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

38                                                    CONTENTS

63    **Figures**

68    **Tables**

74 # Foreword

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

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

# 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.

88

89

# 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 for PCI Express™ 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.

DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.0*

DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.0*

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 1.1*, PCIeV1.1, March 28, 2005,
http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_11.pdf

PCI-SIG, *PCI Express® Base Specification 2.0*, PCIeV2.0, December 20, 2006,
http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_2.pdf

PCI-SIG, *PCI Express® Base Specification*, Revision 2.1, March 4, 2009,
https://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_r2_1_04Mar09.pdf

## 3   Terms and Definitions

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™

122 **4.2**
123 **VDM**

124 Vendor Defined Message

# 5 Conventions

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

## 5.1 Reserved and Unassigned Values

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

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

## 5.2 Byte Ordering

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

# 6 MCTP over PCI Express VDM Transport

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

## 6.1 Packet Format

139 The MCTP over PCI Express (PCIe) VDM transport binding transfers MCTP messages using PCIe Type
140 1 VDMs with data. MCTP messages use the MCTP VDM code value (`0000b`) that uniquely differentiates
141 MCTP messages from other DMTF VDMs.

142 Figure 1 shows the encapsulation of MCTP packet fields within a PCIe VDM for PCIe 2.1.
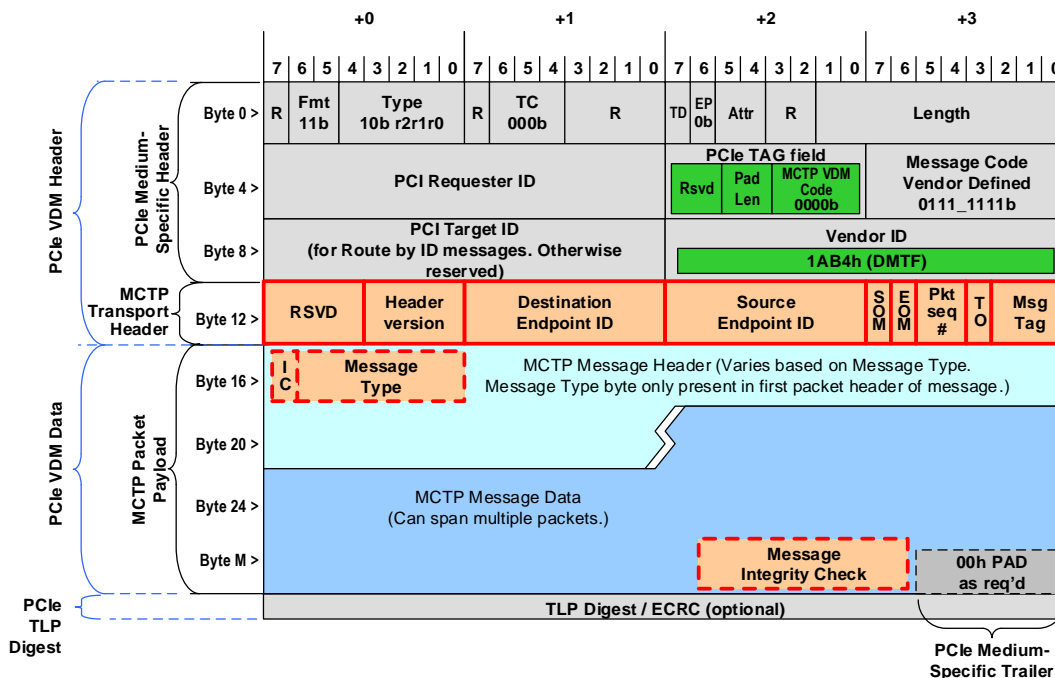
143

144                          **Figure 1 – MCTP over PCI Express Packet Format for PCIe 2.1**

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

152    Table 1 lists the PCIe medium-specific fields and field values.

153                          **Table 1 – PCI Express Medium-Specific MCTP Packet Fields**

| Field | Description |
|---|---|
| R or<br>Fmt[2] | PCIe 1.1/2.0: PCIe reserved bit (1 bit).<br>PCIe 2.1: 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).<br>[4:3] Set to `10b` to indicate a message<br>[2:0] PCI message routing (`r2r1r0`)<br>    `000b` : Route to Root Complex<br>    `010b` : Route by ID<br>    `011b` : Broadcast from Root Complex<br>Other routing fields values are not supported for MCTP. |
| R | PCIe reserved bits (1 bit). Refer to the PCI Express™ specification (PCIe). |
| TC | Traffic Class (3 bits). Set to `000b` for all MCTP over PCIe VDM. |

| Field | Description |
|---|---|
| R or<br>R \| Attr \| R \| TH | PCIe 1.1/2.0: PCIe reserved bits (4 bits).<br>PCIe 2.1: PCIe reserved bit (1 bit), Attr[2] (1 bit) – Set to 0b, reserved bit (1bit), 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 Present (1 bit). Set to `0b` for all MCTP over PCIe VDM. |
| Attr | Attributes (2 bits). Set to `00b` or `01b` for all MCTP over PCIe VDM. |
| R or<br>AT | PCIe 1.1: PCIe reserved bits (2 bits).<br>PCIe 2.0/2.1: 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 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 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)<br>`0001b` : For MCTP devices that conform to the *MCTP Base Specification* and this version of the PCIe VDM transport binding.<br>All other settings: Reserved to support future packet header field expansion or header version. |
| `0x00` PAD | Pad bytes. 0 to 3 bytes of `0x00` 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`. |

| Field | Description |
|---|---|
| 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: "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, 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

155 This physical transport binding has been designed to work with the following media specified in Table 2.
156 Use of this binding with other types of physical media is not covered by this specification.

157                                    **Table 2 – Supported Media**

| Physical Media Identifier | Description |
|---|---|
| 0x08 | PCIe 1.1 compatible |
| 0x09 | PCIe 2.0 compatible |
| 0x0A | PCIe 2.1 compatible |

## 6.3   Physical Address Format for MCTP Control Messages

159 The address format shown in Table 3 is used for MCTP control commands that require a physical
160 address parameter to be returned for a bus that uses this transport binding with one of the supported
161 media types listed in 6.2. This includes commands such as the Resolve Endpoint ID, Routing Information
162 Update, and Get Routing Table Entries commands.

163                                    **Table 3 – Physical Address Format**

| Format Size | Layout and Description | |
|---|---|---|
| 2 bytes | byte 1 | [7:0] – Bus number |
| | byte 2 | [7:3] – Device number |
| | | [2:0] – Function number |

## 6.4   Message Routing

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

169 • **Route by ID**

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

172 The bus owner also can use Route by ID for messages to individual endpoints.

173 PCIe endpoints are required to capture the PCIe requester ID and the MCTP source EID when
174 receiving an EID assignment request message. This is because this command can only be
175 issued by the PCIe bus owner.

176      •     **Route to Root Complex**

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

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

181      •     **Broadcast from Root Complex**

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

184    ### 6.4.1    Routing Peer Transactions

185 Because the PCIe specification does not require peer support in root complexes, MCTP over PCIe
186 messages are not required to be routed to peer devices directly. In this case, all messages between two
187 MCTP endpoints shall be routed to or through the PCIe bus owner as an MCTP bridge. If the PCIe bus
188 does support peer-to-peer routing, the bus owner can support the use of direct physical addressing
189 between endpoints.

190    ### 6.4.2    Routing Messages between PCIe and Other Buses

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

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

197    ## 6.5    Bus Owner Address

198 The PCIe VDM bus owner functionality shall be accessible through "Route-to-Root Complex" addressing.

199    ## 6.6    Bus Address Assignment for PCIe

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

201    ## 6.7    Host Dependencies

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

205      •     **Reset**

206         Assertion of "Fundamental Reset" on the bus causes both the host functionality as well as the
207         manageability portion of an MCTP PCIe endpoint to be reset. From the assertion "Fundamental
208         Reset" until the PCIe fabric has been configured and enumerated, no "MCTP over PCI Express"
209         messages can be sent.

210         Similarly, if MCTP PCI-e VDM communication is supported on a function, a function level reset
211         (FLR) could reset MCTP PCIe endpoint as well as MCTP PCIe VDM communication on that
212         function.

213    • **Configuration and Enumeration**

214    Following the de-assertion "Fundamental Reset", the software running on the host CPU
215    configures and enumerates the PCIe fabric. Failure of the host CPU or boot software to properly
216    configure and enumerate the PCIe fabric prevents it from being used for MCTP messaging.

217    • **Power Management States**

218    The host (as defined in the context of the [PCI Express™ specification](#)) controls PCIe bus power
219    management. The host may power down PCIe devices and links, or place them in sleep states,
220    independent of management controllers, which may cause MCTP PCIe VDM communication to
221    be unavailable. Depending on the device usage in the system, a PCIe device may retain or lose
222    states such as EID, "discovered" state, and routing information (if the device is a bridge). A
223    PCIe device that loses MCTP PCIe VDM communication state needs to be reinitialized and/or
224    rediscovered after it returns to a power state that supports MCTP communication.

225    • **Link States**

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

## 229    6.8    Discovery Notify Message Use for PCIe

230    An MCTP control Discovery Notify message shall be sent from a PCIe endpoint to the PCIe bus owner
231    whenever the physical address for the device changes (that is, the endpoint receives a Type 0
232    configuration write request and the bus number is different than the currently stored bus number). This
233    occurs on the first Type 0 configuration write following a PCIe bus reset during initial enumeration, or
234    during re-enumeration where the bus number has changed (for example, because of a hot plug event,
235    bus reset, and so on).

236    Endpoints use the Discovery Notify command to inform the bus owner that it needs to update the
237    endpoint's ID. The Discovery Notify command shall be sent with the PCIe message routing set to `000b`
238    (Route-to-Root Complex), the Destination Endpoint ID for the Discovery Notify message shall be set to
239    the Null Destination EID. The Source Endpoint ID field shall be set to the Null Source EID if the device
240    has not yet been assigned an EID; otherwise, it shall contain the assigned EID value.

## 241    6.9    MCTP over PCIe Endpoint Discovery

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

### 243    6.9.1    Discovered Flag

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

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

247    The Prepare for Endpoint Discovery message causes each recipient endpoint on the PCIe bus to set their
248    respective Discovered flag to the *undiscovered* state. For the Prepare for Endpoint Discovery request
249    message, the routing in the physical transport header should be set to `011b` (Broadcast from Root
250    Complex).

251    An endpoint also sets the flag to the *undiscovered* state at the following times:

252    • Whenever the PCI bus/device/function number associated with the endpoint is initially assigned
253    or is changed to a different value.

254    • Whenever an endpoint first appears on the bus and requires an EID assignment. A device shall
255    have been enumerated on PCI and have a bus/device/function number before it can do this.

256     • During operation if an endpoint enters a state that causes it to lose its EID assignment.

257     • For hot-plug endpoints that have already received an EID assignment: After exiting any
258       temporary state where the hot-plug endpoint was unable to respond to MCTP control requests
259       for more than $T_{RECLAIM}$ seconds.

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

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

### 6.9.2   PCIe Endpoint Announcement

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

### 6.9.3   Full Endpoint Discovery/Enumeration

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

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

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

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

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

295     5) If the PCIe bus owner received any responses to the Endpoint Discovery request issued in Step
296        3, then it repeats steps 3 and 4 until it no longer gets any responses to the Endpoint Discovery
297        request. In this case, then the PCIe bus owner is allowed to send the next Endpoint Discovery
298        request without waiting for MT2 time interval. If no responses were received by the PCIe bus
299        owner to the Endpoint Discovery request within the MT2 time interval, then the discovery
300        process is completed.

301 After the initial endpoint enumeration, it is recommended that the bus owner maintains a list of the unique
302 IDs for the endpoints it has discovered, and reassigns the same IDs to those endpoints if a
303 bus/device/function number changes during system operation.

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

**Full PCIe MCTP Endpoint Discovery**



305
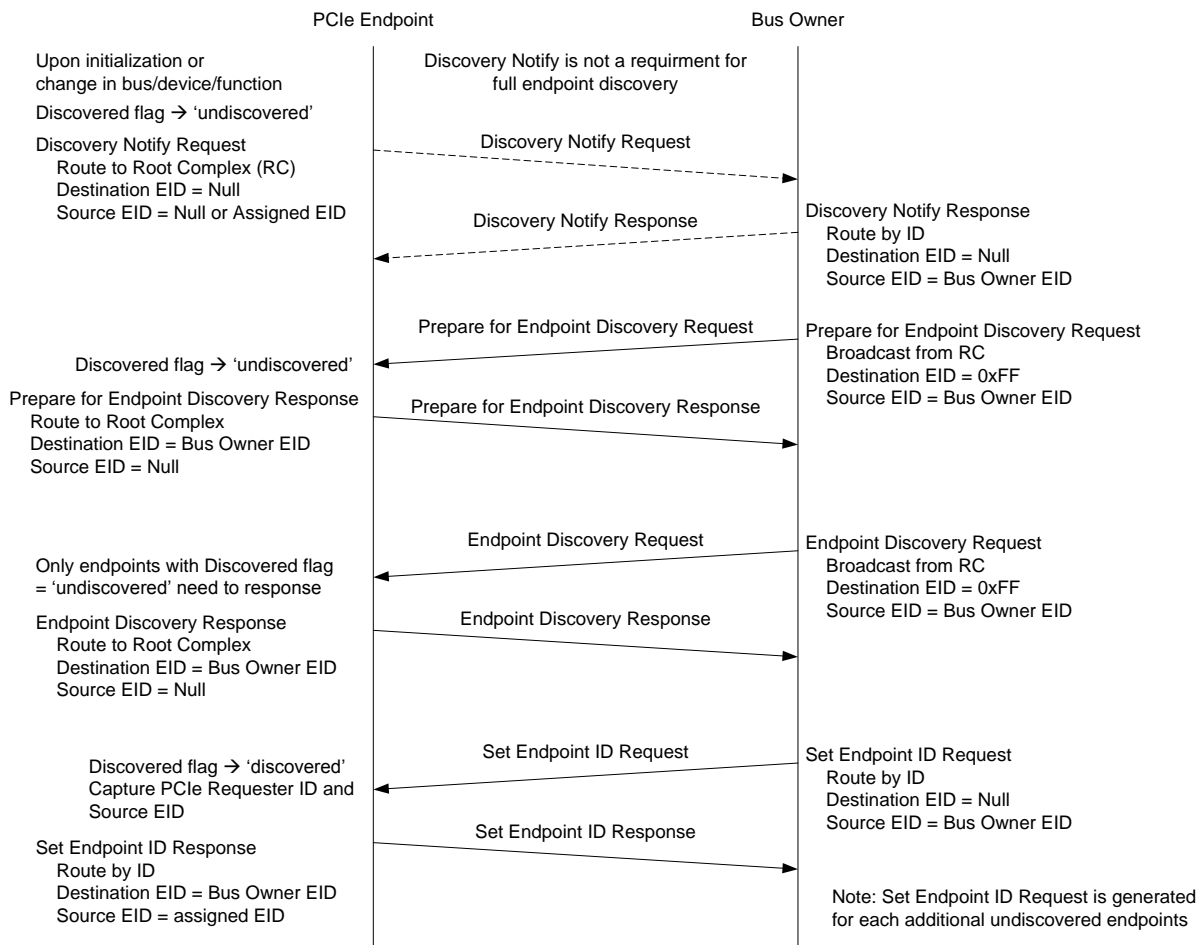
306                 **Figure 2 – Flow of Operations for Full PCIe MCTP Discovery**

### 6.9.4   Partial Endpoint Discovery/Enumeration

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

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

316 The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a
317 Discovery Notify message to the bus owner. The bus owner may also elect to periodically issue a

318    broadcast Endpoint Discovery message to test for whether any undiscovered endpoints have been
319    missed. The Discovery Notify message provides the bus owner with the bus/device/function number of
320    the MCTP PCIe endpoint. The bus owner can then send a directed Endpoint Discovery message to the
321    endpoint to confirm that the device has not been discovered. The bus owner then issues a Set Endpoint
322    ID command to the physical address for the endpoint which causes the endpoint to set its Discovered flag
323    to *discovered*.

324    It is recommended that the bus owner maintains a list of the unique IDs for the endpoints it has
325    discovered, and reassigns the same IDs to those endpoints if a bus/device/function number changes
326    during system operation.

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

**Partial PCIe MCTP Endpoint Discovery**
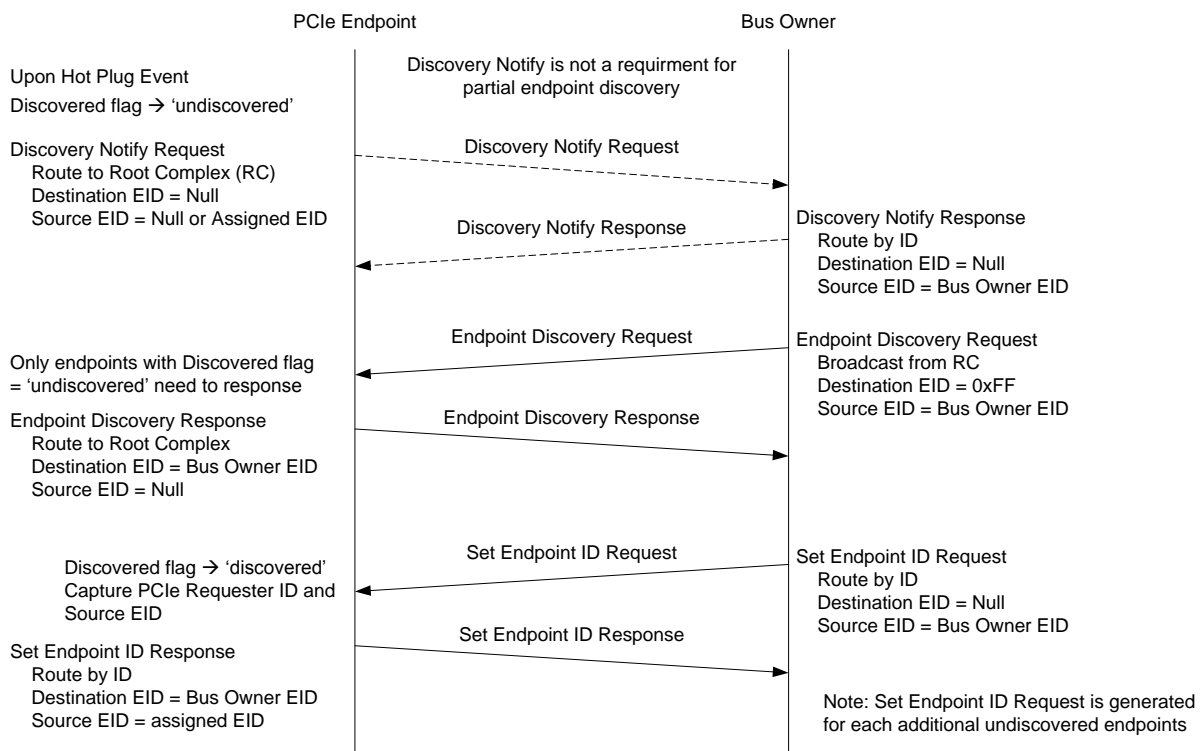


328
329                              **Figure 3 – Flow of Operations for Partial Endpoint Discovery**

330    **6.9.5   Endpoint Re-enumeration**

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

## 336   **6.10 MCTP Messages Timing Requirements**

337   Table 4 lists MCTP-specific timing requirements for MCTP Control messages and operation on the PCIe
338   VDM medium.

339                  **Table 4 – Timing Specifications for MCTP Control Messages on PCIe VDM**

| Timing Specification | Symbol | Min | Max | Description |
|---|---|---|---|---|
| Endpoint ID reclaim | $T_{RECLAIM}$ | – | 5 sec | Maximum interval that a hot-plug endpoint is allowed to be non-responsive to MCTP control messages before its EID can be reclaimed by the bus owner. |
| 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.<br><br>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. |

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.

340          # ANNEX A
341          ## (informative)
342
343          # Notations and Conventions


344    **A.1    Notations**

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

346    - 2:N          In field descriptions, this will typically be used to represent a range of byte offsets
347                   starting from byte two and continuing to and including byte N. The lowest offset is on
348                   the left, the highest is on the right.

349    - (6)          Parentheses around a single number can be used in message field descriptions to
350                   indicate a byte field that may be present or absent.

351    - (3:6)        Parentheses around a field consisting of a range of bytes indicates the entire range
352                   may be present or absent. The lowest offset is on the left, the highest is on the right.

353    - PCIe         Underlined, blue text is typically used to indicate a reference to a document or
354                   specification called out in Clause 2, "Normative References" or to items hyperlinked
355                   within the document.

356    - rsvd         Abbreviation for Reserved. Case insensitive.

357    - [4]          Square brackets around a number are typically used to indicate a bit offset. Bit offsets
358                   are given as 0-based values (that is, the least significant bit [LSb] offset = 0).

359    - [7:5]        A range of bit offsets. The most significant bit is on the left, the least significant bit is
360                   on the right.

361    - 1b           The lower case "b" following a number consisting of 0s and 1s is used to indicate the
362                   number is being given in binary format.

363    - 0x12A        A leading "0x" is used to indicate a number given in hexadecimal format.

364                              # ANNEX B

365                              (informative)

366

367

368                              # 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. |
| 1.02 | 2014-12-07 | Clarifications to TD bit usage. Added TLP Digest/ECRC to packet figure and to field descriptions table. |

369

370