International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.709/Y.1331

(02/2012)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – General

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Transport

Interfaces for the optical transport network

Recommendation ITU-T G.709/Y.1331

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.709/Y.1331

Interfaces for the optical transport network

Summary

Recommendation ITU-T G.709/Y.1331 defines the requirements for the optical transport module of order n (OTM-n) signals of the optical transport network, in terms of:

– optical transport hierarchy (OTH)
– functionality of the overhead in support of multi-wavelength optical networks
– frame structures
– bit rates
– formats for mapping client signals.

The first revision of this Recommendation included the text of Amendment 1 (ODUk virtual concatenation, ODUk multiplexing, backward IAE), extension of the physical interface specification, ODUk APS/PCC signal definition and several editorial enhancements.

The second revision of this Recommendation included the text of Amendments 1, 2 and 3, corrigenda 1 and 2, Erratum 1, the Implementers Guide (2005), support for an extended (unlimited) set of constant bit rate client signals, a flexible ODUk, which can have any bit rate and a bit-rate tolerance up to ±100 ppm, a client/server independent generic mapping procedure to map a client signal into the payload of an OPUk, or to map an ODUj signal into the payload of one or more tributary slots in an OPUk and the ODUk delay measurement capability.

The third revision of this Recommendation includes the text of Amendments 1 and 2, Corrigendum 1 and Erratum 1 to Recommendation ITU-T G.709/Y.1331 (2009) and support for more client signals.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.709/Y.1331	2001-02-09	15
1.1	ITU-T G.709/Y.1331 (2001) Amd.1	2001-11-29	15
2.0	ITU-T G.709/Y.1331	2003-03-16	15
2.1	ITU-T G.709/Y.1331 (2003) Amd.1	2003-12-14	15
2.2	ITU-T G.709/Y.1331 (2003) Cor.1	2006-12-14	15
2.3	ITU-T G.709/Y.1331 (2003) Amd.2	2007-11-22	15
2.4	ITU-T G.709/Y.1331 (2003) Cor.2	2009-01-13	15
2.5	ITU-T G.709/Y.1331 (2003) Amd.3	2009-04-22	15
3.0	ITU-T G.709/Y.1331	2009-12-22	15
3.1	ITU-T G.709/Y.1331 (2009) Cor.1	2010-07-29	15
3.2	ITU-T G.709/Y.1331 (2009) Amd.1	2010-07-29	15
3.3	ITU-T G.709/Y.1331 (2009) Amd.2	2011-04-13	15
4.0	ITU-T G.709/Y.1331	2012-02-13	15
4.1	ITU-T G.709/Y.1331 (2012) Cor. 1	2012-10-29	15
4.2	ITU-T G.709/Y.1331 (2012) Amd. 1	2012-10-29	15

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Page

Recommendation ITU-T G.709/Y.1331

Interfaces for the optical transport network

1 Scope

The optical transport hierarchy (OTH) supports the operation and management aspects of optical networks of various architectures, e.g., point-to-point, ring and mesh architectures.

This Recommendation defines the interfaces of the optical transport network to be used within and between subnetworks of the optical network, in terms of:

– optical transport hierarchy (OTH)
– functionality of the overhead in support of multi-wavelength optical networks
– frame structures
– bit rates
– formats for mapping client signals.

The interfaces defined in this Recommendation can be applied at user-to-network interfaces (UNI) and network node interfaces (NNI) of the optical transport network. It is recognized, for interfaces used within optical subnetworks, that aspects of the interface are optical technology dependent and subject to change as technology progresses. Therefore, optical technology dependent aspects (for transverse compatibility) are not defined for these interfaces to allow for technology changes. The overhead functionality necessary for operations and management of optical subnetworks is defined.

The second revision of this Recommendation introduced:

– support for an extended (unlimited) set of constant bit rate client signals;
– a flexible ODUk, which can have any bit rate and a bit-rate tolerance up to ±100 ppm;
– a client/server independent generic mapping procedure to map a client signal into the payload of an OPUk, or to map an ODUj signal into the payload of one or more tributary slots in an OPUk;
– ODUk delay measurement capability.

The third revision of this Recommendation introduces the text of Amendments 1 and 2, Corrigendum 1 and Erratum 1 to Recommendation ITU-T G.709/Y.1331 (2009) and support for more client signals.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.652] Recommendation ITU-T G.652 (2009), *Characteristics of a single-mode optical fibre and cable*.

[ITU-T G.653] Recommendation ITU-T G.653 (2006), *Characteristics of a dispersion-shifted single-mode optical fibre and cable*.

[ITU-T G.655] Recommendation ITU-T G.655 (2009), *Characteristics of a non-zero dispersion-shifted single-mode optical fibre and cable*.

[ITU-T G.693] Recommendation ITU-T G.693 (2009), *Optical interfaces for intra-office systems*.

[ITU-T G.695] Recommendation ITU-T G.695 (2009), *Optical interfaces for coarse wavelength division multiplexing applications*.

[ITU-T G.707] Recommendation ITU-T G.707/Y.1322 (2003), *Network node interface for the synchronous digital hierarchy (SDH)*.

[ITU-T G.780] Recommendation ITU-T G.780/Y.1351 (2008), *Terms and definitions for synchronous digital hierarchy (SDH) networks*.

[ITU-T G.798] Recommendation ITU-T G.798 (2006), *Characteristics of optical transport network hierarchy equipment functional blocks*.

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.

[ITU-T G.806] Recommendation ITU-T G.806 (2009), *Characteristics of transport equipment – Description methodology and generic functionality*.

[ITU-T G.870] Recommendation ITU-T G.870/Y.1352 (2004), *Terms and definitions for optical transport networks (OTN)*.

[ITU-T G.872] Recommendation ITU-T G.872 (2001), *Architecture of optical transport networks*.

[ITU-T G.873.1] Recommendation ITU-T G.873.1 (2006), *Optical Transport Network (OTN): Linear protection*.

[ITU-T G.873.2] Recommendation ITU-T G.873.2 (2012), *ODUk shared ring protection (SRP)*.

[ITU-T G.959.1] Recommendation ITU-T G.959.1 (2009), *Optical transport network physical layer interfaces*.

[ITU-T G.984.6] Recommendation ITU-T G.984.6 (2008), *Gigabit-capable passive optical networks (GPON): Reach extension*.

[ITU-T G.987.4] Recommendation ITU-T G.987.4 (2012), 10 *Gigabit-capable passive optical networks (XG-PON): Reach extension*.

[ITU-T G.7041] Recommendation ITU-T G.7041/Y.1303 (2003), *Generic framing procedure (GFP)*.

[ITU-T G.7042] Recommendation ITU-T G.7042/Y.1305 (2001), *Link capacity adjustment scheme (LCAS) for virtual concatenated signals*.

[ITU-T G.7044] Recommendation ITU-T G.7044/Y.1347 (2011), *Hitless Adjustment of ODUflex(GFP)*.

[ITU-T G.7714.1] Recommendation ITU-T G.7714.1/Y.1705.1 (2010), *Protocol for automatic discovery in SDH and OTN networks*.

[ITU-T G.8011.1] Recommendation ITU-T G.8011.1/Y.1307.1 (2009), *Ethernet private line service*.

[ITU-T I.432.1] Recommendation ITU-T I.432.1 (1999), *B-ISDN user-network interface – Physical layer specification: General characteristics*.

[ITU-T M.1400] Recommendation ITU-T M.1400 (2001), *Designations for interconnections among operators' networks*.

[ITU-T M.3100 Amd.3] Recommendation ITU-T M.3100 (1995) Amd.3 (2001), *Generic network information model – Amendment 3: Definition of the management interface for a generic alarm reporting control (ARC) feature*.

[ITU-T O.150] Recommendation ITU-T O.150 (1996), *General requirements for instrumentation for performance measurements on digital transmission equipment*.

[ITU-T T.50] Recommendation ITU-T T.50 (1992), *International Reference Alphabet (IRA) (Formely International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.

[IEEE 802.3] IEEE Std. 802.3:2008, *IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*.

[IEEE 802.3ba] IEEE Std 802.3ba-2010, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 4: Media Access Control Parameters, Physical Layers and Management Parameters for 40 Gb/s and 100 Gb/s Operation*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 Terms defined in [ITU-T G.780]:

– BIP-X

– network node interface

3.1.2 Terms defined in [ITU-T G.805]:

– adapted information (AI)

– characteristic information (CI)

– network

– subnetwork

3.1.3 Terms defined in [ITU-T G.870]:

– CBR10G

– CBR2G5

– CBR40G

– completely standardized OTUk (OTUk)

– connection monitoring end point (CMEP)

– functionally standardized OTUk (OTUkV)

– hitless activation/deactivation of a connection monitor

– inter-domain interface (IrDI)
– intra-domain interface (IaDI)
– link capacity adjustment scheme (LCAS)
– non associated overhead (naOH)
– OCC with full functionality (OCC)
– OCC with reduced functionality (OCCr)
– OCG with full functionality (OCG n)
– OCG with reduced functionality (OCG nr)
– ODUk path (ODUkP)
– ODUk TCM (ODUkT)
– optical carrier group of order n (OCG n[r])
– optical channel (OCh[r])
– optical channel carrier (OCC[r])
– optical channel data unit (ODUk)
– optical channel payload unit (OPUk)
– optical channel transport unit (OTUk[V])
– optical channel with full functionality (OCh)
– optical channel with reduced functionality (OChr)
– optical multiplex unit (OMU n, n ≥ 1)
– optical physical section of order n (OPSn)
– optical supervisory channel (OSC)
– optical transport hierarchy (OTH)
– optical transport module (OTM n[r].m)
– optical transport network (OTN)
– optical transport network node interface (ONNI)
– OTH multiplexing
– OTM overhead signal (OOS)
– OTM with full functionality (OTM n.m)
– OTM with reduced functionality (OTM-0.m, OTM-nr.m)

3.1.4 Terms defined in [ITU-T G.872]:

– optical multiplex section (OMS)
– optical transmission section (OTS)

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 ODUk.ts: The ODUk.ts is an increment of bandwidth which when multiplied by a number of tributary slots gives the recommended size of an ODUflex(GFP) optimized to occupy a given number of tributary slots of a higher order OPUk.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

16FS	16 columns with Fixed Stuff
3R	Reamplification, Reshaping and Retiming
ACT	Activation (in the TCM ACT byte)
AI	Adapted Information
AIS	Alarm Indication Signal
AMP	Asynchronous Mapping Procedure
API	Access Point Identifier
APS	Automatic Protection Switching
ASI	Asynchronous Serial Interface for DVB
BDI	Backward Defect Indication
BDI-O	Backward Defect Indication Overhead
BDI-P	Backward Defect Indication Payload
BEI	Backward Error Indication
BI	Backward Indication
BIAE	Backward Incoming Alignment Error
BIP	Bit Interleaved Parity
BMP	Bit-synchronous Mapping Procedure
CAUI	(Chip to) 100 Gb/s Attachment Unit Interface
CB	Control Block
CBR	Constant Bit Rate
CI	Characteristic Information
CM	Connection Monitoring
C_m	number of m-bit Client data entities
CMEP	Connection Monitoring End Point
CMGPON_D	Continuous Mode GPON Downstream
CMGPON_U2	Continuous Mode GPON Upstream 2
CMOH	Connection Monitoring Overhead
CMXGPON_D	Continuous Mode XGPON Downstream
CMXGPON_U2	Continuous Mode XGPON Upstream 2
C_n	number of n-bit client data entities
C_{nD}	difference between C_n and ($m/n \times C_m$)
CPRI	Common Public Radio Interface
CRC	Cyclic Redundancy Check
CS	Client Specific
CSF	Client Signal Fail

CTRL	Control word sent from source to sink
DAPI	Destination Access Point Identifier
DDR	Double Data Rate
DMp	Delay Measurement of ODUk path
DMti	Delay Measurement of TCMi
DNU	Do Not Use
DVB	Digital Video Broadcast
EDC	Error Detection Code
EOS	End Of Sequence
ESCON	Enterprise Systems Connection
EXP	Experimental
ExTI	Expected Trace Identifier
FAS	Frame Alignment Signal
FC	Fibre Channel
FC	Flag Continuation
FDI	Forward Defect Indication
FDI-O	Forward Defect Indication Overhead
FDI-P	Forward Defect Indication Payload
FEC	Forward Error Correction
GCC	General Communication Channel
GID	Group Identification
GMP	Generic Mapping Procedure
GPON	Gigabit-capable Passive Optical Networks
IaDI	Intra-Domain Interface
IAE	Incoming Alignment Error
IB	InfiniBand
IrDI	Inter-Domain Interface
JC	Justification Control
JOH	Justification Overhead
LCAS	Link Capacity Adjustment Scheme
LF	Local Fault
LLM	Logical Lane Marker
LSB	Least Significant Bit
MFAS	MultiFrame Alignment Signal
MFI	Multiframe Indicator
MS	Maintenance Signal
MSB	Most Significant Bit

MSI	Multiplex Structure Identifier
MST	Member Status
naOH	non-associated Overhead
NJO	Negative Justification Opportunity
NNI	Network Node Interface
NORM	Normal Operating Mode
NOS	Not_Operational Sequence
OCC	Optical Channel Carrier
OCCo	Optical Channel Carrier – overhead
OCCp	Optical Channel Carrier – payload
OCCr	Optical Channel Carrier with reduced functionality
OCG	Optical Carrier Group
OCGr	Optical Carrier Group with reduced functionality
OCh	Optical Channel with full functionality
OChr	Optical Channel with reduced functionality
OCI	Open Connection Indication
ODTUG	Optical channel Data Tributary Unit Group
ODTUjk	Optical channel Data Tributary Unit j into k
ODTUk.ts	Optical channel Data Tributary Unit k with ts tributary slots
ODU	Optical channel Data Unit
ODUk	Optical channel Data Unit-k
ODUk.ts	Optical channel Data Unit k fitting in ts tributary slots
ODUkP	Optical channel Data Unit-k Path monitoring level
ODUkT	Optical channel Data Unit-k Tandem connection monitoring level
ODUk-Xv	X virtually concatenated ODUks
OH	Overhead
OMFI	OPU Multi-Frame Identifier
OMS	Optical Multiplex Section
OMS-OH	Optical Multiplex Section Overhead
OMU	Optical Multiplex Unit
ONNI	Optical Network Node Interface
OOS	OTM Overhead Signal
OPS	Optical Physical Section
OPSM	Optical Physical Section Multilane
OPU	Optical channel Payload Unit
OPUk	Optical channel Payload Unit-k
OPUk-Xv	X virtually concatenated OPUks

OSC	Optical Supervisory Channel
OTH	Optical Transport Hierarchy
OTL	Optical channel Transport Lane
OTLC	Optical Transport Lane Carrier
OTLCG	Optical Transport Lane Carrier Group
OTM	Optical Transport Module
OTN	Optical Transport Network
OTS	Optical Transmission Section
OTS-OH	Optical Transmission Section Overhead
OTU	Optical channel Transport Unit
OTUk	completely standardized Optical channel Transport Unit-k
OTUkV	functionally standardized Optical channel Transport Unit-k
OTUk-v	Optical channel Transport Unit-k with vendor specific OTU FEC
PCC	Protection Communication Channel
P-CMEP	Path-Connection Monitoring End Point
PCS	Physical Coding Sublayer
PJO	Positive Justification Opportunity
PLD	Payload
PM	Path Monitoring
PMA	Physical Medium Attachment sublayer
PMI	Payload Missing Indication
PMOH	Path Monitoring Overhead
PN	Pseudo-random Number
POS	Position field
ppm	parts per million
PRBS	Pseudo Random Binary Sequence
PSI	Payload Structure Identifier
PT	Payload Type
QDR	Quad Data Rate
RES	Reserved for future international standardization
RF	Remote Fault
RS	Reed-Solomon
RS-Ack	Re-sequence Acknowledge
SAPI	Source Access Point Identifier
SBCON	Single-Byte command code sets Connection
SDI	Serial Digital Interface
SDR	Single Data Rate

Sk	Sink
SM	Section Monitoring
SMOH	Section Monitoring Overhead
SNC	Subnetwork Connection
SNC/I	Subnetwork Connection protection with Inherent monitoring
SNC/N	Subnetwork Connection protection with Non-intrusive monitoring
SNC/S	Subnetwork Connection protection with Sublayer monitoring
So	Source
SQ	Sequence Indicator
TC	Tandem Connection
TC-CMEP	Tandem Connection-Connection Monitoring End Point
TCM	Tandem Connection Monitoring
TCMOH	Tandem Connection Monitoring Overhead
TS	Tributary Slot
TSOH	Tributary Slot Overhead
TTT	Timing Transparent Transcoding
TxTI	Transmitted Trace Identifier
UNI	User-to-Network Interface
VCG	Virtual Concatenation Group
VCOH	Virtual Concatenation Overhead
vcPT	virtual concatenated Payload Type
XGPON	10 Gigabit-capable Passive Optical Networks

5 Conventions

This Recommendation uses the following conventions defined in [ITU-T G.870]:

– k

– m

– n

– r.

The functional architecture of the optical transport network as specified in [ITU-T G.872] is used to derive the ONNI. The ONNI is specified in terms of the adapted and characteristic information present in each layer as described in [ITU-T G.805].

Transmission order: The order of transmission of information in all the diagrams in this Recommendation is first from left to right and then from top to bottom. Within each byte the most significant bit is transmitted first. The most significant bit (bit 1) is illustrated at the left in all the diagrams.

Mapping order: The serial bit stream of a constant bit rate signal is inserted into the OPU payload so that the bits will be transmitted on the OPU/ODU in the same order that they were received at the input of the AMP, BMP or GMP mapper function. If m bits b_a, b_b, b_c up to b_m are client signal bits of which b_a is the bit that is received first and b_m is the bit that is received last, then b_a will be mapped into bit 1 of a first OPU byte and b_m will be mapped into bit 8 of an n^{th} OPU byte (with $n = m/8$).

Value of reserved bit(s): The value of an overhead bit, which is reserved or reserved for future international standardization shall be set to "0".

Value of non-sourced bit(s): Unless stated otherwise, any non-sourced bits shall be set to "0".

OTUk, ODUk and OPUk overhead assignment: The assignment of an overhead in the optical channel transport/data/payload unit signal to each part is defined in Figure 5-1.

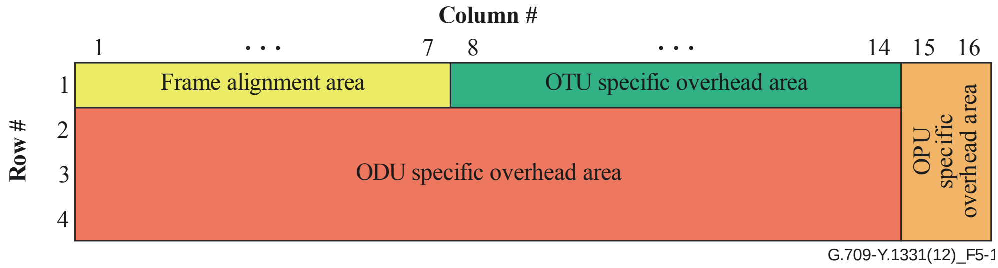


Figure 5-1 – OTUk, ODUk and OPUk overhead

6 Optical transport network interface structure

The optical transport network as specified in [ITU-T G.872] defines two interface classes:

- inter-domain interface (IrDI)
- intra-domain interface (IaDI).

The OTN IrDI interfaces are defined with 3R processing at each end of the interface.

The optical transport module-n (OTM-n) is the information structure used to support OTN interfaces. Two OTM-n structures are defined:

- OTM interfaces with full functionality (OTM-n.m)
- OTM interfaces with reduced functionality (OTM-0.m, OTM-nr.m, OTM-0.mvn).

The reduced functionality OTM interfaces are defined with 3R processing at each end of the interface to support the OTN IrDI interface class.

6.1 Basic signal structure

The basic structure is shown in Figure 6-1.

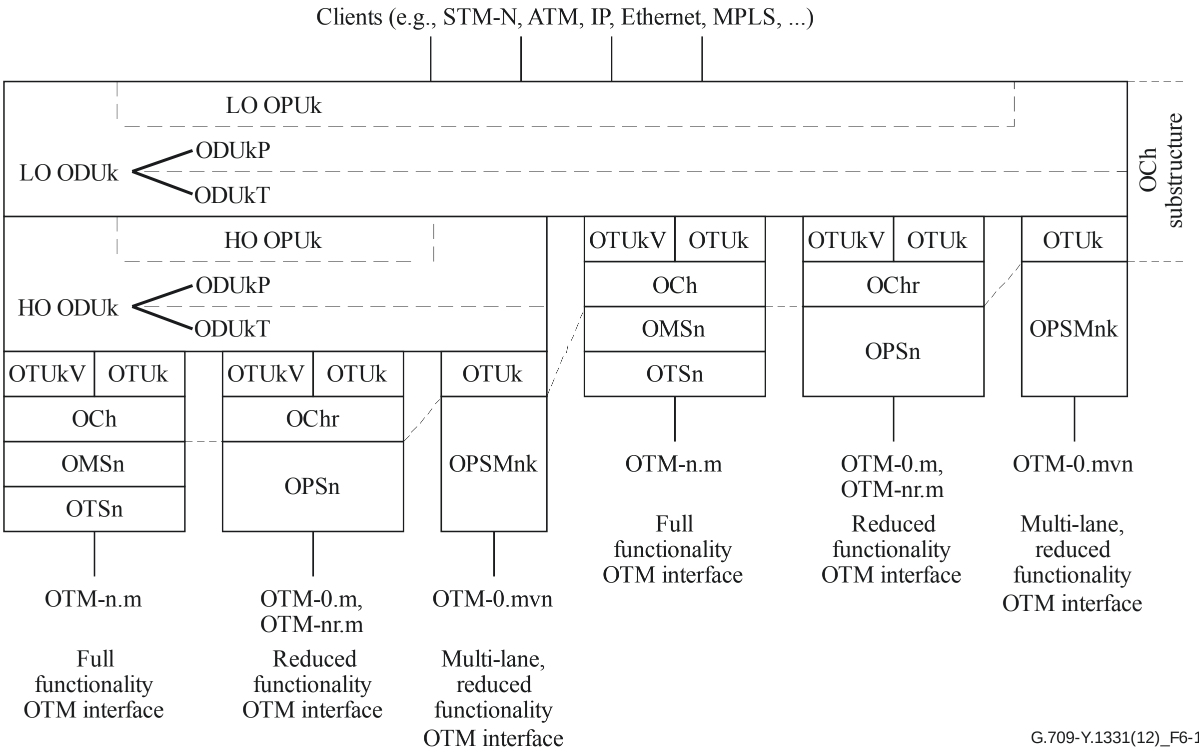


Figure 6-1 – Structure of the OTN interfaces

6.1.1 OCh substructure

The optical channel layer as defined in [ITU-T G.872] is further structured in layer networks in order to support the network management and supervision functionalities defined in [ITU-T G.872]:

– The optical channel with full (OCh) or reduced functionality (OChr), which provides transparent network connections between 3R regeneration points in the OTN.

– The completely or functionally standardized optical channel transport unit (OTUk/OTUkV) which provides supervision and conditions the signal for transport between 3R regeneration points in the OTN.

– The optical channel data unit (ODUk) which provides:
  - tandem connection monitoring (ODUkT)
  - end-to-end path supervision (ODUkP)
  - adaptation of client signals via the optical channel payload unit (OPUk)
  - adaptation of OTN ODUk signals via the optical channel payload unit (OPUk).

6.1.2 Full functionality OTM-n.m (n ≥ 1) structure

The OTM-n.m (n ≥ 1) consists of the following layers:

- optical transmission section (OTSn)
- optical multiplex section (OMSn)
- full functionality optical channel (OCh)
- completely or functionally standardized optical channel transport unit (OTUk/OTUkV)
- one or more optical channel data unit (ODUk).

6.1.3 Reduced functionality OTM-nr.m and OTM-0.m structure

The OTM-nr.m and OTM-0.m consist of the following layers:

- optical physical section (OPSn)

- reduced functionality optical channel (OChr)
- completely or functionally standardized optical channel transport unit (OTUk/OTUkV)
- one or more optical channel data unit (ODUk).

6.1.4 Parallel OTM-0.mvn structure

The OTM-0.mvn consists of the following layers:

- optical physical section (OPSMnk)
- completely standardized optical channel transport unit (OTUk)
- one or more optical channel data unit (ODUk).

6.2 Information structure for OTN interfaces

The information structure for OTN interfaces is represented by information containment relationships and flows. The principal information containment relationships are described in Figures 6-2, 6-3, 6-4 and 6-5. The information flows are illustrated in Figure 6-6.

For supervision purposes in the OTN, the OTUk/OTUkV signal is terminated whenever the OCh signal is terminated.

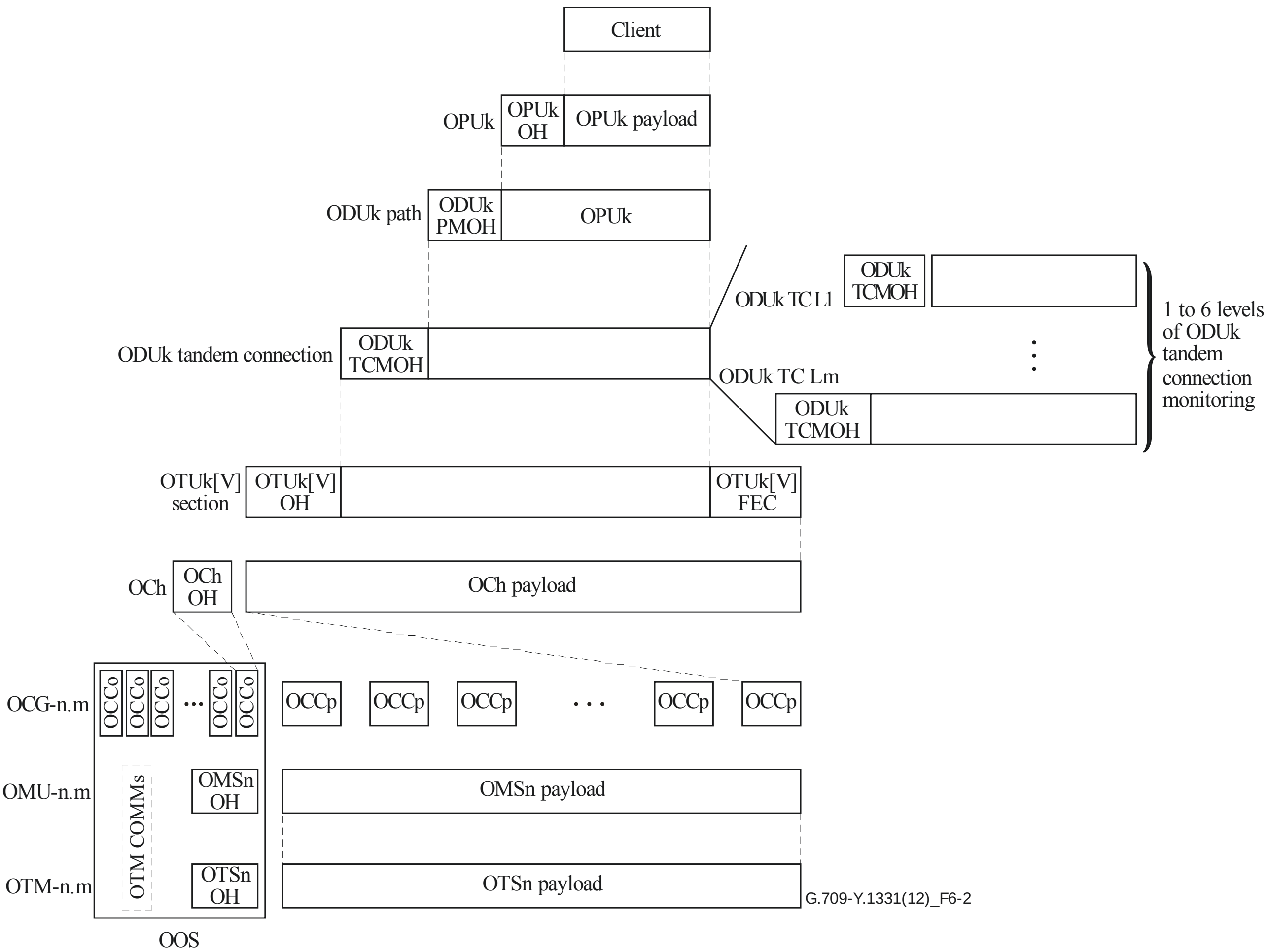


Figure 6-2 – OTM-n.m principal information containment relationships

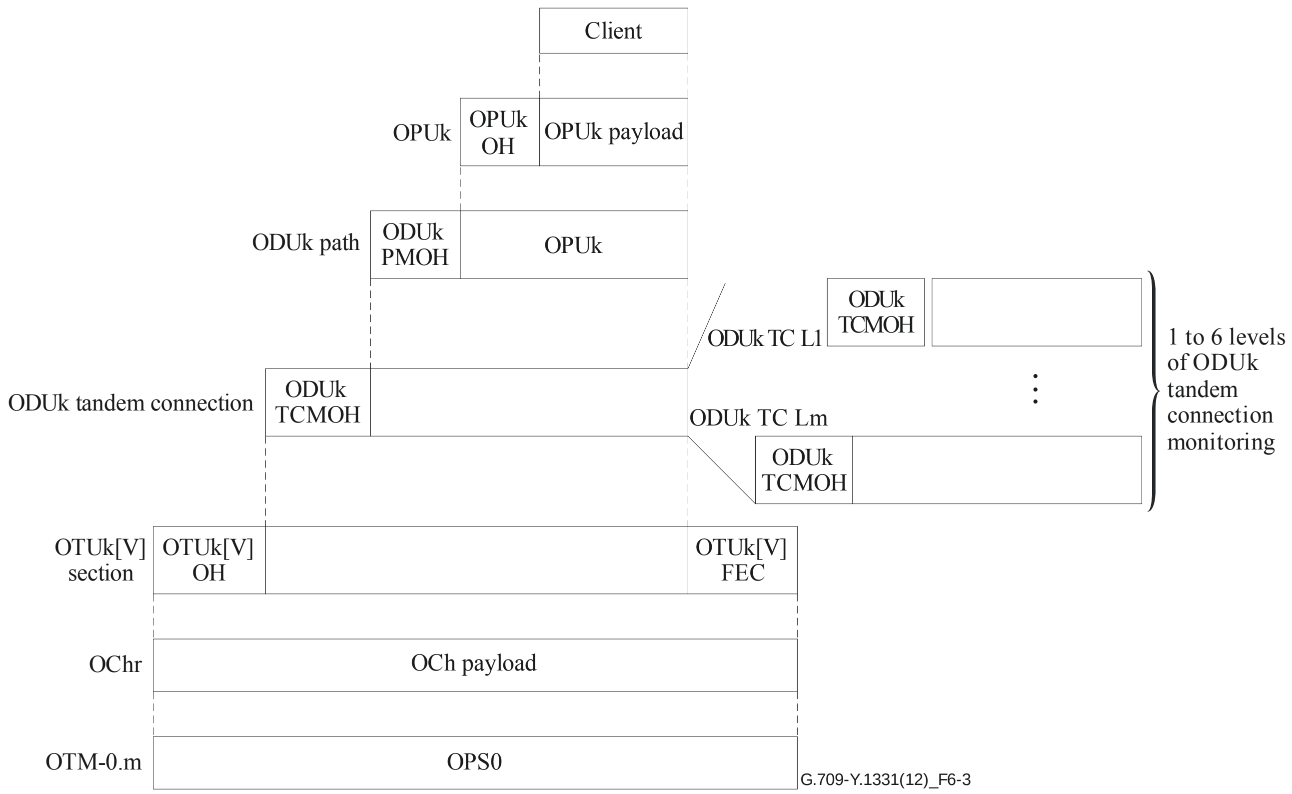


Figure 6-3 – OTM-0.m principal information containment relationships

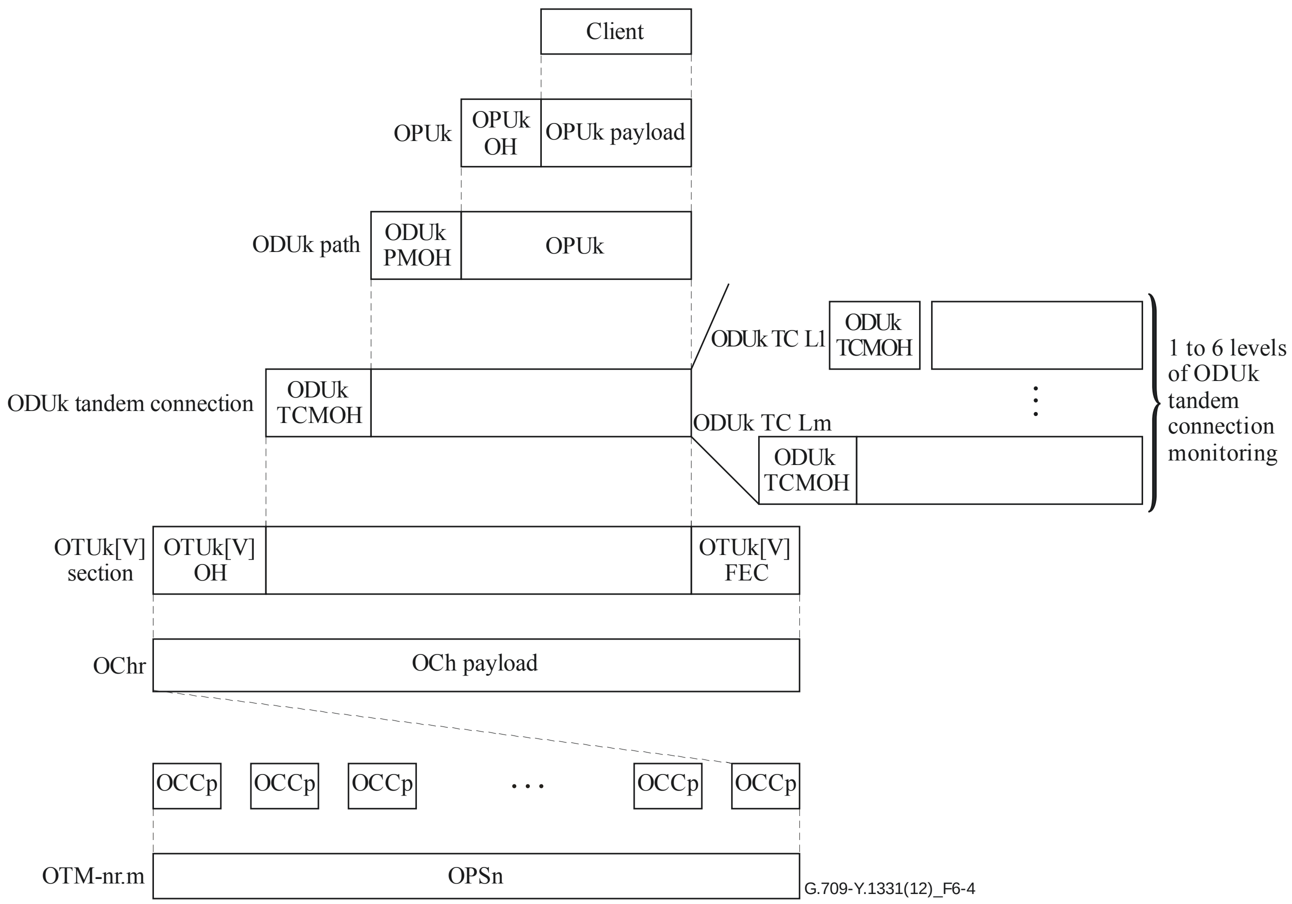


Figure 6-4 – OTM-nr.m principal information containment relationships

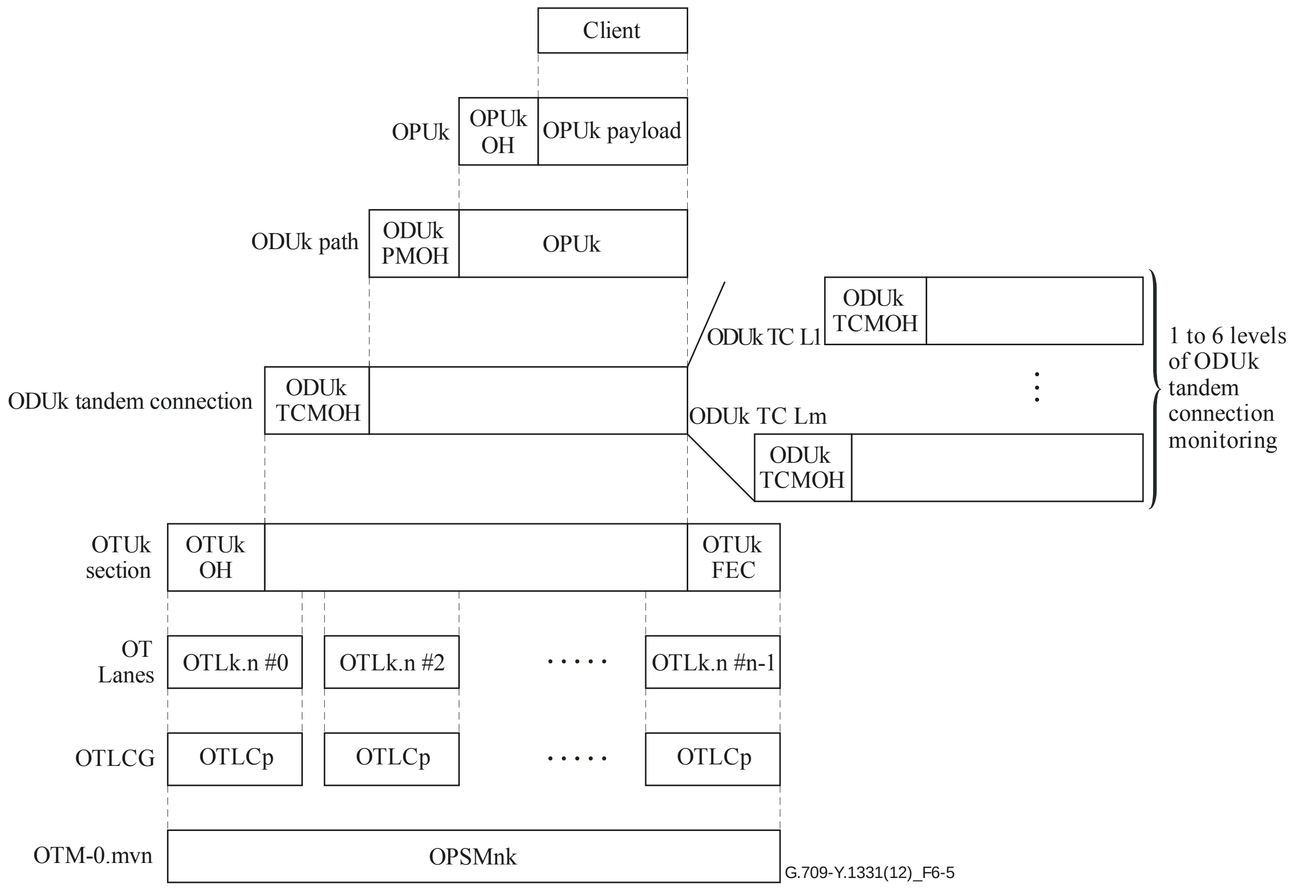


Figure 6-5 – OTM-0.mvn principal information containment relationships

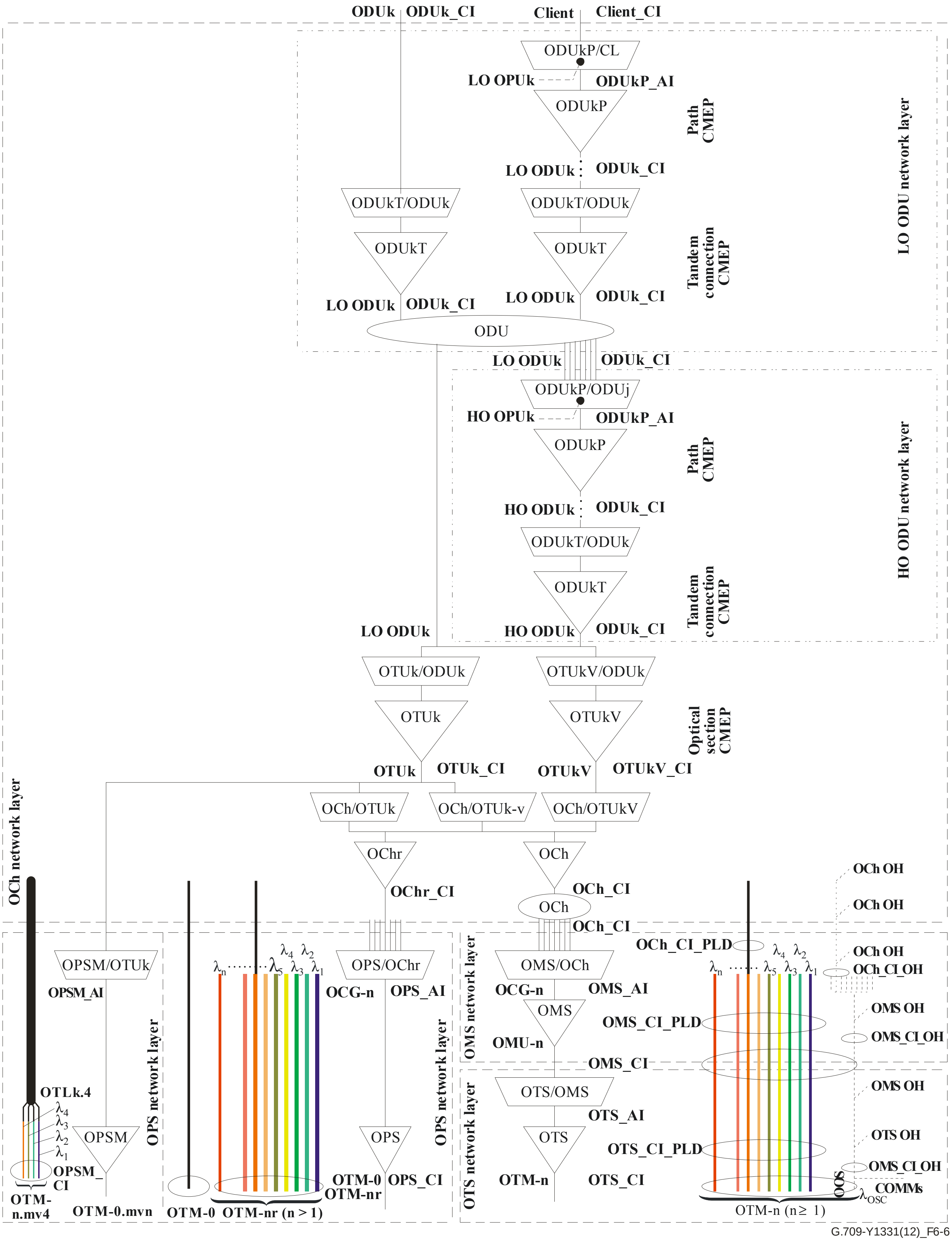


Figure 6-6 – Example of information flow relationship

7 Multiplexing/mapping principles and bit rates

Figures 7-1A and 7-1B show the relationship between various information structure elements and illustrate the multiplexing structure and mappings (including wavelength and time division multiplexing) for the OTM-n. In the multi-domain OTN any combination of the ODUk multiplexing layers may be present at a given OTN NNI. The interconnection of and visibility of ODUk multiplexing layers within an equipment or domain is outside the scope of this Recommendation. Refer to [ITU-T G.872] for further information on interconnection of and multiplexing of ODUk layers within a domain. Figure 7-1A shows that a (non-OTN) client signal is mapped into a lower order OPU, identified as "OPU (L)". The OPU (L) signal is mapped into the associated lower order ODU, identified as "ODU (L)". The ODU (L) signal is either mapped into the associated OTU[V] signal, or into an ODTU. The ODTU signal is multiplexed into an ODTU Group (ODTUG). The ODTUG signal is mapped into a higher order OPU, identified as "OPU (H)". The OPU (H) signal is mapped into the associated higher order ODU, identified as "ODU (H)". The ODU (H) signal is mapped into the associated OTU[V].

The OPU (L) and OPU (H) are the same information structures, but with different client signals. The concepts of lower order and high order ODU are specific to the role that the ODU plays within a single domain.

Figure 7-1B shows that an OTU[V] signal is mapped either into an optical channel signal, identified as OCh and OChr, or into an OTLk.n. The OCh/OChr signal is mapped into an optical channel carrier, identified as OCC and OCCr. The OCC/OCCr signal is multiplexed into an OCC group, identified as OCG-n.m and OCG-nr.m. The OCG-n.m signal is mapped into an OMSn. The OMSn signal is mapped into an OTSn. The OTSn signal is presented at the OTM-n.m interface. The OCG-nr.m signal is mapped into an OPSn. The OPSn signal is presented at the OTM-nr.m interface. A single OCCr signal is mapped into an OPS0. The OPS0 signal is presented at the OTM-0.m interface. The OTLk.n signal is mapped into an optical transport lane carrier, identified as OTLC. The OTLC signal is multiplexed into an OTLC group, identified as OTLCG. The OTLCG signal is mapped into an OPSMnk. The OPSMnk signal is presented at the OTM-0.mvn interface.

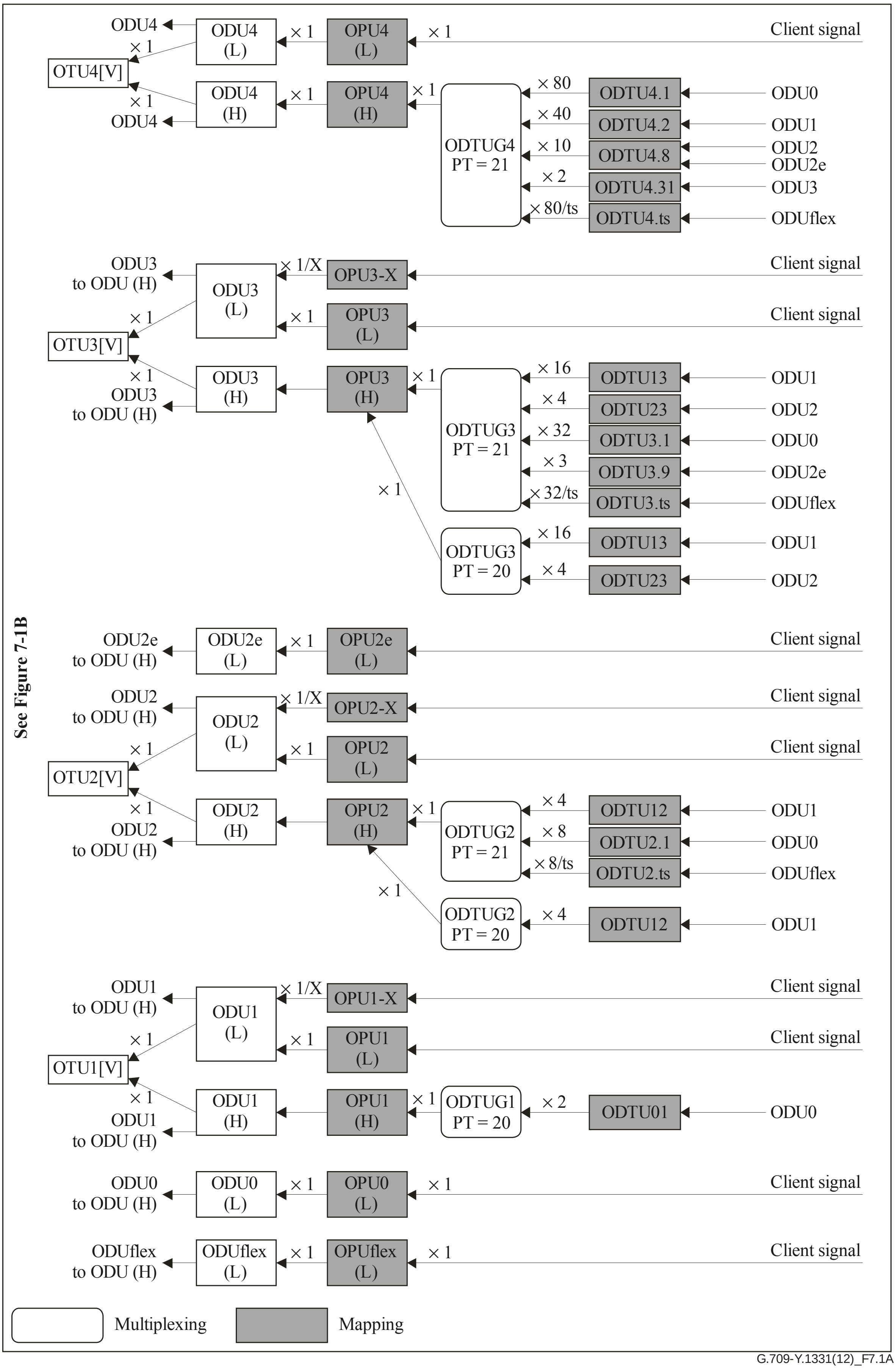


Figure 7-1A – OTM multiplexing and mapping structures (Part I)

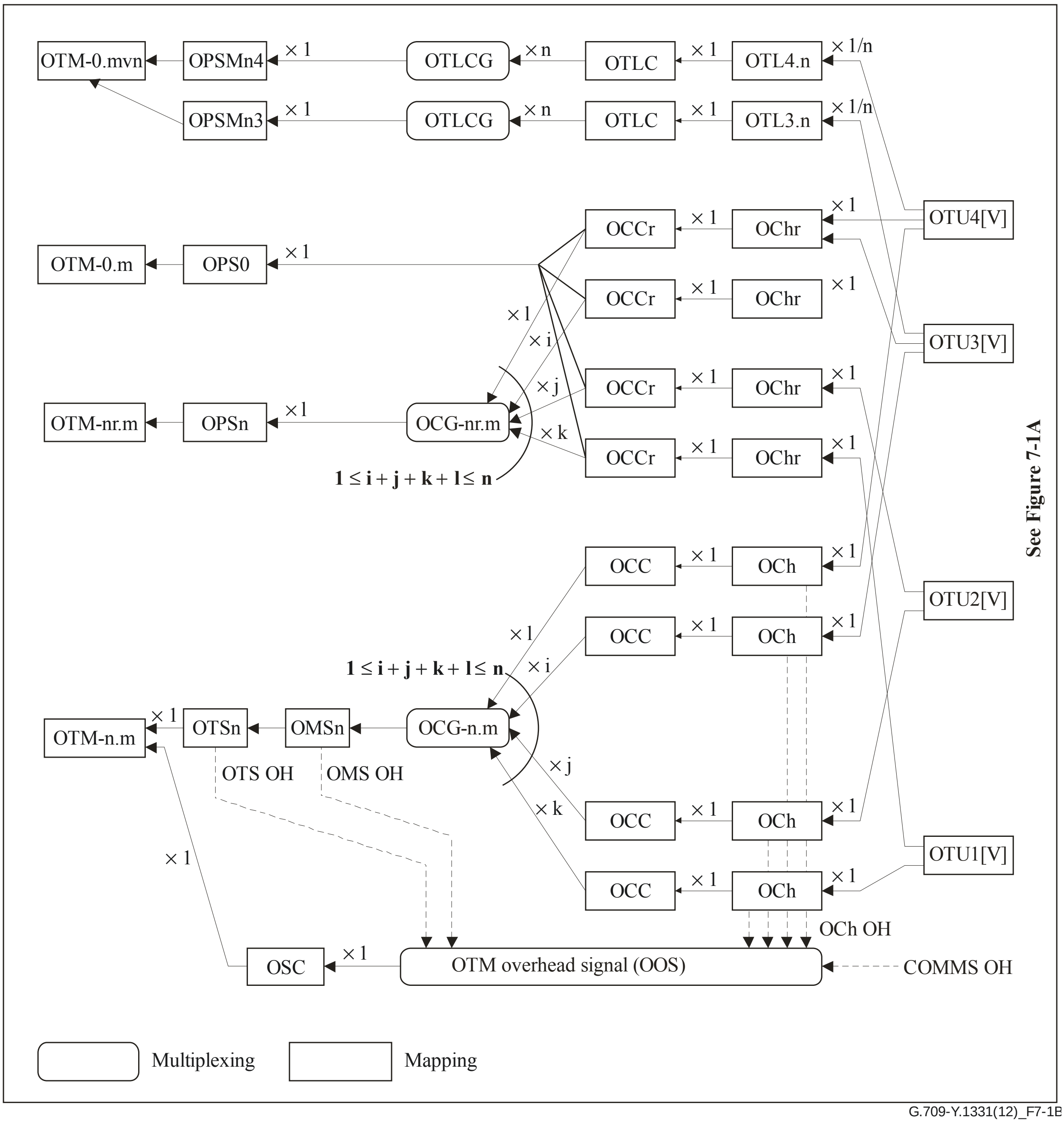


Figure 7-1B – OTM multiplexing and mapping structures (Part II)

The OTS, OMS, OCh and COMMS overhead is inserted into the OOS using mapping and multiplexing techniques which are outside the scope of this Recommendation.

7.1 Mapping

The client signal or an optical channel data tributary unit group (ODTUGk) is mapped into the OPUk. The OPUk is mapped into an ODUk and the ODUk is mapped into an OTUk[V]. The OTUk[V] is mapped into an OCh[r] and the OCh[r] is then modulated onto an OCC[r]. The OTUk may also be mapped into n OTLk.n and an OTLk.n is then modulated onto an OTLC.

7.2 Wavelength division multiplex

Up to n (n ≥ 1) OCC[r] are multiplexed into an OCG-n[r].m using wavelength division multiplexing. The OCC[r] tributary slots of the OCG-n[r].m can be of different size.

The OCG-n[r].m is transported via the OTM-n[r].m. For the case of the full functionality OTM-n.m interfaces the OSC is multiplexed into the OTM-n.m using wavelength division multiplexing.

n OTLC are aggregated into an OTLCG using wavelength division multiplexing. The OTLCG is transported via the OTM-0.mvn.

7.3 Bit rates and capacity

The bit rates and tolerance of the OTUk signals are defined in Table 7-1.

The bit rates and tolerance of the ODUk signals are defined in clause 12.2 and Table 7-2.

The bit rates and tolerance of the OPUk and OPUk-Xv payload are defined in Table 7-3.

The OTUk/ODUk/OPUk/OPUk-Xv frame periods are defined in Table 7-4.

The types and bit rates of the OTLk.n signals are defined in Table 7-5.

The 2.5G and 1.25G tributary slot related HO OPUk multiframe periods are defined in Table 7-6.

The ODTU payload area bandwidths are defined in Table 7-7. The bandwidth depends on the HO OPUk type (k=1,2,3,4) and the mapping procedure (AMP or GMP). The AMP bandwidths include the bandwidth provided by the NJO overhead byte. GMP is defined without such NJO bytes.

The bit rates and tolerance of the ODUflex(GFP) are defined in Table 7-8.

The number of HO OPUk tributary slots required by LO ODUj are summarized in Table 7-9.

Table 7-1 – OTU types and bit rates

OTU type	OTU nominal bit rate	OTU bit-rate tolerance
OTU1	255/238 × 2 488 320 kbit/s	±20 ppm
OTU2	255/237 × 9 953 280 kbit/s	
OTU3	255/236 × 39 813 120 kbit/s	
OTU4	255/227 × 99 532 800 kbit/s	
NOTE 1 – The nominal OTUk rates are approximately: 2 666 057.143 kbit/s (OTU1), 10 709 225.316 kbit/s (OTU2), 43 018 413.559 kbit/s (OTU3) and 111 809 973.568 kbit/s (OTU4). NOTE 2 – OTU0, OTU2e and OTUflex are not specified in this Recommendation. ODU0 signals are to be transported over ODU1, ODU2, ODU3 or ODU4 signals, ODU2e signals are to be transported over ODU3 and ODU4 signals and ODUflex signals are transported over ODU2, ODU3 and ODU4 signals.		

Table 7-2 – ODU types and bit rates

ODU type	ODU nominal bit rate	ODU bit-rate tolerance
ODU0	1 244 160 kbit/s	±20 ppm
ODU1	239/238 × 2 488 320 kbit/s	
ODU2	239/237 × 9 953 280 kbit/s	
ODU3	239/236 × 39 813 120 kbit/s	
ODU4	239/227 × 99 532 800 kbit/s	

Table 7-2 – ODU types and bit rates

ODU type	ODU nominal bit rate	ODU bit-rate tolerance
ODU2e	239/237 × 10 312 500 kbit/s	±100 ppm
ODUflex for CBR client signals	239/238 × client signal bit rate	±100 ppm (Notes 2, 3)
ODUflex for GFP-F mapped client signals	configured bit rate (see Table 7-8)	±100 ppm
NOTE 1 – The nominal ODUk rates are approximately: 2 498 775.126 kbit/s (ODU1), 10 037 273.924 kbit/s (ODU2), 40 319 218.983 kbit/s (ODU3), 104 794 445.815 kbit/s (ODU4) and 10 399 525.316 kbit/s (ODU2e). NOTE 2 – The bit-rate tolerance for ODUflex(CBR) signals is specified as ±100 ppm. This value may be larger than the tolerance for the client signal itself (e.g., ±20 ppm). For such case, the tolerance is determined by the ODUflex(CBR) maintenance signals, which have a tolerance of ±100 ppm. NOTE 3 – For ODUflex(CBR) signals with nominal bit rates close to the maximum ODTUk.ts payload bit rate and client rate tolerances less than ±100 ppm (e.g., ±10 ppm), the ODUflex(CBR) maintenance signal bit rates may exceed the ODTUk.ts payload bit rate. For such cases either an additional tributary slot may be used (i.e., ODTUk.(ts+1)), or the nominal bit rate of the ODUflex(CBR) signal may be artificially reduced to a value of 100 ppm below the maximum ODUflex(CBR) signal bit rate.		

Table 7-3 – OPU types and bit rates

OPU type	OPU payload nominal bit rate	OPU payload bit-rate tolerance
OPU0	238/239 × 1 244 160 kbit/s	±20 ppm
OPU1	2 488 320 kbit/s	
OPU2	238/237 × 9 953 280 kbit/s	
OPU3	238/236 × 39 813 120 kbit/s	
OPU4	238/227 × 99 532 800 kbit/s	
OPU2e	238/237 × 10 312 500 kbit/s	±100 ppm
OPUflex for CBR client signals	client signal bit rate	client signal bit-rate tolerance, with a maximum of ±100 ppm
OPUflex for GFP-F mapped client signals	238/239 × ODUflex signal rate	±100 ppm
OPU1-Xv	X × 2 488 320 kbit/s	±20 ppm
OPU2-Xv	X × 238/237 × 9 953 280 kbit/s	
OPU3-Xv	X × 238/236 × 39 813 120 kbit/s	
NOTE – The nominal OPUk payload rates are approximately: 1 238 954.310 kbit/s (OPU0 Payload), 2 488 320.000 kbit/s (OPU1 Payload), 9 995 276.962 kbit/s (OPU2 Payload), 40 150 519.322 kbit/s (OPU3 Payload), 104 355 975.330 (OPU4 Payload) and 10 356 012.658 kbit/s (OPU2e Payload). The nominal OPUk-Xv payload rates are approximately: X × 2 488 320.000 kbit/s (OPU1-Xv Payload), X × 9 995 276.962 kbit/s (OPU2-Xv Payload) and X × 40 150 519.322 kbit/s (OPU3-Xv Payload).		

Table 7-4 – OTUk/ODUk/OPUk frame periods

OTU/ODU/OPU type	Period (Note)
ODU0/OPU0	98.354 µs
OTU1/ODU1/OPU1/OPU1-Xv	48.971 µs
OTU2/ODU2/OPU2/OPU2-Xv	12.191 µs
OTU3/ODU3/OPU3/OPU3-Xv	3.035 µs
OTU4/ODU4/OPU4	1.168 µs
ODU2e/OPU2e	11.767 µs
ODUflex/OPUflex	CBR client signals: 121856/client_signal_bit_rate
	GFP-F mapped client signals: 122368/ODUflex_bit_rate
NOTE – The period is an approximated value, rounded to 3 decimal places.	

Table 7-5 – OTL types and bit rates

OTL type	OTL nominal bit rate	OTL bit-rate tolerance
OTL3.4	255/236 × 9 953 280 kbit/s	±20 ppm
OTL4.4	255/227 × 24 883 200 kbit/s	
NOTE – The nominal OTL rates are approximately: 10 754 603.390 kbit/s (OTL3.4) and 27 952 493.392 kbit/s (OTL4.4).		

Table 7-6 – HO OPUk multiframe periods for 2.5G and 1.25G tributary slots

OPU type	1.25G tributary slot multiframe period (Note)	2.5G tributary slot multiframe period (Note)
OPU1	97.942 µs	–
OPU2	97.531 µs	48.765 µs
OPU3	97.119 µs	48.560 µs
OPU4	93.416 µs	–
NOTE – The period is an approximated value, rounded to 3 decimal places.		

Table 7-7 – ODTU payload bandwidth (kbit/s)

ODTU type	ODTU payload nominal bandwidth	ODTU payload bit-rate tolerance
ODTU01	(1904 + 1/8)/3824 × ODU1 bit rate	±20 ppm
ODTU12	(952 + 1/16)/3824 × ODU2 bit rate	
ODTU13	(238 + 1/64)/3824 × ODU3 bit rate	
ODTU23	(952 + 4/64)/3824 × ODU3 bit rate	
ODTU2.ts	ts × 476/3824 × ODU2 bit rate	
ODTU3.ts	ts × 119/3824 × ODU3 bit rate	
ODTU4.ts	ts × 47.5/3824 × ODU4 bit rate	

Table 7-7 – ODTU payload bandwidth (kbit/s)

ODTU type	ODTU payload nominal bandwidth		ODTU payload bit-rate tolerance
	Minimum	**Nominal**	**Maximum**
ODTU01	1 244 216.796	1 244 241.681	1 244 266.566
ODTU12	2 498 933.311	2 498 983.291	2 499 033.271
ODTU13	2 509 522.012	2 509 572.203	2 509 622.395
ODTU23	10 038 088.048	10 038 288.814	10 038 489.579
ODTU2.ts	ts × 1 249 384.632	ts × 1 249 409.620	ts × 1 249 434.608
ODTU3.ts	ts × 1 254 678.635	ts × 1 254 703.729	ts × 1 254 728.823
ODTU4.ts	ts × 1 301 683.217	ts × 1 301 709.251	ts × 1 301 735.285
NOTE – The bandwidth is an approximated value, rounded to 3 decimal places.			

Table 7-8 – Recommended ODUflex (GFP) bit rates and tolerance

ODU type	Nominal bit-rate	Tolerance
ODU2.ts (Note)	1'249'177.230 kbit/s	
ODU3.ts (Note)	1'254'470.354 kbit/s	
ODU4.ts (Note)	1'301'467.133 kbit/s	
ODUflex(GFP) of n tributary slots, 1 ≤ n ≤ 8	n × ODU2.ts	±100 ppm
ODUflex(GFP) of n tributary slots, 9 ≤ n ≤ 32	n × ODU3.ts	±100 ppm
ODUflex(GFP) of n tributary slots, 33 ≤ n ≤ 80	n × ODU4.ts	±100 ppm
NOTE – The values of ODUk.ts are chosen to permit a variety of methods to be used to generate an ODUflex(GFP) clock. See Appendix XI for the derivation of these values and example ODUflex(GFP) clock generation methods.		

Table 7-9 – Number of tributary slots required for ODUj into HO OPUk

	# 2.5G tributary slots		# 1.25G tributary slots			
LO ODU	**OPU2**	**OPU3**	**OPU1**	**OPU2**	**OPU3**	**OPU4**
ODU0	–	–	1	1	1	1
ODU1	1	1	–	2	2	2
ODU2	–	4	–	–	8	8
ODU2e	–	–	–	–	9	8
ODU3	–	–	–	–	–	31
ODUflex(CBR)	–	–	–	Note 1	Note 2	Note 3
– ODUflex(IB SDR)	–	–	–	3	3	2
– ODUflex(IB DDR)	–	–	–	5	5	4
– ODUflex(IB QDR)	–	–	–	–	9	8
– ODUflex(FC-400)	–	–	–	4	4	4
– ODUflex(FC-800)	–	–	–	7	7	7

Table 7-9 – Number of tributary slots required for ODUj into HO OPUk

	# 2.5G tributary slots		# 1.25G tributary slots			
LO ODU	**OPU2**	**OPU3**	**OPU1**	**OPU2**	**OPU3**	**OPU4**
– ODUflex(FC-1600)	–	–	–	–	12	11
– ODUflex(3G SDI) (2 970 000)	–	–	–	3	3	3
– ODUflex(3G SDI) (2 970 000/1.001)	–	–	–	3	3	3
ODUflex(GFP)	–	–	–	n	n	n

NOTE 1 – Number of tributary slots = Ceiling(ODUflex(CBR) nominal bit rate/(T×ODTU2.ts nominal bit rate) × (1+ODUflex(CBR) bit-rate tolerance)/(1−HO OPU2 bit-rate tolerance)).

NOTE 2 – Number of tributary slots = Ceiling(ODUflex(CBR) nominal bit rate/(T×ODTU3.ts nominal bit rate) × (1+ODUflex(CBR) bit-rate tolerance)/(1−HO OPU3 bit-rate tolerance)).

NOTE 3 – Number of tributary slots = Ceiling(ODUflex(CBR) nominal bit rate/(T×ODTU4.ts nominal bit rate) × (1+ODUflex(CBR) bit-rate tolerance)/(1−HO OPU4 bit-rate tolerance)).

NOTE 4 – T represents the transcoding factor. Refer to clauses 17.7.3, 17.7.4 and 17.7.5.

7.4 ODUk time-division multiplex

Figure 7-1A shows the relationship between various time-division multiplexing elements that are defined below and illustrates possible multiplexing structures. Table 7-10 provides an overview of valid tributary slot types and mapping procedure configuration options.

Up to 2 ODU0 signals are multiplexed into an ODTUG1 (PT=20) using time-division multiplexing. The ODTUG1 (PT=20) is mapped into the OPU1.

Up to 4 ODU1 signals are multiplexed into an ODTUG2 (PT=20) using time-division multiplexing. The ODTUG2 (PT=20) is mapped into the OPU2.

A mixture of p (p ≤ 4) ODU2 and q (q ≤ 16) ODU1 signals can be multiplexed into an ODTUG3 (PT=20) using time-division multiplexing. The ODTUG3 (PT=20) is mapped into the OPU3.

A mixture of p (p ≤ 8) ODU0, q (q ≤ 4) ODU1, r (r ≤ 8) ODUflex signals can be multiplexed into an ODTUG2 (PT=21) using time-division multiplexing. The ODTUG2 (PT=21) is mapped into the OPU2.

A mixture of p (p ≤ 32) ODU0, q (q ≤ 16) ODU1, r (r ≤ 4) ODU2, s (s ≤ 3) ODU2e and t (t ≤ 32) ODUflex signals can be multiplexed into an ODTUG3 (PT=21) using time-division multiplexing. The ODTUG3 (PT=21) is mapped into the OPU3.

A mixture of p (p ≤ 80) ODU0, q (q ≤ 40) ODU1, r (r ≤ 10) ODU2, s (s ≤ 10) ODU2e, t (t ≤ 2) ODU3 and u (u ≤ 80) ODUflex signals can be multiplexed into an ODTUG4 (PT=21) using time-division multiplexing. The ODTUG4 (PT=21) is mapped into the OPU4.

NOTE – The ODTUGk is a logical construct and is not defined further. ODTUjk and ODTUk.ts signals are directly time-division multiplexed into the tributary slots of an HO OPUk.

Table 7-10 – Overview of ODUj into OPUk mapping types

	2.5G tributary slots		1.25G tributary slots			
	OPU2	OPU3	OPU1	OPU2	OPU3	OPU4
ODU0	–	–	AMP (PT=20)	GMP (PT=21)	GMP (PT=21)	GMP (PT=21)
ODU1	AMP (PT=20)	AMP (PT=20)	–	AMP (PT=21)	AMP (PT=21)	GMP (PT=21)
ODU2	–	AMP (PT=20)	–	–	AMP (PT=21)	GMP (PT=21)
ODU2e	–	–	–	–	GMP (PT=21)	GMP (PT=21)
ODU3	–	–	–	–	–	GMP (PT=21)
ODUflex	–	–	–	GMP (PT=21)	GMP (PT=21)	GMP (PT=21)

Figures 7-2, 7-3 and 7-4 show how various signals are multiplexed using the ODTUG1/2/3 (PT=20) multiplexing elements. Figure 7-2 presents the multiplexing of four ODU1 signals into the OPU2 signal via the ODTUG2 (PT=20). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 1 into 2 (ODTU12) using the AMP justification overhead (JOH). The four ODTU12 signals are time-division multiplexed into the optical channel data tributary unit group 2 (ODTUG2) with payload type 20, after which this signal is mapped into the OPU2.

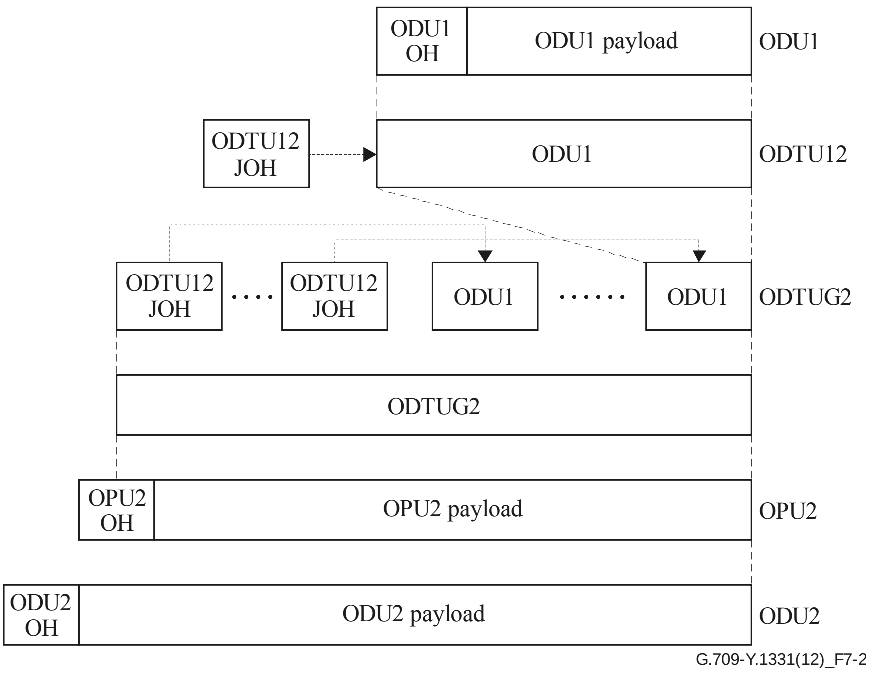


Figure 7-2 – ODU1 into ODU2 multiplexing method via ODTUG2 (PT=20)

Figure 7-3 presents the multiplexing of up to 16 ODU1 signals and/or up to 4 ODU2 signals into the OPU3 signal via the ODTUG3 (PT=20). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 1 into 3 (ODTU13) using the AMP justification overhead (JOH). An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 2 into 3 (ODTU23) using the AMP justification overhead (JOH). "x" ODTU23 (0 ≤ x ≤ 4) signals and "16-4x" ODTU13 signals are time-division multiplexed into the optical channel data tributary unit group 3 (ODTUG3) with payload type 20, after which this signal is mapped into the OPU3.

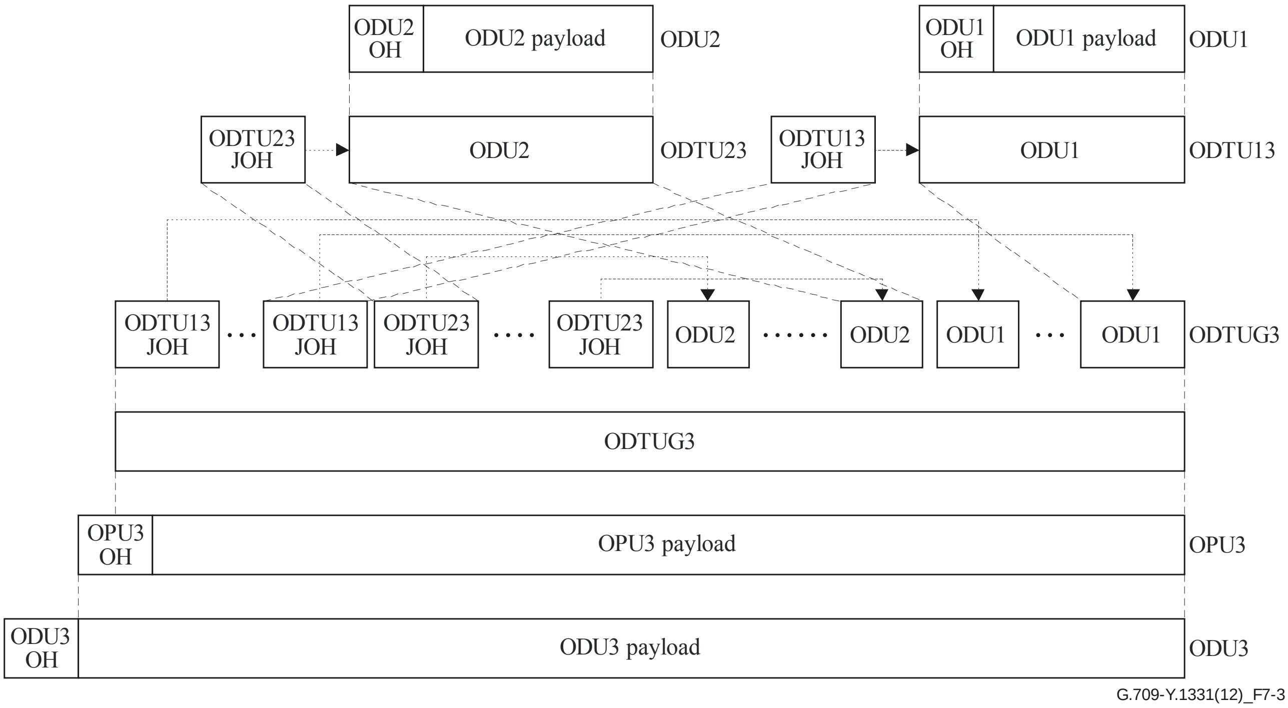


Figure 7-3 – ODU1 and ODU2 into ODU3 multiplexing method via ODTUG3 (PT=20)

Figure 7-4 presents the multiplexing of two ODU0 signals into the OPU1 signal via the ODTUG1 (PT=20). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 0 into 1 (ODTU01) using the AMP justification overhead (JOH). The two ODTU01 signals are time-division multiplexed into the optical channel data tributary unit group 1 (ODTUG1) with payload type 20, after which this signal is mapped into the OPU1.

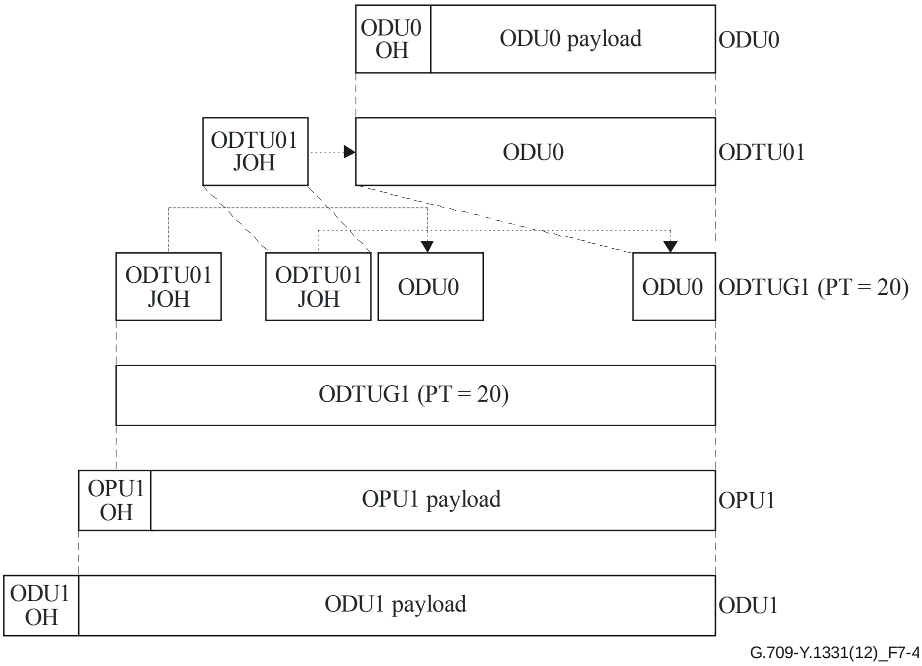


Figure 7-4 – ODU0 into ODU1 multiplexing method via ODTUG1 (PT=20)

Figures 7-5, 7-6 and 7-7 show how various signals are multiplexed using the ODTUG2/3/4 (PT=21) multiplexing elements.

Figure 7-5 presents the multiplexing of up to eight ODU0 signals, and/or up to four ODU1 signals and/or up to eight ODUflex signals into the OPU2 signal via the ODTUG2 (PT=21). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 1 into 2 (ODTU12) using the AMP justification overhead (JOH). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 2.1 (ODTU2.1) using the GMP justification overhead. An ODUflex signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 2.ts (ODTU2.ts) using the GMP justification overhead. Up to eight ODTU2.1 signals, up to four ODTU12 signals and up to eight ODTU2.ts signals are time-division multiplexed into the optical channel data tributary unit group 2 (ODTUG2) with payload type 21, after which this signal is mapped into the OPU2.

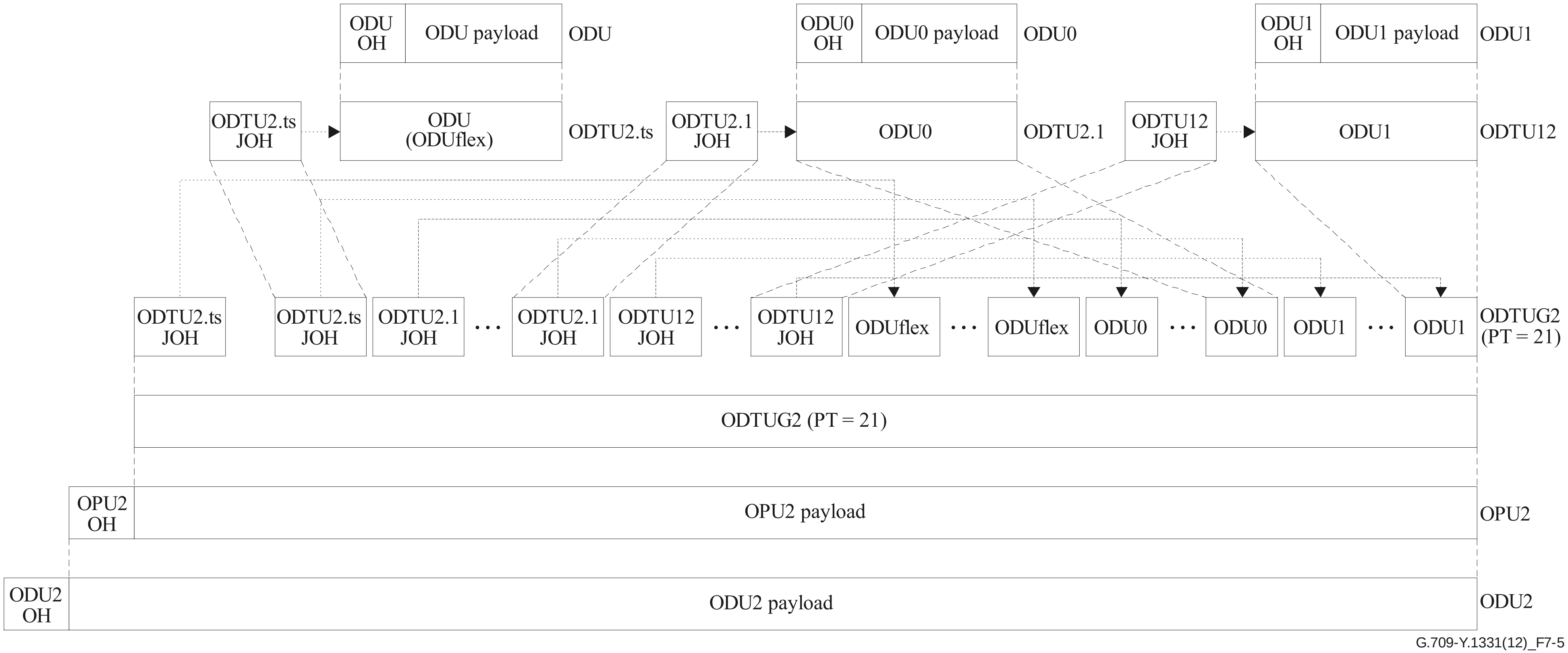


Figure 7-5 – ODU0, ODU1 and ODUflex into ODU2 multiplexing method via ODTUG2 (PT=21)

Figure 7-6 presents the multiplexing of up to thirty-two ODU0 signals and/or up to sixteen ODU1 signals and/or up to four ODU2 signals and/or up to three ODU2e signals and/or up to thirty-two ODUflex signals into the OPU3 signal via the ODTUG3 (PT=21). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 1 into 3 (ODTU13) using the AMP justification overhead (JOH). An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 2 into 3 (ODTU23) using the AMP justification overhead. An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 3.1 (ODTU3.1) using the GMP justification overhead. An ODU2e signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 3.9 (ODTU3.9) using the GMP justification overhead. An ODUflex signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 3.ts (ODTU3.ts) using the GMP justification overhead. Up to thirty-two ODTU3.1 signals, up to sixteen ODTU13 signals, up to four ODTU23 signals, up to three ODTU3.9 and up to thirty-two ODTU3.ts signals are time-division multiplexed into the optical channel data tributary unit group 3 (ODTUG3) with payload type 21, after which this signal is mapped into the OPU3.

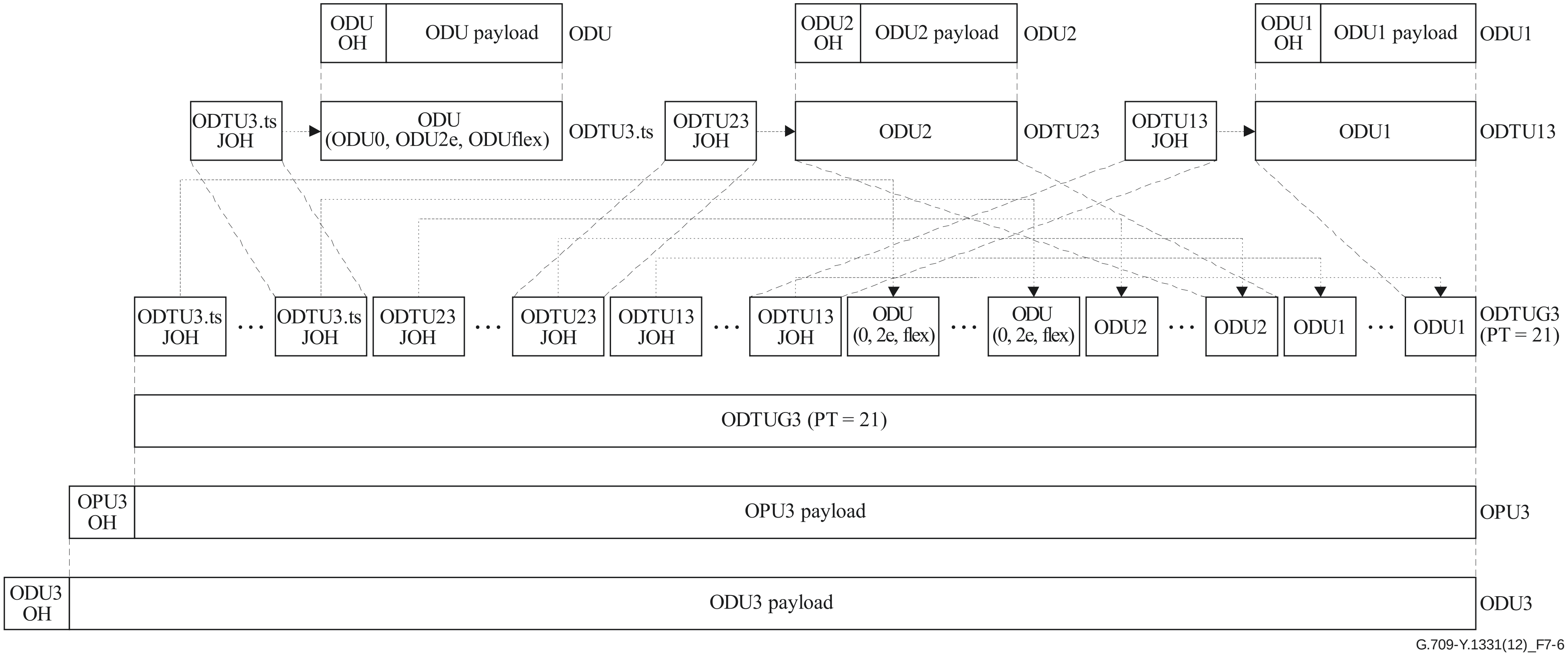


Figure 7-6 – ODU0, ODU1, ODU2, ODU2e and ODUflex into ODU3 multiplexing method via ODTUG3 (PT=21)

Figure 7-7 presents the multiplexing of up to eighty ODU0 signals and/or up to forty ODU1 signals and/or up to ten ODU2 signals and/or up to ten ODU2e signals and/or up to two ODU3 signals and/or up to eighty ODUflex signals into the OPU4 signal via the ODTUG4 (PT=21). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.1 (ODTU4.1) using the GMP justification overhead (JOH). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.2 (ODTU4.2) using the GMP justification overhead. An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.8 (ODTU4.8) using the GMP justification overhead (JOH). An ODU2e signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.8 (ODTU4.8) using the GMP justification overhead. An ODU3 signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.31 (ODTU4.31) using the GMP justification overhead. An ODUflex signal is extended with a frame alignment overhead and asynchronously mapped into the optical channel data tributary unit 4.ts (ODTU4.ts) using the GMP justification overhead (JOH). Up to eighty ODTU4.1 signals, up to forty ODTU4.2 signals, up to ten ODTU4.8 signals, up to two ODTU4.31 and up to eighty ODTU4.ts signals are time-division multiplexed into the optical channel data tributary unit group 4 (ODTUG4) with payload type 21, after which this signal is mapped into the OPU4.

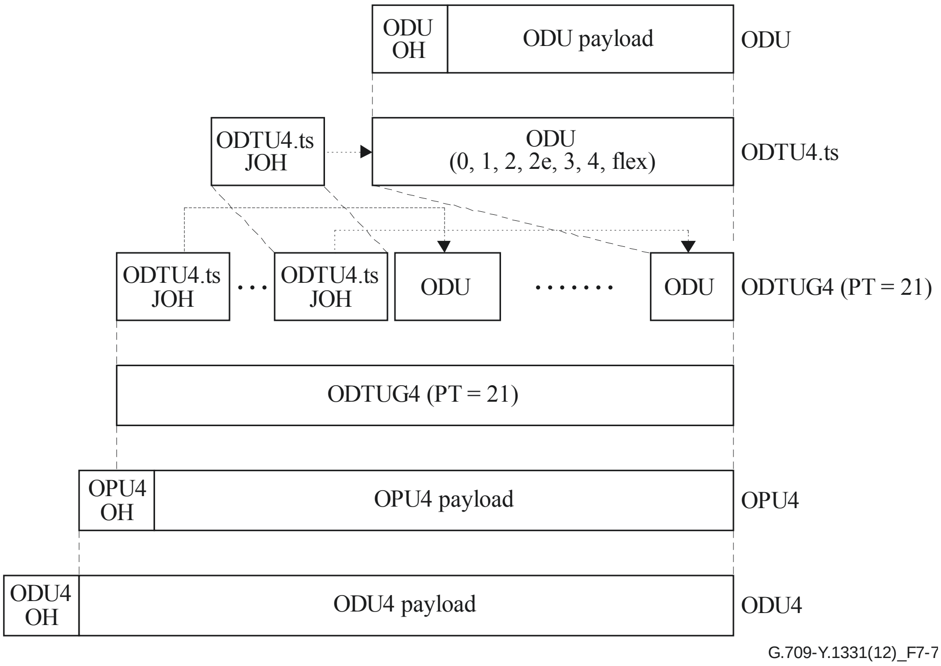


Figure 7-7 – ODU0, ODU1, ODU2, ODU2e, ODU3 and ODUflex into ODU4 multiplexing method via ODTUG4 (PT=21)

Details of the multiplexing method and mappings are given in clause 19.

Some examples illustrating the multiplexing of 2 ODU0 signals into an ODU1 and of 4 ODU1 signals into an ODU2 are presented in Appendix III.

8 Optical transport module (OTM-n.m, OTM-nr.m, OTM-0.m, OTM-0.mvn)

Two OTM structures are defined, one with full functionality and one with reduced functionality. For the IrDI only reduced functionality OTM interfaces are currently defined. Other full or reduced functionality OTM IrDIs are for further study.

Table 8-1 provides an overview of the OTU, OTU FEC, OCh/OChr, OPS, OPSM and OMS/OTS elements in the OTM structures specified in this clause.

Table 8-1 – Overview of OTM structures

	OTUk frame	OTUkV frame	OTUk FEC	OTUkV FEC	OChr	OCh	OPS	OPSM	OMS OTS	IaDI	IrDI
OTM-n.m	X		X			X			X	X	
OTM-n.m	X			X		X			X	X	
OTM-n.m		X		X		X			X	X	
OTM-16/32r.m	X		X		X		X			X	X
OTM-16/32r.m	X			X	X		X			X	
OTM-16/32r.m		X		X	X		X			X	
OTM-0.m	X		X		X		X			X	X
OTM-0.m	X			X	X		X			X	
OTM-0.mvn	X		X					X		X	X

8.1 OTM with reduced functionality (OTM-0.m, OTM-nr.m, OTM-0.mvn)

The OTM-n supports n optical channels on a single optical span with 3R regeneration and termination of the OTUk[V] on each end. As 3R regeneration is performed on both sides of the OTM-0.m, OTM-nr.m and OTM-0.mvn interfaces access to the OTUk[V] overhead is available and maintenance/supervision of the interface is provided via this overhead. Therefore a non-associated OTN overhead is not required across the OTM-0.m, OTM-nr.m and OTM-0.mvn interfaces and an OSC/OOS is not supported.

Three OTM interfaces classes with reduced functionality are defined, OTM-0.m, OTM-nr.m and OTM-0.mvn. Other reduced functionality interfaces classes are for further study.

8.1.1 OTM-0.m

The OTM-0.m supports a non-coloured optical channel on a single optical span with 3R regeneration at each end.

Four OTM-0.m interface signals (see Figure 8-1) are defined, each carrying a single channel optical signal containing one OTUk[V] signal:

– OTM-0.1 (carrying an OTU1[V])

– OTM-0.2 (carrying an OTU2[V])

– OTM-0.3 (carrying an OTU3[V])

– OTM-0.4 (carrying an OTU4[V]).

In generic terms: OTM-0.m.

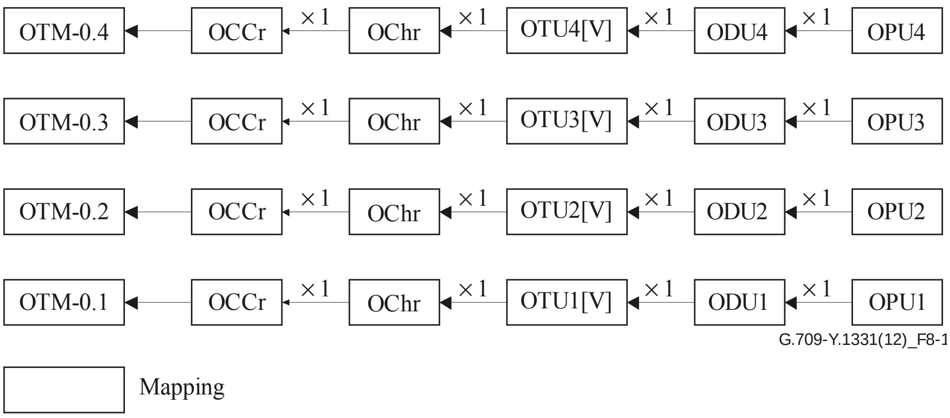


Figure 8-1 – OTM-0.m structure

Figure 8-1 shows the relationship between various information structure elements that are defined below and illustrates possible mappings for the OTM-0.m.

An OSC is not present and there is no OOS either.

8.1.2 OTM-nr.m

8.1.2.1 OTM-16r.m

This OTM-16r.m supports 16 optical channels on a single optical span with 3R regeneration at each end.

Several OTM-16r interface signals are defined. Some examples:

– OTM-16r.1 (carrying i (i ≤ 16) OTU1[V] signals);
– OTM-16r.2 (carrying j (j ≤ 16) OTU2[V] signals);
– OTM-16r.3 (carrying k (k ≤ 16) OTU3[V] signals);
– OTM-16r.4 (carrying l (l ≤ 16) OTU4[V] signals);
– OTM-16r.1234 (carrying i (i ≤ 16) OTU1[V], j (j ≤ 16) OTU2[V], k (k ≤ 16) OTU3[V] and l (l ≤ 16) OTU4[V] signals with i + j + k +l ≤ 16);
– OTM-16r.123 (carrying i (i ≤ 16) OTU1[V], j (j ≤ 16) OTU2[V] and k (k ≤ 16) OTU3[V] signals with i + j + k ≤ 16);
– OTM-16r.12 (carrying i (i ≤ 16) OTU1[V] and j (j ≤ 16) OTU2[V] signals with i + j ≤ 16);
– OTM-16r.23 (carrying j (j ≤ 16) OTU2[V] and k (k ≤ 16) OTU3[V] signals with j + k ≤ 16);
– OTM-16r.34 (carrying k (k ≤ 16) OTU3[V] and l (l ≤ 16) OTU4[V] signals with k + l ≤ 16);

which in generic terms are identified as OTM-16r.m.

The OTM-16r.m signal is an OTM-nr.m signal (see Figure 6-6) with 16 optical channel carriers (OCCr) numbered OCCr #0 to OCCr #15. An optical supervisory channel (OSC) is not present and there is no OOS either.

At least one of the OCCrs is in service during normal operation and transporting an OTUk[V].

There is no predefined order in which the OCCrs are taken into service.

Some examples of the defined OTM-16r.m interface signals and the OTM-16r.m multiplexing structure are shown in Figure 8-2.

NOTE – OTM-16r.m OPS overhead is not defined. The interface will use the OTUk[V] SMOH in this multi-wavelength interface for supervision and management. OTM-16r.m connectivity (TIM) failure reports will be computed from the individual OTUk[V] reports by means of failure correlation in fault management. Refer to the equipment Recommendations for further details.

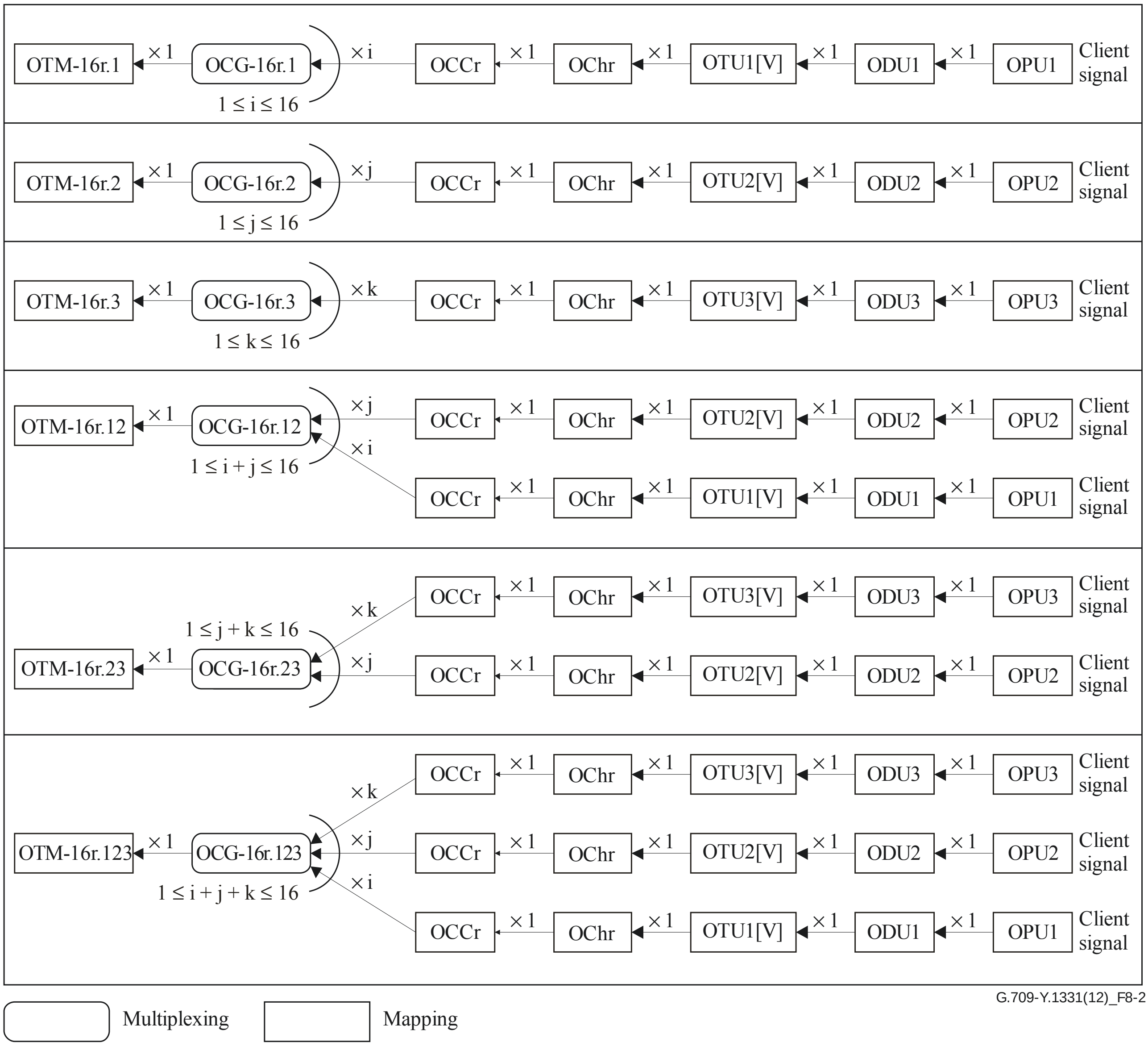


Figure 8-2 – OTM-16r.m multiplexing structure examples

8.1.2.2 OTM-32r.m

This OTM-32r.m supports 32 optical channels on a single optical span with 3R regeneration at each end.

Several OTM-32r interface signals are defined. Some examples:

- OTM-32r.1 (carrying i (i ≤ 32) OTU1[V] signals);
- OTM-32r.2 (carrying j (j ≤ 32) OTU2[V] signals);
- OTM-32r.3 (carrying k (k ≤ 32) OTU3[V] signals);
- OTM-32r.4 (carrying l (l ≤ 32) OTU4[V] signals);
- OTM-32r.1234 (carrying i (i ≤ 32) OTU1[V], j (j ≤ 32) OTU2[V], k (k ≤ 32) OTU3[V] and l (l ≤ 32) OTU4[V] signals with i + j + k + l ≤ 32);

– OTM-32r.123 (carrying i ($i \leq 32$) OTU1[V], j ($j \leq 32$) OTU2[V] and k ($k \leq 32$) OTU3[V] signals with $i + j + k \leq 32$);
– OTM-32r.12 (carrying i ($i \leq 32$) OTU1[V] and j ($j \leq 32$) OTU2[V] signals with $i + j \leq 32$);
– OTM-32r.23 (carrying j ($j \leq 32$) OTU2[V] and k ($k \leq 32$) OTU3[V] signals with $j + k \leq 32$);
– OTM-32r.34 (carrying k ($k \leq 32$) OTU3[V] and l ($l \leq 32$) OTU4[V] signals with $k + l \leq 32$);

which in generic terms are identified as OTM-32r.m.

The OTM-32r.m signal is an OTM-nr.m signal (see Figure 6-6) with 32 optical channel carriers (OCCr) numbered OCCr #0 to OCCr #31. An optical supervisory channel (OSC) is not present and there is no OOS either.

At least one of the OCCrs is in service during normal operation and transporting an OTUk[V].

There is no predefined order in which the OCCrs are taken into service.

NOTE – OTM-32r.m OPS overhead is not defined. The interface will use the OTUk[V] SMOH in this multi-wavelength interface for supervision and management. OTM-32r.m connectivity (TIM) failure reports will be computed from the individual OTUk[V] reports by means of failure correlation in fault management. Refer to the equipment Recommendations for further details.

8.1.3 OTM-0.mvn

The OTM-0.mvn supports a multi-lane optical signal on a single optical span with 3R regeneration at each end.

Two OTM-0.mvn interface signals are defined, each carrying a four-lane optical signal containing one OTUk signal striped across the four optical lanes:

- OTM-0.3v4 (carrying an OTU3)
- OTM-0.4v4 (carrying an OTU4).

In generic terms: OTM-0.mvn.

The optical lanes are numbered of each OTLCx, x=0 to n–1 where x represents the optical lane number of the corresponding [ITU-T G.959.1] or [ITU-T G.695] application code for the multilane applications.

Figure 8-3 shows the relationship between various information structure elements for the OTM-0.3v4 and OTM-0.4v4.

An OSC is not present and there is no OOS either.

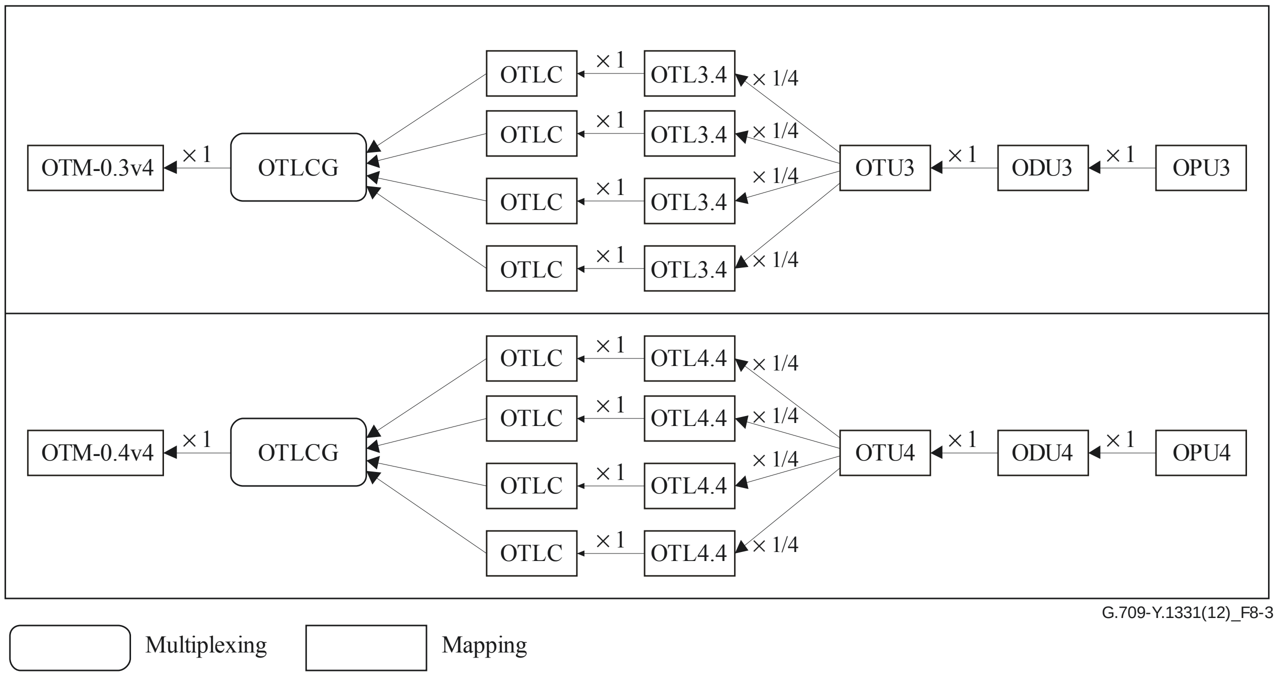


Figure 8-3 – OTM-0.3v4 and OTM-0.4v4 structure

8.2 OTM with full functionality (OTM-n.m)

The OTM-n.m interface supports up to n optical channels for single or multiple optical spans. 3R regeneration is not required at the interface.

Several OTM-n interface signals are defined. Some examples:

– OTM-n.1 (carrying i (i ≤ n) OTU1[V] signals);

– OTM-n.2 (carrying j (j ≤ n) OTU2[V] signals);

– OTM-n.3 (carrying k (k ≤ n) OTU3[V] signals);

– OTM-n.4 (carrying l (l ≤ n) OTU4[V] signals);

– OTM-n.1234 (carrying i (i ≤ n) OTU1[V], j (j ≤ n) OTU2[V], k (k ≤ n) OTU3[V] and l (l ≤ n) OTU4[V] signals with i + j + k +l ≤ n);

– OTM-n.123 (carrying i (i ≤ n) OTU1[V], j (j ≤ n) OTU2[V] and k (k ≤ n) OTU3[V] signals with i + j + k ≤ n);

– OTM-n.12 (carrying i (i ≤ n) OTU1[V] and j (j ≤ n) OTU2[V] signals with i + j ≤ n);

– OTM-n.23 (carrying j (j ≤ n) OTU2[V] and k (k ≤ n) OTU3[V] signals with j + k ≤ n);

– OTM-n.34 (carrying k (k ≤ n) OTU3[V] and l (l ≤ n) OTU4[V] signals with k + l ≤ n);

which in generic terms are identified as OTM-n.m.

An OTM-n.m interface signal contains up to "n" OCCs associated with the lowest bit rate that is supported as indicated by m and an OSC (see Figure 8-4). It is possible that a reduced number of higher bit rate capable OCCs are supported. The value of "n", "m" and the OSC are not defined in this Recommendation.

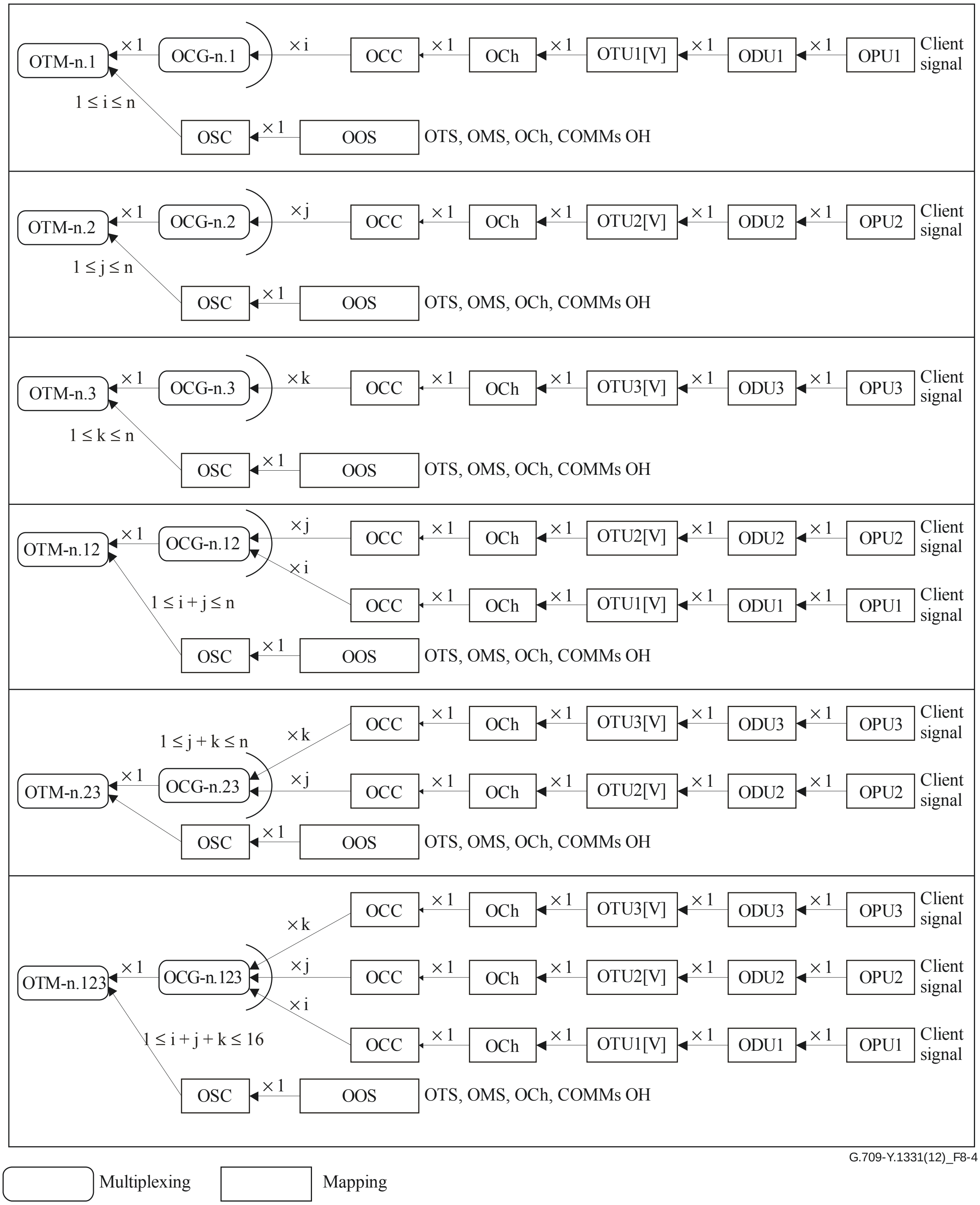


Figure 8-4 – OTM-n.m multiplexing structure examples

9 Physical specification of the ONNI

9.1 OTM-0.m

Specifications for physical optical characteristics of the OTM-0.1, OTM-0.2 and OTM-0.3 signals are contained in [ITU-T G.959.1] and [ITU-T G.693].

Specifications for physical optical characteristics of the OTM-0.4 are for further study.

9.2 OTM-nr.m

9.2.1 OTM-16r.m

Specifications for physical optical characteristics of the OTM-16r.1, OTM-16r.2 and OTM-16r.12 signals are contained in [ITU-T G.959.1].

Specifications for physical optical characteristics of other OTM-16r.m are for further study.

9.2.2 OTM-32r.m

Specifications for physical optical characteristics of the OTM-32r.1, OTM-32r.2, and OTM-32r.12 signals are contained in [ITU-T G.959.1].

Specifications for physical optical characteristics of other OTM-32r.m are for further study.

9.3 OTM-n.m

Specifications for physical optical characteristics of the OTM-n.m are vendor specific and outside the scope of this Recommendation.

9.4 OTM-0.mvn

Specifications for physical optical characteristics of the OTM-0.3v4 and OTM-0.4v4 signals are contained in [ITU-T G.695] and [ITU-T G.959.1], respectively.

10 Optical channel (OCh)

The OCh transports a digital client signal between 3R regeneration points. The OCh client signals defined in this Recommendation are the OTUk signals.

10.1 OCh with full functionality (OCh)

The optical channel with a full functionality (OCh) structure is conceptually shown in Figure 10-1. It contains two parts: OCh overhead and OCh payload.

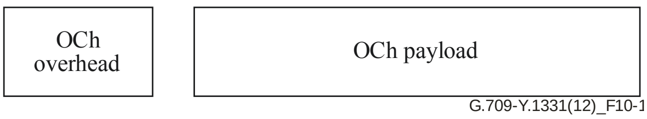


Figure 10-1 – OCh information structure

10.2 OCh with reduced functionality (OChr)

The optical channel with a reduced functionality (OChr) structure is conceptually shown in Figure 10-2. It contains: OCh payload.

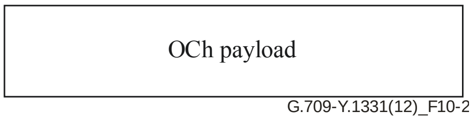


Figure 10-2 – OChr information structure

11 Optical channel transport unit (OTU)

The OTUk[V] conditions the ODUk for transport over an optical channel network connection. The OTUk frame structure, including the OTUk FEC is completely standardized. The OTUkV is a frame structure, including the OTUkV FEC that is only functionally standardized (i.e., only the required functionality is specified); refer to Appendix II. Besides these two, there is an OTUkV in which the completely standardized OTUk frame structure is combined with a functionally standardized OTUkV FEC; refer to Appendix II. This combination is identified as OTUk-v.

11.1 OTUk frame structure

The OTUk (k = 1,2,3,4) frame structure is based on the ODUk frame structure and extends it with a forward error correction (FEC) as shown in Figure 11-1. 256 columns are added to the ODUk frame for the FEC and the reserved overhead bytes in row 1, columns 8 to 14 of the ODUk overhead are used for an OTUk specific overhead, resulting in an octet-based block frame structure with four rows and 4080 columns. The MSB in each octet is bit 1, the LSB is bit 8.

NOTE – This Recommendation does not specify an OTUk frame structure for k=0, k=2e or k=flex.

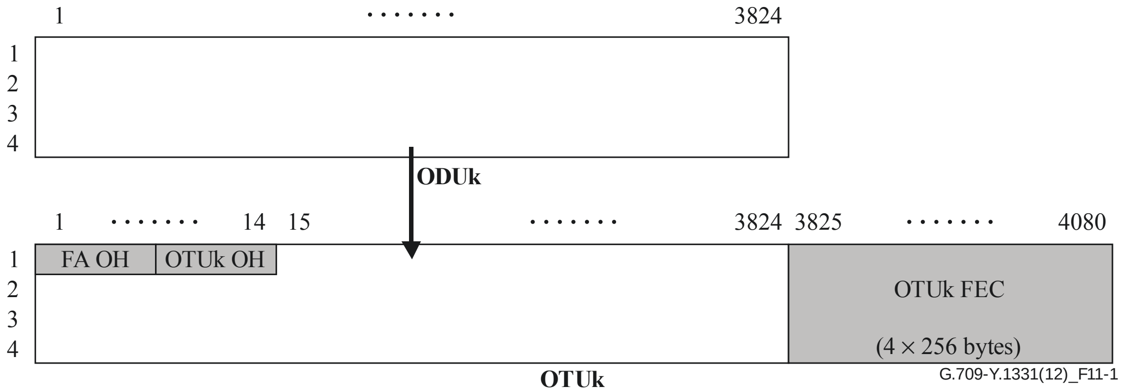


Figure 11-1 – OTUk frame structure

The bit rates of the OTUk signals are defined in Table 7-1.

The OTUk (k=1,2,3,4) forward error correction (FEC) contains the Reed-Solomon RS(255,239) FEC codes. Transmission of the OTUk FEC is mandatory for k=4 and optional for k=1,2,3. If no FEC is transmitted, fixed stuff bytes (all-0s pattern) are to be used.

The RS(255,239) FEC code shall be computed as specified in Annex A.

For interworking of equipment supporting FEC, with equipment not supporting FEC (inserting fixed stuff all-0s pattern in the OTUk (k=1,2,3) FEC area), the FEC supporting equipment shall support the capability to disable the FEC decoding process (ignore the content of the OTUk (k=1,2,3) FEC).

The transmission order of the bits in the OTUk frame is left to right, top to bottom, and MSB to LSB (see Figure 11-2).

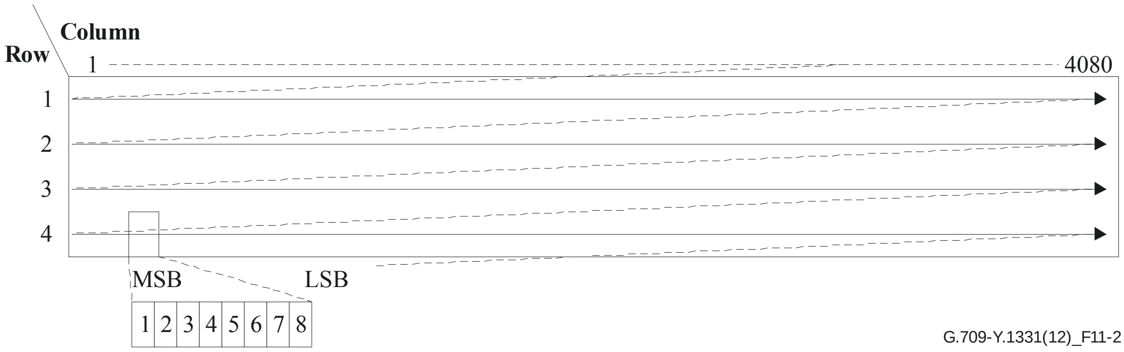


Figure 11-2 – Transmission order of the OTUk frame bits

11.2 Scrambling

The OTUk signal must have sufficient bit timing content at the ONNI. A suitable bit pattern, which prevents a long sequence of "1"s or "0"s, is provided by using a scrambler.

The operation of the scrambler shall be functionally identical to that of a frame synchronous scrambler of sequence length 65535 operating at the OTUk rate.

The generating polynomial shall be $1 + x + x^3 + x^{12} + x^{16}$. Figure 11-3 shows a functional diagram of the frame synchronous scrambler.

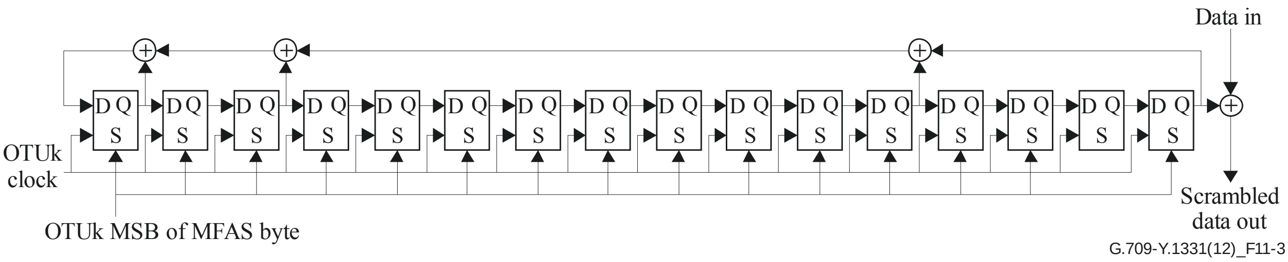


Figure 11-3 – Frame synchronous scrambler

The scrambler shall be reset to "FFFF" (HEX) on the most significant bit of the byte following the last framing byte in the OTUk frame, i.e., the MSB of the MFAS byte. This bit, and all subsequent bits to be scrambled shall be added modulo 2 to the output from the x^{16} position of the scrambler. The scrambler shall run continuously throughout the complete OTUk frame. The framing bytes (FAS) of the OTUk overhead shall not be scrambled.

Scrambling is performed after FEC computation and insertion into the OTUk signal.

12 Optical channel data unit (ODUk)

12.1 ODUk frame structure

The ODUk (k = 0,1,2,2e,3,4,flex) frame structure is shown in Figure 12-1. It is organized in an octet-based block frame structure with four rows and 3824 columns.

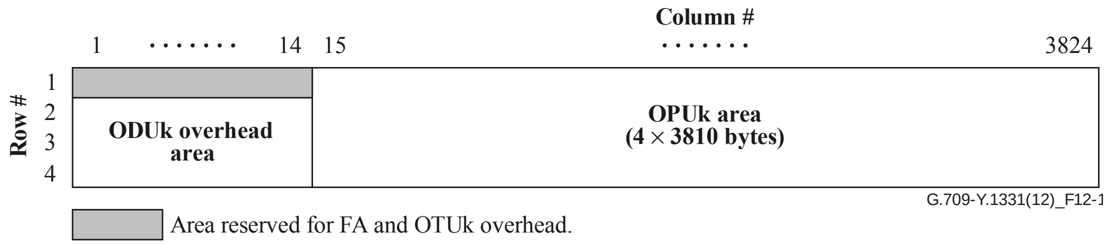


Figure 12-1 – ODUk frame structure

The two main areas of the ODUk frame are:

- ODUk overhead area
- OPUk area.

Columns 1 to 14 of the ODUk are dedicated to ODUk overhead area.

NOTE – Columns 1 to 14 of row 1 are reserved for a frame alignment and OTUk specific overhead.

Columns 15 to 3824 of the ODUk are dedicated to OPUk area.

12.2 ODUk bit rates and bit-rate tolerances

ODUk signals may be generated using either a local clock, or the recovered clock of the client signal. In the latter case the ODUk frequency and frequency tolerance are locked to the client signal's frequency and frequency tolerance. In the former case the ODUk frequency and frequency tolerance are locked to the local clock's frequency and frequency tolerance. The local clock frequency tolerance for the OTN is specified to be ±20 ppm.

ODUk maintenance signals (ODUk AIS, OCI, LCK) are generated using a local clock. In a number of cases this local clock may be the clock of a higher order signal over which the ODUk signal is transported between equipment or through equipment (in one or more of the tributary slots). For these cases, the nominal justification ratio should be deployed to comply with the ODUk's bit-rate tolerance specification.

12.2.1 ODU0, ODU1, ODU2, ODU3, ODU4

The local clocks used to create the ODU0, ODU1, ODU2, ODU3 and ODU4 signals are generated by clock crystals that are also used for the generation of SDH STM-N signals. The bit rates of these ODUk (k=0,1,2,3,4) signals are therefore related to the STM-N bit rates and the bit-rate tolerances are the bit-rate tolerances of the STM-N signals.

The ODU0 bit rate is 50% of the STM-16 bit rate.

The ODU1 bit rate is 239/238 times the STM-16 bit rate.

The ODU2 bit rate is 239/237 times 4 times the STM-16 bit rate.

The ODU3 bit rate is 239/236 times 16 times the STM-16 bit rate.

The ODU4 bit rate is 239/227 times 40 times the STM-16 bit rate.

ODU1, ODU2 and ODU3 signals which carry an STM-N (N = 16, 64, 256) signal may also be generated using the timing of these client signals.

Refer to Table 7-2 for the nominal bit rates and bit-rate tolerances.

12.2.2 ODU2e

An ODU2e signal is generated using the timing of its client signal.

The ODU2e bit rate is 239/237 times the 10GBASE-R client bit rate.

Refer to Table 7-2 for the nominal bit rate and bit-rate tolerances.

12.2.3 ODUflex for CBR client signals

An ODUflex(CBR) signal is generated using the timing of its client signal.

The ODUflex bit rate is 239/238 times the CBR client bit rate.

The client signal may have a bit-rate tolerance up to ±100 ppm.

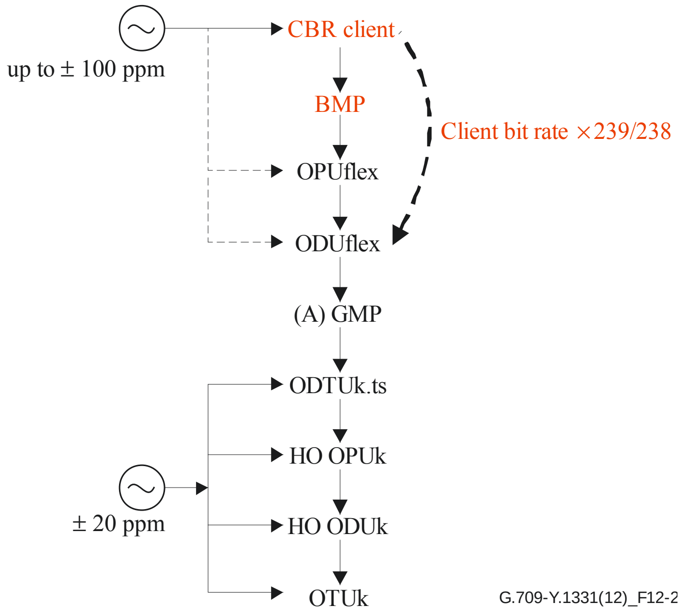


Figure 12-2 – ODUflex clock generation for CBR signals

12.2.4 ODUflex for PRBS and Null test signals

ODUflex(CBR) connections may be tested using a PRBS or NULL test signal as the client signal instead of the CBR client signal. For such a case, the ODUflex(PRBS) or ODUflex(NULL) signal should be generated with a frequency within the tolerance range of the ODUflex(CBR) signal.

If the CBR client clock is present such ODUflex(PRBS) or ODUflex(NULL) signal may be generated using the CBR client clock, otherwise the ODUflex(PRBS) or ODUflex(NULL) signal is generated using a local clock.

12.2.5 ODUflex for GFP-F mapped packet client signals

ODUflex(GFP) signals are generated using a local clock. This clock may be the local HO ODUk (or OTUk) clock, or an equipment internal clock of the signal over which the ODUflex is carried through the equipment.

Any bit rate is possible for an ODUflex(GFP) signal, however it is suggested for maximum efficiency that the ODUflex(GFP) fills an integral number of tributary slots of the smallest HO ODUk path over which the ODUflex(GFP) may be carried. The recommended bit-rates to meet this criteria are specified in Table 7-8. The derivation of the specific values is provided in Appendix XI.

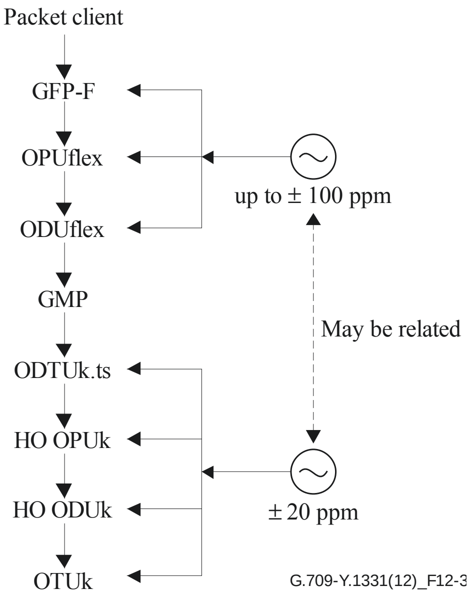


Figure 12-3 – ODUflex clock generation for GFP-F mapped packet client signals

13 Optical channel payload unit (OPUk)

The OPUk (k = 0,1,2,2e,3,4,flex) frame structure is shown in Figure 13-1. It is organized in an octet-based block frame structure with four rows and 3810 columns.

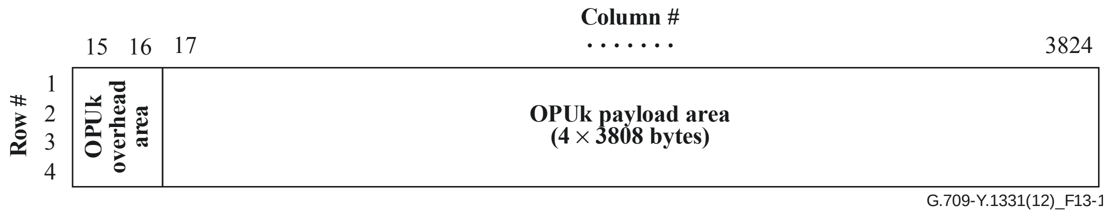


Figure 13-1 – OPUk frame structure

The two main areas of the OPUk frame are:

- OPUk overhead area
- OPUk payload area.

Columns 15 to 16 of the OPUk are dedicated to an OPUk overhead area.

Columns 17 to 3824 of the OPUk are dedicated to an OPUk payload area.

NOTE – OPUk column numbers are derived from the OPUk columns in the ODUk frame.

14 OTM overhead signal (OOS)

The OTM overhead signal (OOS) consists of the OTS, OMS and OCh overhead. The format, structure and bit rate of the OOS is not defined in this Recommendation. The OOS is transported via an OSC.

Depending on an operator's logical management overlay network design, general management communications may also be transported within the OOS. Therefore, the OOS for some applications may also transport general management communications. General management communications may include signalling, voice/voiceband communications, software download, operator-specific communications, etc.

15 Overhead description

An overview of OTS, OMS and OCh overhead is presented in Figure 15-1.

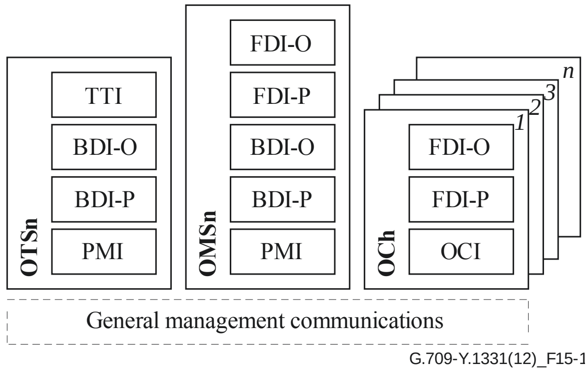


Figure 15-1 – OTSn, OMSn and OCh overhead as logical elements within the OOS

An overview of OTUk, ODUk and OPUk overhead is presented in Figures 15-2 and 15-3.

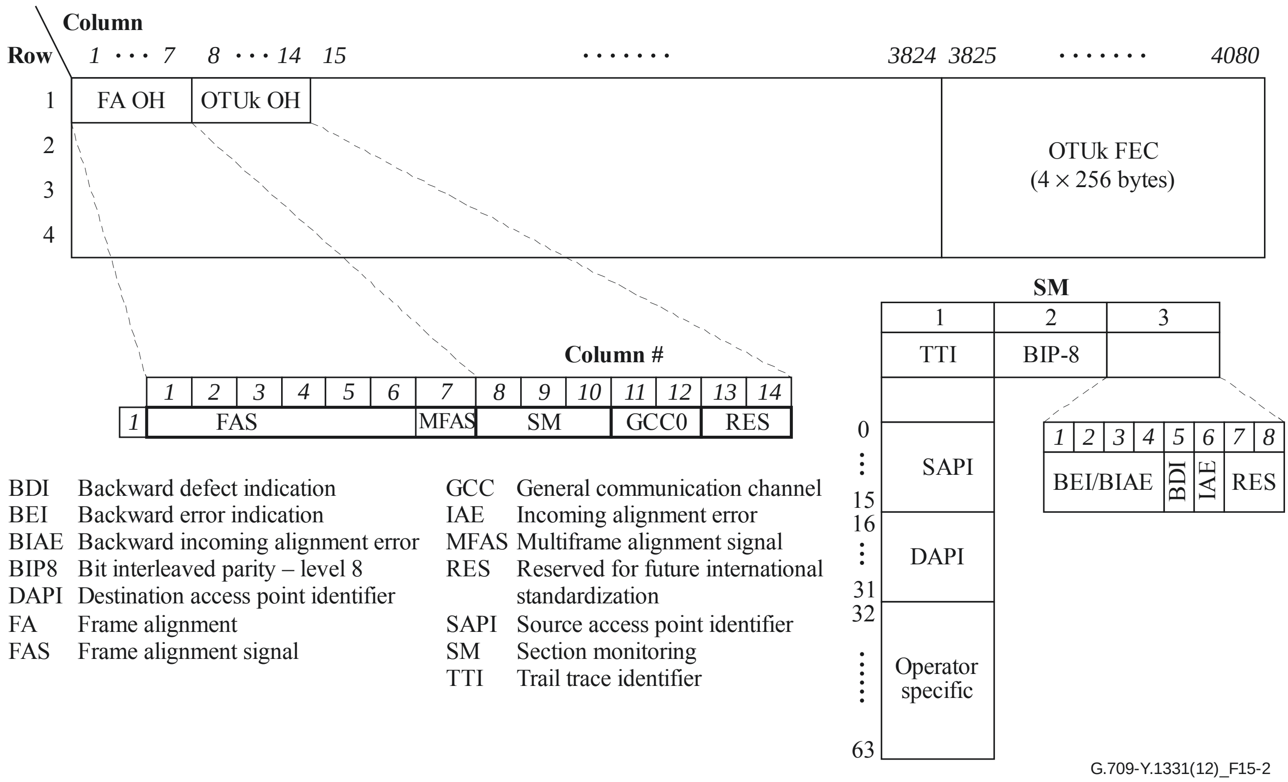


Figure 15-2 – OTUk frame structure, frame alignment and OTUk overhead

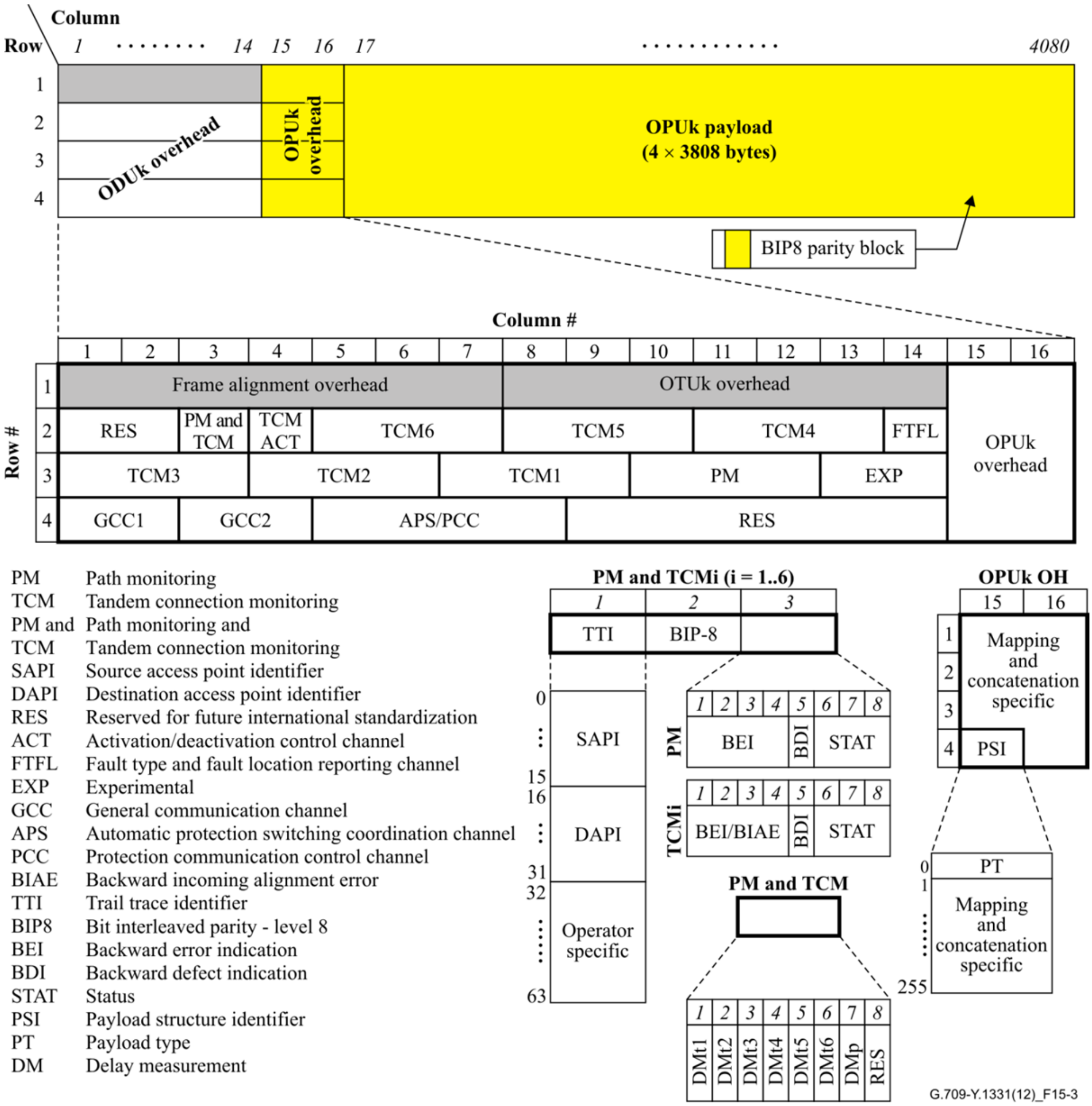


Figure 15-3 – ODUk frame structure, ODUk and OPUk overhead

15.1 Types of overhead

15.1.1 Optical channel payload unit overhead (OPUk OH)

OPUk OH information is added to the OPUk information payload to create an OPUk. It includes information to support the adaptation of client signals. The OPUk OH is terminated where the OPUk is assembled and disassembled. The specific OH format and coding is defined in clause 15.9.

15.1.2 Optical channel data unit overhead (ODUk OH)

ODUk OH information is added to the ODUk information payload to create an ODUk. It includes information for maintenance and operational functions to support optical channels. The ODUk OH consists of portions dedicated to the end-to-end ODUk path and to six levels of tandem connection monitoring. The ODUk path OH is terminated where the ODUk is assembled and disassembled. The TC OH is added and terminated at the source and sink of the corresponding tandem connections, respectively. The specific OH format and coding is defined in clauses 15.6 and 15.8.

15.1.3 Optical channel transport unit overhead (OTUk OH)

OTUk OH information is part of the OTUk signal structure. It includes information for operational functions to support the transport via one or more optical channel connections. The OTUk OH is terminated where the OTUk signal is assembled and disassembled. The specific OH format and coding is defined in clauses 15.6 and 15.7.

The specific frame structure and coding for the non-standard OTUkV OH is outside the scope of this Recommendation. Only the required basic functionality that has to be supported is defined in clause 15.7.3.

15.1.4 Optical channel non-associated overhead (OCh OH)

OCh OH information is added to the OTUk to create an OCh. It includes information for maintenance functions to support fault management. The OCh OH is terminated where the OCh signal is assembled and disassembled.

The specific frame structure and coding for the OCh OH is outside the scope of this Recommendation. Only the required basic functionality that has to be supported is defined in clause 15.5.

15.1.5 Optical multiplex section overhead (OMS OH)

OMS OH information is added to the OCG to create an OMU. It includes information for maintenance and operational functions to support optical multiplex sections. The OMS OH is terminated where the OMU is assembled and disassembled.

The specific frame structure and coding for the OMS OH is outside the scope of this Recommendation. Only the required basic functionality that has to be supported is defined in clause 15.4.

15.1.6 Optical transmission section overhead (OTS OH)

OTS OH information is added to the information payload to create an OTM. It includes information for maintenance and operational functions to support optical transmission sections. The OTS OH is terminated where the OTM is assembled and disassembled.

The specific frame structure and coding for the OTS OH is outside the scope of this Recommendation. Only the required basic functionality that has to be supported is defined in clause 15.3.

15.1.7 General management communications overhead (COMMS OH)

COMMS OH information is added to the information payload to create an OTM. It provides general management communication between network elements. The specific frame structure and coding for the COMMS OH is outside the scope of this Recommendation.

15.2 Trail trace identifier and access point identifier definition

A trail trace identifier (TTI) is defined as a 64-byte string with the following structure (see Figure 15-4):

- TTI[0] contains the SAPI[0] character, which is fixed to all-0s.
- TTI[1] to TTI[15] contain the 15-character source access point identifier (SAPI[1] to SAPI[15]).
- TTI[16] contains the DAPI[0] character, which is fixed to all-0s.
- TTI[17] to TTI[31] contain the 15-character destination access point identifier (DAPI[1] to DAPI[15]).
- TTI[32] to TTI[63] are operator specific.

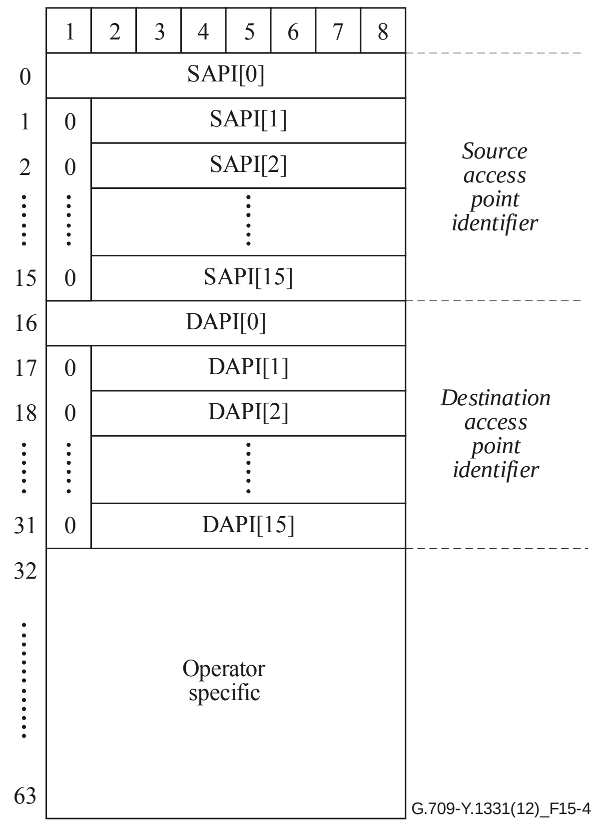


Figure 15-4 – TTI structure

The features of access point identifiers (APIs) are:

– Each access point identifier must be globally unique in its layer network.

– Where it may be expected that the access point may be required for path set-up across an inter-operator boundary, the access point identifier must be available to other network operators.

– The access point identifier should not change while the access point remains in existence.

– The access point identifier should be able to identify the country and network operator which is responsible for routing to and from the access point.

– The set of all access point identifiers belonging to a single administrative layer network should form a single access point identification scheme.

– The scheme of access point identifiers for each administrative layer network can be independent from the scheme in any other administrative layer network.

It is recommended that the ODUk, OTUk and OTM should each have the access point identification scheme based on a tree-like format to aid routing control search algorithms. The access point identifier should be globally unambiguous.

The access point identifier (SAPI, DAPI) shall consist of a three-character international segment and a twelve-character national segment (NS) (see Figure 15-5). These characters shall be coded according to [ITU-T T.50] (International Reference Alphabet – 7-bit coded character set for information exchange).

IS character #			NS character #											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CC			ICC	UAPC										
CC			ICC		UAPC									
CC			ICC			UAPC								
CC			ICC				UAPC							
CC			ICC					UAPC						
CC			ICC						UAPC					

Figure 15-5 – Access point identifier structure

The international segment field provides a three-character ISO 3166 geographic/political country code (G/PCC). The country code shall be based on the three-character uppercase alphabetic ISO 3166 country code (e.g., USA, FRA).

The national segment field consists of two subfields: the ITU carrier code (ICC) followed by a unique access point code (UAPC).

The ITU carrier code is a code assigned to a network operator/service provider, maintained by the ITU-T Telecommunication Standardization Bureau (TSB) as per [ITU-T M.1400]. This code shall consist of 1-6 left-justified characters, alphabetic, or leading alphabetic with trailing numeric.

The unique access point code shall be a matter for the organization to which the country code and ITU carrier code have been assigned, provided that uniqueness is guaranteed. This code shall consist of 6-11 characters, with trailing NUL, completing the 12-character national segment.

15.3 OTS OH description

The following OTM-n OTSn overhead is defined:

– OTSn-TTI

– OTSn-BDI-P

– OTSn-BDI-O

– OTSn-PMI.

15.3.1 OTS trail trace identifier (TTI)

The OTSn-TTI is defined to transport a 64-byte TTI as specified in clause 15.2 for OTSn section monitoring.

15.3.2 OTS backward defect indication – Payload (BDI-P)

For OTSn section monitoring, the OTSn-BDI-P signal is defined to convey in the upstream direction the OTSn payload signal fail status detected in the OTSn termination sink function.

15.3.3 OTS backward defect indication – Overhead (BDI-O)

For OTSn section monitoring, the OTSn-BDI-O signal is defined to convey in the upstream direction the OTSn overhead signal fail status detected in the OTSn termination sink function.

15.3.4 OTS payload missing indication (PMI)

The OTS PMI is a signal sent downstream as an indication that upstream at the source point of the OTS signal no payload is added, in order to suppress the report of the consequential loss of signal condition.

15.4 OMS OH description

The following OTM-n OMSn overhead is defined:

– OMSn-FDI-P
– OMSn-FDI-O
– OMSn-BDI-P
– OMSn-BDI-O
– OMSn-PMI.

15.4.1 OMS forward defect indication – Payload (FDI-P)

For OMSn section monitoring, the OMSn-FDI-P signal is defined to convey in the downstream direction the OMSn payload signal status (normal or failed).

15.4.2 OMS forward defect indication – Overhead (FDI-O)

For OMSn section monitoring, the OMSn-FDI-O signal is defined to convey in the downstream direction the OMSn overhead signal status (normal or failed).

15.4.3 OMS backward defect indication – Payload (BDI-P)

For OMSn section monitoring, the OMSn-BDI-P signal is defined to convey in the upstream direction the OMSn payload signal fail status detected in the OMSn termination sink function.

15.4.4 OMS backward defect indication – Overhead (BDI-O)

For OMSn section monitoring, the OMSn-BDI-O signal is defined to convey in the upstream direction the OMSn overhead signal fail status detected in the OMSn termination sink function.

15.4.5 OMS payload missing indication (PMI)

The OMS PMI is a signal sent downstream as an indication that upstream at the source point of the OMS signal none of the OCCps contain an optical channel signal, in order to suppress the report of the consequential loss of signal condition.

15.5 OCh OH description

The following OTM-n OCh overhead is defined:

– OCh-FDI-P
– OCh-FDI-O
– OCh-OCI.

15.5.1 OCh forward defect indication – Payload (FDI-P)

For OCh trail monitoring, the OCh-FDI-P signal is defined to convey in the downstream direction the OCh payload signal status (normal or failed).

15.5.2 OCh forward defect indication – Overhead (FDI-O)

For OCh trail monitoring, the OCh-FDI-O signal is defined to convey in the downstream direction the OCh overhead signal status (normal or failed).

15.5.3 OCh open connection indication (OCI)

The OCh OCI is a signal sent downstream as an indication that upstream in a connection function the matrix connection is opened as a result of a management command. The consequential detection of the OCh loss of signal condition at the OCh termination point can now be related to an open matrix.

15.6 OTUk/ODUk frame alignment OH description

15.6.1 OTUk/ODUk frame alignment overhead location

The OTUk/ODUk frame alignment overhead location is shown in Figure 15-6. The OTUk/ODUk frame alignment overhead is applicable for both the OTUk and ODUk signals.

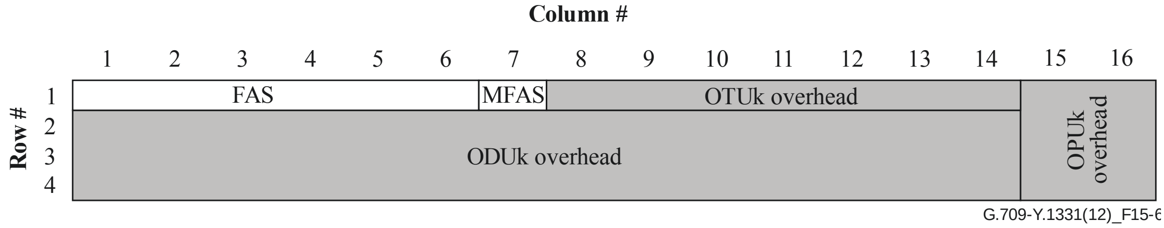


Figure 15-6 – OTUk/ODUk frame alignment overhead

15.6.2 OTUk/ODUk frame alignment overhead definition

15.6.2.1 Frame alignment signal (FAS)

A six byte OTUk-FAS signal (see Figure 15-7) is defined in row 1, columns 1 to 6 of the OTUk overhead. OA1 is "1111 0110". OA2 is "0010 1000".

FAS OH byte 1	FAS OH byte 2	FAS OH byte 3	FAS OH byte 4	FAS OH byte 5	FAS OH byte 6
1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
OA1	OA1	OA1	OA2	OA2	OA2

G.709-Y.1331(12)_F15-7

Figure 15-7 – Frame alignment signal overhead structure

15.6.2.2 Multiframe alignment signal (MFAS)

Some of the OTUk and ODUk overhead signals will span multiple OTUk/ODUk frames. Examples are the TTI and TCM-ACT overhead signals. These and other multiframe structured overhead signals require multiframe alignment processing to be performed, in addition to the OTUk/ODUk frame alignment.

A single multiframe alignment signal (MFAS) byte is defined in row 1, column 7 of the OTUk/ODUk overhead for this purpose (see Figure 15-8). The value of the MFAS byte will be incremented each OTUk/ODUk frame and provides as such a 256-frame multiframe.

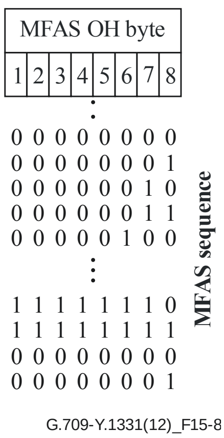


Figure 15-8 – Multiframe alignment signal overhead

Individual OTUk/ODUk overhead signals may use this central multiframe to lock their 2-frame, 4-frame, 8-frame, 16-frame, 32-frame, etc., multiframes to the principal frame.

NOTE – The 80-frame HO OPU4 multiframe cannot be supported. A dedicated 80-frame OPU4 multiframe indicator (OMFI) is used instead.

15.7 OTUk OH description

15.7.1 OTUk overhead location

The OTUk overhead location is shown in Figures 15-9 and 15-10.

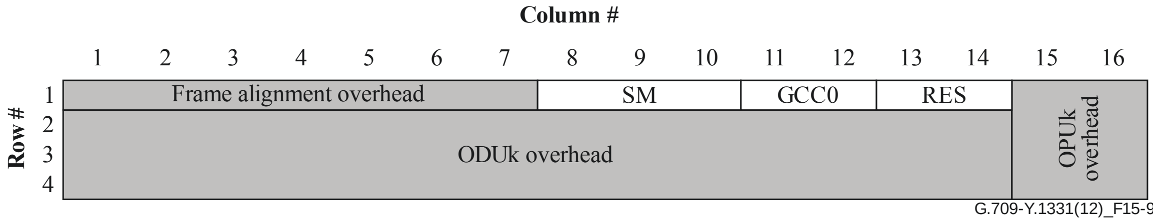


Figure 15-9 – OTUk overhead

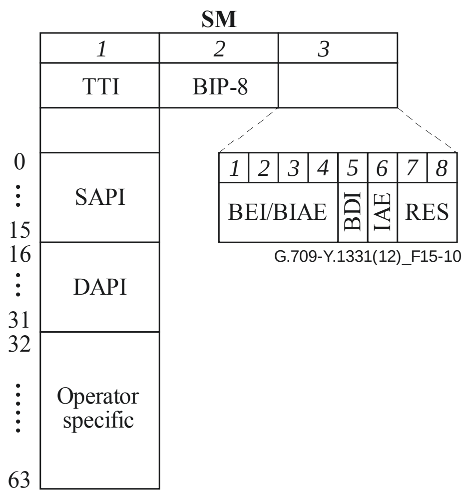


Figure 15-10 – OTUk section monitoring overhead

15.7.2 OTUk overhead definition

15.7.2.1 OTUk section monitoring (SM) overhead

One field of OTUk section monitoring (SM) overhead is defined in row 1, columns 8 to 10 to support section monitoring.

The SM field contains the following subfields (see Figure 15-10):

– trail trace identifier (TTI)
– bit interleaved parity (BIP-8)
– backward defect indication (BDI)
– backward error indication and backward incoming alignment error (BEI/BIAE)
– incoming alignment error (IAE)
– bits reserved for future international standardization (RES).

15.7.2.1.1 OTUk SM trail trace identifier (TTI)

For section monitoring, a one-byte trail trace identifier (TTI) overhead is defined to transport the 64-byte TTI signal specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1].

The 64-byte TTI signal shall be aligned with the OTUk multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at OTUk multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.7.2.1.2 OTUk SM error detection code (BIP-8)

For section monitoring, a one-byte error detection code signal is defined. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation *BIP-8* refers only to the number of BIP bits and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8 refer to BIP-X definition in [ITU-T G.707].

The OTUk BIP-8 is computed over the bits in the OPUk (columns 15 to 3824) area of OTUk frame i, and inserted in the OTUk BIP-8 overhead location in OTUk frame i+2 (see Figure 15-11).

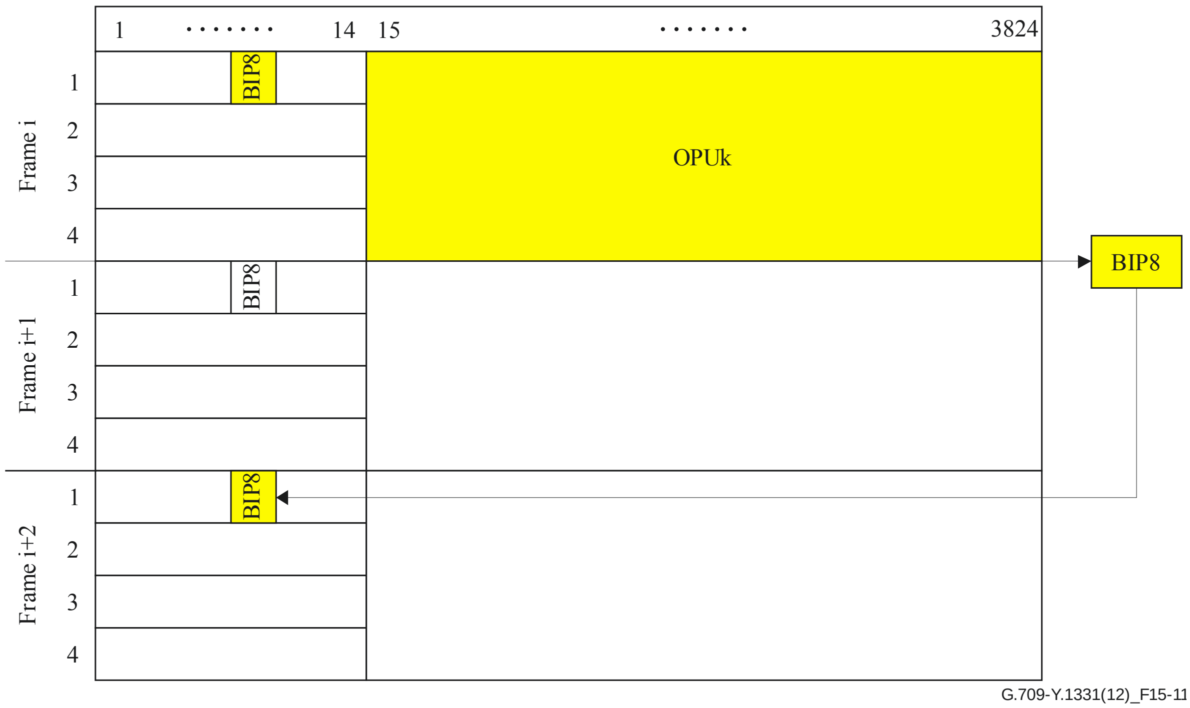


Figure 15-11 – OTUk SM BIP-8 computation

15.7.2.1.3 OTUk SM backward defect indication (BDI)

For section monitoring, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a section termination sink function in the upstream direction.

BDI is set to "1" to indicate an OTUk backward defect indication; otherwise, it is set to "0".

15.7.2.1.4 OTUk SM backward error indication and backward incoming alignment error (BEI/BIAE)

For section monitoring, a four-bit backward error indication (BEI) and backward incoming alignment error (BIAE) signal is defined. This signal is used to convey in the upstream direction the count of interleaved-bit blocks that have been detected in error by the corresponding OTUk section monitoring sink using the BIP-8 code. It is also used to convey in the upstream direction an incoming alignment error (IAE) condition that is detected in the corresponding OTUk section monitoring sink in the IAE overhead.

During an IAE condition the code "1011" is inserted into the BEI/BIAE field and the error count is ignored. Otherwise the error count (0-8) is inserted into the BEI/BIAE field. The remaining six possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-1) and BIAE not active.

Table 15-1 – OTUk SM BEI/BIAE interpretation

OTUk SM BEI/BIAE **bits 1 2 3 4**	**BIAE**	**BIP violations**
0 0 0 0	false	0
0 0 0 1	false	1
0 0 1 0	false	2
0 0 1 1	false	3
0 1 0 0	false	4
0 1 0 1	false	5
0 1 1 0	false	6
0 1 1 1	false	7
1 0 0 0	false	8
1 0 0 1, 1 0 1 0	false	0
1 0 1 1	true	0
1 1 0 0 to 1 1 1 1	false	0

15.7.2.1.5 OTUk SM incoming alignment error overhead (IAE)

A single-bit incoming alignment error (IAE) signal is defined to allow the S-CMEP ingress point to inform its peer S-CMEP egress point that an alignment error in the incoming signal has been detected.

IAE is set to "1" to indicate a frame alignment error, otherwise it is set to "0".

The S-CMEP egress point may use this information to suppress the counting of bit errors, which may occur as a result of a frame phase change of the OTUk at the ingress of the section.

15.7.2.1.6 OTUk SM reserved overhead (RES)

For section monitoring, two bits are reserved (RES) for future international standardization. They are set to "00".

15.7.2.2 OTUk general communication channel 0 (GCC0)

Two bytes are allocated in the OTUk overhead to support a general communications channel or a discovery channel as specified in [ITU-T G.7714.1] between OTUk termination points.

This general communication channel is a clear channel and any format specification is outside of the scope of this Recommendation. These bytes are located in row 1, columns 11 and 12 of the OTUk overhead.

15.7.2.3 OTUk reserved overhead (RES)

Two bytes of an OTUk overhead are reserved for future international standardization. These bytes are located in row 1, columns 13 and 14. These bytes are set to all-0s.

15.7.3 OTUkV overhead

The functionally standardized OTUkV frame should support, as a minimum capability, section monitoring functionality comparable to the OTUk section monitoring (see clause 15.7.2.1) with a trail trace identifier as specified in clause 15.2. Further specification of this overhead is outside the scope of this Recommendation.

15.8 ODUk OH description

15.8.1 ODUk OH location

The ODUk overhead location is shown in Figures 15-12, 15-13 and 15-14.

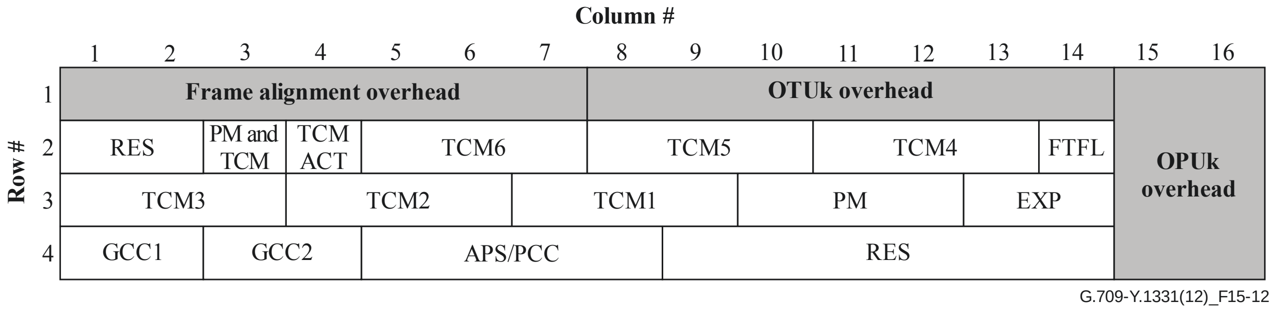


Figure 15-12 – ODUk overhead

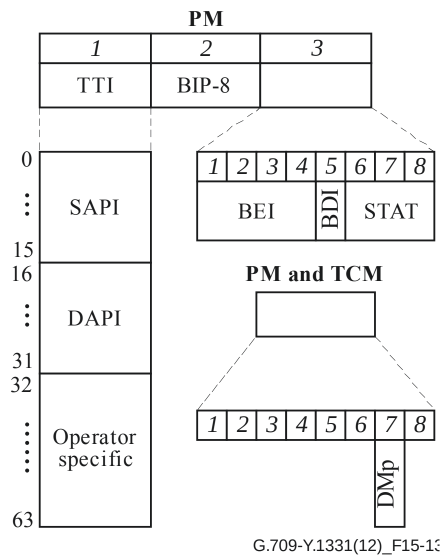


Figure 15-13 – ODUk path monitoring overhead

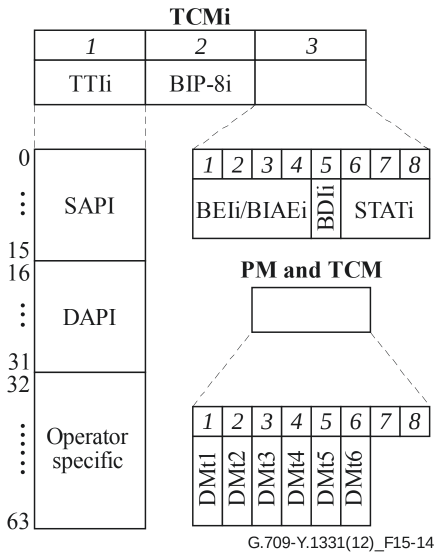


Figure 15-14 – ODUk tandem connection monitoring #i overhead

15.8.2 ODUk OH definition

15.8.2.1 ODUk path monitoring (PM) overhead

One field of an ODUk path monitoring overhead (PM) is defined in row 3, columns 10 to 12 to support path monitoring and one additional bit of path monitoring is defined in row 2, column 3, bit 7.

The PM field contains the following subfields (see Figure 15-13):

- trail trace identifier (TTI)
- bit interleaved parity (BIP-8)
- backward defect indication (BDI)
- backward error indication (BEI)
- status bits indicating the presence of a maintenance signal (STAT).

The PM&TCM field contains the following PM subfield (see Figure 15-13):

- path delay measurement (DMp).

The content of the PM field, except the STAT subfield, will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODUk-AIS, ODUk-OCI, ODUk-LCK). The content of the PM&TCM field will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal. Refer to clause 16.5.

15.8.2.1.1 ODUk PM trail trace identifier (TTI)

For path monitoring, a one-byte trail trace identifier (TTI) overhead is defined to transport the 64-byte TTI signal specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1].

The 64-byte TTI signal shall be aligned with the ODUk multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at ODUk multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.8.2.1.2 ODUk PM error detection code (BIP-8)

For path monitoring, a one-byte error detection code signal is defined. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation BIP-8 refers only to the number of BIP bits and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8, refer to the BIP-X definition in [ITU-T G.707].

Each ODUk BIP-8 is computed over the bits in the OPUk (columns 15 to 3824) area of ODUk frame i, and inserted in the ODUk PM BIP-8 overhead location in the ODUk frame i+2 (see Figure 15-15).

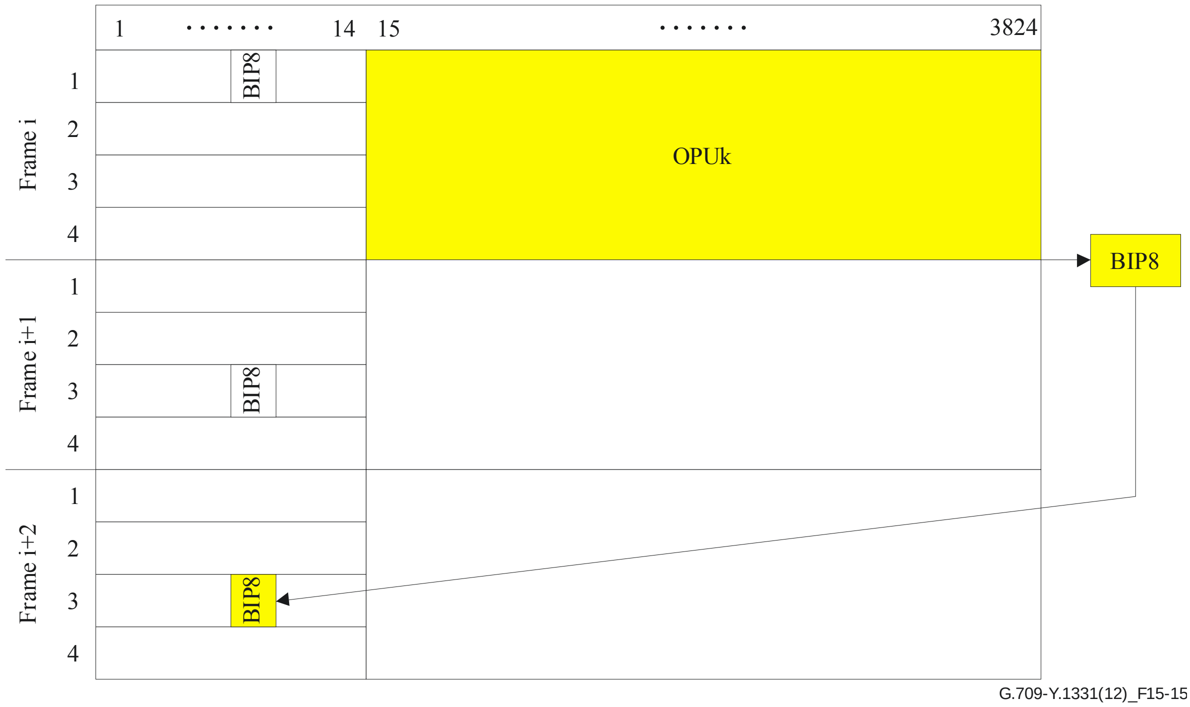


Figure 15-15 – ODUk PM BIP-8 computation

15.8.2.1.3 ODUk PM backward defect indication (BDI)

For path monitoring, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a path termination sink function in the upstream direction.

BDI is set to "1" to indicate an ODUk backward defect indication, otherwise it is set to "0".

15.8.2.1.4 ODUk PM backward error indication (BEI)

For path monitoring, a four-bit backward error indication (BEI) signal is defined to convey in the upstream direction the count of interleaved-bit blocks that have been detected in error by the corresponding ODUk path monitoring sink using the BIP-8 code. This count has nine legal values, namely 0-8 errors. The remaining seven possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-2).

Table 15-2 – ODUk PM BEI interpretation

ODUk PM BEI bits 1 2 3 4	BIP violations
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2

Table 15-2 – ODUk PM BEI interpretation

ODUk PM BEI bits 1 2 3 4	BIP violations
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1 to 1 1 1 1	0

15.8.2.1.5 ODUk PM status (STAT)

For path monitoring, three bits are defined as status bits (STAT). They indicate the presence of a maintenance signal (see Table 15-3).

A P-CMEP sets these bits to "001".

Table 15-3 – ODUk PM status interpretation

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
1 0 1	Maintenance signal: ODUk-LCK
1 1 0	Maintenance signal: ODUk-OCI
1 1 1	Maintenance signal: ODUk-AIS

15.8.2.1.6 ODUk PM delay measurement (DMp)

For ODUk path monitoring, a one-bit path delay measurement (DMp) signal is defined to convey the start of the delay measurement test.

The DMp signal consists of a constant value (0 or 1) that is inverted at the beginning of a two-way delay measurement test. The transition from 0→1 in the sequence …0000011111…, or the transition from 1→0 in the sequence …1111100000… represents the path delay measurement start point. The new value of the DMp signal is maintained until the start of the next delay measurement test.

This DMp signal is inserted by the DMp originating P-CMEP and sent to the far-end P-CMEP. This far-end P-CMEP loops back the DMp signal towards the originating P-CMEP. The originating P-CMEP measures the number of frame periods between the moment the DMp signal value is inverted and the moment this inverted DMp signal value is received back from the far-end P-CMEP. The receiver should apply a persistency check on the received DMp signal to be tolerant for bit errors emulating the start of delay measurement indication. The additional frames that are used for such persistency checking should not be added to the delay frame count. The looping P-CMEP should loop back each received DMp bit within approximately 100 µs.

Refer to [ITU-T G.798] for the specific path delay measurement process specifications.

NOTE 1 – Path delay measurements can be performed on-demand, to provide the momentary two-way transfer delay status, and pro-active, to provide 15-minute and 24-hour two-way transfer delay performance management snapshots.

NOTE 2 – Equipment designed according to the 2008 or earlier versions of this Recommendation may not be capable of supporting this path delay monitoring. For such equipment, the DMp bit is a bit reserved for future international standardization and set to zero.

NOTE 3 – This process measures a round trip delay. The one way delay may not be half of the round trip delay in the case where the transmit and receive directions of the ODUk path are of unequal lengths (e.g., in networks deploying unidirectional protection switching).

15.8.2.2 ODUk tandem connection monitoring (TCM) overhead

Six fields of an ODUk tandem connection monitoring (TCM) overhead are defined in row 2, columns 5 to 13 and row 3, columns 1 to 9 of the ODUk overhead; and six additional bits of tandem connection monitoring are defined in row 2, column 3, bits 1 to 6. TCM supports monitoring of ODUk connections for one or more of the following network applications (refer to [ITU-T G.805], [ITU-T G.872], [ITU-T G.873.2] and [ITU-T G.7714.1]):

– optical UNI-to-UNI tandem connection monitoring; monitoring the ODUk connection through the public transport network (from public network ingress network termination to egress network termination);
– optical NNI-to-NNI tandem connection monitoring; monitoring the ODUk connection through the network of a network operator (from operator network ingress network termination to egress network termination);
– sublayer monitoring for linear 1+1, 1:1 and 1:n optical channel subnetwork connection protection switching, to determine the signal fail and signal degrade conditions;
– sublayer monitoring for optical channel data unit shared ring protection (SRP-1) protection switching as specified in [ITU-T G.873.2], to determine the signal fail and signal degrade conditions;
– sublayer monitoring for optical channel data unit connection passing through two or more concatenated ODUk link connections (supported by back-to-back OTUk trails), to provide a discovery message channel as specified in [ITU-T G.7714.1];
– monitoring an optical channel tandem connection for the purpose of detecting a signal fail or signal degrade condition in a switched optical channel connection, to initiate automatic restoration of the connection during fault and error conditions in the network;
– monitoring an optical channel tandem connection for, e.g., fault localization or verification of delivered quality of service.

The six TCM fields are numbered TCM1, TCM2, ..., TCM6.

Each TCM field contains the following subfields (see Figure 15-14):

– trail trace identifier (TTI)
– bit interleaved parity 8 (BIP-8)
– backward defect indication (BDI)
– backward error indication and backward incoming alignment error (BEI/BIAE)
– status bits indicating the presence of a TCM overhead, incoming alignment error, or a maintenance signal (STAT).

The PM&TCM field contains the following TCM subfields (see Figure 15-14):

– tandem connection delay measurement (DMti, i=1 to 6).

The content of the TCM fields, except the STAT subfield, will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODUk-AIS, ODUk-OCI, ODUk-LCK). The content of the PM&TCM field will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal. Refer to clause 16.5.

A TCM field and PM&TCM bit is assigned to a monitored connection as described in clause 15.8.2.2.6. The number of monitored connections along an ODUk trail may vary between 0 and 6. These monitored connections may be nested, cascaded or both. Nesting and cascading are the default operational configurations. Overlapping is an additional configuration for testing purposes only. Overlapped monitored connections must be operated in a non-intrusive mode in which the maintenance signals ODUk-AIS and ODUk-LCK are not generated. For the case where one of the endpoints in an overlapping monitored connection is located inside an SNC protected domain while the other endpoint is located outside the protected domain, the SNC protection should be forced to working when the endpoint of the overlapping monitored connection is located on the working connection, and forced to protection when the endpoint is located on the protection connection.

Nesting and cascading configurations are shown in Figure 15-16. Monitored connections A1-A2/B1-B2/C1-C2 and A1-A2/B3-B4 are nested, while B1-B2/B3-B4 are cascaded. Overlapping is shown in Figure 15-17 (B1-B2 and C1-C2).

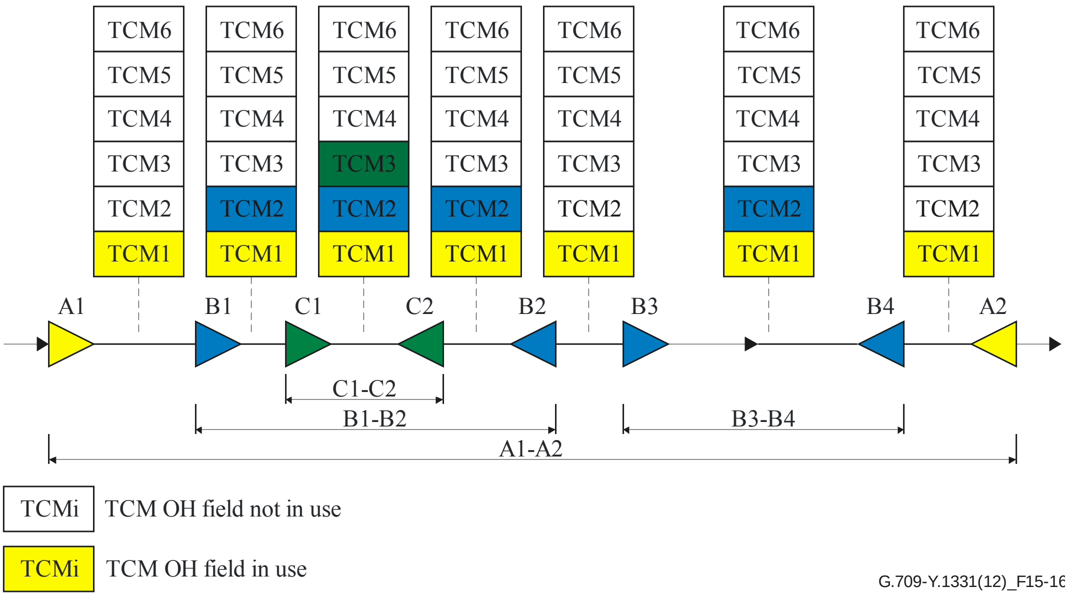


Figure 15-16 – Example of nested and cascaded ODUk monitored connections

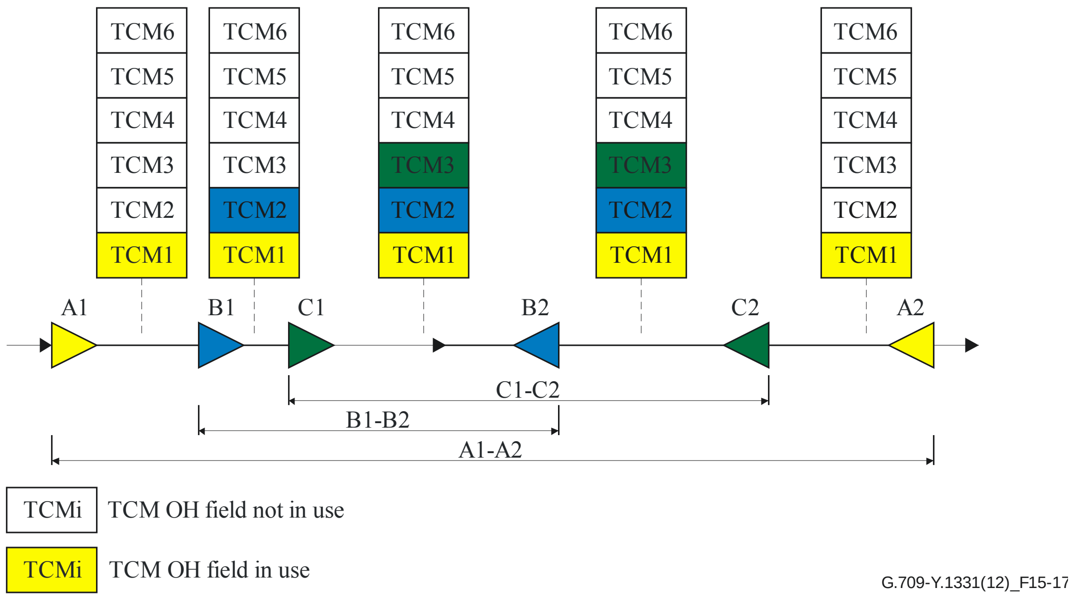


Figure 15-17 – Example of overlapping ODUk monitored connections

15.8.2.2.1 ODUk TCM trail trace identifier (TTI)

For each tandem connection monitoring field, one byte of overhead is allocated for the transport of the 64-byte trail trace identifier (TTI) specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1] for TCM6.

The 64-byte TTI signal shall be aligned with the ODUk multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at ODUk multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.8.2.2.2 ODUk TCM error detection code (BIP-8)

For each tandem connection monitoring field, a one-byte error detection code signal is defined. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation *BIP-8* refers only to the number of BIP bits, and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8 refer to the BIP-X definition in [ITU-T G.707].

Each ODUk TCM BIP-8 is computed over the bits in the OPUk (columns 15 to 3824) area of ODUk frame i, and inserted in the ODUk TCM BIP-8 overhead location (associated with the tandem connection monitoring level) in ODUk frame i+2 (see Figure 15-18).

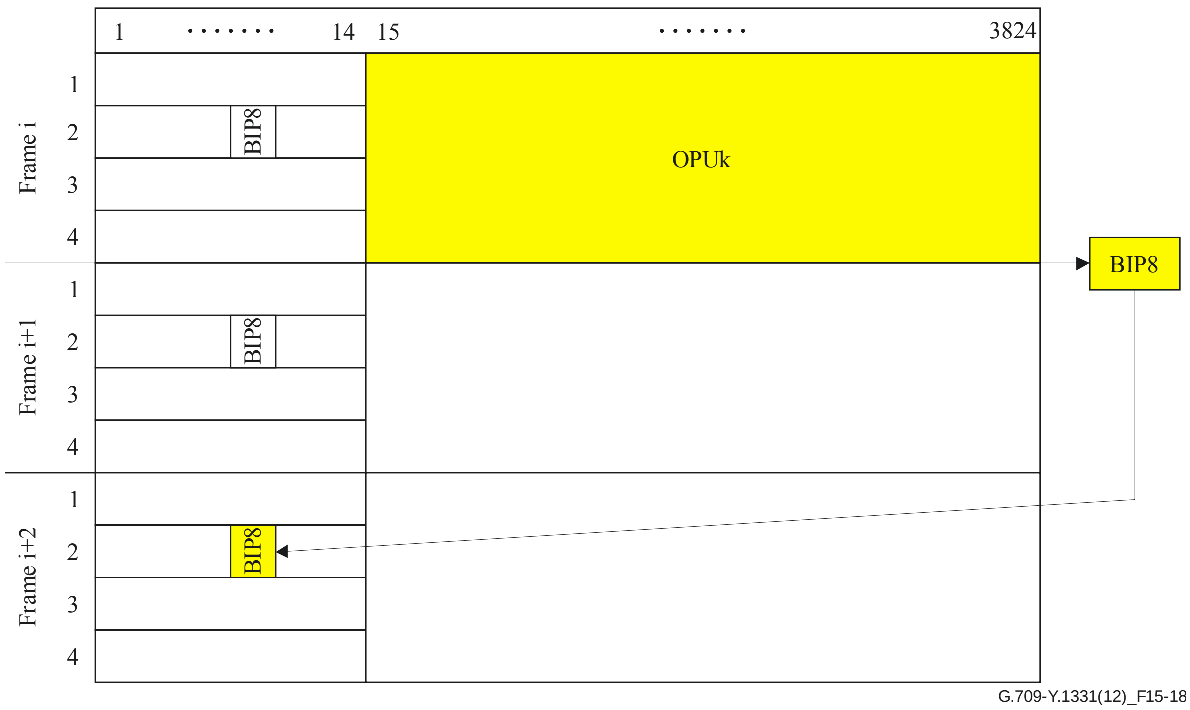


Figure 15-18 – ODUk TCM BIP-8 computation

15.8.2.2.3 ODUk TCM backward defect indication (BDI)

For each tandem connection monitoring field, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a tandem connection termination sink function in the upstream direction.

BDI is set to "1" to indicate an ODUk backward defect indication; otherwise, it is set to "0".

15.8.2.2.4 ODUk TCM backward error indication (BEI) and backward incoming alignment error (BIAE)

For each tandem connection monitoring field, a 4-bit backward error indication (BEI) and backward incoming alignment error (BIAE) signal is defined. This signal is used to convey in the upstream direction the count of interleaved-bit blocks that have been detected as being in error by the corresponding ODUk tandem connection monitoring sink using the BIP-8 code. It is also used to convey in the upstream direction an incoming alignment error (IAE) condition that is detected in the corresponding ODUk tandem connection monitoring sink in the IAE overhead.

During an IAE condition the code "1011" is inserted into the BEI/BIAE field and the error count is ignored. Otherwise the error count (0-8) is inserted into the BEI/BIAE field. The remaining six possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-4) and BIAE not active.

Table 15-4 – ODUk TCM BEI/BIAE interpretation

ODUk TCM BEI/BIAE bits 1 2 3 4	BIAE	BIP violations
0 0 0 0	false	0
0 0 0 1	false	1
0 0 1 0	false	2
0 0 1 1	false	3
0 1 0 0	false	4

Table 15-4 – ODUk TCM BEI/BIAE interpretation

ODUk TCM BEI/BIAE bits 1 2 3 4	BIAE	BIP violations
0 1 0 1	false	5
0 1 1 0	false	6
0 1 1 1	false	7
1 0 0 0	false	8
1 0 0 1, 1 0 1 0	false	0
1 0 1 1	true	0
1 1 0 0 to 1 1 1 1	false	0

15.8.2.2.5 ODUk TCM status (STAT)

For each tandem connection monitoring field, three bits are defined as status bits (STAT). They indicate the presence of a maintenance signal, if there is an incoming alignment error at the source TC-CMEP, or if there is no source TC-CMEP active (see Table 15-5).

Table 15-5 – ODUk TCM status interpretation

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
1 0 1	Maintenance signal: ODUk-LCK
1 1 0	Maintenance signal: ODUk-OCI
1 1 1	Maintenance signal: ODUk-AIS

A P-CMEP sets these bits to "000".

A TC-CMEP ingress point sets these bits to either "001" to indicate to its peer TC-CMEP egress point that there is no incoming alignment error (IAE), or to "010" to indicate that there is an incoming alignment error.

The TC-CMEP egress point may use this information to suppress the counting of bit errors, which may occur as a result of a frame phase change of the ODUk at the ingress of the tandem connection.

15.8.2.2.6 TCM overhead field assignment

Each TC-CMEP will be inserting/extracting its TCM overhead from one of the 6 TCM_i overhead fields and one of the 6 DMt_i fields. The specific TCM_i/DMt_i overhead field is provisioned by the network operator, network management system or switching control plane.

At a domain interface, it is possible to provision the maximum number (0 to 6) of tandem connection levels which will be passed through the domain. The default is three. These tandem connections should use the lower TCM_i/DMt_i overhead fields $TCM_1/DMt_1...TCM_{MAX}/DMt_{MAX}$.

Overhead in TCM/DMt fields beyond the maximum (TCM_{max+1}/DMt_{max+1} and above) may/will be overwritten in the domain.

The TCM6 overhead field is assigned to monitor an ODUk connection which is supported by two or more concatenated ODUk link connections (supported by back-to-back OTUk trails). [ITU-T G.7714.1] specifies a discovery application which uses the TCM6 TTI SAPI field as discovery message channel. [ITU-T G.873.2] specifies an ODU SRP-1 protection application which uses the TCM6 field to monitor the status/performance of the ODU connection between two adjacent ODU SRP-1 nodes.

Example

For the case of an ODUk leased circuit, the user may have been assigned one level of TCM, the service provider one level of TCM and each network operator (having a contract with the service provider) four levels of TCM. For the case where a network operator subcontracts part of its ODUk connection to another network operator, these four levels are to be split; e.g., two levels for the subcontracting operator.

This would result in the following TCM OH allocation:

– User: TCM1/DMt1 overhead field between the two user subnetworks, and TCM1/DMt1..TCM6/DMt6 within its own subnetwork;
– Service provider (SP): TCM2/DMt2 overhead field between two UNIs;
– Network operators NO1, NO2, NO3 having contract with service provider: TCM3/DMt3, TCM4/DMt4, TCM5/DMt5, TCM6/DMt6. Note that NO2 (which is subcontracting) cannot use TCM5/DMt5 and TCM6/DMt6 in the connection through the domain of NO4;
– NO4 (having subcontract with NO2): TCM5/DMt5, TCM6/DMt6.

See Figure 15-19.

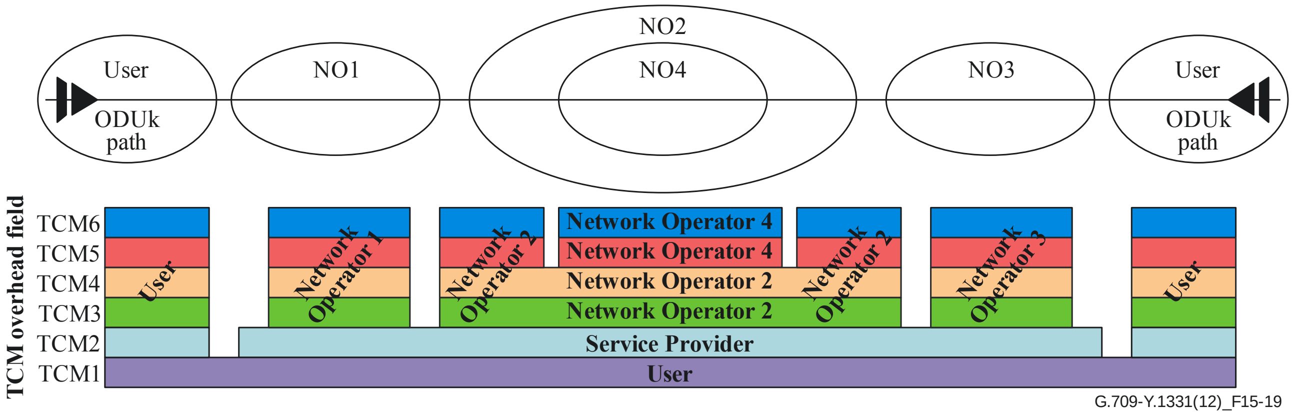


Figure 15-19 – Example of TCM overhead field assignment

15.8.2.2.7 ODUk tandem connection monitoring activation/deactivation coordination protocol

A one-byte TCM activation/deactivation field is located in row 2, column 4. Its definition is for further study.

15.8.2.2.8 ODUk TCM delay measurement (DMti, i=1 to 6)

For ODUk tandem connection monitoring, a one-bit tandem connection delay measurement (DMti) signal is defined to convey the start of the delay measurement test.

The DMti signal consists of a constant value (0 or 1) that is inverted at the beginning of a two-way delay measurement test. The transition from 0→1 in the sequence …0000011111…, or the transition from 1→0 in the sequence …1111100000… represents the path delay measurement start point. The new value of the DMti signal is maintained until the start of the next delay measurement test.

This DMti signal is inserted by the DMti originating TC-CMEP and sent to the far-end TC-CMEP. This far-end TC-CMEP loops back the DMti signal towards the originating TC-CMEP. The originating TC-CMEP measures the number of frame periods between the moment the DMti signal value is inverted and the moment this inverted DMti signal value is received back from the far-end TC-CMEP. The receiver should apply a persistency check on the received DMti signal to be tolerant for bit errors emulating the start of delay measurement indication. The additional frames that are used for such persistency checking should not be added to the delay frame count. The looping TC-CMEP should loop back each received DMti bit within approximately 100 µs.

Refer to [ITU-T G.798] for the specific tandem connection delay measurement process specifications.

NOTE 1 – Tandem connection delay measurements can be performed on-demand, to provide the momentary two-way transfer delay status, and pro-active, to provide 15-minute and 24-hour two-way transfer delay performance management snapshots.

NOTE 2 – Equipment designed according to the 2008 or earlier versions of this Recommendation may not be capable of supporting this tandem connection delay monitoring. For such equipment, the DMti bit is a bit reserved for future international standardization.

NOTE 3 – This process measures a round trip delay. The one way delay may not be half of the round trip delay in the case where the transmit and receive directions of the ODUk tandem connection are of unequal lengths (e.g., in networks deploying unidirectional protection switching).

15.8.2.3 ODUk general communication channels (GCC1, GCC2)

Two fields of two bytes are allocated in the ODUk overhead to support two general communications channels or two discovery channels as specified in [ITU-T G.7714.1] between any two network elements with access to the ODUk frame structure (i.e., at 3R regeneration points).

These general communication channels are clear channels and any format specification is outside of the scope of this Recommendation. The bytes for GCC1 are located in row 4, columns 1 and 2, and the bytes for GCC2 are located in row 4, columns 3 and 4 of the ODUk overhead.

15.8.2.4 ODUk automatic protection switching and protection communication channel (APS/PCC)

A four-byte ODUk-APS/PCC signal is defined in row 4, columns 5 to 8 of the ODUk overhead. Up to eight levels of nested APS/PCC signals may be present in this field. The APS/PCC bytes in a given frame are assigned to a dedicated connection monitoring level depending on the value of MFAS as follows:

Table 15-6 – Multiframe to allow separate APS/PCC for each monitoring level

MFAS bits 6 7 8	APS/PCC channel applies to connection monitoring level	Protection scheme using the APS/PCC channel (Note 1)
0 0 0	ODUk Path	ODUk SNC/Ne, ODUj CL-SNCG/I, Client SNC/I, ODU SRP-p
0 0 1	ODUk TCM1	ODUk SNC/S, ODUk SNC/Ns
0 1 0	ODUk TCM2	ODUk SNC/S, ODUk SNC/Ns
0 1 1	ODUk TCM3	ODUk SNC/S, ODUk SNC/Ns
1 0 0	ODUk TCM4	ODUk SNC/S, ODUk SNC/Ns
1 0 1	ODUk TCM5	ODUk SNC/S, ODUk SNC/Ns

Table 15-6 – Multiframe to allow separate APS/PCC for each monitoring level

1 1 0	ODUk TCM6	ODUk SNC/S, ODUk SNC/Ns, ODU SRP-1
1 1 1	ODUk server layer trail (Note 2)	ODUk SNC/I
NOTE 1 – An APS channel may be used by more than one protection scheme and/or protection scheme instance. In case of nested protection schemes, care should be taken when an ODUk protection is to be set up in order not to interfere with the APS channel usage of another ODUk protection on the same connection monitoring level, e.g., protection can only be activated if that APS channel of the level is not already being used. NOTE 2 – Examples of ODUk server layer trails are an OTUk or an HO ODUk (e.g., an ODU3 transporting an ODU1).		

For linear protection schemes, the bit assignments for these bytes and the bit-oriented protocol are given in [ITU-T G.873.1]. Bit assignment and byte-oriented protocol for ring protection schemes are given in [ITU-T G.873.2].

15.8.2.5 ODUk fault type and fault location reporting communication channel (FTFL)

One byte is allocated in the ODUk overhead to transport a 256-byte fault type and fault location (FTFL) message. The byte is located in row 2, column 14 of the ODUk overhead.

The 256-byte FTFL message shall be aligned with the ODUk multiframe (i.e., byte 0 of the 256-byte FTFL message shall be present at ODUk multiframe position 0000 0000, byte 1 of the 256-byte FTFL message shall be present at ODUk multiframe position 0000 0001, byte 2 of the 256-byte FTFL message shall be present at ODUk multiframe position 0000 0010, etc.).

The 256-byte FTFL message consists of two 128-byte fields as shown in Figure 15-20: the forward and backward fields. The forward field is allocated to bytes 0 to 127 of the FTFL message. The backward field is allocated to bytes 128 to 255 of the FTFL message.

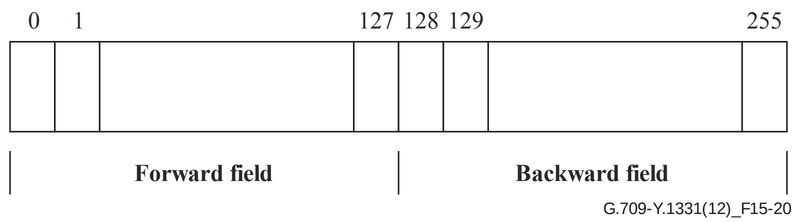


Figure 15-20 – FTFL message structure

The forward and backward fields are further divided into three subfields as shown in Figure 15-21: the forward/backward fault type indication field, the forward/backward operator identifier field, and the forward/backward operator-specific field.

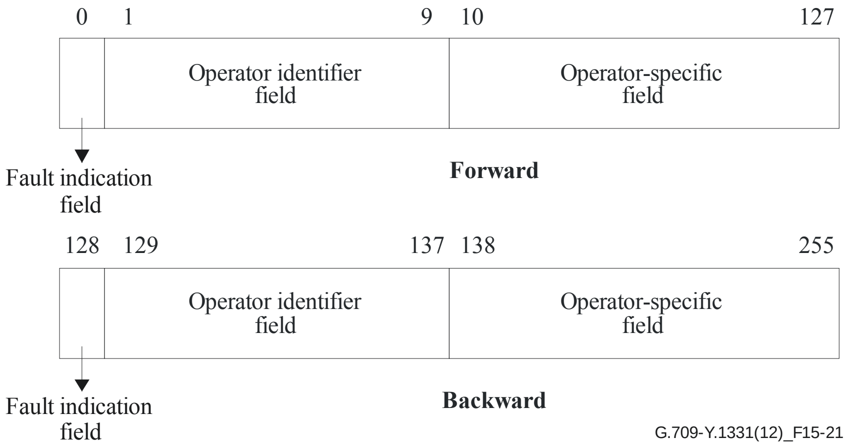


Figure 15-21 – Forward/backward field structure

15.8.2.5.1 Forward/backward fault type indication field

The fault type indication field provides the fault status. Byte 0 of the FTFL message is allocated for the forward fault type indication field. Byte 128 of the FTFL message is allocated for the backward fault type indication field. The fault type indication fields are coded as in Table 15-7. Code 0000 0000 shall indicate no fault, code 0000 0001 shall indicate signal fail, and code 0000 0010 shall indicate signal degrade. The remaining codes are reserved for future international standardization.

Table 15-7 – Fault indication codes

Fault indication code	Definition
0000 0000	No fault
0000 0001	Signal fail
0000 0010	Signal degrade
0000 0011 . . . 1111 1111	Reserved for future international standardization

15.8.2.5.2 Forward/backward operator identifier field

The operator identifier field is 9 bytes. Bytes 1 to 9 are allocated for the forward operator identifier field. Bytes 129 to 137 are allocated for the backward operator identifier field. The operator identifier field consists of two subfields: the international segment field, and the national segment field as shown in Figure 15-22.

<table>
<tr><td>Byte allocation in backward field</td><td>129</td><td>130</td><td>131</td><td>132</td><td>133</td><td>134</td><td>135</td><td>136</td><td>137</td></tr>
<tr><td>Byte allocation in forward field</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr>
<tr><td></td><td colspan="3">Country code</td><td colspan="6">National segment code</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td>ICC</td><td colspan="5">NUL padding</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td colspan="2">ICC</td><td colspan="4">NUL padding</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td colspan="3">ICC</td><td colspan="3">NUL padding</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td colspan="4">ICC</td><td colspan="2">NUL padding</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td colspan="5">ICC</td><td>NUL padding</td></tr>
<tr><td></td><td colspan="3">G/PCC</td><td colspan="6">ICC</td></tr>
<tr><td></td><td colspan="9">NUL</td></tr>
</table>

G.709-Y.1331(12)_F15-22

Figure 15-22 – Operator identifier field structure

The international segment field provides a three-character ISO 3166 geographic/political country code (G/PCC). The first three bytes of the 9-byte operator identifier field (i.e., bytes 1 to 3 for the forward operator identifier field and bytes 129 to 131 for the backward operator identifier field) are reserved for the international segment field. The country code shall be based on the three-character uppercase alphabetic ISO 3166 country code (e.g., USA, FRA).

The national segment field provides a 1-6 character ITU carrier code (ICC). The ICC is maintained by the ITU-T Telecommunication Standardization Bureau (TSB) as per [ITU-T M.1400]. The national segment field is 6 bytes and provides a 1-6 character ITU carrier code (ICC) with trailing null characters to complete the 6-character field.

15.8.2.5.3 Forward/backward operator-specific field

Bytes 10 to 127 are allocated for the forward operator-specific field as shown in Figure 15-21. Bytes 138 to 255 are allocated for the backward operator-specific field. The operator-specific fields are not subject to standardization.

15.8.2.6 ODUk experimental overhead (EXP)

Two bytes are allocated in the ODUk overhead for experimental use. These bytes are located in row 3, columns 13 and 14 of the ODUk overhead.

The use of these bytes is not subject to standardization and outside the scope of this Recommendation.

An experimental overhead is provided in the ODUk OH to allow a vendor and/or a network operator within their own (sub)network to support an application, which requires an additional ODUk overhead.

There is no requirement to forward the EXP overhead beyond the (sub)network; i.e., the operational span of the EXP overhead is limited to the (sub)network with the vendor's equipment, or the network of the operator.

15.8.2.7 ODUk reserved overhead (RES)

Eight bytes and one bit are reserved in the ODUk overhead for future international standardization. These bytes are located in row 2, columns 1 to 2 and row 4, columns 9 to 14 of the ODUk overhead. The bit is located in row 2, column 3, bit 8 of the ODUk overhead. These bytes and bit are set to all-0s.

15.9 OPUk OH description

15.9.1 OPUk OH location

The OPUk overhead consists of: payload structure identifier (PSI) including the payload type (PT), the overhead associated with concatenation and the overhead (e.g., justification control and opportunity bits) associated with the mapping of client signals into the OPUk payload. The OPUk PSI and PT overhead locations are shown in Figure 15-23.

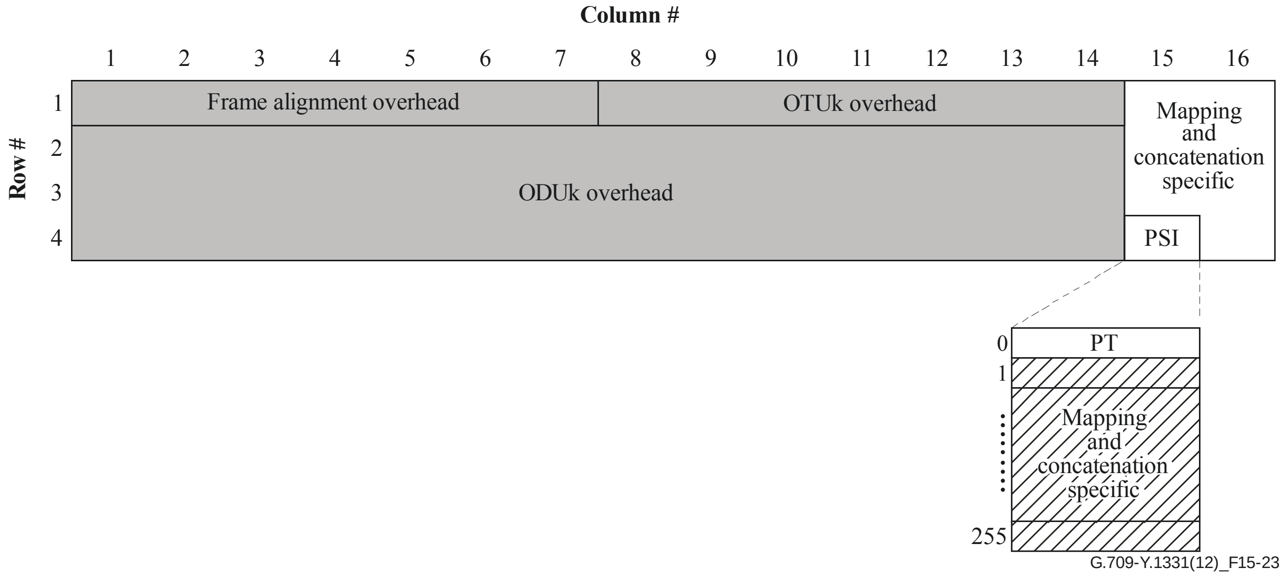


Figure 15-23 – OPUk overhead

15.9.2 OPUk OH definition

15.9.2.1 OPUk payload structure identifier (PSI)

One byte is allocated in the OPUk overhead to transport a 256-byte payload structure identifier (PSI) signal. The byte is located in row 4, column 15 of the OPUk overhead.

The 256-byte PSI signal is aligned with the ODUk multiframe (i.e., PSI[0] is present at ODUk multiframe position 0000 0000, PSI[1] at position 0000 0001, PSI[2] at position 0000 0010, etc.).

PSI[0] contains a one-byte payload type. PSI[1] to PSI[255] are mapping and concatenation specific, except for PT 0x01 (experimental mapping) and PTs 80-0x8F (for proprietary use).

15.9.2.1.1 OPUk payload type (PT)

A one-byte payload type signal is defined in the PSI[0] byte of the payload structure identifier to indicate the composition of the OPUk signal. The code points are defined in Table 15-8.

Table 15-8 – Payload type code points

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 0 0	0 0 0 1	01	Experimental mapping (Note 3)
0 0 0 0	0 0 1 0	02	Asynchronous CBR mapping, see clause 17.2
0 0 0 0	0 0 1 1	03	Bit-synchronous CBR mapping, see clause 17.2
0 0 0 0	0 1 0 0	04	ATM mapping, see clause 17.3
0 0 0 0	0 1 0 1	05	GFP mapping, see clause 17.4
0 0 0 0	0 1 1 0	06	Virtual concatenated signal, see clause 18 (Note 5)
0 0 0 0	0 1 1 1	07	PCS codeword transparent Ethernet mapping: 1000BASE-X into OPU0, see clauses 17.7.1 and 17.7.1.1 40GBASE-R into OPU3, see clauses 17.7.4 and 17.7.4.1 100GBASE-R into OPU4, see clauses 17.7.5 and 17.7.5.1
0 0 0 0	1 0 0 0	08	FC-1200 into OPU2e mapping, see clause 17.8.2
0 0 0 0	1 0 0 1	09	GFP mapping into extended OPU2 payload, see clause 17.4.1 (Note 6)
0 0 0 0	1 0 1 0	0A	STM-1 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 0 1 1	0B	STM-4 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 1 0 0	0C	FC-100 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 1 0 1	0D	FC-200 mapping into OPU1, see clause 17.7.2
0 0 0 0	1 1 1 0	0E	FC-400 mapping into OPUflex, see clause 17.9
0 0 0 0	1 1 1 1	0F	FC-800 mapping into OPUflex, see clause 17.9
0 0 0 1	0 0 0 0	10	Bit stream with octet timing mapping, see clause 17.6.1
0 0 0 1	0 0 0 1	11	Bit stream without octet timing mapping, see clause 17.6.2
0 0 0 1	0 0 1 0	12	IB SDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 0 1 1	13	IB DDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 1 0 0	14	IB QDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 1 0 1	15	SDI mapping into OPU0, see clause 17.7.1
0 0 0 1	0 1 1 0	16	(1.485/1.001) Gbit/s SDI mapping into OPU1, see clause 17.7.2
0 0 0 1	0 1 1 1	17	1.485 Gbit/s SDI mapping into OPU1, see clause 17.7.2
0 0 0 1	1 0 0 0	18	(2.970/1.001) Gbit/s SDI mapping into OPUflex, see clause 17.9
0 0 0 1	1 0 0 1	19	2.970 Gbit/s SDI mapping into OPUflex, see clause 17.9
0 0 0 1	1 0 1 0	1A	SBCON/ESCON mapping into OPU0, see clause 17.7.1
0 0 0 1	1 0 1 1	1B	DVB_ASI mapping into OPU0, see clause 17.7.1
0 0 0 1	1 1 0 0	1C	FC-1600 mapping into OPUflex, see clause 17.9
0 0 1 0	0 0 0 0	20	ODU multiplex structure supporting ODTUjk only, see clause 19 (AMP only)
0 0 1 0	0 0 0 1	21	ODU multiplex structure supporting ODTUk.ts or ODTUk.ts and ODTUjk, see clause 19 (GMP capable) (Note 7)
0 1 0 1	0 1 0 1	55	Not available (Note 2)

Table 15-8 – Payload type code points

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 1 1 0	0 1 1 0	66	Not available (Note 2)
1 0 0 0	x x x x	80-8F	Reserved codes for proprietary use (Note 4)
1 1 1 1	1 1 0 1	FD	NULL test signal mapping, see clause 17.5.1
1 1 1 1	1 1 1 0	FE	PRBS test signal mapping, see clause 17.5.2
1 1 1 1	1 1 1 1	FF	Not available (Note 2)

NOTE 1 – There are 205 spare codes left for future international standardization. Refer to Annex A of [ITU-T G.806] for the procedure to obtain one of these codes for a new payload type.

NOTE 2 – These values are excluded from the set of available code points. These bit patterns are present in ODUk maintenance signals.

NOTE 3 – Value "01" is only to be used for experimental activities in cases where a mapping code is not defined in this table. Refer to Annex A of [ITU-T G.806] for more information on the use of this code.

NOTE 4 – These 16 code values will not be subject to further standardization. Refer to Annex A of [ITU-T G.806] for more information on the use of these codes.

Table 15-8 – Payload type code points

NOTE 5 – For the payload type of the virtual concatenated signal a dedicated payload type overhead (vcPT) is used, see clause 18.

NOTE 6 – Supplement 43 (2008) to the ITU-T G-series of Recommendations indicated that this mapping recommended using payload type 87.

NOTE 7 – Equipment supporting ODTUk.ts for OPU2 or OPU3 must be backward compatible with equipment which supports only the ODTUjk. ODTUk.ts capable equipment transmitting PT=21 which receives PT=20 from the far end shall revert to PT=20 and operate in ODTUjk only mode. Refer to [ITU-T G.798] for the specification.

15.9.2.2 OPUk mapping specific overhead

Seven bytes are reserved in the OPUk overhead for the mapping and concatenation specific overhead. These bytes are located in rows 1 to 3, columns 15 and 16 and column 16 row 4. In addition, 255 bytes in the PSI are reserved for mapping and concatenation specific purposes.

The use of these bytes depends on the specific client signal mapping (defined in clauses 17 and 19), the use of concatenation (see clause 18) and use of hitless adjustment of ODUflex(GFP) (see [ITU-T G.7044]).

16 Maintenance signals

An alarm indication signal (AIS) is a signal sent downstream as an indication that an upstream defect has been detected. An AIS signal is generated in an adaptation sink function. An AIS signal is detected in a trail termination sink function to suppress defects or failures that would otherwise be detected as a consequence of the interruption of the transport of the original signal at an upstream point.

A forward defect indication (FDI) is a signal sent downstream as an indication that an upstream defect has been detected. An FDI signal is generated in an adaptation sink function. An FDI signal is detected in a trail termination sink function to suppress defects or failures that would otherwise be detected as a consequence of the interruption of the transport of the original signal at an upstream point.

NOTE – AIS and FDI are similar signals. AIS is used as the term when the signal is in the digital domain. FDI is used as the term when the signal is in the optical domain; FDI is transported as a non-associated overhead in the OTM overhead signal (OOS).

An open connection indication (OCI) is a signal sent downstream as an indication that upstream the signal is not connected to a trail termination source. An OCI signal is generated in a connection function and output by this connection function on each of its output connection points, which are not connected to one of its input connection points. An OCI signal is detected in a trail termination sink function.

A locked (LCK) is a signal sent downstream as an indication that upstream the connection is "locked", and no signal has passed through.

A payload missing indication (PMI) is a signal sent downstream as an indication that upstream at the source point of the signal, either none of the tributary slots have an optical signal or an optical signal with no payload. This indicates that the transport of the optical tributary signal is interrupted.

A PMI signal is generated in the adaptation source function and it is detected in the trail termination sink function which suppresses the LOS defect that arises under this condition.

16.1 OTS maintenance signals

16.1.1 OTS payload missing indication (OTS-PMI)

OTS-PMI is generated as an indication that the OTS payload does not contain an optical signal.

16.2 OMS maintenance signals

Three OMS maintenance signals are defined: OMS-FDI-P, OMS-FDI-O and OMS-PMI.

16.2.1 OMS forward defect indication – Payload (OMS-FDI-P)

OMS-FDI-P is generated as an indication of an OMS server layer defect in the OTS network layer.

16.2.2 OMS forward defect indication – Overhead (OMS-FDI-O)

OMS-FDI-O is generated as an indication when the transport of OMS OH via the OOS is interrupted due to a signal fail condition in the OOS.

16.2.3 OMS payload missing indication (OMS-PMI)

OMS-PMI is generated as an indication when none of the OCCs contain an optical signal.

16.3 OCh maintenance signals

Three OCh maintenance signals are defined: OCh-FDI-P, OCh-FDI-O and OCh-OCI.

16.3.1 OCh forward defect indication – Payload (OCh-FDI-P)

OCh-FDI is generated as an indication for an OCh server layer defect in the OMS network layer.

When the OTUk is terminated, the OCh-FDI is continued as an ODUk-AIS signal.

16.3.2 OCh forward defect indication – Overhead (OCh-FDI-O)

OCh-FDI-O is generated as an indication when the transport of OCh OH via the OOS is interrupted due to a signal fail condition in the OOS.

16.3.3 OCh open connection indication (OCh-OCI)

The OCh-OCI signal indicates to downstream transport processing functions that the OCh connection is not bound to, or not connected (via a matrix connection) to a termination source function. The indication is used in order to distinguish downstream between a missing optical channel due to a defect or due to the open connection (resulting from a management command).

NOTE – OCI is detected at the next downstream OTUk trail terminating equipment. If the connection was opened intentionally, the related alarm report from this trail termination should be disabled by using the alarm reporting control mode (refer to [ITU-T M.3100]).

16.4 OTUk maintenance signals

16.4.1 OTUk alarm indication signal (OTUk-AIS)

The OTUk-AIS (see Figure 16-1) is a generic-AIS signal (see clause 16.6.1). Since the OTUk capacity (130 560 bits) is not an integer multiple of the PN-11 sequence length (2047 bits), the PN-11 sequence may cross an OTUk frame boundary.

NOTE – OTUk-AIS is defined to support a future server layer application. OTN equipment should be capable of detecting the presence of such a signal within OTM-0.1, OTM-0.2, OTM-0.3 interface signals and within OChr interface signals (in an OTM-nr.m interface) carrying an OTU1, OTU2 or OTU3; it is not required to generate such a signal.

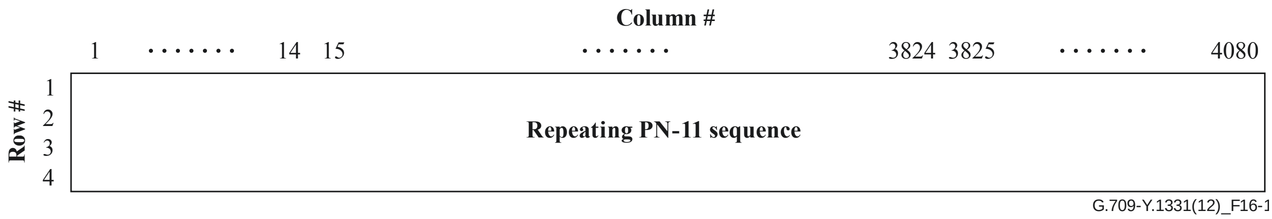


Figure 16-1 – OTUk-AIS

16.5 ODUk maintenance signals

Three ODUk maintenance signals are defined: ODUk-AIS, ODUk-OCI and ODUk-LCK.

16.5.1 ODUk alarm indication signal (ODUk-AIS)

ODUk-AIS is specified as all "1"s in the entire ODUk signal, excluding the frame alignment overhead (FA OH), OTUk overhead (OTUk OH) and ODUk FTFL (see Figure 16-2).

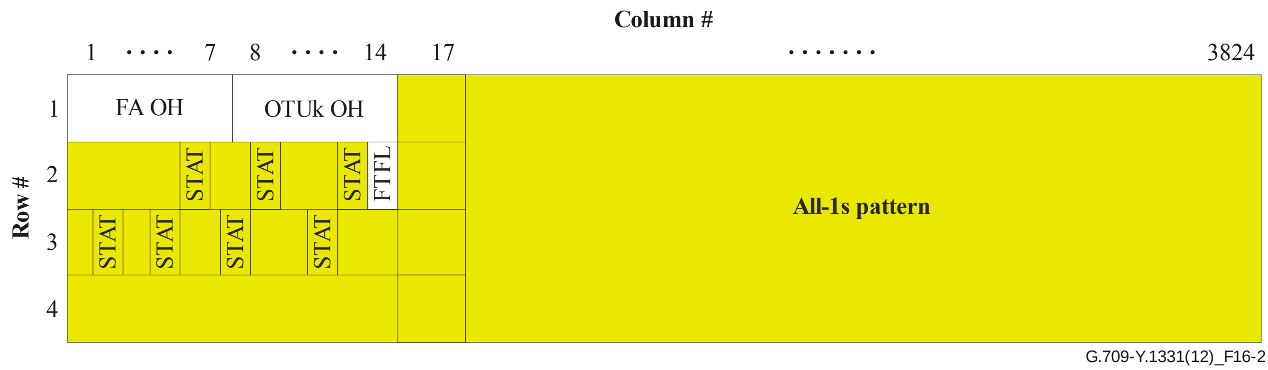


Figure 16-2 – ODUk-AIS

In addition, the ODUk-AIS signal may be extended with one or more levels of ODUk tandem connection, GCC1, GCC2, EXP and/or APS/PCC overhead before it is presented at the OTM interface. This is dependent on the functionality between the ODUk-AIS insertion point and the OTM interface.

The presence of the ODUk-AIS is detected by monitoring the ODUk STAT bits in the PM and TCMi overhead fields.

16.5.2 ODUk open connection indication (ODUk-OCI)

ODUk-OCI is specified as a repeating "0110 0110" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH) (see Figure 16-3).

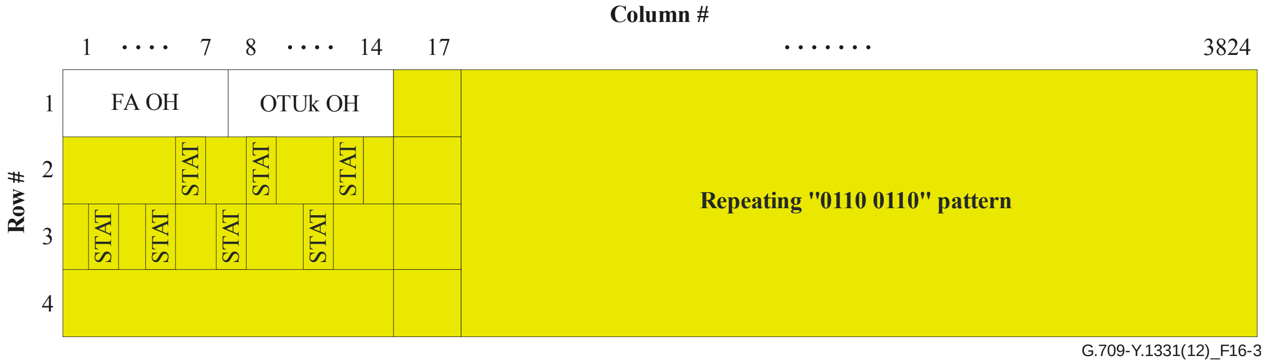


Figure 16-3 – ODUk-OCI

NOTE – The repeating "0110 0110" pattern is the default pattern; other patterns are also allowed as long as the STAT bits in the PM and TCMi overhead fields are set to "110".

In addition, the ODUk-OCI signal may be extended with one or more levels of ODUk tandem connection, GCC1, GCC2, EXP and/or APS/PCC overhead before it is presented at the OTM interface. This is dependent on the functionality between the ODUk-OCI insertion point and the OTM interface.

The presence of ODUk-OCI is detected by monitoring the ODUk STAT bits in the PM and TCMi overhead fields.

16.5.3 ODUk locked (ODUk-LCK)

ODUk-LCK is specified as a repeating "0101 0101" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH) (see Figure 16-4).

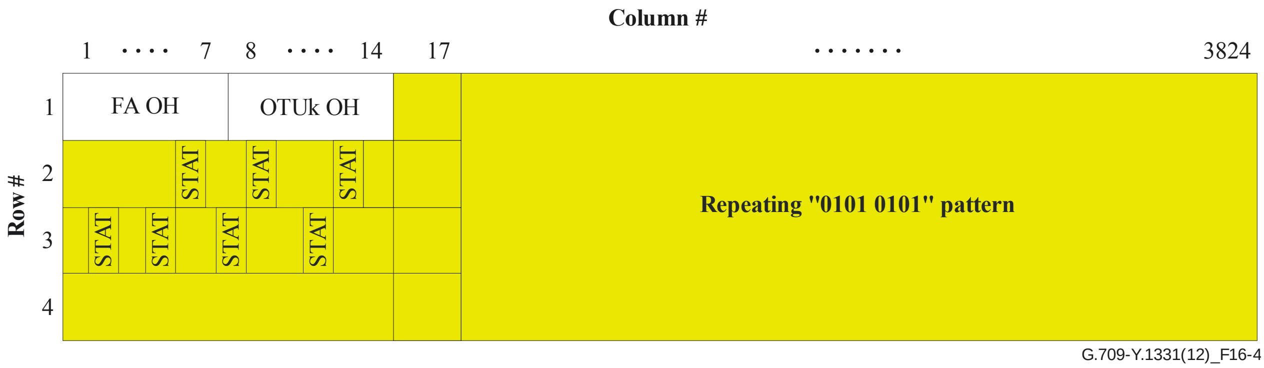


Figure 16-4 – ODUk-LCK

NOTE – The repeating "0101 0101" pattern is the default pattern; other patterns are also allowed as long as the STAT bits in the PM and TCMi overhead fields are set to "101".

In addition, the ODUk-LCK signal may be extended with one or more additional levels of ODUk tandem connection, GCC1, GCC2, EXP and/or the APS/PCC overhead before it is presented at the OTM interface. This is dependent on the functionality between the ODUk-LCK insertion point and the OTM interface.

The presence of ODUk-LCK is detected by monitoring the ODUk STAT bits in the PM and TCMi overhead fields.

16.6 Client maintenance signal

16.6.1 Generic AIS for constant bit rate signals

The generic-AIS signal is a signal with a 2 047-bit polynomial number 11 (PN-11) repeating sequence.

The PN-11 sequence is defined by the generating polynomial $1 + x^9 + x^{11}$ as specified in clause 5.2 of [ITU-T O.150]. (See Figure 16-5.)

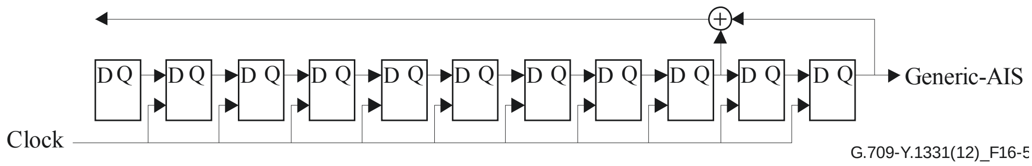


Figure 16-5 – Generic-AIS generating circuit

17 Mapping of client signals

This clause specifies the mapping of:

– STM-16, STM-64, STM-256 constant bit rate client signals into OPUk using client/server specific asynchronous or bit-synchronous mapping procedures (AMP, BMP);

– 10GBASE-R constant bit rate client signal into OPU2e using client/server specific bit-synchronous mapping procedure (BMP);

– FC-1200 constant bit rate client signal after timing transparent transcoding (TTT) providing a 50/51 rate compression into OPU2e using client/server specific byte-synchronous mapping procedure;

– constant bit rate client signals with bit rates up to 1.238 Gbit/s into OPU0 and up to 2.488 Gbit/s into OPU1 using a client agnostic generic mapping procedure (GMP) possibly preceded by a timing transparent transcoding (TTT) of the client signal to reduce the bit rate of the signal to fit the OPUk payload bandwidth;

– constant bit rate client signals into OPU1, OPU2, OPU3 or OPU4 respectively using a client agnostic generic mapping procedure (GMP) possibly preceded by a timing transparent transcoding (TTT) of the client signal to reduce the bit rate of the signal to fit the OPUk payload bandwidth;

– other constant bit rate client signals into OPUflex using a client agnostic bit-synchronous mapping procedure (BMP);

– asynchronous transfer mode (ATM);

– packet streams (e.g., Ethernet, MPLS, IP) which are encapsulated with the generic framing procedure (GFP-F);

– test signals;

– continuous mode GPON constant bit rate client signal into OPU1 using asynchronous mapping procedure (AMP);

– continuous mode XGPON constant bit rate client signal into OPU2 using asynchronous mapping procedure (AMP);

into OPUk.

17.1 OPUk client signal fail (CSF)

For support of local management systems, a single-bit OPUk client signal fail (CSF) indicator is defined to convey the signal fail status of the CBR and Ethernet private line client signal mapped into an LO OPUk at the ingress of the OTN to the egress of the OTN.

OPUk CSF is located in bit 1 of the PSI[2] byte of the payload structure identifier. Bits 2 to 8 of the PSI[2] byte are reserved for future international standardization. These bits are set to all-0s.

OPUk CSF is set to "1" to indicate a client signal fail indication, otherwise it is set to "0".

NOTE – Equipment designed prior to this revision of the Recommendation will generate a "0" in the OPUk CSF and will ignore any value in OPUk CSF.

17.2 Mapping of CBR2G5, CBR10G, CBR10G3 and CBR40G signals into OPUk

The mapping of a CBR2G5, CBR10G or CBR40G signal (with up to ±20 ppm bit-rate tolerance) into an OPUk (k = 1,2,3) may be performed according to the bit-synchronous mapping procedure based on one generic OPUk frame structure (see Figure 17-1). The mapping of a CBR2G5, CBR10G or CBR40G signal (with up to ±45 ppm bit-rate tolerance) into an OPUk (k = 1,2,3) may be performed according to the asynchronous mapping procedure. The mapping of a CBR10G3 signal (with up to ±100 ppm bit-rate tolerance) into an OPUk (k = 2e) is performed using the bit-synchronous mapping procedure.

NOTE 1 – Examples of CBR2G5, CBR10G and CBR40G signals are STM-16 and CMGPON_D/U2 (refer to [ITU-T G.984.6]), STM-64 and CMXGPON_D/U2 [ITU-T G.987.4] and STM-256. An example of a CBR10G3 signal is 10GBASE-R.

NOTE 2 – The maximum bit-rate tolerance between an OPUk and the client signal clock, which can be accommodated by the asynchronous mapping scheme, is ±65 ppm. With a bit-rate tolerance of ±20 ppm for the OPUk clock, the client signal's bit-rate tolerance can be ±45 ppm.

NOTE 3 – For OPUk (k=1,2,3) the clock tolerance is ±20 ppm. For OPU2e the clock tolerance is ±100 ppm and asynchronous mapping cannot be supported with this justification overhead.

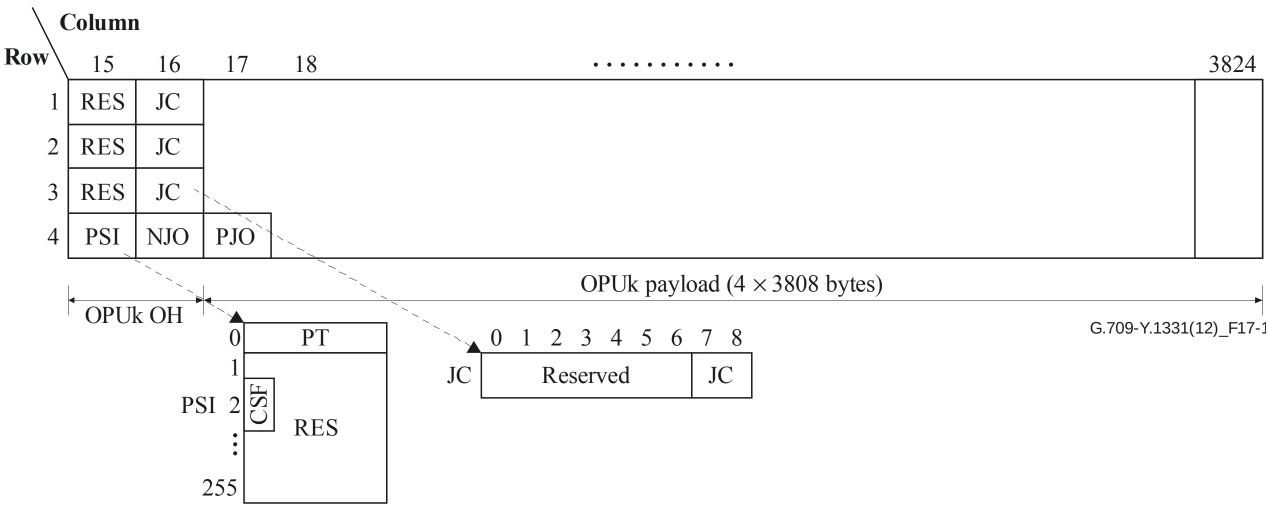


Figure 17-1 – OPUk frame structure for the mapping of a CBR2G5, CBR10G or CBR40G signal

The OPUk overhead for these mappings consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits reserved for future international standardization (RES), three justification control (JC) bytes, one negative justification opportunity (NJO) byte, and three bytes reserved for future international standardization (RES). The JC bytes consist of two bits for justification control and six bits reserved for future international standardization.

The OPUk payload for these mappings consists of 4 × 3808 bytes, including one positive justification opportunity (PJO) byte.

The justification control (JC) signal, which is located in rows 1, 2 and 3 of column 16, bits 7 and 8, is used to control the two justification opportunity bytes NJO and PJO that follow in row 4.

The asynchronous and bit-synchronous mapping processes generate the JC, NJO and PJO according to Tables 17-1 and 17-2, respectively. The de-mapping process interprets JC, NJO and PJO according to Table 17-3. Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

Table 17-1 – JC, NJO and PJO generation by an asynchronous mapping process

JC bits 7 8	NJO	PJO
0 0	justification byte	data byte
0 1	data byte	data byte
1 0	not generated	
1 1	justification byte	justification byte

Table 17-2 – JC, NJO and PJO generation by a bit-synchronous mapping process

JC bits 7 8	NJO	PJO
0 0	justification byte	data byte
0 1	not generated	
1 0		
1 1		

Table 17-3 – JC, NJO and PJO interpretation

JC bits 7 8	NJO	PJO
0 0	justification byte	data byte
0 1	data byte	data byte
1 0 (Note)	justification byte	data byte
1 1	justification byte	justification byte
NOTE – A mapper circuit does not generate this code. Due to bit errors a de-mapper circuit might receive this code.		

The value contained in NJO and PJO when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming CBR2G5, CBR10G or CBR40G client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the generic-AIS signal as specified in clause 16.6.1, and is then mapped into the OPUk.

During a signal fail condition of the incoming 10GBASE-R type CBR10G3 client signal (e.g., in the case of a loss of input signal), this failed incoming 10GBASE-R signal is replaced by a stream of 66B blocks, with each block carrying two local fault sequence ordered sets (as specified in [IEEE 802.3]). This replacement signal is then mapped into the OPU2e.

During the signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the generic-AIS pattern as specified in clause 16.6.1 is generated as a replacement signal for the lost CBR2G5, CBR10G or CBR40G signal.

During the signal fail condition of the incoming ODU2e/OPU2e signal (e.g., in the case of an ODU2e-AIS, ODU2e-LCK, ODU2e-OCI condition) a stream of 66B blocks, with each block carrying two local fault sequence ordered sets (as specified in [IEEE 802.3]) is generated as a replacement signal for the lost 10GBASE-R signal.

NOTE 4 – Local fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH=10 0x55 00 00 01 00 00 00 01.

NOTE 5 – Equipment developed prior to the 2008 version of this Recommendation may generate a different 10GBASE-R replacement signal (e.g., Generic-AIS) than the local fault sequence ordered set.

Asynchronous mapping

The OPUk signal for the asynchronous mapping is created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the CBR2G5, CBR10G or CBR40G (i.e., $4^{(k-1)} \times 2\,488\,320$ kbit/s (k = 1,2,3)) client signal.

The CBR2G5, CBR10G, CBR40G (i.e., $4^{(k-1)} \times 2\,488\,320$ kbit/s (k = 1,2,3)) signal is mapped into the OPUk using a positive/negative/zero (pnz) justification scheme.

Bit-synchronous mapping

The OPUk clock for bit-synchronous mapping is derived from the CBR2G5, CBR10G, CBR40G or CBR10G3 client signal. During signal fail conditions of the incoming CBR2G5, CBR10G, CBR40G or CBR10G3 signal (e.g., in the case of a loss of input signal), the OPUk payload signal bit rate shall be within the limits specified in Table 7-3 and neither a frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming CBR2G5, CBR10G, CBR40G or CBR10G3 signal shall be done without introducing a frequency or frame phase discontinuity.

The CBR2G5, CBR10G, CBR40G or CBR10G3 signal is mapped into the OPUk without using the justification capability within the OPUk frame: NJO contains a justification byte, PJO contains a data byte, and the JC signal is fixed to 00.

17.2.1 Mapping a CBR2G5 signal (e.g., STM-16, CMGPON_D/CMGPON_U2) into OPU1

Groups of eight successive bits (not necessarily being a byte) of the CBR2G5 signal are mapped into a data (D) byte of the OPU1 (see Figure 17-2). Once per OPU1 frame, it is possible to perform either a positive or a negative justification action.

Column #

Row #	15	16	17	18	....	3824
1	RES	JC	D	D	3805D	D
2	RES	JC	D	D	3805D	D
3	RES	JC	D	D	3805D	D
4	PSI	NJO	PJO	D	3805D	D

G.709-Y.1331(12)_F17-2

Figure 17-2 – Mapping of a CBR2G5 signal into OPU1

17.2.2 Mapping a CBR10G signal (e.g., STM-64, CMXGPON_D/CMXGPON_U2) into OPU2

Groups of eight successive bits (not necessarily being a byte) of the CBR10G signal are mapped into a data (D) byte of the OPU2 (see Figure 17-3). 64 fixed stuff (FS) bytes are added in columns 1905 to 1920. Once per OPU2 frame, it is possible to perform either a positive or a negative justification action.

Column #

Row #	15	16	17	····· 1904	1905 ····· 1920	1921 ····· 3824
1	RES	JC	118 × 16D		16FS	119 × 16D
2	RES	JC	118 × 16D		16FS	119 × 16D
3	RES	JC	118 × 16D		16FS	119 × 16D
4	PSI	NJO	PJO	15D + 117 × 16D	16FS	119 × 16D

G.709-Y.1331(12)_F17-3

Figure 17-3 – Mapping of a CBR10G signal into OPU2

17.2.3 Mapping a CBR40G signal (e.g., STM-256) into OPU3

Groups of eight successive bits (not necessarily being a byte) of the CBR40G signal are mapped into a data (D) byte of the OPU3 (see Figure 17-4). 128 fixed stuff (FS) bytes are added in columns 1265 to 1280 and 2545 to 2560. Once per OPU3 frame, it is possible to perform either a positive or a negative justification action.

Column #

Row #	15	16	17	····· 1264	1265 ··· 1280	1281 ····· 2544	2545 ··· 2560	2561 ··· 3824
1	RES	JC	78 × 16D		16FS	79 × 16D	16FS	79 × 16D
2	RES	JC	78 × 16D		16FS	79 × 16D	16FS	79 × 16D
3	RES	JC	78 × 16D		16FS	79 × 16D	16FS	79 × 16D
4	PSI	NJO	PJO	15D + 77 × 16D	16FS	79 × 16D	16FS	79 × 16D

G.709-Y.1331(12)_F17-4

Figure 17-4 – Mapping of a CBR40G signal into OPU3

17.2.4 Mapping a CBR10G3 signal (e.g., 10GBASE-R) into OPU2e

Groups of eight successive bits (not necessarily being a byte) of the CBR10G3 signal are bit-synchronously mapped into a data (D) byte of the OPU2e (see Figure 17-5). 64 fixed stuff (FS) bytes are added in columns 1905 to 1920.

NOTE – The NJO byte will always carry a stuff byte, the PJO byte will always carry a data (D) byte and the JC bytes will always carry the all-0s pattern.

Column #

Row #	15	16	17	····· 1904	1905 ····· 1920	1921 ····· 3824
1	RES	JC	118 × 16D		16FS	119 × 16D
2	RES	JC	118 × 16D		16FS	119 × 16D
3	RES	JC	118 × 16D		16FS	119 × 16D
4	PSI	NJO	PJO	15D + 117 × 16D	16FS	119 × 16D

G.709-Y.1331(12)_F17-5

Figure 17-5 – Mapping of a CBR10G3 signal into OPU2e

17.3 Mapping of ATM cell stream into OPUk (k=0,1,2,3)

A constant bit rate ATM cell stream with a capacity that is identical to the OPUk (k=0,1,2,3) payload area is created by multiplexing the ATM cells of a set of ATM VP signals. Rate adaptation is performed as part of this cell stream creation process by either inserting idle cells or by discarding cells. Refer to [ITU-T I.432.1]. The ATM cell stream is mapped into the OPUk payload area with the ATM cell byte structure aligned to the ODUk payload byte structure (see Figure 17-6). The ATM cell boundaries are thus aligned with the OPUk payload byte boundaries. Since the OPUk payload capacity (15232 bytes) is not an integer multiple of the cell length (53 bytes), a cell may cross an OPUk frame boundary.

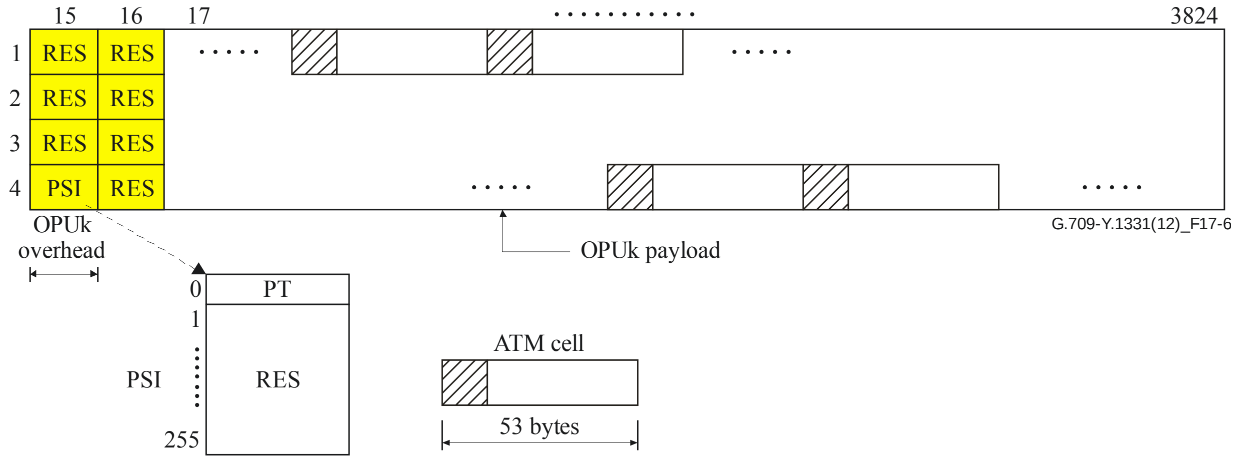


Figure 17-6 – OPUk frame structure and mapping of ATM cells into OPUk

The ATM cell information field (48 bytes) shall be scrambled before mapping into the OPUk. In the reverse operation, following termination of the OPUk signal, the ATM cell information field will be descrambled before being passed to the ATM layer. A self-synchronizing scrambler with generator polynomial x43 + 1 shall be used (as specified in [ITU-T I.432.1]). The scrambler operates for the duration of the cell information field. During the 5-byte header the scrambler operation is suspended and the scrambler state retained. The first cell transmitted on start-up will be corrupted because the descrambler at the receiving end will not be synchronized to the transmitter scrambler. Cell information field scrambling is required to provide security against false cell delineation and the cell information field replicating the OTUk and ODUk frame alignment signal.

When extracting the ATM cell stream from the OPUk payload area after the ODUk termination, the ATM cells must be recovered. The ATM cell header contains a header error control (HEC) field, which may be used in a similar way to a frame alignment word to achieve cell delineation. This HEC method uses the correlation between the header bits to be protected by the HEC (32 bits) and the control bit of the HEC (8 bits) introduced in the header after computation with a shortened cyclic code with generating polynomial $g(x) = x^8 + x^2 + x + 1$.

The remainder from this polynomial is then added to the fixed pattern "01010101" in order to improve the cell delineation performance. This method is similar to conventional frame alignment recovery where the alignment signal is not fixed but varies from cell to cell.

More information on HEC cell delineation is given in [ITU-T I.432.1].

The OPUk overhead for the ATM mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES).

The OPUk payload for the ATM mapping consists of 4 × 3808 bytes.

17.4 Mapping of GFP frames into OPUk (k=0,1,2,3,4,flex)

The mapping of generic framing procedure (GFP) frames is performed by aligning the byte structure of every GFP frame with the byte structure of the OPUk payload (see Figure 17-7). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a frame may cross the OPUk (k=0,1,2,3,4,flex) frame boundary.

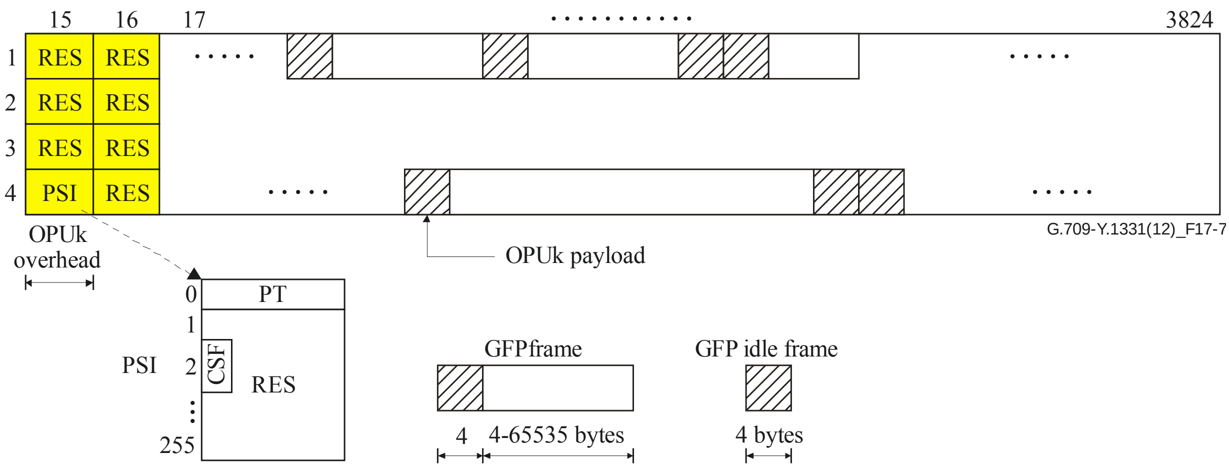


Figure 17-7 – OPUk frame structure and mapping of GFP frames into OPUk

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPUk payload area, due to the insertion of idle frames at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE 1 – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPUk overhead for the GFP mapping consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES). The CSF indicator should be used only for Ethernet private line type 1 services; for other packet clients the CSF bit is fixed to 0.

The OPUk payload for the GFP mapping consists of 4 × 3808 bytes.

NOTE 2 – The OPUflex(GFP) bit rate may be any configured bit rate as specified in Tables 7-3 and 7-8.

17.4.1 Mapping of GFP frames into an extended OPU2 payload area

The mapping of generic framing procedure (GFP) frames in an extended OPU2 payload area is performed by aligning the byte structure of every GFP frame with the byte structure of the extended OPU2 payload (see Figure 17-8). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a frame may cross the OPU2 frame boundary.

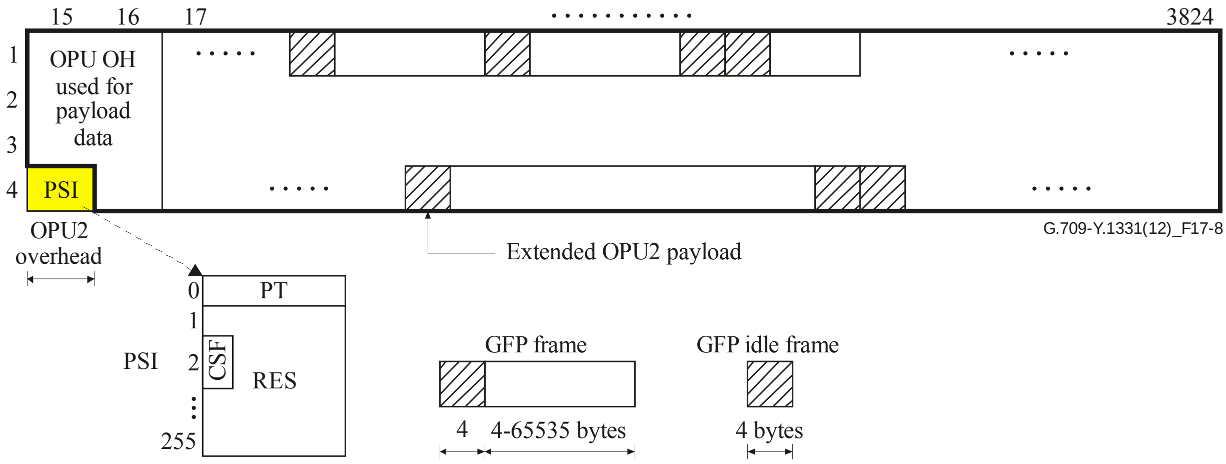


Figure 17-8 – OPU2 frame structure and mapping of GFP frames into an extended OPU2 payload area

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPU2 payload area, due to the insertion of GFP-idle frames at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPU2 overhead for the GFP mapping consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits of reserved for future international standardization (RES).

The extended OPU2 payload for the GFP mapping consists of 4 × 3808 bytes from the OPU2 payload plus 7 bytes from the OPU2 overhead.

17.5 Mapping of test signal into OPUk

17.5.1 Mapping of a NULL client into OPUk

An OPUk payload signal with an all-0s pattern (see Figure 17-9) is defined for test purposes. This is referred to as the NULL client.

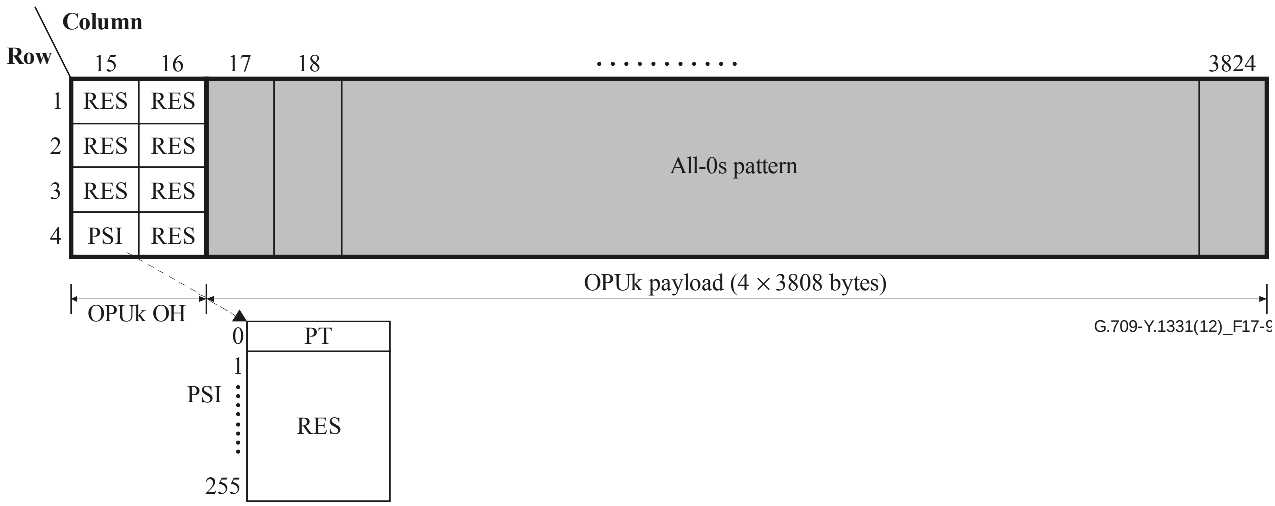


Figure 17-9 – OPUk frame structure and mapping of a NULL client into OPUk

The OPUk overhead for the NULL mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES).

The OPUk payload for the NULL mapping consists of 4 × 3808 bytes.

17.5.2 Mapping of PRBS test signal into OPUk

For test purposes, a 2 147 483 647-bit pseudo-random test sequence ($2^{31} - 1$) as specified in clause 5.8 of [ITU-T O.150] can be mapped into the OPUk payload. Groups of eight successive bits of the 2 147 483 647-bit pseudo-random test sequence signal are mapped into 8 data bits (8D) (i.e., one byte) of the OPUk payload (see Figure 17-10).

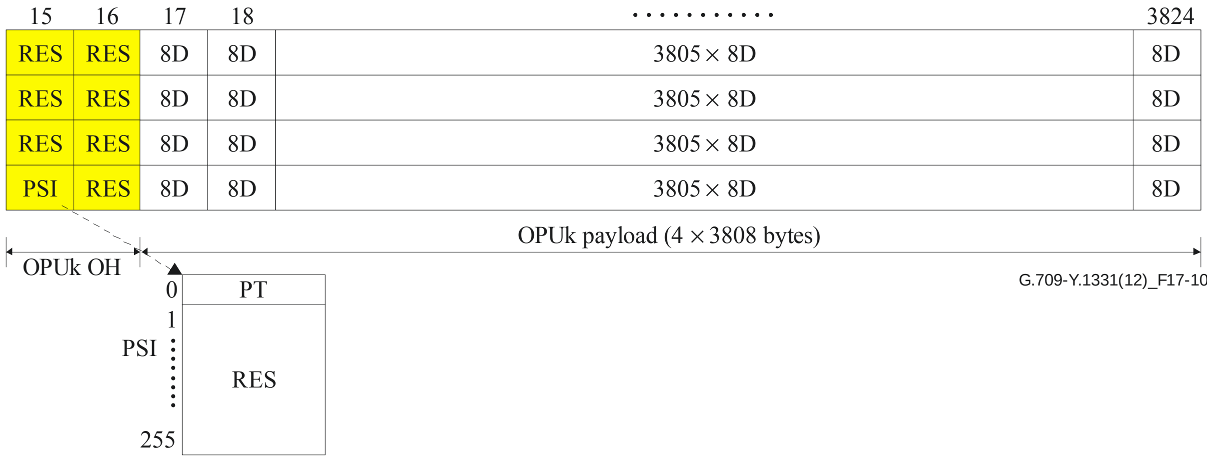


Figure 17-10 – OPUk frame structure and mapping of 2 147 483 647-bit pseudo-random test sequence into OPUk

The OPUk overhead for the PRBS mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES).

The OPUk payload for the PRBS mapping consists of 4 × 3808 bytes.

17.6 Mapping of a non-specific client bit stream into OPUk

In addition to the mappings of specific client signals as specified in the other subclauses of this clause, a non-specific client mapping into OPUk is specified. Any (set of) client signal(s), which after encapsulation into a continuous bit stream with a bit rate of the OPUk payload, can be mapped into the OPUk payload (see Figure 17-11). The bit stream must be synchronous with the OPUk signal. Any justification must be included in the continuous bit stream creation process. The continuous bit stream must be scrambled before mapping into the OPUk payload.

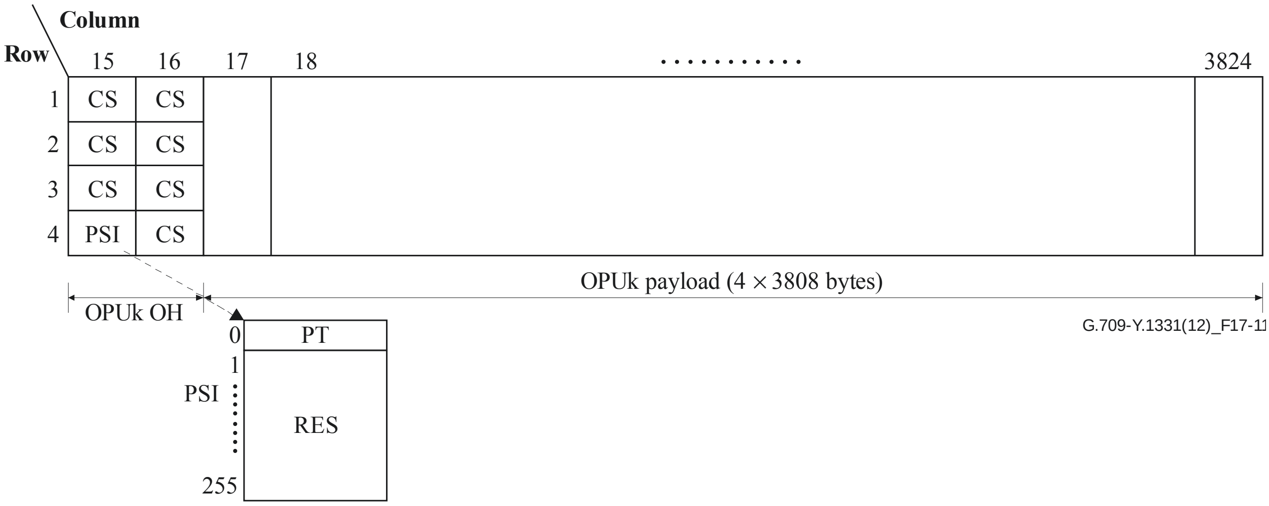


Figure 17-11 – OPUk frame structure for the mapping of a synchronous constant bit stream

The OPUk overhead for the mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes for client-specific (CS) purposes. The definition of these CS overhead bytes is performed within the encapsulation process specification.

The OPUk payload for this non-specific mapping consists of 4 × 3808 bytes.

17.6.1 Mapping bit stream with octet timing into OPUk

If octet timing is available, each octet of the incoming data stream will be mapped into a data byte (octet) of the OPUk payload.

17.6.2 Mapping bit stream without octet timing into OPUk

If octet timing is not available, groups of eight successive bits (not necessarily an octet) of the incoming data stream will be mapped into a data byte (octet) of the OPUk payload.

17.7 Mapping of other constant bit-rate signals with justification into OPUk

Mapping of other CBR client signals (with up to ±100 ppm bit-rate tolerance) into an OPUk (k = 0, 1, 2, 3, 4) is performed by the generic mapping procedure as specified in Annex D.

During a signal fail condition of the incoming CBR client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the appropriate replacement signal as defined in the clauses hereafter.

During a signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition), the failed client signal is replaced by the appropriate replacement signal as defined in the clauses hereafter.

The OPUk overhead for this mapping consists of a:

– payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-8, the client signal fail (CSF) and 254 bytes plus 7 bits reserved for future international standardization (RES);
– three justification control (JC1, JC2, JC3) bytes carrying the value of GMP overhead Cm;
– three justification control (JC4, JC5, JC6) bytes carrying the value of GMP overhead ΣCnD and
– one byte reserved for future international standardization (RES).

The JC1, JC2 and JC3 bytes consist of a 14-bit C_m field (bits C1, C2, .., C14), a 1-bit Increment Indicator (II) field, a 1-bit Decrement Indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The JC4, JC5 and JC6 bytes consist of a 10-bit ΣC_{nD} field (bits D1, D2, .., D10), a 5-bit CRC-5 field which contains an error check code over the bits 4 to 8 in the JC4, JC5 and JC6 fields and nine bits reserved for future international standardization (RES). The default value of n in ΣC_{nD} is 8. The support for n=1 is client dependent and specified in the clauses hereafter when required.

17.7.1 Mapping a sub-1.238 Gbit/s CBR client signal into OPU0

Table 17-4A specifies the clients defined by this Recommendation and their GMP c_m and C_m with m=8 (c_8, C_8) minimum, nominal and maximum parameter values. Table 17-4B specifies the GMP c_n and C_n with n=8 (c_8, C_8) or n=1 (c_1, C_1) for those clients. Table 17-5 specifies the replacement signals for those clients.

The support for 1-bit timing information (C_1) is client dependent. Clients for which the 8-bit timing information in C_m with m=8 is sufficient will not deploy the ability to transport ΣC_{1D} and the JC4/5/6 value will be fixed to all-0s.

The OPU0 payload for this mapping consists of 4 × 3808 bytes. The bytes in the OPU0 payload area are numbered from 1 to 15232. The OPU0 payload byte numbering for GMP 1-byte (8-bit) blocks is illustrated in Figure 17-12. In row 1 of the OPU0 frame the first byte will be labelled 1, the next byte will be labelled 2, etc.

Groups of eight successive bits (not necessary being a byte) of the client signal are mapped into a byte of the OPU0 payload area under control of the GMP data/stuff control mechanism. Each byte in the OPU0 payload area may either carry 8 client bits, or carry 8 stuff bits. The stuff bits are set to zero.

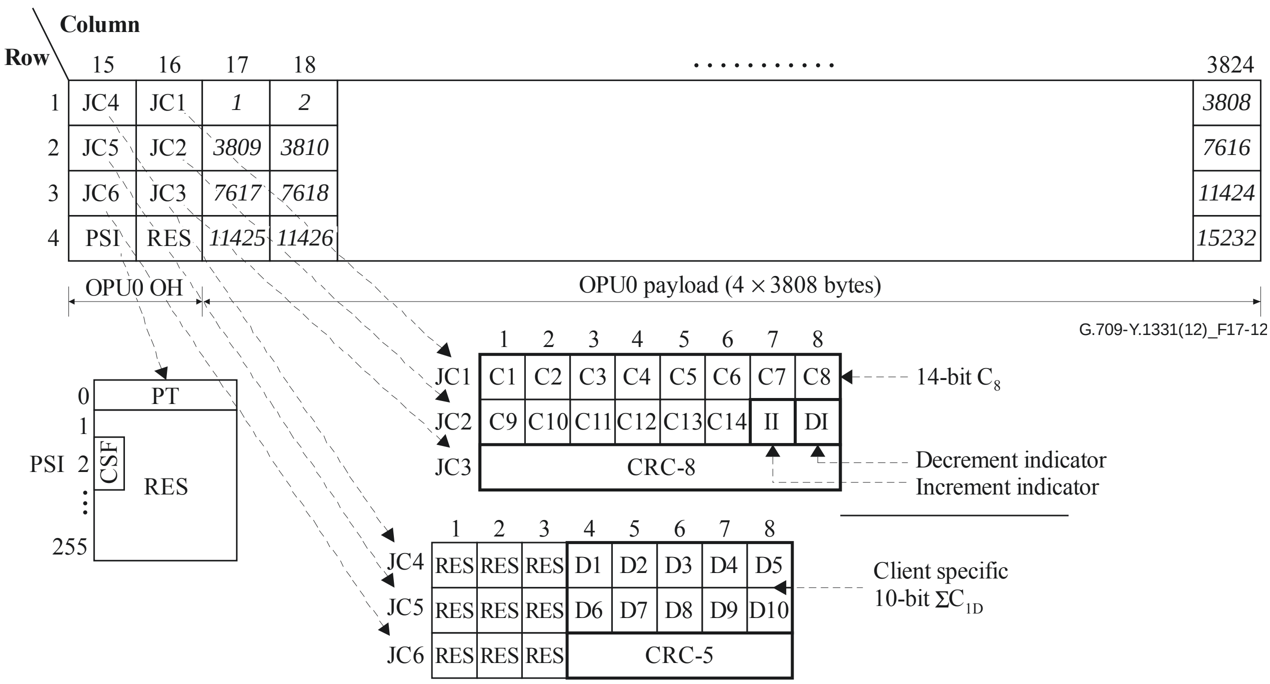


Figure 17-12 – OPU0 frame structure for the mapping of a sub-1.238 Gbit/s client signal

Table 17-4A – Cm (m=8) for sub-1.238G clients into OPU0

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
Transcoded 1000BASE-X (see clause 17.7.1.1)	15/16 × 1 250 000	±100	14405	14405.582	14407.311	14409.040	14410
STM-1	155 520	±20	1911	1911.924	1912.000	1912.076	1913
STM-4	622 080	±20	7647	7647.694	7648.000	7648.306	7649
FC-100	1 062 500	±100	13061	13061.061	13062.629	13064.196	13065
SBCON/ESCON	200 000	±200	2458	2458.307	2458.848	2459.389	2460
DVB-ASI	270 000	±100	3319	3319.046	3319.444	3319.843	3320
SDI	270 000	±2.8	3319	3319.369	3319.444	3319.520	3320

NOTE – Floor $C_{m,min}$ (m=8) and Ceiling $C_{m,max}$ (m=8) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_m values outside the range $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 17-4B – C_n (n=8 or 1) for sub-1.238G clients into OPU0

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
Transcoded 1000BASE-X (see clause 17.7.1.1)	15/16 × 1 250 000	±100	14405	14405.582	14407.311	14409.040	14410
FC-100	1 062 500	±100	13061	13061.061	13062.629	13064.196	13065
			Floor $C_{1,min}$ (note)	Minimum c_1	Nominal c_1	Maximum c_1	Ceiling $C_{1,max}$ (Note)
STM-1	155 520	±20	15295	15295.338	15296.000	15296.612	15297
STM-4	622 080	±20	61181	61181.553	61184.000	61186.447	61187
SDI	270 000	±2.8	For further study				

NOTE – Floor $C_{n,min}$ (n=8,1) and Ceiling $C_{n,max}$ (n=8,1) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_n values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n values outside the range $C_{n, min}$ to $C_{n, max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 17-5 – Replacement signal for sub-1.238G clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
STM-1	Generic-AIS	±20
STM-4	Generic-AIS	±20
1000BASE-X	Link Fault	±100
FC-100	NOS	±100
SBCON/ESCON	NOS	±200
DVB-ASI	Generic-AIS	±100
SDI	Generic-AIS	For further study

17.7.1.1 1000BASE-X transcoding

The 1000BASE-X signal (8B/10B coded, nominal bit rate of 1 250 000 kbit/s and a bit-rate tolerance up to ±100 ppm) is synchronously mapped into a 75-octet GFP-T frame stream with a bit rate of 15/16 × 1 250 000 kbit/s ±100 ppm (approximately 1 171 875 kbit/s ±100 ppm). This process is referred to as "timing transparent transcoding (TTT)". The 15/16 × 1 250 000 kbit/s ±100 ppm signal is then mapped into an OPU0 by means of the generic mapping procedure as specified in clause 17.7.1 and Annex D.

For 1000BASE-X client mapping, 1-bit timing information (C_1) is not needed, so OPU0 JC4/JC5/JC6 OH value will be fixed to all-0s.

The mapping of the 1000BASE-X signal into GFP-T is performed as specified in [ITU-T G.7041] with the following parameters:

– each GFP-T frame contains one superblock
– the 65B_PAD character is not used
– GFP idle frames are not used
– the GFP frame pFCS is not used.

During a signal fail condition of the incoming 1000BASE-X client signal (e.g., in the case of a loss of input signal), either:

– this failed incoming 1000BASE-X signal is replaced by a stream of 10B blocks, with a bit rate of 1 250 000 kbit/s ±100 ppm, each carrying a link fault indication as specified in [IEEE 802.3], which stream is then applied at the GFP-T mapper, or
– the GFP-T signal is replaced by a stream of GFP client signal fail (CSF) and GFP-idle frames as specified in [ITU-T G.7041] with a bit rate of 15/16 x 1 250 000 kbit/s ±100 ppm.

During either

– a signal fail condition of the incoming ODU0/OPU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), or
– incoming CSF frames as specified in [ITU-T G.7041]

the GFP-T de-mapper process generates a stream of 10B blocks, with each block carrying a link fault indication as specified in [IEEE 802.3] as a replacement signal for the lost 1000BASE-X signal.

NOTE – The Ethernet link fault indication is a stream of repeating /C1/C2/C1/C2/ … ordered sets, where C1 = /K28.5/D21.5/D0.0/D0.0/ and C2 = /K28.5/D2.2/D0.0/D0.0/. This character stream is then processed by the GFP-T mapper process in the same manner as if it were the received 8B/10B data stream, mapping it into GFP-T superblocks for transmission.

17.7.1.2 FC-100

During a signal fail condition of the incoming FC-100 signal (e.g., in the case of a loss of input signal), this failed incoming FC-100 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-100 signal.

17.7.1.3 SBCON/ESCON

During a signal fail condition of the incoming SBCON/ESCON signal (e.g., in the case of a loss of input signal), this failed incoming SBCON/ESCON signal is replaced by an NOS sequence as specified in [b-ANSI INCITS 296].

NOTE – The NOS sequence ordered set is defined as /K28.5/D0.2/.

During a signal fail condition of the incoming ODU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), NOS sequence ordered sets as specified in [b-ANSI INCITS 296] are generated as a replacement signal for the lost SBCON/ESCON signal.

17.7.2 Mapping a supra-1.238 to sub-2.488 Gbit/s CBR client signal into OPU1

Table 17-6A specifies the clients defined by this Recommendation and their GMP c_m and C_m with m=16 (c_{16}, C_{16}) minimum, nominal and maximum parameter values. Table 17-6B specifies the GMP c_n and C_n with n=8 (c_8, C_8) or n=1 (c_1, C_1) for those clients. Table 17-7 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU1 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU1 JC4/JC5/JC6 OH is client dependent.

The OPU1 payload for this mapping consists of 4×3808 bytes. The groups of 2 bytes in the OPU1 payload area are numbered from 1 to 7616. The OPU1 payload byte numbering for GMP 2-byte (16-bit) blocks is illustrated in Figure 17-13. In row 1 of the OPU1 frame the first 2-bytes will be labelled 1, the next 2-bytes will be labelled 2, etc.

Groups of sixteen successive bits of the client signal are mapped into a group of 2 successive bytes of the OPU1 payload area under control of the GMP data/stuff control mechanism. Each group of 2 bytes in the OPU1 payload area may either carry 16 client bits, or carry 16 stuff bits. The stuff bits are set to zero.

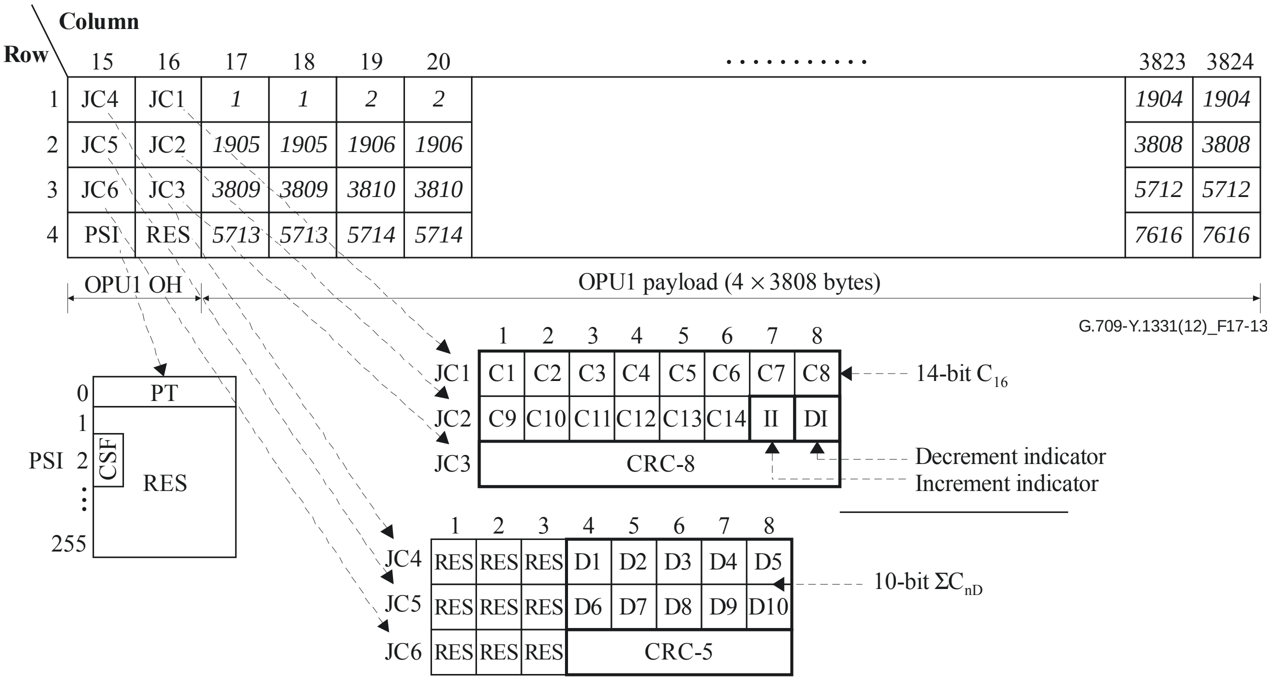


Figure 17-13 – OPU1 frame structure for the mapping of a supra-1.238 to sub-2.488 Gbit/s client signal

Table 17-6A – C_m (m=16) for supra-1.238 to sub-2.488G clients into OPU1

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	Floor $C_{16,min}$ (Note)	Minimum c_{16}	Nominal c_{16}	Maximum c_{16}	Ceiling $C_{16,max}$ (Note)
FC-200	2 125 000	±100	6503	6503.206	6503.987	6504.767	6505
1.5G SDI	1 485 000	±10	4545	4545.003	4545.139	4545.275	4546
1.5G SDI	1 485 000/1.001	±10	4540	4540.462	4540.598	4540.735	4541
NOTE – Floor $C_{m,min}$ (m=16) and Ceiling $C_{m,max}$ (m=16) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_m values outside the range $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-6B – C_n (n=8 or 1) for supra-1.238 to sub-2.488G clients into OPU1

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
FC-200	2 125 000	±100	13006	13006.412	13007.973	13009.534	13010
1.5G SDI	1 485 000	±10	For further study				
1.5G SDI	1 485 000/1.001	±10	For further study				
			Floor $C_{1,min}$ (Note)	Minimum c_1	Nominal c_1	Maximum c_1	Ceiling $C_{1,max}$ (Note)

1.5G SDI	1 485 000	±10	For further study					
1.5G SDI	1 485 000/1.001	±10	For further study					
NOTE – Floor Cn,min (n=8,1) and Ceiling Cn,max (n=8,1) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated Cn values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that Cn values outside the range Cn, min to Cn, max may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.								

Table 17-7 – Replacement signal for supra-1.238 to sub-2.488 Gbit/s clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
FC-200	NOS	±100
1.5G SDI	Generic-AIS	For further study

17.7.2.1 FC-200

During a signal fail condition of the incoming FC-200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-200 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODU1 signal (e.g., in the case of an ODU1-AIS, ODU1-LCK, ODU1-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-200 signal.

17.7.3 Mapping CBR client signals into OPU2

Table 17-8A specifies the clients defined by this Recommendation and their GMP c_m and C_m with m=64 (c_{64}, C_{64}) minimum, nominal and maximum parameter values. Table 17-8B specifies the GMP c_n and C_n with n=8 (c_8, C_8) or n=1 (c_1, C_1) for those clients. Table 17-9 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU2 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU2 JC4/JC5/JC6 OH is client dependent.

The OPU2 payload for this mapping consists of 4 × 3808 bytes. The groups of eight bytes in the OPU2 payload area are numbered from 1 to 1904. The OPU2 payload byte numbering for GMP 8-byte (64-bit) blocks is illustrated in Figure 17-14. In row 1 of the OPU2 frame the first 8-bytes will be labelled 1, the next 8-bytes will be labelled 2, etc.

Groups of sixty-four successive bits of the client signal are mapped into a group of eight successive bytes of the OPU2 payload area under control of the GMP data/stuff control mechanism. Each group of eight bytes in the OPU2 payload area may either carry 64 client bits, or carry 64 stuff bits. The stuff bits are set to zero.

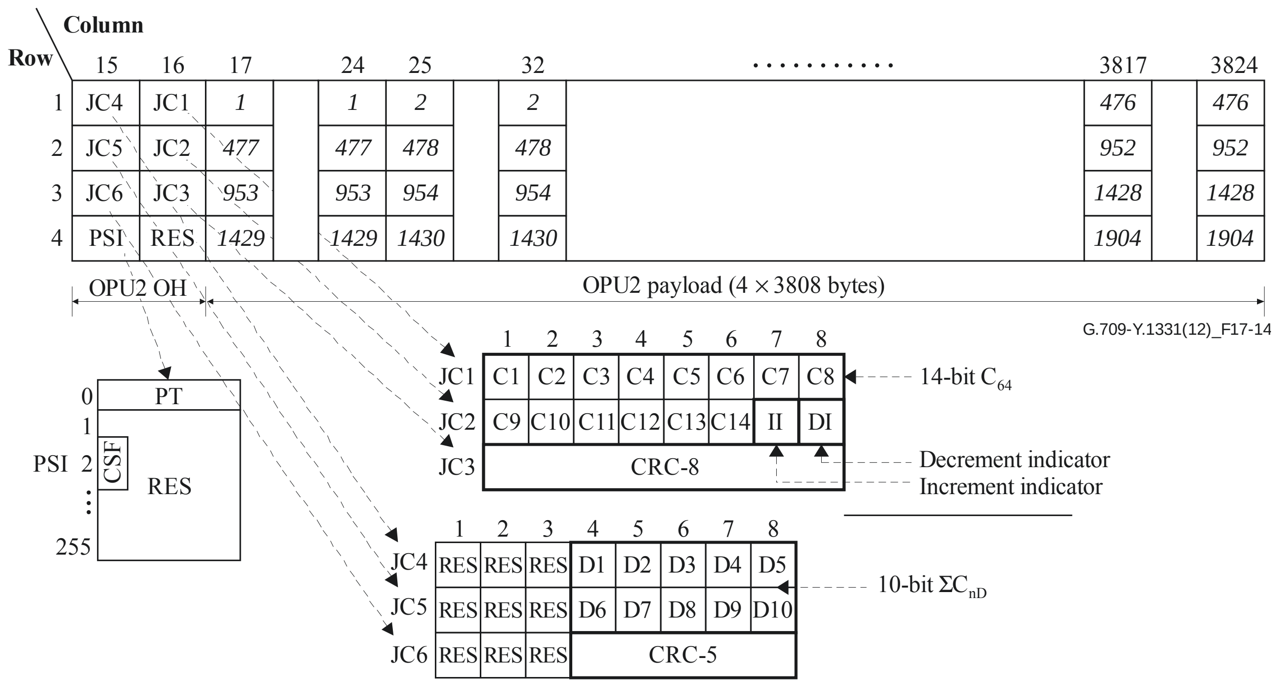


Figure 17-14 – OPU2 frame structure for the mapping of a CBR client signal

Table 17-8A – C_m (m=64) for CBR clients into OPU2

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{64,min}$ (Note)	Minimum c_{64}	Nominal c_{64}	Maximum c_{64}	Ceiling $C_{64,max}$ (Note)
For further study							
NOTE – Floor $C_{m,min}$ (m=64) and Ceiling $C_{m,max}$ (m=64) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_m values outside the range $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-8B – C_n (n=8 or 1) for CBR clients into OPU2

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
For further study							
			Floor $C_{1,min}$ (Note)	Minimum c_1	Nominal c_1	Maximum c_1	Ceiling $C_{1,max}$ (Note)
For further study							
NOTE – Floor $C_{n,min}$ (n=8,1) and Ceiling $C_{n,max}$ (n=8,1) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_n values throughout this range							

but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n values outside the range $C_{n, min}$ to $C_{n, max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 17-9 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
For further study		

17.7.4 Mapping CBR client signals into OPU3

Table 17-10A specifies the clients defined by this Recommendation and their GMP c_m and C_m with m=256 (c_{256}, C_{256}) minimum, nominal and maximum parameter values. Table 17-10B specifies the GMP c_n and C_n with n=8 (c_8, C_8) or n=1 (c_1, C_1) for those clients. Table 17-11 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU3 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU3 JC4/JC5/JC6 OH is client dependent.

The OPU3 payload for this mapping consists of 4 × 3808 bytes. The groups of 32 bytes in the OPU3 payload area are numbered from 1 to 476. The OPU3 payload byte numbering for GMP 32-byte (256-bit) blocks is illustrated in Figure 17-15. In row 1 of the OPU3 frame the first 32-bytes will be labelled 1, the next 32-bytes will be labelled 2, etc.

Groups of two hundred-fifty-six successive bits of the client signal are mapped into a group of 32 successive bytes of the OPU3 payload area under control of the GMP data/stuff control mechanism. Each group of 32 bytes in the OPU3 payload area may either carry 256 client bits, or carry 256 stuff bits. The stuff bits are set to zero.

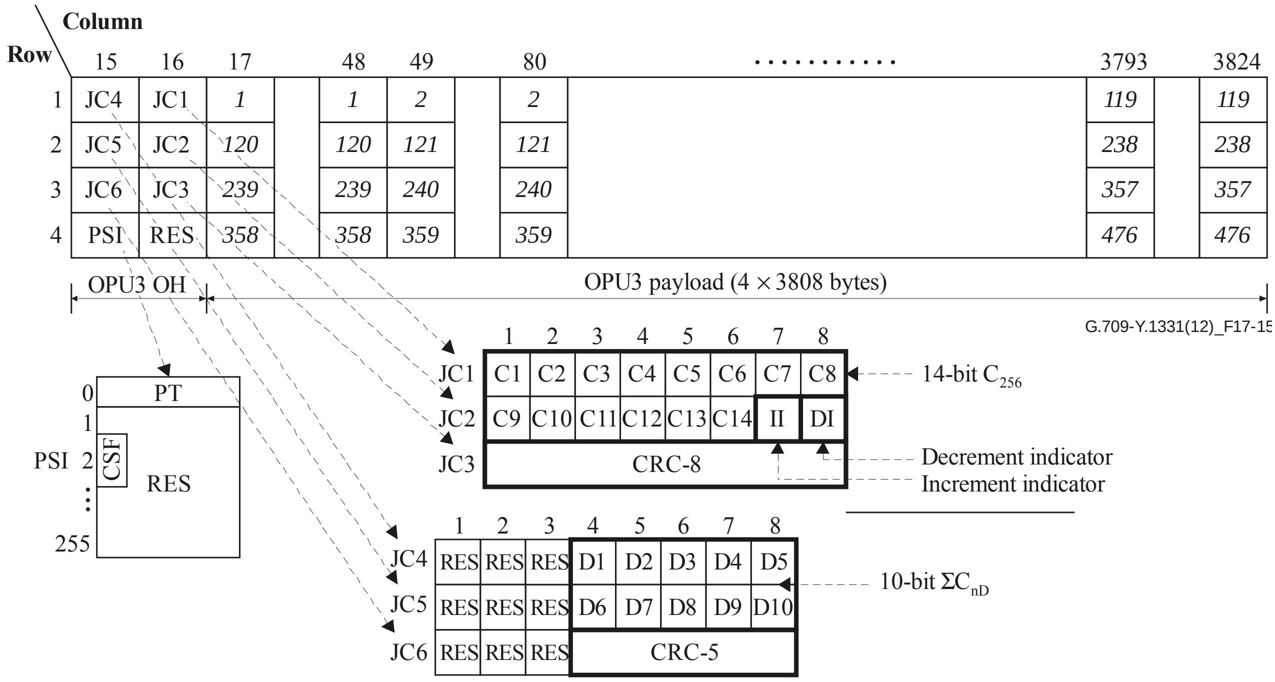


Figure 17-15 – OPU3 frame structure for the mapping of a CBR client signal

Table 17-10A – C_m (m=256) for CBR clients into OPU3

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{256,min}$ (Note)	Minimum c_{256}	Nominal c_{256}	Maximum c_{256}	Ceiling $C_{256,max}$ (Note)
Transcoded 40GBASE-R (see clause 17.7.4.1)	1027/1024 × 64/66 × 41 250 000	±100	475	475.548	475.605	475.662	476
NOTE – Floor $C_{m,min}$ (m=256) and Ceiling $C_{m,max}$ (m=256) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_m values outside the range $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-10B – C_n (n=8 or 1) for CBR clients into OPU3

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
Transcoded 40GBASE-R (see clause 17.7.4.1)	1027/1024 × 64/66 × 41 250 000	±100	15217	15217.529	15219.355	15221.181	15222
			Floor $C_{1,min}$ (Note)	**Minimum c_1**	**Nominal c_1**	**Maximum c_1**	**Ceiling $C_{1,max}$** (Note)
For further study							
NOTE – Floor $C_{n,min}$ (n=8,1) and Ceiling $C_{n,max}$ (n=8,1) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_n values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n values outside the range $C_{n, min}$ to $C_{n, max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-11 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
40GBASE-R	Continuous 40GBASE-R local fault sequence ordered sets with four PCS lane alignment markers inserted after each 16383 x 4 sixty-six-bit blocks	±100

A 40GBASE-R local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

17.7.4.1 40GBASE-R multi-lane processing and transcoding

The 40GBASE-R client signal (64B/66B encoded, nominal aggregate bit-rate of 41 250 000 kbit/s, ±100 ppm) is recovered using the process described in Annex E for parallel 64B/66B interfaces. The lane(s) of the physical interface are bit-disinterleaved, if necessary, into four streams of 10 312 500 kbit/s. 66B block lock and lane alignment marker lock are acquired on each PCS lane, allowing the 66B blocks to be de-skewed and reordered.

The resulting sequence is descrambled and transcoded according to the process described in Annex B into 513B code blocks. Each pair of two 513B code blocks is combined according to the process described in Annex F into a 1027B block, resulting in a bit stream of 1027/1024 × 40 000 000 kbit/s ±100 ppm (40,117,187.500 kbit/s ±100 ppm). This process is referred to as "timing transparent transcoding (TTT)", mapping a bit stream which is 1027/1056 times the bit-rate of the aggregate Ethernet signal.

In the mapper, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

The de-mapper will insert a compensated Ethernet PCS lane BIP as described in Annex E. In addition, as described in Annex E, the combined error mask resulting from the PCS BIP-8 error mask and the OTN BIP-8 error mask may be used as a non-intrusive monitor.

For 40GBASE-R client mapping, 1-bit timing information (C_1) is not needed.

The de-mapper will recover from the output of the GMP processor 1027B block lock, and then trans-decode each 1027B block to sixteen 66B blocks as described in Annex E. Trans-decoded lane alignment markers are constructed with a compensated BIP-8. The 66B blocks are then re-distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

17.7.5 Mapping CBR client signals into OPU4

Table 17-12A specifies the clients defined by this Recommendation and their GMP c_m and C_m with m=640 (c_{640}, C_{640}) minimum, nominal and maximum parameter values. Table 17-12B specifies the GMP c_n and C_n with n=8 (c_8, C_8) or n=1 (c_1, C_1) for those clients. Table 17-13 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU4 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU4 JC4/JC5/JC6 OH is client dependent.

The OPU4 payload for this mapping consists of 4 × 3800 bytes for client data and 4 × 8 bytes with fixed stuff. The groups of 80 bytes in the OPU4 payload area are numbered from 1 to 190. The OPU4 payload byte numbering for GMP 80-byte (640-bit) blocks is illustrated in Figure 17-16. In row 1 of the OPU4 frame the first 80-bytes will be labelled 1, the next 80-bytes will be labelled 2, etc.

Groups of six hundred and forty successive bits of the client signal are mapped into a group of 80 successive bytes of the OPU4 payload area under control of the GMP data/stuff control mechanism. Each group of 80 bytes in the OPU4 payload area may either carry 640 client bits, or carry 640 stuff bits. The stuff bits are set to zero.

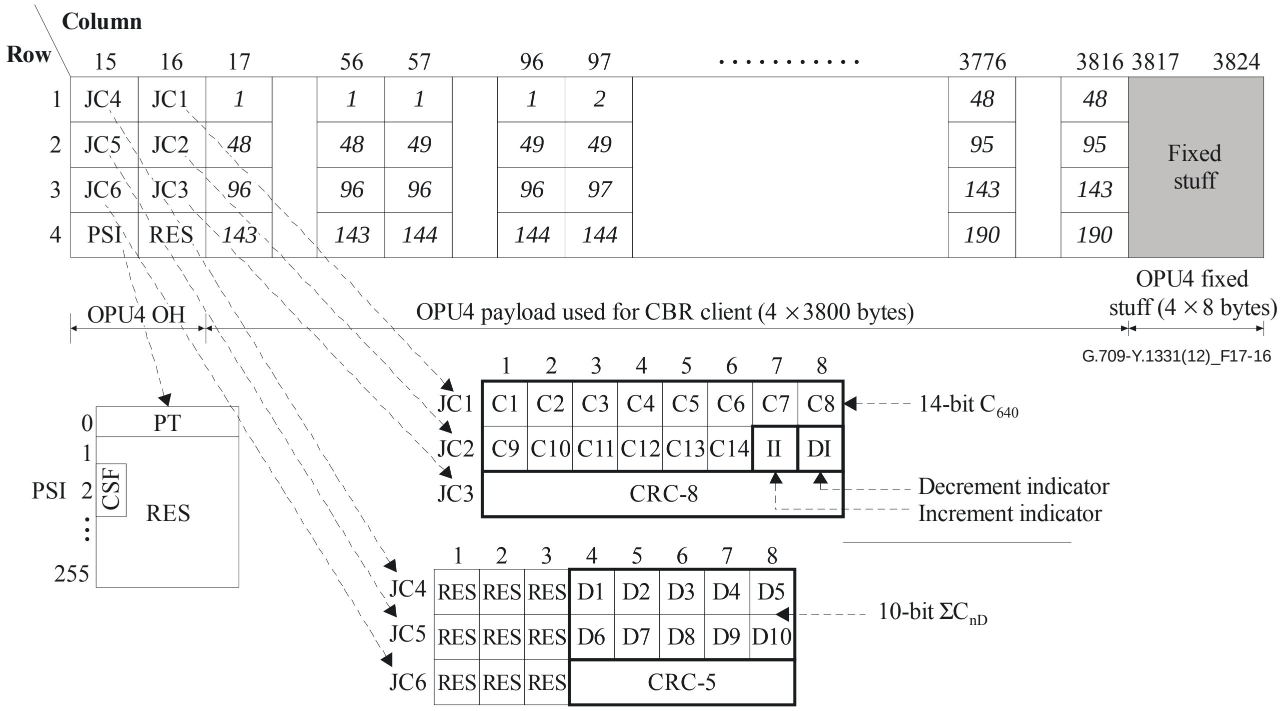


Figure 17-16 – OPU4 frame structure for the mapping of a CBR client signal

Table 17-12A – C_m (m=640) for CBR clients into OPU4

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	Floor $C_{640,min}$ (Note)	Minimum c_{640}	Nominal c_{640}	Maximum c_{640}	Ceiling $C_{640,max}$ (Note)
100GBASE-R (see 17.7.5.1)	103 125 000	±100	188	188.131	188.154	188.177	189
NOTE – Floor $C_{m,min}$ (m=640) and Ceiling $C_{m,max}$ (m=640) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_m values outside the range $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-12B – C_n (n=8 or 1) for CBR clients into OPU4

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
100GBASE-R (see 17.7.5.1)	103 125 000	±100	15050	15050.518	15052.324	15054.131	15055
			Floor $C_{1,min}$ (Note)	**Minimum c_1**	**Nominal c_1**	**Maximum c_1**	**Ceiling $C_{1,max}$** (Note)
For further study							
NOTE – Floor $C_{n,min}$ (n=8,1) and Ceiling $C_{n,max}$ (n=8,1) values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_n values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n values outside the range $C_{n,\ min}$ to $C_{n,\ max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.							

Table 17-13 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
100GBASE-R (see 17.7.5.1)	Continuous 100GBASE-R local fault sequence ordered sets with 20 PCS lane alignment markers inserted after each 16383 x 20 sixty-six-bit blocks	±100

A 100GBASE-R local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate a "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

17.7.5.1 100GBASE-R multi-lane processing

The 100GBASE-R client signal (64B/66B encoded, nominal aggregate bit-rate of 103 125 000 kbit/s ±100 ppm) is recovered using the process described in Annex E for parallel 64B/66B interfaces. The lane(s) of the physical interface are bit-disinterleaved, if necessary, into twenty streams of 5 161 250 kbit/s. 66B block lock and lane alignment marker lock are acquired on each PCS lane, allowing the 66B blocks to be de-skewed and reordered.

In the mapper, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

The de-mapper will pass through the PCS lane BIP from the ingress as described in Annex E. In addition, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

For 100GBASE-R client mapping, 1-bit timing information (C_1) is not needed.

The de-mapper will recover from the output of the GMP processor 64B/66B block lock per the state diagram in Figure 82-10 [IEEE 802.3ba]. The 66B blocks are re-distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

17.8 Mapping a 1000BASE-X and FC-1200 signal via timing transparent transcoding into OPUk

17.8.1 Mapping a 1000BASE-X signal into OPU0

Refer to clause 17.7.1 for the mapping of the transcoded 1000BASE-X signal and to clause 17.7.1.1 for the transcoding of the 1000BASE-X signal.

17.8.2 Mapping an FC-1200 signal into OPU2e

The nominal line rate for FC-1200 is 10 518 750 kbit/s ± 100 ppm, and must therefore be compressed to a suitable rate to fit into an OPU2e.

The adaptation of the 64B/66B encoded FC-1200 client is done by transcoding a group of eight 66B blocks into one 513B block (as described in Annex B), assembling eight 513B blocks into one 516-octet superblock and encapsulating seventeen 516-octet superblocks into an 8800 octet GFP frame as illustrated in Figure 17-18. The GFP frame consists of 2200 rows with 32 bits per row. The first row contains the GFP core header, the second row the GFP payload header. The next four rows contain 16 bytes reserved for future international standardization. The next seventeen times 129 rows contain the seventeen superblocks #1 to #17. The last row contains the GFP payload FCS. The flag (F) bit of 513B block #i (i = 0..7) is carried in Flag #i bit located in the superblock flags field. The remaining 512 bits of each of the eight 513B blocks of a superblock are carried in 16 rows of the superblock data field; bits of 513B block #0 in the first 16 rows of the superblock, bits of 513B block #1 in the next 16 rows, etc. Each 513B block contains 'j' (j = 0..8) control blocks (CB1 to CBj) and '8-j' all-data blocks (DB1..DB8-j) as specified in Annex B. Figure 17-18 presents a 513B block with three control blocks and five all-data blocks. A 513B block may contain zero to eight control blocks and a superblock may contain thus zero to sixty-four control blocks.

NOTE 1 – The GFP encapsulation stage does not generate GFP-idle frames and therefore the generated GFP stream is synchronous to the FC-1200 client stream. The adaptation process performs a 50/51 rate compression, so the resulting GFP stream has a signal bit rate of 50/51 × 10.51875 Gbit/s ± 100 ppm (i.e., 10 312 500 kbit/s ± 100 ppm).

The stream of 8800 octet GFP frames is byte-synchronous mapped into the OPU2e payload by aligning the byte structure of every GFP frame with the byte structure of the OPU2e payload (see Figure 17-17). Sixty-four fixed stuff (FS) bytes are added in columns 1905 to 1920 of the OPU2e payload. All the GFP frames have the same length (8800 octets). The GFP frames are not aligned with the OPU2e payload structure and may cross the boundary between two OPU2e frames.

During a signal fail condition of the incoming FC-1200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-1200 signal is replaced by a stream of 66B blocks, with each block carrying two local fault sequence ordered sets as specified in [b-ANSI INCITS 364]. This replacement signal is then applied at the transcoding process.

NOTE 2 – Local fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH=10 0x55 00 00 01 00 00 00 01.

During a signal fail condition of the incoming ODU2e/OPU2e signal (e.g., in the case of an ODU2e-AIS, ODU2e-LCK, ODU2e-OCI condition) a stream of 66B blocks, with each block carrying two local fault sequence ordered sets as specified in [b-ANSI INCITS 364] is generated as a replacement signal for the lost FC-1200 signal.

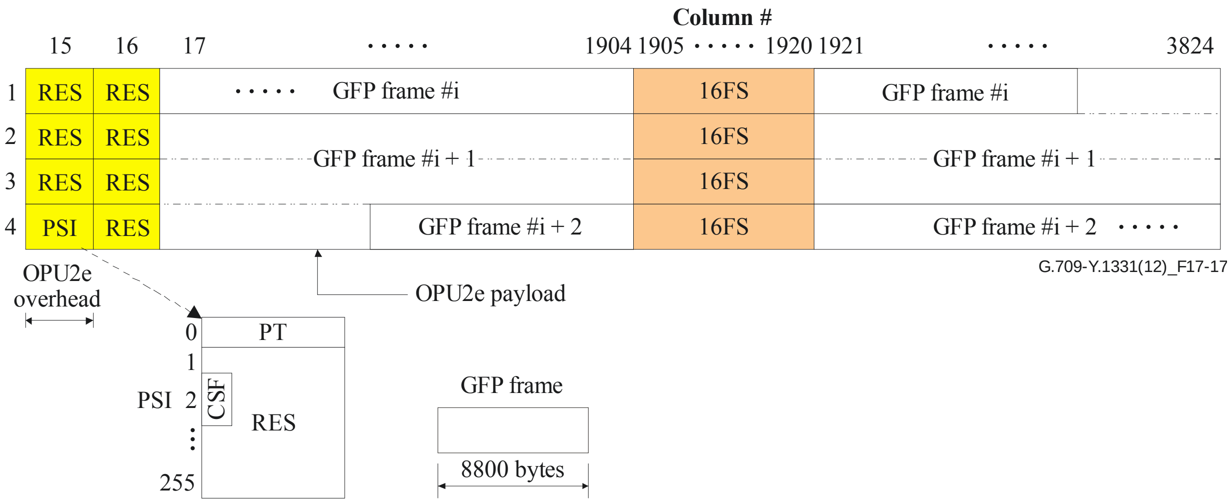


Figure 17-17 – Mapping of transcoded FC-1200 into OPU2e

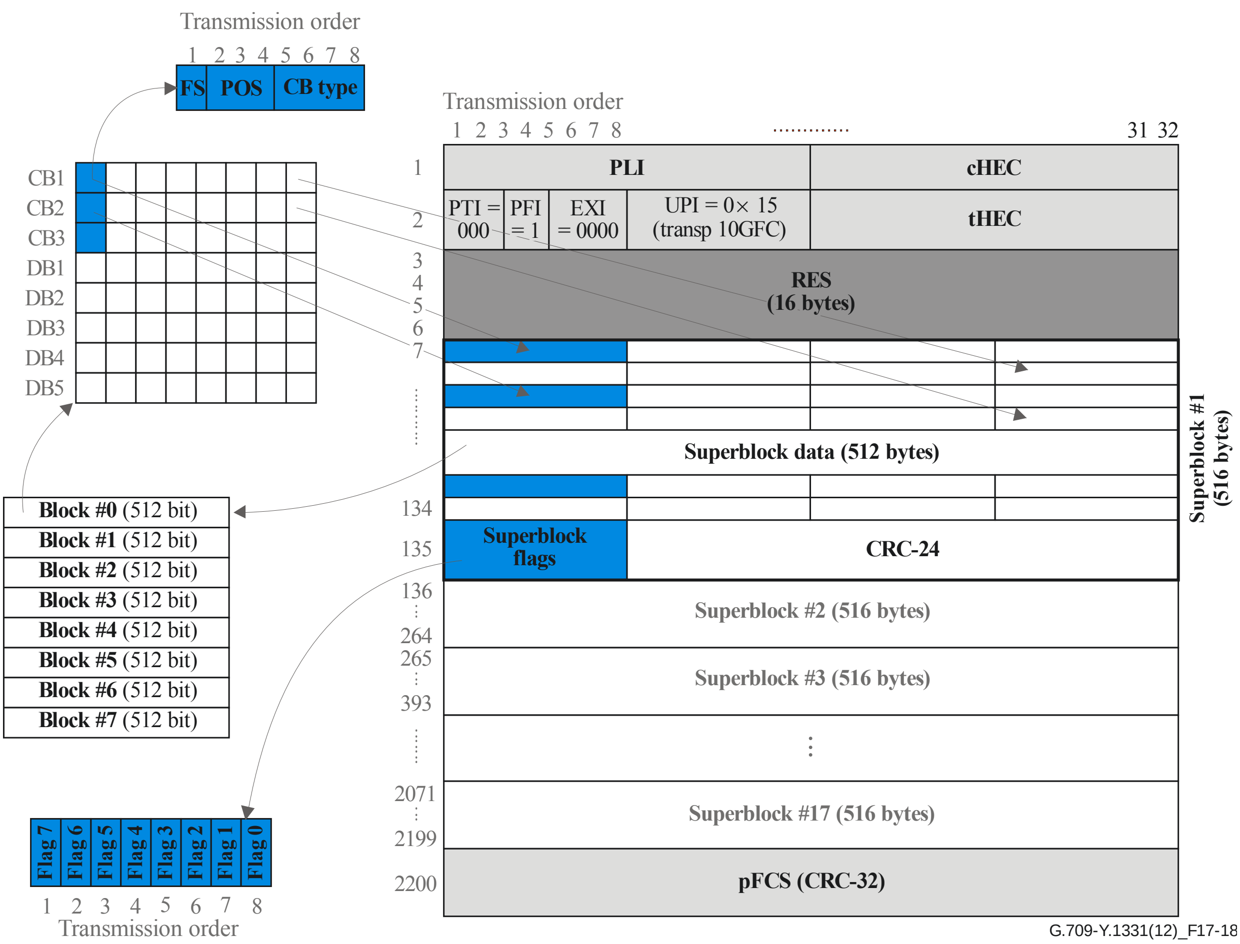


Figure 17-18 – GFP frame format for FC-1200

GFP framing is used to facilitate delineation of the superblock structure by the receiver. The leading flag bits from each of the eight 513B blocks are relocated into a single octet at the end of the 513-octet superblock data field (labelled "Superblock flags").

To minimize the risk of incorrect decoding due to errors in the 1 to 65 octets of "control" information (Flags, FC, POS, CB_Type), a CRC-24 is calculated over the 65 octets within each superblock that may contain such "control" information and appended to form a 516 octet superblock. The 65 octets in the 516-octet superblock over which the CRC-24 is calculated are the octets (1+8n) with n=0..64 (i.e., octets 1, 9, 17, .., 513). The generator polynomial for the CRC-24 is $G(x) = x^{24} + x^{21} + x^{20} + x^{17} + x^{15} + x^{11} + x^{9} + x^{8} + x^{6} + x^{5} + x + 1$ with an all-ones initialization value, where x^{24} corresponds to the MSB and x^{0} to the LSB. This superblock CRC is generated by the source adaptation process using the following steps:

1) The 65 octets of "control" information (Flags, POS, CB_Type) are taken in network octet order (see Figure 17-18), most significant bit first, to form a 520-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 519.
2) $M(x)$ is multiplied by x^{24} and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 23 or less.
3) The coefficients of $R(x)$ are considered to be a 24-bit sequence, where x^{23} is the most significant bit.
4) After inversion, this 24-bit sequence is the CRC-24.

Exactly 17 of these 516-octet superblocks are prefixed with the standard GFP core and type headers and 16 octets of "reserved" (padding). Because the number of 516-octet superblocks per GFP frame is known a priori, it is possible for this mapping scheme to operate in a cut-through (as opposed to store and forward) fashion, thus minimizing the mapping latency.

The payload FCS (a CRC-32) is appended to the end of each GFP frame and is calculated across the payload information field of the GFP frame as per [ITU-T G.7041]. The purpose of the payload FCS is to provide visibility of bit errors occurring anywhere in the GFP payload information field and thus augments the coverage provided by the per-superblock CRC-24 (which only provides coverage for the "control" overhead in each superblock). The payload FCS is only for the purposes of gathering statistics.

All octets in the GFP payload area are scrambled using the $X^{43} + 1$ self-synchronous scrambler, again as per [ITU-T G.7041].

17.9 Mapping a supra-2.488 CBR Gbit/s signal into OPUflex

Mapping of a supra-2.488 CBR Gbit/s client signal (with up to ±100 ppm bit-rate tolerance) into an OPUflex is performed by a bit-synchronous mapping procedure (BMP). Table 17-14 specifies the clients defined by this Recommendation.

The bit-synchronous mapping processes deployed to map constant bit rate client signals into an OPUflex does not generate any justification control signals.

The OPUflex clock for the bit-synchronous mapping is derived from the client signal. During a signal fail condition of the incoming client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the appropriate replacement signal as defined in Table 17-15. The OPUflex payload signal bit rate shall be within the limits specified in Table 7-3 and neither a frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming client signal shall be done without introducing a frequency or frame phase discontinuity.

During a signal fail condition of the incoming ODUflex/OPUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), the failed client signal is replaced by the appropriate replacement signal as defined in Table 17-15.

The OPUflex overhead for this mapping consists of:

– a payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-8, the client signal fail (CSF) and 254 bytes plus 7 bits reserved for future international standardization (RES);

– three justification control (JC) bytes, consisting of two bits for justification control (with fixed 00 value) and six bits reserved for future international standardization;
– one negative justification opportunity (NJO) byte (carrying a justification byte); and
– three bytes reserved for future international standardization (RES).

NOTE – To allow the use of a common asynchronous/bit-synchronous de-mapper circuit for CBR client signals into ODUk (k=1,2,3 and flex), JC, NJO and PJO fields are assumed to be present in the OPUflex frame structure for the mapping of a supra-2.488G CBR client signal (Figure 17-19). This OPUflex frame structure is now compatible with the OPUk frame structure for the mapping of a CBR2G5, CBR10G or CBR40G signal (Figure 17-1). As a CBR signal is mapped into the OPUflex without justification, the NJO field contains a justification byte (stuff), the PJO field contains a data byte (D), and the JC bits are fixed to 00.

The OPUflex payload for this mapping consists of 4 × 3808 bytes (Figure 17-19). Groups of eight successive bits (not necessarily being a byte) of the client signal are mapped into a data (D) byte of the OPUflex payload area under control of the BMP control mechanism. Each data byte in the OPUflex payload area carries 8 client bits.

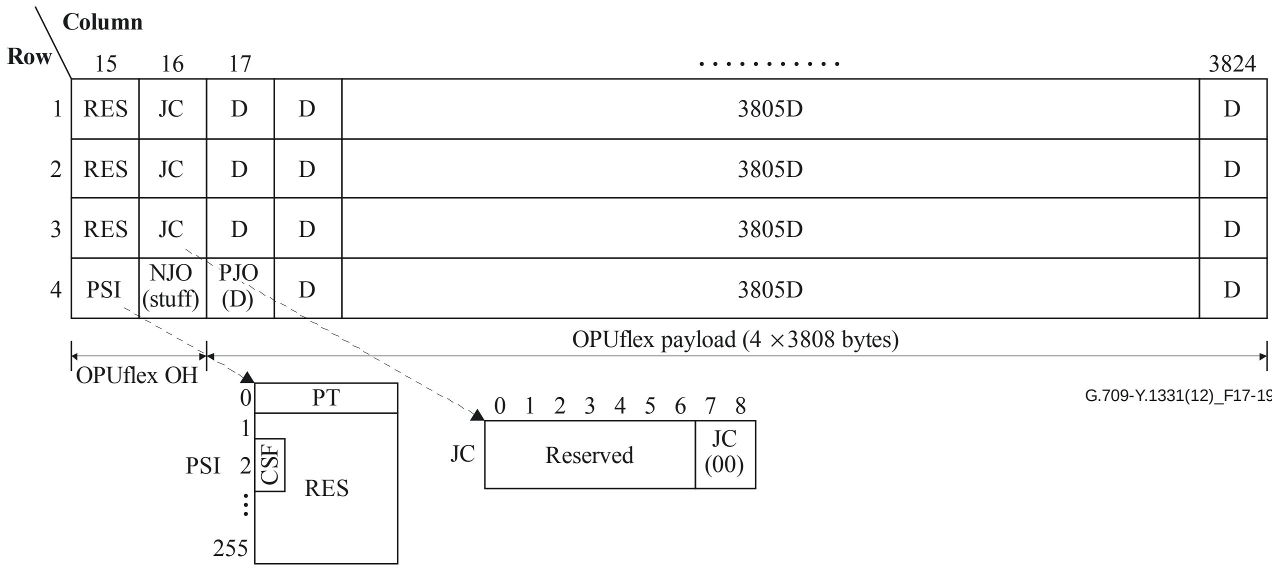


Figure 17-19 – OPUflex frame structure for the mapping of a supra-2.488 Gbit/s client signal

Table 17-14 – supra-2.488G CBR clients

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)
FC-400	4 250 000	±100
FC-800	8 500 000	±100
FC-1600	14 025 000	±100
IB SDR	2 500 000	±100
IB DDR	5 000 000	±100
IB QDR	10 000 000	±100
3G SDI	2 970 000	± 10
3G SDI	2 970 000/1.001	±10

Table 17-15 – Replacement signal for supra-2.488 Gbit/s clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
FC-400	NOS	±100
FC-800	NOS	±100
FC-1600	NOS	±100
IB SDR	For further study	±100
IB DDR	For further study	±100
IB QDR	For further study	±100
3G SDI	Generic-AIS	For further study

17.9.1 FC-400 and FC-800

During a signal fail condition of the incoming FC-400/FC-800 signal (e.g., in the case of a loss of input signal), this failed incoming FC-400/FC-800 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-400/FC-800 signal.

17.9.2 FC-1600

During a signal fail condition of the incoming FC-1600 signal (e.g., in the case of a loss of input signal), this failed incoming FC-1600 signal is replaced by a NOS primitive sequence as specified in [b-INCITS 470].

During signal fail condition of the incoming ODUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-1600 signal.

18 Concatenation

Concatenation in the OTN is realized by means of virtual concatenation of OPUk signals.

18.1 Virtual concatenation of OPUk

18.1.1 Virtual concatenated OPUk (OPUk-Xv, k = 1 .. 3, X = 1 .. 256)

The OPUk-Xv (k = 1,2,3) frame structure is shown in Figure 18-1. It is organized in an octet-based block frame structure with 4 rows and X × 3810 columns.

NOTE 1 – Virtual concatenation for OPUk with k=0,2e,4,flex is not supported.

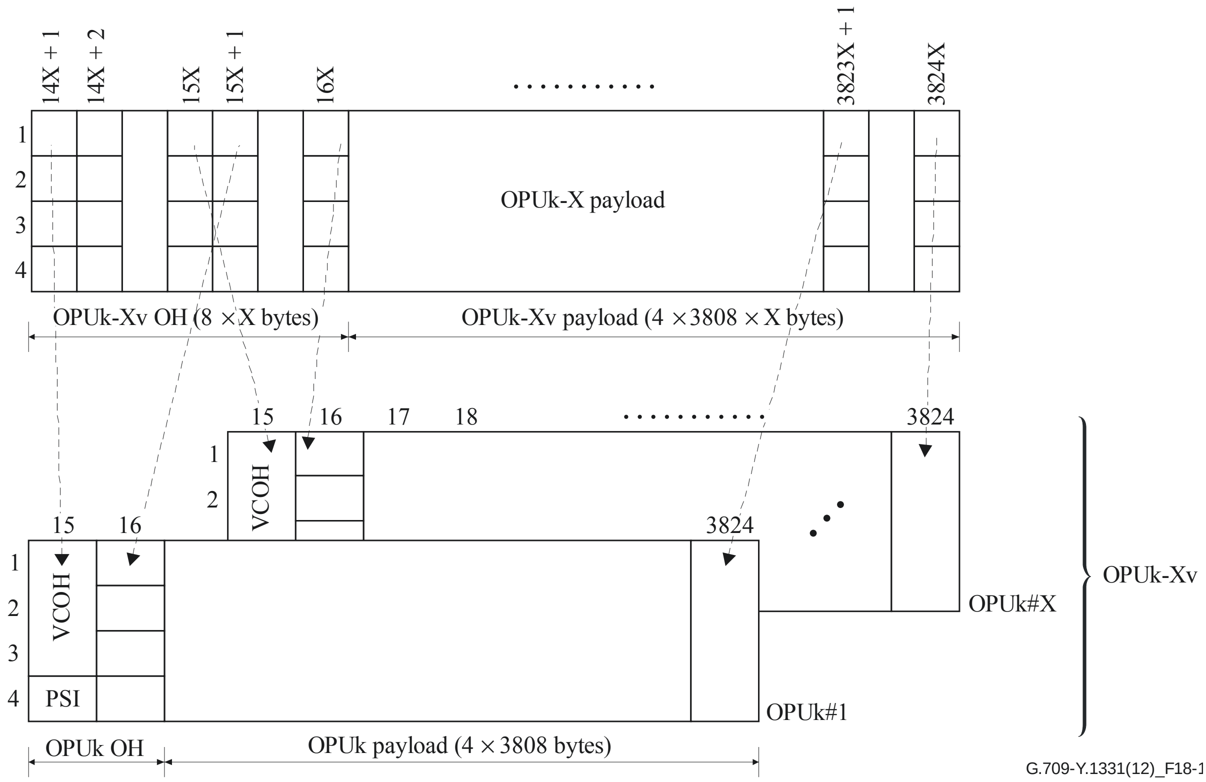


Figure 18-1 – OPUk-Xv structure

The two main areas of the OPUk-Xv frame are:

- OPUk-Xv overhead area
- OPUk-Xv payload area.

Columns 14X+1 to 16X of the OPUk-Xv are dedicated to the OPUk-Xv overhead area.

Columns 16X+1 to 3824X of the OPUk-Xv are dedicated to the OPUk-Xv payload area.

NOTE 2 – OPUk-Xv column numbers are derived from the OPUk columns in the ODUk frame.

An OPUk-Xv provides a contiguous payload area of X OPUk payload areas (OPUk-X-PLD) with a payload capacity of $X \times 238/(239\text{-}k) \times 4^{(k-1)} \times 2\ 488\ 320$ kbit/s ±20 ppm as shown in Figure 18-1. The OPUk-X-PLD is mapped in X individual OPUks which form the OPUk-Xv.

Each OPUk in the OPUk-Xv is transported in an ODUk and the X ODUks form the ODUk-Xv.

Each ODUk of the ODUk-Xv is transported individually through the network. Due to different propagation delay of the ODUks, a differential delay will occur between the individual ODUks and thus OPUks. This differential delay has to be compensated and the individual OPUks have to be realigned for access to the contiguous payload area.

18.1.2 OPUk-Xv OH description

18.1.2.1 OPUk-Xv OH location

The OPUk-Xv overhead consists of: X times a payload structure identifier (PSI) including the payload type (PT) and client signal fail (CSF), X times the virtual concatenation (VCOH) overhead used for a virtual concatenation specific sequence and multiframe indication and an overhead (e.g., justification control and opportunity bits) associated with the mapping of client signals into the OPUk payload as shown in Figure 18-1. The PSI and VCOH overhead is specific for each individual OPUk of the OPUk-Xv, while the mapping specific overhead is related to the concatenated signal.

The OPUk-Xv VCOH consists of a 3-byte VCOH per OPUk. The VCOH bytes in each OPUk are used as defined in Figure 18-2.

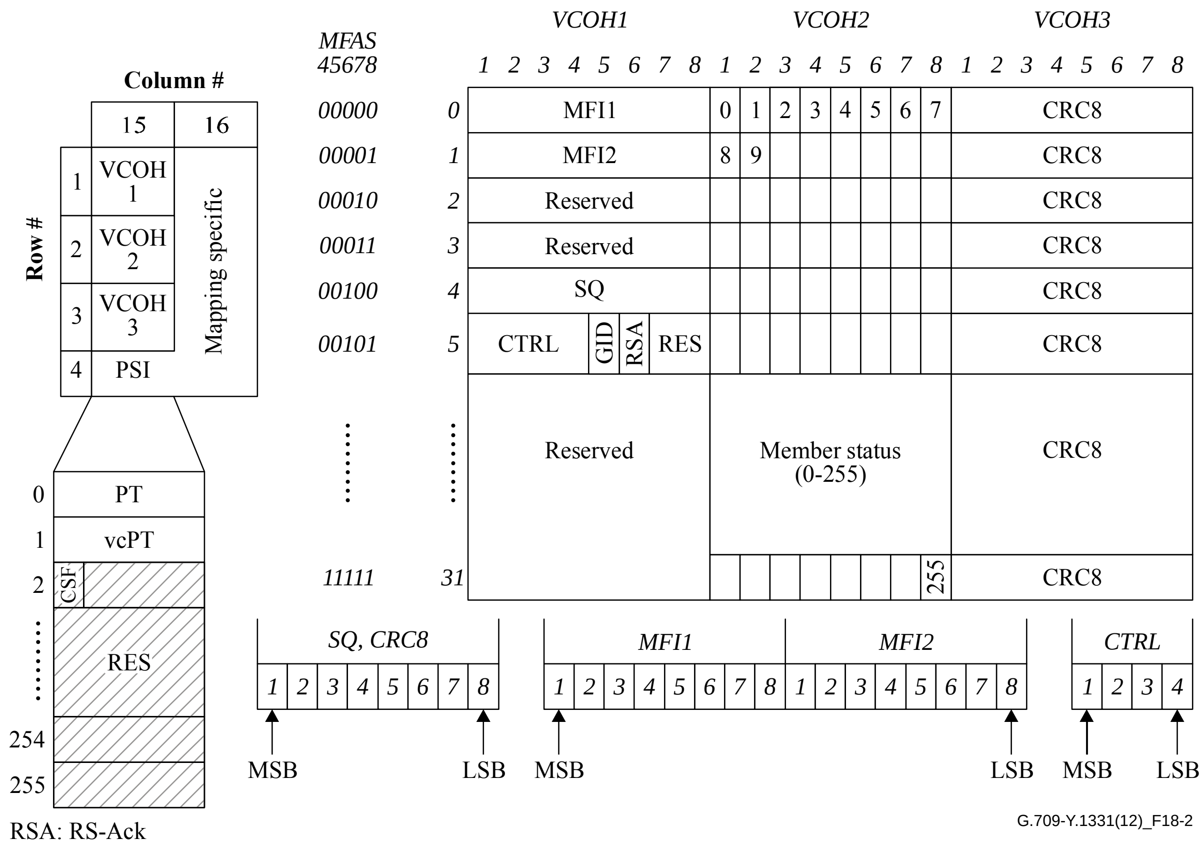


Figure 18-2 – OPUk-Xv virtual concatenation overhead

18.1.2.2 OPUk-Xv OH definition

18.1.2.2.1 OPUk-Xv payload structure identifier (PSI)

In each OPUk of the OPUk-Xv one byte is allocated in row 4, column 15 (Figure 18-2) to transport a 256-byte payload structure identifier (PSI) signal as defined in clause 15.9.2.

PSI[1]is used for a virtual concatenation specific payload type identifier (vcPT).

The PSI content is identical for each OPUk of the OPUk-Xv.

18.1.2.2.1.1 OPUk-Xv payload type (vcPT)

A one-byte OPUk-Xv payload type signal is defined in the PSI[1] byte of the payload structure identifier to indicate the composition of the OPUk-Xv signal. The code points are defined in Table 18-1.

Table 18-1 – Payload type (vcPT) code points for virtual concatenated OPUk (OPUk-Xv) signals

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 0 0	0 0 0 1	01	Experimental mapping (Note 3)
0 0 0 0	0 0 1 0	02	Asynchronous CBR mapping, see clauses 18.2.1 and 18.2.2
0 0 0 0	0 0 1 1	03	Bit-synchronous CBR mapping, see clauses 18.2.1 and 18.2.2
0 0 0 0	0 1 0 0	04	ATM mapping, see clause 18.2.3

Table 18-1 – Payload type (vcPT) code points for virtual concatenated OPUk (OPUk-Xv) signals

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 0 0	0 1 0 1	05	GFP mapping, see clause 18.2.4
0 0 0 1	0 0 0 0	10	Bit stream with octet timing mapping, see clause 18.2.6
0 0 0 1	0 0 0 1	11	Bit stream without octet timing mapping, see clause 18.2.6
0 1 0 1	0 1 0 1	55	Not available (Note 2)
0 1 1 0	0 1 1 0	66	Not available (Note 2)
1 0 0 0	x x x x	80-8F	Reserved codes for proprietary use (Note 4)
1 1 1 1	1 1 0 1	FD	NULL test signal mapping, see clause 18.2.5.1
1 1 1 1	1 1 1 0	FE	PRBS test signal mapping, see clause 18.2.5.2
1 1 1 1	1 1 1 1	FF	Not available (Note 2)

NOTE 1 – There are 228 spare codes left for future international standardization. Refer to Annex A of [ITU-T G.806] for the procedure to obtain one of these codes for a new payload type.

NOTE 2 – These values are excluded from the set of available code points. These bit patterns are present in ODUk maintenance signals.

NOTE 3 – Value "01" is only to be used for experimental activities in cases where a mapping code is not defined in the above table. Refer to Annex A of [ITU-T G.806] for more information on the use of this code.

NOTE 4 – These 16 code values will not be subject to further standardization. Refer to Annex A of [ITU-T G.806] for more information on the use of these codes.

18.1.2.2.1.2 OPUk-Xv payload structure identifier reserved overhead (RES)

253 bytes plus 7 bits are reserved in the OPUk PSI for future international standardization. These bytes and bits are located in PSI[2] to PSI[255] of the OPUk overhead. These bytes are set to all-0s.

18.1.2.2.1.3 OPUk-Xv client signal fail (CSF)

For support of local management systems, a single-bit OPUk-Xv client signal fail (CSF) indicator is defined to convey the signal fail status of the client signal mapped into an OPUk-Xv at the ingress of the OTN to the egress of the OTN.

OPUk-Xv CSF is located in bit 1 of the PSI[2] byte of the payload structure identifier. Bits 2 to 8 of the PSI[2] byte are reserved for future international standardization. These bits are set to all-0s.

OPUk-Xv CSF is set to "1" to indicate a client signal fail indication, otherwise it is set to "0".

NOTE – Equipment designed prior to this revision of the Recommendation will generate a "0" in the OPUk-Xv CSF and will ignore any value in OPUk-Xv CSF.

18.1.2.2.2 OPUk-Xv virtual concatenation overhead (VCOH1/2/3)

Three bytes per individual OPUk of the OPUk-Xv are used to transport an 8 × 3 byte × 32 frame structure for a virtual concatenation specific overhead. These bytes are located in rows 1, 2 and 3 of column 15 as shown in Figure 18-2.

The structure is aligned with the ODUk multiframe and locked to bits 4, 5, 6, 7 and 8 of the MFAS. The structure is repeated 8 times in the 256-frame multiframe.

The structure is used to transport multiframe sequences and the LCAS control overhead.

18.1.2.2.2.1 OPUk-Xv virtual concatenation multiframe indicator (MFI1, MFI2)

A two-stage multiframe is introduced to cover a differential delay measurement (between the member signals within the virtual concatenated group) and compensation (of those differential delays) by the realignment process within the receiver.

The first stage uses MFAS in the frame alignment overhead area for the 8-bit multiframe indicator. MFAS is incremented every ODUk frame and counts from 0 to 255.

The second stage uses the MFI1 and MFI2 overhead bytes in the VCOH. They form a 16-bit multiframe counter with the MSBs in MFI1 and the LSBs in MFI2.

MFI1 is located in VCOH1[0] and MFI2 in VCOH1[1].

The multiframe counter of the second stage counts from 0 to 65535 and is incremented at the start of each multiframe of the first stage (MFAS = 0).

The resulting overall multiframe (a combination of 1st multiframe and 2nd multiframe counter) is 16 777 216 ODUk frames long.

At the start of the OPUk-Xv the multiframe sequence of all individual OPUks of the OPUk-Xv is identical.

The realignment process has to be able to compensate a differential delay of at least 125 µs.

18.1.2.2.2.2 OPUk-Xv sequence indicator (SQ)

The sequence indicator SQ identifies the sequence/order in which the individual OPUks of the OPUk-Xv are combined to form the contiguous OPUk-X-PLD as shown in Figure 18-1.

The 8-bit sequence number SQ (which supports values of X up to 256) is transported in VCOH1[4]. Bit 1 of VCOH1[4] is the MSB, bit 8 is the LSB.

Each OPUk of an OPUk-Xv has a fixed unique sequence number in the range of 0 to (X–1). The OPUk transporting the first time slot of the OPUk-Xv has the sequence number 0, the OPUk transporting the second time slot has the sequence number 1 and so on up to the OPUk transporting time slot X of the OPUk-Xv with the sequence number (X–1).

For applications requiring a fixed bandwidth the sequence number is fixed assigned and not configurable. This allows the constitution of the OPUk-Xv either to be checked without using the trace, or to be transported via a number of ODUk signals which have their trail termination functions as being part of an ODUk trail termination function resource group.

Refer to [ITU-T G.7042] for use and operation.

18.1.2.2.2.3 OPUk-Xv LCAS control words (CTRL)

The LCAS control word (CTRL) is located in bits 1 to 4 of VCOH1[5]. Bit 1 of VCOH1[5] is the MSB, bit 4 is the LSB.

Refer to [ITU-T G.7042] for the LCAS control commands, their coding and operation.

18.1.2.2.2.4 OPUk-Xv LCAS member status field (MST)

The LCAS member status field (MST) reports the status of the individual OPUks of the OPUk-Xv.

One bit is used per OPUk to report the status from sink to source. VCOH2[0] to VCOH2[31] are used as shown in Figure 18-2. Refer to [ITU-T G.7042] for coding and operation.

The status of all members (256) is transferred in 1567 µs (k = 1), 390 µs (k = 2) and 97 µs (k = 3).

18.1.2.2.2.5 OPUk-Xv LCAS group identification (GID)

The LCAS group identification (GID) provides the receiver with a means of verifying that all the arriving channels originated from one transmitter. Refer to [ITU-T G.7042] for coding and operation.

Bit 5 of VCOH1[5] is used for the GID.

18.1.2.2.2.6 OPUk-Xv LCAS re-sequence acknowledge (RS-Ack)

Re-sequence acknowledge, an indication from sink to source that a re-sequence, a sequence increase or a sequence decrease has been detected. Refer to [ITU-T G.7042] for coding and operation.

Bit 6 of VCOH1[5] is used for the RS-Ack.

18.1.2.2.2.7 OPUk-Xv LCAS cyclic redundancy check (CRC)

An 8-bit CRC check for fast acceptance of VirtConc LCAS OH is provided. The CRC-8 is calculated over VCOH1 and VCOH2 on a frame per frame basis and inserted into VCOH3. The CRC_8 Polynomial is $x^8 + x^3 + x^2 + 1$. Refer to [ITU-T G.7042] for operation.

18.1.2.2.2.8 OPUk-Xv VCOH reserved overhead

The reserved VCOH is set to all-0s.

18.1.2.2.3 OPUk mapping specific overhead

X times four bytes are reserved in the OPUk overhead for the mapping specific overhead. These bytes are located in columns 15X+1 to 16X.

The use of these bytes depends on the specific client signal mapping (defined in clause 18.2).

18.2 Mapping of client signals

18.2.1 Mapping of CBR signals (e.g., STM-64/256) into OPUk-4v

Mapping of a CBR signal (with up to ±20 ppm bit-rate tolerance) into an OPUk-4v may be performed according to two different modes (asynchronous and bit-synchronous) based on one generic OPUk-4v frame structure (see Figure 18-3).

NOTE 1 – Examples of such signals are STM-64 and STM-256.

NOTE 2 – The maximum bit-rate tolerance between OPUk-4v and the client signal clock, which can be accommodated by this mapping scheme, is ±65 ppm. With a bit-rate tolerance of ±20 ppm for the OPUk-4v clock, the client signal's bit-rate tolerance can be ±45 ppm.

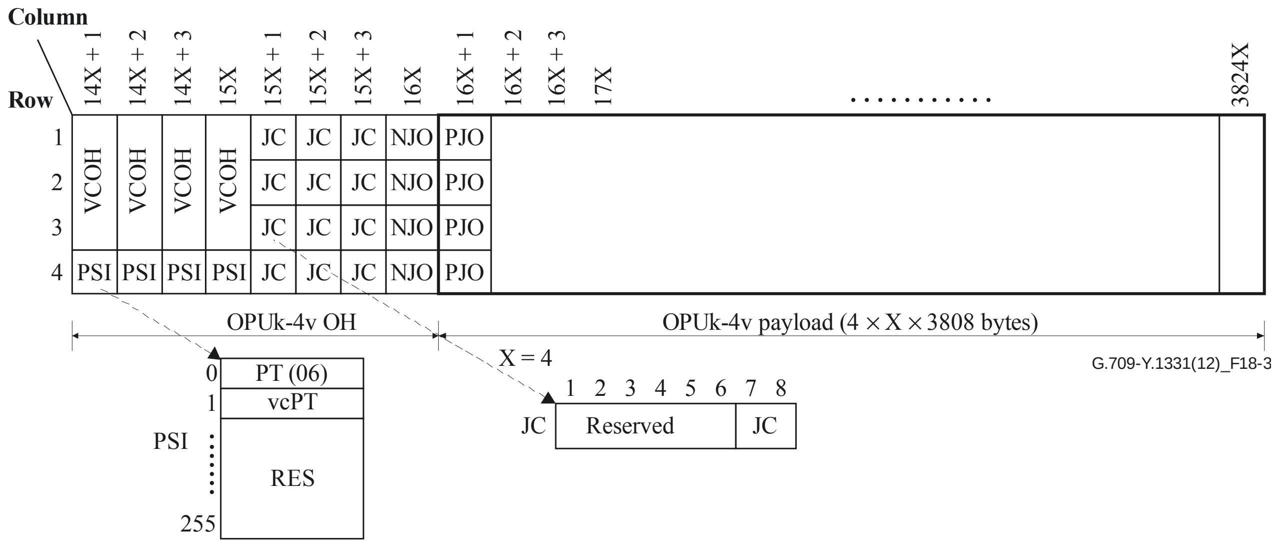


Figure 18-3 – OPUk-4v frame structure for the mapping of a CBR10G or CBR40G signal

The OPUk-4v overhead for these mappings consists of an X (X = 4) times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times the virtual concatenation overhead (VCOH), three justification control (JC) bytes and one negative justification opportunity (NJO) byte per row. The JC bytes consist of two bits for justification control and six bits reserved for future international standardization.

The OPUk-4v payload for these mappings consists of X (X = 4) times 4 × 3808 bytes, including one positive justification opportunity (PJO) byte per row.

The justification control (JC) signals, which are located in columns 15X+1 (61), 15X+2 (62) and 15X+3 (63) of each row, bits 7 and 8, are used to control the two justification opportunity fields NJO and PJO that follow in column 16X (64) and 16X+1 (65) of each row.

The asynchronous and bit-synchronous mapping processes generate the JC, NJO and PJO according to Tables 17-1 and 17-2, respectively. The de-mapping process interprets JC, NJO and PJO according to Table 17-3. Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

The value contained in NJO and PJO when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming CBR client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the generic-AIS signal as specified in clause 16.6.1, and is then mapped into the OPUk-4v.

During a signal fail condition of the incoming ODUk/OPUk-4v signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the generic-AIS pattern as specified in clause 16.6.1 is generated as a replacement signal for the lost CBR signal.

Asynchronous mapping

The OPUk-4v signal for the asynchronous mapping is created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the CBR (i.e., $4^{(k)}$ × 2 488 320 kbit/s) client signal.

The CBR (i.e., $4^{(k)}$ × 2 488 320 kbit/s) signal is mapped into the OPUk-4v using a positive/negative/zero (pnz) justification scheme.

Bit-synchronous mapping

The OPUk-4v clock for bit-synchronous mapping is derived from the CBR (i.e., $4^{(k)} \times 2\,488\,320$ kbit/s) client signal. During signal fail conditions of the incoming CBR signal (e.g., in the case of a loss of input signal), the OPUk-4v payload signal bit rate shall be within the limits specified in Table 7-3 and neither a frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming CBR signal shall be done without introducing a frequency or frame phase discontinuity.

The CBR (i.e., $4^{(k)} \times 2\,488\,320$ kbit/s) signal is mapped into the OPUk-4v without using the justification capability within the OPUk-Xv frame: NJO contains four justification bytes, PJO contains four data bytes, and the JC signal is fixed to 00.

18.2.1.1 Mapping a CBR10G signal (e.g., STM-64) into OPU1-4v

Groups of eight successive bits (not necessarily being a byte) of the CBR10G signal are mapped into a data (D) byte of the OPU1-4v (see Figure 18-4). Once per OPU1-4v row (and thus four times per OPU1-4v frame), it is possible to perform either a positive or a negative justification action.

X = 4

	14X + 1	14X + 2	14X + 3	15X	15X + 1	15X + 2	15X + 3	16X	16X + 1	16X + 2, 16X + 3, 17X … 3824X
1	VCOH	VCOH	VCOH	VCOH	JC	JC	JC	NJO	PJO	4 × 3808D – 1
2					JC	JC	JC	NJO	PJO	4 × 3808D – 1
3					JC	JC	JC	NJO	PJO	4 × 3808D – 1
4	PSI	PSI	PSI	PSI	JC	JC	JC	NJO	PJO	4 × 3808D – 1

G.709-Y.1331(12)_F18-4

Figure 18-4 – Mapping of a CBR10G signal into OPU1-4v

18.2.1.2 Mapping a CBR40G signal (e.g., STM-256) into OPU2-4v

Groups of eight successive bits (not necessarily being a byte) of the CBR40G signal are mapped into a data (D) byte of the OPU2-4v (see Figure 18-5). X times 64 fixed stuff (FS) bytes are added in columns 1904X+1 to 1920X. Once per OPU2-Xv row (and thus four times per OPU2-4v frame), it is possible to perform either a positive or a negative justification action.

	14X + 1	14X + 2	14X + 3	15X	15X + 1	15X + 2	15X + 3	16X	16X + 1	16X + 2, 16X + 3, 17X 1904X	1904X + 1 … 1920X	1920X + 1 … 3824X
1	VCOH	VCOH	VCOH	VCOH	JC	JC	JC	NJO	PJO	4 × 118 × 16D – 1	4 × 16FS	4 × 119 × 16D
2					JC	JC	JC	NJO	PJO	4 × 118 × 16D – 1	4 × 16FS	4 × 119 × 16D
3					JC	JC	JC	NJO	PJO	4 × 118 × 16D – 1	4 × 16FS	4 × 119 × 16D
4	PSI	PSI	PSI	PSI	JC	JC	JC	NJO	PJO	4 × 118 × 16D – 1	4 × 16FS	4 × 119 × 16D

G.709-Y.1331(12)_F18-5

Figure 18-5 – Mapping of a CBR40G signal into OPU2-4v

18.2.2 Mapping of CBR signals (e.g., STM-256) into OPUk-16v

Mapping of a CBR signal (with up to ±20 ppm bit-rate tolerance) into an OPUk-16v may be performed according to two different modes (asynchronous and bit-synchronous) based on one generic modified OPUk-16v frame structure (see Figure 18-6). This modified OPUk-16v frame structure has part of its OPUk-16v OH distributed over the frame; consequently, columns 15X+5 to 16X are now within the OPUk-16v payload area.

NOTE 1 – Examples of such signals are STM-256.

NOTE 2 – The maximum bit-rate tolerance between OPUk-16v and the client signal clock, which can be accommodated by this mapping scheme, is ±65 ppm. With a bit-rate tolerance of ±20 ppm for the OPUk-16v clock, the client signal's bit-rate tolerance can be ±45 ppm.

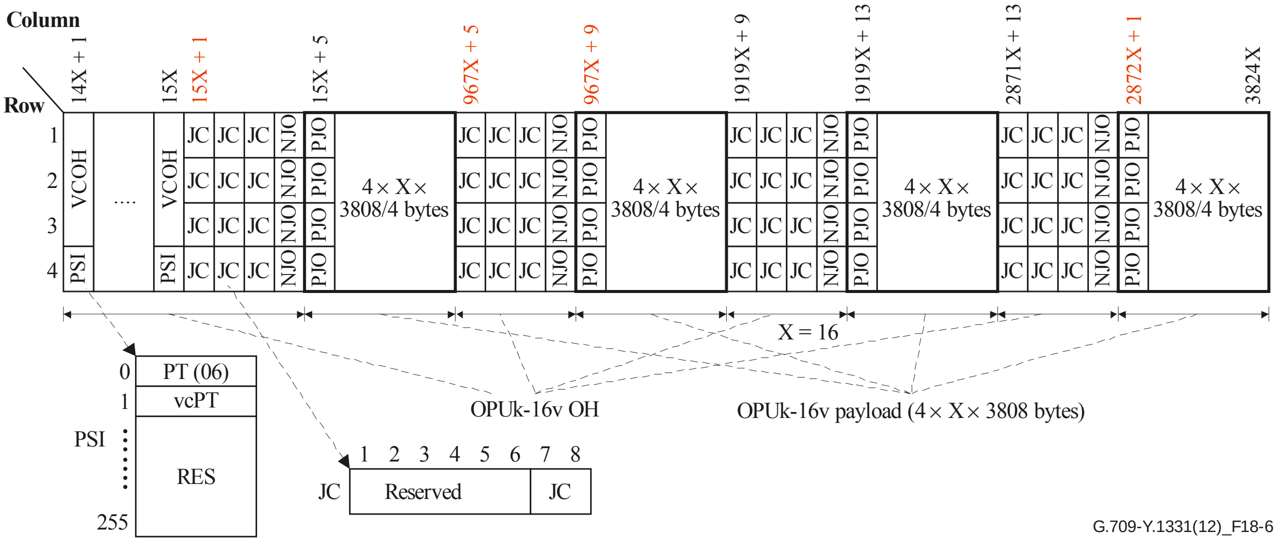


Figure 18-6 – OPUk-16v frame structure for the mapping of a CBR signal

The OPUk-16v overhead for these mappings consists of an X (X = 16) times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times the virtual concatenation overhead (VCOH), 4 × 3 justification control (JC) bytes and 4 × 1 negative justification opportunity (NJO) bytes per row. The JC bytes consist of two bits for justification control and six bits reserved for future international standardization.

The OPUk-16v payload for these mappings consists of 4 blocks of 4 × 15232 bytes, including 4 × 1 positive justification opportunity (PJO) bytes per row.

The justification control (JC) signals, which are located in the locations indicated in Figure 18-3, bits 7 and 8, are used to control the two justification opportunity fields NJO and PJO that follow in the next two columns of each row.

The asynchronous and bit-synchronous mapping processes generate the JC, NJO and PJO according to Tables 17-1 and 17-2, respectively. The de-mapping process interprets JC, NJO and PJO according to Table 17-3. Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

The value contained in NJO and PJO when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming CBR client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the generic-AIS signal as specified in clause 16.6.1, and is then mapped into the OPUk-16v.

During a signal fail condition of the incoming ODUk/OPUk-16v signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the generic-AIS pattern as specified in clause 16.6.1 is generated as a replacement signal for the lost CBR signal.

Asynchronous mapping

The OPUk-16v signal for the asynchronous mapping is created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the CBR (i.e., $4^{(k+1)}$ × 2 488 320 kbit/s) client signal.

The CBR (i.e., $4^{(k+1)} \times 2\,488\,320$ kbit/s) signal is mapped into the OPUk-16v using a positive/negative/zero (pnz) justification scheme.

Bit-synchronous mapping

The OPUk-16v clock for the bit-synchronous mapping is derived from the CBR client signal. During signal fail conditions of the incoming CBR signal (e.g., in the case of a loss of input signal), the OPUk-16v payload signal bit rate shall be within the limits specified in Table 7-3 and neither a frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming CBR signal shall be done without introducing a frequency or frame phase discontinuity.

The CBR (i.e., $4^{(k+1)} \times 2\,488\,320$ kbit/s) signal is mapped into the OPUk-16v without using the justification capability within the OPUk-16v frame: NJO contains four justification bytes, PJO contains four data bytes, and the JC signal is fixed to 00.

18.2.2.1 Mapping a CBR40G signal (e.g., STM-256) into OPU1-16v

Groups of eight successive bits (not necessarily being a byte) of the CBR40G signal are mapped into a data (D) byte of the OPU1-16v (see Figure 18-7). Four times per OPU1-16v row (and thus sixteen times per OPU1-16v frame), it is possible to perform either a positive or a negative justification action.

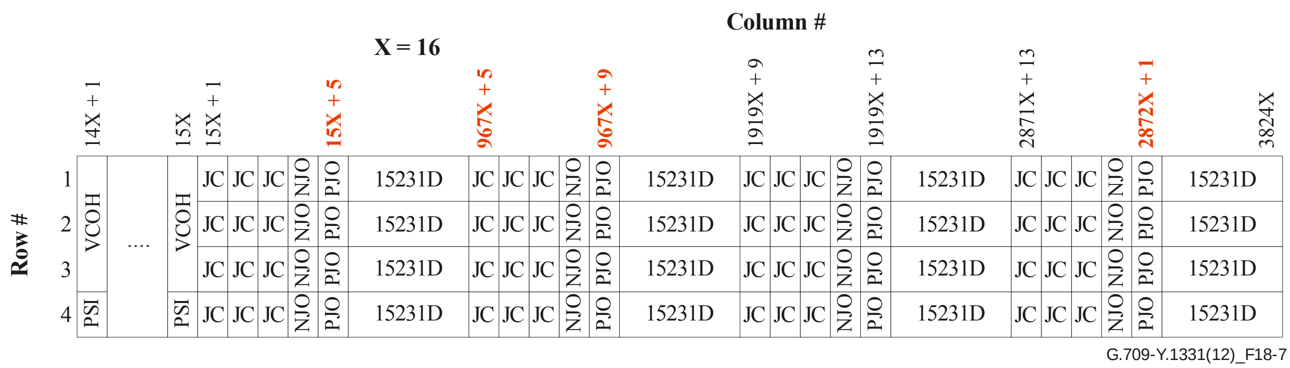


Figure 18-7 – Mapping of a CBR40G signal into OPU1-16v

18.2.3 Mapping of an ATM cell stream into OPUk-Xv

A constant bit rate ATM cell stream with a capacity that is identical to the OPUk-Xv payload area is created by multiplexing the ATM cells of a set of ATM VP signals. Rate adaptation is performed as part of this cell stream creation process by either inserting idle cells or by discarding cells. Refer to [ITU-T I.432.1]. The ATM cell stream is mapped into the OPUk-Xv payload area with the ATM cell byte structure aligned to the OPUk-Xv payload byte structure (see Figure 18-8). The ATM cell boundaries are thus aligned with the OPUk-Xv payload byte boundaries. Since the OPUk-Xv payload capacity ($X \times 15232$ bytes) is not an integer multiple of the cell length (53 bytes), a cell may cross an OPUk-Xv frame boundary.

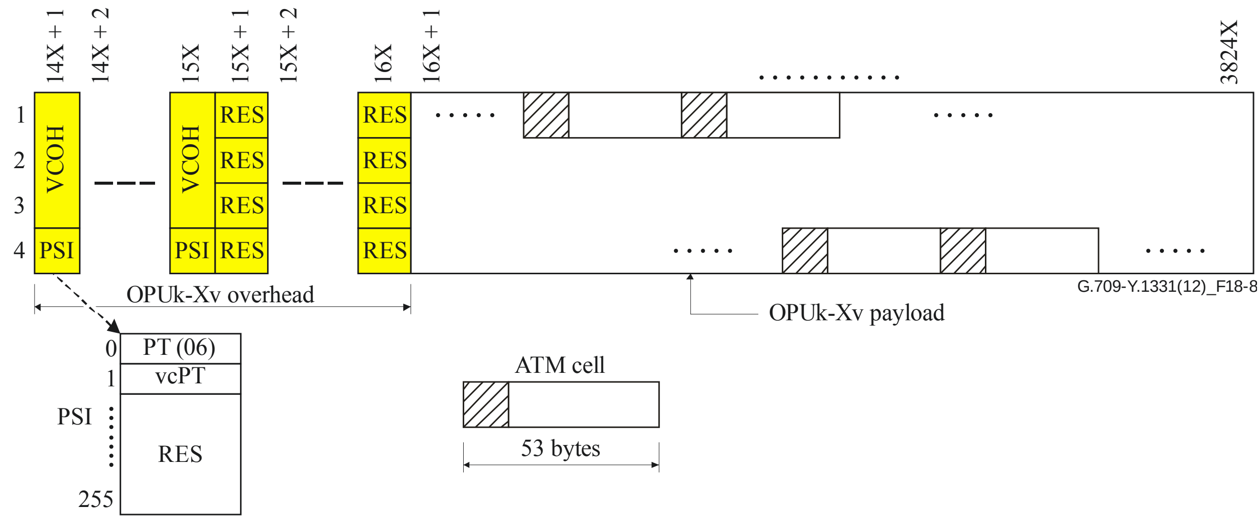


Figure 18-8 – OPUk-Xv frame structure and mapping of ATM cells into OPUk-Xv

The ATM cell information field (48 bytes) shall be scrambled before mapping into the OPUk-Xv. In the reverse operation, following termination of the OPUk-Xv signal, the ATM cell information field will be descrambled before being passed to the ATM layer. A self-synchronizing scrambler with generator polynomial $x^{43} + 1$ shall be used (as specified in [ITU-T I.432.1]). The scrambler operates for the duration of the cell information field. During the 5-byte header the scrambler operation is suspended and the scrambler state retained. The first cell transmitted on start-up will be corrupted because the descrambler at the receiving end will not be synchronized to the transmitter scrambler. Cell information field scrambling is required to provide security against false cell delineation and the cell information field replicating the OTUk and ODUk frame alignment signal.

When extracting the ATM cell stream from the OPUk-Xv payload area after the ODUk terminations, the ATM cells must be recovered. The ATM cell header contains a header error control (HEC) field, which may be used in a similar way to a frame alignment word to achieve cell delineation. This HEC method uses the correlation between the header bits to be protected by the HEC (32 bits) and the control bit of the HEC (8 bits) introduced in the header after computation with a shortened cyclic code with generating polynomial $g(x) = x^8 + x^2 + x + 1$.

The remainder from this polynomial is then added to the fixed pattern "01010101" in order to improve the cell delineation performance. This method is similar to conventional frame alignment recovery where the alignment signal is not fixed but varies from cell to cell.

More information on HEC cell delineation is given in [ITU-T I.432.1].

The OPUk-Xv overhead for the ATM mapping consists of X times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times three virtual concatenation overhead (VCOH) bytes and X times four bytes reserved for future international standardization (RES).

The OPUk-Xv payload for the ATM mapping consists of 4X × 3808 bytes.

18.2.4 Mapping of GFP frames into OPUk-Xv

The mapping of generic framing procedure (GFP) frames is performed by aligning the byte structure of every GFP frame with the byte structure of the OPUk-Xv payload (see Figure 18-9). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a GFP frame may cross the OPUk frame boundary. A GFP frame consists of a GFP header and a GFP payload area.

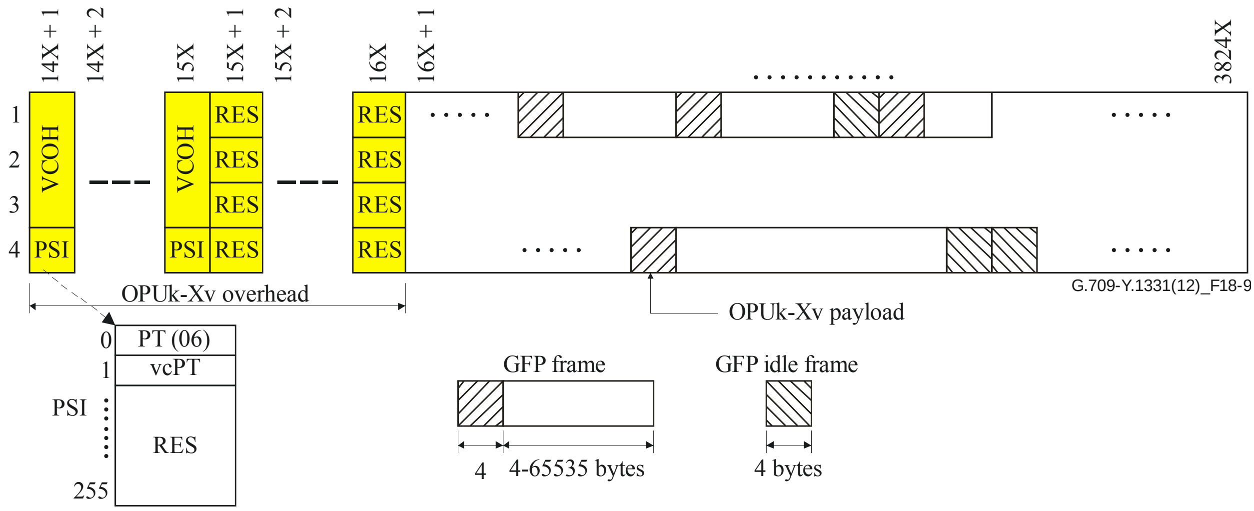


Figure 18-9 – OPUk-Xv frame structure and mapping of GFP frames into OPUk-Xv

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPUk-Xv payload area, due to the insertion of GFP idles at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPUk-Xv overhead for the GFP mapping consists of X times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times three virtual concatenation overhead (VCOH) bytes and X times four bytes reserved for future international standardization (RES).

The OPUk-Xv payload for the GFP mapping consists of 4X × 3808 bytes.

18.2.5 Mapping of a test signal into OPUk-Xv

18.2.5.1 Mapping of a NULL client into OPUk-Xv

An OPUk-Xv payload signal with an all-0s pattern (see Figure 18-10) is defined for test purposes. This is referred to as the NULL client.

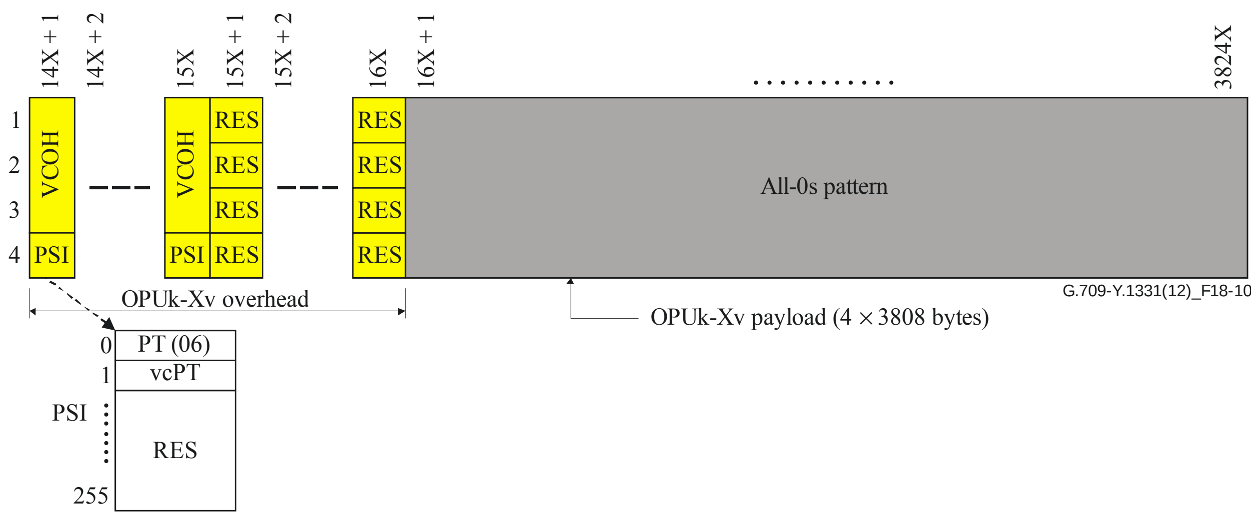


Figure 18-10 – OPUk-Xv frame structure and mapping of the NULL client into OPUk-Xv

The OPUk-Xv overhead for the NULL mapping consists of X times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times three virtual concatenation overhead (VCOH) bytes and X times four bytes reserved for future international standardization (RES).

The OPUk-Xv payload for the NULL mapping consists of 4X × 3808 bytes.

18.2.5.2 Mapping of PRBS test signal into OPUk-Xv

For test purposes, a 2 147 483 647-bit pseudo-random test sequence ($2^{31} - 1$) as specified in clause 5.8 of [ITU-T O.150] can be mapped into the OPUk-Xv payload. Groups of eight successive bits of the 2 147 483 647-bit pseudo-random test sequence signal are mapped into eight data bits (8D) (i.e., one byte) of the ODU3 payload (see Figure 18-11).

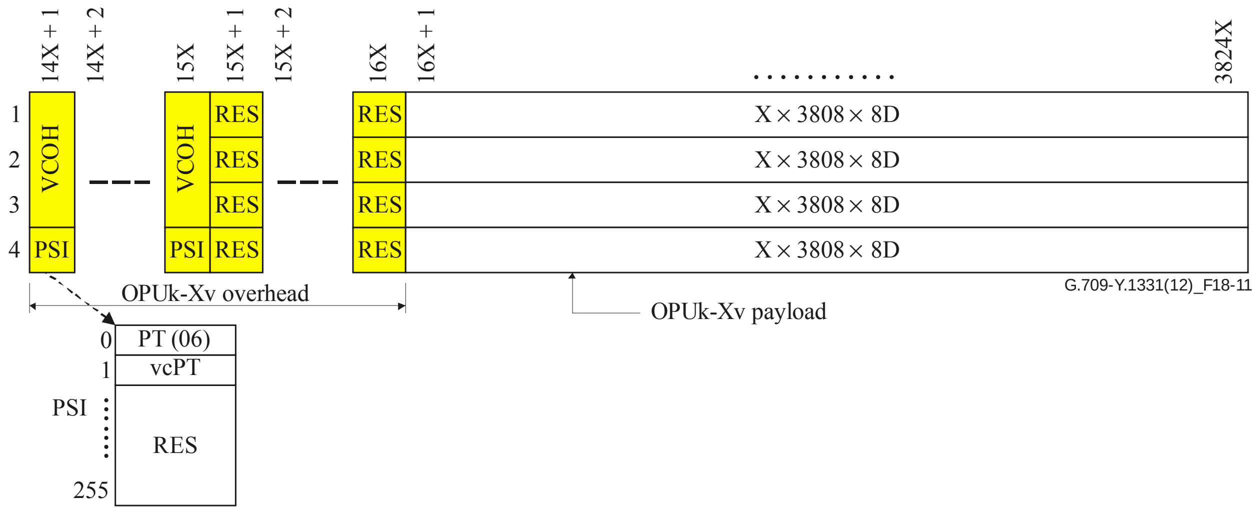


Figure 18-11 – OPUk-Xv frame structure and mapping of 2 147 483 647-bit pseudo-random test sequence into OPUk-Xv

The OPUk-Xv overhead for the PRBS mapping consists of X times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times three virtual concatenation overhead (VCOH) bytes and X times four bytes reserved for future international standardization (RES).

The OPUk-Xv payload for the PRBS mapping consists of 4X × 3808 bytes.

18.2.6 Mapping of a non-specific client bit stream into OPUk-Xv

In addition to the mappings of specific client signals as specified in the other subclauses of this clause, a non-specific client mapping into OPUk-Xv is specified. Any (set of) client signal(s), which after encapsulation into a continuous bit stream with a bit rate of the OPUk-Xv payload can be mapped into the OPUk-Xv payload (see Figure 18-12). The bit stream must be synchronous with the OPUk-Xv signal. Any justification must be included in the continuous bit stream creation process. The continuous bit stream must be scrambled before mapping into the OPUk-Xv payload.

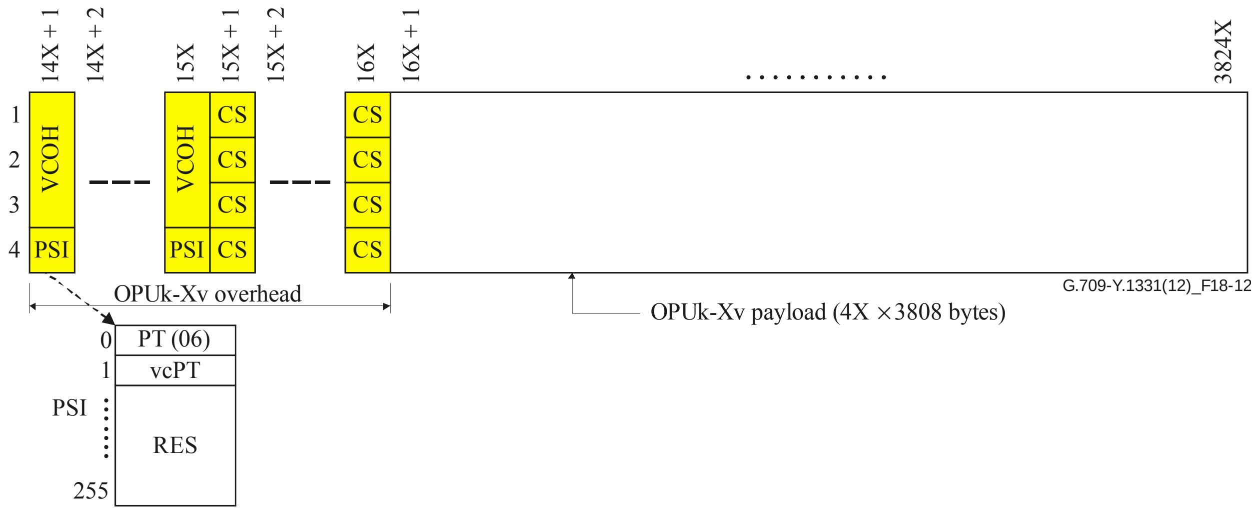


Figure 18-12 – OPUk-Xv frame structure for the mapping of a synchronous constant bit stream

The OPUk-Xv overhead for the mapping consists of X times a payload structure identifier (PSI), which includes the payload type (PT) and virtual concatenation payload type (vcPT), X times three virtual concatenation overhead (VCOH) bytes and X times four bytes for client specific purposes (CS). The definition of these CS overhead bytes is performed within the encapsulation process specification.

The OPUk-Xv payload for this non-specific mapping consists of 4X × 3808 bytes.

18.2.6.1 Mapping bit stream with octet timing into OPUk-Xv

If octet timing is available, each octet of the incoming data stream will be mapped into a data byte (octet) of the OPUk-Xv payload.

18.2.6.2 Mapping bit stream without octet timing into OPUk-Xv

If octet timing is not available, groups of eight successive bits (not necessarily an octet) of the incoming data stream will be mapped into a data byte (octet) of the OPUk-Xv payload.

18.3 LCAS for virtual concatenation

Refer to [ITU-T G.7042].

19 Mapping ODUj signals into the ODTU signal and the ODTU into the HO OPUk tributary slots

This clause specifies the multiplexing of:

– ODU0 into HO OPU1, ODU1 into HO OPU2, ODU1 and ODU2 into HO OPU3 using client/server specific asynchronous mapping procedures (AMP);

– other ODUj into HO OPUk using a client agnostic generic mapping procedure (GMP).

This ODUj into HO OPUk multiplexing is performed in two steps:

1) asynchronous mapping of ODUj into optical channel data tributary unit (ODTU) using either AMP or GMP;

2) byte-synchronous mapping of ODTU into one or more HO OPUk tributary slots.

19.1 OPUk tributary slot definition

The OPUk is divided into a number of tributary slots (TS) and these tributary slots are interleaved within the OPUk. A tributary slot includes a part of the OPUk OH area and a part of the OPUk payload area. The bytes of the ODUj frame are mapped into the ODTU payload area and the ODTU bytes are mapped into the OPUk tributary slot or slots. The bytes of the ODTU justification overhead are mapped into the OPUk OH area.

There are two types of tributary slots:

1) Tributary slot with a bandwidth of approximately 2.5 Gbit/s; an OPUk is divided into n tributary slots, numbered 1 to n.

2) Tributary slot with a bandwidth of approximately 1.25 Gbit/s; an OPUk is divided into 2n tributary slots, numbered 1 to 2n.

HO OPU2 and HO OPU3 interface ports supporting 1.25 Gbit/s tributary slots must also support the 2.5 Gbit/s tributary slot mode for interworking with interface ports supporting only the 2.5G tributary slot mode (i.e., interface ports compliant with issues of this Recommendation: prior to the definition of 1.25G tributary slots). When operated in 2.5G tributary slot mode, 1.25G tributary slots "i" and "i+n" (i = 1 to n, n = 4 (OPU2) and n = 16 (OPU3)) function as one 2.5G tributary slot.

19.1.1 OPU2 tributary slot allocation

Figure 19-1 presents the OPU2 2.5G tributary slot allocation and the OPU2 1.25G tributary slot allocation. An OPU2 is divided into four 2.5G tributary slots numbered 1 to 4, or in eight 1.25G tributary slots numbered 1 to 8.

– An OPU2 2.5G tributary slot occupies 25% of the OPU2 payload area. It is a structure with 952 columns by 16 (4 × 4) rows (see Figures 19-1 and 19-7) plus a tributary slot overhead (TSOH). The four OPU2 TSs are byte interleaved in the OPU2 payload area and the four OPU2 TSOHs are frame interleaved in the OPU2 overhead area.

– An OPU2 1.25G tributary slot occupies 12.5% of the OPU2 payload area. It is a structure with 476 columns by 32 (8 × 4) rows (see Figures 19-1 and 19-7) plus a tributary slot overhead (TSOH). The eight OPU2 TSs are byte interleaved in the OPU2 payload area and the eight OPU2 TSOHs are frame interleaved in the OPU2 overhead area.

An OPU2 2.5G tributary slot "i" (i = 1,2,3,4) is provided by two OPU2 1.25G tributary slots "i" and "i+4" as illustrated in Figure 19-1.

The tributary slot overhead (TSOH) of OPU2 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU2 frame.

The TSOH for a 2.5G tributary slot is available once every 4 frames. A 4-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 7 and 8 of the MFAS byte as shown in Table 19-1 and Figure 19-1.

The TSOH for a 1.25G tributary slot is available once every 8 frames. An 8-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 6, 7 and 8 of the MFAS byte as shown in Table 19-1 and Figure 19-1.

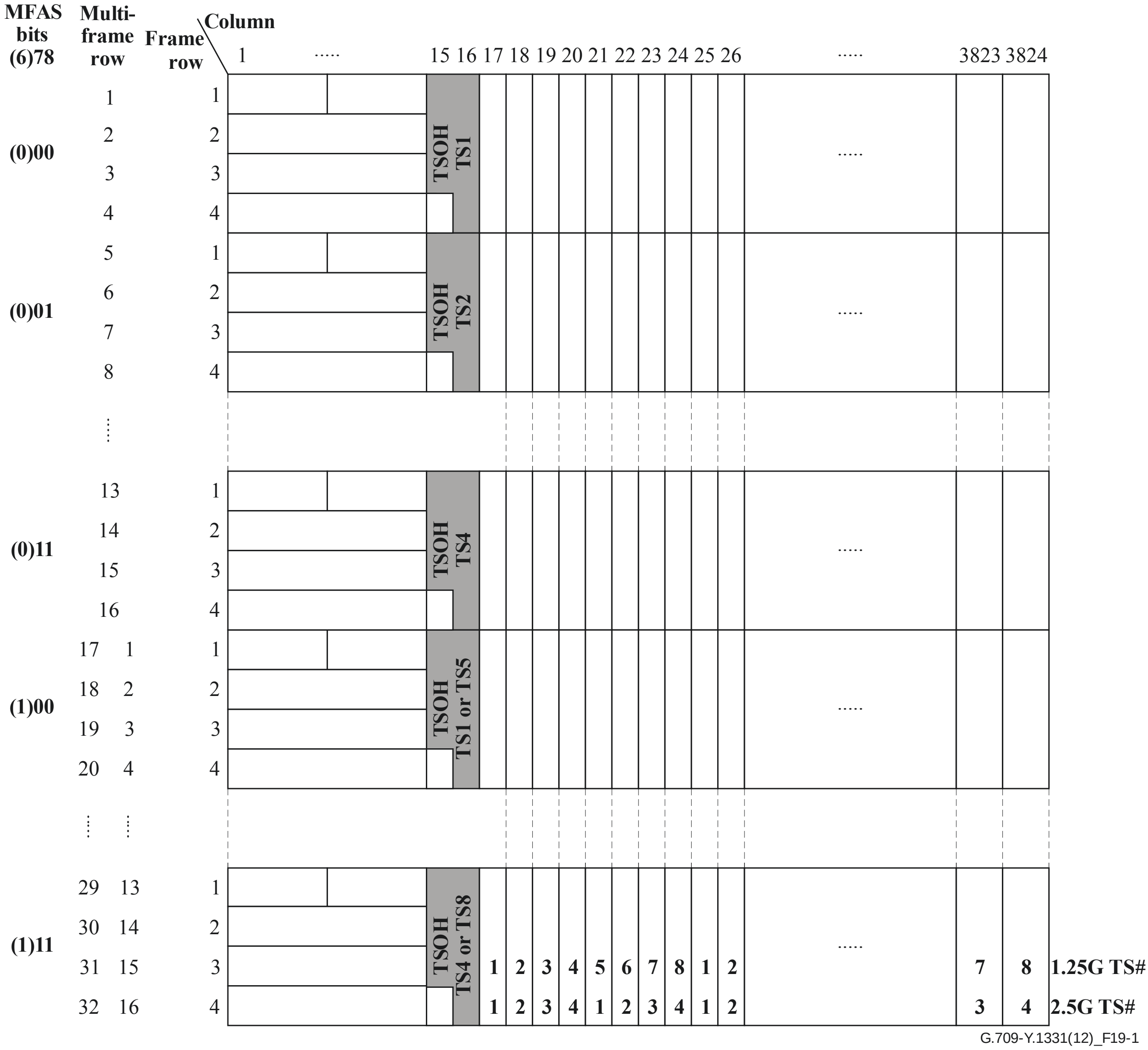


Figure 19-1 – OPU2 tributary slot allocation

Table 19-1 – OPU2 tributary slot OH allocation

MFAS bits 7 8	TSOH 2.5G TS
0 0	1
0 1	2
1 0	3
1 1	4

MFAS bits 6 7 8	TSOH 1.25G TS
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6
1 1 0	7
1 1 1	8

19.1.2 OPU3 tributary slot allocation

Figure 19-2 presents the OPU3 2.5G tributary slot allocation and the OPU3 1.25G tributary slot allocation. An OPU3 is divided into sixteen 2.5G tributary slots numbered 1 to 16, or in thirty-two 1.25G tributary slots numbered 1 to 32.

– An OPU3 2.5G tributary slot occupies 6.25% of the OPU3 payload area. It is a structure with 238 columns by 64 (16 × 4) rows (see Figures 19-2 and 19-8) plus a tributary slot overhead (TSOH). The sixteen OPU3 2.5G TSs are byte interleaved in the OPU3 payload area and the sixteen OPU3 TSOHs are frame interleaved in the OPU3 overhead area.
– An OPU3 1.25G tributary slot occupies 3.125% of the OPU3 payload area. It is a structure with 119 columns by 128 (32 × 4) rows (see Figures 19-2 and 19-8) plus a tributary slot overhead (TSOH). The thirty-two OPU3 1.25G TSs are byte interleaved in the OPU3 payload area and the thirty-two OPU3 TSOHs are frame interleaved in the OPU3 overhead area.

An OPU3 2.5G tributary slot "i" (i = 1,2,..16) is provided by two OPU3 1.25G tributary slots "i" and "i+16" as illustrated in Figure 19-2.

The tributary slot overhead (TSOH) of OPU3 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU3 frame.

The TSOH for a 2.5G tributary slot is available once every 16 frames. A 16-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 5, 6, 7 and 8 of the MFAS byte as shown in Table 19-2 and Figure 19-2.

The TSOH for a 1.25G tributary slot is available once every 32 frames. A 32-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 4, 5, 6, 7 and 8 of the MFAS byte as shown in Table 19-2 and Figure 19-2.

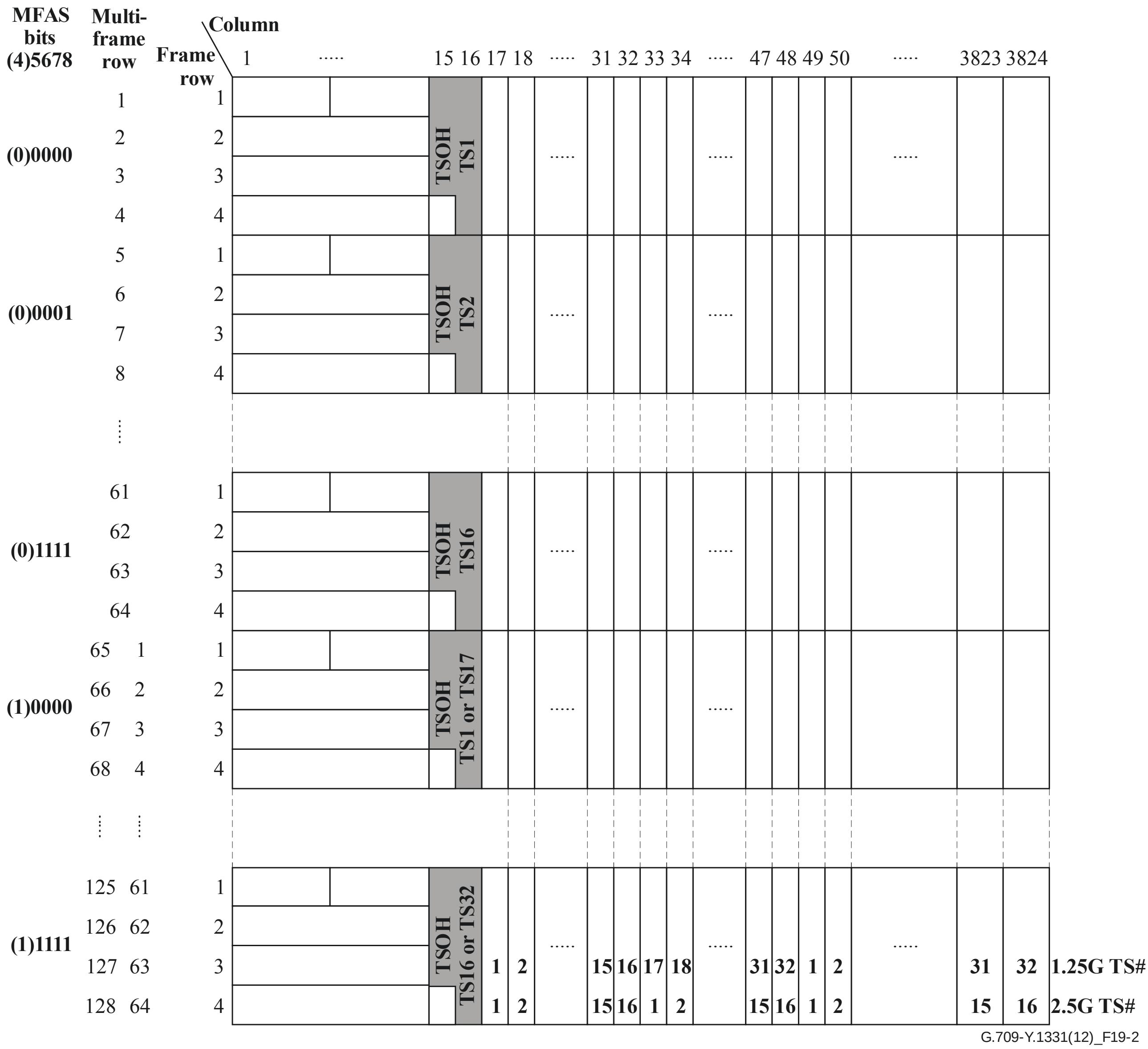


Figure 19-2 – OPU3 tributary slot allocation

Table 19-2 – OPU3 tributary slot OH allocation

MFAS bits 5 6 7 8	TSOH 2.5G TS	MFAS bits 4 5 6 7 8	TSOH 1.25G TS	MFAS bits 4 5 6 7 8	TSOH 1.25G TS
0 0 0 0	1	0 0 0 0 0	1	1 0 0 0 0	17
0 0 0 1	2	0 0 0 0 1	2	1 0 0 0 1	18
0 0 1 0	3	0 0 0 1 0	3	1 0 0 1 0	19
0 0 1 1	4	0 0 0 1 1	4	1 0 0 1 1	20
0 1 0 0	5	0 0 1 0 0	5	1 0 1 0 0	21
0 1 0 1	6	0 0 1 0 1	6	1 0 1 0 1	22
0 1 1 0	7	0 0 1 1 0	7	1 0 1 1 0	23
0 1 1 1	8	0 0 1 1 1	8	1 0 1 1 1	24
1 0 0 0	9	0 1 0 0 0	9	1 1 0 0 0	25
1 0 0 1	10	0 1 0 0 1	10	1 1 0 0 1	26
1 0 1 0	11	0 1 0 1 0	11	1 1 0 1 0	27
1 0 1 1	12	0 1 0 1 1	12	1 1 0 1 1	28
1 1 0 0	13	0 1 1 0 0	13	1 1 1 0 0	29
1 1 0 1	14	0 1 1 0 1	14	1 1 1 0 1	30
1 1 1 0	15	0 1 1 1 0	15	1 1 1 1 0	31
1 1 1 1	16	0 1 1 1 1	16	1 1 1 1 1	32

19.1.3 OPU1 tributary slot allocation

Figure 19-3 presents the OPU1 1.25G tributary slot allocation. An OPU1 is divided into two 1.25G tributary slots numbered 1 to 2.

– An OPU1 1.25G tributary slot occupies 50% of the OPU1 payload area. It is a structure with 1904 columns by 8 (2 × 4) rows (see Figure 19-3) plus a tributary slot overhead (TSOH). The two OPU1 1.25G TSs are byte interleaved in the OPU1 payload area and the two OPU1 TSOHs are frame interleaved in the OPU1 overhead area.

The tributary slot overhead (TSOH) of OPU1 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU1 frame.

The TSOH for a 1.25G tributary slot is available once every 2 frames. A 2-frame multiframe structure is used for this assignment. This multiframe structure is locked to bit 8 of the MFAS byte as shown in Table 19-3 and Figure 19-3.

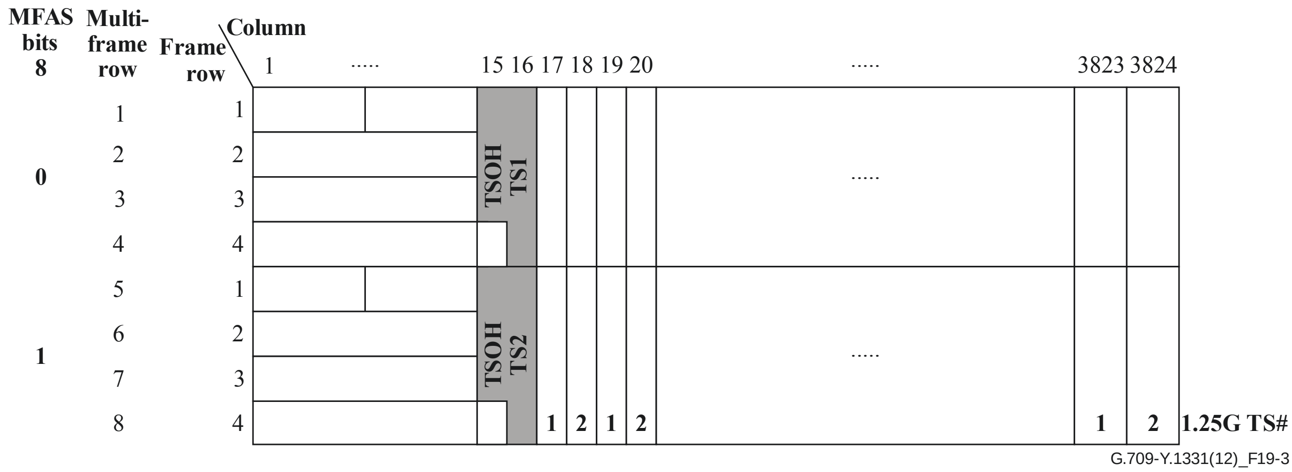


Figure 19-3 – OPU1 tributary slot allocation

Table 19-3 – OPU1 tributary slot OH allocation

MFAS bit 8	TSOH 1.25G TS
0	1
1	2

19.1.4 OPU4 tributary slot allocation

Figures 19-4A and 19-4B present the OPU4 1.25G tributary slot allocation. An OPU4 is divided into eighty 1.25G tributary slots (numbered 1 to 80), which are located in columns 17 to 3816, and 8 columns of fixed stuff located in columns 3817 to 3824. The OPU4 frame may be represented in a 320 row by 3810 column format (Figure 19-4A) and in a 160 row by 7620 column format (Figure 19-4B).

– An OPU4 1.25G tributary slot occupies 1.247% of the OPU4 payload area. It is a structure with 95 columns by 160 ($80 \times 4/2$) rows (see Figure 19-4B) plus a tributary slot overhead (TSOH). The eighty OPU4 1.25G TSs are byte interleaved in the OPU4 payload area and the eighty OPU4 TSOHs are frame interleaved in the OPU4 overhead area.

The tributary slot overhead (TSOH) of OPU4 tributary slots is located in rows 1 to 3, columns 15 and 16 of the OPU4 frame.

The TSOH for a 1.25G tributary slot is available once every 80 frames. An 80-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 2, 3, 4, 5, 6, 7 and 8 of the OMFI byte as shown in Table 19-4.

OMFI bits 2345678	Multi-frame row	Frame row \ Column	1	.....	15	16	17	18	.....	55	56	57	58	.....	95	96	97	98	.....	3815	3816	3817	3818	3819	3820	3821	3822	3823	3824
0000000	1	1			TSOH TS1		1	2	.....	39	40	41	42	.....	79	80	1	2	.....	39	40	FS	FS	FS	FS	FS	FS	FS	FS
	2	2					41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
	3	3					1	2		39	40	41	42		79	80	1	2		39	40	FS	FS	FS	FS	FS	FS	FS	FS
	4	4			PSI	OMFI	41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
0000001	5	1			TSOH TS2		1	2	.....	39	40	41	42	.....	79	80	1	28	.....	39	40	FS	FS	FS	FS	FS	FS	FS	FS
	6	2					41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
	7	3					1	2		39	40	41	42		79	80	1	2		39	40	FS	FS	FS	FS	FS	FS	FS	FS
	8	4			PSI	OMFI	41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
	⋮																												
1001110	313	1			TSOH TS79		1	2	.....	39	40	41	42	.....	79	80	1	2	.....	39	40	FS	FS	FS	FS	FS	FS	FS	FS
	314	2					41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
	315	3					1	2		39	40	41	42		79	80	1	2		39	40	FS	FS	FS	FS	FS	FS	FS	FS
	316	4			PSI	OMFI	41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
1001111	317	1			TSOH TS80		1	2	.....	39	40	41	42	.....	79	80	1	2	.....	39	40	FS	FS	FS	FS	FS	FS	FS	FS
	318	2					41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS
	319	3					1	2		39	40	41	42		79	80	1	2		39	40	FS	FS	FS	FS	FS	FS	FS	FS
	320	4			PSI	OMFI	41	42		79	80	1	2		39	40	41	42		79	80	FS	FS	FS	FS	FS	FS	FS	FS

G.709-Y.1331(12)_F19-4A

Figure 19-4A – OPU4 1.25G tributary slot allocation

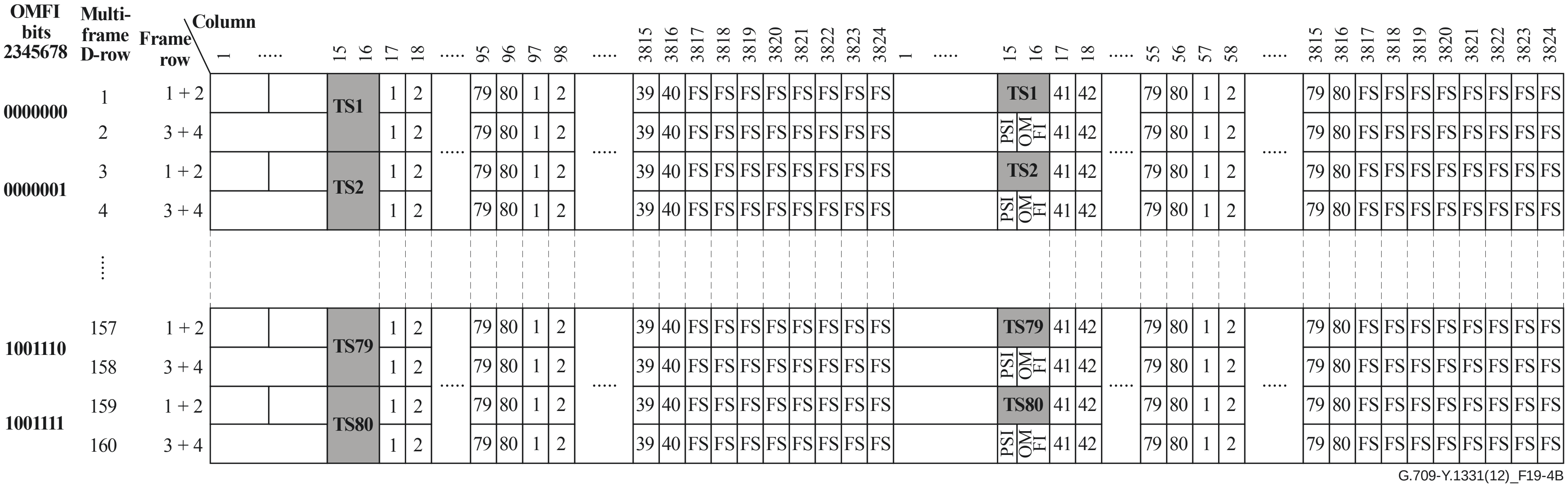


Figure 19-4B – OPU4 tributary slots in 160 row x 7620 column format

Table 19-4 – OPU4 tributary slot OH allocation

OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS
0 0 0 0 0 0 0	1	0 0 1 0 1 0 0	21	0 1 0 1 0 0 0	41	0 1 1 1 1 0 0	61
0 0 0 0 0 0 1	2	0 0 1 0 1 0 1	22	0 1 0 1 0 0 1	42	0 1 1 1 1 0 1	62
0 0 0 0 0 1 0	3	0 0 1 0 1 1 0	23	0 1 0 1 0 1 0	43	0 1 1 1 1 1 0	63
0 0 0 0 0 1 1	4	0 0 1 0 1 1 1	24	0 1 0 1 0 1 1	44	0 1 1 1 1 1 1	64
0 0 0 0 1 0 0	5	0 0 1 1 0 0 0	25	0 1 0 1 1 0 0	45	1 0 0 0 0 0 0	65
0 0 0 0 1 0 1	6	0 0 1 1 0 0 1	26	0 1 0 1 1 0 1	46	1 0 0 0 0 0 1	66
0 0 0 0 1 1 0	7	0 0 1 1 0 1 0	27	0 1 0 1 1 1 0	47	1 0 0 0 0 1 0	67
0 0 0 0 1 1 1	8	0 0 1 1 0 1 1	28	0 1 0 1 1 1 1	48	1 0 0 0 0 1 1	68
0 0 0 1 0 0 0	9	0 0 1 1 1 0 0	29	0 1 1 0 0 0 0	49	1 0 0 0 1 0 0	69
0 0 0 1 0 0 1	10	0 0 1 1 1 0 1	30	0 1 1 0 0 0 1	50	1 0 0 0 1 0 1	70
0 0 0 1 0 1 0	11	0 0 1 1 1 1 0	31	0 1 1 0 0 1 0	51	1 0 0 0 1 1 0	71
0 0 0 1 0 1 1	12	0 0 1 1 1 1 1	32	0 1 1 0 0 1 1	52	1 0 0 0 1 1 1	72
0 0 0 1 1 0 0	13	0 1 0 0 0 0 0	33	0 1 1 0 1 0 0	53	1 0 0 1 0 0 0	73
0 0 0 1 1 0 1	14	0 1 0 0 0 0 1	34	0 1 1 0 1 0 1	54	1 0 0 1 0 0 1	74
0 0 0 1 1 1 0	15	0 1 0 0 0 1 0	35	0 1 1 0 1 1 0	55	1 0 0 1 0 1 0	75
0 0 0 1 1 1 1	16	0 1 0 0 0 1 1	36	0 1 1 0 1 1 1	56	1 0 0 1 0 1 1	76
0 0 1 0 0 0 0	17	0 1 0 0 1 0 0	37	0 1 1 1 0 0 0	57	1 0 0 1 1 0 0	77
0 0 1 0 0 0 1	18	0 1 0 0 1 0 1	38	0 1 1 1 0 0 1	58	1 0 0 1 1 0 1	78
0 0 1 0 0 1 0	19	0 1 0 0 1 1 0	39	0 1 1 1 0 1 0	59	1 0 0 1 1 1 0	79
0 0 1 0 0 1 1	20	0 1 0 0 1 1 1	40	0 1 1 1 0 1 1	60	1 0 0 1 1 1 1	80

19.2 ODTU definition

The optical channel data tributary unit (ODTU) carries a justified ODU signal. There are two types of ODTUs:

1) ODTUjk ((j,k) = {(0,1), (1,2), (1,3), (2,3)}; ODTU01, ODTU12, ODTU13 and ODTU23) in which an ODUj signal is mapped via the asynchronous mapping procedure (AMP) as defined in clause 19.5;

2) ODTUk.ts ((k,ts) = (2,1..8), (3,1..32), (4,1..80)) in which a lower order ODU (ODU0, ODU1, ODU2, ODU2e, ODU3, ODUflex) signal is mapped via the generic mapping procedure (GMP) defined in clause 19.6.

Optical channel data tributary unit jk

The optical channel data tributary unit jk (ODTUjk) is a structure which consists of an ODTUjk payload area and an ODTUjk overhead area (Figure 19-5). The ODTUjk payload area has c columns and r rows (see Table 19-5) and the ODTUjk overhead area has "ts" times 4 bytes, of which "ts" times 1 byte can carry payload. The ODTUjk is carried in "ts" 1.25G or 2.5G tributary slots of an HO OPUk.

The location of the ODTUjk overhead depends on the OPUk tributary slot(s) used when multiplexing the ODTUjk in the OPUk (see clauses 19.1.1, 19.1.2, 19.1.3). The ts instances of the ODTUjk overhead might not be equally distributed.

The ODTUjk overhead carries the AMP justification overhead as specified in clause 19.4.

NOTE – The 1.25G and 2.5G tributary slot versions of an ODTU12 are identical when the two 1.25G tributary slots carrying the ODTU12 are TSa and TSa+4. The 1.25G and 2.5G tributary slot versions of an ODTU13 are identical when the two 1.25G tributary slots carrying the ODTU12 are TSa and TSa+16. The 1.25G and 2.5G tributary slot versions of an ODTU23 are identical when the eight 1.25G tributary slots carrying the ODTU23 are TSa, TSb, TSc, TSd, TSa+16, TSb+16, TSc+16 and TSd+16.

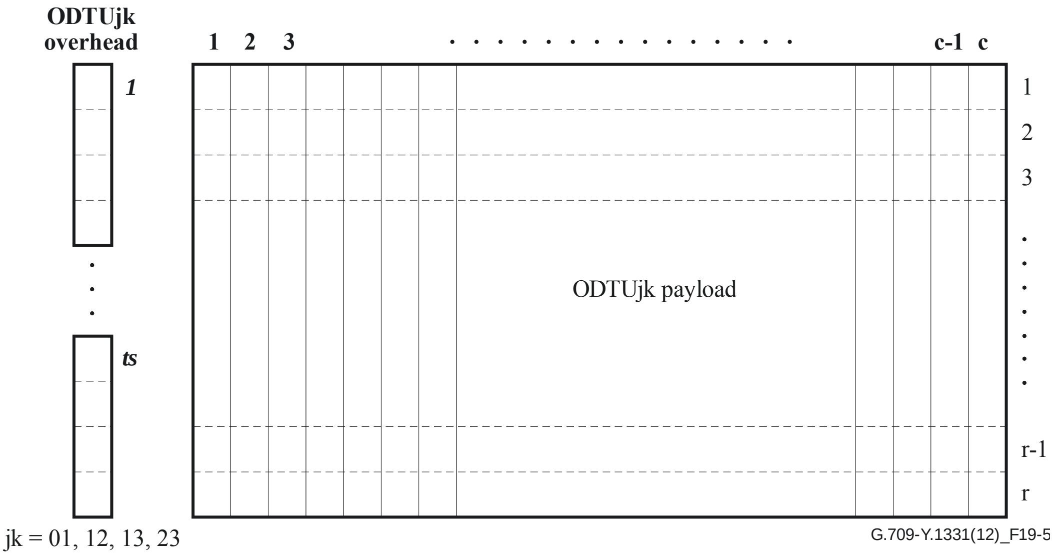


Figure 19-5 – ODTUjk frame formats

Table 19-5 – ODTUjk characteristics for 2.5G and 1.25G tributary slots

2.5G TS	c	r	ts	ODTUjk payload bytes	ODTUjk overhead bytes
ODTU12	952	16	1	15232	1 x 4
ODTU13	238	64	1	15232	1 x 4
ODTU23	952	64	4	60928	4 x 4

1.25G TS	c	r	ts	ODTUjk payload bytes	ODTUjk overhead bytes
ODTU01	1904	8	1	15232	1 × 4
ODTU12	952	32	2	30464	2 × 4
ODTU13	238	128	2	30464	2 × 4
ODTU23	952	128	8	121856	8 × 4

Optical channel data tributary unit k.ts

The optical channel data tributary unit k.ts (ODTUk.ts) is a structure which consists of an ODTUk.ts payload area and an ODTUk.ts overhead area (Figure 19-6). The ODTUk.ts payload area has j x ts columns and r rows (see Table 19-6) and the ODTUk.ts overhead area has one times 6 bytes. The ODTUk.ts is carried in "ts" 1.25G tributary slots of an HO OPUk.

The location of the ODTUk.ts overhead depends on the OPUk tributary slot used when multiplexing the ODTUk.ts in the OPUk (see clauses 19.1.1, 19.1.2, 19.1.4). The single instance of an ODTUk.ts overhead is located in the OPUk TSOH of the last OPUk tributary slot allocated to the ODTUk.ts.

The ODTUk.ts overhead carries the GMP justification overhead as specified in clause 19.4.

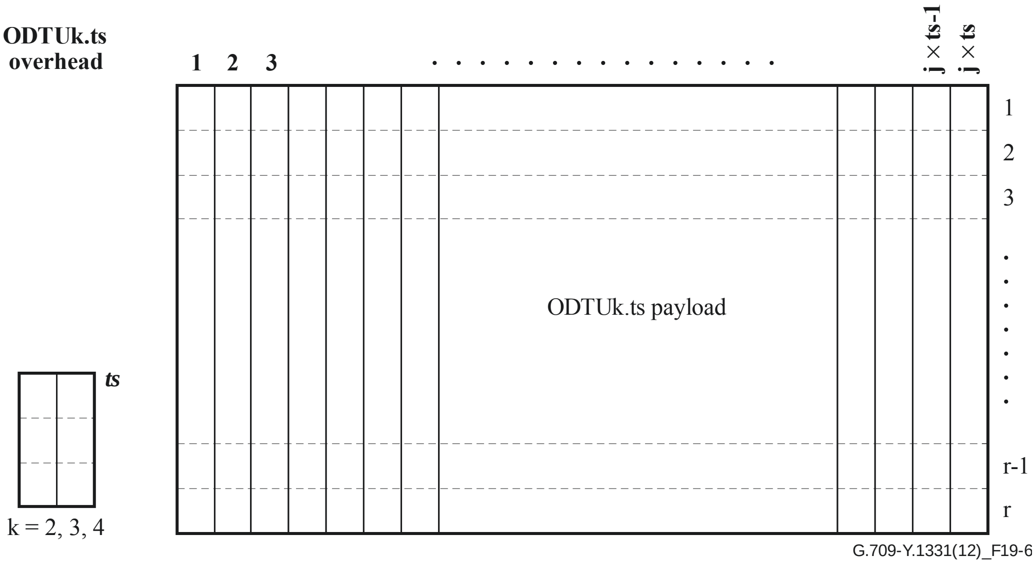


Figure 19-6 – ODTUk.ts frame formats

Table 19-6 – ODTUk.ts characteristics

	j	**r**	**ts**	**ODTUk.ts payload bytes**	**ODTUk.ts overhead bytes**
ODTU2.ts	476	32	1 to 8	15232 × ts	1 × 6
ODTU3.ts	119	128	1 to 32	15232 × ts	1 × 6
ODTU4.ts	95	160	1 to 80	15200 × ts	1 × 6

19.3 Multiplexing ODTU signals into the OPUk

Multiplexing an ODTU01 signal into an OPU1 is realized by mapping the ODTU01 signal in one of the two OPU1 1.25G tributary slots.

Multiplexing an ODTU12 signal into an OPU2 is realized by mapping the ODTU12 signal in one of the four OPU2 2.5G tributary slots or in two (of the eight) arbitrary OPU2 1.25G tributary slots: OPU2 TSa and TSb with $1 \le a < b \le 8$.

Multiplexing an ODTU13 signal into an OPU3 is realized by mapping the ODTU13 signal in one of the sixteen OPU3 2.5G tributary slots or in two (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa and TSb with $1 \le a < b \le 32$.

Multiplexing an ODTU23 signal into an OPU3 is realized by mapping the ODTU23 signal in four (of the sixteen) arbitrary OPU3 2.5G tributary slots: OPU3 TSa, TSb, TSc and TSd with $1 \le a < b < c < d \le 16$ or in eight (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa, TSb, TSc, TSd, TSe, TSf, TSg and TSh with $1 \le a < b < c < d < e < f < g < h \le 32$.

NOTE – a, b, c, d, e, f, g and h do not have to be sequential (a = i, b = i+1, c = i+2, d = i+3, e=i+4, f=i+5, g=i+6, h=i+7); the values can be arbitrarily selected to prevent bandwidth fragmentation.

Multiplexing an ODTU2.ts signal into an OPU2 is realized by mapping the ODTU2.ts signal in ts (of the eight) arbitrary OPU2 1.25G tributary slots: OPU2 TSa, TSb, .. , TSp with $1 \le a < b < .. < p \le 8$.

Multiplexing an ODTU3.ts signal into an OPU3 is realized by mapping the ODTU3.ts signal in ts (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa, TSb, .. , TSq with $1 \le a < b < .. < q \le 32$.

Multiplexing an ODTU4.ts signal into an OPU4 is realized by mapping the ODTU4.ts signal in ts (of the eighty) arbitrary OPU4 1.25G tributary slots: OPU4 TSa, TSb, .. , TSr with $1 \le a < b < .. < r \le 80$.

The OPUk overhead for these multiplexed signals consists of a payload type (PT), the multiplex structure identifier (MSI), the OPU4 multiframe identifier (k=4), the OPUk tributary slot overhead carrying the ODTU overhead and depending on the ODTU type one or more bytes reserved for future international standardization.

19.3.1 ODTU12 mapping into one OPU2 tributary slot

A byte of the ODTU12 payload signal is mapped into a byte of an OPU2 2.5G TS #i (i = 1,2,3,4) payload area, as indicated in Figure 19-7 (left). A byte of the ODTU12 overhead is mapped into a TSOH byte within column 16 of the OPU2 2.5G TS #i.

A byte of the ODTU12 signal is mapped into a byte of one of two OPU2 1.25G TS #A, B (A,B = 1,2,..,8) payload areas, as indicated in Figure 19-7 (right). A byte of the ODTU12 overhead is mapped into a TSOH byte within column 16 of the OPU2 1.25G TS #a,b.

The remaining OPU2 TSOH bytes in column 15 are reserved for future international standardization.

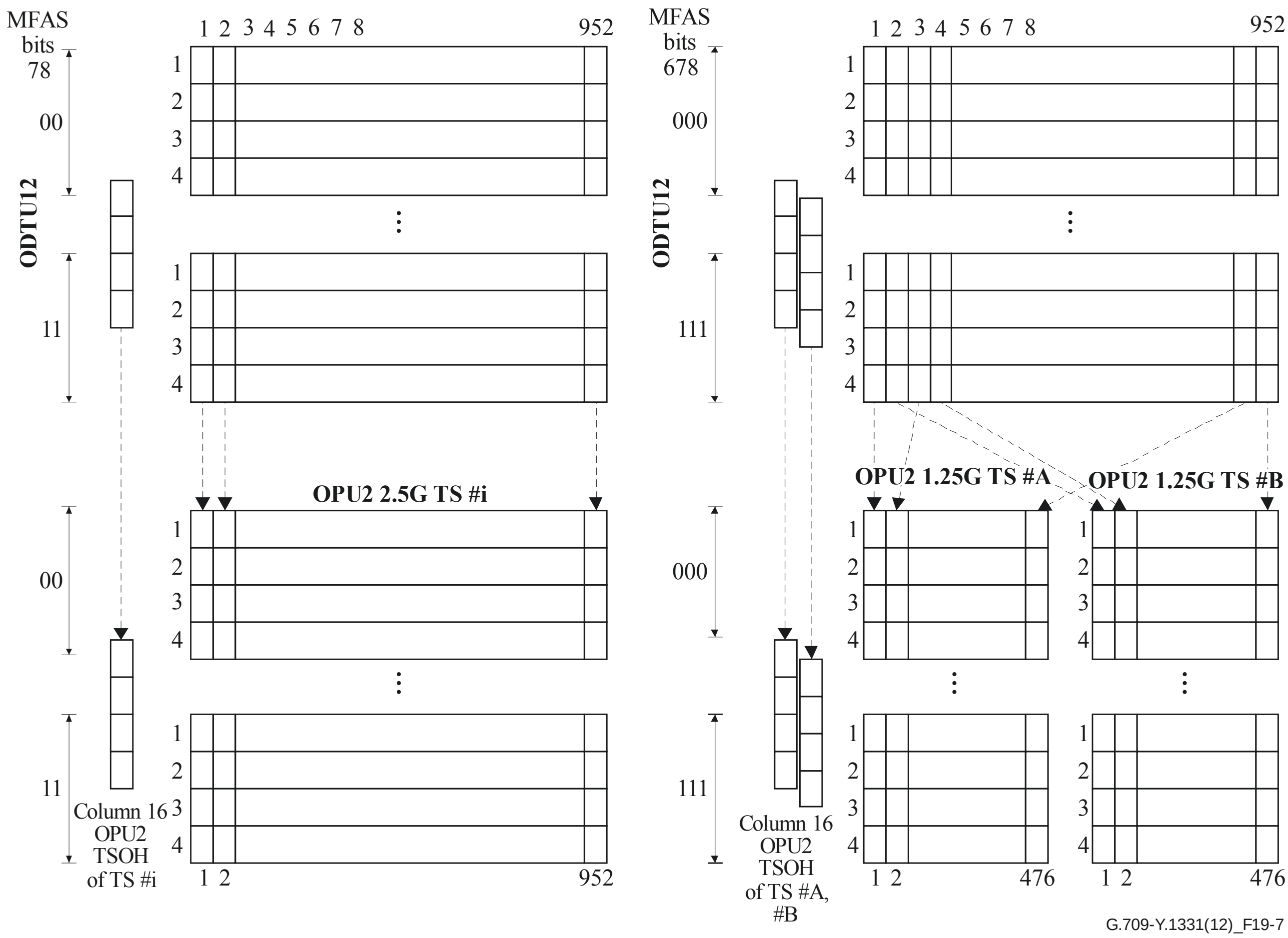


Figure 19-7 – Mapping of ODTU12 into one OPU2 2.5G tributary slot (left) and two OPU2 1.25G tributary slots (right)

19.3.2 ODTU13 mapping into one OPU3 tributary slot

A byte of the ODTU13 signal is mapped into a byte of an OPU3 2.5G TS #i (i = 1,2,..,16) payload area, as indicated in Figure 19-8 (left). A byte of the ODTU13 overhead is mapped into a TSOH byte within column 16 of the OPU3 2.5G TS #i.

A byte of the ODTU13 signal is mapped into a byte of one of two OPU3 1.25G TS #A, B (A,B = 1,2,..,32) payload areas, as indicated in Figure 19-8 (right). A byte of the ODTU13 overhead is mapped into a TSOH byte within column 16 of the OPU3 1.25G TS #a,b.

The remaining OPU3 TSOH bytes in column 15 are reserved for future international standardization.

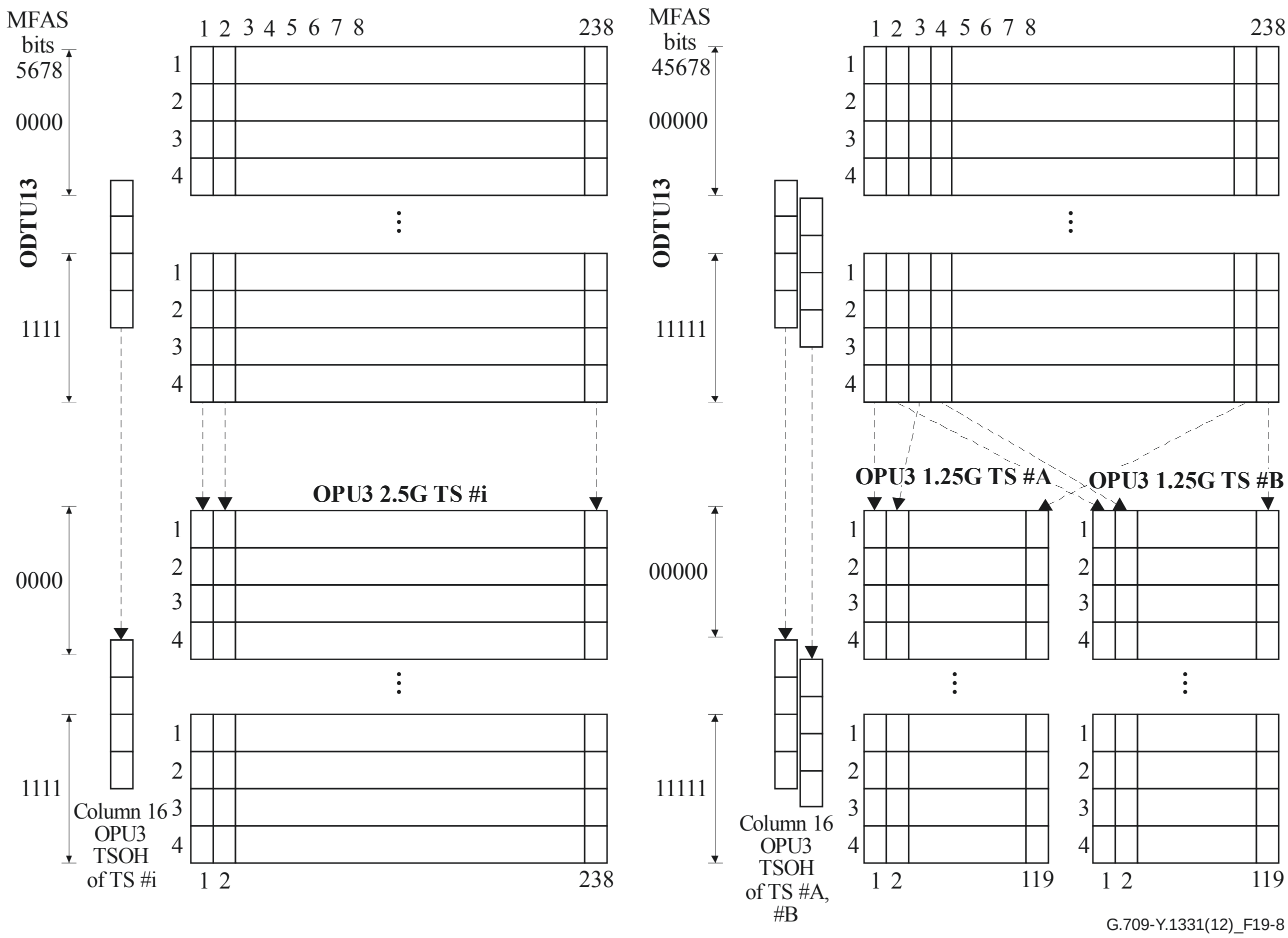


Figure 19-8 – Mapping of ODTU13 into one OPU3 2.5G tributary slot (left) and two OPU3 1.25G tributary slots (right)

19.3.3 ODTU23 mapping into four OPU3 tributary slots

A byte of the ODTU23 signal is mapped into a byte of one of four OPU3 2.5G TS #A,B,C,D (A,B,C,D = 1,2,..,16) payload areas, as indicated in Figure 19-9 (top). A byte of the ODTU23 overhead is mapped into a TSOH byte within column 16 of the OPU3 TS #a,b,c,d.

A byte of the ODTU23 signal is mapped into a byte of one of eight OPU3 1.25G TS #A, B, C, D, E, F, G, H (A,B,C,D,E,F,G,H = 1,2,..,32) payload areas, as indicated in Figure 19-9 (bottom). A byte of the ODTU23 overhead is mapped into a TSOH byte within column 16 of the OPU3 1.25G TS #a,b,c,d,e,f,g,h.

The remaining OPU3 TSOH bytes in column 15 are reserved for future international standardization.

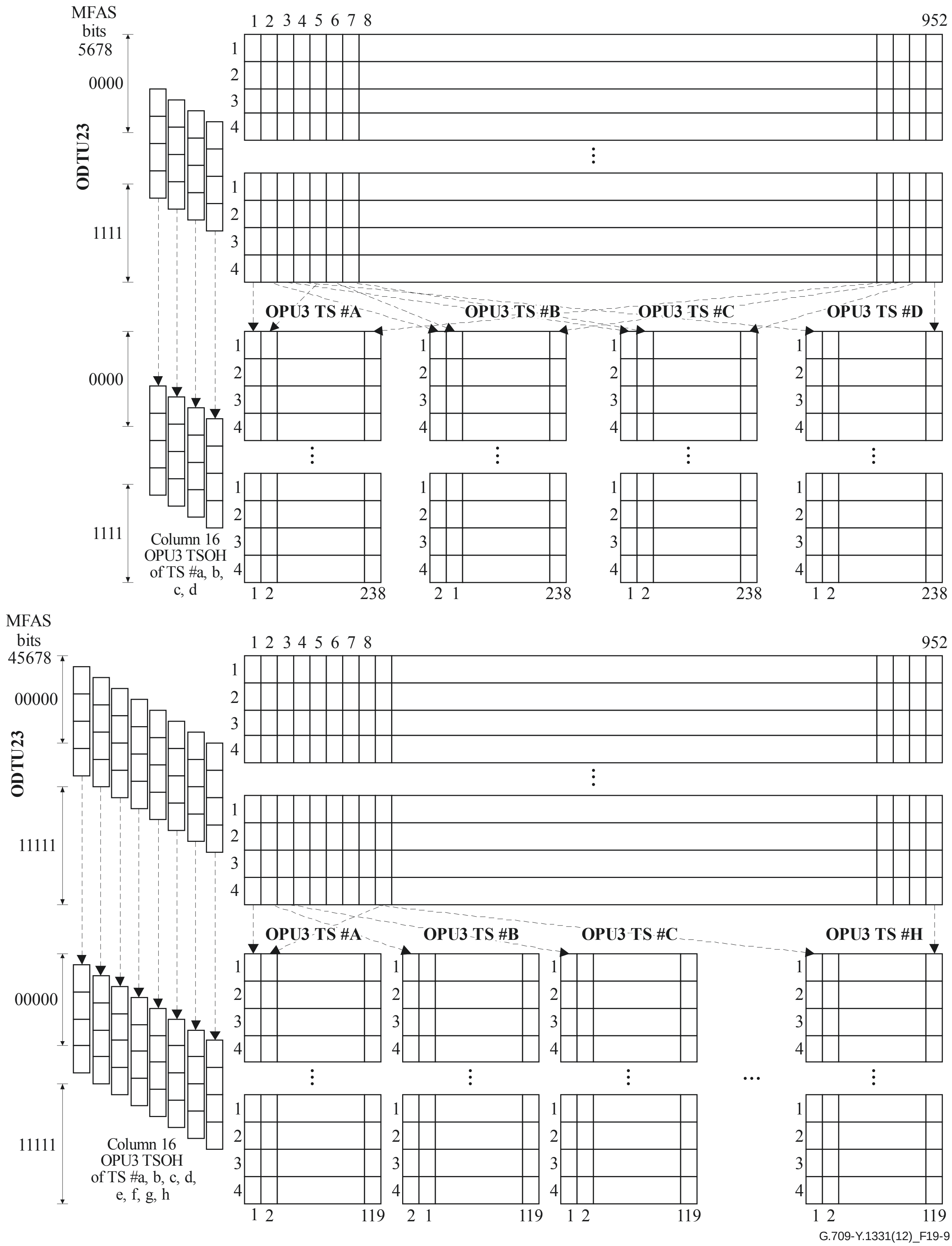


Figure 19-9 – Mapping of ODTU23 into 4 OPU3 2.5G tributary slots (#A, #B, #C, #D with A<B<C<D) (top) and 8 OPU3 1.25G tributary slots (#A, #B, #C, #D, #E, #F, #G, #H with A<B<C<D<E<F<G<H) (bottom)

19.3.4 ODTU01 mapping into one OPU1 1.25G tributary slot

A byte of the ODTU01 signal is mapped into a byte of an OPU1 1.25G TS #i (i = 1,2), as indicated in Figure 19-10 for a group of 4 rows out of the ODTU01.

A byte of the ODTU01 TSOH is mapped into a TSOH byte within column 16 of the OPU1 1.25G TS #i.

The remaining OPU1 TSOH bytes in column 15 are reserved for future international standardization.

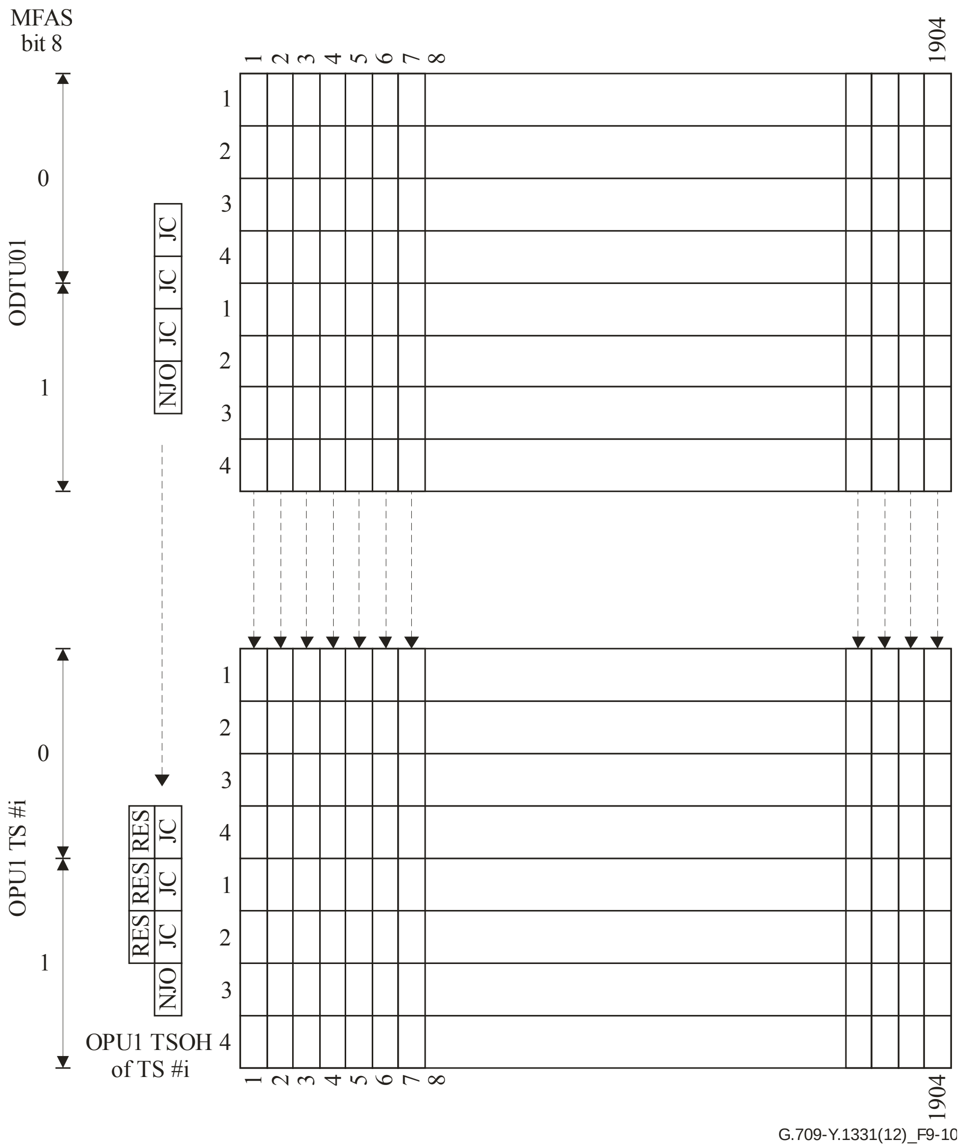


Figure 19-10 – Mapping of ODTU01 (excluding JOH) into OPU1 1.25G tributary slot

19.3.5 ODTU2.ts mapping into ts OPU2 1.25G tributary slots

A byte of the ODTU2.ts payload signal is mapped into a byte of an OPU2 1.25G TS #i (i = 1,..,ts) payload area, as indicated in Figure 19-11.

A byte of the ODTU2.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU2 1.25G tributary slot allocated to the ODTU2.ts.

The remaining OPU2 TSOH bytes are reserved for future international standardization.

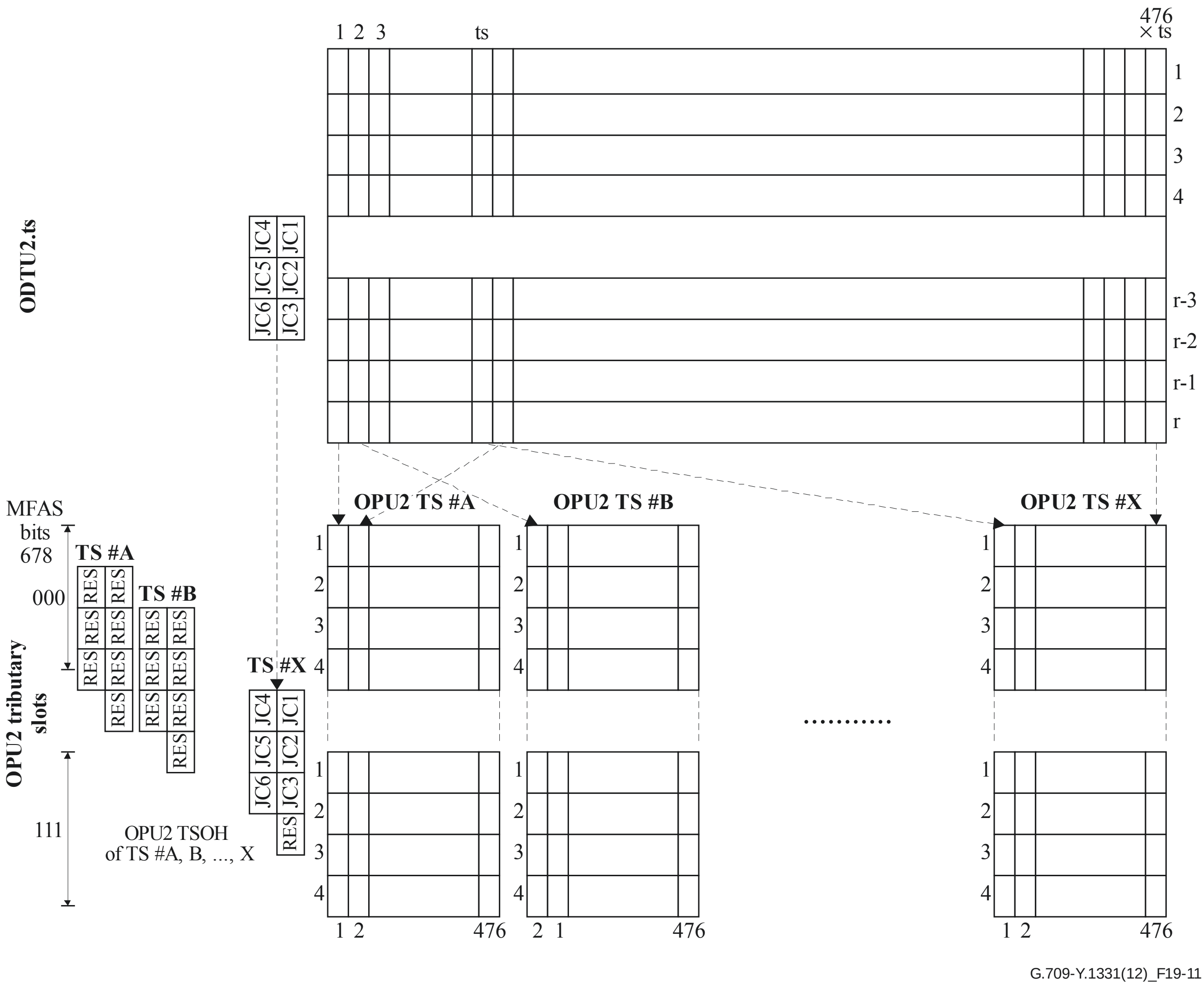


Figure 19-11 – Mapping of ODTU2.ts into 'ts' OPU2 1.25G tributary slots

19.3.6 ODTU3.ts mapping into ts OPU3 1.25G tributary slots

A byte of the ODTU3.ts payload signal is mapped into a byte of an OPU3 1.25G TS #i (i = 1,..,ts) payload area, as indicated in Figure 19-12.

A byte of the ODTU3.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU3 1.25G tributary slot allocated to the ODTU3.ts.

The remaining OPU3 TSOH bytes are reserved for future international standardization.

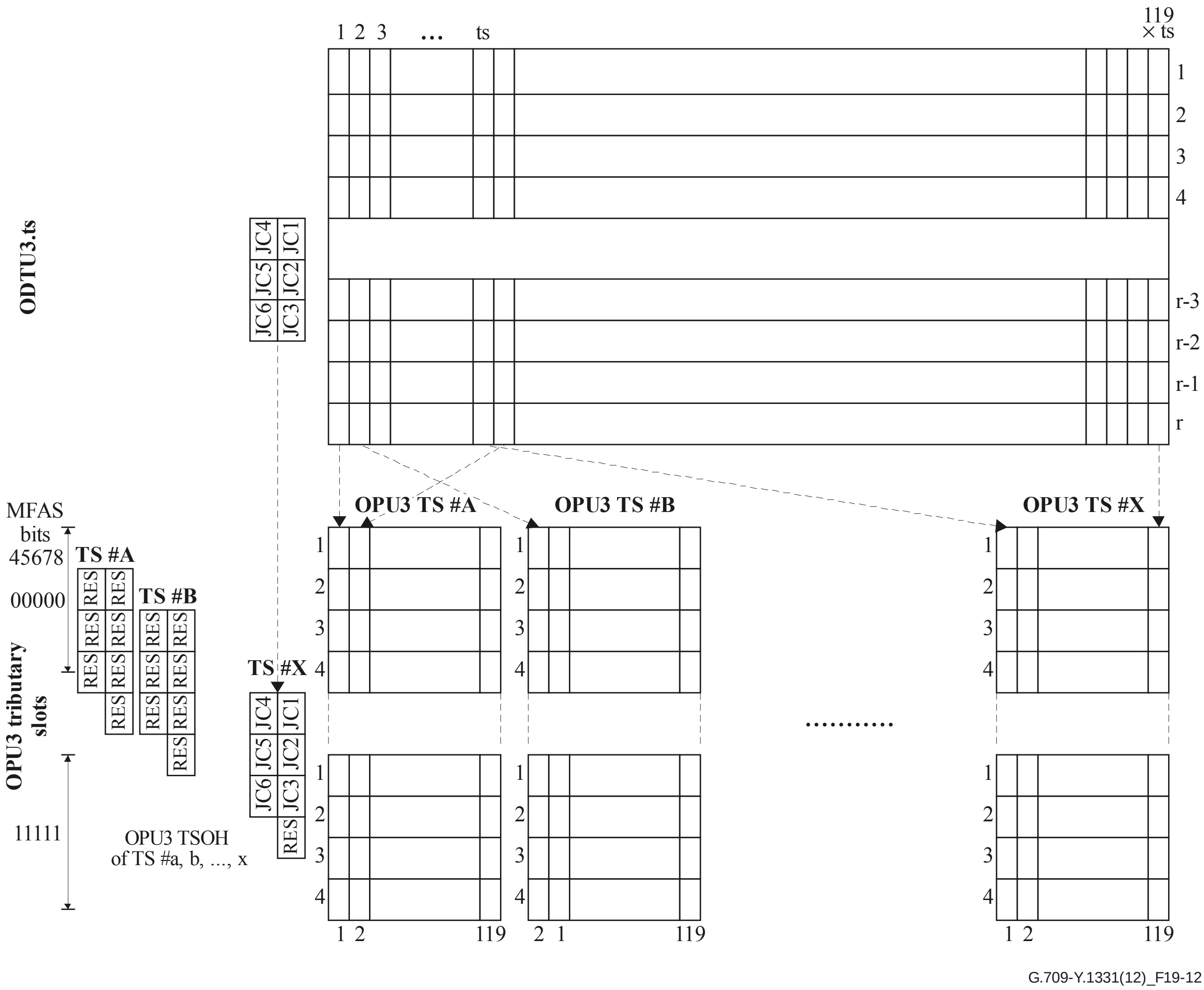


Figure 19-12 – Mapping of ODTU3.ts into 'ts' OPU3 1.25G tributary slots

19.3.7 ODTU4.ts mapping into ts OPU4 1.25G tributary slots

A byte of the ODTU4.ts payload signal is mapped into a byte of an OPU4 1.25G TS #i (i = 1,..,ts) payload area, as indicated in Figure 19-13.

A byte of the ODTU4.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU4 1.25G tributary slot allocated to the ODTU4.ts.

The remaining OPU4 TSOH bytes are reserved for future international standardization.

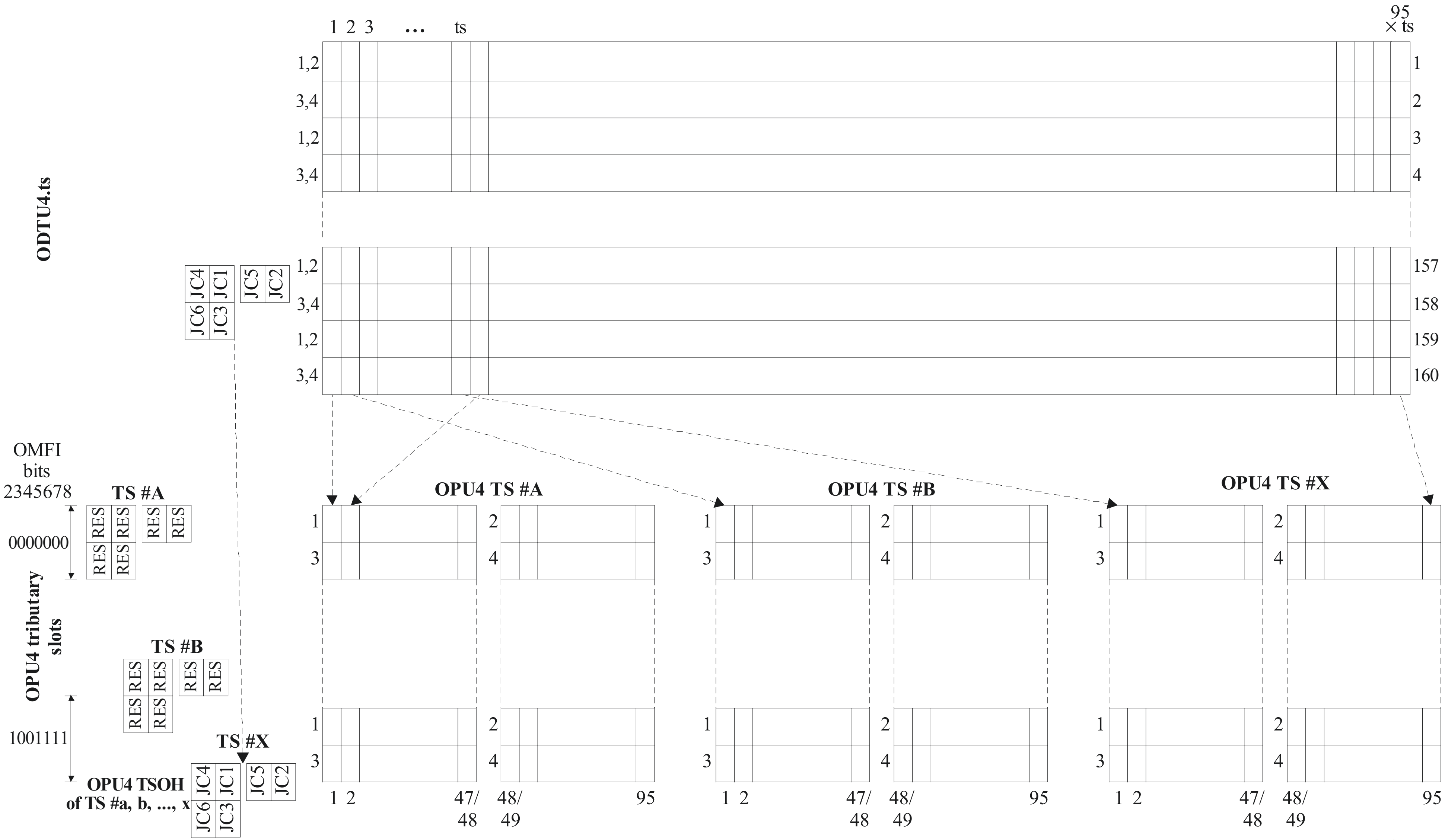


Figure 19-13 – Mapping of ODTU4.ts into 'ts' OPU4 1.25G tributary slots

19.4 OPUk multiplex overhead and ODTU justification overhead

The OPUk (k=1,2,3,4) multiplex overhead consists of a multiplex structure identifier (MSI) and an ODTU overhead. The OPUk (k=4) multiplex overhead contains an OPU multiframe identifier (OMFI).

The OPUk MSI overhead locations are shown in Figures 19-14A, 19-14B and 19-14C and the OMFI overhead location is shown in Figure 19-14C.

ODTUjk overhead

The ODTUjk overhead carries the AMP justification overhead consisting of justification control (JC) and negative justification opportunity (NJO) signals in column 16 of rows 1 to 4. ODTUjk overhead bytes in column 15 rows 1, 2 and 3 are reserved for future international standardization.

The ODTUjk overhead consists of 3 bytes of justification control (JC) and 1 byte of negative justification opportunity (NJO) overhead. The JC and NJO overhead locations are shown in Figures 19-14A and 19-14B. In addition, two or n times two positive justification overhead bytes (PJO1, PJO2) are located in the ODTUjk payload area. Note that the PJO1 and PJO2 locations are multiframe, ODUj and OPUk tributary slot dependent.

The PJO1 for an ODU1 in OPU2 or OPU3 2.5G tributary slot #i (i: 1..4 or 1..16 respectively) is located in the first column of OPUk 2.5G tributary slot #i (OPUk column 16+i) and the PJO2 is located in the second column of OPUk 2.5G tributary slot #i (OPU2 column 20+i, OPU3 column 32+i) in frame #i of the four or sixteen frame multiframe.

EXAMPLE – ODU1 in OPU2 or OPU3 TS(1): PJO1 in column 16+1=17, PJO2 in column 20+1=21 (OPU2) and 32+1=33 (OPU3). ODU1 in OPU2 TS(4): PJO1 in column 16+4=20, PJO2 in column 20+4=24. ODU1 in OPU3 TS(16): PJO1 in column 16+16=32, PJO2 in column 32+16=48.

The four PJO1s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #a (OPU3 column 16+a) in frames #a, #b, #c and #d of the sixteen frame multiframe. The four PJO2s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #b (OPU3 column 16+b) in frames #a, #b, #c and #d of the sixteen frame multiframe. Figure 19-14A presents an example with four ODU2s in the OPU3 mapped into 2.5G tributary slots (1,5,9,10), (2,3,11,12), (4,14,15,16) and (6,7,8,13).

EXAMPLE – ODU2 in OPU3 TS(1,2,3,4): PJO1 in column 16+1=17, PJO2 in column 16+2=18. ODU2 in OPU3 TS(13,14,15,16): PJO1 in column 16+13=29, PJO2 in column 16+14=30.

The PJO1 for an ODU0 in OPU1 1.25G tributary slot #i (i: 1,2) is located in the first column of OPU1 1.25G tributary slot #i (OPU1 column 16+i) and the PJO2 is located in the second column of OPU1 1.25G tributary slot #i (OPU1 column 18+i) in frame #i of the two frame multiframe.

The PJO1 for an ODU1 in OPU2 or OPU3 1.25G tributary slots #a and #b (a: 1..7 or 1..31 respectively; b: 2..8 or 2..32 respectively) is located in the first column of OPUk 1.25G tributary slot #a (OPUk column 16+a) and the PJO2 is located in the first column of OPUk 1.25G tributary slot #b (OPUk column 16+b) in frames #a and #b of the eight or thirty-two frame multiframe.

EXAMPLE – ODU1 in OPU2 or OPU3 TS(1,2): PJO1 in column 16+1=17, PJO2 in column 16+2=18. ODU1 in OPU2 TS(7,8): PJO1 in column 16+7=23, PJO2 in column 16+8=24. ODU1 in OPU3 TS(31,32): PJO1 in column 16+31=47, PJO2 in column 16+32=48.

The eight PJO1s for an ODU2 in OPU3 1.25G tributary slots #a, #b, #c, #d, #e, #f, #g and #h are located in the first column of OPU3 1.25G tributary slot #a (OPU3 column 16+a) in frames #a, #b, #c, #d, #e, #f, #g and #h of the thirty-two frame multiframe. The eight PJO2s for an ODU2 in OPU3 1.25G tributary slots #a, #b, #c, #d, #e, #f, #g and #h are located in the first column of OPU3 1.25G tributary slot #b (OPU3 column 16+b) in frames #a, #b, #c, #d, #e, #f, #g and #h of the thirty-two frame multiframe. Figure 19-14B presents an example with two ODU2s and two ODU1s in the OPU3 mapped into 1.25G tributary slots (1,5,9,10,17,19,20,21), (25,26,27,28,29,30,31,32), (2,3) and (4,24).

EXAMPLE – ODU2 in OPU3 TS(1,2,3,4,5,6,7,8): PJO1 in column 16+1=17, PJO2 in column 16+2=18. ODU2 in OPU3 TS(25,26,27,28,29,30,31,32): PJO1 in column 16+25=41, PJO2 in column 16+26=42.

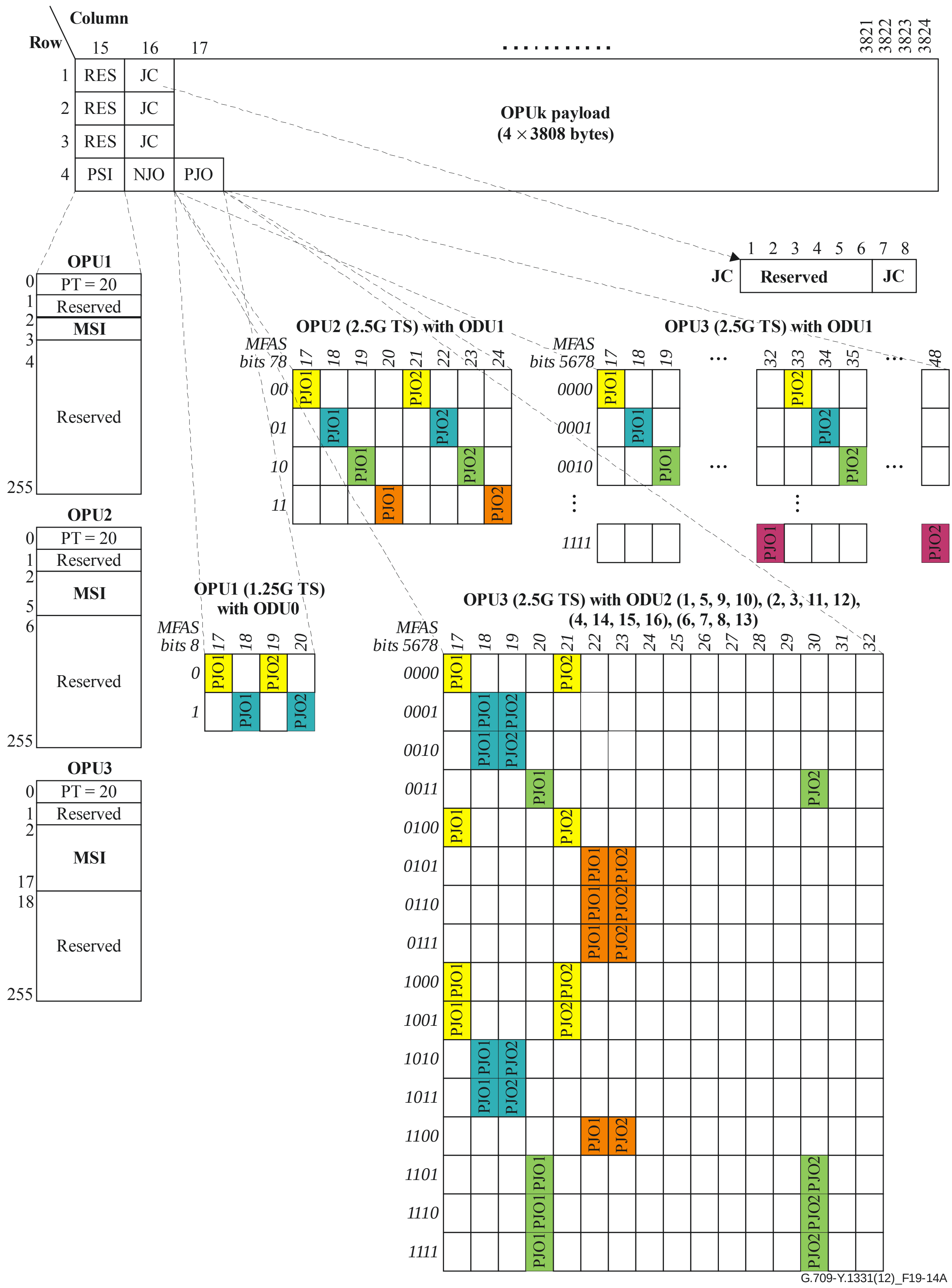


Figure 19-14A – OPUk (k=1,2,3) multiplex overhead associated with an ODTUjk only (payload type = 20)

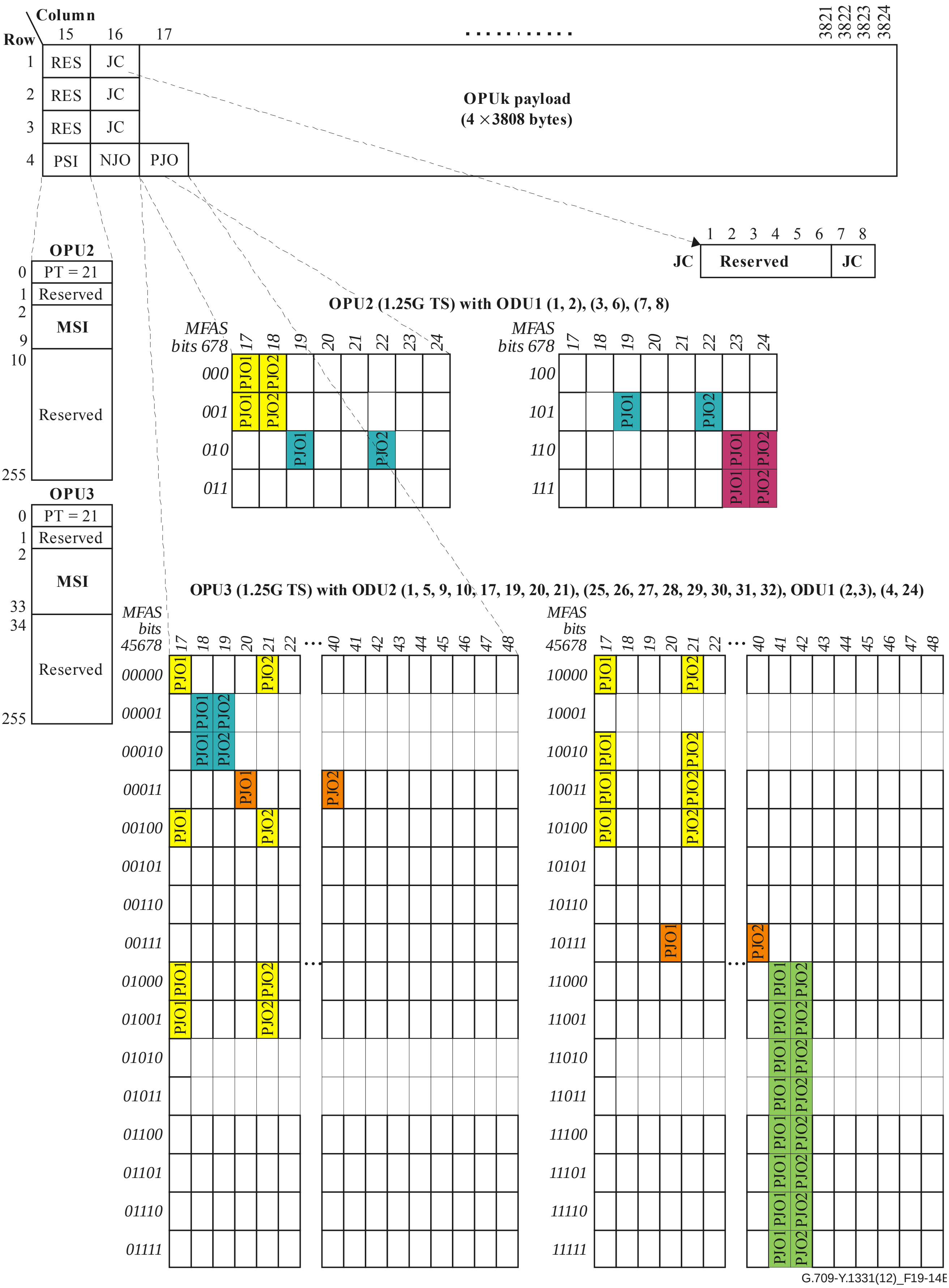


Figure 19-14B – OPUk (k=2,3) multiplex overhead associated with an ODTUjk only (payload type = 21)

ODTUk.ts overhead

The ODTUk.ts overhead carries the GMP justification overhead consisting of 3 bytes of justification control (JC1, JC2, JC3) which carry the 14-bit GMP C_m information and client/LO ODU specific 3 bytes of justification control (JC4, JC5, JC6) which carry the 10-bit GMP ΣC_{8D} information.

The JC1, JC2, JC3, JC4, JC5 and JC6 overhead locations are shown in Figure 19-14C.

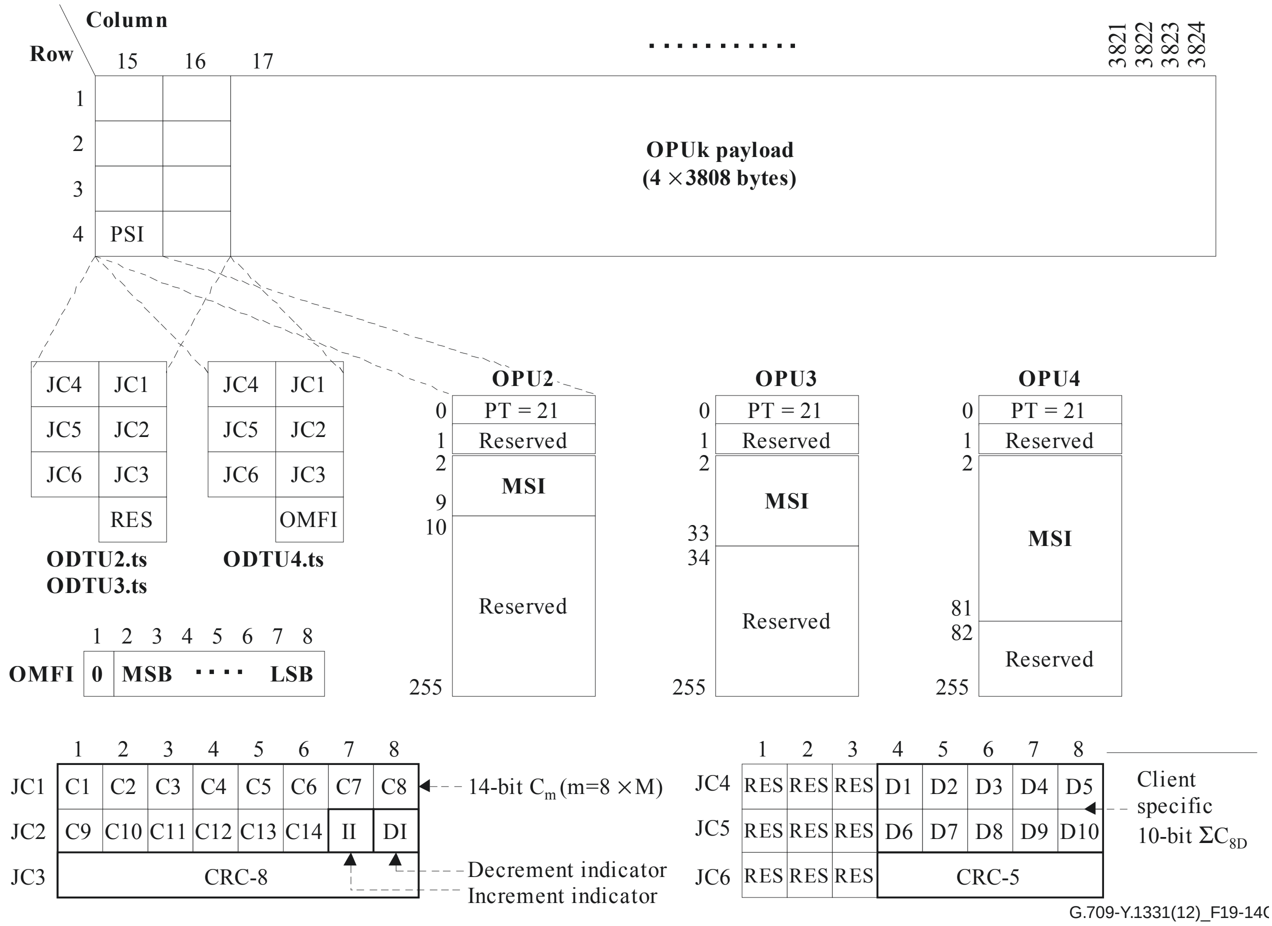


Figure 19-14C – OPUk (k=2,3,4) multiplex overhead associated with an ODTUk.ts (payload type = 21)

19.4.1 OPUk multiplex structure identifier (MSI)

The OPUk (k=1,2,3,4) multiplex structure identifier (MSI) overhead, which encodes the ODU multiplex structure in the OPU, is located in the mapping specific area of the PSI signal (refer to Figure 19-14A for the MSI location in OPUk with PT=20, refer to Figures 19-14B and 19-14C for the MSI location in OPUk with PT=21). The MSI has an OPU and tributary slot (2.5G, 1.25G) specific length (OPU1: 2 bytes, OPU2: 4 or 8 bytes, OPU3: 16 or 32 bytes, OPU4: 80 bytes) and indicates the ODTU content of each tributary slot (TS) of an OPU. One byte is used for each TS.

19.4.1.1 OPU2 multiplex structure identifier (MSI) – Payload type 20

For the 4 OPU2 2.5G tributary slots four bytes of the PSI are used (PSI[2] .. PSI[5]) as MSI bytes as shown in Figures 19-14A and 19-15. The MSI indicates the ODTU content of each tributary slot of the OPU2. One byte is used for each tributary slot.

– The ODTU type in bits 1 and 2 is fixed to 00 to indicate the presence of an ODTU12.

– The tributary port # indicates the port number of the ODU1 that is being transported in this 2.5G TS; the assignment of ports to tributary slots is fixed, the port number equals the tributary slot number.

	1 2	3 4 5 6 7 8	
PSI[2]	00	00 0000	*TS1*
PSI[3]	00	00 0001	*TS2*
PSI[4]	00	00 0010	*TS3*
PSI[5]	00	00 0011	*TS4*

Figure 19-15 – OPU2-MSI coding – Payload type 20

19.4.1.2 OPU3 multiplex structure identifier (MSI) – Payload type 20

For the 16 OPU3 2.5G tributary slots 16 bytes of the PSI are used (PSI[2] .. PSI[17]) as MSI bytes as shown in Figures 19-14A, 19-16A and 19-16B. The MSI indicates the ODTU content of each tributary slot of the OPU3. One byte is used for each tributary slot.

– The ODTU type in bits 1 and 2 indicates if the OPU3 TS is carrying ODTU13 or ODTU23. The default ODTU type is ODTU13; it is present when either a tributary slot carries an ODTU13, or is not allocated to carry an ODTU. Refer to Appendix V for some examples.

– The tributary port # in bits 3 to 8 indicates the port number of the ODTU13/23 that is being transported in this 2.5G TS; for the case of ODTU23, a flexible assignment of tributary ports to tributary slots is possible, for the case of ODTU13, this assignment is fixed, the tributary port number equals the tributary slot number. ODTU23 tributary ports are numbered 1 to 4.

	1 2	3 4 5 6 7 8	
PSI[2]	ODTU type	Tributary Port #	*TS1*
PSI[3]	ODTU type	Tributary Port #	*TS2*
PSI[4]	ODTU type	Tributary Port #	*TS3*
PSI[5]	ODTU type	Tributary Port #	*TS4*
PSI[6]	ODTU type	Tributary Port #	*TS5*
PSI[7]	ODTU type	Tributary Port #	*TS6*
PSI[8]	ODTU type	Tributary Port #	*TS7*
PSI[9]	ODTU type	Tributary Port #	*TS8*
PSI[10]	ODTU type	Tributary Port #	*TS9*
PSI[11]	ODTU type	Tributary Port #	*TS10*
PSI[12]	ODTU type	Tributary Port #	*TS11*
PSI[13]	ODTU type	Tributary Port #	*TS12*
PSI[14]	ODTU type	Tributary Port #	*TS13*
PSI[15]	ODTU type	Tributary Port #	*TS14*
PSI[16]	ODTU type	Tributary Port #	*TS15*
PSI[17]	ODTU type	Tributary Port #	*TS16*

Figure 19-16A – OPU3-MSI coding – Payload type 20

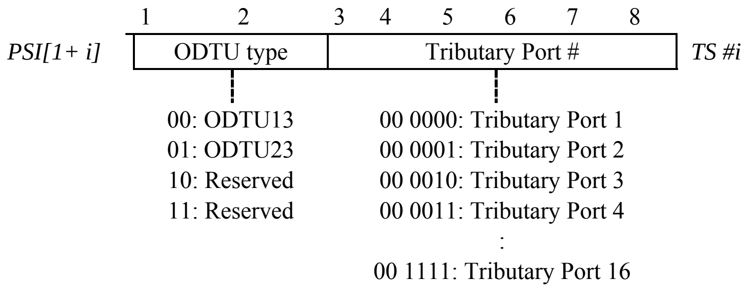


Figure 19-16B – OPU3 MSI coding – Payload type 20

19.4.1.3 OPU1 multiplex structure identifier (MSI) – Payload type 20

For the 2 OPU1 1.25G tributary slots 2 bytes of the PSI are used (PSI[2], PSI[3]) as MSI bytes as shown in Figures 19-14A and 19-17. The MSI indicates the ODTU content of each tributary slot of the OPU1. One byte is used for each tributary slot.

– The ODTU type in bits 1 and 2 is fixed to 11 to indicate the presence of an ODTU01.

– The tributary port # in bits 3 to 8 indicates the port number of the ODTU01 that is being transported in this 1.25G TS; the assignment of ports to tributary slots is fixed, the port number equals the tributary slot number.

	1 2	3 4 5 6 7 8	1.25G TS
PSI[2]	11	00 0000	*TS1*
PSI[3]	11	00 0001	*TS2*

Figure 19-17 – OPU1 MSI coding – Payload type 20

19.4.1.4 OPU4 multiplex structure identifier (MSI) – Payload type 21

For the eighty OPU4 1.25G tributary slots 80 bytes of the PSI are used (PSI[2] to PSI[81]) as MSI bytes as shown in Figures 19-14C, 19-18A and 19-18B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

– The TS occupation bit 1 indicates if the tributary slot is allocated or unallocated.

– The tributary port # in bits 2 to 8 indicates the port number of the ODTU4.ts that is being transported in this TS; for the case of an ODTU4.ts carried in two or more tributary slots, a flexible assignment of tributary port to tributary slots is possible. ODTU4.ts tributary ports are numbered 1 to 80. The value is set to all-0s when the occupation bit has the value 0 (tributary slot is unallocated).

	1	2 3 4 5 6 7 8	1.25G TS
PSI[2]	TS occupied	Tributary Port #	*TS1*
PSI[3]	TS occupied	Tributary Port #	*TS2*
PSI[4]	TS occupied	Tributary Port #	*TS3*
PSI[5]	TS occupied	Tributary Port #	*TS4*
PSI[6]	TS occupied	Tributary Port #	*TS5*
PSI[7]	TS occupied	Tributary Port #	*TS6*
PSI[8]	TS occupied	Tributary Port #	*TS7*
PSI[9]	TS occupied	Tributary Port #	*TS8*
:	:	:	:
:	:	:	:
PSI[81]	TS occupied	Tributary Port #	*TS80*

Figure 19-18A – OPU4 1.25G TS MSI coding – Payload type 21

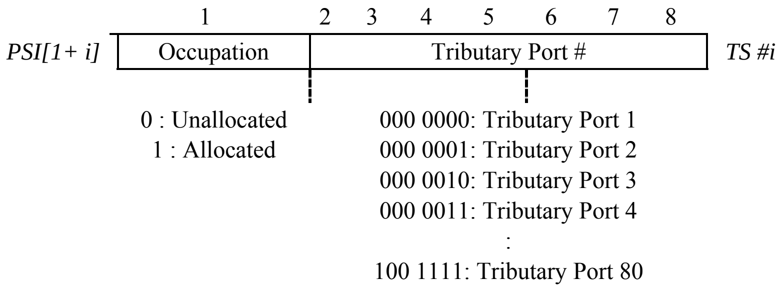


Figure 19-18B – OPU4 MSI coding – Payload type 21

19.4.1.5 OPU2 multiplex structure identifier (MSI) – Payload type 21

For the eight OPU2 1.25G tributary slots 8 bytes of the PSI (PSI[2] to PSI[9]) are used as MSI bytes as show in Figures 19-14B, 19-19A and 19-19B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

– The ODTU type in bits 1 and 2 indicates if the OPU2 1.25G TS is carrying an ODTU12 or ODTU2.ts. The default ODTU type is 11 (unallocated); it is present when a tributary slot is not allocated to carry an ODTU.

– The tributary port # in bits 3 to 8 indicates the port number of the ODTU that is being transported in this TS; a flexible assignment of tributary ports to tributary slots is possible, ODTU12 tributary ports are numbered 1 to 4, and ODTU2.ts tributary ports are numbered 1 to 8. The value is set to all-0s when the ODTU type has the value 11 (tributary slot is unallocated).

	1 2	3 4 5 6 7 8	
PSI[2]	ODTU type	Tributary Port #	*TS1*
PSI[3]	ODTU type	Tributary Port #	*TS2*
PSI[4]	ODTU type	Tributary Port #	*TS3*
PSI[5]	ODTU type	Tributary Port #	*TS4*
PSI[6]	ODTU type	Tributary Port #	*TS5*
PSI[7]	ODTU type	Tributary Port #	*TS6*
PSI[8]	ODTU type	Tributary Port #	*TS7*
PSI[9]	ODTU type	Tributary Port #	*TS8*

Figure 19-19A – OPU2 MSI coding – Payload type 21

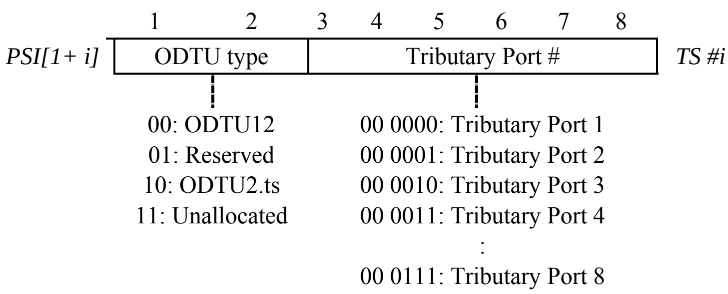


Figure 19-19B – OPU2 MSI coding – Payload type 21

19.4.1.6 OPU3 with 1.25G tributary slots (payload type 21) multiplex structure identifier (MSI)

For the thirty-two OPU3 1.25G tributary slots 32 bytes of the PSI (PSI[2] to PSI[33]) are used as MSI bytes as shown in Figures 19-14B, 19-20A and 19-20B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

– The ODTU type in bits 1 and 2 indicates if the OPU3 1.25G TS is carrying an ODTU13, ODTU23 or ODTU3.ts. The default ODTU type is 11 (unallocated); it is present when a tributary slot is not allocated to carry an ODTU.

– The tributary port # in bits 3 to 8 indicates the port number of the ODTU that is being transported in this TS; a flexible assignment of tributary ports to tributary slots is possible, ODTU13 tributary ports are numbered 1 to 16, ODTU23 tributary ports are numbered 1 to 4 and ODTU3.ts tributary ports are numbered 1 to 32. The value is set to all-0s when the ODTU type has the value 11 (tributary slot is unallocated).

	1 2	3 4 5 6 7 8	
PSI[2]	ODTU type	Tributary Port #	*TS1*
PSI[3]	ODTU type	Tributary Port #	*TS2*
PSI[4]	ODTU type	Tributary Port #	*TS3*
PSI[5]	ODTU type	Tributary Port #	*TS4*
PSI[6]	ODTU type	Tributary Port #	*TS5*
:	:	:	:
:	:	:	:
PSI[33]	ODTU type	Tributary Port #	*TS32*

Figure 19-20A – OPU3 MSI coding – Payload type 21

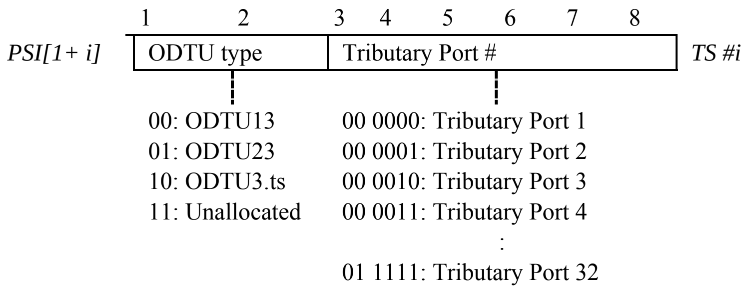


Figure 19-20B – OPU3 MSI coding – Payload type 21

19.4.2 OPUk payload structure identifier reserved overhead (RES)

253 (OPU1), 251 or 247 (OPU2), 239 or 223 (OPU3) and 175 (OPU4) bytes are reserved in the OPUk PSI for future international standardization. These bytes are located in PSI[1] and PSI[4] (OPU1), PSI[6] or PSI[10] (OPU2), PSI[18] or PSI[34] (OPU3), PSI[82] (OPU4) to PSI[255] of the OPUk overhead. These bytes are set to all-0s.

19.4.3 OPUk multiplex justification overhead (JOH)

Two mapping procedures are used for the mapping of ODUj: either the asynchronous mapping procedure (AMP) or generic mapping procedure (GMP) into ODTUjk or ODTUk.ts, respectively. AMP uses ODUj and OPUk specific fixed stuff and justification opportunity definitions (ODTUjk). GMP uses ODUj and OPUk independent stuff and justification opportunity definitions (ODTUk.ts). Stuff locations within an ODTUk.ts are determined by means of a formula which is specified in clause 19.4.3.2.

19.4.3.1 Asynchronous mapping procedures (AMP)

The justification overhead (JOH) located in column 16 of the OPUk (k=1,2,3) as indicated in Figures 19-14A and 19-14B consists of three justification control (JC) bytes and one negative justification opportunity (NJO) byte. The three JC bytes are located in rows 1, 2 and 3. The NJO byte is located in row 4.

Bits 7 and 8 of each JC byte are used for justification control. The other six bits are reserved for future international standardization.

19.4.3.2 Generic mapping procedure (GMP)

The justification overhead (JOH) for the generic mapping procedure consists of two groups of three bytes of justification control; the general (JC1, JC2, JC3) and the client to LO ODU mapping specific (JC4, JC5, JC6). Refer to Figure 19-14C.

The JC1, JC2 and JC3 bytes consist of a 14-bit C_m field (bits C1, C2, .., C14), a 1-bit increment indicator (II) field, a 1-bit decrement indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The JC4, JC5 and JC6 bytes consist of a 10-bit ΣC_{nD} field (bits D1, D2, .., D10), a 5-bit CRC-5 field which contains an error check code over bits 4 to 8 in the JC4, JC5 and JC6 fields and nine bits reserved for future international standardization (RES).

The value of 'm' in C_m is 8 × 'ts' (number of tributary slots occupied by the ODTUk.ts).

The value of 'n' represents the timing granularity of the GMP C_n parameter, which is also present in ΣC_{nD}. The value of n is 8.

The value of C_m controls the distribution of groups of 'ts' LO ODUj data bytes into groups of 'ts' ODTUk.ts payload bytes. Refer to clause 19.6 and Annex D for further specification of this process.

The value of ΣC_{nD} provides additional 'n'-bit timing information, which is necessary to control the jitter and wander performance experienced by the LO ODUj signal.

The value of C_n is computed as follows: $C_n(t) = m \times C_m(t) + (\Sigma C_{nD}(t) - \Sigma C_{nD}(t-1))$. Note that the value CnD is effectively an indication of the amount of data in the mapper's virtual queue that it could not send during that multiframe due to it being less than an M-byte word. For the case where the value of ΣC_{nD} in a multiframe 't' is corrupted, it is possible to recover from such error in the next multiframe 't+1'.

19.4.4 OPU multiframe identifier overhead (OMFI)

An OPU4 multiframe identifier (OMFI) byte is defined in row 4, column 16 of the OPU4 overhead (Figure 19-21). The value of bits 2 to 8 of the OMFI byte will be incremented each OPU4 frame to provide an 80 frame multiframe for the multiplexing of LO ODUs into the OPU4.

NOTE – It is an option to align the OMFI = 0 position with MFAS = 0 position every 1280 (the least common multiple of 80 and 256) frame periods.

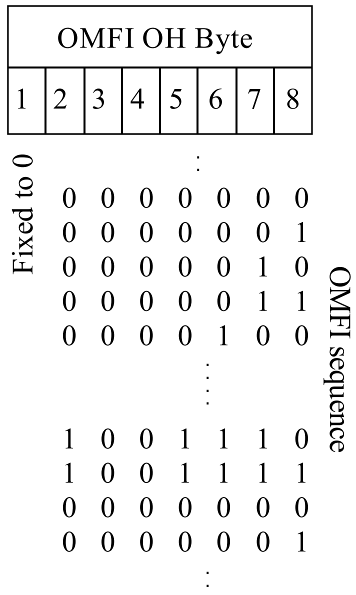


Figure 19-21 – OPU4 multiframe identifier (OMFI) overhead

19.5 Mapping ODUj into ODTUjk

The mapping of ODUj signals (with up to ±20 ppm bit-rate tolerance) into the ODTUjk signal ((j,k) = {(0,1), (1,2); (1,3), (2,3)}) is performed as an asynchronous mapping.

NOTE 1 – The maximum bit-rate tolerance between OPUk and the ODUj signal clock, which can be accommodated by this mapping scheme is –130 to +65 ppm (ODU0 into OPU1), –113 to +83 ppm (ODU1 into OPU2), –96 to +101 ppm (ODU1 into OPU3) and –95 to +101 ppm (ODU2 into OPU3).

The ODUj signal is extended with a frame alignment overhead as specified in clauses 15.6.2.1 and 15.6.2.2 and an all-0s pattern in the OTUj overhead field (see Figure 19-22).

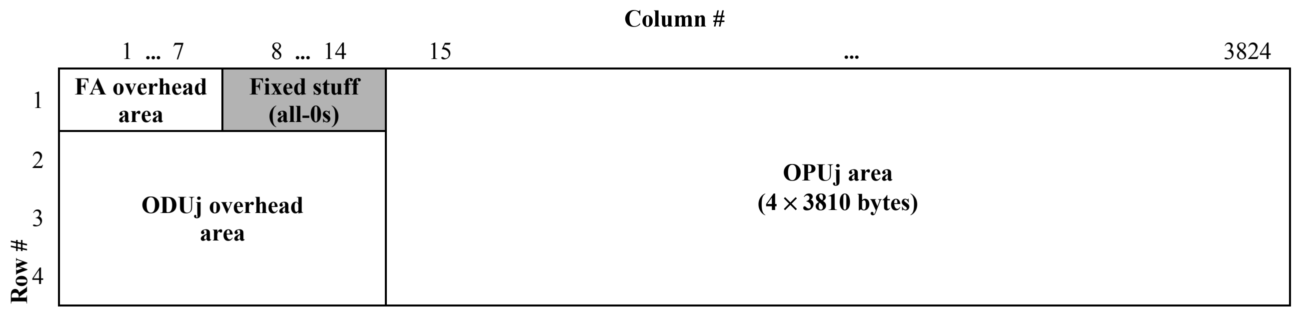


Figure 19-22 – Extended ODUj frame structure (FA OH included, OTUj OH area contains fixed stuff)

The OPUk signal and therefore the ODTUjk (k = 1,2,3) signals are created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the ODUj (j = 0,1,2) client signals.

The extended ODUj (j = 0,1,2) signal is mapped into the ODTUjk (k = 1,2,3) using an asynchronous mapping with –1/0/+1/+2 positive/negative/zero (pnz) justification scheme.

An extended ODUj byte is mapped into an ODTUjk byte.

The asynchronous mapping process generates the JC, NJO, PJO1 and PJO2 according to Table 19-7. The de-mapping process interprets JC, NJO, PJO1 and PJO2 according to Table 19-7.

Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

Table 19-7 – JC, NJO, PJO1 and PJO2 generation and interpretation

JC 7 8	NJO	PJO1	PJO2	Interpretation
0 0	justification byte	data byte	data byte	no justification (0)
0 1	data byte	data byte	data byte	negative justification (–1)
1 0 (Note)	justification byte	justification byte	justification byte	double positive justification (+2)
1 1	justification byte	justification byte	data byte	positive justification (+1)
NOTE – Note that this code is not used for the case of ODU0 into OPU1.				

The value contained in NJO, PJO1 and PJO2 when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming ODUj client signal (e.g., OTUj-LOF), this failed incoming signal will contain the ODUj-AIS signal as specified in clause 16.5.1. This ODUj-AIS is then mapped into the ODTUjk.

For the case where the ODUj is received from the output of a fabric (ODU connection function), the incoming signal may contain (in the case of an open matrix connection), the ODUj-OCI signal as specified in clause 16.5.2. This ODUj-OCI signal is then mapped into the ODTUjk.

NOTE 2 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such a unit is intentionally absent (i.e., not installed), the associated ODTUjk signals should carry an ODUj-OCI signal. If such a unit is installed but temporarily removed as part of a repair action, the associated ODTUjk signal should carry an ODUj-AIS signal.

The de-mapping of ODUj signals from the ODTUjk signal (j = 0,1,2; k = 1,2,3) is performed by extracting the extended ODUj signal from the OPUk under control of its justification overhead (JC, NJO, PJO1, PJO2).

NOTE 3 – For the case where the ODUj signal is output as an OTUj signal, frame alignment of the extracted extended ODUj signal is to be recovered to allow frame synchronous mapping of the ODUj into the OTUj signal.

During a signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the ODUj-AIS pattern as specified in clause 16.5.1 is generated as a replacement signal for the lost ODUj signal.

19.5.1 Mapping ODU1 into ODTU12

A byte of the ODU1 signal is mapped into an information byte of the ODTU12 (see Figure 19-23A). Once per 4 OPU2 frames, it is possible to perform either a positive or a negative justification action. The frame in which justification can be performed is related to the TSOH of the OPU2 2.5G TS in which the ODTU12 is mapped (Figure 19-1). Figure 19-23A shows the case with mapping in OPU2 2.5G TS1.

A byte of the ODU1 signal is mapped into an information byte of the ODTU12 (see Figure 19-23B). Twice per 8 OPU2 frames, it is possible to perform either a positive or a negative justification action. The frames in which justification can be performed are related to the TSOH of the OPU2 1.25G TSs in which the ODTU12 is mapped (Figure 19-1). Figure 19-23B shows the case with mapping in OPU2 1.25G TS2 and TS4.

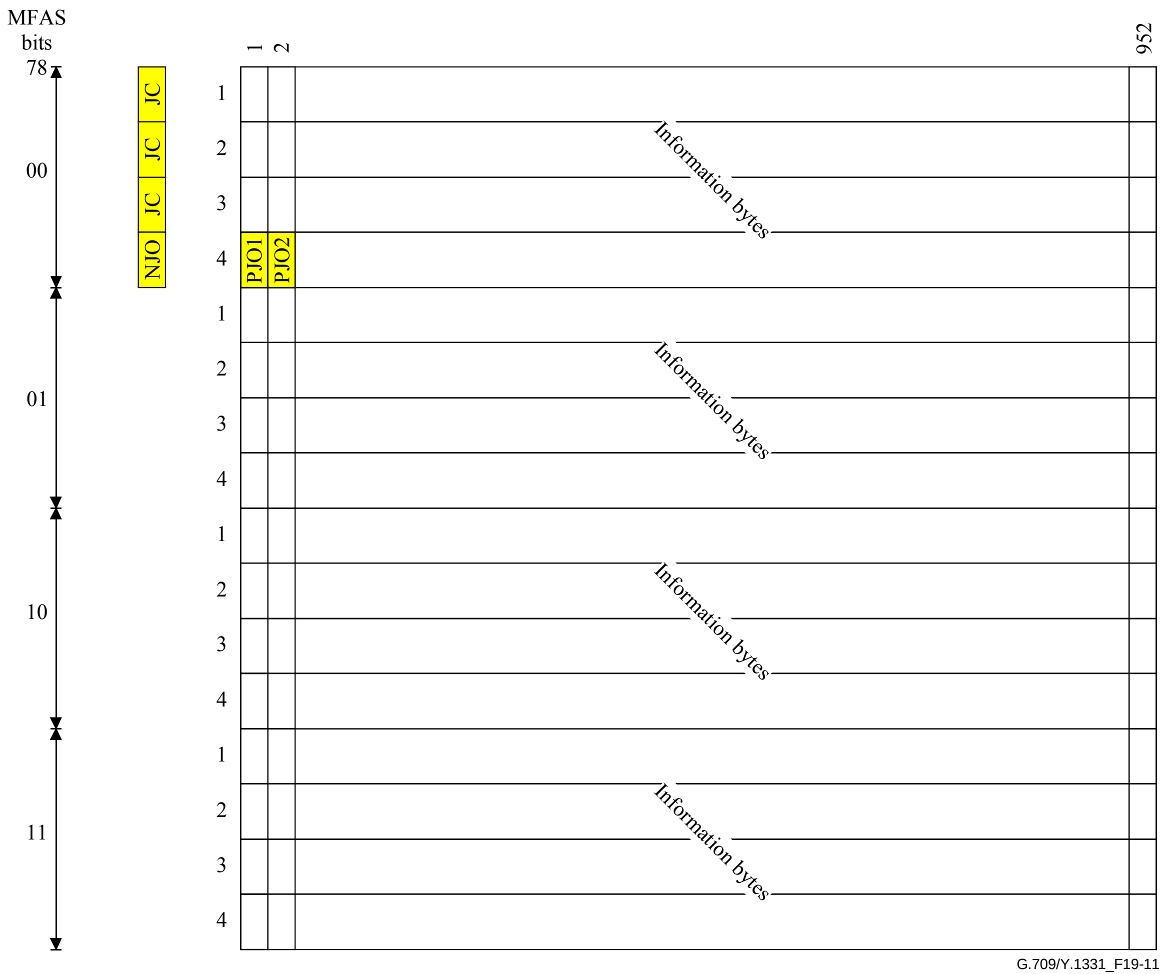


Figure 19-23A – ODTU12 frame format and mapping of ODU1 (mapping in 2.5G TS1)

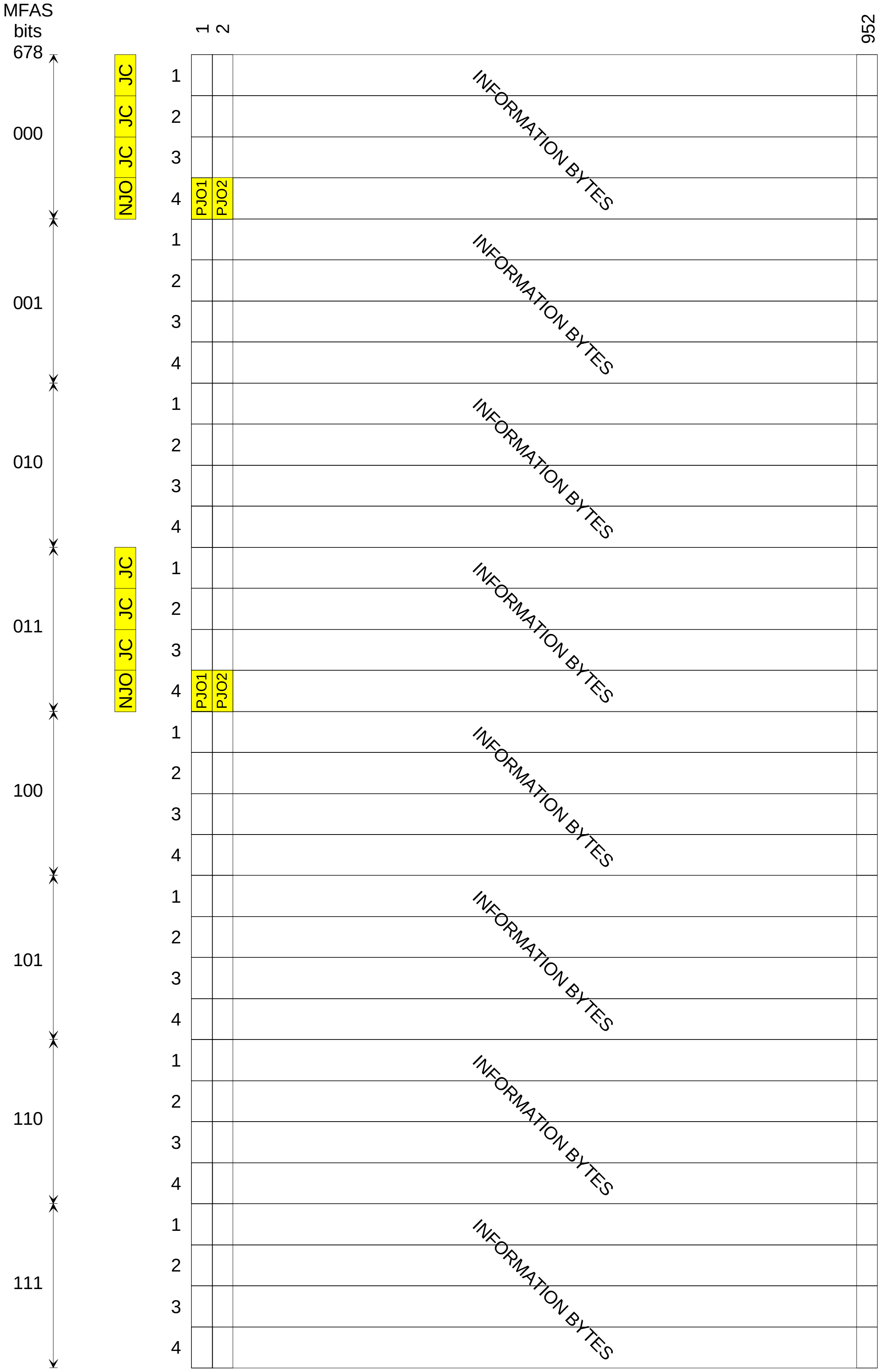


Figure 19-23B – ODTU12 frame format and mapping of ODU1 (mapping in 1.25G TS1 and TS4)

19.5.2 Mapping ODU1 into ODTU13

A byte of the ODU1 signal is mapped into an information byte of the ODTU13 (Figure 19-24A). Column 119 of the ODTU13 is fixed stuff. An all-0s pattern is inserted in the fixed stuff bytes. Once per 16 OPU3 frames, it is possible to perform either a positive or a negative justification action. The frame in which justification can be performed is related to the TSOH of the OPU3 2.5G TS in which the ODTU13 is mapped (Figure 19-2). Figure 19-24A shows the case with mapping in OPU3 2.5G TS3.

A byte of the ODU1 signal is mapped into an information byte of the ODTU13 (see Figure 19-24B). Column 119 of the ODTU13 is fixed stuff. An all-0s pattern is inserted in the fixed stuff bytes. Twice per 32 OPU3 frames, it is possible to perform either a positive or a negative justification action. The frames in which justification can be performed are related to the TSOH of the OPU3 1.25G TSs in which the ODTU13 is mapped (Figure 19-2). Figure 19-24B shows the case with mapping in OPU3 1.25G TS2 and TS25.

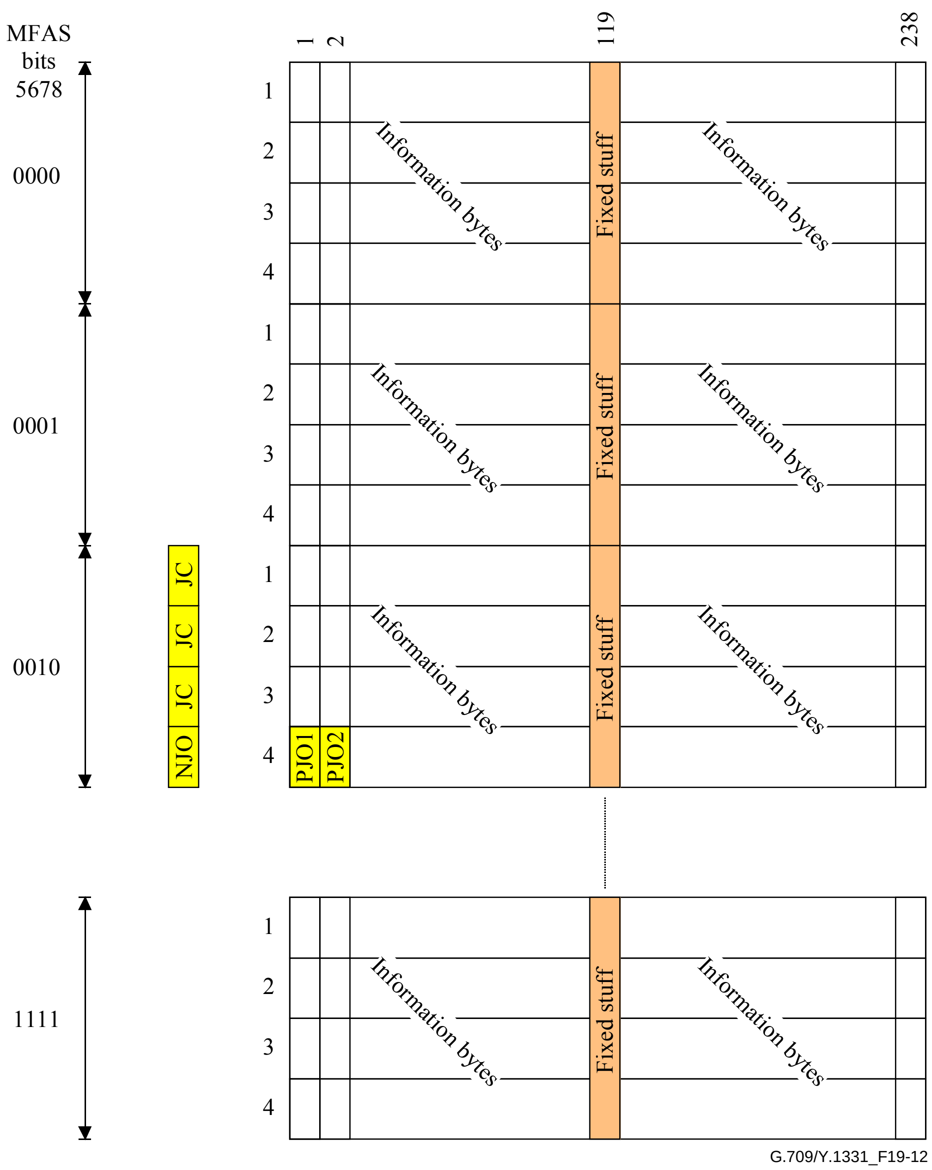


Figure 19-24A – ODTU13 frame format and mapping of ODU1 (mapping in 2.5G TS3)

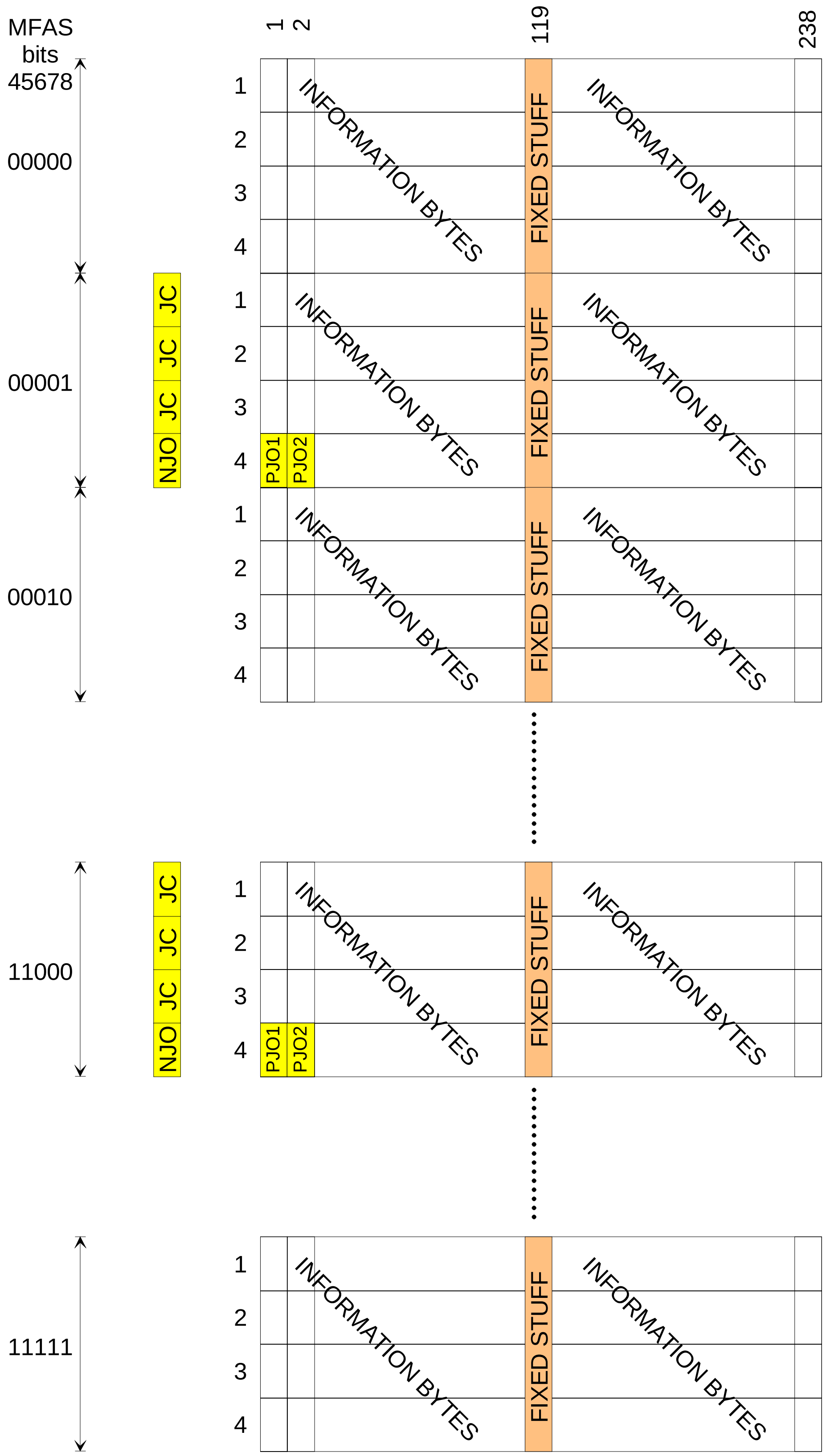


Figure 19-24B – ODTU13 frame format and mapping of ODU1 (mapping in 1.25G TS2 and TS25)

19.5.3 Mapping ODU2 into ODTU23

A byte of the ODU2 signal is mapped into an information byte of the ODTU23 (Figure 19-25A). Four times per sixteen OPU3 frames, it is possible to perform either a positive or a negative justification action. The four frames in which justification can be performed are related to the TSOH of the OPU3 2.5G TSs in which the ODTU23 is mapped (Figure 19-2). Figure 19-25A shows the case with mapping in OPU3 2.5G TS1, TS5, TS9 and TS10.

A byte of the ODU2 signal is mapped into an information byte of the ODTU23 (see Figure 19-25B). Eight times per 32 OPU3 frames, it is possible to perform either a positive or a negative justification action. The frames in which justification can be performed are related to the TSOH of the OPU3 1.25G TSs in which the ODTU23 is mapped (Figure 19-2). Figure 19-25B shows the case with mapping in OPU3 1.25G TS 1, 2, 5, 9, 10, 25, 26 and 32.

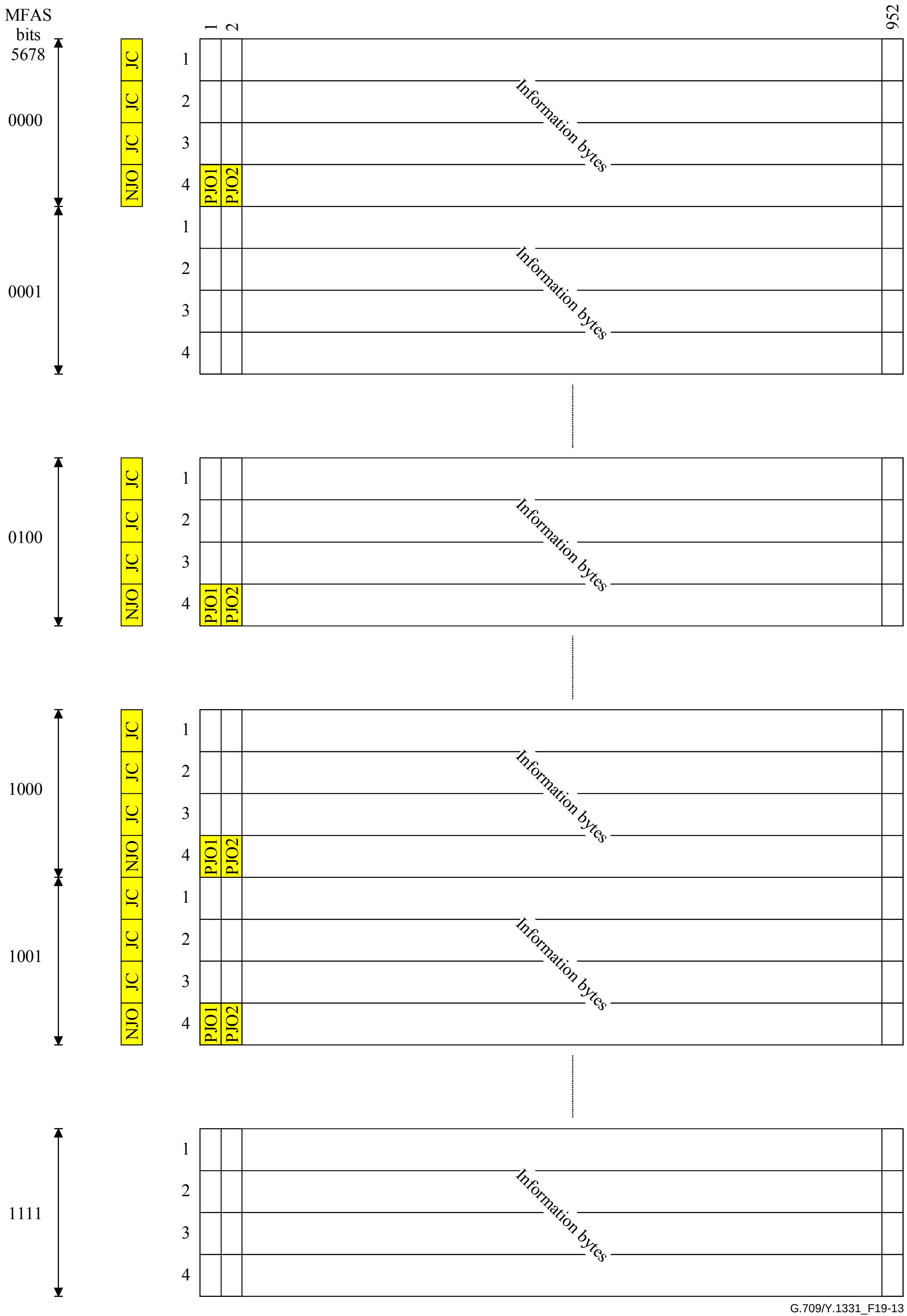


Figure 19-25A – ODTU23 frame format and mapping of ODU2 (mapping in 2.5G TS 1,5,9,10)

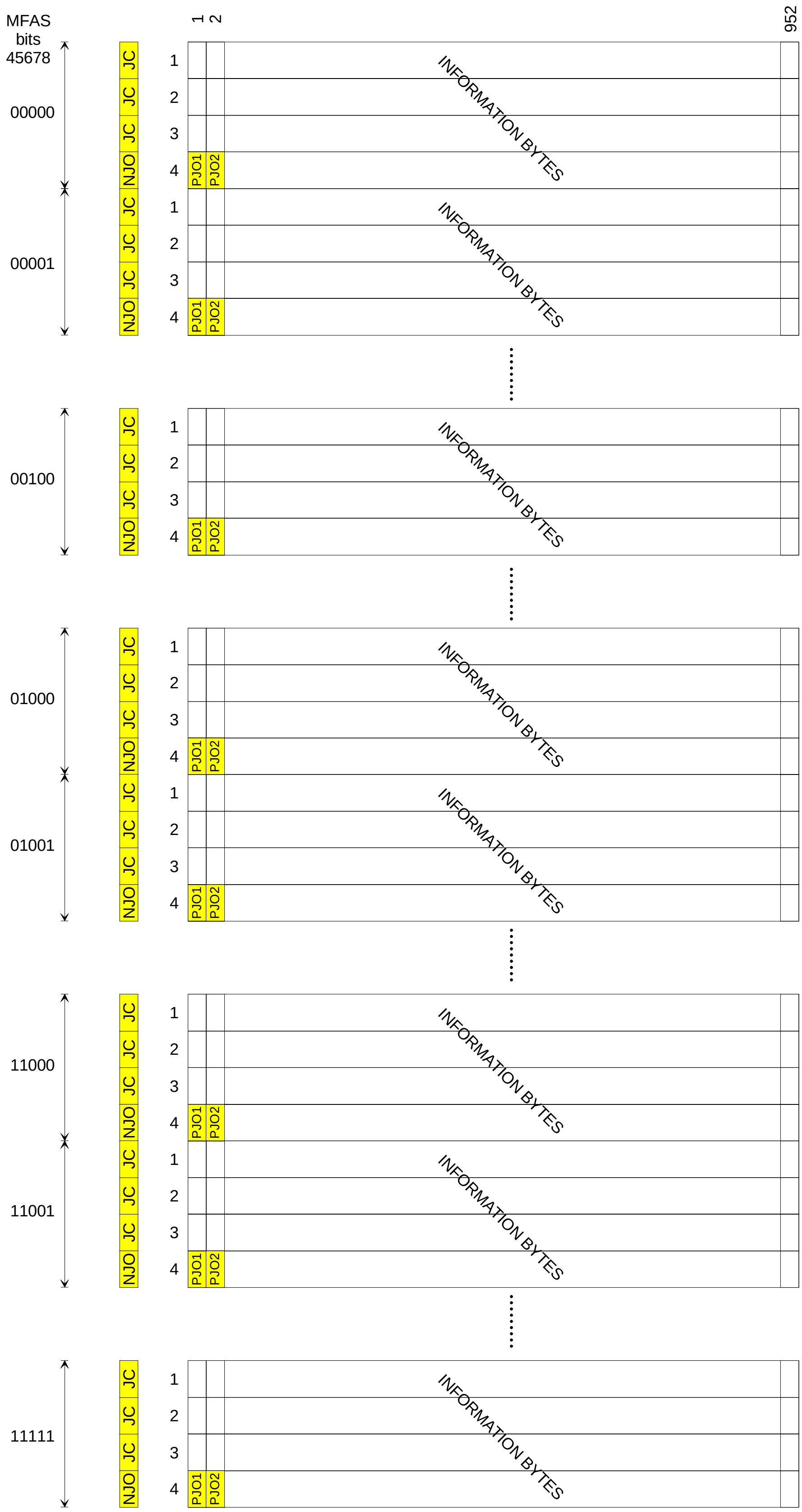


Figure 19-25B – ODTU23 frame format and mapping of ODU2 (mapping in 1.25G TS 1,2,5,9,10,25,26,32)

19.5.4 Mapping ODU0 into ODTU01

A byte of the ODU0 signal is mapped into an information byte of the ODTU01 (see Figure 19-26). Once per 2 OPU1 frames, it is possible to perform either a positive or a negative justification action.

The frame in which justification can be performed is related to the TSOH of the OPU1 TS in which the ODTU01 is mapped (Figure 19-3). Figure 19-26 shows the case with mapping in OPU1 TS1.

NOTE – The PJO2 field will always carry an information byte.

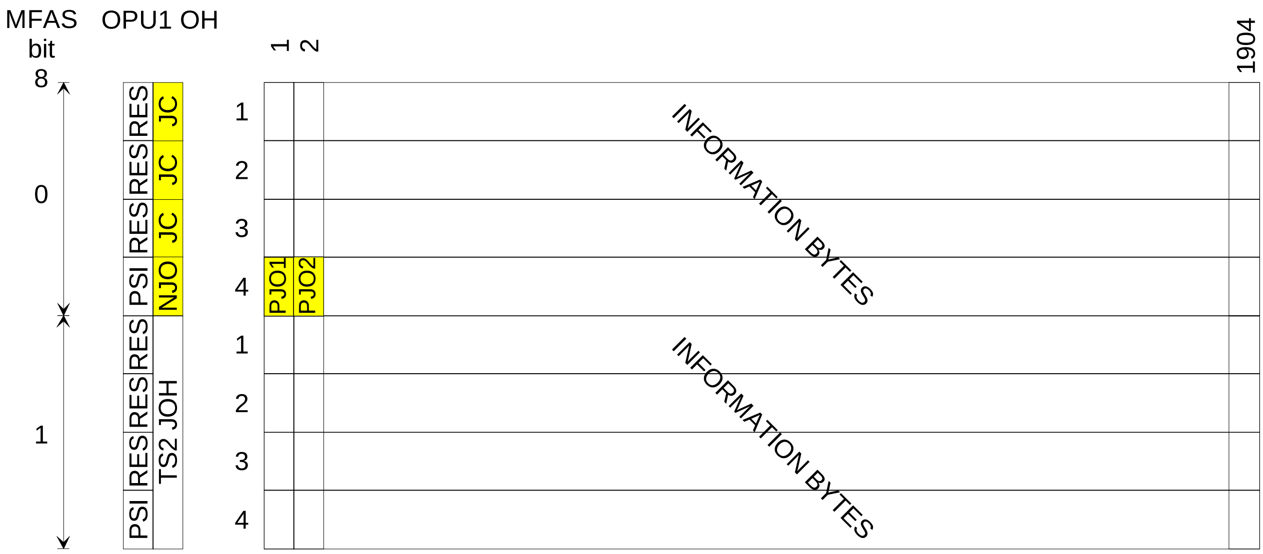


Figure 19-26 – Mapping of ODU0 in OPU1 TS1

19.6 Mapping of ODUj into ODTUk.ts

The mapping of ODUj (j = 0, 1, 2, 2e, 3, flex) signals (with up to ±100 ppm bit-rate tolerance) into the ODTUk.ts (k = 2,3,4; ts = M) signal is performed by means of a generic mapping procedure as specified in Annex D.

The OPUk and therefore the ODTUk.ts (k = 2,3,4) signals are created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the ODUj client signal.

The ODUj signal is extended with a frame alignment overhead as specified in clauses 15.6.2.1 and 15.6.2.2 and an all-0s pattern in the OTUj overhead field (see Figure 19-22).

The extended ODUj signal is adapted to the locally generated OPUk/ODTUk.ts clock by means of a generic mapping procedure (GMP) as specified in Annex D. The value of n in c_n and $C_n(t)$ and $C_{nD}(t)$ is specified in Annex D. The value of M is the number of tributary slots occupied by the ODUj; ODTUk.ts = ODTUk.M.

A group of 'M' successive extended ODUj bytes is mapped into a group of 'M' successive ODTUk.M bytes.

The generic mapping process generates for the case of ODUj (j = 0,1,2,2e,3,flex) signals once per ODTUk.M multiframe the $C_m(t)$ and $C_{nD}(t)$ information according to Annex D and encodes this information in the ODTUk.ts justification control overhead JC1/JC2/JC3 and JC4/JC5/JC6. The de-mapping process decodes $C_m(t)$ and $C_{nD}(t)$ from JC1/JC2/JC3 and JC4/JC5/JC6 and interprets $C_m(t)$ and $C_{nD}(t)$ according to Annex D. CRC-8 shall be used to protect against an error in JC1,JC2,JC3 signals. CRC-5 shall be used to protect against an error in JC4,JC5,JC6 signals.

During a signal fail condition of the incoming ODUj signal, this failed incoming signal will contain the ODUj-AIS signal as specified in clause 16.5.1. This ODUj-AIS is then mapped into the ODTUk.M.

For the case where the ODUj is received from the output of a fabric (ODU connection function), the incoming signal may contain (in the case of an open matrix connection) the ODUj-OCI signal as specified in clause 16.5.2. This ODUj-OCI signal is then mapped into the ODTUk.M.

NOTE 1 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such a unit is intentionally absent (i.e., not installed), the associated ODTUk.M signals should carry an ODUj-OCI signal. If such a unit is installed but temporarily removed as part of a repair action, the associated ODTUk.M signal should carry an ODUj-AIS signal.

A group of 'M' successive extended ODUj bytes is de-mapped from a group of 'M' successive ODTUk.M bytes.

NOTE 2 – For the case where the ODUj signal is output as an OTUj signal, frame alignment of the extracted extended ODUj signal is to be recovered to allow frame synchronous mapping of the ODUj into the OTUj signal.

During a signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the ODUj-AIS pattern as specified in clause 16.5.1 is generated as a replacement signal for the lost ODUj signal.

19.6.1 Mapping ODUj into ODTU2.M

Groups of M successive bytes of the extended ODUj (j = 0, flex) signal are mapped into a group of M successive bytes of the ODTU2.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU2.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU2.M payload area are numbered from 1 to 15232.

The ODTU2.M payload byte numbering for GMP M-byte (m-bit) blocks is illustrated in Figure 19-27. In row 1 of the ODTU2.M multiframe the first M-bytes will be labelled 1, the next M-bytes will be labelled 2, etc.

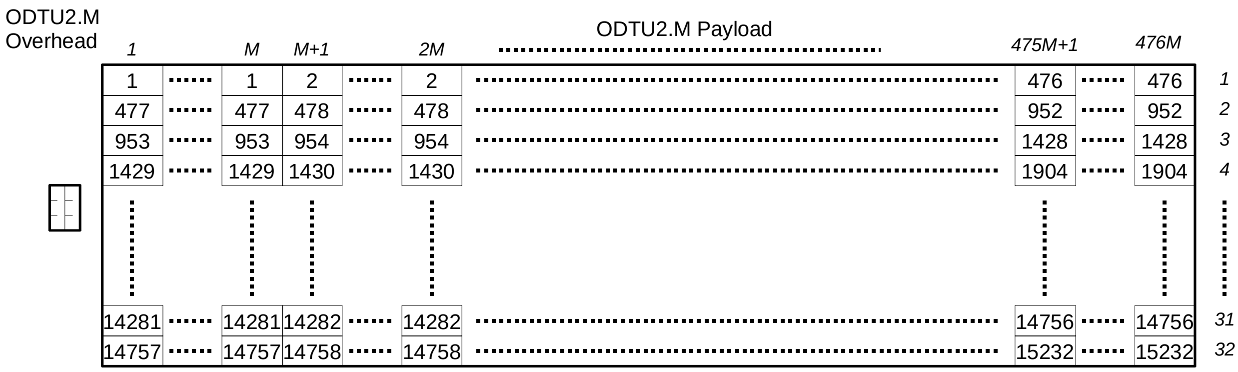


Figure 19-27 – ODTU2.M GMP byte numbering

Table 19-8 – C_m and C_n (n=8) for ODUj into ODTU2.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$ (Note)	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$ (Note)
ODU0	1	8	15167	15167.393	15168.000	15168.607	15169
ODUflex(GFP), n=1..8	n	8 × n	15227	15227.339	15229.167	15230.994	15231
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	3	24	10200	10200.928	10202.152	10203.376	10204
– ODUflex(IB DDR)	5	40	12241	12241.113	12242.582	12244.051	12245
– ODUflex(FC-400)	4	32	13006	13006.183	13007.744	13009.305	13010
– ODUflex(FC-800)	7	56	14864	14864.209	14865.993	14867.777	14868
– ODUflex(3G SDI) (2 970 000)	3	24	12118	12118.702	12120.156	12121.611	12122
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	12106	12106.595	12108.048	12109.501	12110
			Floor $C_{8,min}$ (Note)	**Minimum** c_8	**Nominal** c_8	**Maximum** c_8	**Ceiling** $C_{8,max}$ (Note)
ODU0	1	8	15167	15167.393	15168.000	15168.607	15169
ODUflex(GFP), n=1..8	n	8 × n	ODUflex(GFP) see Table 19-8A				
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	3	24	30602	30602.783	30606.456	30610.128	30611
– ODUflex(IB DDR)	5	40	61205	61205.566	61212.911	61220.257	61221
– ODUflex(FC-400)	4	32	52024	52024.731	52030.974	52037.218	52038
– ODUflex(FC-800)	7	56	104049	104049.462	104061.949	104074.437	104075
– ODUflex(3G SDI) (2 970 000)	3	24	36356	36356.106	36360.469	36364.833	36365
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	36319	36319.786	36324.145	36328.504	36329

NOTE – Floor $C_{m,min}$, Floor $C_{n,min}$ (n=8), Ceiling $C_{m,max}$ and Ceiling $C_{n,max}$ (n=8) values represent the boundaries of ODUj/ODTU2.M ppm offset combinations (i.e., min. ODUj/max. ODTU and max. ODUj/min. ODTU). In steady state, given instances of ODUj/ODTU offset combinations should not result in generated C_n and C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 19-8A –C_n (n=8) for ODUj (j=flex, ODU2.ts) into ODTU2.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	1	8	15227	15227.339	15229.167	15230.994	15231
ODUflex(GFP)	2	16	30454	30454.679	30458.334	30461.989	30462
ODUflex(GFP)	3	24	45682	45682.018	45687.501	45692.983	45693
ODUflex(GFP)	4	32	60909	60909.358	60916.667	60923.978	60924
ODUflex(GFP)	5	40	76136	76136.697	76145.834	76154.972	76155
ODUflex(GFP)	6	48	91364	91364.036	91375.001	91385.966	91386
ODUflex(GFP)	7	56	106591	106591.376	106604.168	106616.961	106617
ODUflex(GFP)	8	64	121818	121818.715	121833.335	121847.955	121848

19.6.2 Mapping ODUj into ODTU3.M

Groups of M successive bytes of the extended ODUj (j = 0, 2e, flex) signal are mapped into a group of M successive bytes of the ODTU3.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU3.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU3.M payload area are numbered from 1 to 15232.

The ODTU3.M payload byte numbering for GMP M-byte (m-bit) blocks is illustrated in Figure 19-28. In row 1 of the ODTU3.M multiframe the first M-bytes will be labelled 1, the next M-bytes will be labelled 2, etc.

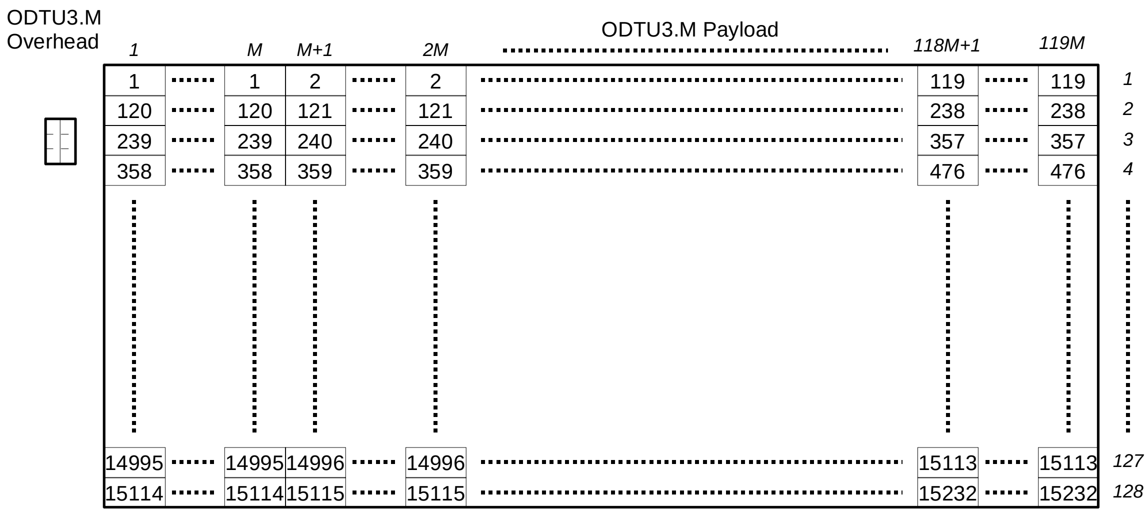


Figure 19-28 – ODTU3.M GMP byte numbering

Table 19-9 – C_m and C_n (n=8) for ODUj into ODTU3.M

ODUj signal	M	m=8× M	Floor $C_{m,min}$ (Note)	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$ (Note)
ODU0	1	8	15103	15103.396	15104.000	15104.604	15105
ODU2e	9	72	14026	14026.026	14027.709	14029.392	14030
ODUflex(GFP), n=1..8, ODU2.ts	n	8 × n	15163	15163.089	15164.909	15166.729	15167
ODUflex(GFP), n=9..32, ODU3.ts	n	8 × n	15227	15227.339	15229.167	15230.994	15231
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	3	24	10157	10157.886	10159.105	10160.324	10161
– ODUflex(IB DDR)	5	40	12189	12189.463	12190.926	12192.389	12193
– ODUflex(IB QDR)	9	72	13543	13543.848	13545.473	13547.099	13548
– ODUflex(FC-400)	4	32	12951	12951.304	12952.859	12954.413	12955
– ODUflex(FC-800)	7	56	14801	14801.491	14803.267	14805.043	14806
– ODUflex(FC-1600)	12	96	14246	14246.435	14248.144	14249.854	14250
– ODUflex(3G SDI) (2 970 000)	3	24	12067	12067.568	12069.016	12070.465	12071
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	12055	12055.513	12056.960	12058.406	12059
			Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODU0	1	8	15103	15103.396	15104.000	15104.604	15105
ODU2e	9	72	126234	126234.232	126249.381	126264.532	126265
ODUflex(GFP), n=1..32	n	8 × n	ODUflex(GFP),see Tables 19-9A and 19-9B				
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	3	24	30473	30473.657	30477.314	30480.972	30481
– ODUflex(IB DDR)	5	40	60947	60947.314	60954.629	60961.943	60962
– ODUflex(IB QDR)	9	72	121894	121894.629	121909.258	121923.887	121924
– ODUflex(FC-400)	4	32	51805	51805.217	51811.434	51817.652	51818
– ODUflex(FC-800)	7	56	103610	103610.434	103622.869	103635.304	103636
– ODUflex(FC-1600)	12	96	170957	170957.217	170977.734	170998.251	170999
– ODUflex(3G SDI) (2 970 000)	3	24	36202	36202.705	36207.049	36211.394	36212
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	36166	36166.538	36170.879	36175.219	36176

Table 19-9 – C_m and C_n (n=8) for ODUj into ODTU3.M

NOTE – Floor $C_{m,min}$, Floor $C_{n,min}$ (n=8), Ceiling $C_{m,max}$ and Ceiling $C_{n,max}$ (n=8) values represent the boundaries of ODUj/ODTU3.M ppm offset combinations (i.e., min. ODUj/max. ODTU and max. ODUj/min. ODTU). In steady state, given instances of ODUj/ODTU offset combinations should not result in generated C_n and C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 19-9A –C_n (n=8) for ODUj (j=flex, ODU2.ts) into ODTU3.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	1	8	15163	15163.089	15164.909	15166.729	15167
ODUflex(GFP)	2	16	30326	30326.178	30329.818	30333.457	30334
ODUflex(GFP)	3	24	45489	45489.267	45494.726	45500.186	45501
ODUflex(GFP)	4	32	60652	60652.356	60659.635	60666.914	60667
ODUflex(GFP)	5	40	75815	75815.445	75824.544	75833.643	75834
ODUflex(GFP)	6	48	90978	90978.534	90989.453	91000.372	91001
ODUflex(GFP)	7	56	106141	106141.623	106154.361	106167.100	106168
ODUflex(GFP)	8	64	121304	121304.712	121319.270	121333.829	121334

Table 19-9B –Cn (n=8) for ODUj (j=flex, ODU3.ts) into ODTU3.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	9	72	137045	137045.095	137062.502	137079.909	137080
ODUflex(GFP)	10		152272	152272.176	152291.668	152311.162	152312
ODUflex(GFP)	11		167499	167499.226	167520.835	167542.446	167543
ODUflex(GFP)	12		182726	182726.245	182750.002	182773.760	182774
ODUflex(GFP)	13		197953	197953.234	197979.169	198005.105	198006
ODUflex(GFP)	14		213180	213180.193	213208.336	213236.480	213237
ODUflex(GFP)	15		228407	228407.121	228437.503	228467.886	228468
ODUflex(GFP)	16		243634	243634.019	243666.670	243699.322	243700
ODUflex(GFP)	17		258860	258860.886	258895.836	258930.788	258931
ODUflex(GFP)	18		274087	274087.723	274125.003	274162.285	274163
ODUflex(GFP)	19		289314	289314.529	289354.170	289393.812	289394
ODUflex(GFP)	20		304541	304541.305	304583.337	304625.370	304626
ODUflex(GFP)	21		319768	319768.051	319812.504	319856.959	319857
ODUflex(GFP)	22		334994	334994.766	335041.671	335088.577	335089
ODUflex(GFP)	23		350221	350221.450	350270.838	350320.227	350321
ODUflex(GFP)	24		365448	365448.104	365500.004	365551.906	365552

Table 19-9B –Cn (n=8) for ODUj (j=flex, ODU3.ts) into ODTU3.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	25		380674	380674.728	380729.171	380783.617	380784
ODUflex(GFP)	26		395901	395901.321	395958.338	396015.357	396016
ODUflex(GFP)	27		411127	411127.884	411187.505	411247.128	411248
ODUflex(GFP)	28		426354	426354.416	426416.672	426478.930	426479
ODUflex(GFP)	29		441580	441580.918	441645.839	441710.762	441711
ODUflex(GFP)	30		456807	456807.389	456875.005	456942.624	456943
ODUflex(GFP)	31		472033	472033.830	472104.172	472174.517	472175
ODUflex(GFP)	32		487260	487260.241	487333.339	487406.441	487407

19.6.3 Mapping ODUj into ODTU4.M

Groups of M successive bytes of the extended ODUj (j = 0, 1, 2, 2e, 3, flex) signal are mapped into a group of M successive bytes of the ODTU4.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU4.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU4.M payload area are numbered from 1 to 15200.

The ODTU4.M payload byte numbering for GMP M-byte (m-bit) blocks is illustrated in Figure 19-29. In row 1 of the ODTU4.M multiframe the first M-bytes will be labelled 1, the next M-bytes will be labelled 2, etc.

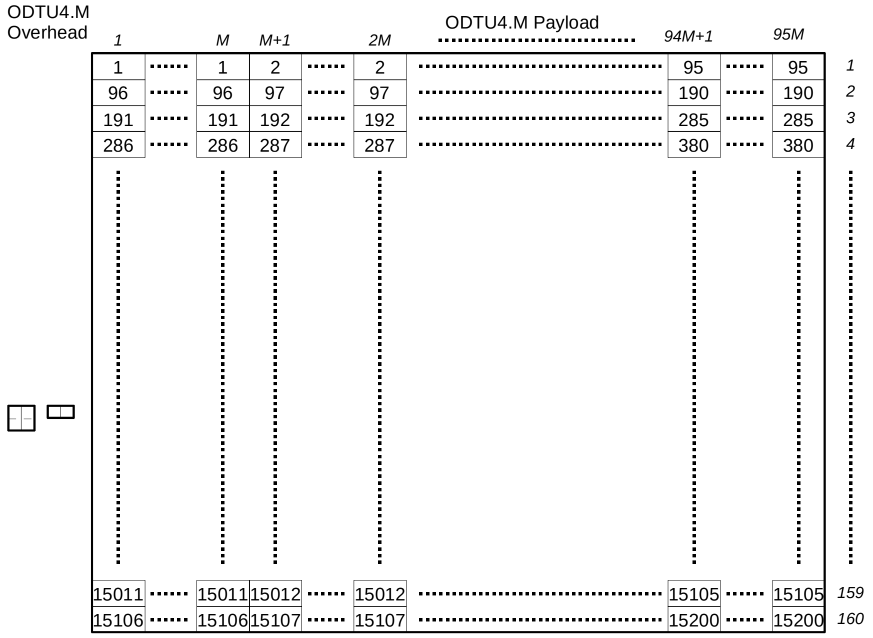


Figure 19-29 – ODTU4.M GMP byte numbering

Table 19-10 – Cm and Cn (n=8) for ODUj into ODTU4.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$ (Note)	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$ (Note)
ODU0	1	8	14527	14527.419	14528.000	14528.581	14529
ODU1	2	16	14588	14588.458	14589.042	14589.626	14590
ODU2	8	64	14650	14650.013	14650.599	14651.185	14652
ODU2e	8	64	15177	15177.527	15179.348	15181.170	15182
ODU3	31	248	15186	15186.673	15187.280	15187.888	15188
ODUflex(GFP), n=1..8, ODU2.ts	n	8 × n	14584	14584.836	14586.586	14588.336	14589
ODUflex(GFP), n=9..32, ODU3.ts	n	8 × n	14646	14646.636	14648.394	14650.151	14651
ODUflex(GFP), n=33..80, ODU4.ts	n	8 × n	15195	15195.349	15197.173	15198.996	15199
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	2	16	14655	14655.763	14657.522	14659.281	14660
– ODUflex(IB DDR)	4	32	14655	14655.763	14657.522	14659.281	14660
– ODUflex(IB QDR)	8	64	14655	14655.763	14657.522	14659.281	14660
– ODUflex(FC-400)	4	32	12457	12457.399	12458.894	12460.389	12461
– ODUflex(FC-800)	7	56	14237	14237.027	14238.736	14240.444	14241
– ODUflex(FC-1600)	11	88	14948	14948.878	14950.672	14952.467	14953
– ODUflex(3G SDI) (2 970 000)	3	24	11607	11607.364	11608.757	11610.150	11611
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	11595	11595.769	11597.160	11598.552	11599
			Floor $C_{8,min}$ (Note)	**Minimum** c_8	**Nominal** c_8	**Maximum** c_8	**Ceiling** $C_{8,max}$ (Note)
ODU0	1	8	14527	14527.419	14528.000	14528.581	14529
ODU1	2	16	29176	29176.917	29178.084	29179.251	29180
ODU2	8	64	117200	117200.105	117204.793	117209.482	117210
ODU2e	8	64	121420	121420.214	121434.786	121449.359	121450
ODU3	31	248	470786	470786.863	470805.695	470824.528	470825
ODUflex(GFP), n=1..80	n	8 × n	ODUflex(GFP), see Tables 19-10A, 19-10B and 19-10C				
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(IB SDR)	2	16	29311	29311.526	29315.044	29318.562	29319
– ODUflex(IB DDR)	4	32	58623	58623.052	58630.088	58637.124	58638
– ODUflex(IB QDR)	8	64	117246	117246.105	117260.176	117274.247	117275
– ODUflex(FC-400)	4	32	49829	49829.595	49835.575	49841.555	49842
– ODUflex(FC-800)	7	56	99659	99659.189	99671.149	99683.110	99684
– ODUflex(FC-1600)	11	88	164437	164437.662	164457.396	164477.132	164478

Table 19-10 – Cm and Cn (n=8) for ODUj into ODTU4.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$ (Note)	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$ (Note)
– ODUflex(3G SDI) (2 970 000)	3	24	34822	34822.093	34826.272	34830.451	34831
– ODUflex(3G SDI) (2 970 000/1.001)	3	24	34787	34787.306	34791.481	34795.656	34796

NOTE – Floor $C_{m,min}$, Floor $C_{n,min}$ (n=8), Ceiling $C_{m,max}$ and Ceiling $C_{n,max}$ (n=8) values represent the boundaries of ODUj/ODTU4.M ppm offset combinations (i.e., min. ODUj/max. ODTU and max. ODUj/min. ODTU). In steady state, given instances of ODUj/ODTU offset combinations should not result in generated C_n and C_m values throughout this range but rather should be within as small a range as possible. Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

Table 19-10A –C_n (n=8) for ODUj (j=flex, ODU2.ts) into ODTU4.M

ODUj signal	M	m=8×M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	1	8	14584	14584.836	14586.586	14588.336	14589
ODUflex(GFP)	2	16	29169	29169.671	29173.172	29176.673	29177
ODUflex(GFP)	3	24	43754	43754.507	43759.758	43765.009	43766
ODUflex(GFP)	4	32	58339	58339.342	58346.344	58353.346	58354
ODUflex(GFP)	5	40	72924	72924.178	72932.930	72941.682	72942
ODUflex(GFP)	6	48	87509	87509.014	87519.516	87530.018	87531
ODUflex(GFP)	7	56	102093	102093.849	102106.102	102118.355	102119
ODUflex(GFP)	8	64	116678	116678.685	116692.688	116706.691	116707

Table 19-10B –Cn (n=8) for ODUj (j=flex, ODU3.ts) into ODTU4.M

ODUj signal	M	m=8×M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	9	72	131819	131819.722	131835.542	131851.362	131852
ODUflex(GFP)	10		146466	146466.358	146483.935	146501.514	146502
ODUflex(GFP)	11		161112	161112.993	161132.329	161151.665	161152
ODUflex(GFP)	12		175759	175759.629	175780.722	175801.817	175802
ODUflex(GFP)	13		190406	190406.265	190429.116	190451.968	190452
ODUflex(GFP)	14		205052	205052.901	205077.510	205102.119	205103
ODUflex(GFP)	15		219699	219699.536	219725.903	219752.271	219753
ODUflex(GFP)	16		234346	234346.172	234374.297	234402.422	234403

Table 19-10B –Cn (n=8) for ODUj (j=flex, ODU3.ts) into ODTU4.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	17		248992	248992.808	249022.690	249052.573	249053
ODUflex(GFP)	18		263639	263639.444	263671.084	263702.725	263703
ODUflex(GFP)	19		278286	278286.080	278319.477	278352.876	278353
ODUflex(GFP)	20		292932	292932.715	292967.871	293003.028	293004
ODUflex(GFP)	21		307579	307579.351	307616.264	307653.179	307654
ODUflex(GFP)	22		322225	322225.987	322264.658	322303.330	322304
ODUflex(GFP)	23		336872	336872.623	336913.051	336953.482	336954
ODUflex(GFP)	24		351519	351519.258	351561.445	351603.633	351604
ODUflex(GFP)	25		366165	366165.894	366209.838	366253.784	366254
ODUflex(GFP)	26		380812	380812.530	380858.232	380903.936	380904
ODUflex(GFP)	27		395459	395459.166	395506.625	395554.087	395555
ODUflex(GFP)	28		410105	410105.801	410155.019	410204.239	410205
ODUflex(GFP)	29		424752	424752.437	424803.413	424854.390	424855
ODUflex(GFP)	30		439399	439399.073	439451.806	439504.541	439505
ODUflex(GFP)	31		454045	454045.709	454100.200	454154.693	454155
ODUflex(GFP)	32		468692	468692.344	468748.593	468804.844	468805

Table 19-10C –Cn (n=8) for ODUj (j=flex, ODU4.ts) into ODTU4.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	33		15195	15195.349	15197.173	15198.996	15199
ODUflex(GFP)	34		30390	30390.698	30394.346	30397.993	30398
ODUflex(GFP)	35		45586	45586.048	45591.518	45596.989	45597
ODUflex(GFP)	36		60781	60781.397	60788.691	60795.986	60796
ODUflex(GFP)	37		75976	75976.746	75985.864	75994.982	75995
ODUflex(GFP)	38		91172	91172.004	91183.037	91194.070	91195
ODUflex(GFP)	39		106367	106367.231	106380.210	106393.188	106394
ODUflex(GFP)	40		121562	121562.429	121577.382	121592.337	121593
ODUflex(GFP)	41		136757	136757.595	136774.555	136791.516	136792
ODUflex(GFP)	42		151952	151952.732	151971.728	151990.725	151991
ODUflex(GFP)	43		167147	167147.838	167168.901	167189.965	167190
ODUflex(GFP)	44		182342	182342.914	182366.074	182389.235	182390
ODUflex(GFP)	45		197537	197537.959	197563.246	197588.535	197589
ODUflex(GFP)	46		212732	212732.974	212760.419	212787.866	212788
ODUflex(GFP)	47		227927	227927.958	227957.592	227987.227	227988

Table 19-10C –Cn (n=8) for ODUj (j=flex, ODU4.ts) into ODTU4.M

ODUj signal	M	m=8× M	Floor $C_{8,min}$ (Note)	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$ (Note)
ODUflex(GFP)	48		243122	243122.912	243154.765	243186.619	243187
ODUflex(GFP)	49		258317	258317.836	258351.938	258386.041	258387
ODUflex(GFP)	50		273512	273512.729	273549.110	273585.493	273586
ODUflex(GFP)	51		288707	288707.592	288746.283	288784.976	288785
ODUflex(GFP)	52		303902	303902.424	303943.456	303984.489	303985
ODUflex(GFP)	53		319097	319097.227	319140.629	319184.033	319185
ODUflex(GFP)	54		334291	334291.998	334337.802	334383.607	334384
ODUflex(GFP)	55		349486	349486.740	349534.974	349583.211	349584
ODUflex(GFP)	56		364681	364681.450	364732.147	364782.846	364783
ODUflex(GFP)	57		379876	379876.131	379929.320	379982.511	379983
ODUflex(GFP)	58		395070	395070.781	395126.493	395182.207	395183
ODUflex(GFP)	59		410265	410265.401	410323.666	410381.933	410382
ODUflex(GFP)	60		425459	425459.990	425520.838	425581.689	425582
ODUflex(GFP)	61		440654	440654.549	440718.011	440781.476	440782
ODUflex(GFP)	62		455849	455849.078	455915.184	455981.293	455982
ODUflex(GFP)	63		471043	471043.576	471112.357	471181.141	471182
ODUflex(GFP)	64		486238	486238.044	486309.530	486381.019	486382
ODUflex(GFP)	65		15195	15195.349	15197.173	15198.996	15199
ODUflex(GFP)	66		30390	30390.698	30394.346	30397.993	30398
ODUflex(GFP)	67		45586	45586.048	45591.518	45596.989	45597
ODUflex(GFP)	68		60781	60781.397	60788.691	60795.986	60796
ODUflex(GFP)	69		75976	75976.746	75985.864	75994.982	75995
ODUflex(GFP)	70		91172	91172.004	91183.037	91194.070	91195
ODUflex(GFP)	71		106367	106367.231	106380.210	106393.188	106394
ODUflex(GFP)	72		121562	121562.429	121577.382	121592.337	121593
ODUflex(GFP)	73		136757	136757.595	136774.555	136791.516	136792
ODUflex(GFP)	74		151952	151952.732	151971.728	151990.725	151991
ODUflex(GFP)	75		167147	167147.838	167168.901	167189.965	167190
ODUflex(GFP)	76		182342	182342.914	182366.074	182389.235	182390
ODUflex(GFP)	77		197537	197537.959	197563.246	197588.535	197589
ODUflex(GFP)	78		212732	212732.974	212760.419	212787.866	212788
ODUflex(GFP)	79		227927	227927.958	227957.592	227987.227	227988
ODUflex(GFP)	80		243122	243122.912	243154.765	243186.619	243187

Annex A

Forward error correction using 16-byte interleaved RS(255,239) codecs

(This annex forms an integral part of this Recommendation.)

The forward error correction for the OTU-k uses 16-byte interleaved codecs using a Reed-Solomon RS(255,239) code. The RS(255,239) code is a non-binary code (the FEC algorithm operates on byte symbols) and belongs to the family of systematic linear cyclic block codes.

For FEC processing, an OTU row is separated into 16 sub-rows using byte-interleaving as shown in Figure A.1. Each FEC encoder/decoder processes one of these sub-rows. The FEC parity check bytes are calculated over the information bytes 1 to 239 of each sub-row and transmitted in bytes 240 to 255 of the same sub-row.

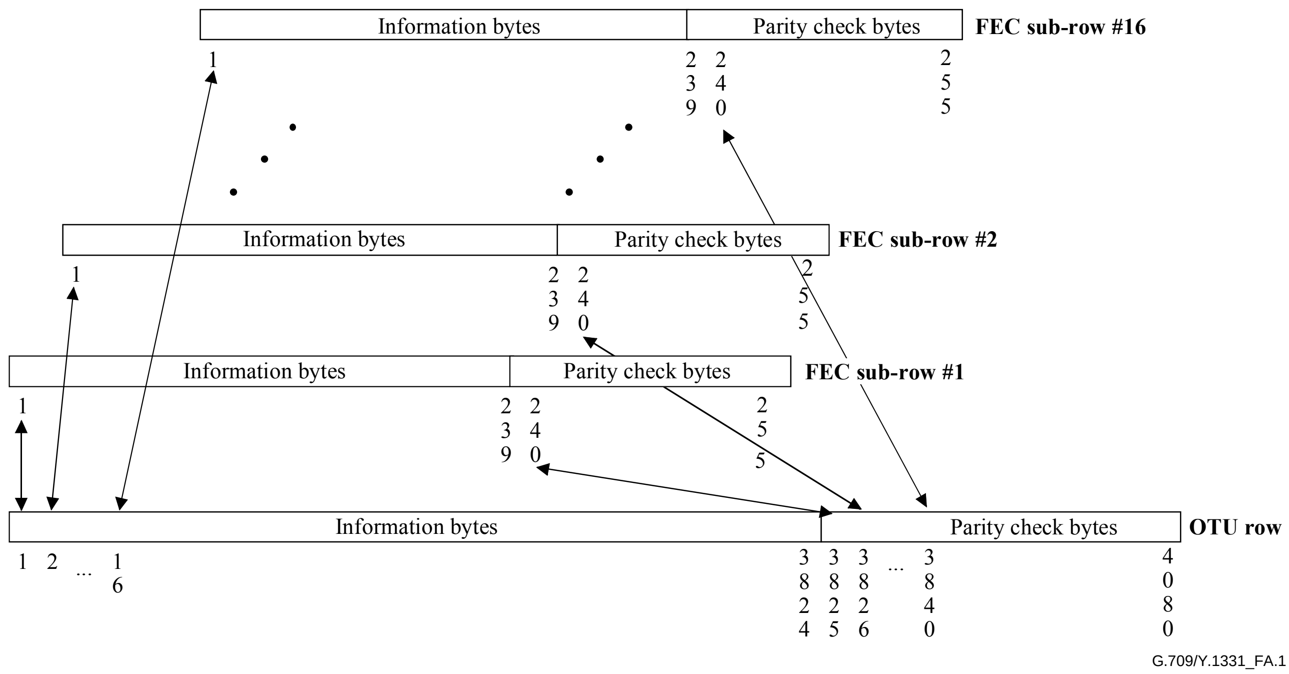


Figure A.1 – FEC sub-rows

The bytes in an OTU row belonging to FEC sub-row X are defined by: $X + 16 \times (i - 1)$ (for $i = 1...255$).

The generator polynomial of the code is given by:

$$G(z) = \prod_{i=0}^{15} (z - \alpha^i)$$

where α is a root of the binary primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

The FEC code word (see Figure A.2) consists of information bytes and parity bytes (FEC redundancy) and is represented by the polynomial:

$$C(z) = I(z) + R(z)$$

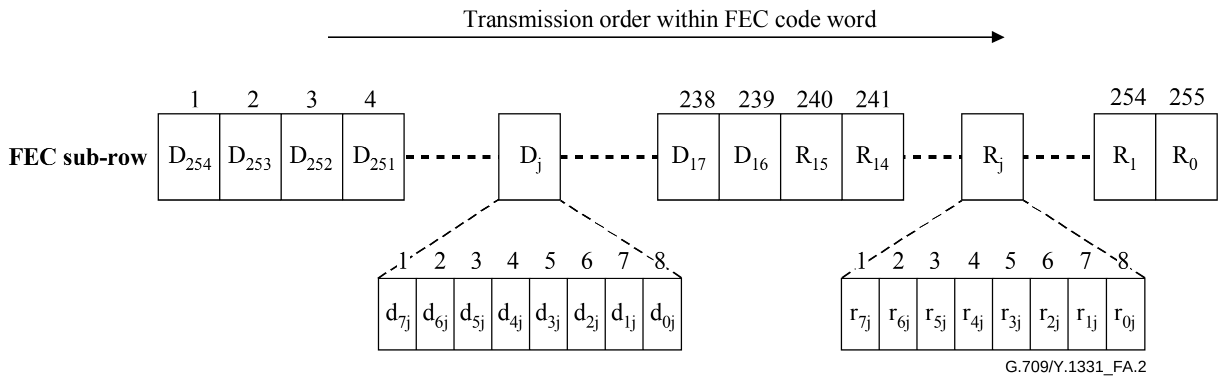


Figure A.2 – FEC code word

Information bytes are represented by:

$$I(z) = D_{254} . z^{254} + D_{253} . z^{253} + ... + D_{16} . z^{16}$$

Where D_j (j = 16 to 254) is the information byte represented by an element out of GF(256) and:

$$D_j = d_{7j} \cdot \alpha^7 + d_{6j} \cdot \alpha^6 + ... + d_{1j} \cdot \alpha^1 + d_{0j}$$

Bit d_{7j} is the MSB and d_{0j} the LSB of the information byte.

D_{254} corresponds to byte 1 in the FEC sub-row and D_{16} to byte 239.

Parity bytes are represented by:

$$R(z) = R_{15} \cdot z^{15} + R_{14} \cdot z^{14} + ... + R_1 \cdot z^1 + R_0$$

Where R_j (j = 0 to 15) is the parity byte represented by an element out of GF(256) and:

$$R_j = r_{7j} \cdot \alpha^7 + r_{6j} \cdot \alpha^6 + ... + r_{1j} \cdot \alpha^1 + r_{0j}$$

Bit r_{7j} is the MSB and r_{0j} the LSB of the parity byte.

R_{15} corresponds to the byte 240 in the FEC sub-row and R_0 to byte 255.

R(z) is calculated by:

$$R(z) = I(z) \bmod G(z)$$

where "mod" is the modulo calculation over the code generator polynomial *G(z)* with elements out of the GF(256). Each element in GF(256) is defined by the binary primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

The Hamming distance of the RS(255,239) code is d_{min} = 17. The code can correct up to 8 symbol errors in the FEC code word when it is used for error correction. The FEC can detect up to 16 symbol errors in the FEC code word when it is used for error detection capability only.

Annex B

Adapting 64B/66B encoded clients via transcoding into 513B code blocks

(This annex forms an integral part of this Recommendation.)

Clients using 64B/66B coding can be adapted in a codeword and timing transparent mapping via transcoding into 513B code blocks to reduce the bit rate that is required to transport the signal. The resulting 513B blocks can be mapped in one of several ways depending on the requirements of the client and the available bandwidth of the container into which the client is mapped. This mapping can be applied to serial or parallel client interfaces.

B.1 Transmission order

The order of transmission of information in all the diagrams in this annex is first from left to right and then from top to bottom.

B.2 Client frame recovery

For 40GBASE-R and 100GBASE-R clients, framing recovery consists of the recovering 64B/66B block lock per the state diagram in Figure 82-10 of [IEEE 802.3ba]. For other 64B/66B encoded clients, block lock is achieved as per the state diagram in Figure 49-12 of [IEEE 802.3]. Descrambling is performed as per the process shown in Figure 49-10 of [IEEE 802.3].

Each 66B codeword (after block lock) is one of the following:

– a set of eight data bytes with a sync header of "01";

– a control block (possibly including seven or fewer data octets) beginning with a sync header of "10".

The 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane markers) according to the polynomial $G(x) = 1 + x^{39} + x^{58}$. The 64B/66B PCS receive process will descramble the bits other than (1) the sync header of 66B data and control blocks, and (2) the PCS lane markers.

Figure B.1 illustrates the ordering of 64B/66B code blocks after the completion of the recovering process for an interface.

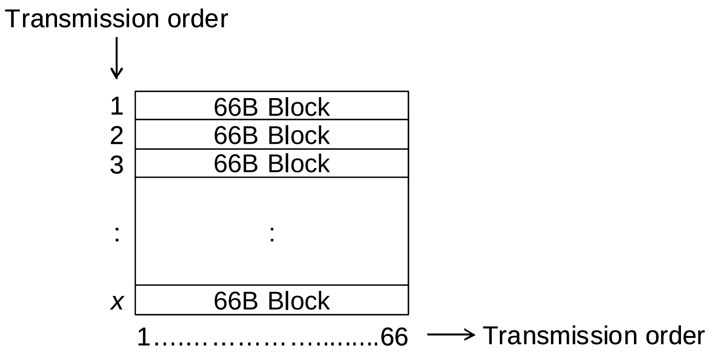


Figure B.1 – Stream of 64B/66B code blocks for transcoding

B.3 Transcoding from 66B blocks to 513B blocks

The transcoding process at the encoder operates on an input sequence of 66B code blocks.

66B control blocks (after descrambling) follow the format shown in Figure B.2.

A group of eight 66B blocks is encoded into a single 513B block. The format is illustrated in Figure B.3.

Input Data / Data Block Format	SYNC (bit 0)	SYNC (bit 1)	Block Payload (bits 2–9)								4-bit code
D0D1D2D3D4D5D6D7	0	1	D0	D1	D2	D3	D4	D5	D6	D7	
Control block formats			**Block type field**								**4-bit code**
C0C1C2C3C4C5C6C7	1	0	0x1e	C0	C1	C2	C3	C4	C5	C6	C7	0001
C0C1C2C3O4D5D6D7	1	0	0x2d	C0	C1	C2	C3	O4	D5	D6	D7	0010
C0C1C2C3S4D5D6D7	1	0	0x33	C0	C1	C2	C3		D5	D6	D7	0111
O0D1D2D3S4D5D6D7	1	0	0x66	D1	D2	D3	O0		D5	D6	D7	1011
O0D1D2D3O4D5D6D7	1	0	0x55	D1	D2	D3	O0	O4	D5	D6	D7	1101
S0D1D2D3D4D5D6D7	1	0	0x78	D1	D2	D3	D4	D5	D6	D7		1110
O0D1D2D3C4C5C6C7	1	0	0x4b	D1	D2	D3	O0	C4	C5	C6	C7	1000
T0C1C2C3C4C5C6C7	1	0	0x87		C1	C2	C3	C4	C5	C6	C7	0011
D0T1C2C3C4C5C6C7	1	0	0x99	D0		C2	C3	C4	C5	C6	C7	0101
D0D1T2C3C4C5C6C7	1	0	0xaa	D0	D1		C3	C4	C5	C6	C7	1001
D0D1D2T3C4C5C6C7	1	0	0xb4	D0	D1	D2		C4	C5	C6	C7	1010
D0D1D2D3T4C5C6C7	1	0	0xcc	D0	D1	D2	D3		C5	C6	C7	1100
D0D1D2D3D4T5C6C7	1	0	0xd2	D0	D1	D2	D3	D4		C6	C7	0110
D0D1D2D3D4D5T6C7	1	0	0xe1	D0	D1	D2	D3	D4	D5		C7	0000
D0D1D2D3D4D5D6T7	1	0	0xff	D0	D1	D2	D3	D4	D5	D6		1111

Figure B.2 – 66B Block coding

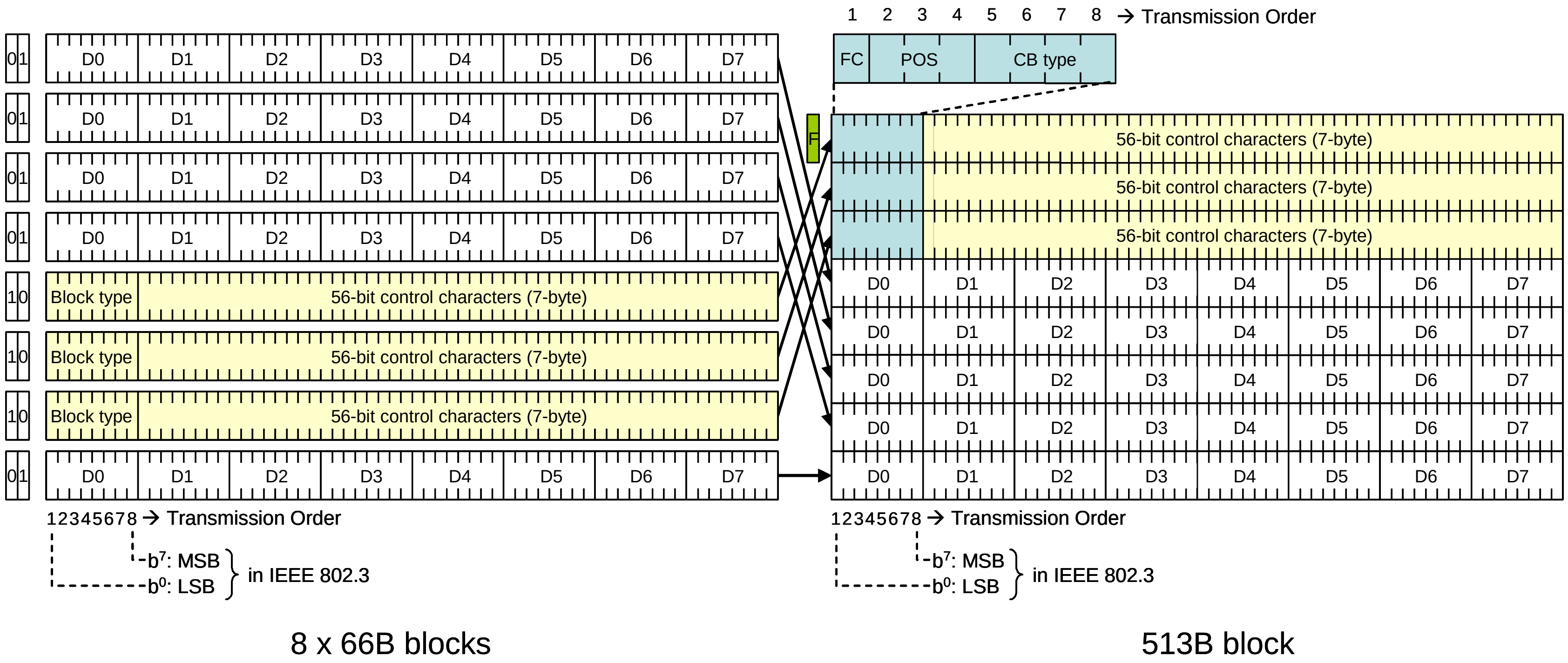


Figure B.3 – 513B block code format

Each of the 66B blocks is encoded into a row of the 8-byte by 8-row structure. Any 66B control blocks (CBi) are placed into the uppermost rows of the structure in the order received, while any all-data 66B blocks (DBi) are placed into the lowermost rows of the structure in the order received.

The flag bit "F" is 1 if the 513B structure contains at least one 66B control block, and 0 if the 513B structure contains eight all-data 66B blocks. Because the 66B control blocks are placed into the uppermost rows of the 513B block, if the flag bit "F" is 1, then the first row will contain a mapping of a 66B control block.

A 66B control block is encoded into a row of the structure shown in Figure B.3 as follows: the sync header of "10" is removed. The byte representing the block type field (see Figure B.2) is replaced by the structure shown in Figure B.4:

Figure B.4 – 513B block's control block header

The byte indicating the control block type (one of 15 legal values) is translated into a 4-bit code according to the rightmost column of Figure B.2. The 3-bit POS field is used to encode the position in which this control block was received in the sequence of eight 66B blocks. The flag continuation bit "FC" will be set to a 0 if this is the final 66B control block or PCS lane alignment marker encoded in this 513B block, or to a 1 if one or more 66B control blocks or PCS lane alignment markers follow this one. At the decoder, the flag bit for the 513B block as a whole, plus the flag continuation bits in each row containing the mapping of a 66B control block or PCS lane alignment marker will allow identification of those rows, which can then be restored to their original position amongst any all-data 66B blocks at the egress according to the POS field. The remaining 7 bytes of the row are filled with the last 7 bytes of the 66B control block.

An all-data 66B block is encoded into a row of the 513B block by dropping the sync header and copying the remaining eight bytes into the row. If all eight rows of the 513B block are placements of 66B all-data blocks, the flag bit "F" will be 0. If fewer than eight rows of the 513B block are placements of 66B all-data blocks, they will appear at the end, and the row containing the placement of the final 66B control block will have a flag continuation bit "FC" value of 0.

The decoder operates in the reverse of the encoder to reconstruct the original sequence of 66B blocks. If flag bit "F" is 1, then 66B control blocks starting from the first row of the block are reconstructed and placed in the position indicated by the POS field. This process continues through all of the control blocks working downward from the top row. The final 66B control block placed within the 513B block will be identified when the flag continuation bit "FC" is zero.

The structure of the 512B/513B code block is shown in Figure B.5. For example, if there is a single 64B/66B control block CB1 in a 512B/513B code block and it was originally located between 64B/66B data blocks DB2 and DB3, the first octet of the 64B character will contain 0.010.1101.CB1; the leading bit in the control octet of 0 indicates the flag continuation "FC" that this 64B control block is the last one in the 512B/513B code block, the value of 010 indicates CB1's position "POS" between DB2 and DB3, and the value of 1101 is a four-bit representation of the control code's block type "CB TYPE" (of which the eight-bit original block type is 0x55).

Input client characters	Flag bit	512-bit (64-Octet) field							
All data block	0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
7 data block 1 control block	1	0 AAA aaaa CB1	DB1	DB2	DB3	DB4	DB5	DB6	DB7
6 data block 2 control block	1	1 AAA aaaa CB1	0 BBB bbbb CB2	DB1	DB2	DB3	DB4	DB5	DB6
5 data block 3 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	0 CCC cccc CB3	DB1	DB2	DB3	DB4	DB5
4 data block 4 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	0 DDD dddd CB4	DB1	DB2	DB3	DB4
3 data block 5 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	0 EEE eeee CB5	DB1	DB2	DB3
2 data block 6 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	0 FFF ffff CB6	DB1	DB2
1 data block 7 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	0 GGG gggg CB7	DB1
8 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	1 GGG gggg CB7	0 HHH hhhh CB8

- Leading bit in a 66B control block FC = 1 if there are more than 66Bcontrol block and = 0 if this payload contains the last control block in that 513B block
- AAA = 3-bit representation of the first control code's original position (First control code loca tor: POS)
- BBB = 3-bit representation of the second control code's original position (Second control code lo cator: POS)
.....
- HHH = 3-bit representation of the eighth control code's original position (Eighth control code lo cator: POS)
- aaa = 4-bit representation of the first control code's type (first control block type: CB TYPE)
- bbb = 4-bit representation of the second control code's type (Second control block type: CB TYPE)
.....
-hhh = 4-bit representation of the eighth control code's type (Eighth control block type: CB TYPE)
- CBi = 56-bit representation of the i-th control code caracters
- DBi = 64-bit representation of the i-th data value in order of transmission

G.709-Y.1331(12)_FB.5

Figure B.5 – 513B code block components

B.3.1 Errors detected before 512B/513B encoder

A set of errors might be detected at the 64B/66B PCS receive process which, in addition to appropriate alarming, needs to send the appropriate signal downstream.

Errors encountered before the encoder, such as loss of client signal, will result in the insertion of an Ethernet LF sequence ordered set prior to this process, which will be transcoded as any other control block. The same action should be taken as a result of failure to achieve 66B block lock on an input signal.

An invalid 66B block will be converted to an error control block before transcoding and the OTN BIP-8 calculation as described in clause E.4.1. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10" and a control block type field which does not appear in Figure B.2. An error control block has sync bits of "10", a block type code of 0x1E, and 8 seven-bit/E/error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

B.3.2 Errors detected by 512B/513B decoder

Several mechanisms will be employed to reduce the probability that the decoder constructs erroneous 64B/66B encoded data at the egress if bit errors have corrupted. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Mechanisms for improving the robustness and for 513B block lock are discussed in Annex F.

B.4 Link fault signalling

In-band link fault signalling in the client 64B/66B code (e.g., if a local fault or remote fault sequence ordered set is being transmitted between Ethernet equipments) is carried transparently according to this transcoding.

Annex C

Adaptation of OTU3 and OTU4 over multichannel parallel interfaces

(This annex forms an integral part of this Recommendation.)

NOTE 1 – This mechanism is designed to allow the use of the optical modules being developed for IEEE 40GBASE-R and 100GBASE-R signals for short-reach client-side OTU3 and OTU4 interfaces respectively. The corresponding physical layer specifications are being added to [ITU-T G.695] and [ITU-T G.959.1].

OTU3 signals may be carried over parallel interfaces consisting of four lanes.

OTU4 signals may be carried over parallel interfaces consisting of four or ten lanes, which are formed by bit multiplexing of 20 logical lanes.

NOTE 2 – Ten lane IEEE 100GBASE-R interfaces have no corresponding ITU-T physical layer interface specification.

The OTU3 and OTU4 frames are inversely multiplexed over physical/logical lanes on a 16-byte boundary aligned with the OTUk frame as illustrated in Figure C.1. The OTUk frame is divided into 1020 groups of 16-bytes.

	1					*4080*
1	1:16 (FAS)	17:32	33:48	49:64	••••	4065:4080
2	4081:4096	4097:5012	5013:5028	5029:5044		9145:9160
3	9161:9176	9177:9192	9193:9208	9209:9224		12225:12240
4	12241:12256	12257:12272	12273:12288	12289:13304		16305:16320

Figure C.1 – OTU3 and OTU4 frames divided on 16-byte boundary

OTU3 16-byte increment distribution

Each 16-byte increment of an OTU3 frame is distributed round robin, to each of the four physical lanes. On each OTU3 frame boundary the lane assignments are rotated.

For OTU3, the lane rotation and assignment is determined by the two LSBs of the MFAS as described in Table C.1 and Figure C.2, which indicates the starting group of bytes of the OTU3 frame that are sent on each lane.

NOTE 3 – MFAS is scrambled as defined in clause 11.2.

The pattern repeats every 64 bytes until the end of the OTU3 frame. The following OTU3 frame will use different lane assignments according to the MFAS.

Table C.1 – Lane rotation assignments for OTU3

MFAS 7-8	Lane 0	Lane 1	Lane 2	Lane 3
*00	1:16	17:32	33:48	49:64
*01	49:64	1:16	17:32	33:48
*10	33:48	49:64	1:16	17:32
*11	17:32	33:48	49:64	1:16

The distribution of 16-byte blocks from the sequence of OTU3 frames is illustrated in Figure C.2:

The parallel lanes can be reassembled at the sink by first recovering framing on each of the parallel lanes, then recovering the lane identifiers and then performing lane de-skewing. Frame alignment, lane identifier recovery and multi-lane alignment should operate under 10^{-3} bit error rate conditions before error correction. Refer to [ITU-T G.798] for the specific processing details.

The lane rotation mechanism will place the first 16 bytes of the OTU3 frame on each lane once per 4080×4 (i.e., 16320) bytes (the same as an OTU3 itself). The two LSBs of the MFAS will be the same in each FAS on a particular lane, which allows the lane to be identified. Since the MFAS cycles through 256 distinct values, the lanes can be de-skewed and reassembled by the receiver as long as the total skew does not exceed 127 OTU3 frame periods (approximately 385 μs). The receiver must use the MFAS to identify each received lane, as lane positions may not be preserved by the optical modules to be used for this application.

OTU4 16-byte increment distribution

Each 16-byte increment of an OTU4 frame is distributed, round robin, to each of the 20 logical lanes. On each OTU4 frame boundary the lane assignments are rotated.

For distribution of OTU4 to twenty logical lanes, since the MFAS is not a multiple of 20, a different marking mechanism must be used. Since the frame alignment signal is 6 bytes (48 bits) and as per [ITU-T G.798] only 32 bits must be checked for frame alignment, the third OA2 byte position will be borrowed as a logical lane marker (LLM). For maximum skew detection range, the lane marker value will increment on successive frames from 0-239 (240 values being the largest multiple of 20 that can be represented in 8-bits). LLM = 0 position shall be aligned with MFAS = 0 position every 3840 (the least common multiple of 240 and 256) frame periods. The logical lane number can be recovered from this value by a modulo 20 operation. Table C.2 and Figure C.3 illustrate how bytes of the OTU4 are distributed in 16-byte increments across the 20 logical lanes.

The pattern repeats every 320 bytes until the end of the OTU4 frame.

The following OTU4 frame will use different lane assignment according to the LLM MOD 20.

Table C.2 – Lane rotation assignments for OTU4

LLM MOD 20	Lane 0	Lane 1	………………	Lane 18	Lane 19
0	1:16	17:32		289:304	305:320
1	305:320	1:16		273:288	289:304
:					
18	33:48	49:64		1:16	17:32
19	17:32	33:48		305:320	1:16

The distribution of 16-byte blocks from the sequence of OTU4 frames is illustrated in Figure C.3:

The parallel lanes can be reassembled at the sink by first recovering framing on each of the parallel lanes, then recovering the lane identifiers and then performing de-skewing of the lanes. Frame alignment, lane identifier recovery and multi-lane alignment should operate under 10^{-3} bit error rate conditions before error correction. Refer to [ITU-T G.798] for specific processing details.

The lane rotation mechanism will place the first 16 bytes of the OTU4 frame on each lane once per 4080×4 (i.e., 16320) bytes (the same as an OTU4 itself). The "LLM MOD 20" will be the same in each FAS on a particular lane, which allows the lane to be identified. Since the LLM cycles through 240 distinct values, the lanes can be de-skewed and reassembled by the receiver as long as the total skew does not exceed 119 OTU4 frame periods (approximately 139 µs). The receiver must use the "LLM MOD 20" to identify each received lane, as lane positions may not be preserved by the optical modules to be used for this application.

The lanes are identified, de-skewed, and reassembled into the original OTU4 frame according to the lane marker. The MFAS can be combined with the lane marker to provide additional skew detection range, the maximum being up to the least common multiple "LCM(240, 256)/2 – 1" or 1919 OTU4 frame periods (approximately 2.241 ms). In mapping from lanes back to the OTU4 frame, the sixth byte of each OTU4 frame which was borrowed for lane marking is restored to the value OA2.

Each physical lane of an OTM-0.4v4 interface is formed by simple bit multiplexing of five logical lanes. At the sink, the bits are disinterleaved into five logical lanes from each physical lane. The sink will identify each logical lane according to the lane marker in the LLM byte. The sink must be able to accept the logical lanes in any position as the ordering of bit multiplexing on each physical lane is arbitrary; the optical module hardware to be used for this application is permitted full flexibility concerning which physical lane will be used for output of each logical lane, and the order of bit multiplexing of logical lanes on each physical output lane.

NOTE 4 – Ten-lane IEEE 100GBASE-R interfaces are specified, although not with ITU-T physical layer specifications. These interfaces may be compatible with a 10-lane interface for OTU4, each lane consisting of two bit-multiplexed logical lanes. Refer to Appendix VII.

This mechanism handles any normally framed OTU3 or OTU4 sequence.

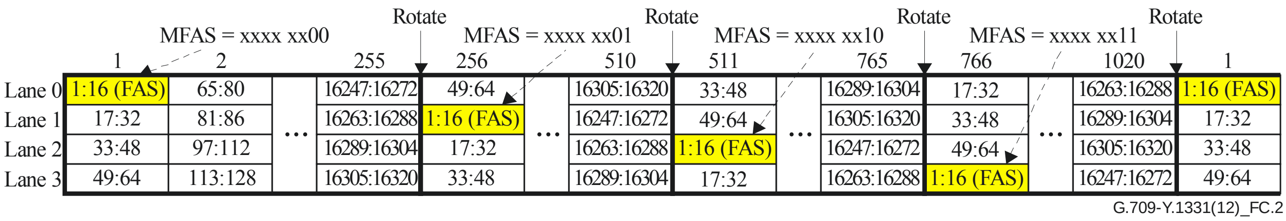


Figure C.2 – Distribution of bytes from OTU3 to parallel lanes

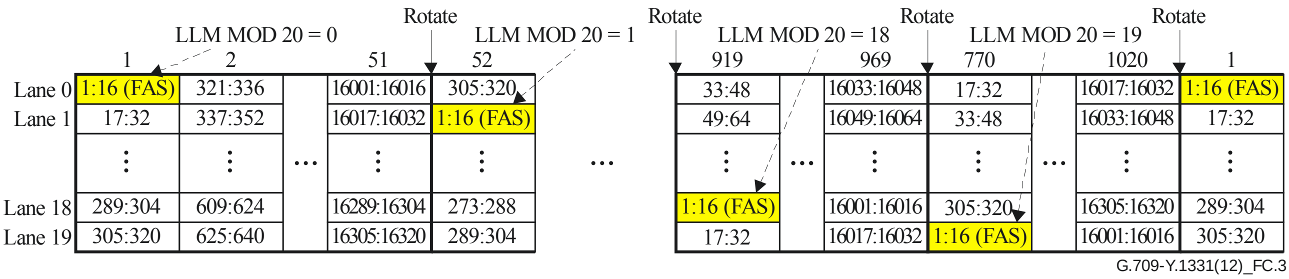


Figure C.3 – Distribution of bytes from OTU4 to parallel lanes

Annex D

Generic mapping procedure principles

(This annex forms an integral part of this Recommendation.)

D.1 Basic principle

For any given CBR client signal, the number of n-bit (e.g., n = 1/8, 1, 8) data entities that arrive during one server frame or server multiframe period is defined by:

$$C_n = \left(\frac{f_{client}}{n} \times T_{server} \right) \quad \text{(D-1)}$$

f_{client}: client bit rate

T_{server}: frame period of the server frame or server multiframe

c_n: number of client n-bit data entities per server frame or server multiframe

As only an integer number of n-bit data entities can be transported per server frame or multiframe, the integer value $C_n(t)$ of c_n has to be used. Since it is required that no client information is lost, the rounding process to the integer value has to take care of the truncated part, e.g., a c_n with a value of 10.25 has to be represented by the integer sequence 10,10,10,11.

$$C_n(t) = \text{int}\left(\frac{f_{client}}{n} \times T_{server} \right) \quad \text{(D-2)}$$

$C_n(t)$: number of client n-bit data entities per server frame t or server multiframe t (integer)

For the case c_n is not an integer, $C_n(t)$ will vary between:

$$C_n(t) = floor\left(\frac{f_{client}}{n} \times T_{server} \right) \quad \text{(D-3)}$$

and

$$C_n(t) = ceiling\left(\frac{f_{client}}{n} \times T_{server} \right) = 1 + floor\left(\frac{f_{client}}{n} \times T_{server} \right) \quad \text{(D-4)}$$

The server frame or multiframe rate is defined by the server bit rate and the number of bits per server frame or multiframe:

$$T_{server} = \frac{B_{server}}{f_{server}} \quad \text{(D-5)}$$

f_{server}: server bit rate

B_{server}: bits per server frame or multiframe

Combining (D-5) with (D-1) and (D-2) results in:

$$c_n = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{n} \right) \quad \text{(D-6)}$$

and

$$C_n(t) = \text{int}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{n}\right) \quad \text{(D-7)}$$

As the client data has to fit into the payload area of the server signal, the maximum value of C_n and as such the maximum client bit rate is limited by the size of the server payload area.

$$C_n(t) \le P_{server} \quad \text{(D-8)}$$

$$f_{client} \le f_{server} \times \frac{P_{server}}{B_{server}} \times n \quad \text{(D-9)}$$

P_{server}: maximum number of (n bits) data entities in the server payload area

The client and server bit rate are independent. This allows specifying the server bit rate independently from the client bit rates. Furthermore, client clock impairments are not seen at the server clock.

If the client or server bit rate changes due to client or server frequency tolerances, c_n and $C_n(t)$ change accordingly. A special procedure has to take care that $C_n(t)$ is changed fast enough to the correct value during start-up or during a step in the client bit rate (e.g., when the client signal is replaced by its AIS signal or the AIS signal is replaced by the client signal). This procedure may be designed to prevent buffer over-/underflow, or an additional buffer over-/underflow prevention method has to be deployed.

A transparent mapping has to determine $C_n(t)$ on a server (multi)frame per (multi)frame base.

In order to extract the correct number of client information entities at the de-mapper, $C_n(t)$ has to be transported in the overhead area of the server frame or multiframe from the mapper to the de-mapper.

Figure D.1 shows the generic functionality of the mapper and de-mapper circuit.

At the mapper, $C_n(t)$ is determined based on the client and server clocks. The client data is constantly written into the buffer memory. The read out is controlled by the value of $C_n(t)$.

At the de-mapper, $C_n(t)$ is extracted from the overhead. $C_n(t)$ controls the write enable signal for the buffer. The client clock is generated based on the server clock and the value of $C_n(t)$.

$C_n(t)$ has to be determined first, then it has to be inserted into the overhead and afterwards $C_n(t)$ client data entities have to be inserted into the payload area of the server as shown in Figure D.2.

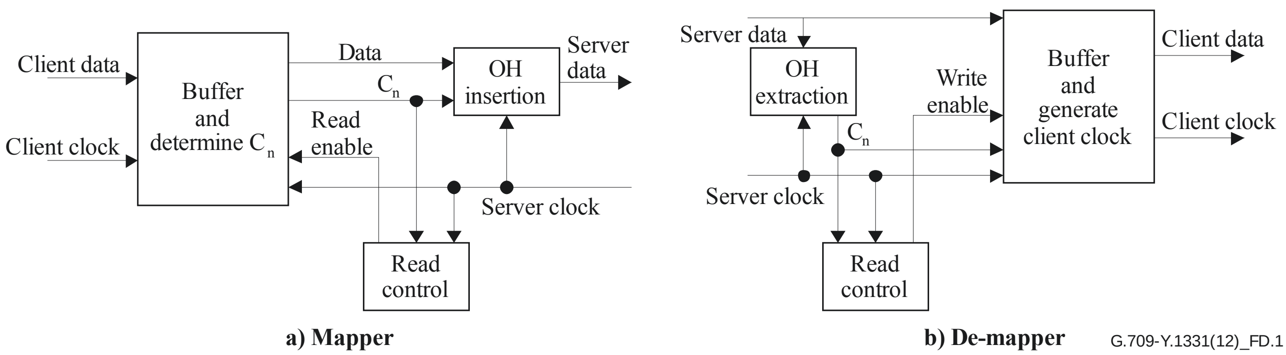


Figure D.1 – Generic functionality of a mapper/de-mapper circuit

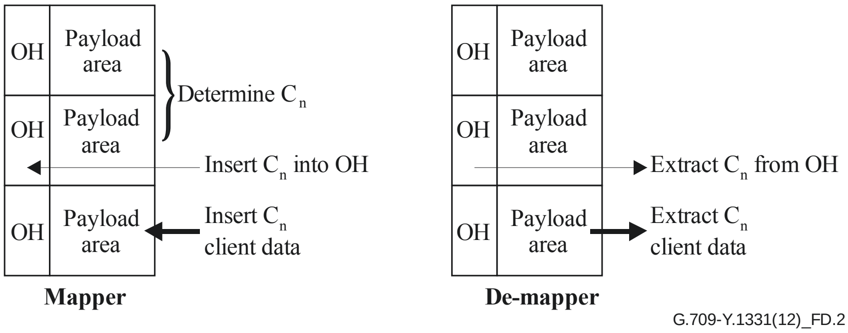


Figure D.2 – Processing flow

$C_n(t)$ client data entities are mapped into the payload area of the server frame or multiframe using a sigma-delta data/stuff mapping distribution. It provides a distributed mapping as shown in Figure D.3. Payload field j (j = 1 .. P_{server}) carries:

– client data (D) if $(j \times C_n(t)) \bmod P_{server} < C_n(t)$ (D-10)

– stuff (S) if $(j \times C_n(t)) \bmod P_{server} \geq C_n(t)$. (D-11)

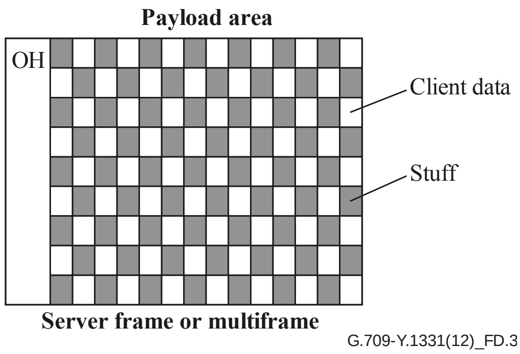


Figure D.3 – Sigma-delta based mapping

$C_n(t)$ client data entities have to be distributed over P_{server} locations. A client data entity has therefore to be inserted with a spacing of $\frac{P_{server}}{C_n(t)}$. This is normally not an integer value, however it can be emulated by an integer calculation using the sigma-delta method based on an overflow accumulator as shown in Figure D.4.

The accumulator memory is reset to 0 at every frame start of the server frame. At every location of the payload area, $C_n(t)$ is added to the memory and the result is compared with P_{server}. If the result is lower than P_{server}, it is stored back into the memory and no client data is indicated for this payload position. If it is equal or greater than P_{server}, P_{server} is subtracted from the result and the new result is stored back in the memory. In addition, client data is indicated for the client position.

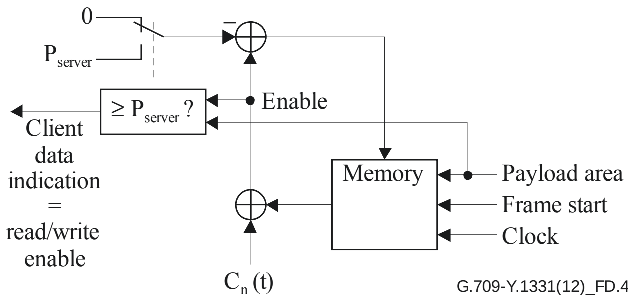


Figure D.4 – Sigma-delta accumulator

As the same start value and $C_n(t)$ are used at the mapper and de-mapper the same results are obtained and interworking is achieved.

D.2 Applying GMP in OTN

Clauses 17.7 and 19.6 specify GMP as the asynchronous generic mapping method for the mapping of CBR client signals into LO OPUk and the mapping of LO ODUj signals into an HO OPUk (via the ODTUk.ts).

NOTE – GMP complements the traditional asynchronous client/server specific mapping method specified in clauses 17.6 and 19.5. GMP is intended to provide the justification of new CBR type client signals into OPUk.

Asynchronous mappings in the OTN have a default 8-bit timing granularity. Such 8-bit timing granularity is supported in GMP by means of a c_n with n=8 (c_8). The jitter/wander requirements for some of the OTN client signals are such that for those signals an 8-bit timing granularity may not be sufficient. For such a case, a 1-bit timing granularity is supported in GMP by means of c_n with n=1 (c_1).

Clauses 17.7 and 19.6 specify that the mapping of CBR client bits into the payload of an LO OPUk and the mapping of an LO ODUj bits into the payload of an ODTUk.ts is performed with 8 × M-bit (M-byte) granularity.

The insertion of CBR client data into the payload area of the OPUk frame and the insertion of LO ODUj data into the payload area of the ODTUk.ts multiframe at the mapper is performed in M-byte (or m-bit, m = 8 × M) data entities, denoted as $C_m(t)$. The remaining $C_{nD}(t)$ data entities are signalled in the justification overhead as additional timing/phase information.

$$C_m = \left(\frac{n \times c_n}{m}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{m}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{8 \times M}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \quad \text{(D-12)}$$

As only an integer number of m-bit data entities can be transported per server frame or multiframe, the integer value $C_m(t)$ of c_m has to be used. Since it is required that no information is lost, the rounding process to the integer value has to take care of the truncated part, e.g., a c_m with a value of 10.25 has to be represented by the integer sequence 10,10,10,11.

$$C_m(t) = \text{int}(c_m) = \text{int}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \quad \text{(D-13)}$$

For the case c_m is not an integer, $C_m(t)$ will vary between:

$$C_m(t) = floor\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \text{and}\, C_m(t) = ceiling\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \tag{D-14}$$

The remainder of c_n and $C_m(t)$ is:

$$c_{nD} = c_n - \left(\frac{8 \times M}{n} \times C_m(t)\right) \tag{D-15}$$

As only an integer number of c_{nD} n-bit data entities can be signalled per server frame or multiframe, the integer value $C_{nD}(t)$ of c_{nD} has to be used.

$$C_{nD}(t) = \text{int}(c_n) - \left(\frac{8 \times M}{n} \times C_m(t)\right) = C_n(t) - \left(\frac{8 \times M}{n} \times C_m(t)\right) \tag{D-16}$$

$C_{nD}(t)$ is a number between $1 - \frac{8 \times M}{n}$ and $\frac{8 \times M}{n} - 1$.

As the client data has to fit into the payload area of the server signal, the maximum value of C_m and as such the maximum client bit rate is limited by the size of the server payload area.

$$C_m(t) \le P_{m,server} \tag{D-17}$$

$P_{m,server}$: maximum number of (m bits) data entities in the server payload area

In order to extract the correct number of client information entities at the de-mapper, $C_m(t)$ has to be transported in the overhead area of the server frame or multiframe from the mapper to the de-mapper.

At the mapper, $C_n(t)$ is determined based on the client and server clocks. The client data is constantly written into the buffer memory. The read out is controlled by the value of $C_m(t)$.

At the de-mapper, $C_m(t)$ and $C_{nD}(t)$ are extracted from the overhead and used to compute $C_n(t)$. $C_m(t)$ controls the write enable signal for the buffer. The client clock is generated based on the server clock and the value of $C_n(t)$.

$C_n(t)$ has to be determined first, then it has to be inserted into the overhead as $C_m(t)$ and $\Sigma C_{nD}(t)$ and afterwards $C_m(t)$ client data entities have to be inserted into the payload area of the server as shown in Figure D.5.

The $C_n(t)$ value determines the $C_m(t)$ and $C_{nD}(t)$ values; $C_m(t)$ = floor (n/m × $C_n(t)$) and $C_{nD}(t)$ = $C_n(t)$ − (m/n × $C_m(t)$). The values of $C_{nD}(t)$ are accumulated and if $\Sigma C_{nD}(t) \ge m/n$ then m/n is subtracted from $\Sigma C_{nD}(t)$ and $C_m(t)$ is incremented with +1. These latter two values are then encoded in the overhead bytes. This $C_m(t)$ value is applied as input to the sigma-delta process.

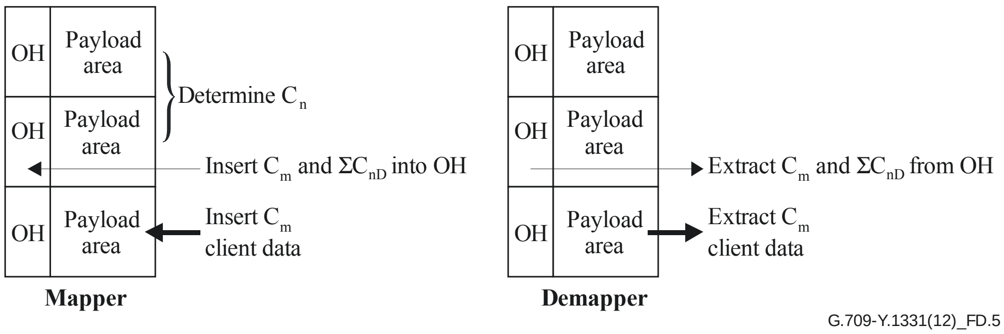


Figure D.5 – Processing flow for GMP in OTN

During start-up or during a step in the client bit rate, the value of $C_n(t)$ will not match the actual number of n-bit client data entities arriving at the mapper buffer and the $C_n(t)$ determination process has to adjust its value to the actual number of n-bit client data entities arriving. This adjustment method is implementation specific. During the mismatch period, the mapper buffer fill level may increase if more n-bit client data entities arrive per multiframe than there are transmitted, or decrease if less n-bit client data entities arrive per multiframe than there are transmitted.

To prevent overflow or underflow of the mapper buffer and thus data loss, the fill level of the mapper buffer has to be monitored. For the case where too many m-bit client data entities are in the buffer, it is necessary to insert temporarily more m-bit client data entities in the server (multi)frame(s) than required by $C_n(t)$. For the case too few m-bit client data entities are in the buffer, it is necessary to insert temporarily fewer m-bit client data entities in the server (multi)frame(s) than required by $C_n(t)$. This behaviour is similar to the behaviour of AMP under these conditions.

The OTN supports a number of client signal types for which transfer delay (latency) and transfer delay variation are critical parameters. Those client signal types require that the transfer delay introduced by the mapper plus de-mapper buffers is minimized and that the delay variation introduced by the mapper plus de-mapper buffers is minimized.

In steady state periods, $C_n(t)$ is a value in the range $C_{n,min}$ to $C_{n,max}$. A value outside this range indicates that there is a misalignment of the expected client bit rate and the actual client bit rate. During transient periods after e.g., a frequency step, $C_n(t)$ may be temporarily outside the range $C_{n,min}$ to $C_{n,max}$.

$C_m(t)$ client data entities are mapped into the payload area of the server frame or multiframe using a sigma-delta data/stuff mapping distribution. It provides a distributed mapping as shown in Figure D.3. Payload field j (j = 1 .. $P_{m,server}$) carries

– client data (D) if $(j \times C_m(t)) \bmod P_{m,server} < C_m(t)$; (D-18)

– stuff (S) if $(j \times C_m(t)) \bmod P_{m,server} \geq C_m(t)$. (D-19)

Values of n, m, M, f_{client}, f_{server}, T_{server}, B_{server}, and $P_{m,server}$ for LO OPU and ODTUk.ts

The values for n, m, M, f_{client}, f_{server}, T_{server}, B_{server}, and $P_{m,server}$ are specified in Table D.1.

Table D.1 – LO OPUk and ODTUk.ts GMP parameter values

GMP parameter	CBR client into LO OPUk	LO ODUj into HO OPUk (ODTUk.ts)
n	8 (default) 1 (client specific)	8
$m = 8 \times M$	OPU0: $8 \times 1 = 8$ OPU1: $8 \times 2 = 16$ OPU2: $8 \times 8 = 64$ OPU3: $8 \times 32 = 256$ OPU4: $8 \times 80 = 640$	ODTU2.ts: $8 \times$ ts ODTU3.ts: $8 \times$ ts ODTU4.ts: $8 \times$ ts
f_{client}	CBR client bit rate and tolerance	LO ODUj bit rate and tolerance (Table 7-2)
f_{server}	OPUk Payload bit rate and tolerance (Table 7-3)	ODTUk.ts Payload bit rate and tolerance (Table 7-7)
T_{server}	ODUk/OPUk frame period (Table 7-4)	OPUk multiframe period (Table 7-6)
B_{server}	OPU0: 8×15232 OPU1: 8×15232 OPU2: 8×15232 OPU3: 8×15232 OPU4: 8×15200	ODTU2.ts: $8 \times$ ts $\times 15232$ ODTU3.ts: $8 \times$ ts $\times 15232$ ODTU4.ts: $8 \times$ ts $\times 15200$
$P_{m,server}$	OPU0: 15232 OPU1: 7616 OPU2: 1904 OPU3: 476 OPU4: 190	ODTU2.ts: 15232 ODTU3.ts: 15232 ODTU4.ts: 15200
ΣC_{8D} range	OPU0: N/A OPU1: 0 to +1 OPU2: 0 to +7 OPU3: 0 to +31 OPU4: 0 to +79	ODTUk.1: N/A ODTUk.2: 0 to +1 ODTUk.3: 0 to +2 ODTUk.4: 0 to +3 : ODTUk.8: 0 to +7 : ODTUk.32: 0 to +31 : ODTUk.79: 0 to +78 ODTUk.80: 0 to +79
ΣC_{1D} range (for selected clients)	OPU0: 0 to +7 OPU1: 0 to +15 OPU2: 0 to +63 OPU3: 0 to +255 OPU4: 0 to +639	Not applicable

D.3 $C_m(t)$ encoding and decoding

$C_m(t)$ is encoded in the ODTUk.ts justification control bytes JC1, JC2 and JC3 specified in clause 19.4.

$C_m(t)$ is a binary count of the number of groups of m LO OPU payload bits that carry m client bits; it has values between Floor($C_{m,min}$) and Ceiling($C_{m,max}$), which are client specific. The Ci (i=1..14) bits that comprise $C_m(t)$ are used to indicate whether the $C_m(t)$ value is incremented or decremented from the value in the previous frame, that is indicated by $C_m(t-1)$. Table D.2 shows the inversion patterns for the Ci bits of $C_m(t-1)$ that are inverted to indicate an increment or decrement of the $C_m(t)$ value. An "I" entry in the table indicates an inversion of that bit.

The bit inversion patterns apply to the $C_m(t-1)$ value, prior to the increment or decrement operation that is signalled by the inversion pattern when $|C_m(t) – C_m(t-1)| \leq 2$ (except $C_m(t) – C_m(t-1) = 0$). The incremented or decremented $C_m(t)$ value becomes the base value for the next GMP overhead transmission.

– When $0 < C_m(t) – C_m(t-1) \leq 2$, indicating an increment of +1 or +2, a subset of the Ci bits containing $C_m(t-1)$ is inverted as specified in Table D.2 and the increment indicator (II) bit is set to 1.

– When $0 > C_m(t) – C_m(t-1) \geq -2$, indicating a decrement of -1 or -2, a subset of Ci bits containing $C_m(t-1)$ is inverted as specified in Table D.2 and the decrement indicator (DI) bit is set to 1.

– When the value of $C_m(t)$ is changed with a value larger than +2 or –2 from the value of $C_m(t-1)$, both the II and DI bits are set to 1 and the Ci bits contain the new $C_m(t)$ value. The CRC-8 verifies whether the $C_m(t)$ value has been received correctly.

– When the value of $C_m(t)$ is unchanged from the value of $C_m(t-1)$, both the II and DI bits are set to 0.

The above encoding process is illustrated in Figure D.6.

Table D.2 – $C_m(t)$ increment and decrement indicator patterns

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Change
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	I	U	I	U	1	0	+1
U	I	U	I	U	I	U	I	U	I	U	I	U	I	0	1	–1
U	I	I	U	U	I	I	U	U	I	I	U	U	I	1	0	+2
I	U	U	I	I	U	U	I	I	U	U	I	I	U	0	1	–2
binary value														1	1	More than +2/–2
NOTE – I indicates inverted Ci bit – U indicates unchanged Ci bit																

The CRC-8 located in JC3 is calculated over the JC1 and JC2 bits. The CRC-8 uses the $g(x) = x^8 + x^3 + x^2 + 1$ generator polynomial, and is calculated as follows:

1) The JC1 and JC2 octets are taken in network octet order, most significant bit first, to form a 16-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 15.

2) $M(x)$ is multiplied by x^8 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 7 or less.

3) The coefficients of $R(x)$ are considered to be an 8-bit sequence, where x^7 is the most significant bit.

4) This 8-bit sequence is the CRC-8 where the first bit of the CRC-8 to be transmitted is the coefficient of x^7 and the last bit transmitted is the coefficient of x^0.

The de-mapper process performs steps 1-3 in the same manner as the mapper process. In the absence of bit errors, the remainder shall be 0000 0000.

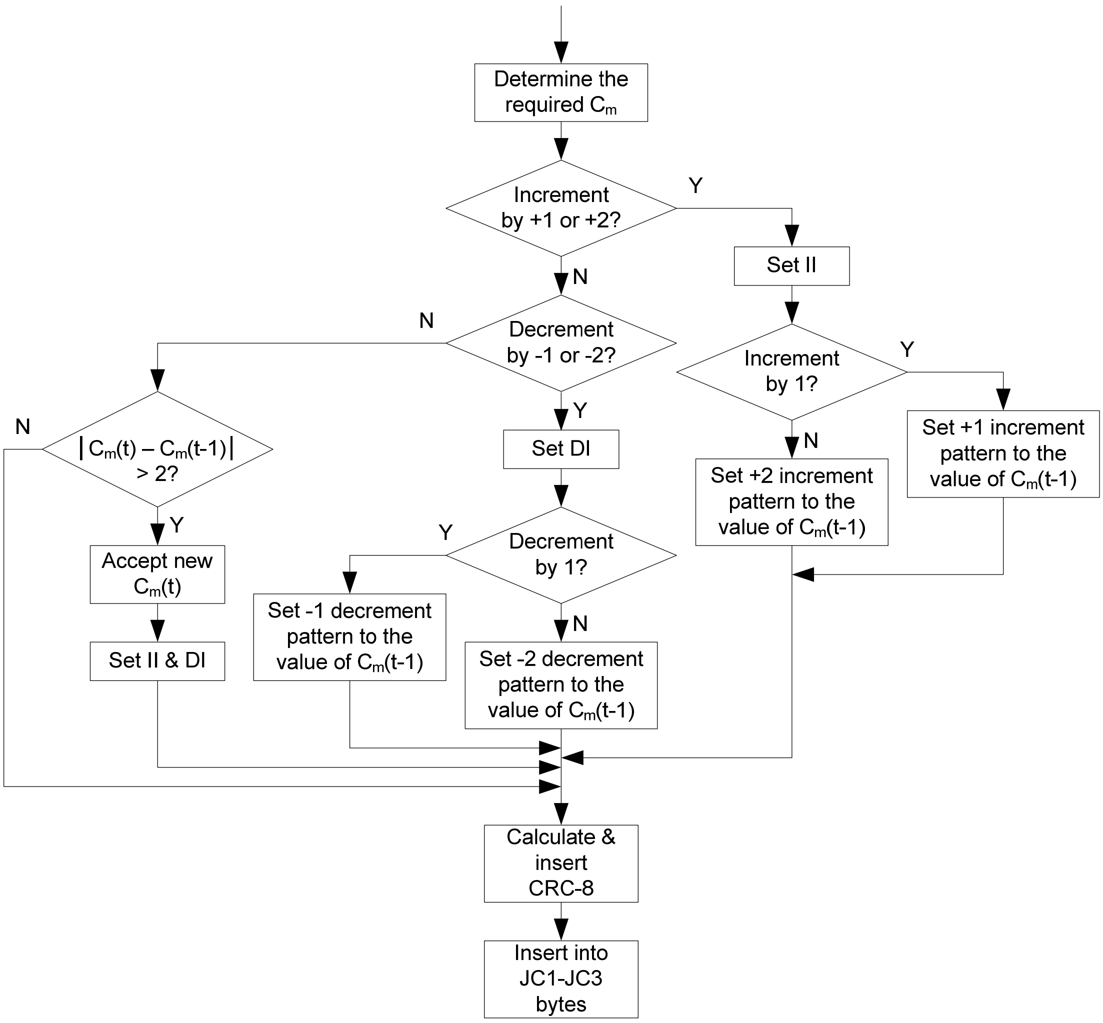
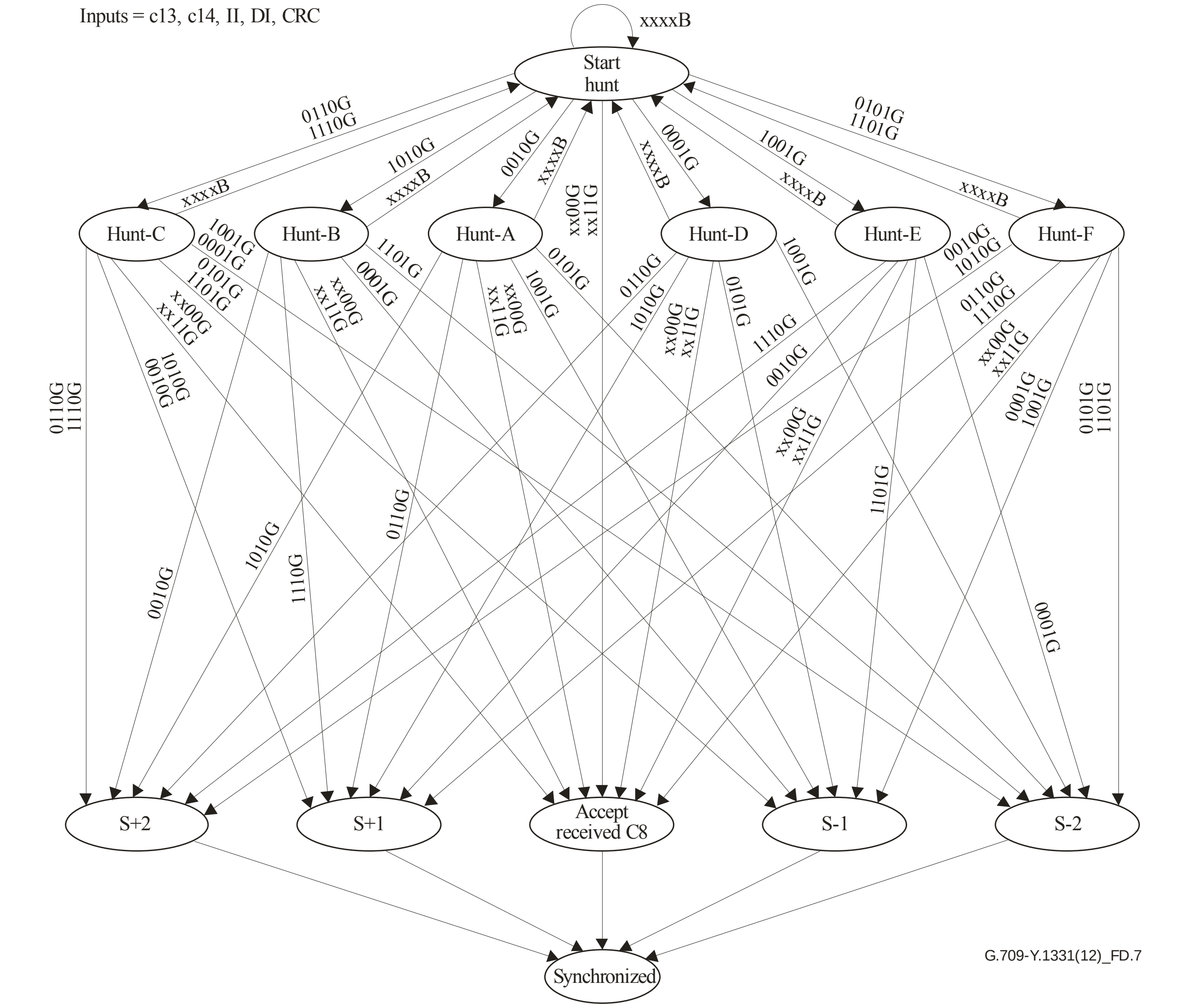


Figure D.6 – JC1, JC2 and JC3 generation

A parallel logic implementation of the source CRC-8 is illustrated in Appendix VI.

The GMP sink synchronizes its $C_m(t)$ value to the GMP source through the following process, which is illustrated in Figure D.7. When the received JC octets contain II = DI and a valid CRC-8, the GMP sink accepts the received C1-C14 as its $C_m(t)$ value for the next frame. At this point the GMP sink is synchronized to the GMP source. When II ≠ DI with a valid CRC-8 in the current received frame (frame *i*), the GMP sink must examine the received JC octets in the next frame (frame *i*+1) in order to obtain $C_m(t)$ synchronization. II ≠ DI in frame *i* indicates that the source is performing a count increment or decrement operation that will modify the $C_m(t)$ value it sends in frame *i*+1. Since this modification to the $C_m(t)$ will affect C13, C14, or both, the GMP sink uses C13, C14, II, and DI in frame *i* to determine its synchronization hunt state (Hunt – A-F in Figure D.7) when it receives frame *i*+1. If II = DI with a valid CRC-8 in frame *i*+1, $C_m(t)$ synchronization is achieved by directly accepting the received C1-C14 as the new $C_m(t)$. If II ≠ DI with a valid CRC-8 in frame *i*+1, the sink uses the new C13, C14, II and DI values to determine whether the source is communicating an increment or decrement operation and the magnitude of the increment/decrement step. This corresponds to the transition to the lower row of states in Figure D.7. At this point, the GMP sink has identified the type of increment or decrement operation that is being signalled in frame *i*+1. The sink then applies the appropriate bit inversion pattern from Table D.2 to the received C1-C14 field to determine the transmitted $C_m(t)$ value. Synchronization has now been achieved since the GMP sink has determined the current $C_m(t)$ and knows the expected $C_m(t)$ change in frame i+2.



S+2 Count = C1–C14 after inverting C2, C3, C6, C7, C10, C11 and C14; Increment +2 for the next frame

S+1 Count = C1–C14 after inverting C1, C3, C5, C7, C9, C11 and C13; Increment +1 for the next frame

S–1 Count = C1–C14 after inverting C2, C4, C6, C8, C10, C12 and C14; Decrement –1 for the next frame

S–2 Count = C1–C14 after inverting C1, C4, C5, C8, C9, C12 and C13; Decrement –2 for the next frame

Figure D.7 – GMP sink count synchronization process diagram

Note that the state machine of Figure D.7 can also be used for off-line synchronization checking.

When the GMP sink has synchronized its $C_m(t)$ value to the GMP source, it interprets the received JC octets according to the following principles:

– When the CRC-8 is good and II = DI, the GMP sink accepts the received $C_m(t)$ value.

– When the CRC-8 is good and II ≠ DI, the GMP sink compares the received $C_m(t)$ value to its expected $C_m(t)$ value to determine the difference between these values. This difference is compared to the bit inversion patterns of Table D.2 to determine the increment or decrement operation sent by the source and updates its $C_m(t)$ accordingly. Since the CRC-8 is good, the sink can use either JC1 or JC2 for this comparison.

– When the CRC-8 is bad, the GMP sink compares the received $C_m(t)$ value to its expected $C_m(t)$ value. The sink then compares the difference between these values, per Table D.2, to the valid bit inversion patterns in JC1, and the bit inversion, II and DI pattern in JC2.

- If JC1 contains a valid pattern and JC2 does not, the sink accepts the corresponding increment or decrement indication from JC1 and updates its $C_m(t)$ accordingly.
- If JC2 contains a valid pattern and JC1 does not, the sink accepts the corresponding increment or decrement indication from JC2 and updates its $C_m(t)$ accordingly.
- If both JC1 and JC2 contain valid patterns indicating the same increment or decrement operation, this indication is accepted and the sink updates its $C_m(t)$ accordingly.
- If neither JC1 nor JC2 contain valid patterns, the sink shall keep its current count value and begin the search for synchronization.

NOTE – If JC1 and JC2 each contain valid patterns that are different from each other, the receiver can either keep the current $C_m(t)$ value and begin a synchronization search, or it can use the CRC-8 to determine whether JC1 or JC2 contains the correct pattern.

The GMP sink uses the updated $C_m(t)$ value to extract the client data from the next LO OPU frame or ODTUk.ts multiframe.

D.4 $\Sigma C_{nD}(t)$ encoding and decoding

The cumulative value of $C_{nD}(t)$ ($\Sigma C_{nD}(t)$) is encoded in bits 4-8 of the LO OPUk and ODTUk.ts justification control bytes JC4, JC5 and JC6. Bits D1 to D10 in JC4 and JC5 carry the value of $\Sigma C_{nD}(t)$. Bit D1 carries the most significant bit and bit D10 carries the least significant bit.

The CRC-5 located in JC6 is calculated over the D1-D10 bits in JC4 and JC5. The CRC-5 uses the $g(x) = x^5 + x + 1$ generator polynomial, and is calculated as follows:

1) The JC4 bits 4-8 and JC5 bits 4-8 octets are taken in network transmission order, most significant bit first, to form a 10-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 9.
2) $M(x)$ is multiplied by x^5 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 4 or less.
3) The coefficients of $R(x)$ are considered to be a 5-bit sequence, where x^4 is the most significant bit.
4) This 5-bit sequence is the CRC-5 where the first bit of the CRC-5 to be transmitted is the coefficient of x^4 and the last bit transmitted is the coefficient of x^0.

The de-mapper process performs steps 1-3 in the same manner as the mapper process. In the absence of bit errors, the remainder shall be 00000.

A parallel logic implementation of the source CRC-5 is illustrated in Appendix VI.

Annex E

Adaptation of parallel 64B/66B encoded clients

(This annex forms an integral part of this Recommendation.)

E.1 Introduction

IEEE 40GBASE-R and 100GBASE-R interfaces specified in [IEEE 802.3ba]-2010 are parallel interfaces intended for short-reach (up to 40 km) interconnection of Ethernet equipment. This annex describes the process of converting the parallel format of these interfaces into a serial bit stream to be carried over the OTN.

The order of transmission of information in all the diagrams in this annex is first from left to right and then from top to bottom.

E.2 Clients signal format

40GBASE-R and 100GBASE-R clients are initially parallel interfaces, but in the future they may be serial interfaces. Independent of whether these interfaces are parallel or serial, or what the parallel interface lane count is, 40GBASE-R signals are comprised of four PCS lanes, and 100GBASE-R signals are comprised of twenty PCS lanes. If the number of physical lanes on the interface is fewer than the number of PCS lanes, the appropriate number of PCS lanes is bit-multiplexed onto each physical lane of the interface. Each PCS lane consists of 64B/66B encoded data with a PCS lane alignment marker inserted on each lane once per 16384 66-bit blocks. The PCS lane alignment marker itself is a special format 66B codeword.

The use of this adaptation for 40GBASE-R into OPU3 also applies the transcoding method that appears in Annex B and the framing method of Annex F. The adaptation described in this annex alone can be used for the adaptation of 100GBASE-R into OPU4.

E.3 Client frame recovery

Client framing recovery consists of the following:

– bit-disinterleave the PCS lanes, if necessary. This is necessary whenever the number of PCS lanes and the number of physical lanes is not equal, and is not necessary when they are equal (e.g., a 4-lane 40GBASE-R interface);
– recover 64B/66B block lock as per the state diagram in Figure 82-10 of [IEEE 802.3ba];
– recover lane alignment marker framing on each PCS lane as per the state diagram in Figure 82-11 of [IEEE 802.3ba];
– reorder and de-skew the PCS lanes into a serialized stream of 66B blocks (including lane alignment markers). Figure E.1 illustrates the ordering of 66B blocks after the completion of this process for an interface with p PCS lanes.

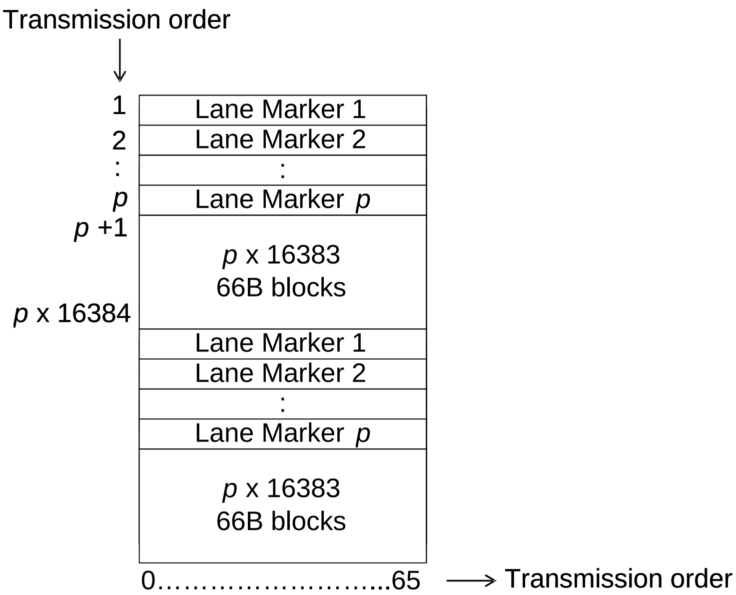


Figure E.1 – De-skewed/serialized stream of 66B blocks

Each 66B codeword is one of the following:

– a set of eight data bytes with a sync header of "01";

– a control block (possibly including seven or fewer data octets) beginning with a sync header of "10";

– a PCS lane alignment marker, also encoded with a sync header of "10". Of the 8 octets following the sync header, 6 octets have fixed values allowing the PCS lane alignment markers to be recognized (see Tables E.1 and E.2). The fourth octet following the sync header is a BIP-8 calculated over the data from one alignment marker to the next as defined in Table 82-4 of [IEEE 802.3ba]. The eighth octet is the complement of this BIP-8 value to maintain DC balance. Note that the intended operation is to pass these BIP-8 values transparently as they are used for monitoring the error ratio of the Ethernet link between Ethernet PCS sublayers. For the case of 100GBASE-R, the BIP-8 values are not manipulated by the mapping or de-mapping procedure. For the case of 40GBASE-R, a BIP-8 compensation is done as described in clause E.4.1.

For all-data blocks and control blocks, the 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane alignment markers) according to the polynomial $G(x) = 1 + x^{39} + x^{58}$.

After 64B/66B block lock recovery as per the state diagram in Figure 82-10 of [IEEE 802.3ba] to the single-lane received aggregate signal, these 66B blocks are re-distributed to PCS lanes at the egress interface. The 66B blocks (including PCS lane alignment markers) resulting from the decoding process are distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

E.3.1 40GBASE-R client frame recovery

PCS lane alignment markers have the values shown in Table E.1 for 40GBASE-R signals which use PCS lane numbers 0-3.

Table E.1 – PCS lane alignment marker format for 40GBASE-R

Lane Number	SH	Encoding {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}
0	10	0x90, 0x76, 0x47, BIP_3, 0x6f, 0x89, 0xb8, BIP_7
1	10	0xf0, 0xc4, 0xe6, BIP_3, 0x0f, 0x3b, 0x19, BIP_7
2	10	0xc5, 0x65, 0x9b, BIP_3, 0x3a, 0x9a, 0x64, BIP_7
3	10	0xa2, 0x79, 0x3d, BIP_3, 0x5d, 0x86, 0xc2, BIP_7

Since a 40GBASE-R client signal must be transcoded into 1024B/1027B for rate reduction, the 64B/66B PCS receive process at the ingress interface further descrambles the bit-stream skipping sync headers and PCS lane alignment markers, and the 64B/66B PCS transmit process at the egress interface scrambles the bit-stream again skipping sync headers and PCS lane alignment markers, as shown in Figure E.1.

E.3.2 100GBASE-R client frame recovery

PCS lane alignment markers have the values shown in Table E.2 for 100GBASE-R signals which use PCS lane numbers 0-19.

The lane alignment markers transported over the OPU4 are distributed unchanged to the PCS lanes.

Table E.2 – PCS lane alignment marker format for 100GBASE-R

Lane Number	SH	Encoding {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}	Lane Number	SH	Encoding {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}
0	10	0xc1, 0x68, 0x21, BIP_3, 0x3e, 0x97, 0xde, BIP_7	10	10	0xfd, 0x6c, 0x99, BIP_3, 0x02, 0x93, 0x66, BIP_7
1	10	0x9d, 0x71, 0x8e, BIP_3, 0x62, 0x8e, 0x71, BIP_7	11	10	0xb9, 0x91, 0x55, BIP_3, 0x46, 0x6e, 0xaa, BIP_7
2	10	0x59, 0x4b, 0xe8, BIP_3, 0xa6, 0xb4, 0x17, BIP_7	12	10	0x5c, 0xb9, 0xb2, BIP_3, 0xa3, 0x46, 0x4d, BIP_7
3	10	0x4d, 0x95, 0x7b, BIP_3, 0xb2, 0x6a, 0x84, BIP_7	13	10	0x1a, 0xf8, 0xbd, BIP_3, 0xe5, 0x07, 0x42, BIP_7
4	10	0xf5, 0x 07, 0x09, BIP_3, 0x0a, 0xf8, 0xf6, BIP_7	14	10	0x83, 0xc7, 0xca, BIP_3, 0x7c, 0x38, 0x35, BIP_7
5	10	0xdd, 0x14, 0xc2, BIP_3, 0x22, 0xeb, 0x3d, BIP_7	15	10	0x35, 0x36, 0xcd, BIP_3, 0xca, 0xc9, 0x32, BIP_7
6	10	0x9a, 0x4a, 0x26, BIP_3, 0x65, 0xb5, 0xd9, BIP_7	16	10	0xc4, 0x31, 0x4c, BIP_3, 0x3b, 0xce, 0xb3, BIP_7
7	10	0x7b, 0x45, 0x66, BIP_3, 0x84, 0xba, 0x99, BIP_7	17	10	0xad, 0xd6, 0xb7, BIP_3, 0x52, 0x29, 0x48, BIP_7

Table E.2 – PCS lane alignment marker format for 100GBASE-R

Lane Number	SH	Encoding {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}	Lane Number	SH	Encoding {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}
8	10	0xa0, 0x24, 0x76, BIP_3, 0x5f, 0xdb, 0x89, BIP_7	18	10	0x5f, 0x66, 0x2a, BIP_3, 0xa0, 0x99, 0xd5, BIP_7
9	10	0x68, 0xc9, 0xfb, BIP_3, 0x97, 0x36, 0x04, BIP_7	19	10	0xc0, 0xf0, 0xe5, BIP_3, 0x3f, 0x0f, 0x1a, BIP_7

E.4 Additions to Annex B transcoding for parallel 64B/66B clients

When OPUk is large enough for the serialized 66B block stream (e.g., for 100GBASE-R client signals into OPU4), the recovered client frames are adapted directly as per this annex.

When used in combination with the transcoding into 513B code blocks described in Annex B (e.g., for 40GBASE-R client signals into OPU3), this clause describes the additions to the Annex B transcoding process for transport of PCS lane alignment markers.

Ethernet path monitoring is the kind of behaviour that is desirable in the case where the Ethernet equipment and the OTN equipment are in different domains (e.g., customer and service provider) and from the standpoint of the Ethernet equipment. It is also the default behaviour which would result from the current mapping of 100GBASE-R where the 66B blocks would be mapped into the OPU4 container after management of skew. It may also be perceived as a transparency requirement that BIP-8 work end-to-end. Additional functionality as described below has to be built in to allow BIP-8 transparency for 40GBASE-R client signals.

PCS lane alignment markers are encoded together with 66B control blocks into the uppermost rows of the 513B code block shown in Figure B.3. The flag bit "F" of the 513B structure is 1 if the 513B structure contains at least one 66B control block or PCS lane alignment marker, and 0 if the 513B structure contains eight all-data 66B blocks.

The transcoding into 512B/513B must encode PCS lane alignment marker into a row of the structure shown in Figure B.3 as follows: The sync header of "10" is removed. The received M_0, M_1 and M_2 bytes of the PCS alignment marker encodings as shown in Table E.1 are used to forward the lane number information. The first byte of the row will contain the structure shown in Figure B.4, with a CB-TYPE field of "0100". The POS field will indicate the position where the PCS lane alignment marker was received among the group of eight 66B codewords being encoded into this 513B block. The flag continuation bit "FC" will indicate whether any other 66B control blocks or PCS lane alignment markers are encoded into rows below this one in the 513B block. Beyond this first byte, the next four bytes of the row are populated with the received M_0, M_1, M_2 and ingress BIP_3 bytes of the PCS alignment marker encodings at the encoder. At the decoder, a PCS lane alignment marker will be generated in the position indicated by the POS field among any 66B all-data blocks contained in this 513B block, the sync header of "10" is generated followed by the received M_0, M_1 and M_2 bytes, the egress BIP_3 byte, the bytes M_4, M_5 and M_6 which are the bit-wise inverted M_0, M_1 and M_2 bytes received at the decoder, and the egress BIP_7 byte which is the bit-wise inverted egress BIP_3 byte.

It will then be up to the Ethernet receiver to handle bit errors within the OTN section that might have altered the PCS alignment marker encodings (for details refer to clause 82.2.18.3 and Figure 82-11 in [IEEE 802.3ba]).

The egress BIP_3 and the egress BIP_7 bytes are calculated as described in clause E.4.1.

Figure E.2 below shows the transcoded lane marker format.

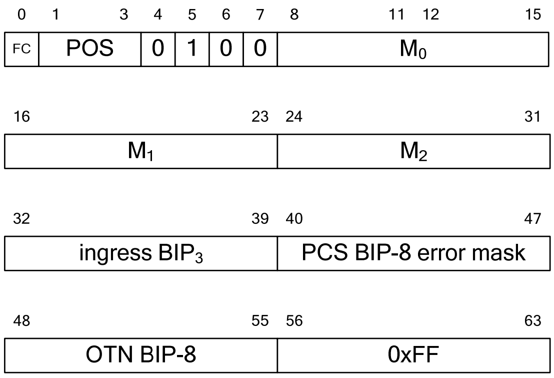


Figure E.2 – Transcoded lane marker format

E.4.1 BIP-8 transparency

The transcoding method used for 40GBASE-R is timing and PCS codeword transparent. In normal operation, the only aspects of the PCS encoded bitstream that are not preserved given the mapping described in annexes B, E and F are for one the scrambling, since the scrambler does not begin with a known state and multiple different encoded bitstreams can represent the same PCS encoded content, and secondly the BIP-8 value in the Ethernet path or more precisely, the bit errors that occur between the Ethernet transmitter and the ingress point of the OTN domain and within the OTN domain. The BIP-8 values can be preserved with the scheme described below. As the scrambling itself does not contain any information that has to be preserved, no effort has been made to synchronize the scrambler states between OTN ingress and OTN egress.

Unfortunately, since the BIP-8 is calculated on the scrambled bitstream, a simple transport of the BIP-8 across the OTN domain in the transcoded lane marker will not result in a BIP-8 value that is meaningful for detecting errors in the received, descrambled, transcoded, trans-decoded, and then rescrambled bit stream.

To preserve the bit errors between the Ethernet transmitter and the egress side of the OTN domain, the bit-error handling is divided into two processes, one that takes place at the OTN ingress side, or encoder, and one on the OTN egress side, or decoder.

At the OTN ingress an 8-bit error mask is calculated by generating the expected BIP-8 for each PCS lane and XORing this value with the received BIP-8. This error mask will have a "1" for each bit of the BIP-8 which is wrong, and a "0" for each bit which is correct. This value is shown as a PCS BIP-8 error mask in Figure E.2.

In the event no errors are introduced across the OTN (as an FEC protected network can be an essentially zero error environment), the PCS BIP-8 error mask can be used to adjust the newly calculated PCS BIP-8 at the egress providing a reliable indication of the number of errors that are introduced across the full Ethernet path. If errors are introduced across the OTN, this particular BIP-8 calculation algorithm will not see these errors.

To overcome this situation, a new BIP-8 per lane for the OTN section is introduced. In the following this new BIP-8 will be identified as OTN BIP-8 in order to distinguish it from the PCS BIP-8.

It should be noted that the term OTN BIP-8 does not refer to and should not be confused with the BIP-8 defined in the OTUk overhead (byte SM[2]).

The OTN BIP-8 is calculated similarly to the PCS BIP-8 as described in clause 82.2.8 of [IEEE 802.3ba] with the exception that the calculation will be done over unscrambled PCS lane data, the original received lane alignment marker, after error control block insertion and before transcoding. Figure E.2 shows the byte location of the OTN BIP-8 in the transcoded lane marker.

The transcoded lane marker is transmitted together with the transcoded data blocks over the OTN section as defined in Annex B. At the OTN egress after trans-decoding and before scrambling, the ingress alignment marker is recreated using M_0, M_1, M_2 and ingress BIP_3 of the transcoded alignment marker followed by the bit-wise inversion of these bytes. This recreated alignment marker together with the trans-decoded and unscrambled data blocks is used to calculate the expected OTN BIP-8 for each PCS lane (refer to clause 82.2.8 of [IEEE 802.3ba]). The expected value will be XORed with the received OTN BIP-8. This error mask will have a "1" for each bit of the OTN BIP-8 which is wrong, and a "0" for each bit which is correct.

The egress BIP_3 for each PCS lane is calculated over the trans-decoded and scrambled data blocks including the trans-decoded alignment marker (refer to clause E.4) following the process depicted in clause 82.2.8 of [IEEE 802.3ba].

The egress BIP_3 is then adjusted for the errors that occurred up to the OTN egress by first XORing with the PCS BIP-8 error mask and then XORing with the OTN BIP-8 error mask. This combined error mask will be used to compute the number of BIP errors when used for non-intrusive monitoring.

The BIP_7 is created by bit-wise inversion of the adjusted BIP_3.

E.4.2 Errors detected by mapper

Errors encountered before the mapper, such as loss of client signal on any physical lane of the interface, will result in the insertion of an Ethernet LF sequence ordered set prior to this process. The same action should be taken as a result of failure to achieve 66B block lock on any PCS lane, failure to achieve lane alignment marker framing on each PCS lane, or failure to de-skew because the skew exceeds the buffer available for de-skew.

An invalid 66B block will be converted to an error control block before transcoding. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10", is not a valid PCS lane alignment marker and has a control block type field which does not appear in Figure B.2 or has one of the values 0x2d, 0x33, 0x66, or 0x55 which are not used for 40GBASE-R or 100GBASE-R. An error control block has sync bits of "10", a block type code of 0x1e, and 8 seven-bit/E/error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

E.4.3 Errors detected by de-mapper

Several mechanisms will be employed to reduce the probability that the de-mapper constructs erroneous parallel 64B/66B encoded data at the egress if bit errors have corrupted. Since detectable corruption normally means that the proper order of 66B blocks to construct at the de-mapper cannot be reliably determined, if any of these checks fail, the de-mapper will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Mechanisms for improving the robustness and for 513B block lock including PCS lane alignment markers are discussed in Annex F.

Annex F

Improved robustness for mapping of 40GBASE-R into OPU3 using 1027B code blocks

(This annex forms an integral part of this Recommendation.)

F.1 Introduction

When a parallel 40GBASE-R signal is transcoded as per annexes B and E and directly mapped into OPU3 without GFP framing, another method is needed to locate the start of 513B blocks and to provide protection to prevent bit errors creating an unacceptable increase in mean time to false packet acceptance (MTTFPA).

F.2 513B code block framing and flag bit protection

The mapping of 513B code blocks into OPU3 requires a mechanism for locating the start of the code blocks. A mechanism is also needed to protect the flag bit, whose corruption could cause data to be erroneously interpreted as control and vice versa.

Both of these requirements can be addressed by providing parity across the flag bits of two 513B blocks produced from the transcoding of Annex B.

Figure F.1 illustrates the flag parity bit across two 513B blocks. This creates a minimum two-bit Hamming distance between valid combinations of flag bits.

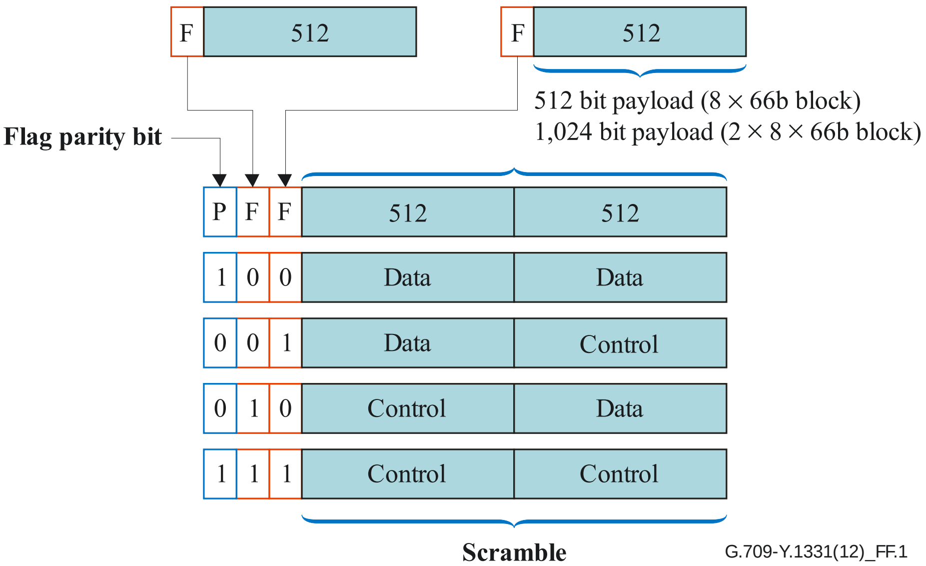


Figure F.1 – Flag parity bit on two 513B blocks (1027B code)

The flag parity bit creates a sequence that can be used for framing to locate the 513B blocks in a stream of bits. The state diagram of Figure 49-12 of [IEEE 802.3] is applied to locate a 3-bit pattern appearing once per 1027 bits (rather than a 2-bit pattern appearing once per 66 bits) where four out of eight 3-bit sequences (rather than two out of four two-bit values as used in IEEE 802.3) match the pattern. The additional step required is to scramble the non-flag bits so that the legal sequences of these bits are not systematically mimicked in the data itself. The scrambler to be used for this purpose is the Ethernet self-synchronous scrambler using the polynomial $G(x) = 1 + x^{39} + x^{58}$.

At the de-mapper, invalid flag bit parity will cause both of the 513B blocks across which the flag bit parity applies to be decoded as 8 × 2 66B error control blocks ("10" sync header, control block type 0x1e, followed by eight 7-bit/E/control characters).

F.3 66B block sequence check

Bit error corruption of the position or flag continuation bits could cause 66B blocks to be de-mapped from 513B code blocks in the incorrect order. Additional checks are performed to prevent that this results in incorrect packet delineation. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Other checks are performed to reduce the probability that invalid data is delivered at the egress in the event that bit errors have corrupted any of the POS fields or flag continuation bits "FC".

If flag bit "F" is 1 (i.e., the 513B block includes at least one 64B/66B control block), for the rows of the table up until the first one with a flag continuation bit of zero (the last one in the block), it is verified that no two 66B control blocks or lane alignment markers within that 513B block have the same value in the POS field, and further, that the POS field values for multiple control or lane alignment rows are in ascending order, which will always be the case for a properly constructed 513B block. If this check fails, the 513B block is decoded into eight 66B error control blocks.

The next check is to ensure that the block sequence corresponds to well-formed packets, which can be done according to the state diagram in Figures F.2 and F.3. This check will determine if 66B blocks are in an order that does not correspond to well-formed packets, e.g., if during an IPG an all-data 66B block is detected without first seeing a control block representing packet start, or if during a packet a control/idle block is detected without first seeing a control block representing packet termination, control blocks have likely been misordered by corruption of either the POS bits or a flag continuation bit. Failure of this check will cause the 513B block to be decoded as eight 66B error control blocks. Note that PCS lane alignment markers are accepted in either state and do not change state as shown in Figure F.3.

The sequence of PCS lane alignment markers is also checked at the decoder. For an interface with p PCS lanes, the PCS lane alignment markers for lanes 0 to p-1 will appear in a sequence, followed by 16383×p non-lane-marker 66B blocks, followed by another group of PCS lane alignment markers. A counter is maintained at the decoder to keep track of when the next group of lane alignment markers is expected. If, in the process of decoding lane alignment markers from a 513B block, a lane alignment marker is found in a position where it is not expected, or a lane alignment marker is missing in a position where it would have been expected, the entire 513B block is decoded as eight 66B error control blocks as shown in Figures F.2, F.3, and F.4.

F.3.1 State diagram conventions

The body of this clause is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

The notation used in the state diagrams follows the conventions of clause 21.5 of [IEEE 802.3]. State diagram timers follow the conventions of clause 14.2.3.2 of [IEEE 802.3]. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

F.3.2 State variables

F.3.2.1 Constants

EBLOCK_T<65:0>

66-bit vector to be sent to the PCS containing /E/ in all the eight character locations.

Mi<65:0>

66-bit vector containing the trans-decoded alignment marker of i-th PCS lane ($0 < i \le p$). (p=4 for 40GBASE-R, and p=20 for 100GBASE-R).

F.3.2.2 Variables

1027B_block_lock

Indicates the state of the block_lock variable when the state diagram of Figure 49-12 of [IEEE 802.3] is applied to locate a 3-bit pattern appearing once per 1027 bits (rather than a 2-bit pattern appearing once per 66 bits) as described in clause F.2. Set true when sixty-four contiguous 1027-bit blocks are received with valid 3-bit patterns, set false when sixteen 1027-bit blocks with invalid 3-bit patterns are received before sixty-four valid blocks.

1027B_high_ber

Indicates a Boolean variable when the state diagram of Figure 49-13 of [IEEE 802.3] is applied to count invalid 3-bit sync headers of 1027-bit blocks (rather than 2-bit sync headers of 66-bit blocks) within the current 250 µs (rather than 125 µs). Set true when the ber_cnt exceeds 8 (rather than 16) indicating a bit error ratio $>10^{-4}$.

Mseq_violation

Indicates a Boolean variable that is set and latched in each rx513_raw<527:0> PCS lane alignment marker cycle based on the PCS lane marker position and order. It is true if the unexpected marker sequence is detected and false if not.

POS_violation

A Boolean variable that is set in each rx513_raw<527:0> based on the POS field values for rx_tcd<65:0>. It is true if the two or more have the same POS values or if they are not in ascending order, and false if their POS values are in ascending order.

reset

A Boolean variable that controls the resetting of the PCS. It is true whenever a reset is necessary including when reset is initiated from the MDIO, during power on, and when the MDIO has put the PCS into low-power mode.

Rx513_coded<512:0>

A vector containing the input to the 512B/513B decoder.

rx513_raw<527:0>

A vector containing eight successive 66-bit vectors (tx_coded).

rx_tcd<65:0>

A 66-bit vector transcode-decoded from a 513-bit block following the rules shown in Figure B.5.

seq_violation

A Boolean variable that is set in each rx513_raw<527:0> based on the sequence check on an rx_tcd<65:0> stream. It is true if the unexpected sequence is detected and false if not.

F.3.2.3 Functions

DECODE(rx513_coded<512:0>)

Decodes the 513-bit vector returning rx513_raw<527 :0> which is sent to the client interface. The DECODE function shall decode the block as specified in Figure F.2.

R_BLOCK_TYPE = {C, S, T, D, E, M}

This function classifies each 66-bit rx_tcd vector as belonging to one of the six types depending on its contents.

Values: C, S, T, and D are defined in clause 49.2.13.2.3 of [IEEE 802.3].

M: the vector contains a sync header of 10 and is recognized as a valid PCS lane alignment marker by using the state machine shown in Figure F.3.

E: the vector does not meet the criteria for any other value.

R_TYPE(rx_tcd<65:0>)

Returns the R_BLOCK_TYPE of the rx_tcd<65:0> bit vector.

R_TYPE_NEXT

Prescient end of packet check function. It returns the R_BLOCK_TYPE of the rx_tcd vector immediately following the current rx_tcd vector.

F.3.2.4 Counters

cnt

Count up to a maximum of *p* of the number of PCS lanes.

F.3.3 State diagrams

The receive state machine for a series of 513-bit blocks shown in Figure F.2 determines whether the 513-bit block contains valid eight 66-bit blocks or not.

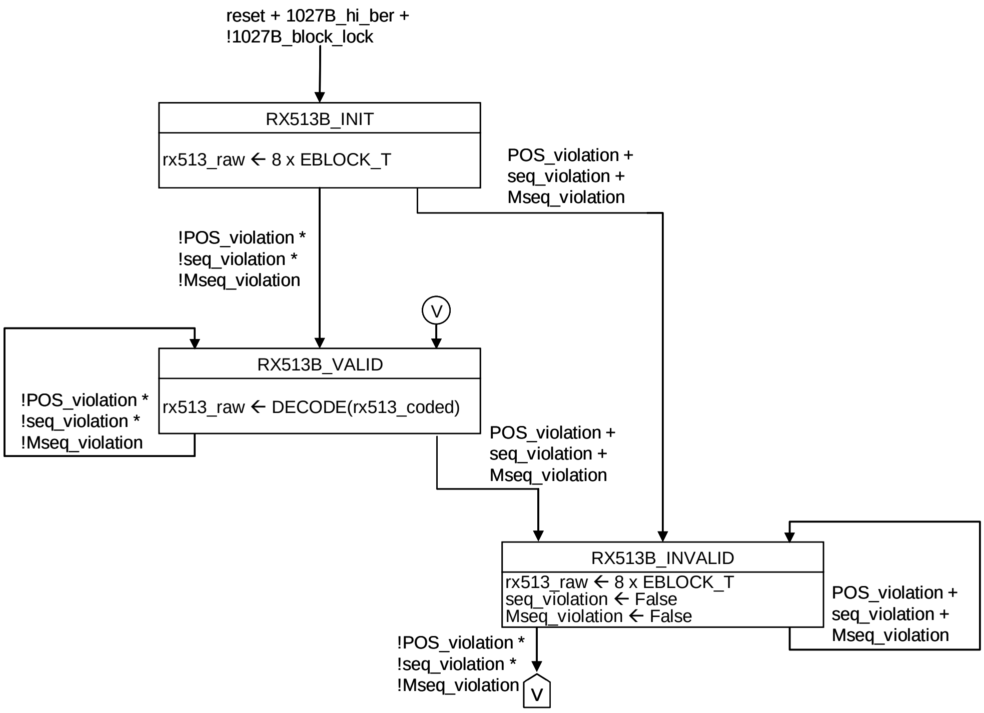


Figure F.2 – Receive state machine for the 512B/513B code blocks including lane alignment markers

The trans-decode state machine for a series of 66-bit blocks shown in Figure F.3 checks the block type sequence of recovered 66-bit blocks.

The PCS lane alignment marker state machine for a series of 66-bit blocks shown in Figure F.4 detects the alignment markers every $p \times 16384$ blocks and checks whether the marker is in ascendant order or not.

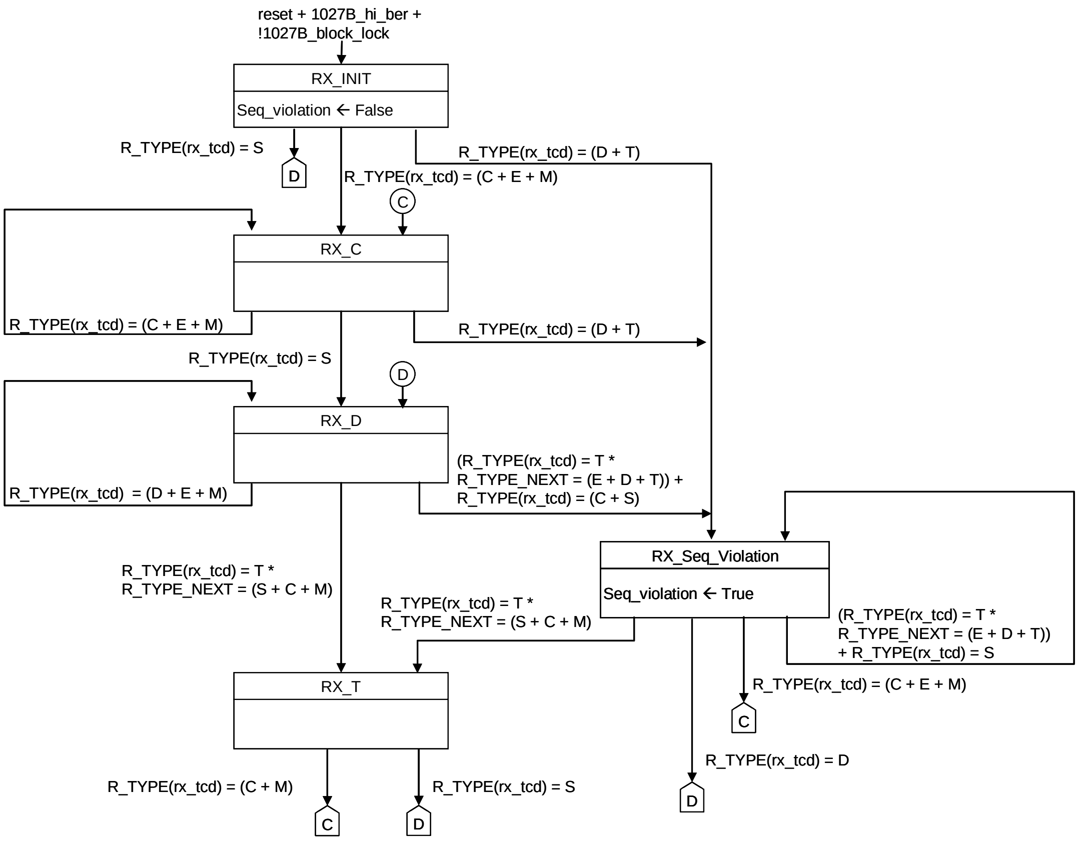


Figure F.3 – Trans-decode state machine for the 64B/66B code blocks including the lane alignment markers

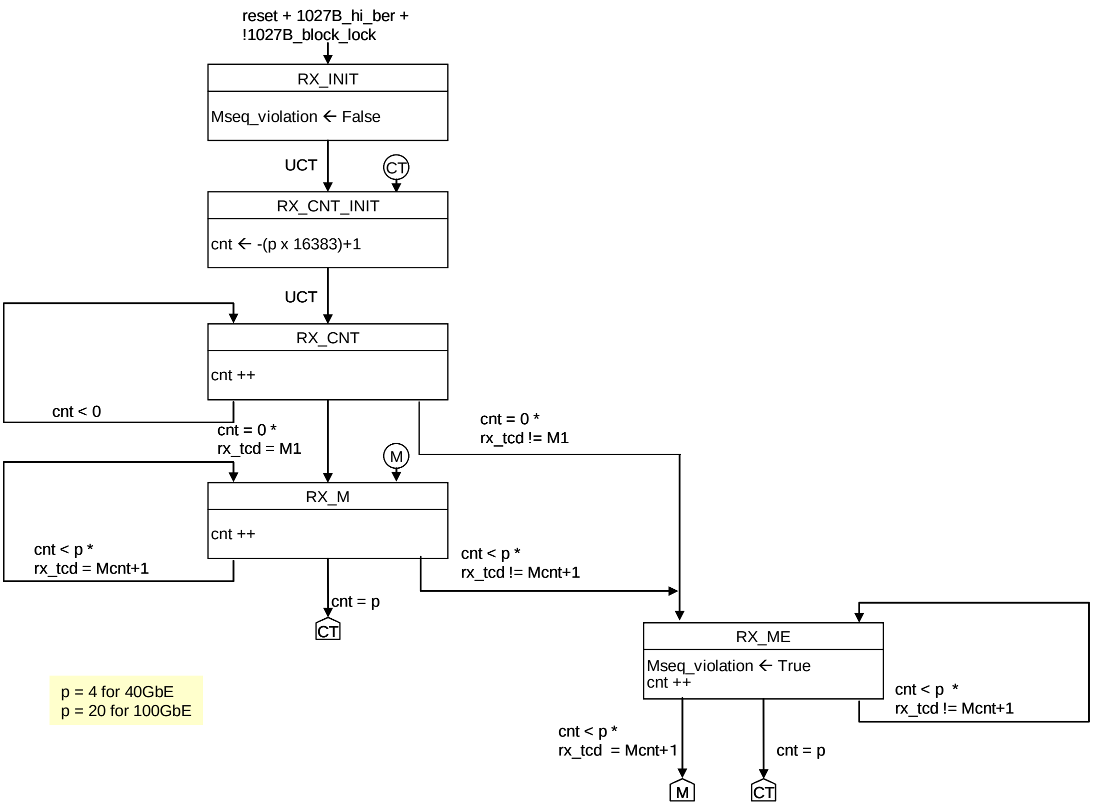


Figure F.4 – Receive state machine for the lane alignment markers

Appendix I

Range of stuff ratios for asynchronous mappings of CBR2G5, CBR10G, and CBR40G clients with ±20 ppm bit-rate tolerance into OPUk, and for asynchronous multiplexing of ODUj into ODUk (k > j)

(This appendix does not form an integral part of this Recommendation.)

Clause 17.2 describes asynchronous and bit-synchronous mappings of CBR2G5, CBR10G, and CBR40G clients with ±20 ppm bit-rate tolerance into ODU1, 2, and 3, respectively. Clause 19 describes asynchronous mapping (multiplexing) of ODUj into ODUk (k > j). For asynchronous CBR client mappings, any frequency difference between the client and local OPUk server clocks is accommodated by the +1/0/–1 justification scheme. For asynchronous multiplexing of ODUj into ODUk (k > j), any frequency difference between the client ODUj and local OPUk server clocks is accommodated by the +2/+1/0/–1 justification scheme. The OPUk payload, ODUk, and OTUk bit rates and tolerances are given in clause 7.3. The ODU1, ODU2, and ODU3 rates are 239/238, 239/237, and 239/236 times 2 488 320 kbit/s, 9 953 280 kbit/s, and 39 813 120 kbit/s, respectively. The ODUk bit-rate tolerances are ±20 ppm. This appendix shows that each justification scheme can accommodate these bit rates and tolerances for the respective mappings, and also derives the range of justification (stuff) ratio for each mapping.

The +1/0/–1 mapping in clause 17.2 provides for one positive justification opportunity (PJO) and one negative justification opportunity (NJO) in each ODUk frame. The +2/+1/0/–1 mapping in clause 19 provides for 2 PJOs and one NJO in each ODUk frame. For the case of ODU multiplexing (i.e., the latter case), the ODUj being mapped will get only a fraction of the full payload capacity of the ODUk. There can be, in general, a number of fixed stuff bytes per ODUj or CBR client. Note that in both mapping cases, there is one stuff opportunity in every ODUk frame. For mapping of a CBR client into ODUk, the CBR client is allowed to use all the stuff opportunities (because only one CBR client signal is mapped into an ODUk). However, for mapping ODUj into ODUk (k > j), the ODUj can only use 1/2 (ODU0 into ODU1), 1/4 (ODU1 into ODU2 or ODU2 into ODU3) or 1/16 (ODU1 into ODU3) of the stuff opportunities. The other stuff opportunities are needed for the other clients being multiplexed into the ODUk.

Traditionally, the justification ratio (stuff ratio) for purely positive justification schemes is defined as the long-run average fraction of justification opportunities for which a justification is done (i.e., for a very large number of frames, the ratio of the number of justifications to the total number of justification opportunities). In the +1/0/–1 scheme, positive and negative justifications must be distinguished. This is done by using different algebraic signs for positive and negative justifications. With this convention, the justification ratio can vary at most (for sufficiently large frequency offsets) from –1 to +1 (in contrast to a purely positive justification scheme, where the justification ratio can vary at most from 0 to 1). In the case of ODUk multiplexing, the justification ratio is defined relative to the stuff opportunities available for the client in question. Then, the justification ratio can vary (for sufficiently large frequency offsets) from –1 to +2. (If the justification ratio were defined relative to all the stuff opportunities for all the clients, the range would be –1/2 to +1 for multiplexing ODU0 into ODU1, –1/4 to +1/2 for multiplexing ODU1 into ODU2 and ODU2 into ODU3, and –1/16 to +1/8 for multiplexing ODU1 into ODU3.)

Let α represent the justification ratio ($-1 \le \alpha \le 1$ for CBR client into ODUk mapping; $-2 \le \alpha \le 1$ for ODUj into ODUk mapping (k > j)), and use the further convention that positive α will correspond to negative justification and negative α to positive justification (the reason for this convention is explained below).

Define the following notation (the index j refers to the possible ODUj client being mapped, and the index k refers to the ODUk server layer into which the ODUj or CBR client is mapped):

N = number of fixed stuff bytes in the OPUk payload area associated with the client in question (note that this is not the total number of fixed stuff bytes if multiple clients are being multiplexed)

S = nominal STM-N or ODUj client rate (bytes/s)

T = nominal ODUk frame period(s)

y_c = client frequency offset (fraction)

y_s = server frequency offset (fraction)

p = fraction of OPUk payload area available to this client

N_f = average number of client bytes mapped into an ODUk frame, for the particular frequency offsets (averaged over a large number of frames)

Then N_f is given by:

$$N_f = ST\frac{1+y_c}{1+y_s} \quad \text{(I-1)}$$

For frequency offsets small compared to 1, this may be approximated:

$$N_f = ST(1+y_c-y_s) \equiv ST\beta \quad \text{(I-2)}$$

The quantity β–1 is the net frequency offset due to client and server frequency offset.

Now, the average number of client bytes mapped into an ODUk frame is also equal to the total number of bytes in the payload area available to this client (which is (4)(3808)p = 15232p), minus the number of fixed stuff bytes for this client (N), plus the average number of bytes stuffed for this client over a very large number of frames. The latter is equal to the justification ratio α multiplied by the fraction of frames p corresponding to justification opportunities for this client. Combining this with equation I-1 produces:

$$ST\beta = \alpha p + 15232p - N \quad \text{(I-3)}$$

In equation I-3, a positive α corresponds to more client bytes mapped into the ODUk, on average. As indicated above, this corresponds to negative justification. This sign convention is used so that α enters in equation I-3 with a positive sign (for convenience).

Equation I-3 is the main result. For mapping STM-N (CBR clients) into ODUk, the quantity p is 1.

The range of stuff ratio may now be determined for mapping STM-N or ODUj clients into ODUk, using equation I-3. In what follows, let R_{16} be the STM-16 rate, i.e., 2.48832 Gbit/s = 3.1104×10^8 bytes/s.

Asynchronous mapping of CBR2G5 (2 488 320 kbit/s) signal into ODU1

The nominal client rate is $S = R_{16}$. The nominal ODU1 rate is (239/238)S (see clause 7.3). But the nominal ODU1 rate is also equal to (4)(3824)/T. Then:

$$\mathrm{ST} = (4)(3824)\frac{238}{239} = 15232 \quad \text{(I-4)}$$

Inserting this into equation I-3, and using the fact that $N = 0$ (no fixed stuff bytes) for this mapping produces

$$\alpha = 15232(\beta - 1) \quad \text{(I-5)}$$

Since the ODUk and client frequency tolerances are each ±20 ppm, β ranges from 0.99996 to 1.00004. Using this in equation I-5 gives as the range of α:

$$-0.60928 \le \alpha \le +0.60928 \tag{I-6}$$

Asynchronous mapping of CBR10G (9 953 280 kbit/s) signal into ODU2

The nominal client rate is $S = 4R_{16}$. The nominal ODU2 rate is (239/237)S (see clause 7.3). But the nominal ODU2 rate is also equal to (4)(3824)/T. Then:

$$ST = (4)(3824)\frac{237}{239} = 15168 \tag{I-7}$$

Inserting this into equation I-3, and using the fact that $N = 64$ (number of fixed stuff bytes) for this mapping produces:

$$\alpha = 15168\beta + 64 - 15232 = 15168(\beta - 1) \tag{I-8}$$

As before, the ODUk and client frequency tolerances are ±20 ppm, and β ranges from 0.99996 to 1.00004. Using this in equation I-8 gives as the range of α:

$$-0.60672 \le \alpha \le +0.60672 \tag{I-9}$$

Asynchronous mapping of CBR40G (39 813 120 kbit/s) signal into ODU3

The nominal client rate is $S = 16R_{16}$. The nominal ODU3 rate is (239/236)S (see clause 7.3). But the nominal ODU3 rate is also equal to (4)(3824)/T. Then:

$$\mathrm{ST} = (4)(3824)\frac{236}{239} = 15104 \tag{I-10}$$

Inserting this into equation I-3, and using the fact that $N = 128$ (number of fixed stuff bytes) for this mapping produces:

$$\alpha = 15104\beta + 128 - 15232 = 15104(\beta - 1) \tag{I-11}$$

As before, the ODUk and client frequency tolerances are ±20 ppm, and β ranges from 0.99996 to 1.00004. Using this in equation I-11 gives as the range of α:

$$-0.60416 \le \alpha \le +0.60416 \tag{I-12}$$

ODU1 into ODU2 multiplexing

The ODU1 nominal client rate is (see clause 7.3):

$$\mathrm{S} = \frac{239}{238}\mathrm{R}_{16} \tag{I-13}$$

The ODU2 nominal frame time is:

$$\mathrm{T} = \frac{(3824)(4)}{\frac{239}{237}(4\mathrm{R}_{16})} \tag{I-14}$$

The fraction p is 0.25. Inserting into equation I-3 produces:

$$\frac{239}{238}\mathrm{R}_{16}\frac{(3824)(4)}{\frac{239}{237}(4\mathrm{R}_{16})}\beta = \frac{\alpha}{4} + 3808 - \mathrm{N} \tag{I-15}$$

Simplifying and solving for α produces:

$$\alpha = \frac{237}{238}(15296)\beta + 4N - 15232 \quad \text{(I-16)}$$

Now let β = 1 + y, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned}\alpha &= \frac{237}{238}(15296) - 15232 + 4\text{N} + \frac{237}{238}(15296)\text{y} \\ &= 4\text{N} - 0.2689076 + 15231.731092\text{y}\end{aligned} \quad \text{(I-17)}$$

The number of fixed stuff bytes N is zero, as given in clause 19.5.1. The client and mapper frequency offsets are in the range ±20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ±40 ppm. Inserting these values into equation I-17 gives for the range for α:

$$\begin{aligned}\alpha &= 0.340362 && \text{for } y = +40\text{ ppm} \\ \alpha &= -0.268908 && \text{for } y = 0\text{ ppm} \\ \alpha &= -0.878177 && \text{for } y = -40\text{ ppm}\end{aligned} \quad \text{(I-18)}$$

In addition, stuff ratios of –2 and +1 are obtained for frequency offsets of –113.65 ppm and 83.30 ppm, respectively. The range of frequency offset that can be accommodated is approximately 197 ppm. This is 50% larger than the range that can be accommodated by a +1/0/–1 justification scheme (see above), and is due to the additional positive stuff byte.

ODU2 into ODU3 multiplexing

The ODU2 nominal client rate is (see clause 7.3):

$$S = \frac{239}{237}(4R_{16}) \quad \text{(I-19)}$$

The ODU3 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{236}(16R_{16})} \quad \text{(I-20)}$$

The fraction p is 0.25. Inserting into equation I-3 produces:

$$\frac{239}{237}4R_{16}\frac{(3824)(4)}{\frac{239}{236}(16R_{16})}\beta = \frac{\alpha}{4} + 3808 - N \quad \text{(I-21)}$$

Simplifying and solving for α produces:

$$\alpha = \frac{236}{237}(15296)\beta + 4N - 15232 \quad \text{(I-22)}$$

As before, let β = 1 + y, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned}\alpha &= \frac{236}{237}(15296) - 15232 + 4N + \frac{236}{237}(15296)y \\ &= 4N - 0.5400844 + 15231.459916y\end{aligned} \quad \text{(I-23)}$$

The number of fixed stuff bytes N is zero, as given in clause 19.5.3. The client and mapper frequency offsets are in the range ±20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ±40 ppm. Inserting these values into equation I-23 gives for the range for α:

$$\begin{aligned} \alpha &= 0.0691740 \quad &\text{for} \quad y = +40\,\text{ppm} \\ \alpha &= -0.5400844 \quad &\text{for} \quad y = 0\,\text{ppm} \\ \alpha &= -1.149343 \quad &\text{for} \quad y = -40\,\text{ppm} \end{aligned} \tag{I-24}$$

In addition, stuff ratios of –2 and +1 are obtained for frequency offsets of –95.85 ppm and 101.11 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 197 ppm, which is 50% larger than the range that can be accommodated by a +1/0/–1 justification scheme (see above) due to the additional positive stuff byte.

ODU1 into ODU3 multiplexing

The ODU1 nominal client rate is (see clause 7.3):

$$S = \frac{239}{238}(R_{16}) \tag{I-25}$$

The ODU3 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{236}(16R_{16})} \tag{I-26}$$

The fraction p is 0.0625. Inserting into equation I-3 produces:

$$\frac{239}{238}R_{16}\frac{(3824)(4)}{\frac{239}{236}(16R_{16})}\beta = \frac{\alpha}{16} + 952 - N \tag{I-27}$$

Simplifying and solving for α produces:

$$\alpha = \frac{236}{238}(15296)\beta + 16N - 15232 \tag{I-28}$$

As before, let $\beta = 1 + y$, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned} \alpha &= \frac{236}{238}(15296) - 15232 + 16N + \frac{236}{238}(15296)y \\ &= 16N - 64.5378151 + 15167.462185y \end{aligned} \tag{I-29}$$

The total number of fixed stuff bytes in the ODU3 payload is 64, as given in clause 19.5.2; the number for one ODU1 client, N, is therefore 4. The client and mapper frequency offsets are in the range ±20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ±40 ppm. Inserting these values into equation I-29 gives for the range for α:

$$\begin{aligned} \alpha &= 0.0688834 \quad &\text{for} \quad y = +40\,\text{ppm} \\ \alpha &= -0.5378151 \quad &\text{for} \quad y = 0\,\text{ppm} \\ \alpha &= -1.144514 \quad &\text{for} \quad y = -40\,\text{ppm} \end{aligned} \tag{I-30}$$

In addition, stuff ratios of –2 and +1 are obtained for frequency offsets of –96.40 ppm and 101.39 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 197 ppm, which is 50% larger than the range that can be accommodated by a +1/0/–1 justification scheme (see above) due to the additional positive stuff byte.

ODU0 into ODU1 multiplexing

The ODU0 nominal client rate is (see clause 7.3):

$$S = \frac{1}{2}(R_{16}) \tag{I-31}$$

The ODU1 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{238}(R_{16})} \tag{I-32}$$

The fraction p is 0.5. Inserting into equation I-3 produces:

$$\frac{1}{2}R_{16}\frac{(3824)(4)}{\frac{239}{238}(R_{16})}\beta = \frac{\alpha}{2} + 7616 - N \tag{I-33}$$

Simplifying and solving for α produces:

$$\alpha = \frac{238}{239}(15296)\beta + 2N - 15232 \tag{I-34}$$

As before, let β = 1 + y, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned}\alpha &= \frac{238}{239}(15296) - 15232 + 2N + \frac{238}{239}(15296)y \\ &= 2N + 15232y\end{aligned} \tag{I-35}$$

The total number of fixed stuff bytes N is zero, as given in clause 19.5.4. The client and mapper frequency offsets are in the range ±20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ±40 ppm. Inserting these values into equation I-35 gives for the range for α:

$$\begin{aligned}&\alpha = 0.6092800 &&\text{for} \quad y = +40\,\text{ppm} \\ &\alpha = 0.0000000 &&\text{for} \quad y = 0\,\text{ppm} \\ &\alpha = -0.6092800 &&\text{for} \quad y = -40\,\text{ppm}\end{aligned} \tag{I-36}$$

In addition, stuff ratios of –2 and +1 are obtained for frequency offsets of –130 ppm and 65 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 195 ppm.

Appendix II

Examples of functionally standardized OTU frame structures

(This appendix does not form an integral part of this Recommendation.)

This appendix provides examples of functionally standardized OTU frame structures. These examples are for illustrative purposes and by no means imply a definition of such structures. The completely standardized OTUk frame structure as defined in this Recommendation is shown in Figure II.1. Functionally standardized OTUkV frame structures will be needed to support, e.g., alternative FEC. Examples of OTUkV frame structures are:

- OTUkV with the same overhead byte allocation as the OTUk, but use of an alternative FEC as shown in Figure II.2;
- OTUkV with the same overhead byte allocation as the OTUk, but use of a smaller, alternative FEC code and the remainder of the OTUkV FEC overhead area filled with fixed stuff as shown in Figure II.3;
- OTUkV with a larger FEC overhead byte allocation as the OTUk, and use of an alternative FEC as shown in Figure II.4;
- OTUkV with no overhead byte allocation for FEC as shown in Figure II.5;
- OTUkV with a different frame structure than the OTUk frame structure, supporting a different OTU overhead (OTUkV overhead and OTUkV FEC) as shown in Figure II.6;
- OTUkV with a different frame structure than the OTUk frame structure, supporting a different OTU overhead (OTUkV overhead) and with no overhead byte allocation for FEC as shown in Figure II.7.

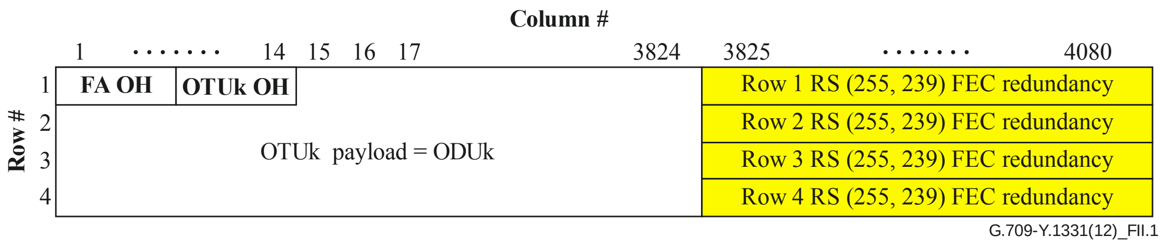


Figure II.1 – OTUk (with RS(255,239) FEC)

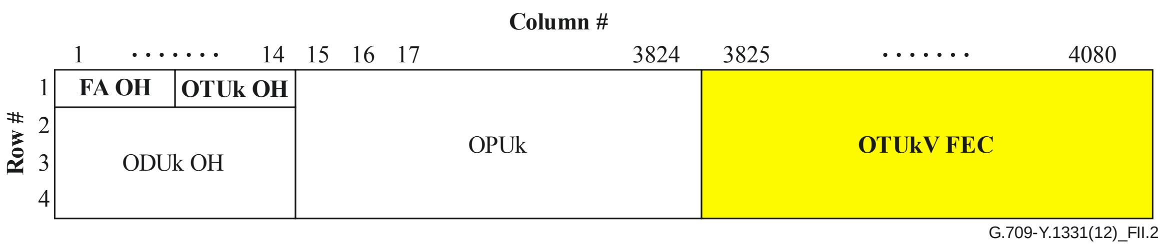


Figure II.2 – OTUk with alternative OTUkV FEC (OTUk-v)

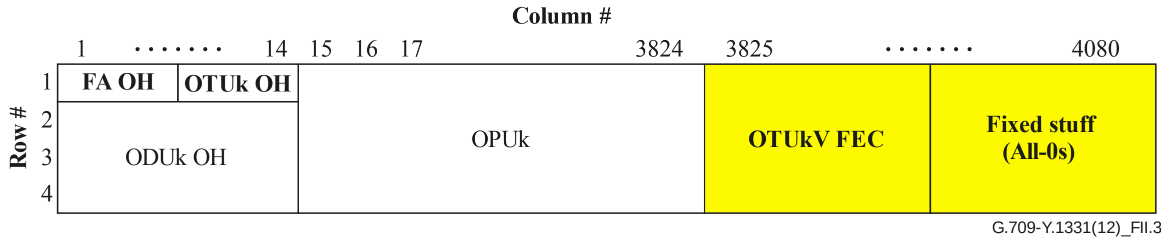


Figure II.3 – OTUk with a smaller OTUkV FEC and the remainder of an FEC area filled with fixed stuff

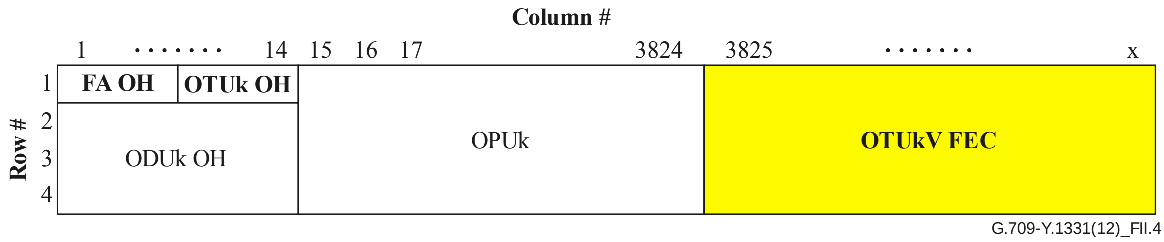


Figure II.4 – OTUk with a larger OTUkV FEC

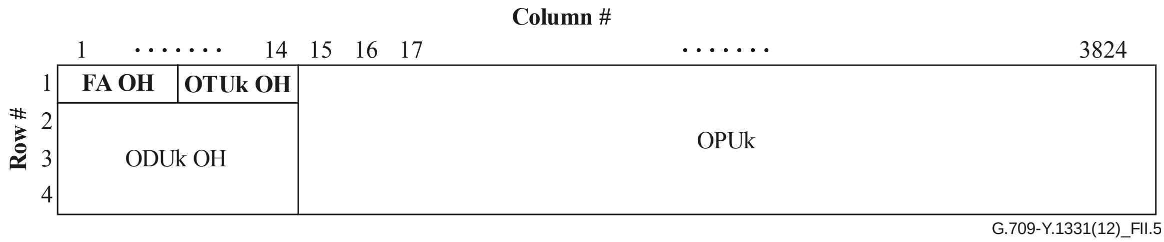


Figure II.5 – OTUk without an OTUkV FEC area

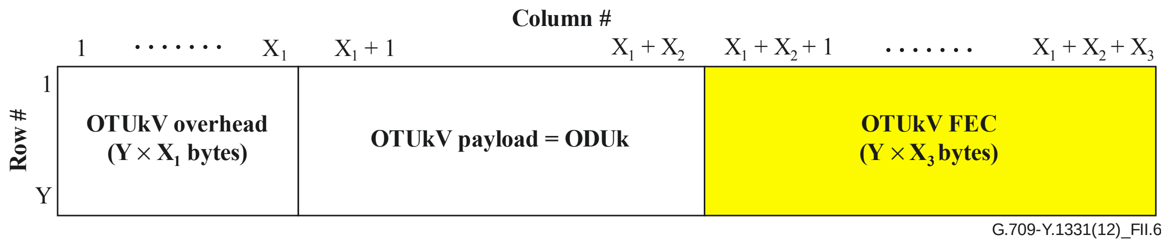


Figure II.6 – OTUkV with a different frame structure

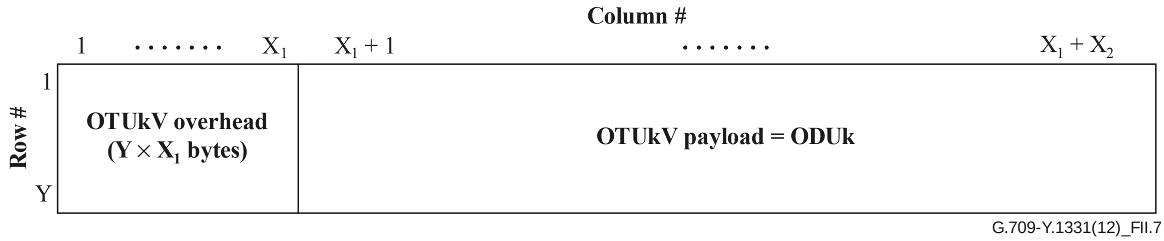


Figure II.7 – OTUkV with a different frame structure and without FEC area

For the case of Figures II.6 and II.7, the mapping of the ODUk signal can be either asynchronous, bit-synchronous, or frame synchronous.

For the case of asynchronous mapping, the ODUk and OTUkV bit rates can be asynchronous. The ODUk signal is mapped as a bit stream into the OTUkV payload area using a stuffing technique.

For the case of bit-synchronous mapping, the ODUk and OTUkV bit rates are synchronous. The ODUk signal is mapped into the OTUkV payload area without stuffing. The ODUk frame is not related to the OTUkV frame.

For the case of a frame synchronous mapping, the ODUk and OTUkV bit rates are synchronous and the frame structures are aligned. The ODUk signal is mapped into the OTUkV payload area without stuffing and with a fixed position of the ODUk frame within the OTUkV frame. (See Figure II.8.)

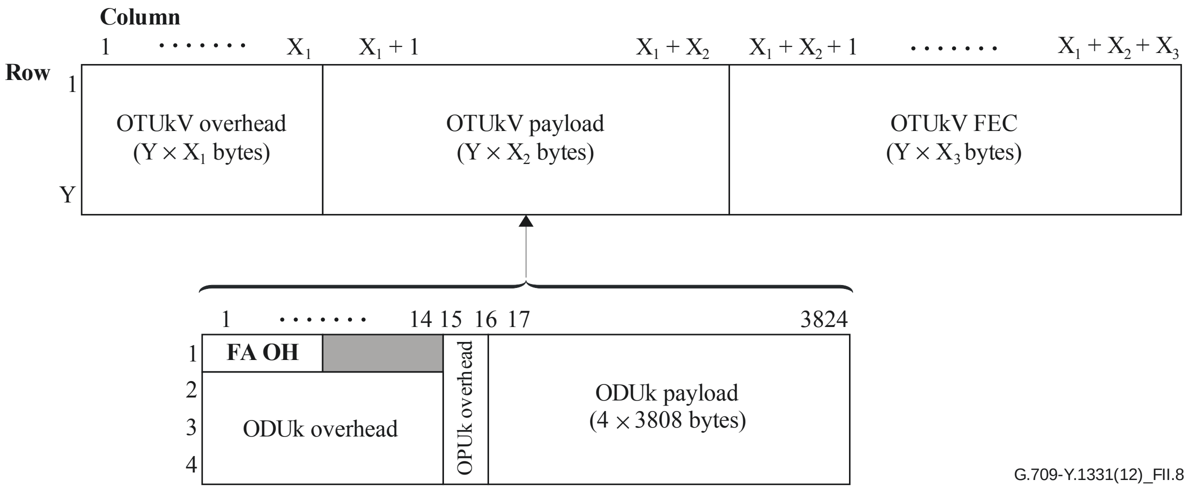


Figure II.8 – Asynchronous (or bit-synchronous) mapping of ODUk into OTUkV

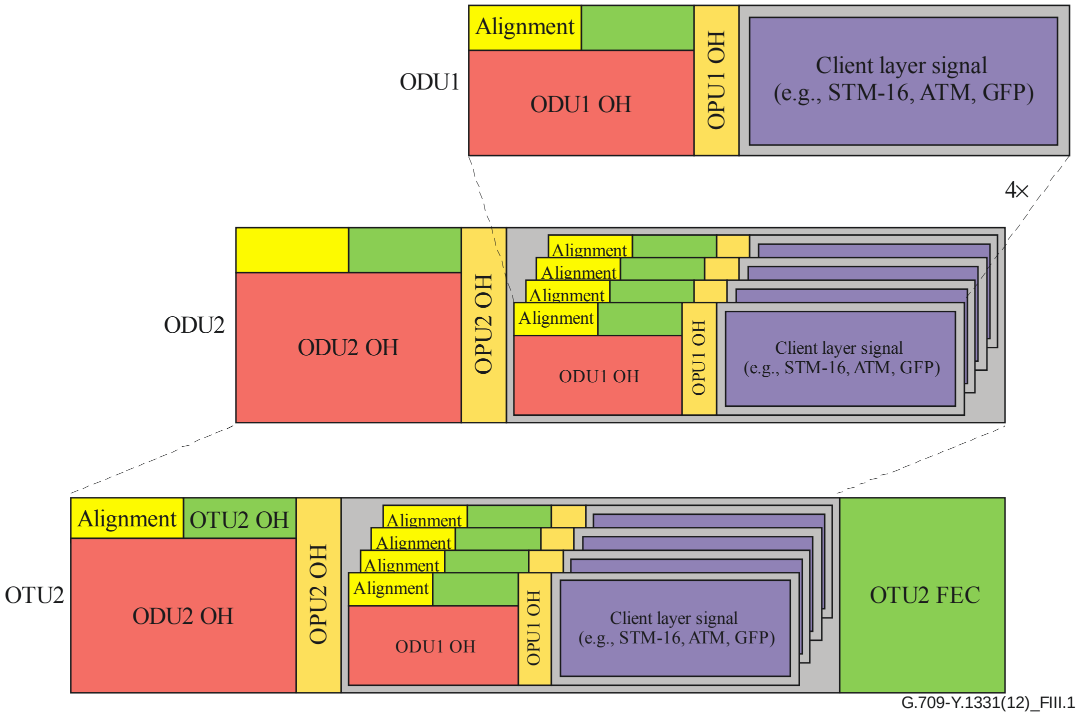
Appendix III

Example of ODUk multiplexing

(This appendix does not form an integral part of this Recommendation.)

Figure III.1 illustrates the multiplexing of four ODU1 signals into an ODU2. The ODU1 signals including the frame alignment overhead and an all-0s pattern in the OTUk overhead locations are adapted to the ODU2 clock via justification (asynchronous mapping). These adapted ODU1 signals are byte interleaved into the OPU2 payload area, and their justification control and opportunity signals (JC, NJO) are frame interleaved into the OPU2 overhead area.

ODU2 overhead is added after which the ODU2 is mapped into the OTU2 [or OTU2V]. OTU2 [or OTU2V] overhead and frame alignment overhead are added to complete the signal for transport via an OTM signal.



NOTE – The ODU1 floats in a quarter of the OPU2 payload area. An ODU1 frame will cross multiple ODU2 frame boundaries. A complete ODU1 frame (15296 bytes) requires the bandwidth of (15296/3808) 4.017 ODU2 frames. This is not illustrated.

Figure III.1 – Example of multiplexing 4 ODU1 signals into an ODU2

Figure III.2 illustrates the multiplexing of two ODU0 signals into an ODU1. The ODU0 signals including the frame alignment overhead and an all-0s pattern in the OTUk overhead locations are adapted to the ODU1 clock via justification (asynchronous mapping). These adapted ODU0 signals are byte interleaved into the OPU1 payload area, and their justification control and opportunity signals (JC, NJO) are frame interleaved into the OPU1 overhead area and ODU1 overhead is added.

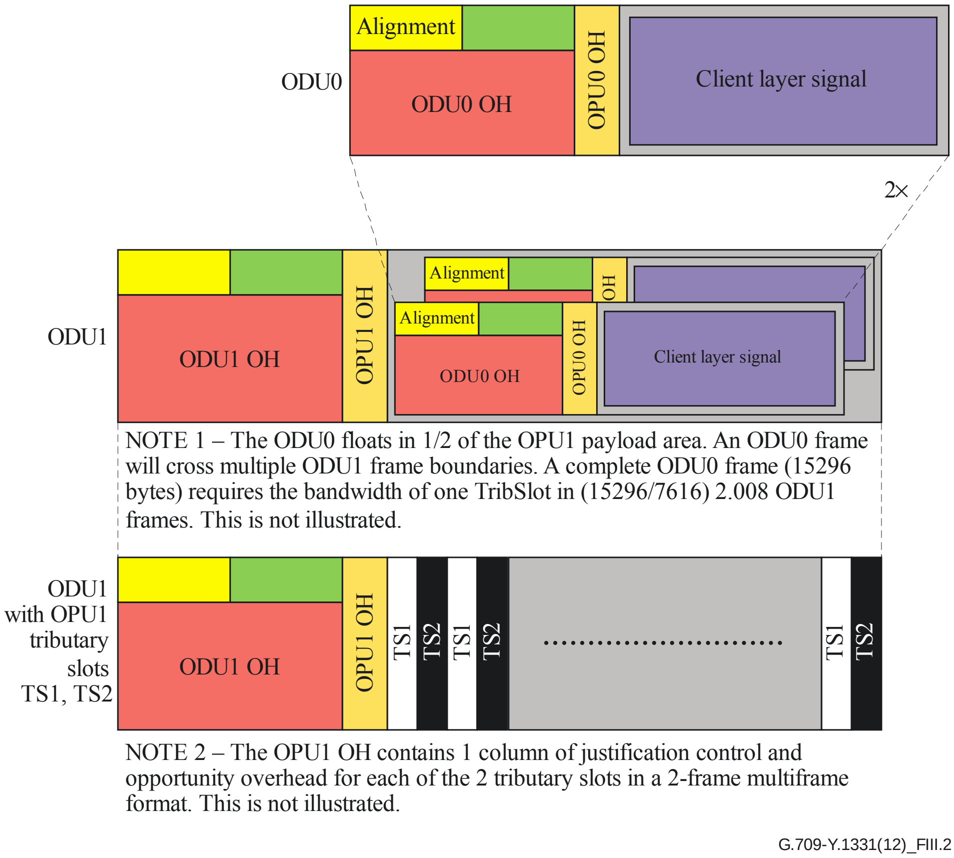


Figure III.2 – Example of multiplexing 2 ODU0 signals into an ODU1

Appendix IV

Example of fixed stuff in OPUk with multiplex of lower-order ODUk signals

(This appendix does not form an integral part of this Recommendation.)

When an OPU3 transports 16 ODU1 signals, columns 1905 to 1920 of the OPU3 contain fixed stuff, one fixed stuff column for each of the 16 ODU1 signals.

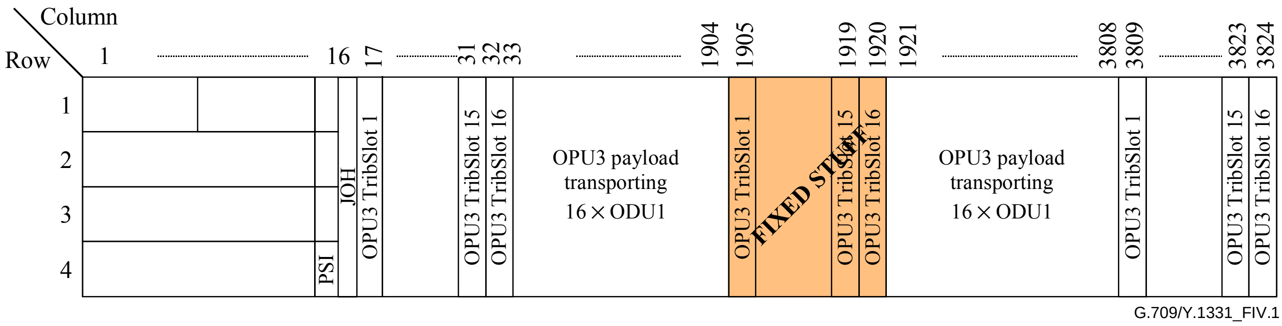


Figure IV.1 – Fixed stuff locations when mapping 16 × ODU1 into OPU3

Appendix V

ODUk multiplex structure identifier (MSI) examples

(This appendix does not form an integral part of this Recommendation.)

The following figures present four examples of ODU1 and ODU2 carriage within an OPU3 and the associated MSI encoding.

	1 2	3 4 5 6 7 8	
PSI[2]	00	000000	*TS*1
PSI[3]	00	000001	*TS*2
PSI[4]	00	000010	*TS*3
PSI[5]	00	000011	*TS*4
PSI[6]	00	000100	*TS*5
PSI[7]	00	000101	*TS*6
PSI[8]	00	000110	*TS*7
PSI[9]	00	000111	*TS*8
PSI[10]	00	001000	*TS*9
PSI[11]	00	001001	*TS*10
PSI[12]	00	001010	*TS*11
PSI[13]	00	001011	*TS*12
PSI[14]	00	001100	*TS*13
PSI[15]	00	001101	*TS*14
PSI[16]	00	001110	*TS*15
PSI[17]	00	001111	*TS*16

Figure V.1 – OPU3-MSI coding for the case of 16 ODU1s into OPU3

	1 2	3 4 5 6 7 8	
PSI[2]	01	000000	*TS*1
PSI[3]	01	000001	*TS*2
PSI[4]	01	000010	*TS*3
PSI[5]	01	000011	*TS*4
PSI[6]	01	000000	*TS*5
PSI[7]	01	000001	*TS*6
PSI[8]	01	000010	*TS*7
PSI[9]	01	000011	*TS*8
PSI[10]	01	000000	*TS*9
PSI[11]	01	000001	*TS*10
PSI[12]	01	000010	*TS*11
PSI[13]	01	000011	*TS*12
PSI[14]	01	000000	*TS*13
PSI[15]	01	000001	*TS*14
PSI[16]	01	000010	*TS*15
PSI[17]	01	000011	*TS*16

Figure V.2 – OPU3-MSI coding for the case of 4 ODU2s into OPU3 TS# (1, 5, 9, 13), (2, 6, 10, 14), (3, 7, 11, 15) and (4, 8, 12, 16)

	1 2	3 4 5 6 7 8	
PSI[2]	01	000000	*TS*1
PSI[3]	01	000001	*TS*2
PSI[4]	01	000001	*TS*3
PSI[5]	01	000010	*TS*4
PSI[6]	01	000000	*TS*5
PSI[7]	01	000011	*TS*6
PSI[8]	01	000011	*TS*7
PSI[9]	01	000011	*TS*8
PSI[10]	01	000000	*TS*9
PSI[11]	01	000000	*TS*10
PSI[12]	01	000001	*TS*11
PSI[13]	01	000001	*TS*12
PSI[14]	01	000011	*TS*13
PSI[15]	01	000010	*TS*14
PSI[16]	01	000010	*TS*15
PSI[17]	01	000010	*TS*16

Figure V.3 – OPU3-MSI coding for the case of 4 ODU2s into OPU3 TS# (1, 5, 9, 10), (2, 3, 11, 12), (4, 14, 15, 16) and (6, 7, 8, 13)

	1 2	3 4 5 6 7 8	
PSI[2]	01	000000	*TS*1
PSI[3]	00	000001	*TS*2
PSI[4]	00	000010	*TS*3
PSI[5]	01	000001	*TS*4
PSI[6]	01	000000	*TS*5
PSI[7]	00	000101	*TS*6
PSI[8]	00	000110	*TS*7
PSI[9]	01	000001	*TS*8
PSI[10]	01	000000	*TS*9
PSI[11]	01	000001	*TS*10
PSI[12]	00	001010	*TS*11
PSI[13]	00	001011	*TS*12
PSI[14]	01	000000	*TS*13
PSI[15]	00	001101	*TS*14
PSI[16]	00	001110	*TS*15
PSI[17]	01	000001	*TS*16

Figure V.4 – OPU3-MSI coding for the case of 5 ODU1s and 2 ODU2s into OPU3 TS# (2), (6), (11), (12), (14), (1, 5, 9, 13) and (4, 8, 10, 16) and OPU3 TS# 3, 7, 15 unallocated (default to ODU1)

Appendix VI

Parallel logic implementation of the CRC-8 and CRC-5

(This appendix does not form an integral part of this Recommendation.)

CRC-8

Table VI.1 illustrates example logic equations for a parallel implementation of the CRC-8 using the g(x) = $x^8 + x^3 + x^2 + 1$ polynomial over the JC1-JC2. An "X" in a row of the table indicates that the message bit of that column is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC1.C1 corresponds to the first bit (MSB) of the first mapping overhead octet (JC1), JC1.C2 corresponds to bit 2 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc8 are inserted into the JC3 octet with crc1 occupying MSB and crc8 the LSB of the octet.

Table VI.1 – Parallel logic equations for the CRC-8 implementation

Mapping overhead bits	CRC checksum bits							
	crc1	**crc2**	**crc3**	**crc4**	**crc5**	**crc6**	**crc7**	**crc8**
JC1.C1		X				X		X
JC1.C2	X		X			X		
JC1.C3		X		X			X	
JC1.C4			X		X			X
JC1.C5	X			X			X	
JC1.C6		X			X			X
JC1.C7	X		X				X	
JC1.C8		X		X				X
JC2.C9	X		X		X	X	X	
JC2.C10		X		X		X	X	X
JC2.C11	X		X		X	X		X
JC2.C12	X	X		X				
JC2.C13		X	X		X			
JC2.C14			X	X		X		
JC2.II				X	X		X	
JC2.DI					X	X		X

CRC-5

Table VI.2 illustrates example logic equations for a parallel implementation of the CRC-5 using the g(x) = $x^5 + x + 1$ polynomial over the JC4-JC5 C_nD fields. An "X" in a row of the table indicates that the message bit of that column is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC4.D1 corresponds to the first bit (MSB) of the first mapping overhead octet (JC1), JC4.D2 corresponds to bit 2 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc5 are inserted into the JC6 octet with crc1 occupying JC6 bit 4 and crc5 the JC6 bit 8.

Table VI.2 – Parallel logic equations for the CRC-5 implementation

Mapping overhead bits	CRC checksum bits				
	crc1	**crc2**	**crc3**	**crc4**	**crc5**
JC4.D1	X		X	X	
JC4.D2		X		X	X
JC4.D3	X		X		
JC4.D4		X		X	
JC4.D5			X		X
JC5.D6	X			X	X
JC5.D7	X	X			
JC5.D8		X	X		
JC5.D9			X	X	
JC5.D10				X	X

Appendix VII

OTL4.10 structure

(This appendix does not form an integral part of this Recommendation.)

The purpose of the OTM-0.4v4 interface, as defined in clause 8.1.3, is to enable the re-use of modules developed for Ethernet 100GBASE-LR4 or 100GBASE-ER4 interfaces. These modules have corresponding optical specifications for OTU4 interfaces with the optical parameters as specified for the application codes 4I1-9D1F and 4L1-9C1F, respectively, in [ITU-T G.959.1]. These modules have a four-lane WDM interface to and from a transmit/receive pair of ITU-T G.652 optical fibres, and connect to the host board via a 10-lane electrical interface. The conversion between 10 and 4 lanes is performed using an IEEE 802.3ba PMA sublayer as specified in [IEEE 802.3ba] clause 83. The specification of the 10-lane electrical chip-to-module interface (CAUI) is found in [IEEE 802.3ba] Annex 83B. The application of the OTL4.10 interface is illustrated in Figure VII.1:

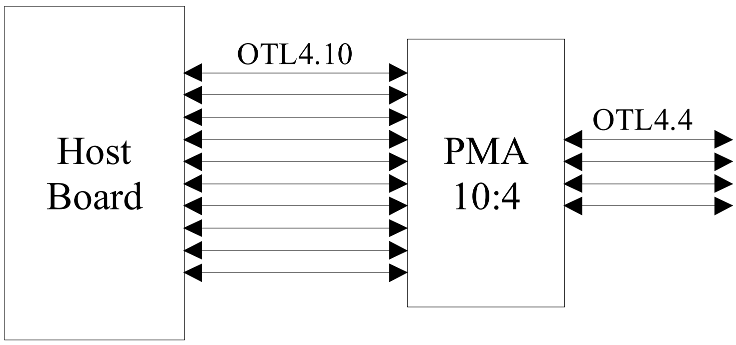


Figure VII.1 – Illustration of the application of an OTL4.10 interface

Each OTL4.10 lane carries two bit-multiplexed logical lanes of an OTU4 as described in Annex C. The logical lane format has been chosen so that the [IEEE 802.3ba] 10:4 PMA (gearbox) will convert the OTU4 signal between a format of 10 lanes of OTL4.10 and four lanes of OTL4.4. Each OTL4.4 lane carries five bit-multiplexed logical lanes of an OTU4 as described in Annex C.

The bit rate of an OTL4.10 lane is indicated in Table VII.1.

Table VII.1 – OTL types and capacity

OTL type	OTL nominal bit rate	OTL bit-rate tolerance
OTL4.10	255/227 × 9 953 280 kbit/s	±20 ppm
NOTE – The nominal OTL4.10 rate is approximately: 11 180 997.357 kbit/s.		

Appendix VIII

CPRI into LO ODU mapping

(This appendix does not form an integral part of this Recommendation.)

CPRI constant bit rate signals (CPRI options 1 to 6) may be transported over an ODUk connection. These CBR signals are mapped into an LO OPUk via the generic mapping procedure as specified in clause 17.7 for CPRI options 1 to 3 and via the bit-synchronous mapping procedure as specified in clause 17.9 for CPRI options 4 to 6.

Two CPRI signals (options 1 and 2) are transported via OPU0, one CPRI signal (option 3) is transported via OPU1 and three CPRI signals (options 4, 5 and 6) are transported via OPUflex. The GMP C_m and C_n (n=1) values associated with the CPRI options 1 to 3 signals are presented in Tables VIII.1 and VIII.2.

The use of the "Experimental mapping" payload type (code 0x01) is suggested.

NOTE – OTN transport of CPRI [b-CPRI] is intended for use within an administrative domain. Users of this Recommendation should not assume that the required performance for the CPRI client is met. It is the responsibility of the network operator to determine if the required performance can be met. The noise generated by the OTN would have to be handled by the CPRI system in order to meet the application requirements. This is considered as a complex task according to the current OTN specification. The OTN network should also be designed in order to meet the applicable symmetry requirements.

Further details are provided below.

Simulation analyses were done for the transport of CPRI Option 2, Option 3, and Option 4 clients over OTN for the following four cases:

a) CPRI Option 2 client signal → ODU0 → ODU2 → OTU2 → ODU2 → ODU0 → CPRI Option 2 client signal

b) CPRI Option 3 client signal → ODU1 → ODU2 → OTU2 → ODU2 → ODU1 → CPRI Option 3 client signal

c) CPRI Option 3 client signal → ODU1 → OTU1 → ODU1 → CPRI Option 3

d) CPRI Option 4 client signal → ODU2 → OTU2 → ODU2 → CPRI Option 4

In accordance with this Appendix VIII, the mappings of the CPRI Option 2 client to ODU0 and the CPRI Option 3 client to ODU1 are via GMP. The CPRI Option 4 client is mapped to ODUflex, and the ODUflex is mapped to ODU2 via GMP. Finally, in (a) the ODU0 is mapped to ODU2 via GMP, and in (b) the ODU1 is mapped to ODU2 via AMP. Cases (a) and (b) have a single mapping of the CPRI client to OTN and one level of OTN multiplexing. Cases (c) and (d) have a single mapping to OTN and no OTN multiplexing.

Simulations were run for no use of additional phase information for the CPRI client to LO ODU mapper (i.e., C_n with n = 8) and 1 UI of additional phase information for the CPRI client to LO ODU mapper (i.e., C_n with n = 1). The desynchronizer bandwidth for the HO ODU to LO ODU demappers was 300 Hz.

The simulation results indicated that, for CPRI client desynchronizer bandwidth in the range of 100-300 Hz (current OTN client desynchronizers are 300 Hz or, in a few cases, 100 Hz or 200 Hz) RMS frequency offset ranges from approximately 113 ppb to 190 ppb for transport of CPRI Option 2 for case (a) and 156 ppb to 317 ppb for transport of CPRI Option 3 for case (b). In addition, for the same range of desynchronizer bandwidths, RMS frequency offset ranges from approximately 29 ppb to 116 ppb for CPRI option 3 for case (c) and 32 ppb to 130 ppb for CPRI Option 4 for case (d).

The simulation results also indicated that, for CPRI client desynchronizer bandwidth in the range of 100-300 Hz, peak-to-peak jitter ranges from approximately 6.9-14.2 UIpp (unit intervals peak-to-peak) for transport of CPRI Option 2 for case (a) and 6.7-14.1 UIpp for transport of CPRI Option 3 for case (b). In addition, for the same range of desynchronizer bandwidths, peak-to-peak jitter ranges from approximately 0.8-7.2 UIpp for CPRI option 3 for case (c) and 0.76-7.2 UIpp for CPRI Option 4 for case (d).

In order to allow compatibility with OTN transport, CPRI REs would need to be designed to tolerate and filter properly at least the noise added by the OTN transport, which is not currently budgeted by CPRI. Additional sources of noise may also exist. The OTN network should also be designed in order to meet the applicable CPRI stringent symmetry requirements; this is something that has not been studied. Interworking between OTN and the CPRI REs, in terms of jitter and wander, is still unknown and has to be considered.

The CPRI replacement signal is the link fault signal as defined in clause 17.7.1.1.

Table VIII.1A – C_m (m=8) for sub-1.238G clients into OPU0

Client signal	**Nominal bit rate (kbit/s)**	**Bit rate tolerance (ppm)**	**Floor $C_{8,min}$**	**Minimum c_8**	**Nominal c_8**	**Maximum c_8**	**Ceiling $C_{8,max}$**
CPRI option 1	614 400	±0.002	7553	7553.429	7553.580	7553.731	7554
CPRI option 2	1 228 800	±0.002	15106	15106.858	15107.160	15107.463	15108

Table VIII.1B – C_n (n=8 or 1) for sub-1.238G clients into OPU0

Client signal	**Nominal bit rate (kbit/s)**	**Bit rate tolerance (ppm)**	**Floor $C_{8,min}$**	**Minimum c_8**	**Nominal c_8**	**Maximum c_8**	**Ceiling $C_{8,max}$**
-	–	–	–	–	–	–	–
			Floor $C_{1,min}$	**Minimum** c_1	**Nominal** c_1	**Maximum** c_1	**Ceiling** $C_{1,max}$
CPRI option 1	614 400	±0.002	60427	60427.433	60428.642	60429.851	60430
CPRI option 2	1 228 800	±0.002	120854	120854.867	120857.284	120859.701	120860

Table VIII.2A – C_m (m=16) for supra-1.238 to sub-2.488G clients into OPU1

Client signal	**Nominal bit rate (kbit/s)**	**Bit rate tolerance (ppm)**	**Floor $C_{16,min}$**	**Minimum c_{16}**	**Nominal c_{16}**	**Maximum c_{16}**	**Ceiling $C_{16,max}$**
CPRI option 3	2 457 600	±0.002	7521	7521.825	7521.975	7522.126	7523

Table VIII.2B – C_n (n=8 or 1) for supra-1.238 to sub-2.488G clients into OPU1

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	Floor $C_{8,min}$	Minimum c_8	Nominal c_8	Maximum c_8	Ceiling $C_{8,max}$
–	–	–	–	–	–	–	–
			Floor $C_{1,min}$	Minimum c_1	Nominal c_1	Maximum c_1	Ceiling $C_{1,max}$
CPRI option 3	2 457 600	±0.002	241709	241709.733	241714.568	241719.403	241720

Table VIII.3 – supra-2.488G CBR clients

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)
CPRI option 4	3 072 000	±0.002
CPRI option 5	4 915 200	±0.002
CPRI option 6	6 144 000	±0.002

Table VIII.4 – Replacement signal for CPRI clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
CPRI option 1	Link Fault	±100
CPRI option 2	Link Fault	±100
CPRI option 3	Link Fault	±100
CPRI option 4	Link Fault	±100
CPRI option 5	Link Fault	±100
CPRI option 6	Link Fault	±100

Table VIII.5 – Number of tributary slots required for ODUj into HO OPUk

	# 2.5G tributary slots		# 1.25G tributary slots			
LO ODU	**OPU2**	**OPU3**	**OPU1**	**OPU2**	**OPU3**	**OPU4**
ODUflex(CBR)						
– ODUflex(CPRI Opt 4)	–	–	–	3	3	3
– ODUflex(CPRI Opt 5)	–	–	–	4	4	4
– ODUflex(CPRI Opt 6)	–	–	–	5	5	5

Table VIII.6 – C_m and C_n (n=8) for ODUj into ODTU2.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	12534	12534.900	12536.404	12537.909	12538
– ODUflex(CPRI 5)	4	32	15041	15041.880	15043.685	15045.490	15046
– ODUflex(CPRI 6)	5	40	15041	15041.880	15043.685	15045.490	15046
			Floor $C_{8,min}$	**Minimum c_8**	**Nominal c_8**	**Maximum c_8**	**Ceiling $C_{8,max}$**
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	37604	37604.700	37609.213	37613.726	38614
– ODUflex(CPRI 5)	4	32	60167	60167.519	60174.740	60181.961	60182
– ODUflex(CPRI 6)	5	40	75209	75209.399	75218.425	75227.452	75228

Table VIII.7 – C_m and C_n (n=8) for ODUj into ODTU3.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	12482	12482.010	12483.508	12485.006	12486
– ODUflex(CPRI 5)	4	32	14978	14978.412	14980.210	14982.007	14983
– ODUflex(CPRI 6)	5	40	14978	14978.412	14980.210	14982.007	14983
			Floor C8,min	**Minimum c8**	**Nominal c8**	**Maximum c8**	**Ceiling C8,max**
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	37446	37446.030	37450.524	37455.018	37456
– ODUflex(CPRI 5)	4	32	59913	59913.648	59920.838	59928.029	59929
– ODUflex(CPRI 6)	5	40	74892	74892.060	74901.048	74910.036	74911

Table VIII.8 – C_m and C_n (n=8) for ODUj into ODTU4.M

ODUj signal	M	m=8×M	Floor $C_{m,min}$	Minimum c_m	Nominal c_m	Maximum c_m	Ceiling $C_{m,max}$
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	12006	12006.001	12007.442	12008.883	12009
– ODUflex(CPRI 5)	4	32	14407	14407.201	14408.930	14410.659	14411
– ODUflex(CPRI 6)	5	40	14407	14407.201	14408.930	14410.659	14411
			Floor $C_{8,min}$	**Minimum c_8**	**Nominal c_8**	**Maximum c_8**	**Ceiling $C_{8,max}$**
ODUflex(CBR)	ODUflex(CBR) dependent						
– ODUflex(CPRI 4)	3	24	36018	36018.003	36022.326	36026.649	36027
– ODUflex(CPRI 5)	4	32	57628	57628.805	57635.722	57642.638	57643
– ODUflex(CPRI 6)	5	40	72036	72036.007	72044.652	72053.297	72054

Appendix IX

Overview of CBR clients into LO OPU mapping types

(This appendix does not form an integral part of this Recommendation.)

As there are many different constant bit rate client signals and multiple mapping procedures, Table IX.1 provides an overview of the mapping procedure that is specified for each client.

Table IX.1 – Overview of CBR client into LO OPU mapping types

	OPU0	**OPU1**	**OPU2**	**OPU2e**	**OPU3**	**OPU4**	**OPUflex**
STM-1	GMP with C_{1D}	–	–	–	–	–	–
STM-4	GMP with C_{1D}	–	–	–	–	–	–
STM-16	–	AMP, BMP	–	–	–	–	–
STM-64	–	–	AMP, BMP	–	–	–	–
STM-256	–	–	–	–	AMP, BMP	–	–
1000BASE-X	TTT+GMP no C_{nD}	–	–	–	–	–	–
10GBASE-R	–	–	–	16FS+BMP	–	–	–
40GBASE-R	–	–	–	–	TTT+GMP with C_{8D}	–	–
100GBASE-R	–	–	–	–	–	GMP with C_{8D}	–
FC-100	GMP no C_{nD}	–	–	–	–	–	–
FC-200	–	GMP with C_{8D}	–	–	–	–	–
FC-400	–	–	–	–	–	–	BMP
FC-800	–	–	–	–	–	–	BMP
FC-1200	–	–	–	TTT+16FS+ BMP (Note)	–	–	–
CPRI option 1	GMP TBD C_{1D}	–	–	–	–	–	–
CPRI option 2	GMP TBD C_{1D}	–	–	–	–	–	–
CPRI option 3	–	GMP TBD C_{1D}	–	–	–	–	–
CPRI option 4	–	–	–	–	–	–	BMP
CPRI option 5	–	–	–	–	–	–	BMP
CPRI option 6	–	–	–	–	–	–	BMP

Table IX.1 – Overview of CBR client into LO OPU mapping types

	OPU0	OPU1	OPU2	OPU2e	OPU3	OPU4	OPUflex
CM_GPON	–	AMP	–	–	–	–	–
CM_XGPON	–	–	AMP	–	–	–	–
IB SDR	–	–	–	–	–	–	BMP
IB DDR	–	–	–	–	–	–	BMP
IB QDR	–	–	–	–	–	–	BMP
SBCON/ESCON	GMP no C_{nD}	–	–	–	–	–	–
DVB_ASI	GMP no C_{nD}	–	–	–	–	–	–
SDI	GMP TBD C_{nD}	–	–	–	–	–	–
1.5G SDI	–	GMP TBD C_{nD}	–	–	–	–	–
3G SDI	–	–	–	–	–	–	BMP
NOTE – For this specific case the mapping used is byte synchronous.							

Appendix X

Overview of LO ODU into HO OPU mapping types

(This appendix does not form an integral part of this Recommendation.)

As there are many different LO ODU bit rate signals and multiple mapping procedures, Table X.1 provides an overview of the mapping procedure that is specified for each LO ODU.

Table X.1 – Overview of LO ODU client into HO OPU mapping types

	2.5G tributary slots		**1.25G tributary slots**			
	OPU2	**OPU3**	**OPU1**	**OPU2**	**OPU3**	**OPU4**
ODU0	–	–	ODTU01 AMP (PT=20)	ODTU2.1 GMP (PT=21)	ODTU3.1 GMP (PT=21)	ODTU4.1 GMP (PT=21)
ODU1	ODTU12 AMP (PT=20)	ODTU13 AMP (PT=20)	–	ODTU12 AMP (PT=21)	ODTU13 AMP (PT=21)	ODTU4.2 GMP (PT=21)
ODU2	–	ODTU23 AMP (PT=20)	–	–	ODTU23 AMP (PT=21)	ODTU4.8 GMP (PT=21)
ODU2e	–	–	–	–	ODTU3.9 GMP (PT=21)	ODTU4.8 GMP (PT=21)
ODU3	–	–	–	–	–	ODTU4.31 GMP (PT=21)
ODUflex	–	–	–	ODTU2.ts GMP (PT=21)	ODTU3.ts GMP (PT=21)	ODTU4.ts GMP (PT=21)
ODUflex(IB SDR)	–	–	–	ODTU2.3 GMP (PT=21)	ODTU3.3 GMP (PT=21)	ODTU4.2 GMP (PT=21)
ODUflex(IB DDR)	–	–	–	ODTU2.5 GMP (PT=21)	ODTU3.5 GMP (PT=21)	ODTU4.4 GMP (PT=21)
ODUflex(IB QDR)	–	–	–	–	ODTU3.9 GMP (PT=21)	ODTU4.8 GMP (PT=21)
ODUflex(FC-400)	–	–	–	ODTU2.4 GMP (PT=21)	ODTU3.4 GMP (PT=21)	ODTU4.4 GMP (PT=21)
ODUflex(FC-800)	–	–	–	ODTU2.7 GMP (PT=21)	ODTU3.7 GMP (PT=21)	ODTU4.7 GMP (PT=21)

Table X.1 – Overview of LO ODU client into HO OPU mapping types

	2.5G tributary slots		1.25G tributary slots			
	OPU2	OPU3	OPU1	OPU2	OPU3	OPU4
ODUflex(CPRI option 4)	–	–	–	ODTU2.3 GMP (PT=21)	ODTU3.3 GMP (PT=21)	ODTU4.3 GMP (PT=21)
ODUflex(CPRI option 5)	–	–	–	ODTU2.4 GMP (PT=21)	ODTU3.4 GMP (PT=21)	ODTU4.4 GMP (PT=21)
ODUflex(CPRI option 6)	–	–	–	ODTU2.5 GMP (PT=21)	ODTU3.5 GMP (PT=21)	ODTU4.5 GMP (PT=21)
ODUflex(GFP), n=1, … ,8 (ts=n)	–	–	–	ODTU2.ts (GMP) (PT=21)	ODTU3.ts (GMP) (PT=21)	ODTU4.ts (GMP) (PT=21)
ODUflex(GFP), n=9, … ,32 (ts=n)	–	–	–	–	ODTU3.ts (GMP) (PT=21)	ODTU4.ts (GMP) (PT=21)
ODUflex(GFP), n=33, … ,80 (ts=n)	–	–	–	–	–	ODTU4.ts (GMP) (PT=21)

Appendix XI

Derivation of recommended ODUflex(GFP) bit-rates and examples of ODUflex(GFP) clock generation

(This appendix does not form an integral part of this Recommendation.)

XI.1 Introduction

The recommended bit-rates for ODUflex(GFP) are provided in Table 7-8. While in principle an ODUflex(GFP) may be of any bit-rate, there are a variety of reasons for recommending particular rates:

- To encourage a common set of bit-rates which can be expected to be supported by multiple manufacturers.
- To provide the largest amount of bandwidth possible within a given amount of resources (number of tributary slots) independent of the HO ODUk over which the ODUflex(GFP) may be routed.
- To maintain the number of tributary slots required if the ODUflex(GFP) must be rerouted, e.g., during a restoration.
- To satisfy a protocol requirement for ODUflex hitless resizing that a resizable ODUflex must occupy the same number of tributary slots on every HO ODUk path over which it is carried, and that a resize operation must always add or remove at least one tributary slot.

XI.2 Tributary slot sizes

ODUflex(GFP) is mapped via GMP into a certain number of 1.25G tributary slots of an HO OPU2, OPU3, or OPU4. Each of these have different tributary slot sizes:

$$OPU2\_TS = \frac{238}{237} \times 4 \times STM16 \times \frac{476\,columns}{3808\,columns} = 1249409.620 \text{ kbit/s} \pm 20\text{ppm}$$

$$OPU3\_TS = \frac{238}{236} \times 16 \times STM16 \times \frac{119\,columns}{3808\,columns} = 1254703.729 \text{ kbit/s} \pm 20\text{ppm}$$

$$OPU4\_TS = \frac{238}{227} \times 40 \times STM16 \times \frac{47.5\,columns}{3808\,columns} = 1301709.251 \text{ kbit/s} \pm 20\text{ppm}$$

An ODUflex(GFP) that occupies 8 or fewer tributary slots may be routed over HO OPU2, OPU3, or OPU4. The smallest tributary slot that may be encountered along the route of the ODUflex(GFP) is that of HO OPU2. Even if the initially selected route does not choose a link of HO OPU2, the ODUflex(GFP) should be sized to a multiple of the OPU2 tributary slot size to preserve the possibility to restore the ODUflex(GFP) over a route that includes HO OPU2 without changing the size of the ODUflex or the number of tributary slots it occupies.

An ODUflex(GFP) that occupies at least 9 but no more than 32 tributary slots may be routed over HO OPU3 or OPU4. It does not fit over HO OPU2. Therefore such an ODUflex may be sized to a multiple of the OPU3 tributary slot size. Even if the initially selected route does not choose a link of HO OPU3, the ODUflex(GFP) should be sized to a multiple of the OPU3 tributary slot size to preserve the possibility to restore the ODUflex(GFP) over a route that includes HO OPU3 without changing the size of the ODUflex or the number of tributary slots it occupies.

An ODUflex(GFP) that occupies at least 33 but no more than 80 tributary slots may only be carried via HO OPU4, and may therefore take advantage of the full size of the OPU4 tributary slot size.

A small margin must be left between the ODUflex(GFP) size and the integral multiple of the tributary slot size to accommodate possible clock variation along a sequence of HO OPUk links without overflowing the range of C_m in the GMP mapper.

Physical layers for data interfaces such as Ethernet and fibre channel have historically used a clock tolerance of ±100 ppm. This range is sufficiently wide that specifying this as the clock tolerance for ODUflex(GFP) can accommodate a variety of mechanisms for generating an ODUflex(GFP) clock and remain within the clock tolerance range.

ODUk.ts as shown in Table 7-8 is an increment of bandwidth which when multiplied by a number of tributary slots, gives the recommended size of an ODUflex(GFP) optimized to occupy a given number of tributary slots of a higher order OPUk. These values are chosen to allow a sufficient margin that allows the HO OPUk and the ODUflex(GFP) to independently vary over their full clock tolerance range without exceeding the capacity of the allocated tributary slots.

The nominal values for ODUk.ts are chosen to be 186 ppm below the bandwidth of a single 1.25G tributary slot of a higher order OPUk. This allows the ODUflex(GFP) clock to be as much as 100 ppm above its nominal rate and the higher order OPUk to be as much as 20 ppm below its nominal clock rate, allowing approximately 66 ppm of margin to accommodate jitter and to ensure that the largest average C_m value even in the worst-case situation of the HO OPUk at -20 ppm from its nominal value and the ODUflex(GFP) at +100 ppm from its nominal value will be one less than the maximum value (i.e., the maximum average C_m is no more than 15231 out of 15232 for ODUflex carried over OPU2 or OPU3, and no more than 15199 out of 15200 for ODUflex carried over OPU4).

XI.3 Example methods for ODUflex(GFP) clock generation

XI.3.1 Generating an ODUflex(GFP) clock from a higher order OPUk clock

The clock for an ODUflex(GFP) may be generated from the initial higher order OPUk over which the ODUflex is carried by setting the value of C_m to a fixed value on the initial segment. Normal GMP processing on subsequent segments avoids the need to couple the higher order OPUk clocks along the path.

Table XI.1 – Generation of ODUflex(GFP) clock from higher order OPUk clock using fixed C_m

Higher order OPUk	Nominal bit rate	Nominal 1.25G TS bit rate	ODU2.ts		ODU3.ts		ODU4.ts	
			C_m out of	Bit-rate per TS	C_m out of	Bit-rate per TS	C_m out of	Bit-rate per TS
OPU2	9'995'276.962	1'249'409.620	15230 15232	1'249'245.570				
			+20 ppm	1'249'270.555				
			-20 ppm	1'249'220.585				
OPU3	40'150'519.322	1'254'703.729	15165 15232	1'249'184.746	15230 15232	1'254'538.983		
			+20 ppm	1'249'209.729		1'254'564.074		
			-20 ppm	1'249'159.762		1'254'513.892		
OPU4	104'355'975.330	1'301'709.251	14587 15200	1'249'212.687	14649 15200	1'254'522.291	15198 15200	1'301'537.974
			+20 ppm	1'249'237.671		1'254'547.381		1'301'564.004
			-20 ppm	1'249'187.703		1'254'497.200		1'301'511.943
		ODUk.ts	nominal	1'249'177.230		1'254'470.354		1'301'467.133
			+100 ppm	1'249'302.148		1'254'595.801		1'301'597.280
			-100 ppm	1'249'052.312		1'254'344.907		1'301'336.986

Table XI.1 illustrates how a clock for an ODUflex(GFP) occupying n × ODUk.ts can be derived from the higher order OPUk clock using a fixed value of C_m in the initial segment of the path.

For example, for an ODUflex(GFP) occupying up to 8 tributary slots should be based on ODU2.ts, and therefore have a clock frequency of n × 1'249'177.230 kbit/s ±100 ppm. This allows the ODUflex(GFP) to have a frequency of between n × 1'249'052.312 kbit/s and n × 1'249'302.148 kbit/s.

- If the initial segment over which the ODUflex(GFP) is carried is an OPU2, a clock in this range can be generated by fixing the value of C_m on the initial segment to 15230, which will result in the ODUflex having a clock of n × 1'249'245.570 kbit/s ±20 ppm. While the centre frequency of this range differs from the nominal value of ODU2.ts, the clock tolerance is narrower, being locked to the higher order OPU2, so the possible clock range is fully within the ±100 ppm range allowed.
- If the initial segment is a higher order OPU3, the ODUflex(GFP) of a multiple of ODU2.ts can be generated using a fixed value of C_m=15165 on the initial ODU3 segment, which will result in the ODUflex having a clock of n × 1'249'184.746 kbit/s ±20 ppm.
- If the initial segment is a higher order OPU4, the ODUflex(GFP) of a multiple of ODU2.ts can be generated using a fixed value of C_m=14587 on the initial OPU4 segment, which will result in the ODUflex having a clock of n × 1'249'212.687 kbit/s ±20 ppm.

The centre frequencies of all of these ODUflex(GFP) are slightly different but the resulting ranges for the clocks all fall within the ±100 ppm window (see Figure XI.1). Fixed C_m for generating ODU3.ts and ODU4.ts from the initial higher order OPUk can similarly be found from this table.

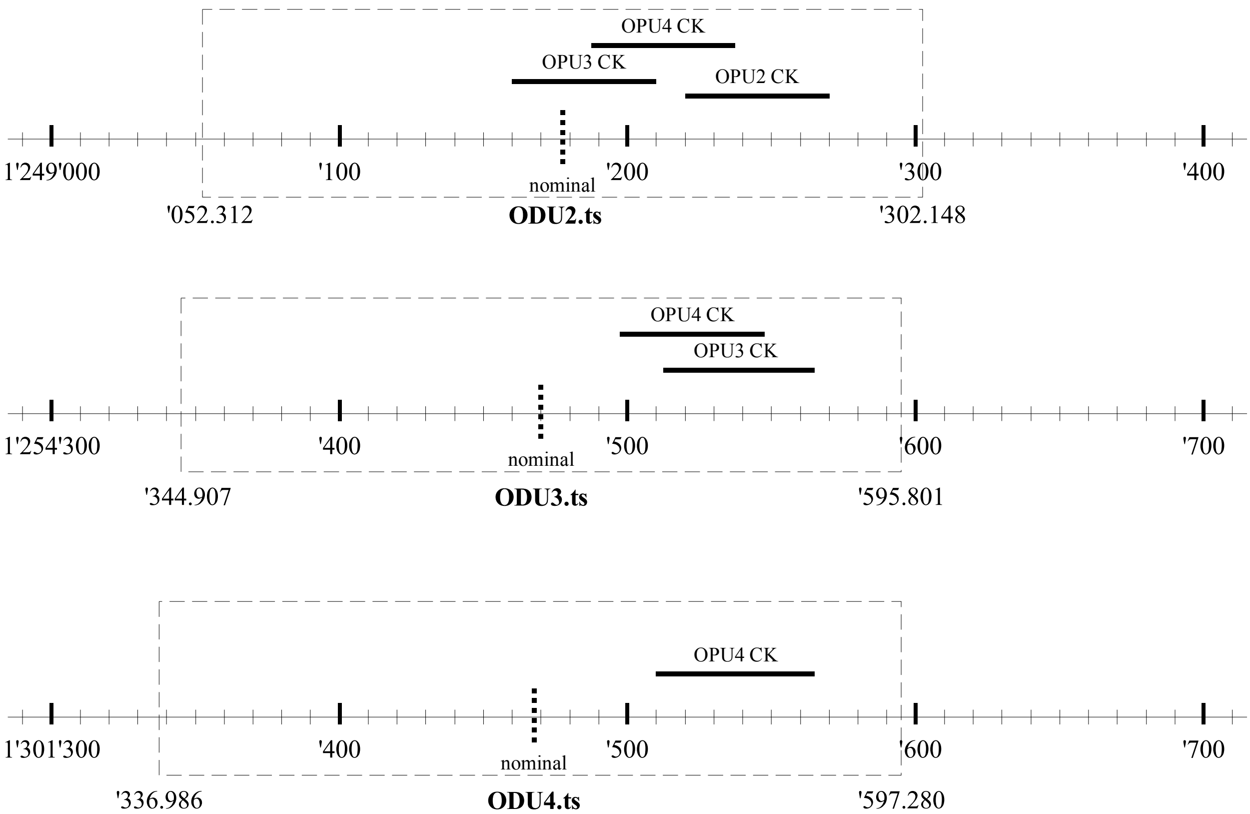


Figure XI.1 – Graphical representation of frequency ranges in Table XI.1

To ensure that this method is future proof, likely rates for future OPU5, OPU6, and OPU7 have been checked to ensure that it is possible to select a fixed C_m to generate ODUflex(GFP) clocks based on any ODUk.ts value. As future tiers of the hierarchy are yet to be agreed it would be premature to list them here, but the following reasoning ensures that this mechanism can be extended to future tiers: based on the M-byte mechanism for GMP mapping into tributary slots, each increment of fixed C_m represents a 65-66 ppm difference in the resulting ODUflex frequency. There will generally be three (exceptionally four) values of C_m for which, if the higher order OPUk is running at nominal frequency, would generate an ODUflex clock that falls within a ±100 ppm window. At least one of these possible values of C_m is 67 ppm or more from each end of the ±100 ppm range. Since the actual variation of the clock for an ODUflex whose clock is generated in this manner is only ±20 ppm and the higher order OPUk for downstream segments can also vary by ±20 ppm, at least 40 ppm difference is needed between the centre frequency of an ODUflex(GFP) generated from a future OPUk (k>4) and each end of the ±100 ppm range. Since only 40 ppm is required and at least 67 ppm are available, it will be possible to select fixed C_m values to generate ODUflex(GFP) clocks from future higher order OPUk.

XI.3.2 Generating an ODUflex(GFP) clock from a system clock

The clock for an ODUflex(GFP) may be generated using a multiplier from the internal system clock. Normally the internal system clock will have an accuracy of at least ±20 ppm, perhaps even ±4.6 ppm for a network element that supports both SDH and OTN interfaces. The exact multiplier to be used is implementation specific, and chosen so that the range of the generated clock falls within the specified ±100 ppm window around the nominal value of $n \times$ ODUk.ts.

Bibliography

[b-ITU-T G-Sup.43]	ITU-T G-series Recommendations – Supplement 43 (2008), *Transport of IEEE 10GBASE-R in optical transport networks (OTN).*
[b-ANSI INCITS 296]	ANSI INCITS 296-1997, *Information Technology – Single-Byte Command Code Sets CONnection (SBCON) Architecture).*
[b-ANSI INCITS 364]	ANSI INCITS 364-2003, *Information Technology – Fibre Channel 10 Gigabit (10GFC).*
[b-IB ARCH]	InfiniBand Trade Association (2006), *InfiniBand Architecture Specification, Volume 2, Release 1.2.1*.
[b-INCITS 470]	INCITS 470:2011, *Information Technology – Fibre Channel – Framing and Signaling – 3 (FC-FS-3).*
[b-CPRI]	CPRI Specification V5.0 (2011-09-21) Common Public Radio Interface (CPRI), *Interface Specification*.

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	**Y.1300–Y.1399**
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Smart ubiquitous networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
Future networks	Y.3000–Y.3099

For further details, please refer to the list of ITU-T Recommendations.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2012