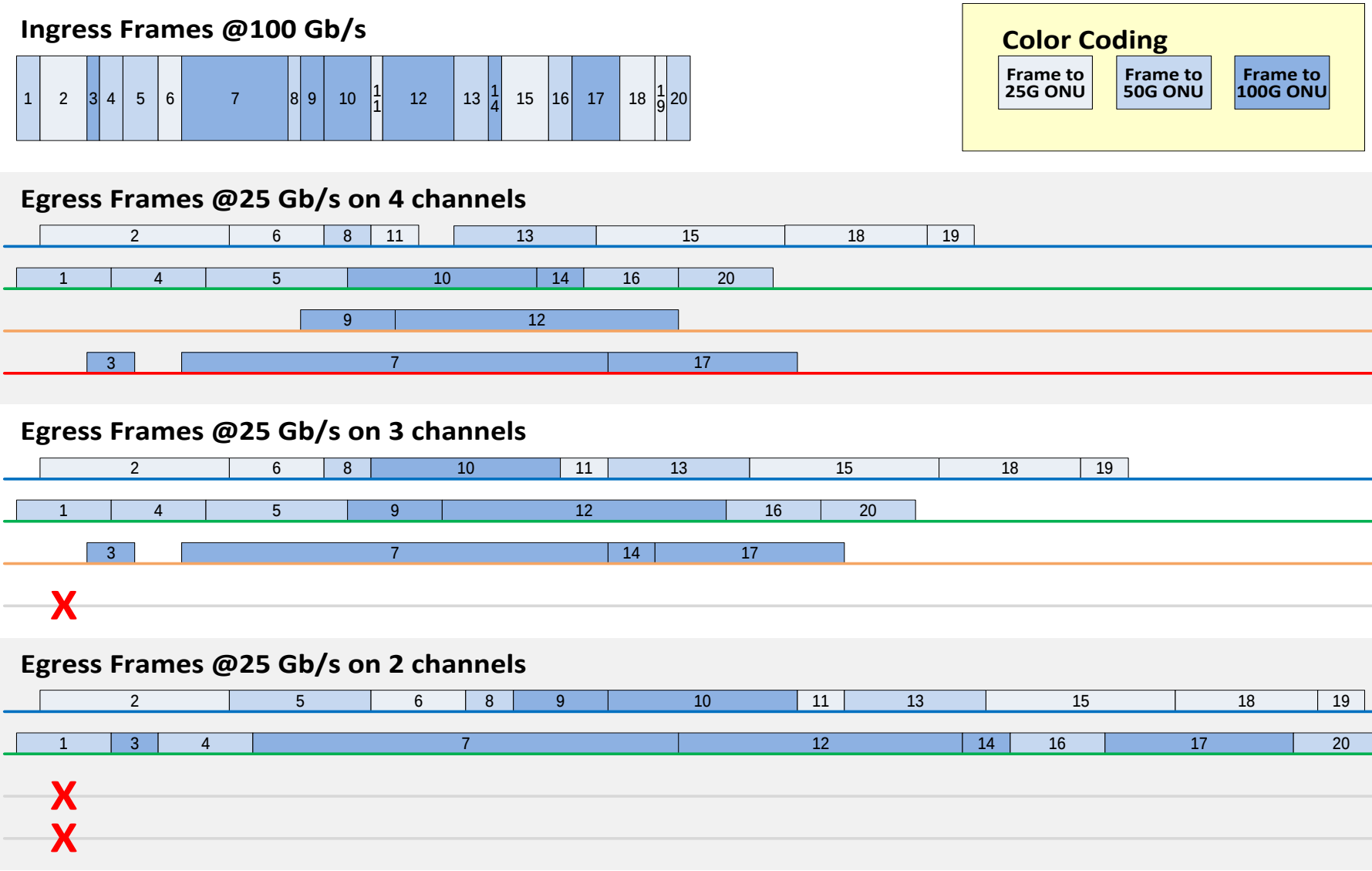
- Downstream MPCP+ operation:
  1. **MPCP+ (Tx@OLT):**
     Uses *Lane-Aware Frame Distributor* (LAFD) algorithm
  2. **MPCP+ (Rx@ONU):**
     Uses *Lane-Aware Frame Combiner* (LAFC) algorithm
- Upstream MPCP+ operation :
  3. **MPCP+ (Tx@ONU):**
     Uses *Grant-Aware Frame Distributor* (GAFD) algorithm
  4. **MPCP+ (Rx@OLT):**
     Uses *Grant-Aware Frame Combiner* (GAFC) algorithm
- Four new state diagrams, but not any more difficult than what we have in 1G-EPON/10G-EPON

Thank You

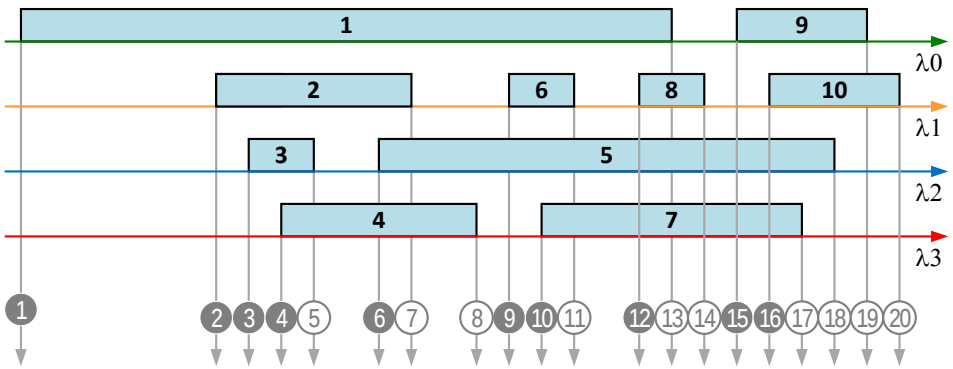
Backup

Tx over Reduced Number of Channels

Illustration of the same downstream frame sequence sent over 4, 3, and 2 channels (same LAFD algorithm, different DLC tables)



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