Contents

8 Bandwidth allocation mechanisms

8.1 Introduction

8.2 Principles of bandwidth allocation in Nx25G-EPON

8.3 Downstream transmission

8.4 Upstream transmission

8.5 Group Logical Links

8.5.1 The concept of GLID

In a Nx25G-EPON system with a very high number of logical links, the scheduling resources required at the OLT can be quite high. To ease the OLT's burden, the Nx25G-EPON system may optionally support consolidation of several LLIDs into arbitrary groups using the Group Link ID (GLID). Using the GLIDs allows the traffic scheduling functionality and resource allocation to be partially shifted from the OLT to ONUs.

GLID is a special type of LLID that is used only for the purposes of bandwidth granting by the OLT and reporting by the ONU. The scheduler located at the OLT treats the GLID as a single scheduled element, same as any other traffic-bearing LLID (PLID, MLID, or ULID). The OLT scheduler allocates envelopes to the GLID and may request ONUs to report GLID queue status.

At the ONU, the bandwidth granted to a GLID is distributed among the LLIDs that are members of the group identified by the given GLID. The methods by which the granted bandwidth is distributed among the group member are described in 8.5.4. The GLID report contains the sum of all queue lengths of member LLIDs from that ONU.

There may be various strategies for grouping LLIDs into a single scheduled element. For example, all LLIDs for a specific subscriber hosted on a multi-subscriber ONU could be grouped together into a single GLID; in another example all LLIDs supporting a specific traffic class (e.g., best-effort traffic) could be grouped together.

GLID values are used only for the purposes of bandwidth granting by the OLT and reporting by the ONU. The actual envelope transmission is identified by a PLID, an MLID, or a ULID value, associated with a specific MAC instance that sourced the data (i.e., the LLID field in the envelope headers may only contain a PLID, MLID, or ULID, but never a GLID).

8.5.2 GLID querying and provisioning

The ONU's capabilities for supporting GLIDs are queried using the read-only attribute *aOnuLlidCapability* (see 14.4.1.7). The *sGroups* sub-attribute of the above attribute returns the maximum number of GLIDs that may be provisioned into the queried ONU. If the ONU does not support GLIDs, it shall report *sGroups* value of zero.

The *aOnuLlidCapability* attribute also reports the maximum number of member LLIDs that each GLID is able to support (using the *sGlidMaxSize* sub-attribute) and the supported GLID scheduling policies (using the *sGlidPolicy* sub-attribute).

GLID entities may be created and destroyed dynamically by the NMS using the eOAM action *acConfigGlid* (see 14.6.2.20). After the GLID entity was created, the NMS may add the traffic-bearing LLIDs as members of this group. The member LLIDs are added and deleted using the eOAM action *acConfigGlidMember* (see 14.6.2.11). The action of deleting a GLID entity neither deletes the LLIDs that are members of this GLID, nor it affects any data stored in the LLID's queues.

The NMS may query the ONU for the currently provisioned GLIDs using the *aGlidType* attribute (see 14.4.2.19). The current membership of each GLID can be queried using the *aGlidMembership* attribute defined in 14.4.2.20.

8.5.3 Reporting and Granting a GLID

At the OLT, a GLID is treated as a regular scheduled element, same as any other bidirectional LLID (i.e., an LLID that gets scheduled for upstream transmission). The OLT scheduler allocates upstream bandwidth to a GLID by placing `EnvAlloc` structures for this GLID into GATE MPCPDUs (see 8.4.1).

The format of `EnvAlloc` structure is shown in IEEE Std 802.3, 144.3.6.1. The `EnvAlloc` structure assigned to a GLID contains the GLID value in the `LLID` field. The `EnvLength` represents the total length of the transmission allocated to GLID members. The ONU further sub-divides this envelope length among the GLID members and each GLID member transmits its own envelope in the upstream. The sum of lengths of all envelopes transmitted by the GLID members, including the respective envelope headers, shall not exceed the `EnvLength` value allocated to this GLID.

The GLID `EnvAlloc` structure also includes the `ForceReport` and `Fragmentation` flags. As explained in 8.4.1.4, the asserted `ForceReport` flag causes the ONU to report the length of the upstream queue of corresponding LLID. For the GLID, the ONU shall report the sum of upstream queue lengths of all LLIDs that are members of the given GLID at the moment of REPORT MPCPDU generation. If the sum of queue lengths exceeds the maximum reportable value of 2^{24} EQ (approximately, 134 Mbytes), the ONU shall report the value of 2^{24} EQs.

The effects of the `Fragmentation` flag on the upstream transmission for traffic-bearing LLIDs are explained in 8.4.3.2. The `Fragmentation` flag for the GLID is applied to each GLID member LLID, but the exact fragmentation behavior depends on the provisioned GLID scheduling policy, as detailed in 8.5.4.

8.5.3.1 Processing of GLID `EnvAlloc` at the ONU

The GATE MPCPDUs are received and processed at the ONU Multipoint Control Protocol (MPCP) entity of the MAC Control sublayer (see IEEE Std. 802.3, 144.3.2 and Figure 144-10). Figure 8-9 below illustrates a similar functional diagram, but here the emphasis is made on the flow of bandwidth-allocating information.

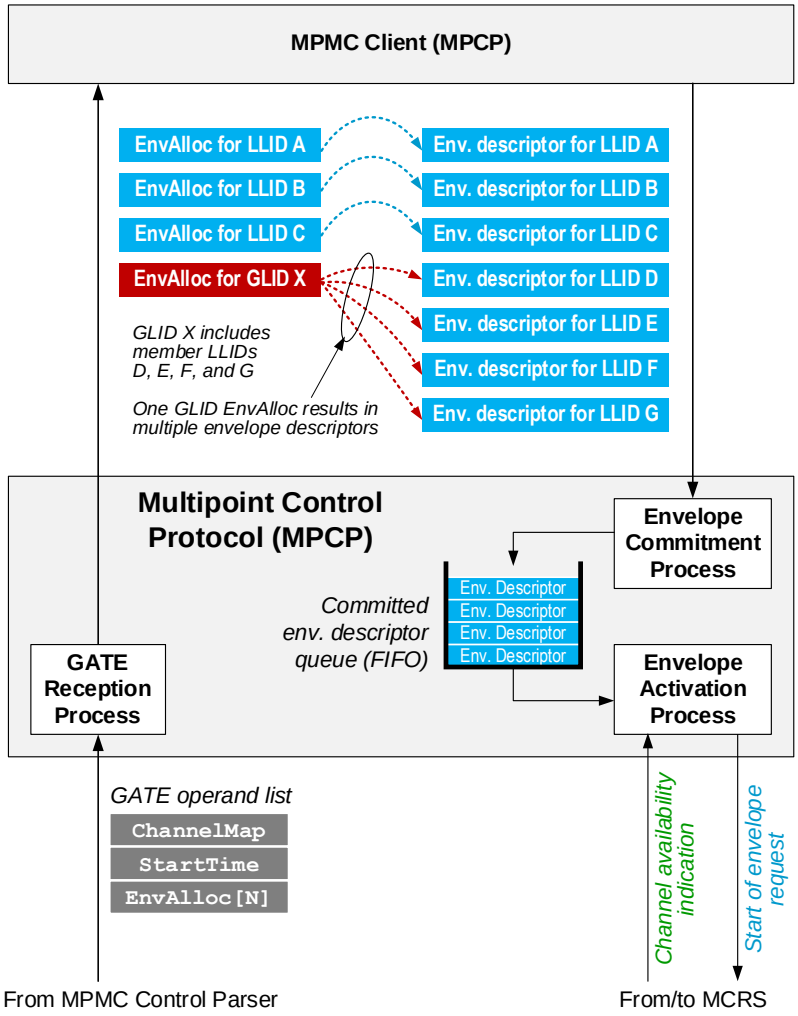

Figure 8-9 – OMU MPCP functional block diagram and flow of bandwidth-allocating information

Upon reception of the GATE MPCPDU, the ONU verifies its validity in the GATE Reception process. If the GATE is valid (i.e., it is not late and allocates transmission on enabled channels), the list of envelope allocations is passed to the MPCP Client.

Based on the envelope allocations received from the Gate Reception process and the state of local upstream queues, the MPCP client generates a set of *envelope descriptors*. An envelope descriptor is a structure that includes a value of a traffic-bearing LLID, the start time of the allocated envelope, and the transmission length (in units of EQ). The MPCP client shall generate the envelope descriptors not earlier than the grant cut-off time, as defined in 8.4.1.6.

In general, each `EnvAlloc` for a regular traffic-bearing LLID produces a single envelope descriptor. According to the IEEE 802.3, 144.3.1.2, the ONU generates the upstream envelope descriptors following the order of corresponding `EnvAlloc` structures in each GATE MPCPDU. However, for the `EnvAlloc` structure assigned to a GLID, the MPCP client may produce multiple envelope descriptors: one for each non-idle member LLID. The relative order of these envelope descriptors is determined by member LLID priority if the GLID scheduling policy is priority-based (see 8.5.4.1 and 8.5.4.2). Otherwise the order of envelope descriptors assigned to member LLIDs of a given GLID is implementation-specific.

The envelope descriptors are passed to the Envelope Commitment process that queues them in a FIFO queue. There is an independent envelope descriptor FIFO per each enabled upstream channel and each envelope descriptor applies to a single channel only. Envelopes are activated, one at a time, when the MCRS signals to the Envelope Activation process that the corresponding upstream channel became available.

8.5.4 GLID service policies

GLID service policy determines the mechanism for sharing the bandwidth granted to a GLID among the GLID's member LLIDs. The NMS provisions the GLID service policy at the time of GLID creation (see 14.6.2.20). Once provisioned, the service policy for the given GLID cannot change. However, the same set of LLIDs can be made the members of two or more GLIDs, each with a different service policy.

The service policy defines the overall principle of sharing the GLID bandwidth among the member LLIDs. It does not prescribe a specific scheduling or bandwidth-allocating algorithm. Typically, product developers are able to choose a scheduling algorithm from a family of algorithms related to the given policy.

This standard defines four GLID service policies:

— Priority scheduling (see 8.5.4.1)

— Strict priority scheduling (see 8.5.4.2)

— EQ-based proportional (weighted) sharing (see 8.5.4.3)

— Frame-based proportional (weighted) sharing (see 8.5.4.4)

GLID service policy affects the fragmentation of the upstream traffic at the ONU.

8.5.4.1 Priority scheduling

The priority scheduling policy serves a higher-priority LLID to exhaustion before serving any lower-priority LLIDs. Under this policy, the envelope descriptors are committed after the grant cut-off time, but before the grant start time. The grant cut-off time preceeds the grant start time by approximately 16 μs (see 8.4.1.6). This policy is not strict: frames of any priority that arrive after the envelope descriptors were committed are deferred until the next grant.

Besides its implementation simplicity, this policy has an advantage of minimizing the number of fragments because higher-priority LLIDs are served first and cannot be preempted by any later LLIDs. If the `Fragmentation` flag in the corresponding GLID `EnvAlloc` is set to one, the GLID grant shall contain at most one new fragment, which would be in the last frame in the GLID grant. If the `Fragmentation` flag is set to zero, no new frames shall be fragmented under this policy. A frame that doesn't entirely fit in the remainder of GLID grant is deferred until the next grant, leaving the remainder in the current grant unused.

The major disadvantage of this policy is that the lower-priority LLIDs may consistently starve if all the bandwidth allocated to the GLID is consumed by its higher-priority members.

8.5.4.2 Strict Priority scheduling

This policy is similar to the (non-strict) priority scheduling described in 8.5.4.1, but with a notable distinction: a later-arriving higher-priority frame preempts the transmission of the lower-priority LLID. The exact moment of such preemption depends on the value of the `Fragmentation` flag in the GLID `EnvAlloc`. If the flag is set to one, the preemption may happen on any EQ boundary, leaving a fragment of the preempted frame waiting for another transmission opportunity (either later in the same grant or in a

future grant). If the `Fragmentation` flag was set to 0, the current frame shall be allowed to complete its transmission before the higher priority frame starts its transmission.

Like the non-strict priority policy above, this policy also suffers from potential starvation of lower-priority LLIDs. In addition, this policy allows a large number of fragments to be created within the GLID grant when the fragmentation is enabled (i.e., when `Fragmentation` flag = 1).

8.5.4.3 EQ-based proportional (weighted) sharing

The EQ-based proportional (weighted) sharing scheduler allocates envelopes to member LLIDs in proportion of their weights. Taking the length of the GLID envelope allocation to be equal to *L* EQs, the proportional (fair) share target for $LLID_i$ is equal to

$$Target_i = L \times \frac{w_i}{\sum_{all\ non-idle\ members} w} \ , \qquad \text{(Eq.1)}$$

where w_i is the weight assigned to $LLID_i$.

This scheduler is work-conserving: The ONU shall not leave any portions of the GLID grant unutilized if there is any data (frames or frame fragments) available in any of the upstream queues belonging to member LLIDs. If the length Q_i of the upstream queue of the $LLID_i$ is less than the target value $Target_i$, the transmission envelope length L_i for the $LLID_i$ is set to Q_i plus 1 EQ for the envelope start header (ESH). The surplus bandwidth left by $LLID_i$ ($Target_i - L_i$) is proportionally divided among the remaining busy LLIDs. The work-conservation principle may necessitate an iterative approach to calculating the member LLID's fair shares.

This policy achieves the accurate bandwidth target on a shorter timescales, but in a situation when the fragmentation is allowed, it produces a large number of fragments. In extreme case, a single GLID grant may introduce a new fragment for every member LLID.

When fragmentation is disallowed, the calculated target envelope lengths of member LLIDs are adjusted up or down to nearest frame boundaries. The mechanism of this adjustment is described in 8.5.4.4.

8.5.4.4 Frame-based proportional (weighted) sharing

The frame-based proportional (weighted) sharing scheduler calculates the fair share of bandwidth member LLIDs in proportion of their weights. This calculated fair share target for each LLID is then adjusted up or down to the nearest frame boundaries.

Typically, frame-based (or packet-based) proportional schedulers rely on so-called *deficit counter* instantiated per each member of the group. A positive value of the counter denotes the accumulated deficit, i.e., the amount of extra bandwidth required to bring the running total up to the calculated target (i.e., the fair share) level. A negative value of the counter denotes the overage (i.e., the bandwidth consumed in excess of the fair share).

The target bandwidth for $LLID_i$ in each grant is calculated as shown in Eq 1. If the length Q_i of the upstream queue of the $LLID_i$ is less than or equal to the target value $Target_i$, the entire LLID's queue is served and the deficit counter D_i of the $LLID_i$ is left unchanged. If the length Q_i of the $LLID_i$ queue is greater than the $Target_i$, the envelope length L_i for the $LLID_i$ is calculated as follows:

$$L_i = \begin{cases} smallest\ frame\ boundary \geq Target_i, & if\ D_i \geq 0 \\ largest\ frame\ boundary < Target_i, & if\ D_i < 0 \end{cases} \qquad \text{(Eq 2)}$$

After the L_i is calculated, the deficit counter value D_i is updated for the next cycle:

$$D_i = D_i + Target_i - L_i \qquad \text{(Eq.3)}$$

The implementation may also add other enhancements, such as serving LLIDs in decreasing order of their deficit counters.

If the fragmentation is allowed, this scheduling policy produces at most a single new fragment at the end of the grant. If fragmentation is disallowed, no new fragments are added, but the grant may have an unused remainder at the end.