Mobile Optical Pluggables Alliance (MOPA), Remote Performance Monitoring Specification

May 21, 2023

Abstract: This specification has been produced by the Mobile Optical Pluggables Alliance (MOPA). It defines the method of sending remote performance monitoring data using ITU-T G.698.4 frame structure as a basis. A small set of Type of Message (TOM) will be defined for the purpose of sending special messages specifically for SFF-8472 Transceiver Management register device addresses, pages and bytes. The memory map defined and associated with this specification is detailed in SFF-8472 Rev 12.4.1 and above.

POINTS OF CONTACT:

Hock Gin Lim
Lumentum
1001 Ridder Park Drive
San Jose, CA 95131
Ph: +1 (408)-546-5984
hockgin.lim@lumentum.com

Ernest Muhigana
Lumentum
1001 Ridder Park Drive
San Jose, CA 95131
Ph: +1 (408)-546-5148
ernest.muhigana@lumentum.com

Fabio Cavaliere
Ericsson
Via Moruzzi 1 c/o CNR
56124 Pisa, Italy
Ph: +39 050 5492381
Fabio.cavaliere@ericsson.com

Alberto Deho
Ericsson
Via Melen 77
16152 Genoa, Italy
alberto.deho@ericsson.com

Alberto Mari
Ericsson
Via Melen 77
16152 Genoa, Italy
alberto.mari@ericsson.com

Carlo Campanelli
Ericsson
Via Melen 77
16152 Genoa, Italy
carlo.campanelli@ericsson.com

Foreword

The development work on this specification was originally contributed to MOPA by Lumentum Inc. and integrating contribution by Ericsson for the State Machine part.
This specification is informational.

Revision History

Version 0.90.5 March 22nd, 2023
- Initial limited release to MOPA members

Version 1.0 May 21, 2013
- For general publication.

Contents

Figures

Tables

1 Scope and Purpose

This document defines a method of transmitting and receiving data in packets using G.698.4 frames. The purpose of these packets is to request, send and receive the memory map used for managing a transceiver remotely over the media interface. This document specifically defines messages that will request, send and receive the SFF-8472 memory map. In future other CMIS memory maps may be considered for similar applications.

The physical method of sending the message over the media is not defined by this document. It assumes that the method described by ITU G.698.4 will be used as a basis, which defines the amplitude modulation index and framing and other transmission characteristics.

The main difference from G.698.4 is that transmission is at 5kbps and the transmission clock will have a fairly loose specification of +/- 5% absolute accuracy but will have a drift specification which is still quite large compared with the 100ppm defined by G.698.4. This is to allow for implementation of a communication scheme using firmware only solution running on small micro-controllers with minimum hardware support. A clock drift specification will be introduced as clock frequency drift per minute.

2 References and Conventions

2.1 Industry Documents

The following documents are relevant to this specification:

- SFF-8472 Rev 12.4.1 (pre-release version) or higher.

2.2 Sources

Copies of IEEE standards may be obtained from the Institute of Electrical and Electronics Engineers (IEEE) (https://www.ieee.org).

Copies of OIF Implementation Agreements may be obtained from the Optical Internetworking Forum (http://www.oiforum.com).

Copies of Common Language Equipment Identification specifications may be obtained from www.commonlanguage.com

Copies of SNIA-SFF documents may be obtained from the SNIA (https://www.snia.org/sff/specifications2).

2.3 Conventions

The following conventions are used throughout this document:

DEFINITIONS

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in the definitions or in the text where they first appear. The word or term may also be printed in **bold font**.

ORDER OF PRECEDENCE

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values.

LISTS

Lists sequenced by lowercase or uppercase letters show no ordering relationship between the listed items.

Lists sequenced by numbers show an ordering relationship between the listed items.

Lists are associated with an introductory paragraph or phrase, and are numbered relative to that paragraph or phrase (i.e., all lists begin with an a. or 1. entry).

NUMBERING CONVENTIONS

The ISO convention of decimal numbering is used (i.e., the thousands and higher multiples are separated by a space and a period is used as the decimal point). This is equivalent to the English/American convention of a comma and a period.

American	French	ISO
0.6	0,6	0.6
1,000	1 000	1 000
1,323,462.9	1 323 462,9	1 323 462.9

Logical Operations

Logical operators are written in uppercase (OR, AND, NOT) and parentheses are used to clarify precedence.

Numerical Constants

Numerals without suffix are understood as numbers in decimal notation (e.g. 16)
Hexadecimal literals are marked with a suffix ‘h’ (eg. 10h), often written with leading zeroes (0010h).
Binary literals are marked with a suffix ‘b’ (e.g. 10000b), often written with leading zeroes (00010000b)
The suffixes may be omitted for unambiguous cases like 0=0b=0h and 1=1b=1h.
Spaces may be inserted to make long hexadecimal or binary digit strings readable(0001 0000b).

Referencing Module Resources

In this specification, all references to lane numbers are based on the electrical connector interface lanes, unless otherwise indicated.
In cases where a status or control aspect is applicable only to lanes after multiplexing or demultiplexing has occurred, the status or control is intended to apply to all lanes in the data path, unless otherwise indicated.
All references to host lanes or media lanes in this document refer to the registers that control or describe those signals. When the term ‘lane’ is used without reference to ‘host’ or ‘media’, a host lane perspective is assumed.

Examples

Examples are printed in grey font for visual differentiation from specification text.

Auxiliary Test

Auxiliary text such as hints or notes are printed in *italic font* for visual differentiation from specification text.

3 Keywords, Acronyms, and Definitions

For the purposes of this document, the following keywords, acronyms, and definitions apply.

3.1 Keywords

May / may not: Indicates flexibility of choice with no implied preference.

Obsolete: Indicates that an item was defined in prior specifications but has been removed from this specification.

Optional: Describes features which are not required by the specification. However, if any feature defined by the specification is implemented, it shall be done in the same way as defined by the specification. Describing a feature as optional in the text is done to assist the reader.

Prohibited: Describes a feature, function, or coded value that is defined in a referenced specification to which this specification makes a reference, where the use of said feature, function, or coded value is not allowed for implementations of this specification.

Reserved: Defines the signal on a connector contact when its actual function is set aside for future standardization. It is not available for vendor specific use. Where this term is used for bits, bytes, fields, and code values; the bits, bytes, fields, and code values are set aside for future standardization. The default value shall be zero. The originator is required to define a Reserved field or bit as zero, but the receiver should not check Reserved fields or bits for zero.

Restricted: Refers to features, bits, bytes, words, and fields that are set aside for other standardization purposes. If the context of the specification applies the restricted designation, then the restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word, or field (e.g., a restricted byte uses the same value as defined for a reserved byte).

Shall: Indicates a mandatory requirement in the context of this informational description that are considered important for interoperability between products that use this specification.

Should: Indicates flexibility of choice with a strongly preferred alternative.

Vendor specific: Indicates something (e.g., a bit, field, code value) that is not defined by this specification. Specification of the referenced item is determined by the manufacturer and may be used differently in various implementations.

3.2 Acronyms and Abbreviations

ACK:	Acknowledge
ASCII:	American Standard Code for Information Interchange (the numerical representation of a character)
BER:	Bit Error Rate
CDR:	Clock and Data Recovery
COR:	Clear on read
DWDM:	Dense Wavelength Division Multiplexing
FEC:	Forward Error Correction

FERC: Frame Error Count
HE: Head End
TE: Tail End
NACK: Not Acknowledge
RO: Read-Only
RW Readable and Writeable
TOM Type Of Message
TWI: Two Wire Interface
WDM: Wavelength Division Multiplexing

3.3 Definitions

Application: TBD

Checksum: a number derived from a block of digital data for the purpose of detecting errors.

Custom: Custom fields and formats are defined by the module manufacturer and may be unique to a specific vendor.

Host Interface: TBD

Interface: TBD

Lower Memory: The 128 bytes addressed by byte addresses 00h through 7fh.

Module: TBD

Media Interface: TBD

NV: Non-Volatile memory: a type of memory that can retrieve stored information even after having been power cycled

Optical Power: TBD

Signal Power: TBD

OMA: Optical Modulation Amplitude: The difference between two optical power levels, of a digital signal generated by an optical source, *e.g.,* a laser diode.

OSNR: Optical Signal to Noise Ratio: The ratio between the optical signal power in a given signal bandwidth and the noise power in a given noise reference bandwidth.

Page: A management memory segment of 128 bytes that can be mapped into Upper Memory.

Pave: Average Power: The average optical power

RO: Read-Only

RPM: Remote Performance Monitoring

Rx: an electronic component (Rx) that converts an input signal (optical or electrical) to an electrical (retimed or non-retimed) output signal.

Tx: a circuit (Tx) that converts an electrical input signal to a signal suitable for the communications media (optical or electrical).

Upper Memory: The 128 bytes addressed by byte addresses 80h through ffh.

4 RPM Introduction

Remote Performance Monitoring (RPM) is an implementation that allows low speed data communication between “paired” pluggable transceiver modules at both ends of a fiber link. Two such systems are shown in Figure 4-1 where a low speed communication channel will allow TX1_RX1 and TX2_RX2 to exchange low speed data specifically digital diagnostics. This allows the remote pluggable transceiver on the TEE to be remotely managed.

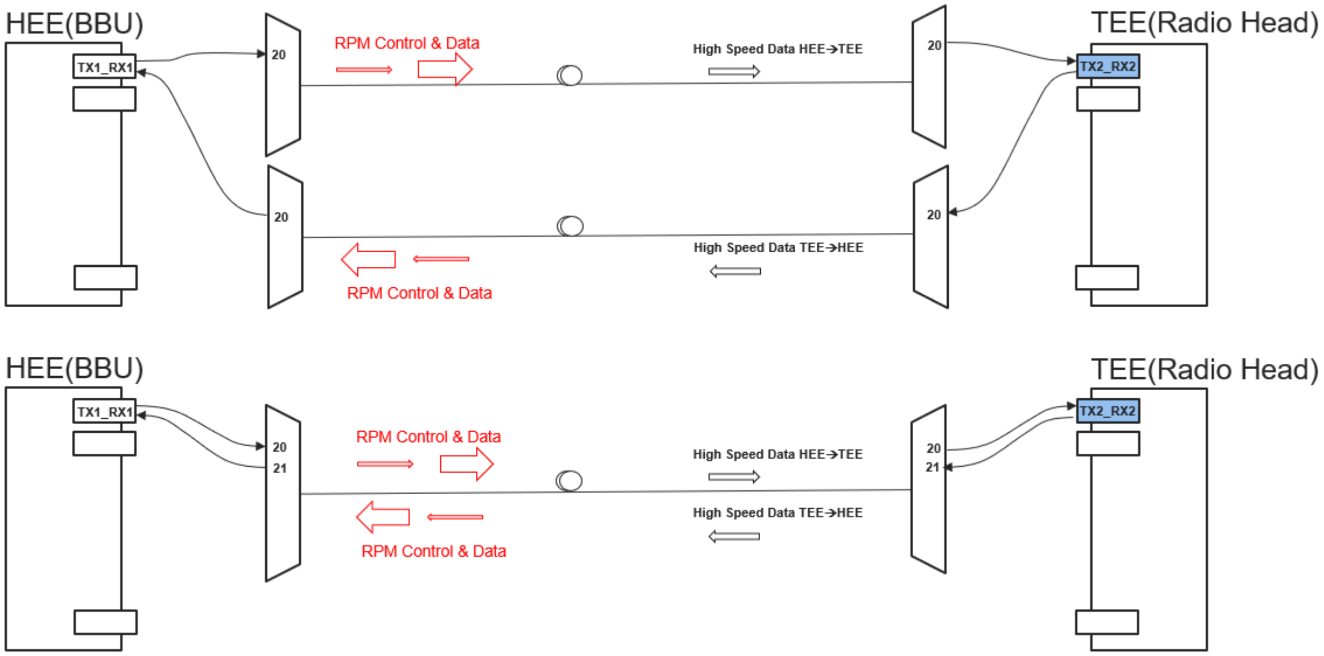

Figure 4-1 Uni-directional and Bi-directional fiber system. This illustration is for WDM based systems but also gray optical links can used the described functionality.

At present this RPM specification is designed to only work with pluggable transceivers using the SFF-8472 management interface and memory map. A summary of the memory map of SFF-8472 with RPM registers is described in Figure 4-2. As shown the remote pluggable transceivers memory map is reflected in Page 20h to 27h.

Table 4-1 list the summary of the memory map that are of importance to this specification. It breaks out sections of the memory map that are mainly constant non-volatile, non-changes, static information as well as digital diagnostics and control registers. These regions will be used in this specification is enable faster transfer of information.

Table 4-1 Summary of SFF-8472 memory map ranges

Dev	Page:Bytes	Description of Pages
A0h	0-63	Consist of mainly RO identification, inventory information, including Vendor name, part numbers, configuration. This includes a checksum at Byte 63.
A0h	64-95	Additional pluggable transceiver options and module Serial Number. This includes a checksum at Byte 95.
A0h	96-127	Additional custom RO information.
A0h	128-255	Custom RO data.
A2h	0-55	RO Alarm and Warning Thresholds
A2h	56-95	RO Calibrations or Enhanced Features.
A2h	96-119	RO, RW Digital Diagnostics, Controls, Latched Alarms.
A2h	120-127	Vendor and table data in byte 127.
A2h	P0:128-255	Page 00/01h: User EEPROM in bytes 128-247 (120 bytes). Vendor EEPROM 248-255.
A2h	P2:128-171	Page 02h: Tunable Registers. For RPM, these registers controls tunability and hence by the time RPM is active, these registers will remain static. If these registers are changing, RPM will most likely loose lock as they reflect the status of the physical media that RPM relies on for communication.
A2h	P2:172-191	Consist of addition local latched alarms local to Page 2 tunable features as well as RPM latched alarms.
A2h	P2:192-207	RPM status, error counters.
A2h	P2:208-239	RPM controls and reserved registers.
A2h	P2:240-255	RPM User Data
A2h	P20h:128-255	Remote Transceivers. SFF-8472 A0h:128-255
A2h	P21h:128-255	Remote Transceivers. SFF-8472 A0h:128-255
A2h	P22h:128-255	Remote Transceivers. SFF-8472 A2h Low Memory Bytes 0-127
A2h	P23h:128-255	Remote Transceivers. SFF-8472 A2h Page 00h/01h Bytes 128-255
A2h	P24h:128-255	Remote Transceivers. SFF-8472 A2h Page 02h Bytes 128-255
A2h	P25h:128-255	Remote Transceivers. Reserved.
A2h	P26h:128-255	Remote Transceivers. Vendor
A2h	P27h:128-255	Remote Transceivers. Vendor

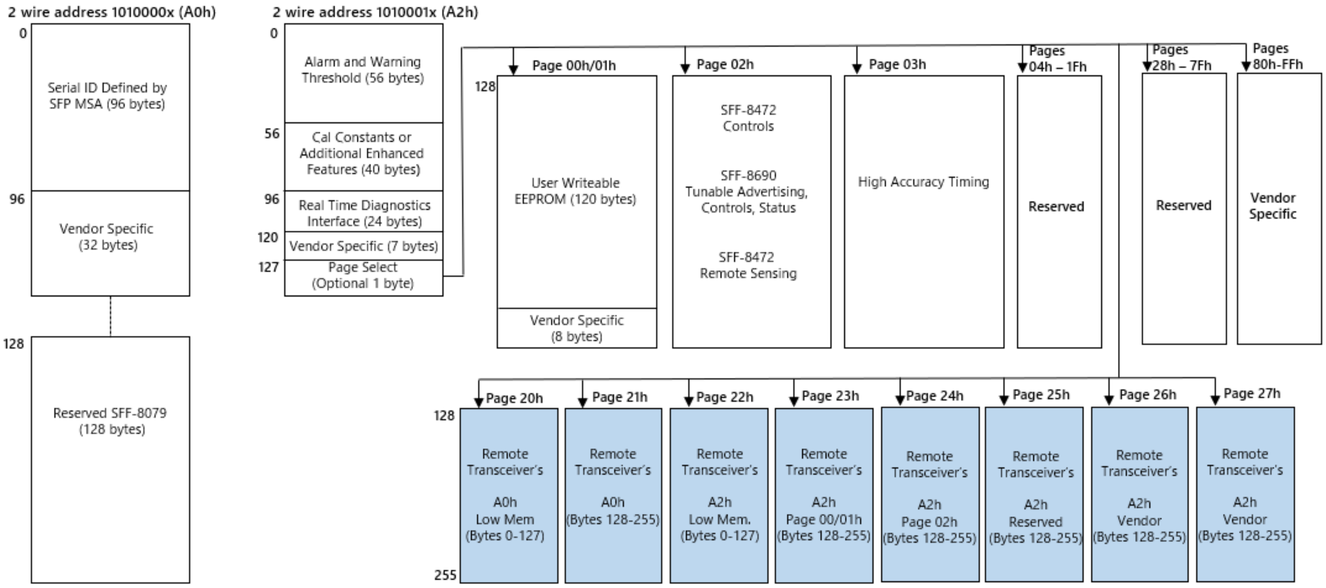

Figure 4-2 SFF_8472 Memory Map

4.1 RPM Requirements

Requirements

- Exchange Inventory information in less than < 2s.
- Periodically exchange Digital Diagnostics every 2s.
- Allow host to send “User Data” over the RPM channel.
- Low Speed communication and exchange of information is bi-directional.
- If enabled, start exchange information automatically.
- If enabled, RxLOS de-assert (eg. fiber break, fiber pull), will automatically restart information exchange.

RPM described here assumes the following

- Pluggable Transceiver Memory Map as per SFF-8472.
- Each TOM::MSG frame addresses and transmits 2 bytes of memory map information.
- Memory Map Data is transferred in units of octet, 8 bytes.
- Framing per G.698.4 and messages, protocol is described in Section 6.
- Message efficiency is 33%, as there are 48 bits in the frame. The framer uses hamming codes for error detection which is capable of detecting 2 bits of error in the TOM and 2 bits of error in the message.

4.2 RPM Module Algorithm

The transmission times quoted in this section uses the nominal transmission parameters described in section 5. The transmission message formats are described in section 6.

The basis of the algorithm depends on defining two transmission data states (S1_Data and S2_Data). These are managed by the state machine described in section 4.2.1.

- State: Transmitting S1 Data (see Figure 4-3)
  - Bytes A0h:0-95 and Bytes A2h:96-119, are being transmitted repeatedly.
  - Messages sent may or may not be received by the remote pluggable module.
  - Commands not accepted are
    - Writing to A2h:P2:208-210 to change remote transceivers memory map.
  - Command frame will be injected in the transmitted bitstream once the RX receives A0h:0-95 message packets and the checksum bytes in A0h:65 and A0h:95 are both valid.
  - Writing to A2h:P2:240-247 to inject user data will be accepted but there may be no guarantees that the RX will receive this message in this state. The host is responsible for any retransmission.
  - Maximum Duration is 576.0 msec without data injection.
  - Each data frame injection will take 9.6 msec.
  - If configured (A2h:P2:211.7) S1_Data will transmit a command to enable the remote TX.

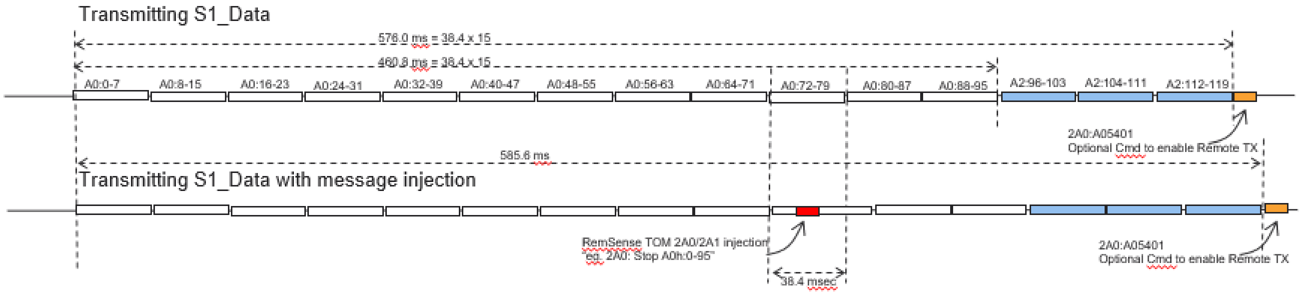

Figure 4-3 State "S1 Data" Transmission

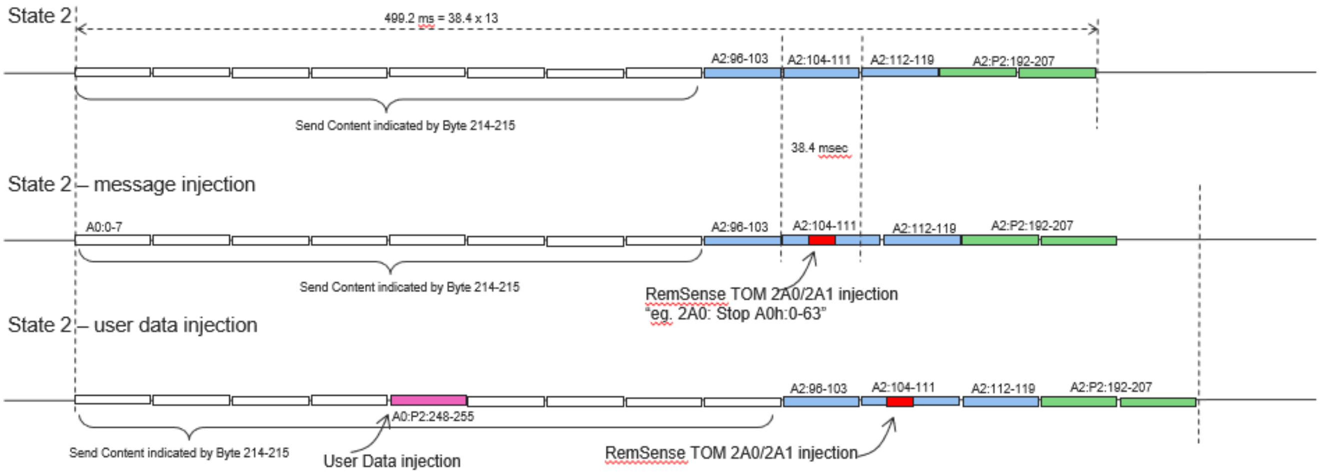

Figure 4-4 State "S2 Data" Transmission

- State: Transmitting S2 Data (see Figure 4-4)
  - Bytes A0h:0-95 are not transmitted.

  - Bytes A2h:96-119 and A2h:P2:192-207 are continuously transmitted.
    - These consists of 5 octets.
    - These 5 octets will be transmitted in between transmitting every 8 octets
  - Bytes defined by register A2h:P2:215 will be continuously transmitted in a round robin every 8 octets.
  - Bytes defined by register A2h:P2:216 if set will be transmitted using 1 of the 8 octets.
  - User Data may be injected. Writing to user data A2h:P2:240-247.
  - Writing to A2h:P2:208-210 to send command to change memory map of remote transceiver will be accepted.
  - Typical duration to for updating digital diagnostics, controls and alarms is about 500 msec.
    - This will increase by 38.4 msec if “user data” is transmitted.
    - This will increase by 9.6 msec if command frames are injected.
  - State S2 Data will stop transmitting and revert back to S1 data transmission as defined by the state machine, described in Section 4.2.1

4.2.1 RPM State Machine

RPM state machines consist of a Transmit and a Receive State Machine. Both the transmit and receive state machine centers around the transmission and reception of S1_Data and S2_Data. Figure 4-5 show the TX and RX state machines which are linked by events. These state machine will synchronize the states of the TX and RX of both local and remote pluggable transceivers.

Some examples of use cases of these state machine are shown in sections 4.2.1.3, 4.2.1.4 and 4.2.1.6.

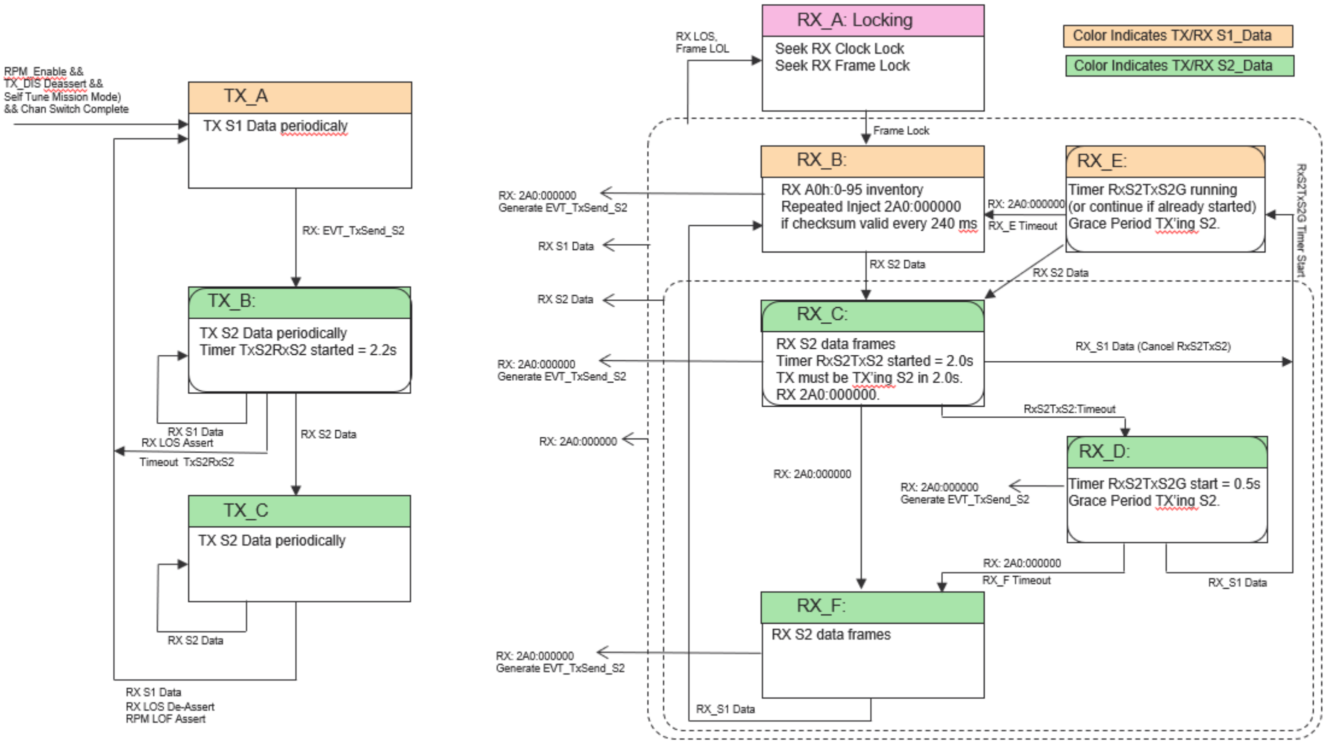

Figure 4-5 Module TX and RX State machines

Transmitter States

- TX_A
  - The Transmitter transmits S1_Data as described in section 4.2 Figure 4-3
  - The module goes into this state whenever
    - the module powers up and completes tuning the laser on a tunable transceiver.
    - RX_LOS de-assert event. (fiber break & maintenance)
    - RX RPM Frame LOF de-assert.
  - Exits State from TX_A to State TX_B when it receives and event EVT_TxSend_S2. The EVT_TxSend_S2 is generated by the RX State machine whenever it receives a message from the remote transceiver, indicating that the remote transceiver has received A0h:0-95 page correctly by verifying checksums at A0h:63 and A0h:95.
- TX_B (Transient State)
  - The Transmitter transmits S2_Data as described in section 4.2 Figure 4-4
  - As this state is entered, a timer TxS2RxS2 is started. This timer shall be set to 2.25 secs.
  - The remote transmitter may still be sending S1_Data. Hence the RX will be receiving S1_Data.
  - RX S1_Data in this state will not cause any effect on the TX_B state.
  - Under normal conditions, the receiver is expected to receive S2_Data within the timeout period.
  - When RX S2_Data event is received, the state now cancels TxS2RxS2 timer and transition to TX_C.
- TX_C
  - The expectation is S2_Data is going to be received.
  - When the RX now receives S1_Data, the transmitter will now change it state back to TX_A. One condition leading to this may be from an LOS event causing the remote transmitter to start sending S1 Data. This is illustrated in section 4.2.1.6.

Receiver States

- RX_A
    - This simply indicates that the RX has not yet achieved frame lock.
    - One frame lock has been achieved, it will transition to RX_B.
- RX_B
    - It is assumed that S1_Data is being received. The state machine will ensure that both remote and local transceivers will be sending S1_Data.
    - When S1_Data is being received, the RX will continuously validate the A0:0-63 data checksum. It will do this every-time an A0 packet is received with data byte address 0 to 63. (See Cmd 0x2A8). Once the checksum is valid, the RX will inject the message 0x2A0:000000. As the local receiver has received the A0:0-63 data with correct checksum, this message will tell the remote transceiver to stop sending S1 and start sending S2. Hence 0x2A0:000000 is the message to tell the receiver to stop sending S1_Data and start sending S2_Data.
    - It will continue to send 0x2A0:000000 at least once every 160-320 msec.
    - Transition to RX_C only after it has received S2_Data.
- RX_C (Transient State)
    - Timer RxS2TxS2 started. (2.0 sec). This timer is to ensure that both local and remote transceivers ends up in the steady state TX_C and RX_F.
    - Receiving S2_Data, timer RxS2TxS2 running.
    - In this state, we are simply waiting for the remote transceiver to lock onto the S1_Data and send the 0x2A0:000000 stop command.
    - If the stop command is received within this state, it will transition to RX_F.
    - If S1_Data is received, it will transition to RX_E and cancel timer RxS2TxS2. It will start the RxS2TxS2G (0.5 sec) timer before entering RX_E from this state.
    - If RxS2TxS2 timer expires, it will transition to state RX_D.
- RX_D (Transient State)
    - This state is entered if RxS2TxS2 timer expires and it is still receiving S2 Data.
    - Start RxS2TxS2G (0.5 sec). At timeout, it will transition to RX_E.
    - In this state, if the 0x2A0:000000 stop command is received it will transition to RX_F and cancel the RxS2TxS2G timer. It will also generate the EVT_TxSendS2.
    - If S1_Data is received, it will transition to RX_E. Since the RxS2TxS2G timer is already running, it will not re-start the timer.
- RX_E (Transient State)
    - In this state the timer RxS2TxS2 is running. S1_Data is being received.
    - If 0x2A0:000000 stop command is received, it will **NOT** generate the EVT_TxSendS2.
    - Upon the expiry of RxS2TxS2 timer it will transition to RX_B.
    - In this state, since it is receiving S1_Data, it will NOT compute the checksum.
- RX_F
    - If 0x2A0:000000 is received, it will generate EVT_TxSendS2
    - If S1_Data is received, the state will transition to RX_B.

4.2.1.1 RPM State machine Timers

There are 3 state machine timers. All these timers act to ensure that both HE and TE pluggable transceivers synchronize and exchange memory map information. The timeout duration is designed to be related to each other (bearing in mind that the remote HE and TE cpu's clock may differ but up to 10%). The timeout also takes into consideration that the time taken to send S1_Data is < 450 msec.

Table 4-2 Summary of SFF-8472 memory map ranges

Timer	Duration	Description
TxS2RxS2	2.25s	TX state machine timer. Started when state machine transitions from TX_A to TX_B. The remote transmitter may still be transmitting S1_Data, hence this RX will be receiving S1_Data.
RxS2TxS2	2.00s	RX state machine timer. Started when RX_B transitions to RX_C.
RxS2TxS2G	0.5s	RX state machine timer. This timer is either running in RX_D or Rx_E states.

4.2.1.2 RPM State machine events.

Table 4-3 RPM State Machine Events

Event	Description
FrameLock	The G.698.4 frame lock has been acquired.
RX S1 Data	Receiving messages with TOM::MSG where TOM=2A8h and MSG is a 24-bit message denote by AABBCCh, where AA values is between 0..47.
RX S2 Data	Receiving messages with TOM::MSG where TOM=2A8h and MSG is a 24-bit message denote by AABBCCh, where AA values is != 0..47.
RX DDM	Receiving TOM::MSG 2A9::AABBCC where AA is between 48 to 59. These messages are sent in both S1_Data and S2_Data states.
EVT_TxSendS2	This event is generated when the TOM::MSG 2A0::000000h is received whilst the RX State machine is in RX_B, RXC, RX_D and RX_F. The event is not generated when the 2A0:000000h message is received whilst in RX_A or RX_E states. In RX_A state, there should not be any messages being received since there is no frame lock.
rsRxLOS	RxLOS events for RPM always uses DC RxLOS, that is optical power has dropped below a threshold for RxLOS Assert and the optical power has restored above a threshold for RxLOS DeAssert.

	This should be independent from the module’s “main” memory map RxLOS which may be configured for AC RxLOS, in which the RxLOS is generated when primary modulation signal at say 10G or 25G has been lost. In 90% of use cases both DC and AC LOS occurs at the similar time. However, there may be uses cases where only AC LOS occurs and the optical signal power is still present at normal levels, hence the RPM channel is still active.
rsRxLOS Assert	The “rsRxLOS Assert” is to be generated when the DC RxLOS is raised. There assumption is that there is no additional “hold” time. TX_C state. Do not change state. TX_B state. Change to TX_A state, since this will eventually timeout. Keep current values in Pages 20h to 27h.
rsRxLOS DeAssert	The “rsRxLOS De-Assert” is to be generated when the DC RxLOS is cleared, but there will be a requirement for additional hold time, depending on whether “Smart Tuning – via smart Tunable MSA” is enabled or not on a tunable device. This feature can be read and determined from register A2h:Pg2:128. If “Smart Tuning” is disabled, then additional hold time will be 0. If “Smart Tuning” is enabled, then “rsRxLOS DeAssert” event will require an additional hold time of 100 msec. This is mainly because smart tuning will generate signaling pulses in the optical power with up to 80 msec pulse width. When this event is received, the module shall: - Clear values in Pages 20h to 27h. - Transition to TX_A state.

4.2.1.3 Example One. HEE/TEE module defaults TX and RX RPM Enable

In this example both HEE and TEE are both assumed to have been set to enable both TX RPM and RX_PRM. When both module powers up, this show a typical scenario where both HEE and TEE pluggable modules completes exchange of data and both sides ends up sending S2_Data.

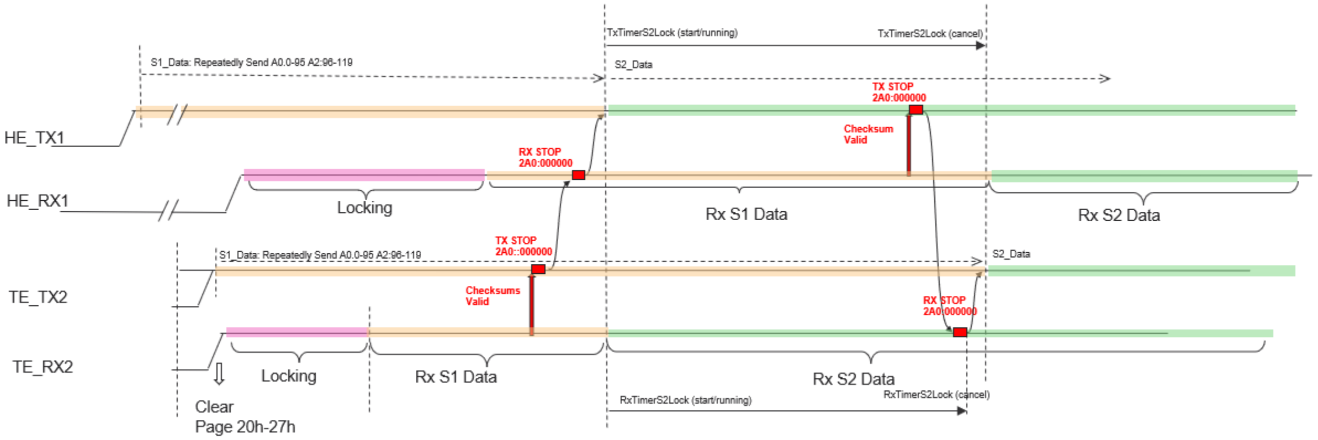

Figure 4-6 HEE/TEE RPM Timing Sync Example TX/RX Enable

4.2.1.4 Example Two. Host Managed. HEE/TEE defaults TX RPM Disable and RX RPM Enable

In this example both HEE and TEE are both assumed to have been set to have TX RPM Disable and RX_RPM Enable. When both module powers up, this show a typical scenario where both HEE and TEE pluggable modules completes exchange of data and both sides ends up sending S2_Data. In this case the events are triggered by the host on one side of the link to enable TX_RPM by setting A2h:Pg02:212.0=1. The figure only show the message to enable the transmitter from the HE_TX1, but it can be sent bi-directionally, this is done to keep the use case as clear as possible.

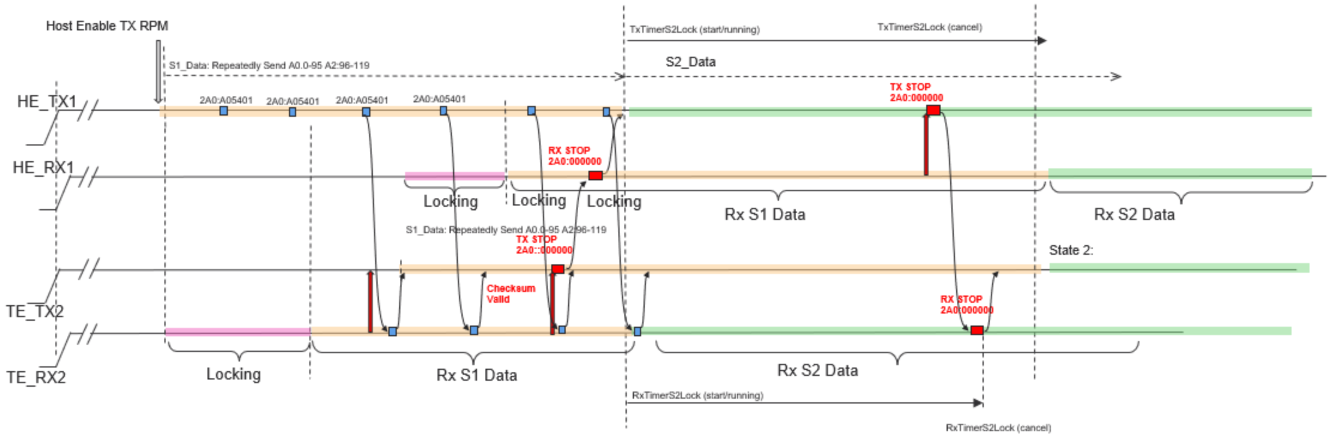

Figure 4-7 HEE/TEE Host Managed Example Data Synchronization

4.2.1.5 HEE, TEE Power Up. Timeout

Figure 4-8 and Figure 4-9 continues from the example in Figure 4-6 but in these examples, synchronization and or locking issues may have occurred, causing the timeouts defined to expire. These two examples show timeout occurring at different states.

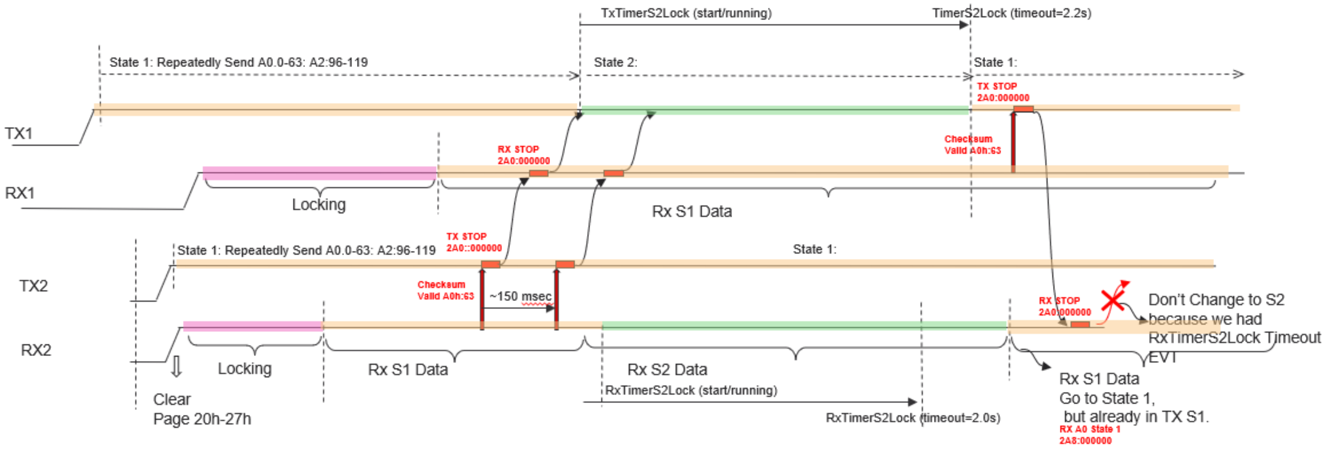

Figure 4-8 HEE/TEE Message Synchronization with Timeout Example 1

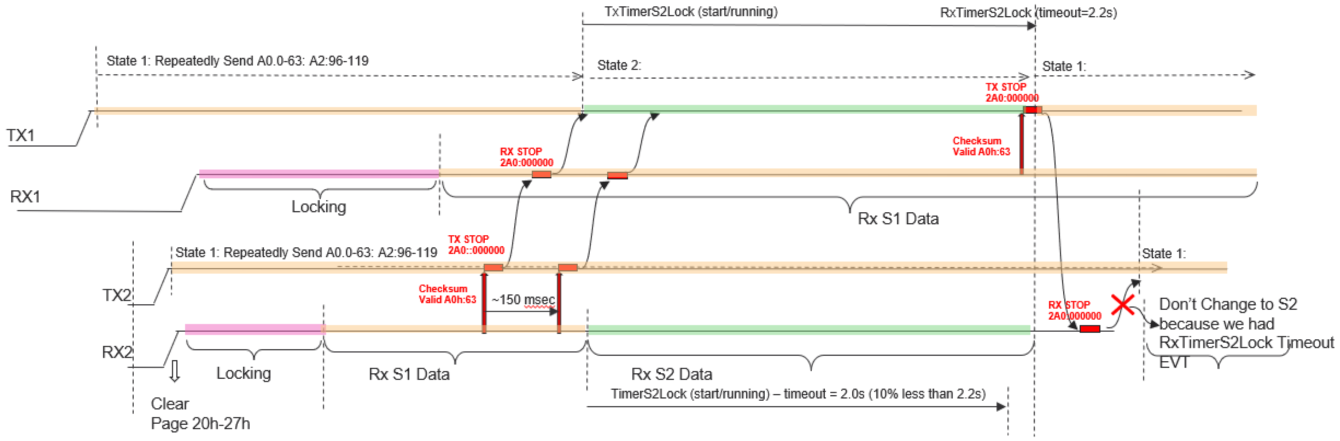

Figure 4-9 HEE/TEE Message Synchronization with Timeout Example 2

4.2.1.6 RX LOS Recovery

The state machine will automatically force both sides to (re)-synchronize the memory map whenever an LOS occurs. In this example, only 1 pluggable transceiver experience and LOS event (fiber pull, or patch panel rework).

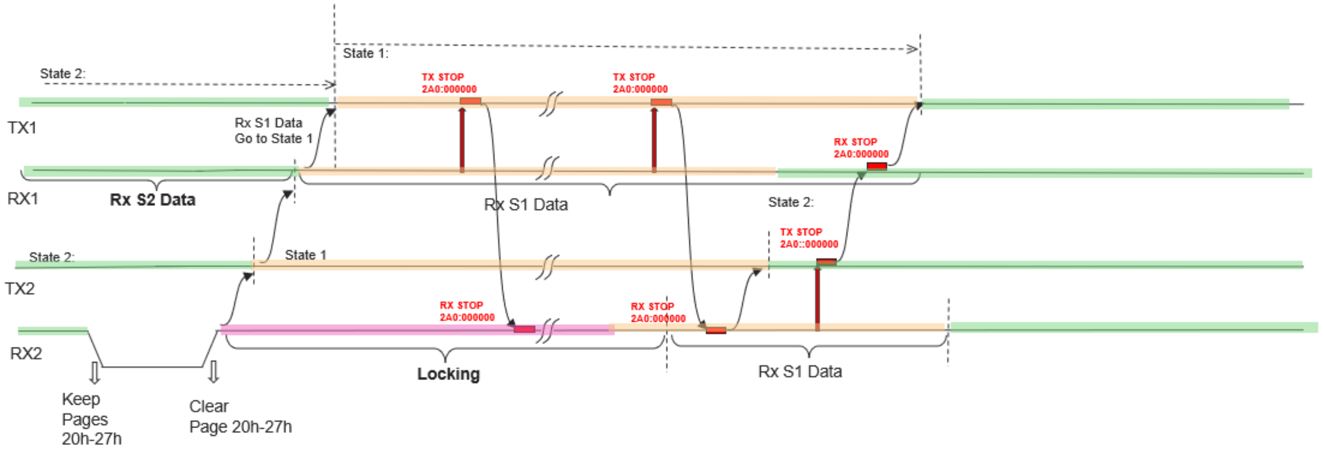

Figure 4-10 HEE/TEE Message Synchronization after RxLOS

NOTE: The remote monitored page data is not cleared until the RX_LOS de-assert event. The TX continues to transmit S2_Data.

4.2.1.7 RPM RX LOF Recovery

The state machine will automatically force both sides to (re)-synchronize the memory map whenever an RPM LOF occurs. In this example, only 1 pluggable transceiver experience and LOF event (spurious burst event that causes 6 consecutive G.698.4 frames to have and error). Each frame is 9.6 msec thus an event > 57.6 msec.

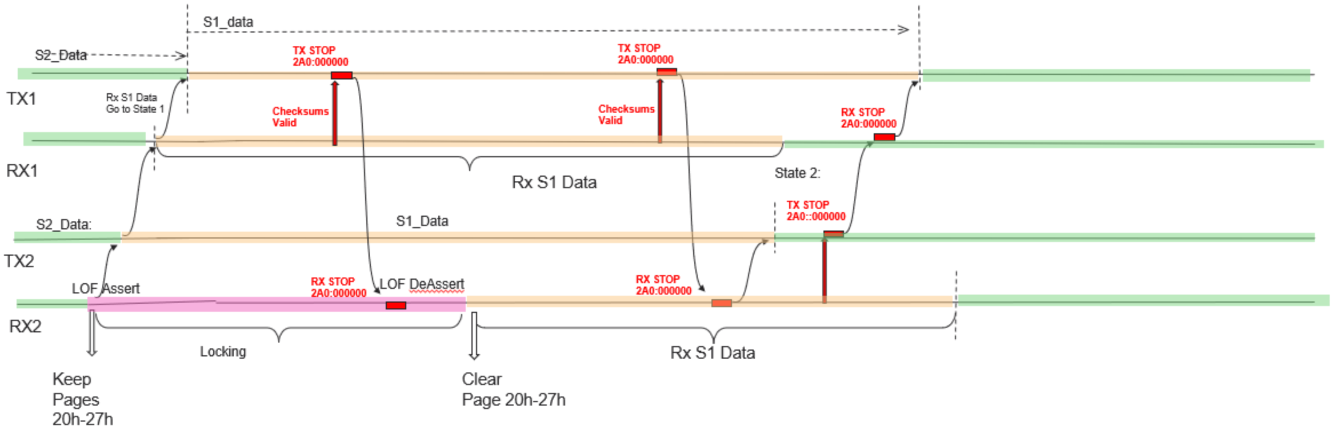

Figure 4-11 HEE/TEE Message Synchronization after RX RPM LOF

NOTE: The remote monitored page data is not cleared until the RPM RX_LOF de-assert event. However, The TX starts sending S1_Data once the LOF event is detected.

5 TX, RX, Framing

This section describes a method of sending and receiving low speed digital data between a pair of optical transceivers at the edges of the link, on top of high speed traffic data. The transmitted data is sent by envelope modulating the transmitted optical signal with a 3-10% modulation index, framed into 48-bit frames per G.698.4.

The low speed received signal is extracted from the RX input power and then demodulated. Typically modulated AC content of the signal is amplified and filtered into the frequency band of interest, to extract the signal content at the desired communication frequency. Amplification and Filtering is typically performed in hardware.

The signal is then sampled, in which additional digital filtering techniques may be employed. After the optional digital filtering of the signal (to increase the SNR of the desired signal in the frequency band of interest), the modulated data stream is then sent to a Clock Recovery mechanism. After clock recovery, we will then have symbol recovery and the frame recovery. In order to minimize system cost, a lot of these steps can be implemented within a low-cost microcontroller using firmware / software algorithms.

The method of transmission, framing and reception of frames used here is defined by ITU G.698.4.

The departure from G.698.4 transmission method:

- 5 kbps used here instead of 47.5 to 52.5 kbps defined in G.698.4
- Absolute clock different between TX and RX may be up to 100,000 ppm clock (+/- 5% to 5 kbps) instead of 100ppm.
- Clock drift over temperature and time are also considered. Although this parameter is very difficult to generate and test.

The important Transmission Characteristics assumed for this implementation is summarized by Table 5-1.

Table 5-1 Transmission Characteristics

Parameter	**Value**	**Comment**
Symbol Rate	5 kbps +/- 5%.	It is assumed that there is no high precision clock in the transmitter and receiver.
Clock Accuracy	100,000 ppm	This is the absolute clock accuracy and potential difference in the transmitter and receiver clock frequency prior to clock locking. This represent the use of low precision crystal oscillators used in MCU's. NOTE this represent +/- 5% of the nominal MCU clock.
Clock Drift	Clock drift should be defined as 500 ppm/min maximum.	This is the change in clock as function of temperature. (or slow drift in clock once the clock is locked). Assume 3C/min typical change, clock drift may come from several sources, including temperature. Typical is 200 ppm/C
Encoding	Manchester Encoding	A combination of 5 kbps symbol rate and Manchester Encoding, implies that there will be a transition every 100 usec or 200 usec, depending on the symbol. A transmission stream of 00000000b or 11111111b will generate a 5 kHz signal A transmission stream of 01010101b or 1010101010b will generate a 2.5 kHz signal.
Framing	Continuous Framing	As defined by G.698.4
Bits per frame	48 bits	Frame time = 48 * 0.2 = 9.6 msec.
Clock Mark/Space Ratio	50/50	Clock high and low has to be as close as possible. Should be between +/- 2%. Aka 48/52 % or 52/48%.

5.1 G.698.4 frames

G.698.4 defines a 48-bit frame structure that consist of

- 11-bit TOM
- 5-bit TOM checksum
- 24 bit MSG
- 8-bit MSG checksum

TOM	TOM checksum	MSG	MSG Checksum

Figure 5-1 G.698.4 Frame Structure

Hamming code for 5-bit TOM checksum and 8-bit MSG checksum from G.698.4 is adopted and not detailed in this document. Theoretically, a hamming code is able to detect 2-bit errors. This mean that up to 2 bit errors in the TOM, and 2 bit errors independently detectable in the MSG file.

We would like to verify this as there was a comment of the possibility of receiving a packet that has an-undetectable error. Check with Fabio?

5.2 Frame Transmission

The transmitter simply transmits the frame using it's local clock. If the input to the framer has no data, the transmitter shall insert "IDLE" frames. IDLE frames are simply frames with a special TOM code.

G.698.4 defines IDLE frames as TOM=0, MSG=0.

In this specification we define an alternate (extended) IDLE frame as TOM=0x2AA MSG=24-bit counter. The purpose of this is to generate a more distinct signal with more equal phase changes to support richer signal for clock recovery. This is because; for a 5 kbps Manchester Encoded stream, a 00000000b pattern generates a higher frequency pattern with 100 usec between transitions (or 5 kHz clock) and a 10101010b pattern generates a lower frequency pattern with 200 usec between transitions (or 2.5 kHz clock). This provides more information for the clock recovery to find both phases of the clock transitions, to lock on the correct Manchester Encoded symbol phase.

The alternate IDLE frame MSG counter being a 24-bit counter can also be used as a detection mechanism that new idle frames are coming into the system, like a keep alive message.

5.2.1 Other Transmission and Control Considerations

The modulation index of the modulated signal which is amplitude modulated may be very small. Care has to be taken to ensure that other optical power, wavelength, aging-loop, that either modulates optical power, change wavelength (changes optical power at a 2nd order), or any other control loops do not cause a random optical power change that may be perceived as a symbol change to the transmitted modulation.

The exact methods are not described here, simply to ensure that optical control loop changes has to be either smaller than the modulation index so as it is not going to periodically change a bit from 1 to 0 or vice versa. One such method may be to ensure that the optical control loop changes the symbol synchronous to the frame boundary and ensure that the change does not affect and momentarily change the modulation index such that the data of the bit is not changed. A transmitter system should be able to characterize these changes.

The key issue here is that we do not want optical control loops to periodically or randomly cause an error in the modulated bit-stream.

5.3 Frame Detection

The algorithm used to lock to a frame is defined by G.698.4, which details the requirement to lock to a frame, but does not define the whole process of extracting a signal to acquiring frame lock. In this section, we try to summaries the functions needed to acquire a signal from clock recover to frame lock. CPU resources will be briefly discussed where it is appropriate.

5.3.1 Clock Detection

Receiving a signal starts with clock detection. For this discussion we assume a 5 kbps Manchester Encoded stream. This signal will consist of expected clock transitions at 100 usec and 200 usec intervals. (and in 10 kbps, clock transitions at 50 and 100 usec respectively). The receiver based on it's local oscillator can sample and look to these two distinct frequencies in a noisy environment, as well as no other significant frequencies being present. The receiver also has to be able to detect that the signal is not good enough and simply consist of a noisy signal that it is not capable of decoding.

At the beginning of clock detection, the receiver due to the large variation in local and receive clock oscillators, specified to be +/- 5%, the receiver also has to tune it's clock phase lock loop to dial into these large variations. The worst case is that the local clock is at -5% nominal and the receiver clock is at +5% of nominal.

In summary the clock detector needs to:

- RX a clear signal that is declared as Manchester Encoded bitstream in noisy environment.
- RX need to lock into a remote clock with up to 10% variation from it's local clock.

Clock detection would need a large number of clocks symbols to be received and appropriately threshold to detect a valid Manchester Encoded bitstream. Technically, all the bits transitions received may be sent to the framer for frame detection, but frame detection implementation per bit or per clock transition received will be computationally costly when implemented in software.

Once clock is detected the clock needs to be tracked. This is to continuously synchronize to the now slower clock drift changes due to environment, voltage and other effects cause by non-ideal clock oscillators. Clock tracking can use a local PLL to track the remote clock. However the expected variation in clock frequencies are now much lower that the original 100,000 ppm. In the table above, we define up to 500ppm change which should be specified as a clock drift specification of ppm change per second. This defines the tracking capabilities of the clock recovery circuit.

Currently there are no specifications on the time to lock and detect a clock signal. This is difficult to specify as the characteristics of the RX signal needs to be fully specified. It is currently assumed that it may take < 3s for clock recovery and frame recovery at a specific operating conditions (To be Specified). It may also depend on RX power as well as presence and Noise Figures of EDFA's.

5.3.2 Frame Recovery

In one implementation, frame recovery is only started when we have a clear indication of the data bit-stream transition and when the clock is recovered. This minimizes CPU processing power as opposed to handling every bit transition and passing it to the framer. Note that clock recovery does not indicate bit phase recovery as Manchester Encoded signals has a "data phase" where the transition carry the symbol and a "non-date phase" at 180 out of phase to the data phase.

Figure 5-2 show a typical Manchester Encoded Stream. As shown a 0000's bit-stream and a 1111's bit-stream generates a high frequency content and a 101010's or 010101's bit-stream generates a low frequency content. When decoding a bit-stream, there will be a data phase and a non-data phase (180° out of phase). When decoding a receive signal, we need to recover the clock as well as the data phase.

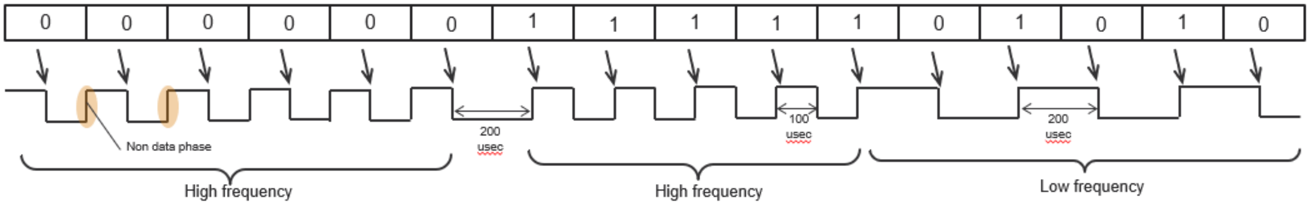

Figure 5-2 Typical 5 kbps Manchester Encoded Stream

Frame detection starts by buffering 48 bits of information, 48 * 0.2 = 9.6 msec worth of data transitions. Data transitions at the non-data phase will have missing transitions at the expected data phases. Depends on the decoding algorithm, all of the data transitions has to be stored for a full 9.6 msec for a 5 kbps Manchester Encoded bit-stream. For each data bit phase change, the framer calculates the checksum of the perceived 48-bits of receive data, and treat missing bits in the appropriate bit-position appropriately. If the checksum bits do not match, the bit clock is shifted to the next incoming bit phase. This bit phase could be at 100 usec or 200 usec elapsed time, depending on the incoming symbol. It is assumed that the incoming bitstream when trying to recover lock may be either TOM=idle or extended idle message, or the incoming bitstream my be frames containing other types of continuous TOM message that sends continuous live real-time DDM (Digital Diagnostics Monitoring) data. This is continue until the frame is lock. It may take up to 48 msec for a single frame lock.

NOTE: implementation in software, this frame recovery process is fairly computationally intensive. Looking at the bit-stream in Figure 5-2, for a 48-bit frame as defined by G.698.4, there could be a average of approximately 60 transitions, (between 48 to 96) data transitions. The framer will have to check every data transition (or every data transition a the data phase) for a valid frame. Firmware has to be sized to process all transitionsfor frame checksum matches. In this case, there may be a need to calculate the TOM and MSG checksum and compare to input bitstream once every 100 usec, hence checking for framing is fairly computationally intensive

Once the first frame is locked, the framer needs to wait for the second full frame, and simply advance 48-bit period for the next frame, hence it will be less computation intensive once frame lock has been achieved. Per G.698.4 frame lock is only declared when 2 consecutive frames have matching checksums. Otherwise, the frame recovery has to continue.

Firmware has to be appropriately sized to ensure all other operations are not affected by the computational resources required during frame acquisition. NOTE that increase the data frame from (eg. From 5 to 10 kbps) will double the computational requirement needed to seek a frame. If the receiver portion is enabled by default connected to a transmitter that is not generating any remote performance monitoring signal; a random signal may causes the false detection of clock locking, and hence frame detection. All this has to be accounted for in processing requirements. This also highlights that continuous clock detection of the modulated signal may be required to ensure robustness of the detection of signals.

5.3.3 Frame Lock State

The frame lock state is when the bit phase boundary and the frame boundary has been found. The receiver now declares that the frame is lock. Now, since the frames are sent as continuous frames, the framer will simply advance the bit-stream by 48 bits. Once frame has been locked, the computational load requirements is predicted to drop by at least 48 times, since we do not need to process every potential bit change for frame lock.

The checksums for TOM and MSG continues to run for all incoming frames. As per G.698.4 if the checksum for 6 consecutive TOM frames are in error, the framer will be declare Loss of Lock and return to Frame Recovery or Clock Recovery States. Here we recommend that both TOM and MSG checksum errors are processed equally. Any frame that has an error will be discarded and not sent to the upper communications layer.

5.3.4 Special Frames

There has been proposal around to use special frames, in additional to standard G.698.4 frames. Special frames are frames that are not framed by G.698.4 framing structure as defined here. It may have say only 3 or 4 bytes or even up to 256 bytes or larger. This current proposal in this document does not use or address the use of special frames except being mentioned here in this section. Smaller special frame should consider defining a new TOM to suite the purpose of the function of the special frame.

The use of special frame will cause additional effort in standardization. Things to consider are:

- Frequency of special frames.
- Frame locking algorithm in the presence of special frame, lock time and locking algorithm.
- Detection of special frame and handling of special frame.
- Relocking or resynchronization to frame lock after special frame.
- Potentially larger vulnerability to errors and clock frequency variation (in case of special larger frames)

In the event if special frames are being used or received, the frame lock state shall be robust enough to recognized the special frame and advance the number of bytes to ensure frame is still locked after the special frame. Definition of special frame and interoperability with G.698.4 frames will also have to be properly define.

In summary special frame will require the framer to perform additional functions and needs to be investigated further.

6 Messages for Remote Data Transmission

In this proposal to send remote data between transceivers, 4 new TOM messages will be introduced. Existing TOM messages that has been defined by G.698.4 per Table 6-1.

Table 6-1 G.698.4 TOM messages (from 03/2018)

TOM value	TOM value	Message Type	Message Content
0	0x000	Idle	
1	0x001	Frequency	Nominal Optical Frequency
2	0x002	Tuning Power	Tuning power relationship to received power
3	0x003	Pilot tone frequency	Frequency to be used for TEE to HEE label pilot tone
4	0x004	Start Sweep	
5	0x005	Turn off	
6	0x006	Stop Sweep	
7	0x007	Change power	New optical Power level
8	0x008	Change frequency	Change in optical frequency
9	0x009	Send traffic	
10	0x00A	Send pilot tone	
11	0x00B	Stop sending pilot tone	

Table 6-2 list the 4 new TOM messages proposed for sending remote data. We would also like to reserve the message code

Table 6-2 New TOM message defined for Remote Performance Monitoring

TOM value	Message Type	Description and Message Content	Section
0x2A0	Command and Response	Send command and respond message.	6.1
0x2A1	Retransmission Command and Response	This is being considered for managing retransmission. If used, the MSG content will be identical to the TOM=0x2A0 message content.	6.1
0x2A1-0x2A7	Reserved	Reserved	
0x2A8	Send A0h and vendor page data	Encoded to send page A0h data and vendor page data.	6.3
0x2A9	Send A2h and reserved page data	Tuning power relationship to received power	6.3
0x2AA	Extended Idle	A 24-bit running counter.	6.4
0x2AB-0x2AF	Reserved	Reserved	

6.1 TOM: 0x2A0. CTRL_CMD / RSP message

This TOM is defined for the sending a message to the remote node. It allows a host to send a message to command message to the remote host, including writing to the memory map of the remote transceiver (SFF-8472 based transceiver). A command message that is sending commands shall be sent at a higher priority than other messages sending data packets in an application where bi-directional RDDM is being continuously sent.

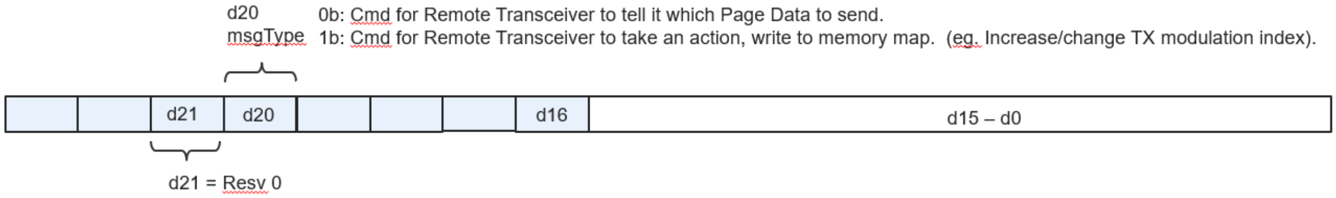

Figure 6-1 TOM 0x2A0 Generic Message

Figure 6-1 show the 24 bit MSG of the TOM 0x2A0. This include

- 1 bit of message type
    - 0b. Indicating message is sending page data, include RDDM
    - 1b. Indicating a command message to affect modules memory map.
- 1 bit of reserved
- 22 bits of message content that is dependent on the message type field (d23)

6.1.1 TOM 0x2A0. msgType = 0b

TOM 0x2A0 msgType 0b is to be used to command the receiver to transmit page data. When this message is received, the receiver should start transmitting data using TOM 0x2A8 and 0x2A9 message. Figure 6-2 show the fields in this message. This message has no “ACK” bit or CMD/RSP bit as per TOM 0x2A0 msgType 1b.

The receiver upon reception of this message, shall start sending page data the prescribed number of times per command in the repeatEnc portion of the message or continuously. The data shall be sent using either TOM 0x2A8 or 0x2A9.

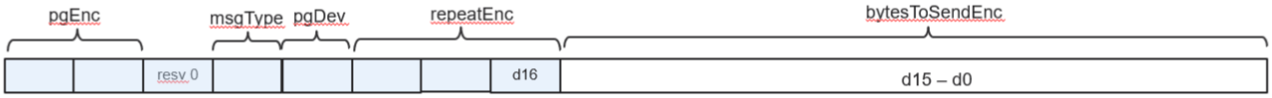

Figure 6-2 TOM 0x2A0 msgType 0b fields

This message consists of:

- msgType is 0b.
- pgEnc (page Encoding) and pgDev (page Devices) indicates which page data to be transmitted, together with the bytesToSendEnc mask field.
    - pgDev 0 pgEnc 0. Send A0h Low data (Bytes 0 to 127).
    - pgDev 0 pgEnc 1. Send A0h Hi data (Bytes 128 to 255)
    - pgDev 0 pgEnc 2. Resv for Vendor Page
    - pgDev 0 pgEnc 3. Resv for Vendor Page

  - pgDev 1 pgEnc 0. Send A2h Low data (Bytes 0 to 127)
  - pgDev 1 pgEnc 1. Send A2h Page 00h/01h. (Bytes 128-255)
  - pgDev 1 pgEnc 2. Send A2h Page 02h. (Bytes 128-255)
  - pgDev 1 pgEnc 3. Resv. Could be potentially used to send A2h. Page 03h.
- repeatEnc defines how many times the page data shall be sent.
  - Value of 0 mean to stop sending the data.
  - Values between 1 to 6 is to tell the receiver to transmit data (in round robin) between 1 and 6 times and stop after sending the same data 1 to 6 times.
  - Value of 7 mean that the data requested shall be sent repeatedly (RDDM data).
- bytesToSendEnc defines a 16-bit mask. Each bit defines an 8 byte group (octet) of addresses within the page. The start address of the octet in the page = 8*bitpos. (Values are 0,8,16,…,112,120). This is a 7 bit address, so this represent the lower 7 bits address in the upper I2C address space.

6.1.2 TOM 0x2A0. msgType = 1b

TOM 0x2A0 msgType 1b is to be used to command the receiver to write to a memory map location or take an action. Upon reception of this message, the receiver shall send an acknowledgement back on the TX. This command and acknowledge message will be sent as higher priority than data message. In order to ensure these command messages are secure in an errored media channel, 3 frames of the command messages shall be sent. This will protect against

- Error'ed frames that are being dropped.
- Error in frames that are not detected by the TOM and MSG checksums.
- Prevents incorrect memory address or action being taken.

The receiver should check to ensure that at least 2 frames of the same command is received back to back before taking an action. (back to back or within 4 frame period).

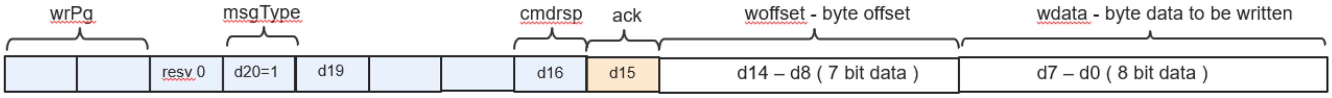

Figure 6-3 TOM 0x2A0 msgType 1b fields

This message consists of:

- msgType 1b.
- wrPg 2 bit encoded field.
  - wrPg 0. Write to Device A2h low.
  - wrPg 1. Write to Device A2h Page 00h/01h.
  - wrPg 2. Write to Device A2h Page 02h.
  - wrPg 3. Resv for vendor.
- cmdRsp flag.
  - 0 bit indicates this is a command.
  - 1 bit indicates this is a response message to the command. In the case of the response message the other bits like wrPg, woffset and wdata of the command shall be echoed.
- Ack or Nack flag.
  - In a CMD message, cmdRsp = 0, this bit shall be set to 0.
  - In a RSP message, cmdRsp = 1, this bit shall indicate 0 (NACK) or 1 (ACK). The NACK mean that the action was not taken. The reason for the NACK is not defined. An ACK mean that the action was taken.
- Woffset

    - This is the 7 bit address within the page to be written. If writing to upper page, then this is the lowest 7 bit of the address.
- Wdata
    - This is the 8 bit data (byte) data of the byte to be written.
- All other bits undefined shall be resv and set to 0.

This command allows any I2C memory of the remote transceiver in A2h Pages 00h/01h/02h to be written. The only difference is the memory map is written from a command received in the media channel as opposed to the local I2C interface. Currently it is up to the vendor's firmware to accept or not accept these writes from the media interface or only allow a certain set of registers to be writeable from the media interface.

This message is a critical message that may affect the remote module's operation state and hence it may be prudent for some of the commands to be rejected by the module. Hence, due to the importance of this message, and a Response bit is defined for this message. The receiver when receiving a command shall response with the same message and it shall indicate this by setting the cmdRsp flag in a response message to 1b. It will also indicate an ACK or an NACK. A NACK may be indicated if the firmware has chosen not to allow the write to occur or is not being handled. In an errored media, the CMD message may be received in error and discarded. In this case there will be no response message and the sender shall implement an appropriate timeout. The response message only acknowledges that the memory map write command message is received and accepted or not accepted. The response message does not indicate the memory map write command has been received and executed. It only indicates that the message has been received.

6.2 TOM: 0x2A1. CTRL_CMD / RSP message

In the event of OSI Layer 3 communication being implemented for requesting message retries, this high priority message shall be used in place of 0x2A0 command. The format is identical to TOM 0x2A0 message.

6.3 TOM: 0x2A8/02A9. Sending Page Data

These two message types are used to data page data content. Each frame sends only 2 bytes. The frame structure is define in Figure 6-4. TOM 0x2A8 is used to send devEnc 0b message (see 6.1.1) and TOM 0x2A9 is used to send devEnc 1b.

- TOM 0x2A8
    - Send Data from A0h low
    - Send Data from A0h high
    - Send Data from Vendor pages
- TOM 0x2A9
    - Send Data from A2h low
    - Send Data from A2h Page 00h/01h.
    - Send Data from A2h Page 02h.

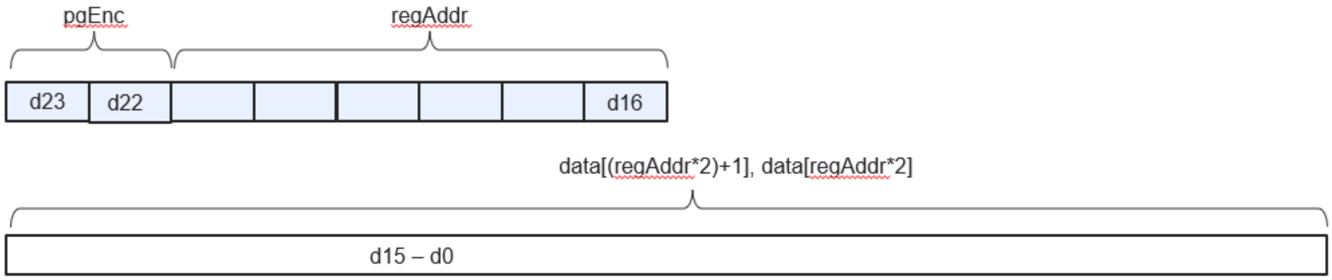

Figure 6-4 TOM 0x2A8/0x2A9 fields

This message consists of

- pgEnc.
    - TOM 0x2A8.
        - pgEnc 00b. Send data from Page A0h low
        - pgEnc 01b. Send data from Page A0h high.
        - pgEnc 10b. vendor.
        - pgEnc 11b. vendor.
    - TOM 0x2A9.
        - pgEnc 00b. Send data from Page A2h low.
        - pgEnc 01b. Send data from Page A2h Page 00h/01h.
        - pgEnc 10b. Send data from Page A2h Page 02h.
        - pgEnc 11b. Reserved.
- regAddr
    - This is the top 6 bits of the byte address within a page.
- Page data (2 bytes).

6.4 TOM: 0x2AA. Extended Idle

An extended IDLE message is define. The purpose of this idle is recognizing that Manchester Encoding is used and hence this create a richer distribution of the frequency of the output signal.

- More even distribution of 2.5 and 5 kHz clock transitions.
- MSG value is a 24 bit counter. Receiver can actually use this as a heartbeat.
- IDLE message may be inserted by the framer if the TX frame buffer has underflowed.

TOM Code : 0x2AA	TOM Chksum	MSG Value = 24 bit counter	MSG Chksum

Figure 6-5 TOM 0x2A8/0x2A9 fields

7 Transmit and Receive Rules

7.1 I2C memory map

In SFF-8472 Rev 12.4.1, the remote performance monitoring memory map was define to have the summary as described in Table 7-1. This table contains the same information as the SFF-8472. See SFF-8472 Revision 12.4.1 or higher.

Table 7-1 Page 02h Register Summary for Remote performance monitoring

A2h Address Bytes	Function	Description	
192-197	Status/Debug	Clock Status and Debug	
198-207	Error Counters	Return frame error counters, enables calculation of BER or FER	
208-210	Tx Remote Cmd	Allow remote memory map to be written.	
211	Tx Mod Index	Modulation Index	
212-215	Control	Controls	
216-239	Reserved	Reserved for Alarm Bit Mask of frames.	
240-255	User Data	User Data	

7.2 Reception of Frames

The receiver upon reception of any frame without any errors in the TOM and MSG packets shall update the debug registers in A2h:Pg02h:Bytes 192-197 with the TOM and MSG received without errors. This allow the host to read the “last received” TOM and MSG to allow the system to be debugged. At 5 kbps transmission each frame will be received within 9.6 msec, hence the TOM and MSG debug registers will be updated at this rate.

If there is an error or a LOF (Loss of Frame), the debug registers shall hold the last valid received TOM and MSG values.

Frames that have been received with an error on either the TOM or the MSG field or both fields shall be discarded. As described in G.698.4, if 6 consecutive frames have been received in error, the receiver will declare a LOF state, and restart framing and or clock recovery. G.698.4 only stipulates that TOM checksum are recognized as errors, but we recommend both TOM and MSG checksum shall be equally treated as errors to declare the LOF state.

7.3 TX and RX Idle Frames

Idle frames shall be transmitted by the framer if there is no other message or data frames to be sent. The framer shall automatically insert idle frame if there are no active messages.

The receiver upon reception of idle frames may simply discard the IDLE frame and take no additional action. The debug registers are already updated whenever a frame has been received.

7.4 Transmitting Data and Cmd Frames

Data frames are sent using TOM 0x2A8 and 0x2A9 messages. Command frames are sent using TOM 0x2A0 messages. Command frames shall be sent at higher priority compared to the transmission of data frames.

A typical application would want to update the digital diagnostics information at the fastest possible rate. Thus data will be sent continuously. Command messages are to be considered higher priority message and shall interrupt the normal transmission of data frames. In the current message definition, there is only one TOM=0x2A0 command message. Thus if there is any CMD message, it should be sent in the next possible frame, interrupt and deferring the data frame that will be sent.

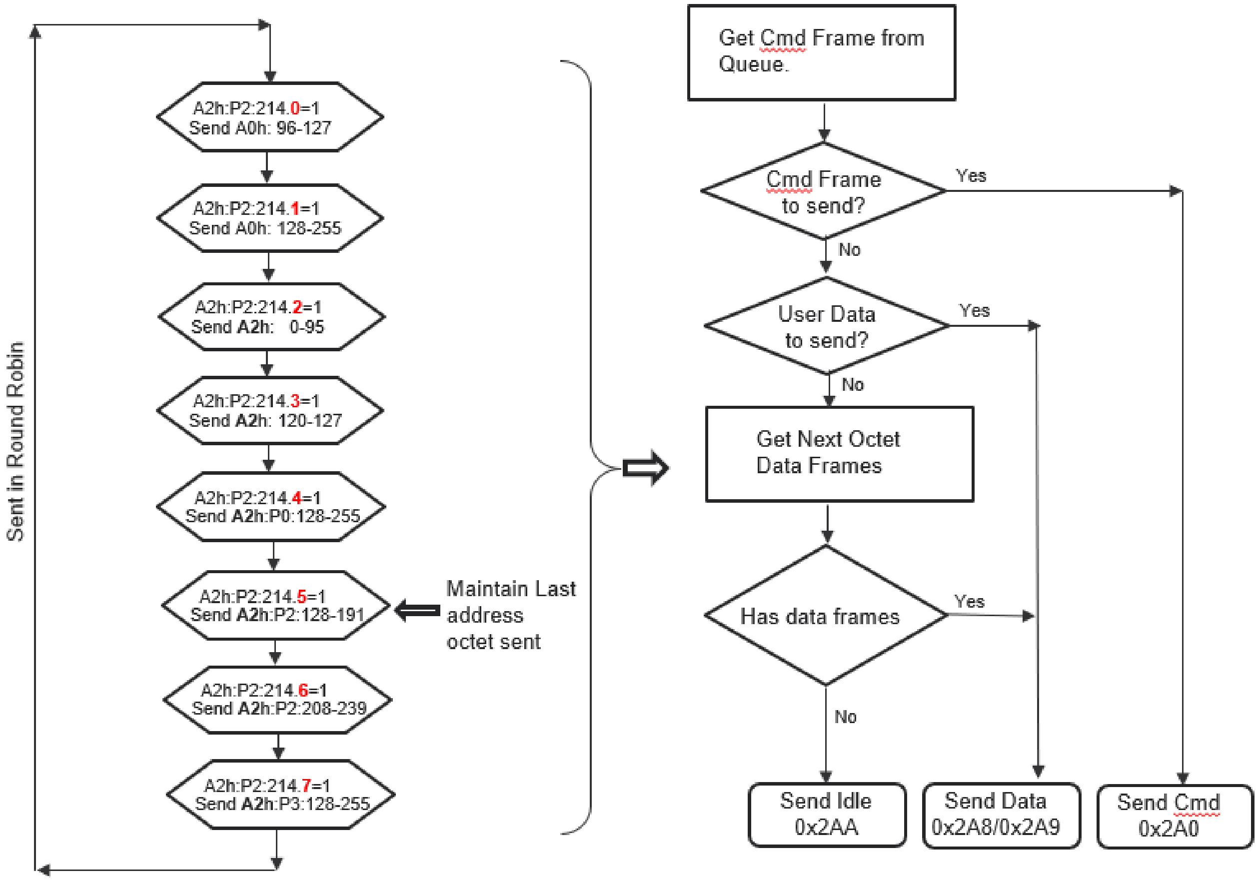

Figure 7-1 Getting Next Octet of Bytes to be Transmitted in S2_Data

Figure 7-1 show the flow-chart to fill up the framing buffer prior to framer requiring to send the next frame. Firstly the application should check if there is a command message to be transmitted or not, then it checks if there is any page data to be transmitted. If any of these results in a frame to be sent, the command frames will have priority. It is up to the application to limit the flow rate of commands to the module. These are expected to be at a low rate.

Sending page data depends on the commanded page data to be sent. Each page shall store a single 16 bit mask and a 4-bit counter. This matches the range that the command message may request page data. Whenever a new message is receive that is the command to change that page mask or repeat count, the respective individual page send mask or repeat count will be replaced by that in the incoming message. This mean that a message which changes the mask to 0 or the repeat count to 0 will stop the current page data transmission.

7.5 Undetectable Errors

Do we need this section, this complicates the interface. I think it's simpler to block some messages to avoid catastrophic messages from bringing the link down. The hamming code can detect 2 bits in error in 16 bits and 2 bits in error in 32 bits, and hence can detect up to 4 bits in error in 48 bits. To calculate, we can detect 1/48 bits in error hence 2E-2.

7.5.1 Transmitting message multiple times

Currently, there is a consideration to send a command message multiple times (3 times), in consecutive frames. This is to combat against receiving a message that was in error but was not being detected by either the TOM or MSG checksum fields. This may be used for critical messages for example messages that cause the laser to tune to another channel, hence bring the link down. This remains something to be considered. If implemented, then command messages shall be sent in 3 consecutive frames and the reception of these command messages shall only accept this message if 2 or more of this message frames has been received identically. This requires special handling by the framer.

In an errored environment the Hamming code is able to detect 2-bit error in the TOM and 2-bit error in the MSG content, hence this type of protection may only be necessary for if the error rates are very high.

7.5.2 More Robust Message for critical commands

Creating a repeated more robust MSG, including additional parity or making use of the 24 bit MSG fields with repeated bytes for robustness. (eg. If sending tuning commands, send the 8 bit channel number 2x within the command).

8 Remote Performance Monitoring

This section provides additional examples of message exchange between HEE and TEE.

8.1 Typical Application Network

Figure 8-1 show a typical network with a local HEE(BBU) transceiver and a remote TEE(Radio Head) transceiver. The message flow in these example assumes asymmetrical operation in which the transceiver behaves differently depends on whether the transceiver is plugged into the HEE or the TEE.

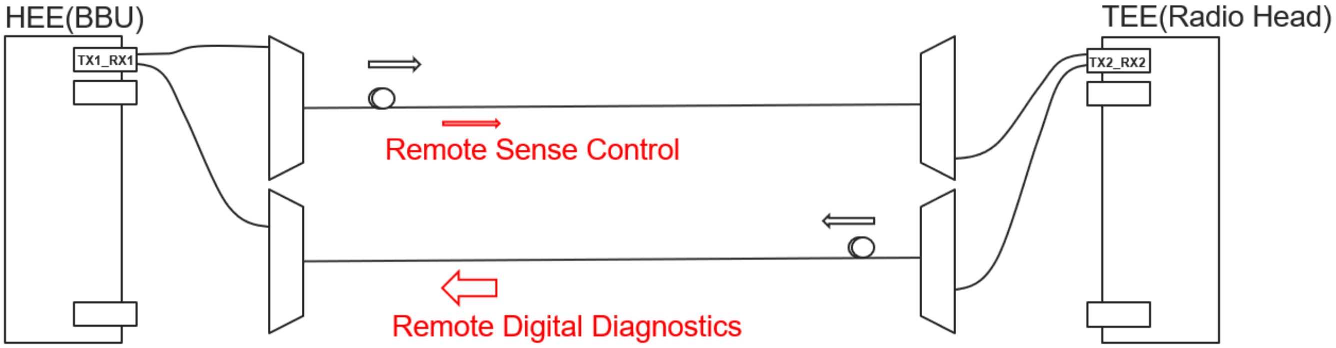

Figure 8-1 TE and HE definition

Upon initial optical link being establish (whether it is a HEE or TEE) it is assumed that remote performance monitoring TX (at a default unspecified modulation index) is automatically enabled when the link is established. This could be either via manual tuning or when the “Self Tuning” via “Self Tuning MSA” has established a channel. Initial bring-up includes the first time LOS is cleared. The data content in the example represent the byte addresses instead of the data.

Module @ HEE(BBU)	
TX	**RX**
Module TX Starts	Module RX Starts
S1 (0x2A8:0x000001)	-
S1 (0x2A8:0x010203)	-
S1 (0x2A8:0x020405)	-
S1 (0x2A8:0x030607)	LOS clear
S1 (0x2A8:0x040809)	Locking to Frames
S1 (0x2A8:0x050A0B)	Locking to Frames
S1 (0x2A8:0x060C0D)	Locking to Frames
S1 (0x2A8:0x070E0F)	S1 (0x2A8:0x030607)
S1 (0x2A8:0x081011)	S1 (0x2A8:0x040809)
S1 (0x2A8:0x091213)	S1 (0x2A8:0x050A0B)
S1 (0x2A8:0x0A1415)	S1 (0x2A8:0x060C0D)
…	…
S1 (0x2A8:0x1F3E3F)	S1 (0x2A9:0x3B7677)
S1 (0x2A9:0x306061)	S1 (0x2A8:0x000001)
…	S1 (0x2A8:0x010203)
S1 (0x2A9:0x3B7677)	S1 (0x2A8:0x020405)
S1 (0x2A8:0x000001)	S1 (0x2A8:0x030607) ➔

Module @ TEE (Radio Head)	
TX	**RX**
Module TX Starts	Module RX Starts
S1 (0x2A8:0x000001)	LOS clear
S1 (0x2A8:0x010203)	Locking to Frames
S1 (0x2A8:0x020405)	Locking to Frames
S1 (0x2A8:0x030607)	Locking to Frames
S1 (0x2A8:0x040809)	Locking to Frames
S1 (0x2A8:0x050A0B)	Locking to Frames
S1 (0x2A8:0x060C0D)	Locking to Frames
…	S1 (0x2A8:0x091012)
S1 (0x2A9:0x3B7677)	…
S1 (0x2A8:0x000001)	
S1 (0x2A8:0x010203)	
S1 (0x2A8:0x020405)	
S1 (0x2A8:0x030607)	
S1 (0x2A8:0x040809)	

	checksum valid, inject stop
STOP (**0x2A0:0x000000**)	-
S1 (0x2A8:0x010203)	-
S1 (0x2A8:0x020405)	-
…	
S1 (0x2A8:0x0C1819)	S2 (0x2A8:0x204041)
S1 (0x2A8:0x0D1A1B)	S2 (0x2A8:0x214243)
S1 (0x2A8:0x0E1C1D)	STOP (**0x2A0:0x000000**)
S2 (0x2A8:0x204041)	S2 (0x2A8:0x224445)
S2 (0x2A8:0x214243)	S2 (0x2A8:0x234647)
S2 (0x2A8:0x224445)	S2 (0x2A8:0x244849)

S1 (0x2A8:0x050A0B)	
S1 (0x2A8:0x060C0D)	STOP (**0x2A0:0x000000**)
S1 (0x2A8:0x070E0F)	S1 (0x2A8:0x010203)
S2 (0x2A8:0x204041)	…
S2 (0x2A8:0x214243)	S1 (0x2A8:0x0B1617) ➔ checksum valid, inject stop
STOP (**0x2A0:0x000000**)	S1 (0x2A8:0x0C1819)
S2 (0x2A8:0x224445)	S1 (0x2A8:0x0D1A1B)
S2 (0x2A8:0x234647)	S1 (0x2A8:0x0E1C1D)
S2 (0x2A8:0x244849)	S2 (0x2A8:0x204041)
S2 (0x2A8:0x254A4B)	S2 (0x2A8:0x214243)

Figure 8-2 TOM Establishing a Link

9 Tuning Automation and Remote performance monitoring

Automation of the tuning of transceiver allow network operators deployment flexibility. There are multiple ways of achieving this goal. It depends on the network topology of the operator. Here two different options will be described.

9.1 Smart Tuning and Remote performance monitoring

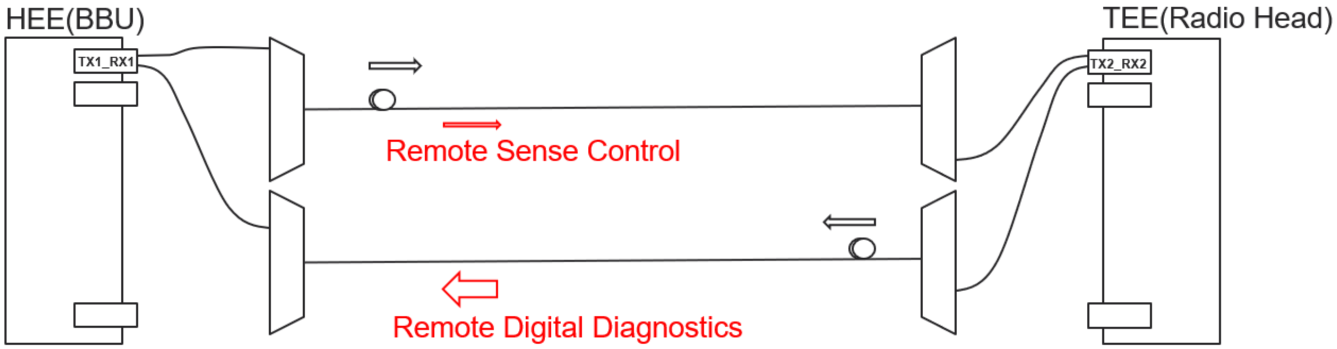

Figure 9-1 HEE and TEE definition

Smart tunable MSA defines a method in which both the HEE and TEE transceivers will automatically tune to the frequencies in which the fibers are connected through the AWG’s.

In this application, if remote performance monitoring control and management were enabled, the transceiver at the Radio Head may not be connected to a host. Here there are two possible solutions.

- The remote performance monitoring RX portion of the circuit is always active.
- Both the remote performance monitoring TX and RX portion of the circuit is always active.

In both these cases, the smart tunable MSA will have a state to declare that the optical frequencies of both BBU and the Radio Head transceivers are locked. Upon achieving that state, the remote performance monitoring circuit shall automatically activate, in either of the use case above.

Module @ HEE(BBU)	
HOST	**Module**
	Module Inserted Power Up
Module supports Self Tuning A2h.Pg2.B128.3=1b A2h.Pg2.B151.1=1b (auto)	Module autonomously starts smart tuning. (Since A2h.Pg2.B151.1=1b). If this bit was 0b, then module will want for host command.
	Module continues smart tuning until smart tuning algorithm locks to the remote node.
	Module continues smart tuning.
	Smart Tuning negotiation
	Smart Tuning locked to channel

Module @ TEE (Radio Head)	
Module	**HOST (not present)**
Module Inserted Power Up	
Module autonomously starts smart tuning. (Since A2h.Pg2.B151.1=1b). If this bit was 0b, then module will want for host command.	
Smart Tuning negotiation	
Smart Tuning locked to channel	

	Remote PM TX disabled	Remote PM TX Disabled	
	Remote PM RX enabled	Remote PM RX Enabled	
Host Initiates Remote PM We can define a state machine here for timeouts and increasing or decreasing TX modulation index.	Remote PM TX enable Start sending IDLE message.	Upon RX of IDLE message remote Sends TX automatically using a default modulation index of 10%.	
	RX remote PM IDLE messages. Set A2h.Pg2.Byte 192 status flags. Set A2h.Pg2 update remote PM error counters.	Continue sending IDLE packets.	
Initiate the State Machine to transfer RPM data	Initiate the State Machine to transfer RPM data.	Initiate the State Machine to transfer RPM data	Initiate the State Machine to transfer RPM data.

Figure 9-2 Example of Smart Tunable and Remote Performance Monitoring

9.2 Tuning Commands via Remote Performance Monitoring Command Message

Another application of the remote command messages defined in this draft is when the optical frequency from the BBU to the radio head is already known. Hence the Radio Head Transceiver will have it's RX and it's Remote Performance Monitoring Receiver enabled. Once the transceiver is ready, it will be ready to receiver commands to allow it to tune it's transceiver.

Module @ HEE(BBU)		**Module @ TEE (Radio Head)**	
HOST	**Module**	**Module**	**HOST (not present)**
	Module Inserted Power Up		
Module does not supports Self Tuning via Smart Tuning. A2h.Pg2.B128.3=0b Assume module support remote performance monitoring.			
	Module is Ready, TX can be enabled or disable.		
Start State machine to use remote PM to tune remote transceivers.			
Host State machine robust enough to timeout and continue.	Send IDLE message tune message and idle message.	Module is off.	
	Send IDLE message tune message and idle message.	Module is Inserted Power Up.	
	Send IDLE message tune message and idle message.	If RX is channel independent, then RX will lock to IDLE messages and eventually get the message to tune to the channel. If RX is not channel independent, then it host state machine on BBU shall set appropriate to tune to a channel and send remote PM data with sufficient time and IDLE message for this node to lock.	
	RX LOS clear will clear when the Remote TX has tuned.		
Initiate the State Machine to transfer RPM data	Initiate the State Machine to transfer RPM data.	Initiate the State Machine to transfer RPM data	Initiate the State Machine to transfer RPM data.
Host can read error counters and then send command to remote transceiver to change modulation index for optimum performance.			

Figure 9-3 Example of Tuning via G.698.4 message (Remote Sense)

End of Document