61.0.0.1 PHY PMI Aggregation Receive function

The PHY PMI aggregation receive function requires per-PMI queues as well as a per-MAC packet buffer for fragment reassembly. The algorithm assumes only “good” fragments are placed on the per-loop receive queues ("bad" fragments are discarded according to the rules in 61.2.2.5).

During initial bring-up and in the event of certain errors, the receive algorithm has to determine which sequence number is expected next (expectedFragmentSequenceNumber). When the link state is changed to UP, the expected sequence number is unknown and no frame sequence errors shall be recorded. As fragments are received, expectedFragmentSequenceNumber is initialized to the smallest sequence number of fragments at the head of per-PMI queues when either all active queues are non-empty or at least one queue has been non-empty for 64,000 bit times.

In addition to the expected sequence number, it is necessary to determine the next sequence number (nextFragmentSequenceNumber). This is defined as the smallest sequence number of fragments at the head of per-PMI queues.Note that the sequence number rolls over after it reaches the maximum value.

The receive function then executes the following algorithm:

a) Determine the next sequence number via the preceeding algorithm
b) If the next sequence number is equal to the expected sequence number, process that fragment . Otherwise wait for that condition or follow the error handling rules described in 61.2.2.5.
c) Accept the fragment into the packet buffer. If that fragment is an end-of-packet, pass the packet to the MAC-PHY Rate Matching layer.
d) Increment the expected sequence number
e) Repeat processing.

The PMD control of aggregated links must ensure that the maximum latency difference between any two aggregated links correponds to no more than 64,000 bit times. This must be achieved by adjusting the bit rate, error correction and interleaving functions in the PMA/PMD of each link. Note that the burst noise protection offered by the error correction and interleaving functions is directly proportional to the latency, therefore it is logical that multiple aggregated links in the same environment should be optimized to have the similar latencies.