Contents

8 Bandwidth allocation mechanisms

8.1 Introduction

Clause 8 provides definitions of mechanisms, parameters, and functions related to bandwidth allocation. The bandwidth allocation mechanisms have a direct impoact on quality of service (QoS) and performance of different services.

8.1.1 Traffic types and services

As converged network architecture, EPON is expected to support a variety of traffic types and services, including the following:

- Real-time flows with periodic fixed-size data frames
- Circuit emulation service
- Mobile backhaul service
- Real-time flows with variable-size data frames or with periodic inactivity
- VoIP service
- IPTV service
- Non-real-time flows that require throughput/frame loss guarantees
- Guaranteed data service
- Non-real-time, non-guaranteed flows
- Best effort service

The definitions of these services, including their bandwidth profiles, frame size distributions, and latency requirements are given in IEEE Std 1904.1, subclause 8.2.

8.1.2 QoS parameters and metrics

As described in IEEE Std 1904.1, subclause 8.2, various traffic types require different network performance. The performance of a frame-based network (and EPON in particular) can be conveniently characterized by several parameters: bandwidth (throughput), frame delay (latency), delay variation (jitter), and frame loss ratio. The definitions and measurement methodologies for these parameters are given in IEEE Std 1904.1, subclause 8.3.

8.2 Principles of bandwidth allocation in Nx25G-EPON

As was detailed in Clause 4, in the Nx25G-EPON architecture, the OLT and the ONUs instantiate multiple MAC instances (LLIDs). Only a single MAC instance may be transmitting at any time on a given channel. Each MAC instance transmits at a pre-defined time and for a pre-defined duration.

The transmission by an individual MAC instance on a single channel is called an *envelope*, and the assignment of such transmission window is called the *envelope allocation*.

8.2.1 Transmission envelopes

A transmission envelope represents a continuous transmission by a specific MAC instance (LLID) on a specific MCRS channel. A transmission envelope is always transmitted on a single MCRS channel. The concept of transmission envelope is explained in IEEE Std 802.3, subclause 143.2.4.2.

The envelopes are formed within the MCRS according to transmission requests issued by the local MPCP client through the *MCRS_CTRL.request()* primitives (see IEEE Std 802.3, 143.3.1.2.1).

8.2.2 Transmission over multiple channels in 50G-EPON

In 50G-EPON systems, multiple channels exist in the downstream direction, and may also exist in the upstream direction. In the transmission direction where multiple channels are present, there can be an independent scheduler for each channel, or a single common scheduler may be used to schedule all channels simultaneously (see 8.4.1.2.1).

In case of independent schedulers, different MAC instances may be scheduled to transmit on different channels at the same time. Transmission envelopes on each channel may start and stop at different times with no alignment of envelopes.

In implementations where a single scheduler is used, a given MAC instance is always scheduled to transmit on all available channels and envelope start and stop times are synchronized across all channels.

8.2.2.1 Dynamic channel bonding

In 50G-EPON systems with multiple channels, the transmission envelopes allocated to a given MAC instance (LLID) may overlap in time. When a single scheduler is used, the envelopes overlap completely, i.e., for every grant, the scheduled LLID starts and stops the transmission on all channels at the same time. In case of multiple independent schedulers, the envelopes allocated to a given LLID may overlap fully or partially, as shown in Figure 8-1.

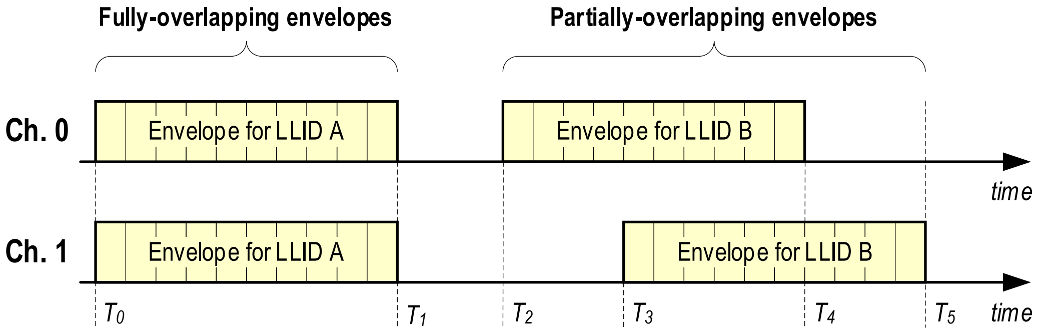


Figure 8-1 – Overlapping envelopes in 50G-EPON system

Whenever the envelopes allocated to the same MAC instance overlap in time on multiple channels, the channel bonding mechanism of MCRS activates automatically (see IEEE Std 802.3, Clause 143). During the envelope overlap interval, the given MAC instance is transmitting at full 50 Gb/s data rate.

8.3 Downstream transmission

In the downstream direction, each MAC instance in the OLT transmits data toward a single MAC instance in one of the ONUs (i.e., a P2P LLID) or towards a set of MAC instances in multiple ONUs (i.e., a P2MP LLID). The OLT internally arbitrates its MAC instances to ensure that only a single instance is transmitting at any time.

8.3.1 Scheduling of downstream envelopes

In the downstream direction, the OLT MPCP client schedules the envelopes based on state of downstream queues, previsioned QoS parameters, and the downstream scheduling algorithm implemented by the OLT. The scheduling algorithm is outside the scope of this standard.

There is flexibility in how the OLT's local MPCP client may issue the transmission requests (i.e., the *MCRS_CTRL.request()* primitives) to the MCRS. A separate transmission request may be issued for each individual downstream frame, or for a group of frames stored in a single queue. In 50G-EPON systems, the MPCP client may issue a set of transmission requests allocating multiple overlapping envelopes to a single frame or to a group of frames in the same queue.

8.3.2 Frame fragmentation in downstream direction

The OLT shall not fragment any data frames transmitted in the downstream direction. The local MPCP client has visibility into the contents of the downstream queues and is able to issue transmission requests (i.e., *MCRS_CTRL.request()* primitives) with the `env_length` parameter that accommodates one or more complete frames.

In systems with multiple channels, envelopes may overlap and a frame may be striped over multiple channels with each channel transporting parts of this frame (see 8.2.2.1). However, at the conclusion of the overlapped transmission, no frame may remain fragmented. Therefore, in situations where the MPCP client schedules overlapping envelopes to a MAC instance, the sum of all `env_length` parameters in these overlapping transmission requests has to accommodate a number of complete frames, while the individual `env_length` values may not be frame-size aligned.

A simple scheduling algorithm that allocates an envelope to an individual frame with the highest transmission priority on a channel with the earliest availability satisfies the above requirements. Subclause 8.3.3 illustrates an example of such scheduling.

Since the fragmentation in the downstream direction is disallowed, the ONUs do not need to implement frame reassembly buffers in their receive data paths.

8.3.3 Downstream scheduling example

In this example, OLT has two downstream queues: a higher priority LLID A queue containing a single frame A1 and a lower priority LLID B queue containing frames B1 and B2 (see Figure 8-2 (a)). Frame preambles and IPGs are not stored in the queues.

The scheduler allocates a separate envelope to each frame on the first available channel. The allocated envelope's length is increased by two or three EQs in order to accommodate the Envelope Start Header (ESH), frame preamble, and possibly an Idle EQ. In the MCRS sublayer, frame preamble is replaced by the Envelope Continuation Header (ECH). The presence of Idle EQ depends on the preceding frame's length as explained in IEEE Std 802.3, 143.2.4.4. In this example, Idle EQs are necessary.

As shown in Figure 8-2 (b), at time T_0, both channels are available, and so an envelope for frame A1 is allocated on channel 0 and an envelope for frame B1 is allocated on channel 1. At time T_1, the transmission of frame A1 completes and the channel 0 becomes available again. The envelope for frame B2 is scheduled to start on channel 0 at time T_1.

Note that actual envelope scheduling may happen ahead of envelope start times, since the future time at which each channel will become available is known in advance. However, delaying the envelope scheduling until a channel is about to become available allows the scheduler to react to late-arriving higher-priority frames.

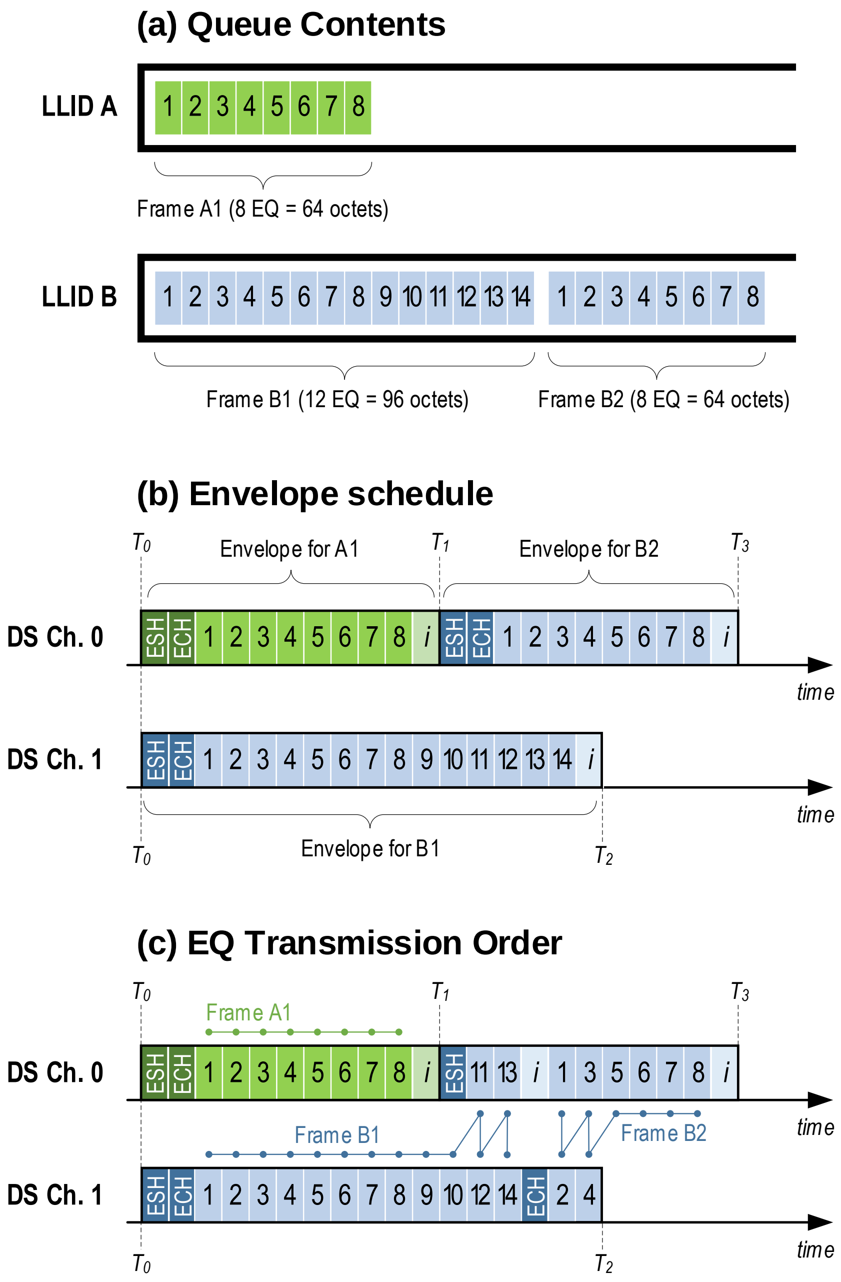

(a) Queue Contents
LLID A
1 2 3 4 5 6 7 8
Frame A1 (8 EQ = 64 octets)
LLID B
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8
Frame B1 (12 EQ = 96 octets)
Frame B2 (8 EQ = 64 octets)
(b) Envelope schedule
T0
Envelope for A1
T1
Envelope for B2
T3
DS Ch. 0
ESH ECH 1 2 3 4 5 6 7 8 i ESH ECH 1 2 3 4 5 6 7 8 i
time
DS Ch. 1
ESH ECH 1 2 3 4 5 6 7 8 9 10 11 12 13 14 i
time
T0
Envelope for B1
T2
(c) EQ Transmission Order
T0
T1
T3
Frame A1
DS Ch. 0
ESH ECH 1 2 3 4 5 6 7 8 i ESH 11 13 i 1 3 5 6 7 8 i
time
Frame B1
Frame B2
DS Ch. 1
ESH ECH 1 2 3 4 5 6 7 8 9 10 12 14 ECH 2 4
time
T0
T2

Figure 8-2 -- Downstream scheduling using a separate envelope for each frame

When the OLT local time approaches T_0, the channel 0 in the above example starts transmitting the frame A1 and channel 1 starts transmitting the frame B1. At time T_1, the frame A1 completes its transmission and channel 0 starts serving the queue of LLID B. While the envelope was initially allocated to accommodate the frame B2, the preceding frame B1 has not completed its transmission yet. Therefore, as shown in Figure 8-2 (c), at time T_1, both channels 0 and 1 serve the remainder of the frame B1. Because a portion of frame B1 was transmitted over both channels (i.e., transmitted at 50 Gb/s), this frame completes the transmission sooner, while the envelope initially allocated for the B1 is still active. At that time, the transmission of frame B2 starts on both channels. At time T_2, the envelope for LLID B on channel 1 closes and the remainder of the frame B2 is transmitted only on channel 0.

This example illustrates that while the overlapping envelopes are allocated based on individual frame sizes, the frame transmission order strictly follows the serial order of frames in each queue. A later frame does not start transmission before the transmission of the earlier frame is completed -- this is a key feature of the MCRS specified in IEEE Std 802.3, Clause 143.

8.3.4 Double-header suppression (?)

<TBD>

8.4 Upstream transmission

In the upstream direction, each MAC instance in an ONU transmits data towards one of the MAC instances in the OLT. To avoid upstream data collisions, transmission windows for all MAC instances at the ONUs are centrally controlled in such a way that only a single MAC instance's transmission reaches the OLT port at any given time. This arbitration is achieved by allocating envelopes to MAC instances in the ONUs. Each MAC instance defers its transmission until the start of its envelope. When the envelope transmission starts, the MAC instance transmits its queued frames at full line rate for the duration of its assigned envelope.

To simplify bandwidth allocation and to ensure full utilization of allocated envelopes, the Nx25G-EPON systems support frame fragmentation in the upstream direction. The upstream frame fragmentation mechanisms are defined in 8.4.3.2.

The upstream envelopes are formed within the ONU MCRS according to transmission requests issued by the local MPCP client through the *MCRS_CTRL.request()* primitives (see IEEE Std 802.3, 143.3.1.2.1). The ONU MPCP client schedules the envelopes based on the envelope allocations received from the OLT and the state of local upstream queues. The mechanism of allocating envelopes (i.e., granting bandwidth) to the ONUs is defined in 8.4.1.

Reporting of a queue occupancy state or congestion by different ONUs assists the OLT in optimal allocation of the transmission windows across the PON. ONUs report the queue state on a per-LLID basis, as detailed in 8.4.2.

8.4.1 Bandwidth granting

8.4.1.1 Definition of GATE MPCPDU

The format and semantic of the fields of GATE MPCPDU are specified in IEEE Std 802.3, 144.3.6.1.

8.4.1.2 Envelope Allocations

The `EnvAlloc` structure in the GATE MPCPDU carries the length of the envelope (in the field `EnvLength`) in units of EQ. One EQ in each envelope is taken by the Envelope Start Header (ESH) and the rest of the EQs are filled with data sourced from a corresponding MAC instance. Each of the EQs sourced from the MAC may be either a data EQ, an idle EQ, or a preamble EQ. Within the transmitting MCRS entity, the preamble EQs are replaced with Envelope Continuation Header (ECH) EQs, and within the receiving MCRS entity, a reverse transformation takes place.

8.4.1.2.1 Envelope allocations on multiple channels

A 50/50G-EPON ONU (i.e., an ONU that supports two upstream channels) may be granted transmission opportunities on both upstream channels. In a case of a single, common scheduler, the GATE MPCPDU contains the `ChannelMap` field with the value 0x03, i.e., the envelope allocations apply to both channels simultaneously. In a case of two independent schedulers, each GATE applies to either channel 0 (if

`ChannelMap == 0x01`) or channel 1 (if `ChannelMap == 0x02`). A GATE MPCPDU arriving on any downstream channel may carry envelope allocations for any of the upstream channels.

8.4.1.3 Scheduling upstream bursts

The sum of all `EnvLength` values in a GATE MPCPDU represents the payload portion of the upstream burst, i.e., the FEC-protected area, as illustrated in IEEE Std 802.3, Figure 142-3. The actual transmission from the ONU also includes optical overhead, line coding overhead, and FEC overhead.

To schedule consecutive bursts, the scheduler in the OLT has to calculate the total burst size that ONU is to transmit for a given grant, including the overhead components mentioned above. Note that different ONUs may be provisioned to use different optical overhead values. The following five steps represent the calculations performed by the OLT:

Step 1 – Calculate the number of 257-bit blocks (*B*) taken by all envelopes together:

$B = \lceil L/4 \rceil$,

where

L – the sum of all envelope lengths in the current grant ($L = \sum_i EnvLength[i]$);

4 is the number of EQs that fit into one 257-bit block

Step 2 – Calculate the total number of FEC codewords (*C*) used in this burst (including possible shortened last codeword):

$C = \lceil B/FEC\_PAYLOAD\_SIZE \rceil$,

where

FEC_PAYLOAD_SIZE = 56 blocks (see IEEE Std. 802.3, 142.2.5.1)

Step 3 – Calculate the total number 257-bit blocks (*P*) taken by FEC-protected portion of the burst:

$P = B + C \times FEC\_PARITY\_SIZE$,

where

FEC_PARITY_SIZE = 10 blocks (see IEEE Std. 802.3, 142.2.5.1)

Step 4 – Calculate the total burst size (*S*) in units of 257-bit blocks:

$S = SP1Length + SP2Length + SP3Length + P + EBD\_length$,

where

SP1Length, *SP2Length*, *SP3Length* – lengths of synchronization pattern regions provisioned into given ONU via REGISTER MPCPDU (see IEEE Std 802.3, 144.3.6.4).

EBD_length = 1 block

Step 5 – Calculate the total burst time (*T*) in units of EQT:

$T = \lceil S \times PCS\_BLK\_SZ/66 \rceil + LaserOffTime$,

where

PCS_BLK_SZ = 257 bits, see IEEE Std. 802.3, 142.3.5.1

66 is the number of bits (at 25.78125 Gb/s line rate) that are transmitted in the interval of time equal one EQT

LaserOffTime – the time required to turn the ONU transmitter off (in units of EQT). This value is conveyed by the ONU to the OLT in the REGISTER_REQ MPCPDU (see IEEE Std 802.3, 144.3.6.3)

8.4.1.4 Requesting LLID reports

The OLT may request ONU to report the length of a given upstream queue by setting the `ForceReport` (FR) flag to one in a corresponding `EnvAlloc` structure. By setting the FR flag to one and the `EnvLength` field to zero, the OLT can request just a queue state report for this LLID without allocating an envelope for its data.

8.4.1.5 PLID envelope allocations

As is explained in IEEE Std 802.3, 143.3.1.2.3 and 144.3.1.1, the value of the `Timestamp` field in MPCPDUs references the transmission (and reception) time of the ESH preceding these MPCPDUs. The OLT does not allocate partially overlapping envelopes to the PLID, because these envelopes may transmit the ESH at different times. However, allocating fully overlapping envelopes is allowed since both ESH fields will be transmitted at the same time (see Figure 8-1).

If an ONU is given partially overlapping PLID envelope allocations, it chooses only one of these envelopes for MPCPDU transmission, and only if the envelope length is enough for at least one complete MPCPDU. The ONU ignores the rest of the overlapping PLID envelope allocations.

There are special considerations for the values of `EnvAlloc` fields when this structure applies to a PLID, as explained below.

8.4.1.5.1 `EnvLength` value

The value of the PLID `EnvLength` field informs the ONU about how many REPORT MPCPDUs it is expected to generate. The ONU shall not generate more REPORT MPCPDUs than it is able to transmit in a given PLID envelope.

To avoid allocation inefficiencies, the OLT should set the length of the PLID EnvLength field to $10 \times N_R + 1$, where N_R is the desired number of the REPORT MPCPDUs. Typically, the number of REPORT MPCPDUs depends on the total number of LLIDs that have `ForceReport` flags asserted (N_{FR}) in their `EnvAlloc` structures in a given grant, and is calculated as $N_R = \lceil N_{FR}/7 \rceil$.

The OLT, at its discretion, may issue a PLID grant allowing a larger number of REPORT MPCPDUs to be transmitted than what is needed to accommodate N_{FR} individual LLID reports. This is further explained in 8.4.2.4.

8.4.1.5.2 `Fragmentation` flag

Per IEEE Std 802.3, 144.3.6.1, the ONU does not fragment MPCPDU frames, regardless of the value of the `Fragmentation` flag in the `EnvAlloc` structure that allocates a PLID envelope.

8.4.1.5.3 `ForceReport` flag

For LLIDs other than PLIDs, the `ForceReport` flag indicates that the ONU is to report the total length of the frames (including IPG and preamble), queued for transmission on this specific LLID. When the respective flag is set to 0, the ONU is not required to report the length of the given queue.

Note that REPORT MPCPDUs are generated just in time, and if the PLID queue length is to be reported, it would always show the length of zero. Therefore, the `ForceReport` flag is assigned a special meaning when it is associated with the PLID `EnvAlloc` structure. If the PLID `ForceReport` flag is asserted, the ONU shall generate at least a single REPORT MPCPDU in the corresponding PLID envelope, and may generate more than one REPORT if the `EnvLength` value allows it. The OLT relies on the presence of REPORT MPCPDUs as an indication of MPCP health (see the definition of `MissedReportCount` in IEEE Std 802.3, 144.3.7.3). If the OLT misses eight or more REPORT MPCPDUs (i.e., `MISSED_REPORT_LIMIT`), it deregisters the ONU.

If the `ForceReport` flag is 0, the ONU is not required to generate any REPORT MPCPDUs in the given PLID envelope if there are no changes to any of the upstream queues in the ONU. The OLT does not count a REPORT MPCPDU as missing (i.e., it shall not increment the `MissedReportCount` variable) if it did not receive a REPORT MPCPDU in a PLID envelope and the corresponding envelope allocation had `ForceReport` flag set to 0. This mechanism allows an idle ONU to keep its transmitter turned off and is one of the key power saving features. This behavior is further defined in 8.4.4.1.

8.4.1.6 Multi-PDU grant

As explained in 8.4.1.1, a GATE MPCPDU contains a transmission start time (`StartTime` field) and up seven envelope allocations (`EnvAlloc[i]` fields). The `EnvAlloc` fields may allocate envelopes to the same or different LLIDs.

When it is necessary to allocate more than seven envelopes in one grant, the OLT may issue multiple GATE MPCPDUs. All GATE MPCPDUs that allocate envelopes for the same grant shall have the same `StartTime` value, which is the time at which the ONU is expected to transmit the first envelope header in a burst (i.e., the ESH immediately following the SBD). The GATE MPCPDUs representing a single grant do not necessarily need to be transmitted back to back.

Per IEEE Std 802.3, 144.3.8, the MPCP client at the ONU processes all GATE MPCPDUs pertaining to the same grant before committing to the set of envelopes covering the entire grant. The MPCP client shall use the following criteria to determine that the sequence of GATE MPCPDUs pertaining to the grant with `StartTime` = *T* has concluded and that it should generate a set of envelope descriptors for the given grant:

a) A GATE MPCPDU is received with the value of the `StartTime` field that is different from *T*, or

b) The MPCP `LocalTime` value has reach the grant cut-off time, which is equal to *T* − `MPCP_PROCESS_DLY`.

The definitions of the above constants and variables are provided in IEEE Std 802.3, 144.3.8.

8.4.2 Queue state reporting

The ONUs report the queue state on a per-LLID basis using the REPORT MPCPDUs. For each LLID, an ONU shall report a single value that represents the total queue length, including the associated framing overhead, i.e., the minimum IPG and frame preambles. The values are reported in units of EQ.

In Nx25G-EPON systems, the ONUs do not need to track and report any intermediate frame boundaries within the queues.

8.4.2.1 Definition of REPORT MPCPDU

The format and semantic of the fields of REPORT MPCPDU are specified in IEEE Std 802.3, 144.3.6.2.

8.4.2.2 Reporting real-time queue lengths

The ONU shall report real-time queue length for all LLIDs at the moment when the REPORT MPCPDUs are generated. The reported queue lengths depend on where the PLID envelope is scheduled relative to the envelopes of LLIDs being reported.

If the PLID envelope is scheduled to be transmitted ahead of an LLID being reported, then the reported queue length will include the LLID's data that is to be transmitted later in the same grant.

If the PLID envelope is scheduled to be transmitted after other LLIDs (i.e., it is scheduled as the last envelope in a grant), then the reported queue lengths will exclude the LLID data that has been transmitted earlier in the same grant. The latter case more accurately represents the volume of queued data available for the next grant.

8.4.2.3 Multi-PDU reporting

As described in 8.4.2.1, a REPORT MPCPDU contains up seven LLID reports (`LlidStatus[i]` structures). Each `LlidStatus[i]` structure reports the queue length of a different LLID.

When it is necessary to report the queue lengths of more than seven LLIDs and if the corresponding PLID envelope length allows it, the ONU shall generate multiple REPORT MPCPDUs. All REPORT MPCPDUs that are transmitted in the same PLID envelope have the same `Timestamp` value that corresponds to the time the PLID ESH is transmitted by the MCRS.

8.4.2.4 Mandatory and gratuitous reports

A *mandatory* LLID report is a report that was explicitly requested by the OLT via setting the `ForceReport` flag to 1 in a previous `EnvAlloc` to this LLID. Conversely, a *gratuitous* LLID report is a report sent by the ONU without an explicit request by the OLT.

The ONU may be allocated more space for LLID reports than the number of `ForceReport` flags set to 1 in the current grant. Recall that a single REPORT MPCPDU can carry up to seven `LlidStatus`, structures, so after the last mandatory LLID report, there may be one to six extra slots in the last REPORT MPCPDU. The OLT may even grant extra envelope space for additional full REPORT MPCPDUs, so the number of potential gratuitous reports may be arbitrarily large.

The Figure 8-3 illustrates the case where a grant had ten `ForceReport` flags set to 1, and also allocated PLID envelope length sufficient for three REPORT MPCPDUs (i.e., `EnvLength` = 3×10 + 1 = 31, according to 8.4.1.5.1). Such granting allows the ONU to generate up to 11 gratuitous LLID reports.

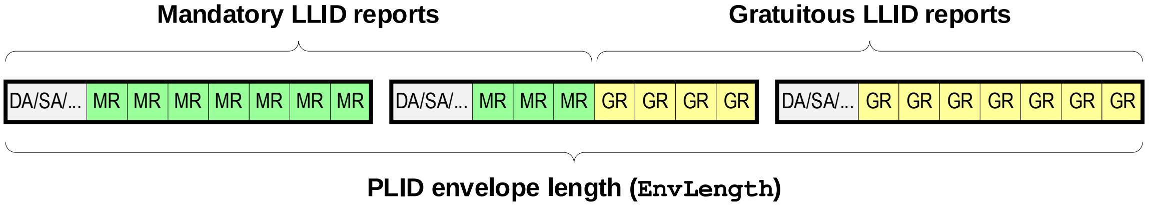


Figure 8-3 – Mandatory and gratuitous LLID reports

8.4.2.5 LLID reporting priorities

The mandatory `LlidStatus` reports are generated first and they are placed in REPORT MPCPDU(s) in the same order as the order of `EnvAlloc` structures with the `ForceReport` flag set to 1 in a corresponding GATE MPCPDU(s).

After generating the mandatory `LlidStatus` reports, the ONU shall use the remaining slots in the REPORT MPCPDUs for gratuitous `LlidStatus` reports. It is possible that not all LLIDs provisioned in the given ONU can be accommodated by the remaining slots in the REPORT MPCPDUs. To ensure that the more important queue changes are conveyed to the OLT in a timely manner, the ONU shall generate the gratuitous reports according to the reporting priority shown in Table 8-1. To determine the reporting priority, the ONU keeps track of the last queue length reported for each LLID and the changes to each queue since its last report.

Table 8-1 – Priority for gratuitous reports

LLID reporting priority	Last reported `QueueLength`	New arrivals
1 – An idle LLID became active	= 0	Yes
2 – LLID was active and had new arrivals	> 0	Yes
3 – LLID has residual data, but no new arrivals	> 0	No
4 – LLID was idle and remains idle[a]	= 0	No

[a] This priority is optional to report, even if there are available `LlidStatus` slots in REPORT MPCPDUs

8.4.3 Upstream burst composition

The structure of the upstream burst is shown in IEEE Std 802.3, 142.1.3. The FEC-protected area of a burst consists of one or more complete envelopes.

8.4.3.1 Order of envelopes in a burst

The GATE MPCPDUs are passed to the MPCP client in the order they are received. The MPCP client at the ONU shall generate the upstream envelopes following the order of `EnvAlloc[i]` fields in each GATE MPCPDU. It is possible for an LLID to be allocated multiple disjoint envelopes within the same grant.

If a GATE MPCPDU contained an envelope allocation for a GLID, then the portion of the upstream transmission that corresponds to this `EnvAlloc` may comprise multiple envelopes carrying data frames for different LLIDs that are members of the given GLID (see 8.5.x.x).

8.4.3.2 Frame fragmentation in upstream direction

To simplify bandwidth allocation and to ensure full utilization of allocated envelopes, the Nx25G-EPON systems support frame fragmentation in the upstream direction. To support the upstream frame fragmentation, the ONUs shall implement a segmentation function and the OLT shall implement a reassembly function. Both segmentation in the ONUs and reassembly in the OLT operate on a per-LLID basis, i.e., different LLIDs may have frame fragments pending reassembly in the OLT reassembly buffers, while the remaining frame fragments are waiting for transmission opportunities in the segmentation buffers in various ONUs.

Per each LLID, the reassembly buffer may contain one or more fragments, collectively comprising the head portion of single frame. The segmentation buffer may contain the tail portion of a frame that is yet to be transmitted in one or more envelopes. As illustrated in Figure 8-4, for each LLID, the fragments stored in the segmentation buffer and the reassembly buffer at any moment of time may belong to the same or different frames, considering that some fragments or even complete frames belonging to the same LLID may be in flight at that time. The reassembly and segmentation buffers for LLID A respectively contain the head portion and tail portion of the same frame A1. The reassembly and segmentation buffers for LLID B contain the head portion and tail portion of different frames, with some frame fragments being in flight.

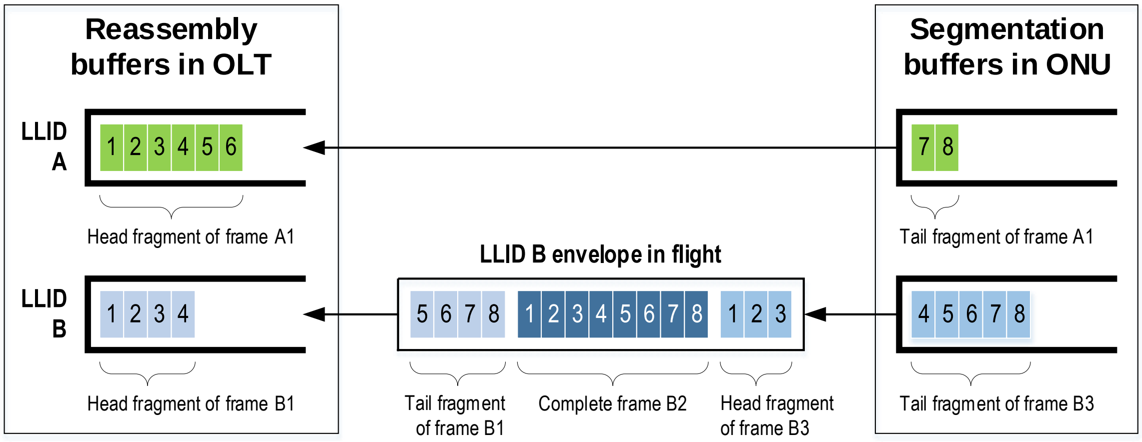

Reassembly buffers in OLT
LLID A
1 2 3 4 5 6
Head fragment of frame A1
LLID B
1 2 3 4
Head fragment of frame B1
LLID B envelope in flight
5 6 7 8
1 2 3 4 5 6 7 8
1 2 3
Tail fragment of frame B1
Complete frame B2
Head fragment of frame B3
Segmentation buffers in ONU
7 8
Tail fragment of frame A1
4 5 6 7 8
Tail fragment of frame B3

Figure 8-4 – Frame fragments stored in the segmentation and reassembly buffers

Any envelope may contain at most two partial frames: a tail fragment at the beginning of the envelope, and a head fragment of another frame at the end of the envelope. There can be any number of complete (non-fragmented) frames in the middle of the envelope.

8.4.3.2.1 Frame reassembly function in the OLT

Every bidirectional ULID and MLID may have upstream frame fragments pending reassembly. The reassembly buffer shall be able to accommodate at least one maximum-size frame that may be encountered on the given LLID. Maximum frame size supported by an ONU may be queried using the *aOnuMaxFrameSizeCapability* attribute (14.4.2.14) and it may be further restricted per individual service port using the *aUniMaxFrameSizeLimit* attribute (14.4.2.15).

Since generally not all LLIDs are active all the time, the OLT may reduce the overall size of the reassembly memory by relying on statistical memory sharing techniques. However, there could be transient periods of increased LLID activity leading to oversubscription of the reassembly buffer. To avoid frame loss due to the reassembly buffer oversubscription, the OLT is able to disable fragmentation dynamically on a per-envelope basis.

The fragmentation is controlled via the `Fragmentation` flag in the `EnvAlloc` structures in GATE MPCPDU (see IEEE Std 802.3, 144.3.6.1). Asserting the `Fragmentation` flag allows the ONU to fragment frames that are to be transmitted in the envelope allocated by the given `EnvAlloc` structure. Setting the `Fragmentation` flag to 0 forbids the ONU from fragmenting new frames in the given envelope. However, in order to preserve the order of frames within an LLID, a pending frame fragment, if exists, is still transmitted by the ONU. Therefore, the MPCP client in the OLT shall clear the `Fragmentation` flags with enough headroom remaining in the reassembly buffers to accommodate the frame fragments that still may be pending in the ONUs or are in-flight.

The OLT shall not sustain frame loss due to overflow of the reassembly buffers.

8.4.3.2.2 Frame segmentation function in the ONU

The segmentation function in the ONU allows a partial frame to be transmitted upstream toward the OLT. Conceptually, a full upstream frame is accepted from a frame storage into a segmentation buffer for a given LLID and then this frame is served (i.e., passed by the MAC instance to MCRS) one EQ at a time when an envelope for this LLID opens/activates. Implementations may choose to combine the segmentation buffer with the main frame storage (i.e., the upstream queue).

Serving the data into the envelopes depends on the value of the `Fragmentation` flag in the corresponding `EnvAlloc` structure received from the OLT.

If the `Fragmentation` flag for the given envelope is set to 1, and assuming there are enough frames queued for the upstream transmission to fill this envelope, the ONU shall fill the allocated envelope completely. The ONU shall not send the Envelope Continuation Header (ECH), which represents a preamble of a next frame, in the last EQ in the envelope. Instead, it shall defer the ECH to the next envelope for the same LLID, and fill the last EQ with the idle code.

If the `Fragmentation` flag for the given envelope is set to zero, the ONU shall not fragment new frames. If a frame fragment is already stored in the segmentation buffer for that LLID, this fragment is transmitted first. After the entire remaining fragment is transmitted, the ONU may start transmission of a new frame, only if the entire frame can fit in the remainder of the envelope, or in case of a multi-channel system, if the entire frame can fit in the multiple overlapping envelopes. Otherwise, the remainder of the envelope is filled with Idle EQs and the next frame is deferred to a future envelope.

Note that even with the `Fragmentation` flag cleared, the ONU is allowed to stripe any frame over multiple channels.

The ONU shall not fragment MPCPDU frames, regardless of the value of the `Fragmentation` flag in the `EnvAlloc` structure that allocates a PLID envelope.

8.4.3.3 Queue service discipline

Each bidirectional LLID is provisioned a single upstream queue via the extended OAM action *acConfigLlid* (14.6.2.8). The nature of this queue is left to implementation. While typically, a FIFO queue is assumed in the models described in Clauses 4 and 6, the upstream queue may also be implemented as a priority queue.

In a priority queue, a higher priority frame is inserted in the queue in front of lower priority frames. However, in a situation when the segmentation buffer contains a fragment of a partially-transmitted frame, that fragment is assumed to be of the highest priority and it shall be transmitted at the earliest opportunity and ahead of any later-arriving frame.

Note that the OLT has no direct control over the order of the frames transmitted by an ONU within a given LLID. If it is desired for the OLT to exercise such control, then the data frames are classified into distinct categories (e.g., by priority, class of service, VLAN tagging, etc.) and a separate LLID are provisioned for each category. Then, the OLT is able to grant each LLID independently.

8.4.3.4 Transmitting multiple bursts on multiple channels

As was described in 8.4.1.2.1, a 50/50G-EPON ONU may be granted transmission opportunities on both upstream channels.

In case of a single, common scheduler, the envelopes on both channels are synchronized, which implies that every MAC instance (LLID) scheduled to transmit in the given burst(s) is transmitting at 50 Gb/s and all LLID’s transmissions are serialized.

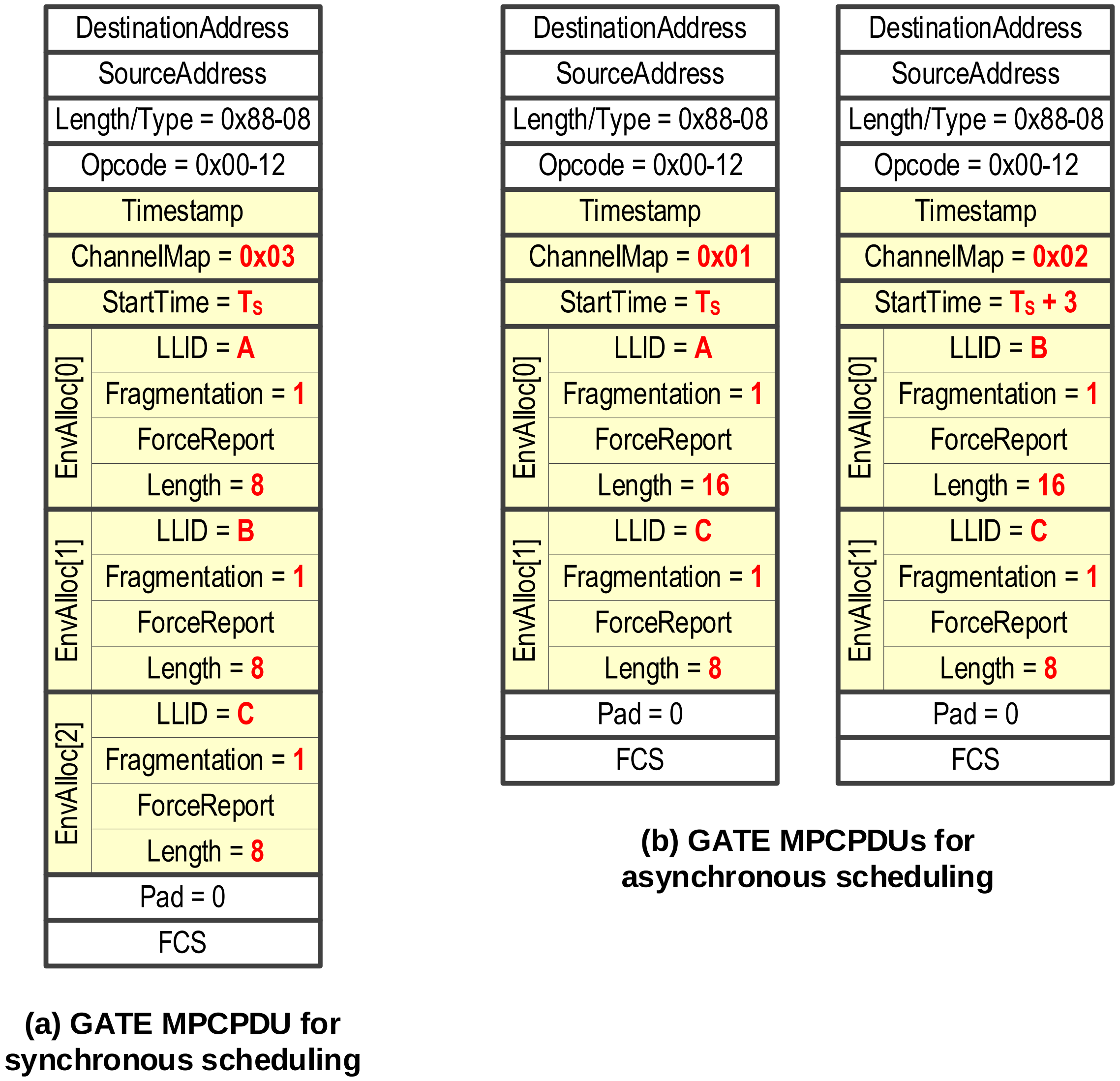
In a case of two independent schedulers, each GATE applies to one of the channels. These grants conveyed by these GATEs can be completely disjoint in time, or they may overlap. Even if the grants to the same ONU overlap in time, the envelopes within each of these grants may be allocated to different LLIDs on different channels, or they may be allocated to the same LLID. If the envelopes on both channels are allocated to the same LLID and these envelopes overlap in time, the MCRS channel-bonding feature (8.2.2.1) activates automatically, and, for the duration of the envelope overlap, the LLID transmits at the data rate of 50 Gb/s.

8.4.3.4.1 Upstream scheduling example

The following example illustrates allocation of upstream envelopes to LLIDs A, B, and C in two alternative scenarios: (a) synchronous allocation using a single common scheduler, and (b) asynchronous allocation using an independent scheduler for each upstream channel. Overall, in both scenarios, 16 EQs are allocated to each of the LLIDs. Figure 8-5 shows the relevant field values in GATE MPCPDU(s) used in this example.

In scenario (a), only a single GATE MPCPDU is issued. This GATE contains the `ChannelMap` field with the value 0x03, which instructs the ONU to apply the given envelope allocations to both channels simultaneously. This GATE allocates to every LLID two synchronous 8-EQ envelopes, which results is a combined allocation of 16 EQs per LLID.

The scenario (b) uses two separate GATEs; each GATE applies to either channel 0 (if `ChannelMap = 0x01`) or channel 1 (if `ChannelMap = 0x02`). Since the two schedulers are independent, the GATE messages have different start times (e.g., T_S and T_S+3, respectively). In this example, the 16-EQ envelopes for the LLIDs A and B are scheduled on separate channels, while the LLID C is scheduled on both channels and the two 8-EQ envelopes for this LLID happen to partially overlap in time.



(b) GATE MPCPDUs for asynchronous scheduling

(a) GATE MPCPDU for synchronous scheduling

Figure 8-5 – Contents of GATE MPCPDUs for (a) synchronous and (b) asynchronous schedule

The contents of the upstream queues for the LLIDs A, B, and C are shown in Figure 8-6. For simplicity, the segmentation buffers are shown as being a part of the upstream queue, thus a pedning fragment appears at the head of the respective queue. Note that LLIDs A and C have pending fragments, but LLID B does not.

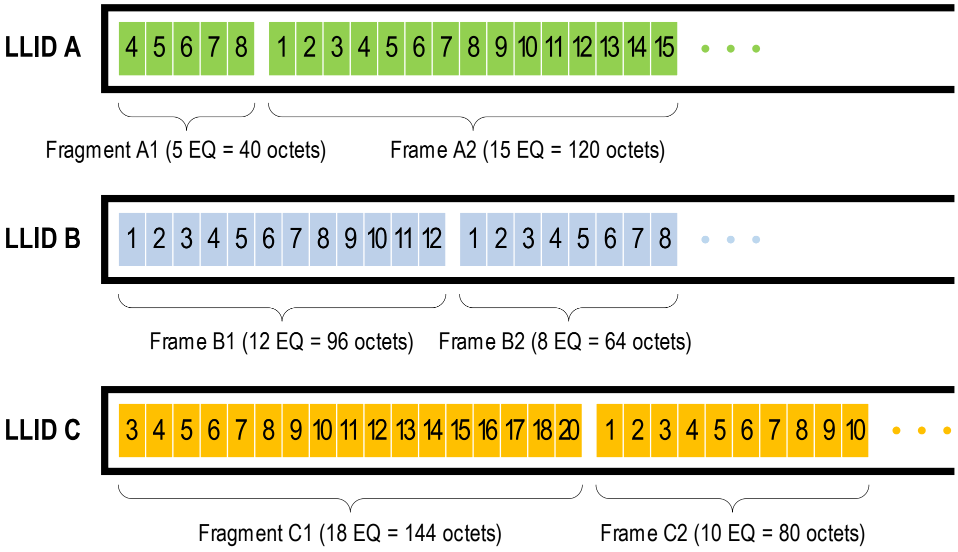
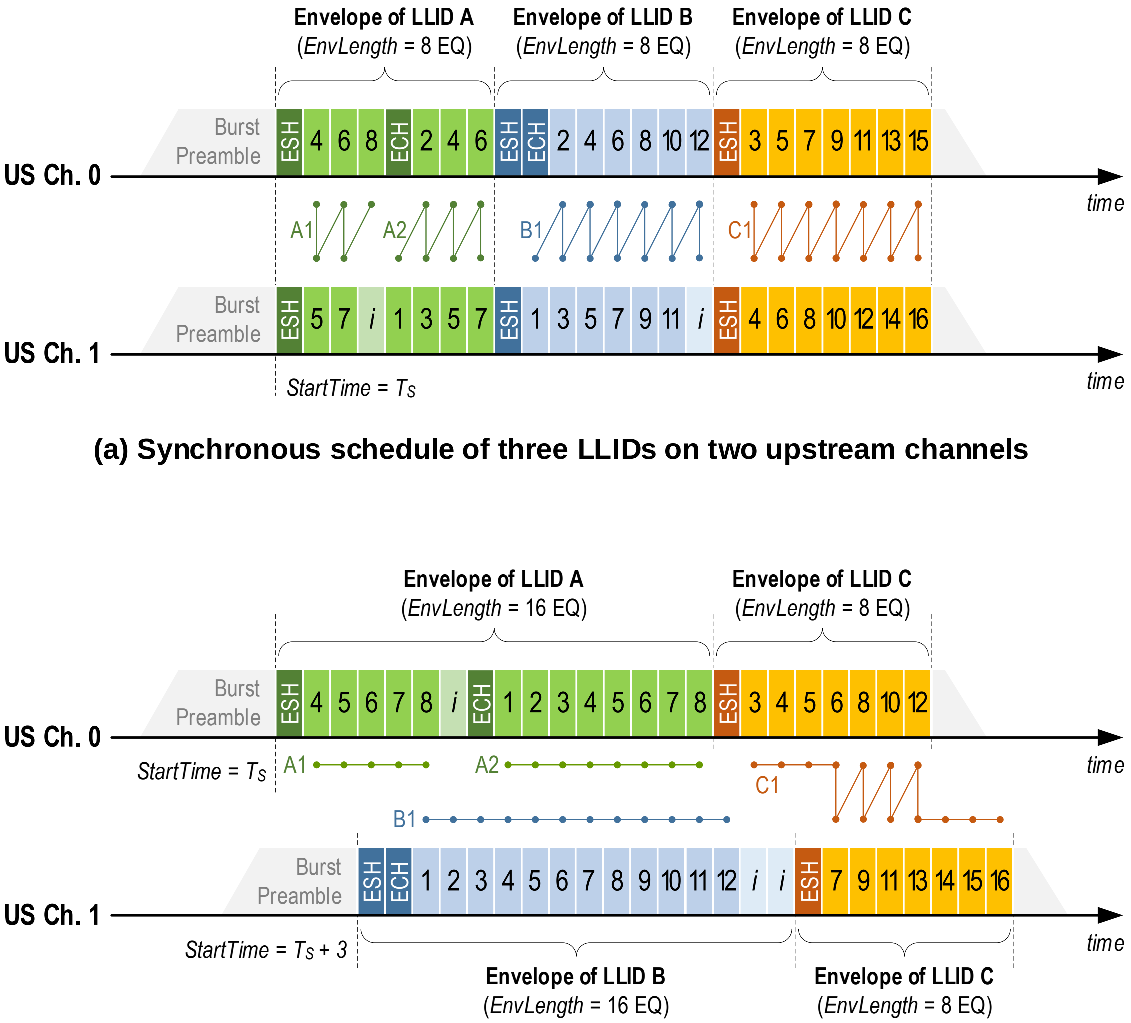

LLID A
4 5 6 7 8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Fragment A1 (5 EQ = 40 octets)
Frame A2 (15 EQ = 120 octets)
LLID B
1 2 3 4 5 6 7 8 9 10 11 12
1 2 3 4 5 6 7 8
Frame B1 (12 EQ = 96 octets)
Frame B2 (8 EQ = 64 octets)
LLID C
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20
1 2 3 4 5 6 7 8 9 10
Fragment C1 (18 EQ = 144 octets)
Frame C2 (10 EQ = 80 octets)

Figure 8-6 – Contents of upstream queues of LLIDs A, B, and C before the upstream transmission

The upstream transmission that results from the bandwidth allocations according to grants in Figure 8-5 is illustrated in Figure 8-7. In both scenarios in this example, the LLID A transmitted two partial frames: a tail fragment A1 and a head fragment of frame A2. The LLID B did not have any pending fragments and the frame B1 could fit entirely in the allocated envelope(s), therefore the complete frame B1 was transmitted. LLID C had a pending fragment that was larger than the allocated envelope length of 16 EQ. This LLID transmitted a fragment of the frame that did not include either the head or the tail portion of the frame.

In the scenario (a) shown in Figure 8-7 (a), every LLID transmits an 8-EQ envelope synchronously on both channels. Every envelope starts with an Envelope Start Header (ESH) and then the data that follows it is striped across both channels. The Envelope Continuation Headers (ECH) are transmitted in place of frame preambles.



(a) Synchronous schedule of three LLIDs on two upstream channels

(b) Asynchronous schedule of three LLIDs on two upstream channels

Figure 8-7—Envelope transmission examples corresponding to the schedules shown in Figure 8-5.

The upstream transmission in scenario (b) is illustrated in Figure 8-7 (b). Here, LLIDs A and B each transmitted a single 16-EQ envelope on channels 0 and 1 respectively. Note that in this case, only a single ESH was transmitted by each of these LLIDs, compared to two ESH fields transmitted by each LLID in scenario (a). This resulted in LLID A being able to transmit one extra data EQ (EQ #8). The LLID B also had an extra EQ available, but that EQ could only transmit the preamble of the next frame (i.e., the ECH). While transmitting ECH as the last EQ in an envelope is allowed per MCRS defined in IEEE Std 802.3, Clause 143, the typical implementation of MPCP Client is expected to simply defer the entire next frame, including the preamble, to a subsequent envelope. Thus, the LLID B in this example transmitted one extra Idle EQ at the end of the envelope, instead of the ECH of the next frame.

After completion of the envelope for LLID A on channel 0, the ONU started transmitting the 8-EQ envelope for LLID C. Three EQTs later, the channel 1 also became available and ONU started transmitting the second 8-EQ envelope for LLID C. After the ESH was transmitted in the second envelope, the LLID C data EQs started spanning both channels, as illustrated in the Figure 8-7 (b). After the envelope on channel 0 closed, the LLID C transmission continued for 3 more EQs on channel 1.

The contents of the upstream queues for the three LLIDs upon completion of the scheduled transmission are shown in Figure 8-8. Note that EQ #8 of the fragment A2 remains in the segmentation buffer only in scenario (a). In scenario (b) it is transmitted in the envelope allocated to LLID A.

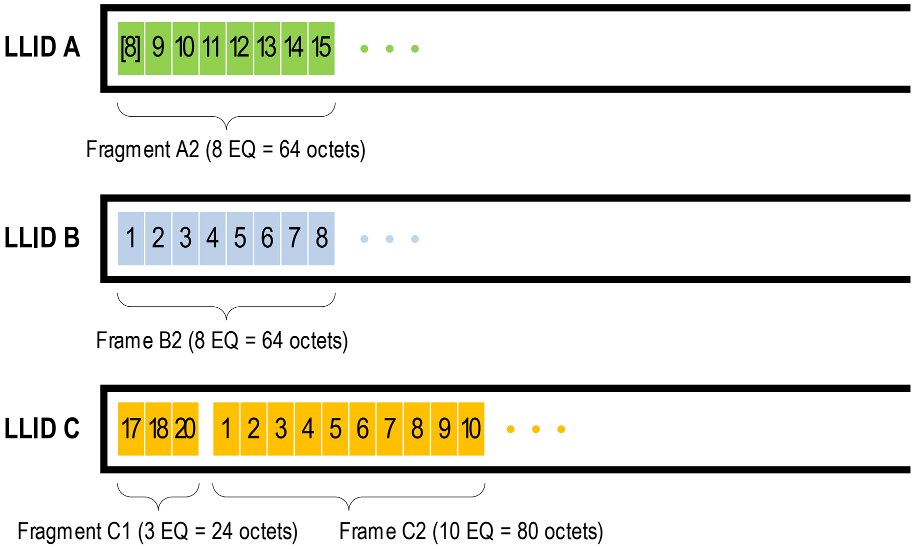

LLID A
[8]
9
10
11
12
13
14
15
Fragment A2 (8 EQ = 64 octets)
LLID B
1
2
3
4
5
6
7
8
Frame B2 (8 EQ = 64 octets)
LLID C
17
18
20
1
2
3
4
5
6
7
8
9
10
Fragment C1 (3 EQ = 24 octets)
Frame C2 (10 EQ = 80 octets)

Figure 8-6 – Contents of upstream queues of LLIDs A, B, and C after the upstream transmission

8.4.4 Polling overhead optimization

Typically, the OLT schedules PLID envelope as part of the same grant that schedules ONU's various LLIDs to transmit their user or management data. However, an idle ONU (i.e., ONU that has no queued data) is still granted a PLID envelope periodically in order to allow it to report the arrival of a new data. Every idle ONU is polled and depending of the latency bounds of the given LLID, the polling interval may need to be fairly short. This, coupled with potentially a large number of ONUs in a system, may result is a significant polling overhead.

The Nx25G-EPON systems include a number of mechanisms that allow the reduction or the polling overhead. In the ONU, key features that allows such optimization are the REPORT MPCPDU suppression (8.4.4.1) and upstream burst suppression (8.4.4.2). In the OLT, the key mechanisms include scheduling overlapping polling grants (8.4.4.3) and shared GATE MPCPDUs (8.4.4.4).

8.4.4.1 REPORT MPCPDU suppression

As was described in 8.4.1.5.3, the value of the `ForceReport` flag in the PLID envelope allocation affects the report generation by the ONU. The ONU shall not generate and transmit any REPORT MPCPDUs if all of the following conditions are true:

a) The PLID `EnvAlloc` structure that allocated the current PLID envelope contained the `ForceReport` flag set to 0.

b) The queues of all LLIDs that are mandatory to report (i.e., LLIDs, other than PLID, that had the `ForceReport` set to 1 in their respective `EnvAlloc` structures) are empty.

c) All LLIDs that are not mandatory to report (i.e., LLIDs that had the `ForceReport` set to 0 in their respective `EnvAlloc` structures) fall into the reporting priorities 3 or 4 (see Table 8-1).

If any of the above conditions are not true, the ONU generates one or more REPORT MPCPDUs, according to the procedures in 8.4.2.

If the REPORT MPCPDUs are suppressed, the ONU shall not transmit the PLID envelope in the upstream burst. Note that under some conditions, the entire upstream bursts may be suppressed (see 8.4.4.2).

8.4.4.2 Upstream burst suppression

If the conditions are met for the REPORT MPCPDU suppression per 8.4.4.1 and all other LLIDs (if there are any) allocated within the same grant have no data to transmit, the ONU shall suppress the entire upstream burst, i.e., for the given grant (burst), it does not turn on the optical transmitter at all.

The power saving mechanism (TX-mode) relies on the ONU's upstream burst suppression feature (see <TBD>).

8.4.4.3 Overlapping polling grants

The overlapping polling grants are grants issued to different ONUs and scheduled such that the upstream bursts from these ONUs are to arrive to the OLT at the same time, i.e., all the GATE MPCPDUs sent to different ONUs have the same `StartTime` value.

Such overlapping grants are issued only to ONUs that have been remained idle for a certain amount of time. The idle duration threshold is implementation-dependent. This feature relies on the fact that it is very unlikely for multiple idle ONUs to become active during the same polling cycle.

These overlapping polling grants shall have the PLID `ForceReport` flag set to zero, which will cause idle ONUs to suppress their upstream transmissions.

In the unlikely event that more than one ONU became active in the same polling cycle, the OLT detects a collision when several simultaneous PLID envelopes reach the OLT's PON port at the scheduled `StartTime`. The collision is assumed if the OLT detects the receive optical power, but is unable to recognize the start of burst delimiter (SBD) at the scheduled time. Or the OLT may be able to find the delimiter, but encounter an uncorrectable FEC codeword after it. An absence of any optical power at the scheduled time `StartTime` indicates that all polled ONUs remain idle.

In case the collision is detected, the OLT shall stop issuing the overlapping polling grants and instead poll all idle ONUs individually as it normally does.

The overlapping polling grants feature allows conservation of upstream bandwidth that would otherwise be consumed by the multiple polling grants. This feature is optional for the OLT to support.

8.4.4.4 Shared GATE MPCPDUs

The overlapping polling grants feature described above does not reduce the overhead caused by the polling GATE MPCPDUs in the downstream direction (although this overhead is significantly smaller than the upstream polling overhead). To reduce the downstream polling overhead, the OLT may optionally use the shared GATE MPCPDUs.

This feature requires a multicast PLID to be provisioned for a set of ONUs (see 7.4.2.1.1). A GATE MPCPDU that is delivered to a set of ONUs over a multicast PLID is referred to as a *shared GATE.*

A shared GATE shall include envelope allocation only for the multicast PLID that has been provisioned into each ONU in a given multicast group.

Shared GATEs are only employed in situations where there is low probability of a response by any ONU in the multicast group. If the OLT detects a collision in a polling grant, it shall stop issuing the shared GATE MPCPDUs and it should handle the collision as described in 8.4.4.3.

8.5 Group Logical Links

<TBD>

8.5.1 The concept of GLID

<TBD>

8.5.2 GLID service policies

<TBD>

8.5.3 GLID provisioning

<TBD>

8.5.4 Reporting and Granting a GLID

<TBD>