Document Identifier: DSP0238

Date: 2021-03-02

Version: 1.2.0

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

Supersedes: 1.1.0

Document Class: Normative

Document Status: Published

Document Language: en-US

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

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

Editors:

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

Contributors:

- Patrick Caporale – Lenovo
- Yuval Itkin – NVIDIA Corporation
- Eliel Louzoun – 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.

100

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

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 1.0 https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.0.pdf

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 1.0* https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.0.pdf

DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.8* https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.8.pdf

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards,* http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse&sort=subtype

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

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

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

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

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

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

PCI-SIG, *PCI Express® Base Specification Revision 5.0*, May 28, 2019, http://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
MCTP PCIe Endpoint
a PCIe endpoint on which MCTP PCIe VDM communication is supported

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™

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 the 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

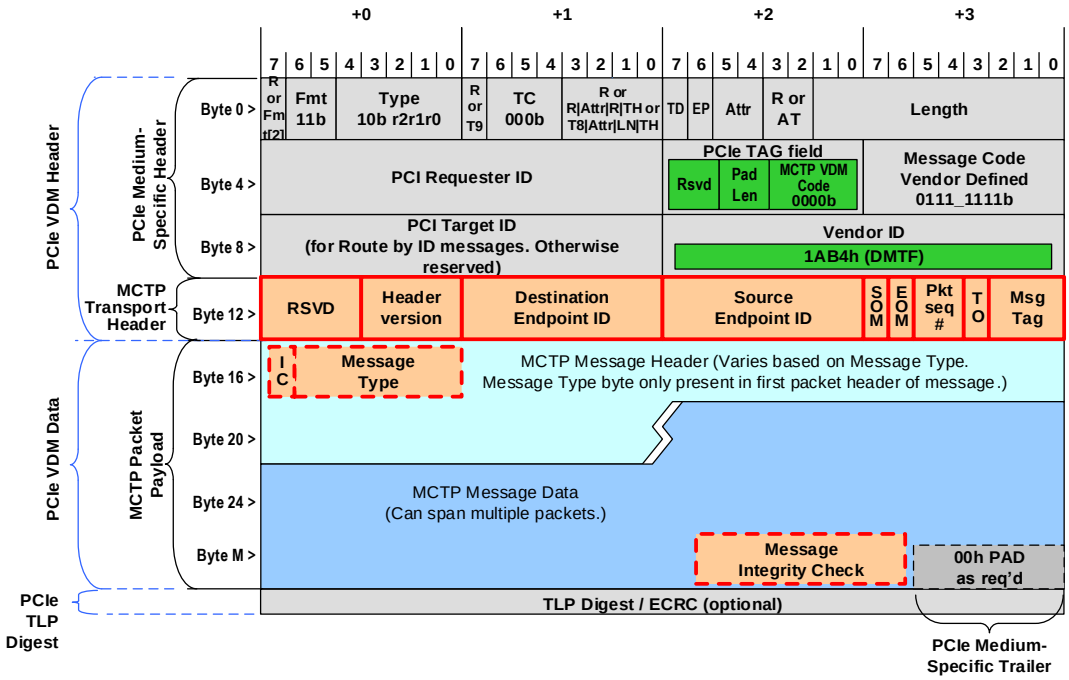
This document defines the medium-specific transport binding for transferring MCTP packets between endpoints on PCI Express™ using PCIe Vendor Defined Messages (VDMs).

A 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 a 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.

6.1 Packet format

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`) that uniquely differentiates MCTP messages from other DMTF VDMs.

Figure 1 shows the encapsulation of MCTP packet fields within a PCIe VDM.



194

195 **Figure 1 – MCTP over PCI Express Vendor Defined Message (VDM) packet format**

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

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

207 **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, 3.X, 4.X, 5.X: 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). Refer to the PCI Express™ specification (PCIe). Set to 0b. PCIe 4.X/5.X: T9 (1bit). Refer to the PCI Express™ specification (PCIe) Gen 4. 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/5.X: 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 2.0/2.1/3.X/4.X/5.X: 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 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 v2.1/3.X/4.X/5.X: "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 v2.1/3.X/4.X/5.X, if the ultimate PCI Express Receiver of the TLP does not support ECRC checking, the receiver must ignore the TLP Digest.

6.2 Supported media

This physical transport binding has been designed to work with the following media as defined in DSP0239 and listed in Table 2. 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 2. Note that the CXL is built on the PCI Express (PCIe) physical and electrical interface.

Table 2 – Supported media

Physical Media Identifier	Description
0x08	PCIe 1.1 compatible
0x09	PCIe 2.0 compatible
0x0A	PCIe 2.1 compatible
0x0B	PCIe 3.X compatible
0x0C	PCIe 4.X compatible
0x0D	PCIe 5.X compatible, CXL 1.X/2.X compatible

6.3 Physical address format for MCTP control messages

The address format shown in Table 3 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 6.2. This includes commands such as the Resolve Endpoint ID, Routing Information Update, and Get Routing Table Entries commands.

Table 3 – Physical address format

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

6.4 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 requester ID 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.

- **Route to root complex**

  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.

  Communication of MCTP PCIe VDM packets that are destined to MCTP bus owner using routed to root complex is implementation specific and is outside the scope of this specification.

- **Broadcast from root complex**

  The MCTP bus owner should use this routing for the Prepare for Endpoint Discovery and Endpoint Discovery messages as part of the MCTP over PCIe VDM discovery process.

6.4.1 Routing peer transactions on a PCIe bus

Because the PCIe specification does not require peer-to-peer routing support in PCIe root complexes, MCTP over PCIe VDM messages are not required to be routed to peer devices directly. When peer-to-peer routing is not supported by a PCIe root complex, all MCTP over PCIe VDM messages between two MCTP endpoints shall be routed to or through the MCTP bus owner as an MCTP bridge. If the PCIe root complex, as the MCTP bus owner, supports peer-to-peer routing, it shall use direct physical addressing to support routing between two MCTP endpoints on the PCIe bus.

6.4.2 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 things like power state and MCTP discovery state.

6.5 Bus owner address

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

6.6 Bus address assignment for PCIe

PCIe bus addresses are assigned per the mechanisms specified in PCIe.

6.7 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” on the bus causes both the host functionality as well as the MCTP PCIe VDM communication on an MCTP PCIe endpoint to be reset. From the assertion “Fundamental Reset” until the PCIe fabric has been configured and enumerated, no “MCTP over PCI Express” messages can be sent.

  Similarly, if MCTP PCIe VDM communication is supported on a function, a function level reset (FLR) could reset MCTP PCIe VDM endpoint as well as MCTP PCIe VDM communication on that function.

- **Configuration and enumeration**

  Following the de-assertion “Fundamental 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. 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 a 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.

6.8 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 is different than the currently stored bus number). This occurs on the first Type 0 configuration write following a PCIe bus reset during initial enumeration, or during re-enumeration where the bus 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.

6.9 MCTP over PCIe endpoint discovery

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

6.9.1 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 PCI bus/device/function or bus/function number 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.

6.9.2 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.

6.9.3 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 of 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 bus/device/function or bus/function number changes during system operation.

Figure 2 provides an example flow of operations for full endpoint discovery.

Full PCIe MCTP Endpoint Discovery

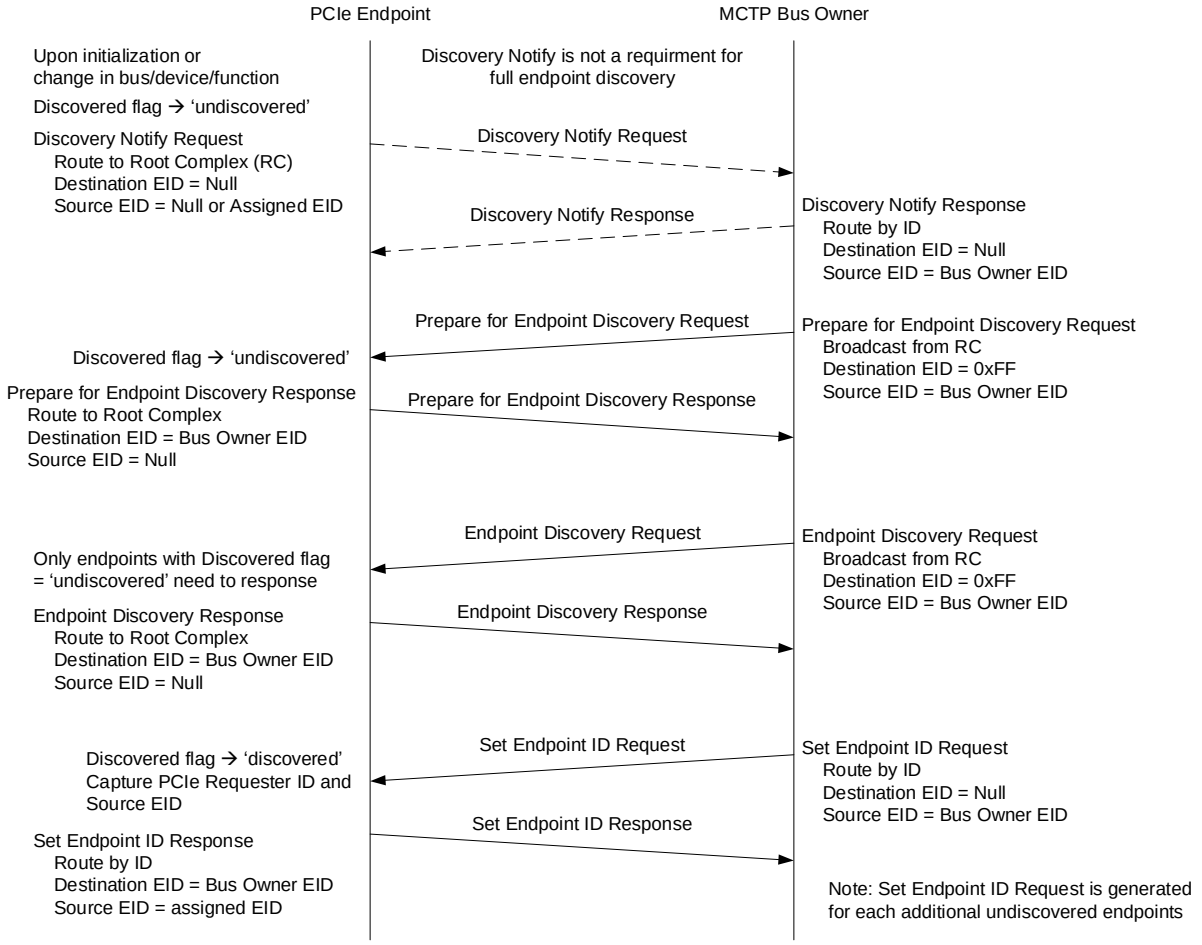


Figure 2 – Flow of operations for full MCTP Discovery over PCIe

6.9.4 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 a full enumeration has been done. This situation can occur if a device has its bus/device/function or bus/function number 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 in order 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 for whether any undiscovered endpoints have been missed. The Discovery Notify message provides the MCTP bus owner with the bus/device/function or bus/function number of the MCTP PCIe endpoint. The MCTP bus owner can then send a directed 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 bus/device/function or bus/function number changes during system operation.

Figure 3 provides an example flow of operations for partial endpoint discovery.

Partial PCIe MCTP Endpoint Discovery

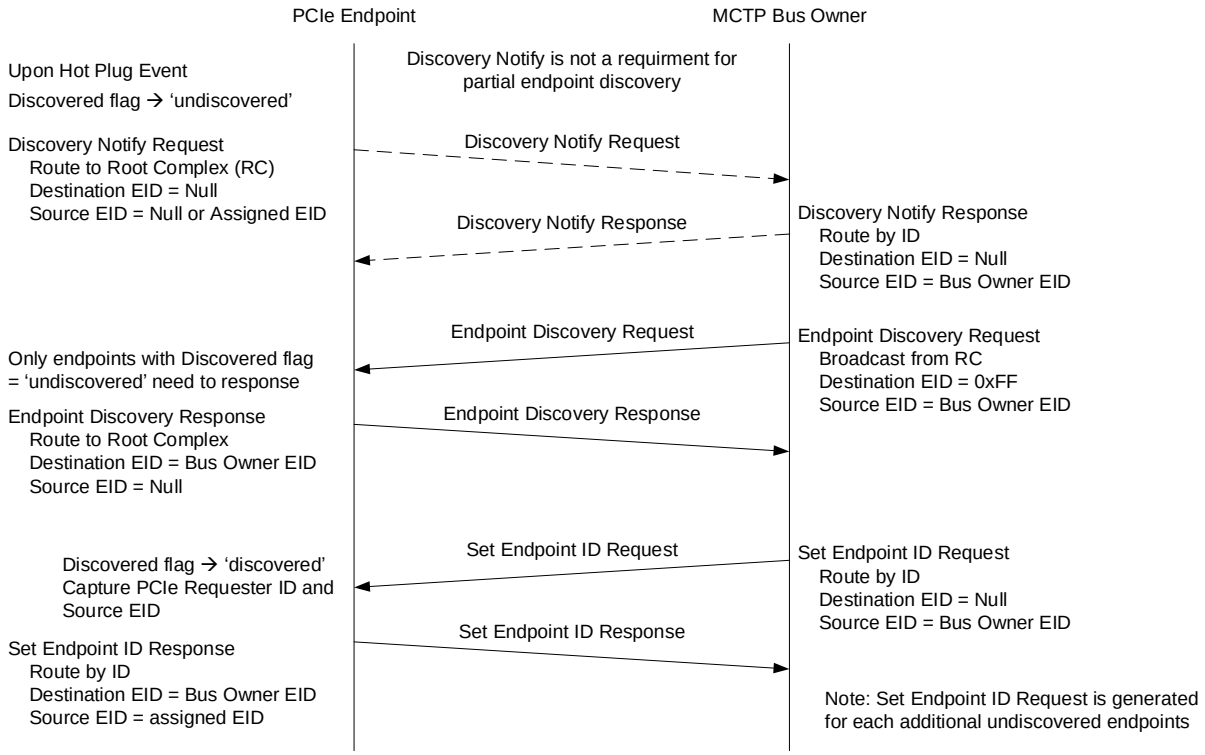


Figure 3 – Flow of operations for Partial Endpoint Discovery

6.9.5 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.

6.10 MCTP messages timing requirements

Table 4 lists MCTP-specific timing requirements for MCTP Control messages and operation on the PCIe VDM medium. All MCTP Control Messages over PCIe VDM shall comply to the timing specification listed in Table 4.

Table 4 – Timing specifications for MCTP Control messages on PCIe VDM

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	–	5 sec	Maximium 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 requirment 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 a 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.
- PCIe — 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	
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.