



1  
2  
3  
4

**Document Number: DSP0238**

**Date: 2009-12-11**

**Version: 1.0.1**

5 **Management Component Transport Protocol**  
6 **(MCTP) PCIe VDM Transport Binding**  
7 **Specification**

8 **Document Type: Specification**  
9 **Document Status: DMTF Standard**  
10 **Document Language: E**  
11

12 Copyright Notice

13 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

14 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
15 management and interoperability. Members and non-members may reproduce DMTF specifications and  
16 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to  
17 time, the particular version and release date should always be noted.

18 Implementation of certain elements of this standard or proposed standard may be subject to third party  
19 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations  
20 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,  
21 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or  
22 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to  
23 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,  
24 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or  
25 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any  
26 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent  
27 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is  
28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  
29 implementing the standard from any and all claims of infringement by a patent owner for such  
30 implementations.

31 For information about patents held by third-parties which have notified the DMTF that, in their opinion,  
32 such patent may relate to or impact implementations of DMTF standards, visit  
33 <http://www.dmtf.org/about/policies/disclosures.php>.

34 PCI-SIG, PCIe, and the PCI HOT PLUG design mark are registered trademarks or service marks of PCI-  
35 SIG.

36 All other marks and brands are the property of their respective owners.

37

38

## CONTENTS

39	Foreword .....	4
40	Introduction .....	5
41	1 Scope .....	7
42	2 Normative References.....	7
43	3 Terms and Definitions.....	7
44	4 Symbols and Abbreviated Terms.....	7
45	5 Conventions .....	8
46	5.1 Reserved and Unassigned Values .....	8
47	5.2 Byte Ordering .....	8
48	6 MCTP over PCI Express VDM Transport.....	8
49	6.1 Packet Format.....	8
50	6.2 Supported Media.....	10
51	6.3 Physical Address Format for MCTP Control Messages .....	11
52	6.4 Message Routing .....	11
53	6.5 Bus Owner Address .....	12
54	6.6 Bus Address Assignment for PCIe .....	12
55	6.7 Host Dependencies.....	12
56	6.8 Discovery Notify Message Use for PCIe.....	12
57	6.9 MCTP over PCIe Endpoint Discovery.....	13
58	6.10 MCTP Messages Timing Requirements.....	17
59	Annex A (informative) Notations and Conventions .....	18
60	Annex B (informative) Change Log.....	19
61		

## 62 Figures

63	Figure 1 – MCTP over PCI Express Packet Format for PCIe 1.1 .....	9
64	Figure 2 – Flow of Operations for Full PCIe MCTP Discovery .....	15
65	Figure 3 – Flow of Operations for Partial Endpoint Discovery .....	16
66		

## 67 Tables

68	Table 1 – PCI Express Medium-Specific MCTP Packet Fields .....	9
69	Table 2 – Supported Media.....	10
70	Table 3 – Physical Address Format.....	11
71	Table 4 – Timing Specifications for MCTP Control Messages on PCIe VDM .....	17
72		

73

## Foreword

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

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

78

## Introduction

79 The Management Component Transport Protocol (MCTP) over PCIe VDM transport binding defines a  
80 transport binding for facilitating communication between platform management subsystem components  
81 (e.g. management controllers, management devices) using PCIe Vendor Defined Messages (VDMs).

82 The [MCTP Base Specification](#) describes the protocol and commands used for communication within and  
83 initialization of an MCTP network. The MCTP over PCIe VDM transport binding definition in this  
84 specification includes a packet format, physical address format, message routing, and discovery  
85 mechanisms for MCTP over PCIe VDM communications.

86

87



# 88 Management Component Transport Protocol (MCTP) PCIe 89 VDM Transport Binding Specification

## 90 1 Scope

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

## 93 2 Normative References

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

97 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.0*,  
98 [http://www.dmtf.org/standards/published\\_documents/DSP0236\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0236_1.0.pdf)

99 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.0*,  
100 [http://www.dmtf.org/standards/published\\_documents/DSP0239\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0239_1.0.pdf)

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

103 PCI-SIG, *PCI Express Base Specification 1.1*, PCIeV1.1, March 28, 2005,  
104 [http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI\\_Express\\_Base\\_1.1.pdf](http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_1.1.pdf)

105 PCI-SIG, *PCI Express Base Specification 2.0*, PCIeV2.0, December 20, 2006,  
106 [http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI\\_Express\\_Base\\_2.0.pdf](http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI_Express_Base_2.0.pdf)

## 107 3 Terms and Definitions

108 Refer to [DSP0236](#) for terms and definitions that are used across the MCTP specifications. For the  
109 purposes of this document, the following additional terms and definitions apply.

### 110 3.1

#### 111 MCTP PCIe Endpoint

112 a PCIe endpoint on which MCTP PCIe VDM communication is supported

113

114

## 115 4 Symbols and Abbreviated Terms

116 Refer to [DSP0236](#) for symbols and abbreviated terms that are used across the MCTP specifications. The  
117 following symbols and abbreviations are used in this document.

### 118 4.1

119 PCIe®

120 PCI Express™

121 **4.2**  
122 **VDM**  
123 Vendor Defined Message

## 124 **5 Conventions**

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

### 126 **5.1 Reserved and Unassigned Values**

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

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

### 131 **5.2 Byte Ordering**

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

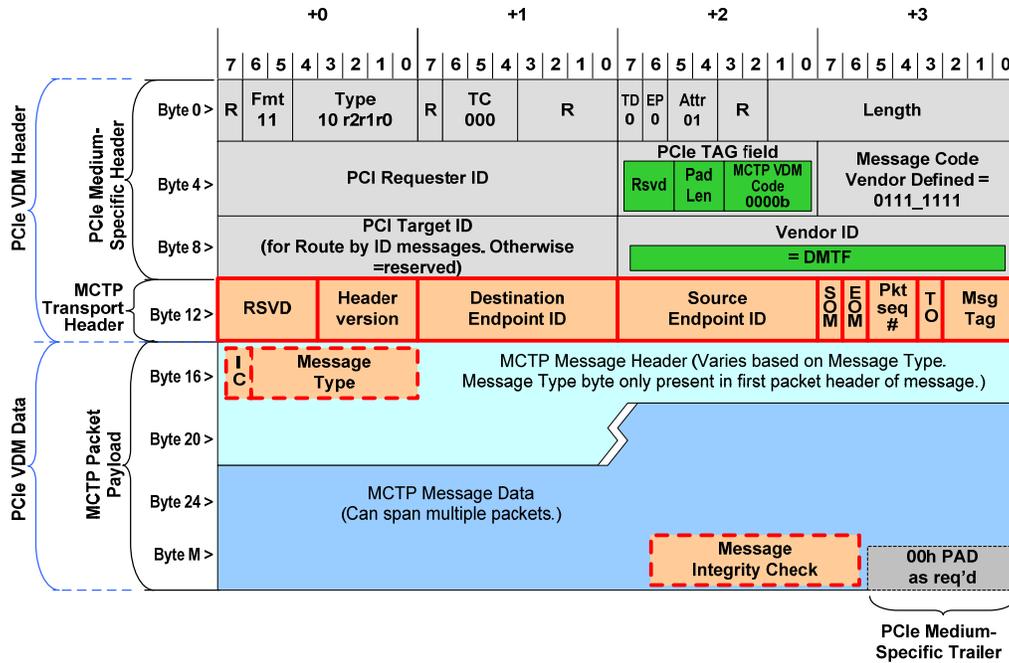
## 134 **6 MCTP over PCI Express VDM Transport**

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

### 137 **6.1 Packet Format**

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

141 Figure 1 shows the encapsulation of MCTP packet fields within a PCIe VDM for PCIe 1.1.



142

143

**Figure 1 – MCTP over PCI Express Packet Format for PCIe 1.1**

144 The fields labeled “PCIe Medium-Specific Header” and “PCIe Medium-Specific Trailer” are specific to  
 145 carrying MCTP packets using PCIe VDMs. The fields labeled “MCTP Transport Header” and “MCTP  
 146 Packet Payload” are common fields for all MCTP packets and messages and are specified in [DSP0236](#).  
 147 This document defines the location of those fields when they are carried in a PCIe VDM. This document  
 148 also specifies the *medium-specific* use of the MCTP “Hdr Version” field.

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

150

**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: 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.
R	PCIe reserved bits (1 bit). Refer to the PCI Express™ specification ( <a href="#">PCIe</a> ).
TC	Traffic Class (3 bits). Set to 000b for all MCTP over PCIe VDM.
R or R   Attr   R   TH	PCIe 1.1/2.0: PCIe reserved bits (4 bits). PCIe 2.1: PCIe reserved bit (1 bit), Attr[2] (1 bit) – Set to 0b, reserved bit (1bit), and TH (1bit) – Set to 0b.

Field	Description
TD	TLP Digest (1 bit). Set to 0b for all MCTP over PCIe VDM.
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 AT	PCIe 1.1: PCIe reserved bits (2 bits). 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 <a href="#">MCTP Base Specification</a> . 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 <b>6836</b> (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 <a href="#">MCTP Base Specification</a> and this version of the PCIe VDM transport binding.  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.

## 151 6.2 Supported Media

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

154

**Table 2 – Supported Media**

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

155 **6.3 Physical Address Format for MCTP Control Messages**

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

160 **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

161 **6.4 Message Routing**

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

- 166 • **Route by ID**

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

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

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

- 173 • **Route to Root Complex**

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

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

- 178 • **Broadcast from Root Complex**

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

181 **6.4.1 Routing Peer Transactions**

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

187 **6.4.2 Routing Messages between PCIe and Other Buses**

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

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

## 194 6.5 Bus Owner Address

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

## 196 6.6 Bus Address Assignment for PCIe

197 PCIe bus addresses are assigned per the mechanisms specified in [PCIe](#).

## 198 6.7 Host Dependencies

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

- 202 • **Reset**

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

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

- 210 • **Configuration and Enumeration**

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

- 214 • **Power Management States**

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

- 222 • **Link States**

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

## 226 6.8 Discovery Notify Message Use for PCIe

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

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

## 238 6.9 MCTP over PCIe Endpoint Discovery

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

### 240 6.9.1 Discovered Flag

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

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

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

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

- 249 • Whenever the PCI bus/device/function number associated with the endpoint is initially assigned  
 250 or is changed to a different value.
- 251 • Whenever an endpoint first appears on the bus and requires an EID assignment. A device shall  
 252 have been enumerated on PCI and have a bus/device/function number before it can do this.
- 253 • During operation if an endpoint enters a state that causes it to lose its EID assignment.
- 254 • For hot-plug endpoints that have already received an EID assignment: After exiting any  
 255 temporary state where the hot-plug endpoint was unable to respond to MCTP control requests  
 256 for more than  $T_{RECLAIM}$  seconds.

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

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

### 262 6.9.2 PCIe Endpoint Announcement

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

### 267 6.9.3 Full Endpoint Discovery/Enumeration

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

- 270 1) The PCIe bus owner issues a broadcast Prepare for Endpoint Discovery message. This  
 271 message causes each discoverable endpoint on the bus to set its PCIe endpoint Discovered  
 272 flag to undiscovered. Depending on the number of endpoints and the buffer space available in  
 273 the PCIe bus owner, the bus owner may not receive all of the response messages. The

274 discovery process does not require the bus owner to receive all the response messages to the  
275 Prepare for Endpoint Discovery request. Because the PCIe bus owner can not determine that  
276 all endpoints have received the Prepare for Endpoint Discovery request, it is recommended that  
277 Prepare for Endpoint Discovery request is retried MN1 times to help ensure that all endpoints  
278 have received the request. The PCIe bus owner is not required to wait for MT2 time interval  
279 between the retries.

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

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

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

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

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

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

Full PCIe MCTP Endpoint Discovery

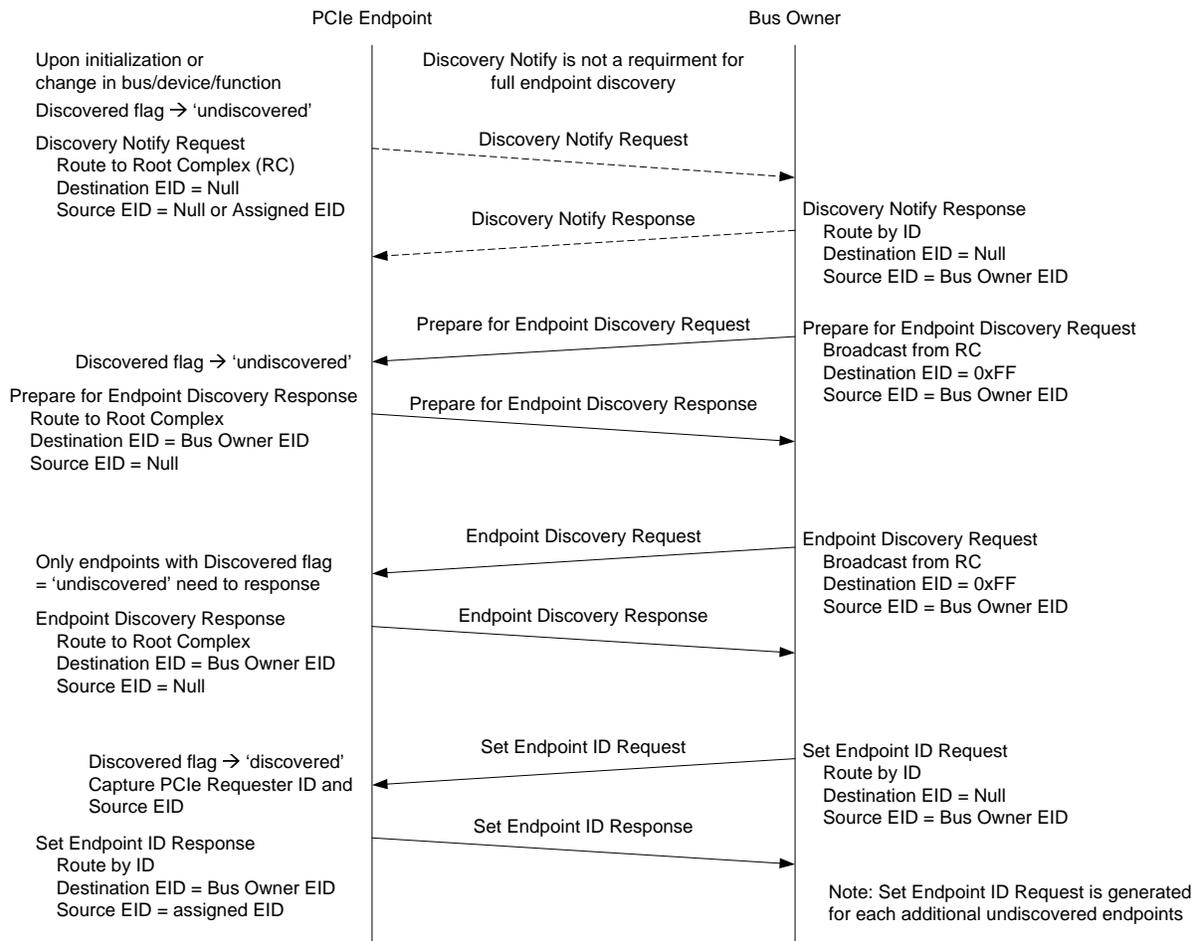


Figure 2 – Flow of Operations for Full PCIe MCTP Discovery

302

303

304 **6.9.4 Partial Endpoint Discovery/Enumeration**

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

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

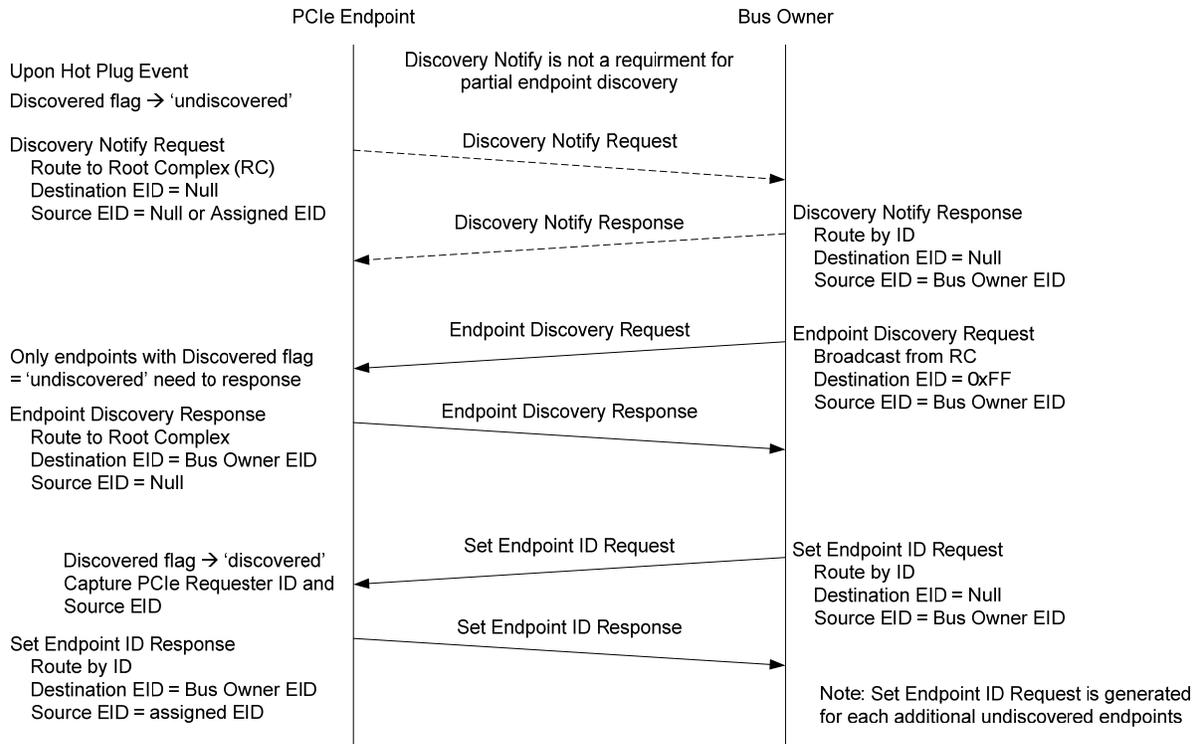
313 The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a  
 314 Discovery Notify message to the bus owner. The bus owner may also elect to periodically issue a  
 315 broadcast Endpoint Discovery message to test for whether any undiscovered endpoints have been  
 316 missed. The Discovery Notify message provides the bus owner with the bus/device/function number of  
 317 the MCTP PCIe endpoint. The bus owner can then send a directed Endpoint Discovery message to the  
 318 endpoint to confirm that the device has not been discovered. The bus owner then issues a Set Endpoint

319 ID command to the physical address for the endpoint which causes the endpoint to set its Discovered flag  
 320 to *discovered*.

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

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

**Partial PCIe MCTP Endpoint Discovery**



325

326

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

327 **6.9.5 Endpoint Re-enumeration**

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

333 **6.10 MCTP Messages Timing Requirements**

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

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

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	–	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 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 <sup>[1]</sup> + 6 ms	MT4, min <sup>[1]</sup>	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 <sup>[2]</sup>	6 sec	Interval after which the instance ID for a given response will expire and become reusable if a response has not been received for the request. This is also the maximum time that a responder tracks an instance ID for a given request from a given requester.
NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.				
NOTE 2: If a requester is reset, it may produce the same sequence number for a request as one that was previously issued. To guard against this, it is recommended that sequence number expiration be implemented. Any request from a given requester that is received more than MT4 seconds after a previous, matching request should be treated as a new request, not a retry.				

## Annex A (informative)

### Notations and Conventions

337  
338  
339  
340

#### 341 A.1 Notations

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

- 343 • 2:N In field descriptions, this will typically be used to represent a range of byte offsets  
344 starting from byte two and continuing to and including byte N. The lowest offset is on  
345 the left, the highest is on the right.
- 346 • (6) Parentheses around a single number can be used in message field descriptions to  
347 indicate a byte field that may be present or absent.
- 348 • (3:6) Parentheses around a field consisting of a range of bytes indicates the entire range  
349 may be present or absent. The lowest offset is on the left, the highest is on the right.
- 350 • PCIe Underlined, blue text is typically used to indicate a reference to a document or  
351 specification called out in Clause 2, "Normative References" or to items hyperlinked  
352 within the document.
- 353 • rsvd Abbreviation for Reserved. Case insensitive.
- 354 • [4] Square brackets around a number are typically used to indicate a bit offset. Bit offsets  
355 are given as 0-based values (that is, the least significant bit [LSb] offset = 0).
- 356 • [7:5] A range of bit offsets. The most significant bit is on the left, the least significant bit is  
357 on the right.
- 358 • 1b The lower case "b" following a number consisting of 0s and 1s is used to indicate the  
359 number is being given in binary format.
- 360 • 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	7/28/2009	DMTF Standard Release
1.0.1	12/11/09	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.

361  
362  
363  
364  
365

366