4

TRANSMISSION CONVERGENCE LAYER

UNCORRECTED PROOF

In this chapter:

- *ONU-ID terminology ambiguity*
- *Payload mapping and framing*
- *FEC, scrambling and encryption*
- *ONU discovery, ranging, and delay compensation*
- *Conveying time of day to the ONU*
- *Physical layer OAM (PLOAM) messages; an embedded operations channel whose details appear in Appendix II*
- *Methods by which upstream bandwidth demand is estimated and capacity is allocated*
- *Energy conservation*
- *Security*

The transmission convergence (TC) layer is the heart and soul of a PON and, in particular, of a G-PON or an XG-PON. Here is where we find solutions to the unique problems posed by a PON, as well as to the ordinary issues of any network protocol. These unique aspects are largely related to the tree structure, the point to multipoint nature of the PON:

- Discovering hitherto unknown ONUs in a way that does not disrupt existing traffic on the PON

Gigabit-capable Passive Optical Networks, First Edition. Dave Hood and Elmar Trojer.

© 2012 John Wiley & Sons, Inc. Published 2012 by John Wiley & Sons, Inc.

- Recovering previously known ONUs onto the PON after power failures, power down, or other disruptions
- Coping with the fact that the various ONUs are at different distances from the OLT and, hence, have different propagation delays
- Accommodating drift in the propagation delay that could be caused, for example, by daily or seasonal temperature changes in the ODN
- Defining the upstream burst header for fast and accurate optical receiver calibration so that the header can be quickly delimited and parsed
- Orchestrating upstream burst transmit timing among ONUs, such that
  (a) No two ONU transmissions collide at the OLT
  (b) The ONUs' varying traffic levels and service-level commitments are honored
- Defining the structure of a payload mapping to preserve efficiency even if upstream payload frames do not fit exactly into the size of the burst
- Protecting the PON from malicious ONUs that might attempt to steal service or invade the privacy of subscriber traffic
- Detecting and diagnosing rogue ONUs, that is, defective ONUs that transmit light at unauthorized times and potentially interfere with normal traffic
- Structuring payload for easy identification of individual flows and grouping of flows into separately manageable traffic classes

Several other topics are addressed by the transmission convergence layer, topics that are not unique to PONs, but are of general interest to any access network:

- Securing the association of OLT and ONU from eavesdropping, tampering, and theft of service
- Adapting (mapping) some set of payload protocols to the transmission medium. The usual G-PON payload mapping is to an Ethernet client layer, but an MPLS mapping is also defined. G-PON also includes a mapping for OMCI, the PON management protocol.
- Measuring the quality of the link, either in terms of optical parameters or in terms of error rate
- Improving the intrinsic quality of the optical link through FEC
- Providing signaling and control channels for the PON link itself, ranging from simple bit-oriented signaling to intermediate-level fast messaging—called PLOAM[*] in G-PON (physical layer operations and maintenance)
- Supporting an embedded high-level operations and management communications channel, OMCC
- Taking steps to actively reduce power demand during periods of light use, with little or no service degradation

[*] Appendix II defines the PLOAM message sets in detail.

UNCORRECTED PROOF

ONU-ID Terminology Ambiguity

It is important to understand that ONU-ID is an ambiguous term.

At the transmission convergence and PLOAM level, where we are in this chapter, ONU-ID is expressed in the form of real bits in real registers and would be visible, for example, to a snooping device on the PON. Every ONU that has progressed past the first phases of initialization on the PON has a TC-layer ONU-ID.

At the provisioning and management level, ONU-ID is a name, a reference. Service orders, installation orders, maintenance, and provisioning refer to the management-level ONU-ID.

An ONU may be fully operational at the TC level but has no ONU-ID at the management level, for example, if it is autodiscovered and has not been provisioned for subscriber service. Likewise, a preprovisioned management-level ONU-ID is valid and meaningful, although it may correspond to nothing at the TC level, for example, if its referent ONU has not yet been installed.

The management-level ONU-ID is usually hierarchical:

```
<network element name>
       <slot number>
              <PON port number>
                   <ONU number>
```

whereas the TC-layer ONU-ID exists in a flat name space {0..253} for G.984 G-PON, {0..1023} for G.987 XG-PON. Observe that the range of TC-layer ONU-IDs includes the value 0. At the management level, the final <ONU number> field almost surely starts with value 1.

The management-level ONU-ID is normally static for the lifetime of a customer subscription, possibly for many years. The TC-layer ONU-ID may be different every time the ONU initializes onto the PON.

An ONU that has been assigned a TC-layer ONU-ID by the OLT is said to be registered on the PON. Once registered, an ONU's TC-layer ONU-ID cannot be reassigned. If it is desired to register the TC-layer ONU-ID with a value equal to the management-level <ONU number>, the OLT may deactivate the ONU, which causes it to reregister. Recognizing the ONU's serial number on the second pass, the OLT can then assign it a different TC-layer ONU-ID. However, it is simpler for the OLT to maintain a mapping between the TC-layer ONU-ID and the management ONU-ID.

In most cases throughout this book, the interpretation of ONU-ID is apparent from the context. When there is potential for confusion, we disambiguate the identifiers as TC-layer ONU-ID versus management- or OMCI-level ONU-ID. In this chapter, ONU-ID is understood to be at the TC layer unless explicitly stated otherwise.

UNCORRECTED PROOF

One way to think of the difference is that a protocol analyzer would see the TC-layer ONU-ID, never the management ONU-ID. In contrast, a management view would never see the TC-layer ONU-ID except perhaps in a troubleshooting or debugging context.

4.1 FRAMING

G-PON uses straightforward TDM in the downstream direction. The OLT broadcasts all downstream traffic onto the fiber; each ONU selects its own particular traffic in ways described in this chapter. At the physical layer, downstream traffic comprises an unbroken stream of fixed-length 125-μs frames.

In the upstream direction, the OLT authorizes each ONU to transmit bursts of traffic (time division multiple access, TDMA) in such a way that:

- Upstream bursts do not collide at the OLT. We discuss timing compensation in Section 4.3.
- Bursts are sized and spaced appropriately for the traffic and the quality of service committed to each ONU. Section 6.3 covers traffic management.

The upstream flow retains the concept of a 125-μs frame, but it simply marks a repetitive series of ticks that serve as convenient reference points.

Depending on the line rate, different amounts of data can be fit into the physical layer frame. Table 4.1 shows the line rates and physical layer frame sizes for G-PON and XG-PON.

Be aware that the term *frame* is overloaded. Figure 4.1 illustrates several uses of the term. In general, a frame has a header section of some kind, a payload, and sometimes a trailer. We discuss the various kinds of frame in detail throughout this section:

- Service data units (SDUs) are often called frames, sometimes packets. In this book, particularly for Ethernet, we usually call them frames.
- Each SDU is encapsulated into a GEM or XGEM[*] frame for transport over the PON. GEM idle frames are defined for occasions when no other payload is available for transmission.

TABLE 4.1 G-PON and XG-PON Bit Rates and Physical Frame Sizes

Bit Rate, Mb/s	Frame Size, Bytes	Use
1244.16	19,440	G.984 G-PON upstream
2488.32	38,880	G.984 G-PON downstream G.987 XG-PON1 upstream
9953.28	155,520	G.987 XG-PON1 downstream

[*] Strictly speaking, the GEM pertains to G.984, while the term XGEM is proper for G.987 XG-PON. Though not identical, they are similar and serve the same purpose. We often use the term *GEM* generically to refer to either. The same is true of the acronyms GTC and XGTC.

UNCORRECTED PROOF

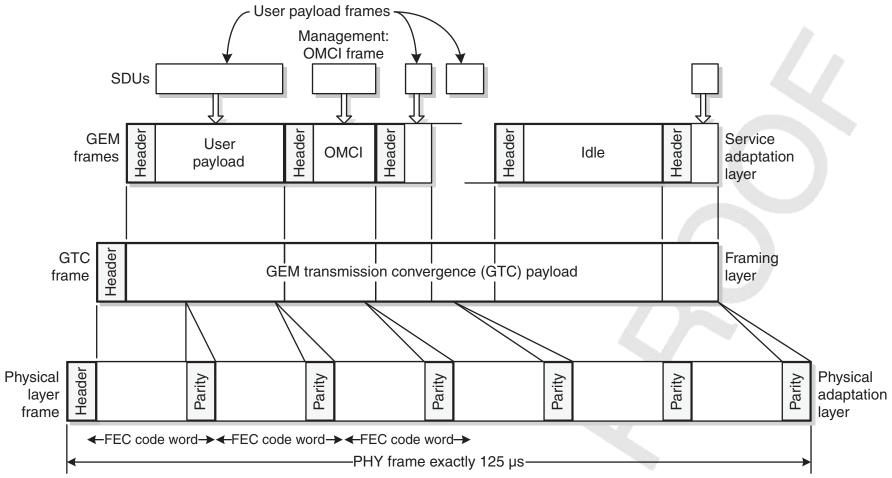

Figure 4.1 Downstream framing, XG-PON.

- A contiguous sequence of GEM frames, together with its overhead, is called a G-PON transmission convergence layer (GTC or XGTC) frame. It repeats at 125-μs intervals.
- The physical layer (PHY) frame extends the GTC frame with a PHY header and FEC. It is a continuous bit sequence at the nominal line rate of 9.953 Gb/s (G.987 XG-PON) or 2.488 Gb/s (G.984 G-PON), with no gaps. The PHY frame header repeats at 125-μs intervals. It is used for synchronization as well as for other functions.

In the upstream direction, ONUs transmit bursts, which are normally much shorter than 125 μs. Their size and timing is governed by the bandwidth map (BWmap), which is broadcast to the ONUs from the OLT; timing and addressing in the BWmap determines which ONU transmits a burst, for how long, and when, as measured from the 125-μs upstream frame boundary. Each burst comprises one or more allocations to distinct traffic aggregation entities in an ONU. At the TC layer, each allocation is identified by an alloc-ID; it largely corresponds to a traffic container (T-CONT) at the management layer. We discuss these and their differences in this chapter and in Section 6.3. With the addition of framing overhead, the series of contiguous allocations corresponds to the GTC frame of Figure 4.1.

There is thus no physically observable 125-μs frame in the upstream direction. However, there are well-defined repetitive instants that mark the conceptual boundaries of upstream frames, and the term is meaningful. Timing discussed in Section 4.3 defines the concept of the upstream frame more precisely.

The process of building a PHY frame may be understood as the combination of three layers: service adaptation, framing, and physical layer adaptation, which correspond to the three layers shown in Figure 4.1.

Downstream and upstream directions are significantly different, and G.984 G-PON and G.987 XG-PON, although clearly siblings, are by no means identical twins.

UNCORRECTED PROOF

We, therefore, discuss them separately, first G.987 XG-PON, then G.984 G-PON. But rather than four distinct expositions, we start by considering the common aspects, and then move into the differences. In the detail of getting payload to and from the PON, the mapping of payload frames into GEM frames (XGEM frames in XG-PON) differs very little.

4.1.1 GEM Framing Layer

For transport over the PON, G-PON and XG-PON must encapsulate service data units (SDUs). SDUs include user data frames and OMCI management messages. The result of the encapsulation is called a GEM (or XGEM) frame: G-PON encapsulation method. Each GEM frame is tagged with a GEM port ID, which uniquely identifies a particular flow on the PON. GEM frames and GEM ports are visible only between the OLT and ONU; they have no wider significance, either toward the subscriber or toward the core network. There are several reasons for GEM:

- *Traffic Multiplexing* Particularly in the downstream direction, the GEM port ID facilitates wire-speed filtering of the entire capacity of the PON by each ONU, which need only pick off the traffic destined for itself. A GEM port ID occupies only 12 bits (G.984) or 16 bits (G.987), a small, uniform label that is easy to look up. This also facilitates upstream demultiplexing.
- *Protocol Efficiency* In contrast to Ethernet, SDUs encapsulated in GEM frames are packed into GTC frames without additional physical layer overhead such as interpacket gaps or preamble-delimiter bytes. GEM frame delineation is accomplished by a short GEM frame header, five bytes in G.984, eight bytes in G.987.
- *Fragmentation* The encapsulation layer supports SDU fragmentation, which effectively allows maximum utilization of GTC frames. SDUs that will not fit the remaining space in a downstream frame or an upstream allocation are fragmented and transmitted across two or more PHY frames or upstream bursts. This again increases protocol efficiency.
- *SDU Agnosticism* G-PON and XG-PON are specified in such a way that they can transport a variety of networking protocols over the PON, as well as transporting OMCI, the management protocol of G-PON itself. The provisioned GEM port ID is used to distinguish one protocol type from another. Although Ethernet is the primary protocol in use today, other protocols such as SDH and native IP over GEM were standardized in G.984. G.984 even specifies TDM over GEM, though with insufficient detail to support implementation. For lack of interest, these latter three have been dropped from G.987. More recently, multiprotocol label switching (MPLS) over GEM was standardized.

4.1.1.1 GEM Frames

Figure 4.2 depicts a G.987 XGEM frame, while Figure 4.3 portrays the structure of a G.984 GEM frame.

UNCORRECTED PROOF

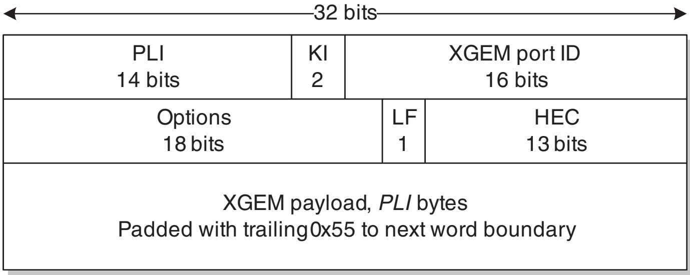
32 bits
PLI
14 bits
KI
2
XGEM port ID
16 bits
Options
18 bits
LF
1
HEC
13 bits
XGEM payload, *PLI* bytes
Padded with trailing 0x55 to next word boundary

Figure 4.2 XGEM frame, G.987 XG-PON.

Observe that the XGEM frame header contains nothing to identify the protocol of its client. G.984.3 GEM contains a payload-type indicator (PTI) field, but as we shall see, it serves this purpose only in a very minimal sense. The client protocol type is known through configuration and distinguished by GEM port ID.

Payload in G.987 XGEM frames is always a multiple of 4 bytes, called a word in XG-PON terminology. G.984 GEM frame payloads have single-byte granularity.

As we see, GEM and XGEM frames share features, but they are not the same. Let us discuss each header field, starting with the common ones.

Common Fields

GEM Port ID The GEM port identifies a traffic flow and facilitates traffic multiplexing as described above. In G.984 G-PON, GEM port IDs lie in the range 0–4095, while G.987 XG-PON port IDs range up to 65,635. In XG-PON, however, ports 0–1022 are preassigned for the OMCI GEM channels of ONUs with TC-layer ONU-IDs 0–1022, respectively, so they are not available for general use. Further, port 65,535 is reserved as the target for idle GEM frames.

PLI A payload length indicator is essential for variable-length frames. The resolution of PLI is bytes, with a range up to 4095 in G.984 G-PON and 16,383 in G.987 XG-PON. In G.984 G-PON, the GEM frame ends after exactly *PLI* bytes of payload; in XG-PON, the frame is padded with as many as 3 bytes of value 0x55 to extend it to the next word boundary, or as many as 7 pad bytes if necessary to create a minimum XGEM payload field of 8 bytes. The actual

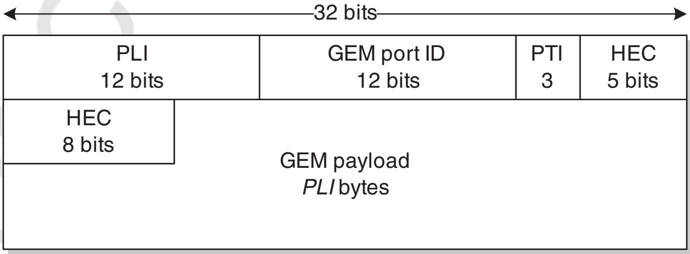

Figure 4.3 GEM frame, G.984.3 G-PON.

UNCORRECTED PROOF

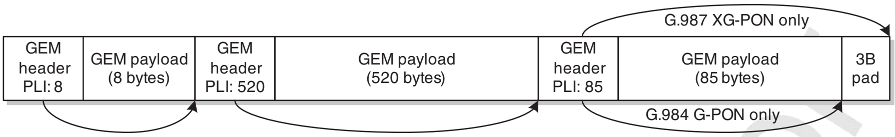

Figure 4.4 Walking from one GEM header to the next.

payload can be as short as 1 byte. No Ethernet SDU is that short—the minimum length is 64 bytes—but it is possible that GEM frame fragmentation could leave a 1-byte orphan.

HEC Starting with a well-defined synchronization instant, GEM frames are concatenated one after another into the GTC frame (in G.984.3: GTC payload). As illustrated in Figure 4.4, the decoder can only find the next GEM frame if it can properly decode the header of the current GEM frame and in particular, its payload length indicator PLI. So there is an error-checking and correcting code across the GEM frame header. In G.984 G-PON and by tradition, this is called a header error control, but since the same algorithm is used in various places in XG-PON, not always in a header, the acronym was morphed into hybrid error control or hybrid error correction. Appendix I describes HEC.

Figure 4.4 also shows how G.987 XGEM frames are padded to the next word boundary.

Walking the list of GEM frames is a convenient time for the ONU to filter downstream traffic by GEM port, extracting only GEM frames of interest for further processing within that ONU. Each GEM frame represents an individual flow and, except for multicast, therefore, goes to (or from) only one ONU.

G.984.3 describes a way to reacquire GEM frame delineation by checking 5-byte candidates in the stream of bits to see if their presumed HEC is correct.

Figure 4.5 shows the G.984 state machine that can potentially recover lost GEM frame delineation during the same GTC frame. Each byte is hypothesized as the start of a GEM frame, and the corresponding HEC is computed for that hypothesis.

Given the extremely low probability of losing GEM frame delineation, and the higher speeds involved, the HEC recovery feature was dropped from G.987. In XG-

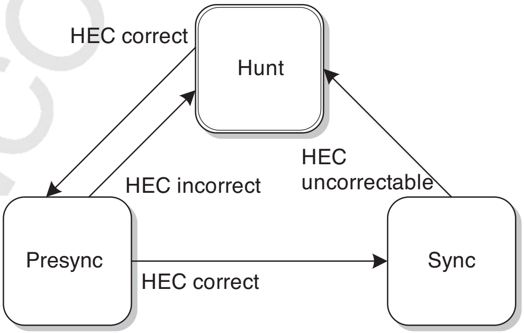

Figure 4.5 GEM frame synchronization, G.984.

UNCORRECTED PROOF

PON, if the frame header is corrupted beyond repair, the remainder of the PHY frame is lost.

Fields Unique to G.987 XG-PON

KI, the Key Index This two-bit field specifies which key, if any, is used to encrypt the current GEM frame. The value 00 indicates an unencrypted payload, and the value 11 is reserved. Values 01 and 10 select one of two possible keys. The reason for two is that during unicast key update, the OLT and ONU may have different views of the validity of a given key. Section 4.6 discusses key negotiation and synchronization in detail.

Options These 18 bits are unused in G.987 XG-PON. They exist primarily to pad the GEM frame header to a word boundary and are available for future use if needed.

LF This bit signals the last fragment of an SDU. If it is 0, the GEM frame contains only a leading or intermediate fragment of an SDU. The receiver must buffer the fragment until the next fragment arrives, and the next, and the next, until the final fragment appears, at which time it can reassemble the SDU. In practice, more than two fragments would only occur if large upstream subscriber frames were trying to squeeze through a very small bandwidth allocation. We discuss fragmentation in more detail below.

If LF = 1, the GEM frame contains either an entire SDU or the final component of a fragmented SDU.

Fields Unique to G.984 G-PON

PTI The 3-bit PTI field reflects the ATM heritage of G-PON. It contains code points to identify SDU fragmentation. In theory, it also distinguishes user traffic from OAM traffic, meaning OMCI. The definition in G.984.3 says OAM traffic comprises 48-byte ATM-like messages, but G.984.4 subsequently evolved to define an extended OMCI message set with variable-length messages. In any event, since G.984 OMCI is carried over a configured GEM port, the distinction is irrelevant. Like the vermiform appendix, the PTI field can complicate life, but not markedly improve it.

4.1.1.2 Payload Mapping to GEM Frames

Now we know what a GEM frame is. How does payload map into a GEM frame? Figure 4.6 illustrates an Ethernet frame mapped into XGEM and GEM frames, respectively.

As we see, the preamble and delimiter are stripped from the Ethernet frame, and the remainder is mapped directly into the payload of a GEM frame. Figure 4.6 shows an untagged Ethernet frame, but tagged frames are encapsulated in the same way: Everything after the start-of-frame delimiter to and including the frame check sequence is copied into the GEM frame. Keeping the Ethernet frame check sequence (FCS) in the GEM frame decouples error detection between the GEM server layer and the Ethernet client layer.

UNCORRECTED PROOF

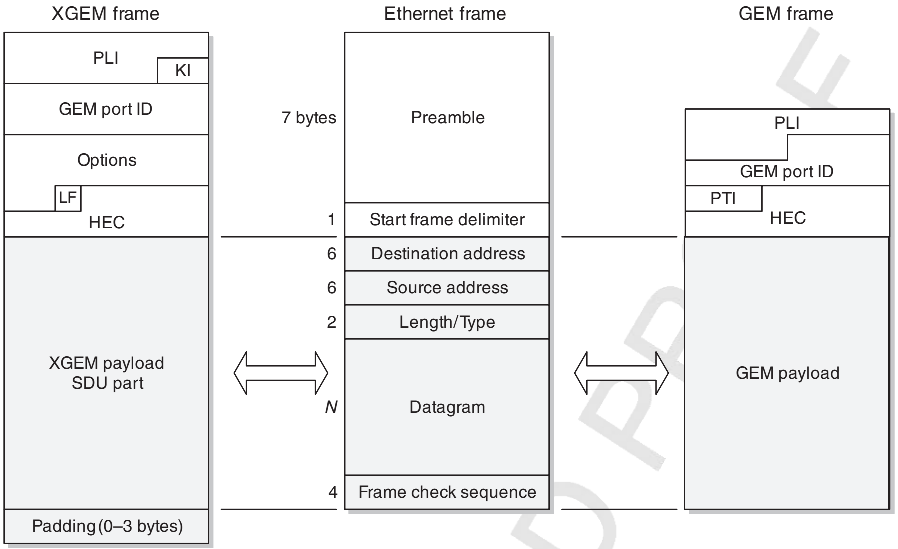

Figure 4.6 Ethernet mapping into GEM.

Shown in Figure 4.7, the MPLS mapping is much the same, except of course with none of the Ethernet frame overhead: preamble, delimiter, or FCS.

Ethernet and MPLS are the only external payload mappings defined for G.987 XG-PON. OMCI messages are also mapped into GEM and XGEM frames, also simply by prepending a GEM header, and in XGEM, possibly a trailing pad to the next word boundary. Other mappings exist for G.984 G-PON, but they have

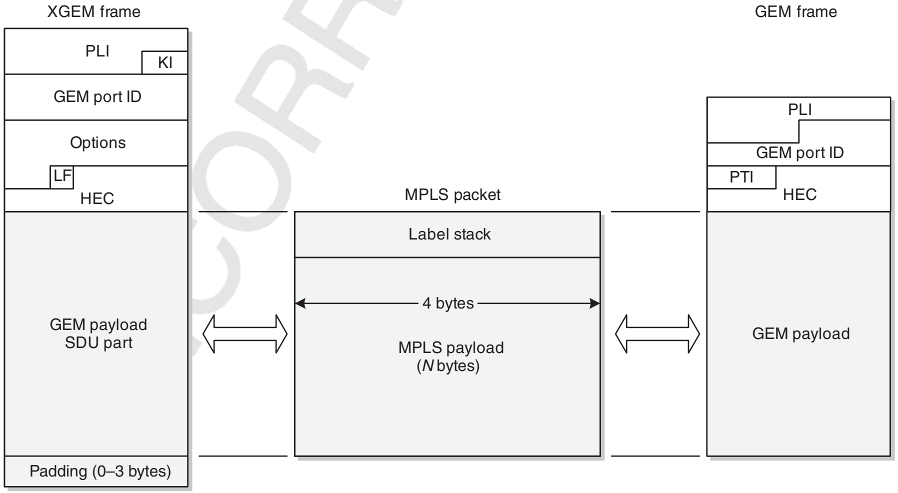

Figure 4.7 MPLS mapping into GEM.

UNCORRECTED PROOF

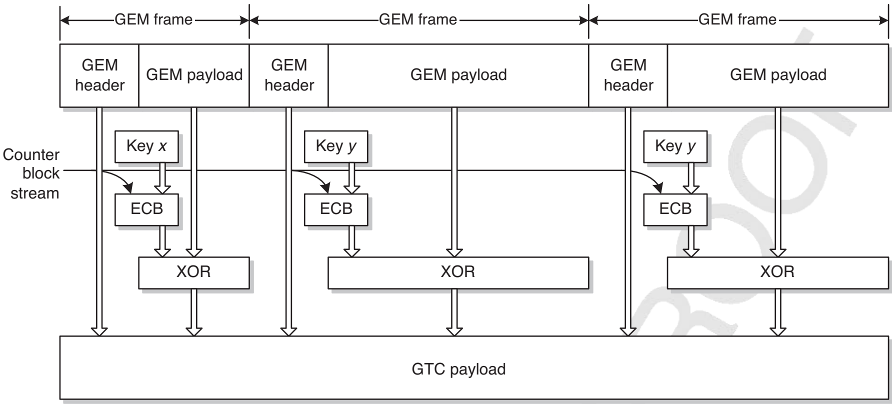

Figure 4.8 GTC payload.

not survived into G.987 because they are of little interest to the industry. We omit illustrations of these other mappings; the interested reader is referred to G.984.3.

GTC Payload and Encryption The GTC payload is the simple concatenation of GEM frames, possibly encrypted. Figure 4.8 illustrates how encryption works.

We do not encrypt the headers of individual GEM frames, nor do we encrypt the additional headers that we will ultimately add onto the structure to build the physical layer frame. The payload of each GEM frame is encrypted (or not) using the advanced encryption system (AES) algorithm in electronic code book (ECB) form. The counter block stream is a bit sequence that can readily be duplicated at both OLT and ONU, while nevertheless including enough complexity to effectively randomize the ECB look-up function:

In G.987 XG-PON, the key for each GEM frame may either be a unicast key or a broadcast key, as suggested by keys x and y in Figure 4.8.

G.984 G-PON does not support broadcast keys. If a GEM frame is encrypted, the only choice is the (unicast) key.

In both G-PON standards, encryption is specified with GEM port granularity. Encryption is sufficiently complex that we do not treat it inline here; it is fully described in Section 4.6.

Fragmentation In principle, one SDU maps into one GEM frame. A GEM frame may never contain more than one SDU, but the inverse relation is not necessarily true. There are three cases in which an SDU may be fragmented into more than one GEM frame.

- *Insufficient Runway*

UNCORRECTED PROOF

Downstream Every 125 µs without fail, the downstream flow is interrupted for a framing structure that contains synchronization and overhead. If there is not enough time to transmit a GEM frame containing the next complete SDU before the next framing overhead is due, the SDU either has to wait or the GEM layer has to fragment the SDU. We reduce packet delay and improve efficiency by choosing the fragmentation option, unless the remaining time is too short to build a useful fragment.

Upstream The situation is much the same. An ONU transmits payload according to its upstream capacity allocations. If too little time remains before the end of its allocation to transmit its next SDU, the ONU must either wait or fragment the SDU. If there is more than a negligible amount of remaining space in the allocation, we fragment the SDU.

- *Oversize Payload* If an SDU were simply too long to fit into a GEM frame (G.984 G-PON: 4095 bytes, G.987 XG-PON: 16,383 bytes), it would have to be fragmented.
- *G.984 G-PON Only* It is imagined that high-priority SDUs might preempt lower priority SDUs. This originally made a certain amount of sense because G.984 was developed with the idea that in the upstream direction, a T-CONT might contain several classes of service, so there could be priority contention between queues within a single T-CONT. However, even a 2000-byte jumbo frame can be transmitted at 1.2 Gb/s in about 13 µs. Further, the industry has moved toward a best-practice service model in which different classes of service are carried in different T-CONTs. There is no preemption between T-CONTs, even within the same ONU. Thus, preemption adds complexity for very little benefit. Preemption is not recognized in G.987.

Once we have determined that our next SDU is to be fragmented, the simple story is that the first GEM frame encapsulates as much of the SDU as will fit. In the first GEM frame header, we indicate that the frame is a fragment. The remainder of the SDU is feedstock for the next GEM frame, which might also contain only a fragment (unlikely, but possible, especially if the upstream allocation were very small). The remaining fragments are at the head of the queue for their particular allocation or (G.984) priority, so they are transmitted as soon as possible. The final GEM frame in the series indicates that it is the last fragment. The receiver must buffer GEM frame fragments until it receives the last one, whereupon it reassembles the original SDU and forwards it.

That is the simple story. Above, we said that we will fragment a frame if we have more than a negligible amount of space into which we can put a fragment. What does *negligible* mean?

In G.984 G-PON, if there remains too little time to send a minimum length GEM frame (6 bytes including GEM header), the transmitter fills the space with all 0 bytes. Otherwise, G.984 will create a fragment with as little as 1 byte of SDU payload.

UNCORRECTED PROOF

In XG-PON, if the space remaining before the break in transmission is too small to contain a minimum length GEM frame—8 bytes of payload—no fragment is created, and the transmitter fills the time with either full or short idle frames, the latter defined to be a sequence of 0 bytes.

It is worth mentioning that a sequence of GEM frames, each with 1 byte of payload, would impose considerable stress on the OLT and ONU devices, especially if encryption were active. G.984.3 does not constrain this possibility; G.987 XG-PON reduces this potential overload by specifying a minimum GEM payload length of 8 bytes. In G.987, our 1-byte fragmented SDU trailer would thus be accompanied by 7 bytes of padding.

Idle GEM Frames It can happen that the transmitter has nothing useful to send. The downstream signal is continuous, and with few exceptions, it is also expected that an ONU completely fills out its upstream transmission allocation, rather than simply going silent. We therefore need the concept of an idle GEM frame.

In G.987 XG-PON, an idle frame is a well-formed GEM frame with complete and correct header, of variable length, addressed to GEM port 0xFFFF, or indeed, to any unused GEM port. The payload content of an idle frame is not specified.

Once a GEM frame is committed to transmission, that is, once its PLI is set, it continues to completion. Long idle frames could delay transmission of recently arrived valid traffic frames, so it is encouraged to keep idle frames short, perhaps as short as 64 or 128 bytes, or even 8 bytes—a GEM header only, no payload.

In G.984 G-PON, an idle frame is simply an all-zeros sequence. Officially it is 5 bytes long, containing only a degenerate header, but since idle frames can be concatenated at will, or truncated to any length necessary to fill the space, the precise definition is somewhat moot. Be aware, however, that a degenerate GEM frame header contains no payload length indication, therefore, it cannot form part of a chain of consecutively linked GEM frames (Fig. 4.4) and, therefore, can have no successor GEM frames during the same transmission interval. G.984 also recognizes the possibility of sending arbitrary but well-formed GEM frames to unused GEM ports.

Both G.984.3 and G.987.3 suggest that idle payload be crafted such that its spectral properties assist in the performance of the PON. But to rely on idle payload for PON performance would require a long-term commitment on the part of the operator to a PON with no more than some specified traffic load, a constraint that clearly lies beyond the bounds of any operator's ODN engineering rules. Imagine if, as traffic levels increased, the PON stopped working!

4.1.2 G.987 XG-PON Downstream Framing

XG-PON framing is based on G.984 G-PON framing, but incorporates several changes, based on experience with G.984 G-PON and expanded requirements, for example, for increased numbers of ONUs. We start by examining G.987 XG-PON and then subset it for G.984 G-PON in the subsequent sections. The additional requirements of XG-PON include:

UNCORRECTED PROOF

- *Four-Byte Alignment* G.984 G-PON aligns fields on byte boundaries. To facilitate parallel processing architectures, particularly in the early days when implementations are likely to be based on field-programmable gate arrays (FPGAs), G.987 XG-PON fields are aligned to 4-byte word boundaries whenever possible.
- *Robustness* In both G-PON and XG-PON, most fields critical for robust frame interpretation and pattern delineation are equipped with hybrid error correction functionality. At the cost of more overhead—13 bits—the HEC code chosen for G.987 XG-PON can detect (and correct) more errors than the 8-bit cyclic redundancy code (CRC) of G.984 G-PON. G.984 G-PON improves its robustness by transmitting some fields twice and some messages thrice; G.987 XG-PON does not.
- *Increased PLOAM Rate* Recall that the TC layer includes an embedded message-based overhead channel for PLOAM. When an entire PON initializes, several messages must be exchanged with each ONU, and as the number of ONUs on the PON increases, the G.984 G-PON limit of a single downstream PLOAM message per 125-μs frame was a bottleneck. In G.987 XG-PON, each downstream frame can contain as many as one broadcast PLOAM message plus one unicast PLOAM message to each ONU. A single PLOAM message per upstream burst was retained for simplicity; this does not represent a bottleneck.
- *Reduced PLOAM Rate* In G.984 G-PON, every downstream frame contains a PLOAM message. In G.987 XG-PON, if the OLT has nothing to say at the PLOAM, it remains silent at the PLOAM.
- *Higher Security*

  The G.987 XG-PON (super)frame counter is 51 bits long, compared to the 32 bits of the G.984 G-PON counter. Together with algorithm enhancements, this improves the security of both physical layer scrambling and payload encryption.

  As well as the downstream unicast encryption of G.984 G-PON, G.987 XG-PON supports upstream encryption and downstream multicast encryption.
- *PON-ID* The concept of PON-ID was introduced, which may be useful for more robust protection switching.
- *Bandwidth Map Restructure* The G.984.3 concept of start time and stop time was changed in G.987.3 to start time and duration, which improves the layering structure of upstream allocations.
- *FEC* Forward error correction is always on in the downstream direction of G.987 XG-PON, contributing to improved link budget and improved error rate. While G.984 G-PON has 10^{-10} as the target bit error rate, G.987 XG-PON is specified at an error rate of 10^{-12}. Since downstream FEC is always on, bit interleaved parity (BIP) adds no value and has been eliminated in the downstream direction. FEC in the upstream direction may be disabled if the optical budget is comfortable, so upstream BIP is retained from its G.984 G-PON ancestor as a way to track errors when FEC is off.

UNCORRECTED PROOF

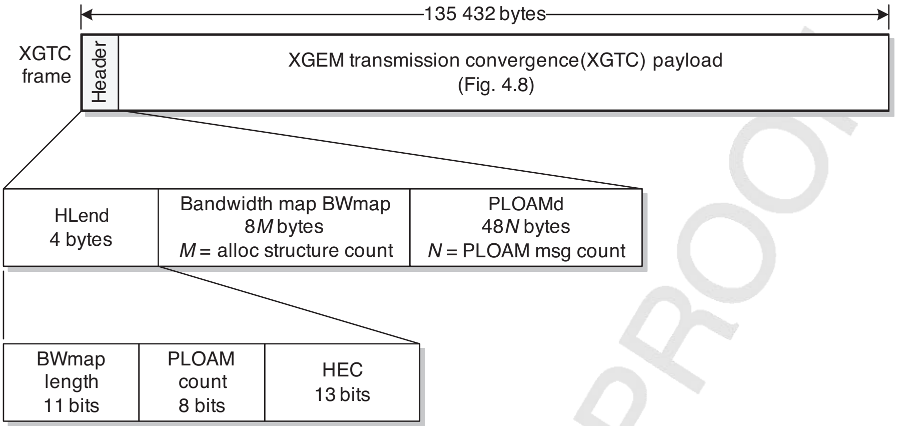
135 432 bytes
XGTC frame
Header
XGEM transmission convergence(XGTC) payload (Fig. 4.8)
HLend 4 bytes
Bandwidth map BWmap 8M bytes M = alloc structure count
PLOAMd 48N bytes N = PLOAM msg count
BWmap length 11 bits
PLOAM count 8 bits
HEC 13 bits

Figure 4.9 XGTC frame header.

UNCORRECTED PROOF

We continue to work our way down the stack from user frames (SDUs) to GEM frames, now to the XGTC frame, then to the PHY frame.

4.1.2.1 XGTC Frame Header

Recall from Figure 4.1 that the GTC frame (XGTC frame in G.987 XG-PON) comprises an XGTC header followed by an XGTC payload that comprises a sequence of GEM frames. Figure 4.9 expands the header structure. There are three fields:

1. *HLend* This 4-byte field tells us the length of the other fields of the header. The BWmap length field is the number of allocation structures in the enclosed bandwidth map. We have 11 bits, but to bound the size and speed required of the algorithms and chips, not more than 512 allocation structures are allowed. Eight bits allow as many as 255 PLOAM messages in the PLOAMd partition, while a 13-bit HEC protects the HLend field from bit errors.
2. *BWmap* In every downstream frame, the OLT authorizes upstream transmission from zero or more ONUs, the authorization to be effective in the corresponding upstream frame. The bandwidth map BWmap is the vehicle for this authorization. The BWmap field contains zero or more so-called allocation structures, each of which authorizes upstream transmission from a traffic-bearing entity in some ONU, or authorizes an ONU discovery transmission. Section 4.1.2.3 examines the BWmap in considerably more detail.
3. *PLOAMd* The downstream PLOAM partition can carry as few as zero PLOAM messages, or a maximum of one broadcast PLOAM message, plus one unicast PLOAM message to each ONU on the PON, bounded by the maximum value 255 of the PLOAM count field. Each PLOAM message is 48 bytes long.

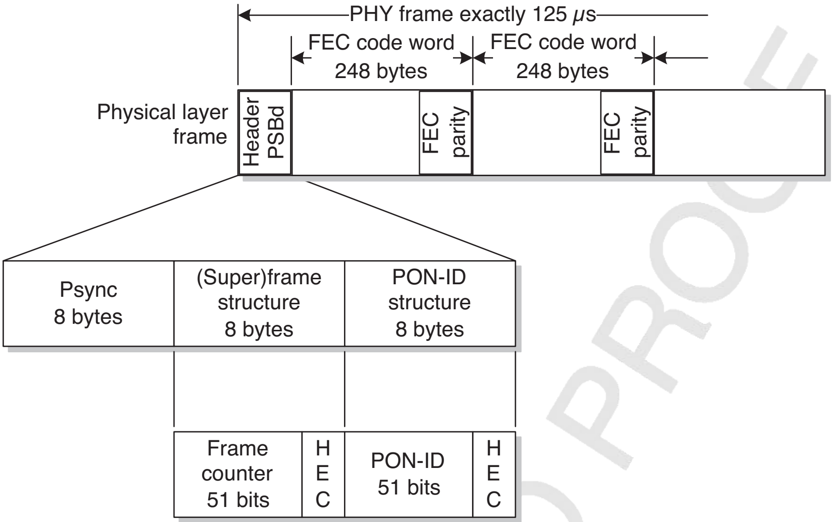

Figure 4.10 PHY frame header.

4.1.2.2 G.987 XG-PON Physical Layer Frame Header

Referring back to Figure 4.1, we see that the downstream XGTC frame is prepended with yet another header to form the physical frame. In XG-PON, this header is called the physical synchronization block downstream (PSBd).

Following the PSBd, the body of the downstream XGTC frame is sliced apart at uniform intervals for the insertion of FEC parity bytes. The original data bytes are not modified in any way. We show how FEC is added and removed in Section 4.1.2.4.

Figure 4.10 illustrates the structure of the physical frame header PSBd. Again, we find three fields:

1. *Psync* The physical layer sync pattern is a fixed sequence of bits, against which the ONU receiver can match, even if the receiver does not have byte sync. As such, it is important that it have a large Hamming distance* from bit-shifted versions of itself—Figure 4.11 challenges us to perform a mental autocorrelation. The PHY frame header is not protected by FEC, and Psync is not even protected by HEC—byte synchronization is a precondition for both of those. The ONU receiver should be designed to be robust to several bit errors in the Psync field.
2. *Superframe Structure* This HEC-protected field contains what is universally known as a superframe counter, even though it actually counts frames, not superframes. With each new downstream frame, the frame count increments by 1, rolling over to 0 if it overflows.

* The Hamming distance between two binary strings of equal length is the number of bit positions that differ between them.

UNCORRECTED PROOF

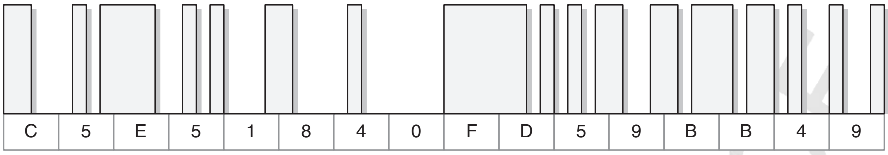

Figure 4.11 Physical synchronization pattern Psync.

3. *PON-ID* Also protected by HEC, the PON-ID has no equivalent in G.984 G-PON. The concept behind the field is that it could be useful in allowing an ONU to detect protection switches when the ONU is dual homed to two separate OLTs. Its default value is zero.

The physical frame header PSBd is even more robust than is apparent at first glance. The receiver can use several clues to remain in sync.

- The frame structure repeats at exact 125-μs intervals.
- The Psync field is fixed and known.
- The value of the frame counter is always one greater than in the previous frame.
- The PON-ID is also stable, not expected to change over perhaps the lifetime of the OLT.

The 16 bytes of HEC-protected frame counter and PON-ID are exclusive-ORed with 0x0F . . . 0F before transmission and again before decoding at the ONU. This anticipates the distinct possibility that at least the PON-ID, and possibly also the frame counter, will be mostly zeros.

The downstream frame is scrambled to reduce its likelihood of containing long sequences of identical bits. Section 4.1.2.5 describes the scrambling process.

At the ONU, the process is reversed. The ONU contains a simple state machine for frame synchronization, depicted in Figure 4.12. The state machine uses the 125-μs interval clue and the (super)frame counter SFC clue along with PSync match, and tolerates three (recommended value for M) errors before it gives up and declares itself to have lost synchronization (loss of downstream synchronization LODS).

Once the ONU receiver has aligned itself with the physical frame header, the ONU is in a position to process the BWmap, looking for upstream transmission allocations to itself. Then the ONU inspects the downstream PLOAM messages, looking either for a broadcast message or a unicast message addressed to itself, or possibly one of each.

In parallel with processing of the header fields, the ONU can descramble the frame, decode FEC blocks, and correct any errors that may have occurred. Then comes the payload of GEM frames, to be filtered according to their GEM port IDs, and finally the decryption process.

In brief, that is how G.987 XG-PON works, downstream.

UNCORRECTED PROOF

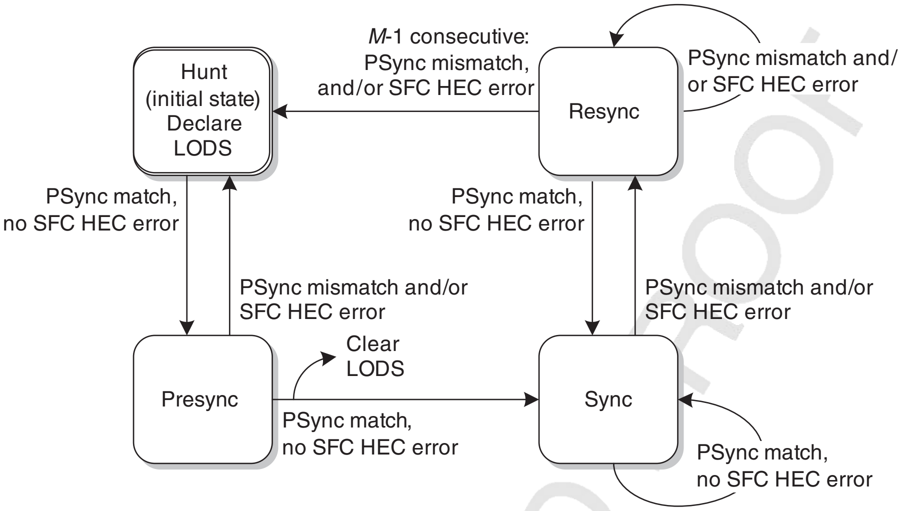

Figure 4.12 G.987 XG-PON frame synchronization state machine.

4.1.2.3 G.987 XG-PON Bandwidth Map

We skipped over the details of the BWmap because it is a significant topic in its own right. Even now, we only discuss part of the upstream capacity allocation process. The DBA discussion in Section 6.3 explains the factors that the OLT takes into account when it builds the BWmap. For our purposes here, we take the BWmap as given.

First, let us consider the structure and the fields of the BWmap. Figure 4.13 reminds us that the XGTC frame header includes a field that tells us how many allocation structures the BWmap contains.

If several consecutive allocation structures are assigned to the same ONU, without a break in transmission time—that is, to be concatenated into a single upstream burst—we refer to them as a contiguous series. The rules for BWmap construction Q1 (Section 6.3.2.2) mean that a contiguous series of allocations is also a contiguous series of allocation structures in the BWmap. A contiguous series uses upstream PON capacity efficiently because it avoids the overhead of separate bursts for multiple allocations to a given ONU.

The fields of an allocation structure are as follows:

1. *Alloc-ID* Each allocation identifier belongs to not more than one ONU. Alloc-IDs in the range 0–1022 are implicitly bound to the ONUs with the, respectively, identical TC-layer ONU-IDs; alloc-ID 1023 is an invitation for any new ONUs to respond with their serial numbers, making this a serial number grant, as described in Section 4.2. Other alloc-IDs are assigned to ONUs through (PLOAM) provisioning. The alloc-ID represents the fundamental unit of granularity at which the OLT manages traffic from the various contending ONUs on the PON.

UNCORRECTED Proof

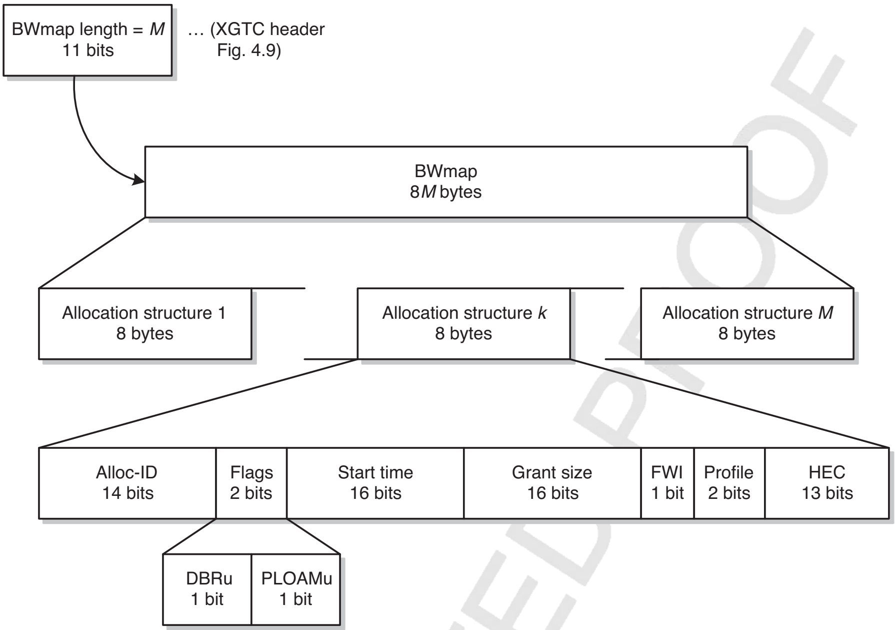

Figure 4.13 BWmap structure.

Alloc-IDs and T-CONTs

Alloc-IDs are frequently confused with T-CONTs, particularly in discussions of G-PON resource management at the OLT (DBA). They are not the same. Their identifiers are different. An alloc-ID exists in a flat namespace per PON, while the external identifier of a T-CONT is hierarchical: <management-layer ONU-ID><T-CONT number>.

Alloc-IDs and T-CONTs are logically distinct in the layering model. Most importantly, there is not even a one-to-one correspondence: T CONTs are assigned to alloc-IDs at the OMCI layer for client traffic, but OMCI itself is carried in a default alloc-ID and needs no T-CONT. See also Section 5.1.1.

This is much the same distinction that we saw between the TC-layer ONU-ID (an alloc-ID is visible in the bits flowing across the PON) and the management-level ONU-ID (a T-CONT is a name that refers to an alloc-ID). Think of it this way: Subscriber traffic is managed through T-CONTs, while the underlying engine only knows alloc-IDs. Although the ambiguity is mostly harmless, it should be kept in mind whenever there is a need for precision.

UNCORRECTED PROOF

2. *Flags*

DBRu If this flag is set, the ONU sends a queue occupancy report in the header of the upstream response for that alloc-ID. This is discussed further in the upstream framing section below. The queue occupancy report is one of the primary inputs to the dynamic bandwidth assignment process, DBA.

PLOAMu The OLT sets this bit to request the ONU to send an upstream PLOAM message in its response. The bit is meaningful only in the first allocation of a contiguous series addressed to a given ONU. If the ONU has no substantive PLOAM message to send, it just sends a heartbeat Acknowledge. The ONU is not free to send upstream PLOAM messages at its own discretion, nor can it ignore a PLOAMu request.

3. *Start Time* In the first grant of a possibly contiguous series, this field specifies the start time of the upstream transmission. Coupled with delay equalization, this is the essential field that allows the OLT to interleave ONU burst responses without collisions. The granularity of the start time is 4 bytes, one word. If it is thought of as a time, rather than as a word count, words flow at the 2.488-Gb/s rate of the upstream PON, not at the 10-Gb/s downstream rate. Start time is an offset relative to the start of the upstream frame, which (Section 4.3.3) is defined to be the first bit following the physical synchronization block, that is, the first bit of the XGTC header.

   In subsequent allocations of a contiguous series, the start time is coded as 0xFFFF, which just indicates continuation without a break from the previous allocation.
4. *Grant Size* This field specifies the number of words authorized by the grant. The word count includes space for the upstream DBR, if there is one, but does not include space for a possible upstream PLOAM message or other overhead such as the burst header or trailer, or FEC, which exist at lower layers in the client–server model.
5. *Forced Wakeup Indicator (FWI)* Suppose the ONU supports low-power modes (Section 4.5), and suppose the OLT wishes to awaken it. The ONU may be sleeping at the time the OLT sends the *Sleep_allow* (off) PLOAM message. In the first allocation of a contiguous series, the FWI bit has the same effect and is more likely to be received by the ONU when the ONU enters its sleep-aware state.
6. *Profile Index* A burst profile specifies the header of the upstream burst. The OLT defines one or more burst profiles, and communicates them to the ONUs through continuing broadcast PLOAM messages, so that new ONUs can learn them. The two-bit profile field in the BWmap specifies which burst profile is to be used by the ONU in its upstream response. If the ONU does not recognize the profile, it must not respond at all.
7. *HEC* Hybrid error correction is described briefly in the GEM frame section above and further in Appendix I.

4.1.2.4 Forward Error Correction

Appendix I also describes FEC and summarizes the algorithms that pertain to the different directions and G-PON technologies. Here, we take FEC and FEC parity as

UNCORRECTED PROOF

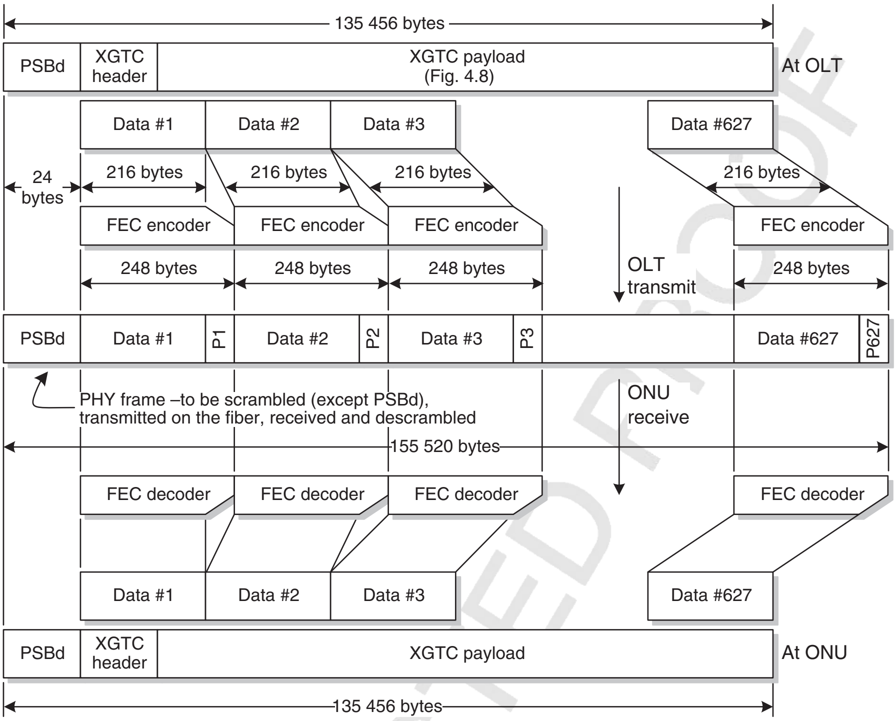

Figure 4.14 Downstream FEC, G.987 XG-PON.

given and show how parity blocks are mapped into the downstream frame. Figure 4.14 shows framing and FEC processing.

G.987 XG-PON downstream uses an RS(248, 216) code, which can correct up to 16 bytes in error. We add 32 bytes of FEC parity to each 216-byte block of XGTC payload and fit the resulting 248-byte code word into the physical frame structure. The size of the code word is chosen such that an integer number of code words fits into the XGTC frame, excluding the PSBd physical header, which would not benefit from FEC in any event.

We then submit the physical frame to scrambling (next section) and transmit it on the fiber. At the ONU, we reverse the sequence, presenting an XGTC frame to the next layer of the stack for header and GEM frame processing.

Downstream FEC in XG-PON is always on. It costs about 13% of the PON's capacity.

4.1.2.5 Scrambling

Scrambling is a technique of pseudorandomizing a data stream to reduce the likelihood of long sequences of identical digits (bits). In non-return-to-zero (NRZ) transmission, as used in G-PON and XG-PON, consecutive identical digits (CID)

UNCORRECTED PROOF

contain no transitions, and a receiving clock can lose synchronization if the dearth of transitions persists too long. G-PON expects receivers to tolerate strings of up to 72 CID. Shifting the balance of 1's and 0's toward 50% also helps the receiver optimize its decision threshold.

A scrambler does not add bits to the flow and therefore imposes no capacity penalty.

It is important to understand that a scrambler does not guarantee to eliminate long, fixed strings of repeated bits. The pseudorandom sequence includes all possible bit sequences up to the order of its generator polynomial, so there necessarily exists a subsequence from the scrambler that exactly matches—or complements—any given payload sequence, resulting in a string of consecutive identical zeros (or ones, if it is a complement match).

Scrambling is not to be confused with encryption: A scrambled signal is easy to decipher. However, scrambling does have a security benefit in that it can improve the network's resistance to denial of service (DoS) attacks. If a malicious user were able to upload, or download, a data sequence that happened to align with the scrambler sequence, the PON might lose synchronization, much to the annoyance of everyone else on the same PON (downstream) or on the same ONU (upstream).

G.984 G-PON uses only 7-bit scramblers, which therefore repeat after 127 bits, and are comparatively easy to attack. G.987 XG-PON uses a much longer scrambler and initializes it with the (super)frame counter, making it many orders of magnitude harder to predict. Also, the attacker has no way to force transmission of the malicious pattern at any particular instant in the frame, making a deterministic attack even more difficult.

That is the overview. Now for the details:

The G.987 scrambler applies to the entire downstream frame excluding the PSBd, direct visibility of which is necessary to achieve frame acquisition. Scrambling starts at the first bit following the PSBd and runs to the end of the frame.

The scrambler polynomial is $x^{58} + x^{39} + 1$. The scrambler is initialized with a different 58-bit value at the beginning of each frame, such that no two nearby frames use the same scrambling sequence. The 58 bits of preload are the 51 bits of the current (super)frame counter, with seven 1 bits at the least significant end.

The scrambler is the same at the OLT and the ONU. Scramblers are conventionally constructed with linear feedback shift registers, as shown in Figure 4.15. The simple and well-defined initialization mechanism allows the ONU receiver to easily generate the same pseudorandom sequence that was used at the OLT transmitter.

4.1.3 G.987 XG-PON Upstream Framing

Figure 4.16 shows how the upstream signal appears at the OLT. Physical layer bursts from a number of ONUs arrive, neatly interleaved in time so that they do not collide. This is the consequence of the OLT's allocation of start times and grant sizes in the downstream BWmap that specified this upstream frame. The boundaries of the

UNCORRECTED PROOF

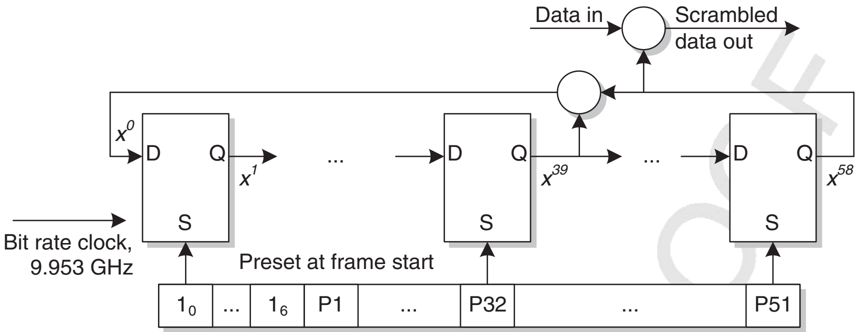

Figure 4.15 G.987 XG-PON scrambler, downstream.

upstream physical layer frame are precisely defined—see timing Section 4.32—but they are not directly physically observable.

Figure 4.17 reminds us of the timing and power relationships of the burst header from Section 3.9. The 0 and 1 levels define the envelope of normal burst transmission, while the different level shown as *off* recognizes the fact that the ONU transmits some amount of light even at logical level 0 and transmits no light when it is off.

Guard Time Each burst is separated from other bursts by the guard time t_g, whose recommended minimum value is 8 byte times. The OLT could choose to make it larger, or even smaller, if it had some reason to do so. The guard time absorbs slight variations in timing (uncertainty t_u) to guarantee that bursts do not collide. It also allows for light decay from the previous ONU as it turns off

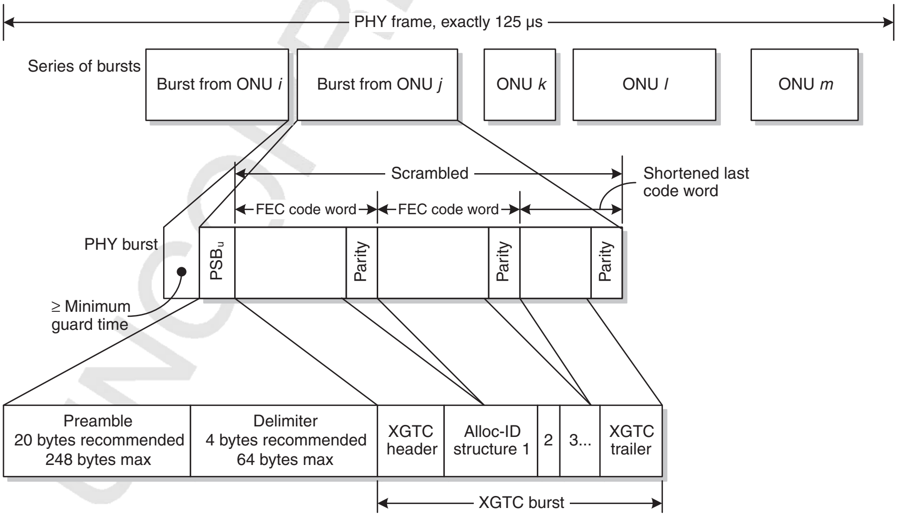

Figure 4.16 G.987 XG-PON upstream frame, OLT view.

UNCORRECTED PROOF

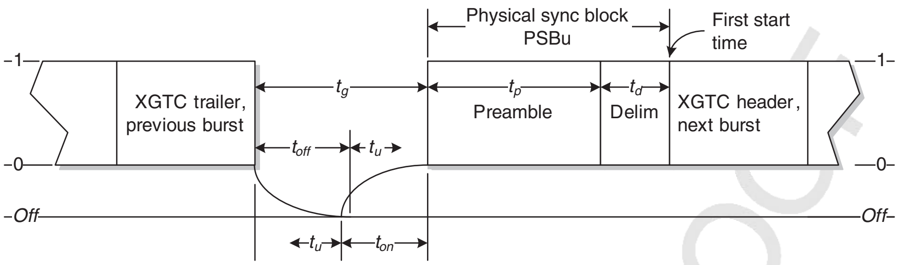

Figure 4.17 Interburst detail.

its transmitter (time t_{off}), and light from the next ONU as it turns its transmitter on (time t_{on}). The t_{off} interval of one ONU may overlap t_{on} from the next.

During the guard interval, an ONU is permitted to transmit no more light than that of a logical 0.

The upstream physical sync block PSBu contains:

Preamble As a sin of omission, Figures 4.16 and 4.17 imply that each burst has the same amplitude. In fact, adjacent bursts can vary by as much as 15 dB, an optical power ratio of 32 : 1. The burst preamble permits the OLT receiver to adjust its gain and thresholds for the amplitude of this particular burst, as well as to acquire clock phase. The preamble length and pattern are specified by the OLT and distributed to the ONUs in the *profile* PLOAM message. To facilitate cost optimization at the OLT receiver,[*] the profile may contain a very long preamble, although 20 bytes is the recommended length, indicated in Figure 4.17 by t_p.

Delimiter The same options are available for the delimiter; the recommended length t_d is 4 bytes, but it may be as long as 64 bytes if necessary. Given that bit synchronization is achieved during the preamble phase, the purpose of the delimiter is to synchronize byte boundaries, and especially the boundary between the delimiter and the first byte of the payload itself. A longer delimiter is more robust in the presence of high bit error rates, especially if the ONU receiver is designed to tolerate some number of bit errors.

The first byte of the physical layer payload begins the XGTC burst. It also begins the coverage of upstream FEC, assuming that FEC is enabled. Given that FEC is enabled, the remainder of the burst, including the burst trailer, is included in a series of FEC code words, the last of which may be shortened. FEC is discussed in Appendix I; downstream FEC is illustrated in Figure 4.14. Construction of the matching figure for upstream bursts is left as an exercise for the serious student: There are no surprises.

[*] In theory, less costly ONU optics could also be accommodated in this way, but, in practice, the OLT may have no way to adaptively discover the need for longer preambles in general or on specific ONUs in particular.

UNCORRECTED PROOF

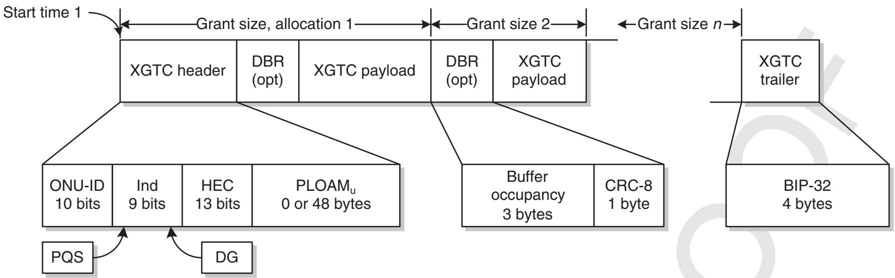

Figure 4.18 XGTC header, upstream

As shown in Figure 4.16, subsequent to FEC parity byte insertion, the burst is scrambled. In a similar way as the downstream frame, the scrambler starts immediately after the PSBu field and runs through the remainder of the burst. It is the same polynomial, with the same preload, as in the downstream direction.

Figure 4.18 expands the XGTC burst—that is, the burst before FEC has been added at the ONU transmitter or after FEC has been removed at the OLT receiver—to show its structure. An XGTC burst comprises an XGTC header and trailer, between which we find a concatenated series of responses to the allocation structures from the BWmap. Each allocation response includes a DBA report (DBR) if it was requested by the BWmap, followed by a series of GEM frames (Fig. 4.8), which may include idle frames. The last frame in each allocation may contain only a fragment of an SDU. Allocation boundaries are respected, even if one allocation is padded with idle frame and the next allocation overflows.

Observe that the first allocation must include space for the XGTC header, as well as whatever payload is intended. Subsequent allocations must include space for DBR fields, if they are needed.

The XGTC header comprises four fields, one of which is the ubiquitous HEC. The others are:

ONU-ID If the ONU is attempting to register onto the PON by responding to a serial number grant (Section 4.4), it uses the unassigned ONU-ID value 1023 (0x3FF) for this field. Otherwise, it uses the TC-layer ONU-ID that was assigned to it by the OLT during registration. Strictly speaking, the ONU-ID is unnecessary; it is merely an extra check that everything is working as expected.

Indicators Only two of the bits in this field are defined. The other 7 bits are reserved.

PLOAM Queue Status The most significant bit of the indicators field is a request by the ONU for a PLOAMu allocation in some future BWmap. If the current BWmap already requests an upstream PLOAM message, the PQS

UNCORRECTED PROOF

bit is set to indicate that, even after transmitting the PLOAM message in the current burst, there still remains at least one PLOAM message in the ONU's queue, awaiting a transmission opportunity.

Dying Gasp The DG bit is a way for the ONU to signal that it expects to disappear from the PON for its own local reasons, usually loss of power.[*] If subsequent troubleshooting is required, the prior reception of a DG assists the operator in isolating the problem to the ONU itself, rather than to the optical distribution network. DG is a best-efforts indication: The ONU may not be able to send DG before it dies. Further, the ONU may recover power before dying, so it may in fact remain alive on the PON and simply cease signaling DG in future bursts.

PLOAMu If the first allocation in the BWmap for this ONU specifies an upstream PLOAM message, it appears here. It is not included in the burst header HEC because it contains its own message integrity check (MIC), which includes an error detection function. A secondary reason to exclude the PLOAM message from HEC is to reuse the same HEC algorithm everywhere; the error-correcting algorithm tops out at 8 bytes, including the HEC field itself.

Following the XGTC header, we encounter zero or more XGTC responses, one for each allocation in the possibly contiguous series specified by the BWmap. Each response begins with a dynamic bandwidth report DBR if and only if the BWmap specifies DBRu for that particular alloc-ID.

The DBR contains two fields:

Buffer Occupancy This field is a word count of the present backlog in the aggregate of upstream transmit queues for this alloc-ID (T-CONT). It does not subtract off the allocation for the current burst, nor does it attempt to estimate traffic arrival rates. It counts client payload size (Ethernet frames) but not GEM frame headers.

To illustrate, suppose we have 1000 words of upstream Ethernet backlog for this T-CONT, suppose the current burst allocation for this alloc-ID is 100 words, and suppose that traffic is continuing to arrive at the rate of 300 words per unspecified interval—one of the reasons we do not attempt to include arrival rates. Given all of these assumptions, the DBR reports 1000 words. The OLT can subtract the 100-word current allocation, as well as any additional allocations that may already be in the pipeline. Since the OLT governs the rate and size of upstream grants, only the OLT is in a position to act on the estimated traffic arrival rate. Finally, since the reported backlog does not include GEM frame headers, the OLT might choose to grant something larger than 1000 words, perhaps 1050 words.

UNCORRECTED PROOF

[*] It has been pointed out that ONU reboot would be another valid reason to signal DG.

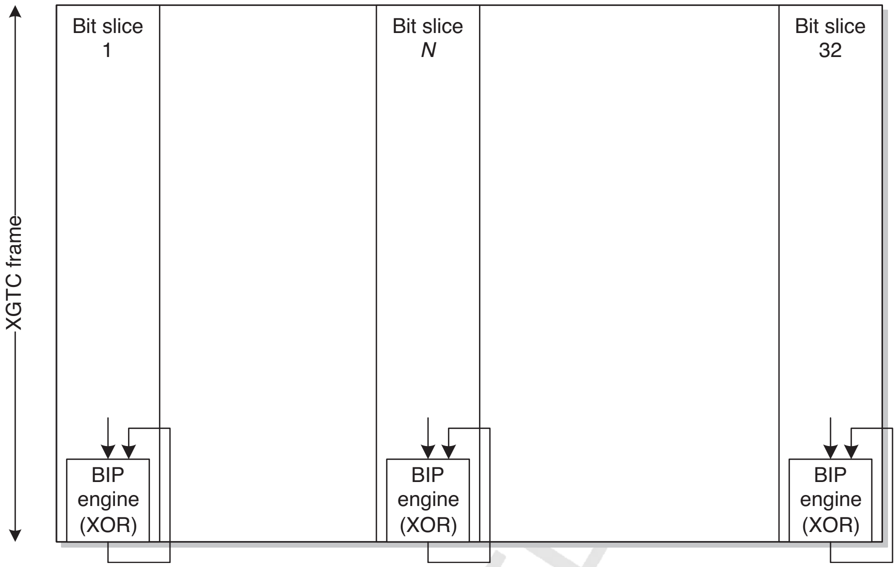

Figure 4.19 BIP-32.

CRC-8 This is just an error check on the buffer occupancy field. Its generator polynomial is $x^8 + x^2 + x + 1$. The standard specifies that cyclic redundancy check failure should cause the OLT to discard the DBR.

To complete our inspection of the XGTC frame, the XGTC trailer is a BIP-32 block (Fig. 4.19). What BIP—bit interleaved parity—means in this context is that the BIP engine looks at the XGTC burst, starting with the XGTC header, as if it were a stream of 32 independent parallel bitwise slices. The BIP engine sets each of its individual bits to produce an even number of 1's in each flow, an exclusive-OR function, counting the BIP itself as the final bit of the checked sequence. At the OLT, the same check detects the case when an odd number of errors occurs in the burst. BIP errors are intended to be counted as a performance monitoring parameter.

For a high-quality channel, the probability of multiple errors in a given bit position across a burst is negligible, and BIP is a good estimate of the total error rate. At higher raw error rates, the probability of two errors in a given bit slice increases, and BIP saturates; the saturation point depends on the length of the burst, but typically lies in the BER range 10^{-4} to 10^{-5}. In practice, it does not much matter where the BIP saturates: If the BIP count exceeds some quite small threshold, it is advisable to turn on upstream FEC. At high raw error rates, BIP errors can still be collected, but the count does not mean much. Even when the BIP count is not saturated, it must be compared with some independent count of the total number of bytes received before the quality of the link can be evaluated.

The OLT also collects FEC statistics: If the number of corrected code words were consistently zero for a given ONU—which is quite possible for ONUs with

UNCORRECTED

comfortable optical budgets—we could gain some additional upstream traffic capacity by disabling upstream FEC for that ONU.

4.1.4 G.984 G-PON Downstream Framing

Though not an identical twin of G.987 XG-PON, G.984 G-PON framing is clearly a sibling. We recognize many family characteristics as we look into its details.

- G.984 G-PON framing differs in part because it was not intended to operate at high raw bit error rates, so it did not need the same level of inherent robustness. FEC is optional in both directions.
- The speed of G.984 G-PON electronics did not warrant 4-byte alignment.
- The fiber was understood to be secure physically, so that upstream content did not need to be encrypted. Downstream multicast was (correctly) expected to be encrypted by the middleware, so that further multicast encryption by the PON layer was thought to be unnecessary.
- With fewer ONUs on the PON—32 envisioned initially, 64 in the current view—there was less need to send several PLOAM messages in a single downstream frame.

Downstream physical layer frames of the G.984 G-PON transmission convergence (GTC) layer occupy 125 µs with continuous transmission, as shown in Figure 4.20.

4.1.4.1 GTC Frame Header

The downstream GTC frame header is called PCBd, the physical control block. It contains seven fields, one of them a repetition.

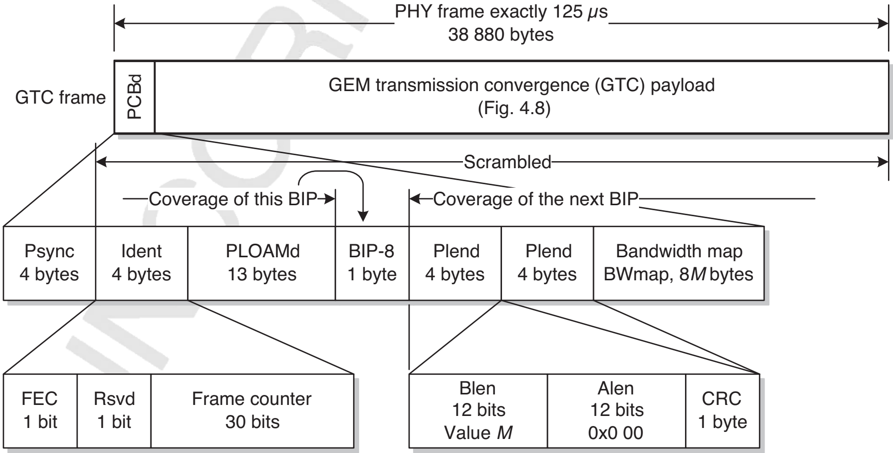

Figure 4.20 Downstream GTC frame.

UNCORRECTED PROOF

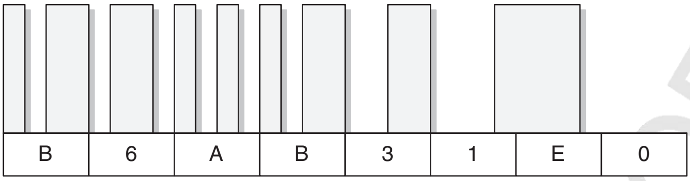

Figure 4.21 Physical sync pattern PSync.

PSync As with G.987 XG-PON, the purpose of the physical synchronization pattern is to provide a well-defined bit sequence against which the ONU receiver can acquire byte delineation. As we would expect, Psync is not scrambled. Figure 4.21 invites another mental autocorrelation exercise to convince ourselves that bit-shifted versions of the Psync pattern are reasonably immune to misinterpretation.

Figure 4.22 shows the state machine by which a G.984 G-PON ONU acquires and maintains frame sync. Examining every bit as a candidate for start of frame, the machine goes from hunt to presync state upon a single match. It then waits exactly 125 µs and checks again. If it sees $M_1 - 1$ additional consecutive Psync patterns at the expected intervals, it goes to sync state, clears the loss of signal/loss of frame indication LOS/LOF, and begins to decode the rest of the downstream frame; but a single Psync error during the presync state puts the state machine back into hunt mode. Once synchronized, the machine remains in sync state until it sees M_2 incorrect Psync fields. The recommended value for M_1 is 2, and for M_2 is 5.

Ident

- *FEC* If this bit is set, the downstream frame is protected with FEC. By comparison, downstream FEC is *always* active in G.987 XG-PON. To protect against bit errors, the ONU requires four consecutive frames with the same FEC bit before it recognizes a change. This presents no service disruption problem in practice because FEC is not intended to be changed dynamically.

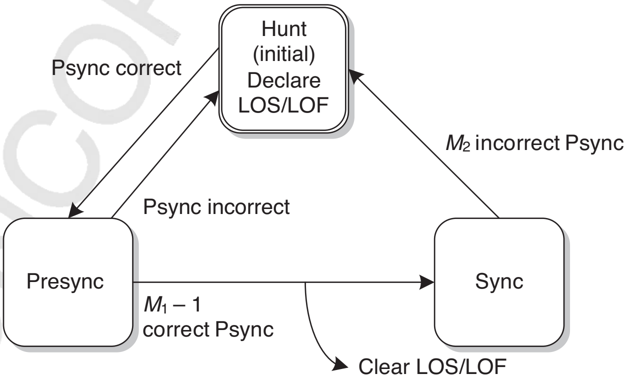

Figure 4.22 Psync state machine.

UNCORRECTED PROOF

- *Rsvd* The reserved bit is not used.
- *Frame Counter* As in G.987 XG-PON, the recommendation calls this a superframe counter. It is a 30-bit circular counter that increments with each downstream frame. Its primary use is in encryption key switching and time-of-day synchronization, both of which are described later in this chapter. Just as in G.987 XG-PON, the predictably incrementing frame counter is available to provide additional robustness in synchronizing to the downstream frame.

  This option is not mentioned in G.984.3; but then, G.984 G-PON was not originally intended to operate at the same high error rates as XG-PON. The introduction of the C+ optical budget class changes that assumption (10^{-4} raw error rate), and it may be wise for G.984 implementations to consider robustness enhancements.

PLOAMd In G.984 G-PON, PLOAM messages are 13 bytes long. Every downstream frame contains one; if the OLT has nothing to say, it sends a *no_message* message. Appendix II discusses PLOAM messages.

BIP-8 This field computes bit-interleaved parity on each of 8 parallel bit slices in the bytes of the frame. Its scope includes all bytes, excluding possible FEC parity bytes, transmitted since the BIP-8 field of the previous frame. At the ONU, BIP is computed *after* a possible FEC correction step. BIP counts can be used for performance monitoring and, in theory, as the input to SDH-like signal fail and signal degrade conditions.

Plend The payload length field does not indicate the length of the payload.

- In its 12-bit field *Blen*, it specifies the number of 8-byte allocations in the bandwidth map BWmap.
- As well as the GEM partition, the original G.984 G-PON specification included an ATM partition, and a similar 12-bit field *Alen* was defined for it. ATM was subsequently removed from the GTC layer, in fact completely removed from G-PON, so the Alen field is always set to zero on transmit and ignored on receive.
- The *CRC* field checks and corrects errors. The specified code, $x^8 + x^2 + x + 1$, can correct one error and detect multiple errors.

The downstream frame includes two copies of Plend for robustness, so the ONU makes an additional check of the CRC results to determine whether the BWmap and the downstream frame are usable. Refer to Table 4.2.

BWmap The bandwidth map specifies the allocations of upstream capacity to the alloc-IDs of the various ONUs on the PON.

4.1.4.2 G.984 G-PON Bandwidth Map

Figure 4.23 portrays the BWmap in G.984 G-PON. As in G.987 XG-PON, it comprises a series of 8-byte allocation structures, each of which grants permission

UNCORRECTED PROOF

TABLE 4.2 PLend Error Processing

First Plend	Second PLend	Corrected Results	Decision
Uncorrectable	Uncorrectable		Discard
Correctable	Correctable	Different	Discard
No errors	No errors	Different	Discard
No errors	No errors	Same	Either PLENd
Correctable	Correctable	Same	Either PLENd
No errors	>0 errors		First PLENd
Correctable	Uncorrectable		First PLENd
>0 errors	No errors		Second PLENd
Uncorrectable	Correctable		Second PLENd

to transmit an upstream burst. As in XG-PON, several allocations to a given ONU may be concatenated into a single burst. We describe the rules for start time and stop time pointers below.

There are five fields in a G.984 G-PON allocation structure:

Alloc-ID The alloc-ID specifies the owner of the upstream grant. Values less than 254 match the corresponding TC-layer ONU-ID directly and authorize upstream OMCI transmission (it is allowed but not encouraged to carry

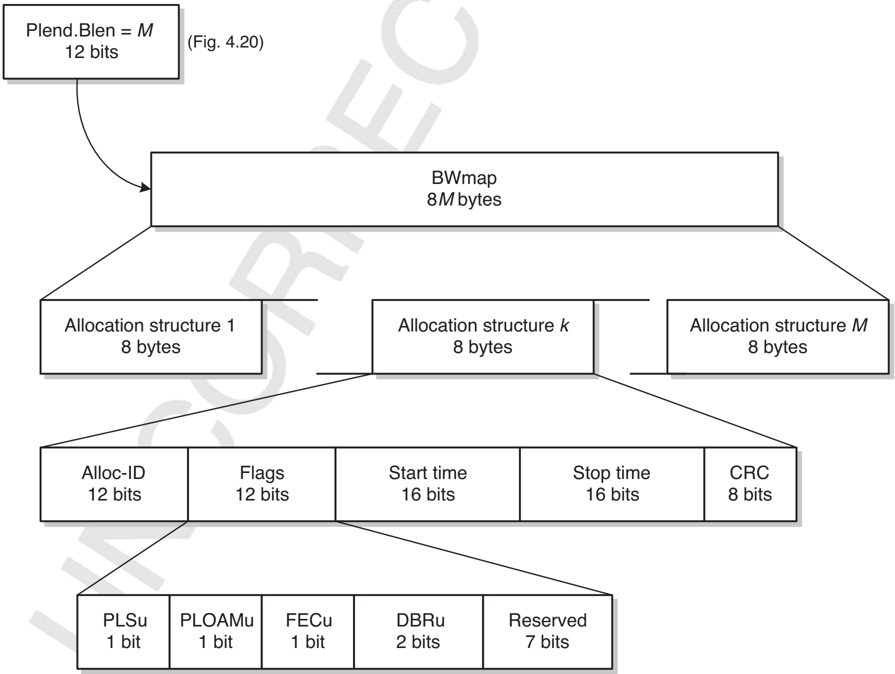

Figure 4.23 G.984 G-PON bandwidth map.

UNCORRECTED PROOF

subscriber traffic in this, the so-called default alloc-ID). Alloc-ID 254 invites a serial_number PLOAM response from undiscovered ONUs, those that do not yet have an assigned TC-layer ONU-ID. Alloc-ID 255 is reserved, and the remainder of the space is available for alloc-IDs that are mapped to T-CONTs, to carry subscriber traffic.

Flags The flags field comprises 12 bits, of which 5 are used.

- *PLSu* G.984 G-PON originally contemplated power leveling. This was a feature whereby the OLT could reduce the dynamic range of the upstream power received from the various ONUs by commanding nearby ONUs to reduce their transmit power levels. The feature has been deprecated; this bit is set to 0 on transmit and ignored on receive.
- *PLOAMu* If this bit is set, the ONU sends a PLOAM message in the upstream allocation. This bit is only meaningful in grants sent to the default alloc-ID. The PLOAM message is often shown pictorially as the first of a contiguous series of allocations, but the default alloc-ID, and its PLOAM message, may in fact occur in any position of the burst.
- *FECu* This bit requests that FEC be enabled in the upstream burst. Unlike G.987 XG-PON, G.984 G-PON ONUs are allowed to not support FEC, in which case they ignore this bit and transmit ordinary data, just as if FEC had not been requested. A bit in the upstream burst header tells the OLT whether FEC is in fact present or not.
- *DBRu* DBR is a 2-bit field that instructs the ONU to send a queue occupancy report in the upstream burst, or not, and which mode to use. G.984 G-PON specifies two modes of DBA reporting (even more, if we go back far enough into history), mode 0 and mode 1. Mode 0 is the default, and is the only mode defined in G.987 XG-PON. In G.984, the mode 0 report is a single byte containing a nonlinear code (Table 4.3) that approximates the total queue occupancy in bytes. Mode 1 contains two similarly encoded bytes, one for green traffic and one for yellow (conformant traffic and traffic beyond the guarantees of its contract, as described in Section 6.3).

Start Time This 16-bit field specifies the offset of the allocation's start time from the beginning of the upstream frame. It is measured in bytes at the upstream rate, 1.2 Gb/s. The burst header occurs prior to the (first) start time, and must be taken into account by the OLT when it plans the BWmap.

Stop Time The stop time also refers to a byte offset from the start of the upstream frame.

CRC The allocation structure is protected by a CRC-8 that can correct two errors. If the CRC detects more errors than it can correct, the ONU discards the allocation.

As with G.987 XG-PON, the concept of contiguous grants also pertains to G.984 G-PON—that is, several grants in a row, all of them directed to the same ONU. An ONU is expected to be able to support up to eight allocation structures in a given BWmap, although it is not specified that they need be contiguous. The way in which

UNCORRECTED PROOF

contiguous grants are indicated in G.984 is that the start time of the subsequent grant is exactly one greater than the stop time of the current grant. If this condition is not met, the OLT must allocate the second grant as a completely separate burst, complete with burst overhead.

Upstream FEC is either on or off for the entire burst. This means the FECu bit should theoretically be the same in each allocation. In practice, the ONU would configure the burst according to the first allocation and ignore the others.

When it is planning the BWmap, the OLT must consider that an upstream PLOAM message consumes capacity from a grant to the default alloc-ID, that a DBA report consumes capacity from each allocation for which it is specified, and that FEC is an overhead of 16 bytes added onto each 239-byte chunk of payload content, starting with the BIP field* of the upstream burst header, running continuously across allocation boundaries, and ending with a possibly shortened last code word.

4.1.4.3 Assembling Downstream G.984 G-PON Frame

The BWmap completes the header of the downstream frame. We now append the GTC payload, which comprises a series of GEM frames, possibly including idle frames. Before transmitting the frame, we optionally insert FEC parity bytes, and we always scramble it.

FEC FEC is discussed in detail in Appendix I and in the downstream XG-PON framing Section 4.1.2. Here we describe the details of G.984 G-PON downstream FEC, as shown in Figure 4.24.

If FEC is disabled, the G.984 G-PON downstream physical frame is identical to the GTC layer frame (but scrambled). If FEC is enabled, the entire frame, including PCBd, is protected by a series of FEC code words. The code is RS (255, 239), which means that 239 data bytes are followed by 16 parity bytes. The first FEC code word begins with the first byte of the PCBd.

The physical frame is 38,880 bytes long, enough for slightly more than one hundred fifty-two 255-byte FEC code words. We, therefore, have a shortened last code word 153 in each frame, 104 bytes of data with the usual 16 bytes of FEC parity. After subtracting FEC parity overhead, 36,432 bytes remain in the downstream frame; the cost of downstream FEC in G.984 G-PON is 6.3%.

A G.984 G-PON ONU is not required to support FEC decoding, but it is required to be able to detect that the downstream FEC bit is set, and if so, skip past the FEC parity bytes and properly receive the payload. In fact, all current G-PON ONUs support FEC.

Scrambling Scrambling is described in Section 4.1.2.5. In G.984 G-PON, the downstream frame is scrambled with a frame-synchronous polynomial, $x^7 + x^6 + 1$, whose output repeats every 127 bits. The scrambler is initialized to all 1's on the first

UNCORRECTED PROOF

* Although FEC starts with the BIP field, which is the first byte of the burst header, the start time and the allocation interval refer to the first byte *after* the burst header. This byte is either the first byte of a PLOAM message, the first byte of a DBA report, or the first byte of a GEM payload frame. Refer to Figure 4.27.

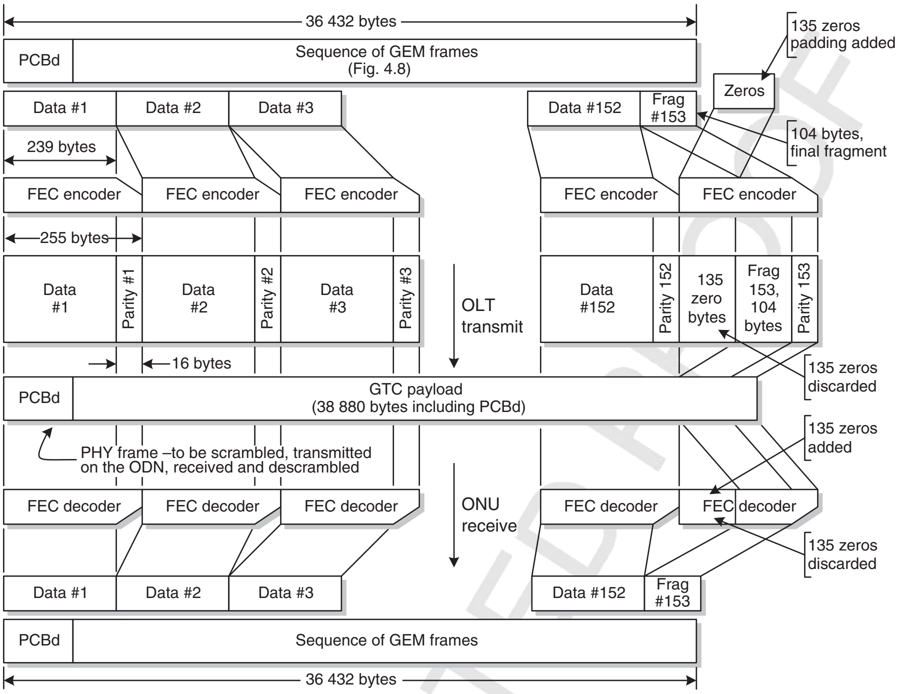

Figure 4.24 G.984 G-PON downstream FEC.

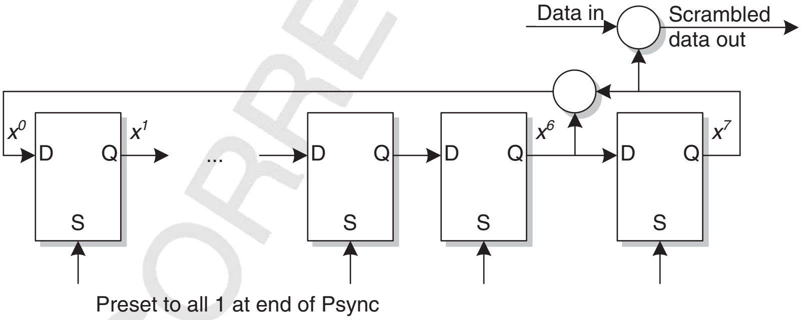

Figure 4.25 G.984 G-PON scrambler.

bit following the Psync field, and as shown in Figure 4.20, runs continuously across the frame, including FEC bytes. The Psync field itself is not scrambled.

Figure 4.25 illustrates the G.984 G-PON scrambler.

The short polynomial and fixed initial state of G.984 G-PON conceivably pose a risk of DoS attacks, supposing that a malicious user were able to transfer data that caused the scrambler to generate a long sequence of identical digits. This risk was

UNCORRECTED PROOF

addressed in the much longer scrambler and varying preset states of the G.987 XG-PON scrambler.

4.1.5 G.984 G-PON Upstream Framing

A G.984 G-PON upstream burst comprises a burst header and a series of allocation intervals. We discuss the burst header after the following explanation of the framing details.

A generalized upstream allocation interval comprises a GTC overhead block, followed by a GTC payload section that contains a sequence of GEM frames. An allocation to the default alloc-ID may include an upstream PLOAM message, as shown in Figure 4.26, while any allocation interval may include a dynamic bandwidth report DBR:

PLOAMu An upstream PLOAM message is present:

- Only if the allocation was directed to the ONU's default alloc-ID or to alloc-ID 254 for ONUs that are attempting to register on the PON,
- Only if the PLOAMu flag was set in the BWmap allocation.

If the ONU has no substantive message to send, it sends a no_message message.

DBR The DBA report is likewise present only if requested by the OLT, but DBR can occur in any allocation structure, reporting on the queue backlog for that particular alloc-ID. The granularity of the report defaults to 48 bytes, for reasons that have to do with the ATM history of ITU-T PONs. It may be provisioned by the OLT through OMCI as the GEM block length attribute of the ANI-G managed entity.

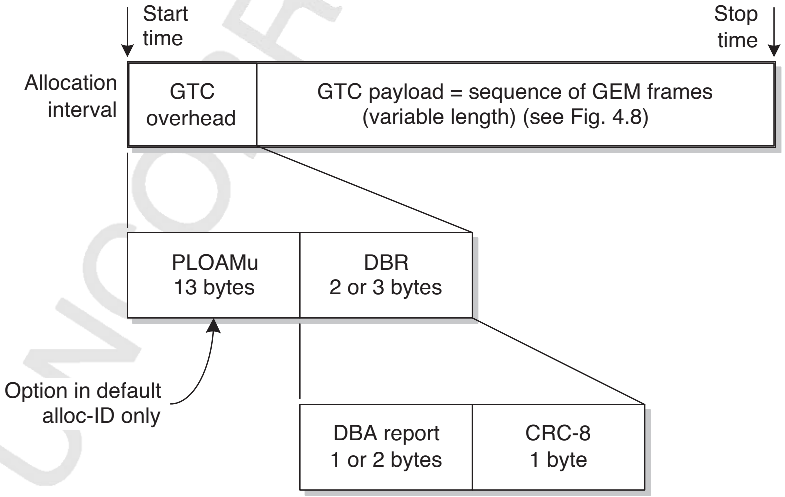

Figure 4.26 Allocation interval structure.

UNCORRECTED PROOF

TABLE 4.3 DBA Report Encoding

Queue Length	Binary Input (ONU)	Coding of Octet	Binary Output (OLT)
0–127	`00000000abcdefg`	`0abcdefg`	`00000000abcdefg`
128–255	`00000001abcdefx`	`10abcdef`	`00000001abcdef1`
256–511	`0000001abcdexxx`	`110abcde`	`0000001abcde111`
512–1023	`000001abcdxxxxx`	`1110abcd`	`000001abcd11111`
1024–2047	`00001abcxxxxxxx`	`11110abc`	`00001abc1111111`
2048–4095	`0001abxxxxxxxxx`	`111110ab`	`0001ab111111111`
4096–8191	`001axxxxxxxxxxx`	`1111110a`	`001a11111111111`
>8191	`01xxxxxxxxxxxxx`	`11111110`	`011111111111111`
Invalid	N/A	`11111111`	N/A

The DBA report comprises 1 byte reporting the entire queue backlog, (mode 0) or 2 bytes reporting green and yellow backlog separately (mode 1). Section 6.3 discusses colored traffic.

It is not specified whether the DBA report should include or exclude the current allocation. The ambiguity is recognized in G.984.3, which requires the ONU to be consistent in its reporting practice, and requires the OLT to tolerate both on an ONU by ONU basis.

Table 4.3 shows the coding of the DBA report octets. The letters a, b, . . . g represent bits that express the next-to-most significant part of the queue length (except for backlog <127, the most significant bit is always 1). Bits shown as x are unspecified: they may be either 0 or 1.

The idea of this compact logarithmic representation is that fine-grained reporting is less important as the queue backlog grows larger. This is appropriate because we do not know exactly how much overhead will be needed for GEM framing or DBR or for new traffic that may arrive before we get a grant for the existing backlog.

To illustrate, suppose the ONU has a backlog of 753 bytes; the OLT will interpret our backlog report to be 767. This 2% overestimation is typical of the accuracy to be expected. Because we need extra overhead for GEM headers and such, it is also appropriate that the OLT overestimate the backlog.

	Pattern	Example Value	Decimal
Backlog at ONU	`000 001a bcdx xxxx`	`000 0010 1111 0001`	`753`
Coding of octet	`1 110a bcd`	`1 1100 111`	
Binary output (OLT)	`000 001a bcd1 1111`	`000 0010 1111 1111`	`767`

Upstream Burst Structure Figure 4.27 shows the concatenation of the burst overhead and burst header with the allocation intervals to form the complete burst, ready for FEC and scrambling. The burst overhead exists to permit the OLT receiver to set the sampling threshold and phase correctly and establish byte delineation; the

UNCORRECTED PROOF

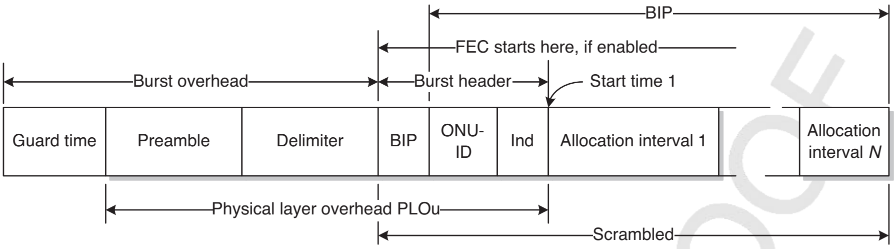

Figure 4.27 G.984 G-PON burst structure.

physical layer overhead PLOu is the complete set of fields transmitted by the ONU prior to the individual allocations. The guard time is not part of PLOu because it does not represent a transmission, merely an elapsed time.

Yes, all these overlaps are confusing. Let us first discuss the burst header, then the burst overhead:

BIP This bytewide field represents the bit-interleaved parity of the burst, excluding preamble, delimiter, and FEC (about which, more below). The BIP computed in this current burst appears in the header of the next burst from the same ONU. This is clumsy because we do not know the quality of the current burst until we receive the next burst, which may be some arbitrary time in the future. It was revised in G.987 XG-PON, whose BIP appears in the burst trailer.

ONU the unique 8-bit identifier of the ONU on the PON (TC-layer ONU-ID).

Ind The bits of this byte are assigned as follows:

7 When set, this bit requests a PLOAMu grant from the OLT: The ONU has something to send. Originally, this bit was defined to indicate that an urgent PLOAM message was waiting, but there are *no urgent* PLOAM messages—or from another viewpoint, there are *no non*urgent PLOAM messages—so urgency was removed from the definition.

6 When set, this bit indicates that upstream FEC is on. Recall that the ONU may disregard the OLT's request to use upstream FEC; this bit represents the reality of the burst.

5 RDI, remote defect indication. This bit informs the OLT of a signal failure in the downstream direction. Since most downstream signal failures prevent the ONU from transmitting, this bit is of limited value. Conceivably, it could signal high BER even after FEC correction; the criteria for setting the bit are not defined.

The other bits are reserved.

G.984 G-PON limits bursts to lie entirely within their respective upstream frames. In comparison, G.987 XG-PON bursts need only have a start time within the upstream frame interval.

UNCORRECTED PROOF

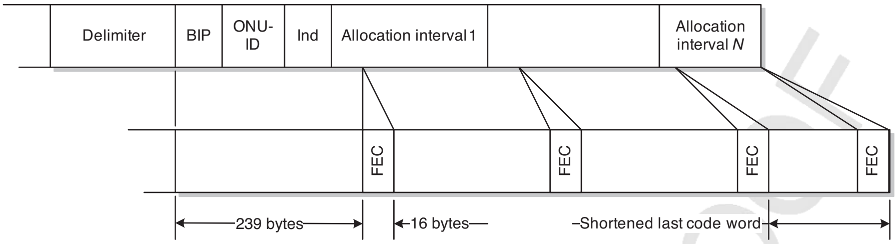

Figure 4.28 G.984 G-PON upstream FEC.

Forward Error Correction When FEC is enabled, the first code word begins with the BIP field. Every 239 bytes of data are followed by 16 bytes of FEC parity, if necessary with a shortened last code word at the end of the burst. Figure 4.28 illustrates the operation.

In G.984 G-PON, the FEC bytes are part of the allocation as defined by start and stop time fields in the BWmap. This implies certain rules in the construction of the BWmap. If the start time pointer of a contiguous allocation series were to point to one of the parity byte locations of a previous allocation, it could be interpreted as an attempt to preempt FEC parity, which is not permitted. It could also be interpreted as a request to start the next allocation on the first byte after the parity block or as a request for a shortened FEC code word. G.984 G-PON, therefore, requires the start time not to point to a parity byte.

Recall that in contiguous allocations, the stop time pointer is exactly one less than the subsequent start time. The restriction on start time value, therefore, implies that embedded stop time pointers are forbidden to point to the last byte before a parity block, or to any of the first 15 parity bytes.

The stop time rule is irrelevant at the end of the burst because there *is* no subsequent start time. The ONU creates a shortened last code word that includes the full complement of 16 parity bytes. The well-behaved OLT avoids a stop time that implies a shortened last code word of fewer than 16 bytes.

Scrambling After applying FEC, the upstream burst is scrambled. The scrambler is the same as is used downstream, with generator polynomial $x^7 + x^6 + 1$. It is preset to all ones on the first bit of the BIP field and runs continuously throughout the burst.

G.984 G-PON Burst Overhead Figure 4.29 reminds us of the burst overhead shown in Figure 4.27. The burst overhead is determined by the OLT, which periodically broadcasts upstream_overhead PLOAM messages and may also broadcast extended_burst_length messages as well. The extended burst length

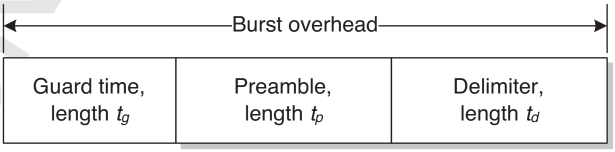

Figure 4.29 G.984 G-PON burst overhead.

UNCORRECTED PROOF

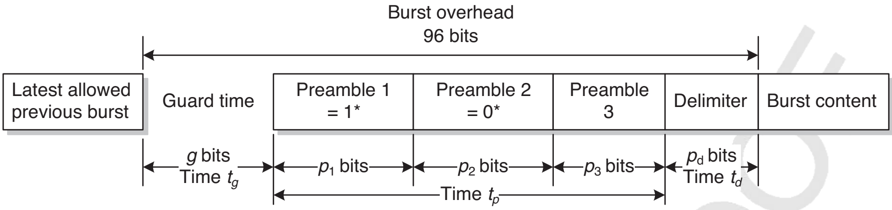

Figure 4.30 Definition of burst overhead fields.

allows for a longer burst header for ONUs that are attempting to register onto the PON, specifically a longer type 3 preamble (Fig. 4.30). Once they have been discovered and brought into service, ONUs use a different type 3 preamble byte count, which is also specified in the extended_burst_length message.

G.984.2 defines the length of the various fields, as shown in Table 4.4. The idea of guard time and the enable and disable times is the same as is illustrated in Figure 4.17.

The upstream_overhead PLOAM message specifies several fields that combine to form the burst overhead. The results are shown in Figure 4.30. Asterisks indicate repetition of a bit value.

g—Number of Guard Bits Guard time t_g is provided to allow time for transmitter turn-off and turn-on, for drift in equalization delay, and for tolerances in measuring and setting equalization delay. Except for computing the length of the type 3 preamble, it is not apparent why the ONU would care about guard time because the burst overhead and the start and stop times are wholly determined by the OLT.

P_1 and P_2

p_1 Number of type 1 preamble bits. Preamble 1 bits are all 1.

p_2 Number of type 2 preamble bits. Preamble 2 bits are all 0.

The p_1 and p_2 intervals are intended to give the OLT receiver a chance to set its decision threshold halfway between 1 and 0 values. Because the preamble 1 and 2 bits contain no transitions, they are useful only in setting the receiver gain and its decision threshold, but not in recovering timing. It is allowed to set either or both of p_1 and p_2 to zero, appropriate if the receiver prefers a preamble type 3 pattern instead.

TABLE 4.4 G.984 G-PON Burst Overhead Components

Component	Symbol	Units	Value
Guard time	t_g	Bits, min	32
Preamble time	t_p	Bits, suggested	44
Delimiter time	t_d	Bits, suggested	20
Total burst overhead	—	Bits, mandatory	96
Transmit enable	—	Bits, max	16
Transmit disable	—	Bits, max	16

UNCORRECTED PROOF

Pattern for Type 3 Preamble Bits one byte. This is likely to be a bit pattern such as 10101010, with a maximum transition density that allows the OLT to recover clock phase.

The length of preamble 3 is computed as

$$p_3 = 96 - g - p_1 - p_2 - p_d \text{ bits} \tag{4.1}$$

where

96 is the length of the burst overhead, as specified in G.984.2.

g, p_1, p_2, and p_d are as defined here.

Delimiter, p_d The delimiter, which has length p_d, is the OLT's choice of a pattern that provides robust byte delineation and frame start indication in the presence of bit errors. The upstream_overhead PLOAM message has space for 3 bytes of delimiter. If fewer than 3 bytes are needed, as suggested in Table 4.4, they may be provisioned to extend the preamble type 3 pattern.

As well as burst header information, the upstream_overhead PLOAM message contains various additional information, not pertinent to this discussion. Appendix II contains the details.

The extended_burst_length PLOAM message, also expanded in Appendix II, is intended to allow the OLT to specify a longer burst overhead for ONUs in the discovery process. The primary content of the extended_burst_length message is a specification of the number of type 3 preamble bytes to transmit while the ONU is in its serial number and ranging states. As a secondary feature, the extended_burst_length message also allows the specification of type 3 preamble repetition for the ONU in operation state. This is an escape route from Eq. (4.1), which may impose an unduly restrictive upstream_overhead message format, especially for ONUs behind a reach extender.

4.2 ONU DISCOVERY

We now have a good view of how payload is mapped into frames, along with the various chunks of necessary overhead, and how the frames are placed onto the fiber. Timing is an important aspect of upstream burst transmission. We discuss timing in Section 4.3, but it may help to understand the details of timing if we first consider the process of discovering an ONU that newly appears on the PON. The discovery and initialization process requires timing measurement and delay adjustment.

We first outline the basic sequence of steps to bring an ONU onto a working PON, then go back and discuss the variations. States and transitions for G.987 XG-PON are shown in Figure 4.31, with G.984 G-PON in Figure 4.32. We consider G.987 XG-PON first, then G.984 G-PON.

4.2.1 G.987 XG-PON Discovery and Ranging

1. When the ONU powers up and initializes, it must remain silent. Anything it transmitted on the PON could disrupt existing traffic. In initial state O1, the

UNCORRECTED PROOF

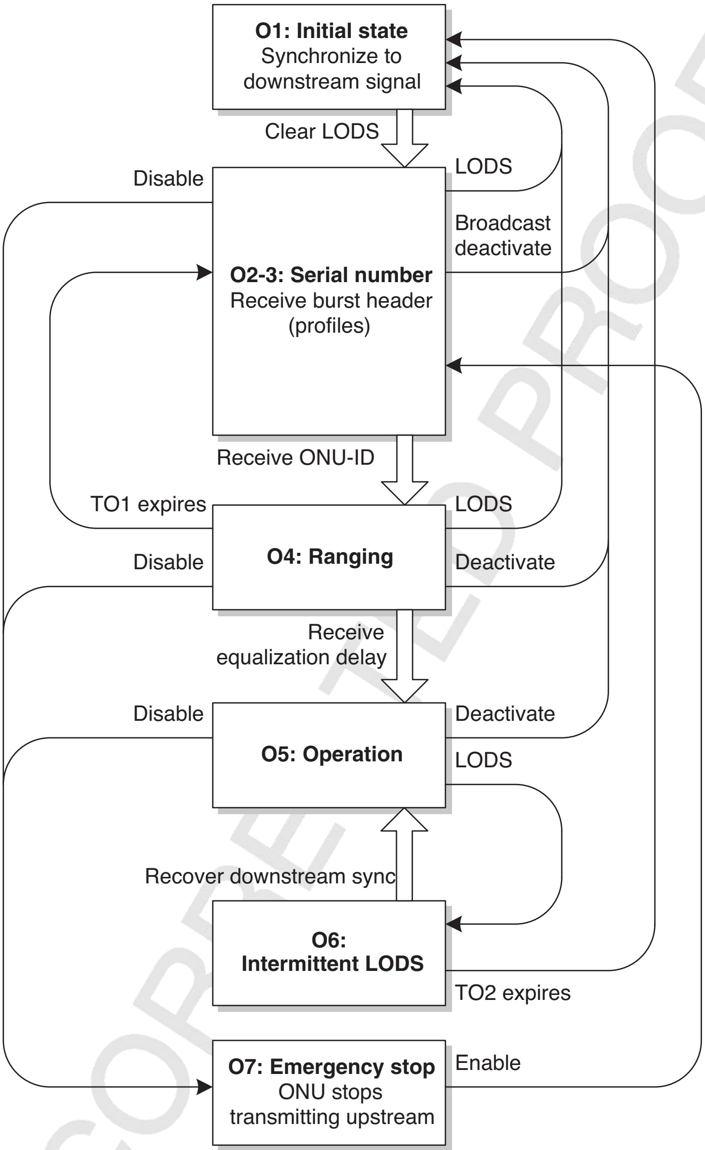

Figure 4.31 G.987 XG-PON ONU states.

UNCORRECTED PROOF

ONU synchronizes itself to the downstream signal, acquiring bit and frame synchronization and frame count. Then it enters serial_number state O2-3 to listen quietly for the burst header information it will need before it can properly sign on.

The ONU also clears, or discards, all TC-layer information it may have had: its ONU-ID, equalization delay, burst profiles, encryption keys, and so

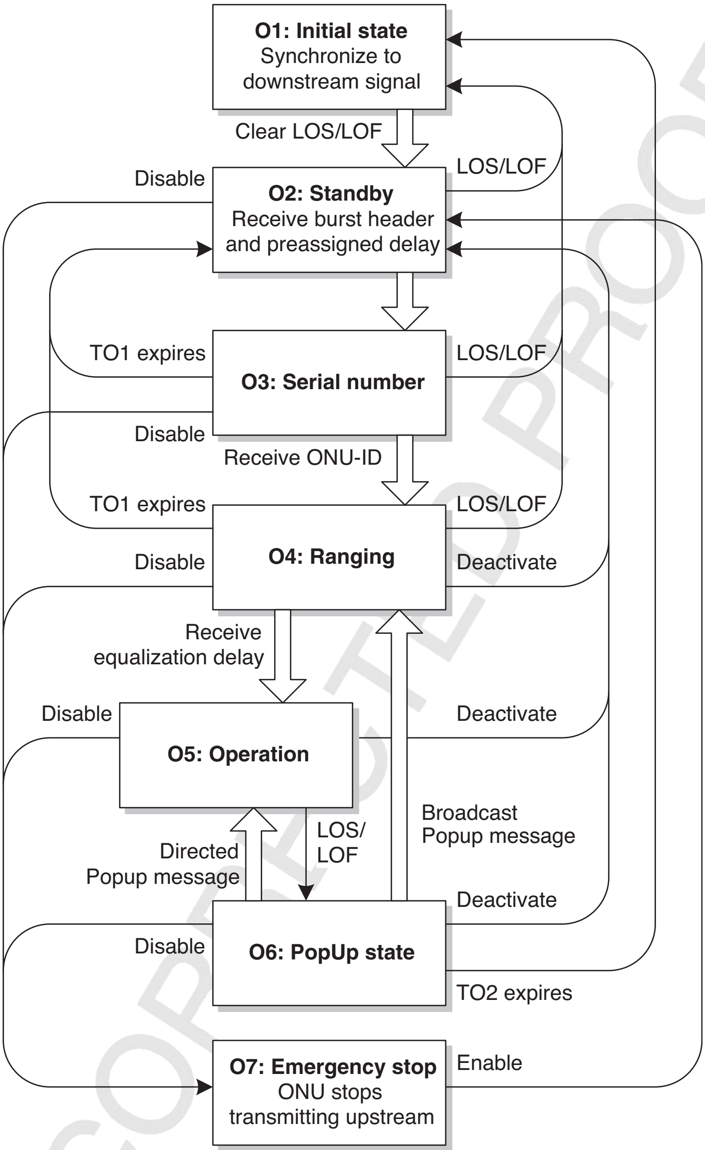

Figure 4.32 G.984 G-PON ONU states.

UNCORRECTED PROOF

forth. The specification is somewhat unclear about whether this occurs only in state O1, or whether it occurs on entry or exit to states O2 or O7.

2. From time to time, the OLT broadcasts header information to define the parameters that ONUs must have before they can attempt to join the PON. In

G.987 XG-PON, this information takes the form of a *profile* PLOAM message, discussed in detail in Appendix II.

3. Once it has acquired one or more profiles, the ONU waits for a serial_number grant that specifies one of the profiles in its repertoire. The ONU is never permitted to transmit if it does not know the specified profile.
4. From time to time, the OLT withholds transmit permission from all known ONUs, thereby opening a so-called quiet window in the upstream direction. Section 4.3 discusses the factors affecting the size and placement of the quiet window. For this window, the OLT generates a so-called serial_number grant, an invitation for any ONU in serial_number state to respond. A serial number grant is a BWmap allocation directed to alloc-ID 1023, requesting a PLOAM message upstream, and allocating zero data capacity.
5. More than one ONU can be waiting to sign on at any given time, and they can be at nearly the same distance from the OLT. If they all transmitted at the same time, their transmissions would collide at the OLT (see Fig. 4.38) and no one would be recognized. Worse, this deterministic collision would recur forever. Therefore, the ONU delays its response by some random value; transmissions may collide during this particular discovery cycle, but sooner or later they will be distinct and the OLT will recognize one of them, then the other.

   The random delay takes on values from 0 to 48 μs; the size of a serial number burst varies depending on the profile settings, but is typically less than 1 μs, so collisions are likely to be resolved reasonably soon, even if several ONUs simultaneously contend for recognition (e.g., when the entire PON is being initialized).
6. The OLT recognizes the serial_number_ONU response from the ONU and assigns a TC-layer ONU-ID to that serial number. If the ONU sees another serial number grant before it gets an ONU-ID, it deduces that it has not been recognized. It generates a new random delay and tries again.
7. The OLT can, and probably does, measure the round-trip delay from the serial_number response PLOAM message. Since the response conveniently includes the value of random delay used by the ONU, round-trip delay is easy to derive.
8. The ONU accepts the ONU-ID as its own and henceforth uses this value for its PLOAM communications as well as for its default alloc-ID for bandwidth grants and OMCI. Gaining a TC-layer ONU-ID allows the ONU to enter ranging state O4.
9. According to the script, the OLT should now measure the propagation and processing delay of the newly assigned ONU. The OLT opens another quiet window and invites the newly assigned ONU to transmit again. The ranging window may be as wide as the serial number window, less an allowance for random delay, which is not present in this step. In practice, if the OLT has even an approximate value for the delay, and merely wishes to make a second, precise measurement, the OLT need not open a full window.

UNCORRECTED PROOF

This grant is called a ranging grant, primarily distinguished because the ONU is in ranging state O4 when it happens. The ranging grant is directed to the newly assigned alloc-ID (ONU-ID); it contains no allocation for payload, only for an upstream PLOAM message. An ONU in ranging state O4 responds with its registration ID, and with no random delay.

10. Here or at serial number acquisition, the OLT measures the delay precisely, computes an offsetting value called equalization delay, and transmits the equalization delay value to the ONU. Henceforth, the ONU delays all of its transmissions by the equalization delay value. Section 4.3 explains equalization delay in detail.
11. Reception of an equalization delay puts the ONU into operation state O5, where it is now ready for service. The OLT is free to audit the ONU's MIB and provision it for service.

Consider a variation on this theme. If the OLT already knew a lot about the ONU—if, for example, the ONU fell off the PON because the subscriber powered it down—the OLT could blindly issue PLOAM messages to assign a TC-layer ONU—ID to the known serial number and set the known equalization delay, whereupon the ONU could go into operation state O5 without ever having responded to either a serial number or a ranging grant.

But see also the discussion in Section 4.6.1 on the PLOAM MIC, which requires the registration ID to be known to the OLT. So unless the registration ID is known, the OLT must send a ranging grant, and the ONU must respond to it. In the case of a previously known ONU, the OLT may well know the ONU's registration ID, but building OLT software around the corner cases may be more trouble than just issuing a ranging grant.

States O6 and O7 are not part of the normal progression. The *intermittent loss of downstream sync* state O6 occurs when the ONU's frame synchronization state machine declares LODS. The ONU starts timer TO2. If TO2 expires, the ONU recognizes that it is no longer on the PON and restarts the initialization sequence from scratch. When the ONU returns to initial state O1, it discards its ONU-ID and all other TC-layer provisioning.

If a G.987 ONU recovers downstream synchronization before TO2 expires, it simply returns to operation state O5 on its own initiative and carries on. There is an assumption that, if the equalization delay is suddenly wrong because of a protection switch, the OLT knows about it and will take whatever action is appropriate. The PON ID field (Fig. 4.10) may also be useful for the ONU itself to detect a protection switch and avoid disruptive responses until it has been redirected by the OLT. Details of how this might work are for further study.*

UNCORRECTED PROOF

* The phrase *for further study* is ITU speak for an admission that a problem has been recognized but that no one wants to take the time and trouble of solving it now. Later, maybe, but only if it becomes important.

O7 is called the emergency stop state. It is intended as a holding pen for rogue ONUs. Assuming that the ONU is capable of responding, the broadcast downstream disable_serial_number PLOAM message puts it into state O7, where it is forbidden to transmit anything upstream. If the ONU receives a subsequent disable message whose action code point is set to enable, then and only then may it return to the serial number state O2-3 and go through the normal ranging process.

More formally, the states are defined as follows:

G.987 State O1, Initial

Behavior The ONU transmits nothing. In this state, the ONU waits for the presence of a downstream signal and then synchronizes to that signal. The ONU discards any TC-layer parameters that it may have known from its previous states: ONU-ID, default alloc-ID, profiles, equalization delay, encryption keys.

G.987.3 states that when the ONU enters O1, including entry from some other state, it should reset its downstream synchronization machine (Fig. 4.12). An ONU that already has downstream sync may well choose to skip this detail.

Exit Criterion Bit, byte, frame, and superframe synchronization achieved: LODS cleared → O2-3.

G.987 State O2-3, Serial Number

Behavior The ONU is never permitted to transmit if it does not know the profile specified in the bandwidth grant. When it sees a serial number grant that specifies a known profile, the ONU transmits a serial_number_ONU PLOAM response, delayed by a locally generated random value. In this state, the ONU continues to respond to serial number grants, each time with a different random delay.

This is the only state in which the ONU is permitted to respond to global bandwidth grants, also the only state in which it uses a random delay.

Exit Criteria

- Assign_ONU-ID PLOAM message → O4.
- LODS → O1.

G.987 State O4, Ranging

Behavior When it enters state O4, the ONU starts timer TO1, whose initial value is recommended to be 10 s.

In this state, the ONU regards any grant directed to itself as a ranging grant. The grant is expected to request an upstream PLOAM message and provide no payload allocation. The ONU responds with the registration PLOAM message. There is no random delay in this message; it is a response to a directed grant, so there is no possibility of collision with autonomous transmitters.

UNCORRECTED PROOF

Having sent the registration PLOAM message, the ONU waits for a ranging_time PLOAM message from the OLT, which establishes its equalization delay.

Exit Criteria

- Ranging_time PLOAM message → O5.
- Timer TO1 expires → O2-3.
- LODS → O1.

G.987 State O5, Operation

Behavior This is the normal state of ONU operation. The ONU responds to grants directed to it, delaying its response by the value of the equalization delay.

Exit Criterion LODS → O6.

G.987 State O6, Intermittent LODS

Behavior When it enters state O6, the ONU starts timer TO2, whose initial value is recommended to be 100 ms. The ONU transmits nothing. In this state, it has lost sync with the downstream signal.

Exit Criteria

- Timer TO2 expires → O1.
- Recovery of downstream sync, LODS cleared → O5.

G.987 State O7, Emergency Stop

Behavior The ONU transmits nothing. It remains in this state until explicitly reenabled, even when power cycled.

Exit Criterion Reception of a disable PLOAM message with enable code point set → O2-3.

Further Notes

- In any state, reception of a disable PLOAM message with a disable code point causes a transition into emergency stop state O7.
- In any state except O7, reception of an assign_ONU-ID PLOAM message that matches one, but not both, of the ONU's serial number and TC-layer ONU-ID causes a transition to O1.
- In any state except O7, reception of a deactivate PLOAM message causes a transition into initial state O1.

4.2.2 G.984.3 G-PON Discovery and Ranging

Comparing Figures 4.31 and 4.32, we see that the XG-PON and G-PON state machines are similar but not identical. The loss of downstream sync LODS of G.987 (Fig. 4.12) is effectively the same as the LOS/LOF of G.984 (Fig. 4.22) and is declared by the downstream synchronization state machine.

UNCORRECTED PROOF

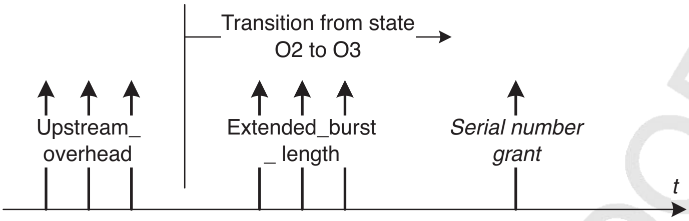

Figure 4.33 ONU discovery sequence, G.984 G-PON.

Because of the similarities, we abbreviate the discussion of G.984, focusing on the differences.

1. In initial state O1, the ONU synchronizes itself to the downstream signal, acquiring bit and frame synchronization and frame count. Then it enters standby state O2 to listen quietly for the burst header information it will need before it can properly sign on. The ONU also clears, or discards, all TC-layer information it may have had: its ONU-ID, equalization delay, burst profiles, and the like.
2. The OLT broadcasts header information to define the parameters that ONUs must have before they can attempt to join the PON. In G.984 G-PON, they are defined by the upstream_overhead and optional extended_burst_length PLOAM messages, discussed in Appendix II. The logical time to do this is just before the OLT issues a serial_number grant (Fig. 4.33). One of the parameters of the upstream_overhead message is a preassigned delay, which allows the OLT to adjust the relative position of the quiet window. G.987 XG-PON has no concept of a preassigned delay; the OLT positions the quiet window through its management of start time pointers.
3. When it acquires the burst header parameters, the ONU waits for a serial_number grant in state O3. By comparison with Figure 4.31, we see that states O2 and O3 were merged in XG-PON because the division served no purpose.
4. The OLT opens a quiet window during which it generates a serial_number grant, an invitation for any ONU in serial_number state to respond. A serial number grant is a BWmap allocation directed to alloc-ID 254, requesting a PLOAM message upstream, and allocating zero payload capacity.
5. The ONU delays its response by the preassigned delay value specified in the upstream_overhead PLOAM message plus a random delay in the range from 0 to 48 µs.
6. The OLT recognizes the serial_number_ONU response from the ONU, and assigns a TC-layer ONU-ID to that serial number. If the ONU sees another serial number grant before it gets an ONU-ID, it generates a new random delay and tries again.
7. The OLT can measure the round-trip delay from the serial_number response PLOAM message. Since the response conveniently includes the value of random delay used by the ONU, round-trip delay is easy to derive.

UNCORRECTED PROOF

8. The ONU accepts the ONU-ID as its own and, henceforth, uses this value for its PLOAM communications as well as for its default alloc-ID for bandwidth grants. Gaining a TC-layer ONU-ID allows the ONU to enter ranging state O4.
9. The OLT now opens another quiet window and issues a ranging grant, primarily distinguished because the ONU is in ranging state O4 when it happens. The ranging grant is directed to the default alloc-ID of the newly assigned ONU; it contains no allocation for payload, only for an upstream PLOAM message. The G.984 ONU in ranging state O4 repeats the serial_number message, delayed only by the preassigned delay, with no random delay. The G.984 ONU uses the preassigned delay only in the serial_number and ranging states.
10. Here or at serial number acquisition, the OLT measures the delay precisely, computes an offsetting equalization delay, and transmits the equalization delay value to the ONU. Henceforth, the ONU delays all transmissions by the equalization delay value.
11. Reception of an equalization delay puts the ONU into operation state O5, where it is now ready for service.

If the OLT already knew a lot about the ONU—if, for example, the ONU fell off the PON because the subscriber powered it down—the OLT could blindly issue PLOAM messages to assign a TC-layer ONU-ID to the known serial number and set the known equalization delay, whereupon the ONU could go into operation state O5 without ever having responded to either a serial number or a ranging grant. The discussion in Section 4.6.1 on the PLOAM MIC does not pertain to G.984 G-PON, which has no MIC; it is a feature only of G.987 XG-PON.

The PopUp state O6 occurs when the ONU's downstream synchronization state machine (Fig. 4.22) declares downstream LOS/LOF. The ONU starts timer TO2. If TO2 expires, the ONU recognizes that it is no longer on the PON and restarts the initialization sequence from scratch. When the ONU returns to initial state O1, it discards its ONU-ID and all other TC-layer provisioning.[*]

Assuming that the ONU recovers synchronization and can receive PLOAM messages, the G.984 OLT can return the ONU to operation state O5 by sending a PopUp message to that specific ONU—it may not help, but it does not hurt anything. The G.984 OLT can also send a broadcast PopUp message, which causes all ONUs in state O6 to enter ranging state O4. This was intended to allow fast recovery in case of protection switches. The PopUp state changed its name and behavior in G.987 XG-PON, and the PopUp PLOAM message was dropped in favor of a broadcast ranging message containing a delay offset.

UNCORRECTED PROOF

[*] G.984.3 is less explicit than G.987 about details such as resetting the TC-layer parameters. The principles make sense, however, in G-PON as well as in XG-PON.

State O7, the emergency stop state, behaves much the same in G.984 as in G.987.

The states are more formally defined as follows, still focusing on differences from G.987 XG-PON:

G.984 State O1, Initial

Behavior The ONU transmits nothing. In this state, the ONU waits for the presence of a downstream signal and then synchronizes to that signal. The ONU discards any TC-layer parameters that it may have known from its previous states: ONU-ID, default alloc-ID, burst header parameters, equalization delay. Although this discard action is only spelled out explicitly in G.987.3, it also makes sense in G.984 G-PON ONUs.

Exit Criterion Bit, byte, frame, and superframe synchronization achieved, LOS/LOF cleared → O2.

G.984 State O2, Standby

Behavior The ONU transmits nothing. In this state, the ONU listens for the broadcast upstream_overhead PLOAM message, which defines the basic parameters of the bursts it will subsequently transmit, including a value for preassigned delay.

If the OLT also broadcasts an extended_burst_length PLOAM message (described in Appendix II), the ONU modifies its response parameters accordingly. This message allows for a longer burst header during ONU discovery—the unranged type 3 preamble bytes field—and also after the ONU is operational—the ranged type 3 preamble bytes field. The idea is that a marginal optical budget may become usable if the ONU's preamble is extended; in particular, a reach-extended ONU is almost certain to require an extended burst overhead, for reasons explained in Section 3.12. The ONU is permitted to ignore this message if it does not support the feature, so in theory, the OLT's effort could be in vain. In practice, current ONUs are expected to support extended burst length capability.

If a newly initialized ONU does not see an extended_burst_length message before it responds to a serial number grant, it ignores subsequent extended_burst_length messages that may appear later on. The intended state and message sequence of G.984 G-PON ONU discovery is quite deterministic, as shown in Figure 4.33.

The OLT is expected to transmit three upstream_overhead messages, followed optionally by three extended_burst_length messages, and only then to issue one or more serial number grants. If a new ONU appears on the PON in time to receive at least one of the upstream_overhead messages, it can be assured of receiving an extended_burst_length message before having the opportunity to respond to a serial number grant.

Exit Criterion Upstream overhead PLOAM received → O3. State transition does *not* depend on having received an extended_burst_length PLOAM message.

UNCORRECTED PROOF

G.984 State O3, Serial Number

Behavior The ONU responds to a serial number grant by generating a serial_number_ONU response PLOAM message, delayed by the preassigned delay plus a random locally generated delay. In state O3, the ONU continues to respond to serial number grants, each time with a different random delay. This is the only state in which the ONU is permitted to respond to global bandwidth grants, also the only state in which it uses a random delay.

Exit Criterion Assign_ONU-ID PLOAM message received → O4.

G.984 State O4, Ranging

Behavior When it enters state O4, the ONU starts timer TO1, whose initial value is recommended to be 10 s. In G.984 G-PON, the response to a ranging grant is another serial_number PLOAM message, delayed by the preassigned delay value. There is no random delay in this message since it is a response to a directed grant.

Exit Criteria

- Ranging_time PLOAM message received → O5.
- Timer TO1 expires → O2.

G.984 State O5, Operation

Behavior This is the normal state of ONU operation. The ONU responds to grants directed to it, delaying its response by the value of the equalization delay.

Exit Criterion Loss of downstream signal/frame LOS/LOF → O6.

G.984 State O6, PopUp

Behavior When it enters PopUp state O6, the ONU starts timer TO2, whose initial value is recommended to be 100 ms. When it enters this state, the ONU has lost sync with the downstream signal. If it remains unsynchronized with the downstream flow, the only exit from state O6 is through expiration of timer TO2. However, while in state O6, the ONU may clear LOS/LOF, whereupon it resumes downstream processing, and can then recognize downstream PLOAM messages.

Whether or not the ONU has downstream sync, it transmits nothing while in state O6.

Exit Criteria

- Timer TO2 expires → O1.
- Directed PopUp PLOAM message received → O5.
- Broadcast PopUp PLOAM message received → O4.

G.984 State O7, Emergency Stop

Behavior The ONU transmits nothing. It remains in this state until explicitly reenabled.

UNCORRECTED PROOF

Exit Criterion Disable PLOAM received, with enable code point set → O2.

Further Notes

- In any state, reception of a disable PLOAM message with a disable code point causes a transition into emergency stop state O7.
- In any state except O7, reception of a deactivate PLOAM message causes a transition into standby state O2.

4.3 ONU TRANSMISSION TIMING AND EQUALIZATION DELAY

This section describes the timing relationships of G.987 XG-PON in detail, and discusses the differences between XG-PON and G.984 G-PON. It is based on the following definitions.

1. The start time of the downstream frame is the moment of transmission/ reception of the first bit of the PSync field.
2. The start time of the upstream PHY frame is the moment of transmission/ reception (either actual or calculated) of the first bit of the word identified by a StartTime pointer of zero value.
3. The start time of an upstream PHY burst is the moment of transmission/ reception of the first bit of the word identified by the StartTime of the corresponding bandwidth allocation structure. This is the first bit of the XGTC burst header in G.987 XG-PON (Fig. 4.18). In G.984 G-PON, it is the first bit following the physical layer overhead PLOu (Fig. 4.27), which may be an upstream PLOAM message, a DBA queue occupancy report DBR, or a GEM frame header, depending on the flags in the BWmap that authorized the burst.

4.3.1 Fundamentals

We start with some fundamental concepts and notation. The terminology—PSBd, for example—is preferentially taken from G.987. The concepts are common to both G-PON and XG-PON.

Consider Figure 4.34.

Suppose that the OLT transmits frame N onto the fiber, starting at time $t_{\mathrm{send}N}$. If we care about time of day (ToD), we require that $t_{\mathrm{send}N}$ be an absolute time, that is, some specific date and time. Otherwise, $t_{\mathrm{send}N}$ is just a convenient reference. In either case, all of the other times are delays, most of them relative to $t_{\mathrm{send}N}$.

Frame N propagates down the fiber to various ONUs, located at varying distances L from the OLT. When we engineer the optical distribution network ODN, we need to specify the distance of the farthest possible ONU that will need to be accommodated. Designate this hypothetical ONU as $\mathrm{ONU}_{\mathrm{max}}$, at distance L_{max}. After propagation delay t^{D}_{max}, frame N reaches $\mathrm{ONU}_{\mathrm{max}}$. If there were a real ONU here at L_{max}, it would perceive frame N to start at time $t_{\mathrm{send}N} + t^{D}_{\mathrm{max}}$.

UNCORRECTED PROOF

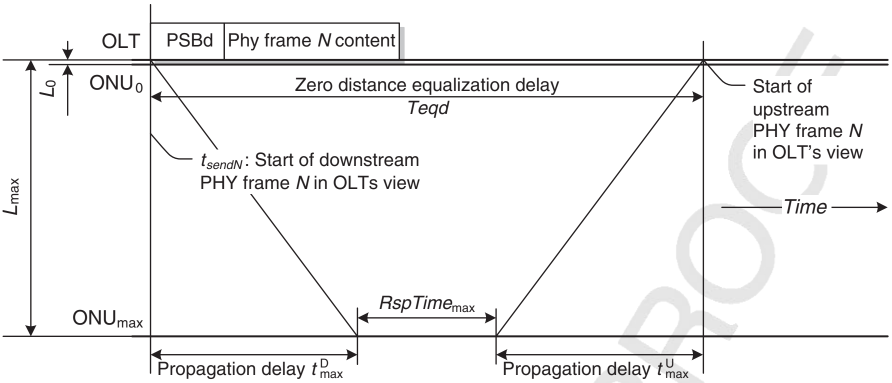
OLT
PSBd
Phy frame N content
L_0
ONU0
Zero distance equalization delay
Teqd
t_{sendN}: Start of downstream PHY frame N in OLTs view
Start of upstream PHY frame N in OLT's view
Time
L_{max}
ONUmax
$RspTime_{max}$
Propagation delay t^{D}_{max}
Propagation delay t^{U}_{max}

Figure 4.34 Basic timing concepts.

Now, a real physical ONU cannot recognize a downstream frame, interpret its BWmap, and generate a corresponding upstream frame in zero time—the content of upstream frame *N* depends on the BWmap information received in downstream frame *N*. We define an ONU response time RspTime to account for this delay. Further, we standardize its value to be 35 ± 1 μs, meaning that all real ONUs are expected to be able to respond within 36 μs, and if an ONU is capable of responding in less than 34 μs, it must introduce additional delay.

ONU_{max} now generates upstream frame *N*, which is guaranteed to start not more than $\text{RspTime}_{\text{max}} = 36\,\mu\text{s}$ later. After upstream propagation delay t^{U}_{max}, the OLT receives frame *N*. At this time, the OLT can be sure that upstream frame *N* will have arrived, sent no matter from which ONU. Let Teqd be the offset in time between the OLT's view of the downstream and the latest possible upstream frames.

Consider a second hypothetical ONU_0 located at zero distance L_0. It is apparent that, if ONU_0 were to generate upstream frame *N* based only on its own RspTime, the frame would arrive at the OLT far too early. The OLT, therefore, compensates by provisioning an additional delay into each ONU, a delay known as equalization delay, EqD. The total delay for an ONU at zero distance is called the zero-distance equalization delay, and is equal to Teqd. The ONU's actual RspTime is implicit in the delay measurement and compensation.

The equalization delay for our hypothetical ONU_{max} with 36 μs RspTime would be 0, and we see from Figure 4.35 that for any ONU *i*, at intermediate distance L_i, there will be some equalization delay EqD_i that lies between 0 and Teqd—more precisely, between 0 and ($\text{Teqd} - \text{RspTime}_{\text{min}}$). By the way, since the OLT knows the propagation delay, it is trivially easy for the OLT to compute and display the ONU's distance, as long as we do not expect really precise results (1% accuracy is the objective stated in the recommendations).

To summarize: the OLT derives the zero-distance equalization delay Teqd based on the provisioned maximum reach of the PON. As it initializes ONUs onto the PON, the OLT compensates each ONU such that its upstream frames arrive at the OLT aligned to a common reference, which could be later than $t_{\text{send}N} + \text{Teqd}$, but not earlier. Strictly speaking, we should, therefore, show Teqd_{min} and $\text{Teqd}_{\text{actual}}$ as two distinct delays; but to remain consistent with the description in the standards, we

UNCORRECTED PROOF

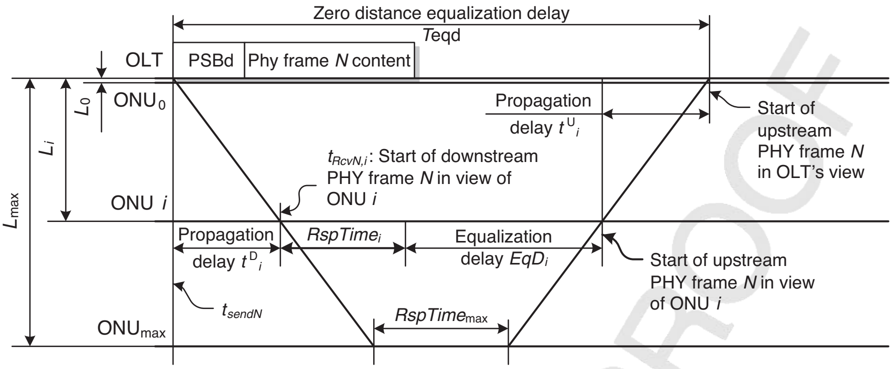
Zero distance equalization delay
Teqd
OLT
PSBd
Phy frame N content
L_0
ONU$_0$
L_i
L_{max}
Propagation
delay $t^U{}_i$
Start of upstream PHY frame N in OLT's view
$t_{RcvN,i}$: Start of downstream PHY frame N in view of ONU i
ONU i
Propagation delay $t^D{}_i$
$RspTime_i$
Equalization delay EqD_i
Start of upstream PHY frame N in view of ONU i
t_{sendN}
$RspTime_{max}$
ONU$_{max}$

Figure 4.35 Values for ONU *i*.

show them as a single value Teqd, with the understanding that a constant additional delay does not affect the ultimate result. Another way to think of it is that $L_{\max}$ can be chosen arbitrarily, as long as it is at least as great as the actual intended reach.

We have defined separate designations for downstream and upstream propagation delays. Although they are indeed approximately the same, it is also true that the group velocity of light in fiber is a function of wavelength, and since the downstream and upstream signals on a G-PON are at different wavelengths, their propagation velocity differs. Not by very much, but if we care about nanoseconds over a reach of 20 km or more—and we do—it matters. To briefly reprise the discussion in Section 3.1: The index of refraction n is a measure of the speed of light v in a particular medium, in comparison to the speed of light c in free space:

$$n = \frac{c}{v} \tag{4.2}$$

Let the downstream group index of refraction be n_D, and the upstream index be n_U. To illustrate the concept, suppose that we sent a signal downstream through a fiber whose refractive index n_D was 1.5, and that we returned the signal upstream through a radio link, through the air, whose refractive index n_U is essentially 1. We would know that 60% of the total round-trip delay was consumed in the downstream direction, 40% upstream. Denote the fraction of round-trip delay claimed by downstream propagation as η, where $\eta = n_D/(n_D + n_U)$. Table 4.5 shows the numbers for G.652 fiber at the G-PON and XG-PON wavelengths. The results are derived in appendices of G.987.3 and G.984.3, respectively, along with estimates of the error.

At 20 km, the difference is about 30 ns (G-PON), about 55 ns (XG-PON), with an expected tolerance of 3 or 4 ns. We shall see η again, as a factor in the fine detail of the time-of-day feature.

4.3.2 Time of Day

By virtue of the physical layer itself, an ONU is always frequency synchronized to the OLT. Historically, it was not necessary for an ONU to know date or time. The

UNCORRECTED PROOF

TABLE 4.5 Propagation Velocity Differences

	G.987 XG-PON	G.984 G-PON
Downstream wavelength λ_D	1577 nm	1490 nm
Upstream wavelength λ_U	1270 nm	1310 nm
Downstream fraction η	50.0134%	50.0065%[a]

[a] The body of G.984.3 amendment 2 suggests 50.0085%, but the appendix recommends 50.0065% as the value to be assumed for compatible calculations across the industry. Q4

OLT took care of time stamps on performance monitoring (PM) or alarm messages. However, G-PON has been extended to backhaul cellular radio traffic, an application that may require ToD to be available at the ONU, depending on the wireless protocol and whether the radio base station contains a separate reference such as GPS. The accuracy requirement on ToD information is the consequence of an allocation of impairments, and the ONU has been pragmatically allocated ±1 μs. Continuing analysis may require that this tolerance be tightened up, perhaps to as little as 100 ns.

IEEE 1588 is a packet timing protocol with the potential to deliver accurate time and frequency information over a packet network. However, the straightforward packet timing algorithms measure round-trip delay and compensate the far end under the assumption that the delay is symmetric and stable. On a PON, upstream bandwidth allocation is driven by the DBA algorithm and tends to run independently of the downstream flow. It is not inconceivable that downstream queuing and the DBA algorithm could be aligned to achieve symmetric and stable delay, but the industry has chosen instead to distribute ToD as a separable feature of the PON TC layer. Here is how it works:

The OLT is assumed to have its own source of accurate ToD information—IEEE 1588, for example, or GPS. The ToD distribution mechanism uses an attribute of the OLT-G managed entity of OMCI. Whenever it initializes an ONU that needs ToD, and from time to time at its own discretion, the OLT sends down the value of a reference frame count N (in G.987 XG-PON, the 32 least significant bits of the 51-bit frame counter), along with a time stamp $T_{\text{stamp}N}$, expressed in IEEE 1588 format and with nanosecond resolution. Refer to Figure 4.36.

During normal operation, ONU i has an equalization delay EqD_i set by the OLT in such a way that upstream bursts from ONU i align with bursts from all other ONUs when they reach the OLT. For ToD distribution, assume that ONU i is properly compensated and operating normally.

As an aid to the following derivation, extend the diagonal time lines of Figure 4.34 past $\text{ONU}_{\max}$ until they meet at a hypothetical reflector at time $T_{\text{stamp}N}$, corresponding to an imaginary ONU with zero RspTime. The downstream propagation delay can be allocated from the total delay according to η (Table 4.5). Teqd and $t_{\text{send}N}$ are known to the OLT, so it can compute $T_{\text{stamp}N}$:

$$\begin{aligned} T_{\text{stamp}N} &= t_{\text{send}N} + t^D_{\text{Ref}l} \\ &= t_{\text{send}N} + \Delta_{\text{OLT}} \\ &= t_{\text{send}N} + \eta \cdot \text{Teqd} \end{aligned} \tag{4.3}$$

UNCORRECTED PROOF

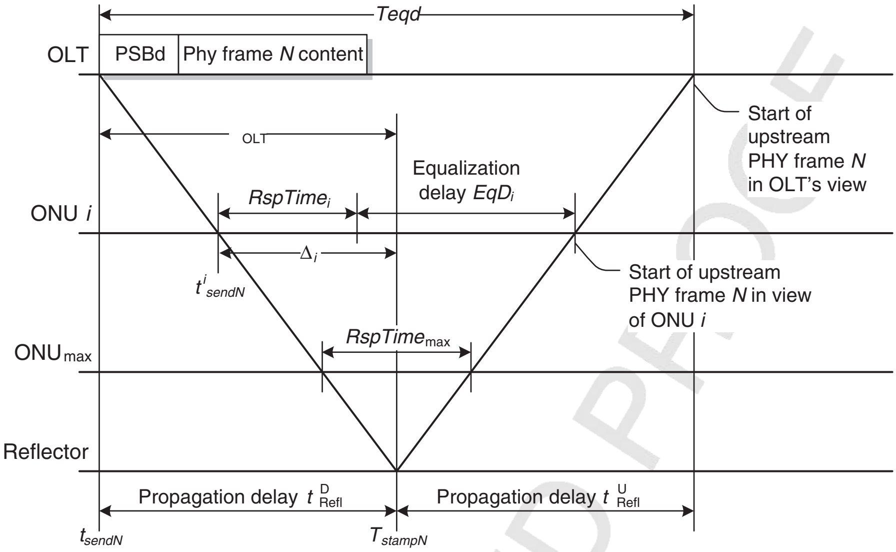

Figure 4.36 Time-of-day computation.

In this application, $t_{\mathrm{send}N}$ is an absolute time, so $T_{\mathrm{stamp}N}$ is also absolute.

The OLT now sends the pair $\{N, T_{\mathrm{stamp}N}\}$ to any or all ONUs via OMCI.

Our objective is to have an accurate value for $T_{\mathrm{stamp}N}$ everywhere throughout the PON. At our imaginary reflector, $T_{\mathrm{stamp}N}$ marks the time of arrival of frame N. But if ONU i were to make the same assumption at time $t^i_{\mathrm{send}N}$, it would be too early. So ONU i needs to derive an offset Δ_i from $t^i_{\mathrm{send}N}$, the instant of arrival of frame N, that corresponds to $T_{\mathrm{stamp}N}$.

ONU i does not know the propagation delays, but it does not need to. Since all triangles in Figure 4.36 are geometrically similar, the ONU need only apply the factor η to the delays it *does* know, namely $\mathrm{RspTime}_i$ and EqD_i. From the perspective of ONU i, frame N arrives at $t^i_{\mathrm{send}N}$, and

$$\begin{aligned} T_{\mathrm{stamp}N} &= t^i_{\mathrm{send}N} + \Delta_i \\ &= t^i_{\mathrm{send}N} + \eta(\mathrm{RspTime}_i + \mathrm{EqD}_i) \end{aligned} \tag{4.4}$$

Observe that frame N need not be in the future or even in the recent past. The ONU can calculate the correct time of day by offsetting the current frame number M by $(M - N)^* \, 125\,\mu\mathrm{s}$.

If the OLT changes the equalization delay EqD_i at some time, due to drift or possibly a protection switch, ONU i retains the time of day derived from its local counter. Changing the delay does not by itself reset the clock. When the OLT someday sends a subsequent ToD calibration update, it would be expected that the altered physical delay would be exactly compensated by the η-weighted amount of the new correction, and the ToD clock would not incur a phase hit.

UNCORRECTED PROOF

Once the ONU knows what time it is, it needs to convey this information to the radio base station. A new IEEE 1588 secondary clock could be instantiated on the ONU; if the radio base station interface is Ethernet, this is probably the most straightforward solution. For non-Ethernet backhaul, no standard interface existed at the time of writing, although the question was under study. A number of existing devices, especially GPS receivers, implement a so-called 1 pulse per second (PPS) interface. This interface uses the edge of an RS-422 signal to designate the start of each second and a message to carry time. It has been called out as a requirement by some operators and is a likely candidate for standardization.

Multiple clock domains—that is, the ONU's ability to honor multiple opinions about date and time—are out of scope. Various options could be pursued if a network client were to require a distinct ToD from that of the network provider, but the question has not arisen in practice.

Having described the standard solution, we mention that other mechanisms have also been proposed for ToD transfer, such as a transparent clock, in which corrections are added by each equipment through which the IEEE 1588 message passes.

4.3.3 ONU Upstream Burst Timing

As we have seen, ONU transmissions are defined relative to the start of the downstream PHY frame carrying the bandwidth map that authorizes the upstream burst. That is, the bandwidth map in downstream frame *N* governs the burst structure of upstream frame *N*. The response time RspTime gives the ONU time to decode the bandwidth map and prepare a response burst accordingly.

Is There Enough Time to Decode the BWmap?

A G.987 XG-PON downstream frame can contain a maximum of 512 eight-byte bandwidth allocation structures, which implies a transmission time of up to 3.3 μs. The BWmap is transmitted before any possible PLOAM messages to give the ONU the maximum possible headroom in preparing a response. The G.987 BWmap is structured in increasing order of start time, giving the ONU further time in which to decode the earliest allocations.

In the G.984 G-PON downstream frame, PLOAM comes before the BWmap, along with several other fields (Fig. 4.20), but there is only one PLOAM message, of only 13 bytes, the other fields are small, and the BWmap is limited to 256 allocation structures. The latest allocation in a BWmap would be transmitted in a bit more than 6.6 μs.

These up-front delays do not seriously impinge on the ONU's 36-μs maximum response time limit.

UNCORRECTED PROOF

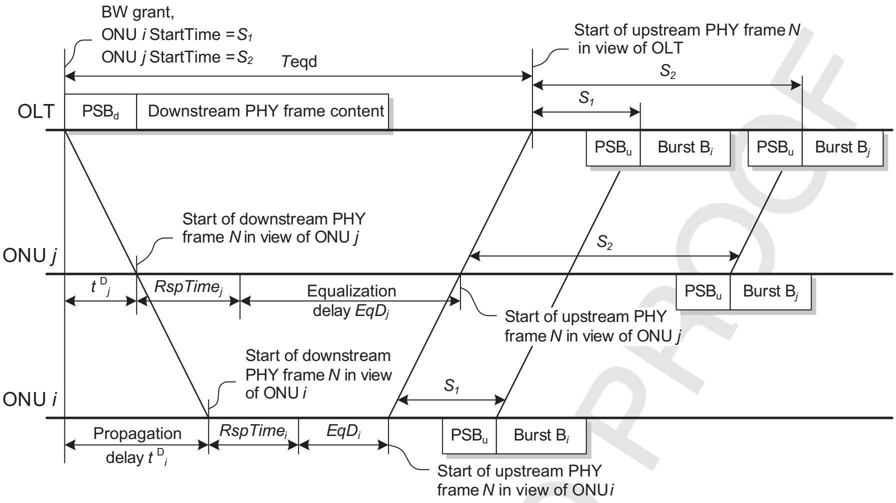
BW grant,
ONU *i* StartTime = S_1
ONU *j* StartTime = S_2
*T*eqd
Start of upstream PHY frame *N* in view of OLT
S_2
S_1
OLT
PSB_d
Downstream PHY frame content
PSB_u
Burst B_i
PSB_u
Burst B_j
Start of downstream PHY frame *N* in view of ONU *j*
S_2
ONU *j*
t^D_j
$RspTime_j$
Equalization delay EqD_j
PSB_u
Burst B_j
Start of upstream PHY frame *N* in view of ONU *j*
Start of downstream PHY frame *N* in view of ONU *i*
S_1
ONU *i*
Propagation delay t^D_i
$RspTime_i$
EqD_i
PSB_u
Burst B_i
Start of upstream PHY frame *N* in view of ONU *i*

Figure 4.37 ONU timing diagram, general case.

Each ONU maintains a running upstream frame counter that is synchronized to the downstream frame count, but delayed by a precise amount, the sum of RspTime and EqD, as shown in Figure 4.37. As a rule of thumb, 10 km of reach corresponds to 100 μs of round-trip delay, so if the PON is configured for its maximum logical reach of 60 km, the OLT will have launched two more downstream frames $N+1$, $N+2$, before it begins to receive upstream frame *N*.

In each allocation, the OLT specifies a start time, which is an offset from the defined start of the upstream frame. The ONU sends the first byte of the burst content at the start time instant (meaning that the burst preamble PSBu *precedes* the start instant—see Figs. 4.17, 4.18, 4.27). The OLT's bandwidth allocation algorithm is responsible to leave sufficient guard time between bursts to avoid collisions. This means the OLT must know the length of the burst header, the length of the payload allocations, and the number of FEC bytes that will be added to the burst. There are enough variables that the OLT could conceivably use an approximation, rather than exact values, as long as it rounds the errors in the conservative direction.

The OLT expects to receive the upstream burst body—the start of the GTC overhead—at an offset of StartTime from Teqd. Figure 4.37 illustrates how differing StartTimes from different ONUs arrive as expected at the OLT. The ranging and equalization delay process removes the effect of the relative delays at the OLT, so the OLT's bandwidth allocation algorithm can do its work in the domain of a single time reference, that is, $t_{\mathrm{start}N} + \mathrm{Teqd}$.

From time to time, or continuously, the OLT measures the actual arrival instant of an upstream burst from ONU *i* against the expected instant and tests the deviation against a threshold, which is recommended to be eight bit times at the G.987 XG-PON upstream bit rate, four bit times in G.984 G-PON. If the difference in delay exceeds the threshold, the OLT sends a ranging_time PLOAM message to ONU *i* with a new value for EqD_i. Exceeding the drift threshold is called *drift of window*

UNCORRECTED PROOF

(DoW) in both G.984 G-PON and G.987 XG-PON. It calls for a correction but is not an exceptional event.

Both G-PON technologies also specify a second drift threshold, twice as large as the DoW value, violation of which suggests a serious problem. Either the delay is changing more rapidly than expected or the ONU is not responding properly to equalization delay correction commands. The OLT recognizes a defect known as transmission interference warning (TIW).

According to G.984.3, the OLT should deactivate the offending G-PON ONU upon TIW. This response is somewhat inappropriate. Deactivation merely resets the ONU, causing it to vie for rediscovery and reregistration. If the propagation delay is in fact changing rapidly for some reason, resetting the ONU does not correct the situation; likewise if the ONU is unable to correctly respond to equalization delay adjustments. G.987.3 specifies deactivation as one option but also suggests disabling the ONU (emergency stop state O7), which may be more appropriate for an uncontrollable ONU.

4.3.4 Timing Relationships During ONU Discovery

We have seen how timing works during an ONU's normal operation, with equalization delay that compensates for the differences between the distances and response times of the various ONUs on the PON. But when an ONU first announces itself on the PON, its distance and delay are unknown and must be discovered. Discovered, moreover, without disrupting existing traffic on the PON.

From time to time, the OLT opens a quiet window for ONU discovery, an interval during which it suspends upstream transmission from all known ONUs. As we shall see, this window is several frames long—250–450 μs—so the OLT needs to plan it in advance.

The OLT transmits a serial number grant, which invites any ONUs that are not already registered on the PON to send a serial number PLOAM message. The OLT times the serial number grant such that all possible responses will occur during the quiet window, given the full range of physical distances, ONU response times, random delay, plus the (G.984 G-PON only) preassigned delay. If it sees a serial number response from an ONU, the OLT assigns it a TC-layer ONU-ID in a downstream PLOAM message, which advances the ONU's state (Section 4.2) so that it does not respond to further serial number grants.

It can also happen that the OLT does not receive the ONU's response. In that case, the ONU remains in serial number state and responds again when it detects the next serial number grant.

The usual reason for the OLT's failure to see an ONU is contention among several unregistered ONUs. This may take the form of a collision in time, in which case neither contender succeeds in registering, or it may simply be that the OLT recognizes the first burst received and does not have processing depth to recognize a second ONU during the same quiet window.

The second case would eventually resolve itself, as the OLT recognized ONUs one by one in order of appearance. ONUs mitigate the first possibility by adding a random delay to each serial number response. Although burst from two or more ONUs may

UNCORRECTED PROOF

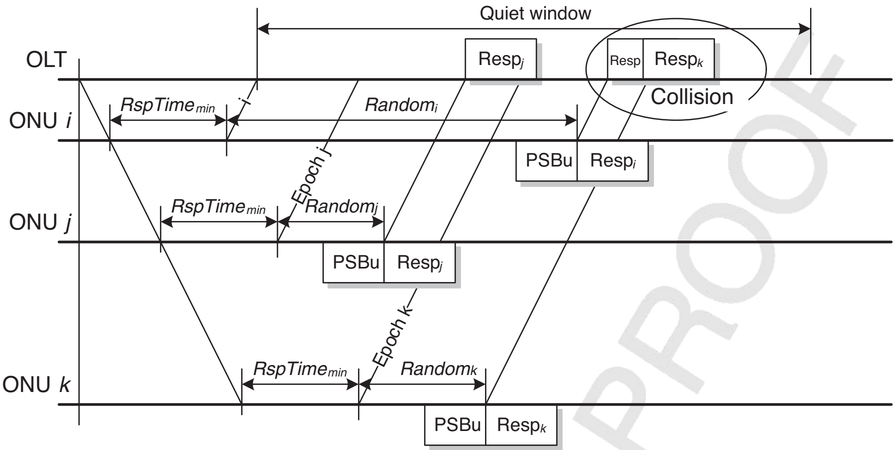

Figure 4.38 Failure to register.

collide during any given quiet window, sooner or later their random delays will relocate their responses so that they can be distinguished. The range of random delay is 0–48 µs.

Figure 4.38 illustrates the possibilities. The epoch lines show the earliest possible responses from the various ONUs. The actual responses are offset by random delays. During the illustrated discovery cycle, ONU *j* succeeds in registering, while, due to unfortunate choices of random delay, burst responses *i* and *k* collide.

Now, how does the OLT determine the quiet window? The quiet window must span the time between the earliest conceivable ONU response and the latest possible ONU response. See Figure 4.39.

The earliest possible response is determined by:

- The minimum round-trip delay, determined by the nearest ONU expected on the PON. L_{min} is a network design parameter; if it is not provisioned, it is implicitly taken as 0.
- The minimum ONU RspTime, 34 µs.

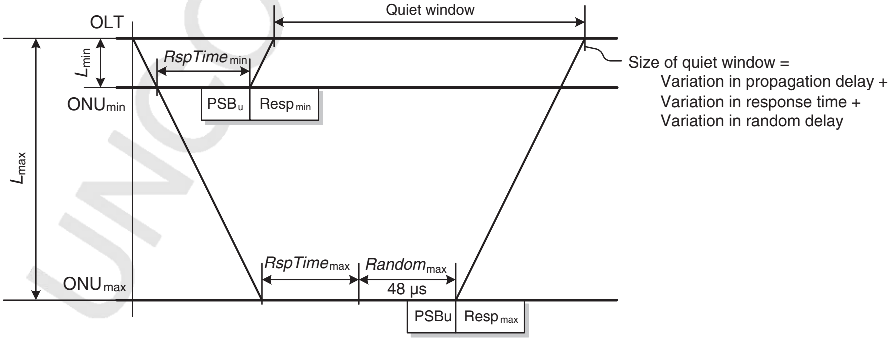

Figure 4.39 Size of quiet window.

UNCORRECTED PROOF

- The minimum random delay generated by candidate ONUs, namely 0.
- The preassigned delay, G.984 only (not shown in the figure).

The latest possible response is determined by corresponding factors:

- The maximum round-trip delay, determined by the maximum reach of the PON, L_{max}, which is likewise a network design parameter. The G-PON standards allow the maximum reach to be as much as 60 km.
- The maximum ONU RspTime, 36 μs.
- The maximum random delay generated by candidate ONUs, 48 μs.
- The preassigned delay, G.984 only.

The default assumption about an ODN is that it spans 0–20 km. As mentioned in Chapter 2, some operators require reach and differential reach of 40 km. In real deployments either maximum reach or differential reach, or both, are potentially provisionable, to optimize the size and placement of quiet windows and thereby gain a bit of efficiency in use of the PON's upstream capacity.

For 20-km differential reach, the variation in round-trip propagation delay is 200 μs. For 40-km differential reach, the variation in round-trip propagation delay is 400 μs. When the differential reach is 20 km, the suggested duration of the serial number quiet window is, therefore, 250 μs, and for 40-km differential reach the suggested duration is 450 μs.

Observe also that the OLT must extend the quiet window on the leading edge to include the PSBu of the earliest possible response burst, and it must extend the trailing edge to include the actual body of the burst containing the latest possible response. Both of these are small factors but are not to be ignored.

The OLT can position the ranging window according to its own convenience. The quiet window typically spans several 125-μs frames, so the OLT must coordinate several bandwidth maps to create the window. G.984 G-PON has two ways to control the position of ONU responses:

- The preassigned delay, broadcast to all prospective ONUs. The ONU adds the preassigned delay to its own delays before generating a serial number response. In G.987, there is no preassigned delay, so this term is 0.
- The serial number grant startTime.

Expanded differential reach is not free. ONUs at the near end of a wide differential reach must be able to store more BWmaps because they incur greater internal delay before the start of the upstream frame.

As to extending the maximum reach, one of the difficulties is the responsiveness of the DBA algorithm to varying traffic rates. The objectives for DBA response are on the order of 2 ms, while 60-km reach would consume 600 μs of that budget just in propagation delay. In Section 6.3, we explore other artifacts of increased differential reach in the DBA algorithm.

UNCORRECTED PROOF

4.4 ONU REGISTRATION

In traditional copper pair telecommunications, there is a direct relationship between a given pair of wires and the terminal at the subscriber's premises. The twisted pair may be cross-connected from one cable to another along its route, maybe more than once, which makes it important to keep accurate records of cable and pair information on each segment. In principle, the same relationship is true in point-to-point optical fiber connections.

In North America, access loop inventory is often maintained in the venerable loop facility assignment and control system (LFACS), a Telcordia product. In LFACS, hence by tradition across the industry, the first cable, the feeder cable from the central office, contains so-called F1 pairs (or fibers), which are cross-connected to F2 pairs (fibers) at a cross-box of some kind in the field, and so on. Tracking outside plant inventory and connectivity is a nontrivial operations expense, but at least in principle, the operator knows from the database exactly where a one-to-one subscriber termination appears at the central office, as well as the exact details of the connection all the way to the subscriber.

In a PON, there is no directly observable physical relationship between a given subscriber terminal and the signal on the PON as viewed from the OLT side of the splitter. That is, at the physical level, the signal at splitter port 1 is indistinguishable from the signal at splitter port 2 or splitter port 32.

It is clearly necessary to identify each individual ONU: A phone number, for example, must be provisioned to the right subscriber, and Internet and IPTV subscriptions must be matched to the subscriber who will be paying the bill. In the world of G-PON, ONU identification is usually done through the ONU's composite serial number,[*] which the OLT learns during the discovery process.

The (composite) serial number gives us a unique identifier for each ONU on the PON, but we still do not know which ONU is installed at which subscriber's premises. There are several ways to make this association, which is referred to as registering the ONU.

If we can, we want to preprovision service on the ONU, that is, set up all of the necessary information on the OLT and further up in the network, before we actually install the ONU. We want the OLT to bring up service as soon as the ONU is installed, so the installer can verify that all services are working properly before leaving the site. Additional truck rolls are expensive, to be avoided if at all possible.

If we could guarantee that only a single ONU was to be installed on a given PON at any given time, it would be easy: We're only expecting one ONU, so the ONU that appears must be the one we're expecting. This is a standard premise for theatrical comedy, but it is hardly feasible when installing ONUs.

UNCORRECTED PROOF

[*] The serial number itself is not necessarily unique until combined with the vendor ID. The industry—and this book—frequently uses the term serial number to designate the combination of vendor ID and the serial number assigned by the ONU vendor.

A second alternative: All of the provisioning information can be preestablished in the OLT, except for the ONU's serial number. When field craft install the new ONU or replace a defective ONU with a new one, they phone the network operations center (NOC) with the address of the subscriber (more likely, the installation or repair work order number) and read off the ONU's serial number from its sticker. The NOC operator then associates the serial number with the particular subscriber and provisions it into the OLT. Until the NOC operator does this, service on the ONU does not come up, and craft in the field must wait, if they are to verify it. This alternative violates our wish to avoid real-time coordination between field craft and NOC operators. It is not impossible, but it is surely the last alternative rather than the first.

As described, this is unsuitable. But there are meaningful use cases that work in essentially the same way, from the viewpoints of OLT and ONU. The historically designated *method B* registration process represents the case in which a new ONU is simply discovered by the OLT, without having been preprovisioned. The ONU's serial number is reported in a message to the management system, whereupon the ONU and OLT merely await further provisioning. The ONU sits harmlessly on the PON; the OLT refrains from provisioning it and grants it no upstream capacity except perhaps an occasional query to see if it is still there.

Even with preprovisioning, this variant of method B could occur if there were an error of some kind, and the ONU were, for example, installed on the wrong PON.

A similar use case is the possibility that a newly built subdivision might be preequipped with ONUs, few of which were actually in service. The ONU's serial number and the corresponding street address might be recorded in a database, awaiting a service order. When the service order was subsequently placed, the ONU could be provisioned accordingly. It is also possible that we might choose not to record the correlation of orphan ONU serial number to address in our database. In that case, a variation of registration method C (below) could be invoked at service order time.

Any G-PON system would naturally support method B. We would surely not want an orphan ONU to sit out there, undiscovered and unreported to the management system!

Making Method B Work

It would be possible for the installer to have a wireless terminal, much as we find at rental car check-ins, but simpler and smaller, because it would not need paper storage or printers. The installer could enter the particulars of an installed ONU on a keyboard or preferably with a bar code or radio frequency identification (RFID) reader, along with subscriber or work order information. The information could then flow through the operator's back office software and back down into the OLT and ONU, turning up preprovisioned service without the need for human coordination at the NOC. All of this except perhaps the RFID reader could be built into an app on a cell phone. If this were put in place, method B could become the preferred installation procedure.

As far as we know, no one is doing this. Yet.

UNCORRECTED PROOF

We avoid the real-time coordination problem in a second form of registration in which we preassociate the ONU's serial number with the given service order and thereby with a given subscriber. From the management system, we preprovision the serial number into the OLT along with the rest of the service parameters. When the physical ONU appears on the PON, the OLT can immediately activate the subscriber's services. The installer verifies service, closes the order, and goes on to the next job. This is known as *method A* registration for historical reasons.

Method A is used when it makes sense. It requires no special support from either the ONU or the OLT. If the OLT does not recognize the ONU's serial number, we have a method B situation.

The complication with method A is that it requires coordination of physical ONUs between service order, warehouse inventory, and the correct truck on the correct day. And what if the preselected ONU turns out to be damaged or defective?

Logistic considerations make this alternative undesirable for single-family ONUs, where we just want to select the next of many identical ONUs from the warehouse, or from the truck. But it may be quite appropriate for the installation of an MDU or a reach extender, devices that are regarded as network equipment and installed under engineering work orders rather than subscriber installation or repair orders. For these, it may well be practical to assign a particular equipment unit to a particular site in advance.

ONU *registration method C* is often called the registration ID method. Registration method C assumes that the ONU has some kind of local interface that lets an installer—or the subscriber, for that matter—enter an identifier string, designated the registration ID. Not infrequently, the local interface takes the form of a simple web page available to whomever connects a PC to one of the ONU's Ethernet ports, but it can be as simple as a craft butt set connected to a POTS port.

In method C, the ONU's serial number is never provisioned into the OLT. Instead, the OLT discovers the serial number. Either during—recall the G.987 XG-PON registration PLOAM message in Section 4.2.1—or after ONU discovery, the OLT also retrieves the registration ID from the ONU. The OLT then associates the serial number with a particular preprovisioned management-layer ONU-ID (subscriber) through the registration ID.

In G.987 XG-PON, the registration ID is a string of up to 36 bytes, conveyed from ONU to OLT in the registration PLOAM message. In G.984 G-PON, the registration ID is conveyed in a confusingly named 10-byte variable called *password* that is conveyed in the password PLOAM message. The name came from the early days of B-PON, when it was expected that some sort of identifier would be useful, but the registration concepts had not yet been fully worked out. The G.984 registration ID is usually a character string, but it could be anything that can fit into 10 bytes, for example, a 20-nibble hex string or a cryptographic hash of some other string such as a passphrase.

In the ONU-G managed entity (Chapter 5), OMCI has been adapted to allow for 36-character credentials (actually 24 plus 12) for operators who cannot fit the

UNCORRECTED PROOF

registration ID into the 10 bytes of G.984. The same managed entity includes an attribute through which the OLT can signal the status of the ONU's credentials back to the subscriber or installer. This feedback facilitates a second try if things go wrong, for example, because of typing errors.

Registration method C enables the OLT to turn up preprovisioned services immediately. No personnel coordination is needed, and no logistics are involved in getting the right physical ONU to the right subscriber. For good reason, this is the most popular method used in one form or another by most operators.

The registration ID is at least potentially accessible to the subscriber, so the long-term security of this technique must be addressed. One of the ways in which the registration ID process can be secured is the so-called lock mechanism. When the ONU appears initially on the PON, the OLT learns its serial number. The OLT associates the serial number with the given subscriber through the registration ID. The OLT then locks the serial number association, either immediately, or after a predetermined time that allows for the possibility of discovering and replacing a defective ONU or upon command from the network operations center. The OLT subsequently recognizes the ONU only by its serial number and ignores its registration ID.[*] If the ONU later fails and needs to be replaced, the network operations center must unlock the association—reenabling registration ID recognition—before the replacement ONU can be brought into service.

It has not become a formal registration method in the standards, but it would also be possible for a subscriber to do the complete service order process directly from an autodiscovered ONU, with no hands-on involvement by the operator at all. This would require the network to set up a promiscuous IP association for any auto-discovered ONU, an association that would only accept HTTP(S) (hypertext transfer protocol secure), and would be redirected to the operator's own service order process. The ONU used by the subscriber to order the service would be implicitly registered by its serial number in the process—a variant of type B—and provisioning would flow in real time to the necessary network elements, so that service could be instantly available. No preassigned registration ID would be needed.

This is done with DSL service; it can be done with PON.

4.5 ONU ENERGY CONSERVATION

After the discussion of the problems of power in Chapter 2—and we hardly mentioned heat dissipation—we trust it is clear that power is a major concern of the industry. Existing power converters may be 85–90% efficient; if it is possible to gain another 1 or 2%, it is well worth doing. Inexpensive batteries that could tolerate temperatures down to −40°C, with indefinite lifetimes, with more stored energy and

[*] But registration ID also plays a rôle in security. See the further discussion in Section 4.6.

UNCORRECTED PROOF

less energy loss during full charge and trickle charge—these are the stuff of which dreams are made.

Reducing the resistance of copper wires would be nice, but it will not happen. What does happen, has happened for years, and shows no signs of stopping is Moore's law reduction of semiconductor size, along with a reduction of the power needed to perform, say, one floating-point multiplication. The progress in power reduction is more impressive than is apparent at first glance because each new generation of semiconductors multiplies the functionality of its predecessor.

But good, solid engineering, the kind we have been doing for years, looking for excess milliwatts and killing them, is not glamorous. We do not do press releases about that 1% gain in power converter efficiency, installing the ONU closer to the power converter, or going to 48 V instead of 12. And it is important to be seen to be doing something about energy consumption.

If we could save 1 W per ONU, and if we had a billion subscribers, we could avoid building two 500-MW power plants! That is pretty exciting. It even has the merit of being true, given the assumptions, which include the questionable ideas that our whole billion subscribers reside within one or two time zones, and that power can be reduced during peak demand hours.

The maximum power reduction at an ONU, of course, occurs when it is switched off, for example, when the subscriber goes on holiday or possibly even overnight. But intrusion monitoring, smart meter reading, the intelligent home, machine-to-machine communications—these services require access even when the homeowner is away for a month. The upside is that the power-down option places the choice on the subscriber, who knows best what his needs are.

Unwilling to rely on subscribers making their own choices—there would be no press releases, and the great unwashed might actually make the wrong choice!—the first standards proposals for saving ONU power contemplated shutting down the entire ONU overnight, without benefit of subscriber consent. Arguably, the ONU could be awakened at any time by subscriber action, but there are times when an emergency call needs to be directed *toward* the ONU, not just *from* the ONU. A sound asleep ONU could not receive any sort of wakeup signal from the OLT. Long periods of unilateral unavailability guaranteed customer dissatisfaction, not to mention liability lawsuits.

The discussion then turned to the idea of brief intervals of sleep, after each of which the ONU and OLT would check with each other to see if anything interesting was going on. That model is formalized in G.987.3, XG-PON. The durations have never been tied down, but a sleep cycle on the order of 10–100 ms is the likely range.

Two modes are defined: sleep (also known as cyclic sleep) and doze. In sleep mode, the ONU's PON interface is entirely powered down; in doze mode, the downstream direction remains operational. Through OMCI, the OLT learns some interval values that are built into the ONU, sets some interval values of its own choice, and then grants permission for the ONU to enter one or either of these low-power modes at its own discretion.

At run time, the OLT and ONU exchange PLOAM messages that keep them in approximate synchronization with regard to low-power states. The ONU may

UNCORRECTED PROOF

exercise either doze or sleep modes at will, assuming that the OLT has authorized both, but it must coordinate with the OLT if it wishes to change mode from doze to sleep or vice versa. We discuss the state machines below.

4.5.1 Sleep Mode

While the ONU is sleeping, its optical transceiver is powered down. Traffic arriving in either direction is discarded by default, or possibly—if there is not very much—buffered for later delivery. A timer in the ONU periodically triggers the transceiver to power up, whereupon it regains synchronization to the PON and exchanges handshakes with the OLT to confirm that it is still alive and well. If necessary, the OLT can bring the ONU into full active mode (active held state—see Fig. 4.40) during one of these handshake episodes.

It is important to understand that the handshakes take the form of bandwidth grants and responses, with the ONU granted the opportunity to send at least one upstream PLOAM message, but *without* a mandatory PLOAM message exchange.[*] This relieves the message processing load, especially from the OLT, which would otherwise have to process several exchanges per second from perhaps a 100 or more ONUs. Only if the ONU does not respond to a grant during the expected interval does a timer in the OLT bring its processor into the picture or, of course, if the ONU signals that it wishes to wake up.

To expedite the process of bringing the ONU into active mode, the OLT can set the FWI bit in the first allocation structure of every burst authorization directed to that ONU—refer back to Figure 4.13. The ONU will see this bit in the first frame it decodes after recovering downstream synchronization, in contrast to the Sleep_allow (off) PLOAM message, which serves the same purpose but that the OLT may send less often because it may involve processor overhead.

Management communications between OLT and ONU are generally expected to be reliable. But when the ONU is asleep, messages may fail to be delivered, including, for example, broadcast PLOAMs that might redefine the burst profile or request a new encryption key from all ONUs. The OLT is responsible to deal with such possibilities, possibly by forcing the ONU awake before beginning a management transaction.

The ONU may also respond to a local stimulus such as a subscriber off-hook event—called a local wakeup indication (LWI)—to awaken itself spontaneously. Whether awakened by a local indication or a forced wakeup from the OLT, the ONU remains awake until released by the OLT to resume low-power operation.

The criteria for what constitutes an LWI are not specified. Examples include the aforementioned off-hook event at an analog telephone interface, upstream messages such as IGMP join requests, heartbeat or alarm messages generated by applications such as premises surveillance, or timed events such as 802.1ag CCMs initiated by the

UNCORRECTED PROOF

[*] A no-op heartbeat acknowledge PLOAM message from the ONU can presumably be discarded before it reaches the processor.

ONU itself. Guided by market requirements, it is the vendor's choice how much of the ONU's functionality to retain during PON interface sleep.

4.5.2 Doze Mode

While the ONU is dozing, its optical transmitter is powered down, but its receiver remains active, and it is expected to forward downstream traffic. Doze mode admittedly saves less power than full sleep mode; on the other hand, doze mode can be used during the many long hours of TV watching—almost entirely downstream traffic, with only occasional upstream IGMP messages—while sleep mode cannot.

As with sleep mode, the dozing ONU periodically powers up its transmitter and shakes hands with the OLT. As with sleep mode, local wakeup indications can bring the ONU into full active mode spontaneously. Unlike sleep mode, the OLT can force the ONU awake at any time, albeit with a start-up delay while the ONU powers up its transmitter.

The question of management transactions remains but is less constrained in doze mode. The ONU does receive downstream PLOAM and OMCI messages; it just cannot respond instantly. But the ONU is allowed 750 μs to respond to a PLOAM message, and 1 s[*] for OMCI. If the ONU regards these messages as remote wakeup indicators, it might be prepared to respond to PLOAM messages within the PLOAM window and can certainly respond to OMCI messages within the OMCI window. However, the standard does not specify that the ONU treat these events as remote wakeup indicators, so the OLT may be well advised to bring even a dozing ONU into full wakefulness before beginning a management transaction.

Doze mode is mandatory in G.987 ONUs; sleep mode is optional.

4.5.3 Tools

Two PLOAM messages are defined for power saving coordination, one in each direction.

- The OLT issues the Sleep_allow PLOAM message either to a single ONU or as a broadcast message to all ONUs. The message contains only a single parameter: Sleep_allow (on) or (off). As would be predicted, the value *off* brings the ONU into full active held state. The value *on* represents permission to the ONU to enter one of its enabled low-power modes—or not—based on its own best judgment. The ONU ignores the message if the designated low-power modes have not been enabled through OMCI.
- The ONU issues the Sleep_request message, which also has only one parameter, chosen from the set {sleep, doze, awake}. The values *sleep* and *doze* indicate which mode the ONU intends to enter, while the value *awake* tells the OLT that

UNCORRECTED PROOF

[*] G.984.4 allows as much as 3 s for low-priority OMCI responses, but low priority is undefined. In G.988, all OMCI responses are expected within 1 s.

the ONU is now fully active and will remain in active held state until permitted by the OLT to reenter one of its low-power modes.

If the ONU supports both sleep and doze modes, the ONU vendor defines two times: for sleep mode, the expected time required for the entire transceiver to power up and regain synchronization to the downstream PON; for doze mode, the time required for only the transmitter to power up and be ready to respond to bandwidth grants. This read-only information is available to the OLT via OMCI to assist it in setting up a satisfactory timing contract with the given ONU.

The OLT determines three intervals that it sends to the ONU via the ONU dynamic power management control managed entity in OMCI.

I_{sleep} Measured as a count of 125-μs frames, this interval is the maximum duration that the ONU is permitted to remain offline during a low-power state. That is, before the interval I_{sleep} elapses, the ONU is expected to have powered itself up, regained synchronization to the downstream PON if necessary, and be fully prepared to respond to grants. The ONU runs a timer T_{sleep} to guarantee this performance. Observe that, if the initial value of T_{sleep} were the same as I_{sleep}, the ONU would have to come fully awake in zero time. It is the ONU's responsibility to appropriately offset the initial value of T_{sleep}, depending on how long it needs to recover. If the OLT's timing requirements were too onerous, the ONU would be expected to remain fully active.

I_{aware} In the absence of external stimuli, the ONU returns to awareness every I_{sleep} frames and remains aware for at least I_{aware} frames. During this time, it responds to bandwidth grants and may exchange PLOAM messages or user traffic with the OLT. Once the interval I_{aware} elapses, in the absence of external stimuli, the ONU returns to the low power state. The ONU runs a timer T_{aware}, whose purpose is to guarantee that the constraints of I_{aware} are satisfied.

I_{hold} When the ONU enters active held state, either because of a local or remote wakeup event, it remains there for a minimum duration, I_{hold} frames. The purpose of I_{hold} is to avoid the race that might occur if the OLT were to send a downstream Sleep_allow (on) PLOAM message that crossed the ONU's upstream Sleep_request (awake) message. Not surprisingly, we find that the ONU runs an analogous timer T_{hold}.

The OLT maintains its own timers, whose values are derived by the OLT itself:

T_{alerted} When the OLT attempts to rouse a sleeping or dozing ONU, it either sets the FWI bit or sends Sleep_allow (off), or both. The OLT must allow time for the ONU to receive this indication, including delay in sending the PLOAM message, round-trip time, ONU recovery time (this is why the ONU declares its power-up times), and delays in granting upstream transmission opportunities. If the OLT does not receive a response from the ONU before T_{alerted} expires, the

UNCORRECTED PROOF

OLT begins counting loss of response events toward the declaration of an ONU loss of bursts (LOB) defect.

C_{lob} This count defines the number of consecutive bursts whose absence triggers a loss of bursts declaration by the OLT. G.987.3 specifies that C_{lob} is 4. The relationship between grant count and elapsed time is not specified; the OLT would be expected to exercise good judgment.

T_{er} The expected response timer T_{er} is similar in a way to T_{alerted}, inasmuch as it defines the latest instant at which a response is expected from the ONU. It differs from T_{alerted} in that it runs when the OLT believes the ONU is in one of its unresponsive low-power states. After I_{sleep} elapses, and subject to delays in granting upstream transmit opportunities, the OLT expects to receive a response. Each time it receives a response, the OLT resets timer T_{er}; if T_{er} times out, the OLT attempts to force the ONU awake to confirm that it is still present and healthy.

4.5.4 State Machines

The low-power state machines in ONU and OLT only exist when the ONU is in its operation state O5 (Section 4.2). The OLT has no direct visibility of ONU state but has strong indications when it is safe to believe that the ONU is in state O5.

Either the OLT or the ONU may be incapable of supporting sleep mode (doze mode is mandatory in G.987), and either OLT or ONU may be provisioned to avoid modes that they are actually capable of supporting. In these cases, the states for unsupported modes simply do not exist; the associated messages are never transmitted and are silently discarded if received. In the discussion below, we assume that both OLT and ONU support both modes.

During its aware states, and when it awakens spontaneously, the ONU responds to bandwidth grants from the OLT. These grants occur at intervals wholly under the control of the OLT. The pertinent interval I_{aware} is also wholly controlled by the OLT. It is, therefore, the OLT's responsibility to issue grants with sufficient frequency to permit the ONU to respond in a timely manner. If the OLT were to permit I_{aware} to elapse without having granted the ONU an upstream transmission opportunity, it could hardly blame the ONU for the timeout!

Transitions between full-power and low-power state groups are signaled between ONU and OLT with PLOAM messages. However, upstream PLOAM messages rely on the PLOAMu grant bit in an upstream bandwidth allocation, which is only available at the OLT's discretion. Where the ONU state machine shows the emission of a Sleep_request (SR) message on a transition, the associated timers do not start running until the PLOAM message is actually transmitted.

Figures 4.40 and 4.41 portray the state machines.

UNCORRECTED PROOF

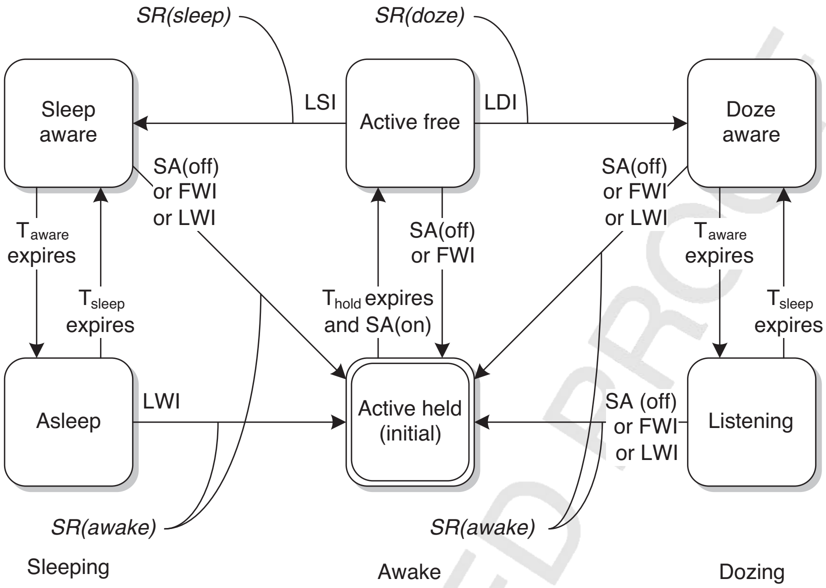

Figure 4.40 ONU power-saving states.

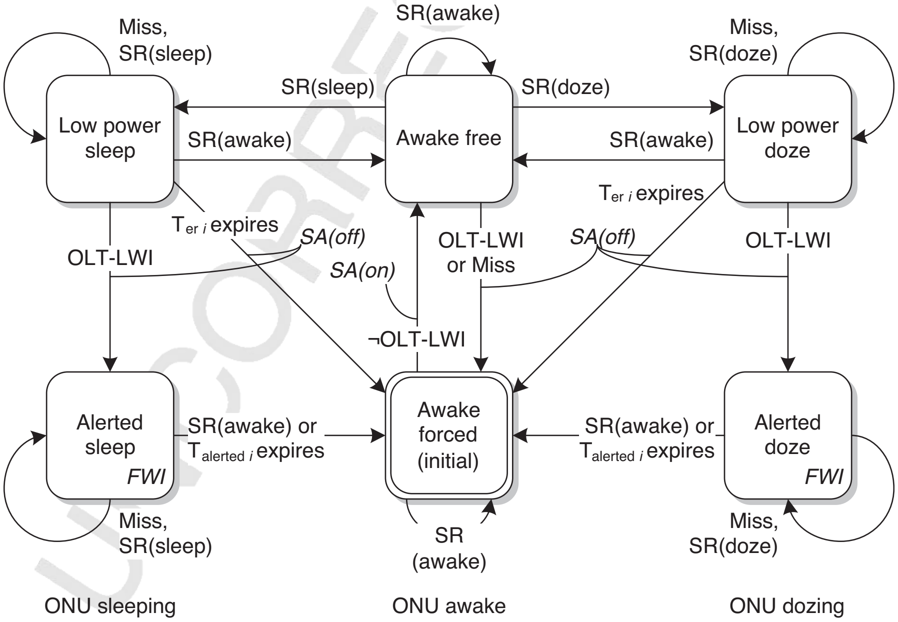

Figure 4.41 OLT power-saving states for ONU *i*.

UNCORRECTED PROOF

4.5.4.1 ONU State Machine

Figure 4.40 illustrates the ONU power-saving state machine. At the time of power-saving provisioning by the OLT, if not before, the ONU creates the state machine and initializes it into active held state. The OLT provisions the necessary modes and intervals into the ONU via OMCI, after which it issues a Sleep_allow (on) PLOAM message. When timer T_{hold} expires, the ONU enters active free state, in which it has permission to enter a low-power state according to its provisioning, its capabilities, and its assessment of the local traffic environment.

Notation

Q2
- Italic text designates the transmission of a PLOAM message. (No-op acknowledge messages are not shown.)
- SR means Sleep_request; SA means Sleep_allow. These are the PLOAM messages whose exchange coordinates ONU and OLT state machines. The OLT sends only the SA message; the ONU sends only SR.
- LWI is the local wakeup indication.
- LDI and LSI are local doze and sleep indications, respectively. Criteria for declaring LDI and LSI are left to the ONU vendor, just as with LWI.
- FWI is the presence of the forced wakeup bit in the first allocation structure of the BWmap.

Depending on circumstances, active traffic, for example, the ONU, may remain in active free state for long periods of time. Suppose that the ONU has remained active free and, for its own reasons, has not entered a low-power state. Now suppose that an event at the OLT causes it to want to keep the ONU active. It can force the ONU back into active held state by sending a Sleep_allow (off) PLOAM message, by setting the FWI bit in a BWmap, or both.

Left to its own devices, the ONU evaluates its environment and sooner or later determines that it is appropriate either to sleep (LSI) or doze (LDI). The criteria for making this decision are not specified but would include at least the level of upstream traffic and the need to send management messages such as OMCI responses.

Suppose that our ONU has decided to doze. The ONU informs the OLT about its decision by sending a Sleep_request (doze) PLOAM message upstream. The ONU enters doze aware state, in which it continues to respond to upstream grants while counting down its local timer T_{aware}, a timer based on I_{aware} as set by the OLT. The aware state keeps the ONU responsive long enough to be sure that the OLT was not in the process of forcing it awake just as the ONU decided to doze.

Assuming that there was no race with the OLT, timer T_{aware} expires after an interval at least as large as I_{aware}. At this point, the ONU enters listening state, stops responding to upstream grants, and powers down its transmitter. The listening ONU continues to process and deliver downstream traffic.

Because the ONU's receiver is still alive, the ONU can respond not only to LWI but also to requests from the OLT, be they delivered via the FWI bit or the

UNCORRECTED PROOF

Sleep_allow (off) PLOAM message, or both. These events cause the ONU to power up its transmitter; after it is capable of responding normally, the ONU signals sleep_request (awake) and returns to active held state.

But if nothing else is going on—the most likely case for 3 AM—timer T_{sleep} expires. After an interval not greater than I_{sleep}, and after having powered up its transmitter, the ONU returns to doze aware state. When it enters doze aware state, the ONU must be fully capable of responding to bandwidth grants.

In doze aware state, the ONU responds to grants from the OLT, primarily as a keep-alive mechanism. It is not forbidden to send real traffic upstream during this time, for example, a brief heartbeat message from our premises surveillance application, as long as the grant specifies the correct alloc-ID for the traffic. It is also possible that the ONU will regard even a small amount of real traffic as an LWI, and will no longer be in doze aware state. It is also possible that the OLT will grant bandwidth only to the default alloc-ID. Observe that there is no PLOAM exchange beyond a heartbeat acknowledge, and therefore minimal or no CPU involvement at the OLT, in the frequent transitions between doze aware and listening states.

Sleep mode operation is much the same, except that downstream messages cannot be delivered while the ONU is asleep.

4.5.4.2 OLT State Machine

In Figure 4.41, we see the matching state machine, an instance of which is maintained by the OLT for each ONU *i* that has a low-power mode enabled. The state names differ slightly from those of the ONU state machine, avoiding the possibility of ambiguous interpretation.

Although the states do not map one for one—beware!—the OLT state machine is synchronized to that of the ONU, subject to delays in propagation, message generation, and bandwidth grants.

The notation is the same as in Figure 4.40, with the following additions:

- *Miss* indicates that the OLT issued a bandwidth grant and received no response.
- *FWI* in the alerted states indicates that the OLT sets the forced wakeup indicator bit in the first allocation structure of each BWmap directed to ONU *i*.
- *OLT-LWI* designates the cessation of the local wakeup indicator at the OLT.

The OLT instantiates the state machine for ONU *i* in awake forced state. The OLT is free to hold the machine in awake forced state indefinitely. Of course, if it wished the ONU never to sleep or doze, the OLT would simply not create the state machine at all. At the ONU, the corresponding initial state would be active held, which would prevent the ONU from ever going into a low-power mode.

Having created its local state machine, having provisioned ONU *i* with suitable interval values, having enabled one or both low-power modes via OMCI, and having determined that traffic levels are low, the OLT emits a Sleep_allow (on) message to permit the ONU to exercise at least one of the low-power modes at its own discretion. This puts the OLT into awake free state.

UNCORRECTED PROOF

If a local wakeup indicator LWI gives the OLT subsequent reason to want the ONU to remain awake, it can send Sleep_allow (off) or FWI and return to awake forced state. One reason for this action could be the OLT's failure to receive a response to an upstream grant (a miss event) without having first received a Sleep_request PLOAM message. This is a safeguard against the possibility that the Sleep_request message from the ONU was lost in transit.

Suppose that the OLT is now in awake free state for ONU i, and that it receives a Sleep_request (doze) PLOAM message in response to one of its bandwidth grants (necessarily a grant to the default alloc-ID with the PLOAMu bit set). Having no objection, the OLT enters low-power doze state and starts timer $T_{\mathrm{er}\,i}$. The OLT does *not* acknowledge the SR message from ONU i.

The OLT continues to send bandwidth grants periodically. Whenever it receives a response from ONU i, it restarts timer $T_{\mathrm{er}\,i}$.

Having given notice that it intends to doze, the ONU stops responding to upstream grants after an interval not less than I_{aware}, whereupon OLT timer $T_{\mathrm{er}\,i}$ begins to count down. The OLT continues to send bandwidth grants, in case the ONU awakens spontaneously, but when it receives no response (miss), it just remains in the same state, as long as timer $T_{\mathrm{er}\,i}$ is still running.

Under normal circumstances, the ONU responds again to upstream grants after an interval not longer than I_{sleep}, T_{er} resets again without expiring, and the cycle repeats indefinitely. Observe that no explicit signaling or state changes occur at the OLT during this repetitive cycle. This is an important feature to offload unnecessary processing from the OLT state machine, an instance of which exists for each of perhaps a 100 ONUs on the PON, and perhaps 8 or 16 PONs on an OLT blade.

In comparing state machines, observe also that a dozing ONU alternates between listening state and doze aware state, during both of which the OLT's state machine remains in a single state, namely low-power doze. The OLT and OLT state machines are synchronized but do not match state for state.

If ONU i dies in its sleep, $T_{\mathrm{er}\,i}$ expires. The OLT sends Sleep_allow (off), probably sets the FWI bit, and enters awake forced state, where it starts deciding whether to declare loss of bursts (LOB) against ONU i.

The OLT may have some less dramatic reason—traffic or management, for example, shown as LWI—to want the ONU to return to full wakefulness. It again sends Sleep_allow (off), probably sets the FWI bit in the first allocation structure of each bandwidth grant to ONU i, and enters alerted doze state, where it waits for the ONU to power up its transmitter and respond. In comparing ONU and OLT state machines, observe that the ONU has no state equivalent to the OLT's alerted states.

When it sees a response from ONU l—the proper response being a Sleep_request (awake) PLOAM message—the OLT enters awake forced state. It does the same if $T_{\mathrm{alerted}\,i}$ expires, but in this case, it is considering whether to declare loss of bursts.

The state machine for sleep mode is exactly the same but subject to slightly different timing and possibly a different policy to force wakefulness.

G.987.3 defines the state machines in more detail, specifically covering the corner cases. There is no equivalent power-saving model for G.984 G-PON.

UNCORRECTED PROOF

4.6 SECURITY

We encourage a skeptical attitude toward security in general and toward G-PON security in particular. There are at least two perspectives for the skeptical eye:

- Is the threat model complete? What did we miss? Of the threats we listed, are they real?
- If the threat were actually to materialize, how effective would the security barriers be?

A web exploration of the ideas of Peter Gutmann,[*] of the University of Auckland, New Zealand, is refreshing and informative.

The security features of G.987 XG-PON go considerably beyond those of G.984 G-PON. In G.984 G-PON, downstream unicast transmissions can be encrypted. Full stop. By default,[†] unicast keys are generated by the ONU and sent upstream to the OLT in the clear, in the expectation that the PON is physically secure in that direction. Control and management traffic is subject to CRC error checks but not to validation.

G.987 XG-PON, in contrast, allows for encryption of unicast traffic in both directions and encryption of downstream multicast traffic. G.987 defines two mechanisms for strong mutual authentication of ONU and OLT—OMCI and IEEE 802.1X—where G.984 defines only the OMCI method. G.987 key exchange is protected by encryption, and PLOAM and OMCI traffic is cryptographically protected from forgery. There is no error correction capability in this layer.

As we have done before, we start with G.987 XG-PON, following which G.984 G-PON is readily to be understood.

4.6.1 Security in G.987 XG-PON

4.6.1.1 Threat Model

G.987.3 lists four threats:

Theft of Service Anyone capable of receiving the downstream PON signal can, in principle, intercept all downstream traffic. This threat is realistic. While we, as subscribers, may have nothing to hide, we would certainly object to someone eavesdropping on our services, even in only the downstream direction. As to multicast, it does matter! The ethically challenged may consider it a venial sin to steal video streams, while on the other hand, operators expect to derive a substantial portion of their G-PON-related revenue from IPTV. Theft of multicast service is discouraged by middleware encryption, but it might be

UNCORRECTED PROOF

[*] At the time of this writing, Gutmann's home page was at http://www.cs.auckland.ac.nz/∼pgut001/.

[†] G.984 G-PON includes an option for OMCI strong authentication. If this option is exercised, the ONU encrypts upstream keys with the resulting master session key.

possible for a knowledgeable group of users to purchase one legitimate subscription, snoop the keys from a legitimate—but hacked—set-top box, and distribute the keys to other members of the group.

ONU Impersonation This threat is a refinement of the theft of service model. Since the serial number of a single-family ONU is readily available—marked right there on the outside of the box—someone capable of counterfeiting that serial number could impersonate a legitimate ONU, and the OLT would happily deliver that subscriber's complete service package to the imposter. If a forged serial number appears on a PON that differs from its home PON, the OLT would simply fail to recognize the serial number (see the autodiscovery discussion in Section 4.4), and no harm would be done. It is not specified what should happen if two ONUs appear on the same PON with the same serial number, but surely the OLT ought to take notice. Of course, if it happens that the legitimate ONU has simply been stolen and relocated by the neighbors while the owner is away, there is little the OLT can do about it.

OLT Impersonation The third threat listed in G.987.3 is that an attacker could gain access to the optical network. ONU impersonation and theft of service are yet once again cited as risks. The only new risk in this category is that an attacker could impersonate the entire OLT. This threat definition came from those who presumably have intelligence connections into the underworld, but one has to ask what the legitimate OLT is doing while its functions are being usurped by an imposter, and why the operator is not out there responding to critical alarms. Be that as it may, this is the justification for *mutual* authentication of ONU and OLT. The ONU is expected not to speak to strangers.

Replay Attacks The final threat listed is that an attacker could record packets and replay them later, either intact or modified, to cause something bad to happen. The discussion typically refers to PLOAM messages as vulnerable, but no one has ever brought forward a detailed example of specific messages or message sequences whose compromise would cause a problem. Replay attacks with subscriber traffic are potentially serious, so we intend to encrypt it. Given the security of the keys, it is fair to say that subscriber traffic is safe from replay.

- The agreed security mechanism does not protect against replay attacks of PLOAM or OMCI messages.
- OMCI can carry information such as session initiation protocol (SIP) (RFC 3261) and remote authentication dial in user service (RADIUS) (RFC 2865) registration, information that admittedly ought to be protected from interception. The OMCI channel can be encrypted.

There is no question that downstream encryption is necessary. It has been part of ITU-T PON since the early days and, in particular, is a feature of G.984 G-PON. A case can be made for multicast encryption, which at least restricts the group of

UNCORRECTED PROOF

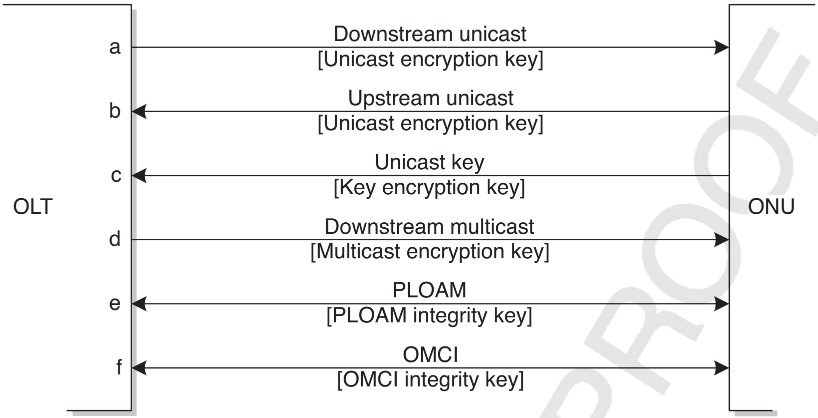

Figure 4.42 G.987 XG-PON security overview.

service thieves to the scope of a single PON. It is harder to be convinced of the dangers of counterfeit OLTs and intercepted PLOAM traffic.

Is excessive security a problem? Only inasmuch as it increases complexity and brittleness and delays implementation and deployment. There is also the risk of complacency if the security turns out not to be as robust as might have been desired.

4.6.1.2 What We Want to Achieve

G.987 XG-PON security appears in any number of contexts and can be confusing. Figure 4.42 is an overview of which information is encrypted and with which key.

(a) Downstream unicast traffic may be encrypted with the unicast encryption key. The key is generated by the ONU. This is the only form of cryptographic security in G.984 G-PON.

(b) Upstream unicast traffic may also be encrypted, also with the unicast key.

(c) When a G.987 XG-PON unicast key is generated or updated, it is conveyed to the OLT under the protection of the key encryption key (KEK). There is also an option for this in G.984 G-PON.

(d) Downstream G.987 XG-PON multicast traffic may be encrypted. The multicast encryption key is generated by the OLT and conveyed to all concerned ONUs via OMCI, protected by the KEK.

(e) G.987 XG-PON PLOAM messages in both directions are sent in cleartext form, but they are protected against alteration by a MIC. The MIC employs the PLOAM integrity key (PLOAM_IK). G.987.3 specifies that the receiver discard a PLOAM message whose MIC fails.

UNCORRECTED PROOF

> The editor of G.987.3 points out that the safety rules on construction sites specify that heavy equipment operators accept emergency stop commands from anyone under all circumstances, even from passers-by on the street. At the time of writing, the same policy was under consideration for PLOAM messages such as deactivate_ONU-ID.

(f) G.987 XG-PON OMCI messages in both directions are sent in cleartext form, but they are protected against alteration by a message integrity check. The MIC employs the OMCI integrity key (OMCI_IK). G.987.3 specifies that the receiver discard an OMCI message whose MIC fails.

In the following sections, we investigate how all of this works. Initialization and update are topics of special interest, and we offer parenthetic observations on the actual degree of security achieved.

4.6.1.3 Security Toolkit

We need to digress at this point to fill in some of the fundamentals and notation. All XG-PON security is based on the advanced encryption standard AES, defined in National Institute of Science and Technology (NIST) Federal Information Processing Standard (FIPS) 197. The underlying AES encryption system can be used in several modes, three of which are applicable to XG-PON. G.987 XG-PON specifies 128-bit keys, so in the recommendations, we often see the algorithms designated explicitly with the trailing term—*128*. We also find the same 128-bit qualifier as an explicit argument in some of the equations.

CMAC A cipher-based message authentication code (NIST special publication 800-38B) does not conceal the original message but instead produces a digest of the message that is sent along with the message itself. The digest is easy for the receiver to regenerate if the message has not been altered and if the CMAC is generated with the same key at each end. The algorithm is designed such that it is computationally difficult to generate a valid digest if either of these conditions is not satisfied, that is, if the message has been altered or if the receiver does not know the key.

We express an AES-CMAC derivation in the following notation:

$$\text{Result} = \text{AES-}CMAC(\text{key, operand, } 128) \tag{4.5}$$

That is, the 128-bit result is derived by applying the AES-CMAC algorithm to a key and an operand.

ECB An electronic code book (NIST special publication 800-38A) is a way of transforming cleartext input to ciphertext through the use of a key. The characteristic of ECB mode is that the same input, with the same key, always produces the same output. Large samples of ECB output could, therefore, be subject to dictionary attack, especially if the corresponding cleartext could

UNCORRECTED PROOF

reasonably be deduced. Key encryption is a good application for ECB mode because the output sample space is small and also because it would not be instantly obvious whether the output of an attack algorithm was or was not the cleartext.

CTR In contrast, AES counter mode continually changes a seed value, such that the output of a given input with a given key is never[*] the same twice, at least never within whatever time scale is regarded to be significant. Successive counter initialization values must be known or predictable by both sides, which is why it is a counter, a simple function that changes continually and is readily predictable by the engines at both ends. In G.987 XG-PON (also in G.984 G-PON), counter mode is used for payload encryption, with material derived from the frame counter and the cleartext's position within the frame or upstream burst as the initial counter value.

4.6.1.4 System Keys

It is apparent from Figure 4.42 that G.987 XG-PON relies on an abundance of keys. The keys used for XG-PON message exchanges are based on two additional keys. Their purpose is to assist with initialization and to make each OLT-ONU association unique, so that there is very low probability of two associations generating the same keying material.

- The session key (SK) is used to generate the key encryption key (KEK), PLOAM_IK and OMCI_IK.
- A master session key (MSK) helps generate the session key.

At every stage of initialization and operation, we need keys that are derived with the same algorithms at OLT and ONU, from raw material that is preferably supplied jointly by the OLT and the ONU but is, of course, known to both. Starting with initialization, let us trace the derivation of the various keying material needed for XG-PON. Refer to Figure 4.43.

Notation

- The vertical bar | designates bit or byte concatenation.
- A hex pair followed by a subscript designates a repeated byte value. For example, $0xFF_{16}$ indicates 16 repeated bytes of all-one values.

The sequence of causality is as follows: Before the ONU can attempt to register on the PON, it must learn the burst profile. The OLT periodically broadcasts the profile,

[*] At the time of writing, a vulnerability in the XG-PON application had been identified in which the counter value is duplicated in two successive 125-μs frames once every 4000 years. The proposed solution is to initialize the counter to zero at OLT boot-up and defer the issue for further study for the subsequent 2000 years.

UNCORRECTED PROOF

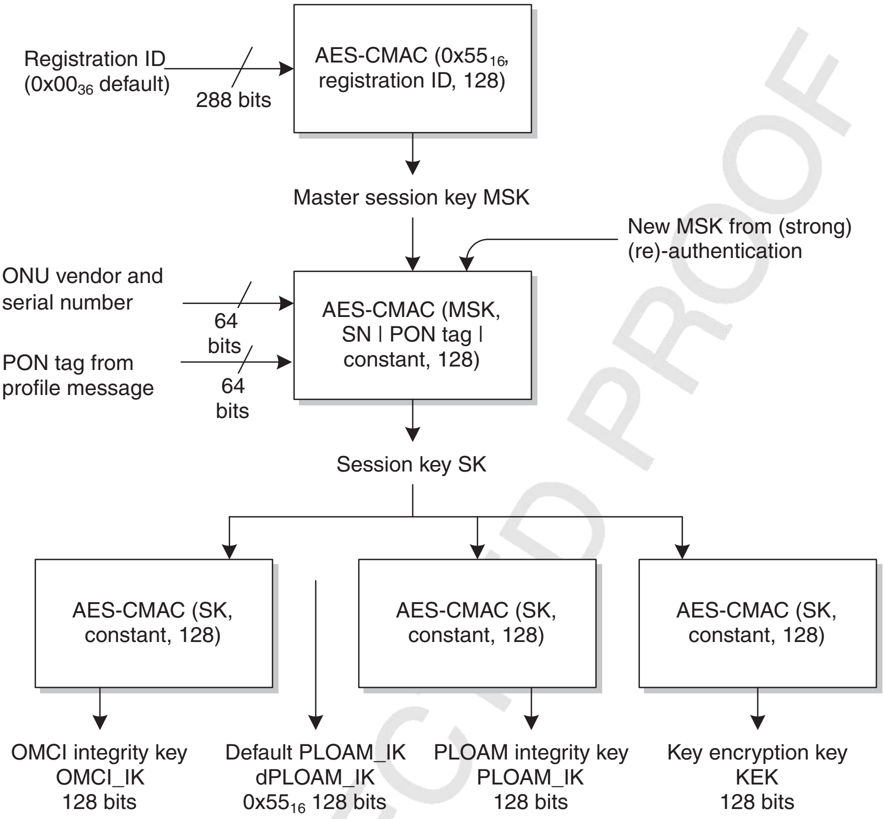
Registration ID
(0x00$_{36}$ default)
288 bits
AES-CMAC (0x55$_{16}$, registration ID, 128)
Master session key MSK
New MSK from (strong) (re)-authentication
ONU vendor and serial number
64 bits
PON tag from profile message
64 bits
AES-CMAC (MSK, SN I PON tag I constant, 128)
Session key SK
AES-CMAC (SK, constant, 128)
AES-CMAC (SK, constant, 128)
AES-CMAC (SK, constant, 128)
OMCI integrity key
OMCI_IK
128 bits
Default PLOAM_IK
dPLOAM_IK
0x55$_{16}$ 128 bits
PLOAM integrity key
PLOAM_IK
128 bits
Key encryption key
KEK
128 bits

Figure 4.43 Key derivation sequence.

a PLOAM message that, like all PLOAM messages, includes a message integrity check. G.987.3 specifies that the ONU discard PLOAM messages whose MIC is not valid, but MIC validation requires a PLOAM integrity key PLOAM_IK. The PLOAM integrity key for the profile message[*] is the efault key 0x55$_{16}$. This default PLOAM_IK,[†] which we designate dPLOAM_IK, permits the ONU to perform error checking on the profile message.

The profile PLOAM message contains an 8-byte PON tag field. It is expected that PON tag remain constant across all profiles that may be broadcast by the OLT, and that it remain constant over extended periods of time. Now that the ONU recognizes the profile, the OLT and ONU share common knowledge of the PON tag.

Anyone snooping the downstream PON also knows the PON tag.

[*] All broadcast PLOAM messages use the default PLOAM_IK, 0x55$_{16}$ and thus have no security whatever.

[†] Be aware that dPLOAM_IK is *not* the same as a PLOAM_IK computed on the basis of the default registration ID.

UNCORRECTED PROOF

When the ONU responds to a serial number grant, it supplies its vendor ID and serial number in an upstream serial_number_ONUPLOAM message. Since we have no other key, the PLOAM message MIC is again based on dPLOAM_IK = $0x55_{16}$. The OLT and ONU now share common knowledge of the ONU's vendor ID and serial number.

Anyone capable of snooping the upstream PON or reading the sticker on the ONU enclosure also knows the ONU's vendor ID and serial number.

The next step is for the OLT to assign a TC-layer ONU-ID, using the dPLOAM_IK key to generate the PLOAM MIC. In the state progression of Section 4.2.1, the ONU is now in ranging state, where it responds to a ranging grant by sending its registration ID, again with MIC based on dPLOAM_IK.

The OLT and ONU now share knowledge of the registration ID and can derive the master session key (MSK), the first step in the sequence of Figure 4.43.

Anyone snooping the upstream PON also knows the registration ID and can generate the same MSK.

Based on common knowledge, information known to both ONU and OLT, with part supplied by each, the OLT and ONU agree when they separately compute the SK and the integrity keys for OMCI and PLOAM, as well as the KEK. We are in business.

Anyone snooping the PON is also in business.

At any time, the OLT may reauthenticate, either by requesting the registration ID (weak)—always computing MIC with dPLOAM_IK—or through the OMCI or 802.1X authentication mechanisms (strong). Any strong reauthentication event causes recomputation of all of the keys at both ONU and OLT.

If a PON snooper can derive the system keys, the payload keys are equally vulnerable. It appears that we must rely on strong authentication.... If strong authentication fails, the recovery mechanism is not well specified.

Corner Case The initialization model of Section 4.2.1 allows the OLT to usher the ONU directly through ranging state O4 and into operation state O5 by simply sending a ranging_time PLOAM message, and without retrieving the registration ID.

In theory, this is feasible. By factory default, the registration ID is a sequence of null octets, $0x00_{36}$. If the operator does not use the registration ID, the keys computed from the default are valid. The other possibility is that the ONU has already registered on some previous occasion, and its registration ID is already known.

However, if the OLT fails to guess the correct registration ID, the ranging_time PLOAM message fails because it is required to be signed with a MIC that is derived from a real PLOAM_IK, not the default. If the ONU discards the ranging_time PLOAM message because of a MIC mismatch, as specified by G.987.3, the OLT has no choice but to send a ranging grant. The OLT only knows whether the equalization delay was successfully set by granting the ONU an upstream transmission opportunity, so this particular corner case could affect service.

Rather than elaborate software to deal with this corner case—and with negligible benefit—a real-world OLT can be expected to always retrieve the ONU's registration ID.

UNCORRECTED PROOF

4.6.1.5 Strong Authentication

We have seen that the simple default approach introduces complexity but hardly security. The alternative is strong encryption, which may be supported in either (or both!) of two ways, one based on OMCI, the other built on IEEE 802.1X. Both methods presuppose some kind of shared secret, or in the latter case, possibly a certificate that can be verified. Even the registration ID is, in some sense, a shared secret. For that matter, even the ONU's MAC address could be considered a shared secret under certain circumstances—almost anything that can be known to both sides, that has at least some degree of shielding from casual observation, and whose complexity minimizes the odds of a lucky guess.

Whatever it is, the shared secret must be known to both OLT and ONU. And thereby hangs a tale.

Authentication Options Even within a single operator's network, authentication methods may vary. For example, an MDU that is regarded as telecommunications equipment may need nothing more than serial number identification, while ONUs that are CPE, easy to move and easy to compromise, may be subject to 802.1X or OMCI strong authentication. The OLT, therefore, needs to know what level of authentication is necessary with which ONU. This policy can be one of the parameters preprovisioned into the OLT as part of the equipment or installation order, presumably looked up in an EMS table that correlates the operator's policy options with the available and approved ONU types.

The problem is a bit more difficult at the ONU. For many good reasons, vendors want to deliver the same ONUs, and the same software, to all customers worldwide. This may mean that the ONU has to support all of the authentication options, including no authentication. So be it. In theory, an ONU practicing mutual authentication should refuse to talk to an illegitimate OLT—but how would it know?

Strong authentication presupposes not only that both sides have the same shared secret information but that both sides know which particular information is the shared secret. If one side uses the MAC address as its shared secret and the other side uses a passphrase, they are not likely to succeed in their negotiation. Nothing in the standards allows the OLT and ONU to discover each other's opinions about the nature of the shared secret.

We hypothesize some out-of-band mechanism to set the value of the shared secret on the ONU, for example, a web page served by the ONU to the subscriber or installer. This mechanism also needs to be able to define which information is to be regarded as the shared secret, and which authentication algorithms the ONU should accept. Sounds a lot like a simple registration ID, does it not!

Keying Material Updates, Proper Treatment of MIC Errors Clearly, both OLT and ONU must have the necessary raw material for key computation. When one of the input values changes, both recompute their keys. But a change in raw material originates either at the ONU or at the OLT, and the other does not know about the change for a certain amount of time. Each side must be prepared to continue to accept

UNCORRECTED PROOF

messages with the old key for a certain amount of time. For management transactions, G.987.3 specifies 10 ms for regeneration of the keys of Figure 4.43. Alternatively, the OLT may simply refrain from issuing management transactions for 10 ms.

The standards do not specify it, but the suggested 10-ms interval would naturally begin when the information change was communicated by one side to the other.

- If the registration ID changes, the timer would begin when the ONU sent an upstream registration PLOAM message with the new value. This might not happen until the next time the ONU reinitialized, which could be many years.
- The PON tag is intended to remain stable indefinitely, but if it were to change, the timer would begin when the OLT transmitted an updated profile PLOAM message.
- In OMCI strong authentication, the timer would begin when the ONU transmitted an attribute value change (AVC) message that confirmed the creation of a new MSK.
- In 802.1X authentication, the timer would begin when the OLT forwarded an extended authentication protocol (EAP) success message (RFC 3748) to the ONU.

The alternative, to recompute keys unilaterally, exposes the OLT-ONU association to difficult corner cases, which we need not explore.

By the way, if the ONU homes onto two OLTs for protection, it is assumed that the OLTs have a way to apprise each other of key updates.

The algorithm for key generation is not specified, except for the admonition to use a cryptographically adequate source of entropy in generating random numbers. We may remark that this is a difficult requirement to satisfy, also difficult to test.

It is also not specified how often to change keys. The strength of the AES algorithm with 128-bit keys is such that it probably suffices to change keys once per month. On the other hand, if the key derivation or the key storage is not secure, no update rate is good enough to ensure security. However, a key update every few seconds or minutes might discourage an attacker.

Consider a use case. If the neighborhood IPTV theft of service club is capable of hacking the middleware key out of a set-top box, its technology expert can also presumably find the broadcast key by delving into a G-PON ONU. But if coordinating a key update with the other members of the club—whose usefulness is limited to the 100 ONUs on the same PON—causes seconds or minutes of disruption to the smooth flow of stolen video, it may simply not be worth the trouble. Instead, the thief with the legitimate subscription will just retransmit the decoded video back up the PON to his friends and neighbors, possibly concealed within his own privately arranged encryption stream.

UNCORRECTED PROOF

MIC Errors There is always the possibility that, somehow, OLT and ONU keys get out of sync. It could expedite recovery from keying errors to define a code point in the acknowledge PLOAM message (or possibly a new message) to flag a PLOAM message MIC error. When it received an acknowledge PLOAM message with this code point, the OLT could reauthenticate. The same could be done at the OMCI level, with the notification delivered either via the OMCI response or via a PLOAM message. At the time of writing, no such mechanisms had been defined. In fact, G.987.3 is clear that a MIC error should cause the silent discard of the associated message.

4.6.1.6 Payload Encryption Key Update

Recall from Figure 4.2 that the XGEM frame header contains a 2-bit key index field. The value 00 specifies no encryption, the value 11 is reserved, and the values 01 and 10 select between two possible keys. Whichever code point is in use at a given instant is the current key; the other code point may be either the old (former) key or the new (next) key.

Orthogonal to the two keys of the key index, there are two key domains (we like to call them key rings), the unicast key ring for each ONU, and the broadcast key ring, which is shared by as many ONUs as need multicast encryption. The ONU may have as many as two key rings, one of each type, while the OLT would have a unicast key ring for each ONU that needed encryption, with one more for multicast (broadcast). Each key ring holds up to two keys, one of which is selected dynamically by the key index field of the XGEM frame header. The other key on the key ring is offline so that it can be updated without concern for payload disruption. As the OLT and ONU successfully negotiate each new key, we expect to progress from the current key to the new key by switching key index.

Each GEM port is provisioned through OMCI to use either the broadcast or the unicast key ring (or no key, of course). If the unicast key ring is specified, OMCI also offers the option to provision encryption only downstream or both ways.

Multicast (broadcast) key update is straightforward. If key index 01 is in use, the OLT simply continues to use index 01 while it generates a new key with index 10, distributes it to all ONUs via OMCI, and collects their acknowledgments. At this point, the OLT is free to start using key index 10. Index 01 is no longer in use, so it is available whenever the OLT again wishes to update the key.

If the OLT wished to invalidate the old key, it would be possible do so through an additional OMCI transaction with each ONU, using a state or a code point that would need to be added into the information model. But the OLT could also continue to use the old key as well as the new one, as long as nothing had changed. G.987.3 is silent on this point with regard to the broadcast key.

OMCI was easy because it is an acknowledged channel, and key switching only occurs when the OLT is satisfied that the new key has been distributed correctly to all concerned ONUs. The same simple acknowledged update could have been done with unicast keys since the PLOAM channel also supports acknowledged message exchanges.

The difference is that, for continuity with G.984 G-PON, it was agreed that the ONU generate the key and send it upstream to the OLT. The acknowledgment thus

UNCORRECTED PROOF

needed to be from the OLT to the ONU, rather than the other way around. Imagine the following PLOAM semantics:

ONU → OLT: New key
OLT → ONU: Acknowledge

However, the OLT is responsible for initiating key updates. This implies a third message:

OLT → ONU: Generate a new key
ONU → OLT: New key
OLT → ONU: Acknowledge

Finally, there were concerns that if the OLT and ONU somehow got out of sync, it might be possible to switch one side to the new key while the other side reverted to an obsolete key that had the same key index. This could have been avoided with three keys on the key ring but was resolved instead by state machines on OLT and ONU, and the message sequence illustrated in Figure 4.44. G.987.3 specifies the state machines precisely and fills in additional detail such as corner case handling.

Let us walk through the sequence of Figure 4.44.

Suppose that encryption has not previously been enabled. Then the OLT's state machine exists (or not) in a state called key inactive, not shown in

CORRECTED PROOF

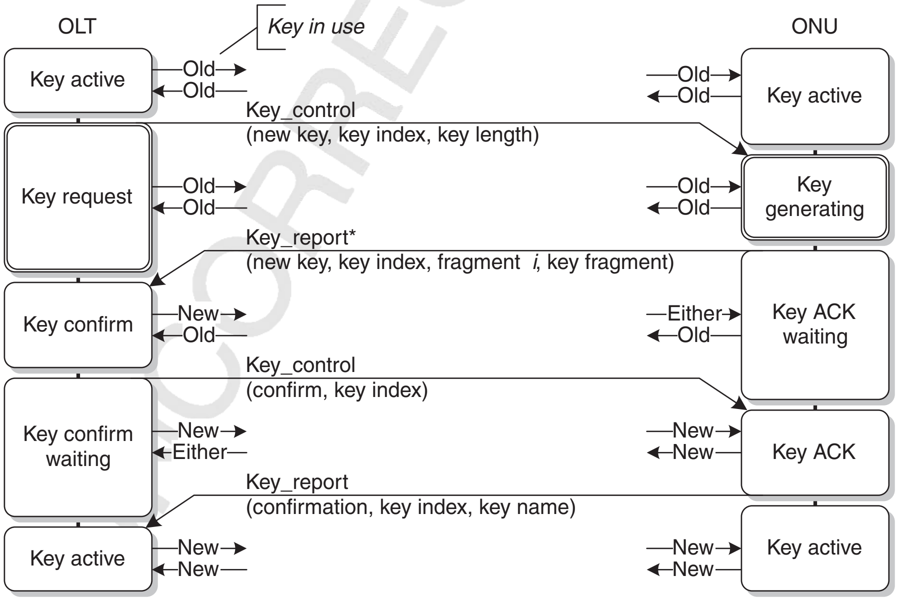

Figure 4.44 Unicast key update.

Figure 4.44. If encryption is then enabled, the state machine is created in (or moves into) the key request state. At the ONU, the state machine is either created in key generating state when the first key_control PLOAM message is received or moves into that state from a key inactive state that is also not shown.

During normal continuing operation, the OLT state machine is in key active state. Suppose that, at the moment, this translates to using key 01 with the given ONU. Suppose that a timer expires, triggering a key update. During the update, the OLT continues to use key 01.

The OLT sends a key_control PLOAM message to the ONU, requesting a new key with index 10, and specifying the length of the desired key. The PLOAM message structure allows for keys of up to 256 bytes, although 16 bytes, 128 bits, suffice for the current need.

As we see in Figure 4.44, the ONU proceeds to generate a new key, which it transmits upstream in one or more key_report PLOAM messages, 32 bytes at a time. The fragments are numbered as a confirmation to the OLT that everything is in order. A single message suffices for the current needs of XG-PON.

Confident that both sides know the new key 10, the OLT starts using it downstream. The ONU accepts traffic with either key during its key ACK waiting transition state, but continues to transmit with the old key 01.

The OLT now sends a key_control PLOAM that requests confirmation of the key, really just a state machine handshake. This informs the ONU that the OLT knows the new key 10. The ONU invalidates the old key 01 and starts transmitting as well as receiving with the new key 10.

Now it is the OLT's turn to accept traffic encrypted with either key.

The ONU generates a 128-bit CMAC of the new key 10, the so-called key name, and returns it in a key_report PLOAM. Given the unicast key *encryptionKey*, the key name is computed as

$$\text{Key Name} = \text{AES-CMAsC}(\text{KEK},\ \text{encryption Key}\,|\,\text{constant},\ 128) \tag{4.6}$$

The constant is 0x3331 3431 3539 3236 3533 3538 3937 3933, which just happens to match the ASCII string "3141592653589793".

When it receives the ONU's key_report, the OLT invalidates the old key in the upstream direction. Both sides are now back in key active state, using the new key 10 in both directions, neither OLT nor ONU having disrupted traffic or imposed potentially onerous real-time synchronization constraints upon the other.

4.6.1.7 Derivation and Use of System Keys

A few details remain to be filled in. Looking back at Figure 4.43, we see that the 128-bit session key SK is generated by an AES-CMAC function, using the MSK as the key, while the message to be validated comprises the bitwise concatenation of the

UNCORRECTED PROOF

ONU vendor ID and serial number, the PON tag and a 128-bit constant whose value is 0x5365 7373 696F 6E4B. (Not coincidentally, this happens to be the ASCII string “SessionK”.)

$$\text{SK} = \text{AES-CMAC}(\text{MSK},\ \text{Vendor ID} \mid \text{SN} \mid \text{PON tag} \mid \text{constant},\ 128) \tag{4.7}$$

The session key SK then becomes the key in further AES-CMAC derivations, each with a different constant string. The equation is the same for each:

$$\text{Key} = \text{AES-CMAC}\ (\text{SK},\ \text{consant},\ 128) \tag{4.8}$$

where the constant strings differ for each key. All of the strings are chosen to be 128 bits long, 16 characters.

Key	Constant (ASCII)
OMCI_IK	“OMCIIntegrityKey”
PLOAM_IK	“PLOAMIntegrtyKey”
KEK	“KeyEncryptionKey”

The MSK and SK are not used for anything else.

PLOAM-IK The PLOAM integrity key validates both upstream and downstream PLOAM messages.

Not to make it too easy, downstream PLOAM messages are prepended with a direction octet whose value is 1, while the direction octet prepended to upstream PLOAM messages has the value 2. The MIC field of the PLOAM message is only 8 bytes, so the CMAC function is only 64 bits long. Q3

$$\text{MIC}_{\text{PLOAM}} = \text{AES-CMAC}\ (\text{PLOAM\_IK},\ \text{direction} \mid \text{Message},\ 64) \tag{4.9}$$

Recall that message validation requires the OLT and ONU to know the common key. To satisfy this requirement for broadcast downstream PLOAM messages, and during ONU discovery, the key used in these circumstances is the default, $\text{dPLOAM\_IK} = 0\text{x}55_{16}$. Appendix II describes the PLOAM messages in detail, including the choice of PLOAM encryption key for each message.

OMCI-IK The OMCI integrity key likewise validates both upstream and downstream OMCI messages. It is used in the same way as is the PLOAM validation key, including prepending a direction octet of value 1 or 2 to the downstream or upstream OMCI message, respectively, before computing the MIC. The OMCI MIC is 4 bytes, so the CMAC function is only 32 bits long.

$$\text{MIC}_{\text{OMCI}} = \text{AES-}CMAC(\text{OMCI\_IK},\ \text{direction} \mid \text{Message}, 32) \tag{4.10}$$

UNCORRECTED PROOF

UNCORRECTED PROOF

There is no need for an OMCI default key because there is always a well-defined MSK and SK when OMCI is operable.

KEK The key encryption key is used to conceal the values of the unicast and the broadcast keys as they traverse the PON via the upstream key_report PLOAM message and downstream via OMCI, respectively. Both of the key encryption functions use AES in electronic code book (ECB) mode.

The ONU is expected to maintain state that remembers whether either OMCI or 802.1X strong authentication was performed during its current session on the PON. If not, the ONU does not use the KEK to encrypt new unicast keys upstream, but sends them in the clear. Whether the upstream key is encrypted or not is indicated by a flag in the key_report PLOAM message, so the OLT always knows how to interpret it. It is not specified by the recommendation, but it would be reasonable for the OLT to object if it expected an encrypted key and got a cleartext key.

Regardless of whether strong authentication was used, the OLT encrypts downstream broadcast keys. As we have seen, unless the OLT-ONU association is strongly authenticated, this provides security more apparent than real.

4.6.1.8 Payload Encryption

The careful reader has by now noticed that we have provision to encrypt almost everything, including the encryption keys themselves, as befits the truly paranoid, and we know how to invoke, generate, and update the keys—but as yet we have no provision to encrypt subscriber payload, whose security is the justification for all of this. Yes, we can indeed encrypt payload.

In the ONU2-G managed entity of OMCI, the ONU declares its payload encryption capabilities, and the OLT chooses one of them. At present, AES is the only defined option, with 128-bit keys.

Payload encryption occurs as a process in the step of transferring GEM frames into the XGTC payload (see Section and Fig. 4.8, reproduced here as Fig. 4.45). The

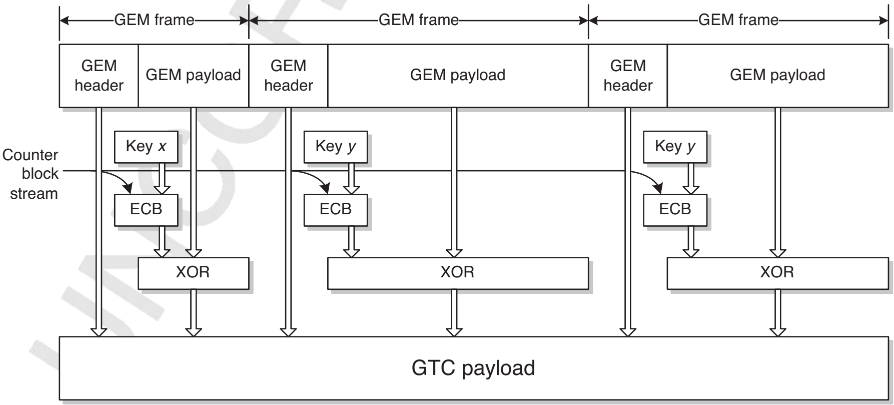

Figure 4.45 Payload encryption.

algorithm is AES in counter mode, which, not surprisingly, relies on counters. Operating at a higher layer, these counters never see the bytes in the underlying server layers, namely the downstream physical frame header, FEC parity bytes, the upstream burst header, upstream PLOAM messages, or upstream DBA reports. To put it a different way, what the counters see is the XGTC header and the GEM frame headers, neither of which is to be encrypted, and the GEM frame payload, which *is* encrypted.

Each downstream GEM port is provisioned through OMCI to be plaintext or to be encrypted with the ONU's unique unicast key or with the common broadcast key. Upstream, the GEM port may be either plaintext or encrypted with the ONU's unicast key. At run time, either unicast or broadcast key 01 or 10 may be specified. Both ONU and OLT must be agile in bringing the correct key into play at a moment's notice.

If AES counter mode is to be secure, its counter values should never repeat—never, at least within some reasonable time frame. We start by using the (super)frame counter as a seed, but we intend to encrypt 128-bit blocks of GEM payload at a time, and there are a lot of 128-bit blocks in an XGTC frame. We can not just use the frame counter for all of them—that would be a repeated value—so we combine the frame count with an intraframe counter.

In the downstream direction, the intraframe counter initializes to zero with the first bit of the XGTC frame and increments at 128-bit intervals, 16 bytes, including both GEM headers and GEM payload. The same initialization logic applies to the upstream direction, but given that the upstream burst is offset from the start of the upstream PHY frame time, counter initialization is more directly expressed by stating that, at the start time instant of the upstream XGTC frame, the counter initializes to the value $\lfloor \text{start\_time}/4 \rfloor$, that is, to the current word count.

After excluding overhead, a 10-Gb/s downstream frame of 125 μs contains 135,432 bytes for XGTC payload. It, therefore, suffices to have a downstream intraframe counter with range $\lceil 135,432/16 \rceil = 8465$, that is, taking on values from 0 to 8464. In the upstream direction, the latest possible start time is 9719 and the largest possible burst is 9720 words. The maximum value that could ever be required from the intraframe counter is, therefore, 4858 (an exercise for the reader). Each direction requires a 14-bit counter.

We precatenate the intraframe counter with the 50 least significant bits of the (super)frame counter (Fig. 4.46), and duplicate the whole thing, to derive a 128-bit value—the initial counter block—that ticks along during each XGTC frame. The important facts about this sequence are that it is unique over time spans of practical interest, and that it is readily predictable at both OLT and ONU, so that it can be easily regenerated for decoding. To avoid duplicate values between downstream and upstream, we also invert the less significant 64-bit end in the upstream direction before loading it into the initial counter block.

On the first bit of each GEM frame header, we latch the current value of the initial counter block into a 128-bit payload counter. Stable for 128 bits at a time, the output of the payload counter forms the counter input to the AES-CTR algorithm.

UNCORRECTED PROOF

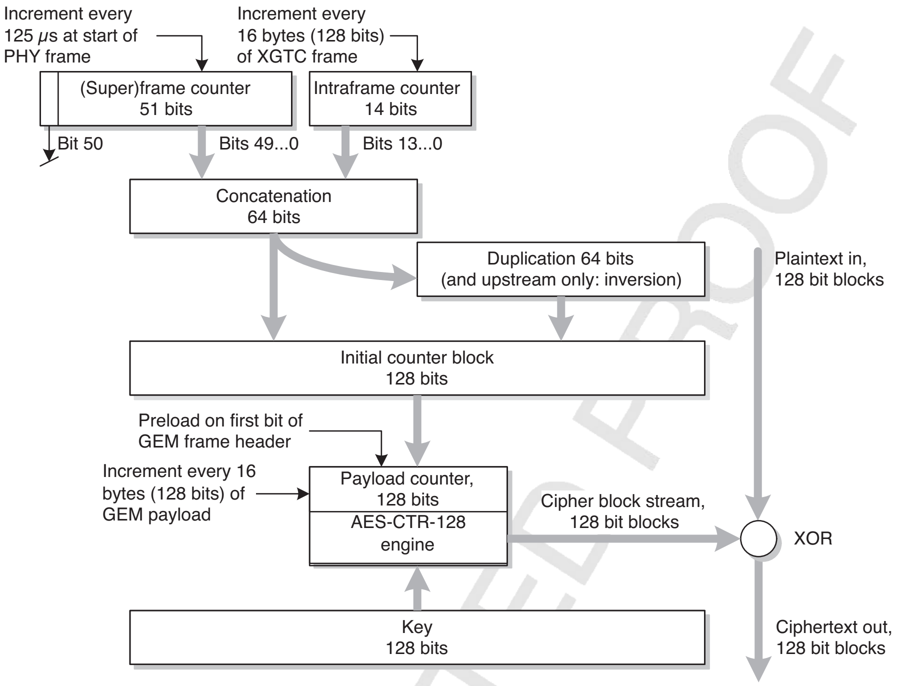

Figure 4.46 Downstream encryption.

Every 16 bytes of payload, the AES-CTR engine increments the payload counter. It is just an ordinary 128-bit counter, that is, with no duplicated 50-bit, 14-bit structure, so the payload counter's value immediately diverges from that of the initial counter block, even if they both happen to increment at the same instants.

That last phrase deserves amplification. Both the intraframe counter and the AES counter increment at 16-byte intervals, but the phase of the intraframe counter is locked to the XGTC payload, which includes GEM headers, while the phase of the AES counter is determined by the start of each successive GEM frame. Because everything in XG-PON is aligned on word boundaries, there are four possible relative phases between the intraframe counter and the AES counter.

Observe also that, while we preload the upstream AES counter at the beginning of the GEM *header*, we do not increment it until byte 16 (the 17th byte) of GEM *payload*. The additional preliminary delay allows the AES engine more time to prepare the first cipher block.

Also using the proper key, which may change with each GEM frame, each new value of the AES counter triggers the encryption engine to generate a 128-bit cipher block, which is then exclusive-ORed with 128 bits of cleartext payload to produce ciphertext. At the receiving end, the same cipher block recovers the cleartext.

The upstream direction is basically the same, except for the initialization instants of the counters. We refer the interested reader to G.987.3 for the details.

UNCORRECTED PROOF

4.6.1.9 Reduced Encryption Strength

Certain administrations—we refrain from editorial comment—do not permit strong encryption within their jurisdictions. At the time of writing, it appears that AES-128 algorithms are acceptable in some of these jurisdictions, as long as the keying material is restricted to 56 bits. The actual key remains 128 bits long, but 72 of its bits are fixed and well-known values. G.987 XG-PON specifies a leading sequence of 0x55 bytes in this application.

Since software parameters can easily be changed, the length of the key needs to be a compile time option, not a run-time setting. OMCI includes a read-only attribute in the enhanced security control managed entity, an attribute that allows an ONU to declare the effective length of its keys.

As to jurisdictions that do not even permit AES, there is currently no agreed-upon solution.

UNCORRECTED PROOF

4.6.2 Security in G.984 G-PON

If G.987 XG-PON security is arguably overkill, G.984 G-PON security might be called PON security lite—lightweight, simple, and perfectly adequate for the needs of many operators. Downstream unicast is the only payload encryption, and with one exception, keys are sent upstream from the ONU in the clear. With the same exception, authentication is a matter of trusting the ONU's serial number or registration ID, and no one expects a counterfeit OLT to suddenly appear in the network.

The exception is that G.984 supports the option for OMCI-based strong authentication. The result of this process is the generation of a master session key MSK, which is used in AES-ECB-128 mode to encrypt the unicast key transmitted upstream in the encryption_key PLOAM message. In G.987 XG-PON, this is done with the key encryption key KEK, but no KEK is defined in G.984 G-PON.

Turning to payload encryption, we first observe that encrypted GEM ports are turned on or off through the encrypted_port-ID PLOAM message, rather than through an OMCI GEM port attribute, as is done in G.987 XG-PON.

As with G.987 XG-PON, the ONU2-G managed entity of OMCI declares the ONU's payload encryption capabilities, and the OLT chooses one of them. AES is the only option presently defined, with 128-bit keys.

Figure 4.47 shows that G.984 G-PON uses a simple request–response PLOAM exchange to update keys.

Details

Because the AES algorithm is fixed at 128 bits, the key is 128 bits long, 16 bytes. The encryption_key PLOAM message carries 8 bytes of key, so there are always two fragments.

Each message is repeated three times, including the key_switching_time message and its acknowledgment.

As would be expected, the key switches at the beginning of the frame whose (super)frame count is specified in the key_switching_time PLOAM message.

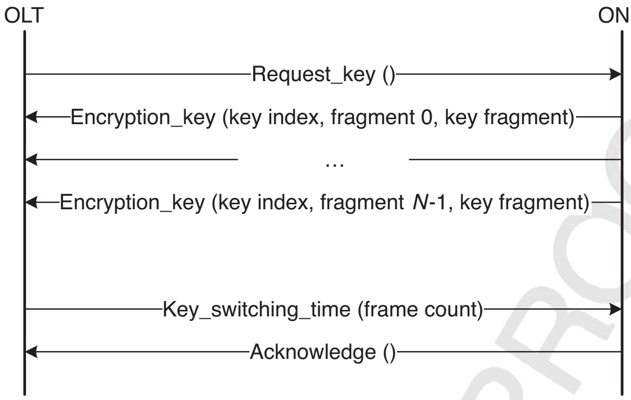
OLT
ONU
Request_key ()
Encryption_key (key index, fragment 0, key fragment)
…
Encryption_key (key index, fragment N-1, key fragment)
Key_switching_time (frame count)
Acknowledge ()

Figure 4.47 Unicast key update, G.984 G-PON.

UNCORRECTED PROOF

It is good to complete the PLOAM transaction before the specified frame number occurs.

Payload encryption is similar to the G.987 XG-PON algorithm, with a frame counter and an intraframe counter, concatenated and latched into an AES-CTR-128 engine for each encrypted GEM frame.

Details

The (super)frame counter in G.984 G-PON is 30 bits wide. The intraframe counter is 16 bits wide, of which only 14 can vary. The concatenated 46-bit pair is replicated three times to produce a field 138 bits wide. The 10 most significant bits are ignored, yielding a 128-bit so-called cryptocounter.

The G.984 G-PON intraframe counter starts with the value 0 on the first byte of the downstream PCBd, that is to say, on the first byte of the physical synchronization pattern. The counter runs continuously throughout the frame, including during FEC bytes. The counter increments every 4 bytes, so that its maximum value is 9719 at the end of the frame.

As with G.987 XG-PON, G.984 G-PON latches the value of the cryptocounter on the first bit of the GEM frame header. Subsequent increments to the AES counter occur at 128-bit intervals in the payload section of the GEM frame.

G.984 G-PON has no key index; it has an active key register and a shadow key register for new and old versions of the unicast key. The active key is the one in use at any given time, while the shadow register is updated by the PLOAM exchange of Figure 4.47. The key index field in the encryption_key PLOAM message (Appendix II) is merely a sequence number that increments with each new key, allowing the OLT to be sure that all (both) upstream encryption_key messages pertain to the same key.

When the specified (super)frame count arrives, both OLT and ONU shift the shadow key into the active key register and begin using it.

In case of confusion for example, intermittent LOS with recovery via the PopUp PLOAM message, the OLT can cancel a pending key switch by sending a new request_key PLOAM message.

Reduced Key Length in G.984 G-PON Reduced key length is not defined for G.984 G-PON. Presumably vendors who sell into these markets will implement proprietary solutions, probably along the lines of the G.987 XG-PON recommendation. If interoperability or standards compliance is someday required in these markets, reduced key length could be formalized in an amendment to G.984.3, subject to the usual risk that some implementations might have hardware implementations that could not be adapted to the standard.

UNCORRECTED PROOF

Author Query

1. We have changed (Section 6.3.2.2.6) to (Section 6.3.2.2) Because level head 4 as unnumbered.
2. Unclear to which text this refers?
3. Please check text cites for Eq. 9.
4. Which appendix?

UNCORRECTED PROOF