INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.998.3

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(01/2005)

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

Digital sections and digital line system – Access networks

Multi-Pair Bonding Using Time-Division Inverse Multiplexing

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. 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/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

ITU-T Recommendation G.998.3

MULTI-PAIR BONDING USING TIME-DIVISION INVERSE MULTIPLEXING

Summary

This Recommendation describes a method for bonding of multiple digital subscriber lines (DSL) using Time-Division Inverse Multiplexing (TDIM). This Recommendation provides a specification of the TDIM protocol in sufficient detail, to allow development and testing of interoperable implementations for both transmitter and receiver. It includes a Multi-pair synchronization frame format, Bonding Communication Channel (BCC), Byte oriented Dispatching, Hitless addition and removal of pairs, Fast removal of pair upon pair failure, Using IEEE 802.3ah handshake for pair discovery, parameter negotiation and setup, and an Optional FEC and Interleaver

1 Scope

This Recommendation defines the bonding function for the TDIM based bonding method, the purpose of which is to provide inverse multiplexing of various service data streams (Ethernet, ATM, TDM) over multiple DSL physical links and to retrieve the original stream at the far-end from these physical links.

This Recommendation is a detailed specification of the TDIM protocol in sufficient detail to allow development and testing of interoperable implementations for both transmitter and receivers. It includes:

1. Multi-pair synchronization frame format.
2. Bonding Communication Channel (BCC).
3. Dispatching algorithm
4. Hitless addition and removal of pairs.
5. Fast removal of pair upon pair failure.
6. Using IEEE 802.3ah handshake for pair discovery, parameter negotiation and setup

This Recommendation defines a new TPS-TC for DSL transceivers. Architecturally, this TPS-TC should be placed above the PMS-TC (at the alpha/beta-interface) of existing or future DSL transceivers. Practically, the exact same result can be obtained by stacking the new TPS-TC defined in this Recommendation on top of the Clear Channel or STM TPS-TC as defined in existing DSL Recommendations.

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

[1] ITU Recommendation G.704 (1998), *Synchronous Frame Structures used at 1544, 6312, 2048, 8488 and 44736 kbit/s Hierarchical Levels*

[2] ITU-T Recommendation G.7041/Y.1303 (2004), *Generic Framing Procedure (GFP)*

[3] ITU-T Recommendation G.991.2 (2003), *Single-pair high-speed digital subscriber line (SHDSL) transceivers.*

[4] ITU-T Recommendation G.992.1 (1999), *Asymmetric Digital Subscriber Line (ADSL) Transceivers.*

[5] ITU-T Recommendation G.992.3 (2002), *Asymmetric Digital Subscriber Line (ADSL) Transceivers – 2 (ADSL2)*

[6] ITU-T Recommendation G.993.1 (2004), *Very-high-speed Digital Subscriber Line Transceivers (VDSL)*

[7] ITU-T Recommendation G.994.1 (2003), *Handshake procedures for Digital Subscriber Line (DSL) Transceivers.*

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

[9] IEEE 802.3ah (2004), *Amendment to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specification*

3 Definitions

There are no additional definitions required in this specification.

4 Abbreviations, Acronyms, and Symbols

ADSL	Asymmetric Digital Subscriber Line
ANSI	American National Standards Institute
ATIS	Alliance for Telecommunications Industry Solutions
ATM	Asynchronous Transfer Mode
BCC	Bonding Communication Channel
bps	bits per second
BTU-C	Bonding Terminating Unit, CO side
BTU-R	Bonding Terminating Unit, RT (or CPE) side
BW	Bandwidth
Clk	Clock
CPE	Customer Premise Equipment
CO	Central Office
CRC	Cyclic Redundancy Check
DS1	Digital Signal 1, (1.544 Mbps)

DS3	Digital Signal 3, (44.736 Mbps)
DSL	Digital Subscriber Line
E1	Electrical interface signal, Level 1, (2.048 Mbps)
E3	Electrical interface signal, Level 3, (34.368 Mbps)
EFM	Ethernet in the First Mile
enum	Enumerated
EOC	Embedded Operations Channel
FCS	Frame Check Sequence
FE	Far End
FEC	Forward Error Correction
GF	Galois Field
GFP	Generic Framing Procedure
HEC	Header Error Check
HS	Handshake
IEEE	The Institute of Electrical and Electronics Engineers
IL	Interleaver
IMA	Inverse Multiplexing for ATM
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
kbps	kilobits per second
LSB	Least Significant Bit / Byte
mbps	megabits per second
MSB	Most Significant Bit / Byte
MUX	Multiplexer
NE	Near End
NS	Number of Services
PLI	Packet Length Identifier
PM	Performance Monitor
PMI	Physical Medium Independent
PSD	Power Spectral Density
RS	Reed Solomon
RT	Remote Terminal
rx	Receive/receiver
SDH	Synchronous Digital Hierarchy
SF	Super Frame
SHDSL	Single-Pair High-Speed Digital Subscriber Line

STM	Synchronous Transfer Mode
TC	Transmission Convergence
TDIM	Time Division Inverse Multiplexing
TDM	Time Division Multiplexing
TPS-TC	Transport Protocol Specific - Transmission Convergence
Tx	Transmit/transmitter
uint 8	Unsigned Integer 8 bits
uint 16	Unsigned Integer 16 bits
UTC	Unable To Comply
VDSL	Very high speed Digital Subscriber Line
µsec	microsecond

5 Data flow

The following figure represents the data flow model which is defined in this proposed text. Data from a mix of services is encapsulated into a single data stream. The data stream can than optionally pass through Forward Error Correction and Interleaving, and then be dispatched (Inverse Multiplexed) over multi-pair modems.

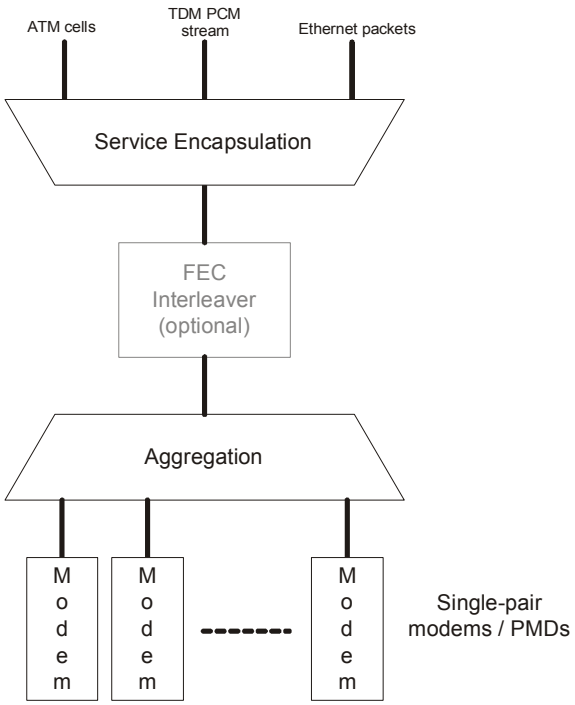

Figure 1-Data flow model

The data flow model using Modem Rate Matching is illustrated in Annex A.

6 Multi-pair Synchronization

6.1 Introduction

The multi-pair Aggregation Group synchronization is done in the Aggregation layer, independently from the underlying physical layer. Time Division Inverse Multiplexing (TDIM) bonding uses a Super Frame format for multi-pair synchronization, as described below.

6.2 Frame format

The multi-pair synchronization frame, called Super-Frame, is made of smaller entities. The following figure describes the frame format for the Super-Frame. Note that wherever applicable, Most Significant Bit / Byte (MSB) bits are always transmitted first in time and Least Significant Bit / Byte (LSB) bits always transmitted last.

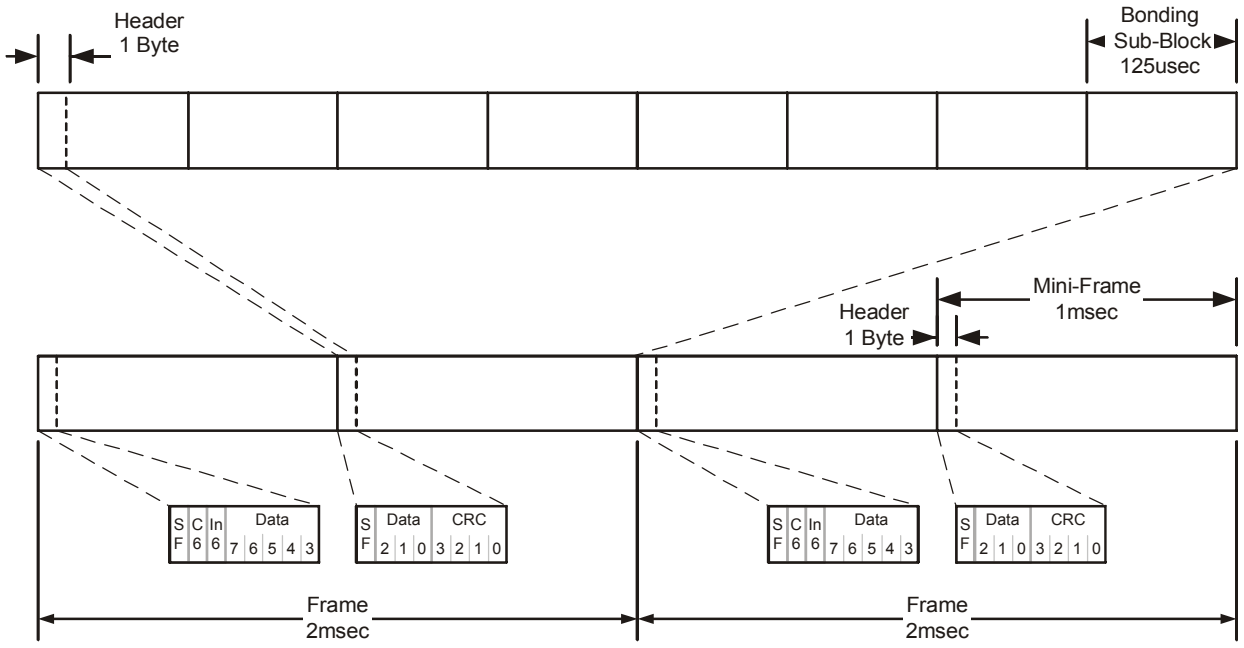

Figure 2-Multi-pair synchronization – frame format

6.2.1 Bonding Sub-Block and Mini-Frames

The Bonding Sub-Block has a fixed time period of 125 µsec. A Sub-Block duration of 125 µsec yields a rate granularity of 8 kbps.

A Mini-Frame contains 8 Bonding Sub-Blocks and has a period of 1 msec. The start of a Mini-Frame is synchronized to the start of a Bonding Sub-Block. The first byte in each Mini-Frame (in each DSL pair) is taken for header. The Mini-Frame header is used to indicate the start of a Super-Frame and to transferring events and messages to the Far End.

The amount of bits in each Sub-Block is a function of the Sub-Block location, the number of bonded pairs and on the total data rate of all the bonded pairs.

The amount of bonding payload bits carried by each of the last 7 Bonding Sub-Blocks of every Mini-Frame is equal to *N(Rate) = Rate/8 kbps*, while:

o *N(Rate)* is the number of bits in each sub-block, depending of the accumulated link data rate.

o *Rate* is the accumulated data rate of all the pairs measured in bits-per-second [bps].

The amount of bonding payload bits carried by the first Bonding Sub-Block in each Mini-Frame is N(Rate)-8M where M is the number of individual pairs in the bonded system.

Note: If rate matching is used (see Annex A), then:

The amount of bits available for bonding payload in the first Bonding Sub-Block of every Min-Frame is *N(Rate)-16M*.

The amount of bits available for bonding payload in the 8-th Bonding Sub-Block of every Mini-Frame is $N(Rate)+8\Delta$, where $\Delta=\Sigma_{i=1..M}\,\Delta i$, and each Δ_i is dependent on the rate variation on pair *i* and can take the values *–1,0,1,2*. Moreover, each Δ_i may take different values in each Mini-Frame (see Annex A for details).

6.2.2 Frame

A Frame is built from 2 Mini-Frames and therefore has a period of 2 msec. There are 16 bits of header in each Frame , one byte in each Mini-Frame. The format of the frame header is described in Figure 3.

Each Frame header has 6 expected bits that are known in advance (Super Frame (SF) bits, Cyclic Redundancy Check (CRC) bits). Error in these bits is used to recognize a pair failure.

10 (ten) consecutive frames with CRC errors or wrong SF bits on a specific pair shall be used to declare this pair faulty.

The following figure defines the content of the Frame header fields:

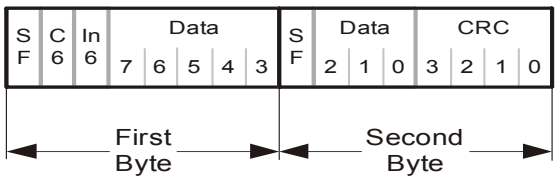

Figure 3- Frame header format

- SF (Super-Frame indication) – indicates the start of a Super-Frame. This bit is located in the start of each Mini-Frame. It shall be set to '1' at the first Mini-Frame in the Super-Frame and otherwise set to '0'.
- C6 (Cyclic Redundancy Check code) – the Super-Frame CRC-6 field is made of six C6 bits C6[5:0] which are part of each Frame. The CRC-6 field is calculated over all the multi-pair data that was transmitted during the previous Super-Frame (excluding the Frame header overhead, and the Modem Rate Matching overhead – if used), and is used to detect errors in

the data in a specific Super-Frame. The encoding of the six bits is defined by the generating polynomial $G(x) = x^6 + x + 1$, (see International Telecommunication Union . Telecommunication Standardization Sector (ITU-T) Recommendation G.991.2, "Single-pair High Speed Digital Subscriber Line (SHDSL) transceivers"). Mathematically, the CRC value corresponding to a given Super-Frame is defined by the following procedure:

– The first 6 bits of the Super-Frame are complemented.

– The N bits of the Super-Frame are then considered to be the coefficients of a polynomial M(x) of degree N–1. (The first bit of the Super-Frame corresponds to the $x^{(N-1)}$ term and the last bit of the Super-Frame corresponds to the x^0 term).

– M(x) is multiplied by x^6 and divided by G(x), producing a remainder R(x) of degree 5.

– The coefficients of R(x) are considered to be a 6-bit sequence.

– The bit sequence is complemented and the result is the CRC.

– The 6 bits of the CRC value are placed in the "C6" bits in the following Super-Frame overhead so that the x^5 term (C6[5]) is located in the first frame of the Super-Frame, and the x^0 term (C6[0]) is located in the last frame of the Super-Frame. The bits of the CRC are thus transmitted in the order x^5, x^4, … x^1, x^0.

- In6 – six bits, one per frame, that make a SF field of six indication bits In6[5:0].
  - The indication bit In6[5] which is transmitted in the first frame of the SF is called M/E and is used for Message / Event indication. It indicates the meaning of the Data[7:0] bits. M/E = '0' indicates that the Data[7:0] bits are part of an event, M/E = '1' indicates that the Data[7:0] bits are part of a message.
  - The indication bit In6[4] has different meaning that depends whether it is transmitted by the Bonding Terminating Unit, Central Office (CO) side (BTU-C) or the Bonding Terminating Unit, Remote Terminal (RT) side (BTU-R):
    – BTU-R: In6[4] = '1' indicates the BTU-R aggregation layer is capable of operating in the regular aggregation mode, without the use of the Modem Rate Matching mechanism .
    – BTU-C: In6[4] = '1' orders the BTU-R to operate in the regular aggregation mode without the modem rate matching.
  - The indication bit In6[3] has different meaning that depends whether it is transmitted by the BTU-C or the BTU-R:
    – BTU-R: In6[3] = '1' indicates the BTU-R aggregation layer is capable of operating with the use of the Modem Rate Matching mechanism.
    – BTU-C: In6[3] = '1' orders the BTU-R to use the Modem Rate Matching mechanism.
  - The indication bits In6[2:0] are reserved for future use, and currently set to '1'.
- Data [7:0] – Byte of data that makes a part of an event or a message (see 12.3), according to the M/E bit

- CRC [3:0] (Cyclic Redundancy Check code) – four bits assigned to a CRC code. A CRC-4 (see ITU-T G.704 clause 2.3.3.5) shall be generated for each frame header (not including the CRC-4 field) and transmitted on the CRC field. The encoding of the four bits is defined by the generating polynomial $G(x) = x^4 + x + 1$. Mathematically, the CRC-4 value corresponding to a given frame header is defined by the following procedure:
  - The first 4 bits of the frame header are complemented.
  - The 12 bits of the frame header are then considered to be the coefficients of a polynomial M(x) of degree 11. The first bit of the frame header corresponds to the x^{11} term and the last bit of the frame header corresponds to the x^0 term.
  - M(x) is multiplied by x^4 and divided by G(x), producing a remainder R(x) of degree 3.
  - The coefficients of R(x) are considered to be a 4-bit sequence.
  - The bit sequence is complemented and the result is the CRC field.
  - The 4 bits of the CRC value are placed in the CRC field of the frame header so that the x^3 term is located in the left most bit of the CRC field, and the x^0 term is located in the right most bit of the CRC field. The bits of the CRC are thus transmitted in the order x^3, x^2, x^1, x^0.

6.2.3 Super-Frame (SF)

A Super-Frame is built from 6 Frames and has a period of 12 msec.

The Super-Frame is used in order to synchronize all the bonded pairs, and to frame Bonding Communications Channel (BCC) events and messages.

The Super-Frame has a CRC-6 field which is made of the six C6 bits which are part of each Frame. The CRC-6 field is calculated over all the multi-pair data that was transmitted during the previous Super-Frame, and is used to detect errors in the data in a specific Super-Frame. This information is used for performance monitoring of the bonded pairs.

6.3 Multi-pair Synchronization State Machine

The following figure describes the Multi-pair Synchronization State Machine, with the states and the transitions between the states. The states and transition conditions are defined below.

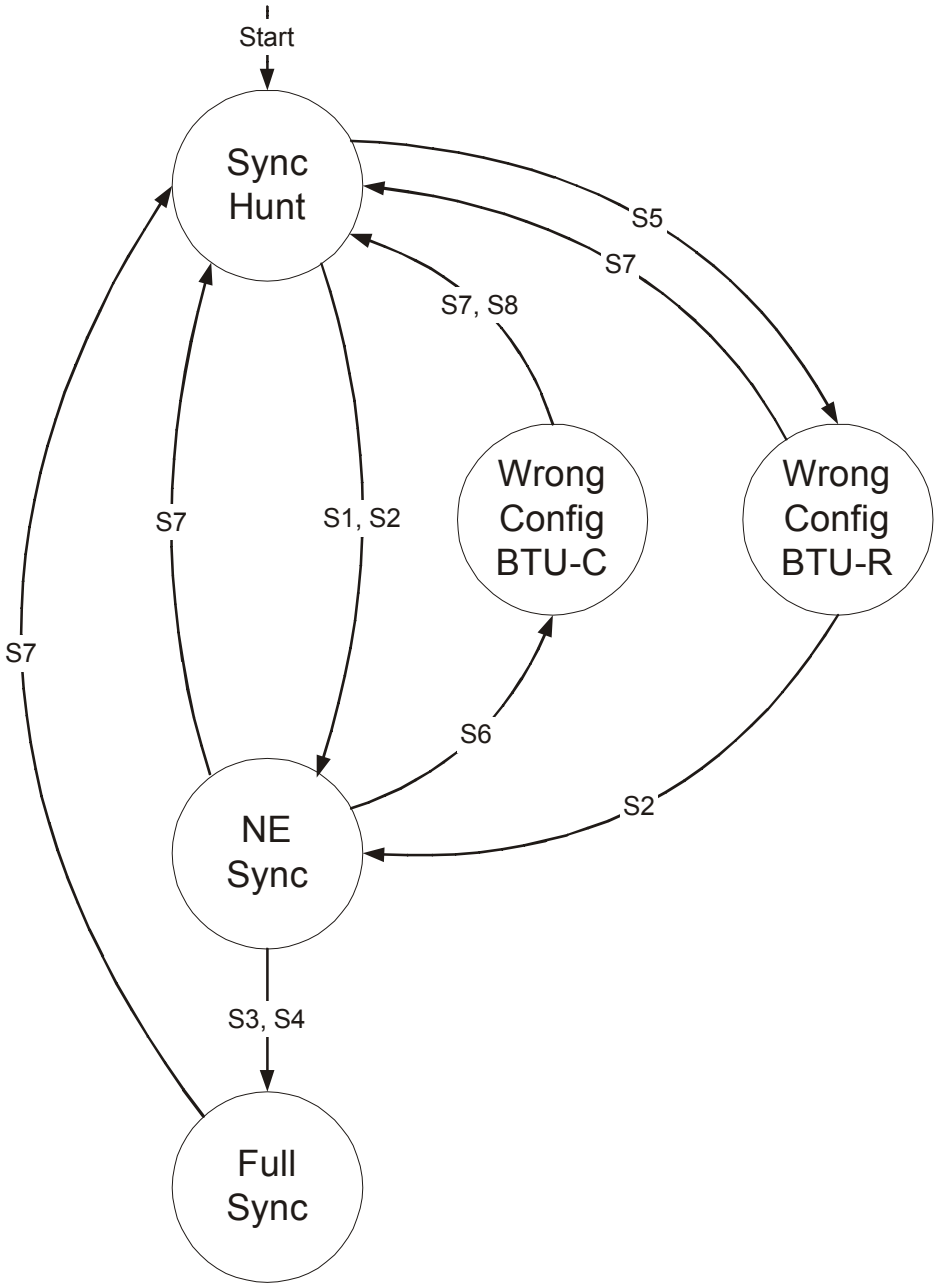

Figure 4- Multi-pair Synchronization State Machine

6.3.1 States

Table 1- Multi-pair Synchronization State Machine states definition

State	Definition	Receiver Actions	Transmitter Actions	Transitions
Sync Hunt	The pair is synchronizing by searching for the Aggregation Group framing, decoding the evSync events, and compensating for delay difference to the other pairs in the Aggregation Group	Searching for Frame headers, decoding evSync events	Sending evSync events with: • Value[0] – indicating no synchronization (0x00) • Value[1] – pair number in the BTU-C and 0xFF in the BTU-R • Value[2] – Aggregation group number in the BTU-C and 0xFF in the BTU-R	• [S1], [S2] to "NE Sync" state • [S5] to "Wrong Config BTU-R" state
NE Sync	The Near End is synchronized to the Aggregation Group	BTU-C is waiting for the BTU-R to get to "NE sync" state (indicated by Value[0] = 0x01 in the received evSync event). BTU-R is waiting for the BTU-C to get to "Full Sync" state (indicated by receiving events other than evSync events)	Sending evSync events with: • Value[0] – indicating NE synchronization (0x01) • Value[1] – the pair number • Value[2] – the Aggregation Group number The BTU-R is taking the pair number and the Aggregation Group number from the events received from the BTU-C	• [S3],[S4] to "Full Sync" state • [S6] to "Wrong Config BTU-C" state • [S7] to "Sync Hunt" state

State	Definition	Receiver Actions	Transmitter Actions	Transitions
Wrong Config BTU-C	This state is for the BTU-C only. The BTU-R is not able to get the Aggregation Group number or the pair number	None	Sending evSync events with: • Value[0] – NE sync (0x01) • Value[1] – the pair number • Value[2] – the Aggregation Group number	• [S7], [S8] to "Sync Hunt" state
Wrong Config BTU-R	This state is for the BTU-R only. There is wrong configuration: The BTU-R is already connected to another Aggregation Group or the pair number assigned to this pair is already in use by the BTU-R	Waiting for evSync events with a different Aggregation Group number or pair number	Sending evSync events with: • Value[0] – Wrong Config (0x80 or 0x81) • Value[1] – the pair number • Value[2] – the Aggregation Group number The BTU-R is sending its own pair number and Aggregation Group number	• [S2] to "NE Sync" state • [S7] to "Sync Hunt" state
Full Sync	The pair is synchronized to the Aggregation Group at both ends. It can be added to the Aggregation Group by using "Sync Change" procedure	Receiving same messages or events as the other pairs in the Aggregation Group	Sending same messages or events as the other pairs in the Aggregation Group	• [S7] to "Sync Hunt" state

6.3.2 Transition conditions

[S1] In the BTU-C: Consecutive three Super-Frames decoded with the same evSync event without errors.

[S2] In the BTU-R: Consecutive three Super-Frames decoded with the same evSync event without errors, with legal Aggregation Group number (equal to the Aggregation Group number received by other synchronized pairs, if exist), and legal pair number (not used by another synchronized pair).

[S3] In the BTU-C: Receiving evSync event with "NE Sync" indication (Value[0] = 0x01) from the BTU-R.

[S4] In the BTU-R: Receiving valid Super-Frame that is not an evSync event.

[S5] In the BTU-R: Consecutive three Super-Frames were decoded with the same evSync event without errors, but with an illegal Aggregation Group number (different from the Aggregation Group number already decoded by the BTU-R from previous synchronized pairs) or with an illegal pair number (i.e. already used by previously synchronized pair).

[S6] In the BTU-C: Receiving evSync event with an indication from the BTU-R of "Wrong Config" (Value[0] = 0x80 or 0x81).

[S7] 10 (ten) consecutive Frames with errors (bad CRC-4 or wrong SF bits).

[S8] In the BTU-C: A Management decision to resynchronize this pair with a different Aggregation Group number and / or a different pair number.

7 Dispatching

7.1 Principles

The Dispatcher maps the data stream bits coming from the Encapsulation layer (or from the Forward Error Correction (FEC)/ Interleaver), which is virtually divided into "Bonding Sub-Blocks" (see Figure 2), to each of the pairs in the Aggregation Group.

7.2 Algorithm

The dispatching algorithm is recurring every Bonding Sub-Block (125 µsec). The dispatching algorithm goes over all of the bits contained in a Bonding Sub-Block, by order from the first to the last. The dispatching algorithm goes over all of the pairs by order of their logical number in the Aggregation Group, and assigns bits from the Bonding Sub-Block to the pairs. Each pair is assigned n_i ($n_i = Rate_i / 8$ kbps) bits, where the first bits in each Mini-Frame are used as header bits and not for data bits. NOTE: For bonding systems implementing Annex A, a slight variation from n_i may occur at the 8-th

Bonding Sub-Block of each Mini-Frame. For a detailed description see Annex A and the dispatching pseudo-code below.

7.3 Pseudo Code

The following pseudo-code defines the dispatching algorithm:

```
Sub_Block_Counter = 1
Repeat every 125 µsec  {
    Bonding_Sub_Block_Bit = 1
    For Pair_Number = 1:Number_of_Pairs  {
        If (Bonding_Sub_Block_Bit==8 and Rate_Matching_Flag==2 ){
        Read Δ_Pair_Number
        }
        else{
```

```
            Δ_Pair_Number=0;
            }

            Bit_Counter = 1
            For Bit_Counter = 1:  n_Pair_Number+8* Δ_Pair_Number) {
            If (Sub_Block_Counter==1 and Bit_Counter<=8*Rate_Matching_Flag) {
                Send_Header (Bit_Counter, Pair_Number)
            }
            else {
                Send_Data (Bonding_Sub_Block_Bit, Pair_Number)
                Bonding_Sub_Block_Bit++
            }
            }
        }
        Sub_Block_Counter++
        If (Sub_Block_Counter > 8) {
            Sub_Block_Counter = 1
        }
    }
```

where:

Sub_Block_Counter The position of this Sub-Block in the Mini-Frame.

Bonding_Sub_Block_Bit Index of the next bit from the Bonding Sub-Block that shall be sent.

Pair_Number Index of the current pair (its logical number in the Aggregation Group).

Number_of_Pairs The amount of pairs currently delivering data in this Aggregation Group.

Bit_Counter Counter of bits already sent on the current pair.

n_i The (base) number of bits assigned to pair *i*.

Δ_i The variation (in Bytes) from n_i (non zero only for Annex A implementations).

Rate_Matching_Flag 1 if Rate Matching is not used, 2 if Rate Matching is used.

Send_Header (k, j) A function that sends bit *k* of the Mini-Frame header to pair *j*.

Send_Data (k, j) A function that sends bit *k* of the Bonding Sub-Block to pair *j*.

8 Differential delay

The framing structure of the Aggregation layer is capable of tolerating a differential delay of 6 millisecond between the fastest pair and the slowest pair, however a TDIM bonding system is required to tolerate a differential delay of only 2 milliseconds.

9 Clock synchronization

9.1 Scope

This clause defines the clock synchronization requirements between the BTU-C and the BTU-R, in the relevant layers and time domains: Service (TDM), Aggregation, and modems.

Figure 5 below shows the time domains in the Bonding system and the clocks that synchronize the time domains.

The directions BTU-C to BTU-R and BTU-R to BTU-C are independent, therefore there are 4 independent clocks:

- Each Time Division Multiplexing (TDM) service clock is transmitted from the BTU-C to the BTU-R (1) over the Aggregation time domain clock using stuffing. The TDM service clock is reconstructed at the BTU-R from the Aggregation time domain clock by stuffing removal.
- The BTU-C Aggregation time domain clock (2) is transmitted over the modems in a plesiochronous manner. The BTU-R Aggregation time domain clock is reconstructed from the modems.
- The BTU-R Aggregation time domain clock (3) is transmitted over the modems in a plesiochronous manner. The BTU-C Aggregation time domain clock is reconstructed from the modems.
- Each TDM service clock is transmitted from the BTU-R to the BTU-C (4) over the Aggregation time domain clock using stuffing. The TDM service clock is reconstructed at the BTU-C from the Aggregation time domain clock by stuffing removal.

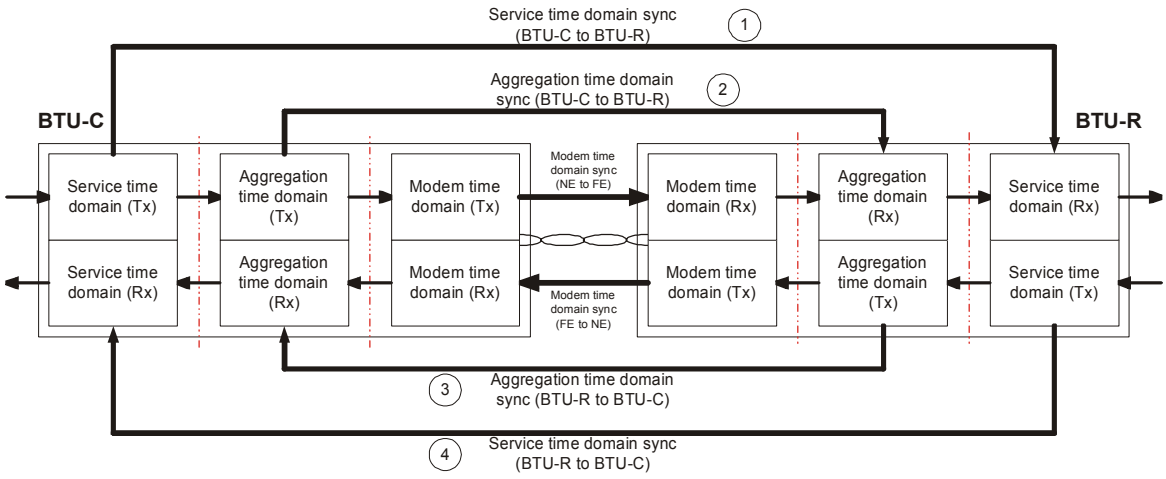

Figure 5- Peer time domains and clock synchronization

Figure 6 below shows a reference model for clock synchronization. It can be seen that the data / clock between the Aggregation layer and the modems are synchronized to the Aggregation time domain. However each modem may use a different line clock (i.e. the Modems time domain) and carry the data and clock from the Aggregation layer using stuffing (like in G.shdsl and in Asymmetric Digital Subscriber Line (ADSL)).

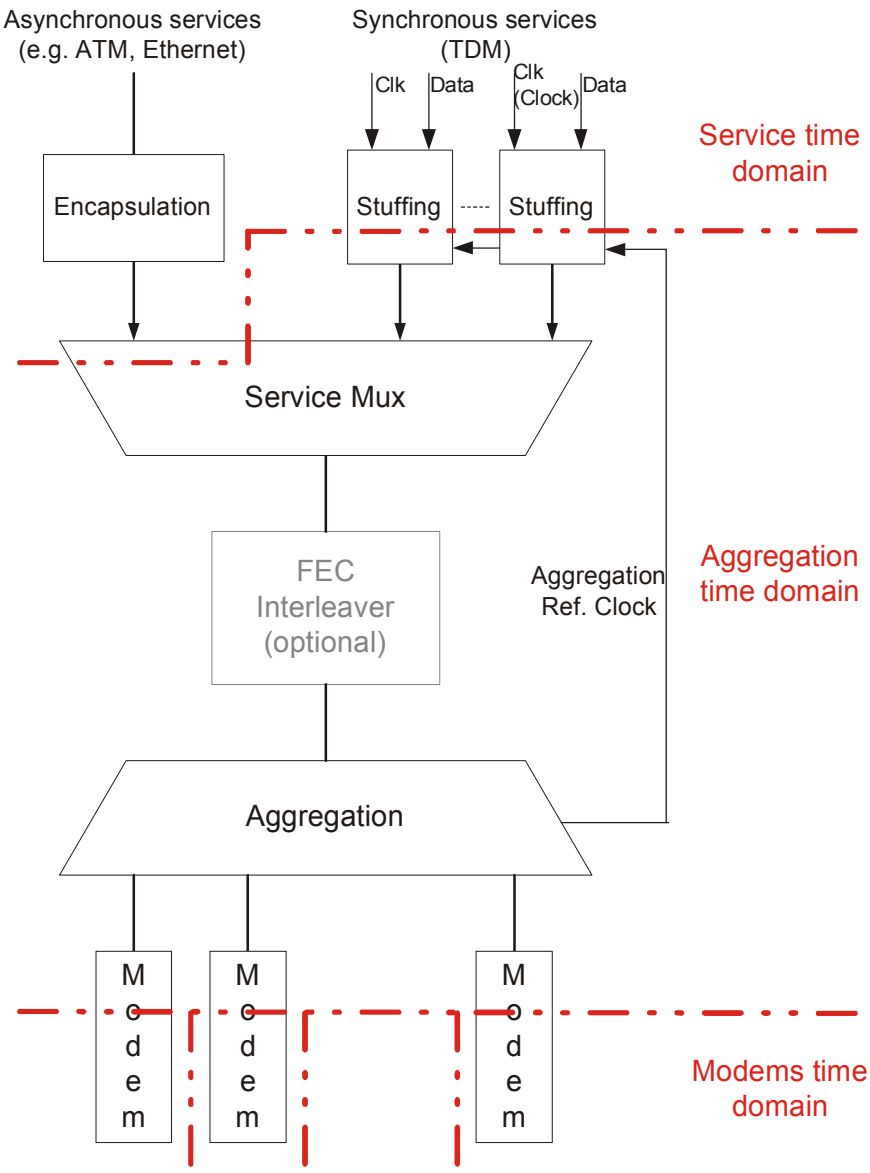

Figure 6- Clock synchronization reference model

Clock synchronization under the use of the Modem Rate Matching mechanism is described in Annex A.

9.2 TDM service synchronization and clock transfer

Each TDM service (e.g. Digital Signal 1, (1.544 Mbps) (DS1), Electrical interface signal, Level 1, (2.048 Mbps) (E1)) data and clock are transferred from the Near End (NE) to the Far End (FE). Plesiochronous services are transferred from the NE to the FE by adding stuffing at the NE and using stuffing removal at the FE. The stuffing is done relatively to the Aggregation reference clock, which synchronizes the Aggregation layers of BTU-C and the BTU-R. For more details see 10.4.

9.3 Aggregation layer synchronization

The Aggregation layer at the BTU-R receiver is locked and synchronized to the Aggregation layer at the BTU-C transmitter. The Aggregation layer at the BTU-C receiver is locked and synchronized to the Aggregation layer at the BTU-R transmitter.

The Aggregation layer issues the Aggregation reference clock which is the reference clock to the Aggregation time domain.

It is required that:

- The Aggregation layer shall be synchronized to the FE as long as at least 1 (one) pair remains in the Aggregation Group.
- In case of loosing synchronization or lock, recovery shall be done in a way that a service is not dropped for more than 50ms.

It is an objective that:

- The Aggregation layer should not loose synchronization to the FE even in cases of pair failure as long as at least 1 (one) pair remains in the Aggregation Group.

9.4 Pair synchronization

All the Pairs participating in an Aggregation Group shall have the same data clock, coming from the Aggregation layer.

Pair clocking under Modem Rate Matching is described in Annex A.

This recommendation does not define how the Pairs achieve synchronization.

It is required that:

- In case a pair looses synchronization, other pairs in the Aggregation Group shall not loose synchronization.

10 Service encapsulation

10.1 Reference model

The Encapsulation layer performs the following functions:

- Adding stuffing information to each TDM service in order to enable accurate clock transfer for each TDM service.
- Encapsulating the frame based service payload (e.g. Ethernet frame in case of an Ethernet service) so the receiver can determine the start and finish of the payload (only to services where applicable, e.g. Ethernet).
- Allocating the service data streams into the N bits of the 125 usec Bonding Sub-Blocks (see 6.2.1).

Figure 7 below illustrates the flow of the various services (Ethernet, ATM, TDM and Generic Framing Procedure (GFP)) through the Encapsulation layer towards the Aggregation layer.

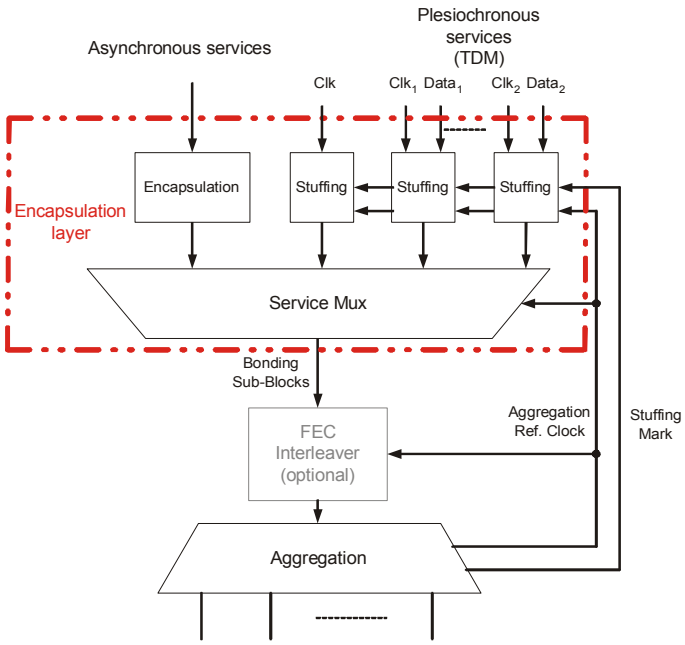

Figure 7_ Service encapsulation reference model

10.2 Service Mux

The service data streams are allocated into the N bits of the 125 usec Bonding Sub-Blocks (see 6.2.1).

The allocation of bits in each Bonding Sub-Block is different, since one in every eight Bonding Sub-Blocks contains a Mini-Frame header, and since a TDM service may require an amount of bytes in 1 msec which is not divisible by 8, resulting in a non even distribution of bits in the Bonding Sub-Blocks. The allocation of bits in the Bonding Sub-Blocks repeats itself every 1 msec (every 8 Bonding Sub-Blocks). In the following text j designates the index of the Bonding Sub-Block in the 1msec Mini-Frame cycle, j = 1:8.

Since TDM services require a constant amount of Bandwidth (BW) they are allocated into a fixed number of bits in the Mini-Frame. TDM_i service is allocated into $N_{i,j}$ bytes in Bonding Sub-Block j. In every Bonding Sub-Block j asynchronous services are allocated into the remaining bits $N_{async,\,j}$. For j = 1, which is the Bonding Sub-Block that contains the Mini-Frame headers, $N_{async,1} = N - \sum_i N_{i,1} - 8 * l * M$, where M is the number of pairs in the Aggregation Group.

l = 2 if the Modem Rate Matching mechanism is used, otherwise l = 1.

For j = 2 to 8, $N_{async,j} = N - \sum_i N_{i,j}$.

The following table defines the values of $N_{i,j}$ for the various TDM service types, including the stuffing information (see 10.4.2). $N_{i,j}$ is the amount of service data bytes (and the associated stuffing) in the j^{th} Bonding Sub-Block in a Mini-Frame.

Note that Fractional Digital Signal 3, (44.736 Mbps) (DS3) and Fractional Electrical interface signal, Level 3, (34.368 Mbps) (E3) are carried as multiple Clear Channel DS1's or E1's, respectively.

The Service type notation is used in the "Service Mapping" and "Service Configuration" messages to specify the $N_{i,j}$ allocation of each service i.

Table 2: Service mapping constants

Service type notation	Service type	$N_{i,1}/8$ [byte]	$N_{i,2}/8$ [byte]	$N_{i,3}/8$ [byte]	$N_{i,4}/8$ [byte]	$N_{i,5}/8$ [byte]	$N_{i,6}/8$ [byte]	$N_{i,7}/8$ [byte]	$N_{i,8}/8$ [byte]
1	Clear Channel DS1	24	24	24	24	24	24	25	25
2	Clear Channel E1	32	32	32	32	32	32	32	33
3	Fractional DS1 (including P x DS0 channels)	P	P	P	P	P	P	P	P+1
4	Fractional E1 (including P channels)	P	P	P	P	P	P	P	P+1
5	DS3	699	699	699	699	699	699	699	700
6	E3	537	537	537	537	537	537	537	538
7	Clock transfer	1	1	1	1	1	1	1	1

10.2.1 Service mapping

Figure 8 describes the service mapping in the time and pairs space, including the Mini-Frames repeating every 1ms. It can be seen that every 1ms, where j = 1, N_{async} is reduced by 8*l*M Bits, where M is the number of pairs in the Aggregation Group, and l = 2 if the Modem Rate Matching is used, and l = 1 otherwise. Note that the dispatching into the pairs is shown here for clarity only, and shall be done according to 7 (Dispatching).

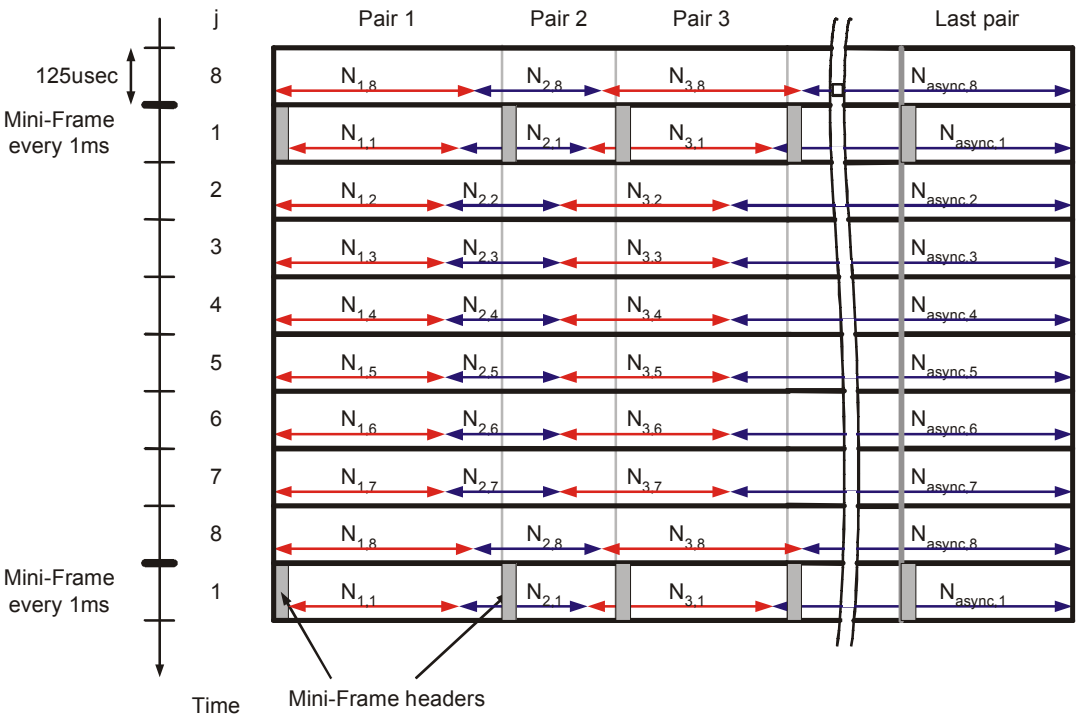

Figure 8- Service mapping in time (and pairs) space

10.2.2 Clock transfer

Clocks (without data) can be transferred from the NE to the FE.

This is done by allocating 1 byte per Bonding Sub-Block for each transferred clock. The first bit (MSB) in each such byte is used to construct a stuffing byte (see

Figure **12**) per Mini-Frame.

The functionality of the stuffing mechanism is described in 10.4.2, with the exception that the receiver functionality is the following:

- If the sum of SC[5:0] equals 0 or 1, then add two clock ticks.
- If the sum of SC[5:0] equals 5 or 6, then subtract two clock ticks.
- If the sum of SC[5:0] equals 2 or 3 or 4, then do nothing.

10.2.3 Service Prioritisation

TDM services have higher priority than asynchronous services. The order of priority from highest to lowest is TDM_1, TDM_2, …, TDM_{last}, Asynchronous service $allocation_1$, Asynchronous service $allocation_2$, Asynchronous service $allocation_{last}$.

Note that there can be more than a single allocation of BW for asynchronous services. In which case the description in 10.2 should be changed accordingly.

When the BW of the Aggregation Group decreases (e.g. due to a pair failure), N decreases accordingly. As a result N_{async} decreases. According to the amount of the BW degradation it could happen that TDM services shall start dropping.

Figure 9 demonstrates this behavior with the BW changes in time marked 1-3:

1. BW decreases. As a result the BW allocated for asynchronous services decreases while the BW allocated for TDM services remains intact.
2. BW decreases to a point where there's not enough BW for TDM_3 so TDM_3 drops. The available BW beyond TDM_1 and TDM_2 is allocated for asynchronous services.
3. BW increases. As a result TDM_3 recovers and the remaining BW is allocated for asynchronous services.

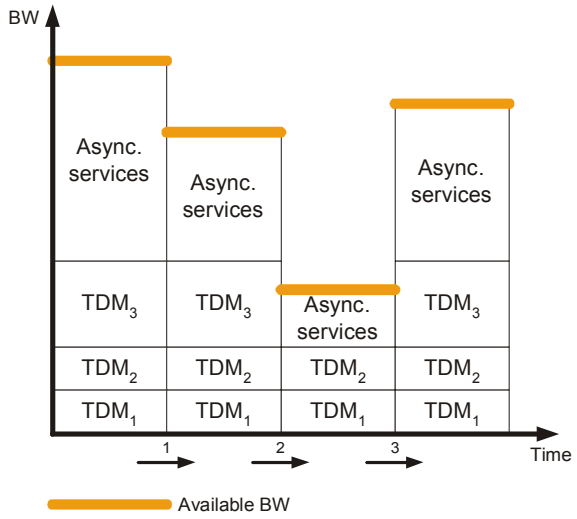

Figure 9 Service prioritization in an available BW

10.3 Asynchronous services

Asynchronous services need to be encapsulated in a manner that enables the receiver to determine the start and finish of the service payload (e.g. Ethernet frame in case of an Ethernet service).

When a few (more than one) asynchronous services are transmitted in the same Bonding system, each asynchronous service shall be allocated with a distinct N_{async_i} bits, and shall be encapsulated separately.

The GFP protocol adds a header information to the incoming service data to identify the type of the encapsulated service, start (and end) of a data unit, and optionally adds some protection.

10.3.1 GFP Format

Standard GFP adds a 4 Byte Core Header to each data packet. This Header consists of 2 Byte long Packet Length Identifier (PLI) and 2 Bytes of Header Error Check (HEC). Encapsulated frames are sent one after another as a continuous Byte stream. The receiver at the FE locks onto the header, using HEC for frame delineation. In addition to the Core Header, a 4 byte Payload Header is pre-pended to the original Payload, containing Payload Type and HEC. Additional extension headers may be added as well for the benefit of applications (up to 60 bytes). GFP also specifies an optional CRC-32 Payload Frame Check Sequence (FCS).

The following figure illustrates GFP frame format:

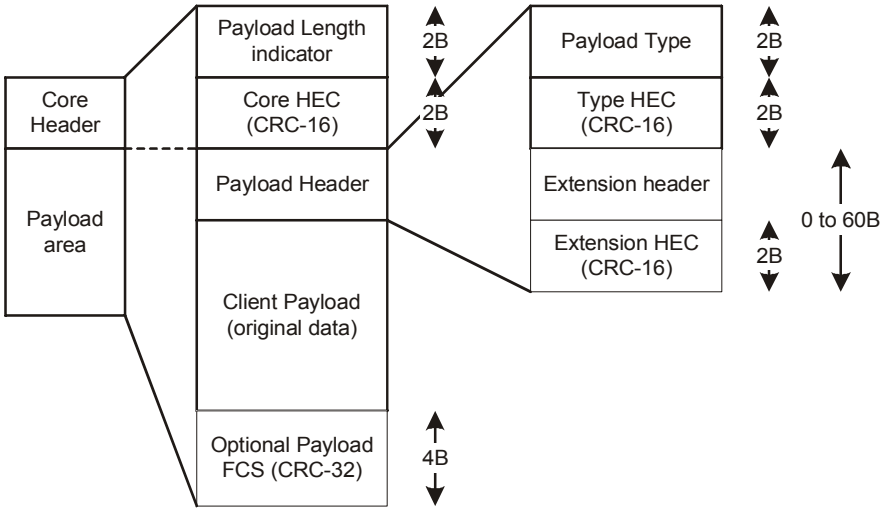

Figure 10- GFP frame format

The payload area is scrambled to improve the robustness of the GFP frame delineation procedure and to provide a sufficient number of 0-1 and 1-0 transitions during idle transmission periods.

In order to compensate for BW differences and achieve a continuous byte stream at a constant rate even when the service BW decreases, Idle frames are inserted into the byte stream when service data is not available. GFP Idle frames are 4 Byte long with PLI=0.

10.3.2 Ethernet

10.3.2.1 Inter-Frame Gap and Preamble deletion and restoring

In Ethernet service transmission Inter-Frame Gaps and Preambles are discarded before the encapsulation, and are regenerated after de-capsulation at the receiver end.

10.3.2.2 Ethernet MAC frame encapsulation

Each Ethernet frame is encapsulated into a single GFP frame. The Ethernet MAC frames (octets from Destination Address through Frame Check Sequence, inclusive) are placed in the GFP Client Payload field. Octet-alignment is maintained and bit identification within octets is maintained.

When Ethernet is the only asynchronous service that is provisioned ("Ethernet only"), a simplified version of GFP shall be used to reduce overhead, with a low fixed overhead of 4 bytes per frame. The following modifications to the original GFP are used in this case:

- Payload Header is removed
- Optional CRC-32 Payload FCS is replaced with CRC-16 (CRC-16 is defined as $G(x) = x^{16} + x^{12} + x^5 + 1$, see ITU-T G.7041/Y.1303 clause 6.1.1.2.1), if FEC between the Encapsulation layer and the Aggregation layer is not used. If FEC between the Encapsulation layer and the Aggregation layer is used the Optional CRC-32 Payload FCS can be omitted to reduce the overhead.

The following encapsulated frame format is used to encapsulate Ethernet frames when Ethernet is the only asynchronous service that is provisioned:

- Header (4 Byte) for frame delineation, including:
    - Payload Length Indicator (PLI:16bit) – 0-65535, indicating length of the Encapsulated frame without the header, i.e. Payload length + optionally 2 Bytes for Frame Check Sequence if used
    - Header CRC (HEC:16bit) – ITU CRC-16 (ITU-T G.7041/Y.1303 clause 6.1.1.2.1)
- Payload - Ethernet Frame = 64 – 1552 Bytes
- Optional Frame Check Sequence (FCS:16bit) – ITU CRC-16 (ITU-T G.7041/Y.1303 clause 6.1.1.2.1), if FEC is not used

The following figure illustrates the simplified GFP frame format:

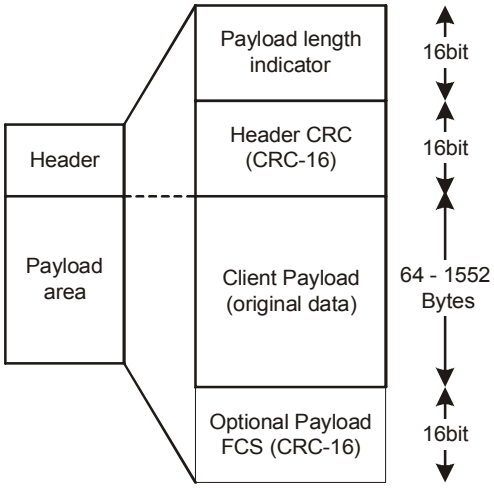

Figure 11: simplified GFP frame format for "Ethernet only"

10.3.2.3 Flow Control

A Flow Control between the Ethernet layer and Encapsulation layer in BTU-C/R shall be implemented, to prevent frame overflow, when Ingress frame/Byte rate exceeds bandwidth available for that Ethernet service in the aggregated link.

10.3.2.4 Data rate decoupling

Data rate decoupling shall be implemented by Idle GFP frames insertion in the transmit direction (downstream) and Idle GFP frames deletion in the receive direction (upstream) at both BTU-C and BTU-R, as specified in ITU-T Rec. G.7041/Y.1303.

NOTE: The GFP Idle frame is a special four-octet GFP control frame consisting of only a GFP Core Header

with the PLI and Core HEC fields set to 0, and no Payload Area.

10.3.2.5 Core header scrambling

The Core Header is scrambled by an exclusive-OR operation (modulo 2 addition) with the hexadecimal number B6AB31E0 as defined in ITU-T Rec. G.7041.Y.1303. The Core Header

Scrambling improves the robustness of the GFP frame delineation procedure and to provide a sufficient number of 0-1 and 1-0 transitions during idle transmission periods.

10.3.2.6 Frame delineation

The GFP delineation function permits the identification of the frame boundaries in the payload. It is based on a coding law using the Header Error Control (HEC) field in the GFP header.

The GFP delineation algorithm shall be as described in ITU-T Rec. G.7041/Y.1303.

10.3.2.7 Mapping of GFP encapsulated frames into Bonding Sub-Blocks

GFP encapsulated frames of Ethernet service j are mapped into the Nasync,j bits of the Bonding Sub-Block on a byte by byte basis. Any alignment to the Bonding Sub-Block is not required. Each byte from the GFP data stream is mapped MSB-first into the next available 8-bits from Nasync,j and possibly split between two Bonding Sub-Blocks.

10.3.2.8 Encapsulation method specification

The Ethernet Service type notation is 8. It is used in the "Service Mapping" and "Service Configuration" messages to specify the encapsulation method of each asynchronous service.

10.3.3 ATM

10.3.3.1 ATM encapsulation functions

The ATM TC layer for the Bonding system is consistent with ITU-T Recommendation I.432.1. It shall provide the following functions, as defined in ITU-T I.432.1:

- Rate decoupling between ATM layer and the synchronous (or plesiochronous) Encapsulation layer.
- Insertion/Extraction (an idle cell inserted at the transmit side has to be extracted at the Far End) of Idle cells.
- Insertion/Extraction (a HEC byte inserted at the transmit side has to be extracted at the Far End) of ATM Header Error Check (HEC) byte.
- Cell payload scrambling / descrambling for SDH-based systems.
- Cell delineation in the receive channel.
- Bit timing and ordering (MSB sent first with bit timing synchronous to the BTU-C downstream timing base).

10.3.3.2 ATM cell encapsulation

When ATM is the only asynchronous service that is provisioned ("ATM only"), it is mapped into the Bonding Sub-Blocks as is without encapsulation, since the ATM cells include all the information required for cell delineation.

10.3.3.3 Flow Control

A Flow Control between the ATM layer and Encapsulation layer in BTU-C/R shall be implemented, to prevent cell overflow, when Ingress cell/Byte rate exceeds bandwidth available for that ATM service in the aggregated link.

10.3.3.4 OAM

The Operations, Administration and Maintenance (OAM) flow across the γ interface exchanges OAM information between the OAM entity and its ATM Encapsulation management functions. OAM flow is bidirectional.

The OAM flow primitives are for *further stud*y.

10.3.3.5 Cell rate decoupling

Cell rate decoupling shall be implemented by Idle cells insertion in the transmit direction and Idle cells deletion in the receive direction, as specified in ITU-T Rec. I.432.1. A standard cell header, also specified in ITU-T Rec. I.432.1 identifies idle cells.

10.3.3.6 HEC generation/verification

The HEC byte shall be generated as described in ITU-T Recs. I.432.x, including the recommended modulo-2 addition (XOR) of the pattern 01010101_2 to the HEC bits. The generator polynomial coefficient set used and the HEC sequence generation procedure shall be in accordance with ITU-T Recs. I.432.x.

The HEC sequence shall be capable of multiple-bit error detection, as defined in ITU-T Recs. I.432.x. The single bit error correction of the cell header shall not be performed.

10.3.3.7 Cell payload randomization and de-randomization

The randomization of the ATM cell payload shall not be performed.

NOTE: Randomization of the transmit ATM cell payload avoids continuous non-variable bit patterns in the ATM cell stream and so improves the efficiency of the cell delineation algorithm. The ATM cell randomization is defined in ITU-T Recs. I.432.x for STM-based transmission.

10.3.3.8 Cell delineation

The cell delineation function permits the identification of the cell boundaries in the payload. It is based on a coding law using the Header Error Control (HEC) field in the cell header.

The cell delineation algorithm shall be as described in ITU-T Recs. I.432.x.

10.3.3.9 Mapping of ATM cells into Bonding Sub-Blocks

ATM cells for service j are mapped into the Nasync,j bits of the Bonding Sub-Block on a byte by byte basis. Any alignment to the Bonding Sub-Block is not required. Each byte from the ATM data stream is mapped MSB-first into the next available 8-bits from Nasync,j and possibly split between two Bonding Sub-Blocks.

10.3.3.10 Encapsulation method specification

The ATM Service type notation is 9. It is used in the "Service Mapping" and "Service Configuration" messages to specify the encapsulation method of each asynchronous service.

10.3.4 GFP

When GFP frames are the only asynchronous service payloads that are transmitted (regardless of what data they encapsulate), they are mapped into the Bonding Sub-Blocks as are without additional encapsulation, since they include all the information required for payload delineation and they also support Idle frames for BW filling.

The GFP Service type notation is 10. It is used in the "Service Mapping" and "Service Configuration" messages to specify the encapsulation method of each asynchronous service.

10.4 TDM service stuffing

Each TDM service (e.g. DS1, E1) data and clock are transferred from the NE to the FE. Plesiochronous services are transferred from the NE to the FE by adding stuffing at the NE and using stuffing removal at the FE. The stuffing is done relatively to the Aggregation reference clock, which synchronizes the Aggregation layers of BTU-C and the BTU-R.

10.4.1 Amount of stuffing

The stuffing mechanism that is described below is suitable for TDM services such as DS1, E1, Fractional DS1, Fractional E1, DS3 and E3.

The following table presents the required amount of stuffing for the various types of TDM services, calculated according to the service rate, its standard clock accuracy and an assumed local clock accuracy of the Aggregation reference clock that is 20 ppm.

Table 3: Required amount of stuffing according to TDM service type

Service type	Nominal rate [kbps]	Number of bytes in 125 usec	Standard accuracy [ppm]	Local clock accuracy	Max clock offset	Number of required stuffing bits in 1ms
CLEAR CHANNEL DS1	1544	24.125	32	20	52	0.08
Clear Channel E1	2048	32	50	20	70	0.14
Fractional DS1 (including Nx DS0)	1544	N+1/8	32	20	52	(N+1/8) x 64K x 52ppm x 1 msec
Fractional E1 (including N channels)	2048	N+1	50	20	70	(N+1) x 64K x 70ppm x 1 msec
DS3	44736	699	20	20	40	1.79
E3	34368	537	20	20	40	1.37

Two (2) stuffing bits in 1 msec are sufficient for the service types in Table 3.

10.4.2 Stuffing mechanism

In addition to its data, each TDM service has an additional 1 byte per 1 msec for the stuffing mechanism. The Encapsulation layer is synching to the 1 msec Mini-Frame according to the Stuffing Mark (see Figure 7).

The format of this stuffing byte is:

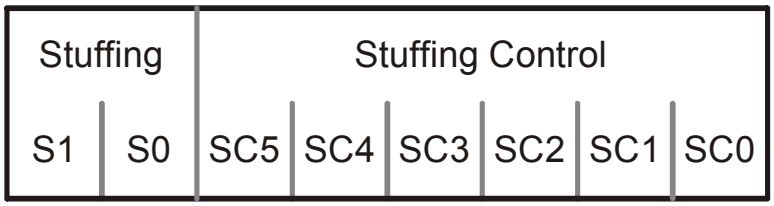

Figure 12- Stuffing byte format

To achieve robustness (the stuffing mechanism is robust for a single bit error) the stuffing byte is evenly scattered throughout the Mini-Frame. For TDM service i: S1 is the first bit of $N_{i,1}$, S0 is the first bit of $N_{i,2}$, SC5 is the first bit of $N_{i,3}$, SC4 is the first bit of $N_{i,4}$, SC3 is the first bit of $N_{i,5}$, SC2 is the first bit of $N_{i,6}$, SC1 is the first bit of $N_{i,7}$, and SC0 is the first bit of $N_{i,8}$.

The transmitter sets the content of S[1:0] and SC[5:0] in the following manner:

- If the transmitter has two additional data bits to transmit, then
  - SC[5:0] = ’000000’
  - S[1:0] in the next Mini-Frame shall have the two additional data bits.
  - The last two data bits in the last byte of the TDM service i in the next Mini-Frame shall be two data bits.
- If the transmitter is lacking two data bits to transmit, then
  - SC[5:0] = ’111111’
  - S[1:0] in the next Mini-Frame shall be set to ‘01’ (a null value, ignored by the receiver).
  - The last two data bits in the last byte of the TDM service i in the next Mini-Frame shall be ‘01’ (a null value, ignored by the receiver).
- On all other cases
  - SC[5:0] = ’101010’
  - S[1:0] in the next Mini-Frame shall be set to ‘01’ (a null value, ignored by the receiver).
  - The last two data bits in the last byte of the TDM service i in the next Mini-Frame shall be two data bits.

The receiver shall follow the following rules:

- If the sum of SC[5:0] equals 0 or 1, then

- S[1:0] in the next Mini-Frame are valid data bits.
- The last two data bits in the last byte of the TDM service i in the next Mini-Frame are valid data bits.

- If the sum of SC[5:0] equals 5 or 6, then
  - S[1:0] in the next Mini-Frame are stuffing bits and shall be dropped.
  - The last two data bits in the last byte of the TDM service i in the next Mini-Frame are stuffing bits and shall be dropped.
- If the sum of SC[5:0] equals 2 or 3 or 4, then
  - S[1:0] in the next Mini-Frame are stuffing bits and shall be dropped.
  - The last two data bits in the last byte of the TDM service i in the next Mini-Frame are valid data bits.

11 FEC and Interleaver

The FEC and Interleaver are optional in the Bonding system.

11.1 FEC

11.1.1 FEC type

The FEC for the Bonding system is similar to the FEC defined for ADSL and Very high speed Digital Subscriber Line (VDSL), i.e. it is derived from a Reed Solomon (RS) code with the generating polynomial

$$G(D) = \prod_{i=0}^{19} \left(D + \alpha^{i}\right)$$

where α is a primitive element that satisfies the primitive binary polynomial $x^8 + x^4 + x^3 + x^2 + 1$ in the Galois Field GF(256). (GF - Galois Field).

Note that G(D) is chosen to be a uniquely chosen polynomial in order to simplify changes to the FEC parameters.

11.1.2 Encoding

$R=R_{RS}$ redundant check bytes c_0 , c_1, ..., c_{R-2}, c_{R-1} shall be appended to $K=K_{RS}$ information bytes m_0, m_1, ..., m_{K-2}, m_{K-1} to form a codeword of size $N_{RS} = K_{RS} + R_{RS}$ bytes. The check bytes are computed from the message byte using the equation:

$$C(D) = M(D)\ D^{20} \text{modulo}\ G(D)$$

where a message polynomial M(D) is defined as:

$$M(D) = m_0\ D^{K-1} + m_1\ D^{K-2} + ... + m_{K-2}\ D + m_{K-1}$$

And the check polynomial is defined as:

$$C(D) = c_0 D^{19} + c_1 D^{18} + ... + c_{R-2} D^{21-R} + c_{R-1} D^{20-R} + c_R D^{19-R} + \ldots + c_{19}$$

Note that only the first $R=R_{RS}$ coefficients of the check polynomial are used in the code word. The decoder shall add the remaining $20-R_{RS}$ coefficients and use erasure decoding.

A data byte (d_7, d_6, ..., d_1, d_0) is identified with the Galois Field element $d_7\alpha^7 + d_6\alpha^6 \ ... + d_1\alpha + d_0$.

11.1.3 FEC parameters

The FEC parameters to be specified are:

- N_{RS} = Code word size
- K_{RS} = Information size of the code word
- $R_{RS} = N_{RS}-K_{RS,}$ Redundancy size of the code word
- S = Number of code words in a Bonding Sub-Block of 125 usec

In addition the following notation shall be used:

- M = Number of active pairs in the Aggregation Group
- l = 2 if the Modem Rate Matching is used, and l = 1 otherwise
- B_i = Number of bits carried by pair i in a Bonding Sub-Block of 125 usec (data rate of pair i divided by 8K)

Note that the parameter S is slightly different than that of S for ADSL. The parameter S that is defined here is associated with 1/S for the parameter S defined in the ADSL Recommendation. The definition of S used in this Recommendation is more suitable for high BW systems, as it naturally leads to integer values of S. S is the number of code words in a Bonding Sub-Block of 125 usec, while S+1 code words exceed the number of bytes in a Bonding Sub-Block.

The following values shall be supported:

- N_{RS}= 5 … 255
- R_{RS} = 2,4,8,16,20

As will be explained below the information rate of the bonding system is $64K_{RS}S-8*l*M$ kbps, therefore the parameter S shall be computed from the values of K_{RS}, and the total BW required by all the services (TotalBW) to be the smallest integer such that:

$$64K_{RS}S-8*l*M \geq \text{TotalBW},$$

where TotalBW is measured in kbps.

The values of N_{RS} and S shall satisfy the following inequality:

$$N_{RS} \cdot S \leq floor\left(\left(\sum_{i=1}^{M} B_i\right)/8\right).$$

In addition, in order to support the TDM services the information size shall satisfy the inequalities ($N_{i,j}$ as in 10.2, j denotes the order of the Bonding Sub-Block in a Mini-Frame, see Figure 8 and Table 2):

For j = 1: $K_{RS}S\text{-}l*M \leq \sum_i N_{i,1}/8$

For j = 2…8: $K_{RS}S \leq \max_j \sum_i N_{i,j}/8$

Note that the $N_{i,j}$ used here are not related to the FEC N_{RS}, but rather to $N_{i,j}$ as in 10.2.

The parameters N_{RS} and R_{RS} are computed by the BTU-C after setting the rates of the pairs and after every change in Dispatching table parameters, and communicated to the BTU-R via a "FEC Capability Request" message.

11.1.4 Data flow

The following figure describes the data flow through the FEC, and the FEC boundaries:

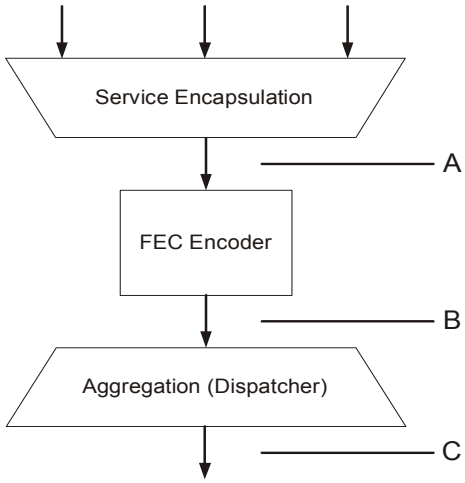

Figure 13- FEC data flow

The Encapsulation unit is set to deliver an output rate (at reference point A) of ($64K_{RS}S\text{-}8*l*M$) kbps.

The RS encoder encodes the first code word in a 1 msec Mini-Frame with a reduced size N_{RS}-M by reducing the information size to K_{RS}-M, while maintaining the redundancy size of R_{RS}. This may be done directly or by preceding the first K_{RS}-M information bytes by M zero bytes. This leaves room for the dispatcher to insert the M Mini-Frame header bytes.

The RS encoder encodes the remaining 8S-1 code words in a 1 msec Mini-Frame with a code word size of N_{RS} and an information size of K_{RS}. This generates an output rate (at reference point B) of ($64N_{RS}S - 8*l*M$) kbps.

The Dispatcher dispatches $\sum_{i=1}^{M} B_i$ bits for every Bonding Sub-Block of 125 usec generating a rate of $8\sum_{i=1}^{M} B_i$ kbps at reference point C. The receiver shall ignore the last $\sum_{i=1}^{M} B_i - 8 \cdot S \cdot N_{RS}$ bits of each Bonding Sub-Block of 125 usec, before the RS decoding.

The Mini-Frame header bytes are inserted in the Aggregation layer, after the FEC.

11.1.5 FEC parameter changes

Changes in the RS parameters shall be performed during a Sync Change procedure or during a Fast Change procedure.

11.1.5.1 Sync Change procedure

Sync Change procedure is initiated by the BTU-C to change the bonding system configuration (pairs that participate in the Aggregation Group, Services mapping, FEC and Interleaver) in a controlled hitless manner. As part of a Sync Change procedure the BTU-C shall compute new FEC parameters and send them to the BTU-R via a "FEC Capability Request" message. When the BTU-C or the BTU-R switch to the new parameters in a synchronized fashion using evConfigSw events, the FEC parameters are changed as well without causing errors.

Since the FEC code words are aligned with the 1 msec Mini-Frames but not with the 125 usec Bonding Sub-Blocks, the parameter changes shall be performed at the beginning of a 1 msec Mini-Frame in order to ensure a hitless transition.

11.1.5.2 Fast Change procedure

Fast Change procedure is initiated by the BTU-C to remove failing pairs from the Aggregation Group as fast as possible. Changing the pairs in the Aggregation Group using the Fast Change procedure is fast (several milliseconds) but does not assure a hitless change. After the BTU-C initiates a Fast Change procedure, and both the BTU-C and the BTU-R have received the Fast Change request, the BTU-C and the BTU-R change parameters independently. The following changes are performed:

1. The parameters K_{RS} and N_{RS} are reduced by $\lceil B_i / 8S \rceil$ for each failed pair i, where $\lceil X \rceil$ denotes the smallest integer not smaller than X.
2. The parameter M is also reduced to denote the number of the remaining pairs in the Aggregation Group.
3. The size of the first codeword in each 1 msec Mini-Frame is reduced by M, where M denotes the number of remaining pairs in the Aggregation Group.
4. The BW of the TDM services is checked vs. the inequalities $K_{RS}S\text{-}1*M \le \sum_i N_{i,1} / 8$ and $K_{RS}S \le \max_j \sum_i N_{i,j} / 8$, and TDM services are removed according to a predefined priority order if there is not enough BW for all of them.

As a result:

- The output rate of the Encapsulation layer (at reference point A) and the output rate of the RS encoder (at reference point B) are each reduced according to the new parameters, N_{RS}, K_{RS}, M.

- The number of bits which are ignored by the receiver at the end of every S code words is also changed according to the new parameters N_{RS}, K_{RS}, M and is computed to be the difference between the aggregate rate and the RS encoder rate.

11.2 Interleaver

A Bonding system that supports an Interleaver shall support two types of Interleavers: a Block Interleaver and a Convolution Interleaver.

Block Interleavers introduce a longer delay than Convolution Interleavers with the same depth, however block Interleavers can support hitless parameters change with a Sync Change procedure.

The Interleaver is defined by 2 parameters: Interleaver type (Block, Convolution) and Depth. These parameters are computed by the BTU-C and sent to the BTU-R via a "FEC Capability Request" message. When the BTU-C or the BTU-R switch to the new parameters in a synchronized fashion using evConfigSw events, the Interleaver parameters are changed as well without causing errors.

The depth of the Interleaver shall be $S \cdot 3^a \cdot 2^b$, where a=0,1 and b=0…5. The maximal Interleaver Depth corresponds to 12ms.

11.2.1 Block Interleaver

In a Block Interleaver code words are (logically) grouped to sets of D code words, and written to a memory array of DxN (D rows, N columns), where code word i is stored in row i and N is the number of bytes in a code word. The code words are then read from the memory and transmitted column-wise, i.e. the first bytes of all D code words are transmitted followed by the second bytes of all D code words etc. This is illustrated in the following figure (for N=7, D=4):

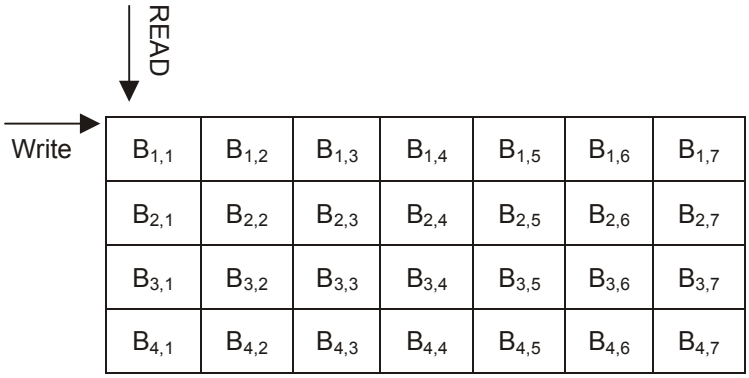

Figure 14- Block Interleaver

The transmit sequence shall be: $B_{1,1}$; $B_{2,1}$; $B_{3,1}$, $B_{4,1}$; $B_{1,2}$; $B_{2,2}$; …. $B_{3,7}$; $B_{4,7}$.

At the De-interleaver at the receiver the role of the read and write is reversed, i.e. received bytes are written column-wise, and then read row-wise into the FEC RS decoder.

In order to align the Interleaver blocks with the 12 msec Super Frames, the depth D shall be a divisor of 96 for S=1. For S>1, the Interleaver depth D shall be a divisor of 96 multiplied by S.

In addition the Interleaver block shall be aligned with the 12 msec Super Frames, i.e. the first code word in a 12 msec Super Frame shall be the first code word in an Interleaver block.

The first M bytes of the first code word in every 1 msec Mini-Frame, which is a shorter code word than the other code words in the Mini-Frame, shall be filled with M dummy bytes prior to interleaving. The dummy bytes shall be removed prior to transmission.

11.2.2 Convolution Interleaver

The interleaving depths supported are the same as for the block interleaver.

Convolution interleaving is defined by the rule:

Each of the $N=N_{RS}$ bytes $B_0, B_1, ..., B_{N-1}$ in a RS code word is delayed by an amount that varies linearly with the byte index. More precisely, byte B_i (with index i) is delayed by $(D–1) \times i$ bytes, where D is the Interleaver depth.

An example for $N = 5$, $D = 2$ is shown in Table 4, where Bj_i denotes the i-th byte of the j-th codeword.

Table 4: Interleaver example for N = 5, D = 2

Interleaver input	Bj_0	Bj_1	Bj_2	Bj_3	Bj_4	$Bj+1_0$	$Bj+1_1$	$Bj+1_2$	$Bj+1_3$	$Bj+1_4$
Interleaver output	Bj_0	$Bj\text{-}1_3$	Bj_1	$Bj\text{-}1_4$	Bj_2	$Bj+1_0$	Bj_3	$Bj+1_1$	Bj_4	$Bj+1_2$

With the above-defined rule, and the chosen interleaving depths, the output bytes from the Interleaver always occupy distinct time slots when N is odd and is co-prime to S.

When N is even or not co-prime to S, dummy bytes shall be added at the beginning of the code word at the input to the Interleaver. The number of dummy bytes shall be minimal to satisfy the previous condition. The resultant code word is then convolutionally interleaved, and the dummy bytes shall then be removed from the output of the Interleaver.

For the first code word in every 1 msec Mini-Frame, which has a different length than the other code words, additional dummy bytes shall be added to the beginning of the code word in order to equate its length with the length of the other code words. The resultant codeword is then convolutionally interleaved, and the dummy bytes shall then be removed from the output of the Interleaver.

12 Processes

12.1 Pair management and control

12.1.1 Pair operations

Pair operations are done according to the Pair state machine, and include:

- Setting up pair parameters, by using G.handshake
- Pair activation
- Synchronizing the pair to an Aggregation Group
- Adding or removing the pair to / from an Aggregation Group

- Handling pair failure in synchronization to Aggregation Group, in activation or in handshake

12.1.2 Pair state machine

The following figure describes the Pair State Machine, with the states and the transitions between the states. The states and transition conditions are defined in 5.8.1.3 and 5.8.1.4.

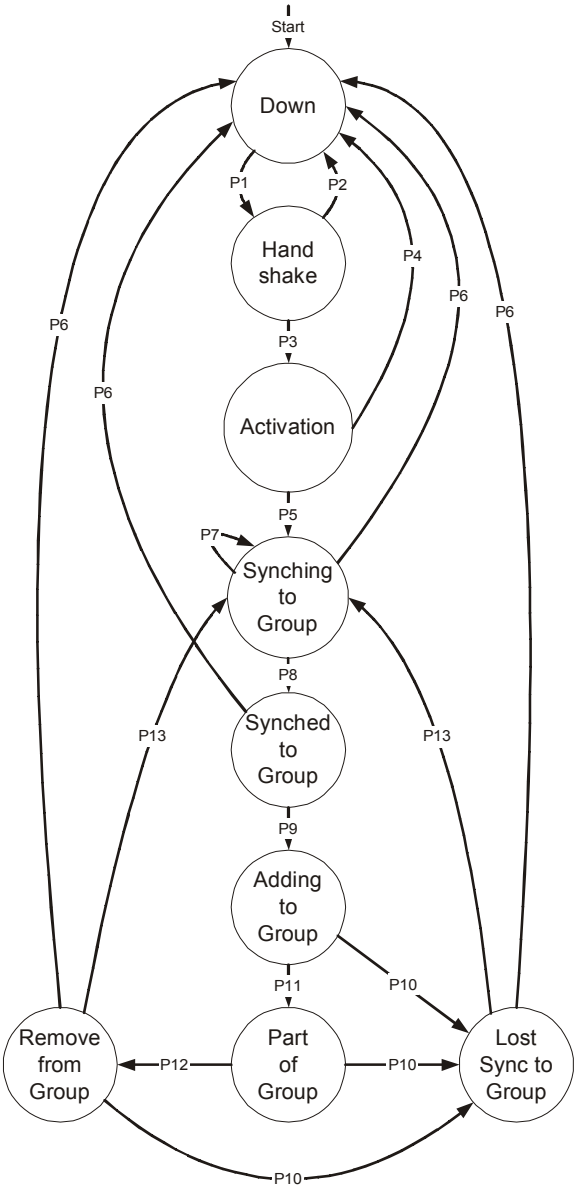

Figure 15- Pair State Machine

12.1.3 States

Table 5: Pair State Machine states definition

State	Definition	Transitions
Down	The pair is not transmitting nor receiving	• [P1] to “Handshake” state
Handshake	The pair is using Handshake (G.994.1) for discovery and to negotiate and set its operational parameters (e.g. rate, modulation, Power Spectral Density (PSD) mask, the bonding scheme (TDIM, Ethernet in the First Mile (EFM), etc.))	• [P2] to “Down” state • [P3] to “Activation” state
Activation	The pair is activating with its operational parameters	• [P4] to “Down” state • [P5] to “Synching to Group” state
Synching to Group	The pair is operating, the Aggregation layer operates “Sync to Group” procedure (12.3.3)	• [P6] to “Down” state • [P7] to “Synching to Group” state • [P8] to “Synched to Group” state
Synched to Group	The pair is synchronized to the Aggregation Group, waiting for a management decision to be added to the Aggregation Group	• [P6] to “Down” state • [P9] to “Adding to Group” state
Adding to Group	The Aggregation layer is attempting to add the pair to the Aggregation Group by using the “Sync Change” procedure (12.3.2)	• [P10] to “Lost Sync to Group” state • [P11] to "Part of Group” state
Part of Group	The Aggregation layer delivers service data through the pair according to the Dispatching Table	• [P10] to “Lost Sync to Group” state • [P12] to “Remove from Group” state
Lost Sync to Group	The Aggregation layer is operating “Fast Change” procedure, after which the pair is removed from the Aggregation Group (12.3.1)	• [P6] to “Down” state • [P13] to “Synching to Group” state
Remove from Group	The Aggregation layer is operating “Sync Change” procedure, after which the pair is removed from the Aggregation Group (12.3.2)	• [P6] to “Down” state • [P10] to “Lost Sync to Group” state • [P13] to “Synching to Group” state

Note:

- A pair is considered "Synchronized to Group" when it is in one of the states: "Synched to Group", "Adding to Group", "Part of Group".

12.1.4 Transition conditions

[P1] A management decision to start a handshake session

[P2] Handshake failure, or a management decision to stop handshake and return to "Down" state

[P3] Successful completion of handshake

[P4] Failure in pair activation

[P5] Successful completion of activation

[P6] Pair failure

[P7] A management decision to synchronize the pair to the Aggregation Group

[P8] Successful completion of synchronization to the Aggregation Group

[P9] A management decision to add the pair to the Aggregation Group

[P10] Lost sync to the Aggregation Group. A pair is considered to have lost sync to the Aggregation Group after: (a) 10 (ten) consecutive Frame header errors (the Frame header is erroneous if its CRC-4 does not match the header content and / or one of the SF indication bits is not received as expected) or (b) 3 consecutive faults in Fast Change procedure (see 12.3.1)

[P11] Successful completion of adding the pair to the Aggregation Group

[P12] A management decision to remove the pair from the Aggregation Group

[P13] A management decision to recover the pair as part of the Aggregation Group

12.2 Aggregation Group management and control

12.2.1 Aggregation Group operations

Aggregation Group operations are done according to the Aggregation Group state machine, and include:

- Setting up the Aggregation Group to Diag mode where control data is transmitted and service data is not transmitted.
- Initializing an Aggregation Group by assigning pairs to it
- Addition or removal of pairs to / from the Aggregation Group
- Handling pair failures

12.2.2 Aggregation Group state machine

The following figure describes the Aggregation Group State Machine, the Aggregation Group States and the transitions between the states. The states and transitions are defined in following paragraphs.

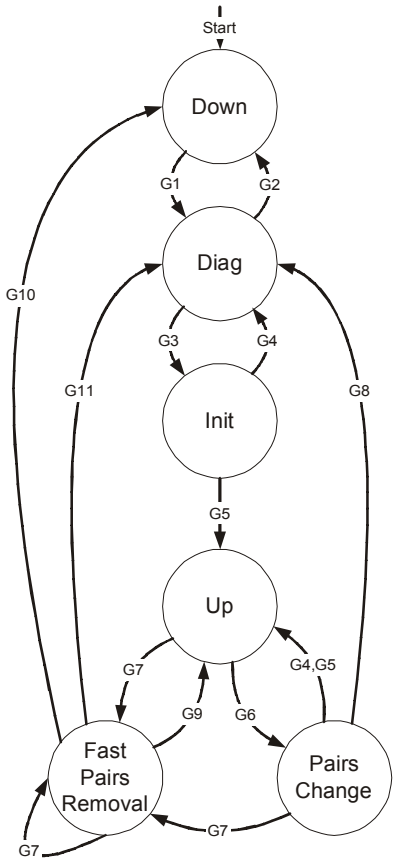

Figure 16- Aggregation Group State Machine

12.2.3 States

Table 6- Aggregation Group State Machine states definition

State	Definition	Transitions
Down	There are no pairs assigned to the Aggregation Group, or there are pairs assigned and none of them is in one of the states in which it is "Synchronized to Group" (i.e. "Synched to Group", "Adding to Group", "Part of Group")	• [G1] to "Diag" state
Diag	One or more of the pairs related to the Aggregation Group are in "Synched to Group" state, and none of the pairs is in "Part of Group" state	• [G2] to "Down" state • [G3] to "Init" state
Init	Initiating the Aggregation Group by using the "Sync Change" procedure with all of the pairs in that are in "Synched to Group" state	• [G4] to "Diag" state • [G5] to "Up" state
Up	The Aggregation Group is working in a steady-state and transferring service data	• [G6] to "Pairs Change" state • [G7] to "Fast Pairs Removal" state
Pairs Change	Hitless pairs addition or removal to / from the Aggregation Group by using the "Sync Change" procedure	• [G4], [G5] to "Up" state • [G7] to "Fast Pairs Removal" state • [G8] to "Diag" state
Fast Pairs Removal	Fast removal of pairs upon pair's loss of sync to the Aggregation Group, by using the "Fast Change" procedure	• [G9] to "Up" state • [G7] to "Fast Pairs Removal" state • [G10] to "Down" state • [G11] to "Diag" state

12.2.4 Transition conditions

[G1] One or more pairs related to the Aggregation Group changed their state to "Synched to Group" state

[G2] None of the pairs related to the Aggregation Group are in one of the states in which they are "Synchronized to Group" (i.e. "Synched to Group", "Adding to Group", "Part of Group")

[G3] Management decision to activate the Aggregation Group

[G4] Failure in "Sync Change" procedure

[G5] "Sync Change" procedure completed successfully, and there is at least one pair participating in the Aggregation Group

[G6] Management decision to add / remove pairs to / from the Aggregation Group

[G7] One or more of the pairs lost its synchronization to the Aggregation Group

[G8] "Sync Change" procedure completed successfully, and no pairs are left in the Aggregation Group

[G9] "Fast Change" procedure completed successfully, and there is at least one pair participating in the Aggregation Group

[G10] Failure in "Fast Change" procedure (after 3 consecutive faults in Fast Change procedure, see 12.3.1.1.1)

[G11] "Fast Change" procedure completed successfully, and no pairs are left in the Aggregation Group

12.3 Procedures

All the procedures are initiated by the BTU-C, except for the Pair Synchronization Loss procedure that can be initiated also by the BTU-R.

12.3.1 Fast Change

Fast Change procedure is initiated by the BTU-C to remove failing pairs from the Aggregation Group as fast as possible. Changing the pairs in the Aggregation Group using the Fast Change procedure is fast (several milliseconds) but does not assure a hitless change.

The following describes the operation at each side, and the event transactions between the BTU-C and the BTU-R during a "Fast Change" procedure:

- The BTU-C sends evFastChange event to the BTU-R with a value indicating the participating pairs after removing the faulty pairs. The BTU-C starts dispatching service data according to the new participating pairs at the beginning of the next sub-block. The BTU-C continues sending the evFastChange event until it receives the same event (with the same pairs bitmap value) from the BTU-R. The BTU-C can start a new Fast Change procedure at any time if the pairs map changes.
- After receiving evFastChange event, the BTU-R decodes the participating pairs from the value field, and sends evFastChange event to the BTU-C, with value indicating the participating pairs. The BTU-R continues sending the evFastChange event until it receives a different event (e.g. evNull) from the BTU-C. The BTU-R starts dispatching service data according to the new participating pairs. The duration between receiving the evFastChange event and the Dispatching Table switch shall not exceed T_{fcp} (T_{fcp}=1 msec).
- After receiving evFastChange event from the BTU-R, the BTU-C compares the participating pairs value, and starts a new Fast Change process in case of discrepancy.

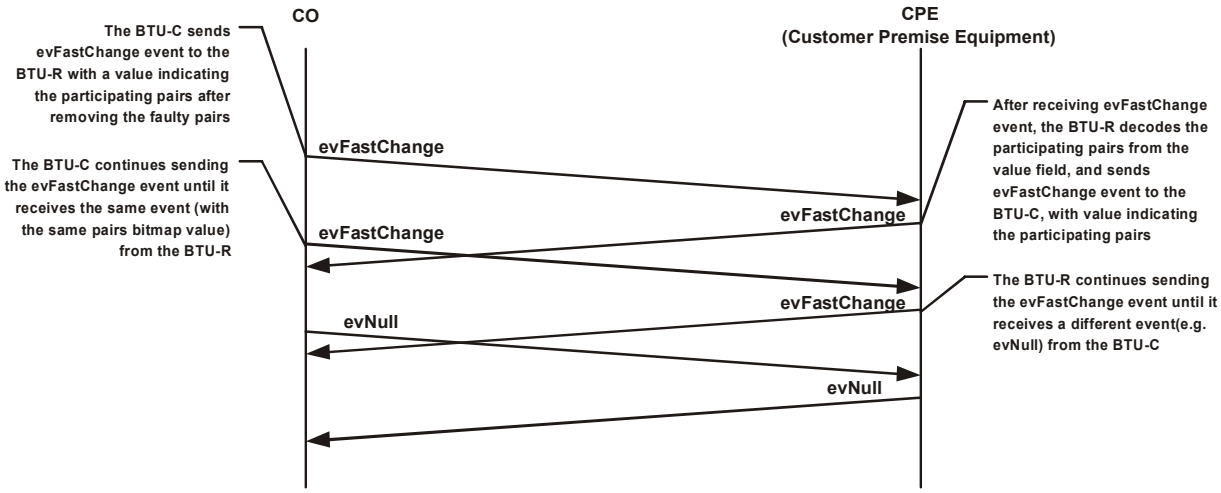

Figure 17- Fast Change transaction

12.3.1.1 Fault handling

12.3.1.1.1 BTU-C

The following failures can be detected in the BTU-C during Fast Change procedure:

- BTU-C is not receiving any evFastChange events from the BTU-R during T_{frs} after starting sending evFastChange events (T_{frs} = 50 msec).
- BTU-C is receiving a pair bitmap that is not identical to the pair bitmap it sent.
- BTU-C is receiving an evFastChange event without have sending any evFastChange event.

Upon detection of any of the above failures the BTU-C shall send at least two evNull events to signal the BTU-R that the previous evFastEvent transaction is completed, and then restart the execution of a Fast Change procedure. In the new Fast Change procedure the participating pair list can be updated.

After three consecutive failures the BTU-C shall change the state of all pairs related to the Aggregation Group to "Lost Sync to Group" state (this action shall cause sending all ones on these pairs) and change the state of the Aggregation Group to "Down" state.

12.3.1.1.2 BTU-R

Upon receiving an evFastChange with unavailable pairs in the bitmap value, the BTU-R shall reply to the BTU-C with an evFastChange event with an empty pair bitmap (a bitmap that does not include any pair). The BTU-C shall then detect discrepancy in the pair bitmap of the received evFastChange event and shall initiate a new Fast Change procedure.

12.3.2 Sync Change

Sync Change procedure is initiated by the BTU-C to change the pairs that participate in the Aggregation Group in a controlled hitless manner. Examples for usage are when the operator needs to remove a pair for maintenance, when more pairs are required to add bandwidth, etc. Changing

the participating pairs using the Sync Change procedure is slower than using Fast Change procedure, but insures that the service data is not disturbed.

The following describes the operation at each side, and the event transactions between the BTU-C and the BTU-R during a "Sync Change" procedure:

- The BTU-C sends evSyncChange event to the BTU-R with a value indicating the participating pairs after the change. The BTU-C continues sending the evSyncChange event until it receives the same event (with the same pairs bitmap value) from the BTU-R. The BTU-C can start a Fast Change procedure at any time if pairs fail during the Sync Change procedure.
- After receiving evSyncChange event, the BTU-R decodes the participating pairs from the value field, and sends evSyncChange event to the BTU-C, with value indicating the new expected participating pairs. The BTU-R continues sending the evSyncChange event until it receives an evConfigSw event from the BTU-C. The duration between receiving the evSyncChange event from the BTU-C and sending the evSyncChange event by the BTU-R shall not exceed T_{scp} (T_{scp}=18 msec).
- After receiving evSyncChange event from the BTU-R, the BTU-C compares the participating pairs value, and starts sending evConfigSw events. The value included in the evConfigSw events shall be a BTUC_tx_counter (starting from a value of at least 0x03, and counting down at each Super Frame) indicating the number of Super-Frames left until the BTU-C transmitter switches to the new configuration including Dispatching Table, Service Configuration and FEC- Interleaver (IL) configuration).
- When receiving the first evConfigSw event, the BTU-R shall:
    - Start sending evConfigSw events back to the BTU-C. The value in the event shall be a BTUR_tx_counter (starting from a value of at least 0x03, and counting down at each Super Frame) indicating the number of Super-Frame left until the BTU-R transmitter switches to the new configuration.
    - Start counting down using BTUR_receiver (rx) counter from the initial BTUC_tx_counter value taken from the received evConfigSw event.
- BTU-C transmitter and BTU-R receiver synchronized change:
    - The Transmitter at the BTU-C shall start working according to new configuration at the beginning of the transmitted Super-Frame following the Super-Frame with BTUC_tx_counter of value 0x01.
    - The BTU-R receiver shall start working according to the new configuration at the beginning of the received Super-Frame at which BTUR_rx_counter reaches 0.
- When receiving the first evConfigSw event, the BTU-C shall start counting down using BTUC_rx_counter from the initial BTUR_tx_counter value taken from the received evConfigSw event.
- BTU-R transmitter and BTU-C receiver synchronized change:
    - The Transmitter at the BTU-R shall start working according to new configuration at the beginning of the transmitted Super-Frame following the Super-Frame with BTUR_tx_counter of value 0x01.
    - The BTU-C receiver shall start working according to the new configuration at the beginning of the received Super-Frame at which BTUC_rx_counter reaches 0.

The "Sync Change" procedure achieves a synchronized change to the new configuration at the BTU-C transmitter and the BTU-R receiver. There is no coupling between a change in the configuration of the receiver and the transmitter at the same side (BTU-C or BTU-R), therefore there may be a time when the receiver and transmitter are working according to different configurations.

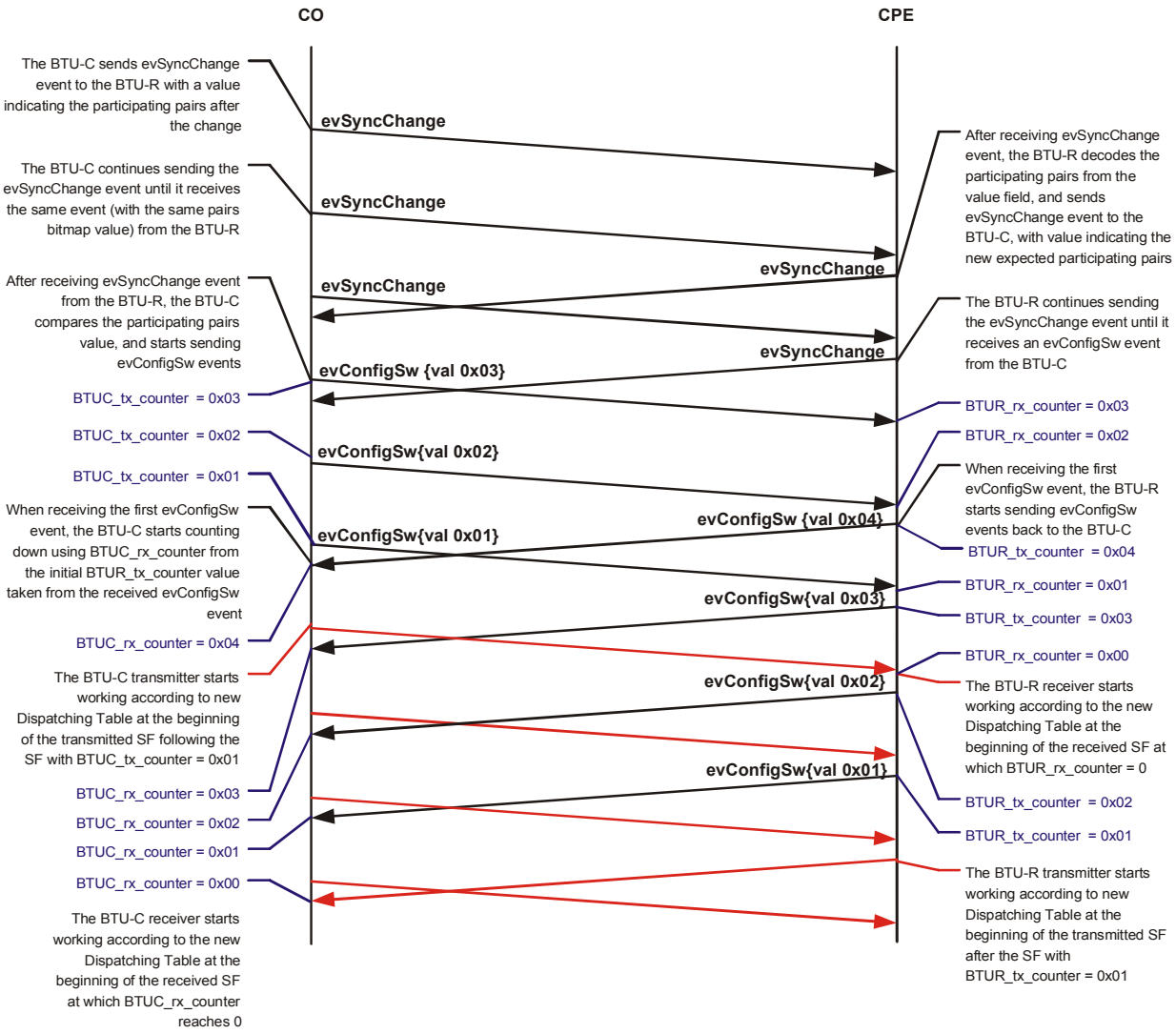

Figure 18- Sync Change transaction

12.3.2.1 Fault handling

12.3.2.1.1 BTU-C

1. The following failures can be detected in the BTU-C during the initialization of a Sync Change procedure:

- BTU-C is not receiving any evSyncChange events from the BTU-R during T_{srs} after starting sending evSyncChange events (T_{srs} = 50 msec).
- BTU-C is receiving a pair bitmap that is not identical to the pair bitmap it sent.
- BTU-C is receiving an evSyncChange event without have sending any evSyncChange event.

These failures may occur in a preliminary stage of the Sync Change procedure, before any changes to the Dispatching Table were done. Upon detection of any of the above failures the BTU-C shall send at least two evNull events to signal the BTU-R that the previous evSyncEvent transaction is completed and shall change back the Aggregation Group State Machine to "Up" state. The decision whether to retry the Sync Change procedure or to take other action is implementation dependent and is not specified in this Recommendation.

2. The following failures can be detected in the BTU-C after sending the first evConfigSw event to the BTU-R.

- BTU-C is not receiving any evConfigSw events from the BTU-R during T_{srs} after starting sending evConfigSw events (T_{srs} = 50 msec).
- BTU-C is receiving an evSyncChange event or an evConfigSw event from the BTU-R not as a response to same event initiated by the BTU-C.

In these failures it is possible that the BTU-R has started changing its Dispatching Table. Therefore upon detection of any of the above failures the BTU-C shall send at least two evNull events to signal the BTU-R that the previous evSyncEvent transaction is completed, and then start the execution of a Fast Change procedure. In the new Fast Change procedure the participating pair list can be updated.

3. Upon receiving an evConfigSw event with a BTUR_tx_counter value that is not correlated to its internal BTUC_rx_counter, the BTU-C shall set its BTUC_rx_counter value according to the received BTUR_tx_counter value.

12.3.2.1.2 BTU-R

1. Upon receiving an evSyncChange with unknown / disabled pairs in the bitmap value, the BTU-R shall reply to the BTU-C with an evSyncChange event with an empty pair bitmap (a bitmap that does not include any pair). The BTU-C shall then detect discrepancy in the pair bitmap of the received evSyncChange event and shall initiate a new Sync Change procedure.

2. Upon receiving an evConfigSw event with a BTUC_tx_counter value that is not correlated to its internal BTUR_rx_counter, the BTU-R shall set its BTUR_rx_counter value according to the received BTUC_tx_counter value.

3. If a Sync Change procedure has already started (i.e. the BTU-R received at least one evSyncChange event), the BTU-R shall stop any sync change actions upon receiving any event that is different than evSyncChange or evConfigSw. The BTU-R shall response normally to such event.

12.3.3 Sync to Group

Sync to Group procedure is initiated by the BTU-C to synchronize pairs to an Aggregation Group. This process can start if the pair is already logically assigned to a group by the Management layer.

The following describes the operation at each side, and the event transactions between the BTU-C and the BTU-R during a "Sync to Group" procedure:

12.3.3.1 At the Transmitter

- All the pairs that are in "Synching to Group" state shall transmit an evSync event, with value according to 13.2.3, and a CRC-6 value fixed to 000000b.
- The pair in the BTU-C shall stop sending evSync events after receiving an evSync event from the BTU-R with Value[0] indicating NE synchronization.
- The pair in the BTU-R shall stop sending evSync events after receiving from the BTU-C a Super-Frame with an event or a message different from evSync event.

12.3.3.2 At the Receiver

- The Super-Frame header shall be detected by finding header bytes. The first byte of the Super-Frame header shall be equal to 10011111b and the next byte shall be 01111011b. The distance between these two bytes shall be equal to $8*n_i$ bits (n_i is related to the pair rate: n_i = rate /8 kbps). The following Super-Frame header bytes shall be at the same $8*n_i$ bits distance and shall complement all the Super-Frame header as specified in 6.2.
- The receiver is expecting to receive evSync events, with a value indicating the pair's allocation to an Aggregation Group, the Pair number, and Synchronization status (see 13.2.3). The Synchronization status indication indicates local synchronization or full synchronization.
- The procedure shall stop if there are Error values (i.e. the MSB is set to '1' in value[0]) that indicate problems in the pair synchronization, and wait for a management decision.

12.3.3.3 Fault handling

Any CRC error, in the Frame header (CRC-4) or in the evSync event (CRC-8), during the Sync to Group procedure shall restart the Sync to Group procedure and shall set the Synchronization status indication (Value[0]) with "no synchronization" value.

12.3.4 Pair Assignment to Aggregation Group

Pair Assignment to Aggregation Group procedure is initiated by the Management layer in order to assign a Pair to an Aggregation Group.

In order to start a Pair Assignment to Aggregation Group procedure the Pair shall be in the "Activation" state.

The Pair Assignment to Aggregation Group procedure includes setting Aggregation Group number and logical pair number, which are used by the Sync to Group procedure to synchronize the Pair to the Aggregation Group. After completing the Sync to Group procedure the pair is in "Synched to Group" state and the Management layer can add the pair to the Aggregation Group using Sync Change procedure. By the end of Sync Change procedure the pair is in "Part of group" state.

12.3.5 Pair Synchronization Loss

A Pair Synchronization Loss procedure is starting upon receiving 10 (ten) consecutive Frames with error (bad CRC-4 or wrong SF bit). This criteria is the same at the BTU-C and at the BTU-R, and both the BTU-C and the BTU-R can initiate the procedure.

Upon receiving 10 (ten) consecutive Frames with error (bad CRC-4 or wrong SF bit), the state of the pair shall change to "Lost Sync to Group" state.

Every pair that is in a "Lost Sync to Group" state (in the BTU-C and in the BTU-R) shall transmit "all ones". The "all ones" pattern shall replace all the payload of the pair including the Framing header. This transmission pattern forces the FE to also initiate Pair Synchronization Loss procedure, if not already started.

Upon declaring a pair (or pairs) as "Lost Sync to Group", the BTU-C shall start the Fast Change procedure to remove these pairs from the Aggregation Group. When the Fast Change procedure is successfully completed, the Management layer should decide whether to change the state of these pairs to "Down" state or to "Synching to Group" state.

Note: the Pair Synchronization Loss procedure should be used to quickly recover from pair cut-off, and not to handle pair removal due to high BER. Pairs with high BER should be removed from the Aggregation Group using the Sync Change procedure.

13 TDIM Bonding Communication Channel (BCC)

13.1 Introduction

The Bonding Communication Channel (BCC) is similar in concept to Embedded Operations Channel (EOC). The BCC is carried in the framing overhead and has a constant B/W of 1 Byte per 2ms. The BCC is used in TDIM bonding to control various aspects of bonding by means of control communication channels between the peers.

There are two communication channels between the peers – high priority Events and low priority Messages. Both channels are using the data bits in the frame header Data[7:0], and are distinguished by the M/E indicator bit. For more details see 6.2.2.

13.2 Events

13.2.1 Introduction

Events are fixed size short datagrams of high priority, dedicated for a fast communication between the peers to handle modem loss, dispatching table switching, FEC and IL parameter switching, and Service configuration changes.

Events are always transmitted on all the pairs that belong to an Aggregation Group at the beginning of a SF.

An event can interrupt a message transmission by de-asserting the Indicator bit in the frame header. The M/E indicator bit shall be set to '0' at the beginning of the SF.

13.2.2 Format

An Event has a fixed length of 6 bytes, with the following format. Events start at the start of a SF.

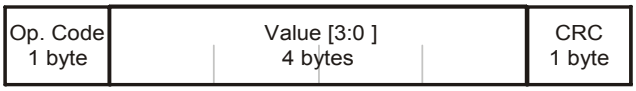

Figure 19- Event format

- Operation Code (Op. Code) - identifies the event, 1 Byte
- Value - contains the information that is being carried by the Event, 4 Bytes
- CRC - is used to validate the correctness of the Event, 1 Byte

13.2.3 Types and Values

The following table specifies the event types and the values in the various event fields:

Table 7- Event codes and values

Op. Code	Event Name	Value [3:0]	Description
0x00	evNull	0x00000000	Null event. Shall be transmitted whenever there is no other Event or Message to transmit.
0x01	evFastChange	Bitmap of the pairs that shall be part of the multipair aggregation group after the change. This bitmap relates to the pair map after the pair discovery stage, according to the pair numbering used in the Value[1] byte of evSync event value. The MSB is transmitted first (in Value[3]) and the LSB is transmitted last (in Value[0]).	Fast Change in the pairs that are part of the Aggregation Group.
0x02	evSyncChange	Bitmap of the pairs that shall be part of the multipair aggregation group after the change. This bitmap relates to the pair map after the pair discovery stage, according to the pair numbering used in the second byte of evSync event value. The MSB is transmitted first (in Value[3]) and the LSB is transmitted last (in Value[0]).	Synchronized (managed) Change in the pairs that are part of the Aggregation Group.

Op. Code	Event Name	Value [3:0]	Description
0x03	evConfigSw	The number of Super-Frames that shall be sent before switching to the new configuration.	Configuration switch indication. Specifies the number of Super-Frames that shall be sent to the peer before switching to a new configuration. The configuration includes: Dispatching Table Service Configuration FEC/Interleaver Configuration
0xFF	evSync	Value[3] – the value 0x5A. Value[2] – the number of the Aggregation Group. A value of 0xFF shall be used by the BTU-C when the pair is not assigned to an Aggregation Group, and by the BTU-R to indicate that a group number was not assigned to it yet. Value[1] – the number of the pair in the Aggregation Group. A value of 0xFF shall be used by the BTU-C when the pair is not assigned to an Aggregation Group, and by the BTU-R to indicate that a pair number was not assigned to it yet. Value[0] – Synchronization status indication, with the following values: 0x00 – no synchronization 0x01 – Near End is synchronized 0x02 – both Near End and Far End are synchronized 0x80 – The BTU-R already has different Aggregation Group number. This value shall be transmitted only by the BTU-R, in case that the Aggregation Group number received by this pair is different from the number already received by other pairs 0x81 – The BTU-R already has this pair number. This value shall be transmitted only by the BTU-R, in case that the Aggregation Group number received by this pair is the same as already received by other pairs and the pair number received by this pair is already used by another pair	Sync attempt. This event shall be sent during pair synchronization to the Aggregation Group. The value indicates whether the pair is already synchronized, the Aggregation Group number and the pair number. The payload data in this pair shall be repetitive 0xE2

13.2.3.1 CRC

An 8 bits CRC shall be generated for each event data (not including the CRC field) and shall be transmitted in the CRC field as specified in 13.2.2.

The encoding of the eight bits CRC field is defined by the generating polynomial $G(x) = x^8+x^7+x^2+1$ (which equals $(x^7+x+1)(x+1)$). This code can detect single, double and triple bit errors (and will fail only on 4 bit errors and up). Mathematically, the CRC value corresponding to a given event message is defined by the following procedure:

– The first 8 bits of the event message are complemented.

– The 40 bits of the event message are then considered to be the coefficients of a polynomial M(x) of degree 39. The first bit of the event message corresponds to the x^{39} term and the last bit of the event message corresponds to the x^0 term.

– M(x) is multiplied by x^8 and divided by G(x), resulting in a remainder R(x) of degree 7.

– The coefficients of R(x) are considered to be a 8 bit sequence.

– The bit sequence is complemented and the result is the CRC.

– The 8 bits of the CRC value are placed in the CRC field of the event so that the x^7 term is located in the left most bit of the CRC field, and the x^0 term is located in the right most bit of the CRC field. The bits of the CRC are thus transmitted in the order x^7 ,x^6 , … x^1 ,x^0.

13.3 Messages

13.3.1 Introduction

Messages are variable size datagrams of low priority, dedicated for a slow communication between the peers to handle non time-critical data, such as inventory, configuration, performance monitoring, maintenance etc. Messages are always transmitted on all the pairs that belong to an aggregation group. As Events, Messages are SF aligned and their total size is always multiple of 6 Bytes.

The total Message length is variable, between 6 and 126 (=6x21) Bytes, thus they can span multiple Super-Frames. The M/E indicator bit shall be set to ‘1’ at the beginning of each SF during an entire message transmission.

If the message is interrupted by an Event in the middle it shall be retransmitted.

13.3.2 Format

A Message has the following format:

Figure 20- Message format

- Length – specifies the length of the message body in bytes, 4..124
- Msg Body – contains the message information, with length in bytes specified by the Length field. Msg Body contains Msg ID (first byte) which determines format of the following data.
- CRC-8 - used to validate the correctness of the Message, 1 Byte.

13.3.3 Types and Content

The BCC Messages are summarized in the following table:

Table 8- BCC Messages

Msg ID	Length (MsgBody)	Message Type	Initiating Unit (BTU-C, BTU-R)	Description
0	4	Unable To Comply (UTC)	-C, -R	Responding unit was unable to comply with request
1	4	Inventory Request	-C, -R	Request TDIM Protocol version and Vendor ID
2	10	Inventory Response	-C, -R	TDIM Protocol version and Vendor ID response
3	4	Performance Monitor (PM)/ Statistics Request	-C, -R	Used to request / initialize all TDIM layer PM / Statistics
4	10	PM / Statistics Response	-C, -R	TDIM layer statistics (CRC-4/6/8 violations etc.)
5	Variable (4-124)	Service Mapping Request	-C, -R (Get only)	Get/Set mapping of Service interface types
6	Variable (4-124)	Service Mapping Response	-C, -R	Mapping of Service interface types
7	Variable (4-124)	TDM Service Configuration Request	-C, -R (Get only)	Get/Set TDM Service Configuration

Msg ID	Length (MsgBody)	Message Type	Initiating Unit (BTU-C, BTU-R)	Description
8	Variable (4-124)	TDM Service Configuration Response	-C, -R	TDM Service Configuration
9	Variable (4-124)	Asynchronous Service Configuration Request	-C, -R (Get only)	Get/Set Asynchronous Service Configuration
10	Variable (4-124)	Asynchronous Service Configuration Response	-C, -R	Asynchronous Service Configuration
11	4	FEC Capability Request	-C, -R (Get only)	Get/Enable/Disable FEC-IL
12	4	FEC Capability Response	-C, -R	FEC Capability response, includes FEC-IL parameters
13	4	Pair Mapping Request	-C, -R	Request for Logical to Physical pair mapping
14	Variable (4-67)	Pair Mapping Response	-C, -R	Logical to Physical pair mapping response
15-127		Reserved		Reserved
128-255		Vendor Specific		Reserved for Vendor Specific messages

13.3.4 Message Contents

Each message shall have the contents specified in the following clauses.

If any message has a message length longer than expected and is received in a frame with a valid CRC, then the known portion of the message shall be used and the extra octets discarded. This will permit addition of new fields to existing messages and maintain backward compatibility. New data fields shall only be placed in reserved bits after the last previously defined data octet.

Reserved bits and octets shall be filled with the value 0x0 for forward compatibility.

Response messages may indicate UTC (Unable to Comply). Note that this is not an indication of non-compliance. UTC indicates that the responding unit was unable to implement the request.

13.3.4.1 Unable To Comply

The Generic UTC message shall be sent back to the source unit in the event that the destination unit is unable to comply with the request. In this case, the definition of UTC is vendor dependent. Note that this message is not meant to replace the UTC bit in those response messages that contain a UTC bit.

Table 9 - Unable To Comply

Octet #	Content	Data Type	Description
1	MsgID = 0	Message ID	
2	MsgID of Request message	Message ID	
3-4	Reserved		

13.3.4.2 Inventory Request

Inventory Request message is used to query inventory information specific to the Aggregation layer at the Far End. The response to this request shall be message ID 2.

Table 10 - Inventory Request

Octet #	Content	Data Type	Description
1	MsgID = 1	Message ID	
2-4	Reserved		

13.3.4.3 Inventory Response

The Inventory Response message is sent in response to the Inventory Request Message (message ID 1). It reports inventory information specific to the Aggregation layer of the source BTU.

Table 11 - Inventory Response

Octet #	Content	Data Type	Description
1	MsgID = 2	Message ID	
2.4-7	Major Version	1-15	Major Version of the TDIM Aggregation Protocol. Current value=1.
2.0-3	Minor Version	0-15	Minor Version of the TDIM Aggregation Protocol. Current value=0.

Octet #	Content	Data Type	Description
3-10	Vendor ID	octet string	Vendor ID of the TDIM Aggregation layer (ordered identically to the Vendor ID in G.994.1)

13.3.4.4 PM / Statistics Request

PM/Statistics Request message is used to query the Aggregation layer related PMs/Statistics at the Far End. The response to this request shall be message ID 4.

Table 12 - PM / Statistics Request

Octet #	Content	Data Type	Description
1	MsgID = 3	Message ID	
2	Request	Unsigned Integer 8 bits (uint 8)	0 – Request PM / Statistics report 1 – Initialize all Far End PM / Statistics counters
3-4	Reserved		

13.3.4.5 PM / Statistics Response

The PM/Statistics Response message is sent in response to the PM/Statistics Request Message (message ID 3) or autonomously at BTU-R discretion. It reports statistics specific to the Aggregation layer of the source BTU.

Table 13 - PM / Statistics Response

Octet #	Content	Data Type	Description
1	MsgID = 4	Message ID	
2,3	CRC-4 Anomaly count	Unsigned Integer 16 bits (uint16)	Frame errors. Sum of CRC-4 errors on all pairs in the Aggregation Group, simultaneous errors on m lines shall be counted m times
4,5	CRC-6 Anomaly count	uint16	SF errors. Sum of all pairs in the Aggregation Group

Octet #	Content	Data Type	Description
6,7	CRC-8 Anomaly count	uint16	Number of corrupted Events / Messages, sum of all pairs in the Aggregation Group
8-10	Reserved		

13.3.4.6 Service Mapping Request

Service Mapping Request message allows a BTU to query and control how Service interfaces of the Far End unit are mapped (by the Service Mux (Multiplexer)) into the Bonding Sub-Blocks of the aggregated link. The response to this request shall be message ID 6.

Table 14 - Service Mapping Request

Octet #	Content	Data Type	Description
1	MsgID = 5	Message ID	Request the Far End to send its Service mapping
2.7	Request	bit (0-1)	0 – Request report 1 – Set (BTU-C only)
2.0-6	Reserved		
3	Number of Services (NS)	uint 8	NS = 0..60
4	Service type notation for service 1 (j=1 in 10.2)	uint 8	Applicable for Set only. Defines which Service type to map into service 1 (highest priority)
5	Service type notation for service 2 (j=2 in 10.2)	uint 8	Applicable for Set only. Defines which Service type to map into service 2 (2^{nd} highest priority)
. . .			
NS+3	Service type notation for service NS (j=NS in 10.2)	uint 8	Applicable for Set only. Defines which type to map into service NS (lowest priority)

Note that the Service Mapping is setting the service priority. Service Mapping Request message with Request value 1 (Set) shall always be followed by the evConfigSw event, which provides synchronized change of the configuration at both the BTU-C and the BTU–R.

The Service type notations values are:

1 Clear Channel DS1

2 Clear Channel E1

3 Fractional DS1

4 Fractional E1

5 DS3

6 E3

7 Clock transfer

8 Ethernet

9 ATM

10 GFP

13.3.4.7 Service Mapping Response

The Service Mapping Response message is sent in response to a Service Mapping Request Message (message ID 5). It reports which Service interfaces of the source BTU are mapped into the services in the aggregated link.

Table 15 - Service Mapping Response

Octet #	Content	Data Type	Description
1	MsgID = 6	Message ID	
2	Number of services (NS)	uint8	NS = 0..60
3	Service type notation for service 1 (j=1 in 10.2)	uint 8	Defines which Service type maps into service 1 (highest priority)
4	Service type notation for service 2 (j=2 in 10.2)	uint 8	Defines which Service type maps into service 2 (2^{nd} highest priority)
. . .			
NS+2	Service type notation for service NS (j=NS in 10.2)	uint 8	Defines which Service type maps into service NS (lowest priority)

Current values for all parameters shall be returned in response to Service Mapping Request with Request value 1 (Set), after they have been set to the requested values by evConfigSw event.

13.3.4.8 TDM Service Configuration Request

TDM Service Configuration Request message allows a BTU to query and control the configuration of TDM services, if additional information is required to "Service Mapping Request" message. The response to this request shall be message ID 8.

Table 16 - TDM Service Configuration Request

Octet #	Content	Data Type	Description
1	MsgID = 7	Message ID	Request the Far End to send its TDM Services Configuration
2.7	Request	bit (0-1)	0 – Request report 1 – Set (BTU-C only)
2.0-6	Reserved		
3	Number of Services (NS)	uint 8	NS = 0..60
4	P_1	uint 8	Number of sub-channels (P, see 10.2) if service 1 is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0
5	P_2	uint 8	Number of sub-channels (P, see 10.2) if service 2 is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0
. . .	. . .		
NS+3	P_{NS}	uint 8	Number of sub-channels (P, see 10.2) if service NS is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0

TDM Service Configuration Request message with Request value 1 (Set) shall always be followed by the evConfigSw event, which provides synchronized change of the configuration at both BTU-C and BTU–R.

13.3.4.9 TDM Service Configuration Response

The TDM Service Configuration Response message is sent in response to a TDM Service Configuration Request Message (message ID 7). It reports the configuration of TDM services with addition information to "Service Mapping Request" message.

Table 17 - TDM Service Configuration Response

Octet #	Content	Data Type	Description
1	MsgID = 8	Message ID	Service Configuration message
2.7	Response	bit (0-1)	0 – OK 1 – UTC
2.0-6	Reserved		
3	Number of Services (NS)	uint8	NS = 0..60
4	P_1	uint 8	Number of sub-channels (P, see 10.2) if service 1 is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0
5	P_2	uint 8	Number of sub-channels (P, see 10.2) if service 2 is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0
. . .	. . .		
NS+3	P_{NS}	uint 8	Number of sub-channels (P, see 10.2) if service NS is of Service type notations 3 (Fractional DS1) or 4 (Fractional E1), otherwise 0

Current values for all parameters shall be returned in response to TDM Service Configuration Request message with Request value 1 (Set), after they have been set to the requested values by an evConfigSw event.

13.3.4.10 Asynchronous Service Configuration Request

Asynchronous Service Configuration Request message allows a BTU to query and control the configuration of Asynchronous services, if additional information is required to "Service Mapping Request" message. The response to this request shall be message ID 10.

Table 18 - Asynchronous Service Configuration Request

Octet #	Content	Data Type	Description
1	MsgID = 9	Message ID	Request the Far End to send its Asynchronous Services Configuration
2.7	Request	bit (0-1)	0 – Request report 1 – Set (BTU-C only)

Octet #	Content	Data Type	Description
2.6	Optional FCS for GFP encapsulation	bit (0-1)	0 – None 1 – Use FCS
2.0-5	Reserved		
3	Number of Services (NS)	uint8	NS = 0..60
4	N_1	uint 8	Maximal number of service 1 bytes in Bonding Sub-Block of 125 usec if service 1 is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0
5	N_2	uint 8	Maximal number of service 2 bytes in Bonding Sub-Block of 125 usec if service 2 is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0
. . .	. . .		
NS+3	N_{NS}	uint 8	Maximal number of service NS bytes in Bonding Sub-Block of 125 usec if service NS is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0

Asynchronous Service Configuration Request message with Request value 1 (Set) shall always be followed by the evConfigSw event, which provides synchronized change of the configuration at both BTU-C and BTU–R.

13.3.4.11 Asynchronous Service Configuration Response

The Asynchronous Service Configuration Response message is sent in response to a Asynchronous Service Configuration Request Message (message ID 9). It reports the configuration of Asynchronous services with addition information to "Service Mapping Request" message.

Table 19 - Asynchronous Service Configuration Response

Octet #	Content	Data Type	Description
1	MsgID = 10	Message ID	Service Configuration message
2.7	Response	bit (0-1)	0 – OK 1 – UTC

Octet #	Content	Data Type	Description
2.6	Optional FCS for GFP encapsulation	bit (0-1)	0 – FCS not used 1 – FCS is used
2.0-5	Reserved		
3	Number of Services (NS)	uint8	NS = 0..60
4	N_1	uint 8	Maximal number of service 1 bytes in Bonding Sub-Block of 125 usec if service 1 is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0
5	N_2	uint 8	Maximal number of service 2 bytes in Bonding Sub-Block of 125 usec if service 2 is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0
. . .	. . .		
NS+3	N_{NS}	uint 8	Maximal number of service NS bytes in Bonding Sub-Block of 125 usec if service NS is of Service type notations 8 (Ethernet) or 9 (ATM) or 10 (GFP), otherwise 0

Current values for all parameters shall be returned in response to Asynchronous Service Configuration Request with Request value 1 (Set), after they have been set to the requested values by an evConfigSw event.

13.3.4.12 FEC Capability Request

The FEC / Interleaver capability is reported and negotiated during service setup via BCC messages. FEC Capability Request is used to query and control FEC capability at the Far End. The response to this request shall be message ID 12.

FEC Capability Request message with Disable or Enable Request values shall always be followed by the evConfigSw event, which provides synchronized change of the configuration at both BTU-C and BTU–R. The FEC Capability response message shall be sent by the BTU-R only after reception of evConfigSw.

Table 20 - FEC Capability Request

Octet #	Content	Data Type	Description
1	MsgID = 11	Message ID	
2.6-7	Request	Enumerated (enum) (0-2)	0 – Request FEC Capability report 1 – Disable FEC (BTU-C only) 2 – Enable FEC (BTU-C only)
2.3-5	Reserved		
2.0-2	Redundancy Word size (2, 4, 8, 16, 20)	enum (1-4)	FEC Redundancy Word size (2^k for k = 1..4, 20 for k = 5). If Request value is 0 or 1 (Request/Disable) the value is ignored.
3	Code Word size (in octets)	enum (20-255)	FEC Code Word size. If Request value is 0 or 1 (Request/Disable) the value is ignored.
4.6-7	Reserved		
4.3-5	Interleaver Param B	enum (0-5)	Interleaver Param B. The Interleaver Depth = $3^A * 2^B$. If Request value is 0 or 1 (Request/Disable) the value is ignored.
4.2	Interleaver Param A	enum (0-1)	Interleaver Param A. If Request value is 0 or 1 (Request/Disable) the value is ignored.
4.0-1	Interleaver Type	enum (0-2)	0 – no Interleaver 1 – Block Interleaver 2 – Convolution Interleaver

13.3.4.13 FEC Capability Response

The FEC Capability Response message is sent in response to a FEC Capability Request message (message ID 11). It reports FEC and Interleaver parameters for the source BTU.

Table 21 - FEC Capability Response

Octet #	Content	Data Type	Description
1	MsgID = 12	Message ID	
2.7	Response	enum (0-1)	0 – OK 1 – UTC (Unable to Comply)

Octet #	Content	Data Type	Description
2.3-6	Reserved		
2.0-2	Redundancy Word size (2, 4, 8, 16, 20)	enum (1-4)	Current/Maximum supported Redundancy word size (2^k for k = 1..4, 20 for k = 5). If FEC is not supported the value shall be 0. When FEC is disabled – Maximum possible value is returned on Request. When FEC is enabled – current value is returned on Request.
3	Code Word size (in octets)	enum (20-255)	Current/Maximum supported FEC Code Word size. If FEC is not supported the value shall be 0. When FEC is disabled – Maximum possible value is returned on Request. When FEC is enabled – current value is returned on Request.
4.6-7	Reserved		
4.3-5	Interleaver Param B	enum (0-5)	Current/Maximum supported Interleaver Param B. The Interleaver Depth = $3^A * 2^B$. If Interleaver is not supported the value shall be 0. When Interleaver is disabled – Maximum possible value is returned on Request. When Interleaver is enabled – current value is returned on Request.
4.2	Interleaver Param A	enum (0-1)	Current/Maximum supported Interleaver Param A. If Interleaver is not supported the value shall be 0. When Interleaver is disabled – Maximum possible value is returned on Request. When Interleaver is enabled – current value is returned on Request.
4.0-1	Interleaver type	enum (0-3)	00 – none supported 01 – Block interleaver supported 10 – Convolution Interleaver supported 11 – Both types supported

If BTU-R does not support FEC, it shall respond with UTC value in Response field on FEC Request from BTU-C to Enable FEC. When FEC is disabled, Max possible values for all parameters shall be returned in response to FEC Request message. Current values for all parameters shall be returned in response to FEC Request to Enable FEC, after they have been set to the requested values by evConfigSw event.

13.3.4.14 Pair Mapping Request

The Pair Mapping Request message is used to determine the mapping between the physical pair (or loop) number labeled on the equipment and the logical wire pair (or loop) ordinal number. While this mapping is vendor specific, this information is useful for troubleshooting circuits. The response to this request shall be message ID 14.

Table 22 - Pair Mapping Request

Octet #	Content	Data Type	Description
1	MsgID = 13	Message ID	
2-4	Reserved		

13.3.4.15 Pair Mapping Response

The Pair Mapping Response message is sent in response to a Mapping Request Message (message ID 13). It is used to determine the mapping between the physical pair (or loop) number and the logical wire pair (or loop) ordinal number. The physical pair number is the number labeled externally on the equipment. The physical pair number is composed of two octets, with the first octet containing the most significant byte, and the second octet containing the least significant byte. For example, if the 16-bit number in octets 3/4 contains the value 4, then logical wire pair 1 in the current group is transported over the equipment's physical pair labeled number 4.

Table 23 - Pair Mapping Response

Octet #	Content	Data Type	Description
1	MsgID = 14	Message ID	
2	Number of Pairs = M	uint8 (1..32)	Number of wire Pairs in the Aggregation Group
3,4	Physical Pair number 1	uint16	Physical number of the pair corresponding to the 1st logical pair in the Aggregation Group
5,6	Physical Pair number 2	uint16	Physical number of the pair corresponding to the 2nd logical pair in the Aggregation Group

Octet #	Content	Data Type	Description
. . .	. . .		
2xM+1,2xM+2	Physical Pair number M	uint16	Physical number of the pair corresponding to the M^{th} logical pair in the Aggregation Group

13.3.4.16 Reserved

The Message IDs 15-127 are reserved.

13.3.4.17 Vendor specific Messages

The Message IDs 128-255 are reserved for Vendor specific (proprietary) messages.

14 Handshaking

14.1 Overview

All of the Bonding system pairs (modems) shall use G.Handshake as defined in ITU-T G.994.1 for startup and choose Synchronous Transfer Mode (STM) mode for each of the pairs. G. Handshake negotiation and modem parameter selection shall be done per modem technology.

The Institute of Electrical and Electronics Engineers (IEEE) 802.3ah standard defines procedure for Pair Discovery. The same discovery shall be used in TDIM bonding, using Physical Medium Independent (PMI) Discovery codepoints defined in G.994.1.

14.2 Bonding Npar(2) codepoint

ITU-T G.994.1 defines Bonding Npar(2) coding in the Identification field (Table 9.37). The TDIM system shall set the "TDIM Bonding" bit to one in the CLR/CL handshake messages on all modems.

14.3 Pair Discovery

IEEE 802.3ah defines process and handshake transactions for Pair Discovery, see 61.2.2.8.3, 61.3.12.1, 61.A2. A TDIM system shall use the same Pair Discovery process and transactions to identify Far End modems, which belong to the same Aggregation Group, by incorporating IEEE 802.3ah standard by reference.

The use of handshake transactions for Pair Discovery provides means for detecting inconsistencies without a need for a full wakeup of the system. However, once the system has initiated, the Pair Mapping messages (see 13.3.4.14, 13.3.4.15) shall be used and shall take precedence over the handshake Pair Discovery.

15 Performance Monitoring

The following PM counters are used in order to monitor the performance of the Aggregation Group:

Table 24- Performance Monitoring counters

PM register	Data Type	Description
CRC-4 Anomaly count	uint16	Frame errors. Sum of CRC-4 errors on all pairs in the Aggregation Group, simultaneous errors on m lines shall be counted m times
CRC-6 Anomaly count	uint16	Super-Frame errors. Sum of all pairs in the Aggregation Group
CRC-8 Anomaly count	uint16	Number of corrupted Events / Messages, sum of all pairs in the Aggregation Group

All the PM counters are "Cleared on Read" and are "stuck" when reaching maximum value (65535).

Annex A: Modem Rate Matching

A.1 Introduction

This annex provides implementation details for the Modem Rate Matching block, which allows bonding over pairs (modems) that are not synchronized to the Aggregation time domain.

The purpose of this annex is to provide a solution for multi loop bonding with pairs with rate that deviates from the nominal n x 8 kbps. This deviation can be derived by either of the following two causes:

1. The DSL pair is asynchronous to the Aggregation time domain.
2. The DSL pair nominal frequency is different from the n x 8 kbps nominal rate, since the DSL technology does not guarantee integer multiples of 8 kbps (e.g. ADSL2).

With Modem Rate Matching, Figure 1 and Figure 6 shall be as described below:

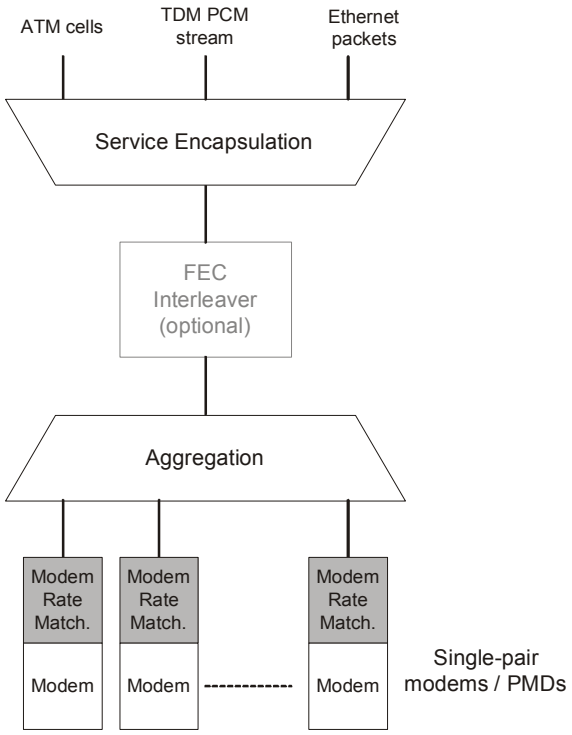

Figure 1a: Data flow model

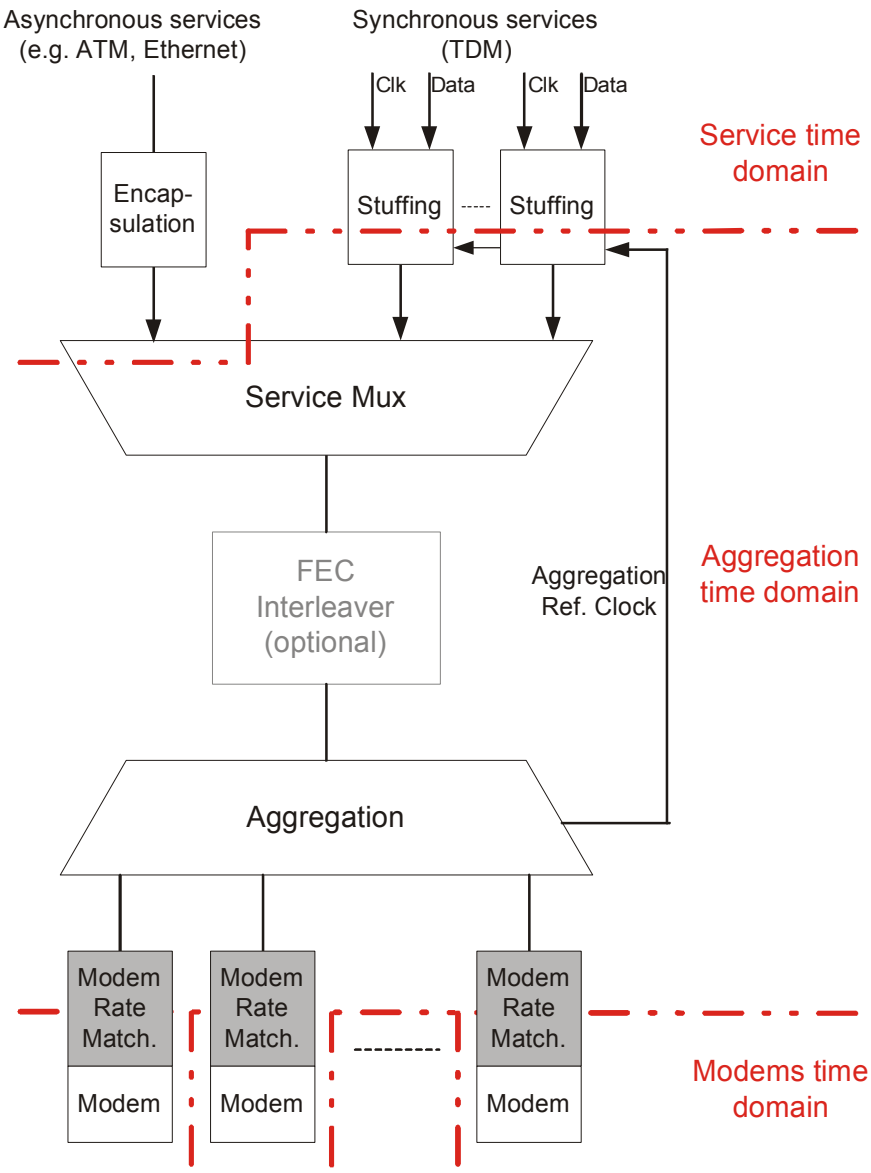

Figure 6a: Clock synchronization reference model

ADSL2 rate cannot be predetermined to provide a fixed rate which is a multiple of 8 kbps. The rate may exceed the desired target by up to 8 kbps. To allow such operation, Modem Rate Matching shall provide a mechanism to add up to 8 kbps.

Additional adaptation of ±8 kbps is provided to overcome frequency deviation (due to the DSL link being derived from a clock which is asynchronous to the Aggregation time domain). ±8 kbps can compensate for frequency deviation of above 70ppm in case of 55.2 megabits per second (mbps) pair (VDSL). Thus, the Modem Rate Matching is defined to compensate for a total deviation of –8 kbps to +16 kbps.

A.2 Work principles

- The Modem Rate Matching mechanism is enabled / disabled for all the Aggregation Group, and not per specific Pairs.
- Modem Rate Matching overhead occupies 8 kbps on each pair.
- Modem payload size granularity is in bytes.
- Modem payload size can change every Bonding Sub-block (1ms).
- Modem payload size change can be done per pair.
- The BTU-R conveys its capabilities over the Frame header bits In6[4:3], and the BTU-C determines the operation mode over those bits.
- The Modem Rate Matching can compensate for total deviation of –8 kbps to +16 kbps.

A.3 Coordination between BTU-C and BTU-R

The use of the Modem Rate Matching is determined by the BTU-C. The BTU-R conveys its capabilities over the Frame header bits In6[4:3], and the BTU-C responds with the required operation mode. Both ends shall operate in the same mode.

Table 25 summarizes the possible options for each BTU:

Table 25: Modem Rate Matching coordination between BTU-C and BTU-R

In6[4]	In6[3]	Transmitting entity	
		BTU-R	**BTU-C**
0	1	Only Modem Rate Matching is supported. In this case both ends shall support it to perform the bonding.	Set the operation mode of both ends to Modem Rate Matching enabled.
1	0	Modem Rate Matching is not supported. In this case both ends shall operate without it.	Set the operation mode of both ends to Modem Rate Matching disabled.
1	1	Modem Rate Matching can be enabled or disabled.	N/A

A.4 Framing format

Modem Rate Matching occupies an additional overhead of 8 kbps per DSL pair. This overhead not consumed if Modem Rate Matching is disabled.

If Modem Rate Matching is enabled, the overhead byte (RM-byte) is added every Mini-Frame (1ms), following the Frame Header field.

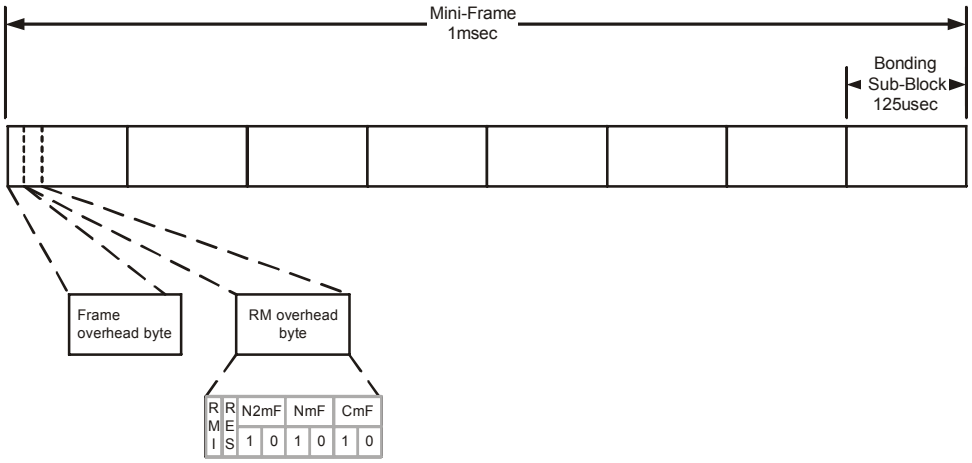

Figure 21: Multi-pair synchronization – frame format with Modem Rate Matching

The fields in the RM overhead byte are defined in Table 26:

Table 26: Fields in the RM overhead byte

Field	Bits	Description
RMI	7	Modem Rate Matching overhead Indication – set to '0'
RES	6	Reserved for future use – set to '1'
N2mF	5:4	The operation to be performed 2 Mini-Frames from the current Mini-Frame.
NmF	3:2	The operation to be performed in the next Mini-Frame.
CmF	1:0	The operation to be performed in the current Mini-Frame.

Table 27 defines the values used in fields N2mF, NmF and CmF.

Table 27 - Values in fields N2mF, NmF and CmF

Value	Description
00	A byte shall be omitted at the end of the relevant Mini-Frame.
01	No change to the nominal number of bytes of the relevant Mini-Frame.
10	A byte shall be added at the end of the relevant Mini-Frame.
11	2 bytes shall be added at the end of the relevant Mini-Frame.

The BTU-C and the BTU-R coordinate the Modem Rate Matching mode of operation as described in section A.3. If Modem Rate Matching is enabled, the byte following the Frame Header overhead byte shall be an RM overhead byte, with its MSB set to '0' to verify that (while the payload data is set to 0xE2 during the synchronization phase).

A.5 Modem Rate Matching operation

In each Mini-Frame the Modem Rate Matching mechanism either omits a byte, adds a byte, adds 2 bytes, or transfers the nominal number of bytes unchanged. The byte is added / omitted to / from the last Bonding Sub-Block of the Mini-Frame.

For each Mini-Frame the values used in fields N2mF, NmF and CmF, that determine the required operation, are conveyed 3 times: The value in field N2mF indicates the operation to be performed 2 Mini-Frames from the current Mini-Frame, the value in field NmF indicates the operation to be performed in the next Mini-Frame, the value in field CmF indicates the operation to be performed in the current Mini-Frame. The receiver uses a "majority vote" to decide on the required operation. Thus a single error which effects one of the fields can be corrected.

Appendix I: Clock synchronization examples

(Informative)

The following text and figures are informative. Figure 22 below elaborates the clock synchronization mechanism in more details.

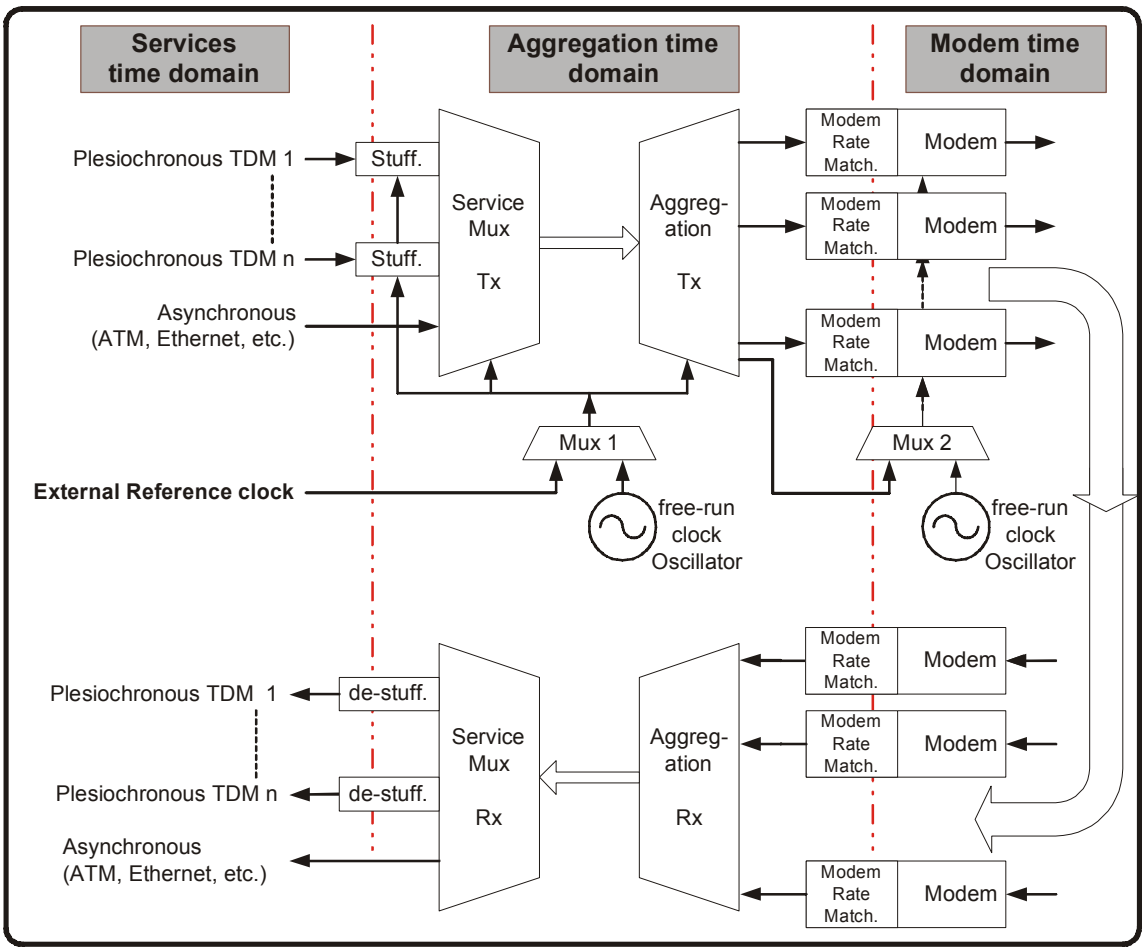

Figure 22: Peer time domains and clock synchronization

I.1 Clock domains

There are 3 clock domains in a TDIM bonding system:

- **Services clock domain:** with the original clock of each TDM plesiochronous service. Each clock should comply with the TDM service clock requirements (e.g. 32PPM for T1, 20 PPM for T3).
- **Aggregation time domain:** can be derived from the External reference clock or from an Internal free-run clock (by Mux 1).
- **Modem time domain:** DSL Modems may work with two clock modes: Synchronous and Plesiochronous. The clock can be derived from an Internal free-run clock or from the Aggregation time domain (by Mux 2). In Synchronous mode all modems use the modem data rate clock as reference for their line symbol rate. In Plesiochronous mode each modem

can use a free-run clock as reference for its line symbol rate. Modem rate matching is required in Plesiochronous mode. Modem rate matching can be done internally in the modem for DSL technologies that support it (e.g. SHDSL) or as part of the bonding system, externally to the modem, for DSL technologies that work only in synchronous mode (e.g. ADSL2).

I.2 Rate adaptation between clock domains

I.2.1 Between Services time domain and Aggregation time domain

Rate adaptation between Services time domain and Aggregation time domain is always implemented.

Rate adaptation is not required for asynchronous services such as Ethernet or ATM.

Rate adaptation is required for TDM Plesiochronous services. It is achieved by the stuffing mechanism explained in [10.4 TDM service stuffing]. The stuffing mechanism adapts the data rate of the each TDM service to the data rate of the Aggregation time domain. The peer system at the FE de-stuffs the data and restores the original TDM service clock.

I.2.2 Between Aggregation time domain and Modem time domain

Rate adaptation between Aggregation time domain and Modem time domain is optional.

In case that the DSL modems support Plesiochronous mode (such as G.SHDSL), rate adaptation is performed by the modems with their stuffing capability. In case that the DSL modems support only synchronous mode (such as ADSL2) rate adaptation between Aggregation time domain and Modem time domain is done in the Modem Rate Matching block (see Figure 22.)

I.3 Timing operation modes

A bonding system may choose to work with Loop timing or with Through timing as seen in

Figure 23.

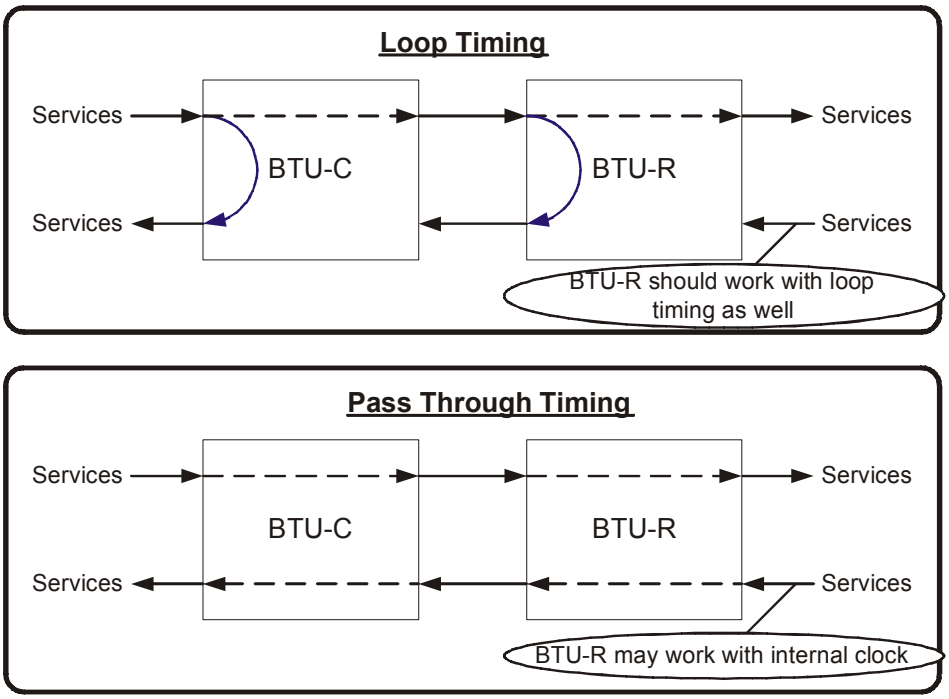

Figure 23: Loop timing and Through timing

I.4 Examples

I.4.1 Plesiochronous services, External reference clock, Loop timing and Plesiochronous DSL modems

In this example Modem Rate Matching is part of the Plesiochronous DSL modems.

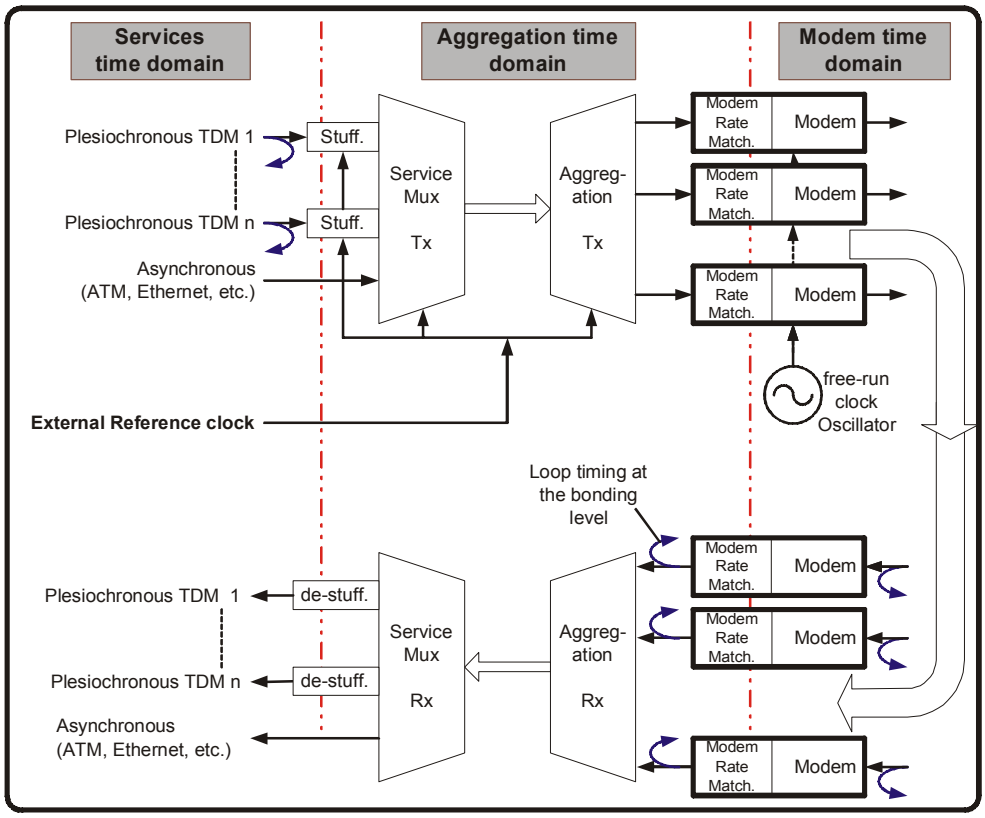

Figure 24

I.4.2 Plesiochronous services, no External reference clock, Loop timing and Synchronous DSL modems

In this example Modem Rate Matching is part of the bonding system, external to the Synchronous DSL modems.

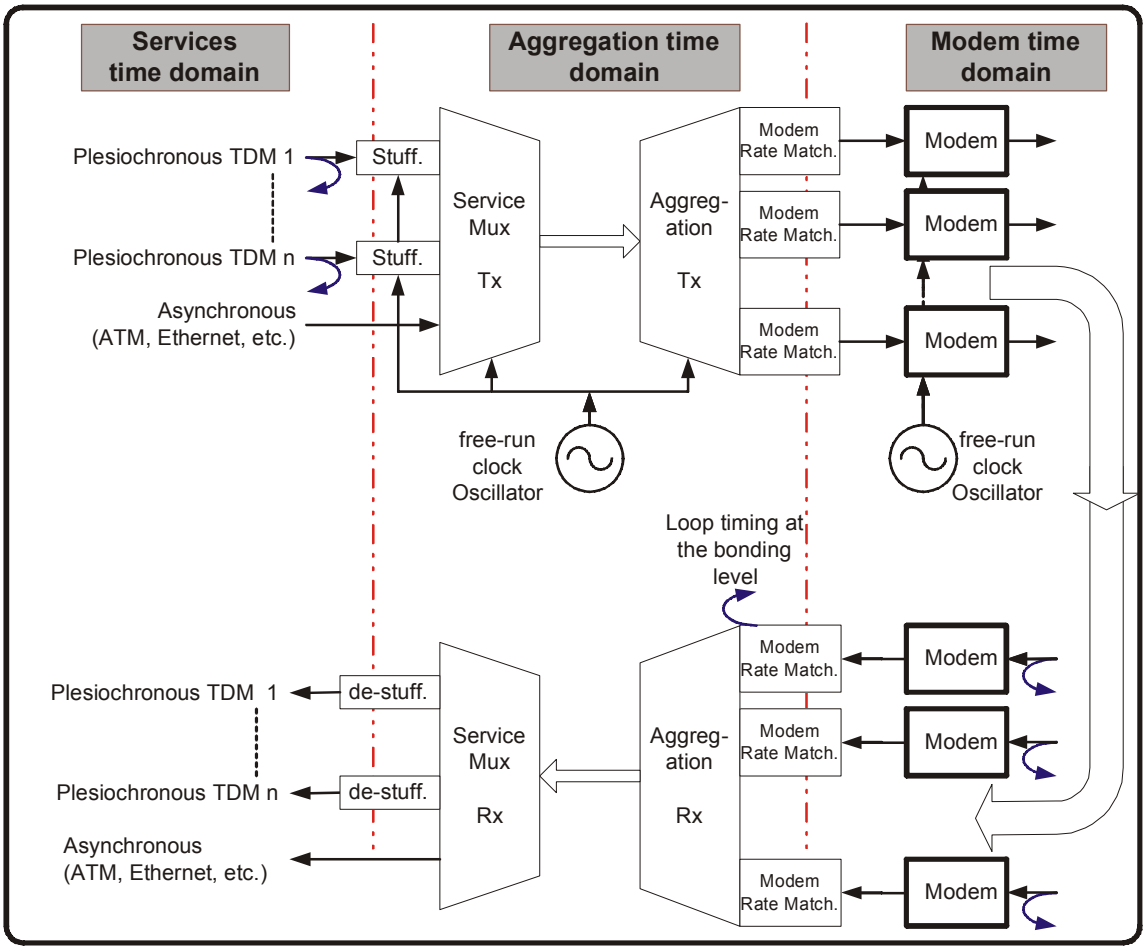

Figure 25: Example of Plesiochronous services, no External reference clock, Loop timing and Synchronous DSL modems

I.4.3 Plesiochronous services, External reference clock, Through timing and Plesiochronous DSL modems

In this example Modem Rate Matching is part of the Plesiochronous DSL modems.

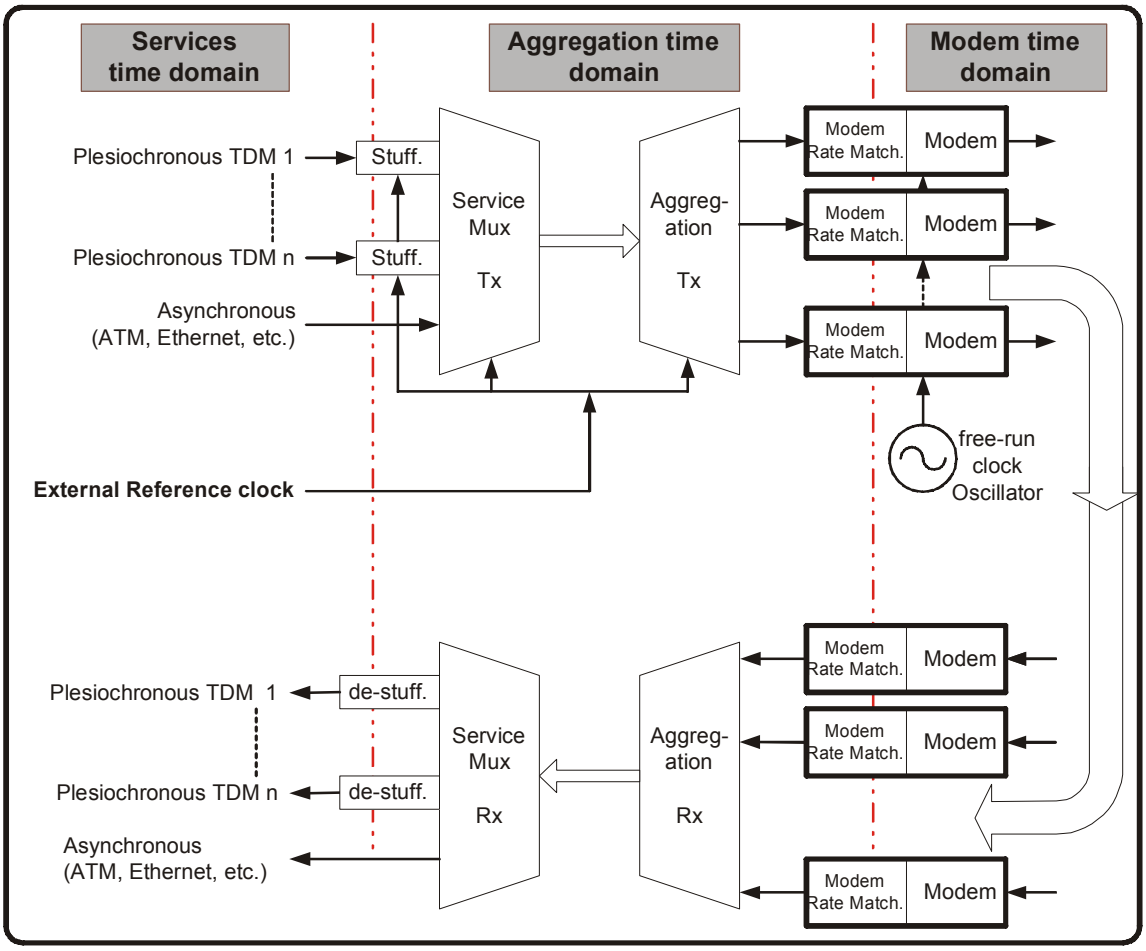

Figure 26: Example of Plesiochronous services, External reference clock, Through timing and Plesiochronous DSL modems

I.4.4 Plesiochronous services, no External reference clock, Through timing and Synchronous DSL modems

In this example Modem Rate Matching is not required, since the DSL modems are synchronous to the Aggregation layer rate allocation and to the Aggregation time domain. This example is applicable for SHDSL modems and not to ADSL2 modems.

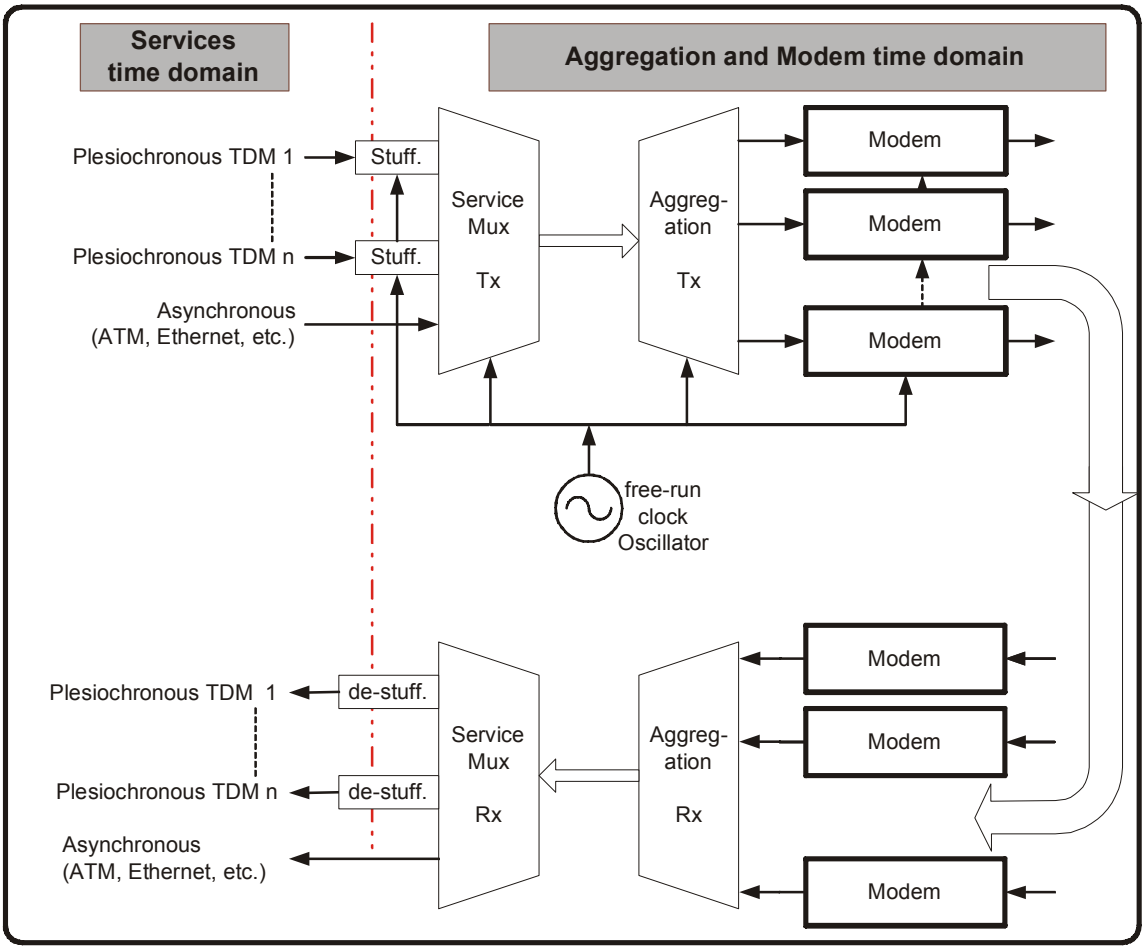

Figure 27: Example of Plesiochronous services, no External reference clock, Through timing and Synchronous DSL modems

Appendix II: Management Objects

(Informative)

This informative Annex provides the Layer Management specification for M^2DSL TDIM functionality of devices implementing this Recommendation. It includes Aggregation Group provisioning, Service Provisioning, FEC/Interleaving provisioning, Group Performance and Pair Status.

a. Management Model and Containment

i. Managed objects

The following objects provide management functionality for the TDIM protocol:

oGroup - This managed object class provides the management controls necessary to allow an instance of a bonded Group to be managed.

oService - this managed object class provides the management controls necessary to allow an instance of a Service in the bonded Group to be managed.

oPair - This managed object class provides the management controls necessary to allow an instance of a pair to be managed.

ii. Capabilities

This Recommendation makes use of the concept of *packages* as defined in ISO/IEC 10165-4: 1992 as a means of grouping behaviour, attributes, actions, and notifications within a managed object class definition. Packages may either be mandatory, or be conditional, that is to say, present if a given condition is true. Within this Recommendation *capabilities* are defined, each of which corresponds to a set of packages, which are components of a number of managed object class definitions and which share the same condition for presence. Implementation of the appropriate basic and mandatory packages is the minimum requirement for claiming conformance to G.Bond TDIM Management. Implementation of an entire optional capability is required in order to claim conformance to that capability. The capabilities and packages for G.Bond TDIM Management are specified in Table 1 below:

Table 28 - TDIM Management Capabilities

			TDIM Bonding Capability (Mandatory)	FEC Capability (Optional)
oGroup managed object class				
aGroupID	ATTRIBUTE	GET	x	
aGroupEnd	ATTRIBUTE	GET	x	
aGroupStatus	ATTRIBUTE	GET	x	
aGroupCapacity	ATTRIBUTE	GET	x	

aGroupRate	ATTRIBUTE	GET	x	
aCRC4Errors	ATTRIBUTE	GET	x	
aCRC6Errors	ATTRIBUTE	GET	x	
aCRC8Errors	ATTRIBUTE	GET	x	
aFECSupported	ATTRIBUTE	GET	x	
aFECAdminState	ATTRIBUTE	GET-SET		x
aFECWordSize	ATTRIBUTE	GET-SET		x
aFECRedundancySize	ATTRIBUTE	GET-SET		x
aFECInterleaverType	ATTRIBUTE	GET-SET		x
aFECInterleaverDepth	ATTRIBUTE	GET-SET		x
oService managed object class				
aServiceID	ATTRIBUTE	GET	x	
aServiceType	ATTRIBUTE	GET-SET	x	
aServiceSize	ATTRIBUTE	GET-SET	x	
oPair managed object class				
aPairID	ATTRIBUTE	GET	x	
aPairStatus	ATTRIBUTE	GET	x	
aPairPhysicalID	ATTRIBUTE	GET	x	
aPairRemotePhysicalID	ATTRIBUTE	GET	x	

b. oGroup managed object class

i. aGroupID

ATTRIBUTE

SYNTAX:

INTEGER

DESCRIPTION:

A read-only value to uniquely identify a bonding group.

ii. aGroupEnd

ATTRIBUTE

SYNTAX:

ENUM {subscriber, office}

DESCRIPTION:

A read-only value to uniquely identify a sub-type of the BTU. The value of subscriber indicates that the BTU operates as BTU-R, the value of office indicates that the BTU operates as BTU-C.

iii. aGroupStatus

ATTRIBUTE

SYNTAX:

ENUM {Down, Init, Up}

DESCRIPTION:

A read-only value indicating current operation status of a bonding group.

iv. aGroupCapacity

ATTRIBUTE

SYNTAX:

INTEGER {1-32}

DESCRIPTION:

A read-only value to specifying a maximum number of PAIRs (modems) which can be grouped in the group identified by aGroupID.

v. aGroupRate

ATTRIBUTE

SYNTAX:

INTEGER

DESCRIPTION:

A read-only value indicating current net data rate of a bonding group. A zero value is returned if aGroupStatus is "Down" or "Init".

vi. aFECSupported

ATTRIBUTE

SYNTAX:

BOOLEAN

DESCRIPTION:

A read-only value indicating if the optional Forward Error Correction (FEC) functionality is supported. A BTU that can perform FEC shall return the value of "true". Otherwise a "false" value shall be returned.

vii. aCRC4Errors

ATTRIBUTE

SYNTAX:

Generalized non-resettable counter.

DESCRIPTION:

The total number of CRC-4 errors (frame header error) on all pairs in the Aggregation Group, simultaneous errors on M lines shall be counted M times.

viii. aCRC6Errors

ATTRIBUTE

SYNTAX:

Generalized non-resettable counter.

DESCRIPTION:

The total number of CRC-6 errors (Super-Frame error) of all pairs in the Aggregation Group, simultaneous errors on M pairs shall be counted 1 time.

ix. aCRC8Errors

ATTRIBUTE

SYNTAX:

Generalized non-resettable counter.

DESCRIPTION:

The total number of CRC-8 errors (Event/Message error) of all pairs in the Aggregation Group, simultaneous errors on M lines shall be counted M times.

x. aFECAdminState

ATTRIBUTE

SYNTAX:

ENUM { Enabled, Disabled }

DESCRIPTION:

A read-write value that indicates the state of the optional FEC function per aggregation group.

A GET operation returns the current state of the FEC function.

A SET operation, allowed on BTU-C only, changes the state of FEC function to the indicated value only if aFECSupported is true and the link is down. If aFECSupported is false or the link is not down the operation has no effect.

xi. aFECWordSize

ATTRIBUTE

SYNTAX:

INTEGER {20-255}

DESCRIPTION:

A read-write value specifying FEC code word size in octets.

A GET operation returns the current value of FEC code word size if aFECAdminState is enabled. Otherwise a maximum supported FEC code word size value is returned.

SET operation, allowed on BTU-C only, changes the FEC code word size to the indicated value only if aFECSupported is true, the link is down. If aFECSupported is false or the link is not down, the operation has no effect.

xii. aFECRedundancySize

ATTRIBUTE

SYNTAX:

INTEGER {2, 4, 8, 16, 20}

DESCRIPTION:

A read-write value specifying FEC Redundancy word size in octets.

A GET operation returns the current value of FEC Redundancy word size if aFECAdminState is enabled. Otherwise a maximum supported FEC Redundancy word size value is returned.

SET operation, allowed on BTU-C only, changes the FEC Redundancy word size to the indicated value only if aFECSupported is true, the link is down. If aFECSupported is false or the link is not down, the operation has no effect.

xiii. aFECInterleaverType

ATTRIBUTE

SYNTAX:

```
ENUM {
    None,
    Block,
    Convolution
}
```

DESCRIPTION:

A read-write value specifying the type of interleaver.

A GET operation returns the current value of interleaver type if aFECAdminState is enabled. Otherwise a bitmask consisting of maximum supported types is returned (e.g. "None", "Block", "Convolution" or "Block"&"Convolution").

A SET operation, allowed on BTU-C only, changes the interleaver type to the indicated value only if aFECSupported is true and the link is down. If aFECSupported is false or the link is not down the operation has no effect.

xiv. aFECInterleaverDepth

ATTRIBUTE

SYNTAX:

INTEGER {1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 96}

DESCRIPTION:

A read-write value specifying interleaver depth.

A GET operation returns the current value of interleaver depth if aFECAdminState is enabled. Otherwise a maximum supported depth value is returned.

SET operation, allowed on BTU-C only, changes the interleaver depth to the indicated value only if aFECSupported is true, the link is down. If aFECSupported is false or the link is not down, the operation has no effect.

c. oServices managed object calss

i. aServiceID

ATTRIBUTE

SYNTAX:

INTEGER {1-60}

DESCRIPTION:

A read-only value to uniquely identify a service. There can be up to 60 services defined over TDIM bonded facility. Services with lower IDs have higher priority in case of bandwidth degradation.

ii. aServiceType

ATTRIBUTE

SYNTAX:

ENUM {DS1, E1, NxDS0, NxE0, DS3, E3, Clock, Ethernet, ATM, GFPnoFCS, GFP}

DESCRIPTION:

A read-write value specifying the service type of the BTU.

A GET operation returns currently assigned value for a specific service identified by aServiceID.

A SET operation, allowed on BTU-C only, changes the service type to the indicated value if the link is down. If the link is not down the operation has no effect.

iii. aServiceSize

ATTRIBUTE

SYNTAX:

INTEGER

DESCRIPTION:

A read-write value specifying the number of bytes per bonding sub-block for a specific service identified by aServiceID (number of channels for fractional DS1/E1 (NxDS0/NxE0) or maximal number of bytes for asynchronous services (Ethernet, ATM, GFPnoFCS and GFP)).

A GET operation returns current value.

A SET operation, allowed on BTU-C only, changes the service size to the indicated value if the link is down. If the link is not down or the service type is fixed rate TDM service (aServiceType is DS1, E1, DS3, E3 or Clock) the operation has no effect.

d. oPair managed object class

i. aPairID

ATTRIBUTE

SYNTAX:

INTEGER {1-32}

DESCRIPTION:

A read-only value to uniquely identify a wire pair (modem) in the bonding group, i.e. logical number of the pair in the aggregation group. This value is never greater than aGroupCapacity.

ii. aPairStatus

ATTRIBUTE

SYNTAX:

ENUM {Down, Handshake, Activation, Synching, Synched, Adding, InGroup, SyncLost, Removing}

DESCRIPTION:

A read-only value indicating current pair status. All the states defined in section 12.1 (Pair management and control).

iii. aPairPhysicalID

ATTRIBUTE

SYNTAX:

INTEGER

DESCRIPTION:

A read-only value indicating physical number of the pair (labeled externally on the equipment).

iv. aPairRemotePhysicalID

ATTRIBUTE

SYNTAX:

INTEGER

DESCRIPTION:

A read-only value indicating physical number of the remote pair (labeled externally on the equipment) connected to the local pair specified by aPairID.