1

Management Component Transport Protocol (MCTP) I3C Transport Binding Specification

38   CONTENTS

69   **Figures**

78 # Tables

87

88   # Foreword

89 The *Management Component Transport Protocol (MCTP) I3C Transport Binding Specification* (DSP0233)
90 was prepared by the DMTF PMCI Working Group in close cooperation with the MIPI Alliance I3C Working
91 Group.

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

94 MIPI Alliance is a collaborative global organization serving industries that develop mobile and mobile-
95 influenced devices. The focus of the organization is to design and promote hardware and software
96 interfaces that simplify the integration of components built into a device, from the antenna and modem to
97 peripherals and the application processor.

98 This version supersedes version 1.0.0. For a list of changes, see the change log in ANNEX B.

99 ## Acknowledgments

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

101 Editors:

102 • Janusz Jurski – Intel Corporation
103 • Myron Loewen – Solidigm
104 • Amit K. Srivastava – Intel Corporation

105 DMTF Contributors:

106 • Patrick Caporale – Lenovo
107 • Yuval Itkin – NVIDIA Corporation
108 • John Leung – Intel Corporation
109 • Eliel Louzoun – Intel Corporation
110 • Balaji Natrajan – Microchip Technology Inc.
111 • Edward Newman – Hewlett Packard Enterprise
112 • Jim Panian – Qualcomm Incorporated
113 • William Scherer III – Hewlett Packard Enterprise
114 • Hemal Shah – Broadcom Inc
115 • Bob Stevens – Dell Inc.
116 • Johan Van De Groenendaal – Intel Corporation

117 MIPI Contributors:

118 • Guruprasad Ardhanari – Intel Corporation
119 • Jerry Backer – Intel Corporation
120 • Rajesh Bhaskar – Intel Corporation
121 • Enrico D. Carrieri – Intel Corporation
122 • Anamitra Chakrabarti – Synopsys, Inc.
123 • Kelly J. Couch – Intel Corporation
124 • Kenneth P. Foust – Intel Corporation
125 • Chris Grigg – MIPI Alliance (Team)
126 • David Harriman – Intel Corporation
127 • Prakash Iyer – Intel Corporation
128 • Paul Kimelman – NXP Semiconductors
129 • Łukasz Łanecki – Intel Corporation

130　　　• 　Richard Marian Thomaiyar – Intel Corporation
131　　　• 　Tim E. McKee – Intel Corporation
132　　　• 　Mariusz Oriol – Intel Corporation
133　　　• 　Ankit Rohatgi – MIPI Alliance (Team)
134　　　• 　Matthew Schnoor – Intel Corporation
135　　　• 　Eyuel Zewdu Teferi – STMicroelectronics
136　　　• 　George Vergis – Intel Corporation

137 # Introduction

138 The Management Component Transport Protocol (MCTP) over I3C transport binding defines a transport
139 binding for facilitating communication between platform management subsystem components (e.g.,
140 management controllers, managed devices) over I3C.

141 The *Management Component Transport Protocol (MCTP) Base Specification* describes the protocol and
142 commands used for communication within and initialization of an MCTP network. The MCTP over I3C
143 transport binding definition in this specification includes a packet format, physical address format,
144 message routing, and discovery mechanisms for MCTP over I3C communications.

1   Scope

This document provides the specifications for the Management Component Transport Protocol (MCTP) transport binding for I3C.

2   Normative references

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

DMTF, DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*, https://www.dmtf.org/dsp/DSP0236

DMTF, DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.8*, https://www.dmtf.org/dsp/DSP0239

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

Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic, Version 1.0, https://www.mipi.org/specifications/i3c-sensor-specification
Note: this specification relies on version 1.0 of the referenced document, but implementers are encouraged to follow the latest published version.

MIPI Mandatory Data Byte (MDB) Values Table, https://www.mipi.org/MIPI_I3C_mandatory_data_byte_values_public

MIPI Device Characteristics Register (DCR) Assignments, https://www.mipi.org/MIPI_I3C_device_characteristics_register

MIPI I3C^{SM} Host Controller Interface^{SM} v1.0 Specification, MIPI, April 2018, https://www.mipi.org/specifications/i3c-hci
Note: this specification relies on version 1.0 of the referenced document but implementers are encouraged to follow the latest published version

System Management Bus (SMBus) Specification v2.0, SBS Implementers Forum, SMBus, August 2000, http://www.smbus.org/specs/smbus20.pdf
Note: this specification relies on version 2.0 of the referenced document, but implementers are encouraged to follow the latest published version.

3   Terms and definitions

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

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

The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 6.

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

189 Refer to *Management Component Transport Protocol (MCTP) Base Specification* for the terms and
190 definitions that are used across the MCTP specifications.

191 For the purposes of this document, the following terms and definitions apply.

192 **3.1**
193 **Address Resolution Protocol**
194 **ARP**

195 refers to the procedure used to dynamically determine the addresses of devices on a shared
196 communication medium

197 **3.2**
198 **ACK**

199 Acknowledge

200 **3.3**
201 **BCR**

202 Bus Characteristics Register – see *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic*
203 for more information

204 **3.4**
205 **BMC**

206 Baseboard management controller

207 **3.5**
208 **CCC**

209 Common Command Code – see *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic*
210 for more information

211 **3.6**
212 **Destination Device**

213 Device receiving the MCTP packet over I3C bus

214 **3.7**
215 **EEPROM**

216 Electrically Erasable Programmable Read-Only Memory

217 **3.8**
218 **EID**

219 Endpoint identifier

220 **3.9**
221 **HCI**

222 Host Controller Interface

223 **3.10**
224 **Hot-Join**

225 Joining the Bus after it is already started – see *Specification for I3C Basic^{SM}, Improved Inter Integrated*
226 *Circuit – Basic* for more information

227 **3.11**
228 **I3C**
229 Improved Inter-Integrated Circuit – see *Specification for I3C Basic^SM, Improved Inter Integrated Circuit –*
230 *Basic* for more information

231 **3.12**
232 **I3C Master / I3C Controller / Primary**
233 The "I3C Master" and "I3C Controller" terms are equivalent – the original revision of *Specification for I3C*
234 *Basic^SM, Improved Inter Integrated Circuit – Basic* defined the "I3C Master" term while the updated MIPI
235 specification revisions used the "I3C Controller" term; the initial revision of this specification sometimes
236 also used the term "Primary" when referring to the "I3C Controller" or "I3C Master"

237 **3.13**
238 **I3C Slave / I3C Target / Secondary**
239 The "I3C Slave" and "I3C Target" terms are equivalent – the original revision of *Specification for I3C*
240 *Basic^SM, Improved Inter Integrated Circuit – Basic* defined the "I3C Slave" term while the updated MIPI
241 specification revisions used the "I3C Target" term; the initial revision of this specification sometimes also
242 used the term "Secondary" when referring to the "I3C Target" or "I3C Slave"

243 **3.14**
244 **IBI**
245 In-Band Interrupt – see *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic* for more
246 information

247 **3.15**
248 **max**
249 maximum

250 **3.16**
251 **MCTP**
252 Management Component Transport Protocol

253 **3.17**
254 **MDB**
255 Mandatory Data Byte

256 **3.18**
257 **MHz**
258 megahertz

259 **3.19**
260 **min**
261 minimum

262 **3.20**
263 **ms**
264 millisecond

265 **3.21**
266 **MSB**
267 most significant byte

268 **3.22**
269 **MTU**
270 Maximum Transmission Unit

271 **3.23**
272 **NACK**
273 not acknowledge

274 **3.24**
275 **PCI**
276 peripheral component interconnect

277 **3.25**
278 **PCIe®**
279 PCI Express™

280 **3.26**
281 **PEC**
282 packet error code

283 **3.27**
284 **PMCI**
285 Platform Management Component Intercommunications

286 **3.28**
287 **SCL**
288 serial clock

289 **3.29**
290 **SDA**
291 serial data

292 **3.30**
293 **sec**
294 second

295 **3.31**
296 **SMBus**
297 System Management Bus

298 **3.32**
299 **Source Device**
300 Device sending the MCTP packet over I3C bus

301 **3.33**
302 **T-bit**
303 Transition bit – see *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* for more
304 information

305 **3.34**
306 **UDID**
307 unique device identifier

4   Conventions

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

4.1   Reserved and unassigned values

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

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

4.2   Byte ordering

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

5   MCTP over I3C transport

The MCTP over I3C transport binding defines how MCTP packets are delivered over a physical I3C medium using I3C transfers. See *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* for complete details about I3C requirements, including the electrical layer. This specification defines additional requirements and supersedes the *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* in any cases when there are differences.

This binding specification has been designed to be able to share the same bus with devices communicating using other I3C protocols (e.g., MIPI Debug for I3C^{SM} – see clause 5.4.1) and compatible SMBus/I2C devices (e.g., EEPROM). Interactions with such devices or protocols are out of scope for this specification.

5.1   MCTP use with I3C

5.1.1   I3C bus physical topology

The physical topology of the I3C bus is presented in Figure 1. There is a single device that plays the role of the I3C Controller (typically it is a Management Controller, Embedded Controller, etc.) and there may be multiple I3C Targets sharing the same I3C bus. *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* defines the Secondary Controller but that is not required for implementing MCTP and is out of scope for this specification.

335
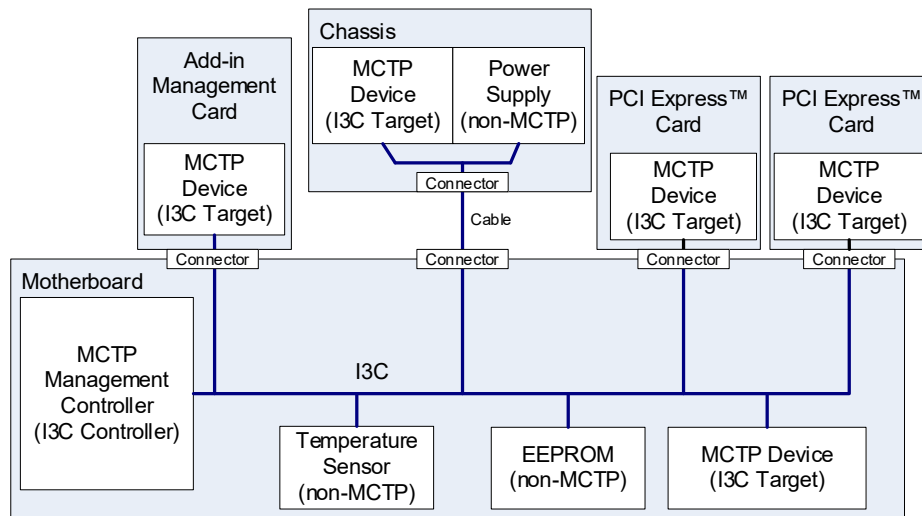
Figure 1 – Physical topology of I3C bus

337

5.1.2   I3C communication logical topology and MCTP packet bridging

339   The topology of the logical communication paths is shown in Figure 2. The I3C Controller can
340   communicate to any of the I3C Targets. Each I3C Target can communicate with the I3C Controller only.
341   Any communications between I3C Targets are implemented by MCTP bridge functionality in the I3C
342   Controller, according to the *Management Component Transport Protocol (MCTP) Base Specification*.
343   Unlike typical MCTP bridges that transfer data to another port, this data may be retransmitted to the same

344 port. When forwarding, the physical addressing and PEC gets changed by the bridge to match the
345 requirements of the destination bus.
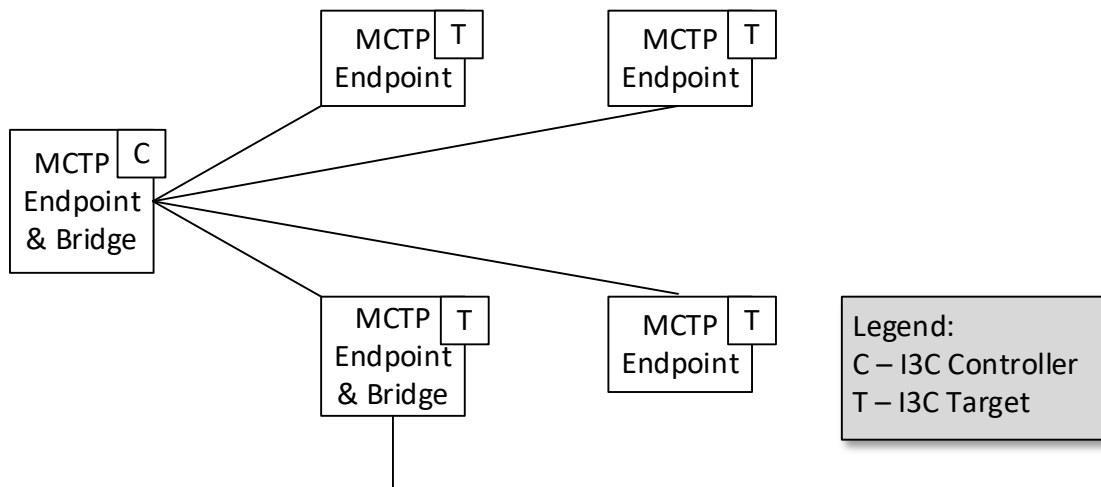
346

347 **Figure 2 – Logical topology of MCTP over I3C communication**

348 Note that the *Specification for I3C Basic[SM], Improved Inter Integrated Circuit – Basic* has its own concept
349 of an I3C bridge device and it requires that I3C bridges implement certain functionality and report their
350 capability using BCR[4] flag. MCTP bridges are a different concept from I3C bridges.

351 There is no relationship between the physical layer I3C addresses and the transport protocol layer MCTP
352 EIDs. I3C addresses are assigned by the I3C Controller, while MCTP EIDs are assigned by the MCTP
353 Bus Owner. These two functions are logically independent but they may be collocated.

354 ### 5.1.3 MCTP Bus Owner for I3C bus[1]

355 As defined in *Management Component Transport Protocol (MCTP) Base Specification*, MCTP Bus Owner
356 device is responsible for MCTP endpoints discovery and managing MCTP EID assignments. EID
357 assignment requires physical addressing to be used (with EID = 0, i.e., Null Destination EID or Null
358 Source EID). On the I3C bus, direct communication can only happen with the I3C Controller either as a
359 source or a destination, as described in the previous clause.

360 There may be multiple logical MCTP buses overlaid on a single I3C physical bus:

361 • Preferably, the I3C Controller is the MCTP Bus Owner. It can discover all the I3C Targets and
362 fulfil the MCTP Bus Owner role for the whole I3C bus (see clause 5.4.1 for the flow details).
363 • Additionally, an I3C Target can be an MCTP Bus Owner but only for the connection between it
364 and the I3C Controller (see clause 5.4.1 for the flow details as well). Other I3C Target devices on
365 the I3C bus are not directly reachable by the I3C Target. I3C Target acting as an MCTP Bus
366 Owner enables it to act as an MCTP bridge from another MCTP bus. An I3C Target that acts as
367 an MCTP Bus Owner cannot be added to an I3C bus using the I3C hot-join mechanism.

---

[1] The term "bus" is used in a different meaning in *Specification for I3C Basic[SM], Improved Inter Integrated Circuit – Basic* and in *Management Component Transport Protocol (MCTP) Base Specification* context. This clause describes a scenario when multiple MCTP buses are overlaid on a single I3C bus.

368   For example, as shown in Figure 3, two logical buses are created and EIDs are assigned as follows:

369   • *Logical MCTP Bus 1*, *Bridge 1* (I3C Target) is the MCTP Bus Owner – *Bridge 1* assigns the EID
370   and EID pool to the I3C Controller because I3C Controller is an endpoint on the *Logical MCTP*
371   *Bus 1*.
372   • *Logical MCTP Bus 2*, *Bridge 2* (I3C Controller) is the MCTP Bus Owner for *Logical MCTP Bus 2* –
373   *Bridge 2* assigns EIDs to all the remaining I3C Targets.

374   This concept can be extended and each device on an I3C bus could be a MCTP Bridge to additional
375   MCTP networks.
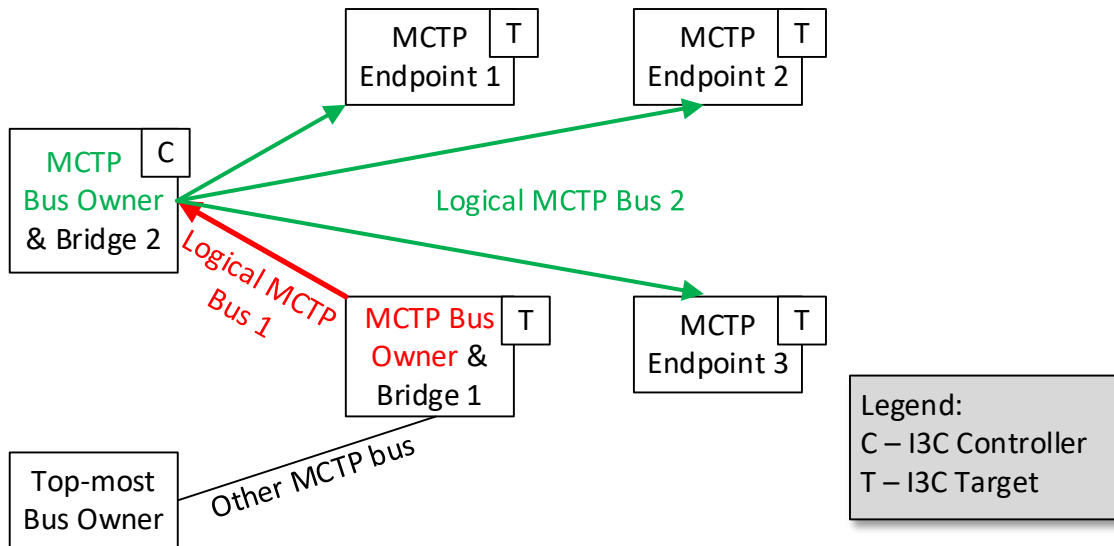
376

377   **Figure 3 – Sample I3C Target as MCTP Bus Owner & bridge**

378

5.2   MCTP packet encapsulation

380   MCTP packet transfers over I3C slightly differ depending on the communication direction:

381   • I3C Controller to I3C Target communication follows encapsulation defined in clause 5.2.1

382   • I3C Target to I3C Controller communication follows encapsulation defined in clause 5.2.2

383   Subclauses below capture the MCTP packet encapsulation details. There is no requirement for the multi-
384   packet MCTP message to be contiguous on the bus.

5.2.1   MCTP packet encapsulation: I3C Controller to I3C Target

5.2.1.1   Overview

387   Transmission of MCTP packets from the I3C Controller to the I3C Target happens using
388   private write transfer initiated by I3C Controller as defined in *Specification for I3C Basic^{SM}, Improved Inter*
389   *Integrated Circuit – Basic*. The transfer shall be directed to the I3C Target address used for MCTP
390   protocol communication. For the purpose of this specification, an I3C Target shall only support the MCTP
391   protocol at its unique I3C Target address and an I3C address shall be dynamically assigned for that

392 purpose. See clause 5.4.1 for discussion about protocol discovery when other protocols may be in use on
393 the I3C bus.

394 The MCTP message header and MCTP message data fields map to I3C payload as indicated in Figure 4.
395 After the MCTP message data, there is a PEC byte added – its role is discussed in clause 5.3.1. Please
396 note that the length of the write transfer is dictated by the I3C Controller using Repeated Start/Stop
397 condition. I3C Controller is expected to obey the discovered maximum write length (see clause 5.4.2 for
398 more information).

399 Note that the I3C Target does not need I3C address of the I3C Controller because all MCTP packets from
400 a given I3C Target will always be directed to the I3C Controller – the I3C Controller has no explicit
401 address as per _Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic_. MCTP Destination
402 EID should be used to route the MCTP packet to another I3C Target if necessary.
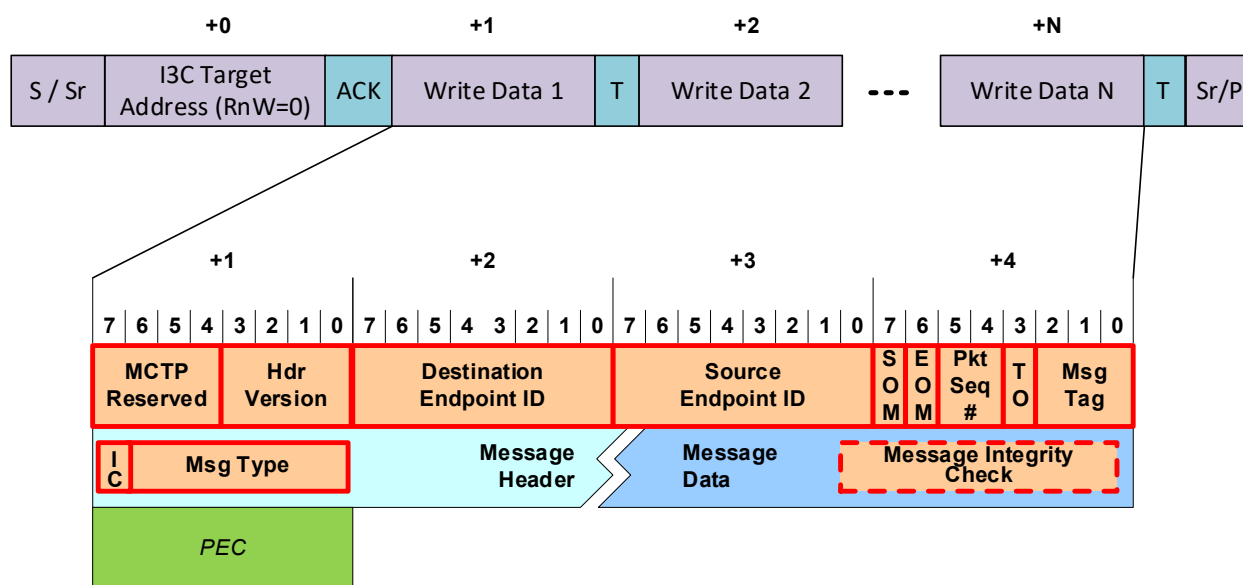
403

404 **Figure 4 – MCTP over I3C packet transfer format: I3C Controller to I3C Target**

405 Please note that the MCTP packet transfer shown may be preceded by the optional I3C Broadcast
406 Address (7'h7E), as defined in the I3C specification. In this transaction, the T-bit is the parity of each byte.

407 As per the _Management Component Transport Protocol (MCTP) Base Specification_, IC and Message
408 Type byte is only present in the first packet of a fragmented MCTP message.

409 **Table 1 – MCTP packet transfer field descriptions**

Byte	I3C Field(s)	Description
0	I3C Target Address RnW	[7:1] I3C Target Address: The address of the I3C Target on the local I3C bus [0] I3C RnW# bit: Shall be set to `0b` as all MCTP messages using I3C write transfers.

Byte	I3C Field(s)	Description
1	Write Data 1	[7:4] MCTP reserved: This nibble is reserved for definition by the _Management Component Transport Protocol (MCTP) Base Specification_. [3:0] MCTP header version: Set to 0001b for MCTP v1 devices that are conformant to the _Management Component Transport Protocol (MCTP) Base Specification_ and this version of the MCTP transport binding. All other values = Reserved.
2	Write Data 2	Destination Endpoint ID (*) as defined in _Management Component Transport Protocol (MCTP) Base Specification_, including special endpoint IDs
3	Write Data 3	Source Endpoint ID (*) as defined in _Management Component Transport Protocol (MCTP) Base Specification_, including special endpoint IDs
4	Write Data 4	[7] SOM: Start Of Message flag (*) [6] EOM: End Of Message flag (*) [5:4] MCTP Packet sequence number (*) [3] Tag Owner (TO) bit (*) [2:0] Message tag (*)
5	Write Data 5	[7] IC: Integrity Check bit (*) [6:0] Message type (*)
6:N-1	Write Data 6:N-1	MCTP message header and data (*)
N	PEC	Packet error code (PEC): All MCTP I3C transfers shall include a PEC byte. The PEC byte shall be transmitted by the source and checked by the destination. Please see clause 5.3.1 for more information.
(*) Indicates a field that is defined by the _Management Component Transport Protocol (MCTP) Base Specification_.		

5.2.1.2   I3C Target address ACKs/NACKs

The I3C Controller can start another write transfer after the Repeated Start condition on the bus, meaning that multiple MCTP packets can follow one after the other in sequence. In case the I3C Target buffer cannot accommodate the maximum packet length (as negotiated according to clause 5.4.2), it shall NACK its address to indicate the potential overflow and a need for retry later. The time to retry is dependent on the implementation – see clause 5.8 for more information.

NACK of an I3C Target address may indicate that the device buffers are full or the physical absence of the device. The I3C Controller may test for the presence of a device after a NACK with the GETSTATUS CCC. The I3C Target shall always respond to GETSTATUS CCC, even if its MCTP data buffer is full. The I3C Controller shall retry GETSTATUS CCC as per _Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic_, clause 5.1.9.2.3 Retry Model for Direct GET CCC Commands, before it considers the device as absent.

5.2.2   MCTP packet encapsulation: I3C Target to I3C Controller

Transmission of MCTP packets from I3C Target to I3C Controller can happen in two modes:

- In-Band Interrupt mode (IBI mode) or
- polling mode (described in clause 5.2.2.4).

426 The I3C Target is required to support both modes of operation and the I3C Controller can enable or
427 disable IBIs as defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic*.

5.2.2.1 Overview – IBI mode

429 Transmission of MCTP packets from I3C Target to I3C Controller according to the IBI mode shall happen
430 using the following general sequence:

431   1. When the I3C Target has a MCTP packet ready for transmission to the I3C Controller, it shall
432   initiate an I3C IBI with MDB = 0xAE (as assigned in *MIPI Mandatory Data Byte (MDB) Values*
433   *Table* registry) to inform the I3C Controller about the data ready.

434   2. The I3C Controller shall read the MCTP packet (or multiple packets) from the I3C Target using
435   *I3C Private Read transfer*.

436 This sequence is illustrated in Figure 5:
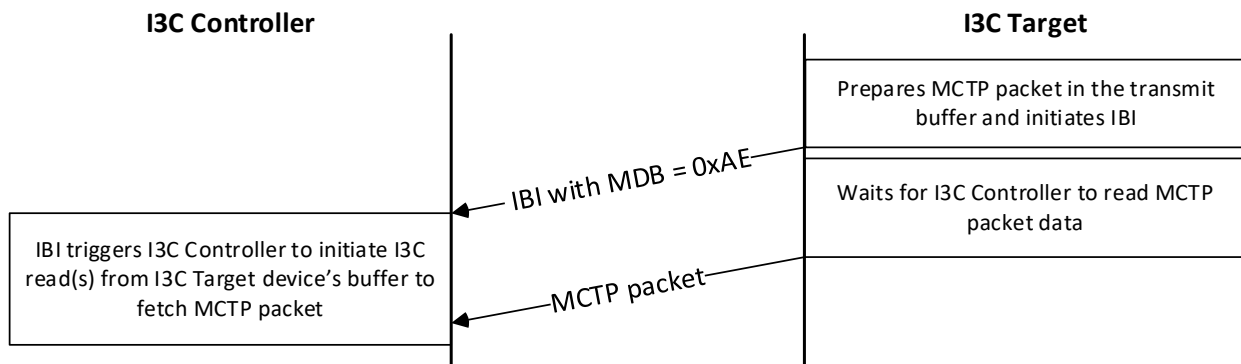
438   **Figure 5 – MCTP over I3C packet transfer sequence: I3C Target to I3C Controller**

439 The transaction field explanations are illustrated in Figure 6 and in Table 2 (for pending read notification)
440 and Table 3 (for the MCTP packet transfer):

441
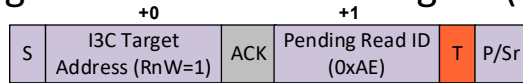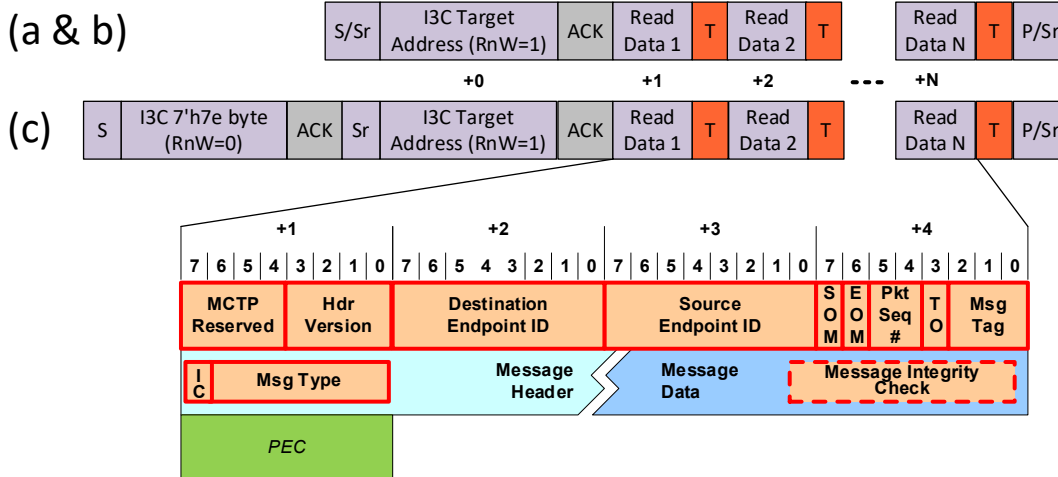
442

443   **Figure 6 – MCTP over I3C packet transfer format: I3C Target to I3C Controller**

444 As defined in the *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic*, the read transfer
445 may start with:

446   (a) Repeated Start condition and the I3C Source (I3C Target) address immediately after the IBI or
447   other traffic – for example, using HCI auto-command as defined in *MIPI I3C^SM Host Controller*
448   *Interface^SM Specification*
449   (b) Start condition and the I3C Target address
450   (c) Start condition with I3C Broadcast Address (7'h7E), then Repeated Start with the I3C Target
451   address

452 In these transactions, the T-bit is zero to indicate the End-of-Data – see *Specification for I3C Basic^SM,*
453 *Improved Inter Integrated Circuit – Basic*, clause *5.1.2.3.4 Ninth Bit of SDR Slave Returned (Read) Data*
454 *as End-of-Data*.

455 As per the *Management Component Transport Protocol (MCTP) Base Specification*, IC and Message
456 Type byte is only present in the first packet of a fragmented MCTP message.

457   **Table 2 – IBI pending read notification field descriptions**

Byte	I3C Field(s)	Description
0	I3C Target Address	[7:1] I3C Target Address: The address of the I3C device that is the source of the MCTP packet
	RnW	[0]: I3C RnW# bit: Shall be set to `1b` for all IBIs

Byte	I3C Field(s)	Description
1	Mandatory Data Byte (MDB)	[7:0] 0xAE value – MCTP Pending Read ID notification as defined in *MIPI Mandatory Data Byte (MDB) Values Table* registry

458

459 **Table 3 – MCTP packet transfer field descriptions**

Byte	I3C Field(s)	Description
0	I3C Target Address RnW	[7:1] I3C Target Address: The address of the I3C device that is the source of the MCTP packet [0]: I3C RnW# bit: Shall be set to `1b` for all read transfers.
1	Read Data 1	[7:4] MCTP reserved: This nibble is reserved for definition by the *Management Component Transport Protocol (MCTP) Base Specification*. [3:0] MCTP header version: Set to 0001b for MCTP v1 devices that are conformant to the *Management Component Transport Protocol (MCTP) Base Specification* and this version of the MCTP transport binding. All other values = Reserved.
2	Read Data 2	Destination Endpoint ID (*) as defined in *Management Component Transport Protocol (MCTP) Base Specification*, including special endpoint IDs
3	Read Data 3	Source Endpoint ID (*) as defined in *Management Component Transport Protocol (MCTP) Base Specification*, including special endpoint IDs
4	Read Data 4	[7] SOM: Start Of Message flag (*) [6] EOM: End Of Message flag (*) [5:4] MCTP Packet sequence number (*) [3] Tag Owner (TO) bit (*) [2:0] Message tag (*)
5	Read Data 5	[7] IC: Integrity Check bit (*) [6:0] Message type (*)
6:N-1	Read Data 6:N-1	MCTP message header and data (*)
N	PEC	Packet error code (PEC): All MCTP I3C transfers shall include a PEC byte. The PEC byte shall be transmitted by the source and checked by the destination. Please see clause 5.3.1 for more information.
(*) Indicates a field that is defined by the *Management Component Transport Protocol (MCTP) Base Specification*.		

460

5.2.2.2 Detailed flow

When a I3C Target has an MCTP packet available to transfer, it initiates the flow by sending an IBI with a Mandatory Data Byte (MDB) value = 0xAE. This is to inform the I3C Controller that an MCTP packet is available for reading from the I3C Target. The I3C Controller should acknowledge the IBI request and read the MDB data from the I3C Target. After accepting the request, the I3C Controller may read the packet immediately with a Repeated Start after the IBI or it may delay the read up to PT timeout – see

467 clause 5.8. The I3C Controller may queue up several IBI notifications from multiple I3C Targets and
468 process them in any order. Delaying reads allows prioritization as well as management of shared buffers
469 but delays may result in IBI retransmissions – see clause 5.2.2.3.

470 When sending the IBI notification, the I3C Target needs to ensure that the MDB has been read by the I3C
471 Controller and ensure the data is available for the next private read request from the I3C Controller. If the
472 I3C Controller NACKs the IBI, then the IBI was not accepted and the I3C Target shall retry the IBI at the
473 next opportunity as per *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic*, *section*
474 *5.1.6.2, Slave Interrupt Request* and conformant to section 5.8. The I3C Target may interpret consecutive
475 NACKs of an IBI as an error and take actions dependent on implementation.

476 Once the I3C Target has sent all the bytes of the MCTP packet and the PEC byte, the I3C Target shall
477 indicate the end of data transfer, and the I3C Controller completes the I3C transaction. The next MCTP
478 packet transfer shall happen in a separate transfer. The error cases for this rule are described in clause
479 5.2.2.7.

5.2.2.3 ACKed IBI retransmissions

480

481 Each I3C Target shall also implement a timeout mechanism in order to retransmit the IBI. The timer shall
482 be started when the I3C Controller acknowledges the IBI with MDB. The I3C Target shall retransmit the
483 IBI if the read has not happened within a certain time. The number of retransmits and the timeout value
484 are implementation dependent – depends on the I3C Target functionality, I3C Controller capabilities, and
485 characteristics of MCTP traffic (urgency of retransmission) and shall conform to clause 5.8 requirements.
486 The I3C Target shall also wait for at least one T_{idle} condition on the bus between retransmits.

5.2.2.4 Polling mode

487

488 The I3C Controller can operate in polling mode when IBIs are disabled. In this case, the I3C Controller
489 can do a GETSTATUS CCC to find if IBIs are pending (pending read notifications are reported as a
490 pending IBI) or it may simply attempt read transfers from the I3C Target and see if it responds with data.
491 When responding to GETSTATUS CCC, MCTP I3C Targets shall report a non-zero value in the pending
492 interrupt field to indicate that at least one MCTP packet is ready to be read. The convention is that the
493 pending interrupt value 1 is the least important priority and 15 is the most important priority. Unless
494 overridden by a different specification or protocol, the default value should be 7. The I3C Controller is
495 expected to read from the I3C Target if a non-zero pending interrupt value is reported. It may use the
496 value as a relative priority hint.

497 To avoid MCTP packet loss, the I3C Controller shall read the pending packet according to the MCTP
498 timeouts defined in clause 5.8.

5.2.2.5 Sequences of multiple MCTP packets and reads without IBIs

499

500 If the I3C Target has multiple MCTP packets to send to the I3C Controller, it may signal multiple IBIs, one
501 for each packet. This may happen even if waiting for the I3C Controller to initiate a private read request
502 on a prior MCTP packet. An I3C Target may only signal multiple ready packets if it is able to service
503 sequential I3C Controller reads separated by a Repeated Start. I3C Targets that are unable to respond
504 quickly enough to a sequence of reads separated by Repeated Start conditions shall delay IBI
505 notifications of additional packets until after the prior packet is read.

506 The I3C Controller may also do multiple MCTP packets reads in a sequence even without having
507 received multiple IBIs, as in the following examples:

508 • If the I3C Controller receives a multi-packet MCTP message, it may attempt to read subsequent
509   MCTP packets until EOM flag is set in the MCTP header,

510 • The I3C Controller knows that the I3C Target transmits MCTP packets on strictly periodical
511   basis,

512   • The I3C Controller expects more MCTP packets, so it decides to continue reading until a NACK
513   is received (see clause 5.2.2.6 for more information).

514   In the above scenarios, the I3C Target shall not send IBIs related to packets that have been read by the
515   I3C Controller.

516   If IBIs are disabled, the I3C Target shall still respond to private reads and provide the next available
517   MCTP packets.

5.2.2.6   NACKs

518

519   If the I3C Controller attempts a read when the I3C Target has no MCTP packets ready to send, then the
520   I3C Target shall NACK the address byte.

521   The I3C Controller shall follow the flow discussed in clause 5.2.1.2 to differentiate between an I3C Target
522   device no longer present and an I3C Target device NACKing the transfers.

5.2.2.7   Early terminated or prolonged reads

523

524   The I3C Target expects the whole MCTP packet to be read by the I3C Controller. It may happen,
525   however, that the I3C Controller terminates the read transfer too early or too late:

526   • The I3C Controller stops before the I3C Target transmits the whole MCTP packet, including the
527   PEC byte, due to unexpected packet content, packet length limit mismatch, or bus errors.

528   • The I3C Controller continues to drive the clock even after the I3C Target indicated the end of
529   transaction due to bit error on T-bit or clock synchronization error.

530   If this happens, the I3C Controller should interpret the last byte of the received data as PEC to detect
531   packet data corruption and discard the packet. The I3C Target shall infer that the I3C Controller received
532   corrupted MCTP packet and retransmit it again from the beginning on the next private read. If IBIs are
533   enabled, the I3C Target shall use an IBI to notify the I3C Controller that it has a packet waiting just as if it
534   had a new packet for transmission.

5.2.2.8   Future performance enhancements

535

536   IBI speeds are limited to I3C SDR mode only, so the amount of data transferred in the IBI was minimized
537   and instead transferred on a subsequent private read. This enables future migration of reads to I3C HDR
538   mode for more efficient transfer of potentially larger MCTP packets.

5.3   Error detection and handling mechanisms

539

5.3.1   MCTP data packets

540

541   MCTP relies on the underlying transport to provide packet-level error detection. For I3C, the PEC byte is
542   used to detect transmission errors as described in clause 5.2. The polynomial for CRC-8 calculation is as
543   follows – same as SMBus PEC – and the initial value and a final XOR values are zeros:

544   $$C(x) = x^8 + x^2 + x^1 + 1$$

545   The PEC is calculated independently for each MCTP packet using the same method as for SMBus PEC,
546   as defined in *System Management Bus (SMBus) Specification*. It is calculated over the I3C
547   source/destination address field, RnW# bit, and all MCTP packet bytes of an I3C private read or write
548   transfer (not for CCCs, IBIs, or other preamble traffic). The PEC is inserted at the end of each MCTP
549   packet prior to its termination with Stop or Repeated Start condition – see Figure 4 and Figure 6. The
550   PEC calculation excludes Start, Repeated Start, Stop, 7'h7E broadcast address, T-bits, ACK, and NACK.

551   The receiver of the MCTP packet shall verify if the PEC byte is correct for the packet content. If it detects
552   an error, it should discard the received packet.

553   When the sender detects the transmission error, it is recommended to retransmit the corrupted packet.
554   These scenarios are:

555   •   In case of Target-to-Controller transfer, if the transfer is terminated too early or too late, the I3C
556   Target can retransmit the packet – see clause 5.2.2.7 for more information,
557   •   If the I3C Target detects error type S6 or the I3C Controller detects error type M1, then it
558   terminates the data transfer early, as defined in *Specification for I3C BasicSM, Improved Inter*
559   *Integrated Circuit – Basic*. In this case, the MCTP packet can be retransmitted.

560   In the above case, thanks to the last byte interpreted as PEC by the receiver, the error is expected to be
561   detected and the corrupted packet data discarded.

562   ## 5.3.2   CCC error detection and handling

563   The following recommendations should be followed in order to lower the probability of silent errors during
564   I3C CCCs:

565   •   The dynamic addresses should be assigned for maximum Hamming distance between any two
566   addresses without using reserved addresses listed in *Specification for I3C BasicSM, Improved*
567   *Inter Integrated Circuit – Basic* – this is to lower the probability of an incorrect device receiving a
568   CCC,
569   •   CCCs should be individually terminated with a Stop condition – this is to prevent getting stuck in
570   Dynamic Address Assignment mode,
571   •   Table 4 recommends the best workarounds to make mandatory CCCs more reliable.

572   Enhancements for optional CCCs are implementation dependent.

573   **Table 4 – Recommended behaviors for robust CCCs**

CCC	Error Description	Recommended Behavior
GETBCR GETDCR GETMRL GETMWL GETPID GETSTATUS	Incorrect value read (due to bit errors in the data field) or value from wrong device (due to bit errors in the address field)	Keep issuing CCC until 2 consecutive read values match. Discard any reads returning invalid values. For GETSTATUS, if the difference between the first and second reading is only the auto cleared Protocol Error flag they should be considered as a match with the Protocol Error flag set.
ENEC DISEC	Incorrect enable/disable event byte value or wrong address (includes S1, S2 error types defined in *Specification for I3C BasicSM, Improved Inter Integrated Circuit – Basic*)	An unexpected IBI received from a device indicates a redundant DISEC IBI command is required. If an I3C Controller times out when waiting for an IBI from an I3C Target, then it is possible that the I3C Target interrrupts are unintentionally disabled and the I3C Controller should use GETSTATUS to see if an interrupt is pending and enable IBIs again.

CCC	Error Description	Recommended Behavior
ENTDAA	Incorrect CCC received (S1 error defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic* or undetected by parity check) or wrong address (due to bit errors in the address field)	Assign addresses to all participating devices that do not yet have a dynamic address then repeat CCC until two consecutive ENTDAAs do not detect additional devices. Confirm address assignments with ACK from directed traffic to that address.
SETNEWDA	If a new address is set incorrectly, wrong device changes address, two devices may end up with same address (S2 error type defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic*)	I3C Controller shall issue a GETPID CCC before and after SETNEWDA to verify if the same device responds at the new address. If there is an error (device does not respond or a different PID is detected), recovery would be via reasssigning all dynamic addresses on the bus with RSTDAA.
ENTAS0	CCC not recognized by I3C Target (S1 error defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic* or undetected by parity check) or wrong address (due to bit errors in the address field)	AS-type CCCs are just hints so errors can be ignored.
RSTDAA	CCC not recognized by I3C Target (S1 error defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic* or undetected by parity check) or wrong address (due to bit errors in the address field)	Issue the CCC twice.
SETMWL SETMRL	Incorrect value (S2 error defined in *Specification for I3C Basic^SM, Improved Inter Integrated Circuit – Basic* or undetected by parity check) or wrong address (due to bit errors in the address field)	Read back to confirm the value twice with the corresponding GET CCC. If the GET values differ, then keep reading until 2 GET values match. If the matched values differ from the written value, then set it again as per clause 5.4.2.

574
575

576   5.3.3   "Stuck SDA" condition handling

577   A possible error condition exists where an I3C Target that is driving the data line (SDA) of the bus could
578   continue driving the data line even when I3C Controller expects it to be released. This can happen, for
579   example, during read transfer due to a missed clock cycle, during ACK, etc.

580   In order to recover, the I3C Controller shall attempt the following sequence in SDR mode with an early
581   exit as soon as SDA goes high, followed by a Stop condition:

582   1.   The I3C Controller shall drive 8 clocks. The I3C Target is required to drive SDA High for the 9th
583   T-Bit. The I3C Controller shall watch for SDA going High, and stop the read by driving SDA Low
584   when the clock line (SCL) is High.
585   2.   The I3C Controller shall hold SCL level (High or Low) for 150 μs. The I3C Target shall implement
586   a detector that determines if the SCL clock has not changed for 100 μs or more and switch SDA
587   to High-Z and wait for Repeated Start or Stop.
588   3.   The I3C Controller should drive SCL low for at least 35ms.

589   The last recovery step attempts to recover devices that implement SMBus timeout $t_{timeout}$ as defined in
590   *System Management Bus (SMBus) Specification* in *Table 1. SMBus AC specifications, Note 2*. Such
591   devices are expected to release the SDA line after 25ms and be ready to receive a new Start condition
592   after at most 35ms.

593   5.4   MCTP support and capabilities discovery

594   I3C Controller shall be a MCTP-aware device (typically a management controller in the system) and a
595   MCTP bridge. I3C Controller can be connected to various I3C Targets that can support different
596   protocols. For this reason, a discovery method is defined in this clause to allow the I3C Controller to find
597   out which devices talk MCTP and what characteristics they support.

598   5.4.1   Initialization and discovery flow

599   MCTP devices are identified by their Device Characteristic Registers (DCR) value of 0xCC as uniquely
600   reserved by *MIPI Device Characteristics Register (DCR) Assignments*. MCTP devices on an I3C bus shall
601   support Dynamic Address Assignment. Hot-Join IBIs should be used to announce device presence on the
602   bus whenever device requires Dynamic Address Assignment and MCTP initialization. Device interrupts
603   shall be enabled by default after the device is powered on or reset.

604   The I3C Controller discovers which devices on the bus are capable of supporting MCTP by reading their
605   DCR. DCR can be obtained while assigning them addresses as shown in Figure 7, for example. This
606   simplified sample flow only shows setup of a single new hot-pluggable device and does not include
607   discovery of support for MCTP packets larger than 64 bytes or alternative methods to read the DCR or

608   assign addresses. It also assumes that MCTP Bus Owner is collocated with the I3C Controller and the
609   I3C Target is the MCTP Endpoint but note that these roles may be swapped as explained in clause 5.1.3.
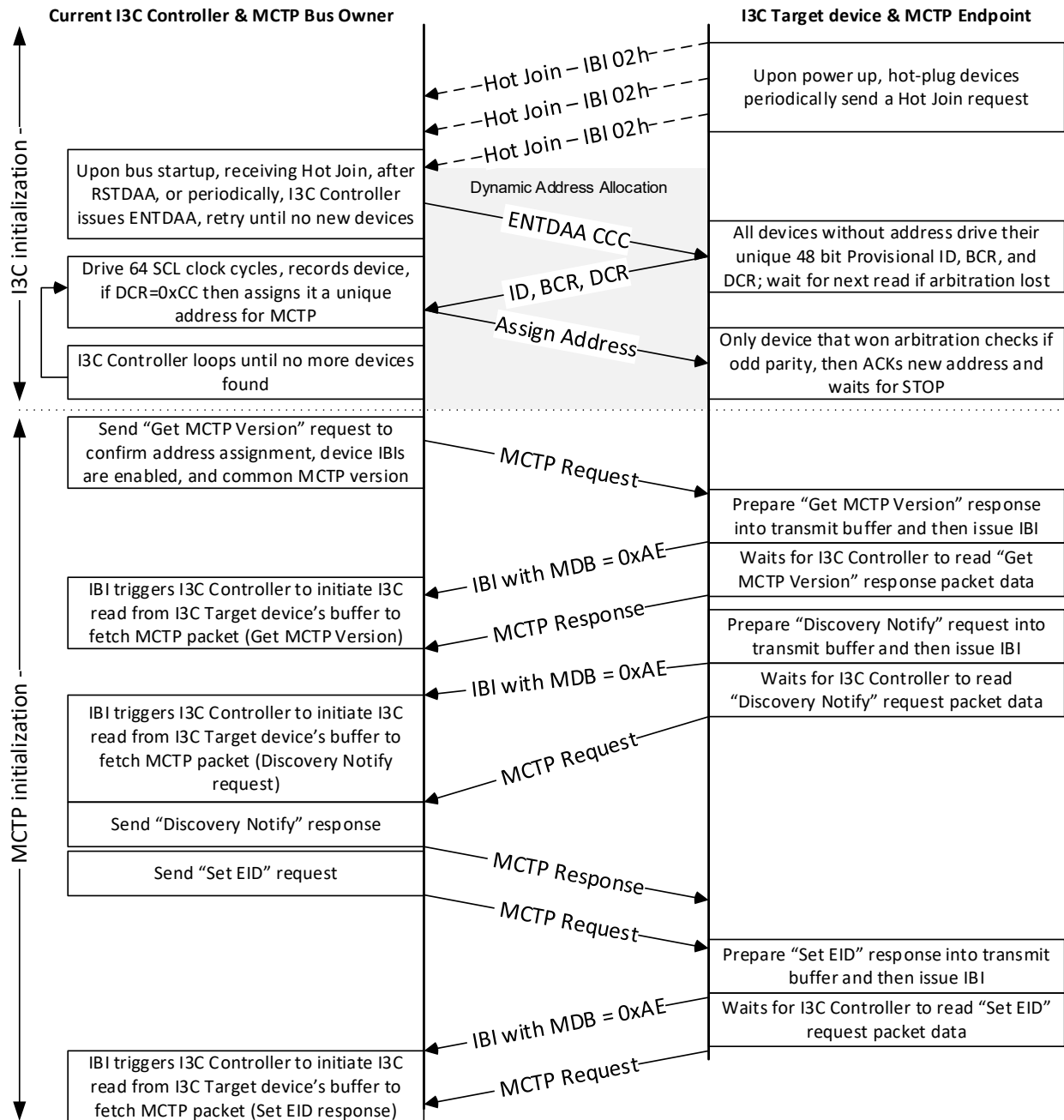
610

611   **Figure 7 – Sample I3C dynamic address assignment flow and MCTP discovery**

612 It is assumed that the I3C Target will only support the MCTP protocol and a unique I3C address will be
613 dynamically assigned for that purpose. In this case, private reads and writes from this I3C address only
614 transfer MCTP packets. After an MCTP-capable I3C Controller discovers that an I3C Target supports
615 MCTP, it shall send an MCTP command to the I3C Target, for example, *Get MCTP Version*. This MCTP
616 command will inform the I3C Target that the I3C Controller supports MCTP.

617 The *Discovery Notify* MCTP message is used during the EID assignment process (see clause 5.1.3 for
618 more information about different logical topologies on I3C bus):

619   • The MCTP Endpoint shall continue sending *Discovery Notify* messages to the MCTP Bus Owner
620   until it is assigned an EID – see MN1 and MT4 in Table 8;
621   • If the I3C Controller is a MCTP Endpoint and does not have the predetermined knowledge about
622   which I3C Target assigns the EID, the I3C Controller is allowed to send the *Discovery Notify*
623   message to multiple I3C Targets.

624 Note that *Prepare for Endpoint Discovery* or *Endpoint Discovery* MCTP control commands are not used to
625 discover MCTP endpoints. I3C devices use the dynamic address assignment process and hot-join
626 mechanisms to discover if other I3C devices are present on or joining the I3C bus (an I3C Target device
627 can only discover the presence of the I3C Controller device, not the rest of the I3C bus, as explained in
628 clauses 5.1.2 and 5.1.3).

629 5.4.2   Transmission unit sizes

630 I3C MCTP devices shall support the minimum of 64 byte MCTP payload as the baseline (see section 8.3
631 in *Management Component Transport Protocol (MCTP) Base Specification*). This results in the minimum
632 I3C transfer size limit that every MCTP over I3C implementation shall support when receiving data: 69
633 bytes (i.e., 64 bytes of MCTP payload + 4 bytes of MCTP header + 1 byte of PEC). The value of 69 is the
634 default baseline transfer length for reads and writes of MCTP over I3C and cannot be negotiated smaller.

635 I3C Target or I3C Controller implementations may support longer transfers than the above default but
636 they shall discover and negotiate their use. Transfer sizes accepted by a particular MCTP Endpoint are
637 discovered as defined in Section 8.3.1 in *Management Component Transport Protocol (MCTP) Base*
638 *Specification*, i.e., via a message type specific mechanism. Transfer sizes of a path are discovered
639 according to section 9.5 in *Management Component Transport Protocol (MCTP) Base Specification*, i.e.,
640 using Query Hop MCTP commands sent to each bridge on the path.

641 In order to respond to Query Hop command, I3C devices that implement the MCTP bridging functionality
642 and transmission units larger than the baseline minimum shall use SETMWL/GETMWL and/or
643 SETMRL/GETMRL I3C CCCs to establish the maximum transfer length from I3C Controller to I3C Target
644 or from I3C Target to I3C Controller, respectively. Each direction may support a different length limit. If
645 these pairs of CCCs are not used or not supported, it means that the baseline minimum is used for a
646 specific direction of communication.

647 For the Controller-to-Target direction (transfers defined in clause 5.2.1), packet size limit bigger than the
648 baseline minimum may be optionally established according to the following flow:

649   (1) The I3C Controller sends SETMWL CCC to the I3C Target with the length equal to the maximum
650   length the I3C Controller would like to send. If the I3C Target is capable to support the new
651   length, it will accept it or otherwise it will change its maximum write length to the largest value it
652   can support.
653   (2) The I3C Controller sends GETMWL CCC to the I3C Target (clause 5.3.2 rules shall be followed
654   to verify the correctness of the transfer).The I3C Target responds with its current maximum write
655   length.

656 For the Target-to-Controller direction (transfers defined in clause 5.2.2), packet size limit bigger than the
657 baseline minimum may be optionally established according to the following flow:

658   (1) The I3C Controller sends SETMRL CCC to the I3C Target with the length equal to the maximum
659   length the I3C Controller would like to receive. If the I3C Target is capable to support the new
660   length, it will accept it or otherwise it will change its maximum read length to the largest value it
661   can support.
662   (2) The I3C Controller sends GETMRL CCC to the I3C Target (clause 5.3.2 rules shall be followed to
663   verify the correctness of the transfer). The I3C Target responds with its current maximum read
664   length.

665 As defined in clause 5.3.2, the above sequences may be repeated to detect and correct any transmission
666 errors.

667 The size values in these CCCs shall include the PEC defined in clause 5.3.1 as well as the MCTP header
668 fields. They do not include the I3C Target address fields[2]. Please note that SETMRL/GETMRL CCCs also
669 need to report the IBI payload size (because an I3C Target that supports MCTP shall support IBIs).

670 If these CCCs are implemented, they indicate the upper bound accepted by the I3C devices. MCTP
671 maximum transmission unit cannot exceed these limits. Not all MCTP packets are of maximum length.
672 Some MCTP packets may be shorter than the above limits (either baseline or negotiated length). I3C
673 transfers will indicate the actual size of a particular packet (for reads, the T-bit is used by the I3C Target
674 to indicate end of data; for writes the I3C Controller ends the transfer with a Stop or Repeated Start). No
675 padding is needed in such a case.

5.5   Supported media

676

677 The transport binding defined in this specification has been designed to work with I3C buses. The I3C
678 media type identifier for this binding spec is defined in *Management Component Transport Protocol*
679 *(MCTP) IDs and Codes*, section 7 *MCTP physical medium identifiers*.

5.6   Physical address format for MCTP control messages

680

681 The address format shown in Table 5 shall be used for MCTP control commands that require a physical
682 address parameter to be returned for a bus that uses this transport binding with one of the supported
683 media types listed in 5.5. This includes commands such as the Resolve Endpoint ID, Routing Information
684 Update, and Get Routing Table Entries commands.

685 **Table 5 – Physical address format**

Format Size	Layout and Description	
1 byte	[7:1]	I3C address bits
	[0]	0b

686 A valid I3C address shall be used to refer to an I3C Target. Since the I3C Controller does not really have
687 any address, a special value of zero (7'h00) is used to indicate the I3C Controller when it is necessary.

---

[2] I3C specification does not clearly define if the I3C address field is included, but this is the interpretation agreed at
MIPI when working on this specification.

688 ## 5.7   Get endpoint ID medium-specific information

689 The medium-specific information as shown in Table 6 shall be used for the medium-specific Information
690 field returned in the response to the Get Endpoint ID MCTP control message.

691   **Table 6 – Medium-specific information**

Description
[7:0]   reserved

692 ## 5.8   MCTP packet and control message timing requirements

693 In I3C, all traffic passes through the I3C Controller and it is responsible for all bus timing and fairness.
694 The I3C Controller should attempt to ensure all device traffic makes progress, but in some cases the I3C
695 Controller may disable interrupts or postpone some traffic to focus on higher priorities.

696 I3C Targets should retry packet transmission (i.e., repeat IBI notifications) until the I3C Controller reads
697 the packet or pauses retries by disabling interrupts. If the PT expires, the endpoint may silently discard
698 the packet. In some implementations a queued packet cannot be modified or retracted.

699 When an I3C Target does not accept MCTP packets from the I3C Controller, the I3C Controller may
700 confirm the presence of the I3C Target with GETSTATUS CCC, as described in clause 5.2.1.2. If the I3C
701 Target is present, the I3C Controller may keep retrying indefinitely or stop after PT elapses.

702   **Table 7 – Timing specifications for MCTP data packets on I3C**

Timing Specification	Symbol	Min	Max	Description
Endpoint packet level timeout	PT	100ms	-	The minimum period for how long an endpoint shall attempt retransmissions: • the I3C Target shall retry an IBI, when it is NACKed by the I3C Controller or ACKed without a read, • the I3C Controller shall retry writing a packet to an I3C Target. There should be at least 8 retry attempts distributed throughout the PT time before timing out. Arbitration losses are not counted as an attempt. *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* defines the minimum retry intervals and conditions for NACKed IBIs, private writes, and arbitration losses. An I3C Target should pause PT when IBIs are disabled.

703   **Table 8 – Timing specifications for MCTP control messages on I3C**

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	T_{RECLAIM}	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	Total of three tries, minimum: the original try plus two retries. The maximum number of retries for a given request is limited by the requirment that all retries shall occur within MT4, max of the initial request.
Request-to-response time	MT1	–	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 (that is, beginning of IBI for transfer initiated by I3C Target or beginning of the write transfer for the transfer initiated by the I3C Controller). 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	Time to take into account transmission delay of an MCTP Control Protocol message. Measured as the 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 measured from 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
Instance ID expiration interval	MT4	5 sec [2]	6 sec	Interval after which the instance ID for a given response will expire and become reusable if a response has not been received for the request. This is also the maximum time that a responder tracks an instance ID for a given request from a given requester.

Timing Specification	Symbol	Min	Max	Description
NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.				
NOTE 2: If a requester is reset, it may produce the same sequence number for a request as one that was previously issued. To guard against this, it is recommended that sequence number expiration be implemented. Any request from a given requester that is received more than MT4 seconds after a previous, matching request should be treated as a new request, not a retry.				

704   **ANNEX A**
705   **(informative)**
706

707   **Notation**

708   **Notations**

709   Examples of notations used in this document are as follows:

710   •   2:N   In field descriptions, this will typically be used to represent a range of byte offsets
711   starting from byte two and continuing to and including byte N. The lowest offset is on
712   the left, the highest is on the right.

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

715   •   (3:6)   Parentheses around a field consisting of a range of bytes indicates the entire range
716   may be present or absent. The lowest offset is on the left, the highest is on the right.

717   •   PCIe   Underlined, blue text is typically used to indicate a reference to a document or
718   specification called out in 2, "Normative References" or to items hyperlinked within the
719   document.

720   •   [4]   Square brackets around a number are typically used to indicate a bit offset. Bit offsets
721   are given as zero-based values (that is, the least significant bit offset = 0).

722   •   [7:5]   A range of bit offsets. The most significant bit is on the left, the least significant bit is
723   on the right.

724   •   1b   The lower case "b" following a number consisting of 0s and 1s is used to indicate the
725   number is being given in binary format.

726   •   0x12A   A leading "0x" is used to indicate a number given in hexadecimal format.

727

728 # ANNEX B
729 # (informative)
730
731 # Change log

Version	Date	Description
1.0.0	2021-06-24	Released as DMTF Standard
1.0.1	2024-03-25	Additional explanations related to: - PT timeout handling and retries during this period, - PEC calculation, - IBI prioritization convention, - consistent use of "I3C Controller" and "I3C Target" terminology, as opposed to "Master" and "Slave" or "Primary" and "Secondary".

732