



1  
2  
3  
4

**Document Identifier: DSP0283**

**Date: 2024-05-21**

**Version: 1.0.1**

5  
6  
7

# **Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) Transport Binding Specification**

8  
9  
10  
11

**Supersedes: 1.0.0**

**Document Class: Normative**

**Document Status: Published**

**Document Language: en-US**

12 Copyright Notice

13 Copyright © 2023–2024 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, or  
21 identify any or all such third-party patent right owners or claimants, nor for any incomplete or inaccurate  
22 identification or disclosure of such rights, owners, or claimants. DMTF shall have no liability to any party,  
23 in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or  
24 identify any such third-party patent rights, or for such party’s reliance on the standard or incorporation  
25 thereof in its products, protocols, or testing procedures. DMTF shall have no liability to any party  
26 implementing such standards, 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28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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 DMTF that, in their opinion, such  
32 patents may relate to or impact implementations of DMTF standards, visit  
33 <https://www.dmtf.org/about/policies/disclosures>.

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

35

## CONTENTS

36	1	Scope .....	6
37	2	Normative references .....	6
38	3	Terms and definitions .....	6
39	4	Symbols and abbreviated terms.....	7
40	5	Conventions .....	7
41	6	MCTP over USB Transport .....	7
42		ANNEX A (informative) Change log.....	20
43			

### 44 **Figures**

45	Figure 1 – Physical topology of USB bus.....	8
46	Figure 2 – Separated USB host and USB Root devices.....	9
47	Figure 3 – A USB Root as MCTP bus owner and MCTP bridge .....	9
48	Figure 4 – An MCTP over USB endpoint as an MCTP bridge.....	10
49	Figure 5 – MCTP 1.x over USB packet format.....	12
50	Figure 6 – USB Bulk transfer principal sequence .....	13
51	Figure 7 – USB Packet with single MCTP packet payload .....	13
52	Figure 8 – USB packet with 2 MCTP packets payload .....	13
53	Figure 9 – Flow of Operations for Full MCTP Discovery over USB .....	15
54	Figure 10 – Flow of Operations for Partial Endpoint Discovery .....	16
55		

### 56 **Tables**

57	Table 1 – MCTP over USB Header Fields .....	12
58	Table 2 – Physical Address Format .....	17
59	Table 3 – Medium-specific information .....	17
60	Table 4 – Timing specifications for MCTP control messages on USB .....	18
61		

62

## Foreword

63 The Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) Transport Binding  
64 Specification (DSP0283) was prepared by the PMCI working group.

65 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
66 management and interoperability. For information about DMTF, see <https://www.dmtf.org>.

67 USB Implementers Forum, Inc. is a non-profit corporation founded by the group of companies that  
68 developed the Universal Serial Bus specification. The USB-IF was formed to provide a support  
69 organization and forum for the advancement and adoption of Universal Serial Bus technology. For  
70 information about USB organization see <https://www.usb.org>.

## 71 Acknowledgments

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

### 73 Editor:

- 74 • Yuval Itkin – NVIDIA Corporation

### 75 Contributors:

- 76 • Patrick Caporale – Lenovo
- 77 • Samer El-Haj-Mahmoud – ARM Inc
- 78 • Michael Garner – Meta
- 79 • Jeff Hilland – Hewlett Packard Enterprise
- 80 • Eliel Louzoun – Intel Corporation
- 81 • Chandra Nelogal – Dell Technologies
- 82 • Edward Newman – Hewlett Packard Enterprise
- 83 • Stevens, Pierre-Philippe – Advanced Micro Devices
- 84 • Paul Sack – Marvell Ltd
- 85 • Hemal Shah – Broadcom Inc.
- 86 • Bob Stevens – Dell Technologies

87

## Introduction

88 The Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) transport binding  
89 defines a transport binding for facilitating MCTP communication between platform management system  
90 components (e.g. management controllers, management devices) over USB 2.0.

91 The [MCTP Base Specification](#) describes the protocol and commands used for communication within and  
92 initialization of an MCTP network. The Management Component Transport Protocol (MCTP) Universal  
93 Serial Bus (USB) transport binding definition in this specification includes a packet format, USB endpoint  
94 descriptors, message routing, and discovery mechanisms for MCTP over USB 2.0 communications.

# 95 Management Component Transport Protocol (MCTP) 96 Universal Serial Bus (USB) Transport Binding Specification

## 97 1 Scope

98 This document provides the specification for the Management Component Transport Protocol (MCTP)  
99 transport binding using USB.

## 100 2 Normative references

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

105 DMTF DSP0004, Common Information Model (CIM) Metamodel 3.0,  
106 [https://www.dmtf.org/standards/published\\_documents/DSP0004\\_3.0.pdf](https://www.dmtf.org/standards/published_documents/DSP0004_3.0.pdf)

107 DMTF DSP0222, Network Controller Sideband Interface (NC-SI) Specification 1.1  
108 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0222\\_1.1.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0222_1.1.pdf)

109 DMTF DSP0223, Generic Operations 1.0,  
110 [https://www.dmtf.org/standards/published\\_documents/DSP0223\\_1.0.pdf](https://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf)

111 DMTF DSP0236, Management Component Transport Protocol (MCTP) Base Specification 1.3  
112 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0236\\_1.3.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.3.pdf)

113 DMTF DSP0239, Management Component Transport Protocol (MCTP) IDs and Codes 1.8  
114 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0239\\_1.8.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.8.pdf)

115 DMTF DSP0256, Management Component Transport Protocol (MCTP) Host Interface Specification 1.0  
116 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0256\\_1.0.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0256_1.0.pdf)

117 DMTF DSP1001, Management Profile Specification Usage Guide 1.1,  
118 [https://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.1.pdf](https://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf)

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

121 USB Implementers Forum, Inc. Universal Serial Bus Specification version 2.0  
122 <https://www.usb.org/document-library/usb-20-specification>

## 123 3 Terms and definitions

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

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

132 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as  
133 described in [ISO/IEC Directives, Part 2](#), Clause 6.

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

137 The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following additional  
138 terms are used in this document.

### 139 3.1

#### 140 MCTP USB Endpoint

141 a USB interface on which MCTP over USB communication is supported

## 142 4 Symbols and abbreviated terms

143 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following  
144 additional abbreviations are used in this document.

### 145 4.1

#### 146 USB

147 Universal Serial Bus

## 148 5 Conventions

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

### 150 5.1 Reserved and Unassigned Values

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

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

### 155 5.2 Byte Ordering

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

## 158 6 MCTP over USB Transport

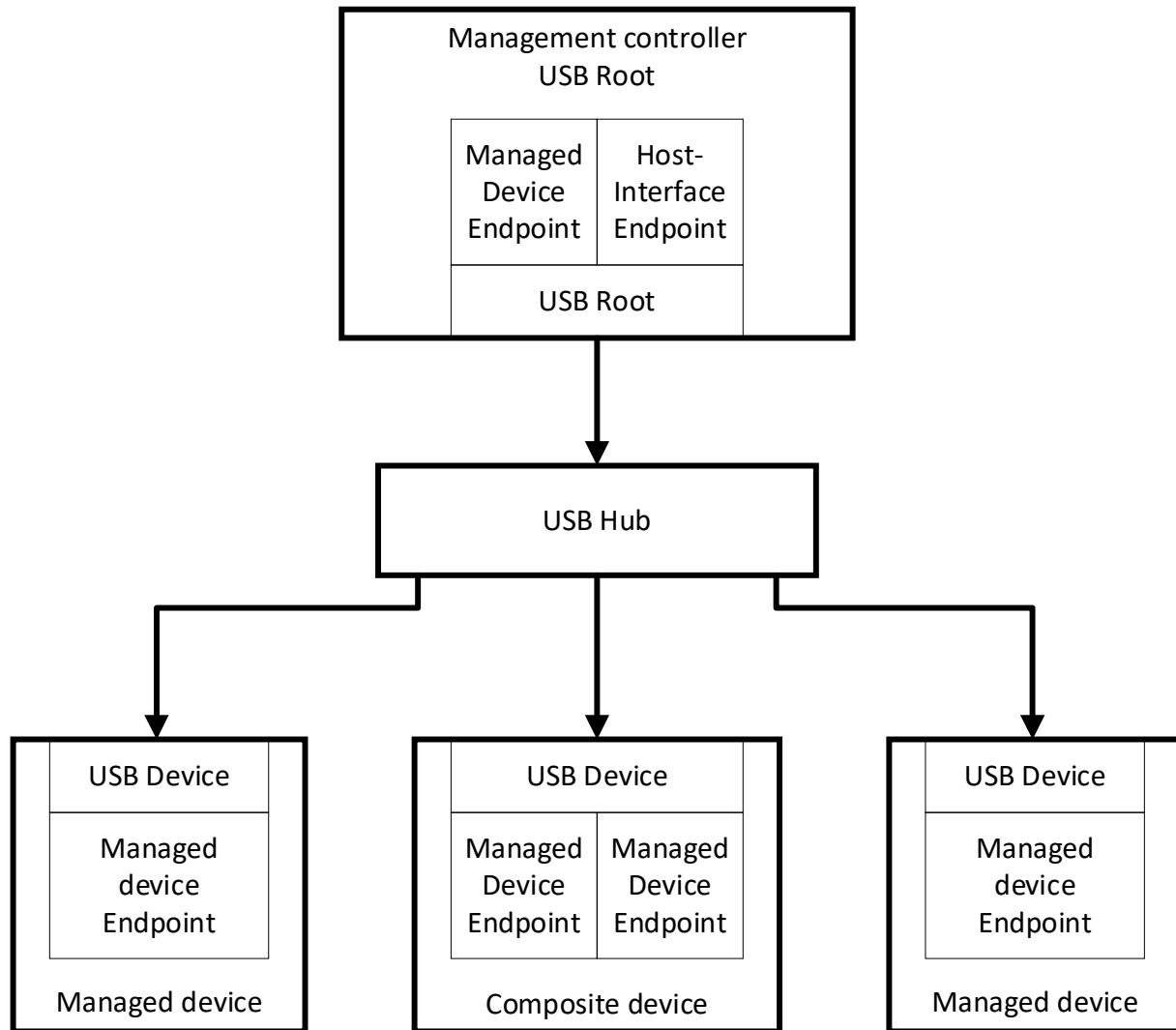
159 This document defines the medium-specific transport binding for transferring MCTP packets between  
160 endpoints on USB using USB Bulk endpoints.

161 A MCTP over USB compliant USB device shall support MCTP over USB communications on at least one  
162 USB interface of the device. If a MCTP over USB compliant USB device supports MCTP over USB  
163 communications on more than one USB interface, then MCTP over USB communication on each such  
164 USB interface shall be independent from MCTP over USB communications on other USB interfaces.

165 **6.1 MCTP use with USB**

166 **6.1.1 USB bus physical topology**

167 The physical topology of the USB bus is presented in Figure 1. There is a single host device that operates  
 168 as the USB tree Root (typically it is a Management Controller, Embedded Controller, etc.) and there may  
 169 be multiple devices sharing the same USB bus tree. A set of USB hubs may be used to enable  
 170 connection of multiple USB devices to the same USB host device.



171

172

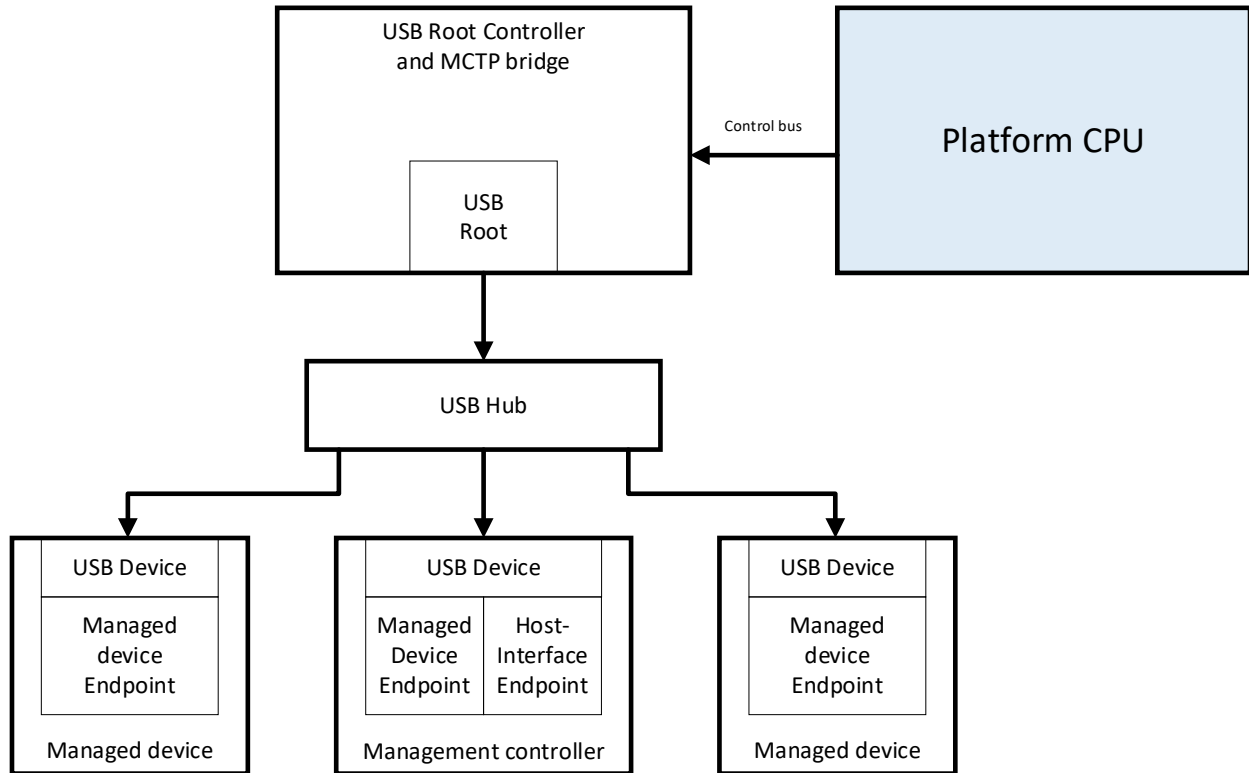
**Figure 1 – Physical topology of USB bus**

173 **6.1.2 MCTP bus owner using USB bus**

174 The MCTP Bus Owner is the USB Root. It is responsible for the discovery and managing the EID  
 175 assignments for the MCTP endpoints on the USB bus. The discovery of the MCTP endpoints is done  
 176 using the provided USB descriptors of each device that has MCTP interface(s) as part of the device  
 177 discovery on the USB bus, as detailed in 6.1.4.

178 The USB host may also be separated from the USB root device. In such a case the USB root is controlled  
 179 by a separate interface as shown in Figure 2.





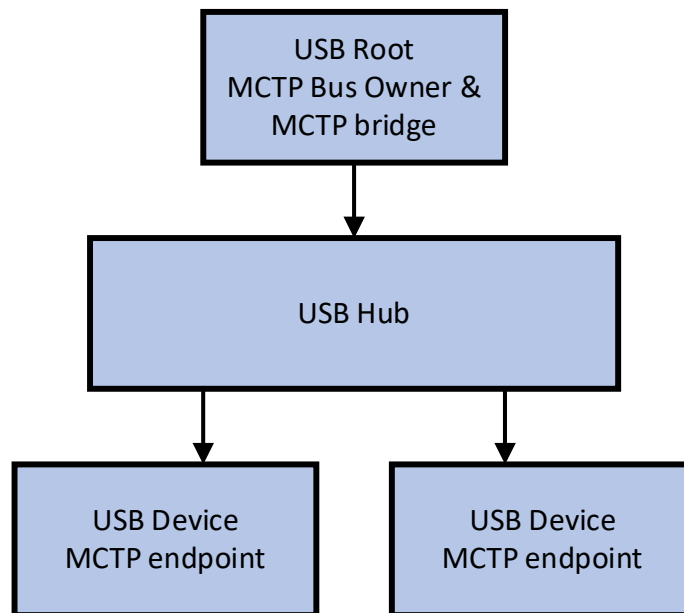
180

181

Figure 2 – Separated USB host and USB Root devices

182 **6.1.3 MCTP bridges over USB**

183 The USB root may act as an MCTP bridge. As USB protocol does not allow direct peer-to-peer  
 184 communication between MCTP endpoints on USB, the USB Root will typically serve as an MCTP bridge  
 185 for all the MCTP endpoints connected to the same USB Root as shown in Figure 3.

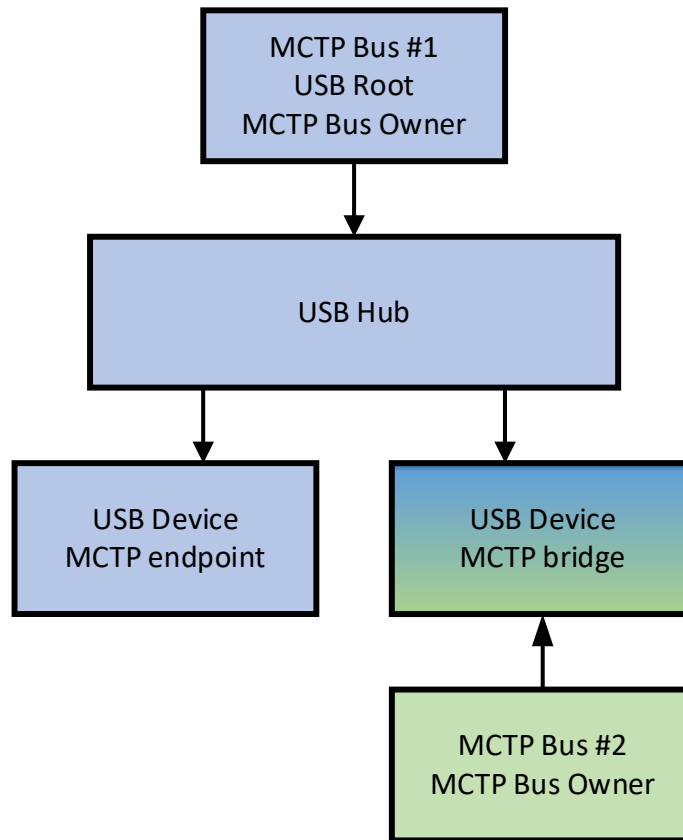


186

187

Figure 3 – A USB Root as MCTP bus owner and MCTP bridge

188 A USB MCTP endpoint can also serve as an MCTP bridge to another MCTP bus as shown in Figure 4.



189

190 **Figure 4 – An MCTP over USB endpoint as an MCTP bridge**

#### 191 6.1.4 Descriptors structure for MCTP endpoint for MCTP over USB

192 An MCTP over USB endpoint is composed of 2 USB Bulk endpoints:

- 193 • Out Bulk endpoint – used to send data from the USB root to the USB device
- 194 • In Bulk endpoint – used to send data from the USB device to the USB root

195 The set of these 2 endpoints is defined as a single USB MCTP interface which is declared by the  
 196 following USB descriptors. A device may have more than one MCTP endpoint. Each such MCTP  
 197 endpoint is an independent USB interface.

198 MCTP over USB is operating in high-speed mode, the endpoint buffer size shall be set to 512 Bytes.

##### 199 6.1.4.1 Interface descriptor

200 For every MCTP endpoint there is a single interface descriptor. This USB interface descriptor defines

- 201 • Class code – A value of 0x14 defines an MCTP endpoint class
- 202 • Sub-Class code
  - 203 • A value of 0x0 defines a Management-controller and Managed-Device endpoints
  - 204 • A value of 0x1 defines an [MCTP Host-Interface](#) endpoint
- 205 • Number of endpoints on the USB MCTP endpoint interface, shall be set to 2

- 206       • Protocol – Class-specific protocol as follows
- 207           • A value of 0x1 defines MCTP 1.x protocol
- 208           • A value of 0x2 defines MCTP 2.x protocol
- 209           • Other values are reserved
- 210       • Alternate settings – shall always be set to 0. An MCTP over USB endpoint shall have no
- 211           alternate settings.

#### 212   **6.1.4.2   Endpoint descriptor**

213   A descriptor is required for every USB Bulk endpoint. Given that there are 2 USB Bulk endpoints for every

214   MCTP interface there are 2 Bulk endpoint descriptors.

215   The 2 Bulk endpoints should use the same USB endpoint number.

##### 216   **6.1.4.2.1   Out Bulk endpoint descriptor**

217   This descriptor declares the USB Bulk endpoint that is used to send data from the USB Root to the USB

218   Device. The following attributes shall be defined in this descriptor:

- 219       • bEndpointAddress – set to the following 8 bits value:
- 220           [7:4] - 0000,
- 221           [3:0] - Bulk\_Endpoint\_Number\_In\_USB\_Device
- 222       • bmAttributes – Set to 0x02 to declare a Bulk endpoint
- 223       • wMaxPacketSize
- 224           • Set to 512, declaring a 512 Bytes buffer size
- 225       • bInterval – set to 0x01
- 226           • High-speed devices, declaring that the host shall not try to access the endpoint again
- 227               during the same micro-frame after receiving a NAK response
- 228               Using this setting minimizes the system idle power by lowering the maximal NAK rate on
- 229               every USB endpoint to 8000 times per second. This sets the maximal additional response
- 230               latency in such a case to 125µsec
- 231               **Implementation note:** While USB specification defines bInterval as a method for setting the maximal
- 232               NAK rate, there are implementations which may not lower the polling rate based on this parameter.

##### 233   **6.1.4.2.2   In Bulk endpoint descriptor**

234   This descriptor declares the USB Bulk endpoint that is used to send data from the USB Device to the

235   USB Root. The following attributes shall be defined in this descriptor

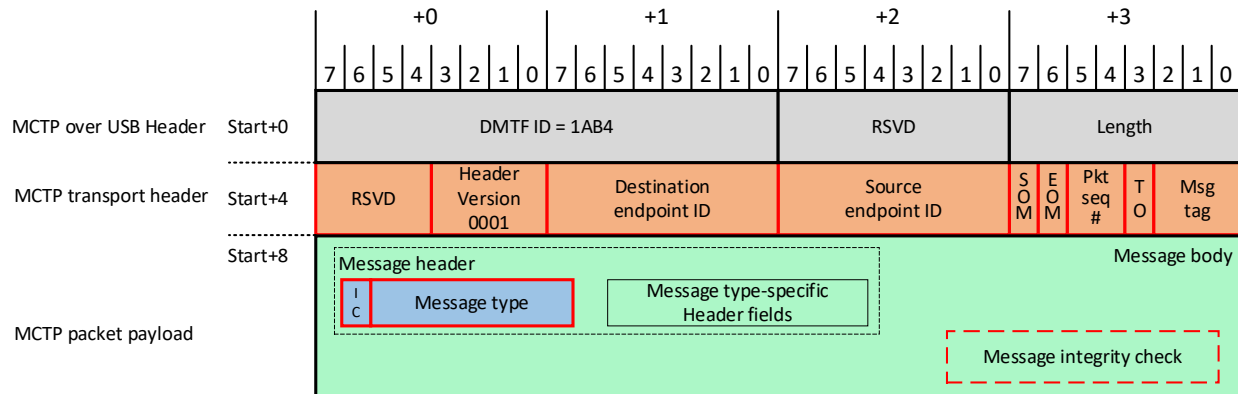
- 236       • bEndpointAddress – set to the following 8 bits value:
- 237           [7:4] - 1000,
- 238           [3:0] - Bulk\_Endpoint\_Number\_In\_USB\_Device
- 239       • bmAttributes – Set to 0x02 to declare a Bulk endpoint
- 240       • wMaxPacketSize
- 241           • Set to 512, declaring a 512 Bytes buffer size
- 242       • bInterval – set to 0x01
- 243           • For high-speed devices, declaring that the host shall not try to access the endpoint again
- 244               during the same micro-frame after receiving a NAK response

245 Using this setting minimizes the system idle power by lowering the maximal NAK rate on  
 246 every USB endpoint to 8000 times per second. This sets the maximal additional response  
 247 latency in such a case to 125µsec

248 **Implementation note:** While USB specification defines bInterval as a method for setting the maximal  
 249 NAK rate, there are implementations which may not lower the polling rate based on this parameter.

250 **6.2 Packet Format**

251 The use of [USB](#) bulk endpoint for MCTP over USB does require adding a medium-specific header for  
 252 each MCTP packet as shown in Figure 5 – MCTP 1.x over USB packet format below.



253

254 **Figure 5 – MCTP 1.x over USB packet format**

255 The fields in the “MCTP over USB Header” are specific to carrying MCTP packets using USB Bulk  
 256 transfers. The fields labeled “MCTP transport header” and “MCTP packet payload” are common fields for  
 257 all MCTP packets and messages and are specified in [MCTP](#). This document defines the location of those  
 258 fields when they are carried in a USB Bulk transfer.

259 Table 1 lists the MCTP over USB Header fields and values.

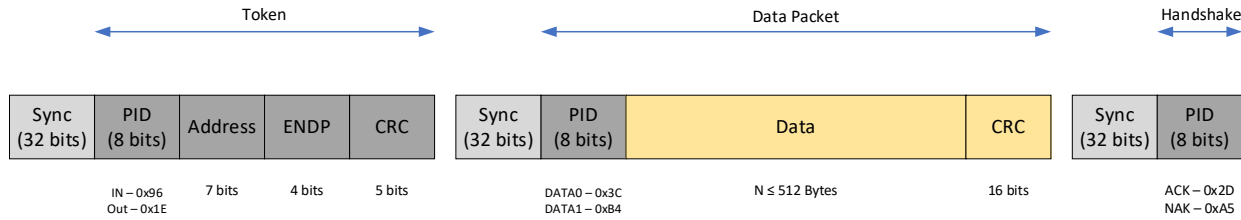
260

**Table 1 – MCTP over USB Header Fields**

Byte offset	Field	Description
0	DMTF ID	DMTF Identifier. Always set to 0x1AB4, matching DMTF Vendor ID as registered in PCI-Sig.
2	Reserved	MCTP Reserved (8 bits). Shall always be set to 0 when generating a packet. Shall be ignored on receive.
3	Length	Length: Length of the MCTP over USB packet in Bytes, starting from the “MCTP over USB Header” to the last byte in the “MCTP packet payload”, implementations shall support the baseline transmission unit defined in <a href="#">DSP0236</a> .

261 The [MCTP](#) packets are sent as the data to the designated USB Bulk endpoint. MCTP traffic over USB  
 262 shall use High-Speed (480M bits-per-second) mode.

263 Figure 6 illustrates HS Bulk data transfer. Every transfer starts with a Token which indicates the data  
 264 transfer direction and the addressed device and endpoint. Following the token, the data packet is  
 265 transferred (IN or OUT), and after the data packet transfer is complete, a handshake PID is used to  
 266 ACK/NAK the transfer. The Token and the Data packet include CRC as shown below.



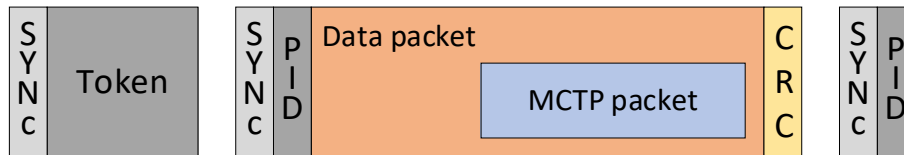
267

268

**Figure 6 – USB Bulk transfer principal sequence**

269 [USB](#) specification does not require data payloads to always be exactly the endpoint buffer size.  
 270 Therefore, if a data payload is less than the endpoint buffer size, it does not need to be padded to the  
 271 endpoint buffer size.

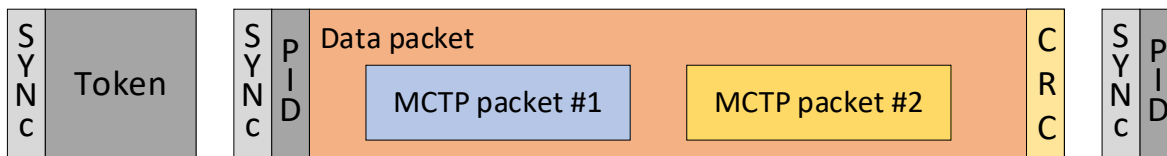
272 The [MCTP](#) packet cannot be larger than the endpoint buffer size. The payload of the [USB](#) packet  
 273 contains any combination of one or more MCTP packets destined to or through the same Endpoint-ID  
 274 (EID). Refer to Figure 7 – USB Packet with single MCTP packet payload, Figure 8 – USB packet with 2  
 275 MCTP packets payload.



276

277

**Figure 7 – USB Packet with single MCTP packet payload**



278

279

**Figure 8 – USB packet with 2 MCTP packets payload**

### 280 **6.3 Error handling**

281 [USB](#) Bulk data transfers reliability is ensured in hardware level using error detection and by invoking a  
 282 limited number of retries. If the retry count is exceeded, the interface shall be reset using a method that is  
 283 out of scope for this specification.

### 284 **6.4 MCTP support and capabilities discovery**

285 An MCTP-capable MCTP over USB bus-owner, shall discover all the MCTP capable interfaces on the  
 286 USB fabric as described below.

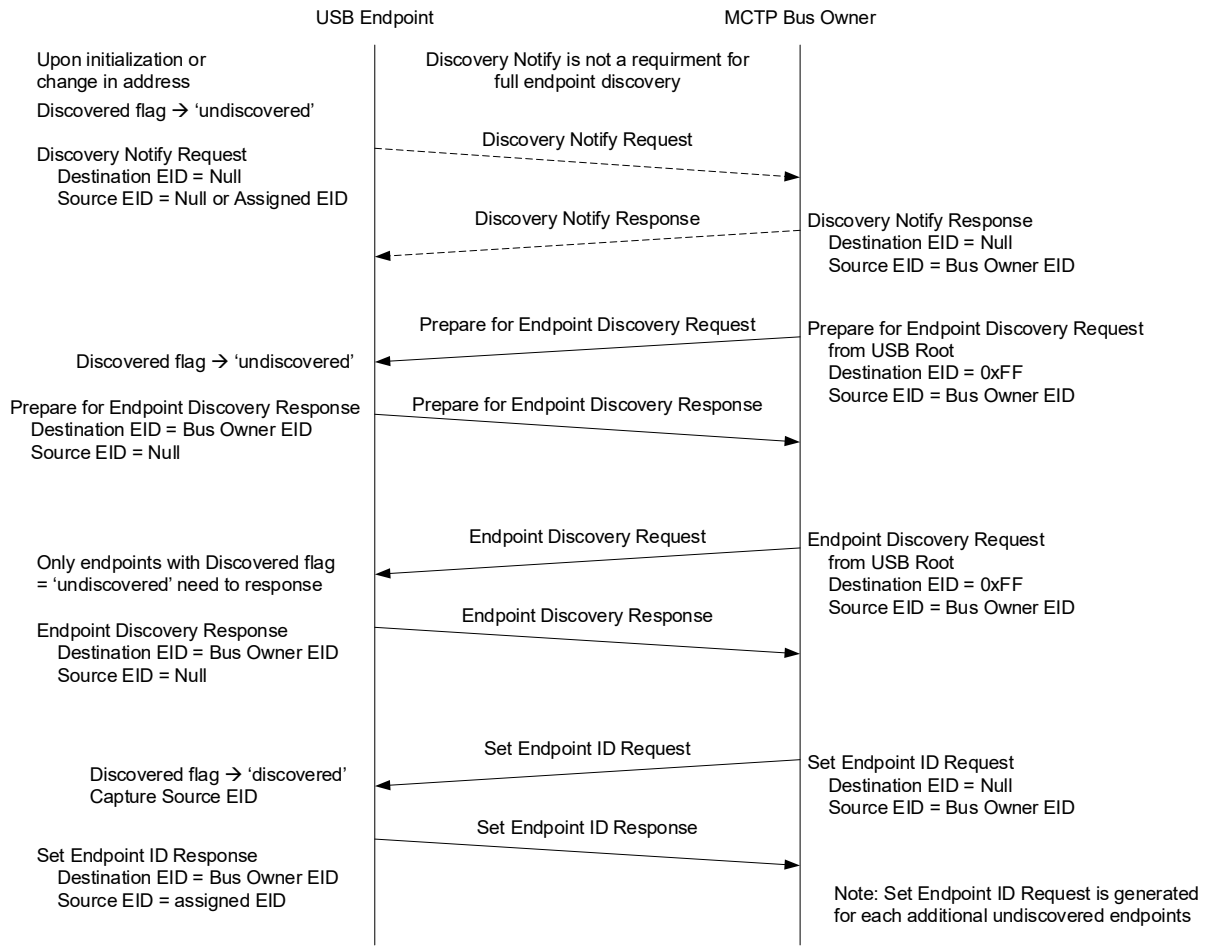
#### 287 **6.4.1 Full Endpoint Discovery/Enumeration**

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

- 290 1) MCTP-capable devices are identified by their USB descriptors as defined in section 6.1.4.
- 291 During USB detection and enumeration phase.

- 292 2) Following its USB enumeration, an MCTP-capable device shall send the *Discovery Notify*  
293 MCTP message , to request EID assignment. A USB interface of a [composite device](#) with more  
294 than one MCTP endpoint shall send the *Discovery Notify* MCTP message for every MCTP  
295 endpoint separately.
- 296 3) The MCTP bus owner issues a Prepare for Endpoint Discovery message for every MCTP-  
297 capable device using the Broadcast EID as the destination EID. When addressing a [composite](#)  
298 [device](#) with more than one MCTP endpoint, the MCTP bus owner shall issue a Prepare for  
299 Endpoint Discovery message for every MCTP-endpoint on that MCTP-capable device using the  
300 Broadcast EID as the destination EID.  
301 This message causes each discoverable endpoint on the bus to set its USB endpoint  
302 Discovered flag to undiscovered.
- 303 4) All MCTP-capable devices that have their Discovered flag set to undiscovered will respond with  
304 an Endpoint Discovery response message.
- 305 5) The MCTP bus owner should wait for at least MT2 time interval to receive the response. This  
306 helps ensure, that all endpoints that received the Prepare for Endpoint Discovery request have  
307 processed the request.
- 308 6) The MCTP bus owner issues an Endpoint Discovery request message for every MCTP endpoint  
309 on an MCTP-capable device using the Broadcast EID as the destination EID. When addressing  
310 a [composite device](#) with more than one MCTP endpoint, the MCTP bus owner shall issue an  
311 Endpoint Discovery message for every MCTP-capable interface using the Broadcast EID as the  
312 destination EID.
- 313 7) For each response message received from an undiscovered MCTP interface of an MCTP-  
314 capable USB device, the MCTP bus owner issues a Set Endpoint ID command to the physical  
315 address for the endpoint. This causes the endpoint to set its Discovered flag to *discovered*.  
316 From this point, the endpoint shall not respond to the Endpoint Discovery command until  
317 another Prepare for Endpoint Discovery command is received, or some other condition causes  
318 the Discovered flag to be set back to *undiscovered*.
- 319 8) If the MCTP bus owner received any responses to the Endpoint Discovery request issued in  
320 Step 6, then it shall repeat steps 6 and 7 until it no longer gets any responses to the Endpoint  
321 Discovery request. In this case, then the MCTP bus owner is allowed to send the next Endpoint  
322 Discovery request without waiting for MT2 time interval. If no responses were received by the  
323 MCTP bus owner to the Endpoint Discovery request within the MT2 time interval, then the  
324 discovery process is completed.
- 325 After the initial endpoint enumeration, it is recommended that the MCTP bus owner maintains a list of the  
326 unique IDs for the endpoints it has discovered and reassigns the same IDs to those endpoints if a USB  
327 endpoint number changes during system operation.
- 328 Figure 9 provides an example flow of operations for full endpoint discovery.

Full USB MCTP Endpoint Discovery



329

330

Figure 9 – Flow of Operations for Full MCTP Discovery over USB

331 **6.4.2 Partial Endpoint Discovery/Enumeration**

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

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

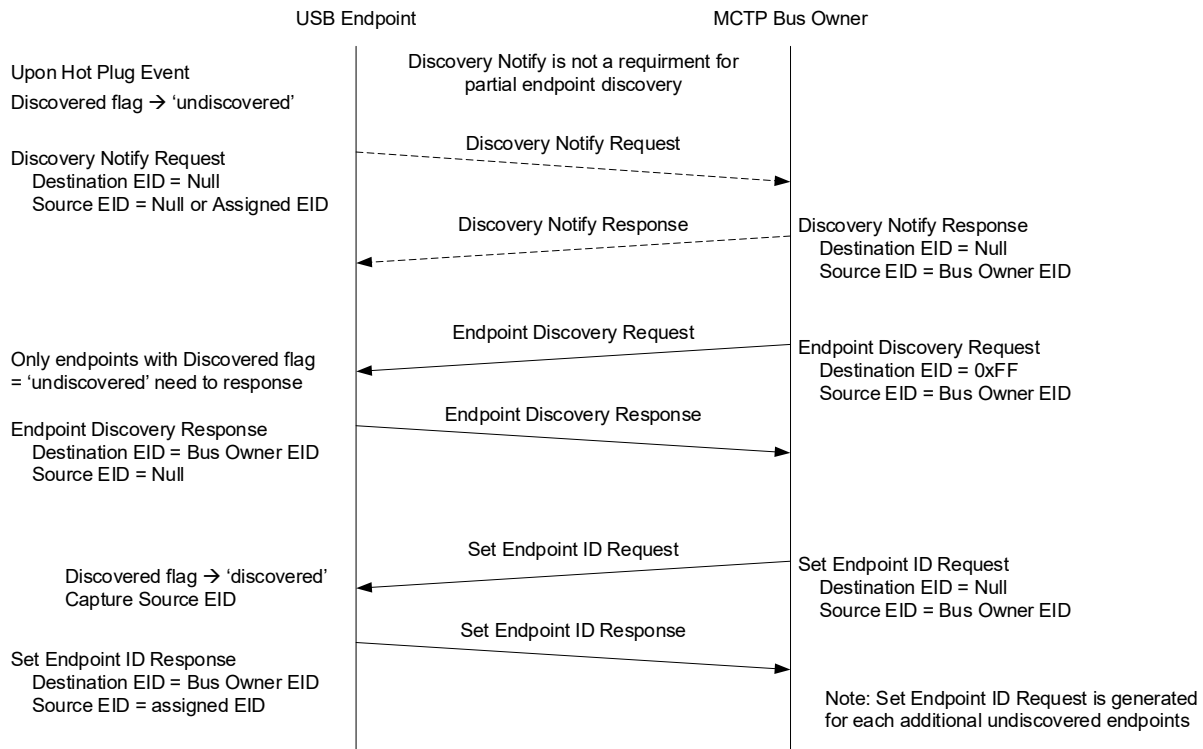
340 The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a  
 341 Discovery Notify message to the MCTP bus owner. The MCTP bus owner may also elect to periodically  
 342 issue a broadcast Endpoint Discovery message to test for whether any undiscovered endpoints have  
 343 been missed. The Discovery Notify message provides the MCTP bus owner with the address/endpoint of  
 344 the MCTP USB endpoint. The MCTP bus owner can then send a directed Endpoint Discovery message a  
 345 to the endpoint to confirm that the device has not been discovered. The MCTP bus owner then issues a

346 Set Endpoint ID command to the physical address for the endpoint which causes the endpoint to set its  
 347 Discovered flag to *discovered*.

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

351 Figure 10 provides an example flow of operations for partial endpoint discovery.

**Partial PCIe MCTP Endpoint Discovery**



352

**Figure 10 – Flow of Operations for Partial Endpoint Discovery**

353

**6.4.3 Endpoint Re-enumeration**

354

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

**6.5 Supported media**

360

361 The transport binding defined in this specification has been designed to work with USB 2.0 compatible  
 362 buses. The USB media type identifiers for this binding spec are defined in [Management Component](#)  
 363 [Transport Protocol \(MCTP\) IDs and Codes](#), in *MCTP physical medium identifiers* section.



364 **6.6 MCTP Messages Routing and USB MCTP bridge**

365 [MCTP](#) packet routing within a [USB](#) bus uses the USB root as an MCTP bridge for routing MCTP packets  
 366 between MCTP endpoints.

367 **6.7 Physical address of MCTP over USB packets**

368 Per [USB](#) specifications, an MCTP over USB endpoint is addressed on the USB fabric using the combined  
 369 7-bits USB Device Address plus 4-bits Endpoint number. The Device Address is configured during the  
 370 interface enumeration process as defined in [USB](#) Bus Enumeration chapter, while the endpoints numbers  
 371 are defined in the endpoints descriptors as described in 6.1.4.2.1 and 6.1.4.2.2.

372 The Device Address and Endpoint number are only used in the Bulk transfer token as shown in Figure 6.  
 373 As the MCTP over USB Header does not include the Device Address and does not include the Endpoint  
 374 number, there is no need for any MCTP endpoint other than the MCTP over USB bus owner to record the  
 375 endpoint address. The bus owner will always add the USB Device Address and Endpoint number of the  
 376 destination endpoint to the USB Bulk packet that is sent to that endpoint.

377 Note: an MCTP over USB endpoint uses 2 Bulk endpoints with the same endpoint number, as described in section  
 378 6.1.4

379 The address format shown in Table 2 is used for MCTP control commands that require a physical address parameter  
 380 to be returned for a bus that uses this transport binding. This includes commands such as the Resolve Endpoint ID,  
 381 Routing Information Update, and Get Routing Table Entries commands.

382 **Table 2 – Physical Address Format**

Format Size	Layout and Description	
2 bytes	Byte 1	[7] – 0 [6:0] – USB Device Address
	Byte 2	[7:4] – 0000 [3:0] – Endpoint Number

383

384 **6.8 Host dependencies**

385 MCTP over USB is not dependent on the operational state of the host system and is operational in all  
 386 power states S5 through S0. The USB bus is only reset on power on reset of the management controller  
 387 or when USB Reset signaling is used as defined in [USB](#).

388 **6.9 Get endpoint ID medium-specific information**

389 The medium-specific information as shown in Table 3 shall be used for the medium-specific Information  
 390 field returned in the response to the Get Endpoint ID MCTP control message.

391 **Table 3 – Medium-specific information**

Description
[7:0] reserved

392

393 **6.10 Composite devices**

394 A composite device which integrates more than a single managed devices entities within the same  
 395 physical device may assign a separate MCTP endpoint to each such managed device entity. In such a  
 396 case, each MCTP endpoint shall use its own MCTP over USB endpoint interface using a shared USB  
 397 connection.

398 **6.11 MCTP over USB packet and control message timing requirements**

399 In USB, all traffic passes through the USB Root.

400 **Table 4 – Timing specifications for MCTP control messages on USB**

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	5 sec	-	Minimum time that a bus owner shall wait before reclaiming the EID for a non-responsive hot-plug endpoint (i.e., not ACKing repeated GETSTATUS CCCs).
Request-to-response time	MT1	-	100 ms	This interval is measured at the responder from the end of the reception of the MCTP Control Protocol request to the beginning of the transmission of the response. 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> + 2 * MT3 max	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.
Transmission Delay	MT3	-	100 ms	Allowed time between the end of the transmission of an MCTP Control Protocol message at the transmitter to the beginning of the reception of the MCTP Control Protocol message at the receiver.
Inter-Packet delay for Multi-Packet messages	MT3a	-	100 ms	Allowed time between the end of the transmission of an MCTP packet with EOM=0 to the beginning of the following MCTP packet of the same Message (see Message assembly in <a href="#">Management Component Transport Protocol (MCTP) Base Specification</a> ), measured at the transmitter

Timing Specification	Symbol	Min	Max	Description
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.
<p>NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.</p> <p>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.</p>				

401

**ANNEX A**  
**(informative)****Change log**

Version	Date	Description
1.0.0	2023-11-02	Initial release.
1.0.1	2024-05-21	Errata fixes Bulk Endpoint parameter • bEndpointAddress bit 7 shall be 1 for IN bulk endpoint per <a href="#">USB 2.0</a> spec. Typo correction in 6.4.1 Corrected document name

402  
403  
404  
405  
406

407