Document Identifier: DSP0284

Date: 2023-08-25

Version: 1.0.0

Management Component Transport Protocol (MCTP) Memory-Mapped BMC Interface (MMBI) Transport Binding Specification

Supersedes: None

Document Class: Normative

Document Status: Published

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Management Component Transport Protocol (MCTP) Memory-Mapped BMC Interface (MMBI) Transport Binding Specification* (DSP0284) was prepared by the DMTF PMCI Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

This version is the first version of this document. Future changes with be detailed in the change log in ANNEX B.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

Editors:

- Janusz Jurski – Intel Corporation

DMTF Contributors:

- Bob Stevens – Dell Inc.
- Edward Newman – Hewlett Packard Enterprise
- Hemal Shah – Broadcom Inc
- Mahesh Natu – Intel Corporation
- Michael Garner – Meta
- Patrick Caporale – Lenovo
- Patrick Schoeller – Intel Corporation
- Rama Bisa – Dell Inc.
- Richard Marian Thomaiyar – Intel Corporation
- Scott Phuong – Cisco
- William Scherer III – Hewlett Packard Enterprise
- Yuval Itkin – NVIDIA Corporation
- Ramesha He – Dell Inc.
- Samer El-Haj-Mahmoud – ARM, Inc
- Chandra Nelogal – Dell Inc.

Introduction

The Management Component Transport Protocol (MCTP) Memory-Mapped BMC Interface (MMBI) transport binding defines a transport binding for facilitating communication between platform management components, typically host software and a BMC (board management controller).

The *Management Component Transport Protocol (MCTP) Base Specification* describes the protocol and commands used for communication within and initialization of an MCTP network. The MCTP MMBI transport binding definition in this specification includes a packet format, physical address format, and discovery mechanisms for MCTP over MMBI communications.

1 Scope

This document provides the specification for the Management Component Transport Protocol (MCTP) transport binding for MMBI.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF, DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*, https://www.dmtf.org/standards/published_documents/DSP0236_1.3.pdf

DMTF, DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.10*, https://www.dmtf.org/standards/published_documents/DSP0239_1.10.pdf

DMTF, DSP0282, *Memory-Mapped BMC Interface (MMBI) 1.0*, https://www.dmtf.org/standards/published_documents/DSP0282_1.0.pdf

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms “shall” (“required”), “shall not”, “should” (“recommended”), “should not” (“not recommended”), “may”, “need not” (“not required”), “can” and “cannot” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 7. The terms in parentheses are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Clause 7 specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms “clause”, “subclause”, “paragraph”, and “annex” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 6.

The terms “normative” and “informative” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled “(informative)” do not contain normative content. Notes and examples are always informative elements.

Refer to *Management Component Transport Protocol (MCTP) Base Specification* for the terms and definitions that are used across the MCTP specifications.

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

3.1
MMBI

Memory-Mapped BMC Interface

3.2
MMIO

Memory-Mapped Input/Output

4 Conventions

The conventions described in the following clauses apply to this specification.

4.1 Reserved and unassigned values

Unless otherwise specified, any reserved, unspecified, or unassigned values in enumerations or other numeric ranges are reserved for future definition by the DMTF.

Unless otherwise specified, numeric or bit fields that are designated as reserved shall be written as 0 (zero) and ignored when read.

4.2 Byte ordering

Unless otherwise specified, byte ordering of multi-byte numeric fields or bit fields is “Big Endian” (that is, the lower byte offset holds the most significant byte, and higher offsets hold less-significant bytes).

5 MCTP over MMBI Transport

The MCTP over MMBI transport binding defines how MCTP packets are delivered over a MMBI. See *Memory-Mapped BMC Interface (MMBI)* for details about MMBI.

A single MMBI instance shall serve as a communication channel between two communicating entities (typically a host software component and the management controller)[1]. MCTP packet bridging to other interfaces is out of scope for this revision of the specification.

5.1 MCTP Endpoint ID Use and MCTP Bus Owner

5.1.1 MCTP Endpoint IDs

This specification only uses MCTP physical addressing as defined in *Management Component Transport Protocol (MCTP) Base Specification*. The sender of an MCTP over MMBI message shall set the Source EID and the Destination EID fields to zero. The receiver of an MCTP over MMBI message shall ignore the Source EID and the Destination EID fields.

5.1.2 MCTP Bus Owner and MCTP Discovery

As defined in *Management Component Transport Protocol (MCTP) Base Specification*, the MCTP Bus Owner device is responsible for MCTP endpoint discovery and management of MCTP EID assignments. EID assignment is not applicable to MMBI, and a Bus Owner is not used. The *Set Endpoint ID* command should not be generated by MCTP over MMBI implementations. MCTP over MMBI implementations that receive such a command shall respond with the `ERROR_UNSUPPORTED_CMD` code (defined in *Management Component Transport Protocol (MCTP) Base Specification*). The *Discovery Notify*, *Prepare for Endpoint Discovery*, or *Endpoint Discovery* MCTP control messages shall not be used to discover MCTP endpoints over MMBI.

5.1.3 Packet Sizes

The normal packet size requirements and fragmentation and reassembly rules apply for MCTP packet size over MMBI, as defined in *Management Component Transport Protocol (MCTP) Base Specification*.

[1] Note that multiple endpoints in the system can be supported using a plurality of MMBI instances.

5.2 MCTP Packet Encapsulation

The MCTP message header and MCTP message data fields map to MMBI payload as shown in Figure 1.

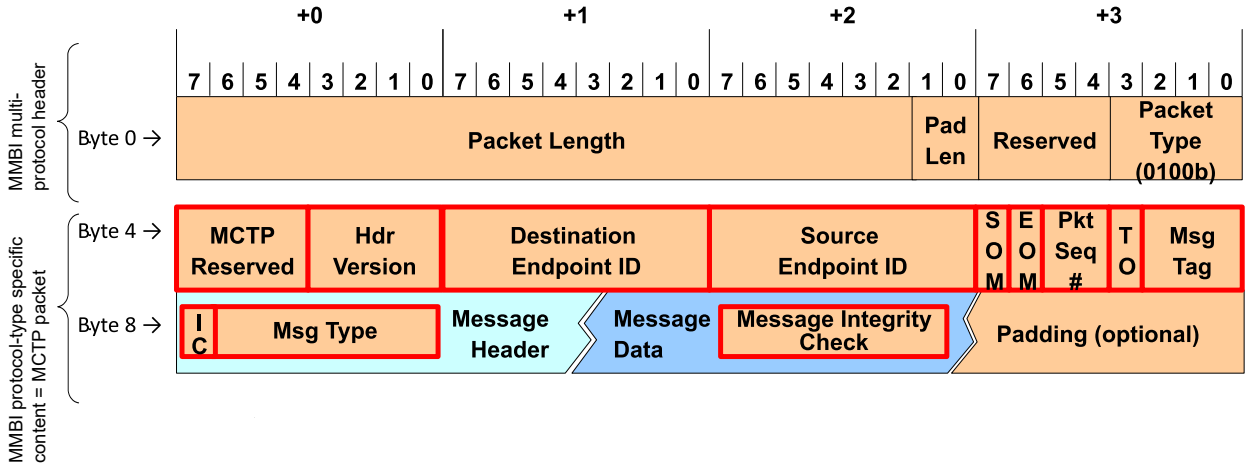

Figure 1 – MCTP over MMBI Encapsulation

The length of the MCTP packet is determined by the MMBI header. Because of the 4-byte alignment requirement, padding must be added if necessary so that the packet length is a multiple of 4-bytes.

Table 1 – MCTP Packet over MMBI – Field Descriptions

Byte	Description
0:2	[23:2] MMBI packet length, as defined in *Memory-Mapped BMC Interface (MMBI)* specification [1:0] Padding length, as defined in *Memory-Mapped BMC Interface (MMBI)* specification
3	[7:4] reserved
	[3:0] Packet Type = MCTP = `0100b`, as defined in *Memory-Mapped BMC Interface (MMBI)* specification
4	[7:4] reserved
	[3:0] Header Version = `0001b` for MCTP, as defined in *Management Component Transport Protocol (MCTP) Base Specification*
5	Destination endpoint ID = Null Destination EID value, as defined in *Management Component Transport Protocol (MCTP) Base Specification*
6	Source endpoint ID = Null Source EID value, as defined in *Management Component Transport Protocol (MCTP) Base Specification*
varies	Padding as defined in *Memory-Mapped BMC Interface (MMBI)* specification

The definitions of all other fields follow *Management Component Transport Protocol (MCTP) Base Specification*,

5.3 Supported media

The MMBI media type identifier for this binding spec is defined in *Management Component Transport Protocol (MCTP) IDs and Codes*, section 7 *MCTP physical medium identifiers*.

5.4 Physical address format for MCTP control messages

The physical address format for MCTP control messages is not defined in the revision of the document. This is because MCTP over MMBI disallows bridging, and the physical address format is only used by commands related to bridging:

- *Resolve Endpoint ID*
- *Resolve UUID*
- *Routing Information Update*
- *Get Routing Table Entries*

MCTP over MMBI and implementations shall not generate these commands. If received, they shall respond with the `ERROR_UNSUPPORTED_CMD` code (defined in *Management Component Transport Protocol (MCTP) Base Specification*).

5.5 Get endpoint ID medium-specific information

The medium-specific information shown in Table 2 shall be used for the medium-specific Information field returned in the response to the *Get Endpoint ID* MCTP control message. Note that the *Get Endpoint ID* MCTP control message should not be typically employed by MCTP over MMBI implementations because this specification only defines the use of the special endpoint ID values: Null Destination EID and Null Source EID—as defined in *Management Component Transport Protocol (MCTP) Base Specification*.

Table 2 – Medium-specific information

	Description
[7:0]	reserved

5.6 MCTP packet and control message timing requirements

With MMBI, which uses a memory-mapping mechanism, the sender and receiver are able to determine if packets have been retrieved from the shared memory buffer. This mechanism can be used as an additional indication to optionally stop MCTP packet retransmissions (i.e., there is no reason to resend a packet if the previous one has not been retrieved from the buffer by the receiver). See Table 3.

208 **Table 3 – Timing specifications for MCTP control messages on MMBI**

Timing Specification	Symbol	Min	Max	Description
Number of request retries	MN1	2	none	Total of three tries, minimum: the original try plus two retries. The maximum number of retries for a given request is limited by the requirement that all retries shall occur within MT4, max of the initial request.
Request-to-response time	MT1	—	100 ms	This interval is measured at the responder from the end of the reception of the MCTP Control Protocol request to the beginning of the transmission of the response.
Time-out waiting for a response	MT2	MT1 max[1] + 2 * MT3 max	MT4, min[1]	This interval at the requester sets the minimum amount of time that a requester should wait before retrying an MCTP control request. This interval is measured at the requester from the end of the successful transmission of the MCTP control request to the beginning of the reception of the corresponding MCTP control response. NOTE: This specification does not preclude an implementation from adjusting the minimum time-out waiting for a response to a smaller number than MT2 based on the measured response times from responders. The mechanism for doing so is outside the scope of this specification.
Transmission delay	MT3	—	20 ms	Time to take into account the transmission delay of an MCTP Control Protocol message. Measured as the time between the end of the transmission of an MCTP Control Protocol message at the transmitter to the beginning of the reception of the MCTP Control Protocol message at the receiver.
Inter-packet delay for multi-packet messages	MT3a	—	100 ms	Allowed time measured from the end of the transmission of an MCTP packet with EOM=0 to the beginning of the following MCTP packet of the same Message (see Message assembly in *Management Component Transport Protocol (MCTP) Base Specification*), measured at the transmitter. The receiver can drop the incomplete message after this timeout.
Instance ID expiration interval	MT4	5 sec [2]	6 sec	Interval after which the instance ID for a given response will expire and become reusable if a response has not been received for the request. This is also the maximum time that a responder tracks an instance ID for a given request from a given requester.

NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.

NOTE 2: If a requester is reset, it may produce the same sequence number for a request as one that was previously issued. To guard against this, it is recommended that sequence number expiration be implemented. Any request from a given requester that is received more than MT4 seconds after a previous matching request should be treated as a new request, not a retry.

ANNEX A
(informative)

Notations

Examples of notations used in this document are as follows:

- 2:N — In field descriptions, this will typically be used to represent a range of byte offsets starting from byte two and continuing to and including byte N. The lowest offset is on the left, the highest is on the right.
- (6) — Parentheses around a single number can be used in message field descriptions to indicate a byte field that may be present or absent.
- (3:6) — Parentheses around a field consisting of a range of bytes indicates the entire range may be present or absent. The lowest offset is on the left, and the highest offset is on the right.
- PCIe — Underlined blue text is typically used to indicate a reference to a document or specification called out in 2, “Normative References” or to items hyperlinked within the document.
- [4] — Square brackets around a number are typically used to indicate a bit offset. Bit offsets are given as zero-based values (that is, the least significant bit offset = 0).
- [7:5] — A range of bit offsets. The most significant bit is on the left, and the least significant bit is on the right.
- `1b` — A number consisting of `0`s and `1`s followed by a lowercase “`b`” indicates that the number is in binary format.
- `0x12A` — A leading “`0x`” indicates that the number is in hexadecimal format.

ANNEX B
(informative)

Change log

Version	Date	Description
1.0.0	2023-08-25	Initial release.