Management Component Transport Protocol (MCTP) PCIe® VDM Transport Binding Specification

Version: 1.3.1

Document Identifier: DSP0238

Date: 2025-07-07

Version History: https://www.dmtf.org/dsp/DSP0238

Supersedes: 1.3.0

Document Class: Normative

Document Status: Published

Document Language: en-US

This document’s normative language is English. Translation into other languages is permitted.

CONTENTS

Figures

Tables

Foreword

The *Management Component Transport Protocol (MCTP) PCIe® VDM Transport Binding Specification* (DSP0238) was prepared by the PMCI Working Group.

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

Acknowledgments

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

Editors:

- Hemal Shah – Broadcom Inc.
- Tom Slaight – Intel Corporation
- Eliel Louzoun – Intel Corporation

Contributors:

- Austin Bolen – Dell Technologies
- Patrick Caporale – Lenovo
- Michał Duzinkiewicz – Intel Corporation
- Samer El-Haj-Mahmoud - ARM
- Steve Glaser – NVIDIA Corporation
- Brett Henning – Broadcom Inc.
- Yuval Itkin – NVIDIA Corporation
- Janusz Jurski – Intel Corporation
- Jose Marinho – ARM
- Mariusz Oriol – Intel Corporation
- Patrick Schoeller – Hewlett Packard Enterprise
- Bob Stevens – Dell Technologies

Introduction

The Management Component Transport Protocol (MCTP) over PCIe VDM transport binding defines a transport binding for facilitating communication between platform management subsystem components (e.g., management controllers, management devices) over PCIe.

The *MCTP Base Specification* describes the protocol and commands used for communication within and initialization of an MCTP network. The MCTP over PCIe VDM transport binding definition in this specification includes a packet format, physical address format, message routing, and discovery mechanisms for MCTP over PCIe VDM communications.

1 Scope

This document provides the specifications for the Management Component Transport Protocol (MCTP) transport binding using PCIe Vendor Defined Messages (VDMs).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

CXL Consortium, *Compute Express Link (CXL) Specification Revision 1.0*, https://www.computeexpresslink.org

CXL Consortium, *Compute Express Link (CXL) Specification Revision 1.1*, https://www.computeexpresslink.org

CXL Consortium, *Compute Express Link (CXL) Specification Revision 2.0*, https://www.computeexpresslink.org

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

DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.9*, https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.9.pdf

ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents*, https://www.iso.org/sites/directives/current/part2/index.xhtml

PCI-SIG, *PCI Express Base Specification Revision 1.1*, March 8, 2005, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 2.0*, December 20, 2006, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 2.1*, March 4, 2009, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 3.0*, November 10, 2010, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 3.1a*, December 7, 2015, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 4.0*, October 5, 2017, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 5.0*, May 28, 2019, https://www.pcisig.com/specifications/

PCI-SIG, *PCI Express Base Specification Revision 6.2*, January 25, 2024, https://www.pcisig.com/specifications/

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 DSP0236 for terms and definitions that are used across the MCTP specifications. For the purposes of this document, the following additional terms and definitions apply.

3.1 Destination Physical Address

the Physical address reflected in the Requester ID field (Non-Flit Mode) or Requester ID and Requester Segment (Flit Mode) fields of the TLP

3.2 Dword

a 32-bit field

3.3 Flit Mode

a mode defined in the PCI Express Base Specification Revision 6.x introducing a new link layer and transaction layer for PCI Express where a TLP header is composed of a 3 to 7 dword TLP Header Base followed by 0 to 7 additional dwords of OHCs (Orthogonal Header Content)

3.4 MCTP PCIe Endpoint

a PCIe endpoint on which MCTP PCIe VDM communication is supported

3.5 PCIe Segment

a PCI Express I/O interconnect topology, in which the Requester IDs must be unique

3.6 Target Physical Address

The Physical address reflected in the Target ID field (Non-Flit Mode) or Target ID and Target Segment (Flit Mode) fields of the TLP

4 Symbols and abbreviated terms

Refer to DSP0236 for symbols and abbreviated terms that are used across the MCTP specifications. The following symbols and abbreviations are used in this document.

4.1 PCIe®

PCI Express®

4.2 VDM

Vendor Defined Message

4.3 CXL®

Compute Express Link®

4.4 FM

Flit Mode

4.5 NFM

Non-Flit Mode

4.6 IDE

Integrity and Data Encryption

4.7 OHC

Orthogonal Header Content

4.8 TLP

Transaction Layer Packet

5 Conventions

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

5.1 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 DMTF.

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

5.2 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 lesser significant bytes).

6 MCTP over PCI Express VDM transport

6.1 Overview

This document defines the medium-specific transport binding for transferring MCTP packets between endpoints on PCI Express using PCIe Vendor Defined Messages (VDMs).

An MCTP over PCIe VDM compliant PCIe device shall support MCTP over PCIe VDM communications on at least one PCIe Physical Function (PF) of the device. If an MCTP over PCIe VDM compliant PCI device supports MCTP over PCIe VDM communications on more than one PCIe function, then MCTP over PCIe VDM communication on each function shall be independent from MCTP over PCIe VDM communications on other PCIe functions.

The MCTP over PCI Express (PCIe) VDM transport binding transfers MCTP messages using PCIe Type 1 VDMs with data. MCTP messages use the MCTP VDM code value (`0000b`) stored in the PCIe TAG field that uniquely differentiates MCTP messages from other DMTF VDMs.

Any MCTP baseline transmission unit for MCTP PCIe VDM communication shall be dword aligned.

The handling of transactions with parameters not matching the values described below is out of scope for this specification. The receiver may choose to accept those or ignore them.

6.2 Packet formats

6.2.1 Non-Flit Mode

Figure 1 shows the encapsulation of MCTP packet fields within a PCIe VDM in Non-Flit Mode.

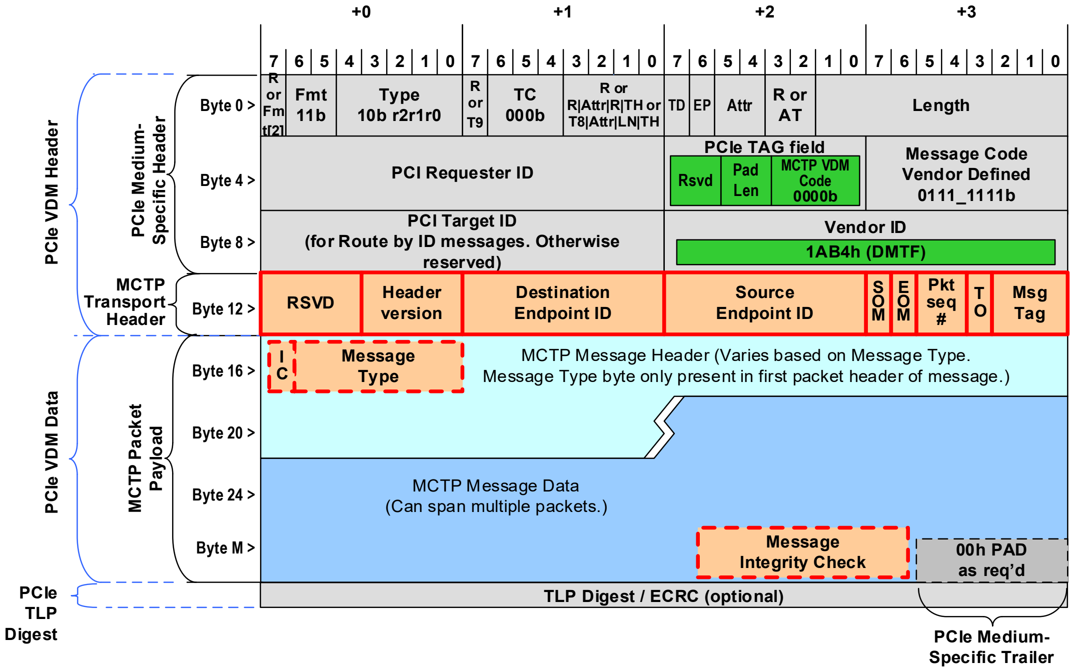

Figure 1 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Non-Flit Mode)

The fields labeled "PCIe Medium-Specific Header" and "PCIe Medium-Specific Trailer" are specific to carrying MCTP packets using PCIe VDMs. The fields labeled "MCTP Transport Header" and "MCTP Packet Payload" are common fields for all MCTP packets and messages and are specified in MCTP. This document defines the location of those fields when they are carried in a PCIe VDM. The PCIe specification allows the last four bytes of the PCIe VDM header to be vendor defined. The MCTP over PCIe VDM transport binding specification uses these bytes for MCTP Transport header fields under the DMTF Vendor ID. This document also specifies the *medium-specific* use of the MCTP "Hdr Version" field.

Table 1 lists the PCIe medium-specific fields and field values that shall be used in MCTP over PCIe VDM communications. When not specified, field values shall be set according to PCIe specifications. Note that the presence of TLP prefixes in MCTP over PCIe VDM packets is implementation dependent and outside the scope of this specification.

Table 1 – PCI Express medium-specific MCTP packet fields

Field	Description
R or Fmt[2]	PCIe 1.1/2.0: PCIe reserved bit (1 bit). PCIe 2.1 and above: Fmt[2]. Set to 0b.
Fmt	Format (2 bits). Set to `11b` to indicate 4 dword header with data.
Type	Type and Routing (5 bits). [4:3] Set to `10b` to indicate a message [2:0] PCI message routing (`r2r1r0`) `000b` : Route to Root Complex `010b` : Route by ID `011b` : Broadcast from Root Complex Other routing fields values are not supported for MCTP.

Field	Description
R or T9	PCIe 1.1/2.0/2.1/3.X: PCIe reserved bit (1 bit). Set to 0b. PCIe 4.X and above: T9 (1bit). Set to 0b.
TC	Traffic Class (3 bits). Set to `000b` for MCTP over PCIe VDM.
R or R | Attr | R | TH or T8 | Attr | LN | TH	PCIe 1.1/2.0: PCIe reserved bits (4 bits). Set to 0000b PCIe 2.1/3.X: PCIe reserved bit (1 bit), Attr[2] (1 bit) – Set to 0b, reserved bit (1bit), and TH (1bit) – Set to 0b. PCIe 4.X and above: T8 bit (1 bit) – Set to 0b, Attr[2] (1 bit) – Set to 0b, LN (1bit) – Set to 0b, and TH (1bit) – Set to 0b
TD	TLP Digest (1 bit). 1b indicates the presence of the TLP Digest field at the end of the PCIe TLP (transaction layer packet). The TD bit should be set in accordance with the devices overall support for the TLP Digest capability, and whether that capability is enabled. See description of the TLP Digest / ECRC field, below, for additional information. Note that earlier versions of this specification erroneously required this bit to be set to 0b, which would have required devices to not support the TLP Digest capability.
EP	Error Poisoned (1 bit).
Attr[1:0]	Attributes (2 bits). Set to `00b` or `01b` for all MCTP over PCIe VDM.
R or AT	PCIe 1.1: PCIe reserved bits (2 bits). PCIe 2.0 and above: Address Type (AT) field. Set to 00b.
Length	Length: Length of the PCIe VDM Data in dwords. Implementations shall support the baseline transmission unit defined in the *MCTP Base Specification*. For example, supporting a baseline transmission unit of 64 bytes requires supporting PCIe VDM data up to 16 dwords. An implementation may optionally support dword aligned larger transfer unit sizes.
PCI Requester ID	Bus/device/function or bus/function number of the managed endpoint sending the message.
Pad Len	Pad Length (2-bits). 1-based count (0 to 3) of the number of `0x00` pad bytes that have been added to the end of the packet to make the packet dword aligned with respect to PCIe. Because only packets with the EOM bit set to `1b` are allowed to be less than the transfer unit size, packets that have the EOM bit set to `0b` will already be dword aligned and will thus not require any pad bytes and will have a pad length of `00b`.
MCTP VDM Code	Value that uniquely differentiates MCTP messages from other DMTF VDMs. Set to `0000b` for this transport mapping as defined in this specification.
Message Code	(8 bits). Set to `0111_1111b` to indicate a Type 1 VDM.
PCI Target ID	(16 bits). For Route by ID messages, this is the bus/device/function number or bus/function number that is the physical address of the target endpoint. This field is ignored for Broadcast and for Route to Root Complex messages.
Vendor ID	(16 bits). Set to **6836** (`0x1AB4`) for DMTF VDMs. The most significant byte is in byte 10, the least significant byte is byte 11.
RSVD	MCTP reserved (4 bits). Set these bits to 0 when generating a message. Ignore them on incoming messages.
Hdr Version	MCTP version (4 bits) `0001b` : For MCTP devices that conform to the *MCTP Base Specification* and this version of the PCIe VDM transport binding. All other settings: Reserved to support future packet header field expansion or header version.

Field	Description
00h PAD	Pad bytes. 0 to 3 bytes of 00h as required to fill out the overall PCIe VDM data to be an integral number of dwords. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned and will thus not require any pad bytes and will have a pad length of 00b.
TLP Digest / ECRC	(32 bits). TLP Digest / ECRC (End-to-end CRC). This field is defined for all PCIe TLPs (Transaction Layer Packets). Device support for this field is optional. However, per PCIe v2.1 and above: "If a device Function is enabled to generate ECRC, it must calculate and apply ECRC for all TLPs originated by the Function. If the device supports generating this field, it must support it for all TLPs." Additionally, per PCIe v2.1 and above, if the ultimate PCI Express Receiver of the TLP does not support ECRC checking, the receiver must ignore the TLP Digest.

Note: In this table, "PCIe X.X and above" means up to the latest PCIe version covered by this specification.

6.2.2 Flit Mode

Figure 2 and Figure 3 show the encapsulation of MCTP packet fields within a PCIe VDM in Flit Mode with or without OHCs.

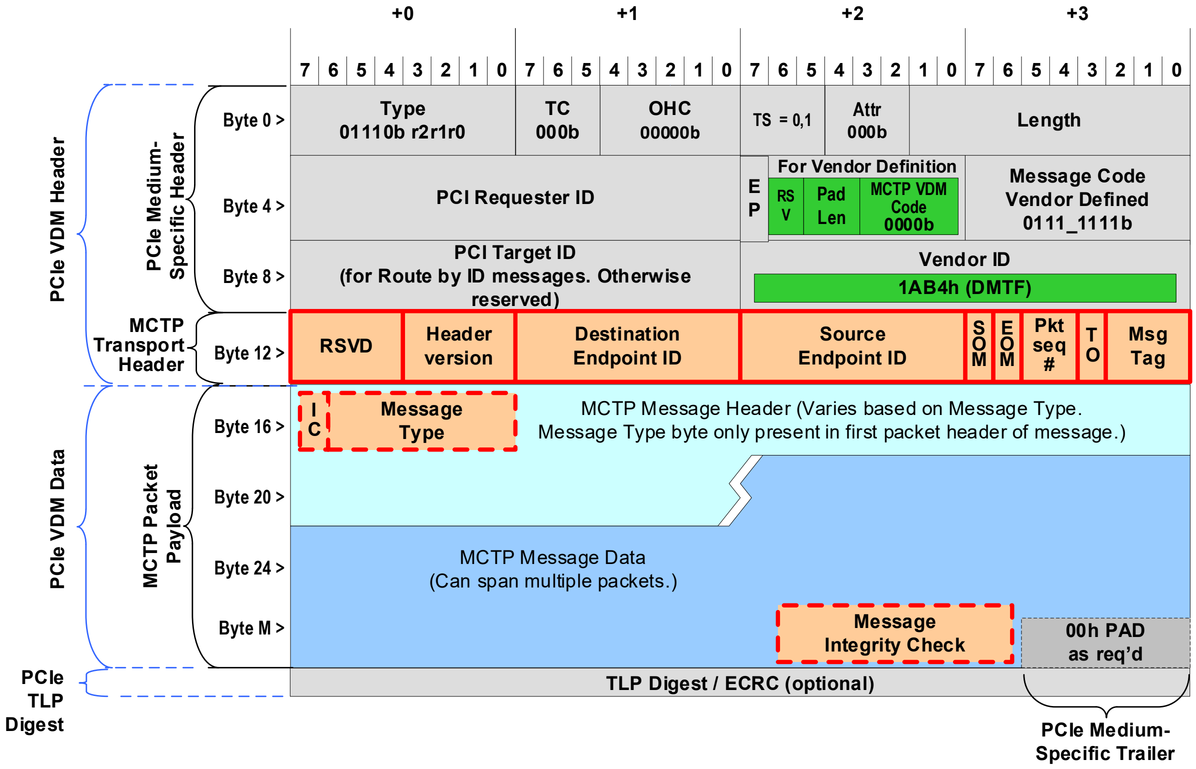

Figure 2 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Flit Mode within a Segment without IDE)

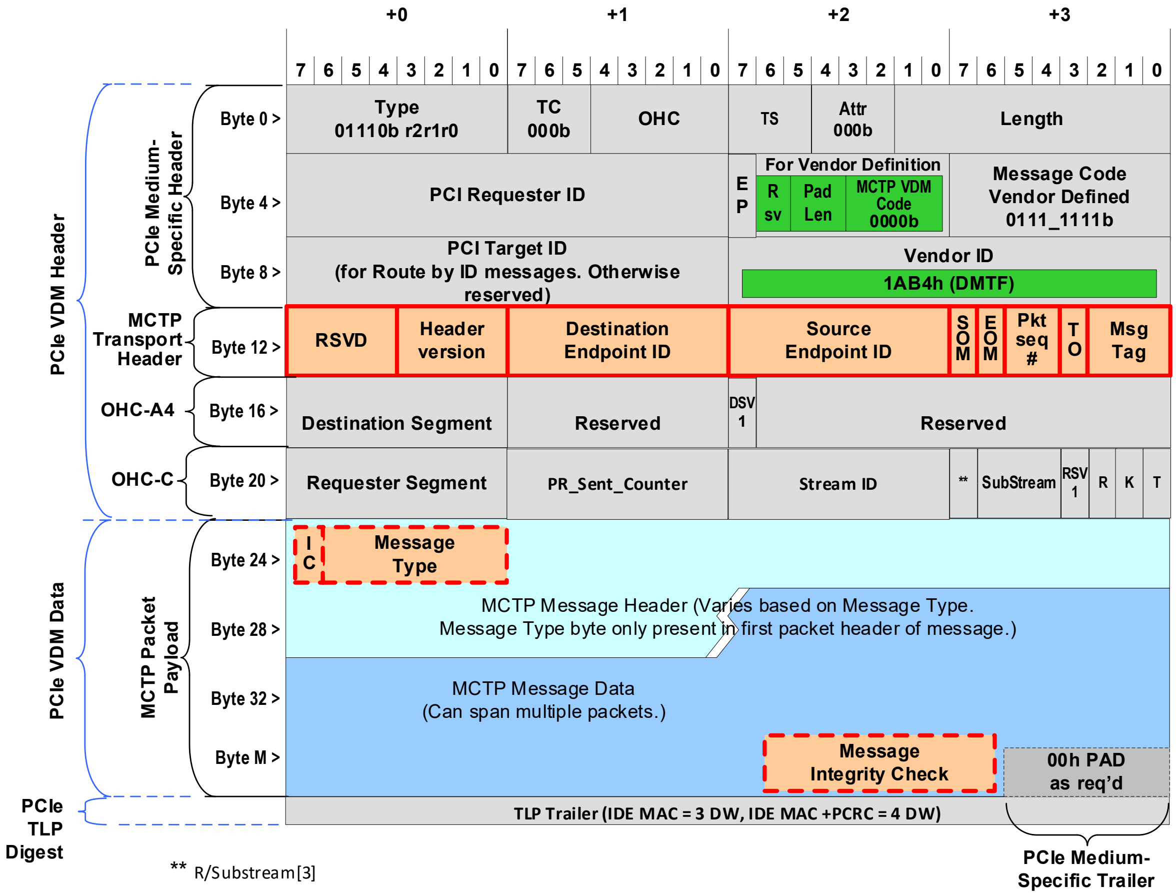

Figure 3 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Flit Mode across segments and/or with IDE)

The fields labeled “PCIe Medium-Specific Header” and “PCIe Medium-Specific Trailer” are specific to carrying MCTP packets using PCIe VDMs. The fields labeled “MCTP Transport Header” and “MCTP Packet Payload” are common fields for all MCTP packets and messages and are specified in MCTP. This section defines the location of those fields when they are carried in a Flit Mode PCIe VDM. The PCIe specification allows the last four bytes of the PCIe VDM header base to be vendor defined. The MCTP over PCIe VDM transport binding specification uses these bytes for MCTP Transport header fields under the DMTF Vendor ID. This document also specifies the *medium-specific* use of the MCTP “Hdr Version” field.

Table 2 lists the PCIe medium-specific fields and field values that shall be used in MCTP over PCIe VDM communications. When not specified, field values shall be set according to PCIe specifications. Note that the presence of additional OHCs in MCTP over PCIe VDM packets is implementation dependent and outside the scope of this specification.

283 **Table 2 – PCI Express medium-specific MCTP Packet Fields in Flit Mode**

Field	Description
PCIe Header	
Type	Type and Routing (8 bits). [7:3] Set to `01110b` to indicate a message [2:0] PCI message routing (`r2r1r0`) `000b` : Route to Root Complex `010b` : Route by ID `011b` : Broadcast from Root Complex Other routing fields values are not supported for MCTP.
TC	Traffic Class (3 bits). Set to `000b` for MCTP over PCIe VDM.
OHC	Indicates the type of OHCs present after the header. Possible values are: 00000b: no OHCs 00100b: Indicates OHC-C is present 00101b: Indicates OHC-A4 and OHC-C are present
TS	TS[2:0] field indicates Trailer Size. The possible encodings for PCIe VDMs are: 000b – No Trailer 001b – 1 dword Trailer containing ECRC 101b – 3 dwords Trailer with IDE MAC if and only if OHC-C present and indicates IDE TLP 110b – 4 dwords Trailer with IDE MAC and PCRC if and only if OHC-C present and indicates IDE TLP
Attr[2:0]	Attributes (3 bits). Set to `000b` for all MCTP over PCIe VDM.
Length	Length: Length of the PCIe VDM Data in dwords. Implementations shall support the baseline transmission unit defined in the *MCTP Base Specification*. For example, supporting a baseline transmission unit of 64 bytes requires supporting PCIe VDM data up to 16 dwords. An implementation may optionally support larger transfer unit sizes.
PCI Requester ID	Bus/device/function or bus/function number of the managed endpoint sending the message.
Pad Len	Pad Length (2-bits). 1-based count (0 to 3) of the number of `0x00` pad bytes that have been added to the end of the packet to make the packet dword aligned with respect to PCIe. Because only packets with the EOM bit set to `1b` are allowed to be less than the transfer unit size, packets that have the EOM bit set to `0b` will already be dword aligned and will thus not require any pad bytes and will have a pad length of `00b`.
MCTP VDM Code	Value that uniquely differentiates MCTP messages from other DMTF VDMs. Set to `0000b` for this transport mapping as defined in this specification.
Message Code	(8 bits). Set to `0111_1111b` to indicate a Type 1 VDM.
PCI Target ID	(16 bits). For Route by ID messages, this is the bus/device/function number or bus/function number that is the physical address of the target endpoint. This field is ignored for Broadcast and for Route to Root Complex messages.
Vendor ID	(16 bits). Set to **6836** (`0x1AB4`) for DMTF VDMs. The most significant byte is in byte 10, the least significant byte is byte 11.
Orthogonal Headers (OHC)	
OHC-A4	Includes the Destination Segment and Destination Segment Valid Note: The PASID and PSV fields in OHC-A4 are not relevant to MCTP VDMs.
OHC-C	Includes Requester Segment, RSV (Requester Segment Valid) and IDE parameters.
MCTP Header	

Field	Description
RSVD	MCTP reserved (4 bits). Set these bits to 0 when generating a message. Ignore them on incoming messages.
Hdr Version	MCTP version (4 bits) 0001b : For MCTP devices that conform to the *MCTP Base Specification* and this version of the PCIe VDM transport binding. All other settings: Reserved to support future packet header field expansion or header version.
MCTP Trailer	
00h PAD	Pad bytes. 0 to 3 bytes of 00h as required to fill out the overall PCIe VDM data to be an integral number of dwords. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned, and will thus not require any pad bytes and will have a pad length of 00b.
PCIe Trailer (3 options if exists)	
TLP Digest / ECRC /	(32 bits). TLP Digest / ECRC (End-to-end CRC).
IDE MAC	(96 bits). IDE Message Authentication Code (MAC).
IDE MAC + PCRC	(128 bits). IDE Message Authentication Code (MAC) + Plaintext CRC

6.3 Supported media

This physical transport binding has been designed to work with the following media as defined in DSP0239 and listed in Table 3. Use of this binding with other types of physical media is not covered by this specification. Refer to DSP0239 for all supported physical media by MCTP transport bindings.

An implementation that is compliant with this specification shall at least support one of the PCIe media listed in Table 3. Note that the CXL is built on the PCI Express (PCIe) physical and electrical interface.

Table 3 – Supported media

Physical Media Identifier	Description
0x08	PCIe revision 1.1 compatible
0x09	PCIe revision 2.0 compatible
0x0A	PCIe revision 2.1 compatible
0x0B	PCIe revision 3.x compatible
0x0C	PCIe revision 4.x compatible
0x0D	PCIe revision 5.x compatible, CXL 1.x/2.x compatible
0x0E	PCIe revision 6.x Non-Flit Mode Compatible,
0x40	PCIe revision 6.x Flit Mode Compatible, CXL 3.X compatible

Note: The compatibility mentioned above is to a specific PCIe specification version and not to a specific speed. For example, a device reporting compatibility to PCIe 4.X may still operate in Gen3 or lower speeds.

6.4 Physical address format for MCTP control messages

The address format shown in Table 4 (Non-Flit Mode) and Table 5 (Flit Mode) is used for MCTP control commands that require a physical address parameter to be returned for a bus that uses this transport binding with one of the supported media types listed in 6.3 This includes commands such as the Resolve Endpoint ID, Routing Information Update, and Get Routing Table Entries commands.

Table 4 – Physical address format (Non-Flit Mode)

Format Size	Address Type	Layout and Description	
2 bytes (BDF ID)	Bus Device Function (BDF)	byte 1	[7:0] – Bus number
		byte 2	[7:3] – Device number [2:0] – Function number
2 bytes (ARI ID)	Alternate Routing Identifier (ARI)	byte 1	[7:0] – Bus number
		byte 2	[7:0] – Function number

Table 5 – Physical address format (Flit Mode)

Format Size	Address Type	Layout and Description	
3 bytes (Segment, BDF ID)	Segment, Bus Device Function (BDF)	byte 1	[7:0] – Segment number
		byte 2	[7:0] – Bus number
		byte 3	[7:3] – Device number [2:0] – Function number
3 bytes (Segment, ARI ID)	Segment, Alternate Routing Identifier (ARI)	byte 1	[7:0] – Segment number
		byte 2	[7:0] – Bus number
		byte 3	[7:0] – Function number

Note: If a message is received in Non-Flit Mode or in Flit Mode without OHC-A4 present, or with OHC-A4 present but DSV bit cleared, then the segment number used to create the physical address is implicitly set to the local segment. In the same manner, to translate a physical address received in Non-Flit Mode format to Flit Mode format, the local segment number shall be added.

6.5 Message routing

Physical packet routing within a PCIe bus uses routing as defined by the PCIe specification. PCIe physical routing/bridging is not the same as MCTP bridging. PCIe physical routing/bridging is generally transparent to MCTP. There are no MCTP-defined functions for configuring or controlling the setup of a PCIe bus. The following types of PCIe addressing are used with MCTP messages:

- **Route by ID**

  All MCTP over PCIe VDM packets between endpoints that are not the bus owner shall use Route by ID for message routing.

  The MCTP bus owner shall use Route by ID for messages to individual MCTP endpoints.

  MCTP endpoints are required to capture the PCIe source physical address (including segment ID in Flit Mode) and the MCTP source EID when receiving an EID assignment MCTP control request message. This is because this request can only be issued by the MCTP bus owner.

  Any MCTP PCIe VDM message received with this routing and Destination EID set to 0xFFh (Broadcast ID) should be dropped.

- **Route to Root Complex (RC)**

  MCTP endpoints shall use this routing for the Discovery Notify request message to the MCTP bus owner as part of the MCTP over PCIe VDM discovery process.

  The MCTP endpoints shall use this routing for responding to the MCTP control request messages that were sent using Broadcast from Root Complex.

  If the MCTP bus owner is in the PCIe root complex, then these messages should be directly accessed by the MCTP bus owner.

  In designs where the PCIe MCTP Bus Owner is not in the PCIe root complex, these messages shall be relayed to the MCTP bus owner. The communication of MCTP PCIe VDM packets that are destined to the MCTP bus owner outside the PCIe root complex relies on the root complex relaying any MCTP over PCIe VDM messages received using Route to Root Complex to the MCTP bus owner. That communication is implementation-specific and is outside the scope of this specification. A receiver of an MCTP PCIe VDM message with this routing may ignore the Destination EID.

- **Broadcast from Root Complex (RC)**

  The MCTP bus owner should use Broadcast from Root Complex (RC) for the transmission of Prepare for Endpoint Discovery and Endpoint Discovery messages.

  If the MCTP bus owner is in the PCIe root complex, the MCTP bus owner should use PCIe root complex for this routing for the Prepare for Endpoint Discovery and Endpoint Discovery messages as part of the MCTP over PCIe VDM discovery process. If the MCTP bus owner is not in the Root Complex, then the MCTP bus owner may use an implementation-specific scheme to signal the transmission of Prepare for Endpoint Discovery and Endpoint Discovery messages.

  If any other MCTP PCIe VDM message is received with this routing, then the message should be dropped. Any MCTP PCIe VDM message using this routing should use a destination EID of 0xFF. A receiver of an MCTP PCIe VDM message with this routing may ignore the Destination EID.

6.5.1 Routing peer transactions on a PCIe bus

The PCIe specification does not require peer-to-peer routing support in PCIe root complexes or switches. For this reason, MCTP over PCIe VDM messages may need to be routed through an MCTP bridge in the MCTP bus owner. When peer-to-peer routing of MCTP PCIe VDM messages between two MCTP endpoints requires routing through a PCIe root complex and the PCIe root complex does not support peer-to-peer routing, then all MCTP over PCIe VDM messages between two MCTP endpoints shall be routed to or through an MCTP bridge located in the PCIe root complex. If the MCTP bus owner is in the PCIe root complex, and the PCIe root complex supports peer-to-peer routing, then the PCIe root complex shall use direct physical addressing to support routing between two MCTP endpoints on the PCIe bus for the non-flit mode.

6.5.2 Routing messages between PCIe and other buses

All MCTP messages that span between PCIe and other buses shall be sent through the MCTP bus owner. The MCTP bus owner has the destination EID routing tables necessary to route messages between the two bus segments.

If an endpoint is aware of multiple routes to a destination over multiple bus types, a higher-level algorithm/protocol above MCTP shall be used to determine which bus/route to use. Typically, this decision can be based on facts such as power state and MCTP discovery state.

6.5.3 Routing Messages in Between Flit Mode and Non-Flit Mode domains

Generally, PCIe rules for routing between segments and presence and values of the Requester Segment and Destination Segment need to be followed so that MCTP can route "transparently" in the presence of PCIe Flit Mode domains.

Figure 4 describes the possible transitions between Flit Mode and Non-Flit Mode domains:

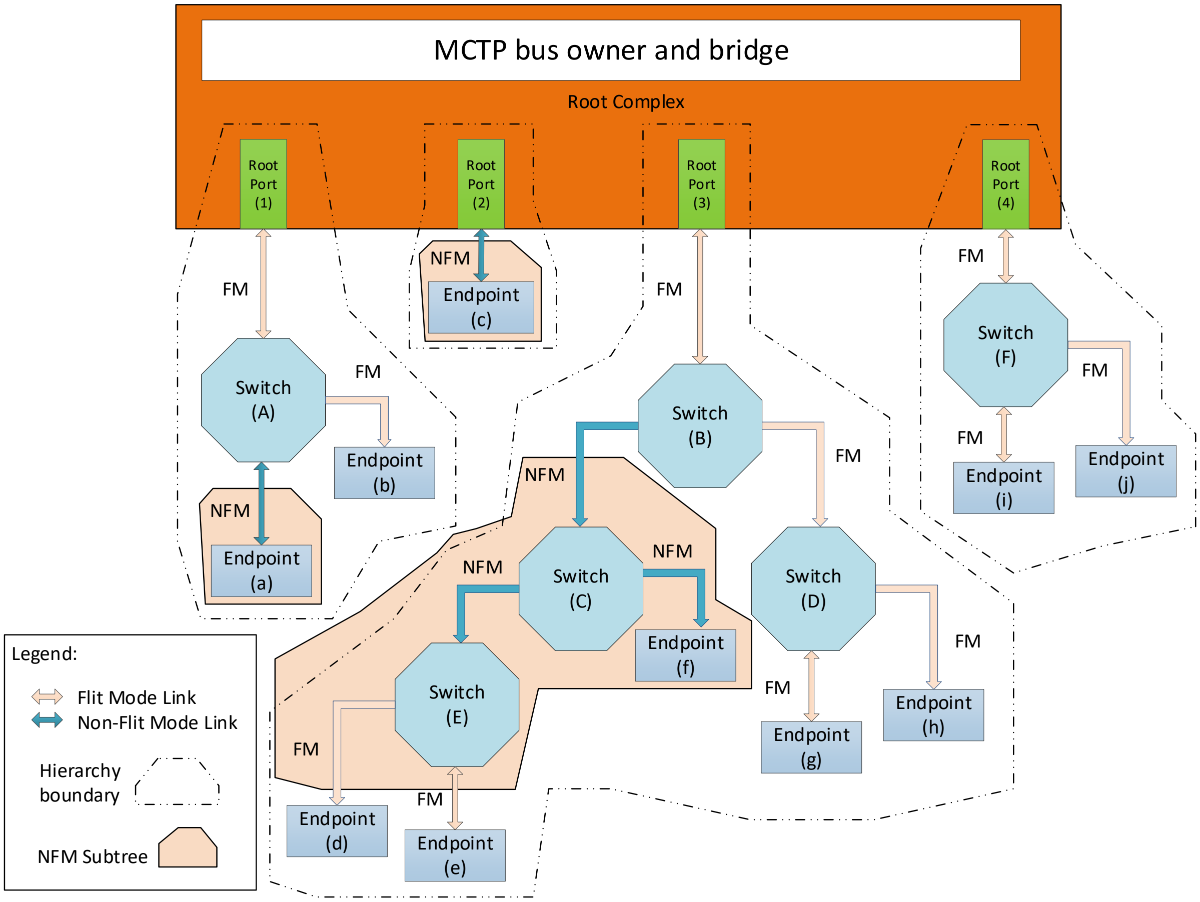

Figure 4 – Flit Mode to Non-Flit Mode routing options

Table 6 – Address Translations when transitioning between Flit Mode and Non-Flit Mode hierarchies

Scenario	Fields translation by Root Port/Bridge	Requires MCTP Bridge?	Example (based on Figure 4)
Flit Mode to Non-Flit Mode or Flit Mode to Flit Mode within segment	Use Target ID and Requester ID as is.	No	(b) to (a) or (i) to (j)

Scenario	Fields translation by Root Port/Bridge	Requires MCTP Bridge?	Example (based on Figure 4)
Flit Mode to Non-Flit Mode across segments	Route to segment indicated by EID lookup. Requester ID in destination segment = {bridge BDF} MCTP bridge shall replace the target ID based on EID lookup.	Yes	(i) to (f)
Flit Mode to Flit Mode across segments	Route to segment indicated by Destination Segment. Target ID kept as is. Requester ID = source Requester ID (no change by MCTP bridge)	No	(j) to (g)
Non-Flit Mode to Flit Mode within segment	Use Target ID and Requester ID as is.	No	(f) to (g)
Non-Flit Mode to Flit Mode across segments	Route to segment indicated by EID lookup. Requester ID in destination segment = {source segment, source BDF} MCTP bridge shall replace the target ID based on EID lookup.	Yes	(f) to (i)
Non-Flit Mode through Flit Mode fabric across segments	Route to segment indicated by EID lookup. Requester ID in destination segment = Bridge ID MCTP bridge shall replace the Target ID based on EID lookup. The MCTP bridge(s) on which Non-Flit Mode island resides shall be aware of it and translate accordingly.	Yes	(e) to (c) or (e) to (i)
Flit Mode to Flit Mode in a Non-Flit Mode island	Use Target ID and Requester ID as is. Segment ID is ignored for the routing. Note: The segment ID seen by these endpoints may be different from the actual ID of the segment as described in the PCIe base specification.	No	(e) to (d)

When a message crosses the boundary between Flit Mode and Non-Flit Mode hierarchies, the Segment part of the address may be removed (Flit Mode to Non-Flit Mode) or added (Non-Flit Mode to Flit Mode). Table 6 describes the translations done when a message crosses between Flit Mode and Non-Flit Mode hierarchies in various scenarios. The “Requires MCTP Bridge” column in Table 6 indicates whether the described scenario requires one or more PCIe Root Ports to function as an MCTP bridge, as DSP0236 describes. The “Example” column in Table 6 provides an example of the described scenario using the topology shown in Figure 4.

To identify the physical address to use, an endpoint may use the Resolve Endpoint ID command that translates an EID to a physical address. As endpoints in Non-Flit Mode may not be aware of the Flit Mode address format, when such an endpoint is requesting the address of a Flit Mode device it should receive a physical address it knows how to use, hence the address shall not include a segment number. In case the resolved EID is in a different segment, the physical address shall point to an MCTP bridge that can translate and route based on the destination EID to a physical address in another segment.

Note: This mechanism assumes the EIDs are unique across all segments, and that physical addressing is not used when crossing between Flit Mode and Non-Flit Mode hierarchies.

A bus owner may discover the Physical medium supported by an endpoint using the Query Supported Interfaces command defined in DSP0236. If the command is not supported, and the bus owner has no other method to detect the supported medium, then a Non-Flit Mode support shall be assumed.

6.5.4 Example of Resolve Endpoint ID response

The table below describes possible responses to Resolve Endpoint ID requests based on the requester and the requested target. The examples are based on Figure 4:

Table 7 – Address used for routing examples

Requester	Requester target	Responding MCTP bus owner	Response (Physical address)
b	a	Root Port 1	{B,D,F} of (a)
i	j	Root Port 4	{S,B,D,F} of (j)
i	f	Root Port 4	{S,B,D,F} of MCTP bridge 4
j	g	Root port 4	{S,B,D,F} of (g)
f	g	Root port 3	{B,D,F} of MCTP bridge 3
f	i	Root port 3	{B,D,F} of MCTP bridge 3
e	c	Root port 3	{B,D,F} of MCTP bridge 3
e	i	Root port 3	{B,D,F} of MCTP bridge 3
e	f	Root port 3	{B,D,F} of (f)

Notes:

- The MCTP bus owner may choose to return its own address instead of the endpoint address in all cases. In this case, all the traffic will be bridged. The chosen routing method is implementation dependent and is outside of the scope of this specification.
- When a {B,D,F} of MCTP bridge is mentioned, it can be any function within the root complex acting as the MCTP bus owner.

6.6 Bus owner address

The MCTP PCIe VDM bus owner functionality shall be accessible through “Route to Root Complex” addressing.

6.7 Bus and Segment address assignment for PCIe

PCIe bus and segment addresses are assigned by the mechanisms specified in PCIe.

6.8 Host dependencies

MCTP over PCIe VDM, when used in a typical “PC” computer system, has a dependency on the host CPU, host software, power management states, link states, and reset. Some of these dependencies are described as follows:

- **Reset**

  Assertion of “Fundamental Reset” or “Conventional Reset” on the bus causes both the host functionality as well as the MCTP PCIe VDM communication on an MCTP endpoint to be reset. From the assertion “Fundamental Reset” or “Conventional Reset” until a physical address is assigned as part of the PCIe fabric enumeration, no MCTP over PCIe VDM messages can be sent to the MCTP endpoint.

  Similarly, if MCTP PCIe VDM communication is supported on a function of a PCIe device, a function level reset (FLR) may reset MCTP PCIe VDM endpoint as well as MCTP PCIe VDM communication on that function. An MCTP PCIe VDM endpoint may not be able to respond to PCIe VDMs during FLR. The PCIe device may use mechanisms outside of this specification to notify function’s FLR to delay any PCIe VDM communication until the FLR processing is complete.

  A PCIe hot reset may reset a MCTP PCIe VDM endpoint. An implementation should retain the EID during a PCIe hot reset.

  An MCTP PCIe VDM endpoint that is reset may need to be reinitialized and/or rediscovered.

- **Configuration and enumeration**

  Following the de-assertion “Fundamental Reset” or “Conventional Reset”, the software running on the host CPU configures and enumerates the PCIe fabric. Failure of the host CPU or boot software to properly configure and enumerate the PCIe fabric prevents it from being used for MCTP over PCIe VDM messaging.

- **Power management states**

  The host (as defined in the context of the PCI Express specification) controls PCIe bus power management. The host may power down PCIe devices and links, or place them in sleep states, independent of management controllers, which may cause MCTP PCIe VDM communication to be unavailable. An MCTP PCIe VDM endpoint on a PCIe device may not be able to respond to PCIe VDMs in low power states or sleep states. The PCIe device may notify using mechanisms outside of this specification of the current power state to delay any PCIe VDM communication until the PCIe device transitions to a state where the PCIe VDM communication is available. Depending on the device usage in the system, a PCIe device may retain or lose states such as EID, “discovered” state, and routing information (if the device is an MCTP bridge). A PCIe device that loses MCTP PCIe VDM communication state needs to be reinitialized and/or rediscovered after it returns to a power state that supports MCTP over PCIe VDM communication.

- **Link states**

  The PCIe link states affect MCTP over PCIe VDM communications. MCTP over PCIe VDM communication can be performed only when the PCIe link is in a state that allows VDM communications. The mechanisms for PCIe link state transitions are outside the scope of this specification.

- **PCIe Root Complex**

  PCIe Root Complex (RC) is responsible for communicating Route to Root Complex MCTP over PCIe VDM discovery messages to the MCTP bus owner.

6.9 Discovery Notify message use for PCIe

An MCTP control Discovery Notify message shall be sent from a PCIe endpoint to the MCTP bus owner whenever the physical address for the device changes (that is, the endpoint receives a Type 0 configuration write request and the bus number and/or the segment number is different from the currently stored bus and segment numbers) and the assigned bus number and optional segment number have been captured by the endpoint. This occurs on the first Type 0 configuration write following a PCIe bus reset during initial enumeration, or during re-enumeration where the bus or segment number has changed (for example, because of a hot plug event, bus reset, and so on).

Endpoints use the Discovery Notify command to inform the MCTP bus owner that it needs to update the endpoint's ID. The Discovery Notify command shall be sent with the PCIe message routing set to `000b` (Route to Root Complex), the Destination Endpoint ID for the Discovery Notify message shall be set to the Null Destination EID. The Source Endpoint ID field shall be set to the Null Source EID if the device has not yet been assigned an EID; otherwise, it shall contain the assigned EID value. If the MCTP bus owner is not in the PCIe root complex, the PCIe root complex provides Discovery Notify command information to the MCTP bus owner using implementation specific mechanisms. These mechanisms are outside the scope of this specification.

6.10 MCTP over PCIe endpoint discovery

This clause describes the steps used to support discovering MCTP endpoints on PCIe.

6.10.1 Discovered flag

Each endpoint (except the bus owner) on the PCIe bus maintains an internal flag called the *Discovered* flag.

The flag is set to the *discovered* state when the Set Endpoint ID command is received.

The Prepare for Endpoint Discovery message causes each recipient endpoint on the PCIe bus to set their respective Discovered flag to the *undiscovered* state. For the Prepare for Endpoint Discovery request message, the routing in the physical transport header should be set to `011b` (Broadcast from Root Complex). An endpoint also sets the flag to the *undiscovered* state at the following times:

- Whenever the PCIe physical address associated with the endpoint is initially assigned or is changed to a different value.
- Whenever an endpoint first appears on the bus and requires an EID assignment. A device shall have been enumerated on PCI and have a bus/device/function or bus/function number before it can do this.
- During operation, if an endpoint enters a state that causes it to lose its EID assignment.
- For endpoints that have already received an EID assignment but are in any temporary state where the endpoint was unable to respond to MCTP control requests for more than $T_{RECLAIM}$ seconds.

Only endpoints that have their Discovered flag set to *undiscovered* shall respond to the Endpoint Discovery message. Endpoints that have the flag set to *discovered* shall not respond to the Endpoint Discovery message.

For PCIe endpoints, an Endpoint Discovery broadcast request message can be sent by the MCTP bus owner to discover all MCTP-capable devices. MCTP-capable endpoints respond with an Endpoint Discovery response message.

An MCTP-capable endpoint shall respond to broadcast MCTP control request messages only if a PCI bus number and in Flit-Mode, a segment number and a bus number are assigned to the associated PCIe function; otherwise, the endpoint should silently discard such MCTP messages.

6.10.2 PCIe endpoint announcement

One or more endpoints may announce their presence and their need for an EID assignment by autonomously sending a Discovery Notify message to the bus owner. This would typically trigger the MCTP bus owner to perform the PCIe endpoint discovery/enumeration processes described in the following subclauses.

6.10.3 Full endpoint Discovery/Enumeration

The following process is typically used when the MCTP bus owner wishes to discover and enumerate all MCTP endpoints on the PCIe bus.

1) The MCTP bus owner issues a broadcast Prepare for Endpoint Discovery message. This message causes each discoverable endpoint on the bus to set its PCIe endpoint Discovered flag to undiscovered. Depending on the number of endpoints and the buffer space available in the MCTP bus owner, the MCTP bus owner may not receive all of the response messages. The discovery process does not require the MCTP bus owner to receive all the response messages to the Prepare for Endpoint Discovery request. Because the MCTP bus owner cannot determine that all endpoints have received the Prepare for Endpoint Discovery request, it is recommended that Prepare for Endpoint Discovery request is retried MN1 times to help ensure that all endpoints have received the request. The MCTP bus owner is not required to wait for MT2 time interval between the retries.

2) The MCTP bus owner should wait for MT2 time interval to help ensure that all endpoints that received the Prepare for Endpoint Discovery request have processed the request.

3) The MCTP bus owner issues a broadcast Endpoint Discovery request message. All MCTP-capable devices that have their Discovered flag set to undiscovered will respond with an Endpoint Discovery response message.

4) Depending on the number of endpoints and the buffer space available in the MCTP bus owner, the MCTP bus owner receives some or all these response messages. For each response message received from an undiscovered MCTP-capable device PCIe bus/device/function or bus/function number, the MCTP bus owner issues a Set Endpoint ID command to the physical address for the endpoint. This causes the endpoint to set its Discovered flag to *discovered*. From this point, the endpoint shall not respond to the Endpoint Discovery command until another Prepare for Endpoint Discovery command is received, or some other condition causes the Discovered flag to be set back to *undiscovered*.

5) If the MCTP bus owner received any responses to the Endpoint Discovery request issued in Step 3, then it shall repeat steps 3 and 4 until it no longer gets any responses to the Endpoint Discovery request. In this case, then the MCTP bus owner is allowed to send the next Endpoint Discovery request without waiting for MT2 time interval. If no responses were received by the MCTP bus owner to the Endpoint Discovery request within the MT2 time interval, then the discovery process is completed.

After the initial endpoint enumeration, it is recommended that the MCTP bus owner maintains a list of the unique IDs for the endpoints it has discovered and reassigns the same IDs to those endpoints if a physical address changes during system operation.

An MCTP-capable endpoint may respond to Route by ID Prepare for Endpoint Discovery and Endpoint Discovery request messages.

Figure 5 provides an example of flow of operations for full endpoint discovery when the MCTP Bus Owner resides in the PCIe RC.

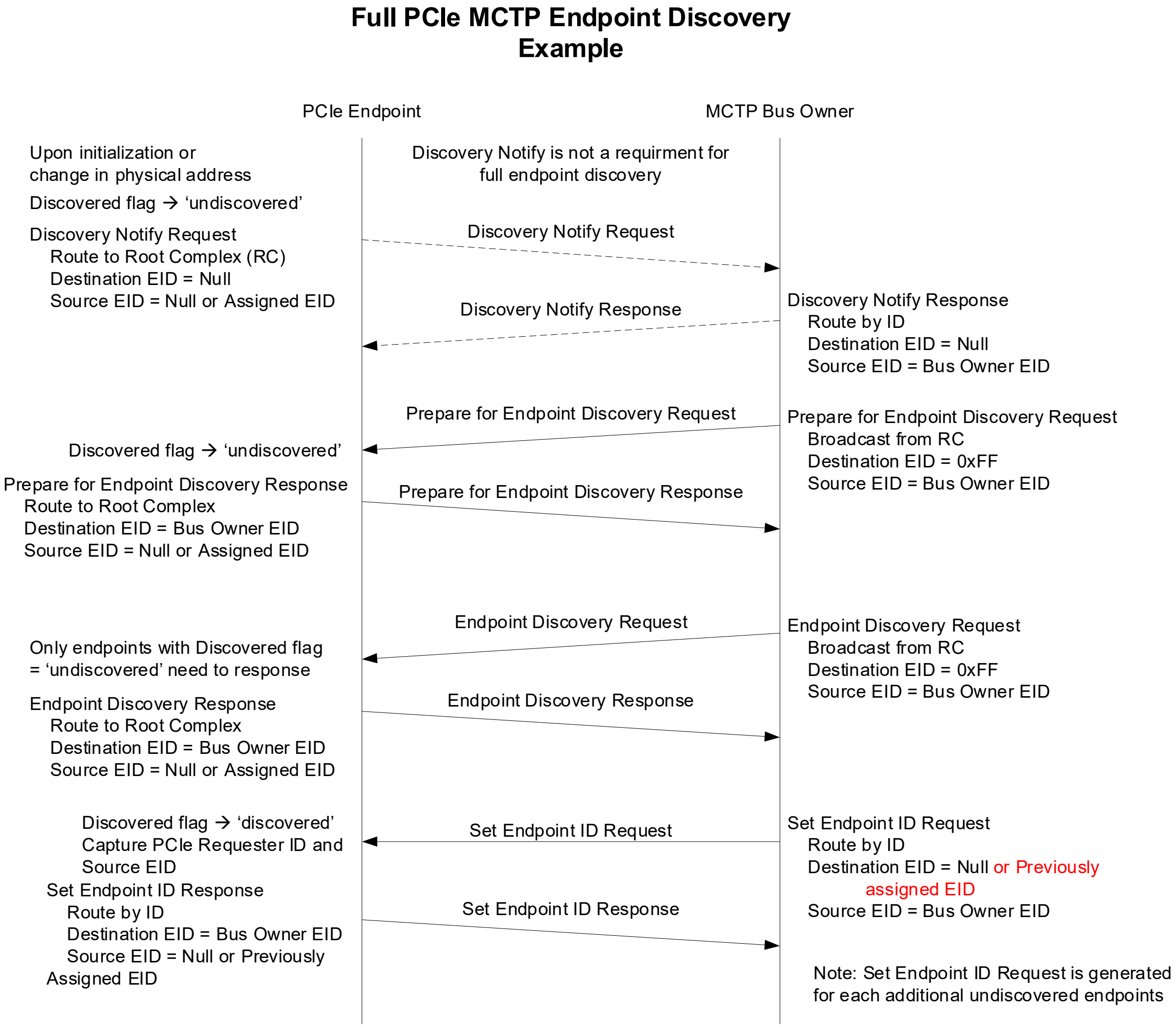
Full PCIe MCTP Endpoint Discovery
Example
PCIe Endpoint
MCTP Bus Owner
Upon initialization or
change in physical address
Discovered flag → 'undiscovered'
Discovery Notify is not a requirment for
full endpoint discovery
Discovery Notify Request
Route to Root Complex (RC)
Destination EID = Null
Source EID = Null or Assigned EID
Discovery Notify Request
Discovery Notify Response
Discovery Notify Response
Route by ID
Destination EID = Null
Source EID = Bus Owner EID
Prepare for Endpoint Discovery Request
Prepare for Endpoint Discovery Request
Broadcast from RC
Destination EID = 0xFF
Source EID = Bus Owner EID
Discovered flag → 'undiscovered'
Prepare for Endpoint Discovery Response
Route to Root Complex
Destination EID = Bus Owner EID
Source EID = Null or Assigned EID
Prepare for Endpoint Discovery Response
Endpoint Discovery Request
Endpoint Discovery Request
Broadcast from RC
Destination EID = 0xFF
Source EID = Bus Owner EID
Only endpoints with Discovered flag
= 'undiscovered' need to response
Endpoint Discovery Response
Route to Root Complex
Destination EID = Bus Owner EID
Source EID = Null or Assigned EID
Endpoint Discovery Response
Discovered flag → 'discovered'
Capture PCIe Requester ID and
Source EID
Set Endpoint ID Response
Route by ID
Destination EID = Bus Owner EID
Source EID = Null or Previously
Assigned EID
Set Endpoint ID Request
Set Endpoint ID Request
Route by ID
Destination EID = Null or Previously
assigned EID
Source EID = Bus Owner EID
Set Endpoint ID Response
Note: Set Endpoint ID Request is generated
for each additional undiscovered endpoints

Figure 5 – Example Flow of operations for full MCTP Discovery over PCIe

Note: In this example flow, the destination EID in Set Endpoint ID Request is set to Null if not assigned before.

6.10.4 Partial endpoint Discovery/Enumeration

This process is used when the MCTP bus owner wishes to discover endpoints that may have been added to the bus after full enumeration has been done. This situation can occur if a device has its physical address change after the full enumeration has been done, or when a hot-plug device is added to the system, or if a device that is already present in the system—but was in a disabled or powered-down state—comes on-line.

The partial discovery process is the same as the full discovery process except that the MCTP bus owner skips the step of broadcasting a Prepare for Endpoint Discovery command to avoid clearing the Discovered flags of already discovered endpoints.

The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a Discovery Notify message to the MCTP bus owner. The MCTP bus owner may also elect to periodically issue a broadcast Endpoint Discovery message to test whether any undiscovered endpoints have been missed. The Discovery Notify message provides the MCTP bus owner with the physical address of the MCTP PCIe endpoint. The MCTP bus owner can then send a direct Endpoint Discovery message to the endpoint to confirm that the device has not been discovered. The MCTP bus owner then issues a Set Endpoint ID command to the physical address for the endpoint which causes the endpoint to set its Discovered flag to *discovered*.

It is recommended that the MCTP bus owner maintains a list of the unique MCTP EIDs for the endpoints it has discovered and reassigns the same MCTP EIDs to those endpoints if a physical address changes during system operation.

An MCTP-capable endpoint may respond to Route by ID Endpoint Discovery request messages.

Figure 6 provides an example of flow of operations for partial endpoint discovery when the MCTP Bus Owner resides in the PCIe RC.

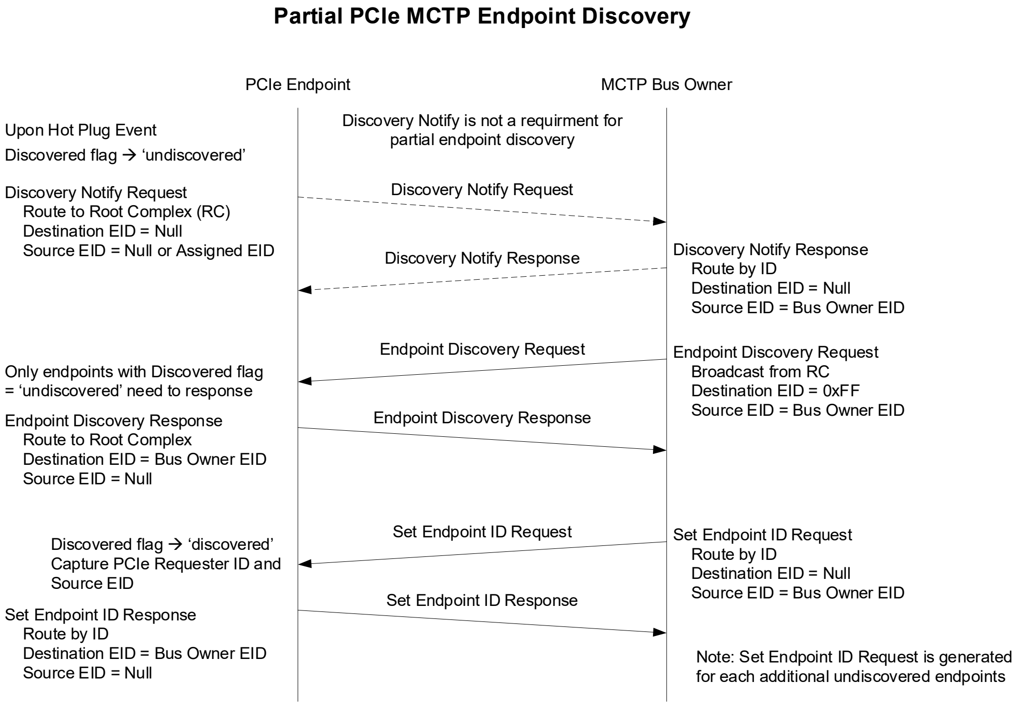
Partial PCIe MCTP Endpoint Discovery
PCIe Endpoint
MCTP Bus Owner
Upon Hot Plug Event
Discovered flag → 'undiscovered'
Discovery Notify is not a requirment for partial endpoint discovery
Discovery Notify Request
Route to Root Complex (RC)
Destination EID = Null
Source EID = Null or Assigned EID
Discovery Notify Request
Discovery Notify Response
Discovery Notify Response
Route by ID
Destination EID = Null
Source EID = Bus Owner EID
Endpoint Discovery Request
Only endpoints with Discovered flag = 'undiscovered' need to response
Endpoint Discovery Request
Broadcast from RC
Destination EID = 0xFF
Source EID = Bus Owner EID
Endpoint Discovery Response
Route to Root Complex
Destination EID = Bus Owner EID
Source EID = Null
Endpoint Discovery Response
Set Endpoint ID Request
Discovered flag → 'discovered'
Capture PCIe Requester ID and Source EID
Set Endpoint ID Request
Route by ID
Destination EID = Null
Source EID = Bus Owner EID
Set Endpoint ID Response
Set Endpoint ID Response
Route by ID
Destination EID = Bus Owner EID
Source EID = Null
Note: Set Endpoint ID Request is generated for each additional undiscovered endpoints

Figure 6 – Flow of operations for Partial Endpoint Discovery

6.10.5 Endpoint re-enumeration

If the bus implementation includes hot-plug devices, the bus owner shall perform a full or partial endpoint discovery any time the MCTP bus owner goes into a temporary state where the MCTP bus owner can miss receiving a Discovery Notify message (for example, if the bus owner device is reset or receives a firmware update). Whether a full or partial endpoint discovery is required is dependent on how much information the MCTP bus owner retains from prior enumerations.

6.11 MCTP messages timing requirements

Table 8 lists MCTP-specific timing requirements for MCTP messages and operation on the PCIe VDM medium. All MCTP Messages over PCIe VDM shall comply with the timing specification listed in Table 8 unless overridden by the specific message type binding.

Table 8 – Timing specifications for MCTP messages on PCIe VDM

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	–	5 sec	Maximum interval that an endpoint is allowed to be non-responsive to MCTP control messages before its EID may be reclaimed by the bus owner. A bus owner shall wait at least for this interval before an EID of the non-responsive endpoint is reclaimed.
Number of request retries	MN1	2	See Description column	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	–	120 ms	This interval is measured at the responder from the end of the reception of an MCTP control request to the beginning of the transmission of the corresponding MCTP control response. This requirement is tested under the condition where the responder can successfully transmit the response on the first try.
Time-out waiting for a response	MT2	MT1 max[1] + 6 ms	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.
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.

Timing Specification	Symbol	Min	Max	Description
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 and conventions

Notations

Examples of notations used in this document are as follows: list into text needed

- 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, the highest is on the right.
- DSP0236 Underlined, blue text is typically used to indicate a reference to a document or specification called out in Clause 2, “Normative References” or to items hyperlinked within the document.
- rsvd Abbreviation for Reserved. Case insensitive.
- [4] Square brackets around a number are typically used to indicate a bit offset. Bit offsets are given as 0-based values (that is, the least significant bit [LSb] offset = 0).
- [7:5] A range of bit offsets. The most significant bit is on the left, the least significant bit is on the right.
- `1b` The lower case “`b`” following a number consisting of 0s and 1s is used to indicate the number is being given in binary format.
- `0x12A` A leading “`0x`” is used to indicate a number given in hexadecimal format.

ANNEX B
(informative)

Change log

Version	Date	Description
1.0.0	2009-07-28	Initial release
1.0.1	2009-10-30	Created erratum to clarify Length field definition of PCIe VDM header for MCTP PCIe VDM transport binding, modify introduction section, and clean up references section.
1.0.2	2014-12-07	Clarifications to TD bit usage. Added TLP Digest/ECRC to packet figure and to field descriptions table.
1.1.0	2018-10-24	Added support for PCIe Gen 3, PCIe Gen 4, and ARI. Fixed Figure 1 to cover PCIe 1.0/2.0/2.1/3.X/4.0. Clarified MCTP over PCIe VDM compliant management device requirements. Clarified Endpoint ID reclaim definition. Clarified MCTP bus owner requirements in the specification. Eliminated PCIe bus owner term and replaced it with PCIe root complex where applicable.
1.2.0	2021-03-02	Added support for PCIe Gen 5.X, CXL 1.X, and CXL 2.X.
1.2.1	2023-12-12	Added a clarification for bus number assignment before responding to broadcast MCTP control messages. Added clarifications that an MCTP-capable endpoint may respond to Route by ID discovery request messages.
1.3.0	2024-08-09	Added PCIe 6.x Flit Mode and Non-Flit mode support. Fixed full endpoint discovery example flow Addressed the following issues: • Removed duplicate reference to DSP0236 • Added default action for non-supported parameters • Clarified the distinction between PCIe gen and PCIe rev. • Clarified the destination EID to use per routing method • Clarified behavior under reset or power down conditions. • Allow route by ID for Endpoint Discovery and Endpoint Discovery request messages. • Generalized timing requirements to all message types.
1.3.1	2025-07-07	Fixed issue #1476 of routing Flit Mode to Non-Flit mode asymmetric routing vs. routing of Non-Flit Mode to Flit mode Fixed issue #1377 When is it safe to send MCTP Discovery Notify over PCIe VDM? Fixed issue #1338 questions on feasibility of an MC as the MCTP PCIe bus owner Fixed issue #1196 Need clarification on EID operation following PCIe Conventional reset