The Definitive Guide to Optical Transport Networks (OTN)

Tutorial White Paper

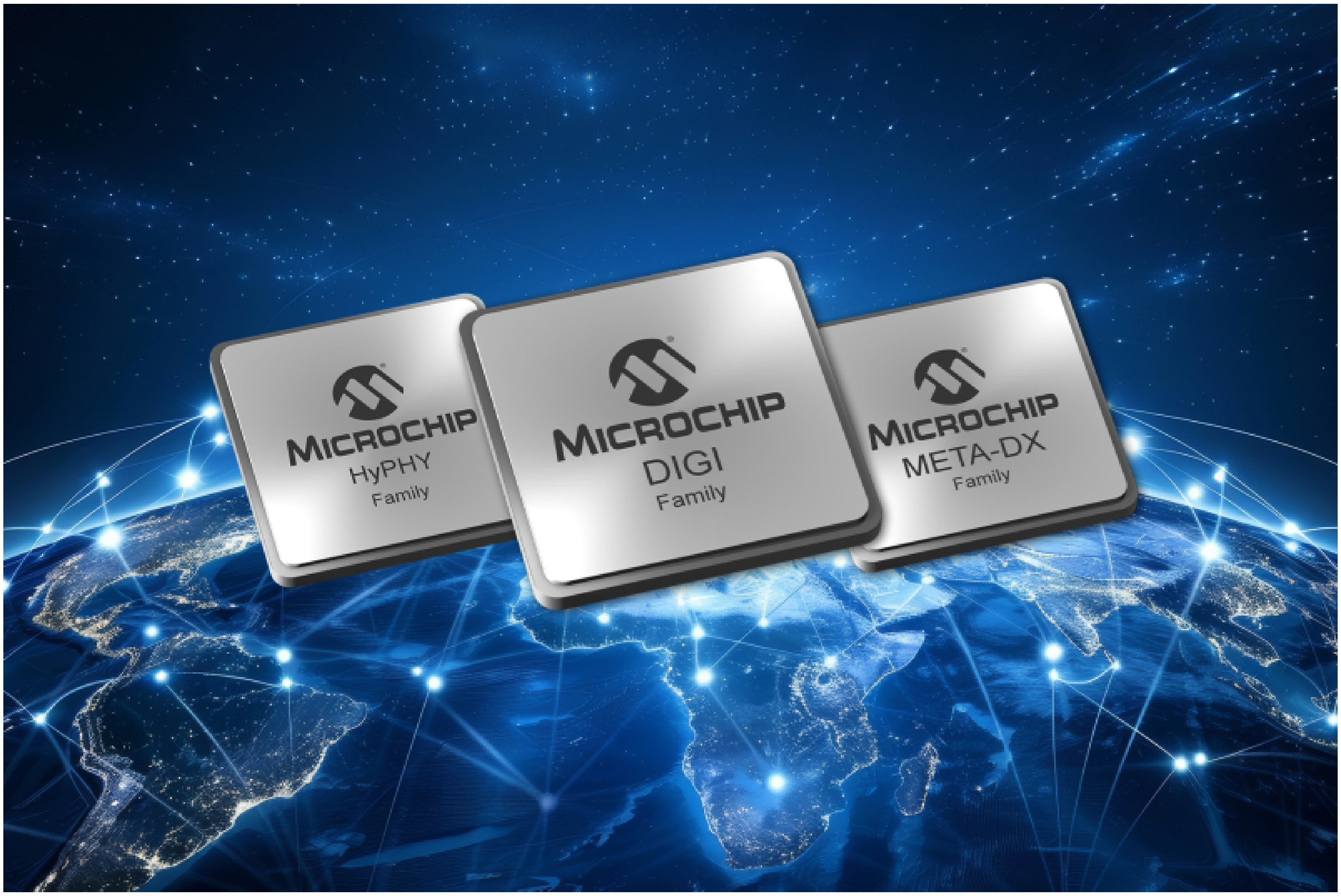


Table of Contents

1. Preface

This Optical Transport Network (OTN) tutorial is a merged, updated version of the author's previous two popular OTN tutorial white papers:

- S Gorshe, "A tutorial on ITU-T G.709 Optical Transport Networks (OTN)," PMC-Sierra, [PMC-2081250], 2009 and 2010
- Steve Gorshe, "The Evolution of ITU-T G.709 Optical Transport Networks (OTN) Beyond 100Gbit/s" Microsemi, [ESC-2170438], March 2017

The merged content of those white papers has been reorganized, enhanced for expanded background and improved clarity. The updates include covering more recent OTN protocol development and extensions. The intention of this tutorial is to introduce the reader to key OTN concepts, including FlexO and FOIC. Specifically, the level of detail in the material and background explanation is intended to help the reader understand the concepts and make effective use of the associated ITU-T OTN Recommendations. Some details are omitted, especially when they pertain to less commonly used OTN features.

About the Author

Steve Gorshe, Ph.D. (Fellow, IEEE) has worked since 1982 in research and development of telecommunications systems and ICs at companies including GTE, Siemens, NEC, PMC-Sierra. He is currently an Associate Fellow at Microchip Technology, where he has contributed to patents and architectural designs related to multiple OTN product families including DIGI, META, and HyPHY. Since 2017 he has been Rapporteur for Q11 of ITU-T SG15, which is responsible for optical transport network standards. He also chaired the IEEE P802.3cx Task Force on accurate timestamping over Ethernet. He has been actively involved in telecom and data communications standards since 1983 and has been author/co-author on over 800 standards contributions across multiple standards organizations. He has 52 patents issued or pending, is author/co-author on multiple telecommunications textbooks and technical papers. Steve received his Ph.D. and MSEE from Oregon State University and BSEE from the University of Idaho.

2. Abstract

The initial ITU-T Optical Transport Network (OTN) standards were developed in the 1998-2001 timeframe. The motivations included providing a cost-effective way to grow the capacity and efficiency of optical transport networks and reducing network operating expenses through improved network management overhead. OTN has proven to be a very flexible protocol, capable of evolving to support new types of client signals and increasing signal rates. For example, as Ethernet became an increasing important client signal, OTN evolved in its ability to efficiently carry Ethernet and became the basis for long-reach Ethernet connectivity. This tutorial covers the basic signal format and features of OTN along with background on how and why they evolved.

3. Introduction

The initial ITU-T Optical Transport Network (OTN) standards were developed in the 1998-2001 timeframe. At that time, optical telecom networks around the world were based on the ITU-T Synchronous Digital Hierarchy (SDH) and equivalent North American Synchronous Optical Network (SONET) standards. The OTN frame format was optimized for carrying SONET/SDH signals at the nominal client rates of 2.5, 10 and 40 Gbit/s.[1] The motivations behind the OTN frame format were to provide a mechanism for carrying client signals with a relatively simple "digital wrapper" that included integrated FEC and improved overhead functionality not available with SONET/SDH. Specifically, the OTN digital wrapper and overhead were optimized for use with the wavelength division multiplexing (WDM) technologies that had become practical. The combination of SONET/SDH time division multiplexing (TDM) and WDM significantly increased the per-line core network capability in a more cost-effective manner than was practical with TDM alone.

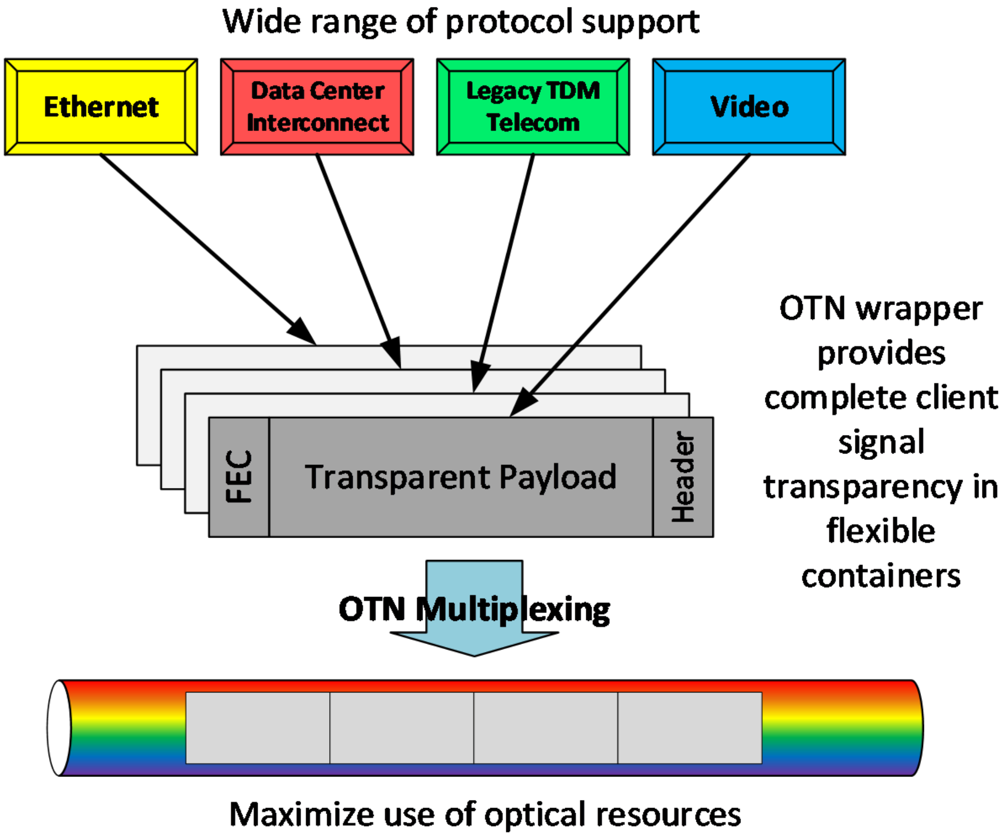
The costs of network operations, administration, and maintenance and provisioning (OAM), collectively known as operating expense (OpEx) are extremely important to network operators. Consequently, network operator decisions are driven by OpEx when competing technologies have comparable equipment capital expense (CapEx) costs. The optimized overhead features of OTN are thus an especially important differentiator between OTN and both previous technologies and more recent alternatives including Ethernet. Network users with higher quality requirements have been willing to pay for the "carrier grade service" provided by telecom networks relative to datacom networks.

OTN subsequently evolved to support new applications and types of clients. The first step was enabling TDM multiplexing of lower rate OTN signals into the higher rates (e.g., allowing a 40 Gbit/s OTN signal to carry 16 of the 2.5 Gbit/s or four of the 10 Gbit/s OTN signals as its clients). Further evolution enabled more flexible TDM and provided optimizations for carrying Ethernet and other packet-based client signals and progression to increasingly higher OTN rates. As shown in the figure below, this led to OTN becoming a common transport network technology for a wide variety of client types. The flexibility and cost-effectiveness of OTN allowed its applications to evolve from use in optical core backbone networks to broad use in metro, metro aggregation and some access applications (refer to the Application Example figure.

---

[1] SONET and SDH were co-developed in parallel and differ primarily in their optimization for their primary regional clients. Specifically, SONET was optimized for carrying the voice network hierarchy based on 1.544 Mbit/s DS1 clients and SDH was optimized for the voice network hierarchy based on the 2.048 Mbit/s E1 clients. In addition to the difference in the lowest client signal rates, SONET was based on a 51.84 Mbit/s STS-1 modularity suited to carrying 45 Mbit/s DS3 signals while SDH was based on 155.52 Mbit/s STM-1 modularity suited to carrying 139 Mbit/s E4 clients. The formats converge in that the 155.52 Mbit/s STM-1 frame is structured as the equivalent of three 51.84 STS-1 frames. The interested reader can find the SDH/SONET details in ITU-T G.707 and their legacy client details in ITU-T G.704.

Figure 3-1. Illustration of OTN as a Common Transport Technology

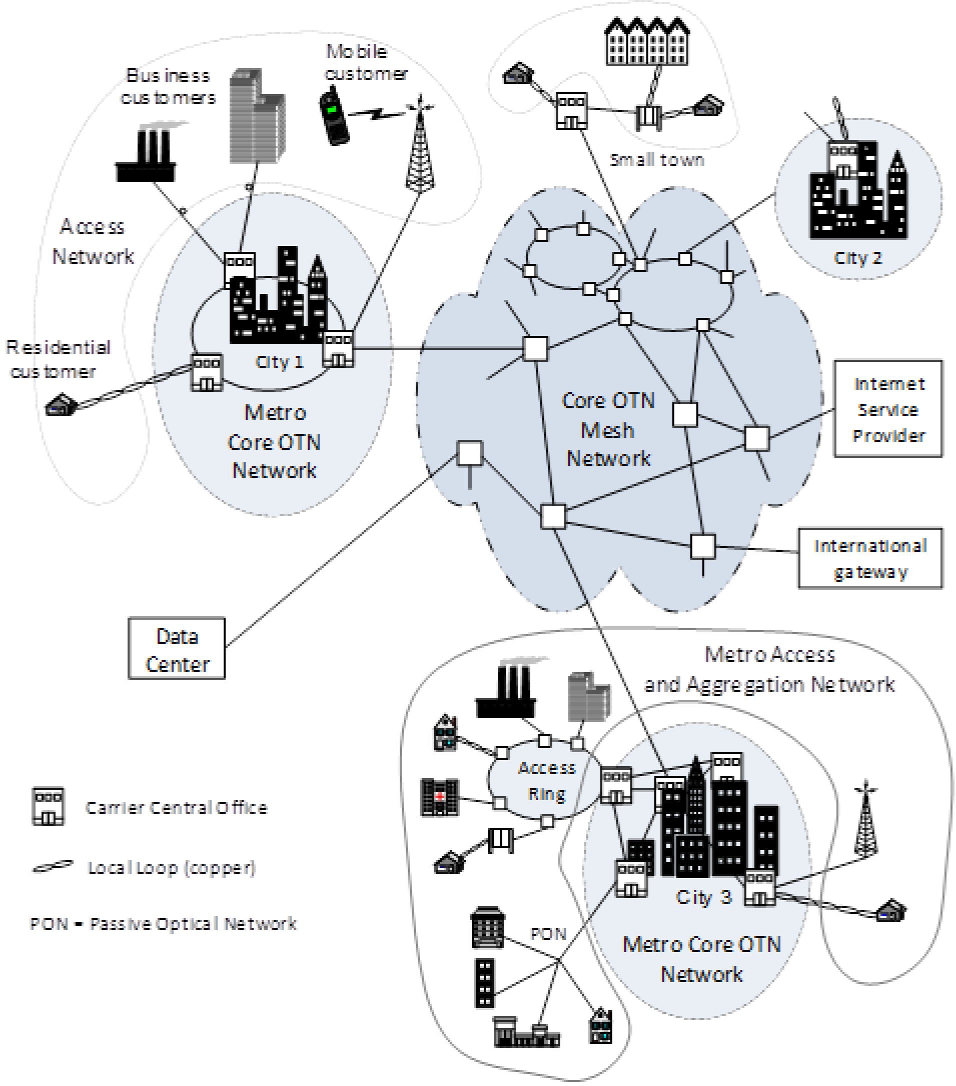


The core ITU-T standards (called Recommendations) pertaining to OTN are:

- ITU-T Rec. ITU-T G.709 (2020), *Interfaces for the optical transport network.* which defines the OTN digital signal format, including rates, overhead, how client signals are mapped and multiplexed into OTN.
- ITU-T Rec. ITU-T G.798 (2023), *Characteristics of optical transport network hierarchy equipment functional blocks*, which defines the functional requirements for network equipment that implements G.709.
- ITU-T Rec. G.709.1 (2024), *Flexible OTN common elements*, which Defines a physical layer signal for carrying OTN signals at rates beyond 100 Gbit/s. The companion G.709.3, G.709.5 and G.709.6 Recommendations define the FEC frame format for different Flexible OTN (FlexO) applications.

See Appendix D: References and Standards Related to OTN for a complete list including related standards.

Figure 3-2. **OTN Application Example**



This tutorial is organized around both historical and functional formats. The historical approach makes it easier to see how the key elements of the OTN frame format evolved to support new functionality. The functional approach allows a more compact description of similar types of overhead, and also allows side-by-side comparisons of different approaches to the same functions.

4. Overview of OTN Topology and Equipment

Figure 4-1 illustrates a Layer 1 telecom network, providing context regarding network topology and introducing some of the terminology used in this tutorial. The Figure 4-1 terminology is explained in Table 4-1. See the Glossary section for a full list of terms and abbreviations.

Table 4-1. Definitions of network terminology used in Figure 4-1

Term	Definition
ADM	Add/Drop Multiplexer
LTE	Line (MS) Terminating Equipment
MS	Multiplex Section – Connection between multiplexer and demultiplexer
Path	End-to-end connection between mapper and demapper
PTE	Path Terminating Equipment
RS	Regenerator Section – Connection between a terminal and a regenerator
STE	Section (RS) Terminating Equipment
Tandem Connection	A portion of the Path with its own monitoring overhead
TCTE	Tandem Connection Terminating Equipment

In general, as illustrated in Figure 4-1, a “Section” layer is point-to-point between either OTN LTE/ADM equipment (i.e., a MS), or between an OTN terminal and a regenerator (i.e., RS). The “Path” layer covers the entire path between OTN endpoints. This can include the terminals where a client signal is inserted into and removed from OTN, or where a multiplexed OTN signal (sometimes called a High Order Path) is generated and terminated. The purpose of a Tandem connection is to provide monitoring for a portion of the Path. Multiple Tandem connections can be concatenated or nested along the path, each associated with a different network operator domain or portion of the network.

Expanding on the network equipment shown in Figure 4-1, there are several different types of optical transport network equipment being deployed based on the OTN standards. The most common types include:

- Regenerators
- OTN terminal equipment
- Reconfigurable Optical Add/Drop Multiplexer (ROADMs)
- Optical cross connect (OXCs)

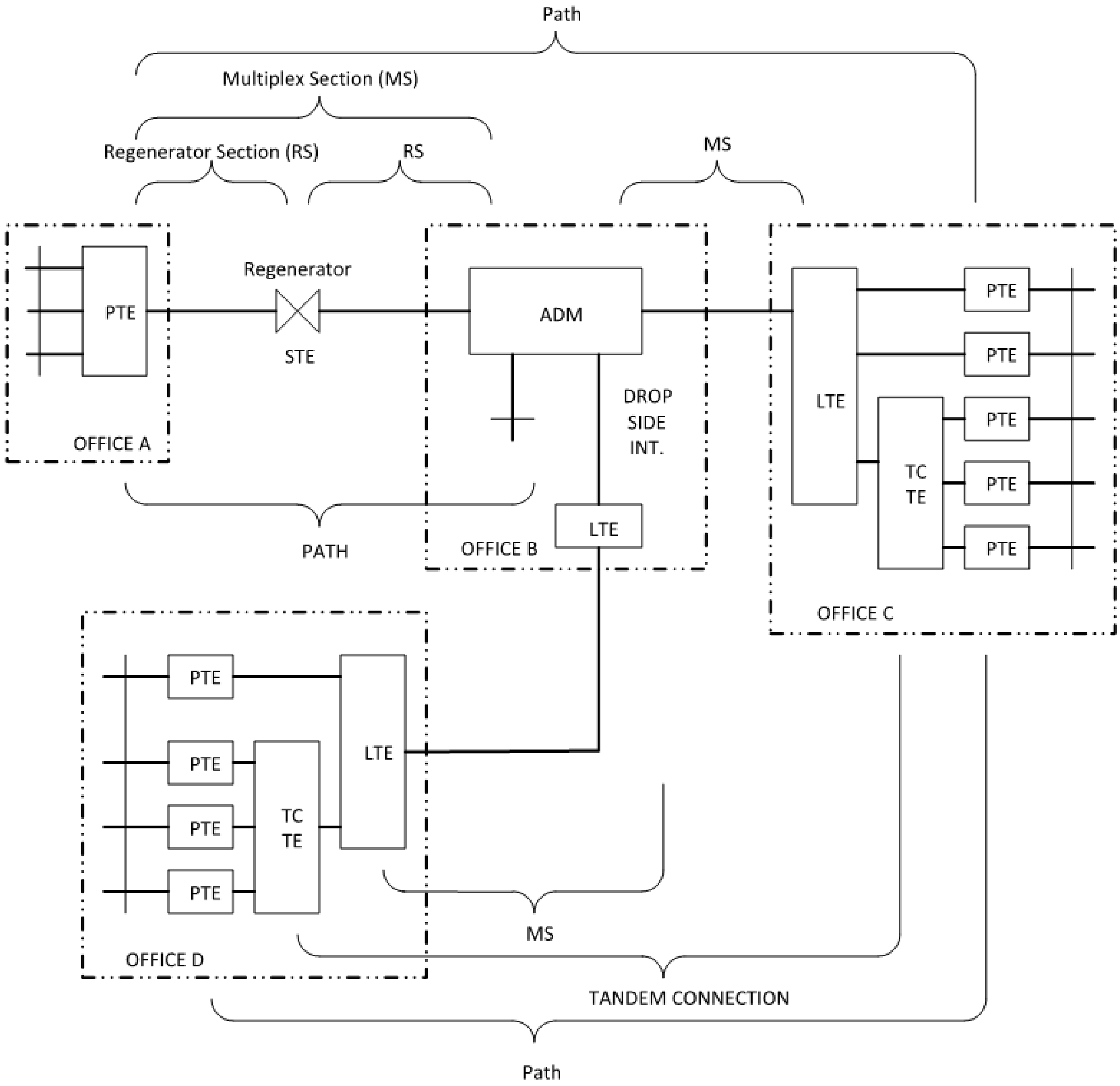
OTN terminal equipment is used for point-to-point connections through WDM networks, mapping the client signals into the OTN frame payload area, sometimes multiplexing multiple signals in the electrical domain, and finally performing mapping/multiplexing in the optical domain. OADMs, OXCs, and some types of regenerators primarily process the OTN signals in optical domain. Refer to Appendix A for more discussion on these three types of equipment.

The key building blocks of today’s ROADM node can be categorized into three primary functions:

1. Wavelength add/drop filters or switches – This is generically referred to as a wavelength fabric and operates only in the optical domain. However, it can be implemented with a number of different technologies, including wavelength blockers and Wavelength Selective Switches (WSSs). The wavelength fabric multiplexes and demultiplexes all of the individual DWDM wavelengths from the client interfacing cards. The wavelength fabric also provides optical protection.
2. Dynamic power control and remote monitoring capabilities at the optical layer – Optical amplification with dispersion compensation and gain equalization, dynamic power control and remote monitoring for the presence/absence of optical signals are just a few of the many advancements that have reduced the need for onsite technicians for node engineering.

3. Optical service channel termination and generation - Traditionally this is in the form of transponders and muxponders.

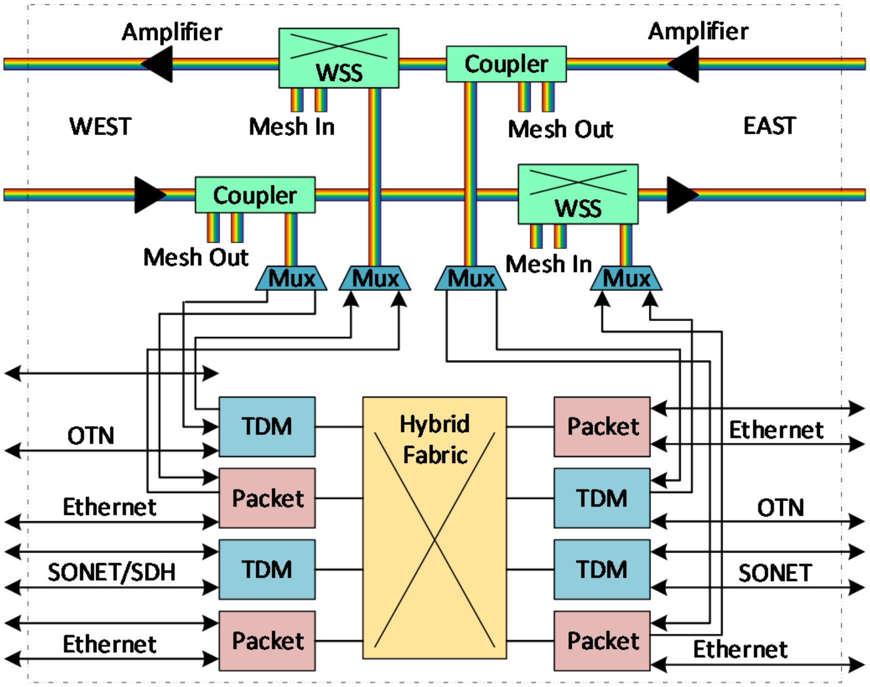
Figure 4-1. Network Overview Illustration



It has become common to augment classic ROADM functionality with switching fabrics in the electrical domain, as illustrated in the Figure below. The electrical domain switching can be TDM, packet switching, or a hybrid of both. This type of network element is sometimes referred to as a Multi-Service Provisioning Platform (MSPP) or Packet Optical Transport Platform (P-OTP) or Packet OTP. The motivation is to allow adding and dropping client signals within the signals carried over the wavelength rather than just adding or dropping the entire wavelength. This finer granularity add/drop allows aggregation or grooming for more efficient use of the wavelengths.

The type of network element illustrated in the Figure below also allows more flexible network topologies. In some carrier networks, ROADMs are being deployed in order to build the network infrastructure for video signal delivery. The ROADM performs the legacy SONET/SDH ADM functions on the wavelengths carrying SONET/SDH signals. The video signals are expected to be carried on separate wavelengths. Optical domain switching can be used to add/drop entire video-bearing wavelengths, and the ROADM packet switch fabric can be used to switch IP-based video signals.

Figure 4-2. Hybrid Optical Transport ROADM Illustration



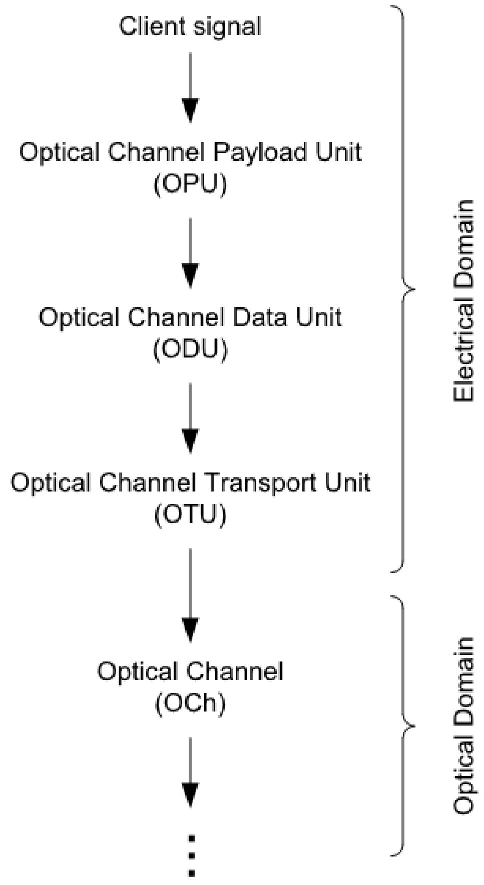
5. OTN High-Level Signal Architecture

The OTN network architecture consists of both optical and electrical (digital) domain layers. While this tutorial focuses on the electrical domain, the optical domain layers can be seen in information flow of the Figure below.

Within the electrical domain, the client signal is inserted into the frame payload area, which, together with some overhead channels, becomes the Optical Payload Unit (OPU). An OPU provides the Path payload area since it is the end-to-end container for the client. OAM overhead is then added to the OPU to create the Optical Data Unit (ODU). If the ODU carries a single client, it corresponds to a Path signal. If multiple client signals are multiplexed into the OPU of an ODU, the ODU corresponds to a Multiplex Section (MS) connection between the multiplexing and demultiplexing points. Transport overhead (e.g., frame alignment overhead) is then added to the ODU to create an Optical Transport Unit (OTU), which is the fully formatted digital signal with Regenerator Section (RS) functionality. As explained below, the “k” at the end of the OPU, ODU and OTU names indicates the signal rate (capacity). The OTU is then transmitted on a wavelength, which constitutes the Optical Channel (OCh). The OCh, OMS, and OTS layers are described in ITU-T Rec. G.872 and will not be discussed further here. The client, OPU, ODU, Tandem Connection and OTU containment relationships are illustrated in the Figure below.

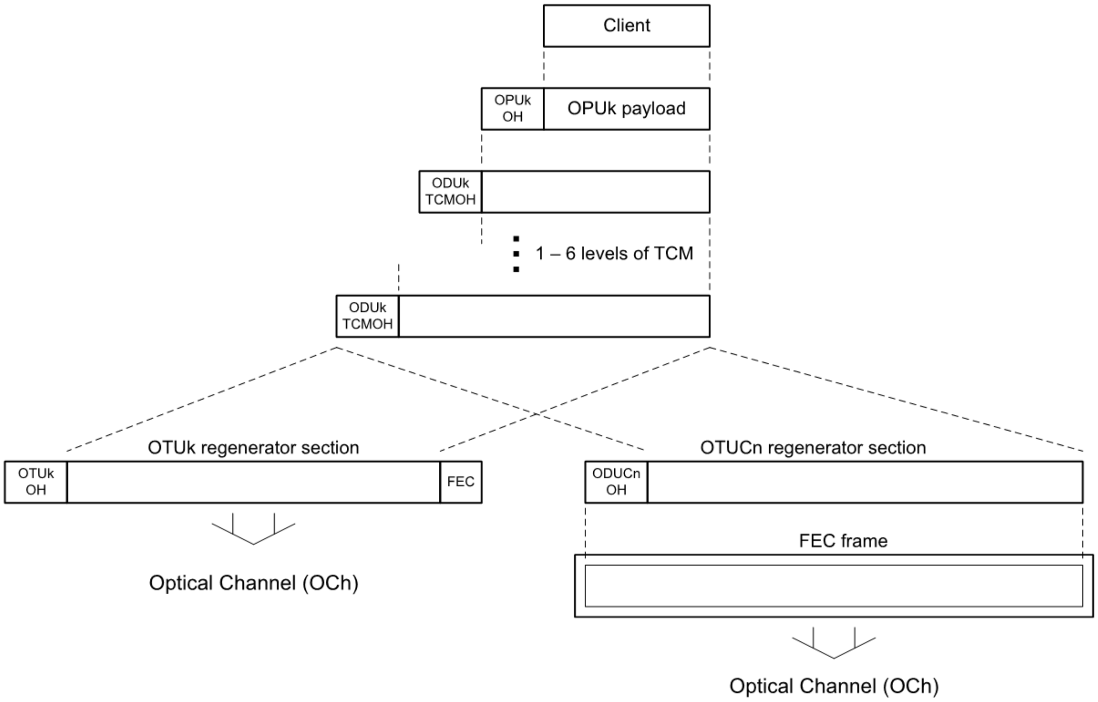
Note: The overhead fields referenced in Figure 5-2 are not new bits or bytes added to the frame at that point in the containment flow. Rather, as illustrated in Figure 6-1, the OPU, ODU, Tandem Connection and OTU refer to overhead fields within the basic OTN frame format that are used by the respective containment layer.

Figure 5-1. Information Flow Illustration for an OTN Signal



The Microchip HyPHY and DIGI device families implement the OTN electrical domain functionality.

Figure 5-2. OTN Information Containment Relationships



6. OTN Signal Formats and Frame Structure

The basic OTN frame format for signal rates up to 100 Gbit/s is illustrated in Figure 6-1. The yellow areas of the frame correspond to the OTUk overhead, the green areas to the ODUk overhead and the blue areas to the OPUk overhead. The brown area (columns 3825-4080) contains the integrated FEC overhead when it is used, or all zeros otherwise. The OTN frame format remains the same for rates from 1.25 Gbit/s (OTU0) through 100 Gbit/s (OTU4). Each column represents a byte (octet)[2], and the transmission order is octet-by-octet, row-by-row. Note that telecom numbers the bits of a byte with bit 1 being the MSB and bit 8 being the LSB. The bit transmission order is 1 through 8[3].

The defined basic OTUk signals are listed in Table 6-1, along with the corresponding ODUk and OPUk rates.

Figure 6-1. OTN Frame Structure and Format

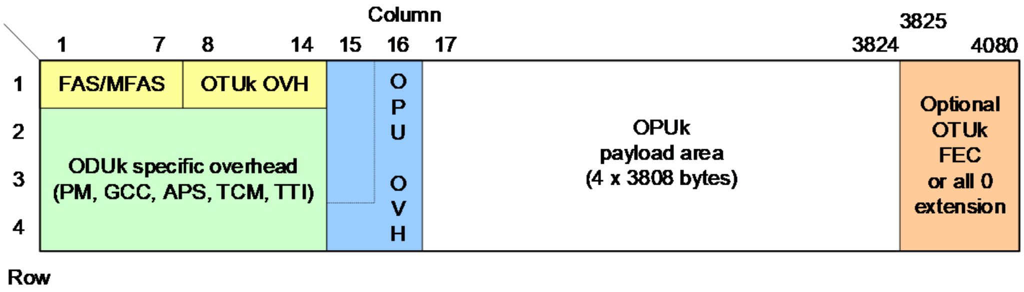


Table 6-1. Defined OTUk Signal Rates and Corresponding ODUk and OPUk

OTUk Type	OTU Nominal Rate (kbit/s)	ODUk Type	ODU Nominal Rate (kbit/s)	OPUk Type	OPU Nominal Rate (kbit/s)
OTU0	255/239 × ODU0 rate	ODU0	1 244 160	OPU0	238/239 × ODU0 rate
OTU1	255/239 ×ODU1 rate	ODU1	239/238 × 2 488 320	OPU1	238/239 × ODU1 rate
OTU2	255/239 × ODU2 rate	ODU2	239/237 × 9 953 280	OPU2	238/239 × ODU2 rate
OTU3	255/239 × ODU3 rate	ODU3	239/236 × 39 813 120	OPU3	238/239 × ODU3 rate
OTU4	255/239 × ODU4 rate	ODU4	239/227 × 99 532 800	OPU4	238/239 × ODU4 rate

1. The OTN signal rates beyond 100 Gbit/s are covered below in section 7.
2. OTN rates at 25 and 50 Gbit/s were added later, as discussed in section 4.2

The reasons for these specific rates are as follows. The ODU1, ODU2 and ODU3 were the initial signals specified for OTN, optimized for carrying SONET STS-48, STS-192 and STS-768[4]. The decimal numbers in the ODU nominal rates column for ODU1, ODU2 and ODU3 correspond to the nominal rates of those SONET/SDH signals. In contrast, the ODU4 rate was chosen such that it would allow an efficient mapping for carrying a bit-transparent 100GBASE-R Ethernet client. The addition of other ODUk rates is discussed below. For the purpose of rate calculation, observe that each region of the OTN frame contains an integer number of 16-byte groups[5].

---

[2] The term “octet” is more commonly used for the transmission of 8-bit TDM information and “byte” is more commonly used for 8-bit computer or data transmission. Since both indicate 8-bit words, byte and octet are used interchangeably within this document.

[3] The Ethernet follows the data comm. convention of having bit 0 represent the LSB of a byte with the MSB as bit 7 and uses a bit transmission order of 0 through 7.

[4] Equivalently, SDH STM-16, STM-64 and STM-256.

The next section of this tutorial describes the OTN overhead channels and functions shown in Figure 6-1 for the different OTN layers. The OTN overhead section introduces the rate justification overhead, which is responsible for adapting between the differences of the client signal rates and the OTN channel rates. The rate justification topic is covered in depth in the OTN Client Signal Payload Mapping and Multiplexing section. Specifically, the mapping and multiplexing section describes the different methods for mapping and multiplexing both constant bit rate and packet-oriented client signals into the OPUk.

6.1. ODUflex

The ODUflex uses the same format as all ODUk signals. The “flex” in the name reflects its flexibility for supporting new client mappings of any bit rate, including both constant bit rate (CBR) and packet clients. With respect to client rate, as explained below in the client signal multiplexing section, the OPU payload area can be divided into multiple TDM channels referred to as Tributary Slots (TS). An ODUflex can be specified to occupy any arbitrary number of TS in the higher-rate OPUk into which it is multiplexed. The first ODUflex types to be defined were ODUflex(CBR), which is used to carry CBR client signals and ODUflex(GFP), which is used to carry client packet streams at rates ≤100 Gbit/s. ODUflex(IMP) was added layer for carrying 64B/66B block-encoded Ethernet or FlexE clients at rates ≥100 Gbit/s. Each of the ODUflex types is a CBR signal like all other ODUk, which allows the network to handle ODUflex signals without needing to know which type of ODUflex they are.

Client mapping into ODUflex signals is defined below in the mapping and multiplexing sections.

6.2. ODU25 and ODU50

As OTN migrated to rates beyond 100 Gbit/s, client signals also migrated to higher rates. When IEEE 802.3 defined 25 and 50 Gbit/s Ethernet interfaces, it became attractive for network operators to provide service interfaces at these rates. This led to adding OTU25 and OTU50 to G.709 for use as UNIs for carrying the corresponding Ethernet rate as a client. While the OTN overhead resulted in somewhat higher bit rates for the OTU25 and OTU50 OTN interface rates, both could be supported with Ethernet optical modules. OTU25 and 25GBASE Ethernet both use NRZ line coding (also known as PAM-2). The 25GBASE Ethernet electrical and optical parameters provided enough margin to directly support the OTU25 rate. The 50GBASE interface was based on 4-level PAM-4 line coding, which provided less margin for accommodating the higher rate of the OTU50. Consequently, the OTU50 has a shorter reach relative to 50GBASE Ethernet in order to compensate for the margin.

For applications where full Ethernet signal reach is required, G.709 added informative (i.e., non-normative) "under-clocked” version of OTU25 and OTU50 that use bit rates identical to the Ethernet signals. The under-clocked versions are referred to as OTU25u and OTU50u, respectively. The obvious tradeoff is that OTU25u and OTU50u are not capable of providing bit-transparent transport for the corresponding Ethernet clients. Each of these signals uses the same OTN frame formats shown in Figure 6-1, specifically, the OTU4 type of frame format.

For the purpose of mapping or multiplexing ODU25, ODU25u, ODU50 or ODU50u into the OTUk network interfaces (k = 3, 4) they are essentially treated as specific types of an ODUflex. See the OTN Client Signal Payload Mapping and Multiplexing section regarding ODUflex multiplexing.

---

[5] Consequently, the 255 term in the numerator of the OTUk rate multiplier corresponds to 4080/16 when the FEC columns are added to the ODUk Ii.e., 4080/3824 = 255/239). Since the OPU does not contain the first 16 columns, its rate is 3808/3824 = 238/239 times the ODU rate.

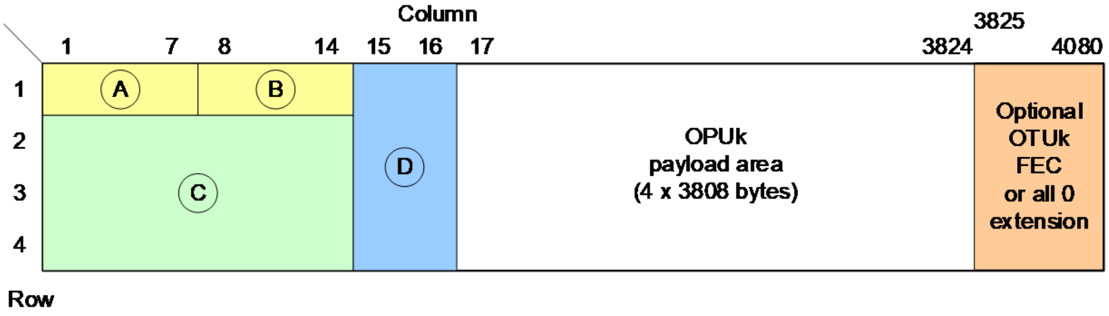
7. OTN Signal Overhead Descriptions

The different portions of the OTN overhead fields are illustrated in Figure 7-1. A summary of the OAM overhead fields used by the different layers is provided in Table 7-1, which also indicates which overhead fields are used for the OTN frame format for rates beyond 100 Gbit/s (OTUC, ODUC, and OPUC). As discussed below, with only two exceptions, the frame format for OTUC, ODUC, and OPUC uses the same overhead fields and definitions as the OTU, ODU and OPU.

Table 7-1. OTN OAM Channel and Function Descriptions

OAM Type	Used in	Function
APS / PCC	ODU, ODUC	Automatic Protection Switching / Protection Communications Channel. The APS/PCC byte is time-shared across the multiframe to create channels for the control of sub-network connection protection at the ODUk Path and each TCM level.
BDI	OTU/OTUC, ODU/ ODUC PM & TCM	Backward Defect Indication – Sent from the overhead sink to the source to indicate that a defect has been detected in the forward direction.
BEI	OTU/OTUC, ODU/ ODUC PM & TCM	Backward Error Indication – A binary count of the number of BIP-8 bits indicating errors, sent from the overhead sink to the source.
BIAE	OTU/OTUC each TCM	Backward Incoming Alignment Error – Indication sent from the overhead sink to the source that it received an IAE.
BIP-8	OTU/OTUC, ODU/ ODUC PM & TCM	8-bit Bit Interleaved Parity- Used in the OTU SM, ODU PM, and each level of TCM overhead. The BIP-8 transmitted in the overhead byte of the current frame i provides bit-wise even parity over the bits in the group of bytes it covered in frame i-2. Specifically, within the 8-bit BIP check symbol, BIP bit *j* provides even parity across bit *j* in all bytes it covers (1 ≤ *j* ≤ 8). See the OTUk FEC overhead description for more information on error rate detection.
GCC	OTU/OTUC, ODU/ ODUC	General Communications Channel. One is available in the OTU overhead and two in the ODU overhead. GCC1 and GCC2 in the ODU are clear channels whose format is not specified in G.709.
IAE	OTU/OTUC	Incoming Alignment Error – Indication sent downstream to inform the receiving NEs that a framing alignment error (e.g., a slip) was detected on the incoming signal.
MFAS	OTU/OTUC	Multiframe Alignment Signal – Binary counter used to establish the 256-frame multiframe that is used for the time-shared overhead channels that spread their content over the course of a multiframe. See Note.
OA	OTU/OTUC	Optical Alignment – Frame alignment signal for the OTU. OA1 = 1111 0110 and. OA2 = 0010 1000
PM & TCM	ODU/ODUC PM & TCM	Delay measurement overhead for the Path (PMd) and TCM level i (DTti)
STAT	ODU PM, each TCM	The STAT field provides monitoring status information for the associated layer. For example, it indicates whether the signal at layer is a normal signal or a type of maintenance signal. In the case of TCM, it also indicates whether that TCM is currently experiencing IAE.
TTI	OTU/OTUC, ODU/ ODUC PM & TCM	Trail Trace Identifier Allows the sink to confirm that it has the correct connection to the source for that layer (i.e., to detect connectivity faults). Its 64-byte string includes a source access point identifier (SAPI), destination access point identifier (DAPI), and operator-specific fields.
Note: The OAM fields that use the OTUk and OTUC overhead MFAS to time-share their bytes include APC/PCC, and TTI. As described below, the OPUk and OPUC overhead also time-shares the justification control (JC) and PSI bytes		

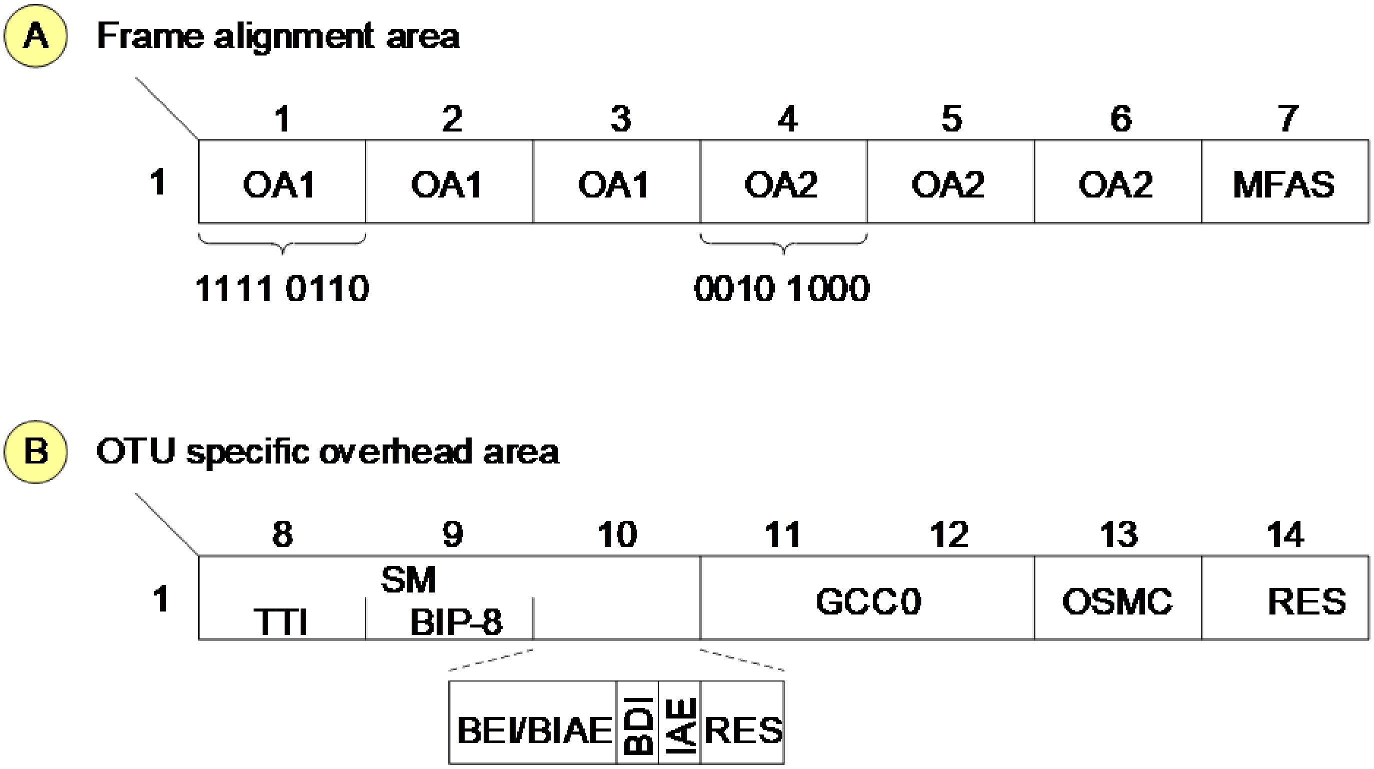
Figure 7-1. OTN Frame Overhead Fields



7.1. OTUk Overhead Fields

The OTUk portions of the OTN overhead are illustrated in the Figure below. The OTU overhead is shown as the A and B areas of Figure 7-1, with an expanded view in in the Figure below. The optional FEC is also part of the OTUk overhead.

Figure 7-2. OTUk Layer Overhead Fields



The A field of Figure 7-2 contains the optical frame alignment (OA) pattern and the multiframe alignment signal (MFAS). See Table 7-1 for the definitions of the OA and MFAS fields.

The B area of Figure 7-1 provides a GCC and section monitoring (SM) information for the OTU. The SM fields include the TTI, BIP-8, IAE, BEI, and BIAE that were discussed above in Table 7-1. In addition, the SM overhead for OTU includes an incoming alignment error (IAE) indicator. The IAE and BIAE are used to disable the error counting in their respective directions during frame alignment loss conditions.

OTN Synchronization Messaging Channel (OSMC)

One of the key early decisions for the original OTN standard was that it would not be required to transport network synchronization as part of the OTN signal since OTN client signals such as SONET/SDH can transport this synchronization. However, since that time applications have emerged that can benefit from carrying precision timing information within the OTN signal itself rather than through one of its clients. One primary application is to be able to distribute precision clock frequency and time-of-day information to the different COs within the network, without having to waste the bandwidth associated with using a client signal for it. Another application is being able to carry precision frequency and time-of-day information to the access edge of the OTN in order to time access links, including those to radio base stations.

For OTUk, the optional OSMC is carried in an OTUk overhead channel[6] . The overhead carries a packet-based precision timing protocol (PTP) using timestamps, which is an adaptation of the IEEE 1588v2 protocol over Ethernet. The Ethernet packets are encapsulated using the G.7041 Generic Framing Procedure (GFP). Note that the Boundary Clock method is used. See G.709 clause 15.7.2.4.1 for additional information regarding details including the timestamp reference point, insertion of the PTP messages into GFP frames and inserting the GFP frames into the OSMC.

FEC

Since one of the objectives behind OTN was having an integrated FEC, it was decided to structure the OTN frame around the Reed-Solomon RS(255,239,8) FEC that was already specified by ITU-T for some undersea cable systems. This integrated FEC choice is evident from the rate multipliers found in Table 6-1. As indicated by the “8” in the FEC name, this RS FEC works on 8-bit code symbols. It is implemented as a “systematic” FEC where the codeword payload area is not changed by the encoding process and the parity check bytes (symbols) appear at the end of the codeword.

In order to provide better protection against burst errors, the OTUk FEC is implemented as the interleaving of 16 independent RS(255,239) codes. See Figure 7-3 where each of the numbered rectangles correspond to one byte in the FEC sub-row. Starting at the beginning of a frame row, each of the first 16 bytes is mapped into a different FEC codeword. The same round-robin byte mapping to FEC codewords is used for each successive set of 16 bytes through the end of the OPU payload area. Each RS(255,239) codeword has 255 – 239 = 16 parity check bytes in addition to its 239 payload bytes. The parity check bytes of the 16 codes are placed into columns 3825-4080 in the same round-robin format that was used for creating the codewords. In other words, the first column of the FEC area (#3825) contains the first parity check byte of the first FEC code, the second byte contains the first parity check byte of the of the second FEC code, etc. through the 16th column of the FEC parity byte area (column #3840) containing the first parity check byte of the 16th FEC code. Likewise, columns 3841-3856 contain the second FEC parity check byte for each of the 16 FEC codes, and etc. until column 4080 contains the 16th parity check byte of the 16th FEC code. Since there are 16 FEC code words, this format provides a 16-byte FEC interleaving pattern. Each RS(255,239) codeword is capable of correcting errors in up to 8 of the 8-bit symbols and detecting errors in up to 16 symbols. Consequently, with the 16-byte interleaving, error bursts of up to 8×16 = 128 symbols can be corrected. When used for error correction, this leads to a coding gain of 6.2 dB for systems with an operating BER of 10-15.

Note that use of the FEC is optional for OTU0 through OTU3, but mandatory for OTU4. When the FEC is not used, the parity byte check locations (columns 3825-4080) are filled with all-zeros. As explained above, the OTUk payload scrambling is performed after the FEC insertion (i.e., the FEC check bytes are based on the unscrambled information).

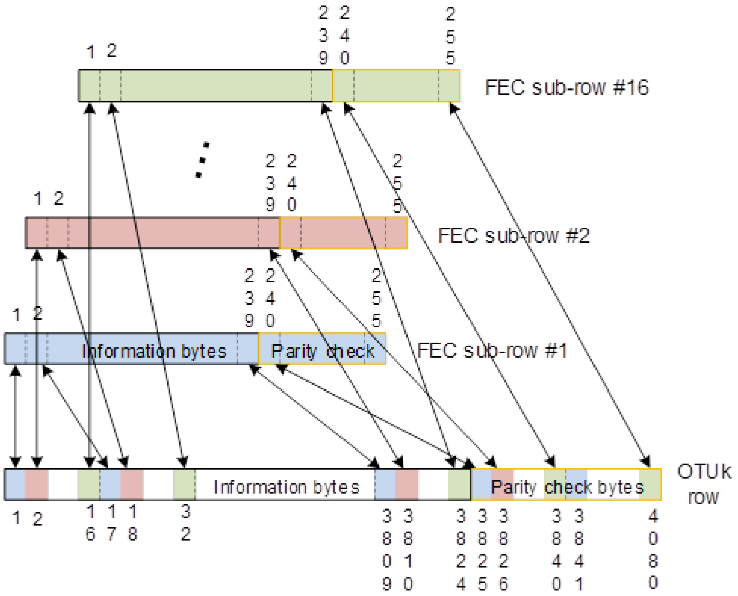
Vendor-proprietary FEC options have been used when longer reach is required. They have typically replaced the FEC bytes of the OTUk frame with FEC check symbols derived for the stronger FEC. For example, some use a combination of two different FEC codes, with each OTN byte covered by both such that if the error is not correctable by one of the FEC codes it will be correctable by the other or

---

[6] For B100G OTUCn (see below), the OSMC is carried in the FlexO signal overhead, since FlexO is the closest digital layer to the PHY.

the two in combination. The one standardized long-reach OTUk FEC is the Staircase FEC for OTU4, defined in G.709.2. Note that it is common for the stronger FEC to be implemented in the optical module rather than the framer IC. In those cases, the G.709 RS(255,239) can cover the electrical module-to-framer (MFI) interface, with the stronger FEC covering the optical span.

Figure 7-3. OTUk Layer Overhead Field Interleavings



Error Detection

As seen in Table 7-1 the BIP-8, which had also been used with SDH/SONET, is used at all OTN signal levels. The BIP-8 was chosen as a simple method to estimate the BER and is especially effective if the bit errors have a random distribution. An individual parity bit can only detect an odd number of errors within the block it covers. The BIP-8 increases the likelihood that multiple errors in the frame will be distributed among different BIP blocks. This both decreases the likelihood that multiple errors will cancel each other within a given block and increases the likelihood that the number of individual BIP bits indicating errors will approximate the total number of errors within that frame. The latter is the basis for having the far-end error BEI error report be the number of individual BIP bits that indicated errors in that received frame.

The usefulness of BIP for BER has diminished over time. First, burst errors are more common in high-speed optical systems than random errors, making “bit” error rate less meaningful. More importantly, the optional FEC integrated into the OTN frame is used in network applications where a higher pre-FEC error rate may be expected and FEC is required for the higher OTN rates where errors would be more common. As explained above, the FEC provides a far more powerful measure of the error rate. Further, if an error pattern occurs that exceeds the ability of the FEC to correct or detect it, an incorrect FEC decoding can result in undetectable errors in many bytes such that the BIP-8 errored bit count would severely underestimate the nature of the error condition. Consequently, at higher rates, especially beyond 100 Gbit/s, it has become more common to use the FEC code’s error detection capability combined with its report of uncorrectable blocks as the basis for determining the signal error condition.

7.2. ODUk Overhead Fields

The ODUk overhead structure is illustrated in the Figure below. It contains the overhead for path performance monitoring (PM), two GCC channels, APS/PCC, six levels of tandem connection monitoring (TCM), and a set of bytes reserved for experimental purposes. See Table 7-1 for the overhead channel descriptions. The PM and TCM overhead consists of a TTI, a BIP-8, status information (STAT), BDI and BEI. The TCM carries the same type of information as the PM with the addition of BIAE. The BIP-8 for the Path and each TCM layer covers all the bits in the OPU for frame i, and is reported in the Path and active TCM BIP fields of frame i+2.

Figure 7-4. ODUk Layer Overhead Fields

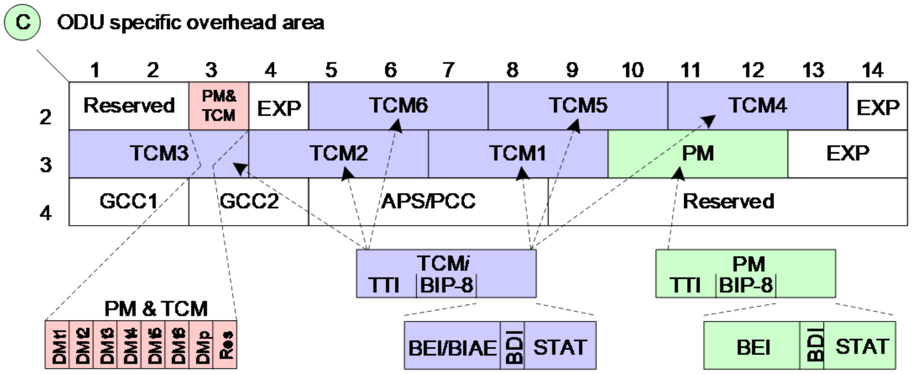


Table 7-2. ODU PM STAT Field Bit and Function Descriptions

PM byte 3 Bits 6 7 8	**Status**
0 0 0	Reserved for future international standardization
0 0 1	Normal path signal
0 1 0	Reserved for future international standardization
0 1 1	Reserved for future international standardization
1 0 0	Reserved for future international standardization (Note 1)
1 0 1	ODU-LCK maintenance signal
1 1 0	ODU-OCI maintenance signal (Note 2)
1 1 1	ODU-AIS
Note 1: A FlexO SquelchText pattern may briefly produce this bit pattern in the STAT byte. Note 2: ODUCn-OCI is reserved for future international standardization.	

Table 7-3. ODU TCM STAT Field Bit and Function Descriptions

TCM byte 3 Bits 6 7 8	**Status**
0 0 0	No source TC
0 0 1	In use without IAE
0 1 0	In use with IAE
0 1 1	Reserved for future international standardization
1 0 0	Reserved for future international standardization (Note 1)
1 0 1	ODU-LCK maintenance signal
1 1 0	ODU-OCI maintenance signal (Note 2)
1 1 1	ODU-AIS
Note 1: A FlexO SquelchText pattern may briefly produce this bit pattern in the STAT byte. Note 2: ODUCn-OCI is reserved for future international standardization.	

Delay Measurement

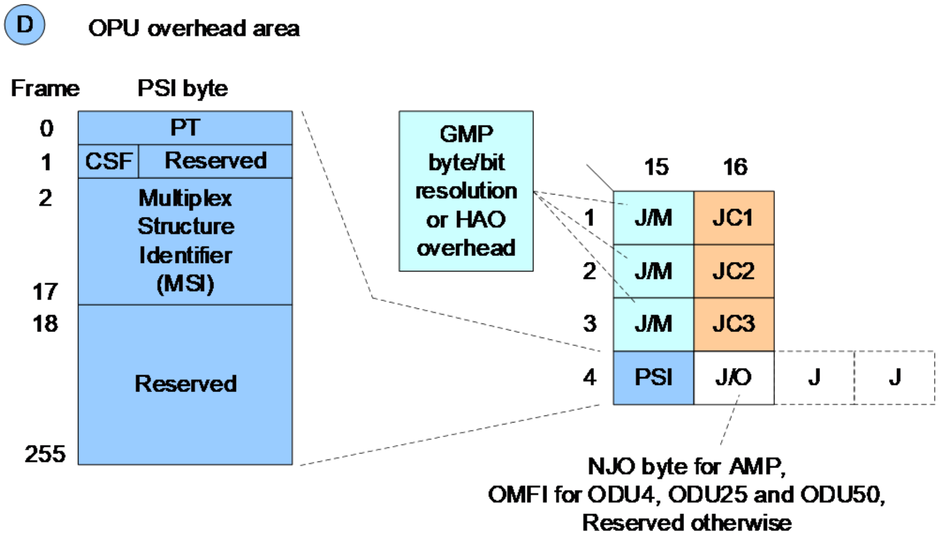
The ODUk overhead also enables a delay measurement capability to perform round trip delay measurement at the Path and TCM levels. This capability was added to OTN in 2009 at the request of many carriers in order to guarantee the performance of some delay-sensitive services carried over OTN. Examples of such service applications include Storage Area Network (SAN) signal transport and financial transaction communications such as for stock or commodity trading.

The delay measurement is performed as follows. One node is provisioned to be the source of the delay measurement pattern, and the node at the other end of the path is provisioned to be the loopback node for that measurement signal. The originating node then measures the delay between sending the pattern and receiving it back from the far-end node. Specifically, the source node initiates the latency measurement by toggling the PM or TCM delay measurement bit and initiating an OTN frame counter. Since only a single bit is required for the pattern, it was possible to locate the PM and all TCM DM bit in the same ODUk overhead byte (the PM&TCM byte of row 2, byte 3). The loopback node for that signal transmits the received delay measurement bit value back to the source in the same overhead bit location. The source node measures the round-trip delay as the number of OTN frame periods between when it transmits the toggled bit and when it receives that bit value from the loopback node. The measurement resolution is approximately two ODUk frames. This type of information would typically be carried in an ODU2 or higher rate ODUk. The ODU2 frame period is approximately 12.2 µs, which provides a much better resolution than the 500 µs resolution required by the target applications.

7.3. OPUk Overhead Fields

The OPUk overhead fields are illustrated in Figure 7-5. As explained below, it carries mapping and adjustment specific overhead information.

Figure 7-5. OPUk Layer Overhead Fields



PSI: The Payload Structure Identifier (PSI) byte carries relatively static information, which makes it possible to time-share this byte across the 256-frame OTN multiframe identified by the MFAS. The first byte of the PSI contains the payload type (PT) information, which species the OPUk payload area structure. The second byte of the PSI includes a client signal fault (CSF) indicator that is set if a fault has been detected on the incoming client signal. The remaining bytes carry the multiplex structure identifier (MSI), which provides client to Tributary Slot mapping information, and a client signal fault (CSF) indicator. See OTN Client Signal Payload Mapping and Multiplexing for the explanation of Tributary Slots, as well as Table 8-4 and Table 8-5 for illustrations of the MSI fields.

JC1 – JC3: The justification control (JC) bytes are used for adapting the payload area to accommodate the rate of the client signal. The JC byte definitions and functions depend on the justification method being used. See CBR Client Mapping for descriptions of the justification method and associated JC byte definitions.

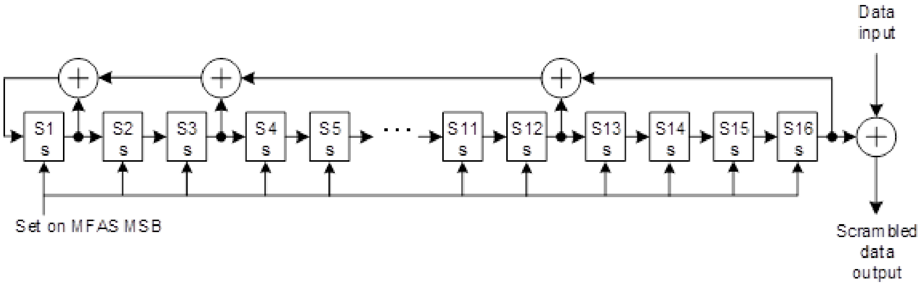
J/O, J/R, J: These bytes are used for justification control in some applications, mapping control functions in other applications. The J/R bytes contain fine resolution GMP overhead when GMP is used and in applications that use hitless resizing of ODUflex(GFP) (HAO), the J/R bytes contain the HAO overhead. The J byte used for a Positive Justification Opportunity with AMP. The J/O byte is used for a Negative Justification Opportunity with AMP and as an OPU Multiframe Indicator (OMFI) for OPU4, OPU25 and OPU50. See CBR Client Mapping for the descriptions of how these bytes are used for the OTN mapping and multiplexing options.

7.4. OTUk Scrambling

A frame-synchronous scrambler is used over the OTUk frame for the following reasons. First, the scrambler prevents long sequences of “1” or “0” patterns, which helps ensure a sufficient number and spacing of bit transitions in the OTUk signal to aid with signal timing recovery at the receiver. Further, it helps to ensure that the receiver framer has correctly determined the start of the OTN frame when it sees the 6-byte OA1 + OA2 pattern. The frame synchronous scrambler operates as follows.

OTUk frame scrambling is performed after the FEC calculation and insertion (see below). The scrambler, which uses $1 + x + x^3 + x^{12} + x^{16}$ generator polynomial, is reset to 0xFFFF on the MSB of the MFAS byte in each frame. See Figure 7-6. Consequently, the OA1 and OA2 frame alignment bytes are not scrambled since their bit pattern must be visible to the receiver framer. The scrambler runs continuously at the OTUk line rate and the output of the x^{16} bit position is XORed (i.e., added modulo 2) to the current OTUk frame bit from the MFAS MSB through the end of the frame (i.e., the last bit of byte 4080 in the fourth row). In other words, with the exception of OA1 and OA2, every bit of the OTN frame, including the FEC fields, is scrambled.

Figure 7-6. OTUk Frame Synchronous Scrambler Illustration (k=0,1,2,3,4)



7.5. Maintenance Signals

Maintenance signals are transmitted when OTN signal connectivity needs to be maintained but a client signal is unavailable due to either a fault condition or administrative activity. OTN uses three types of maintenance signals.

An Alarm Indication Signal (AIS) is transmitted downstream of a detected client fault condition. The AIS signal informs downstream nodes that the problem has already been detected upstream, which avoids a cascade of alarm reports that would complicate detecting the location of the original fault. The AIS signal also preserves the OTN signal rate and in some cases the framing at that level so that service can be restored quickly when the fault condition clears. The OTN AIS signals are defined as follows:

- OTUk-AIS: For OTUk (k=1,2,3), the entire OTUk signal is replaced by a repeating 2047-bit long polynomial number 11 (PN-11) repeating sequence defined by the generator polynomial $1+x^9+x^{11}$.
- OTUCn-AIS: An all-1s pattern that fills all OTUCn bytes except the frame alignment signal (OA) bytes.

- ODUk-AIS: An all-1s pattern that fills all the ODUk bytes (i.e., all the OTUk bytes except the frame alignment, OTUk overhead and FEC bytes). The ODUk-AIS may be extended by not over-writing (i.e., preserving) the GCC1, GCC2, EXP, APS/PCC and/or one or more TCM overhead bytes. ODUk-AIS is detected by monitoring the ODU STAT bits in the PM and TCMi overhead fields.

The open connection indicator (OCI) is used to indicate that no signal is connected at the optical transport module (OTM) interface associated with that ODUk path in the optical domain. It is signaled as a repeating “0110 0110” pattern that fills all the ODUk bytes (i.e., all the OTUk bytes except the frame alignment, OTUk overhead and FEC bytes).

When maintenance activities have locked an ODU so that it is not available for client traffic, the ODU-LCK replacement signal is inserted. The ODUk-LCK repeating 0101 0101 may be extended by not over-writing (i.e., preserving) the GCC1, GCC2, EXP, APS/PCC and/or one or more TCM overhead bytes. ODUk-LCK is detected by monitoring the ODU STAT bits in the PM and TCMi overhead fields.

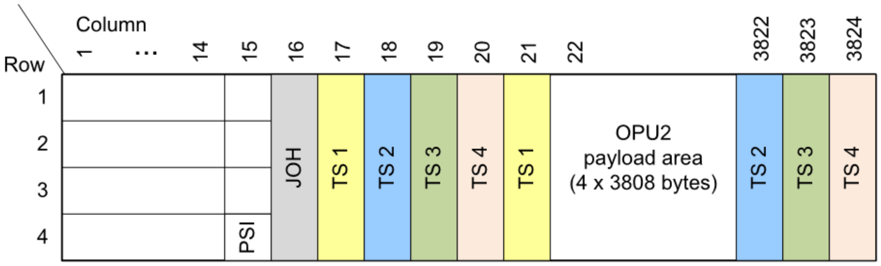
As can be seen from Figures 7-1 and 7-2, the patterns inserted for ODUk-AIS, ODUk-OCI and ODUk-LCK will automatically produce the proper pattern for their detection in the ODU STAT fields.

8. OTN Client Signal Payload Mapping and Multiplexing

G.709 supports transporting both constant bit rate (CBR) client signals and packet-based client signals. For OTN, the term “mapping” applies to applications where the entire OPUk payload area is dedicated to carrying a single client. As the name implies, the term “multiplexing” applies to using TDM for carrying multiple clients within the same OPUk payload area.

In order to support multiplexing, the OPUk payload area is divided into separate regions called Tributary Slots (TS). A given client can be carried in one or more TS, depending on the bandwidth (channel capacity) required for carrying that client. For OPU1, OPU2 and OPU3, the OPUk byte columns are assigned to the different TS on a round robin basis. See Figure 8-1 for an example illustration. More details are provided in the CBR Client Multiplexing section, with Figure 8-8 illustrating OPU1 through OPU3 and Figure 8-9 illustrating OPU4.

Figure 8-1. Illustration of OPUk TS Structure (Example of OPU2 with 2.5G TS)



When multiplexing was first added to OTN, the TS of OPU2 and OPU3 were defined to have a nominal capacity of 2.5 Gbit/s (2.5G TS). This provided 4 TS within OPU2, as illustrated in Figure 8-1, and 16 TS within the OPU3. As mentioned above and explained below, an ODU0 (with a nominal rate of 1.25 Gbit/s) and the ODU4 (with a nominal rate of 100 Gbit/s) were subsequently defined. The desire for efficient multiplexing of the ODU0 into higher rate ODUk payload areas and the desire to improve the OPU4 efficiency through finer TS granularity led to defining a new TS with a nominal capacity of 1.25 Gbit/s (1.25G TS). The resulting number of 1.25G TS within the OPU1, OPU2, OPU3 and OPU4 is 2, 8, 32 and 80, respectively. Note that only a single CBR mapping method is defined for each client signal into a particular OPUk. In this manner, the legacy mappings were preserved rather than creating new options associated with the 1.25G TS.

Since the payload capacity of the OPUk or set of its TS will typically not be an exact match for the client signal rate, a rate justification method is required to accommodate this rate difference. OTN has multiple rate adaptation methods, including three for mapping/multiplexing CBR clients and two associated with mapping packet clients:

- CBR client rate justification methods:
  - Asynchronous mapping procedure (AMP)
  - Bit synchronous mapping procedure (BMP)
  - Generic mapping procedure (GMP)
- Rate justification methods for packet clients:
  - GFP idle frame insertion for clients mapped using G.7041 GFP encapsulation
  - Idle mapping procedure (IMP)

As explained below in the Packet-Oriented Client Mapping section, when packet client signals are mapped into an OPUk, the associated ODUk can then be treated as a CBR signal when it is multiplexed into a higher-rate OPU.

Since most OTN client types are CBR signals, this section begins with descriptions of CBR client mapping and multiplexing. Then the mapping procedures for the increasingly important packet-oriented clients are described.

In the event of a client signal failure, the client signal is replaced by a client maintenance signal. The CBR and packet client replacement signals are defined in their associated client mapping sections.

8.1. CBR Client Mapping

As discussed in the introduction, the initial set of CBR client signal mappings defined for G.709 OTN were SDH STM-16, STM-64, and STM-256 (SONET STS-48, STS-192 and STS-768), which are referred to as CBR2G5, CBR10G and CBR40G and mapped into OPU1, OPU2 and OPU3, respectively. The OPU1 payload area bandwidth was defined to be fully occupied by the CBR2G5 client within the tolerance of the rate justification control explained below. The OPU2 payload area includes 16 fixed stuff byte columns (columns 1905-1920) such that its remaining payload bandwidth is fully occupied by the CBR10G client within the tolerance of the rate justification control explained below. Similarly, The OPU3 payload area includes two sets of 16 fixed stuff byte columns (columns 1265-1280 and 2545-2560) such that its resulting bandwidth is fully occupied by the CBR40G client within the tolerance of the rate justification control explained below. The purpose of the fixed stuff columns is to increase the OPU rate enough to accommodate multiplexing lower rate ODUs into it with the fixed stuff columns removed. Consequently, since the number of TS increases linearly for the OPUk with increasing k and no fixed stuff columns are present when multiplexing into an OPUk, the TS rate increases slightly with increasing k.[7],[8]

The two initial methods defined for mapping CBR client signals into an OPU were the Asynchronous Mapping Procedure (AMP) and Bit-Synchronous Mapping Procedure (BMP). As explained below, the desire for significantly improved flexibility motivated the subsequent definition of the Generic Mapping Procedure (GMP).

8.1.1. CBR Client Maintenance (Replacement) Signal

Before describing the three mapping procedures, it is worthwhile defining the generic AIS signal that replaces (i.e., fills the OPU or OPU TS bandwidth) in the event that a client signal fault makes it unavailable. See the discussion of AIS above in the Maintenance Signals section. Generic AIS consists of a repeating 2047-bit long polynomial number 11 (PN-11) repeating sequence defined by the generator polynomial $1+x^9+x^{11}$.

8.1.2. Bit-Synchronous Mapping Procedure (BMP)

With BMP, the OPU clock is derived directly from the client signal clock (e.g., CBR10G signal). Specifically, as shown below, the ODU rate resulting from BMP is the client rate multiplied by the ratio of the number of ODU columns (3824) divided by the number available payload columns (i.e., 3824 minus overhead and fixed stuff columns). Because the OPU is frequency and phase locked to the client signal, there is no need for dynamic frequency justification. The JC bytes contain fixed

---

[7] Alternatively, one can say that an OPUk rate was defined in order to accommodate multiplexing the LO ODU signals into its payload container. The fixed stuff bytes are required in order to reduce the OPUk container rate to be close enough to the nominal STM-*n* rate that the AMP periodic justification opportunities can handle the remaining frequency offsets.

[8] For example, the rate of an ODU1 signal is (239/238)(STM-16 rate), with the 239/238 factor accounting for the ODU1 and OPU1 overhead. In order to carry four ODU1 signals, the OPU2 payload rate must be at least (4)(239/238)(STM-16 rate) = (239/238)(STM-64 rate). The actual ODU2 rate is (239/237)(STM-64 rate) = (239/238)(238/237)(STM-64 rate). The 239/238 factor here accounts for the ODU2 and OPU2 overhead, and the 238/237 factor accounts for the additional OPU2 payload bandwidth required to carry the ODU1 overhead of the four ODU1 signals. When an STM-64 is mapped into the OPU2, the 16 fixed stuff columns fill the bandwidth corresponding to this 238/237 factor. Similarly, the OPU3 includes enough fixed stuff columns to accommodate carrying 16 ODU1 signals.

values, the NJO contains a justification byte, and the PJO contains a data byte. Note that for most clients, byte alignment is used rather than bit alignment.

BMP may be used for mapping SDH STM-16, STM-64, or STM-256 signals into ODU1, ODU2, or ODU3, respectively. When an STM-16 is bit-synchronously mapped into an OPU1 payload container, the STM-16 rate is used directly for OPU1 container rate. The resulting ODU1 rate is:

```
ODU1 rate = (239/238)(STM-16 client rate)
```

When an STM-64 is bit-synchronously mapped into an OPU2, the resulting ODU rate takes into account the presence of the OPU2 Fixed Stuff columns that are used for the mapping. As a result, the ODU2 rate is:

```
ODU2 rate = (239/238)(STM-64 client rate + fixed column rate) =
```

```
     = (239/237)(STM-64 client rate)
```

The BMP STM-256 into ODU3 mapping similarly takes the Fixed Stuff columns into account.

BMP is also used for mapping all CBR client signals (e.g., Fibre Channel and many of the bit-transparent Ethernet signals) into an ODUflex(CBR), which results in an ODUflex(CBR) rate of.:

```
ODUflex(CBR) rate = (239/238)(CBR client rate)
```

(Note that the ODU2e is a special case of ODUflex(CBR) that is used for carrying bit-transparent 10GBASE-R Ethernet clients. Unlike other ODUflex mappings, ODU2e retains the same fixed stuff columns that are used for STM-64 into ODU2.)

As is explained later in this section, the ODUflex(CBR) signals are then multiplexed into a HO OPUk.

8.1.3. Asynchronous Mapping Procedure (AMP)

With asynchronous mapping, the OPU clock is generated locally. The adaptation between the OPUk payload rate and the client signal rate is performed through the use of the justification control (JC) bytes and their associated Negative Justification Opportunity (NJO) and Positive Justification Opportunity (PJO) bytes. The NJO provides a location for inserting an additional data byte if the client signal is delivering data at a faster rate than the OPUk payload area can accommodate. The PJO provides a stuff opportunity if the client signal is delivering data a lower rate than the OPUk payload area can accommodate.[9] The demapper ignores the contents of the NJO or PJO bytes whenever they carry a justification byte. As illustrated in Table 8-1, bits 7 and 8 of JC are used to indicate the contents of the NJO and PJO. The mapper assigns the same value to these bits in each of the three JC bytes in an OPUk frame so that the demapper can perform a two-of-three majority vote for error correction.

Note that since the justification control is located within the path layer overhead, retiming an OTN signal (e.g., due to switching the signal into a different network clock domain) requires demapping back to the client signal.

Table 8-1. Justification Control and Opportunity Definitions for CBR Client Mappings

JC [78]	Generation by Asynchronous Mapper		Generation by bit-Synchronous Mapper		Interpretation by a Demapper	
NJO	PJO	NJO	PJO	NJO	PJO	
00	justification byte	data byte	justification byte	data byte	justification byte	data byte

---

[9] For the SDH mappings, the OTN signal clock tolerance is ±20ppm and the SDH client signal clock tolerance is ±4.6 ppm. Constant use of the NJO increases the OPUk payload rate by 8/(380848) = 65ppm. Constant use of the PJO decreases the OPUk payload rate by 8/(380848) = 65ppm.

Table 8-1. Justification Control and Opportunity Definitions for CBR Client Mappings (continued)

JC [78]	Generation by Asynchronous Mapper		Generation by bit-Synchronous Mapper	Interpretation by a Demapper	
01	data byte	data byte	not generated	data byte	data byte
10	not generated			justification byte	data byte
11	justification byte	justification byte		justification byte	justification byte
NOTE – Since the mapper never generates the JC [78] = 10, the interpretation by the demapper is based on the assumption that an error has corrupted these bits.					

Note: See Appendix C - Fine Grain (sub-Gbit/s) OTN (fgOTN) for a discussion of sub-ODU1 rate CBR client signals.

8.1.4. Generic Mapping Procedure (GMP)

As explained above, since AMP relies on a small number of bytes (NJO and PJO) for rate adaptation, adding new CBR clients would require a unique set of Fixed Stuff bytes for each client in order to reduce the OPU payload capacity to within the range that AMP can accommodate. The adoption of ODU0 and ODU4 created the opportunity to define both signals with a more flexible mapping method to accommodate new CBR client signals having arbitrary rates and frequency tolerances.

The resulting GMP achieves mapping in a straightforward manner without client-specific accommodations such as unique sets of fixed stuff bytes. It is used for all asynchronous CBR client mappings into ODU0 and ODU4, and for most non-SDH CBR client mappings into ODU1, ODU2 and ODU3.

The concept behind GMP mapping is that the JC bytes of each frame are used to communicate the number of payload words that will be mapped into the OPUk payload area during the next OTN frame. As explained below, the transmitter and receiver use modulo arithmetic based on this count value to determine the location of payload and stuff words within the payload area of the frame. The flexibility of GMP comes from it being able to accommodate a CBR client with a rate that needs between one and 15232 bytes per OPU frame. It also inherently allows great flexibility in accommodating different client frequency clock tolerances, although in practice most CBR client signal clocks are no worse than ±100ppm. Descriptions of the GMP processes for mapping and multiplexing are provided below, including specific examples. Details of the GMP approach, including the associated mathematical concepts and derivations, can be found in Annex D of G.709.

For simplicity, the GMP method will first be described for the mapping case. The distinctions associated with GMP multiplexing are described in the next part of this section. The section concludes with a description of an additional fine-grained phase/frequency information capability that is required with many GMP applications.

8.1.5. GMP for Mapping

The GMP justification method works as follows for mapping. A count value, referred to as C_m,[10] is sent in the JC1-JC3 octets of frame *i* to indicate the number of client signal payload words that will be transmitted in the OPUk payload area during frame *i*+1. The stuff words are distributed throughout the OPU payload container in a manner that the receiver can derive directly from the received count value. As described in the Fine-Grain Phase/Frequency Information with GMP section, GMP also supports fine-grain phase (sub-word) justification resolution.

---

[10] The terminology "C_m" indicates that the data and stuff word size is *m*-bits, and hence the corresponding count increment is *m*-bits. Specifically, $m = 8 \times M$. For mapping, M is the maximum number of TS supported by OPUk and for multiplexing, M is the number of 1.25 Gbit/s TS used by that client.

As data rates increased, there were significant advantages to using wider data paths at lower clock rates within integrated circuits. In order to better facilitate wider data paths, an M-byte word size is used with GMP. For the purposes of mapping, the word size is equal to the maximum number of TS that could be supported by that OPUk in a multiplexed payload. Specifically, ODU0, ODU1, ODU2, ODU3 and ODU4 have word sizes of M = 1, 2, 8, 32 and 80 bytes, respectively. As explained below, for multiplexing the value of M is equal to the number of TS occupied by that client.

The words within an OPUk frame are numbered beginning with word #1 occupying the first M bytes of the first row of the payload area (i.e., row 1, column 17 through column 16+M). Word #2 occupies the next M bytes of row 1, etc through the rest of the OPUk frame payload area. The numbering is illustrated in Figure 8-2 and Figure 8-3 for the ODU0 and ODU1. For ODUk, k= 0-3, the 15232-byte OPU payload area contains an integer number of M-byte words per OPUk frame row (i.e., 3808/M is an integer) and the words within the OPUk frame are numbered 1 through 15232/M. In the case of OPU4, columns 17-3816 carry payload and columns 3817 to 3824 contain fixed stuff[11] (refer to Figure 8-9). Dividing the resulting 3800 payload bytes per OPU4 by M gives 3800/80 = 47.5. Consequently, each row pair contains an integer number of words. Specifically, the end of each odd numbered row contains the first 40 bytes of a word, and each even numbered row begins with the remaining 40 bytes of that word. The words within the 15200-byte OPU4 frame payload area are numbered 1 through 190. See the client mapping example in Figure 8-6 below for an illustration of how the 190 words are arranged with the OPU4 frame.

Note that the byte location designated as NJO for the AMP mappings (refer to the example in Figure 8-10) is not defined for GMP and never carries data. Similarly, the byte location(s) designated as PJO for the AMP mappings is regarded as part of the payload container for GMP.

The method for determining the data and stuff word locations is based on modulo arithmetic. In modulo arithmetic, the modulo remainder of X divided by Y, which is expressed as $(X)\text{mod}Y$, is the integer remainder of X when it is divided by Y. For example, (49)mod13 = 10, since 49 = (3)(13) + 10. In general, P_{server} is the maximum number of payload words in the server frame payload area. Since the entire OPUk is occupied for mapping, P_{server}, is the maximum number of m-bit words in the OPUk for mapping applications. Consequently, the word locations within the OPUk payload area are numbered from 1 to P_{server}, as illustrated in Figure 8-2.)

Let n be the payload word location number and let C_m be the count of the number of m-bit words to be transmitted in the next frame Then, the integer portion of C_m is the average number of payload words per frame is determined by the ratio of the encoded client signal rate to the payload container rate:

$C_{m\ average}$ = integer [(P_{server})(client stream rate / OPUk payload container rate)]

---

[11] A “fixed stuff” byte is dedicated to carrying dummy stuffing rather than client data in a given application.

Figure 8-2. OPU0 Payload Area Word Numbering Example

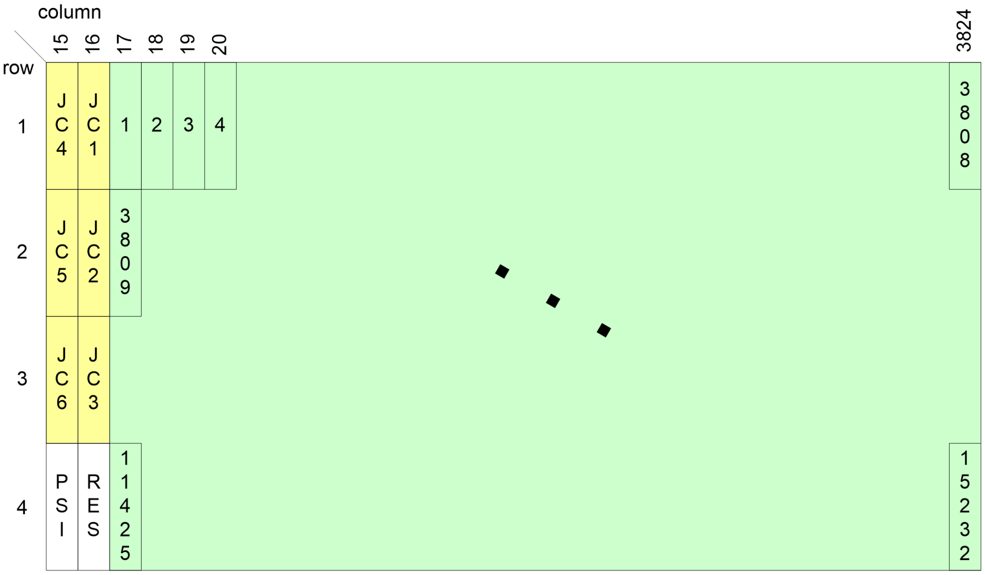
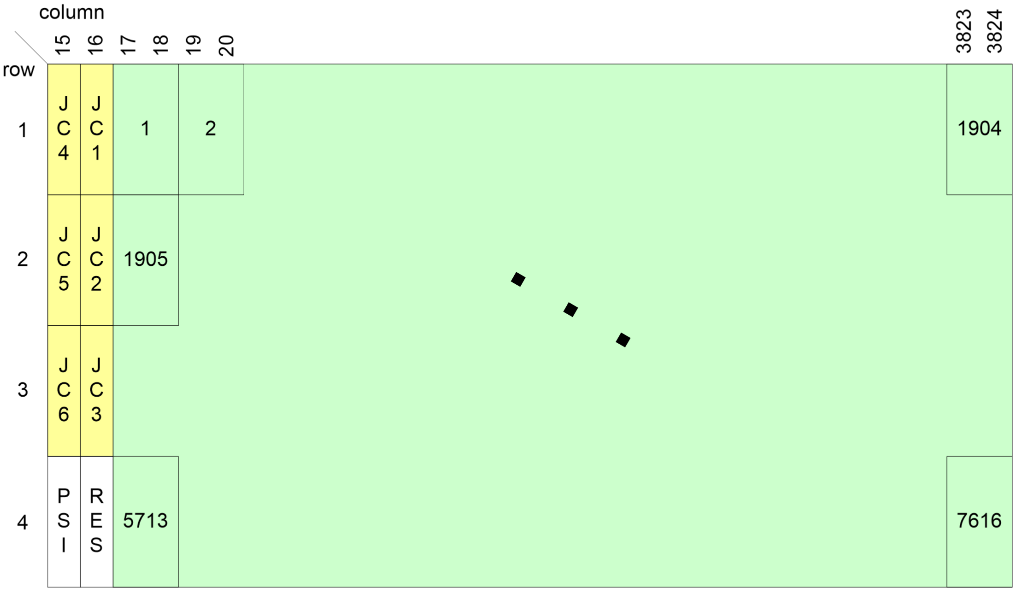


Figure 8-3. OPU1 Payload Area Word Numbering Example



The contents of word *n* in frame *i*+1 is determined by:

$$\text{Word } n = \begin{cases} \text{data} & \text{for: } (n \times C_m) \bmod P_{server} < C_m \\ \text{stuff} & \text{for: } (n \times C_m) \bmod P_{server} \geq C_m \end{cases}$$

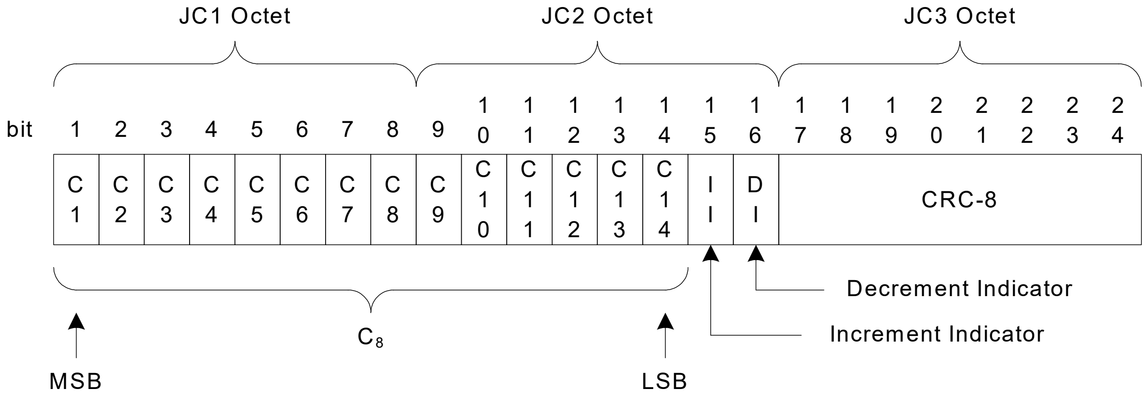
$P_{server} = 15232/M$ for k = 0-3 and $P_{server} = 15200/M$ for k = 4

The result is that beginning with the first word location of the OPU frame, the stuff words are evenly distributed throughout the frame among the payload words.

Another way to describe this mathematical relationship is that each time the payload word location number is incremented by one, the remainder of $(n \times C_m) \bmod P_{server}$ is incremented by an amount equal to the difference between the OPUk word capacity and the number of words to be transmitted that frame (i.e., $P_{server} - C_m$). When this $(n \times (P_{server} - C_m)) \bmod P_{server}$ remainder reaches a value that is less than previous remainder (i.e., less than C_m when the remainder is taken modulo P_{server}) a stuff word is inserted. For example, consider a client that use will use 11424 bytes in the next frame, which is exactly (7/8)(15232). The remainder value increments by 1904 for each increase in the value of *n*, which results in a remainder of zero for every eighth octet. This type of modulo count is easily implemented in hardware.

The OPUk justification control (JC) octet format is illustrated in the figure below. The average value of C_m will rarely be an integer. Consequently, C_m must occasionally be adjusted (incremented or decremented) from frame to frame. While the count increment/decrement is typically by ±1, in some cases, the count may need to be incremented by ±2. Consequently, the mechanism accommodates the ±2 range. Since a mismatch between the source and sink C_m value would cause significant data corruption, it is critical to communicate C_m and its adjustments in a very robust manner.

Figure 8-4. GMP justification control overhead



Robust count communication is achieved through two mechanisms.[12] The first is a count increment or decrement indication based on the combination of indicator bits (II and DI) and inverting a subset of the bits in the current C_m field.[13] The second mechanism is a CRC-8 error detection and correction code over the three-octet JC field.[14]

The source signals the sign and magnitude of the adjustment by transmitting the C_m with different subsets of its bits inverted. These bit inversion patterns, shown in Table 8-2, were chosen to have a per-octet Hamming distance of at least four between every pattern. A “1” in the table cell associated with a C*x* column indicates that bit position will be inverted for the operation indicated by the change (Δ) column. The explicit Increment Indicator (II) and Decrement Indicator (DI) bits provide both a convenient way to detect an increment or decrement on an error-free channel, their location was chosen such that the combination of the C_m bits, II and DI in JC2 will directly mirror the inversion patterns in JC1. For example, when an increment signaled, the II bit is set to one and the C_m value of same bit location in JC1 inverted. The JC fields are protected by a CRC-8 error check code. The CRC-8 allows immediate per-frame changes to the C_m of any magnitude by eliminating the need for the persistency checking.

---

[12] The robust GMP JC octet format with ±2 adjustment capability was originally developed and proposed by PMC-Sierra, now part of Microchip Technology.

[13] Note that while the bit inversion portion of the mechanism has some conceptual similarities to the SDH/SONET pointer adjustment mechanism, the GMP JC encoding provides significant improvements with respect to robustness, flexibility and performance.

[14] The CRC-8 generator polynomial, $G(x) = x^8 + x^3 + x^2 + 1$, was chosen such that it could provide an efficient, low-latency implementation with parallel logic.

Table 8-2. C_m bit inversion patterns to indicate increment and decrement

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Δ
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	+1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	-1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	+2
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	-2
Binary Number														1	1	>±2

The CRC-8 is capable of detecting any 8-bit burst error, and hence can protect against the corruption of any single JC octet.[15] [16] Specifically, the combination of using the count value inversion patterns (including the II and DI) and the CRC-8 allows the receiver to correctly interpret the received C_m value in the presence of any error pattern affecting a single JC octet. The combination of the Increment and Decrement Indicators and the CRC also allow communicating an entirely new C_m, in any situation in which it is necessary. This type of change will typically only occur upon initialization, or upon entering or exiting a client signal fault condition.

The GMP sink interpretation of the received GMP JC values in the presence of potential errors can be summarized as follows:

- When the received CRC is good:
  - Accept the received C_m value when II = DI
  - When II ≠ DI, compare the received C_m bit values to the expected C_m bit values in order to determine the received C_m bit inversion pattern. Interpret the bit inversion pattern according to Table 8-2 to determine and perform the correct increment or decrement operation. Note that since the CRC is good, either JC1 or JC2 can be used for bit inversion pattern check.
- When the received CRC is bad, compare the received C_m value to the sink's current C_m value to determine whether JC1 or JC2 has contains a valid bit inversion pattern (i.e., one of the patterns shown in Table 8-2).
  - If both JC1 and JC2 contain the same valid inversion pattern, it indicates an error in the JC3 CRC. Accept and act on the valid inversion pattern and perform the correct increment or decrement operation.
  - If either JC1 contains a valid inversion pattern and JC2 does not, or JC2 contains a valid inversion pattern and JC1 does not, accept and act on the valid inversion pattern and perform the correct increment or decrement operation.
  - If neither the JC1 nor JC2 contain a valid inversion pattern, the sink should keep its current C_m value and begin a synchronization search.
  - If both JC1 and JC2 contain valid but different inversion patterns, it indicates a burst error in JC1 or JC2 that created a mimic of a different valid inversion pattern. The sink can keep its current C_m value and begin a synchronization search. However, the sink can also use the received CRC to determine whether JC1 or JC2 contains the correct pattern, and accept and act on the valid inversion pattern and perform the correct increment or decrement operation[17].

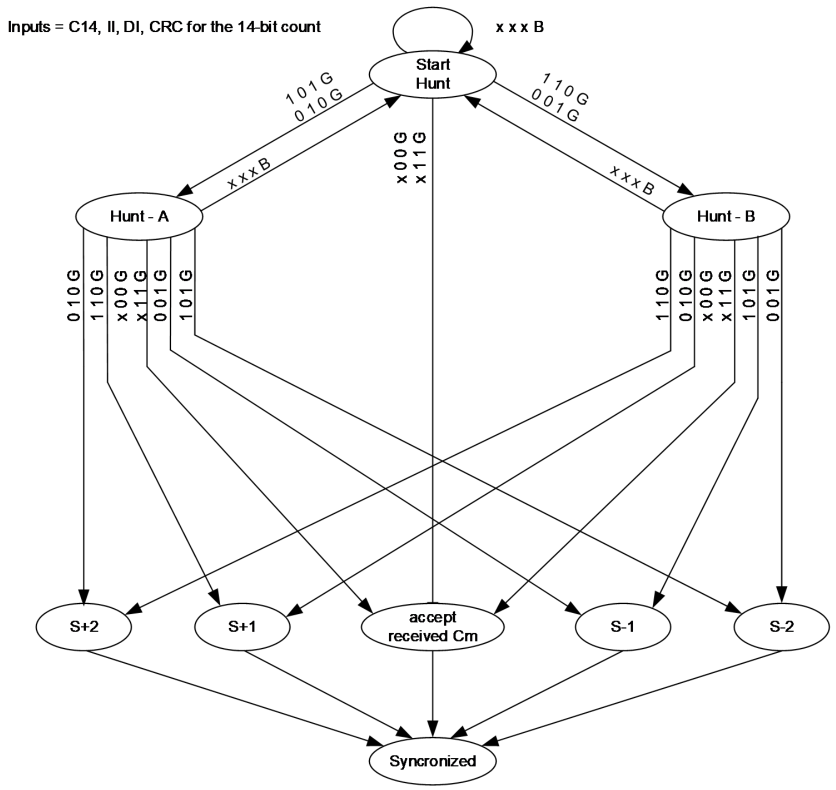
---

[15] Due to the spacing between JC bytes, it is assumed that an error burst will affect no more than a single JC octet per frame.

[16] The CRC-8 polynomial was also chosen to have adequate Hamming distance for single error correction. However, since it is difficult to know a priori which links would be characterized by random errors rather than burst errors, G.709 only specifies using the burst error capability of the JC encoding.

As illustrated in Figure 8-5, initial source-sink C_m synchronization or recovery of from corruption of the sink's expected C_m value can be achieved within two frames, even in the presence of continuous increment and decrement actions.[18] [19] Note that a 10-bit C_m is used for OTN signals for rates beyond 100 Gbit/s (see below). The "G" and "B" state transition inputs in Figure 8-5 represent whether the CRC check was Good or Bad.

Figure 8-5. GMP C_m Sink Synchronization Process Diagram



---

[17] The designer can choose between multiple algorithms to determine the JC byte with the correct pattern.

[18] Achieving synchronization at the receiver requires receiving error free JC octets.

[19] Another criterion for choosing the count value inversion patterns was the requirement to achieve fast synchronization at the receiver with a state machine of minimum complexity. If there is no increment or decrement operation, the receiver can directly accept the received C_m value and achieve synchronization with a single received set of JC octets with a good CRC. While the II and DI indicate whether an increment or decrement is occurring, the receiver cannot immediately determine the magnitude of the change since it does not have a correct base C_m value with which to compare the inversion patterns. However, the increment or decrement operation will change the next transmitter C_m value in a predictable manner. Consequently, there are a small number of valid combinations of the count value LSB (C14 for ODUk, k = 1-4)), II, and DI that could be received in the next frame. As a result, the combination of C14, II and DI in two consecutive frames uniquely identifies the type and magnitude of the increment/decrement operation in the second frame, which allows the receiver to directly determine the transmitter's base C_m value by reversing the inversion pattern.

Table 8-3. State Actions

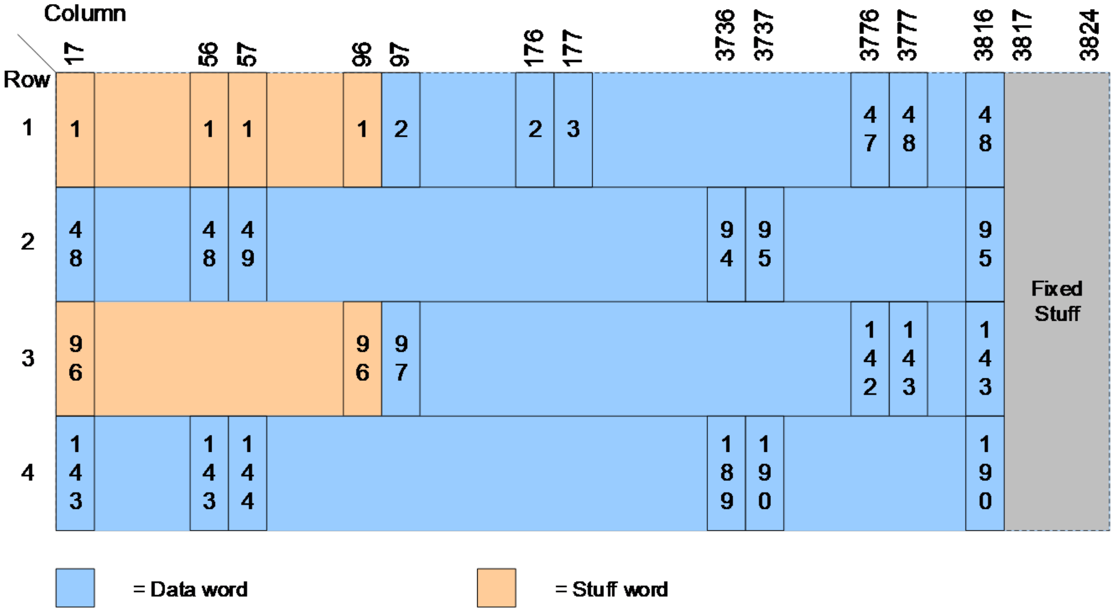
Consequent State Actions for 14-bit C_m			
Increment State	**Action**	**Decrement State**	**Action**
S+2	Count = C1-C14 after inverting C2, C3, C6, C7, C11 and C14; Increment +2 for the next frame	S-1	Count = C1-C14 after inverting C2, C4, C6, C8, C10, C12 and C14; Decrement -1 for next frame
S+1	Count = C1-C14 after inverting C1, C3, C5, C7, C9, C11 and C13; Increment +1 for the next frame	S-2	Count = C1-C14 after inverting C1, C4, C5, C8, C9, C12 and C13; Decrement -2 for next frame

One example of a mapping with GMP is the 100GBASE-R Ethernet client signal. The OPU4 payload capacity was chosen to accommodate this client with its native 64B/66B block coding, which gives a client signal rate of 103.125 Gbit/s. Since the OPU4 frame period is 1.168 μs and GMP mapping word size is 80 bytes for OPU4 (i.e., using C_{640}), at the nominal client and OTN rates the client delivers:

$$Ave.\,client\ words\ per\ OPU4\ frame = \left(\frac{103.125Gbit}{s}\right)\left(\frac{byte}{8\ bits}\right)\left(\frac{word}{80\ bytes}\right)\left(\frac{1.168\ \mu s}{frame}\right)$$
$$= 188.2\ words/frame$$

Consequently, the GMP C_m will vary between 188 and 189 words. As illustrated in Figure 8-6, since 3808 is not evenly divisible by 80, each of the four OPU4 payload rows is padded with 8 fixed stuff bytes so that it contains exactly 47.5 words. This results in each OPU4 frame having the capacity to carry exactly 190 of the 80-byte words. C_m = 188 results in two stuff words per frame.

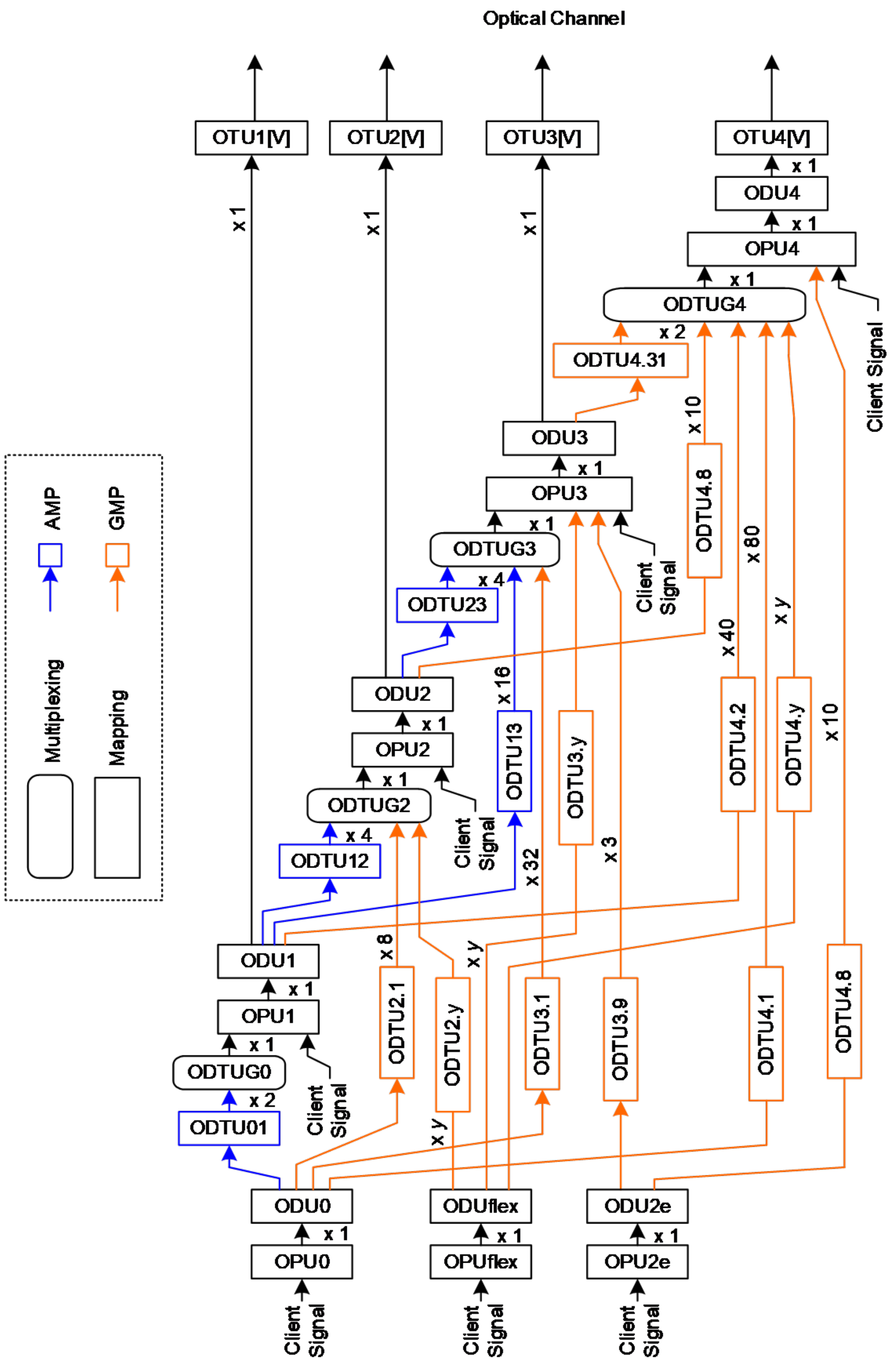
Figure 8-6. 100GBASE Ethernet mapping illustration with GMP C_{640} = 188



8.2. CBR Client Multiplexing

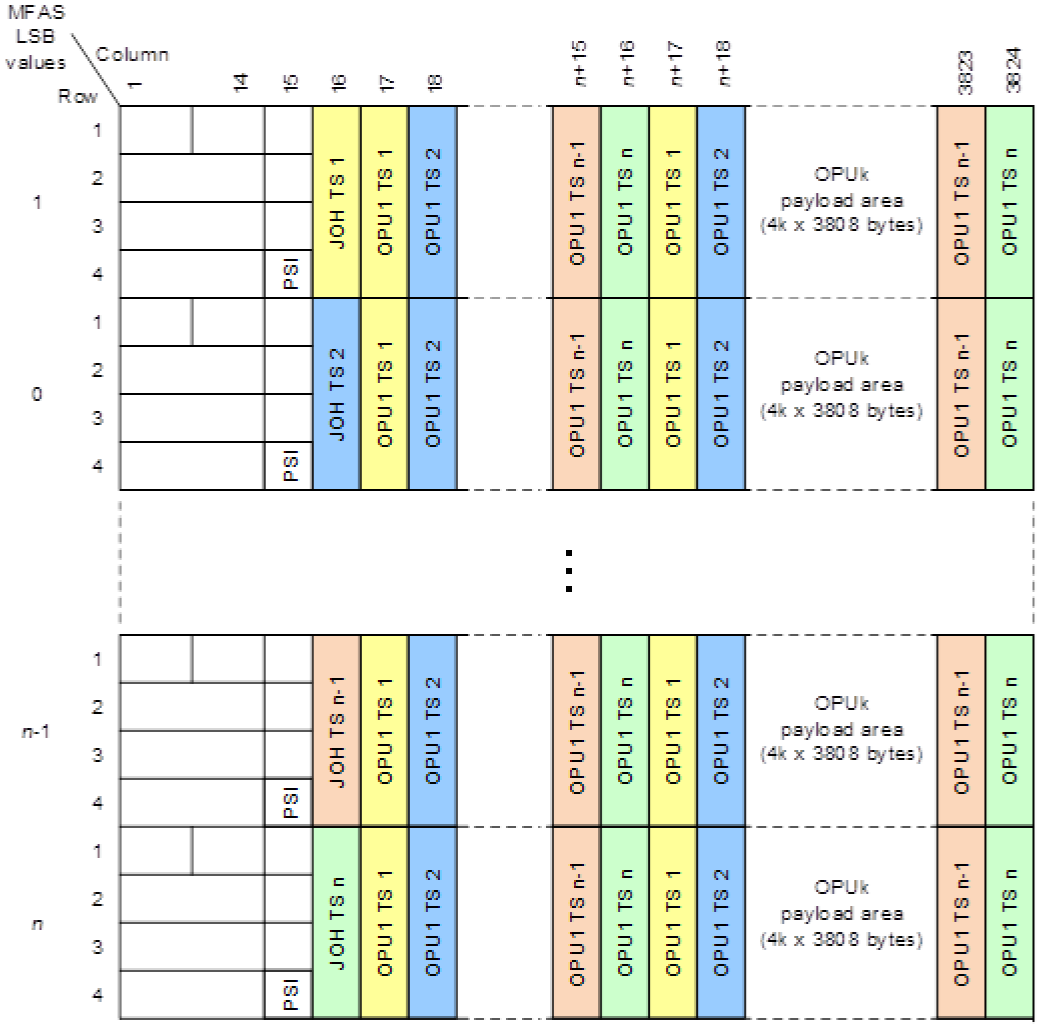
While the initial version of OTN relied on multiplexing in its client domain (e.g., using SONET/SDH), time division multiplexing (TDM) was soon added to OTN to increase the usage efficiency of each optical channel (wavelength). As discussed above, TDM of multiple clients into an OPU is supported by dividing the OPU payload area into Tributary Slots (TS). The OTN TDM hierarchy from ODU0 through ODU4 is illustrated in Figure 8-7. The conceptual ODTU construct used in Figure 8-7 is explained below.

Figure 8-7. OTN Multiplexing Hierarchy Through OTU4



Non-OTN client signals are never multiplexed directly into the TS of an OPUk. Instead, they are always first mapped into their own ODU and these lower rate ODU signals are multiplexed into the higher rate OPUk. The ODU carrying the client signal is assigned to the minimum number of TS in the higher rate ODUk such that the capacity of the set of TS exceeds the rate of the ODU carrying the client signal[20] (1 ≤ #TS ≤ OPUk max). For example, the 25GBASE-R Ethernet rate is 25.78125 Gbit/s. The ODUflex into which is mapped would have a nominal rate of (239/238)×(25.78125G) = 25.88957 Gbit/s. The ODU3 is the lowest rate ODUk that could carry this ODUflex. The OPU3 TS rate is 1.2547 Gbit/s, and (25.88957)/(1.2547) = 20.63, so 21 TS must be used in the higher rate OPUk.

Figure 8-8. Payload and Justification Location Structure for Multiplexing into ODUk, k = 1, 2, 3



The AMP and GMP CBR client mapping methods can also be used for multiplexing. However, since each client signal needs its own justification overhead, as explained next, the JC bytes are time shared on a per-frame basis across a multiframe.

---

[20] With TDM, the ODU into which the client is mapped is sometimes referred to as a “Low Order (LO)” ODU, and the higher rate ODU into which the lower rate ODUs are multiplexed is sometimes referred to as a “High Order (HO) ODU. G.709 moved away from this terminology due to the potential confusion when there is more than one multiplexing stage (e.g., mapping a client into ODU0, multiplexing multiple ODU0 into an ODU2 and multiplexing this ODU2 into an ODU4).

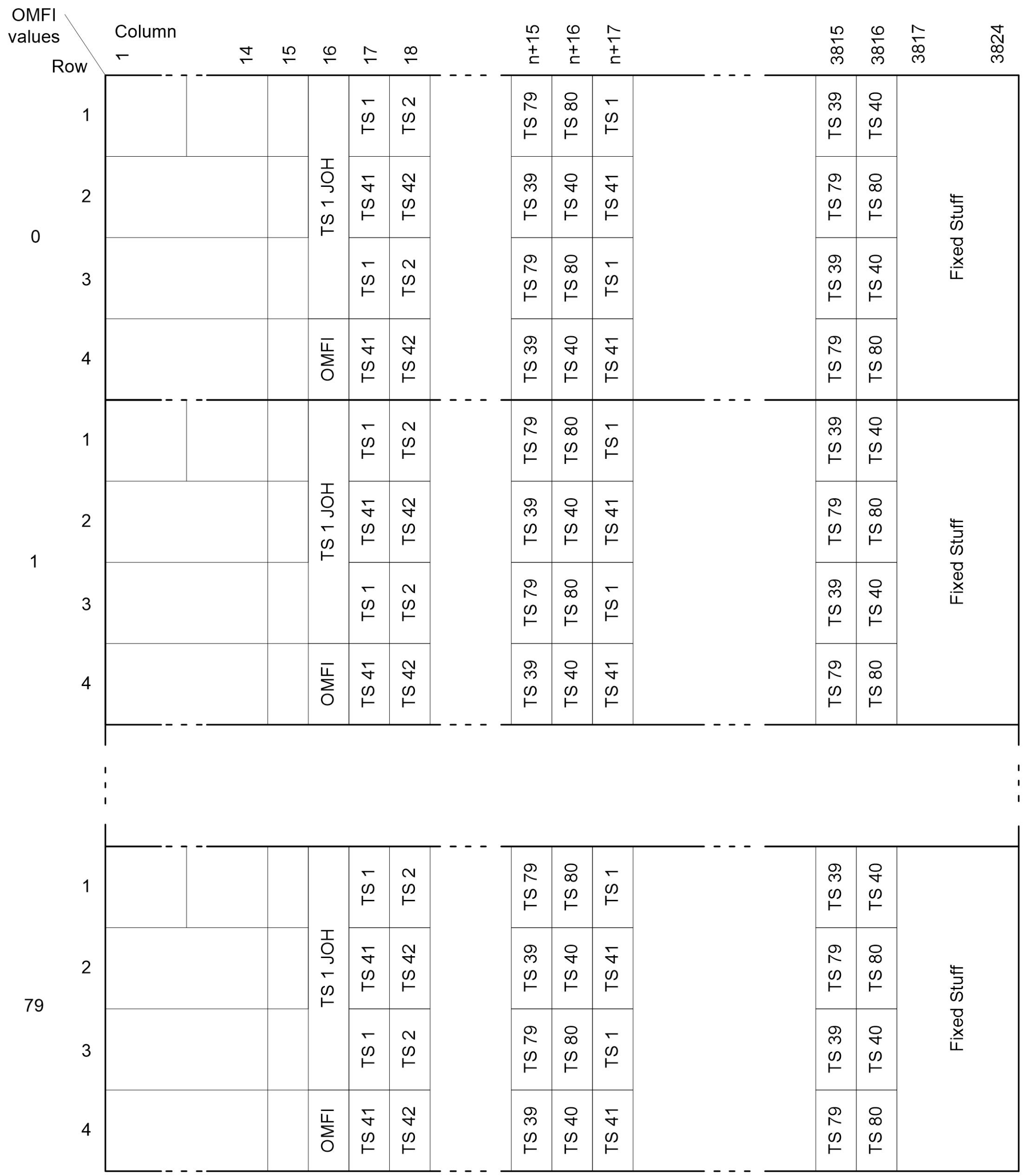
Conceptually, the client signal can be considered as first asynchronously mapped into an intermediate logical construct called an Optical Channel Data Tributary Unit (ODTU)[21]. The overhead of the ODTU is the information required for timing justification. In other words, the ODTU structure contains both the client data in its payload area and the JC byte values associated with the mapping. The ODTU structures are then byte-synchronously multiplexed (byte-interleaved) into their respective TS of the higher rate OPUk with their justification control overhead appearing in the JC bytes of their associated frame within the multiframe.[22] Specifically, the JC bytes of each frame within the multiframe are associated with one of the TS within the OPUk (e.g., the frame with MFAS = 0000 0000 (frame #1) contains the JC bytes associated with TS #1, etc.). Consequently, while the MFAS provides a 256-frame multiframe for time-sharing general overhead, the multiframe length for the justification overhead is equal to the number of TS that OPUk is capable of supporting. For example, an OPU3 with 2.5G TS uses a 16-frame justification multiframe and an OPU2 with 1.25G TS uses an 8-frame justification multiframe. The OPUk justification multiframe repeats on an interval equal to its length. For the example of an OPU2 with 1.25G TS, the multiframe uses MFAS values xxxx x000 through xxxx x111 (i.e., just uses bits 6-8 of the MFAS). Since OPU4 supports 80 TS and 80 is not a power of 2, OPU4 uses a separate OPU Multiframe Identifier (OMFI) that with a modulo count from 0 – 79 to indicate frames 1 – 80. Figure 8-8 illustrates the OPUk structure and multiframe for multiplexing for ODU1, ODU2 and ODU3, and Figure 8-9 illustrates the ODU4 multiplexing frame structure. Refer to the example in Figure 8-13.

---

[21] The ODTU is a useful concept for describing the information associated with the lower rate ODU signal that is multiplexed into the higher rate ODUk. It does not represent an actual signal, nor is it a required intermediate entity for creating the higher rate OPUk.

[22] Note that Clause 7 of G.709 refers to the resulting combined OPUk data and JC fields as an ODTU Group (ODTUGk). However, the ODTUGk is purely a logical construct, since the ODTU constructs are directly time-division multiplexed into the tributary slots of an OPUk. Consequently, the ODTUG term is not used in subsequent clauses of G.709.

Figure 8-9. Payload and Justification Location Structure for Multiplexing into ODU4



As illustrated above, the Payload Structure Indicator (PSI) byte carries the Multiplex Structure Indicator (MSI) field. The MSI carries the information regarding which client signal is assigned to which TS. See Table 8-4 and Table 8-5 for illustrations of the MSI format. As explained below, the ODTUjk names (e.g., ODTU13) are associated with AMP and 2.5G TS, and the ODTUk.TS names (e.g., ODTU2.ts) are associated with GMP and 1.25G TS. While it is possible to use a fixed mapping between tributary ports and tributary slots, the MSI can thus be used to increase the flexibility of the assignments. A different payload type indicator is used when the OPUk is structured for 1.25 Gbit/s (ODU0-capable) tributary slots.

Table 8-4. MSI Illustration for OPU1 – OPU3

MSI Field Illustration				ODTU Type Codes		
Bits 1 2	Bits 3 4 5 6 7 8			**OPUk**	**PT**	**ODTU Type**
				k = 1	20	ODTU01 = 11
ODTU type	Tributary Port # (1 – n)	TS # (1-m)		k = 2	20	ODTU12 = 00
		1			21	ODTU12 = 00 ODTU2.ts = 10
		2		k = 3	20	ODTU13 = 00 ODTU23 = 01
		…				
		m			21	ODTU13 = 00 ODTU23 = 01 ODTU3.ts = 10
m = max. TS for the OPUk n ≤ m						

Table 8-5. MSI Illustration for OPU4

Bits 1	Bits 2 3 4 5 6 7 8	
Occupied	Tributary Port # (1 – n)	TS # (1-80)
		1
		2
		…
		80

8.2.1. AMP for Multiplexing

In the case where AMP is used for TDM multiplexing into 2.5G TS[23] of an OPUk, the JOH structure is modified from the non-multiplexed case. As explained above, each ODU tributary that is being multiplexed into the OPU requires its own justification and associated justification overhead, which is achieved by time-sharing the JC bytes across a justification mutiframe (as illustrated in Figures 8-8 and 8-9). Figure 8-10 illustrates the format of this OPUk JOH sharing for AMP multiplexing. In order to provide the appropriate frequency range accommodation, two PJO bytes, PJO1 and PJO2, were defined. The PJO bytes, of course, need to appear in the column associated with the other data bytes for that tributary, which results in the structure shown in Figure 8-10. The JOH use and interpretation with TDM multiplexing are given in Table 8-6.

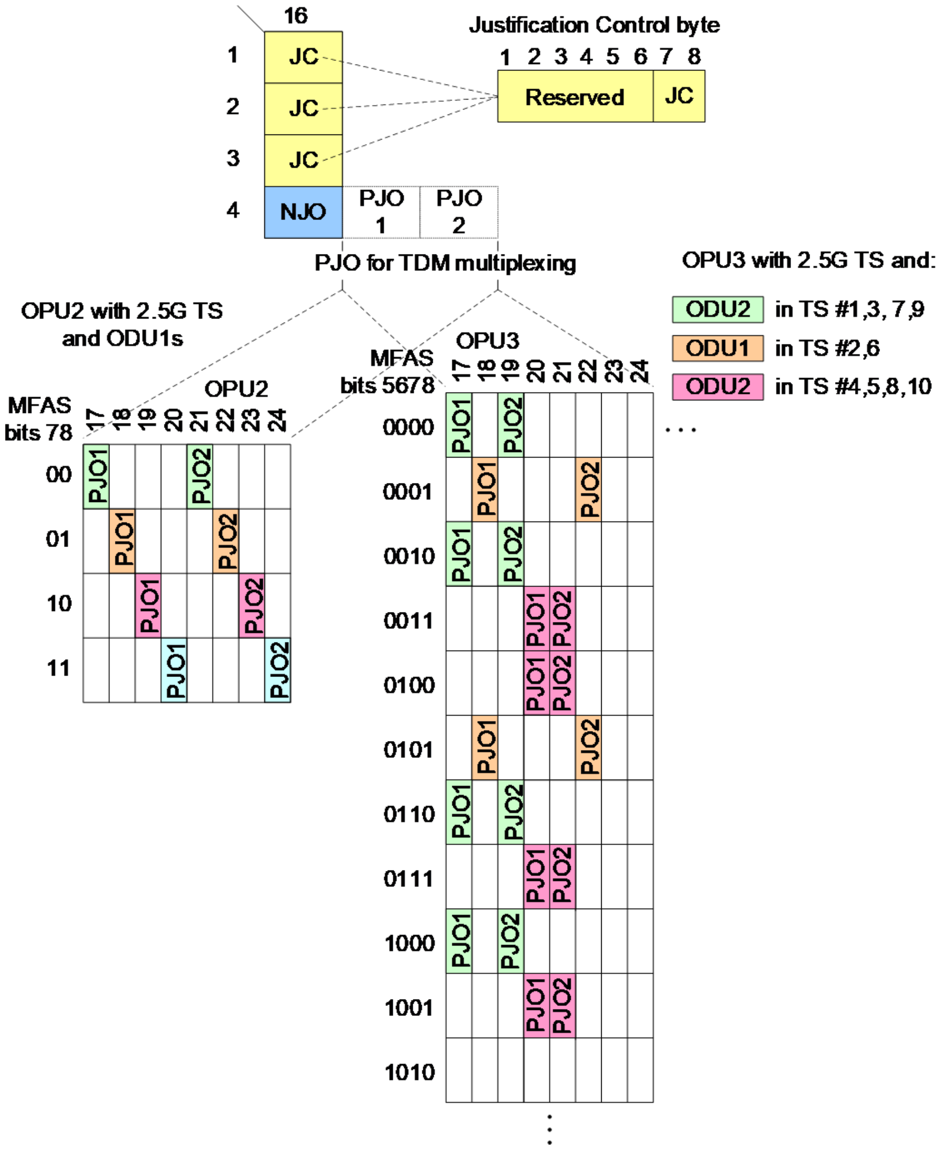
Table 8-6. Justification Control and Opportunity Definitions for TDM Multiplexing

JC [78]	NJO	PJO1	PJO2	Interpretation by the demapper
00	justification byte	data byte	data byte	no justification (0)
01	data byte	data byte	data byte	negative justification (-1)
10	justification byte	justification byte	justification byte	double positive justification (+2)
11	justification byte	justification byte	data byte	positive justification (+1)

---

[23] For multiplexing, AMP is only used with 2.5G TS and is not applicable to 1.25G TS.

Figure 8-10. AMP Justification Overhead Illustration for Multiplexing



Note: Because the justification control is located within the path layer overhead, retiming an OTN signal requires demultiplexing back to the client signal.

8.2.2. GMP for Multiplexing

GMP is specified as the justification method whenever 1.25G TS are used.[24]The GMP method used for multiplexing is similar to the method used for mapping, although with some important differences that are described below.

As with AMP, lower rate ODU clients that are multiplexed into a higher rate OPUk are first (conceptually) mapped into an ODTU that must time share the justification fields in the OPUk overhead. The ODTU for a lower rate ODU mapped with GMP is referred to as an ODTUk.ts, where 'ts' is the number of TS occupied by that ODU in the higher rate OPUk.

---

[24] As discussed below in sections B100G Client Signal Mapping, Multiplexing and Rate Adaptation and GMP Mapping, GMP is also specified for CBR signal multiplexing into OTN signals using the beyond 100Gbit/s structure and FlexO.

The ODTU concept associated with GMP, including the active JC byte location, is illustrated in Figure 8-11. Since there can be an arbitrary spacing between the different TS used by the client, it was simpler to have the GMP count refer to the amount of client data (i.e., number of client data words) sent in the next multiframe rather than the amount of client data in the frames between JOH opportunities for that client. Consequently, unlike AMP where the JC bytes are active in each frame in which they pertain to that client, the GMP JC bytes are only required once per multiframe. Specifically, a client's active GMP JC bytes are located in the ODUk frame associated with the highest number TS used by that client ODU signal. Consequently, for the example of Figure 8-11, since the TS #7 is highest TS number used by that client, the GMP overhead is located in the JC bytes of the 7th frame (MFAS = xxxx x110) of the justification multiframe.

The GMP word size for ODU1-ODU4 is M bytes, where M is the number of TS occupied by that client.[25] Since a client signal using M of the N possible T.S. of an OPU will use M-byte words, for ODU1-ODU3 we have:

(15232 bytes/frame)*(N frames/multiframe)(M/N Trib. Slots / client) / (M bytes/word)

= 15232 words / multiframe[26]

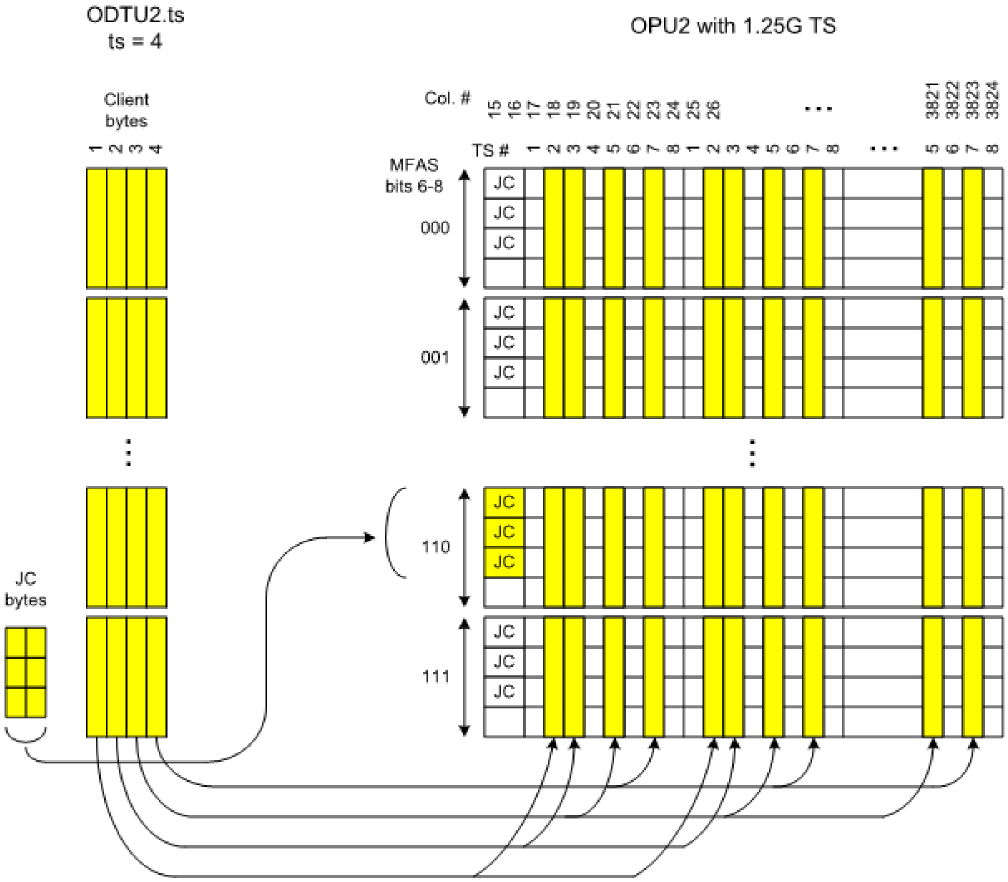
Consequently, even though the GMP count value pertains to a multiframe rather than a frame, it can still use exactly the same GMP count encoding as for the mapping case. Figure 8-12 illustrates the word numbering and locations for the example of an ODUflex using five TS that is multiplexed into an OPU3.

---

[25] This is consistent with GMP mapping where the word size corresponds to using all of the OPUk TS.

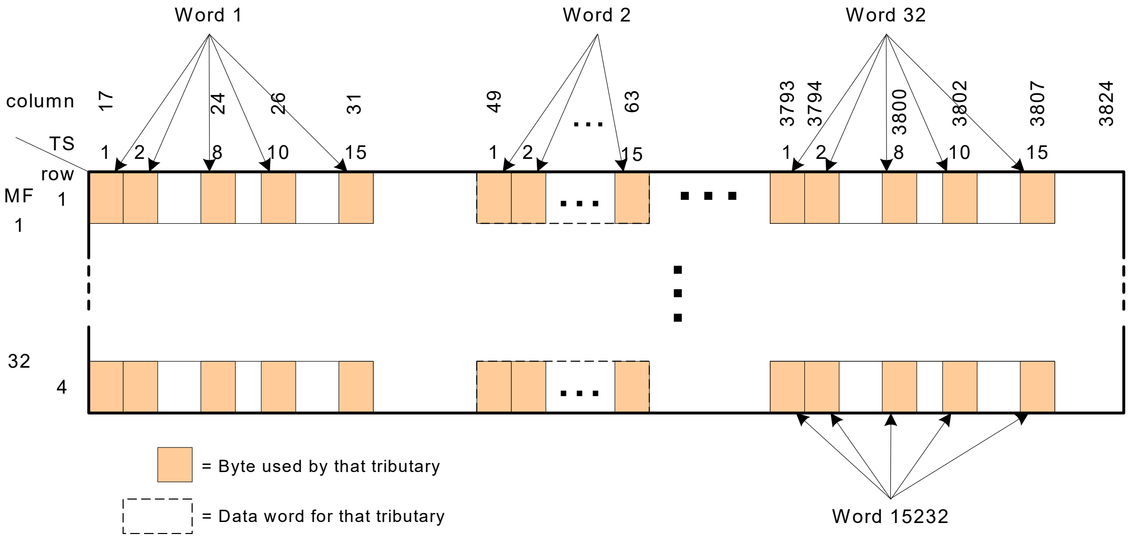
[26] The OPU4 has 15200 bytes rather than 15232.

Figure 8-11. Illustration of the Conceptual ODTU for ODTU2.ts Where TS # 2, 3, 5 and 7 of the OPU2 are Occupied by the Client



Since all ODUflex types, including ODUflex(GFP) and ODUflex(IMP), are CBR signals, the GMP multiplexing works the same way for all ODUflex. Since ODUflex(GFP) and ODUflex(IMP) carry packet rather than CBR clients, they require special considerations regarding choosing the ODUflex rates and justification method for mapping the clients into the OPUflex (see the Clock Generation Methods for ODUflex(GFP) and ODUflex(IMP) section.)

Figure 8-12. Word Numbering Illustration for GMP Multiplexing – ODUflex Tributary using five TS (#1, 2, 8, 10, 15) into the 32-TS structure of an OPU3



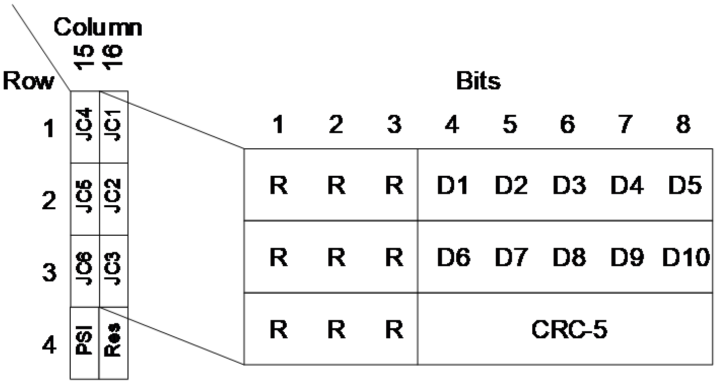
8.2.2.1. Fine-Grain Phase/Frequency Information with GMP

The timing resolution when mapping a signal into OTN is 8 bits, and the timing resolution when multiplexing with GMP is M × 8 bits. It is difficult to satisfy the jitter and wander requirements of some client signals if they are mapped with just byte-level resolution. Examples include mapping SDH/SONET signals into an ODU0 or potentially when carrying a Synchronous Ethernet (SyncE) client mapped into an ODUflex. The M-byte word resolution with GMP multiplexing also potentially increases the jitter and wander challenges when desynchronizing a client signal. For these reasons, an additional capability was added with GMP to provide the receiver with finer resolution phase/ frequency information.

The additional timing information is carried in the J4-J6 bytes, which are located in the first three rows of column 15. The encoding of J4-J6 is illustrated in the Figure below.

A phase offset encoding approach was chosen for C_{nD} rather than a frequency offset encoding approach. A frequency offset approach would have encoded the difference between the number of bits to be transmitted in the next frame (i.e., [word size] × [C_m]) and the average number of bits the client signal delivers during a frame period. As explained below, the phase encoding is essentially a running sum of the number of bits that can’t be transmitted in the next frame. Although both approaches are mathematically equivalent the phase encoding approach is more robust to transmission errors in C_{nD}. A transmission error with the frequency offset approach would result in a frequency error at the receiver that could persist for some time. Transmission errors with the phase offset approach are more readily filtered and at worst result in a transient phase error. Hence, the phase offset approach is more robust for preventing short-term frequency deviations due to transmission errors.

Figure 8-13. GMP Justification Control Overhead for Increased Timing Resolution in JC4-JC6 of Column 15



Since this information can be used for either GMP mapping or multiplexing, the description is written in terms of a mapper with the equivalent multiplexer term in parentheses. The GMP mapper (multiplexer) encoder makes a decision once per frame (multiframe) regarding how many M-byte words of data it will transmit during the next frame (multiframe). There will typically be some number of additional bits or bytes) remaining in the transmitter's buffers that cannot be transmitted since they constitute a fraction of an M-byte word. As illustrated in Figure 8-13, this fractional value is encoded in the D1-D10 field of the J4-J6 bytes as a binary number representing the count of the remainder number of bits (or bytes). This phase count value is referred to as the C_{nD}, which means the Count of the Difference in the number of bits (or bytes) that could be transmitted as a whole byte (M-byte word) and the bits (or bytes) remaining untransmitted at the mapper (multiplexer). Since the C_{nD} remainder accumulates until it reaches or exceeds the GMP word size, it referred to as the $\sum C_{nD}$. The $\sum C_{nD}$ value essentially represents the running phase difference between the received client signal and the client data being transmitted over the server channel. Note that in practice, the transmitter can use an estimated filtered C_{nD} value rather than a strict measurement of the ingress buffer fill remainder. This filtering removes the effects of incoming jitter and wander.

At the GMP receiver, the $\sum C_{nD}$ information helps desynchronizer PLL control by providing a more accurate picture of the client frequency. Specifically, when the received GMP M-byte word count value (J1-J3) is unchanged for several frame (multiframes), the rate of change of the $\sum C_{nD}$ value indicates the frequency offset between the client signal and the OTN channel. Another way to view the $\sum C_{nD}$ is that it provides the receiver with an accurate indication to anticipate upcoming changes in the GMP M-byte word count value, and hence avoids M-byte word sized steps in the PLL control.

The concept of finer-grained frequency/phase communication can be illustrated with the following GMP mapper example where we assume no jitter. Consider a client signal that delivers (Z bytes + Y bits)/frame to the GMP mapper per ODU frame, with $Y < 8$. The mapper will initially send Z bytes of data in the next frame and communicate the Y value in the C_{nD}. This remainder number of bits will accumulate at the mapper over successive frames until the mapper has Z+1 bytes to send and increases its GMP count value accordingly for the next frame. As a specific example, let Z=15100 bytes and Y=5 bits. If we begin at a point where the previous $\sum C_{nD}$ =0, the transmitter will send the following sequence of GMP count (C_m) and C_{nD} values: C_m=Z and $\sum C_{nD}$ =5; C_m=Z+1 and $\sum C_{nD}$ =2; C_m=Z and $\sum C_{nD}$ =7; C_m=Z+1 and $\sum C_{nD}$ =4; C_m=Z+1 and $\sum C_{nD}$ =1; C_m=Z and $\sum C_{nD}$ =6; C_m=Z+1 and $\sum C_{nD}$ =3; C_m=Z+1 bytes and $\sum C_{nD}$ =0; etc. The receiver can anticipate the jump from Z to Z+1 or Z+1 to Z bytes by observing the $\sum C_{nD}$ count changes and can hence smooth the transition in its demapper phase-locked loop (PLL) filter. Consequently, the jitter and wander of the demapped signal can be significantly reduced with a simpler desynchronizer than would otherwise be required.

8.2.3. OTN CBR Client Multiplexing Summary

The CBR client multiplexing types and applications are summarized in Table 8-7.

Table 8-7. Summary of OTN CBR Multiplexing Methods

Multiplexing Method	Application
Asynchronous Mapping Procedure (AMP)	• Multiplexing LO ODU signals into 2.5 Gbit/s tributary slots • Multiplexing some LO ODU signals into 1.25G tributary slots (ODU0 into ODU1, ODU1 into ODU2, ODU1 into ODU3, and ODU2 into ODU3)
Generic Mapping Procedure (GMP)	• Multiplexing LO ODU signals into 1.25 Gbit/s tributary slots (except ODU0 into ODU1, ODU1 into ODU2, ODU1 into ODU3, and ODU2 into ODU3)
Note 1 – The 2.5 Gbit/s and 1.25 Gbit/s tributary slot rates are approximate rate values used for convenience of notation. The actual size of the tributary slot is slightly different for each HO ODU signal rate (refer to the discussion in CBR Client Mapping). Note 2 – Since GMP only supports positive frequency justification, AMP is required when the relative bandwidth of the LO ODU client and the HO OPU TS requires the ability to perform negative frequency justification. The cases for which this is true are elaborated above in this table.	

8.3. Packet-Oriented Client Mapping

Packet-oriented clients are also mapped into a constant rate[27] ODUflex. At a high level, both the rate justification methods associated with carrying packet clients rely on adding padding elements to the encoded client information stream at the source that the sink can recognize and remove rather than relying on the OPUk JC bytes for the dynamic rate justification.

Since packet clients are first mapped into an ODUflex, the associated ODUflex is multiplexed into the server OPUk in same manner as an ODUflex(CBR). Consequently, ODU multiplexing and switching nodes within the network can remain agnostic to the type of ODUflex they multiplex or switch.

Packet Client Maintenance (Replacement) Signal

Packet clients use Ethernet for their Layer 2 encapsulation. Consequently, when a fault condition impacts the client signal, it is replaced by the appropriate fault indication signal for that packet client type.

8.3.1. GFP Frame Mapping

The OTN mapping for packet-oriented clients with rates <100 Gbit/s is to use the packet encapsulation method defined in ITU-T G.7041 Generic Framing Procedure (GFP). GFP encapsulation is defined for a variety of client packet streams, including Ethernet, MPLS and video signals. Each client packet arriving over a data interface is encapsulated into the payload bytes of a GFP frame. See Figure 8-14 for the example of GFP encapsulation of an Ethernet packet.

GFP frames are mapped in a byte-aligned manner into the OPUk. As shown in Figure 8-15, the space (bandwidth) between GFP client data frames in the OPUk payload area is filled by GFP Idle frames.[28] In other words, GFP Idle frames are used for the rate justification when mapping into an OPUk. GFP frame delineation is achieved using the GFP overhead, as defined in G.7041.

---

[27] As described in the Hitless Adjustment of ODUflex(GFP) (HAO) section, the HAO mechanism allows changing the ODUflex(GFP) rate, but even during the resizing the ODUflex rate is incrementally constant.

[28] OTN originally supported a similar mapping for streams of 53-byte Asynchronous Transfer Mode (ATM) cells, which could be used to segment and transport both packet and CBR clients. As ATM technology decreased in importance, the ATM mapping was removed from G.709.

Figure 8-14. GFP Example for Ethernet Frame Encapsulation

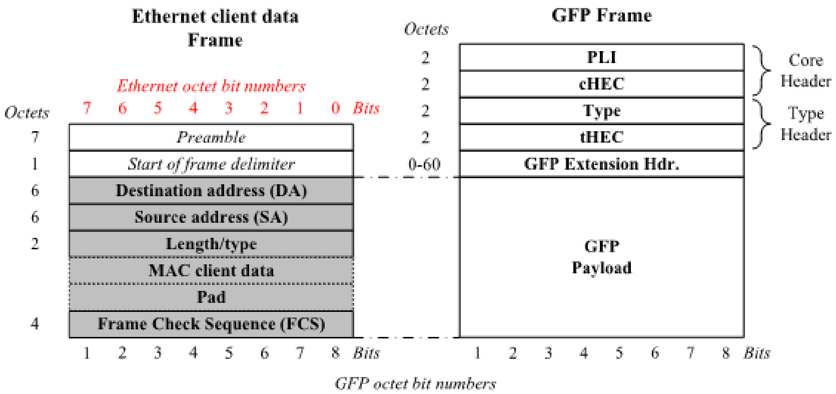
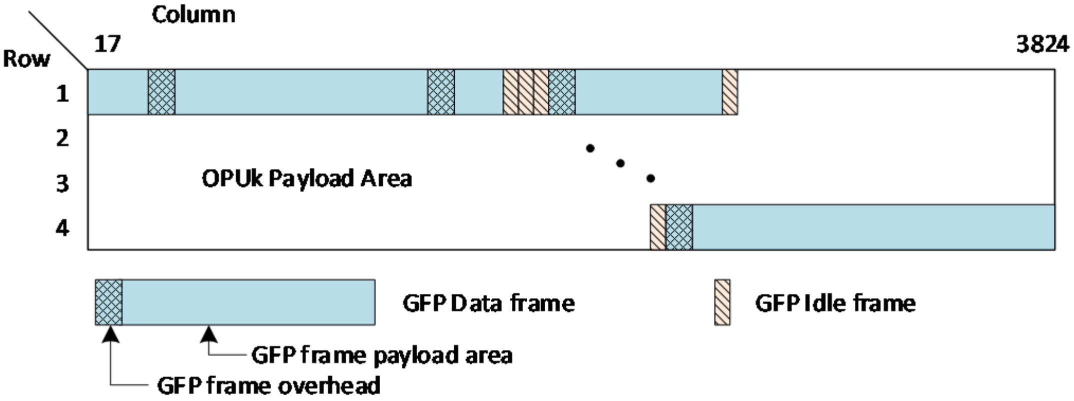


Figure 8-15. GFP frame mapping into an OPUk



8.3.1.1. ODUflex(GFP)

The ODUflex(GFP) is a CBR signal with a frame rate generated from a local source clock. See the discussion below in Clock Generation Methods for ODUflex(GFP) and ODUflex(IMP) regarding the ODUflex(GFP) rate generation approaches. As the name implies, GFP Idle frames provide the mechanism for rate adapting the client packet stream into the OPUflex(GFP) payload area.

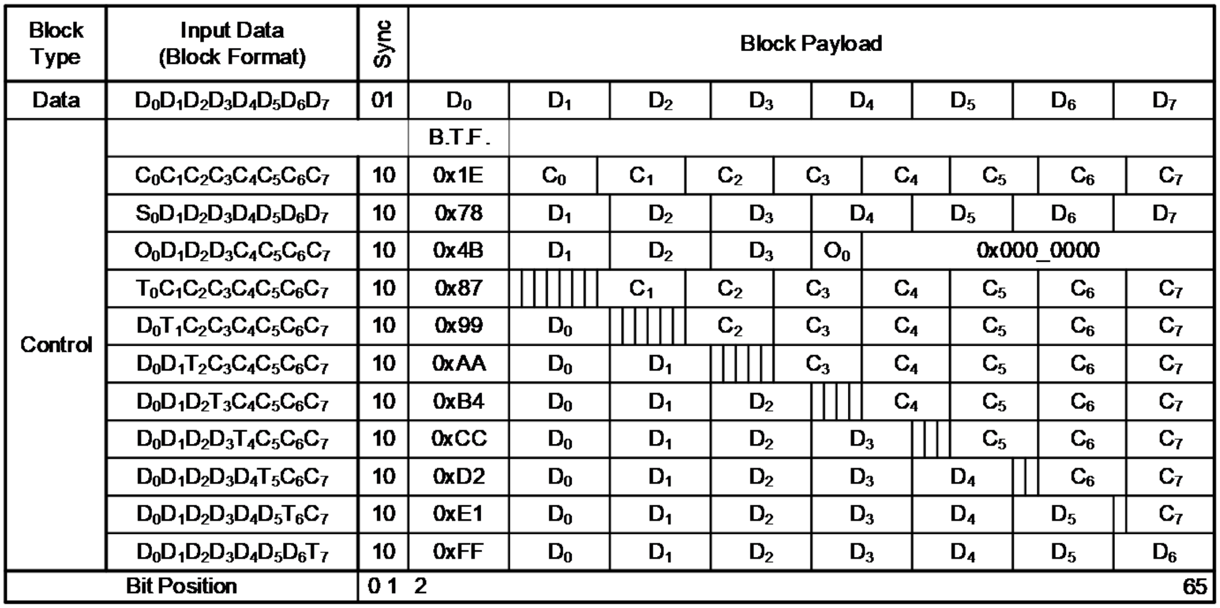
One of the valuable applications for ODUflex(GFP) is that it allows using ODU switching as an economical way to provide pseudo-wire (PW) or Ethernet virtual LAN (VLAN) switching within the OTN domain, thus avoiding the equipment and network complexity of using packet switching.

As described in above, a mechanism referred to as HAO (Hitless rate Adjustment of ODUflex(GFP)) was subsequently defined in ITU-T G.7044 to allow hitless changes to the ODUflex(GFP) rate, and hence its capacity, such that it occupies more or fewer TS.

8.3.2. Idle Mapping Procedure (IMP) into ODUflex(IMP)

As background, when higher rate Ethernet interfaces were defined in IEEE 802.3, it was typical to define them in terms of a stream of 64B/66B blocks. Each 64B/66B block either contained only data from an Ethernet packet or included some type of control information. The blocks with control information include start of packet (/S/), termination of packet (/T/), Ordered sets (/O/) to communicate client status, and idles (/I/) to fill the inter-packet gap (IPG). See the Figure below for an illustration of the 64B/66B block encoding as defined in IEEE 802.3 clause 82.

Figure 8-16. 64B/66B Block Coding Defined in IEEE 802.3 Clause 82



Block Type	Input Data (Block Format)	Sync	Block Payload								
Data	$D_0D_1D_2D_3D_4D_5D_6D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
Control			B.T.F.								
	$C_0C_1C_2C_3C_4C_5C_6C_7$	10	0x1E	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
	$S_0D_1D_2D_3D_4D_5D_6D_7$	10	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
	$O_0D_1D_2D_3C_4C_5C_6C_7$	10	0x4B	D_1	D_2	D_3	O_0	0x000_0000			
	$T_0C_1C_2C_3C_4C_5C_6C_7$	10	0x87	▨	C_1	C_2	C_3	C_4	C_5	C_6	C_7
	$D_0T_1C_2C_3C_4C_5C_6C_7$	10	0x99	D_0	▨	C_2	C_3	C_4	C_5	C_6	C_7
	$D_0D_1T_2C_3C_4C_5C_6C_7$	10	0xAA	D_0	D_1	▨	C_3	C_4	C_5	C_6	C_7
	$D_0D_1D_2T_3C_4C_5C_6C_7$	10	0xB4	D_0	D_1	D_2	▨	C_4	C_5	C_6	C_7
	$D_0D_1D_2D_3T_4C_5C_6C_7$	10	0xCC	D_0	D_1	D_2	D_3	▨	C_5	C_6	C_7
	$D_0D_1D_2D_3D_4T_5C_6C_7$	10	0xD2	D_0	D_1	D_2	D_3	D_4	▨	C_6	C_7
	$D_0D_1D_2D_3D_4D_5T_6C_7$	10	0xE1	D_0	D_1	D_2	D_3	D_4	D_5	▨	C_7
	$D_0D_1D_2D_3D_4D_5D_6T_7$	10	0xFF	D_0	D_1	D_2	D_3	D_4	D_5	D_6	
Bit Position		0 1	2								65

IEEE 802.3 defines a minimum gap between packets, referred to as the inter-packet gap (IPG), which is filled with idle characters. Idle characters are inserted or removed from the transmitted Ethernet signal in order to adapt the rate of the Ethernet packet information flow to the PHY capacity rate. In the case of an interface using 64B/66G encoding, idle characters are grouped into /I/ idle blocks such that idle insertion/removal is performed on a 64B/66B block basis. (See IEEE 802.3 clause 82 for additional information about this process, including the rules for removing /O/ characters for rate adaptation.)

IMP was specified as the mapping method for carrying Ethernet MAC clients and FlexE clients that have a bit rate ≥100 Gbit/s, although it can be used for 64B/66B-encoded clients at any rate. When carrying 64B/66B-encoded Ethernet client signals or FlexE[29] client signals, they are mapped into an ODUflex(IMP). The OPUflex payload area contains the stream of 64B/66B client blocks. Note in order to reduce the complexity for the OTN sink node to perform 64B/66B block synchronization, the 64B/66B blocks are 2-bit aligned within the OPU payload bytes. In other words, the first bit of the sync header (i.e., block bit 0) must be aligned with an odd numbered bit of the OPU payload byte.

Similar to the GFP frame mapping approach, when using a local clock, the rate adaptation is performed within the client information stream (i.e., within the OPU payload area) rather than using the OPUk justification control overhead. Specifically, the mapper uses the IEEE 802.3 clause 82.2.3.6 rate adaptation process of adding or removing /I/ blocks within the IPG regions of the client block stream to adjust its rate to fill the OPUflex(IMP) payload area capacity. Hence the name “Idle Mapping Procedure” (IMP).

8.3.3. Clock Generation Methods for ODUflex(GFP) and ODUflex(IMP)

Choosing the ODUflex(GFP) and ODUflex(IMP) rates requires special considerations. Since they carry packet clients, there are cases where deriving the ODUflex rate from the client signal is not practical or desirable. This section explains the common ODUflex clock generation options for ODUflex(GFP) and ODUflex(IMP), and then discusses the options unique to each.

---

[29] Flexible Ethernet (FlexE) is defined in the OIF FlexE Implementation Agreement as a way to carry multiple Ethernet client signals over groups of Ethernet PHYs. These can be 100, 200 or 400Gbit/s Ethernet PHYs.

8.3.3.1. ODUflex(GFP) and ODUflex(IMP) clock derived from a local clock source

Both ODUflex(GFP) and ODUflex(IMP) allow the option using a local clock to generate their ODUflex rate with a tolerance of ±20 ppm. For example, they can use the same clock source that is used to generate the node's OTUk (or OTUCn) server layer clock or some other equipment-specific clock.

If the server ODUk clock is used for deriving the ODUflex clock, then the relationship between their rates can be accommodated through the choice of either a fixed C_m or using two alternating C_m values for mapping the ODUflex into the server OPUk. The rate is chosen to be high enough to be sufficient to carry the highest expected rate of the GFP-F frame stream or 64B/66B encoded signal associated with that client. As explained below, ODUflex(GFP) supports a local clock option that makes maximum use of the available server TS bandwidth. See G.709 Appendix XV for the discussion regarding using a local clock to generate an appropriate ODUflex rate.

8.3.3.2. Client timing-based option for ODUflex(IMP)

Some clients (e.g., FlexE) provide a 64B/66B stream rate that can be used directly as the basis for the ODUflex(IMP) rate. In this case, the ODUflex(IMP) rate is simply (239/238) times the 64B/66B encoded packet client rate. In other words, it uses the same concept and multiplier as CBR client BMP mapping and directly preserves the client rate.

For FlexE, the client signal rate is specified by the OIF as s × 5,156,250.000 kbit/s ± 100 ppm and s = 2, 8 or n × 5 (n ≥ 1). The associated ODUflex(IMP) is hence labeled as ODUflex(IMP,s), with a nominal bit rate of s × 239/238 × 5,156,250.000 kbit/s ± 100 ppm.

8.3.3.3. ODUflex(GFP) Timing Option for Optimum Server TS Bandwidth Usage

This option is referred to as ODUflex(GFP,n,k), where k designates the OPUk server and n is the number of TS used in the OPUk. The principle behind the ODUflex(GFP,n,k) approach is to allow using the full bandwidth available in the set of OPUk server TS.

Since a lower rate ODU client signal can only be multiplexed into an integer number of TS in the higher rate OPUk, the ODUflex(GFP,n,k) rate can be defined to use the full available bandwidth in an integer number of TS rather than a rate based on the packet client information rate. Otherwise, any fraction of a TS bandwidth that is unused by the ODUflex(GFP) or ODUflex(IMP) client would be lost, since it cannot be used by any other client.

The special consideration with using ODUflex(GFP,n,k) is that the rate of a TS is not a fixed value in the network. As previously noted in Section 8.1, the TS rate increases when moving up the ODUk hierarchy (i.e., to larger values of k). This could cause a potential problem for signals that are carried by different rate HO OPU signals within the network. For example, if an ODUflex(GFP) is initially multiplexed into seven TS of an OPU4 but is then carried by an OPU2 at some point in the network, a signal that fills the seven OPU4 TS will not fit within seven OPU2 TS. That would potentially force the OPU2 to use an inefficient eight TS to carry this signal[30]. Clock frequency differences can create this same problem even for signals that remain at the same HO ODU hierarchical rate through the network.[31]

These rate considerations led to the specification of a set of nominal rates for all possible ODUflex(GFP) sizes. The rates were chosen to take into account the TS rate of the smallest OPUk into which that ODUflex(GFP) signal can fit. For example, since ODUflex(GFP) signals occupying 1-8 TS can fit within an ODU2, the ODU2 TS rate was used to determine their nominal ODUflex(GFP) signal rates. Further, the ODUflex(GFP) rates were chosen such that the worst-case combinations of the ODUflex(GFP) rate, ±100ppm, and the ODUk TS rate, ±20ppm still leaves adequate margin to accommodate jitter or wander[32]. In other words, the rate of an ODUflex(GFP) signal that occupies *N* TS was effectively defined as:

---

[30] Even if the original routing for the ODUflex(GFP) signal in this example only used ODU4 links, it is possible that it could need to be routed over an ODU2 link for fault restoration. In order to simplify the network management, there should be no restrictions on signal routing.

[31] Additional motivations for a precise ODUflex(GFP) rate specification included ensuring proper multi-vendor interoperability of HAO.

(*N*)(nominal TS rate of the smallest applicable OPUk)(Scaling factor)

where the scaling factor takes into account the clock tolerances as described above. G.709 refers to the scaled TS rate as the “ODUk.ts” rate for this application. The resulting ODUk.ts rate specifications are shown in Table 8-8 and Table 8-9.

Fortunately, GMP provides a simple mechanism to implement the scaling factors for each ODUk. The ODUflex(GFP) source node can use a constant C_m value that takes into account the combination of the number of TS used by the ODUflex(GFP) signal, the OPUk clock tolerances, and its hierarchical OTUk rate (i.e., k = 2, 3, or 4). The resulting C_m values and their associated ODUk.ts rates are shown in Table 8-9. These ODUk.ts rates, with their ±20ppm tolerance, fall within the range specified in Table 8-8. Note that as long as the resulting clock rate conforms to the Table 8-8 requirements, it is also possible for the source node to use an internal system clock to generate the ODUflex(GFP) signal rather than implementing the clock as a direct choice of C_m for the transmitted OTUk. See Appendix XI of G.709 for further explanation and elaboration regarding the ODUflex(GFP) rate derivations.

Table 8-8. ODUk.ts rates for ODUflex(GFP)

k	ODUk.ts (Gbit/s)
2	1.249177230 ±100ppm
3	1.254470354 ±100ppm
4	1.301467133 ±100ppm

Table 8-9. ODUflex(GFP) Source C_m Values and ODUk.ts Rates

ODUflex(GFP) rate (Number of TS)	HO OPUk rate at the ODUflex(GFP) source	C_m value at the ODUflex(GFP) source	Resulting bit rate per ODUk.ts (Gbit/s)
1-8	OPU2	15230	1.249245570 ±20ppm
	OPU3	15165	1.249184746 ±20ppm
	OPU4	14587	1.249212687 ±20ppm
9-32	OPU3	15230	1.254538983 ±20ppm
	OPU4	14649	1.254522291 ±20ppm
32-80	OPU4	15498	1.301537974 ±20ppm

8.3.4. Hitless Adjustment of ODUflex(GFP) (HAO)

As explained above, GFP Idle frames are used for rate adapting packet clients into an OPUflex and GMP is used for rate adapting an ODUflex into the TS of a higher-rate OPUk server. This combination created the possibility to dynamically change the number of TS used by an ODUflex and hence the OPUflex client rate capacity. Specifically, the number of OPUk TS occupied by a packet client can be changed dynamically without removing the client’s ODUflex path connection and re-provisioning a new path connection at the desired new rate. In other words, the client server rate can be changed in a hitless manner (i.e., without interrupting the client signal flow during the OTN resizing process). This HAO (hitless adjustment of ODUflex(GFP)) capability was added to OTN with the mechanism and protocol defined in G.7044.

Note: Microchip provided significant technical input to refining the G.7044 protocol, with Microchip devices recognized as a “golden” implementation.

G.7044 HAO consists of two communication/handshake protocols, each with its own resizing control overhead (RCOH):

- Link Connection Resize (LCR) provides handshake communication among the OTN nodes along the path to reserve the new set of TS in each link and within intermediate node switch fabrics.

---

[32] Specifically, the nominal ODUflex(GFP) signal rate occupying *N* TS was chosen to be 186ppm less than *N* times the TS rate of the smallest HO OPUk into which it could fit.

- Bandwidth Resize (BWR) provides the handshake for all the nodes to begin using the new TS set after they are reserved.

Since LCR controls the number of OPUk server layer TS, it uses the OPUk overhead for signaling. Specifically, it is carried in the OPUk OH during the frames associated with the TS(s) to be added or removed. The BWR overhead is located within the OPUflex since the ODUflex must be passed transparently through the path in order to coordinate the ODUflex source and sink nodes. See Figure 8-17 and associated Table 8-10 for the overhead locations and descriptions.

The protocol operations can be summarized as follows.

For rate increase, the process begins with all nodes on the ODUflex path receiving an ADD command from the NMS. The new capacity is then reserved, followed by changing the ODUflex rate and associated client rate to match the new channel rate:

- All nodes enter LCR and the source/sink pair signal ADD in the new TS
- During LCR:
  - The nodes on both ends of a link update their span interface (and internal switch fabric) to correspond to the added TS.
  - After the LCR handshake completes in both directions, a node increases its connection capacity and exits LCR.
  - When the source completes LCR it performs a connectivity check for each new TS, by setting the TSCC (TS Connectivity Check).
  - An intermediate node only forwards the TSCC when it has completed LCR.
- The connectivity is confirmed when the TSCC reaches the sink, which sends an ACK to the source. Then both ends begin BWR and signal it to intermediate nodes (BWR_IND).
- Both ends increase the ODUflex(GFP) rate gradually (512 Mbit/s^2) in order to avoid buffer overflows at nodes along the path.
- BWR completes when the target bandwidth has been reached.

For rate decrease, the process begins with all nodes receiving a REMOVE command from the NMS. The nodes pre-configure for the TS reduction and begin the BWR process under the control of the source. The TS are removed from the connection upon completion of BWR:

- All nodes enter LCR and the source/sink pair signal REMOVE in the affected TS,
- Each node enters BWR relay mode when both it and the node at the other end of the span is properly configured for the TS change (at this point, LCR pauses).
- A connectivity check is performed for the TS(s) being removed, with the TSCC forwarded by intermediate nodes when they have paused LCR.
- The connectivity is confirmed when the TSCC reaches the sink, which sends an ACK to the source. Then both ends begin BWR and signal it to intermediate nodes (BWR_IND).
- Both ends decrease the ODUflex(GFP) rate gradually (512 Mbit/s^2) in order to avoid buffer underflows at nodes along the path.
- BWR completes when the target bandwidth has been reached.
- LCR resumes when BWR has completed in order to remove the now unused TS.

Note: When nodes enter BWR, they must have their GMP processors configured for “GMP special mode.” The special mode allows GMP processors to handle the bandwidth and associated buffer adjustments due to the word size change. The TS connectivity check is only forwarded by an intermediate node after it has entered GMP special mode.

The nodes return to their original configuration if anything prevents the completion of the HAO protocol.

Figure 8-17. RCOH Fields for OPUflex and OPUk Server

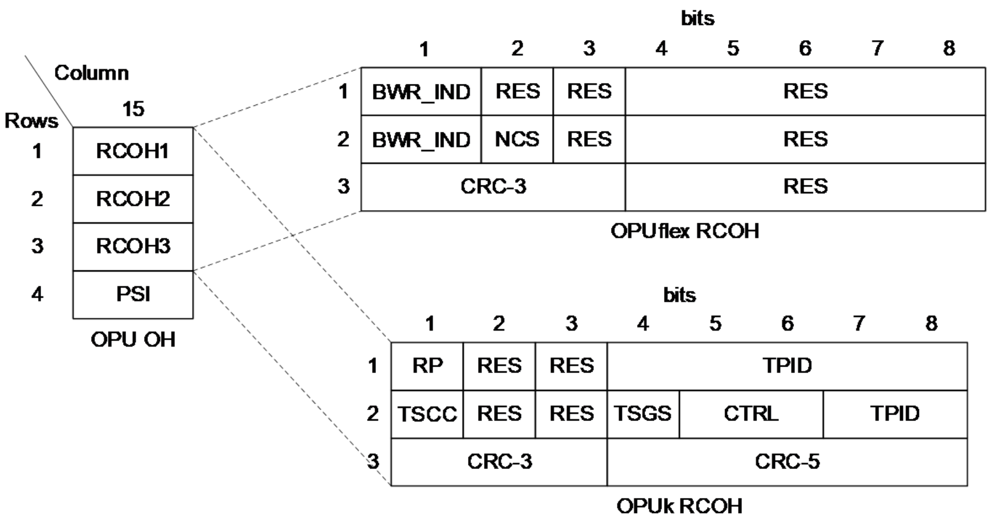


Table 8-10. Definitions of the RCOH Types in the Figure Above

Location	Term	Definition
LCR	CTRL	To communicate LCR status from the source to sink: • 00 = Idle (LCR complete and no new operation). • 01 = ADD (This TS will be added to the ODUflex(GFP)). • 10 = REMOVE (This TS will be removed from the ODUflex(GFP)). • 11 = NORM (LCR will begin at the next resize multiframe boundary, sent after ADD or REMOVE).
	TPID	TPID of the port associated with the bandwidth change.
	TSGS	TS Group Status – Sent by the sink to acknowledge that it is ready to begin using the new TS set.
	TSCC	TS Connectivity Check – Propagated hop-by-hop from source to sink to .
BWR	NCS	Used as an ACK from the sink to the source that the sink has received the correct TSCC=1 value. Receiving TSCC=0 and NCS=0 at the sink indicates BWR completion.
	BWR_IND	Indicates to the sink and all intermediate nodes that the source is adjusting the ODUflex(GFP) rate.

Note: The final target ODUflex rate is specified to match the capacity of the lowest OPUk TS rate along the path.[33]

8.3.5. Summary Information for OTN Client Mapping and Multiplexing

The Table below provides a brief summary of the type of mapping procedure used for some of the important OTN client signals.

Table 8-11. Payload Summary Information for Mapping Major OTN Client Signals

Client Signal	OPUk	Justification Method	Comment
1GE	OPU0	GMP	Uses GFP-T

---

[33] GMP word size is for ODU1-ODU4 is M bytes, where M is the number of TS occupied by that client.

CBR clients: client ≤1.238 Gbit/s	OPU0	GMP	Includes STM-1, STM-4, FC-100
CBR clients: 1.238 Gbit/s < clients ≤ 2.488 Gbit/s	OPU1	GMP	Includes FC-200
STS-48 (SDH STM-16)	OPU1	AMP or BMP	
STS-192 (SDH STM-64)	OPU2	AMP or BMP	
10GE LAN PHY	Extended OPU2	GFP Idles	
10GE LAN PHY	OPU2e	BMP	
Fiber Channel FC-1200	OPU2e	BMP	Uses TTT/GFP-T
STS-768 (SDH STM-256)	OPU3	AMP or BMP	
40GbE	OPU3	GMP	Transcoded
100GbE	OPU4	GMP	
CBR clients: client > 2.488 Gbit/s	OPUflex(CBR)	BMP	Includes FC-400, FC-800, IB-SDR, IB-DDR, IB-QDR
Packet clients	OPUk, k=0,1,2,3,4	GFP Idles	GFP-F
Packet client streams (≤100 Gbit/s)	OPUflex(GFP)	GFP Idles	GFP-F
Packet client streams (≥100 Gbit/s)	OPUflex(IMP)	IMP	

9. OTN Extensions for Rates Beyond 100 Gbit/s

9.1. Introduction

Taking OTN to rates higher than 100Gbit/s posed multiple challenges. Some of these were encountered when the 100Gbit/s OTU4 OTN signal was defined, but some were new challenges. Other objectives and challenges came up during the course of developing the Beyond 100 Gbit/s (B100G) standard. These challenges and objectives included:

- The Shannon channel capacity limits are catching up to optical transport network capabilities. Specifically, the 50 GHz channel spacing of the standard wavelength grid currently used for dense wavelength division multiplexing (DWDM) imposes limits on transporting signals over reasonable distances when they have rates much over 100Gbit/s (200Gbit/s was the practical limit per wavelength for distances of interest in telecom networks at beginning of the B100G project, although higher rates per wavelength will become possible driven by advances in Coherent DSP technology and new higher order modulation formats).
- The old paradigm of adding new discrete rates for OTN had largely reached its practical limits, making a modular rate and frame structure approach more attractive.
- The new OTN format needed to not only carry the then-emerging 400GbE, but also re-use its technology and PHY components in order to benefit from the Ethernet component cost curves.
- The higher bit rates and increased use of multi-lane interfaces poses additional considerations regarding Forward Error Correction (FEC) and performance monitoring.
- Different B100G interface types have different FEC performance capability requirements. Consequently, while the current OTUk frame format had fixed dedicated FEC overhead, it was more appropriate to specify the FEC on a per-interface basis and not make the FEC overhead an integral part of the frame structure.
- Continuing to use 1.25Gbit/s Tributary Slot (TS) sizes in OTN would be impractical for B100G rates, so a larger Tributary Slot size was desirable. As will be discussed below, this decision was complicated by the emergence of the IEEE 802.3 work on 25GbE.
- B100G interfaces should re-use as much IP from the 100Gbit/s OTN interfaces as practical.
- There should not be a new switching layer in OTN associated with B100G.
- The high data rates and the introduction of new data client signals such as the OIF's Flexible Ethernet (FlexE) have made the current byte-oriented mapping approach for data clients impractical. This motivated the desire for a wide-word type of mapping.
- In order to optimize the use of each wavelength, including the transmission reach, there was a desire to transmit the OTN signals at the rate required for the client payload being carried rather than at the full discrete rate of the OTUk signal.

Each of these challenges will be addressed in this white paper, along with its implications for the B100G standard and influence on early working assumptions. Similar considerations are discussed in the Considerations for OTN Beyond 1 Tbit/s section of this white paper regarding Q11 discussion about OTN optimizations for rates beyond 400 Gbit/s.

9.2. Optical Layer Considerations

The ITU-T (Q6 of Study Group 15) defined a standard wavelength grid for dense wavelength division multiplexing (DWDM) with the wavelengths on 50GHz (0.39nm) spacing (refer to Appendix D). Extending OTN up to 100Gbit/s was relatively straightforward in that the signal could be transmitted without using the entire 50GHz channel. In other words, the Shannon limit of the 50GHz channel was not exhausted. While NRZ (Non-Return to Zero) line codes were typically used for simplicity at the lower rates, a more complex modulation scheme was used for all 100Gbit/s and most 40Gbit/s OTN signals. As will be explained in this section, the combined use of more complex modulation schemes and multiple wavelength channels was also required for B100G.

While NRZ is the simplest line code, it has the drawback of relatively high frequency components. The impacts of dispersion in a fiber increase with the frequency content of the signal being carried, and for practical telecom network distances, this limits NRZ use to roughly 25Gbit/s. For higher rates, a more spectrum-efficient modulation is required. In other words, a modulation method was required that uses a lower baud rate[34]. Both Q11 and IEEE 802.3 adopted the PAM-4 (4-level Pulse Amplitude Modulation) line code for electrical and short-reach optical signal rates above 25 Gbit/s. PAM-4 encodes pairs of bits into a 4-level symbol and both Q11 and IEEE 802.3 use gray coding for better error performance

For optical signal transmission, the most commonly used line code for optical rates beyond 100 Gbit/s is DP-QPSK (dual-polarization – quadrature phase shift key), also known as PM-QPSK (Polarization Multiplexing-QPSK). While NRZ transmits one bit/symbol, DP-QPSK divides the transmitted signal onto two different polarization modes and uses two bits/symbol for each polarization. Consequently, the spectrum used by DP-QPSK is approximately ¼ of the spectrum used by NRZ. DP-QPSK was also attractive because it gave the best performance relative to the power dissipated in the optical module. The dual polarization takes advantage of fiber’s ability to preserve the relative polarization of the input signals as they transit the fiber, thus allowing different signals to be transmitted with different polarizations and recovered at the receiver. A 90-degree polarization difference is typically chosen for robustness. For some very long-distance links, such as undersea cables, DP-BPSK (dual-polarization – binary phase shift key) lines codes have been used. The16QAM (quadrature amplitude modulation) line coding is also being used for some metro applications, due to its potentially lower cost. However, it requires linearity, and hence DSP, which increases the power dissipation in the module.

As rates increase beyond 100Gbit/s, fitting within the 50GHz wavelength slot[35] requires increasingly complex modulation, which lowers the signal to noise ratio (SNR). For example, DP-16QAM (Dual Polarization – 16QAM) is expected to be used for rates around 200Gbit/s in order to fit within the 50GHz optical channel. As noted above with respect to metro applications, this requires DSP at the receivers; however, DSP receivers are already commonly used for DP-QPSK receivers due to the implementation advantage they give over traditional receiver designs. The lower SNR of the 16QAM relative to the QPSK reduces the achievable reach of the signal. As it turns out, 200Gbit/s is effectively the maximum rate that can be transmitted over a 50GHz channel for most distances of interest in telecom networks. Consequently, higher rate B100G signals will typically be divided across multiple optical channels[36]. While interfaces using multiple wavelengths were already an option beginning with OTU3 and 50GbE, they are typically required for a OTN signal rates beyond 100Gbit/s.

One alternative to addressing the limits of the 50GHz DWDM channel slots is to either move to wider slots or remove the wavelength grid altogether. A standardized wider slot is unattractive, since it would not be efficient for current signals, and could be too coarse for some common applications. Hence, it would typically not make efficient use of the entire available spectrum.

---

[34] Recall that baud = symbol/second, and the baud rate is hence sometimes referred to as the symbol rate.

[35] Recall that since the optical signal frequency (η) is the speed of light divided by its wavelength (η =c/λ), WDM is essentially a form of FDM, where the transmitted wavelength is modulated similar to the carrier frequency in an FDM system. As with all FDM systems, as the symbol rate increases for a given modulation format, the frequency spectrum used by that signal increases. When the frequency spectrum used by a given optical signal increases to the point where it overlaps with the signal in an adjacent channel, crosstalk occurs.

[36] For the purposes of this white paper, the term “optical channel” is being used in the loosest sense, and essentially equates to a wavelength or a wavelength slot/band or channel slot within the ITU-T DWDM grid. Strictly speaking the correct technical term for this is a network media channel. Also, in the strict OTN sense, an optical channel is the set of wavelengths that carry both the OTN payload signal and the associated optical supervisory channel (OSC) on a separate wavelength.

Removing the grid altogether is not practical, since it would require such a wide range of optical sources. The most practical approach is going to a narrower grid where multiple adjacent slots can be grouped together to create larger slots. This flexible DWDM approach, sometimes referred to as FlexGrid, is defined in ITU-T G.694.1. Specifically, G.694.1 defines a grid with 12.5GHz channels, and illustrates how adjacent slots can be combined to create wider channels. The FlexGrid approach allows the network operator to allocate the optimum wavelength channel size to each transport signal, thus maximizing the use of the available spectrum. However, the inevitable churn from network reconfiguration will quickly produce a fragmented spectrum on the fiber where it will be difficult to find new openings for larger channel slots. While it may be feasible in some applications, FlexGrid does not provide a good general solution. Consequently, the 50GHz grid will continue to be used for some time.

It should be noted that coherent receivers are assumed here. The reason coherent receiver technology became popular is that it provides a linear transformation between the optical and electrical domains. This allows using the electronics to filter/cancel out the effects of both polarization mode dispersion (PMD) and chromatic dispersion. Using coherent receivers does not increase the spectral efficiency or reach, rather it just simplifies the receiver implementation since the dispersion compensation is done by DSP in the electrical domain rather than the optical domain.

For the multi-lane electrical interfaces, it was initially agreed that a B100G signal will use at least 25-28Gbit/s per lane, which was also consistent with 400GbE. Subsequent support for 50 Gbit/s lanes and 112 Gbit/s lanes has been added in both domains. See the OTN Electrical Interfaces – MFI FlexO Interfaces of Order C (FOIC) and OTL (OTN Transport Lane) section for a description of the multi-lane electrical interfaces.

9.3. B100G Signal Formats and Frame Structure

Traditionally, the standard SDH and OTN rates had increased at each step by a factor of four. As noted above, the one exception to this was ODU4, which was chosen to be optimized for carrying 100GbE Ethernet clients rather than choosing a 160Gbit/s rate for carrying four of the 40Gbit/s ODU3 signals. By that time, it had become apparent that Ethernet was an increasingly important client signal for transport over OTN. Also, a rate around 112Gbit/s was much more cost effective for the bandwidth relative to 160Gbit/s, especially for the optical component technology available at that time. This was especially true since the OTU4 could then re-use the Ethernet 100GbE optical interface modules. A final consideration was the potential for OTU4 to become the default long-reach WAN interface for 100GbE. The ITU-T continues to use a similar line of reasoning for rates beyond 100Gbit/s.

Since IEEE 802.3 had begun its work on 400GbE (IEEE P802.3bs) the ITU-T needed to balance the desire to have a rate optimized for carrying and re-using technology from 400GbE, and the need for a more modular rate to carry multiplexed signals at lower rates over longer reach channels. The decision was made to standardize a limited number of B100G rates that matched new Ethernet rates and adopt an n×100G modular structure that also allows constructing signals to match the channel characteristics/quality. This approach was intended to reduce cost by having fewer, higher-volume options. This also fit well with an independent proposal in the OIF to define a modular flexible approach to Ethernet PHYs called “Flexible Ethernet” (FlexE) that is summarized in the Considerations for Flexible Ethernet (FlexE) Client Signals section with respect to FlexE being an OTN client signal. The FlexE concept inspired an analogous approach for OTN B100G interfaces that is called “Flexible OTN” (FlexO), which is discussed further in the Flexible OTN (FlexO) section.

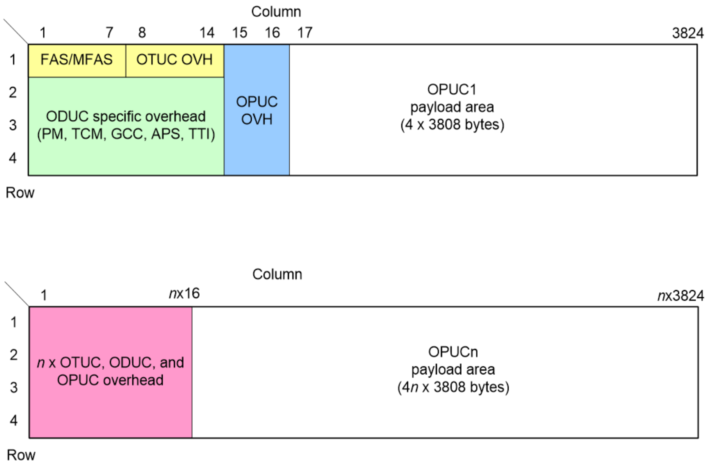
The OTN ODUk structure had used an asynchronous multiplexing approach, with the next higher discrete rate carrying a combination of the lower rate signals. The frame of each OTN ODUk rate was identical but transmitted at a higher signal rate. In contrast, SONET/SDH created its higher rate signals by interleaving an integer multiple of its base rate signals (SONET STS-1 or SDH STM-1). For rates beyond 100Gbit/s, the ITU-T Q11/15 chose to use a hybrid of these two approaches. A new base signal frame was established at around 100Gbit/s, with multiples of this base frame interleaved to create the higher rate signals. The terminology chosen was to call the base frame an ODUC (100

Gbit/s ODU slice), and to call signal constructed from *n* × ODUC slices an ODUCn, where the “C” corresponds to the Roman numeral for 100.

9.3.1. B100G Frame Format

As illustrated in Figure 9-1, the base ODUC frame uses a frame structure identical to the ODU4, except that there are no fixed stuff columns in the payload area. The OPUC payload structure is illustrated in Figure 9-3.

Figure 9-1. Illustration of the OTUC1 Base Frame Format and OTUCn Frame Format



The physical layer of the OTUCn signal depends on the interface. For example, it can be transmitted as a single serial stream, as *n* 100Gbit/s streams, or *n*/2 200Gbit/s streams in the optical domain, or as multiples of 25Gbit/s or 50Gbit/s with electrical domain interfaces. Rather than defining a serial interleaving format for the OTUCn stream, analogous to that of ODUk (k = 0-4), it was left to be specified by the individual interface. As illustrated in Figure 9-4 of OPUCn Payload Area Structure, the OTUC slices are interleaved in a defined manner to form a contiguous OPUCn payload area such that the OPUCn Tributary Slots have a known order. Since the OPUCn tributary slots and word locations are numbered (see Figure 9-3), there is no ambiguity in constructing the OPUCn at the source and recovering it at the sink. For transmission, however, each OTUC is treated individually as a 100Gbit/s entity. While, for example, an OTUC can be transmitted over 4 of the 25Gbit/s lanes or 2 of the 50Gbit/s lanes of an OTUCn interface, the “FlexO” interface discussed below in Flexible OTN (FlexO) was developed as the optical interface format for carrying ODUCn. FlexO transmits each OTUC element as a separate 100Gbit/s optical signal[37] using 100GbE/OTU4 optical modules, and hence requires no OTUC interleaving in the PHY layer.

In addition to the modular frame structure, the B100G signals also differ from the ODUk OTN signals in another important way. For network simplification, an ODUCn signal is only carried point-to-point between network nodes. In other words, the ODUCn signal is only a Multiplex Section layer entity that cannot be switched. It only exists to carry lower rate ODUk signals between a pair of nodes, with all switching being done at the ODUk level. The implications of this decision include: [38]

- No client signals are directly mapped into the OPUCn. They must first be mapped into an ODUk (including ODUflex), which is then mapped or multiplexed into the OPUCn. This topic is discussed below in the B100G Client Signal Mapping, Multiplexing and Rate Adaptation section .

---

[37] As explained below, 200 Gbit/s and 400 Gbit/s FlexO modularity was added when the corresponding Ethernet PHY rates were standardized.

[38] This decision not to have ODUCn switching would appear to imply that TCM is not required. However, as discussed below in the overhead section, TCM is still very useful with ODUCn.

- All the components of the interface signal go through the same fiber and optical switches (i.e., the same Optical Multiplex Section trails) such that the OTUCn signal can be managed as a single entity. Consequently, very limited deskew is required[39].

A fundamental difference between the current OTUk frame format and the OTUCn frame format, as seen when comparing Figures 6-1 and 9-1, is that the OTUCn frame has no dedicated area for FEC. Otherwise, the OTUCn and ODUCn frame formats are identical, except for the population of the OTUC-specific overhead fields. As discussed above, the reason for this choice is the recognition that there will be different interface types for the OTUCn, and each interface will have its own requirements in terms of the strength of FEC that is required. The FEC is provided for the interfaces through the FlexO frame described in section Flexible OTN (FlexO).

The rate of the base OTUC signal was chosen in order to meet the following criteria:

- An OPUC1 must be capable of carrying an ODU4 client
- An OPUC4 must be capable of carrying a 400GbE client.
- The resulting signal rate should be reasonably efficient within the constraints of the first two criteria. For example, when the signal is divided down to the nominally 25Gbit/s rate, the resulting electrical lane interface rate should be compatible with the OIF CEI-28G specification.

The resulting rates are shown in the Table below.

Table 9-1. OTN B100G Signal and Payload Rates

OTUCn/ODUCn signal rate	OPUCn payload area rate	OTUCn/OPUCn frame period
n × (239/226) × 99.5328 Gbit/s = n × 105.258138 Gbit/s	n × (238/226) × 99.5328 Gbit/s = n × 104.817727 Gbit/s	1.163 µs
Note: All rates are ±20 ppm.		

9.3.2. B100G OTUC and ODUC Signal Overhead

As with the OTUk signals, the OTUC shares overhead columns with the ODUC overhead. The OTUC1 overhead is identical to the OTUk overhead described above. The MFAS is also used in removing skew between portions of the OTUCn signal that were transmitted over different wavelengths or different FlexO interface PHYs. As can be seen from Table 9-2, some of the overhead bytes are active on only the first OTUC slice (OTUC #1) , while others are active on all slices. Overhead fields that are only active for OTUC #1 pertain to the whole OTUCn interface and hence only need to appear once.

Table 9-2. OTUC Overhead Usage

OTUC Overhead	Active on OTUC #i Slices
Frame Alignment Signal (FAS) OA1 & OA2	1-n
Multi-frame Alignment Signal (MFAS)	1-n
SM – TTI	1
SM – BIP8	1-n
SM – BEI	1-n
SM – BDI	1
SM – BIAE	1-n
SM – STAT	1
GCC0	1-n*
* The GCC0 channels on all n OTUC slices are merged into a single higher-rate channel for the OTUCn interface. For vendor-specific interfaces, GCC0 can optionally be active on only OTUC slice 1.	

---

[39] The primary contribution to skew is the different propagation speeds of different wavelengths in a fiber when different parts of the OTUCn signal are transmitted on different wavelengths.

Note that scrambling is still required to protect against malicious users who may attempt to interrupt the operation of an OTUCn link by sending long information strings in the client payload that would cause problems for the receiver clock and data recovery circuitry. However, for OTUCn signals, the scrambling is specified as part of the PHY signal (e.g., FlexO) rather than specified in a general sense for the OTUCn.

The ODUCn consists of the ODUCn overhead, which is functionally a Multiplex Section overhead, and the OPUCn. The ODUCn overhead is identical to the ODUk overhead. Table 9-3 shows which ODUC overhead field is active in which slice.

Since the ODUCn is not switched, it does not require TCM in quite the same way that a switched ODUk does. However, for applications where the OTUCn signal goes through 3R repeaters, TCM is valuable for determining the performance of the optical signals on each side of the repeater. A "carrier's carrier" application can use multiple repeaters along the connection. Due to the multiple optical segments and the different service providers along the OTUCn path, multiple levels of TCM are still required. Consequently, it was decided that the B100G overhead will continue to support the same six levels of TCM overhead that are defined for current ODUk.

Table 9-3. ODUC Overhead Usage

ODUC Overhead	Active on ODUC*i* Slices
Delay Measurement (DM)	1
PM – TTI	1
PM – BIP8	1-*n*
PM – BEI	1-*n*
PM – BDI	1
PM – STAT	1
PM – Delay Measurement	1
Expansion (EXP)	1-*n*
GCC1	1-*n**
GCC2	1-*n**
APS/PCC	1

* The GCC1 and GCC2 channels on all *n* ODUC slices are merged into a single higher-rate GCC1 and GCC2 channel for the ODUCn interface. The GCC1 and GCC2 fields can also be optionally combined/merged to create a single larger channel from both.

9.3.3. B100G OPUC Signal Overhead

The OPUCn overhead is illustrated in Figure 9-2, with Table 9-4 indicating the OTUC slices on which the overhead fields are active. As can be seen in comparison to Figure 7-5, it is essentially the same as the OPUk overhead when GMP is used. The differences are explained below in the mapping and multiplexing discussion.

For all applications, the Payload Structure Indicator (PSI) byte contains indicators for the payload type (PT) and Multiplex Structure Identifier (MSI). Note that because no client signals are mapped directly into an OPUCn (i.e., the clients are always first mapped into their own ODUk/ODUflex), PT = 22 ("ODU multiplex structure supporting ODTUCn.ts") is the only PT used for B100G signals. Since the entire OPUCn is a combined payload area, it is only necessary to indicate the PT in OPUC #1. The MSI indicates for each TS of the ODUC whether that TS is available, whether it has been allocated, and the tributary port number associated with the client signal using that TS. The reason for indicating the availability of the TS will become clear in the discussion of "OTUCn-M" below. As shown in Figure 9-2, the set of PSI bytes appear in row 4 of column 15 of each ODUC slice. The correspondence between specific OPUCn overhead bytes and their respective payload area Tributary Slots is discussed below in section 7.5 under payload mappings.

The frequency justification required in order to adapt between the client signal rate and the OPUCn channel payload rate is located in the OPUCn overhead area. The details of the JC bytes are described in the Generic Mapping Procedure (GMP)section, with the correspondence between their location within the multiframe and the associated TS illustrated in Figure 9-3.

Hitless Adjustment of ODUflex(GFP) (HAO) is not supported by the B100G signals.

Figure 9-2. B100G OPUC1 Signal Overhead Structure with the Generic Mapping Procedure

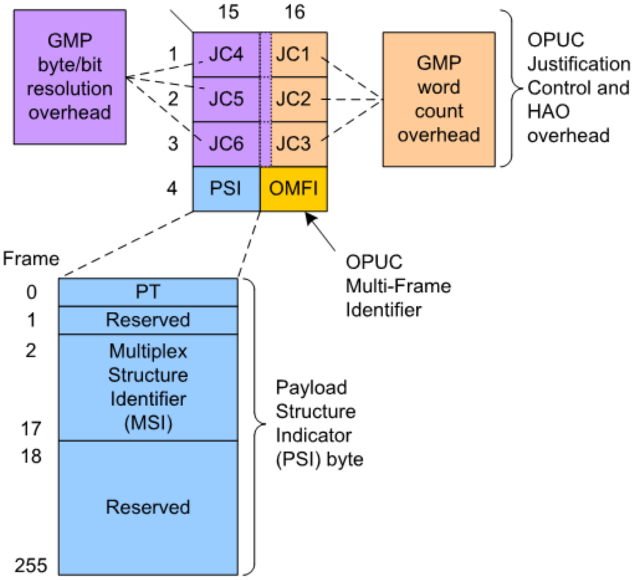


Table 9-4. OPUC Overhead Usage

OPUC Overhead		Active on OPUC*i* Slices
PSI	PT = 22	1
	MSI	1-*n*
GMP JC1-JC3		1-*n*
GMP JC4-JC6		1-*n*

9.4. OPUCn Payload Area Structure

As noted above, it is recognized the single byte base word size is impractical for B100G, as is continuing to use the 1.25 Gbit/s TS rate. A 10 Gbit/s TS rate was initially considered, however the development of new client signals, especially 25GbE Ethernet, made it more practical to adopt using a 5 Gbit/s TS rate. The individual TS are interleaved into the OPUCn contiguous payload area on a 16-byte (128-bit) block basis. In other words, each of the 20*n* TS occupies 16 bytes at a time in a fixed round-robin manner. This 16-byte TS interleaving granularity maintains a consistent 16-byte-modularity for both the OTUC/ODUC/OPUC overhead and the TS. Moving to 16-byte TS interleave granularity rather than the current single byte granularity is much more convenient with the high B100G data rates and the associated wide data buses within ICs that process them.

The OPUCn structure, including an illustration of the TS naming convention, is shown in Figure 9-3 and Figure 9-4. The TS number is designated as TS A.B. The “A” refers to the OPUC slice associated with that column/block (reflecting the fact that the OPUCn logically consists of *n* OPUC slices). The “B” term refers to the order of appearance of each group of 16-byte TS blocks within slice A, up to 20 groups (i.e., B = 1, ..., 20). The order of TS appearance within the OPUCn frame is first by the OPUC slice and then by the TS number within an OPUC slice.[40] In other words, as illustrated in Figure 9-4, the TS A.B appearance order is TS1.1, TS2.1, TS3.1,..., TSn.1, TS1.2, TS2.2,..., TSn.2, TS1.3,..., TS1.20,..., TSn.20. See Figure 9-5 for an example illustration of mapping a client into OPUCn.

The actual TS rate can be calculated as follows: (See Figure 9-3 and Table 9-1.)

TS rate = [OPUC rate]*[ratio of bits/row/TS to the OPUC bits/row] = 5.24089 Gbit/s per TS[41]

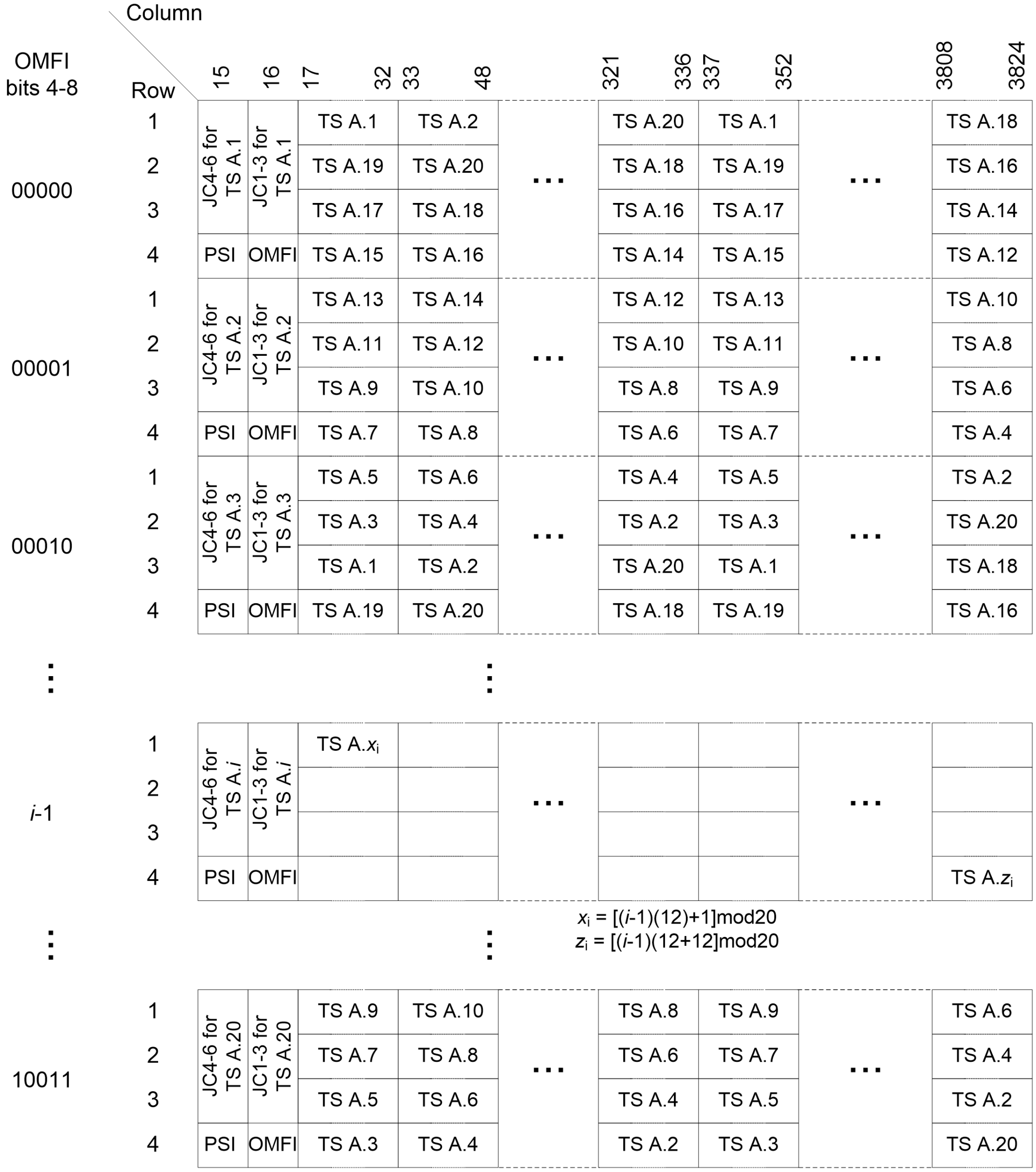
The OTN Generic Mapping Procedure (GMP) is used as the Justification Control to map client signals into the OPUCn. The GMP overhead is described in sections Generic Mapping Procedure (GMP) and B100G Client Signal Mapping, Multiplexing and Rate Adaptation. With the interleaving of *n* OPUC elements to form the OPUCn, there will be *n* appearances of Justification Control overhead (JC1-JC6) per frame (i.e., one per OTUC slice). A 20-frame multiframe provides a unique Justification Control overhead location associated with each of the 20*n* TS. The locations of the JC1-JC6 bytes associated with each TS are illustrated in Figure 9-3. In general, following the TS(A.B) naming convention, the location of the JC1-JC6 bytes associated for a given TS a.b are located in columns 14 and 15 of frame *b* - 1 of OPUC number *a*. As discussed above in Generic Mapping Procedure (GMP), the OPUC GMP word count overhead (C_m carried in bytes JC1-JC3) gives the number of data words carried in the TS(s) during that multiframe.

---

[40] Since there are 3808 payload columns per row in the OPUC, each of the 20*n* TS in the OPUCn appears exactly [(3808*n* columns/OPUCn row)] / [(16 columns/TS)*(20*n* TS)] = 11.9 times/row Consequently, the set of TS will repeat their row alignment every 10 rows. (i.e., exactly 8 times per 20-frame multiframe). Since 10 rows is 2.5 frames, the TS locations repeat their alignment to the OPUCn frame every 5 frames (i.e., 4 times per 20-frame multiframe).

[41] TS rate = [104.817727 Gbit/s]*[(row/3808*8 bits)(128 bits/TS/occur.)(11.9 occur./row)] = 5.24089 Gbit/s per TS

Figure 9-3. OPUC Tributary Slot Interleaving and Overhead Structure (for Slice A)



OMFI bits 4-8	Row	15	16	17–32	33–48		321–336	337–352		3808–3824
00000	1	JC4-6 for TS A.1	JC1-3 for TS A.1	TS A.1	TS A.2	…	TS A.20	TS A.1	…	TS A.18
	2			TS A.19	TS A.20		TS A.18	TS A.19		TS A.16
	3			TS A.17	TS A.18		TS A.16	TS A.17		TS A.14
	4	PSI	OMFI	TS A.15	TS A.16		TS A.14	TS A.15		TS A.12
00001	1	JC4-6 for TS A.2	JC1-3 for TS A.2	TS A.13	TS A.14	…	TS A.12	TS A.13	…	TS A.10
	2			TS A.11	TS A.12		TS A.10	TS A.11		TS A.8
	3			TS A.9	TS A.10		TS A.8	TS A.9		TS A.6
	4	PSI	OMFI	TS A.7	TS A.8		TS A.6	TS A.7		TS A.4
00010	1	JC4-6 for TS A.3	JC1-3 for TS A.3	TS A.5	TS A.6	…	TS A.4	TS A.5	…	TS A.2
	2			TS A.3	TS A.4		TS A.2	TS A.3		TS A.20
	3			TS A.1	TS A.2		TS A.20	TS A.1		TS A.18
	4	PSI	OMFI	TS A.19	TS A.20		TS A.18	TS A.19		TS A.16
⋮				⋮						
i-1	1	JC4-6 for TS A.*i*	JC1-3 for TS A.*i*	TS A.x_i		…			…	
	2									
	3									
	4	PSI	OMFI							TS A.z_i
⋮				⋮						
10011	1	JC4-6 for TS A.20	JC1-3 for TS A.20	TS A.9	TS A.10	…	TS A.8	TS A.9	…	TS A.6
	2			TS A.7	TS A.8		TS A.6	TS A.7		TS A.4
	3			TS A.5	TS A.6		TS A.4	TS A.5		TS A.2
	4	PSI	OMFI	TS A.3	TS A.4		TS A.2	TS A.3		TS A.20

$x_i = [(i-1)(12)+1]\mathrm{mod}20$
$z_i = [(i-1)(12+12]\mathrm{mod}20$

Figure 9-4. Illustration of the OPUCn Tributary Slot Order (from first row of OPUCn Frame)

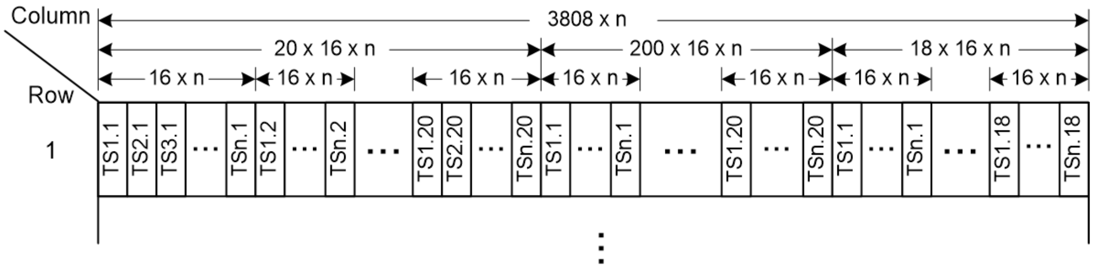
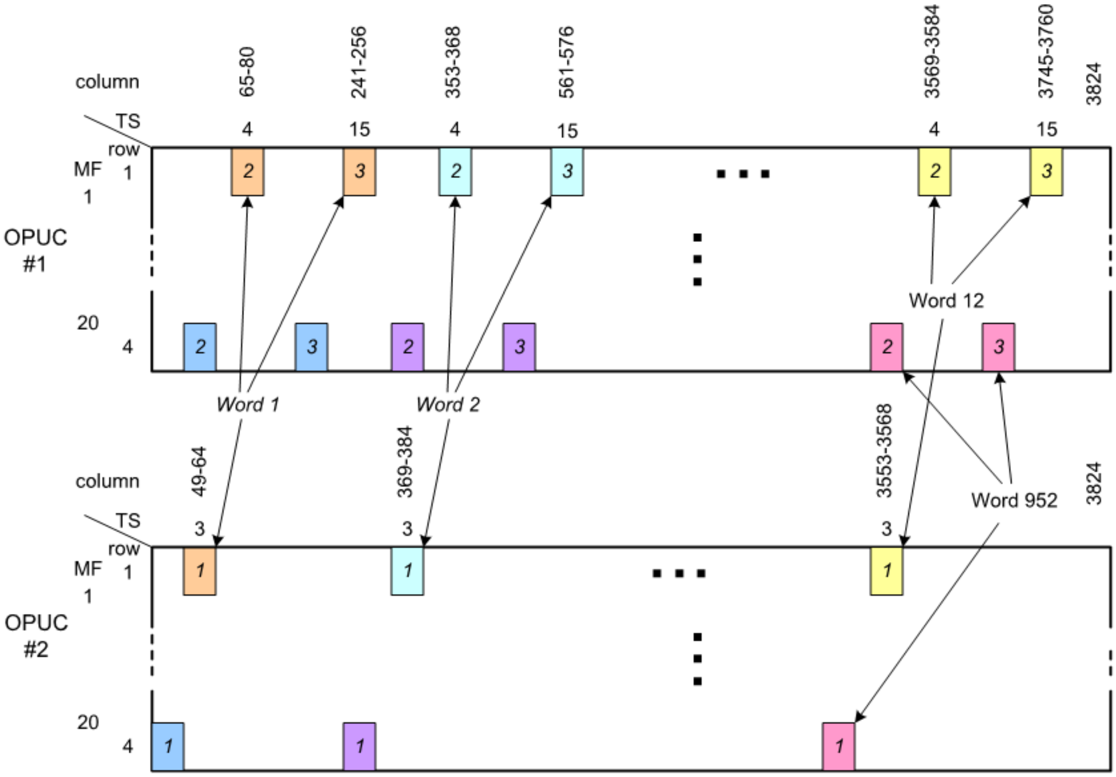


Figure 9-5. Word Numbering for GMP Multiplexing – ODUflex Tributary Using Three TS (#2.3, 1.4 and 1.15) into an OPUC2

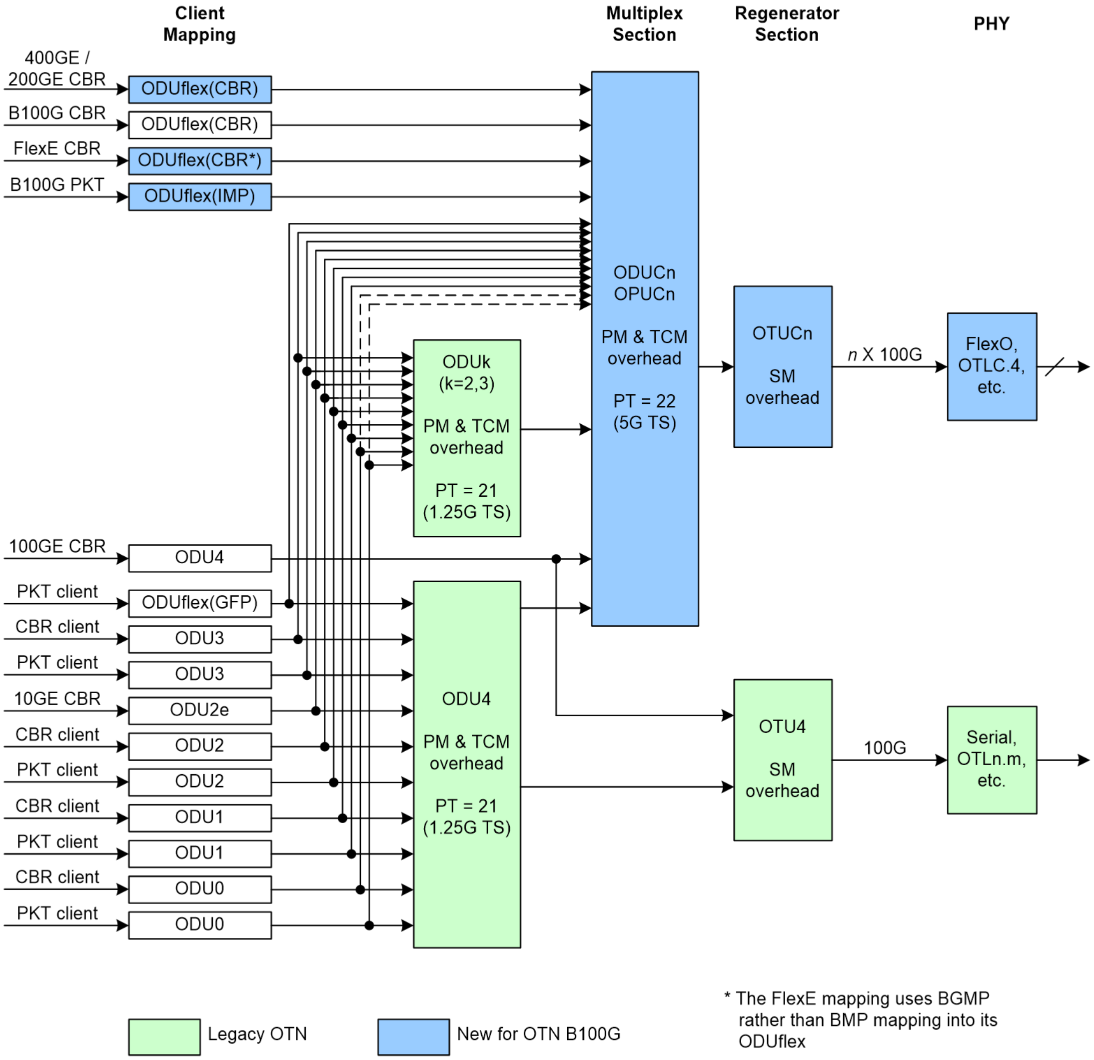


Note: In this example, different colors are used for each word shown for that client. The sequence number of the 16-byte segment of the 3 x 16-byte word is shown in italics.

9.5. B100G Client Signal Mapping, Multiplexing and Rate Adaptation

The defined client signal mapping and multiplexing hierarchy is illustrated in the Figure below, which includes the mappings into ODUk signals.

Figure 9-6. B100G Multiplex Hierarchy Illustration



As noted, all clients carried within an OPUCn are first mapped into an ODUk using one of the methods described in sections CBR Client Mapping and Packet-Oriented Client Mapping. GMP is used for multiplexing all these ODUk signals into the OPUCn.

Since the B100G the TS appearances are structured as 16-byte columns, with 16 bytes (128 bits) of client data mapped into each TS appearance, it made sense to use C_{128} as the base granularity for B100G GMP. A client signal using *m* TS will have a 128*m*-bit GMP word. As shown in Figure 9-3, each 5Gbit/s TS appears in 3808/20 = 190.4 columns of the OPUC. With the 16-byte base GMP word size (C_{128}), there are 952 words/TS/multiframe.[42]

The ODTU concept discussed above in the CBR Client Multiplexing section for OPUk is also used with OPUCn. The client-bearing ODUk signals are first mapped into an ODTUCn.ts structure, which includes the information to be carried in the "*ts*" set of OPUCn TS used by that client, and the associated JC-byte overhead illustrated in Figure 9-7 and Figure 9-8. Each (16)×(*ts*) byte word of the

---

[42] (190.4 bytes/row/TS)(4*20 rows/MF)/(16 bytes/word) = 952 words/TS/MF

ODTUCn.ts is then word-synchronously mapped into the set of the OPUCn TS used by that client such that the ODTUCn.ts consists of a stream of *ts* × 16-byte data or stuff words (see OPUCn Payload Area Structure for description of the TS appearance order). Since each multiframe will contain 952 words (either data or stuff words) associated with that client, for the purposes of GMP, the OPUCn payload words are numbered from 1-952. See Figure 9-5 for an example illustrating the word-to TS mapping associated with a client using 3 TS.

Consistent with GMP multiplexing in an OPUk, when a tributary uses multiple TS, the JC1-JC3 GMP overhead associated with that client signal is located in the last JC overhead appearance in the multiframe associated with that set of TS. See Figure 9-3 for the illustration of mapping between TS number and the associated JC byte location within the multiframe, and the discussion in Section OPUCn Payload Area Structure for the order of TS appearance.

Since each GMP overhead count value is never more than 952, regardless of the size of *n* in the OPUCn, the GMP overhead for B100G uses a 10-bit GMP count field rather than the 14-bit count field used for ODUk. This frees four bits to extend the $\sum C_{nD}$ field for additional resolution with large GMP word sizes, as shown in Figure 9-8. Consider for example the case of a 400GbE mapping. The 400GbE client is first BMP mapped into an ODUflex, which will occupy 4 × 20 = 80 TS, resulting in an 80 × 128 = 10240-bit (1280-byte) GMP word size. An 8-bit $\sum C_{nD}$ resolution is specified for the OPUCn, which requires a $\sum C_{nD}$ field larger than the 10-bit $\sum C_{nD}$ field used for OPUk. The 18-bit OPUCn $\sum C_{nD}$ field can provide byte level resolution for GMP word sizes up to 262144 bytes, which allows an ODUflex to occupy up to (262144 × 8)/128 = 16384 TS, supporting a client that occupies over 320 FlexO instances (e.g., a 3.2 Tbit/s FlexO-320). The GMP field encoding and JC octet format are illustrated in Figure 9-7, Figure 9-8 and Table 9-5. See the discussion of GMP decoding and robustness, along with the $\sum C_{nD}$ discussion in the Generic Mapping Procedure (GMP)section.

Figure 9-7. GMP Justification Control Overhead Fields

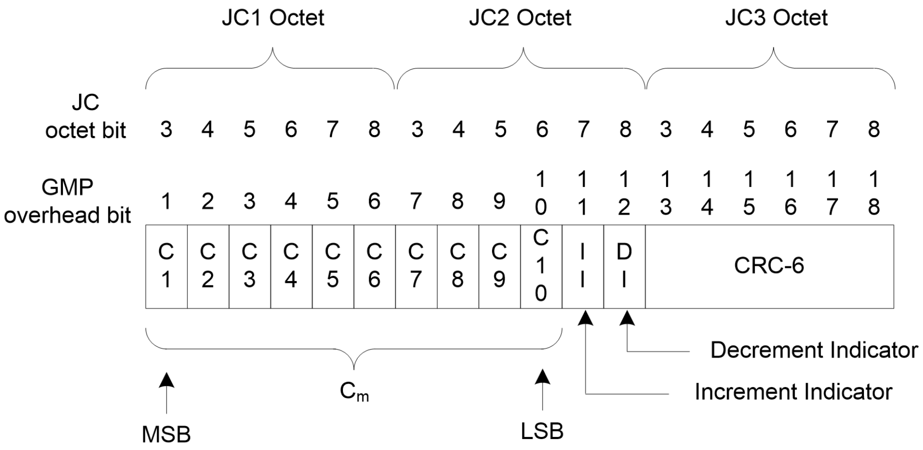


Figure 9-8. Octet Illustration of all GMP Overhead Fields

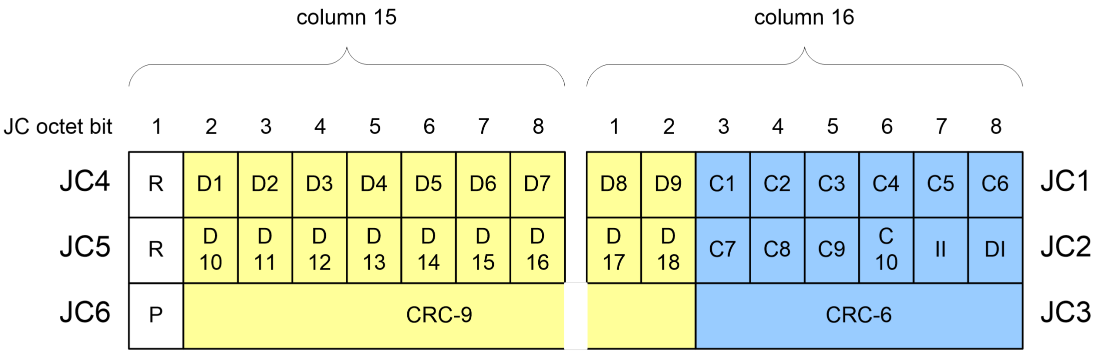


Table 9-5. C_m Bit Inversion Patterns to Indicate Increment and Decrement

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	II	DI	Δ
U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	1	0	+1
I	U	U	I	U	I	I	U	U	I	0	1	–1
U	I	U	I	I	U	U	I	U	I	1	0	+2
U	I	I	U	U	I	U	I	I	U	0	1	–2
Binary value										1	1	>±2
NOTE – I indicates inverted Ci bit – U indicates unchanged Ci bit												

Table 9-5 C_m bit inversion patterns to indicate increment and decrement

9.5.1. Considerations for 200 Gbit/s, 400Gbit/s and 800Gbit/s Ethernet Client Signals

The OTN mapping reference signal for y00GBASE Ethernet PCS clients with y ≤ 4 was a 64B/66B PCS block stream for which IMP mapping (see Clock Generation Methods for ODUflex(GFP) and ODUflex(IMP)) was used. Starting with 800GBASE, the OTN mapping reference signal was moved to be the stream of 256B/257B blocks. Since 256B/257B blocks have no direct equivalent of the 64B/66B Idle block, it would not be possible to use IMP without first transcoding back to 64B/66B and subsequently re-transcoding to 256B/257B. Consequently, beginning with 800GBASE Ethernet (i.e., y ≥ 8) a new mapping method was required.

The mapping method was chosen to be a modified form of BMP. As with any BMP mapping, the ODUflex rate is derived from the client signal rate. As illustrated in Figure 9-9, the client 257b blocks are aligned to the beginning of the OPUflex frame payload area and a 38-bit field at the end of the OPUflex frame fills the remaining bandwidth at the end of the 474 blocks. The 38-bit field contains a 6-bit pad, and an optional CRC-32 discussed below.

At the OTN reference signal point within the 800GBASE PCS, the 256B/257B blocks have been descrambled and the 257-bit Ethernet Alignment Marker (AM) blocks have been removed, since the Ethernet AMs are not carried over the OTN[43]. The mapping process inserts stuff blocks into

---

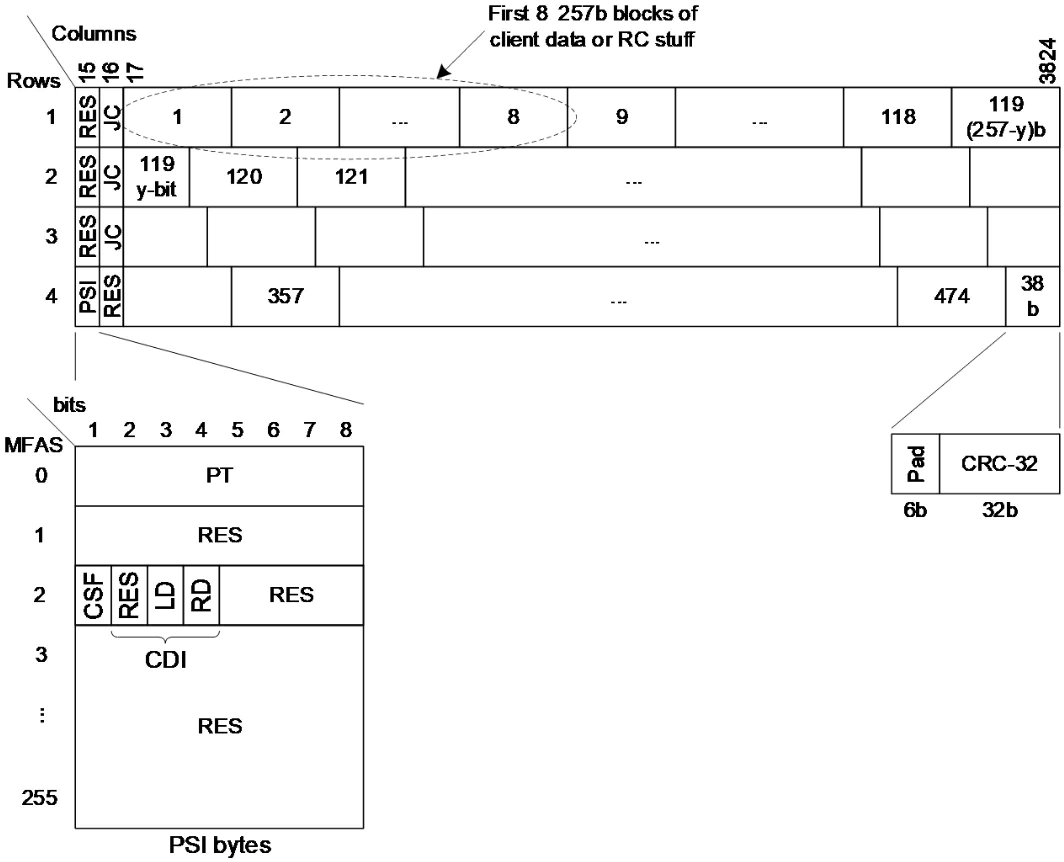
[43] Among the reasons to not carrying the Ethernet AMs across the OTN is that the OTN mapping is specified to be independent of the Ethernet lower layers (e.g., PMD, PMA and FEC). Removing the AMs allows Ethernet interfaces on either side of the OTN network to use a different number of lanes. This allows the OTN mapper and demapper nodes to operate independently and only handle their local Ethernet interface without concern for the Ethernet interface on the other side of the OTN. Note that this decision was based on the assumption that IEEE 802.3 maintains consistency for its initial and all future PMD types, which it did not do with 25GbE.

the OPUflex to compensate for the bandwidth associated with the removed AMs. Specifically, the bandwidth occupied by the stuff blocks is consistent with the nominal bandwidth occupied by the AMs within the 800GBASE client signal.

The stuff block insertion mechanism is to algorithmically place groups of eight 257b stuff blocks at the start of the OPUflex payload area in some of the OPUflex frames. See Figure 9-9. The algorithm deterministically inserts the eight stuff blocks into some OPUflex frames according to a (20k-1)/(20k) ratio under the control of a sigma-delta mechanism. Bit 8 of the JC byte in rows 1-3 is used for indicating whether stuff or client data blocks are present in the next ODUflex frame. A “0” in this bit indicates that the next ODUflex frame will contain client data and a “1’ indicates that it will carry stuff blocks. Majority voting across the three JC bytes provides protection against JC bit errors. A 257b stuff block consists of an all-zero value corresponding to the 256B/257B transcoding of four invalid 64B/66B blocks. This approach ensures that transmission channel errors will not lead to a stuff block accidentally being trans-decoded into valid 64B/66B blocks at the sink.

One of the implications of maintaining a common mapping method pertains to decoding the RS(544,514) FEC used on the y00GBASE PCS interface at the OTN mapper. IEEE 802.3 agreed to provide an FEC Signal Degrade parameter (am_sf) in the AM field of y00GBASE (y > 1) PCS interfaces. When the FEC is terminated at the OTN mapper, the mapping must provide a means of communicating a received FEC SD condition to the OTN demapper so that the indication can be encoded into the egress 200GbE/400GbE/800GbE stream. This client degrade indication (CDI) is carried in PSI byte 2 of the OPUflex overhead, as illustrated in Figure 9-9.

Figure 9-9. 800GBASE Client Signal Degrade Communication Over OTN



The CRC-32 field at the end of OPU payload frame covers the entire OPU frame payload area except the 6-bit pad. A bad CRC means an uncorrectable error has impacted the client signal along the OTN path. The demapper has the option of using the CRC results to perform error marking by decoding all the 257b blocks in the OPUflex frame such that transcoding a 257b block will result in four 64B/66B /E/ blocks. Error marking will improve the Ethernet client's mean time to false packet acceptance (MTTFPA) performance.[44]

9.5.2. Considerations for Flexible Ethernet (FlexE) Client Signals

During the time when the ITU-T was developing the B100G standard, the OIF (Optical Interworking Forum) was defining a new interface called Flexible Ethernet (FlexE) as a mechanism to decouple the Ethernet MAC and Physical Medium Dependent (PMD) sublayers, especially in terms of rates. See Appendix D: References and Standards Related to OTN for a link to a FlexE tutorial.

FlexE defines a CBR signal that is constructed as a FlexE Group carrying one or more FlexE client signals, with each FlexE Group consisting of a set of 100GBASE-R, 200GBASE-R or 400GBASE-R PHYs, all sharing a common PHY clock source. The FlexE signal carried on each PHY is structured as a round-robin repeating set of 5 Gbit/s "Calendar slots" for 64B/66B characters. The FlexE client signals are Ethernet MAC flows that are mapped into one or more Calendar Slots.

There are three options for carrying the FlexE information over OTN:

1. Terminate the FlexE and carrying the FlexE clients as Ethernet clients over OTN (either by using ODUflex(IMP) or by using the associated specified CBR mapping).
2. Simply treat each FlexE Group PHY as a 100GBASE-R, 200GBASE-R or 400GBASE-R signal and transport it accordingly.
3. Exploit calendar fill information within the FlexE stream, as explained below, to allow carrying a lower rate signal over OTN.

Of course, in the first option the FlexE signal itself is not actually transported over OTN. The second option is referred to as the "FlexE unaware" mapping, since the OTN is not required to know that the 100GBASE-R, 200GBASE-R or 400GBASE-R signal is carrying FlexE. The last option is known as a "FlexE aware" mapping and is described below.

The FlexE feature exploited by the FlexE aware mapping is the following. If it is known, or decided, that the clients carried in a FlexE Group do not require the entire Group capacity, the sub-calendars can be provisioned with the unneeded calendar slots marked as unavailable. A FlexE PHY can discard the Unavailable calendar slots in order to use a lower bit rate (e.g., in order to meet an optical reach objective for that PHY). A FlexE aware OTN node can discard the characters from the unavailable FlexE calendar slots during the mapping process in order to construct a lower rate ("crunched") CBR client signal. Since the number of unavailable calendar slots is static, the resulting "crunched" FlexE signal is still a CBR stream that can be mapped into OTN as a partial rate signal with respect to the ingress FlexE signal. Hence, the OTN transport bandwidth requirement for the FlexE signal can be optimized for the amount of bandwidth actually required for the set of constituent FlexE clients. The OTN demapper restores the unavailable calendar slots when it creates the egress FlexE signal.

Bit-synchronous GMP (BGMP) is used for mapping the FlexE signal into an ODUflex(IMP). BGMP generates the C_m values deterministically (e.g., based on a sigma-delta algorithm) rather than deriving them from the input client rate and buffer fill.

9.5.3. Sub-rate OTUCn (OTUCn-M)

The distance over which a signal can be transmitted is a function of the signal rate. Power consumption is also a function of the signal clock rate. Consequently, there will be applications where it's desirable to transmit a B100G signal at a rate less than the discrete *N*×OTUC rate in order to achieve the desired distance/reach for that channel. Such applications could include interconnections between two routers where the packet flow peak rate is less than the *N*×OTUC

---

[44] The disadvantages of using this option are the requirement for a buffer large enough to store the entire OPU frame and the associated latency. Since the FlexO frame is protected by FEC, the value of this option is unclear.

rate, or the interconnections between two OTN cross-connects where the required capacity is less than the N×OTUC rate. It could also address situations in which fiber impairments limit the capacity on a per link basis to less than the full N×OTUC rate. For such applications, the B100G signal definition includes the option of transmitting a signal that has the full set of OTUCn/ODUCn overhead but has an OPUCn consisting of only the active Tributary Slots. Specifically, an OTUCn-M signal consists of n copies of the OTUC, ODUC and OPUC overhead, and M of the 5Gbit/s TS. Since the overhead and TS each occupy 16 bytes of a row, and there are 3808/20/16 = 11.9 bytes/TS/row, the minimum OTUCn-M rate can be calculated as follows:

$\text{OTUCn-M rate} = (\text{OTUCn rate})(\text{OTUCn-M row size}/\text{OTUCn row size})$

$= (\text{OTUCn rate})[(16)(n + 11.9M)]/[(16\text{n})(1 + (11.9)(20))]$

$\text{OTUCn-M rate} = (\text{OTUCn rate})[(n + 11.9M)]/(239n)$

The availability of TS on an OTUCn-M interface is indicated in the OPUCn MSI fields. The specific values of M are a vendor-specific choice. Since OTUCn-M is a single-vendor interconnect application, the specific format of the transmitted signal (e.g., the manner in which the active TS are interleaved into a frame format) is not defined in G.709.

10. Flexible OTN (FlexO)

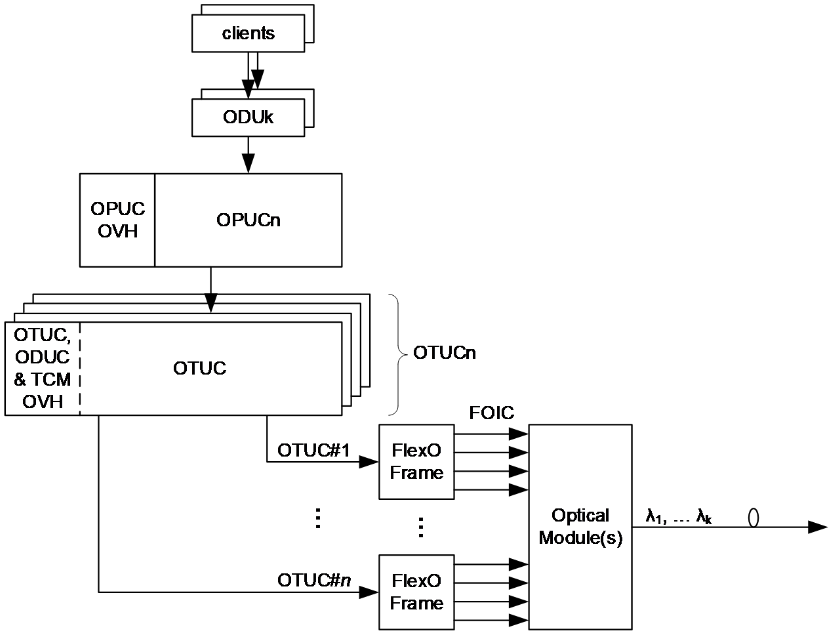
Q11/15 chose to map the OTUCn signals into a newly defined modular flexible OTN (FlexO) PHY format rather than directly using the OTUCn signal as the PHY interface signal in the same manner as the OTUk. The FEC was moved from the OTU frame to the FlexO frame, with the OTUCn stream being mapped into the payload area of a stream of FlexO FEC codewords. Following an approach that had already become common with single-vendor applications, the alignment between the OTUC frame and the FEC codeword stream is arbitrary[45]. This approach allowed modifying the FlexO frame for different FEC options for different rate and reach applications, as explained below. The flexibility of the FlexO signal format has led to it being adopted by OIF, IEEE 802.3 and OpenROADM for use with long-reach *x* × 100G interfaces.

FlexO has conceptual similarities to the OIF FlexE in that FlexO is a modular interface consisting of a set of 100Gbit/s optical PHY streams that are bonded together to carry an OTUCn. For example, with 100 Gbit/s PHYs, a set of *n* 100Gbit/s PHYs are bonded together to carry an OTUCn, with each 100 Gbit/s PHY carrying an OTUC slice. This flexibility allows using any value of "*n*" for an OTUCn interface rather than defining only certain discrete values of *n* (e.g., just *n* = 4 and 10). Further, if for example a set of *m* 100Gbit/s PHYs are available, as a subset of *n* PHYs can be chosen to carry an OTUCn (*n*<*m*), which would allow choosing the subset of PHYs with the best optical channel characteristics or carrying multiple OTUCn signals over a set of PHYs.

As explained below, the short-reach version of FlexO takes advantage of being able to use existing 100GbE/OTU4 optical modules for the individual FlexO 100 Gbit/s PHYs, thus benefiting from the lower cost of these existing higher volume 100GbE/OTU4 optical modules. As future higher rate Ethernet modules became available (e.g., 200 Gbit/s, 400 Gbit/s and 800 Gbit/s PHYs), FlexO was extended to also make use of them. FlexO makes partial reuse of the lane architecture and FEC structure from 100GbE, 400GbE and 800GbE Ethernet in order to also leverage Ethernet concepts.

The common properties across FlexO interfaces are defined in G.709.1. The figure below illustrates the flow of mapping relationships for inserting a client signal into FlexO.

Figure 10-1. FlexO Client Mapping Relationships



---

[45] This approach was typical for strong soft-decision FEC coding for very long-reach.

The general text format used to describe a FlexO signal is:

FlexO-(n/ne/x/xe)-<int>-m.

The fields following “FlexO” are defined as follows:

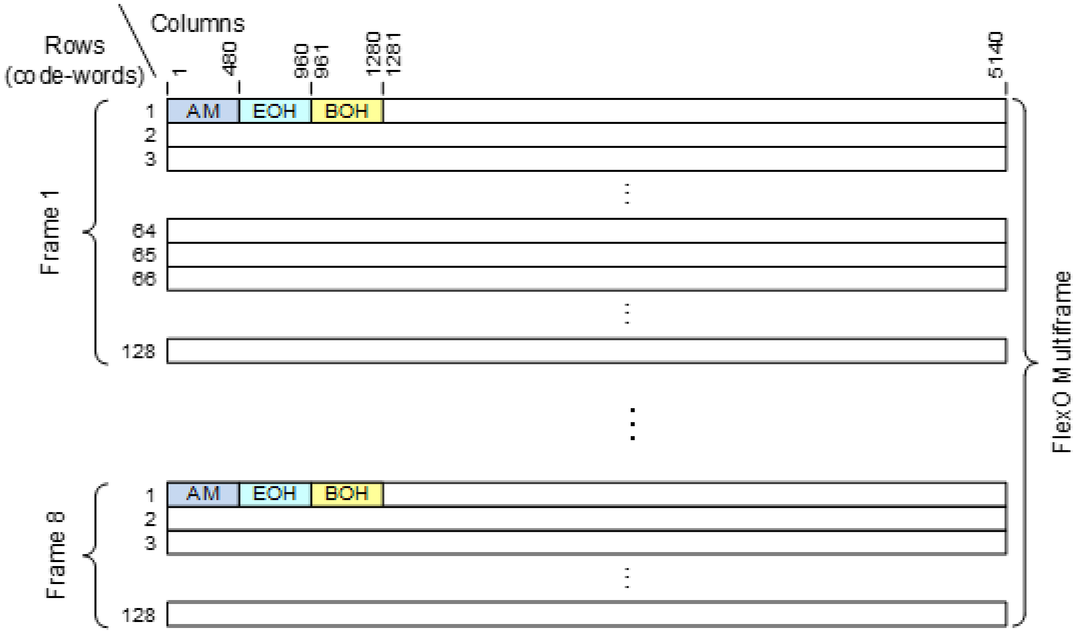
- FlexO instance: A 100G FlexO-1 or FlexO-1e.
- FlexO-n: A FlexO information structure with *n* FlexO instances within the FlexO group used to carry one or more OTUCn client signals
  - The FlexO instances have a bit rate that allows them to carry a 100G OTUC.
- FlexO-ne: A FlexO information structure with *n* FlexO instances within the FlexO group used to carry one or more *y*00Gbit/s Ethernet client signals (*n* = *y*).
  - The FlexO instances have a bit rate optimized for carrying 100 Gbit/s of the *y* × 100 Gbit/s Ethernet signal.
  - Currently *y* = 1, 2, 4 and 8, with 16 in development in IEEE 802.3.
- FlexO-x: A FlexO information structure where x represents the number of interleaved FlexO instances. Hence, x indicates the FlexO-x interface bit rate in 100G increments.
  - e.g., x=1 for 100G, x=2 for 200G, x=4 for 400G, etc.
  - For use in carrying client signals (e.g., OTUCn or Ethernet) over a group of *m* instances $(m = \lceil (n/x) \rceil)$.
- FlexO-xe: A FlexO information structure where x represents the interface bit rate in 100G increments where the 100G bit rate is optimized for carrying *y*00Gbit/s Ethernet client signals.
- FlexO-n(e) and FlexO-x(e) use (e) to indicate that the text describes both the OTN and Ethernet rate optimized versions of the FlexO signal.
- -<int> indicates a specific short or long-reach interface type with respect to FEC or a combination of the FEC and modulation type (for an example, refer to Short-Reach FlexO Interfaces).
- -m indicates the number of interfaces in the FlexO group.
  - Note that m has other meanings within different contexts.

One or more clients are multiplexed to a FlexO-n(e) group, which can be inverse multiplexed across a FlexO-x(e)-<int>-m group interface where n ≤ (m × x). For example, if x=2 and m=2, the FlexO-x(e) consists of a two 200G interface and is capable of carrying a FlexO-n(e) with rate ≤400G (i.e., n ≤ 4). For scenarios where the FlexO-n(e) does not fill the entire FlexO-x(e)-<int>-m (i.e., n < (m × x)), some instances are marked as unequipped.

10.1. FlexO Frame Format

The FlexO frame structure, as shown in the Figure below, consists of 5140 single-bit columns and 128 rows. For short-reach interfaces, each 5140-bit row is the payload area of an FEC codeword. There are eight frames in the FlexO multiframe. The FlexO overhead fields and payload structures are described below.

Figure 10-2. FlexO Frame and Multiframe Illustration



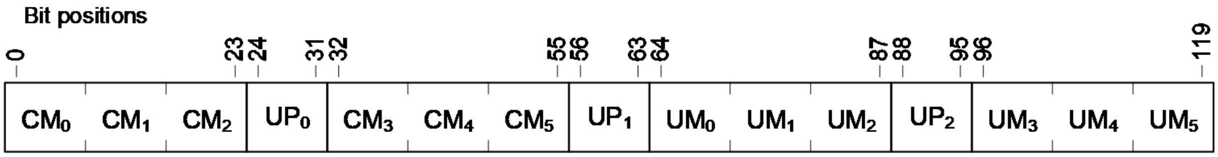
10.1.1. Alignment Mechanism

As shown in Figure 10-2, in the FlexO frame for each instance, the first 5140-bit row of the frame includes a 1280-bit overhead area that begins with an Alignment Markers (AM) field. The 480-bit AM field is followed by an 800-bit FlexO frame overhead area, which is divided into a 480-bit Extended Overhead (EOH) field and a 320-bit Basic Overhead (BOH) field.

The AMs, which are reused from the equivalent rate IEEE 802.3 interfaces, allow the receiver to correctly identify and order the FlexO instances when the interface consists of multiple instances and allows the receiver to perform deskew across the instances. The 480-bit AM field contains a set of four 120-bit AMs. As shown in in the figure below, each AM consists of a set of fields that are common to all AMs (CM_i) and two sets of fields (UP_i and UM_i) that are unique to the AM being used on each logical lane. See Table 10-1 for the CM_i values of the AMs.[46] The sets of values used for the UP_i and UM_i fields are interface specific.

Note: The FlexO frame structure is derived from 100Gbit/s Ethernet clause 91 [IEEE 802.3] FEC alignment and lane architecture, without any 66b alignment or 256b/257b transcoding functions.

Figure 10-3. Alignment Marker Structure



---

[46] Like most data communications protocols, IEEE 802 specifies bytes as being transmitted LSB first. Since OTN follows telecommunications convention, it transmits MSB first. Consequently, G.709.1 translates the 802.3 hexadecimal AM values in order to reflect the MSB-first transmission order. For convenience, the table here shows both conventions for the fields that are common to all AMs.

Table 10-1. FlexO and equivalent IEEE 802.3 Alignment Markers

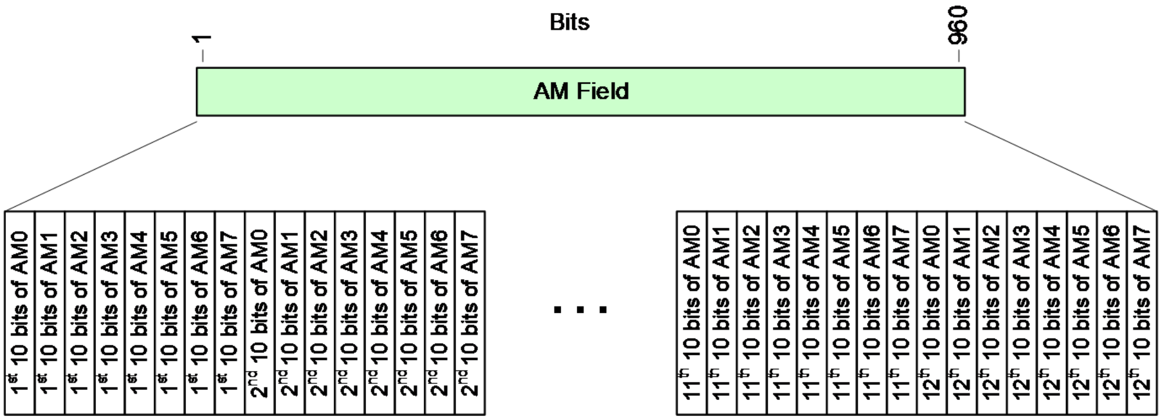
Field	CM0	CM1	CM2	UP0	CM3	CM4	CM5	UP1	UM0	UM1	UM2	UP2	UM3	UM4	UM5
Eth. AM	9a	4a	26	**	65	b5	d9	**	**	**	**	**	**	**	**
FlexO AM*	59	52	64	**	a6	ad	9b	**	**	**	**	**	**	**	**

* Note that the corresponding Ethernet and FlexO AM values are identical in how they appear on a serial interface. This table translates between the respective IEEE 802.3 and ITU-T approaches to expressing the bytes in hexadecimal format for the CM_i fields.

** The values in these bytes are unique for each AM.

The four 120-bit AM fields of each FlexO instance contain the AM values corresponding to the four 25G logical lanes associated with that 100G FlexO-1 instance. For FlexO-x frames with *x*>1, the constituent *x* FlexO frames are aligned and interleaved on a 10-bit round-robin basis such that the resulting FlexO-x frame begins with an *x* × 120-bit AM field, as illustrated in Figure 10.4. The four logical lanes of each FlexO instance are thus interleaved with a 10-bit granularity, beginning with the first 10 bits of each frame row such that the appropriate AM value will appear in each logical lane after 10-bit symbol distribution to the lanes.[47] The Figure below shows an example of the AM interleaving into a FlexO-2 frame. See clause 9 of G.709.5 for the AM values used for FlexO-1, FlexO-2, FlexO-4 and FlexO-8.

Figure 10-4. Example AM interleaving for FlexO-2



10.1.2. FlexO-x(e) Payload Structure and Rates

The FlexO-x(e) for x > 1 is constructed from *x* × 64 rows by 10280 single bit columns[48]. As shown in Figure 10-5, it consists of the interleaving of *x* frame and multiframe-aligned FlexO-1 frames (see Figure 10-2) that are interleaved in order from lowest to highest identification (IID). See G.709.3, G.709.5 and G.709.6 for the *z* value used with the respective interfaces[49].

As explained above, OTUCn uses a 128-bit modularity for its frame overhead and payload Tributary Slots. Consequently, it was desirable to have the OTUC information begin on a 128-bit boundary in the first FEC codeword of the FlexO frame and have each 128 row FlexO frame contain an integer number of 128-bit OTUC blocks. As shown in Figure 10-2, this initial 128-bit alignment is enabled by having a 320-bit FlexO overhead field immediately following the 960-bit AM field, creating a 1280-bit combined field. Since the 5140-bit FEC codeword payload is not divisible by 128, the alignment of

---

[47] This 10-bit granularity corresponds to the 10-bit FEC symbol size of the RS(544,514,10) used for the short-reach FlexO interface.

[48] $x \times 64 \times 10280 = x \times 128 \times 5140$

[49] $z = 10$ for G.709.5 and G.709.3, and $z = 128$ for G.709.6

the codeword and 128-bit block boundaries shift in each successive codeword of the frame. As explained below (see Figure 10-7), the BMP mapping uses 1280-bit FS at the beginning of frames 1-7 in order to have a convenient integer number of 128-bit words in the FlexO multiframe. Since OTUCn was the original client for FlexO, the BMP mapping determined the FlexO rate. Looking at the FlexO multiframe from Figure 10-5, the resulting FlexO-1 rate can be calculated as follows. Excluding the FEC overhead, we have:

Bits/multiframe = (8 frames)(128 codewords)(5140 bits/codeword) = 5263360 = 4112 × 1280 bits

Taking the 1280-bit overhead per frame and the 1280-bit BMP fixed stuff in frames 1-7, the FlexO-1 payload capacity becomes:

Payload/multiframe = 5263360 – (8 + 7)(1280) = 5244160 = 4097 × 1280 payload bits

Hence the resulting FlexO-1 rate with the BMP mapping, excluding the FEC overhead is:

FlexO-1 rate = (4112/4097)(OTUC rate) = (4112/4097)(239/226)(99.5328 Gbit/s) = 105.64351 Gbit/s

See Table 10-2 for the nominal FlexO-n(e) bit rates.

Figure 10-5. FlexO-x Frame and Multiframe Illustration

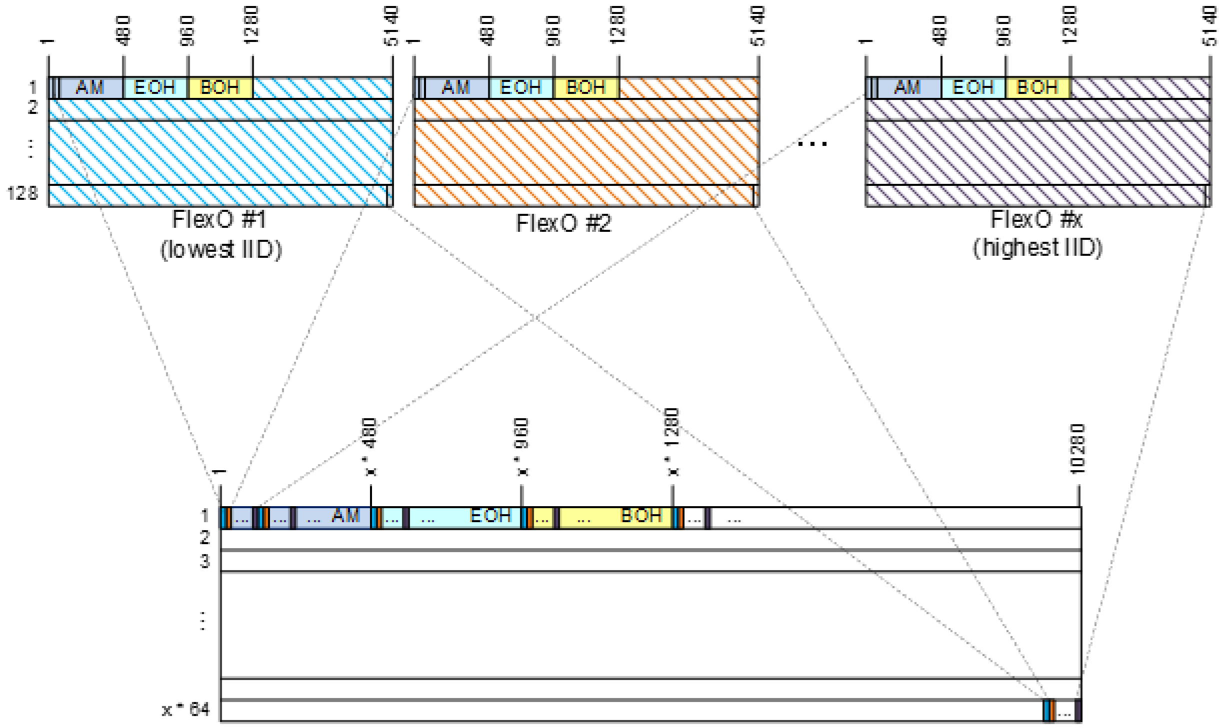


Table 10-2. FlexO-n(e) Bit Rates

Interface Type	Nominal bit rate	Bit rate tolerance
FlexO-n	≈ n × 105 643 510.782 kbit/s	±20 ppm
FlexO-ne	≈ n × 100 622 438.327 kbit/s.	±20 ppm
Note: The actual FlexO-ne rate is derived from a 156.25 MHz clock that is commonly available in Ethernet systems. Specifically, the FlexO-ne rate is n × 511/512 × 514/544 × 1445/1624 × 766 × 156 250 kbit/s.		

10.1.3. FlexO Scrambling

Similar to the OTUk frame, the FlexO signal is scrambled with a frame-synchronous scrambler prior to transmission. The scrambler, which uses the $x^{16} + x^{12} + x^3 + x + 1$ generator polynomial, is reset to 0xFFFF (i.e., all 1s) on the first transmitted bit of the FlexO-1 frame and advances during each subsequent bit. As illustrated in Figure 7-6 for the OTUk case, each FlexO-1 frame bit is replaced by the modulo 2 sum (XOR) of itself and the current MSB value of the scrambler. The AM and FEC check symbol values subsequently overwrite the scrambled bits in their respective fields prior to transmission.

10.2. FlexO Overhead

The FlexO overhead functions are associated with the FlexO PHY operations, including bonding multiple OTUC1 members. The overhead format is divided into basic and extended overhead, illustrated in Figures Figure 10-2 and Figure 10-5 .

10.2.1. Basic Overhead (BOH)

As explained above, the basic overhead (BOH) field is 320 bits (40 bytes) long. For increased field capacity, the BOH area is time shared across an 8-frame overhead multiframe, determined by the three LSBs of its MFAS field that increments every frame. This results in a 2560-bit BOH field, as shown in Figure 10-6. The field definitions and functions can be summarized as follows:

STAT is similar to the OTUCn STAT in that it provides status information about the PHY connection. The only defined STAT function at this time is to report a far end (remote) PHY failure (RPF) condition.

GID (Group Identification) is the number indicating the interface group instance (i.e., the group of PHYs) of which a FlexO-x(e)-<int> is a member. The GID allows the receiver to check whether the interface belongs to the intended FlexO group. (GID = 0 indicates that a PHY is not part of any FlexO group, i.e., is unequipped.) The same GID value is used in both directions.

IID (FlexO Instance Identification) The 8-bit IID field uniquely identifies each of the m interfaces (members) of a FlexO-x(e)-<int>-m interface group, including their order within the group so that they can be correctly reordered at the receiver. The lowest IID value in the group corresponds to the first FlexO-x(e)-<int> in the group. The order of the members in the group is based on increasing IID values, although it is not necessary to arrange the IID values consecutively (sequentially). The same IID value is used in both transport directions.

MAP The MAP field indicates which specific members are used by that FlexO group. For example, since the IID values are not required to be sequential, the MAP confirms to the receiver which IID values it should be receiving for members of that group. The MAP field contains one bit corresponding to each of the 256 possible IID values, and a “1” in a bit position indicates that the corresponding IID is being used by this group. In other words, the “1” bit positions within the MAP correspond to the IID set for the members of the FlexO-x(e)-<int> interface, and the lowest numbered IID uses the highest numbered MAP location within the group.

AVAIL indicates the number of “available” and valid OTUC slices that are mapped into the FlexO frame for that PHY. This field is not especially useful with 100Gbit/s PHYs, since AVAIL can only be “0” or “1.” However, higher rate PHYs (e.g., 200Gbit/s and 400Gbit/s) allow having only a subset of the PHY’s capacity used for carrying actual OTUC slices. AVAIL then tells the receiver which portion of the PHY capacity is in use.

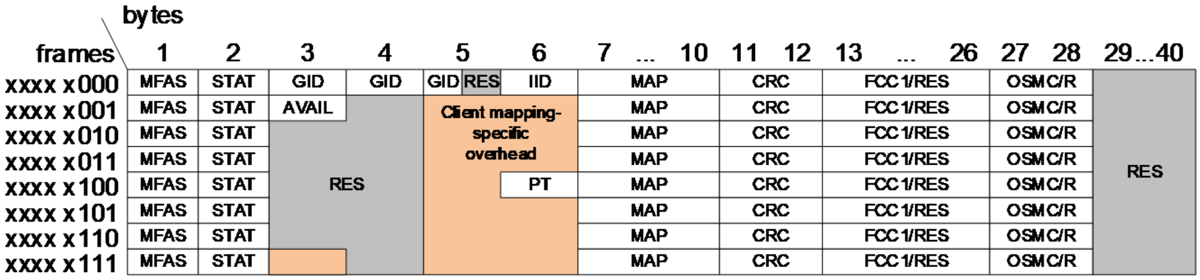
CRC A simple persistency check could have been used for the FlexO overhead, since most of the fields contain static provisioned values and the receiver doesn’t need to respond to the overhead in real-time. However, a CRC was added to cover the fields in columns 2-10 of Figure 10-6, since a CRC can provide a simpler method of regularly confirming the integrity of the received overhead values than performing frequent persistency checks.

FCC1 (FlexO Communications Channel) is included for the purpose of managing the interface. It is not a generic/general purpose communications channel like the OTN GCC*x* channels. The FCC channel rate is slightly less than 18Mbit/s.

OSMC (OTN Synchronization Message Channel). The OSMC is only active on the first member of the FlexO group (i.e., the member with the lowest IID value). The OSMC function is described in the OTUk Overhead Fields section.

PT (Payload Type) specifies the composition of the FlexO payload area. It indicates the type of client being carried (e.g., OTUCn, Ethernet, or type of maintenance signal[50]), and the mapping method being used (BMP or GMP).

Figure 10-6. FlexO Basic Overhead Illustration

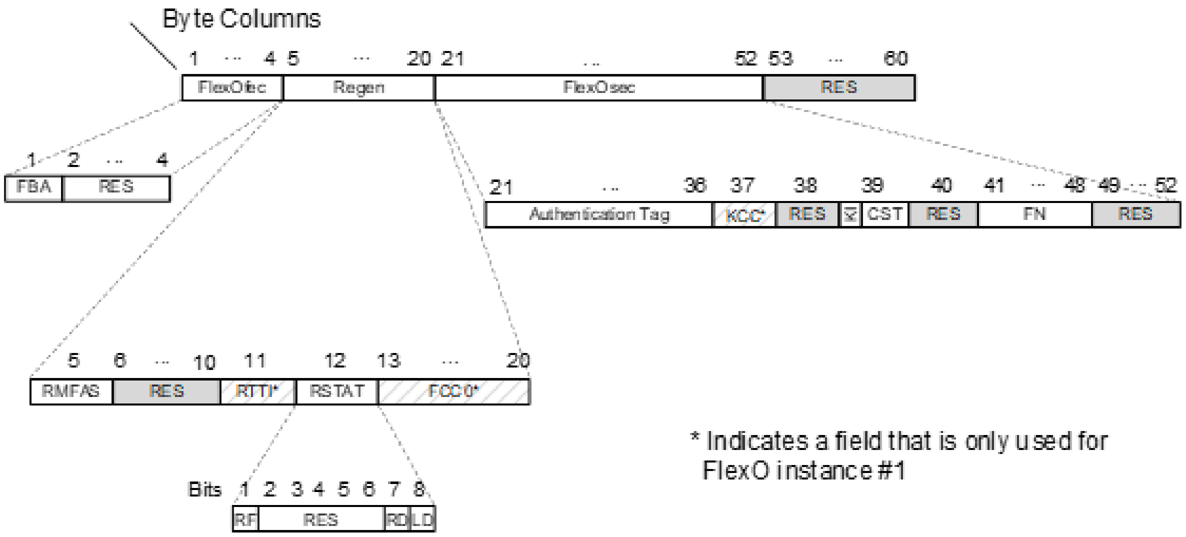


frames \ bytes	1	2	3	4	5	6	7 ... 10	11 12	13 ... 26	27 28	29...40
xxxx x000	MFAS	STAT	GID	GID	GID RES	IID	MAP	CRC	FCC1/RES	OSMC/R	RES
xxxx x001	MFAS	STAT	AVAIL	RES	Client mapping-specific overhead		MAP	CRC	FCC1/RES	OSMC/R	
xxxx x010	MFAS	STAT					MAP	CRC	FCC1/RES	OSMC/R	
xxxx x011	MFAS	STAT					MAP	CRC	FCC1/RES	OSMC/R	
xxxx x100	MFAS	STAT				PT	MAP	CRC	FCC1/RES	OSMC/R	
xxxx x101	MFAS	STAT					MAP	CRC	FCC1/RES	OSMC/R	
xxxx x110	MFAS	STAT					MAP	CRC	FCC1/RES	OSMC/R	
xxxx x111	MFAS	STAT					MAP	CRC	FCC1/RES	OSMC/R	

10.2.2. Extended Overhead (EOH)

The FlexO EOH fields occupy the 60 bytes (480 bits) between the AM and BOH fields. The EOH fields, illustrated in Figure 10-7, repeat every frame rather than being time-shared across the FlexO multiframe. As the name implies, the EOH supports functions that are specific to certain FlexO interfaces or applications. The functions defined at the time of this tutorial include identifying the FEC block alignment for G.709.3 long-reach FlexO interfaces, Flex security (FlexOsec) overhead, and overhead to support using FlexO in regenerator applications. Each of these is described below.

Figure 10-7. FlexO Extended Overhead Illustration



---

[50] PT code values are reserved for use by OIF 800ZR signals, which reuse the FlexO-8 frame format.

FBA (FEC Block Alignment) is used with the G.709.3 long-reach interface (see Appendix B below) to indicate the alignment of the FEC code blocks within the FlexO multiframe.

Regen (Regeneration application overhead) is used when FlexO is used in regenerator applications rather than point-to-point Section layer applications. See the Regen Application section for a description of this application. The 16-byte Regen overhead consists of:

- A regenerator application MFAS (**RMFAS**) byte for timesharing other Regen overhead fields
- A Regen Trail Trace Identifier (**RTTI**) byte
- A Regen status (**RSTAT**) byte with bits to indicate Remote Fault (RF), Remote Defect (RD) and Local Defect (LD)
- **FCC0** (FlexO Communications Channel) is included as a generic FlexO regen management channel per FlexO-x(e)-<int> interface. Its bandwidth is approximately 10 Mbit/s.

FlexOsec (FlexO security) fields to enable secure communication of the FlexO interface, as described below in the separate OTN Security section.

10.3. Client Mapping into FlexO-n

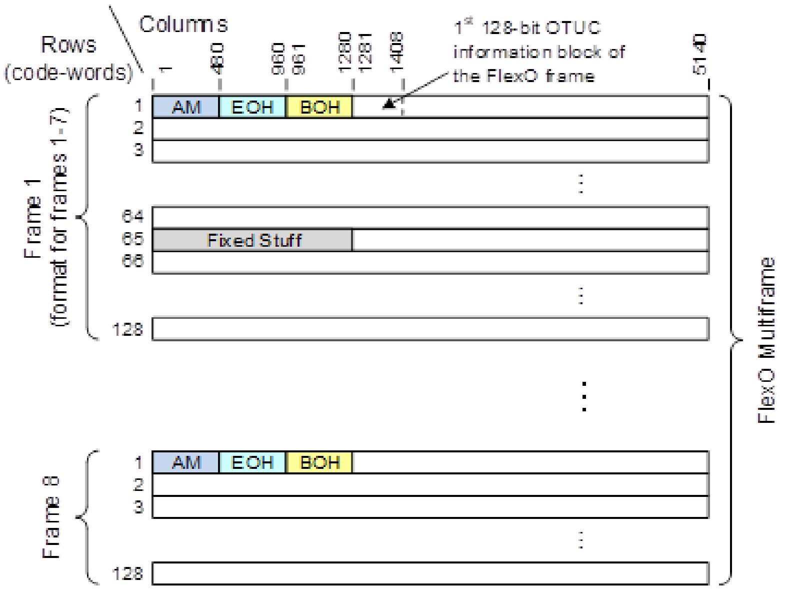
The BMP and GMP mapping methods are defined in this section, followed by a data flow illustration for OTUCn and y00GBASE Ethernet clients.

10.3.1. BMP Mapping of OTUCn into FlexO-n

This mapping was the original FlexO application. The OTUCn signal is mapped into the FlexO-n payload area with each OTUC mapped directly into a FlexO instance (i.e., there is a one-to-one relationship between an OTUC and FlexO instance). Since FlexO was initially designed as the PHY layer for OTUCn clients, BMP was the natural mapping choice. The OTUC occupies the entire FlexO frame payload area, minus the FlexO overhead and fixed stuff bytes. Consequently, the FlexO bit rate is determined directly by the OTUC rate, and no dynamic rate justification is required.

As illustrated in Figure 10-8, the fixed stuff bytes for the BMP mapping are located in the first 1280 bits of row 65 in frames 1-7 of the 8-frame FlexO multiframe. As is also noted in the figure below, the OTUC data is mapped into the payload in 128-bit words. These 128-bit words align to the 128-bit structure of the OTUC frame.

Figure 10-8. Illustration of the FlexO Payload Area with BMP



The payload area of the FlexO multiframe is divided into 128-bit blocks, which are aligned to the start of the FlexO payload area following the 1280-bit set of AM, EOH and BOH fields. While there is an integer number of 128-bit payload blocks within the payload area of a FlexO frame, the payload area of each FlexO frame row is not evenly divisible by 128. Consequently, the last block of most rows will cross over into the next row. Taking into account the eight overhead and seven fixed stuff appearances, the FlexO multiframe contains:

((8 × 128 × 5140) – ((7+8) × 1280) bits/MF) / (128 bits/block) = 40970 payload blocks

As is also noted in Figure 10-8, the OTUC data is mapped into the 128-bit blocks of the FlexO payload area in 128-bit words. These 128-bit words align to the 128-bit structure of the OTUC frame.

Due to the relative lengths of the OTUC frame and FlexO multiframe payload area[51], the OTUC frame structure floats within the FlexO frame (i.e., there is no fixed positional relationship between the OTUC and FlexO frames).

10.3.2. GMP Mapping

The GMP mapping was introduced into FlexO to accommodate both clients with clock tolerances outside the FlexO range (i.e., Ethernet with its ±100ppm range) and to enable the multiplexing of multiple clients into the FlexO-x(e) payload area where each client may have originated in a different source timing domain.

As explained below, the GMP frame payload area P_{server} is 10260 blocks for the OTUC mapping into FlexO-1 and 10220 blocks for the Ethernet mapping into FlexO-1e. Consequently, the same 14-bit C_m coding as the OTN OPUk is used here (see Figure 8-4). As illustrated below in Figure 10-10 and Figure 10-16, the y00GBASE Ethernet client mapping into FlexO-ne uses m = y × 257 bits and for OTUCn mapping into FlexO-n uses m = n × 256 bits.

As illustrated in Figure 10-6, FlexO client mapping specific overhead is primarily carried in overhead columns 5 and 6 of the 40-byte BOH field. The Figure below illustrates the mapping overhead associated with mapping clients into FlexO-n(e), including the GMP overhead. The overhead added for the mapping is:

JC1 – JC3: This GMP justification control overhead is identical to what is used with OPUk (see Figure 8-4 and Table 8-2 above in GMP for Multiplexing).

JC4 – JC6: The 5-bit GMP phase offset encoding ($\sum C_{nD}$) justification control overhead. See the GMP for Multiplexing section for the description of $\sum C_{nD}$.

MSI[*x*]: Mapping Structure Indicator. For FlexO-ne it indicates the Ethernet client content of each FlexO instance payload. Specifically, the Occupied bit (Occ) indicates whether that instance is allocated or not, and the associated Tributary Port if it is allocated.

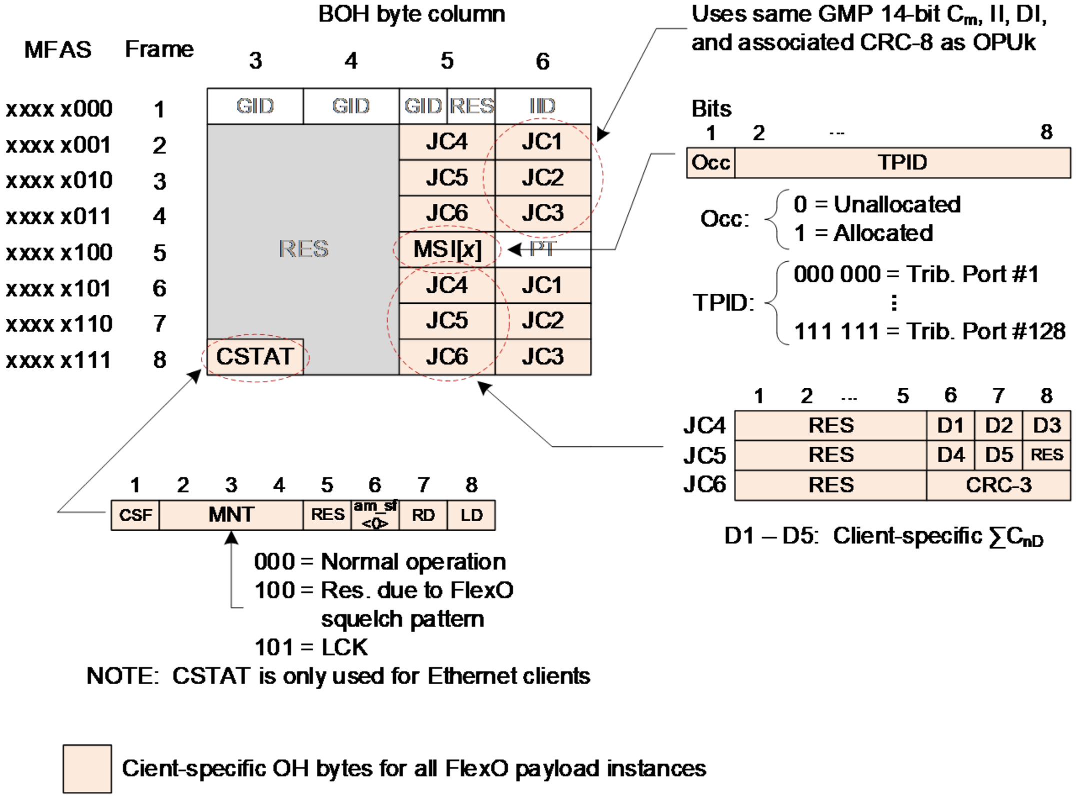
TPID: Tributary Port Identifier is the tributary port number of the Ethernet client carried in this FlexO instance. Note that an all-zero TPID is used when Occ = 0 (unallocated). If Occ = 1 (allocated), an all-zero TPID indicates tributary port #1.

CSTAT: Client Status, including client signal fault and maintenance conditions. The am_sf<0> bit communicates the summary ingress client am_sf<2:0> signal degrade condition for y > 1. The CSTAT field is only used with Ethernet clients.

---

[51] The bit ratio between the FlexO frame and OTUC client is 4112/4097.

Figure 10-9. Mapping Specific Overhead for Client Mapping with GMP



10.3.2.1. GMP Mapping and Multiplexing y00GBASE Ethernet into FlexO-ne

The y00GBASE Ethernet signals have become increasingly common as clients transported over FlexO. Consequently, a direct mapping and multiplexing of y00GBASE Ethernet clients (y = 1, 2, 4, 8) into the FlexO payload area was added to G.709.1, thus avoiding the added bit rate and complexity associated with first mapping the Ethernet client into an ODUflex and subsequently into an OTUCn. Using GMP for this mapping allows carrying one of more separate Ethernet clients (with ≥ ±100 ppm bit-rate tolerance) in n instances of a FlexO-ne.

The mapper recovers the 257b Ethernet block stream prior to mapping by bit de-interleaving the lanes of the client signal into the 25G logical lanes and recovering the lane AMs. Received uncorrectable FEC codewords are replaced with error control blocks transcoded into 257b blocks. At that point, the process details vary for the different Ethernet rates:

- 100GBASE-R: Discard the BIP counters used in the 100GBASE-R AMs. Descrambling is not performed since scrambling is performed by the Ethernet source prior to the 257b transcoding.
- 200GBASE-R and 400GBASE-R: The 3-bit AM status field (am_sf<2:0>) is extracted, since its function is to indicate the degrade status of the now-terminated Ethernet PCS FEC. The 257b blocks are then descrambled. As noted below, the am_sf<2:0> information is communicated across the FlexO connection in the CSTAT overhead so that the appropriate signal status information can be inserted into the egress Ethernet signal.
- 800GBASE-R: The 3-bit AM status field (am_sf<2:0>) is extracted from the 257b streams of both 400G flows and ORed together. The two 257b streams are then descrambled without AMs. As noted below, the am_sf<2:0> information is communicated across the FlexO connection in the CSTAT overhead so that the appropriate signal status information can be inserted into the egress Ethernet signal.

The GMP process for mapping one or more y00GBASE Ethernet clients into FlexO-ne follows the same general process as described above in the OTN Client Mapping and Multiplexing. The specifics of for mapping the y00GBASE Ethernet clients into FlexO-ne are described in this section.

The GMP frame portion of a FlexO-ne multiframe is illustrated in Figure 10-10. The FlexO-ne frame payload area of each FlexO instance is divided into 257b blocks. A 5-bit fixed pad is added immediately following the FlexO frame overhead in the first row of the FlexO frame in order to have an integer number of 257b block locations between the pad and the end of the row. Each subsequent row has 5140/257 = 20 257b blocks.

Similar to the OPUk justification overhead (see Figure 7-5), the GMP frame occupies four consecutive FlexO frames (i.e., MFAS bit 7-8 values 00 to 11).

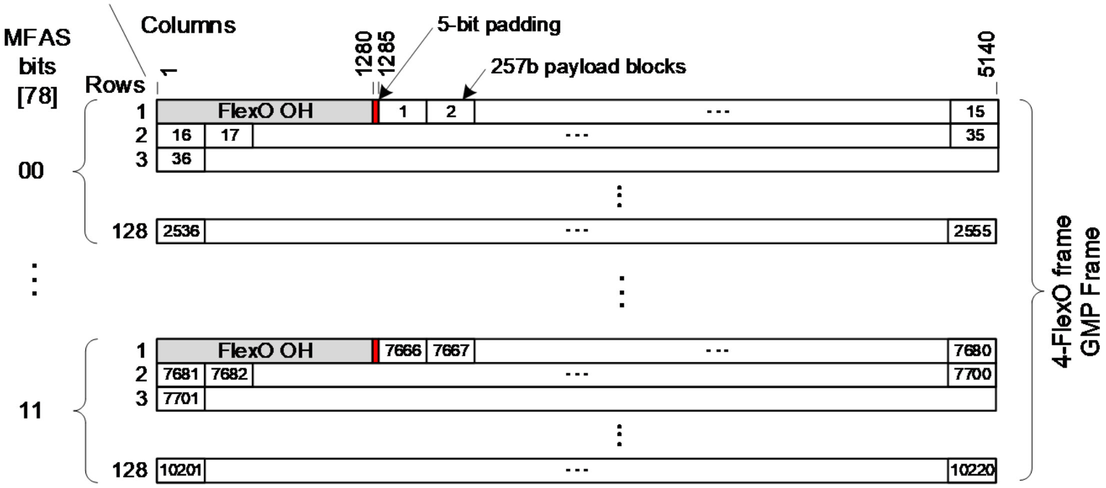
As can be seen from Figure 10-10, the Pserver the per instance is:

Pserver = (((5140 × 128) – 1285 bit/s frame) × (4 frames)) / (257 bits/block) = 10220 blocks

For the 14-bit Cm field, m = y × 257 bits.

The FlexO frames are aligned across the y FlexO payload instances of the FlexO-ne (i.e., multiframe aligned and phase-locked). This allows mapping successive groups of y 257-bit data or stuff blocks into the set of instances. This alignment also allows the GMP overhead of each instance to use identical Cm and ∑CnD in each frame. In other words, while the Cm is actually the number of y × 257b blocks, the alignment means that each FlexO instance can treat its GMP overhead as pertaining to the 257b blocks in its own payload area.

Figure 10-10. FlexO-ne payload area for a GMP frame



The processes are reversed at the FlexO demapper. For 200G, 400G and 800GBASE-R, the client AM status field information is carried across the FlexO interface in the CSTAT byte of the FlexO GMP overhead. At the FlexO demapper, the CSTAT information is merged with any degrade status associated with that FlexO link and inserted into the egress client signal AM status field so that degrade status covers the entire path up to that point. Note that the Ethernet Local Fault (LF) is used as the replacement stream for a failed client.

The GMP mapping overhead fields are illustrated in the figure above, with the JC byte details shown for ∑CnD and the JC1-JC3 bytes using the same format as OTN, as shown in Figure 8-4.

10.3.2.2. GMP Mapping OTUCn into FlexO-n

The GMP process for mapping one or more OTUC clients into FlexO-n is essentially the same as for mapping y00GBASE Ethernet clients into FlexO-ne, including the GMP frame occupying four consecutive FlexO frames (i.e., MFAS bit 7-8 values 00 to 11). The primary difference, as illustrated in

Figure 10-11, is that the payload area of the 4-frame GMP frame is divided into 256b blocks. Since 5140 is not divisible by 256, the blocks at the end of all but the last row wrap around into the next row. Note also from Figure 8-11 that the GMP mapping does not use the BMP fixed stuff bits shown in Figure 8-9. Using these bits for payload provides sufficient bandwidth to use GMP for the OTUC into FlexO without needing to increase the FlexO rate.

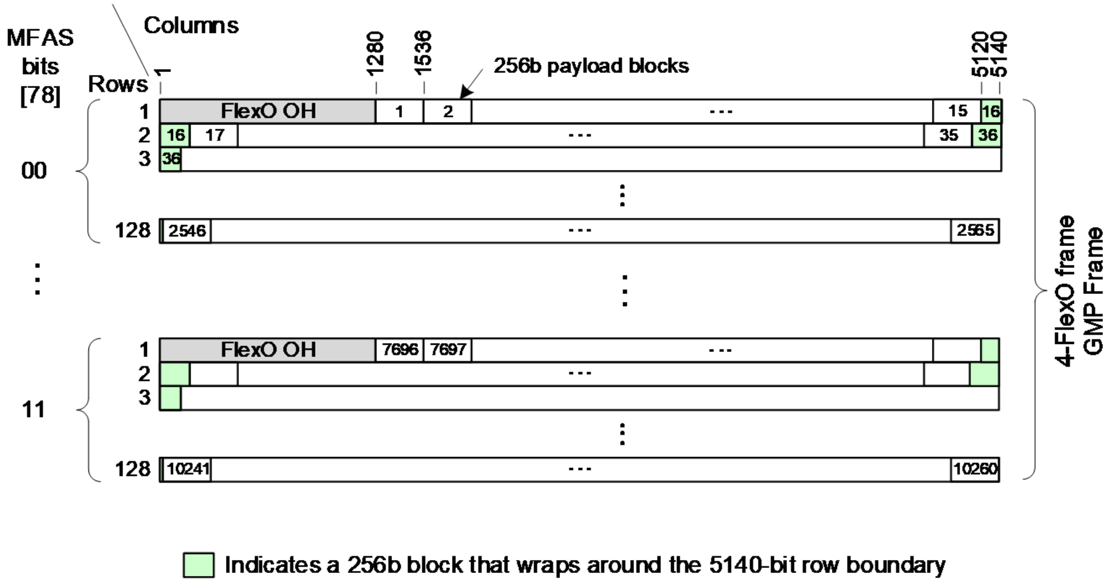
Each OTUC is associated with a FlexO instance (i.e., there is a one-to-one relationship between them). The MSI is used to indicate the OTUC content of each FlexO instance, including the associated TPID. An OTUC, which uses a 128-bit-oriented frame structure, is mapped into the FlexO instance payload in successive sets of two 128-bit data blocks. Consequently, each 256b block within the FlexO instance payload area (see Figure 10-11) is filled with either 32 bytes of the OTUC or a 32-byte GMP stuff word.

As can be seen from Figure 10-11, the P_{server} the per instance is:

P_{server} = (((5140 × 128) – 1280 bit/s frame) × (4 frames)) / (256 bits/block) = 10260 blocks

Here, for the 14-bit C_m field, m = n × 256 bits (32 bytes). As with the Ethernet mapping case, the FlexO frames are aligned across the group of n FlexO payload instances of the FlexO-n, which allows mapping the OTUCn as successive groups of n 256-bit data or stuff blocks into the set of instances. This alignment also allows the GMP overhead of each instance to use identical C_m and $\sum C_{nD}$ in each frame. In other words, while the C_m actually the number of n × 256b blocks, the alignment means that each FlexO instance can treat its GMP overhead as pertaining to the 256b blocks of the OTUC in its own payload area.

Figure 10-11. FlexO-n Payload Area for a GMP Frame



10.4. Short-Reach FlexO Interfaces

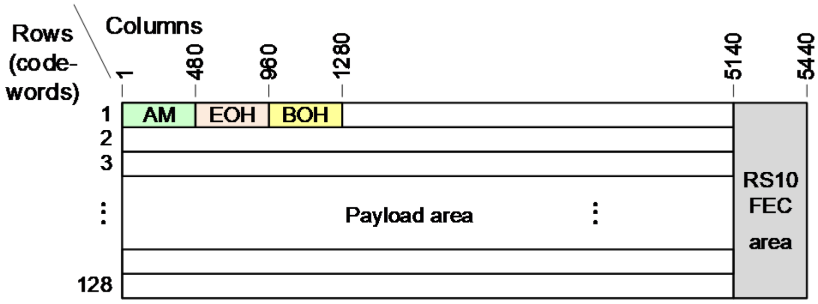
The IEEE 802.3 “KP4” Reed-Solomon RS(544,514,10) FEC is used for the short-reach FlexO interfaces defined in G.709.5.[52] Consequently, these FlexO-x-<int> interfaces are designated as FlexO-k-RS, where RS denotes the use of an RS FEC. As can be seen in the figure below, the FlexO frame format was originally developed around this short-reach application. Each 5140-bit row of the

---

[52] When using the KP4 FEC, Ethernet streams are transcoded from 64B/66B block codes to 256B/257B block codes for increased bandwidth efficiency. The KP4 code was defined by IEEE 802.3 such that an integer number of 256B/257B blocks fit into an FEC codeword (i.e., 5140 = 20×257).

frame formed the payload area becomes a 5440-bit FEC codeword with the FEC check symbol bits appended to the end of the row.[53]

Figure 10-12. Short-reach FlexO-1 Frame Structure



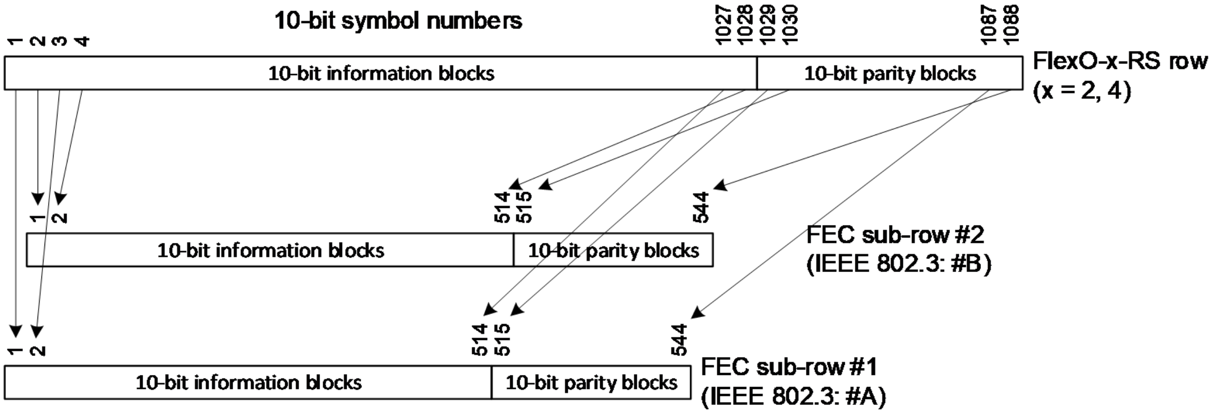
When 200GbE or 400GbE AMs are inserted into the KP4 FEC codeword stream of an Ethernet interface, they are inserted at the beginning of an FEC codeword, followed by padding such that the AMs and padding end on a 257-bit boundary. Since OTUCn does not use the 64B/66B or 256B/257B Ethernet line coding, FlexO was defined around 128-bit payload block granularity, which allowed the payload area of the first row to begin after 1280 overhead bits.

For 100Gbit/s FlexO-1-RS PHYs, the FEC is applied to each FlexO-1. The higher rate interfaces (FlexO-x-RS, *x* > 1), essentially follow the same 10-bit block (i.e., RS symbol) interleaving format as IEEE 802.3 for the equivalent rate by distributing the stream across pairs of FEC engines. As shown in Figure 10-5 above, when multiple FlexO instances are multiplexed into a FlexO-x frame with x > 1, the constituent instances are combined on a 10-bit round-robin basis and the frame row consists of 10280 bits (1028 10-bit symbols). The figure below provides a conceptual illustration of how the FlexO-x row is sub-divided on a 10-bit symbol basis into the two sub-rows that provide the input to the respective FEC engines for FlexO-2 and FlexO-4. This sub-division results in the AM overhead being associated with the correct information blocks. Note that this approach is consistent with IEEE 802.3 clause 119. Note also that per clause 119.2.4.7, the 10-bit re-interleaving of the FEC encoder outputs alternates between the FEC engines. Following the IEEE 802.3 approach for 800GbE, FlexO-8 divides the incoming client stream into two separate 400 Gbit/s streams that each follow the FleO-4 flow until they are merged in the final bit multiplexing to create the FOIC.

---

[53] While the OTN GFEC RS(255,239,8) Reed-Solomon code discussed above, operates on 8-bit symbols, KP4 operates on 10-bit symbols. The overall performance of GFEC and KP4 are similar, however, since 255/239 = 1.0669 and 544/514 = 1.0584, the larger symbol size of KP4 allows achieving this performance with slightly less overhead bandwidth. Although the ODUC rate is higher than ODU4, the lower overhead rate of KP4 when applied to the ODUC achieves a bit rate slightly lower than the OTU4 using GFEC. Specifically, referring to the FlexO-1 rate from Table 10-2, the short-reach FlexO-1 rate is (5440/5140)(105.64351 Gbit/s) = 111.80947 Gbit/s and per Table 6-1 above the OTU4 rate can be calculated as 111.809973568 Gbit/s. The resulting 4.46ppm difference allowed using the same optical module for OTU4 and FlexO-1 for short-reach interfaces.

Figure 10-13. FEC Sub-Row Structure for FlexO-x (x = 2, 4)



Interleaving across two FEC engines improves performance in scenarios where only one of the FEC lanes is experiencing degraded performance, since it is being protected by both engines. However, framer devices like the DIGI family typically need to support different combinations and permutations of FlexO-1, FlexO-2 and FlexO-4 interfaces, which becomes much more complex when each interface type needs to support a different FEC interleaving format. Consequently, G.709.5 allows supporting FlexO-n interfaces (n ≥ 2) by using *n*×FlexO-1 PHYs, each with its own FEC, rather than interleaving the FlexO-n signal across multiple FEC engines. Specifically, G.709.5 allows supporting *n*×100G client signals through the use of *n*×FOIC1.2-RS interfaces (for example, using 4×FOIC1.2-RS rather than FOIC4.8). See the FlexO Data Flow sections below for the description of the FOICx.k and additional elaboration regarding using *n*×FOIC1.2-RS.

10.5. FlexO-n Data Flow Illustrations

10.5.1. FlexO-n Data Flow for OTUCn Mappings

The data flow from the OTUCn client to the output FlexO electrical interface of order C (FOIC) is illustrated in general terms in Figure 10-14, with a more detailed example for n = 2 and 4 provided in Figure 10-15, which illustrates using the short-reach RS-FEC since it used for most FOIC applications. See the Flexible OTN (FlexO) section for the discussion and information regarding FOIC. Note that at a high level, the FlexO data flow is conceptually similar to the IEEE 802.3 Ethernet flow for the same rates, especially with respect to how the interleaving of data into the FEC engines is performed.

As illustrated in Figures Figure 10-14 and Figure 10-15, in the transmit direction, the OTUCn is divided into its n constituent OTUC signals, each of which is then mapped into its own FlexO-1 frame (see the FlexO Frame Format section). With the exception of the AM fields, the FlexO overhead is added to the frames. The n FlexO frames are then multiplexed together via interleaving 10-bits from each in a round-robin manner to form the input to a FlexO-n scrambler (see the OTUk Scrambling section). The scrambler output is distributed in a 10-bit round-robin manner to the set of FEC engines. For n = 2 or 4, the odd numbered FlexO-1 signals are processed in one FEC engine and the even numbered FlexO-1 signal are processed in the other FEC engine. After FEC encoding, multiplexing and 10-bit symbol distribution is performed on the FEC engine outputs to create the 4n logical lanes. The combination of the multiplexing and distribution of the 4n 10-bit symbol streams results in each of the AMs appearing on the appropriate logical lane at this point.

For the case of n = 1, a single FEC engine is used. Consequently, the signal is not divided ahead of the scrambler. The data flow is to first scramble the FlexO-1 frame, then add the AM information. At that point the data is input to the FEC engine. The FlexO-1 with FEC is then distributed to the four logical lanes in a 10-bit round-robin manner, which results in the AM fields appearing on the correct logical lane. The signal is then typically sent to the FOIC block as described in section 8.

The flow for n = 8 is similar to Figure 10-15 with the following exceptions. The n = 8 signal flow is effectively constructed by using a pair of the n = 4 flows. OTUC #1-4 use one flow and OTUC #5-8 use the other flow. Consequently, there are total of four FEC engines. Lanes 0-15 result from the first pair of FEC engines and lanes 16-31 result from the second pair of FEC engines. These 32 logical lanes are combined in pairs using a bit-level multiplexer in order to create the 8-lane FOIC. Specifically, FOIC physical lane 0 receives the rotating sequence of a bit at a time from logical lanes 0, 1, 16 and 17. FOIC physical lane 1 receives a sequence of a bit at a time from logical lanes 2, 3, 18 and 19. Etc., with FOIC physical lane 7 receiving a sequence of a bit at a time from logical lanes 14, 15, 30 and 31..

Having each AM appear on a separate lane allows the receiver to find and align to each lane, and perform lane deskew and de-interleaving of the 10-bit symbols. The de-interleaved 10-bit symbol streams are distributed to the FEC decoder corresponding to the FEC encoder used on these streams. I.e., the odd numbered FlexO-1 signals are processed by one FEC engine and the even numbered FlexO-1 signals are processed by the other FEC engine. For the case of n = 4, the odd numbered FlexO-1 signals are merged by 10-bit interleaving prior to the FEC decoding, and similarly for the even numbered signals. The outputs of the FEC decoders are then distributed to the AM removal function and descrambling is performed on the resulting FlexO-n after the AM removal. After the overhead is removed from the constituent FlexO-1 signals, the AM information is used to reorder them so that the individual OTUC1 elements can be extracted. The OTUCn output is formed after finding frame and multiframe alignment of the OTUC1 elements and removing any skew between them.

Figure 10-14. Generic Illustration of the FlexO OTN Data Flow

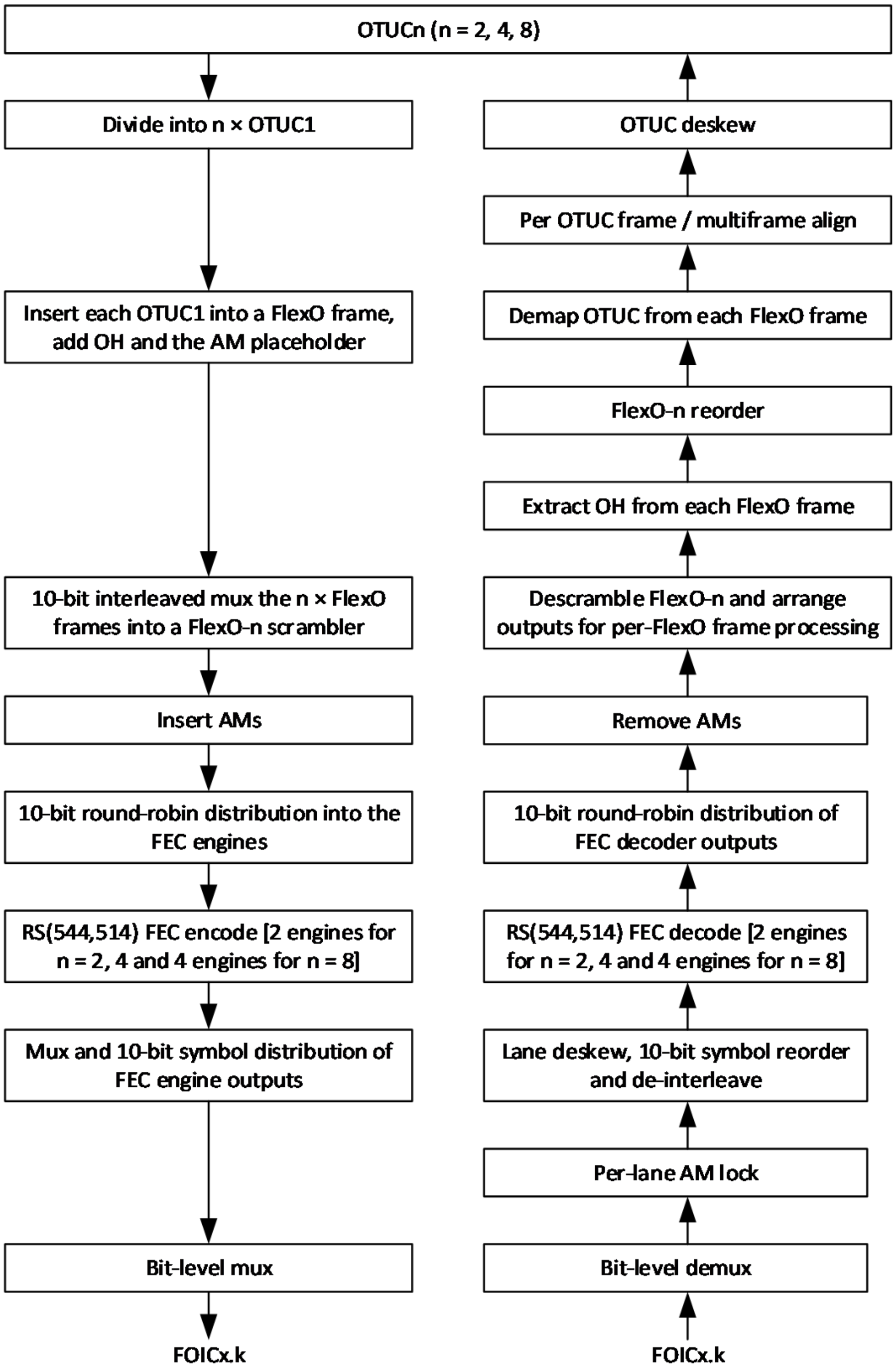
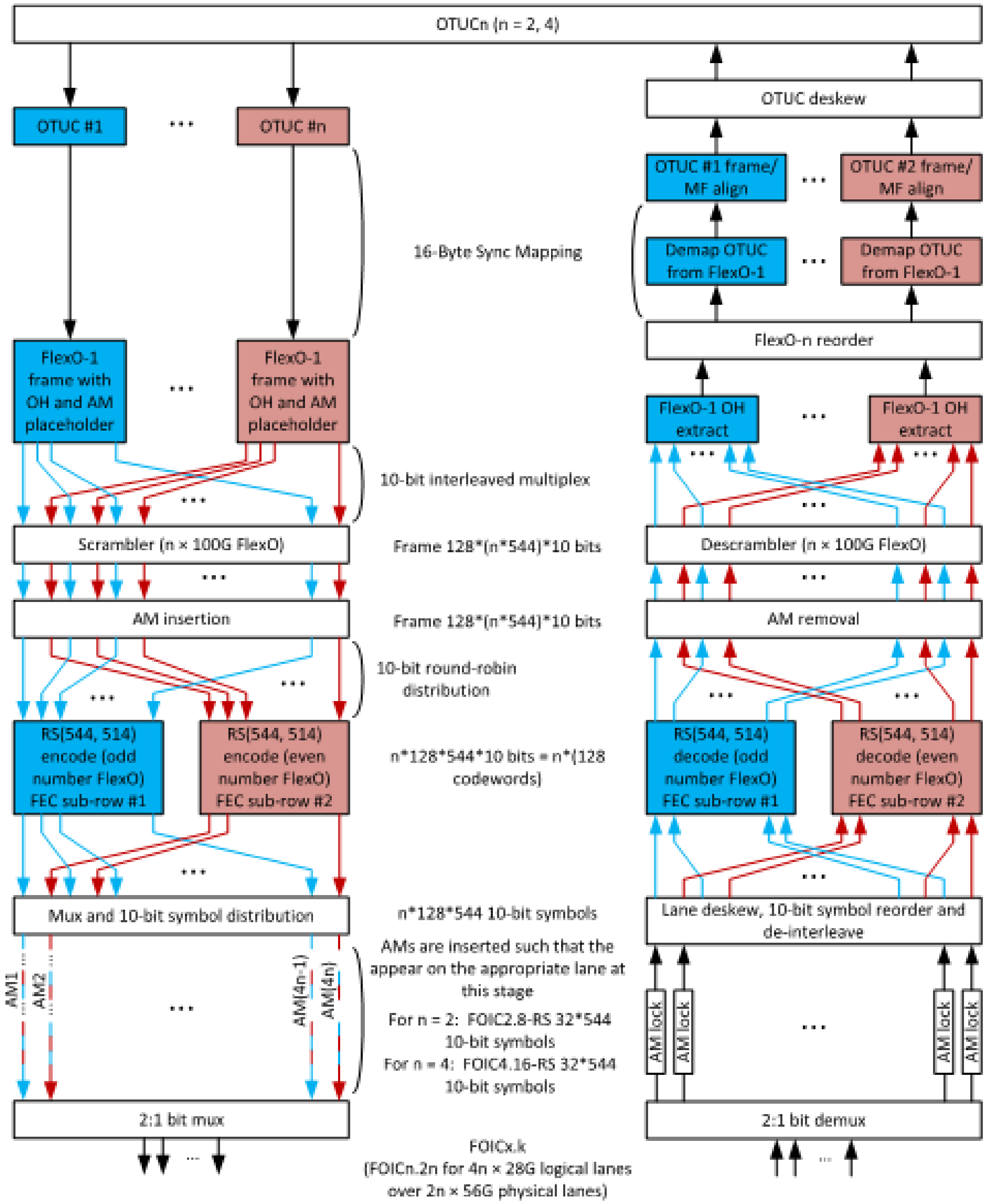


Figure 10-15. Specific OTN Data Flow Example for n = 2, 4



10.5.2. FlexO-ne Data Flow Mapping for y Instances of y00GBASE Ethernet Client

As explained in the GMP Mapping section using GMP for this mapping allows multiplexing one of more separate Ethernet clients into n instances of a FlexO-ne. Since the y00GBASE Ethernet client signal rates are lower than the corresponding OTUCn signal rates, eliminating the ODUflex and OTUCn overhead allowed reducing the FlexO rate to be closer to the Ethernet client rate. The resulting Ethernet optimized FlexO signal is referred to as a FlexO-ne / FlexO-xe (i.e., a FlexO-n with a rate and format optimized for y00GBASE Ethernet clients).

The FlexO-ne signal format has been adopted by OIF and OpenROADM as the mechanism for providing the FEC frame for their long reach Ethernet interfaces (e.g., y00ZR / y00ZR+). IEEE 802.3 subsequently adopted the same frame format for long reach 100GbE in 802.3ct and plans to use it for higher rates in 802.3dj. The main difference between the OIF / IEEE 802.3 signals and the ITU-T FlexO-xe is that the OIF and 802.3 interfaces are only defined for point-to-point applications. In contrast, FlexO includes overhead for telecom networking applications.

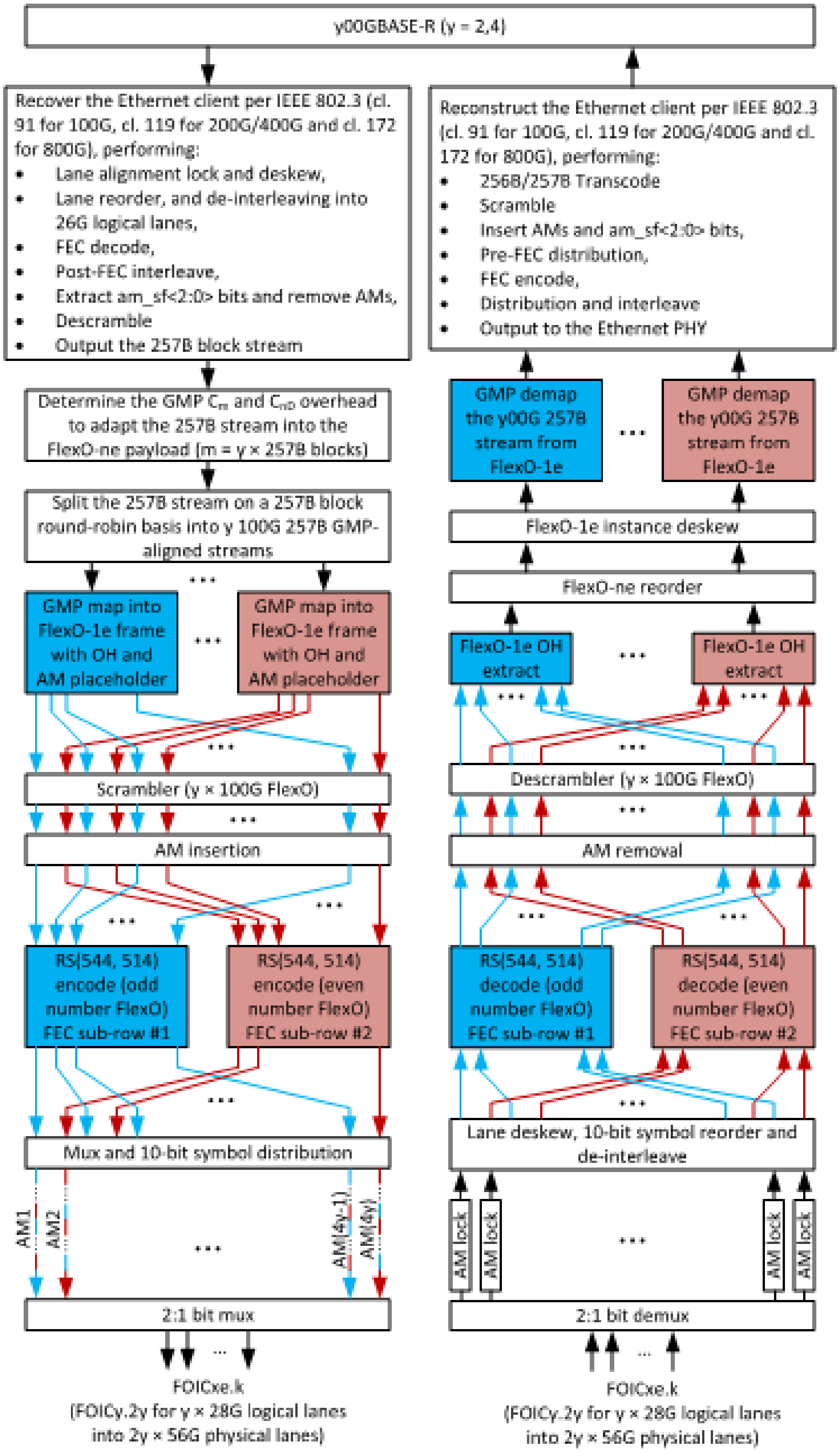
As can be seen by comparing Figure 10-15 and Figure 10-16, mapping y00GBASE Ethernet into FlexO-ne is similar to mapping OTUCn into FlexO-n. In the Ethernet case, the client signal is terminated up to the point of recovering a stream of 256B/257B blocks (e.g., by performing client FEC decoding, descrambling, AM removal, etc.). This client stream is referred to as the OTN reference signal. GMP encoding occurs at this point for mapping y × 257-bit block words into the FlexO-ne payload area. The y00 Gbit/s stream of 257b blocks is split into 100 Gbit/s streams on a round-robin 256B/257B block basis such that the streams are GMP aligned, and the 257-bit word alignment is preserved through the mapping procedure. Each of the 100 Gbit/s streams is then inserted into the payload of a FlexO-1e instance. The GMP alignment across the streams allows each FlexO-1e instance to carry the same GMP C_m and $\sum C_{nD}$ overhead, with the GMP stuff words appearing in the same locations within the respective FlexO-1e payload areas. In other words, the mappers of each FlexO instance across the FlexO-ne are locked.

When the GMP and other FlexO overhead has been added to the set of FlexO-1e instances, the remainder of the flow (FlexO FEC encoding, AM insertion, scrambling, etc.) is the same as for OTUCn client.

Similar to the case with OTUCn clients, there are exceptions to the Figure 10-16 flow for y = 1 and 8. For the input 100GBASE Ethernet, the exceptions include:

- There is a single FlexO FEC engine and hence no need to divide the signal across multiple engines.
- There is no need to perform descrambling since 100GBASE-R does its scrambling prior to the 257b transcoding.
- Similarly, in the demapper direction there is no need to scramble the outgoing blocks since they are already intrinsically scrambled.
- The client AM BIP counters are discarded at the mapper and there is no need to extract and process the am_sf<2:0> bits, since these are not used with 100GBASE Ethernet.

Figure 10-16. Specific y00GBASE Ethernet client data flow example for y = 2, 4



The flow for 800GBASE clients is similar to Figure 10-16 with the following exceptions.

- The y = 8 signal flow is effectively constructed by using a pair of the y = 4 flows, with #1-4 using one flow and #5-8 using the other flow. Consequently, there are total of four FEC engines. Lanes 0-15 result from the first FEC engine pair and lanes 16-31 result from the second FEC engine pair. These 32 lanes are combined to create the FOIC using the 32:8 bit multiplexing order described above in the FlexO-n Data flow for OTUCn Mappings section.
- When the am_sf<2:0> bits are extracted from the client AMs, they are ORed together between both 400G flows.
- Per clause 172.2.4.2, the two 400G 257b streams are descrambled without AMs.

10.6. Regen Application

As an alternative to, or in addition to FEC or optical amplification, longer reaches can be achieved by regenerating the signal at one or more points along the span. Regeneration goes beyond simple repeater functionality in that it also involves terminating the optical signal, recovering the digital signal's clock and data, and mitigating jitter that has accumulated on the signal. The "fresh" digital signal is then re-transmitted with the clean clock over the next optical span[54].

Since the regenerator node is an active piece of equipment, it is important to be able to manage it (e.g., detect equipment faults, determine the quality of the signal it receives, and set the parameters for its transmitted signal). The OTN OTUk overhead was designed to support the Regen function, which is why you will see it referred to as "RS" (Regenerator Section) overhead.

See Extended Overhead (EOH) for the Regen overhead descriptions.

---

[54] In other words, regeneration typically involves O/E, clock and data recovery, jitter filtering, E/O.

11. OTN Electrical Interfaces – MFI FlexO Interfaces of Order C (FOIC) and OTL (OTN Transport Lane)

OTL and FOIC are types of an OTN electrical module to framer interface (MFI). Since the scope of ITU-T SG15 is defining interfaces between network operators and between equipment with a network, intra-system interfaces like MFI are outside its standardization scope. However, because such interfaces are crucial to support the OTN ecosystem, they have been defined in an informational supplement, G.Sup58.

Note: In March 2025, SG15 agreed to move the higher rate FOIC interfaces into a new normative ITU-T Recommendation at its October 2025 meeting.

The OTN signal on the MFI includes an FEC. In most applications, the optical module terminates the MFI FEC and implements a stronger FEC for the optical span in addition to the module's optical DSP (ODSP) function. While the primary application of the electrical interfaces is MFI, both OTL and FOIC can also support options for the digital portion of the MFI electrical interface to pass through the optical module onto the optical interface.

The OTL naming convention is:

OTLk.n-<int> where

- n is the number of parallel electrical lanes in the interface.
- k designates the OTN signal rate (i.e., the rate of the OTUk carried on that interface).
- <int> designates the type of FEC.
  - RS for the G.709.5 Reed-Solomon
  - SC for the OTU4 Staircase FEC of G.709.2

The FOIC naming convention is:

FOICx.k-<int> / FOICxe.k-<int> where

- x is the FlexO-x interface bit rate (i.e., *x* × 100 Gbit/s).
- k is the number of parallel electrical lanes in the interface.
- e designates an Ethernet-optimized FlexO.
- <int> is MFI.

All FOIC interfaces use the same RS FEC as short-reach G.709.5 FlexO optical interfaces.

Note: Within this section, the term "logical lanes" is equivalent to the Ethernet BASE-R PCS lanes over which the FlexO-x-RS signal is distributed. An FOIC physical lane can carry one or multiple multiplexed logical lanes.

Note: The G.Sup58 OTL and FOIC electrical interfaces are typically in support of an optical interface application code from either Rec. G.685 (clause 8) or G.959.1 (clause 8).

11.1. OTL Interfaces

The first OTL interfaces were specified for carrying OTU3 and OTU4 over a multi-lane electrical interface that could re-use the respective 40GBASE-R and 100GBASE-R Ethernet pluggable optical modules. Both interfaces are specified in G.709 clause 8.1 and Annex C. OTL3.4 and OTL4.10 use 11 Gbit/s[55] lane rates for their electrical PHYs. The respective OTU3 or OTU4 is divided into 5G logical lanes on a round-robin 16-byte basis. In the case of OTL4.10, each OTU4.10 physical lane carries two bit-multiplexed logical lanes.

---

[55] (255/236) × 9.95328 bit/s for OTL3.4 and (255/227) × 9.95328 bit/s for OTL4.10

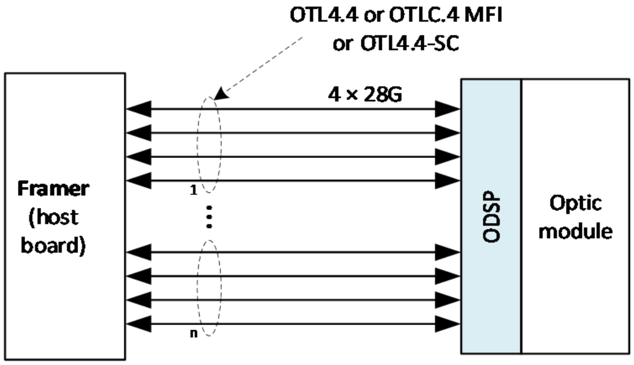
OTL4.4 uses 28 Gbit/s[56] electrical lanes. Since Ethernet four-lane electrical CAUI chip to module interfaces are common in modules, the OTL4 5G logical lane format was specified such that it can use the IEEE 802.3 10:4 PMA/gearbox to convert between OTL4.10 and OTL4.4. Five 5G logical lanes are bit-multiplexed onto each OTL4.4 lane.

Optical modules for OTL3.4 and OTL4.4 have 4-lane WDM optical interfaces, and the gearbox for lane translation between an OTL4.10 MFI and an OTL4.4 optical interface is typically implemented within the optical module.

When 56 Gbit/s electrical interfaces became available, an OTL4.2 was defined that used them.

The set of 4-lane MFI OTL interfaces for 100 Gbit/s OTN is illustrated in the figure below. As explained below, OTLC.4 carries an OTUC1 (i.e., one of the OTUC slices of an OTUCn) over a 4-lane OTLC MFI. The OTL4.4-SC is unique in that it uses the Staircase FEC of G.709.2 (see Appendix B: Long-Reach FlexO Interfaces) on the OTU4 in order to allow the OTL signal to be transmitted over the optical interface for metro/long reach applications rather than adding an FEC within the optical module. Since FlexO was developed for use with B100G optical signals, the FOIC interface is also specified for carrying OTUC1 and is used for all OTUCn MFI interfaces beyond n = 1.

Figure 11-1. Illustration of the OTL4.4/OTUC.4 MFI application



For the OTLC.4 interface, the OTUC frame is extended with 256 RS(255,239) FEC parity byte columns in the same octet-based 4-row by 4080-column block frame structure manner as an OTUk. The frame structure is also scrambled in the same manner as an OTUk. Using an approach consistent with OTL4.4, the resulting signal is divided into 20 × 5G logical lanes that are combined into the four OTLC.4 physical lanes. A logical lane alignment marker is placed into the third OA2 byte location of the frame in order to support reordering of the 20 × 5G logical lanes in the 100G OTUC group.

As explained in ODU25 and ODU50, G.709 added 25G and 50G OTN interfaces as OTU25 and OTU50, respectively. Under-clocked versions of these interfaces that match the corresponding Ethernet rates are designated as OTU25u and OTU50u, which enable re-use of Ethernet modules those respective rates. The corresponding OTL interfaces for OTU25 are OTL25-RS and OTL25u-RS and the OTL interfaces for OTU50 are OTL50.2-RS, OTL50.1-RS, OTL50u.2-RS and OTL50u.1-RS.

11.2. FOIC Interfaces

As noted above, the nomenclature for FOIC interfaces is FOICn.k-MFI, where n is the MFI interface rate and k is the number of parallel electrical lanes in the interface. At the time of writing for this tutorial, the FOIC interfaces explicitly listed in G.Sup58 included:

- MFI using 28 Gbit/s electrical lanes: FOIC1.4, FOIC2.8, FOIC4.16
- MFI using 56 Gbit/s electrical lanes: FOIC1.2, FOIC2.4, FOIC4.8 and n × FOIC1.2
- MFI using 112 Gbit/s electrical lanes: FOIC1.1, FOIC4.4, FOIC8.8

---

[56] (255/227) × 24.8832 bit/s for OTL4.4

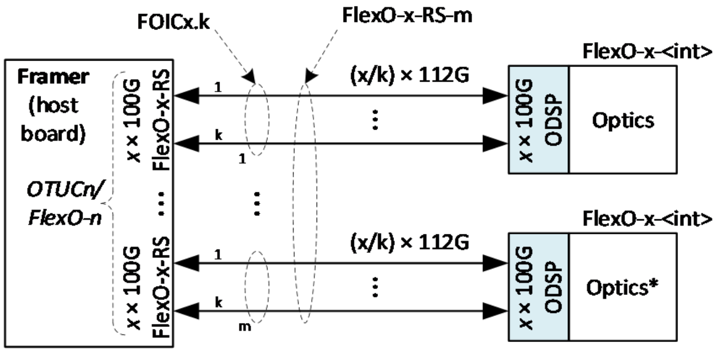
- MFI using 106.5 Gbit/s electrical lanes: FOIC1e.1, FOIC4e.4

The formats of these interfaces are discussed in the following text.

The FOIC concepts are illustrated in the figure below. In this example, the FOIC electrical lane rate is 112 Gbit/s. As shown, a framer could have a single or multiple separate FOIC interfaces to different optical modules.

Since it is common for the same framer device to support a combination of 100G, 200G and 400G interfaces, there are framer implementation advantages to using a 100G modular approach (e.g., n × FOIC1.2-RS) as the basis for all FOIC interfaces with any combination of *n*×100G client signals. This avoids the need to support all the possible permutations of FEC architectures for the different combinations of interfaces (e.g., combinations of FOIC1.2, FOIC2.4 and FOIC4.8). In addition, it is more complex to use an FOIC4 device interface to support applications like 2×ODUC2, especially since that would require additional functionality like GMP mapping of each ODUC2 into the FOIC4.8. Another advantage to adding support for 4×FOIC1.2-RS is that it allows direct support for breakout cable applications[57].

Figure 11-2. Illustration of the Typical FOIC Application and Structure



* An optical module may connect to multiple FOICx.k interfaces (e.g., to create a FlexO-p-<int> where p = *j* × *x*

** While the use of an ODSP has become common, it is not a required element of the optical module

11.2.1. FOIC for FlexO-n MFI

After FEC encoding, the FlexO-k-RS frame is distributed over the 28G logical lanes of an FOIC in a 10-bit round-robin manner from lowest to highest numbered lanes. Since this distribution is the opposite of the 10-bit interleaving of the AM fields, it results in the appropriate AM field appearing on each of the FOIC logical lanes.

For FOIC1.k, the first step is to adapt the FlexO-1-RS to create an FOIC1.4-RS. The four logical lanes are then bit multiplexed into the k physical lanes (k = 1,2,4).

Since FlexO-2 uses two FEC engines, two FEC codewords (i.e., two sub-rows) are 10-bit interleaved and then round-robin distributed to logical lanes, consistent with IEEE 802.3 clause 119 (refer to Figure 10-16). The logical lanes are bit-multiplexed into the k physical lanes of the FOIC2.k.

Similarly, since FlexO-4 uses two FEC engines, two FEC codewords (i.e., two sub-rows) are 10-bit interleaved and then round-robin distributed to logical lanes, consistent with IEEE 802.3 clause 119. The logical lanes are bit-multiplexed into the k physical lanes of the FOIC4.k.

---

[57] While using FOICn for n≥2 has the advantage of distributing the information across multiple FECs, for MFI applications there is no practical performance difference relative to the per-100G FEC with n × FOIC1.2-RS.

FlexO-8 is specified to use the FOIC8.8-RS MFI. Four logical lanes are bit-multiplexed to create the eight physical lanes. Note that each physical lane is specified to carry two logical lanes from lane 0-15 followed by two logical lanes from lane 16-31.

11.2.2. FOIC for FlexO-ne MFI

FlexO-ne signals are carried by FOICke.k interfaces. The FOIC1e.1 and FOIC4e.4 electrical lanes use the 100G Ethernet interface rate of 106.5 Gbit/s. The FEC and AM format of the associated FlexO-ne signals is illustrated in Figure 10-16.

11.2.3. FOICn.k-RS interfaces

As noted above, the typical application with FOIC is to provide the electrical interface between the framer and an optical module that terminates the FOIC and adds a stronger FEC for the optical interface. Since short-reach FlexO optical interfaces use the same RS FEC format as the FOIC MFI, there are applications for which the k FOIC electrical signal lanes can be combined for transmission over the optical interface.

12. OTN Security

Different methods and topologies exist for providing secure communication over OTN. At a high level, the topologies are:

- Customer-provided security (CPE to CPE) on the client data sent over the OTN. Transparent to the OTN path.
- End-to-end security over the OTN path. Transparent to all OTN NEs that do not participate as a security endpoint.
- OTN link-level (PHY-level) security.

The most popular approach for the first type is using the Ethernet MACsec protocol standard (IEEE 802.1AE) on the Ethernet packet stream. This type of security is invisible to the OTN, and hence outside the scope of this tutorial.

The second approach is known as OTNsec or ODUsec. FlexOsec provides a standard for third approach. These two approaches are addressed in this section.

Security involves both encryption, which protects the data through confidentiality, and an authentication protocol that verifies the data's integrity and origin. The encryption process is based on a secret code that can only be reversed with a unique digital key. For added security, the encryption protocols provide a coordinated key change method that enables periodic transitions to using new codes that require a different key. Authentication is critical for ensuring that only unmodified data from the intended source is accepted.

This section covers which OTN / FlexO overhead channels are available for supporting the security functions and which portions of the OTN / FlexO frame information is covered by encryption and authentication. The specific encryption and authentication algorithms are beyond the scope of this white paper.

12.1. FlexOsec

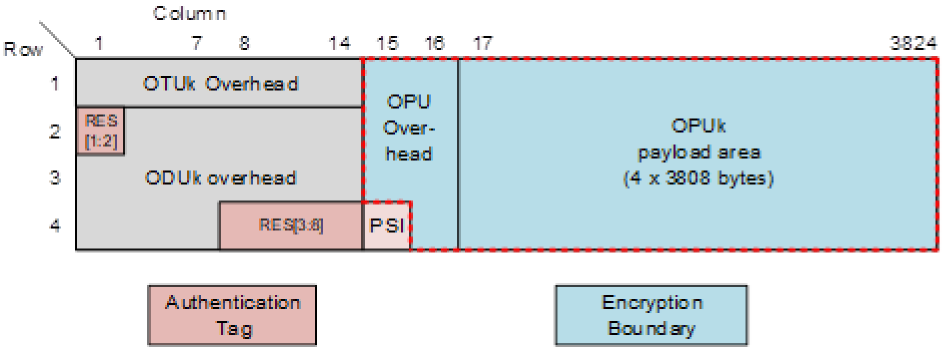
As described in Annex A in G.709.1, when interface confidentiality is required, encryption covers the information in the FlexO frame payload area prior to FEC adaptation and FlexO-x instance interleaving. Encryption is applied per individual FlexO instance. GCM-AES-256 is used for performing the encryption.

Authentication covers the payload area, the BOH field and a portion of the EOH field. The EOH fields pertaining to security begin in EOH byte 21, which is bit 641 of the FlexO frame. The first 16 bytes of the security overhead are an authentication tag calculated for the authenticated bits of the previous frame. The authenticated bits of the current frame begin with EOH byte 37 (bit 768 of the FlexO frame) and run through the end of the frame. Authentication is also applied per individual FlexO instance. The authentication tag is calculated over FlexO frame bits beginning with bit 769 through the end of the frame. Bit 769 corresponds to the first bit of the seventh 128-bit word, which also corresponds to the first bit of the EOH following the authenticate tag inserted for that frame (see Extended Overhead (EOH)).

12.2. ODUk Security

Although no OTNsec / ODUsec approaches have been standardized, Microchip introduced this feature with DIGI-G4. Existing Reserved bytes within the OTN frame are used for carrying encryption and authentication tag overhead information. Note that some of this overhead is carried in reserved bytes of the time-shared csf overhead. The encryption uses an AES-256 cipher.

Figure 12-1. Illustration of the Microchip ODUsec Structure



13. Considerations for OTN Beyond 1 Tbit/s

At the time this white paper was written (early 2025), the IEEE 802.3dj Task Force was working on Ethernet interfaces for MAC and PHY rates beyond 1 Tbit/s (B1T), including new 200 Gbit/s electrical lanes. ITU-T Q11/15 was working in parallel on FlexO extensions associated with both supporting the transport of 802.3dj Ethernet clients and for allowing FlexO reuse of the B1T Ethernet optical modules. Both groups maintained regular liaison correspondence. The following is a brief summary of the Q11/15 preliminary conclusions and key open issues:

Preliminary Q11 conclusions (working understandings and assumptions) as of March 2025:

- A new FlexO-based network will be defined.
    - Q11 will focus primarily on defining the new B1T FlexO Path layer(s).
    - The FlexO TS rate and path switching granularity will be nominally 100 Gbit/s rather than 5 Gbit/s.
- A LO and HO path layer are required.
    - The currently defined FlexO-xe structure and rate is assumed as the choice for the LO path.
    - The HO Path must include MS layer functionality in order to allow multiplexing multiple independent LO FlexO path signals into it.
        - GMP is assumed for client mapping into the LO path and for multiplexing into the HO path (assuming 257-bit transcoding).
- Ethernet will by be the dominant FlexO client at 1.6 Tbit/s.
    - Consequently, the B1T signal format will be optimized for carrying Ethernet clients, including 1.6TbE and y00GbE (y = 1, 2, 4, 8).
    - Direct mappings into the LO FlexO path will be defined for these Ethernet clients (at least for 1.6 TbE) rather than first mapping them into an ODUflex.
- Regarding the new network and legacy pre-B1T OTN:
    - Mappings must be defined for carrying legacy OTN over the new networks.
    - Mappings must be defined for carrying new network signals over legacy OTN networks.
    - Q11 will not address interworking between the old and new networks.

Key open questions for Q11:

- Will the new FlexO section layer interface need a rate somewhat higher than -xe (sometimes referred to as -xo)?
- Is a “Super HO Path” required?
    - To support transparent “tunnelling” through a network domain
    - Is a MS adequate for most of these applications?
- What permutations of LO and/or HO paths would an MS need to support?
    - e.g., how do we identify the layers?
    - e.g., with a payload type or making them mandatory?

14. Conclusions

The OTN standard and technology has a long history of providing the network operators with the capabilities that significantly simplify their OAM and associated OpEx. These capabilities allow network operators to provide guaranteed “carrier grade” quality of service for their users. The protocol has proven to be extremely flexible for accommodating new types of clients (both CBR and packet-oriented) and evolving to support ever higher transmission rates. The goal of this tutorial was to introduce and explain the key aspects of OTN, including the rationale behind how and why they were developed. A related goal is that it will allow the reader to more easily read and understand the related OTN standards when a more detailed understanding is required.

Microchip is a leader in OTN solutions for Muxponders, Switchponders, and Multi-service line cards in OTN systems. Microchip's innovative DIGI, META, and HyPHY product families enable carriers to deliver cost effective transport, grooming, and switching of OTN services end-to-end in their optical networks with resilient and reliable service that exceeds the most stringent Service Level Agreements. For more detailed information on the DIGI, META, and HyPHY product families and the complete portfolio of OTN solutions from Microchip, please visit the Microchip Optical Networking solutions web page (www.microchip.com/en-us/products/high-speed-networking-and-video/optical-networking),

15. Appendix A: Introduction to Wavelength-Division Multiplexing

Glass fibers are not fully transparent to light, and in fact typically have specific wavelength windows where the light attenuation is lowest.[58] The O-band (Original) window is 1260-1360 nm, which supports single mode transmission and is commonly used in client interface and CWDM applications. Commonly used longer wavelength windows are the 1528-1561 nm C-band (Conventional) region and 1565-1625 L-band (Long wavelength) region. Both have significantly lower attenuation than the O-band. The C and L bands require lasers that are more expensive than those for the O-band, but as explained below, the technology is better suited for Dense wavelength-division multiplexing (DWDM) operation.[59]

An ideal unmodulated laser would output light with a single wavelength, which would appear as a single line on a spectral graph. In practice, however, physical realities mean that a laser outputs light with some spread around its central wavelength. Once modulated, the laser signal develops sidebands, and occupies a bandwidth determined by the data rate and modulation scheme. Since this spread looks like a fatter line on a spectral graph, a laser's wavelength spread is often referred to as its line-width. Clearly, the line-width of the lasers determines how many lasers' signals can be combined in a wavelength window with WDM. If the wavelengths of two lasers overlap, they will interfere with each other at their respective receivers. The width of the sidebands (literally, the bandwidth) determines the minimum spacing between laser frequencies in a wavelength-division multiplexing system. Additional guardbands must be added to account for nonidealities in the laser and other optical components. Many DWDM systems use 50GHz, which causes substantial difficulties in transmission of 40Gbps and 100Gbps signals.

Although the lasers for the 1555 nm region are more expensive than those for the 1310 nm region, the good news is that it is also possible to manufacture these lasers with relatively narrow line widths that are compatible with the erbium-doped fiber amplifier (EDFA) optical amplifiers discussed below. Multiple-quantum well (MQW) lasers with distributed feedback (DFB) can achieve line widths of a few hundreds of kHz (a few millionths of a nm).

In Recommendation G.694.1 and G.694.2, the ITU-T has defined "grids" of wavelengths that can be used for WDM. These grids specify the wavelengths that lasers can use, but does not specify which of these may or should be used within a WDM system.[60] In dense WDM (DWDM), the wavelengths in a WDM system are close together, with 50 or 100 GHz spacing. In coarse WDM (CWDM)[61], the wavelengths are much farther apart, which lowers cost at the expense of fiber capacity. As wavelengths closer together on the grid are used, crosstalk can become a problem. The primary

---

[58] Light propagates through a fiber by the process of total internal refraction in which the light going through the core of the fiber is refracted back into the core when it hits the cladding that surrounds the core. (The cladding has a lower index of refraction than the core.) In multi-mode transmission, the light "bounces" through the fiber as it encounters the cladding in such a manner that the portion of a light pulse that encounter the fewest bounces has a shorter path than the one that has the most bounces. The result is a time spreading of the pulse at the receiver that limits possible spacing between pulses (i.e., the possible data rate for a digital signal). In single mode transmission, only the light that goes directly through the core is able to propagate, thus minimizing any pulse spreading.

[59] This section follows a common practice of referring to the regions by their center wavelength, i.e., 1310 and 1555 nm.

[60] The wavelengths in the grid are called the C-band and are on evenly spaced wavelengths of 0.39 nm (50 GHz) starting at 1528.77 nm. The grid was originally assembled largely as a collection of the wavelengths supported by the various laser vendors. In addition to the C-band, DWDM systems can use the L-band (1561-1620 nm). The O-band (1280-1350 nm) is typically used for single wavelength rather than DWDM applications.

[61] A common, extreme example of CWDM is to use just two wavelengths, one at 1310 and the other at 1555 nm.

way to minimize or eliminate cross talk is to require the optical signals to have a sufficiently narrow line width and low enough drift from their center wavelength that their modulated signal does not overlap with adjacent channels. Another phenomenon that creates crosstalk is called four-wave mixing, which arises due to slight non-linearities in the fiber. In four-wave mixing, several optical signals constructively interfere ('beat') with each other to create quite large fields inside the fiber, to which the glass responds in a slightly nonlinear way. The result is the generation of interfering signals at various "beat frequencies." In the case of three or more evenly spaced signal frequencies, some of these "beat frequencies" align with the original signals, causing crosstalk. These crosstalk and mixing problems are familiar to people who are experienced in frequency division multiplexed (FDM) systems, since wavelength modulation is essentially a form of frequency modulation. Bit rate, fiber type, and fiber length are also factors in determining how many channels are possible in a DWDM system. DWDM systems for metro networks will use up to 40 wavelengths (100 GHz spacing), while DWDM for core networks commonly use up to 80 wavelengths (50 GHz spacing).

15.1. Optical Signal Regeneration

There are three aspects to the regeneration of optical signals (the three "Rs"):

- Re-amplification of the optical signal
- Re-shaping of the optical pulses
- Re-timing/re-synchronization

An all-optical amplifier is a 1R amplifier. A 2R regenerator both amplifies and re-shapes the optical pulses. A 3R regenerator also performs clock recovery on the incoming signal and re-times the outgoing signal in order to remove jitter on the pulses. Currently, both 2R and 3R regenerators convert the optical signal to an electrical signal (OE conversion) and create a new optical pulse (EO conversion) after amplification (and re-timing for 3R regenerators).

Clearly, a 3R regenerator needs to be aware of the client signal, at a minimum having the ability to perform clock recovery at that client signal rate, but possibly requiring the ability to frame on the client signal (e.g., to function as a Regenerator Section (RS) terminating NE).

An alternative is to amplify the signal in the optical domain (i.e., a 1R amplifier), thus removing any requirements on the regenerator concerning the rate of the client signal. This is accomplished through optical amplifiers, of which the EDFA is the most common. In an EDFA, the signal passes through a section of fiber that has been doped with erbium. A strong signal from a pump laser is coupled into this fiber segment. The 980 (or 1480) nm wavelength energy of the pump laser excites the erbium atoms, and the presence of 1555 nm signals causes the erbium atoms to transfer their energy to the 1555 nm signal through stimulated emission. Gains of over 20 dB are possible. Other types of amplifiers, such as the Raman amplifier, can amplify a much wider range of wavelengths (1300-1600+ nm). A full discussion of optical amplifier technology is beyond the scope of this white paper. Note that other types of optical amplifiers are available to cover other wavelength bands.

Carriers prefer to use 1R regenerators whenever it is practical in order to preserve maximum signal transparency and minimize the amount of costly electrical domain processing in the network.

15.2. Optical Switching

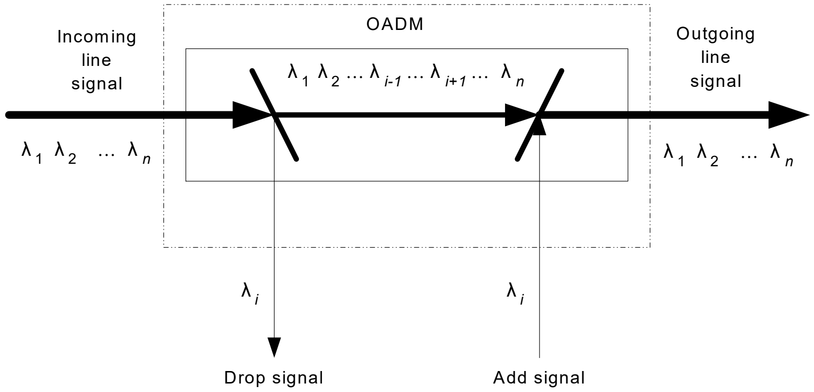
A key motivation for developing all optical networks is that if signals are kept in the optical domain, the network equipment can be agnostic to the client payload signals and eliminate the circuitry required for conversions between electrical and optical domains. For example, switching SONET signals requires STS-1 electronic cross connect fabrics and the OEO functions to convert the signal between the optical and electrical domains. Switching in the optical domain has the promise of lower equipment and provisioning costs at the expense of large granularity in the switched signals. As an additional capability, OTN also supports the use of use cross-connects that switch the client signal in the electrical domain whenever it is desirable to groom the signals that are placed on the wavelengths. Such switches are referred to as hybrid switches.

A number of different technologies exist for switching in the optical domain, including solid-state devices (e.g., directional couplers that are combined to form multi-stage switch fabrics), free-space techniques (e.g., waveguide grating routers), and micro-electrical-mechanical switches (MEMS). MEMS technology allows the construction of an array of mirrors in silicon where each mirror's reflection angle can be controlled by an electrical signal. The optical input signals to the MEMS array can be steered to the appropriate output ports. MEMS switch times are in the order of microseconds. For fast switching, $LiNbO_3$ solid-state switches can achieve switch times in the order of nanoseconds.

Optical equipment with extensive cross-connect capability is known as an optical cross-connect (OXC). Simpler equipment that is capable of adding or dropping wavelengths is known as an optical ADM (OADM). As illustrated in the following figure, an OADM filters an incoming wavelength(s), removing it from the incoming signal and steering it to a drop port. At the transmitter of the OADM, the signal from the add port is then optically merged back into the outgoing signal. OADMs can either add/drop fixed wavelengths or dynamically select which wavelengths to add/drop.

It should be noted that a typical OXC or OADM implementation would have an EDFA at the input to the NE that acts as a pre-amplifier prior to the cross-connect fabric. The pre-amplifier boosts the signal amplitude to compensate for the attenuation over the fiber, and sets it to an appropriate signal level for the switch fabric. Another EDFA is usually present at the output of the OXC or OADM to amplify the signal for transmission.

Figure 15-1. Optical Add/Drop Multiplexing Illustration



16. Appendix B: Long-Reach FlexO Interfaces

Long-reach FlexO interfaces are defined in G.709.3 for rates ≤400G (“B100G”) and G.709.6 for rates ≥400G (“B400G”). Each uses a stronger FEC than the RS10 used with the short-reach FlexO interface. The signal formats for a given FEC have variations based on the specific optical signal modulation technique used to transmit them. Long-reach FEC is typically implemented in the optical modules rather than in framer devices, which only need short-reach FEC for their electrical interfaces to the optical modules. Consequently, these interfaces are described in this Appendix, with pointers to the relevant G.709.3 and G.709.6 clauses for the details.

Since long reach FEC is implemented in the optical modules it is possible to terminate the FlexO-x-RS in the module encapsulate the FlexO signal as the payload of another stronger FEC. Since such optical modules include digital signal processing (DSP) capability, the frame associated with this strong FEC is called a DSP frame. Consequently, the long reach interfaces are typically labelled as FlexO-x-D<fec> interfaces, where D indicates the use of a DSP frame.

16.1. G.709.3 Long Reach Interfaces

16.1.1. FlexO-x-SC-m

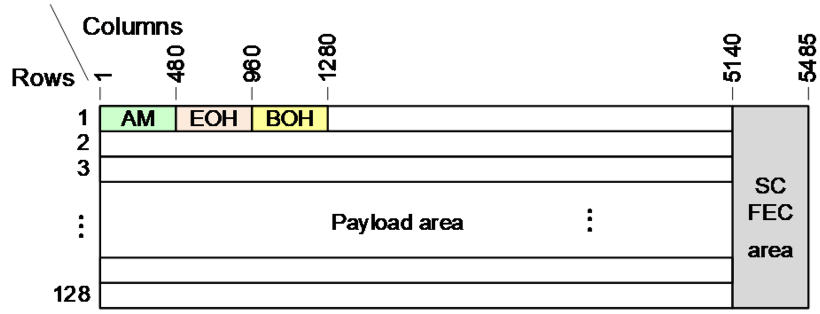
FlexO-x-SC defined in G.709.3 designates a long-reach FlexO interface that uses the iterative continuously recursively interleaved “staircase” (SC) FEC. After going through an error decorrelator in order to reduce the impact of correlated errors, the information bits are arranged into 2-dimensional 512 × 478-bit blocks. Using a base BCH(1022,990) code, the bits of consecutive blocks are arranged in two dimensions to be covered by two separate FEC codewords. The result is a hard decision (HD[62]) FEC with 6.7% FEC overhead. The details of the SC FEC are explained in G.709.2, which defines a long-reach version of the OTU4. Note that SC FEC is not used in G.709.6.

The FlexO-x-SC frame structure, as illustrated in the Figure below, uses the base FlexO payload structure as seen in Figure 10-2. To enable the stronger performance the ratio of FEC overhead to payload bits is 5458/5140 instead of the 5440/5140 ratio used for short reach interfaces. As with Figure 10-5 the payload rows of the FlexO-x-SC frame become 10280 bits long and there are twice as many parity bits per row for x>1.

---

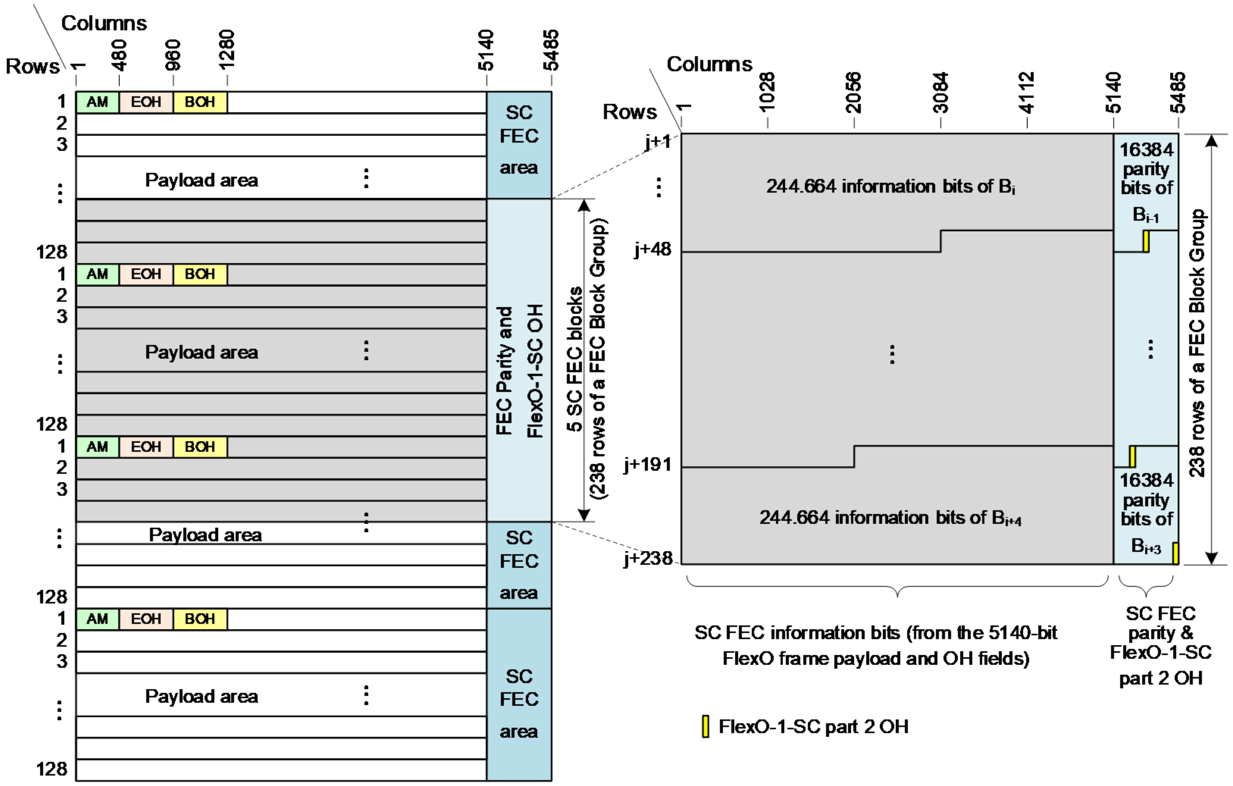
[62] Hard decision FEC decoders first determine the 0 or 1 value of each bit using a fixed decision threshold (i.e., the decision is “hard” in the sense that the value corresponds to one of the possible transmitted bit values). Each FEC codeword is then decoded with an algorithm that uses the codeword’s parity check information to attempt to identify and correct any bits where the receiver decision was erroneous. In contrast, a soft decision (SD) FEC receiver classifies each bit as a value between 0 and 1, indicating the degree of confidence regarding which the receiver’s interpretation of the value that was actually transmitted. Each bit position is covered part of two separate FEC codewords. For example, an FEC information block could have parity bits associated with each row of the block and parity bits associated with each column of the block. In the first pass, both FEC codewords are decoded based on the most likely values for each bit. If the associated error “correction” requires a different value for any of the bits, the confidence levels for the bits are algorithmically adjusted for another pass with the new most likely bit values. The iterations continue until both sets of FEC codewords are satisfied or a fixed number of iterations has been attempted. This type of decoding is sometimes referred to as turbo-decoding.

Figure 16-1. Long-reach FlexO-1 SC FEC Frame Structure



In order to accommodate the lack of alignment between the SC FEC block length and the FlexO-1 frame length, as illustrated in the Figure below, an additional FEC block group (FBG) structure is superimposed on the underlying FlexO-1-SC frame structure. The FBG consists of five contiguous base blocks, each of which consists of 238 consecutive 5485-bit FlexO-1-SC frame rows. The resulting FBG contains 5 × 244,664 information bits, 5 × 16384 parity bits and 5 × 38 FlexO-1-SC OH bits located between the parity bits of successive SC FEC blocks. The FEC Block Alignment (FBA) field of the EOH communicates the alignment of the FBG within the FlexO-SC frame. Note that the FBA is not used in FlexO-1-DSH interfaces discussed below. The “part 2 OH” blocks shown in the Figure below provide a multi-block alignment indication that is used to synchronize the transmitter and receiver error decorrelator controllers.

Figure 16-2. Long-reach FlexO-1 SC FEC Frame Structure – Expanded View



16.1.2. FlexO-x-D<fec>

As noted above, FlexO-x-D<fec> indicates a FlexO DSP frame using a particular type of FEC. The basic FlexO-x-D frame structures for G.709.3 and G.709.6 are shown in the frame structure Figures below. Each column in these figures corresponds to a Z-bit block location within the row. The value of Z depends on the specific interface type (e.g., Z = 4 or 8 in G.709.3 and Z = 128 in G.709.6). DSP is used to enable coherent interfaces. The receiver performance is improved by sending a periodic Training Sequence (TS) and Pilot Sequence (PS). The TS provides frame alignment, and the PS provides FlexO-x-D<fec> frame alignment. In other words, the TS and PS allow the receiver to align to the incoming signal such that it can properly perform the coherent receiver functions.

The FlexO-x-D<fec> frame is essentially a wrapper for carrying the payload and FEC parity bytes of the client signal.

Figure 16-3. FlexO-x-D<fec> Frame Structure - G.709.3 Version

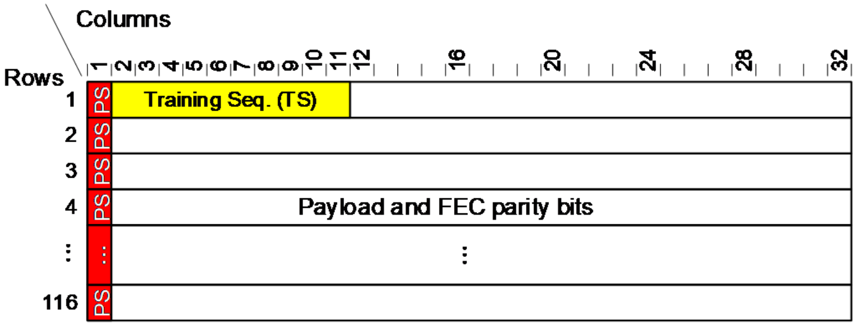
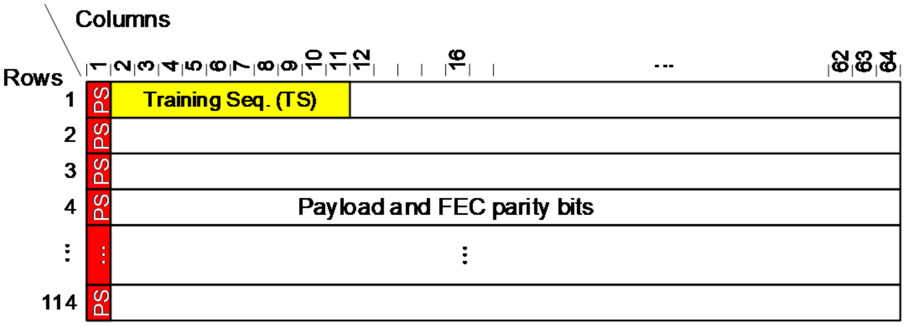


Figure 16-4. FlexO-x-D<fec> Frame Structure - G.709.6 Version



16.1.3. FlexO-x-DSH

The FlexO-x-DSH defined in G.709.3 carries a FlexO-x-SC client combined with an additional Hamming SD FEC (i.e., DSH = FlexO-D with SC + Hamming FEC). The FlexO-x-SC client and the Hamming parity are inserted into the FlexO-DSH frame in 128-bit blocks, which consist of the 119 payload and 9 parity bits of a systematic (128,119) Hamming code, as illustrated in Figure 16-6.

The FlexO-x-DSH multiframe is shown in Figure 16-5. As illustrated, the multiframe provides (116×31) - 10 Z-bit payload blocks in frames 2-49 and (112×31) - 15 Z-bit payload blocks in frame 1, for a total of 175616 payload blocks per multiframe. The MFAS occupies 22 Z-bit blocks across rows 1 and 2 of the first frame of the multiframe. The first frame also contains 76 Z-bit FS blocks across rows 2-4. The FS blocks contain a randomized pattern. The purpose of the FS is to set the FlexO-x-DSH mutiframe payload capacity to correspond to an integer number of SC FBG after they are encoded into 128-bit Hamming code blocks. A Superframe is also defined, which consists of 4 multiframes.

Figure 16-5. FlexO-x-DSH 49-frame Multiframe Structure

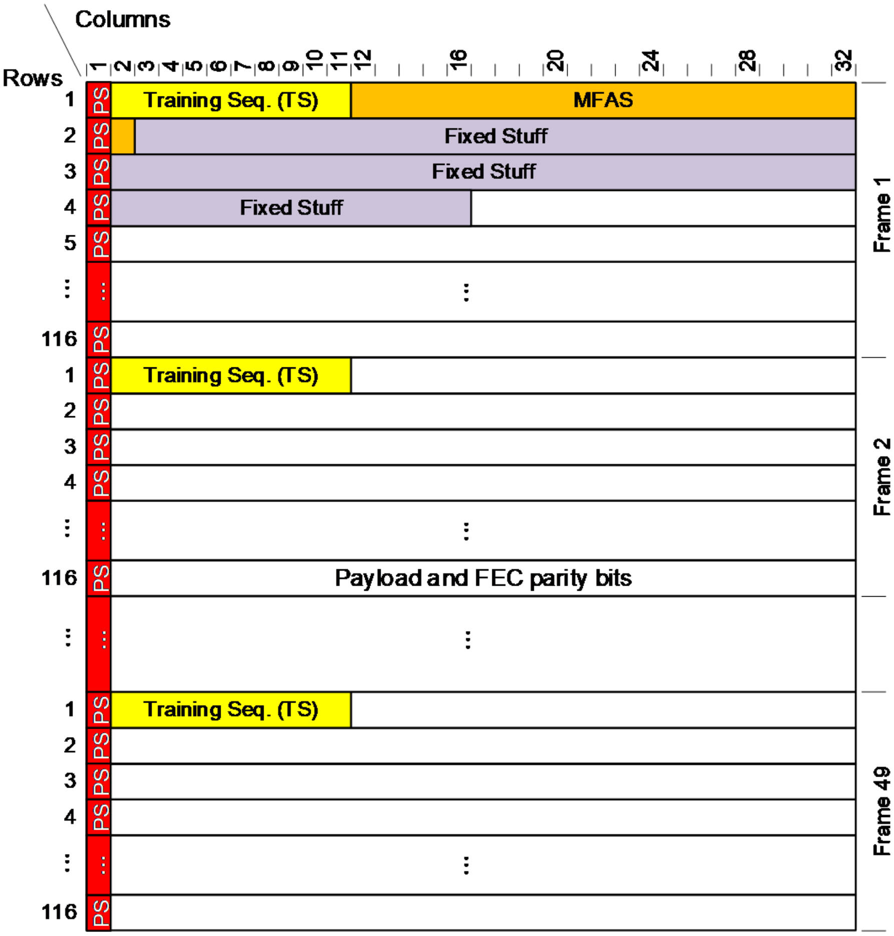
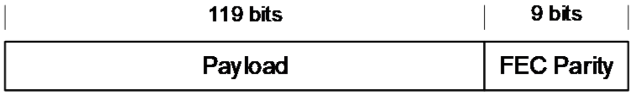


Figure 16-6. 128-bit Payload and FEC Parity Block



16.2. FlexO-x-DO

The “O” in -DO indicates that this is a FlexO-x-D signal using the Open FEC (OFEC) soft decision FEC. OFEC is a spatially coupled type of product code using a BCH(256,239) constituent code where each bit is part of two constituent codewords, each of which is a 256-bit binary vector. An OFEC codeword is a semi-infinite set of bits organized in a matrix with 128 columns and a semi-infinite number of rows.

FlexO-x-DO is one of the FEC options in G.709.3 and the only FEC option used in G.709.6.

As shown in Figure 16-7, the FlexO-x-DO multiframe has a format similar to that of FlexO-x-DSH, with the column numbering representing Z-bit blocks. The main differences are that the FlexO-x-DO multiframe consists of 48 frames rather than 49 and has 74 rather than 76 FS blocks. Note that the -DO and -DSH use different TS and PS signal patterns.

The 172032 k-bit block FlexO-x-DO multiframe payload and FEC parity area carries one OFEC Block Group (OFBG). The OFBG structure is superimposed on the underlying FlexO-x multiframe structure such that it contains 149184×k bits of consecutive FlexO-x multiframe rows, integrity and appended pad bits. See the example in Figure 16-8.

Figure 16-7. FlexO-x-DO 48-frame Multiframe Structure for G.709.3

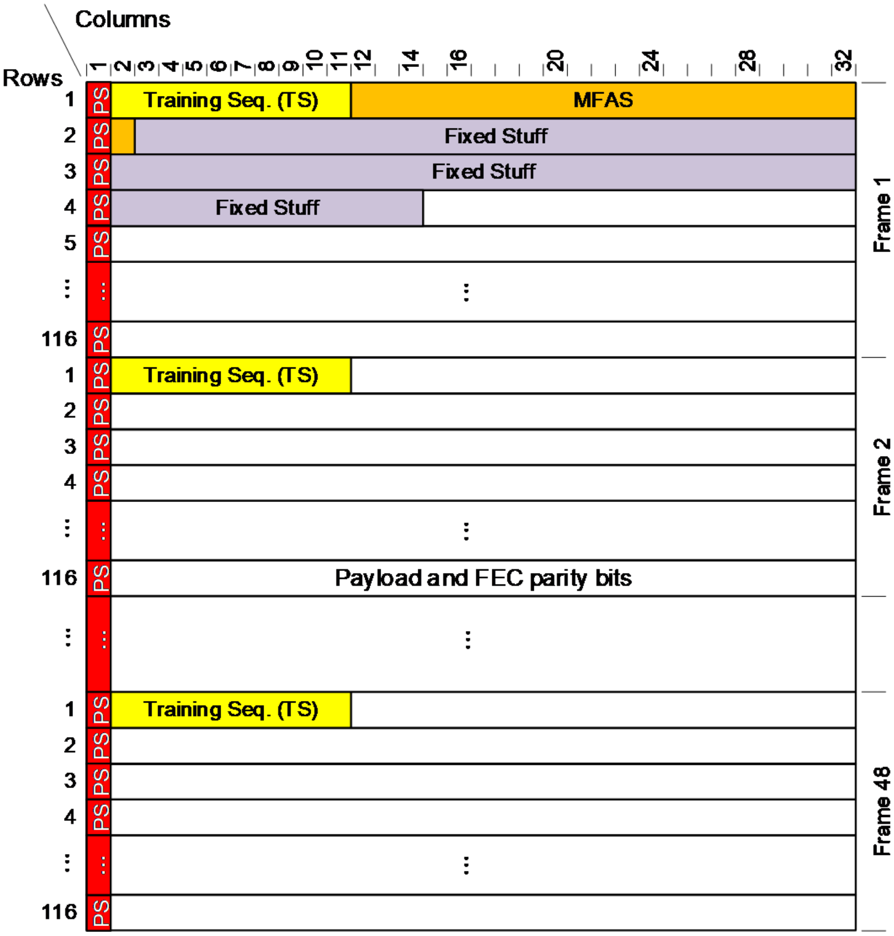
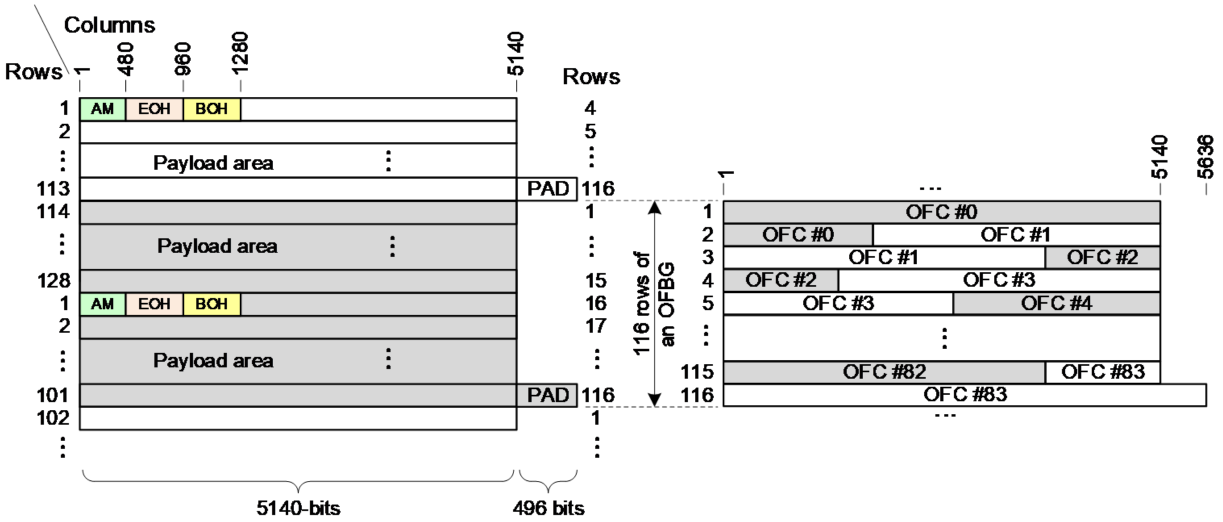


Figure 16-8. OFBG4 Superimposed on FlexO-1 Multiframe Structure



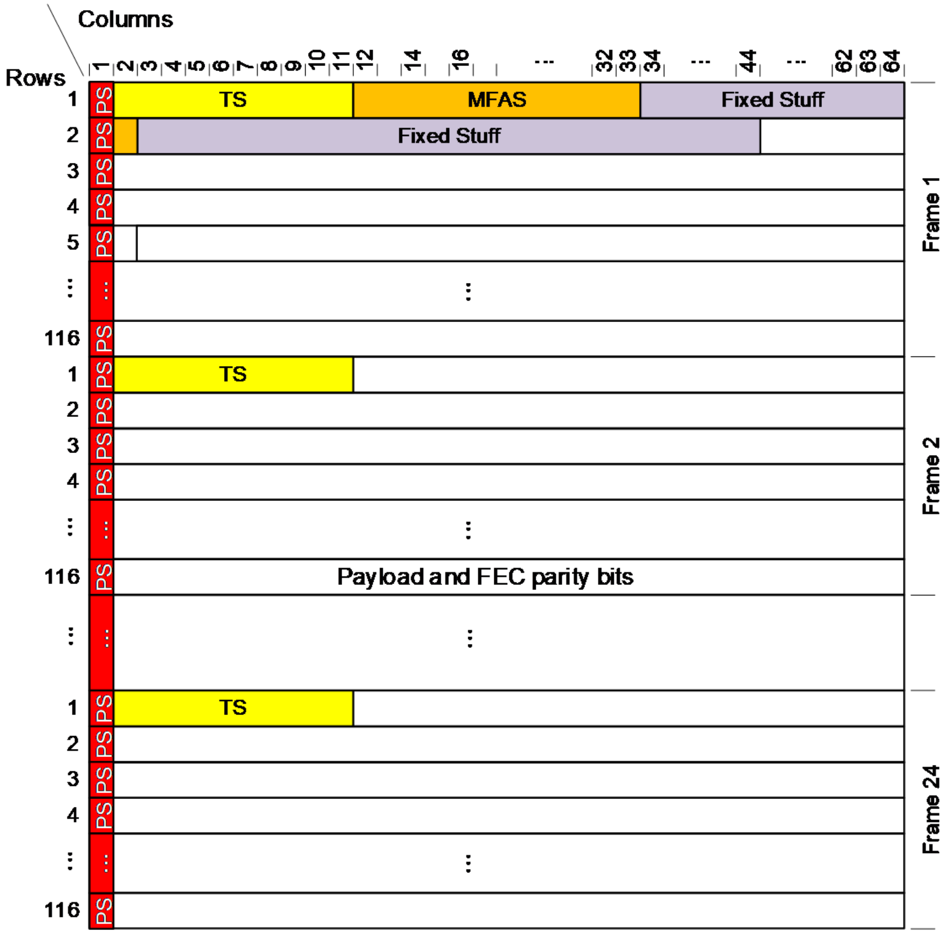
16.3. FlexO-x-DO in G.709.6

The FlexO-x-DO multiframe structure used by G.709.6 is illustrated in Figure 16-9.

As with G.709.3, the OFBG is superimposed on the underlying FlexO-x(e) multiframe structure, as illustrated in Figure 16-10. As illustrated in this figure, padding is added after every four 10280-bit[63] information blocks in order to preserve the FlexO alignment. Unlike the G.709.3 case (see the G.709.3 figure), the padding in G.709.6 includes a CRC-32 that the receiver can optionally use for error marking.

The 400G FlexO-4-DO interfaces are specified to use QPSK modulation (FlexO-4-DO-QPSK and FlexO-4e-DO-QPSK) while the 800G FlexO-8-DO interfaces are specified to use 16QAM modulation (FlexO-8-DO-16QAM and FlexO-8e-DO-16QAM).

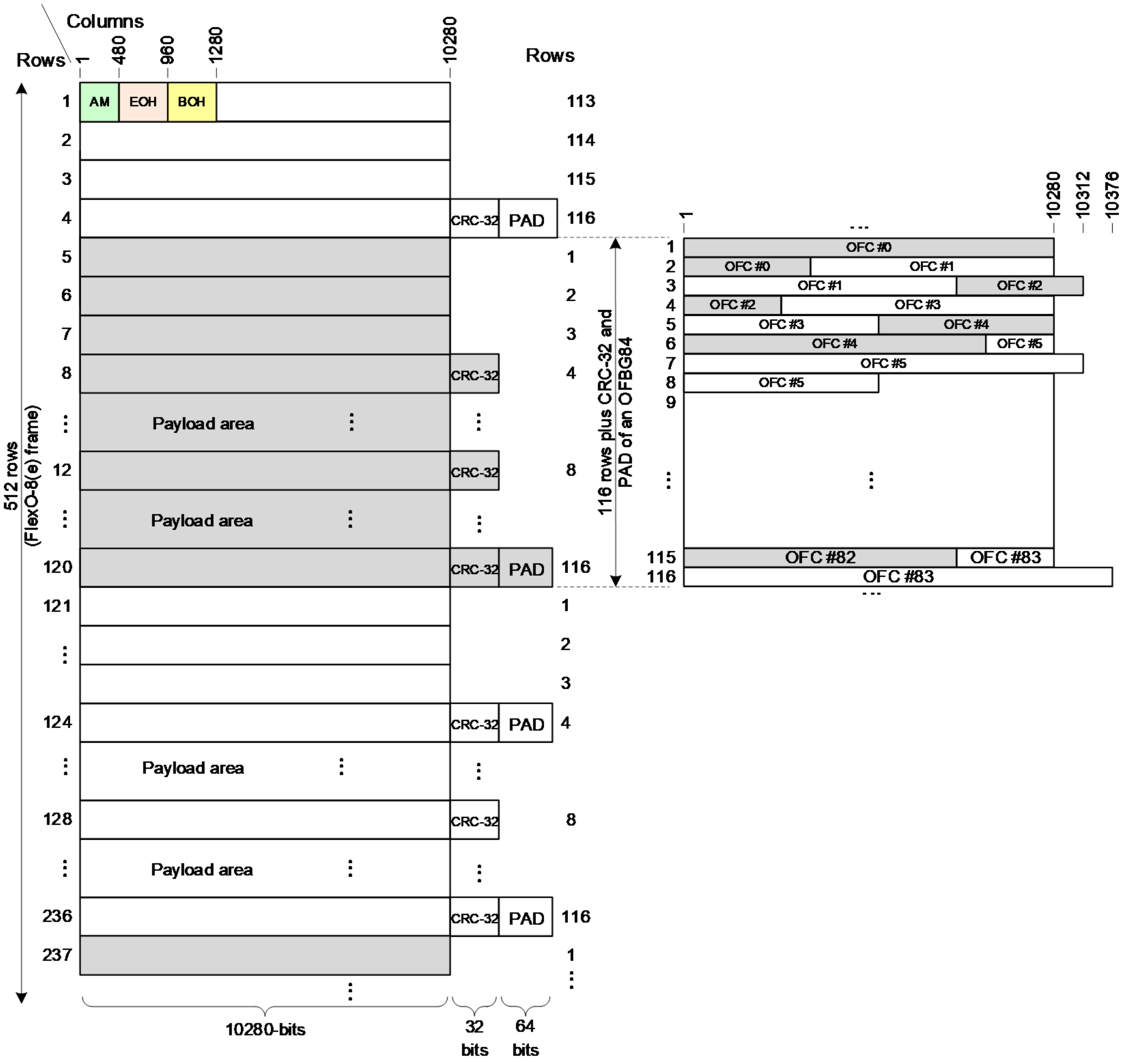
Figure 16-9. FlexO-x-DO 24-frame Multiframe Structure for G.709.6



---

[63] 10280 = 40 × 275

Figure 16-10. OFBG84 Superimposed on FlexO-8(e) Multiframe Structure



17. Appendix C - Fine Grain (sub-Gbit/s) OTN (fgOTN)

As legacy SDH systems are being retired, network operators continue to provide ongoing support for traditional SDH clients by transporting these clients over their OTN networks. As an example, the Microchip HyPHY-20Gflex OTN Processor can efficiently map OC-3/12/48, STM-1/4/16, FE/GE, Fibre Channel, ESCON, and other clients into OTN. However, some network operators requested adding efficient mapping of lower-rate SDH clients directly into OTN. In other words, the application driving fgOTN was efficient OTN transport of CBR and packet clients with sub-Gbit/s rates. The important clients of interest included legacy 155 Mbit/s SDH and 2 Mbit/s E1 CBR clients, and packet clients coming from a 10 Mbit/s or 100 Mbit/s Ethernet UNI. After studying the application requirements, it was agreed that the effective channel bandwidth granularity should be ≈10.4 Mbit/s, which is enough bandwidth to efficiently carry a native 10 Mbit/s Ethernet packet stream.

Significant challenges to reusing OTN techniques[64] led to considering alternative approaches[65]. However, two key insights made it feasible to adopt an approach that leveraged, extended and supplemented elements of OTN. The method, which was added to G.709 in late 2023, is summarized in this appendix. Key features of fgOTN include support for both CBR and packet clients, and not requiring low pass filtering at intermediate nodes.

At a high level, the solution consists of:

- A method to map packet and CBR clients into an ODU-like structure called a fine grain ODUflex (fgODUflex).
- A method to add a 10 Mbit/s fine grain TS (fgTS) structure to an OPUk (k = 0, 1, 2, flex) server in order to carry the encapsulated clients.

17.1. Client Mapping into fgODUflex

The fgODUflex uses a slightly modified version of the ODUk frame format. As illustrated in the figure below, frame columns 1-16 continue to carry traditional ODU and OPU overhead functions with some rearrangement (as shown in the figure below). In order to increase the overhead bandwidth and reduce its latency, columns 1905-1920 carry overhead similar to columns 1-16. Comparing with Figure 7-4 above, the key modifications include:

- FAS is carried per row at the beginning of both overhead column groups.
- Only two TCM levels are supported.
- The TTI, DM and STAT fields have been increased from a few bits to having their own bytes.
- Difference Accumulation (DA) fields have been added, which are explained below.

The fgODUflex Tributary Slot overhead (TSOH) and payload area structures are illustrated in Figure 17-2.

In order to accommodate client and server clock tolerances, the nominal fgODUflex(p) rate and resulting fgOPUflex(p) rate was chosen to be:

- fgODUflex(p) rate = (p / 119.5) × 1.244160 Gbit/s = p × 10.409203 Mbit/s
- fgOPUflex(p) channel rate = (237 / 239) × (fgODUflex(p) rate) = p × 10.322097 Mbit/s

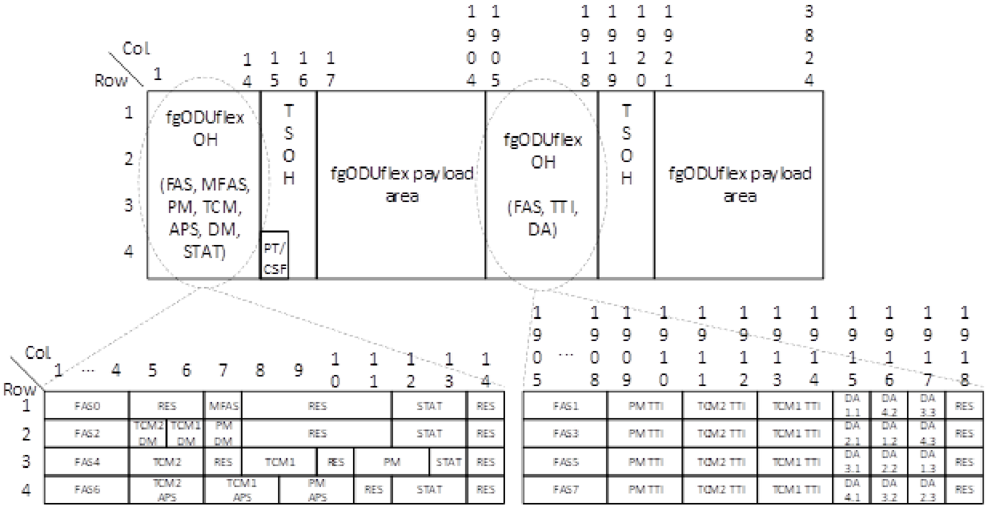
---

[64] One challenge was that having nearly 120 channels in an OPU0 would imply a 120-frame JC multiframe, and multiples of this with an OPU1, OPU2 or OPUflex server. Another challenge is that ODUk switching requires low-pass filtering of the ODUk at intermediate nodes in order to limit the accumulated jitter at the ODUk sink node. This would become unreasonably complex with large number of fine grain clients.

[65] One initial proposal added an SDH-type structure into the OPU (ITU-T G.Sup70). Another proposal was cell-based, with some similarities to Asynchronous Transfer Mode (ATM). Both were relatively complex and difficult to extend

where p is the number of fgTS that the fgODUflex will occupy in the server.

Figure 17-1. fgODUflex Frame Format



17.1.1. CBR Clients

The TSOH area illustrated in Figure 17-3 below distributes the JC bytes across the two sets of TSOH columns such that, as indicated by the horizontal dashed line, the GMP frame repeats every two frame rows rather than across one or more frames of a multiframe. This JC byte spacing maintains the burst error resilience that is critical for GMP. Although a 9-bit C_m would have been sufficient, the JC1-JC6 format was chosen to be identical to that of an ODUk (see Figure 8-13).

The GMP mapping granularity (word size) is fixed at 128 bits (16 bytes). The presence of 16 overhead columns (1905-1920) results in 237 GMP word positions per row, giving 474 positions per 2-row GMP frame (i.e., $P_{server} = 474$). Having two independent payload containers per row allowed for sharing a fgODUflex payload area across up to four 2 Mbit/s E1 clients.

In order for an ODU path sink to correctly interpret its client mapping GMP overhead field, it must know/derive the rate (i.e., the GMP frame period) of that ODU path signal. This requires that the GMP overhead must be terminated and regenerated with appropriate low-pass filtering at each ODUk switching node along the ODUk path. The solution to this avoiding this switch node filtering with fgODUflex was one of the key insights that allowed using a GMP-based ODU-type format for the client mapping with fgOTN[66]. The approach involves adding Difference Accumulator (DA) overhead fields to the fgODUflex that allow accumulating the relative clock offset between adjacent fgODUflex processing nodes along the fgODUflex path. The approach functions as follows:

- Each node has its own independent local reference clock to which all its outgoing OTUk signals are phase locked
- The source node encodes its local clock as the GMP C_m and C_{nD} fields in the fgOPUflex overhead and inserts DA=0 (i.e., its outgoing OTUk signal is the first one to carry the fgODUflex and hence has no frequency offset since the fgODUflex rate and egress OTUk rates are both phase locked to the source's local reference clock).
- Each node along the path determines the normalized difference between the clocks of the ingress OTUk and egress OTUk carrying an fgODUflex and adds the difference to the fgODUflex DA value.
- When the fgODUflex reaches the sink, it compares received OTUk rate and DA value to its own local clock to determine the source node clock.

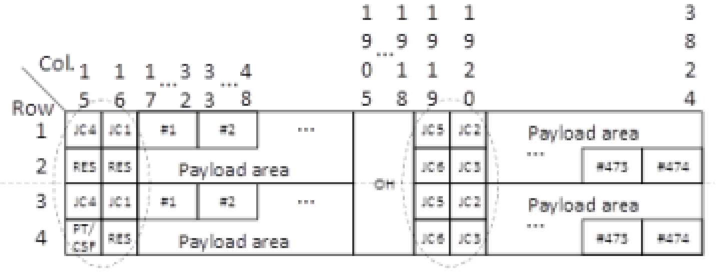
---

[66] This solution was identified, analyzed and proposed by Microchip.

For implementation simplicity, the frequency difference value is encoded into the DA as phase difference (i.e., the difference between number of clock cycles between the two clocks in a sampling interval). Since each node's sampling interval and egress OTUk rates are derived from its local clock, it knows the normalized number of egress OTUk clock cycles in the interval. It compares the normalized number ingress OTUk clock cycles in the sampling interval to this expected value and adds the difference (two's-complement) to the incoming fgODUflex DA value to create the DA it sends on the egress OTUk. Not only is the ingress/egress phase difference much less complex than low pass filtering at the intermediate node, it is further simplified by the number of ingress/egress OTUk pairs typically being much smaller than the potential number of fgODUflex clients[67]. See additional details in the Packet Clients section.

Specifically, the intermediate nodes update the fgODUflex DA field to indicate the accumulating frequency offset. Specifically, an fgODUflex switch or sink node recovers the clock of the ingress ODUk(fgTS) or ODUflex(fgTS,n) server signals described below. For consistency across all nodes, this clock is scaled to a normalized value of 311.04 MHz, which is referred to as the Normalized Multiplexing Server clock (NMSCk). The fgODUflex signals are demultiplexed from the ingress ODU server and re-multiplexed into the egress ODU server. The egress ODU server clock rate is phase locked to that node's local clock, and the fgODUflex DA field is updated as described above in order to reflect the difference between the ingress and egress ODU servers. The same receive functions are performed at the fgODUflex sink node, which uses the difference between the recovered received ODU server rate and its own local clock to determine the final DA update. The CBR client's clock is then recovered based on the combination of the fgODUflex C_m and C_{nD} fields, its local REFCLK and the DA information that provides the offset between the source and sink REFCLK.

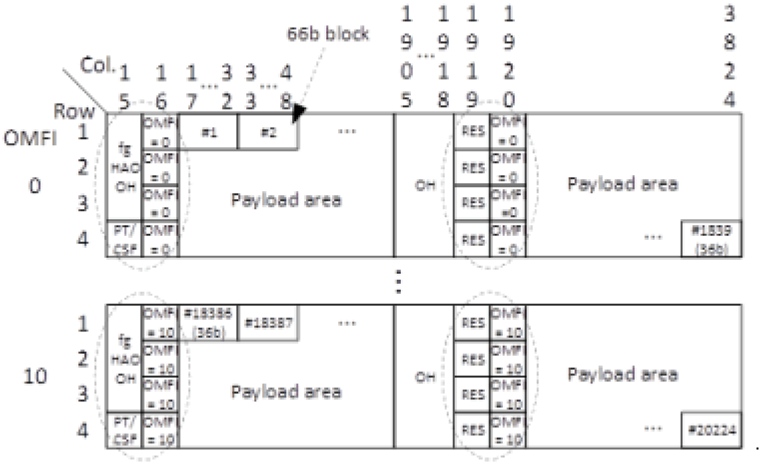
Figure 17-2. fgODUflex TSOH and Payload Area for CBR Clients



---

[67] For example, an intermediate node with 10 OTUk interfaces would have 90 ingress/egress pairs, while a single OPU0 could have up to 119 client fgODUflex.

Figure 17-3. fgODUflex TSOH and Payload Area for Packet Clients



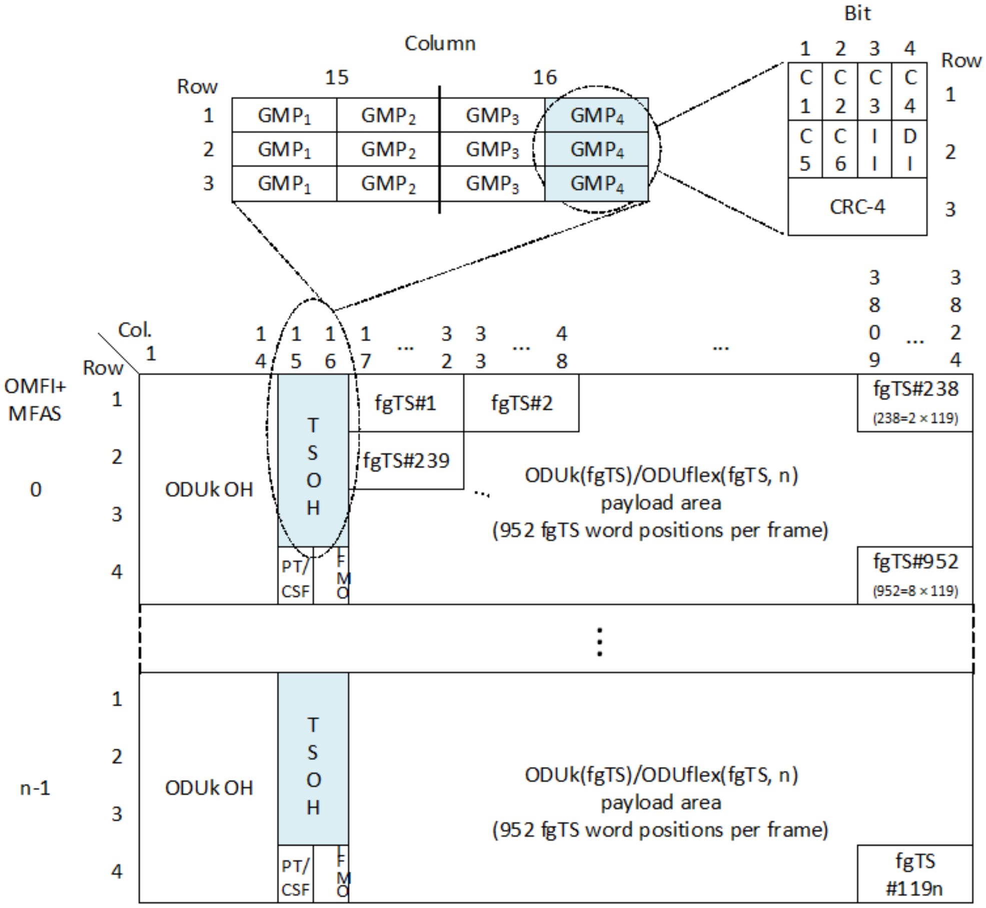
17.1.2. Packet Clients

Packet clients are encoded and mapped as a stream of IEEE 802.3 clause 82 64B/66B blocks. The block alignments are illustrated in Figure 17-3. The 11-frame multiframe allows the first bit of the first block 64B/66B block to align with the first payload area bit of the multiframe and the last bit of the last 64B/66B block (#20224) to align with the last payload area bit of multiframe. The last block of all rows except the last one will be split across two rows. Since JC bytes are not required for packet client mapping, these bytes of column 16 contain a 4-bit (right-justified) OMFI field. Table M.5 of G.709 shows the mapping between the combination of the OMFI and MFAS row number and the bit offset in each row before the start of the first full block in that row.

17.2. OPU0 Server Format for fgOTN

For the server layer ODU, there are 3808 = 32 × 119 byte-columns in a normal OPU payload area. Conveniently, dividing the OPU0 capacity rate by 119 yields a rate of (1.244160 Gbit/s / 119) = 10.455 Mbit/s, which provides a very efficient fgTS structure and channel size for carrying a 10 Mbit/s Ethernet client. The corresponding OPU0 fgOTN server format is shown in Figure below

Figure 17-4. ODUflex(fgTS, n) Server Frame Format for fgOTN



The challenge with using GMP for multiplexing fgODUflex clients into the OPU0 server is that the traditional approach would require a 119-frame multiframe, with 238-frame for OPU1, etc. for OPU2 and OPUflex servers. This would result in very slow GMP response times. The key insight behind the solution to this problem is that, as explained above, the base fgODUflex rate is fixed for all clients using the same number of server fgTS. Consequently, the GMP count MSBs will be unchanged across all GMP periods. This situation allowed defining an OPUk C_m base value (C_mB) and using the GMP overhead to only communicate the small difference between the actual C_m associated with this client fgODUflex and its base value ($C_m – C_mB = C_mT$[68]).

It was determined that a 6-bit C_mT is adequate. It was further determined that the GMP ΣCnD is not needed for this application. Consequently, as illustrated in Figure 17-4, this allowed carrying the GMP overhead for four fgTS in each OPUk server frame, making the multiframe lengths reasonable.

A 16-byte word size is used, resulting in 238 fgTS locations per row.

Note that hitless resizing is supported for fgODUflex signals carrying packet clients. In order to reduce complexity and provide faster resizing, the approach uses a streamlined version of the HAO described in section 6.3.4 above for ODUflex. The specifics, which can be found in G.709 Annex O, are beyond the scope of this white paper.

---

[68] The CmT name comes from it originally being described as the “Top off” value that is added to the CmB to indicate the actual Cm.

18. Appendix D: References and Standards Related to OTN

The date provided for these standards is the release dates of their current base versions. Most have been subsequently updated with amendments and/or corrigenda or errata. See https://www.itu.int/rec/T-REC-G/en for the most recent versions.

Core OTN Standards

- ITU-T Rec. ITU-T G.709 (2020), *Interfaces for the optical transport network.* This standard defines the OTN digital signal format, including rates, overhead, how client signals are mapped and multiplexed into OTN.
- ITU-T Rec. ITU-T G.798 (2023), *Characteristics of optical transport network hierarchy equipment functional blocks*. This defines the functional requirements for network equipment that implements G.709.
- ITU-T Rec. G.709.1 (2024), *Flexible OTN common elements*. Defines a physical layer signal for carrying OTN signals at rates beyond 100 Gbit/s. The companion G.709.3, G.709.5 and G.709.6 Recommendations define the FEC frame format for different Flexible OTN (FlexO) applications.
- ITU-T G.Sup58 (2024) *Optical transport network module framer interfaces*. Defines the electrical interfaces between OTN framer and optical interface modules.

Additional OTN-related Standards

- ITU-T Rec. G.709.2 (2018), *OTU4 long-reach interface*.Defines an FEC structure to support longer reach OTU4 interfaces.
- ITU-T Rec. G.709.3 (2024), *Flexible OTN long-reach interfaces*. Defines FEC structures for longer reach FlexO interfaces at rates between 100 and 400 Gbit/s.
- ITU-T Rec. G.709.4 (2020), *OTU25 and OTU50 short-reach interfaces*. Defines OTN interfaces for 25 and 50 Gbit/s OTN signals that can reuse Ethernet optical modules at the same nominal rates.
- ITU-T Rec. G.709.5 (2024), *Flexible OTN short-reach interfaces*. Defines FEC structures for short reach FlexO interfaces at any rate. Note that this structure is used for all FOIC interfaces.
- ITU-T Rec. G.709.6 (2024), *Flexible OTN B400G long-reach interfaces*. Defines FEC structures for longer reach for FlexO interfaces with rates at and beyond 400 Gbit/s
- ITU-T Rec. G.806 (2022), *Characteristics of transport equipment - Description methodology and generic functionality - Amendment 1*. Provides the basis for G.798 at a higher and more generic level.
- ITU-T Rec. G.872 (2022), *Architecture of the optical transport network*. High-level description of optical transport network architectures.
- ITU-T Rec. G.873.1 (2022), *Optical transport network: Linear protection*.
- ITU-T Rec. G.8201 (2011), *Error performance parameters and objectives for multi-operator international paths within optical transport networks*.

Standards related to important clients and their transport over OTN

- IEEE 802.3-2022, *Standard for Ethernet*
- OIF, Flex *Ethernet 2.0 Implementation Agreement* (2018).
- ITU-T Rec. G.7041 (2016), *Generic framing procedure (GFP)*.
- ITU-T Rec. G.7044 (2011), *Hitless Adjustment of ODUflex(GFP)*.
- ITU-T Rec. G.8021 (2022), *Characteristics of Ethernet transport network equipment functional blocks*.
- ITU-T Rec. G.8023 (2018), *Characteristics of equipment functional blocks supporting Ethernet physical layer and Flex Ethernet interfaces*.

Timing-related standards

- ITU-T Rec. G.8251 (2022), *The control of jitter and wander within the optical transport network (OTN)*.
- ITU-T Rec. G.8260 (2022), *Definitions and terminology for synchronization in packet networks*.
- ITU-T Rec. G.8261 (2022), *Timing and synchronization aspects in packet networks*.
- ITU-T Rec. G.8262 (2022), *Timing characteristics of synchronous equipment slave clock*.
- ITU-T Rec. G.8265.1 (2022), *Precision time protocol telecom profile for frequency synchronization*.

Standards for the optical channels that carry OTN

- ITU-T Rec. G.695 (2018), *Optical interfaces for coarse wavelength division multiplexing applications*.
- ITU-T Rec. G.959.1 (2024), *Optical transport network physical layer interfaces*.

Additional informative references

- OIF FlexE Application Note: http://www.oiforum.com/documents/educational-white-papers/.
- National Institute of Standards and Technology (2007), Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. [NIST SP 800-38D].

19. Glossary and Abbreviations

This Annex defines commonly used abbreviations and acronyms associated with OTN and FlexO, along with pointers to the sections where additional definition of the terms are provided.

Term	Definition
3R	Re-amplification, Reshaping and Retiming
10GbE	10 Gbit/s Ethernet
40GbE	40 Gbit/s Ethernet
100GbE	100 Gbit/s Ethernet
ADM	Add/Drop Multiplexer
AES	Advanced Encryption Standard (from NIST)
AIS	Alarm Indication Signal
AM	Alignment Marker (Ethernet and FlexO)
AMP	Asynchronous Mapping Procedure
APS/PCC	Automatic Protection Switching / Protection Communications Channel
AVAIL	The number of "available" and valid OTUC slices that are mapped into the FlexO frame for that PHY
B1T	Related to rates beyond 1 Tbit/s
B100G	Related to rates beyond 100 Gbit/s
BDI	Backward Defect Indication
BEI	Backward Error Indication
BIAE	Backward Incoming Alignment Error
BIP-8	8-bit Bit Interleaved Parity
BMP	Bit-Synchronous Mapping Procedure
BOH	Basic FlexO overhead
CBR	Constant Bit Rate
CDI	Client Defect Indication
C_m	Number of data words transmitted in the next GMP frame period
CMx	Portions of the Alignment Markers common to all lanes
$\sum C_{nD}$	GMP phase offset encoding: Count of the Difference in the number of bits (bytes) that could be transmitted as a whole GMP word and the bits (bytes) remaining untransmitted at the mapper (multiplexer)
CO	Central Office (of a telephone network provider)
CPE	Customer Premise Equipment
CRC-*n*	*n*-bit Cyclic Redundancy Check error detection code
CSF	Client Signal Fail
CST	Cipher Suite Type
CSTAT	Ethernet client status overhead in FlexO
CWDM	Coarse Wavelength Division Multiplexing
DA	Difference Accumulation (Used with fgOTN)
Deskew	To straighten a skewed image or document.
DFB	Distributed Feedback laser
DM	Delay Measurement overhead

Glossary and Abbreviations (continued)

Term	Definition
DWDM	Dense Wavelength Division Multiplexing
DSP	Digital Signal Processor
EDFA	Erbium-Doped Fiber Amplifier
EOH	Extended FlexO overhead
FBA	FEC Block Alignment
FCC	FlexO Communications Channel
FEC	Forward Error Correction
fgODUflex	Fine grain ODUflex
fgOTN	Fine grain OTN
fgTS	Fine grain TS (≈10.4 Mbit/s)
FlexE	Flexible Ethernet (from OIF)
FlexO	Flexible OTN (G.709.1)
FOIC	FlexO Interface
FlexOsec	FlexO security
GCC	General Communications Channel (in the OTU and ODU overhead)
GCM	Galois/Counter Mode block cipher for authenticated encryption
GFP	Generic Framing Procedure (ITU-T Rec. G.7041)
GFP-T	Transparent mode of GFP
GID	FlexO Group ID
GMAC	Galois Message Authentication Code – authentication-only variant of GCM
GMP	Generic Mapping Procedure
HAO	Hitless Adjustment of ODUflex(GFP) signals
IaDI	Intra-Domain Interface
IAE	Incoming Alignment Error
IMP	Idle Mapping Procedure
IrDI	Inter-Domain Interface
ITU-T	International Telecommunications Telecom Standardization sector special agency of the United Nations
JC	Justification Control
JOH	Justification Overhead
KP4	Reed-Solomon RS(544,514,10) FEC from IEEE 802.3
LD	Ethernet Local Degrade indicator
LTE	Line layer Termination Equipment
MACsec	Ethernet Media Access Control (MAC) security protocol
MAP	FlexO overhead field for the OTUC to FlexO PHY mapping
MFAS	MultiFrame Alignment Signal
MFI	Module to Framer Interface
MS	Multiplex Section – Connection between multiplexer and demultiplexer
MSI	Multiplex Structure Identifier
NE	Network Element
NJO	Negative Justification Opportunity

Glossary and Abbreviations (continued)

Term	Definition
NMS	Network Management System
NNI	Node-Node Network Interface
NRZ	Non-Return to Zero 2-level line code
OA	Optical Alignment (OTN frame alignment field)
OADM	Optical Add-Drop Multiplexer
OCC	Optical Channel Carrier
OCh	Optical channel with full functionality
ODSP	Optical Digital Signal Processor
ODTU	Optical channel Data Tributary Unit
ODTUCn	ODTU for multiplexing an ODU signal into an OPUCn
ODTUCn.ts	ODTU for multiplexing an ODU into “ts” Tributary Slots of an OPUCn
ODTUjk	Optical channel Data Tributary Unit j into k
ODU	Optical Channel Data Unit
ODUC	100Gbit/s element (slice) of an ODUCn
ODUCn	*n* × 100Gbit/s Optical Channel Data Unit
ODUflex(CBR)	Flexible rate ODU for carrying CBR client signals
ODUflex(GFP)	Flexible rate ODU for carrying packet client signals that use a GFP-F mapping into the OPUflex
ODUflex(IMP)	Flexible rate ODU for carrying packet client signals with Ethernet Idle characters used for rate adaptation when mapping into the OPUflex
ODUk	Optical Channel Data Unit-k
ODUk-Xv	X virtually concatenated ODUk's
OH	Overhead
OIF	Optical Interworking Forum
OMFI	OPU Multiframe Indicator
OMS	Optical Multiplex Section
OpEx	Operating Expense
OPU	Optical Channel Payload Unit
OPUC	100Gbit/s element (slice) of an OPUCn
OPUCn	*n* × 100Gbit/s Optical Channel Payload Unit
OPUk	Optical Channel Payload Unit-k
OSC	Optical Supervisory Channel
OSMC	OTN Synchronization Message Channel
OTL	Optical Transport Lane
OTP	Optical Transport Platform
OTU	Optical Channel Transport Unit
OTUC	100Gbit/s element (slice) of an OTUCn
OTUCn	*n* × 100Gbit/s Optical Channel Transport Unit
OTUk	completely standardized Optical Channel Transport Unit-k
OXC	Optical cross-connect equipment
P-OTP	Packet Optical Transport Platform

Glossary and Abbreviations (continued)

Term	Definition
PAM-x	x-level Pulse Amplitude Modulation line code
Path	End-to-end connection between mapper and demapper
PID	FlexO PHY ID
PJO	Positive Justification Opportunity
PM	Path Monitoring
PMD	(Ethernet) Physical Medium Dependent sub-layer
PSI	Payload Structure Identifier
PT	Payload Type
PTE	Path Terminating Equipment
PW	Pseudo-Wire
Q11/15	Question 11 of ITU-T Study Group 15, which is the standards group responsible for OTN standardization
QAM	Quadrature Amplitude Modulation
RD	Ethernet Remote Degrade indicator
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RS	Regenerator Section (Connection between a terminal and a regenerator) or Reed-Solomon FEC
SAN	Storage Area Network
SD	Signal Degrade condition
SDH	Synchronous Digital Hierarchy (defined by the ITU-T in parallel with the North American SONET standard)
SM	Section Monitoring
SONET	Synchronous Optical Network (defined in North America in parallel with the ITU-T SDH standard)
STAT	STAT field provides monitoring status information for the associated layer
STE	Section (RS) Terminating Equipment
STM	Synchronous Transport Module structure within SDH signals
Tandem Connection	A portion of the Path with its own monitoring overhead
TC	Tandem Connection
TCM	Tandem Connection Monitoring
TDM	Time-Division Multiplexing
TPID	FlexO and OPUk Tributary Port Identifier
TS	Tributary Slot
TTI	Trail Trace Identifier Allows the sink to confirm that it has the correct connection to the source for that layer (i.e., to detect connectivity faults).
UMx	Portions of the Alignment Markers unique to each lane
UNI	User-Network Interface
WDM	Wavelength Division Multiplexing
WSS	Wavelength Selective Switch

20. Revision History

Revision A (January 2026)

Initial release of this document.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-2020-1

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.