



1

2 **Management Component Transport Protocol**
3 **(MCTP) Universal Serial Bus (USB) Transport**
4 **Binding Specification**

5 **Version: 1.1.0WIP80**

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or its members. Because this document is a Work in Progress, this document may still change, perhaps profoundly and without notice. This document is available for public review and comment until superseded.

Provide any comments through the DMTF Feedback Portal:

<http://www.dmtf.org/standards/feedback>

6 **Document Identifier: DSP0283**

7 **Date: 2026-03-03**

8 **Version History: <https://www.dmtf.org/dsp/DSP0283>**

9 **Supersedes: 1.0.1**

10 **Document Class: Normative**

11 **Document Status: Work in Progress**

12 **Document Language: en-US**

13 Copyright Notice

14 Copyright © 2022–2024, 2026 DMTF. All rights reserved.

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

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29 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

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

36

CONTENTS

37	1	Scope	7
38	2	Normative references	7
39	3	Terms and definitions	7
40	4	Symbols and abbreviated terms.....	9
41	5	Conventions	9
42	6	MCTP over USB Transport Binding	9
43		ANNEX A (informative) Change log.....	27
44			

45 **Figures**

46	Figure 1 – Typical physical topology of USB bus	10
47	Figure 2 – Separated USB host and USB Root Hub devices	11
48	Figure 3 – A USB host as MCTP bus owner and MCTP bridge	11
49	Figure 4 – An MCTP over USB device as an MCTP bridge	12
50	Figure 5 - MCTP 1.x over USB packet format	16
51	Figure 6 – USB Transaction with single MCTP packet payload	17
52	Figure 7 – USB Transaction with 2 MCTP packets payload	17
53	Figure 8 – MCTP packet transfer over USB data Packets with non-equal MCTP packet fragments	18
54	Figure 9 – MCTP packet transfer over USB data Packets with equal MCTP packet fragments	18
55	Figure 10 – Multiple MCTP packets transfer over exactly N USB data packets	18
56	Figure 11 – Multiple MCTP packets transfer over N USB packets and Zero-length USB data packet	18
57		

58 **Tables**

59	Table 1 Interface descriptor parameters without packet spanning	13
60	Table 2 Bulk Out descriptor parameters without packet spanning	14
61	Table 3 Bulk In descriptor parameters without packet spanning	14
62	Table 4 Interface descriptor parameters with packet spanning	15
63	Table 5 Bulk Out descriptor parameters with packet spanning	15
64	Table 6 Bulk In descriptor parameters with packet spanning	16
65	Table 7 - MCTP over USB Header Fields	16
66	Table 8 – Physical Address Format	23
67	Table 9 – Medium-specific information	24
68	Table 10 – Timing specifications for MCTP control messages on USB	25
69		

70

Foreword

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

73 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
74 management and interoperability. For information about the DMTF, see dmf.org.

75 USB Implementers Forum, Inc. is a non-profit corporation founded by the group of companies that
76 developed the Universal Serial Bus specification. The USB-IF was formed to provide a support
77 organization and forum for the advancement and adoption of Universal Serial Bus technology. For
78 information about the USB organization see usb.org.

79 Acknowledgments

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

81 Editor:

- 82 • Yuval Itkin – NVIDIA Corporation

83 Contributors:

- 84 • Patrick Caporale – Lenovo
- 85 • Samer El-Haj-Mahmoud – ARM Inc.
- 86 • Michael Garner – Meta
- 87 • Jeff Hilland – Hewlett Packard Enterprise
- 88 • Abdul R. Ismail – Intel Corporation
- 89 • Janusz Jurski – Intel Corporation
- 90 • Eliel Louzoun – Intel Corporation
- 91 • Chandra Nelogal – Dell Technologies
- 92 • Edward Newman – Hewlett Packard Enterprise
- 93 • Mariusz Oriol – NVIDIA Corporation
- 94 • Paul Sack – Marvell Ltd
- 95 • Patrick Schoeller – Intel Corporation
- 96 • Hemal Shah – Broadcom Inc.
- 97 • Bob Stevens – Dell Technologies
- 98 • Pierre-Philippe Stevens – Advanced Micro Devices
- 99 • Kasper Wszolek – Intel Corporation

100

101

Introduction

102 The Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) transport binding
103 defines a transport binding for facilitating MCTP communication between platform management system
104 components (e.g. management controllers, management devices) over USB 1.1 or later USB versions.

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

109 This document specifies the USB Base Class 14h (MCTP over USB Protocol Endpoint Device Class)
110 normative usage in USB endpoint descriptors supporting this class.

111 Management Component Transport Protocol (MCTP) 112 Universal Serial Bus (USB) Transport Binding Specification

113 1 Scope

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

116 2 Normative references

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

121 DMTF DSP0222, Network Controller Sideband Interface (NC-SI) Specification 1.1

122 https://www.dmtf.org/sites/default/files/standards/documents/DSP0222_1.1.pdf

123 DMTF DSP0236, Management Component Transport Protocol (MCTP) Base Specification 1.3

124 https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.3.pdf

125 DMTF DSP0239, Management Component Transport Protocol (MCTP) IDs and Codes 1.8

126 https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.8.pdf

127 DMTF DSP0256, Management Component Transport Protocol (MCTP) Host Interface Specification 1.0

128 https://www.dmtf.org/sites/default/files/standards/documents/DSP0256_1.0.pdf

129 *DMTF DSP1001*, Management Profile Specification Usage Guide 1.1

130 https://www.dmtf.org/sites/default/files/standards/documents/DSP1001_1.1.pdf

131 ISO/IEC Directives, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents

132 <https://www.iso.org/sites/directives/current/part2/index.xhtml>

133 USB Implementation Forum Inc. Universal Serial Bus Specification version 2.0

134 <https://www.usb.org/document-library/usb-20-specification>

135 USB Implementation Forum Inc. Universal Serial Bus Specification version 3.2 Revision 1.1 (June 2022)

136 <https://www.usb.org/document-library/usb-32-revision-11-june-2022>

137 3 Terms and definitions

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

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

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

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

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

153 3.1

154 MCTP Endpoint Halt

155 a state when an MCTP endpoint becomes non-responsive for any reason

156 3.2

157 MCTP over USB Interface

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

159 3.3

160 MCTP Host-Interface endpoint

161 an MCTP endpoint that serves as [Host interface](#)

162 3.4

163 MCTP packet spanning

164 an MCTP packet transfer over USB where the MCTP packet is transferred using one or more USB
165 packets

166 3.5

167 USB Function

168 a function as defined in [USB 3.2](#)

169 3.6

170 USB Interface

171 an interface as defined in [USB 2.0](#)

172 3.7

173 USB Root Hub

174 USB Root Hub as defined in [USB 2.0](#)

175 3.8

176 USB Transaction

177 Transaction as defined in [USB 2.0](#)

178 3.9

179 USB Transfer

180 Transfer as defined in [USB 2.0](#)

181

182 4 Symbols and abbreviated terms

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

185 4.1

186 USB

187 Universal Serial Bus

188

189 5 Conventions

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

191 5.1 Reserved and Unassigned Values

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

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

196 5.2 Byte Ordering

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

199 6 MCTP over USB Transport Binding

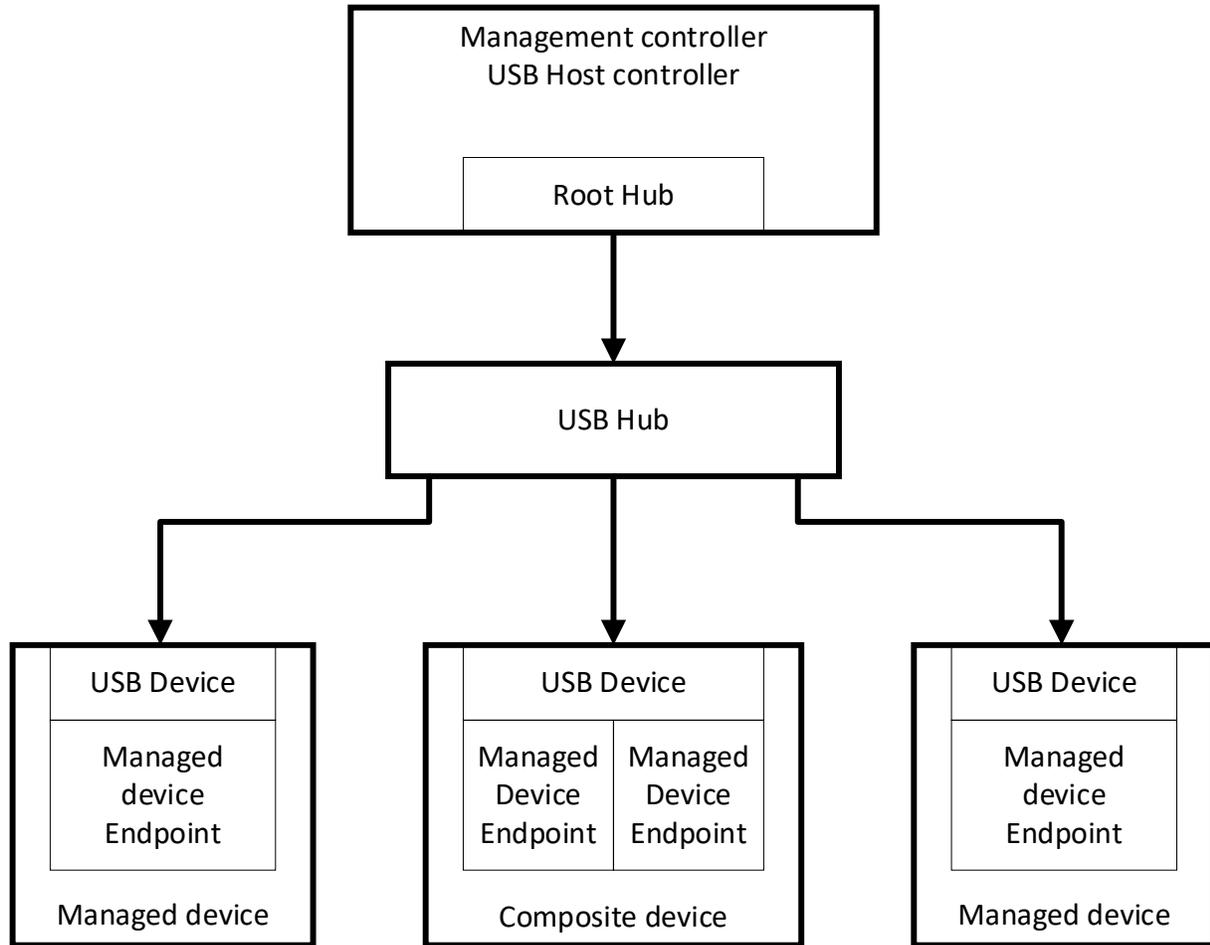
200 This document defines the medium-specific transport binding for transferring MCTP packets between
201 MCTP endpoints on USB using USB Bulk transactions.

202 A MCTP over USB compliant USB device shall support MCTP over USB communications on at least one
203 USB interface of the device. If an MCTP over USB compliant USB device supports MCTP over USB
204 communications on more than one USB interface, then MCTP over USB communication on each USB
205 interface shall be independent from MCTP over USB communications on other USB interfaces.

206 6.1 MCTP use with USB

207 6.1.1 USB bus physical topology

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



212

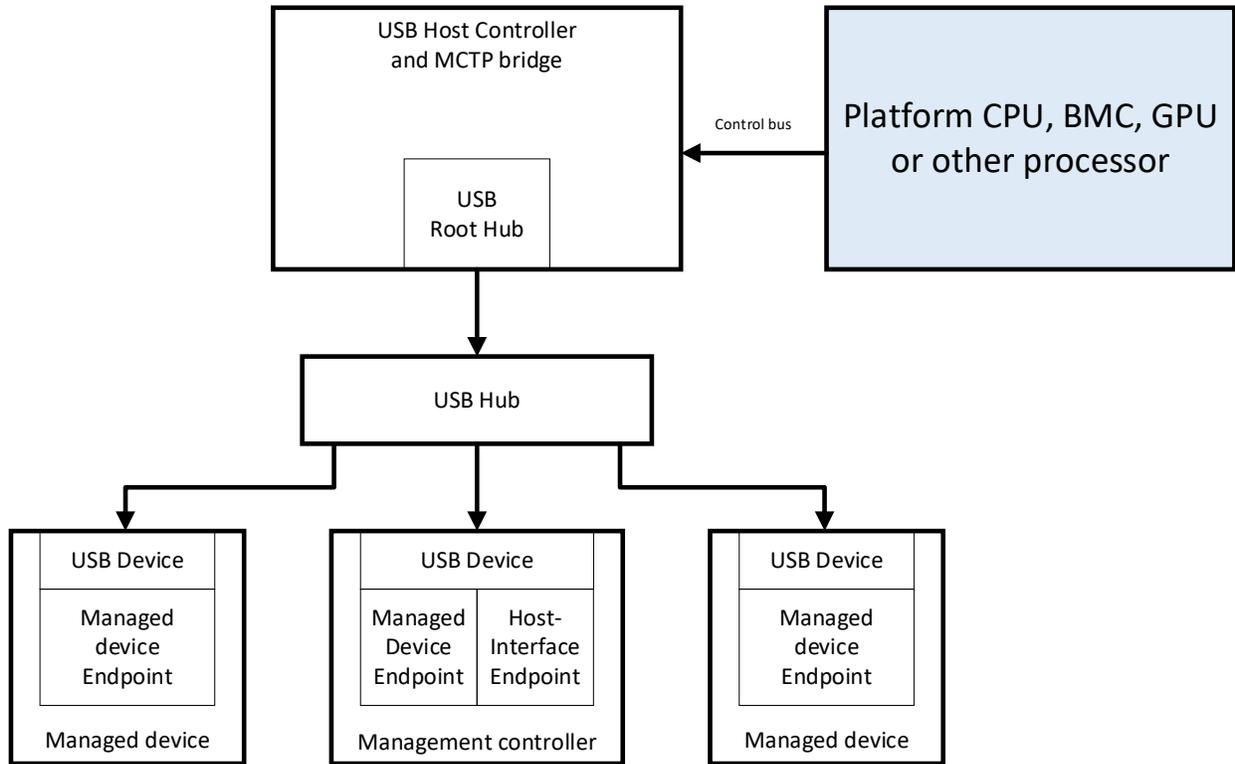
213

Figure 1 – Typical physical topology of USB bus

214 **6.1.2 MCTP bus owner using USB bus**

215 The USB Host is an MCTP bus owner. It is responsible for the discovery and managing the EID
 216 assignments for the MCTP endpoints on the USB bus. The discovery of the MCTP endpoints is done
 217 using the provided USB descriptors of each device that has MCTP interface(s) as part of the device
 218 discovery on the USB bus, as detailed in 6.2.

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



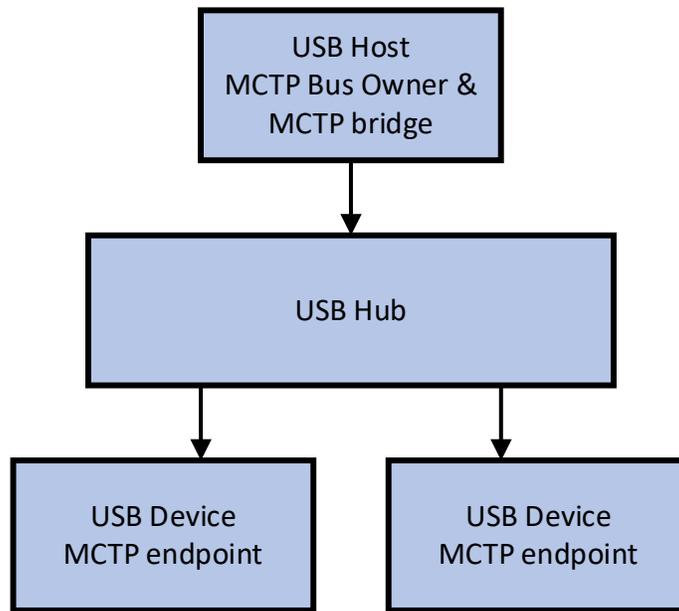
221

222

Figure 2 – Separated USB host and USB Root Hub devices

223 **6.1.3 MCTP bridging over USB**

224 The USB host may act as an MCTP bridge. As USB protocol does not allow direct peer-to-peer
 225 communication between USB endpoints over a USB bus, the USB host will typically serve as an MCTP
 226 bridge for all the MCTP endpoints connected to the same USB host as shown in Figure 3.

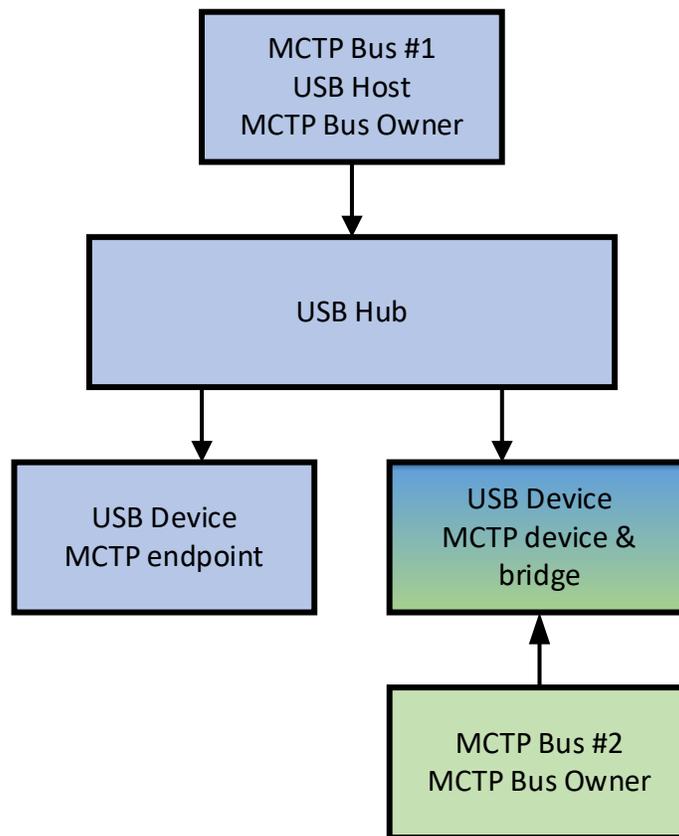


227

228

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

229 A USB MCTP endpoint can also serve as an MCTP bridge to another MCTP bus as shown in Figure 4.
 230 Figure 4 illustrates two MCTP buses that use a USB physical bus for MCTP bus #1 and another physical
 231 bus for MCTP bus #2.



232

233

Figure 4 – An MCTP over USB device as an MCTP bridge

234 6.2 Descriptors structure for MCTP endpoint for MCTP over USB

235 The following sections describe the USB descriptors which are used for MCTP endpoints with and without
 236 support for MCTP packet spanning.

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

- 238 • Bulk OUT endpoint – used to send data from the USB host to the USB device
- 239 • Bulk IN endpoint – used to send data from the USB device to the USB host

240 The set of these 2 USB endpoints is defined as a single USB MCTP interface which is declared by the
 241 following USB descriptors. A device may have at most one MCTP endpoint on a USB interface..

242 6.2.1 USB interface without MCTP packet spanning Descriptors structure

243 MCTP over USB without MCTP packet spanning shall operate only in USB high-speed mode and the
 244 endpoint maximum bulk data payload buffer size shall be set to 512 Bytes.

245 6.2.1.1 Interface descriptor of MCTP Interface without MCTP packet spanning

246 For every MCTP endpoint there is a single interface descriptor as shown below.

247

Table 1 Interface descriptor parameters without packet spanning

Descriptor Field	Value description
Class code	0x14 defines an MCTP endpoint class
Sub-Class code	0x0 - defines a Management-controller and Managed-Device endpoints without MCTP packet spanning support
	0x1 defines an MCTP Host-Interface endpoint without MCTP packet spanning support
	Other - reserved
Number of endpoints	0x2 - Number of endpoints on the USB MCTP endpoint interface
Protocol	0x1 - MCTP 1.x protocol
	0x2 – Reserved for MCTP 2.x and earlier protocol
	Other – reserved
Alternate settings	0 - Shall be set to 0 if no alternate interface
	1 - Used for an MCTP over USB interface that has an alternate settings supporting other MCTP over USB settings

248 A USB Endpoint Descriptor is required for every USB Bulk endpoint. Given that there are 2 USB Bulk
 249 endpoints for every MCTP interface there are 2 Bulk endpoint descriptors.

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

251 **6.2.1.2 Bulk OUT endpoint descriptor of MCTP Interface without MCTP packet spanning**

252 This descriptor declares the USB Bulk endpoint that is used to send data from the USB host to the USB
 253 Device. The following attributes shall be defined in this descriptor:

254

Table 2 Bulk Out descriptor parameters without packet spanning

Descriptor Field	Value description
bEndpointAddress	[7:4] – 0000 [3:0] - Bulk_Endpoint_Number_In_USB_Device
bmAttributes	0x2 - Bulk endpoint
wMaxPacketSize	512 – Declaring 512 Bytes maximum bulk data payload sizes
bInterval	Refer to the USB 2.0 section 9.6.6 to set bInterval and wMaxPacketSize[12..11] to the right value so that it fits the intended use cases supported by the USB device. For Full-Speed devices, declaring that the host shall not try to access the endpoint again during the same frame after receiving a NAK response Implementation note: While USB specification defines bInterval as a method for setting the maximal NAK rate, there are implementations which may not lower the polling rate based on this parameter

255

6.2.1.3 Bulk IN endpoint descriptor of MCTP Interface without MCTP packet spanning

256

This descriptor declares the USB Bulk endpoint that is used to send data from the USB Device to the USB host. The following attributes shall be defined in this descriptor.

257

258

Table 3 Bulk In descriptor parameters without packet spanning

Descriptor Field	Value description
bEndpointAddress	[7:4] – 1000 [3:0] - Bulk_Endpoint_Number_In_USB_Device
bmAttributes	0x2 - Bulk endpoint
wMaxPacketSize	512 – Declaring 512 Bytes maximum bulk data payload sizes
bInterval	Refer to the USB 2.0 section 9.6.6 to set bInterval and wMaxPacketSize[12..11] to the right value so that it fits the intended use cases supported by the USB device. For Full-Speed devices, declaring that the host shall not try to access the endpoint again during the same frame after receiving a NAK response Implementation note: While USB specification defines bInterval as a method for setting the maximal NAK rate, there are implementations which may not lower the polling rate based on this parameter

259

6.2.2 MCTP interface with MCTP packet spanning Descriptors structure

260

MCTP over USB with MCTP packet spanning is allowed for any USB Bulk endpoints regardless of the size of the MCTP packet and wMaxPacketSize.

261

262

6.2.2.1 Interface descriptor of MCTP Interface with MCTP packet spanning

263

For every MCTP endpoint there is a single USB interface descriptor as shown below.

264

Table 4 Interface descriptor parameters with packet spanning

Descriptor Field	Value description
Class code	0x14 defines an MCTP endpoint class
Sub-Class code	0x2 - Management-controller and Managed-Device endpoints supporting MCTP packet spanning
	0x3 MCTP Host-Interface endpoint supporting MCTP packet spanning
	Other
Number of endpoints	0x3 - Number of endpoints on the USB MCTP endpoint interface
Protocol	0x1 - MCTP 1.x protocol
	0x2 – Reserved for MCTP 2.x and earlier protocol
	Other - reserved
Alternate settings	0 - Shall be set to 0 if no alternate interface
	1 - Used for an MCTP over USB interface that has an alternate interface settings

265 **6.2.2.2 Bulk OUT endpoint descriptor**

266 This descriptor declares the USB Bulk endpoint that is used to send data from the USB host to the USB
 267 Device. The following attributes shall be defined in this descriptor:

268 **Table 5 Bulk Out descriptor parameters with packet spanning**

Descriptor Field	Value description
bEndpointAddress	[7:4] – 0000 [3:0] - Bulk_Endpoint_Number_In_USB_Device
bmAttributes	0x2 - Bulk endpoint
wMaxPacketSize	N - N Bytes maximum bulk data payload sizes for the speed at which the device is connected
bInterval	Refer to the USB 2.0 section 9.6.6 to set bInterval and wMaxPacketSize[12..11] to the right value so that it fits the intended use cases supported by the USB device. For Full-Speed devices, declaring that the host shall not try to access the endpoint again during the same frame after receiving a NAK response Implementation note: While USB specification defines bInterval as a method for setting the maximal NAK rate, there are implementations which may not lower the polling rate based on this parameter

269 **6.2.2.3 Bulk IN endpoint descriptor**

270 This descriptor declares the USB Bulk endpoint that is used to send data from the USB Device to the
 271 USB host. The following attributes shall be defined in this descriptor.

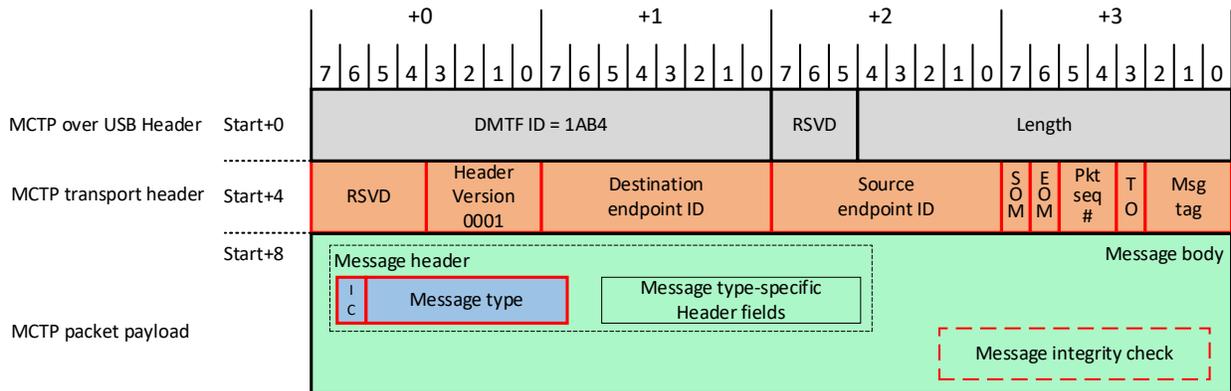
272

Table 6 Bulk In descriptor parameters with packet spanning

Descriptor Field	Value description
bEndpointAddress	[7:4] – 1000 [3:0] - Bulk_Endpoint_Number_In_USB_Device
bmAttributes	0x2 - Bulk endpoint
wMaxPacketSize	N - N Bytes maximum bulk data payload sizes for the speed at which the device is connected
bInterval	Refer to the USB 2.0 section 9.6.6 to set bInterval and wMaxPacketSize[12..11] to the right value so that it fits the intended use cases supported by the USB device. For Full-Speed devices, declaring that the host shall not try to access the endpoint again during the same frame after receiving a NAK response Implementation note: While USB specification defines bInterval as a method for setting the maximal NAK rate, there are implementations which may not lower the polling rate based on this parameter

273 **6.3 Packet Format**

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



276

277 **Figure 5 - MCTP 1.x over USB packet format**

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

282 Table 7 lists the MCTP over USB Header fields and values.

283

Table 7 - 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 (3 bits). Shall always be set to 0 when generating a packet. Shall be ignored on receive.

Byte offset	Field	Description
	Length	Length (13 bits): 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 DSP0236 .

284 The fields labeled “MCTP transport header” and “MCTP packet payload” are common fields for all MCTP
 285 packets and messages and are defined in [MCTP](#).

286

287 **6.4 MCTP transfer over USB**

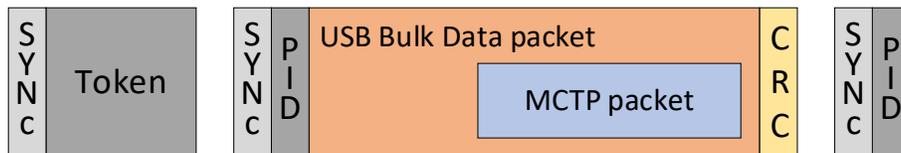
288 MCTP over USB has 2 transfer modes:

- 289 • MCTP over USB without MCP packet spanning – this mode of operation is limited to USB High-
 290 speed only,
- 291 • MCTP over USB with MCTP packet spanning – this mode of operation is independent of the USB
 292 speed.

293 **6.4.1 MCTP over USB without MCTP packet spanning**

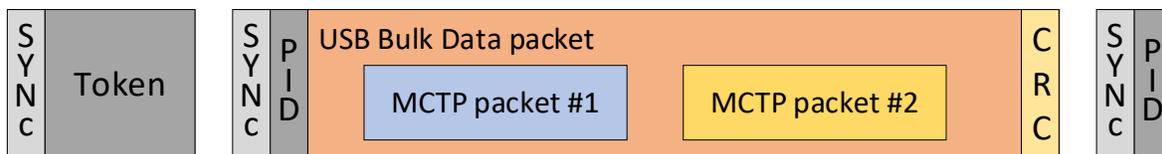
294 This mode can only be used in USB High-Speed mode or later. In this mode, the [MCTP](#) packet cannot be
 295 larger than the endpoint maximum bulk data payload sizes. In this mode, the payload of the [USB](#) packet
 296 contains any combination of one or more MCTP packets destined to or through the same Endpoint-ID
 297 (EID). This mode is the only supported mode for revision 1.0.x of this specification.

298 Refer to Figure 6 – USB Transaction with single MCTP packet payload, Figure 7 – USB Transaction with
 299 2 MCTP packets payload.



300

301 **Figure 6 – USB Transaction with single MCTP packet payload**



302

303 **Figure 7 – USB Transaction with 2 MCTP packets payload**

304 **6.4.2 MCTP over USB with MCTP packet spanning**

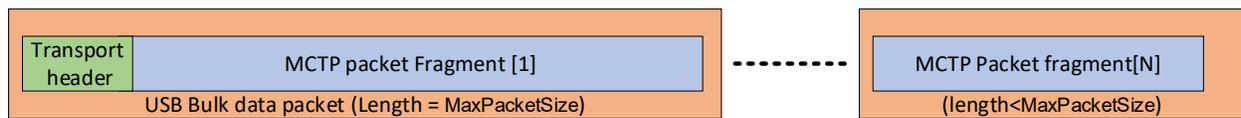
305 In this transfer mode, MCTP over USB packets are transmitted using USB bulk transfers. A single bulk
 306 transfer may be used to transmit multiple MCTP over USB packets by concatenating them together into a
 307 chain as shown in Figure 10 below.

308 When using MCTP over USB with MCTP packet spanning mode, the [MCTP](#) packet can be larger than the
 309 endpoint MaxPacketSize and may be spanning through more than one USB transaction. Using MCTP

310 over USB with MCTP packet spanning mode is allowed on all USB speeds that supports Bulk transfer
 311 type. In this mode, the payload of the [USB](#) packet contains any combination of one or more MCTP
 312 packets or packet fragments that are destined to or through the same Endpoint-ID (EID).

313 When a single [MCTP](#) packet spans over N [USB](#) transactions (N>1), the first USB Bulk transaction
 314 through USB Bulk transaction N-1 shall use data packets size equal to MaxPacketSize of the USB
 315 endpoint. If the Nth USB data packet is less than MaxPacketSize the USB transaction of this short packet
 316 will identify the end of the [MCTP](#) packet. If the data payload within the Nth USB data packet is exactly the
 317 MaxPacketSize, it shall be followed by a zero-length USB packet transaction (which is using a short USB
 318 Bulk transaction) assuring the end of the [MCTP](#) packet is properly identified. See Figure 8 – MCTP
 319 packet transfer over USB data Packets with non-equal MCTP packet , and Figure 9 – MCTP packet
 320 transfer over USB data Packets with equal MCTP packet fragments.

321



322

323 **Figure 8 – MCTP packet transfer over USB data Packets with non-equal MCTP packet fragments**



324

325 **Figure 9 – MCTP packet transfer over USB data Packets with equal MCTP packet fragments**

326 When multiple [MCTP](#) packet span over N [USB](#) transactions, the first USB packet through USB packet N-
 327 1 shall be of MaxPacketSize size. If the Nth USB data packet is less than MaxPacketSize the USB
 328 transfer of this short packet will identify the end of the USB transfer. If the Nth USB data packet is exactly
 329 MaxPacketSize, it shall be followed by a zero-length USB packet (which is a short packet) to assure the
 330 end of [MCTP](#) transfer is properly identified. See Figure 10 – Multiple MCTP packets transfer over exactly
 331 N USB data packets, and Figure 11 – Multiple MCTP packets transfer over N USB packets and Zero-
 332 length USB data packet



333

334 **Figure 10 – Multiple MCTP packets transfer over exactly N USB data packets**



335

336 **Figure 11 – Multiple MCTP packets transfer over N USB packets and Zero-length USB data packet**

337 **6.5 Error handling**

338 [USB](#) Bulk data transfer reliability is ensured in hardware level using error detection and by invoking a
 339 limited number of retries as defined in [USB Spec](#). If the retry count is exceeded, the interface should be
 340 reset using a method that is out of scope for this specification.

341 6.6 MCTP support and capabilities discovery

342 An MCTP-capable MCTP over USB bus-owner, typically discovers all the MCTP capable interfaces on
343 the USB fabric as described below.

344 6.6.1 Full Endpoint Discovery/Enumeration

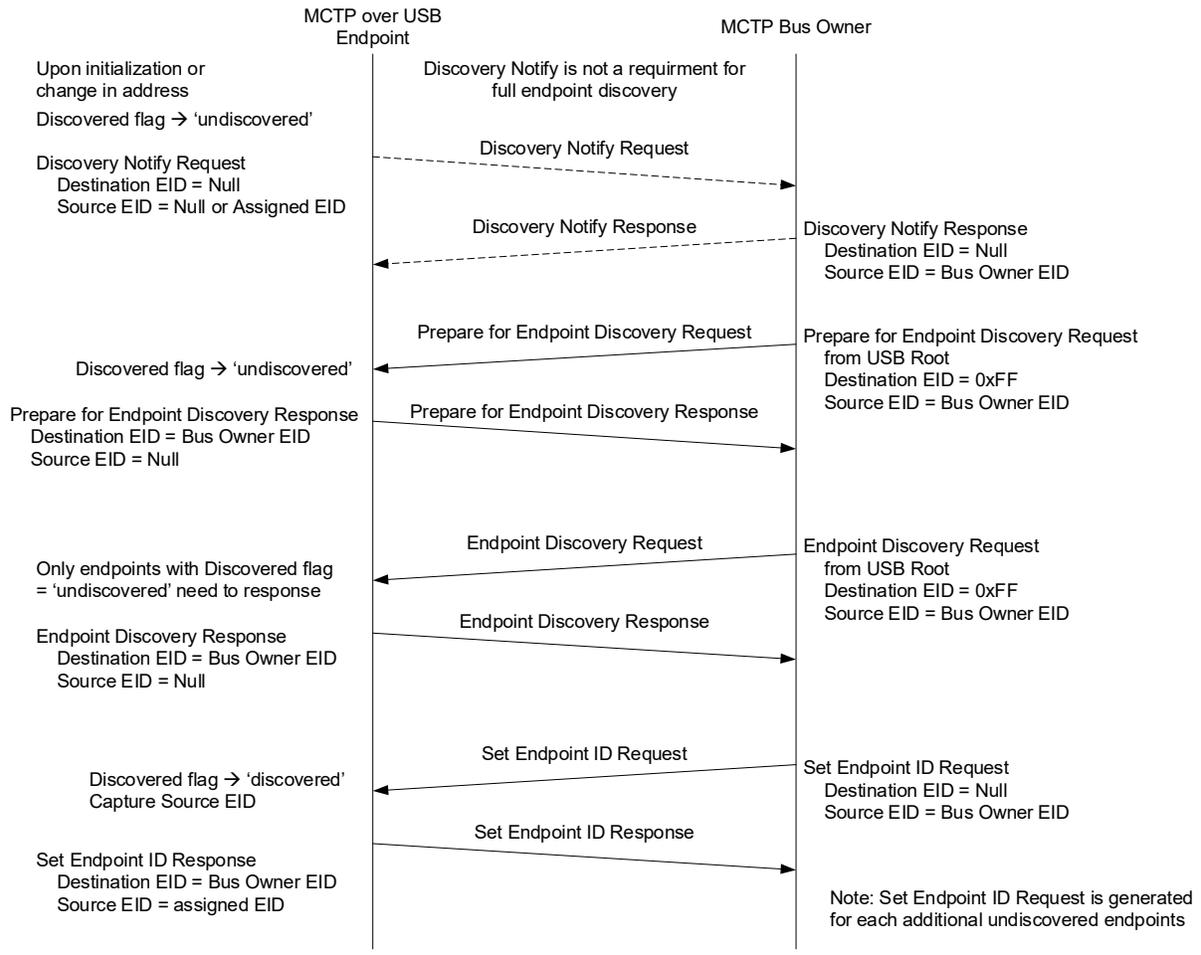
345 The following process is typically used when the MCTP bus owner wishes to discover and enumerate all
346 MCTP endpoints on the USB bus.

- 347 1) MCTP-capable devices are identified by their USB descriptors as defined in section 6.2. During
348 USB detection and enumeration phase.
- 349 2) Following its USB enumeration, an MCTP-capable device should send the *Discovery Notify*
350 MCTP message, to request EID assignment. A USB interface of a [composite device](#) with more
351 than one MCTP endpoint should send the *Discovery Notify* MCTP message for every MCTP
352 endpoint separately.
- 353 3) The MCTP bus owner issues a Prepare for Endpoint Discovery message for every MCTP-
354 capable device using the Broadcast EID as the destination EID. When addressing a [composite](#)
355 [device](#) with more than one MCTP endpoint, the MCTP bus owner shall issue a Prepare for
356 Endpoint Discovery message for every MCTP-endpoint on that MCTP-capable device using the
357 Broadcast EID as the destination EID.
358 This message causes each discoverable endpoint on the bus to set its USB endpoint
359 Discovered flag to undiscovered.
- 360 4) All MCTP-capable devices that have their Discovered flag set to undiscovered will respond with
361 an Endpoint Discovery response message.
- 362 5) The MCTP bus owner should wait for at least MT2 time interval to receive the response. This
363 helps ensure, that all endpoints that received the Prepare for Endpoint Discovery request have
364 processed the request.
- 365 6) The MCTP bus owner issues an Endpoint Discovery request message for every MCTP endpoint
366 on an MCTP-capable device using the Broadcast EID as the destination EID. When addressing
367 a [composite device](#) with more than one MCTP endpoint, the MCTP bus owner shall issue an
368 Endpoint Discovery message for every MCTP-capable interface using the Broadcast EID as the
369 destination EID.
- 370 7) For each response message received from an undiscovered MCTP interface of an MCTP-
371 capable USB device, the MCTP bus owner issues a Set Endpoint ID command to the physical
372 address for the endpoint. This causes the endpoint to set its Discovered flag to *discovered*.
373 From this point, the endpoint shall not respond to the Endpoint Discovery command until
374 another Prepare for Endpoint Discovery command is received, or some other condition causes
375 the Discovered flag to be set back to *undiscovered*.
- 376 8) If the MCTP bus owner received any responses to the Endpoint Discovery request issued in
377 Step 6, then it shall repeat steps 6 and 7 until it no longer gets any responses to the Endpoint
378 Discovery request. In this case, then the MCTP bus owner is allowed to send the next Endpoint
379 Discovery request without waiting for MT2 time interval. If no responses were received by the
380 MCTP bus owner to the Endpoint Discovery request within the MT2 time interval, then the
381 discovery process is completed.

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

385 Figure 12 provides an example flow of operations for full endpoint discovery.

Full USB MCTP Endpoint Discovery



386

387

Figure 12 – Flow of Operations for Full MCTP Discovery over USB

388 6.6.2 Partial Endpoint Discovery/Enumeration

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

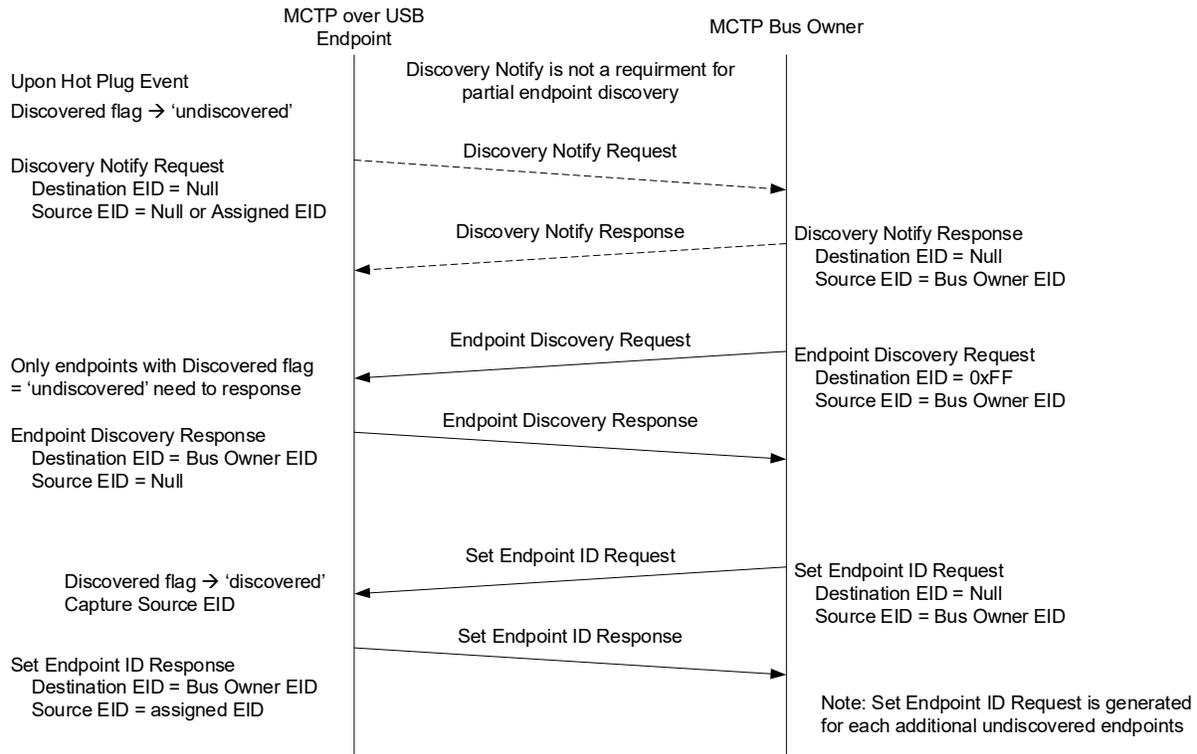
394 The partial discovery process is the same as the full discovery process except that the MCTP bus owner
 395 skips the step of broadcasting a Prepare for Endpoint Discovery command in order to avoid clearing the
 396 Discovered flags of already discovered endpoints.

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

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

408 Figure 13 provides an example flow of operations for partial MCTP endpoint discovery.

Partial USB MCTP Endpoint Discovery



409

Figure 13 – Flow of Operations for Partial MCTP endpoint Discovery

410

6.6.3 MCTP endpoint Re-enumeration

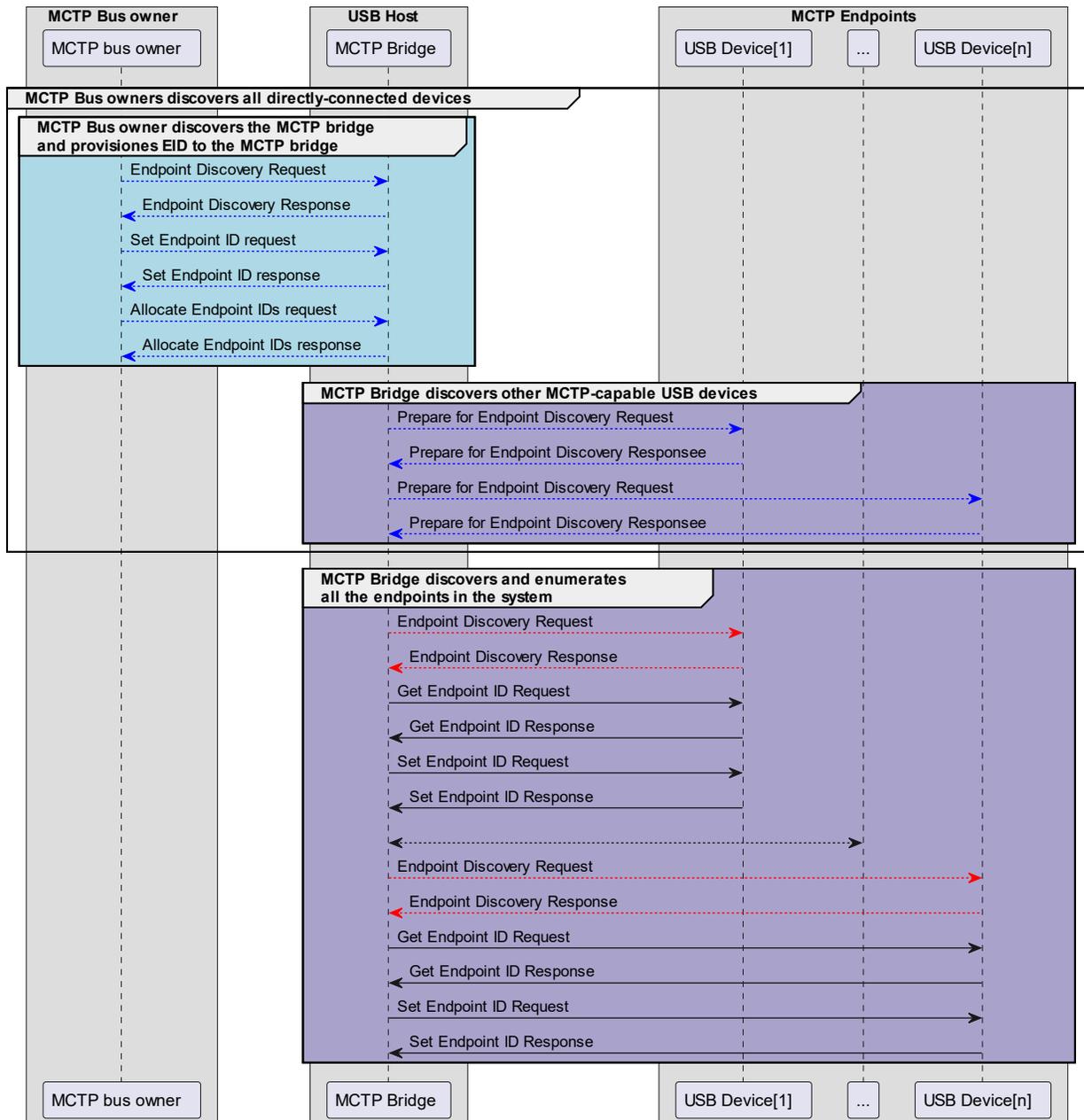
412 If the bus implementation includes hot-plug devices, the bus owner shall perform a full or partial MCTP
 413 endpoint discovery any time the MCTP bus owner goes into a temporary state where the MCTP bus
 414 owner can miss receiving a Discovery Notify message (for example, if the bus owner device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

6.6.4 Discovery when the MCTP bus owner is not the USB host

418 When the USB host is an MCTP bridge while one of the USB devices is either the MCTP top-most bus
 419 owner or another bridge to the MCTP top-most bus owner, then upon reset to the MCTP top-most bus
 420 owner, full discovery shall happen. In this case, the USB bus may not reset when the MCTP top-most bus
 421 owner resets.

422 To accommodate this case the sequence as depicted in Figure 14 shall be performed.

MCTP Endpoints Full Discovery flow



Prepare for Endpoint Discovery is only required when re-enumerating the MCTP bus and is only used when Endpoint Discovery Request is not used
 Endpoint Discovery Request during initial discovery is only needed when Prepare for Endpoint Discovery is not used

423

424

Figure 14 – Full discovery when the MCTP top-most bus owner is not the USB Host

425

The sequence as shown above includes quiescing the MCTP fabric before all the MCTP endpoints are discovered and enumerated. This step uses the “Prepare for Endpoint Discovery” command which sets the discovered flag of all the MCTP endpoints into the undiscovered state.

427

428 As the USB host is operating as an MCTP bridge, once the USB host receives the “Prepare for Endpoint
 429 Discovery” command, it shall then send this command to all the MCTP endpoints which are connected
 430 through the USB Host.

431 Once the MCTP top-most bus owner enumerates the USB host MCTP bridge and assigns it pool of EIDs,
 432 the USB host shall enumerate all the MCTP endpoints which are connected through it.

433 **6.7 Supported media**

434 The transport binding defined in this specification has been designed to work with buses compatible with
 435 USB 1.1 or later USB versions. The USB media type identifiers for this binding spec are defined in
 436 [Management Component Transport Protocol \(MCTP\) IDs and Codes](#), in *MCTP physical medium*
 437 *identifiers* section.

438 **6.8 MCTP Messages Routing and USB MCTP bridge**

439 [MCTP](#) packet routing within a [USB](#) bus uses the USB host as an MCTP bridge for routing MCTP packets
 440 between MCTP endpoints.

441 **6.9 Physical address of MCTP over USB packets**

442 Per [USB](#) specifications, an MCTP over USB endpoint is addressed on the USB fabric using the combined
 443 7-bits USB Device Address plus 4-bits Endpoint number. The Device Address is configured during the
 444 interface enumeration process as defined in [USB](#) Bus Enumeration chapter, while the endpoints numbers
 445 are defined in the endpoints descriptors as described in 6.2.1.2 and 6.2.1.3.

446 The Device Address and USB Endpoint number are only used in the Bulk transfer token as shown in
 447 Figure 8. As the MCTP over USB Header does not include the Device Address and does not include the
 448 USB Endpoint number, there is no need for any MCTP endpoint other than the MCTP over USB bus
 449 owner to record the endpoints addresses. The USB Host, which operates as an MCTP bridge shall add
 450 the USB Device Address and USB Endpoint number of the destination MCTP endpoint to the USB Bulk
 451 transaction that is sent to that MCTP interface.

452 Note: an MCTP endpoint over USB interface uses 2 Bulk endpoints with the same USB endpoint number, as
 453 described in section 6.2

454 The address format shown in Table 8 is used for MCTP control commands that require a physical address parameter
 455 to be returned for a bus that uses this transport binding. This includes commands such as the Resolve Endpoint ID,
 456 Routing Information Update, and Get Routing Table Entries commands.

457 **Table 8 – 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] – USB Endpoint Interface index

458

459 **6.10 Host dependencies**

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

463 6.11 Get endpoint ID medium-specific information

464 The medium-specific information as shown in Table 9 shall be used for the medium-specific Information
465 field returned in the response to the Get Endpoint ID MCTP control message.

466 **Table 9 – Medium-specific information**

Description
[7:0] reserved

467

468 6.12 Composite devices

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

473 6.13 Errors resiliency

474 Comprehensive USB-specific error recovery procedures are defined to ensure high resilience in
475 configurations where USB serves as sideband management interface.

476 To allow errors resiliency, supporting devices shall support the [DSP0236](#) defined **Query Supported**
477 **Recovery Actions** and **Recovery Request** commands.

478 The following reset methods support should be implemented as described below.

479 6.13.1 11.2.1 MCTP application-level reset (L1)

480 This reset type is generic and has no special requirements for this specification. Supporting this reset
481 type is optional per device, according to the device-specific capabilities.

482 6.13.2 MCTP Transport reset (L2)

483 This reset type is used to clear [MCTP Endpoint Halt](#) state. In this reset, the MCTP Endpoint employs
484 sufficient methods to ensure subsequent MCTP Transfers are handled as expected without the need for
485 further intervention. This includes, but is not limited to, flushing internal pipelines, resetting counters etc.

486 6.13.3 Physical bus-level reset (L3)

487 This reset type is used to attempt a USB PORT RESET or BUS RESET as defined in [USB 2p0](#). This
488 reset type allows the USB interface of the device to reset. Following this reset the MCTP endpoint shall
489 be re-enumerated.

490 6.13.4 Device manageability subsystem reset (L4)

491 This reset type is generic and has no special requirements for this specification. Supporting this reset
492 type is optional per device, according to the device-specific capabilities.

493 6.14 MCTP over USB transaction and control message timing requirements

494 In USB, all traffic passes through the USB host.

Table 10 – Timing specifications for MCTP control messages on USB

Timing Specification	Symbol	Min	Max	Description
MCTP 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).
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	-	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 ^[1] + 2 * MT3 max	MT4, min ^[1]	This interval at the requester sets the minimum amount of time that a requester should wait before retrying a MCTP control request. This interval is measured at the requester from the end of the successful transmission of the MCTP control request to the beginning of the reception of the corresponding MCTP control response. NOTE: This specification does not preclude an implementation from adjusting the minimum time-out waiting for a response to a smaller number than MT2 based on the measured response times from responders. The mechanism for doing so is outside the scope of this specification.
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 Management Component Transport Protocol (MCTP) Base Specification), measured at the transmitter

Timing Specification	Symbol	Min	Max	Description
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.				

ANNEX A (informative)

Change log

Version	Date	Description
0.1.0	2022-06-06	Initial draft version
0.1.1	2022-06-22	Updates with RFC comments
0.1.2	2022-07-31	Updates following APTS 2022 ballot and discussion
0.1.4	2022-09-25	Rephrased description on MCTP interface in section 6.2.1.1
0.1.5	2022-11-10	Removed support for USB 1.x
0.1.6	2023-02-28	Updates following review comments
0.1.7	2023-05-06	Added Implementation note regarding NAK rate
0.1.8	2023-07-24	Added medium specific header Clarified multiple MCTP packets within a USB BULK packet are sent to the same EID Clarified the use and mapping of USB addressing, including role of the Bus Owner to add the USB address in the USB Bulk packet. Added Host Dependencies section
0.1.9	2023-07-30	Revised Figure 6 and Figure 7 to include the Token and handshake details
0.1.10	2023-08-09	Editorial & grammatical fixes Added physical address description and format in 6.9
0.1.11	2023-08-15	Typo fixes
0.1.12	2023-08-30	Typo fixes Revised section 6.3 to include MCTP bridge endpoint Updated Figure 5 to show byte ordering Added bytes offsets in Table 7
1.0.1	2024-02-27	Errata fixes Bulk Endpoint parameter • bEndpointAddress bit 7 shall be 1 for Bulk IN endpoint per USB 2.0 spec. Typo correction in 6.6.1 Corrected document name

Version	Date	Description
1.1.0	2026-03-03	Added support for Full-Speed mode transfers Added MCTP packet spanning transfer mode Changed the Discovery process description to be a typical example. Revised support to USB 1.1 or later Added section 6.6.4 Added terms Changed definition of physical address to use the interface index Updated discovery process description Updated description for the setting of bInterval Added parameter MN1 Added section 6.13

502