The flexibility of the GSE packets scheduling over Base band frames allows GSE to have no impact on the MODCOD selection for Base band frames: GSE packets containing the fragments of a PDU do not require to be sent in consecutive Base band frames, the selection of the MODCOD of a frame may therefore be independent of the previous frame.

To optimise as much as possible the capacity gain in ACM/VCM context, a smart placement of GSE packets in Base Band frames should be performed. Hence several functions should be performed jointly: PDU fragmentation, GSE encapsulation, GSE packets scheduling and MODCOD selection of Base band frames.

This clause proposes an informative example of a generic implementation for the Generic Stream Encapsulator. This encapsulator processes IPv4 PDUs, is defined to support ACM operations and generates one generic stream. Its high-level functional architecture is described below.

In this example architecture, two topmost units are defined to retrieve, slice, adapt and transmit the incoming PDUs into a proper sequence of GSE packets (called here EPU for Encapsulated Packet Unit). They are the **PDU Manager** and the **EPU Manager**. The joint element between them is a set of **Scheduler Queues** able to store all received PDU packets.

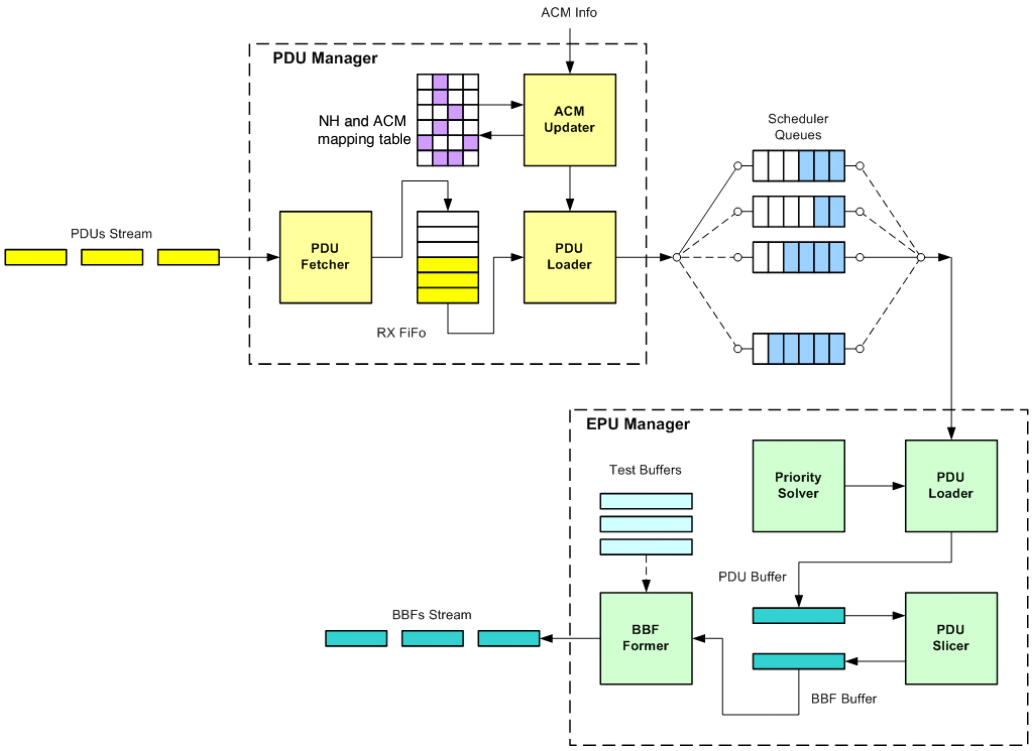


Figure 9: High level scheme of GSE Encapsulator (ACM mode support)

The **PDU Manager** is composed of the following sub-units:

- **PDU Fetcher** - this subunit makes a direct connection with the LAN backbone, loads the incoming packets and stores them inside a RXFIFO.
- **PDU Loader** - the PDU Loader gets recursively packets from the RXFIFO and performs header analysis to check if packets must be processed or not. Three types of conditions have to be satisfied simultaneously to mark the current PDU as "ready for processing":
  - Version matching (IPv4)
  - Protocol matching (TCP, UDP and ICMP check)
  - Destination address vs. Next-hop and ACM mapping table matching (mapping destination IP address to the next-hop neighbour)

After analysis, each packet is put inside a proper scheduler queue. The queue's choice is made by the **ACM Updater** following the mapping rules that the internal "**Next-Hop and ACM mapping Table**" supplies.

- **Next-Hop and ACM mapping Table** - this internal table gives for each destination IP address the IP address of the corresponding Next-hop neighbour, its GSE Label and the appropriate MODCOD.

- **ACM Updater** - this subunit loads periodically ACM Information (from an ACM Manager) indicating for each next-hop IP address the current transmission conditions, and updates the Next-Hop and ACM mapping table accordingly. In addition, this subunit uses the Destination IP Address that the PDU Loader supplies to perform a key-searching on the Next-hop and ACM mapping table in order to associate to the current packet:

  - the next-hop IP address and GSE label (MAC/NPA address);

  - the appropriate MODCOD.

The **EPU Manager** is composed of the following subunits:

- **PDU Loader** - this subunit retrieves the PDU packets from the Scheduler queues using the criteria that the **Priority Solver** supplies, and inserts them in the PDU Buffer.

- **PDU Slicer** - this subunit fragments PDU packets if necessary and encapsulates them into a suitable sequence of GSE packets that best fits the available base band frame payload.

- **Base Band Frame (BBF) Former** - this subunit retrieves from the base band frame (BBF) Buffer the current frame and finalizes it to make possible its transmission as next base band frame.

Implementations must allow to not send the remaining fragment of a fragmented PDU in the next base band frame (allowing this way that the MODCOD for this frame is independent of the previous frame's MODCOD and that a PDU with higher priority is sent in this new frame). A possible solution for this is to push back the fragment from the PDU buffer in its original FIFO.

Scheduling enforces specific priority rules to select PDU packets to send in the next Base band frame and the MODCOD of this frame. They can be based on the combination of different criteria, e.g. MODCOD efficiency, throughput optimisation, etc. They should also take into account the different QoS classes managed by the system (the scheduling should be driven by the system traffic priorities in order not to change the overall Quality of Service).

The example scheduling implementation in the proposed encapsulator is based on a set of scheduler queues reflecting the applied scheduling granularity, i.e. per MODCOD and per QoS, and on the Priority Solver unit, which implements the scheduling algorithm and selects the scheduler queues from which packets will be extracted to fill the base band frames.

The example GSE encapsulator, based on the DiffServ approach, takes into account the following classes of service (the IntServ approach could also be considered):

- Expedited Forwarding traffic (EF)

- Assured Forwarding traffic (AF)

- Best Effort traffic (BE)

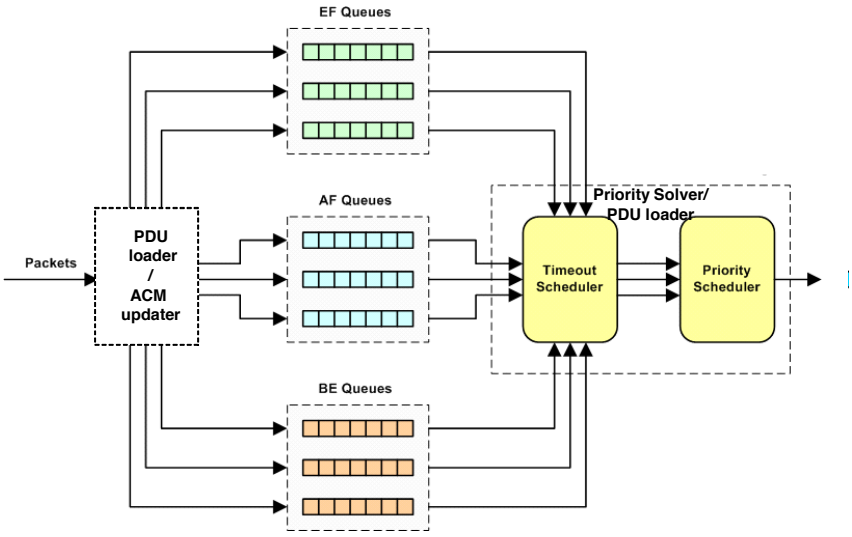
The scheduler queues, also called ACM FIFO queues, are thus divided by priority level (EF, AF and BE), and each priority level contains several FIFO queues, each one being associated with a different ACM mode (figure 10).

The Scheduler operations are as follows:

- When a PDU packet comes from the upper layer, it is classified firstly by priority using its Differentiated Services Code Point (DSCP) marking, i.e. QoS, secondly by required protection mode, i.e. MODCOD, and then it is inserted in the corresponding ACM FIFO queue by the **PDU loader.**

- When a base band frame has to be filled for the next transmission, a specific algorithm is applied by the **Priority Solver** to select the ACM FIFO queue from which packets will be extracted to fill the base band frame.

- The protection mode of the base band frame is determined by the MODCOD of the packets selected during these operations.



NOTE: The logical functionalities "Timeout Scheduler" and "Priority Scheduler" of the Priority Solver are linked to the example priority policy described hereafter.

Figure 10: Scheduler queues for ACM and QoS support

In the example implementation, if the extracted packets cannot fill completely the base band frame payload, then other packets are fetched from the FIFO queues in order to avoid padding. At this regard it is worth noting that a packet can be aggregated only if it requires a protection level lower or equal to the one of the selected base band frame format. If it is not the case, an alternative solution is to modify the MODCOD and the format of the base band frame according to the protection required by this packet.

Different priority policies may be applied on this example architecture.

The following example is based on the use of timeouts related to the waiting time of the first packet in each FIFO (specific timeouts are implemented for each ACM queue, which are controlled separately by the Timeout scheduler) and on a packet prioritisation preferring the highest priority (QoS) packets and among the highest priority packets, packets with the most efficient MODCODs. The algorithm applied when a base band frame has to be filled is described hereafter:

1) The Priority Solver checks if there is one EF packet that is waiting in an EF-ACM queue for a time longer than a certain timeout value. If there is, packets from that EF-ACM queue are selected for the BB-Frame Buffer filling.

2) If no timeout occurs, then the Priority Solver checks if there is one AF packet that is waiting in an AF-ACM queue for a time longer than a certain timeout value (different from the timeout value of EF-ACM queues). If there is, packets from that AF-ACM queue are selected for the BB-Frame Buffer filling.

3) If no timeout occurs, then the Priority Solver checks if there is one BE packet that is waiting in an BE-ACM queue for a time longer than a certain timeout value (different from the timeout value of EF and AF-ACM queues). If there is, packets from that BE-ACM queue are selected for the BB-Frame Buffer filling.

4) If neither of previous conditions is met, then the Priority Solver considers EF-ACM queues and selects packets from the EF-ACM queue having the most efficient MODCOD respect to the least efficient ones.

5) If EF-ACM queues contain no packets, same as point (4) but performed on AF-ACM queues.

6) If AF-ACM queues contain no packets, same as point (4) but performed on BE-ACM queues.

In figure 11, a graphic representation of the priority policy based on the points 4 - 5 - 6 of the algorithm is shown.

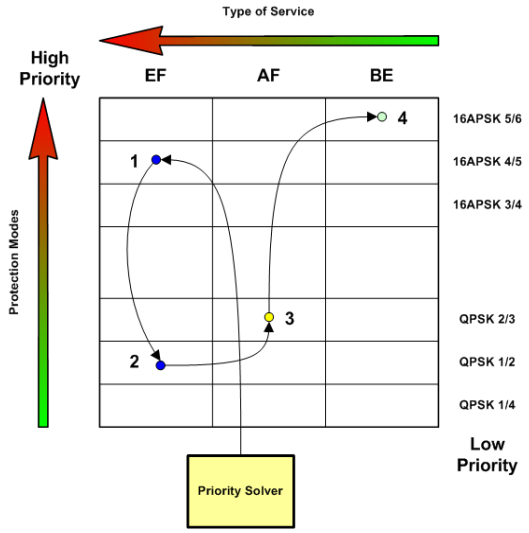


Figure 11: Example of Priority policy for GSE scheduling

Other algorithms may be based on the attribution of a score to each queue, taking into account different criteria such as its QoS priority (EF>AF>BE), its filling level, the latency of first packet to send, etc.

6.4 Receiver

This clause presents an example implementation of the Generic Stream (GSE) Decapsulator (processing one generic stream). Its functional blocks and high-level features are described in figure 12.

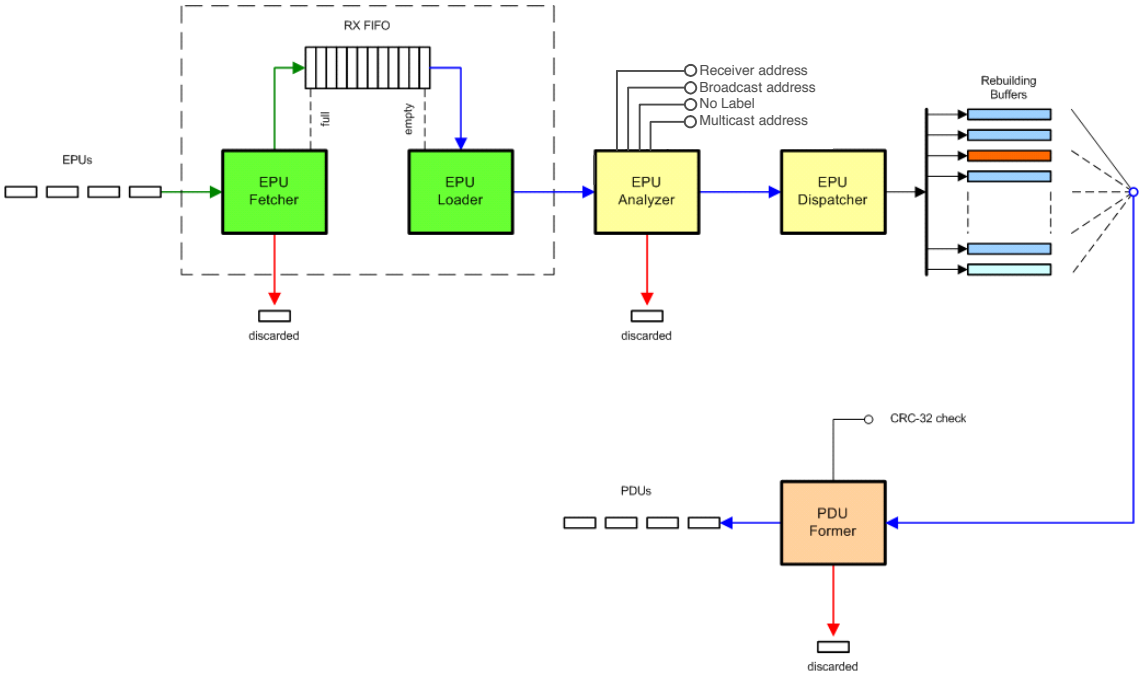


Figure 12: High level scheme of GSE Decapsulator

- **EPU Fetcher** - this subunit is connected to the base band frame Decoder output and loads all the incoming GSE packets. A RXFIFO check is done constantly and when a "Full State" is detected, the EPU Fetcher discards the packets that cannot be stored and processed. An alert is then sent to the decapsulator controller.
- **EPU Loader** - this subunit loads GSE packets and sends them to the EPU Analyzer.