Document Identifier: DSP0233

Date: 2021-06-23

Version: 1.0.0

Management Component Transport Protocol (MCTP) I3C Transport Binding Specification

Supersedes: None

Document Class: Normative

Document Status: Published

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

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

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

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

This version is the first version of this document. Future changes with be detailed in the change log in ANNEX B.

Acknowledgments

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

Editors:

- Janusz Jurski – Intel Corporation
- Myron Loewen – Intel Corporation
- Amit K Srivastava – Intel Corporation

DMTF Contributors:

- Patrick Caporale – Lenovo
- Yuval Itkin – NVIDIA Corporation
- John Leung – Intel Corporation
- Eliel Louzoun – Intel Corporation
- Balaji Natrajan – Microchip Technology Inc.
- Edward Newman – Hewlett Packard Enterprise
- Jim Panian – Qualcomm Incorporated
- William Scherer III – Hewlett Packard Enterprise
- Hemal Shah – Broadcom Inc
- Bob Stevens – Dell Inc.
- Johan Van De Groenendaal – Intel Corporation

MIPI Contributors:

- Guruprasad Ardhanari – Intel Corporation
- Jerry Backer – Intel Corporation
- Rajesh Bhaskar – Intel Corporation
- Enrico D Carrieri – Intel Corporation
- Anamitra Chakrabarti – Synopsys, Inc.
- Kelly J Couch – Intel Corporation
- Kenneth P Foust – Intel Corporation
- Chris Grigg – MIPI Alliance (Team)
- David Harriman – Intel Corporation
- Prakash Iyer – Intel Corporation
- Paul Kimelman – NXP Semiconductors
- Lukasz Lanecki – Intel Corporation

- Richard Marian Thomaiyar – Intel Corporation
- Tim E Mckee – Intel Corporation
- Mariusz Oriol – Intel Corporation
- Ankit Rohatgi – MIPI Alliance (Team)
- Matthew Schnoor – Intel Corporation
- Eyuel Zewdu Teferi – STMicroelectronics
- George Vergis – Intel Corporation

Introduction

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

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

NOTE: The terms Master and Slave used in the referenced document have been replaced in this document with the terms Primary and Secondary.

Management Component Transport Protocol (MCTP) I3C Transport Binding Specification

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

Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic, Version 1.0, https://www.mipi.org/specifications/i3c-sensor-specification

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

System Management Bus (SMBus) Specification v2.0, SBS Implementers Forum, SMBus, August 2000, http://www.smbus.org/specs/smbus20.pdf

MIPI I3C℠ Host Controller Interface℠ v1.0 Specification, MIPI, April 2018, https://www.mipi.org/specifications/i3c-hci

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.

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

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

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

3.1

Address Resolution Protocol

ARP

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

3.2

ACK

Acknowledge

3.3

BCR

Bus Characteristics Register – see *Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic* for more information

3.4

BMC

Baseboard management controller

3.5

CCC

Common Command Code – see *Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic* for more information

3.6

Destination Device

Device receiving the MCTP packet over I3C bus

3.7

EEPROM

Electrically Erasable Programmable Read-Only Memory

3.8

EID

Endpoint identifier

3.9

HCI

Host Controller Interface

3.10

Hot-Join

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

3.11

I3C

Improved Inter-Integrated Circuit – see *Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic* for more information

3.12

IBI

In-Band Interrupt – see *Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic* for more information

3.13

3.14

max

maximum

3.15

MCTP

Management Component Transport Protocol

3.16

MDB

Mandatory Data Byte

3.17

MHz

megahertz

3.18

min

minimum

3.19

ms

millisecond

3.20

MSB

most significant byte

3.21

MTU

Maximum Transmission Unit

3.22

NACK

not acknowledge

3.23

PCI

peripheral component interconnect

3.24

PCIe®

PCI Express™

3.25
PEC

packet error code

3.26
PMCI

Platform Management Component Intercommunications

3.27
Primary

3.28 Alternative term for the current I3C Master as defined in *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic*

SCL

serial clock

3.29
SDA

serial data

3.30
sec

second

3.31
Secondary

Alternative term for I3C Slave as defined in *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic*

3.32
SMBus

System Management Bus

3.33
Source Device

Device sending the MCTP packet over I3C bus

3.34
T-bit

Transition bit – see *Specification for I3C Basic^{SM}, Improved Inter Integrated Circuit – Basic* for more information

3.35
UDID

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 the 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 Primary (typically it is a Management Controller, Embedded Controller, etc.) and there may be multiple Secondaries sharing the same I3C bus. *Specification for I3C Basic[SM], Improved Inter Integrated Circuit – Basic* defines the secondary Primary flow but that is not required for implementing MCTP and is out of scope for this specification.

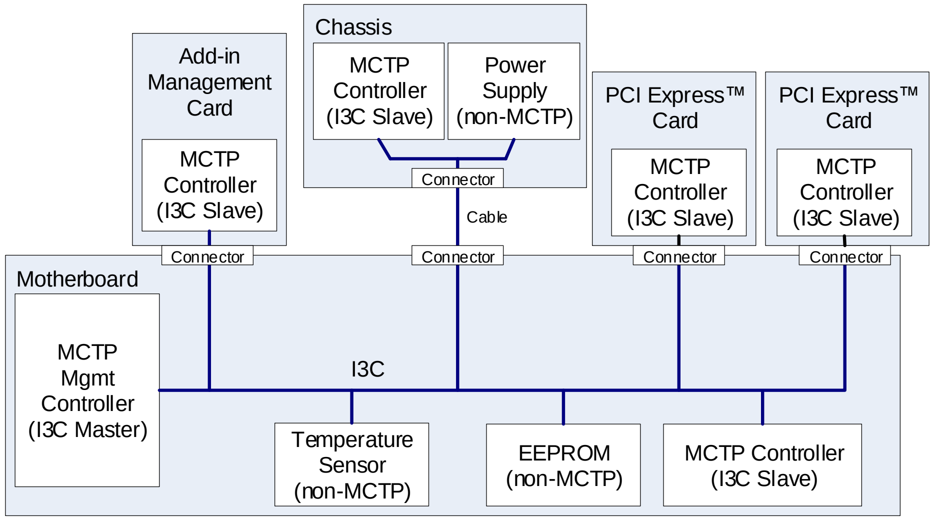

Figure 1 – Physical topology of I3C bus

5.1.2 I3C communication logical topology and MCTP packet bridging

The topology of the logical communication paths is shown in Figure 2. The Primary can communicate to any of the Secondaries. Each Secondary can communicate with the Primary only. Any communications between Secondaries are implemented by MCTP bridge functionality in the Primary, according to the *Management Component Transport Protocol (MCTP) Base Specification*. Unlike typical MCTP bridges that transfer data to another port, this data may be retransmitted to the same port. When forwarding, the physical addressing and PEC gets changed by the bridge to match the requirements of the destination bus.

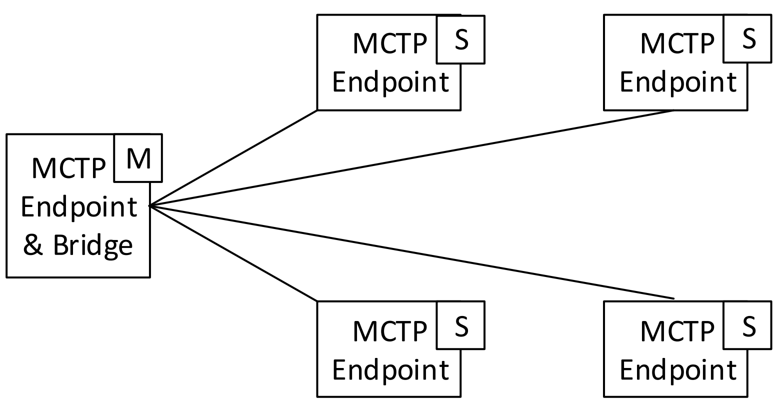

Figure 2 – Logical topology of MCTP over I3C communication

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

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

5.1.3 MCTP Bus Owner for I3C bus[1]

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

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

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

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

- *Logical MCTP Bus 1*, *Bridge 1* (I3C Secondary) is the MCTP Bus Owner – *Bridge 1* assigns the EID and EID pool to the I3C Primary because I3C Primary is an endpoint on the *Logical MCTP Bus 1*.
- *Logical MCTP Bus 2*, *Bridge 2* (I3C Primary) is the MCTP Bus Owner for *Logical MCTP Bus 2* – *Bridge 2* assigns EIDs to all the remaining I3C Secondaries.

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

---

[1] The term "bus" is used in a different meaning in *Specification for I3C Basic℠, 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.

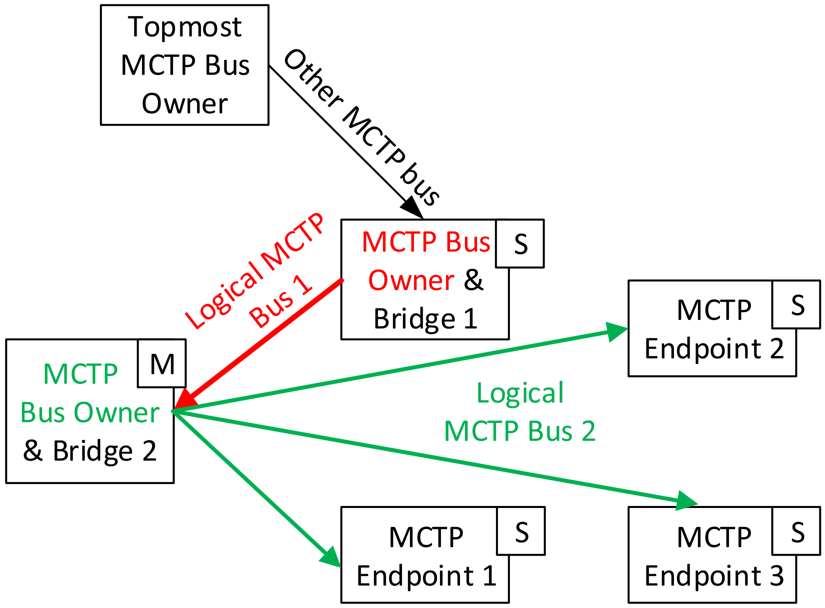

Figure 3 – Sample I3C Secondary as MCTP Bus Owner & bridge

5.2 MCTP packet encapsulation

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

- Primary to Secondary communication follows encapsulation defined in clause 5.2.1
- Secondary to Primary communication follows encapsulation defined in clause 5.2.2

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

5.2.1 MCTP packet encapsulation: Primary to Secondary

5.2.1.1 Overview

Transmission of MCTP packets from the Primary to the Secondary happens using Primary-initiated private write transfer as defined in *Specification for I3C Basic[SM], Improved Inter Integrated Circuit – Basic*. The transfer shall be directed to the Secondary I3C address used for MCTP protocol communication. For the purpose of this specification, a Secondary shall only support the MCTP protocol at its unique Secondary address and an I3C address shall be dynamically assigned for that purpose. See clause 5.4.1 for discussion about protocol discovery when other protocols may be in use on the I3C bus.

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

Note that the Secondary does not need I3C address of the Primary because all MCTP packets from a given Secondary will always be directed to the Primary – the Primary has no explicit address as per *Specification for I3C Basic℠, Improved Inter Integrated Circuit – Basic*. MCTP Destination EID should be used to route the MCTP packet to another Secondary if necessary.

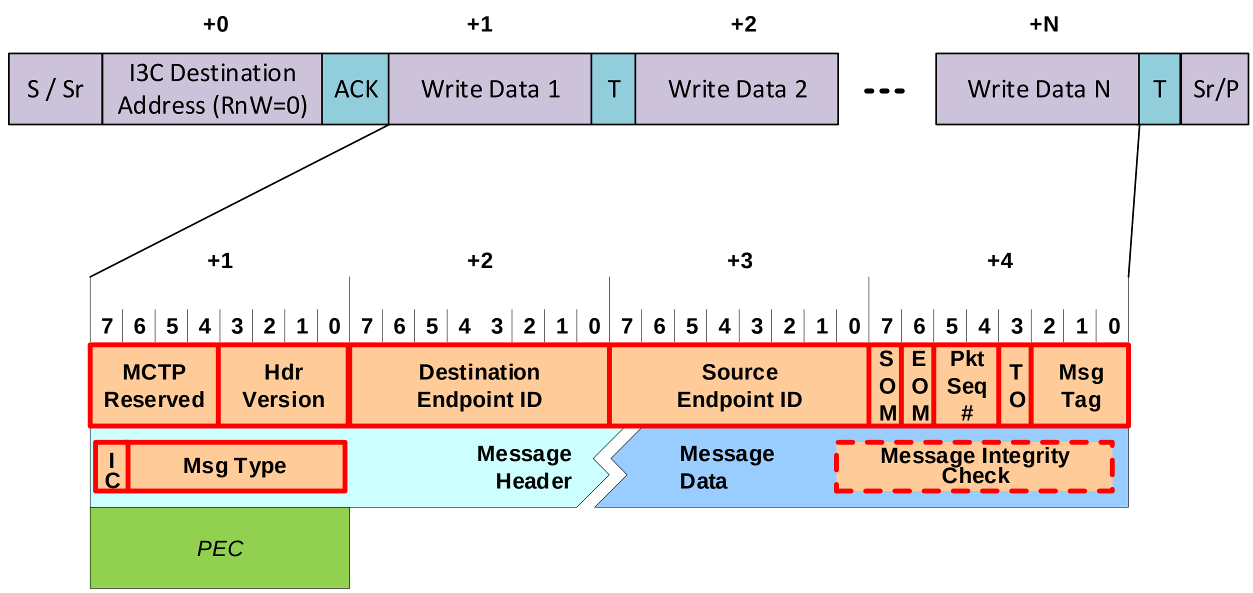

Figure 4 – MCTP over I3C packet transfer format: Primary to Secondary

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

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

Table 1 – MCTP packet transfer field descriptions

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

Byte	I3C Field(s)	Description
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 Secondary address ACKs/NACKs

The Primary 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 Secondary 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 a Secondary address may indicate that the device buffers are full or the physical absence of the device. The Primary may test for the presence of a device after a NACK with the GETSTATUS CCC. The Secondary shall always respond to GETSTATUS CCC, even if its MCTP data buffer is full. The Primary 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: Secondary to Primary

Transmission of MCTP packets from Secondary to Primary can happen in two modes:

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

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

5.2.2.1 Overview – IBI mode

Transmission of MCTP packets from Secondary to Primary according to the IBI mode shall happen using the following general sequence:

1. When the Secondary has a MCTP packet ready for transmission to the Primary, it shall initiate an I3C IBI with MDB = 0xAE (as assigned in *MIPI Mandatory Data Byte (MDB) Values Table* registry) to inform the Primary about the data ready.
2. The Primary shall read the MCTP packet (or multiple packets) from the Secondary using *I3C Private Read transfer*.

This sequence is illustrated in Figure 5:

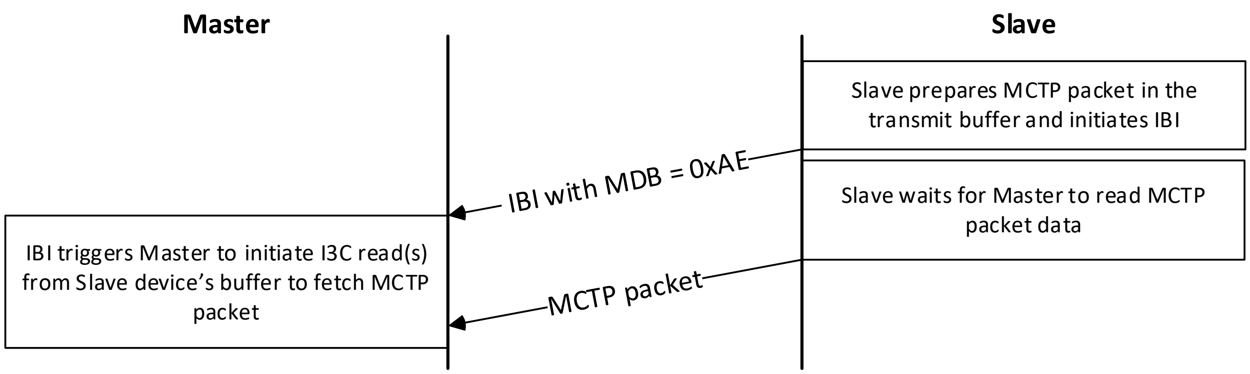

Figure 5 – MCTP over I3C packet transfer sequence: Secondary to Primary

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

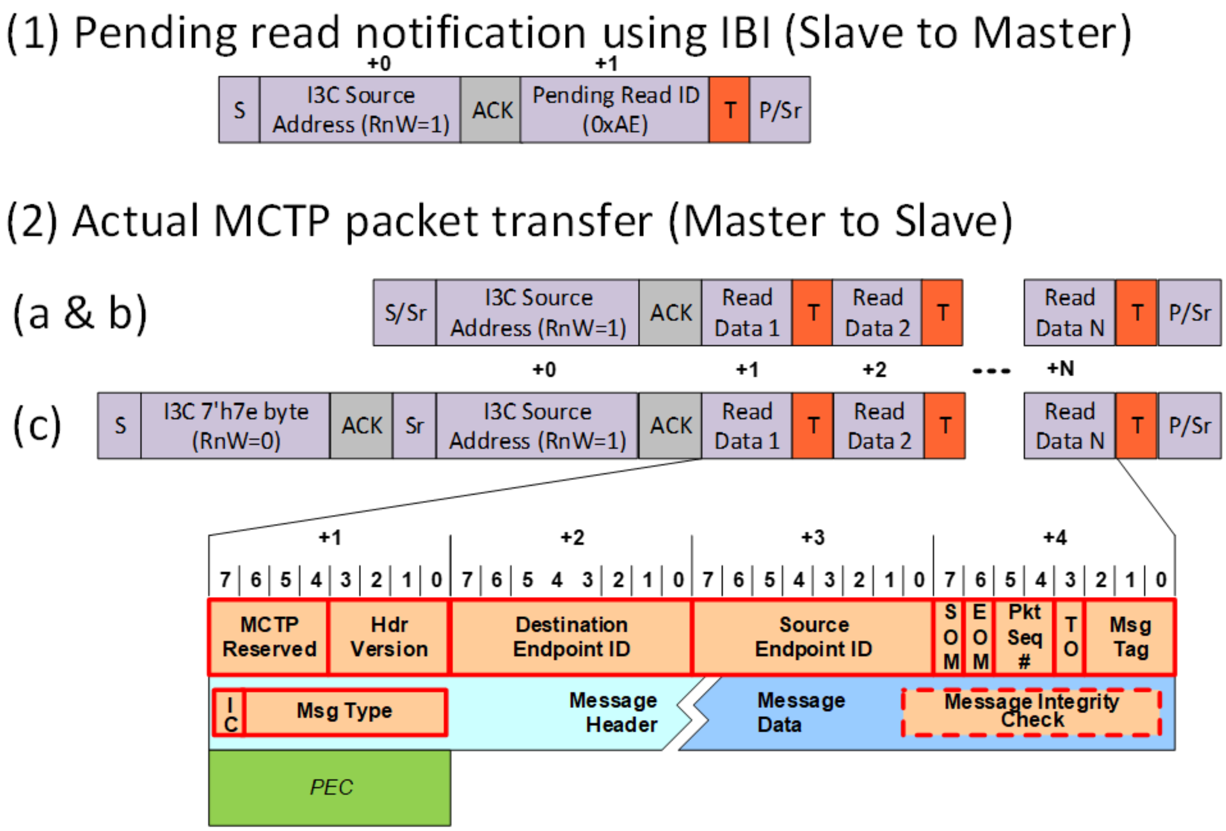

Figure 6 – MCTP over I3C packet transfer format: Secondary to Primary

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

(a) Repeated Start condition and the I3C Source Secondary address immediately after the IBI or other traffic – for example, using HCI auto-command as defined in *MIPI I3C℠ Host Controller Interface℠ Specification*

(b) Start condition and the Secondary address

(c) Start condition with I3C Broadcast Address (7'h7E), then Repeated Start with the Secondary address

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

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

Table 2 – IBI pending read notification field descriptions

Byte	I3C Field(s)	Description
0	Source Address RnW	[7:1] I3C Source Address: The address of the Secondary device [0]: I3C RnW# bit: Shall be set to `1b` for all IBIs
1	Mandatory Data Byte (MDB)	[7:0] 0xAE value – MCTP Pending Read ID notiofication as defined in *MIPI Mandatory Data Byte (MDB) Values Table* registry

Table 3 – MCTP packet transfer field descriptions

Byte	I3C Field(s)	Description
0	Source Address RnW	[7:1] I3C Source Address: The address of the Secondary device [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*.		

5.2.2.2 Detailed flow

When a Secondary 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 Primary that an MCTP packet is available for reading from the Secondary. The Primary should acknowledge the IBI request and read the MDB data from the Secondary. After accepting the request, the Primary may read the packet immediately with a Repeated Start after the IBI or it may do this at a later time. The Primary may queue up several IBI notifications from multiple Secondaries and process them in any order. Delaying reads allows prioritization as well as management of shared buffers.

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

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

5.2.2.3 ACKed IBI retransmissions

Each Secondary shall also implement a timeout mechanism in order to retransmit the IBI. The timer shall be started when the Primary acknowledges the IBI with MDB and the Secondary is allowed to retransmit the IBI if the read has not happened after this timeout expiration. The number of retransmits and the timeout value are implementation dependent – depends on the Secondary functionality and characteristics of MCTP traffic (urgency of retransmission) and shall conform to clause 5.8 requirements. The Secondary shall also wait for at least one T_{idle} condition on the bus between retransmits.

5.2.2.4 Polling mode

The Primary can operate in polling mode when IBIs are disabled. In this case, the Primary can do a GETSTATUS CCC to find if IBIs are pending or it may simply attempt read transfers from the Secondary and see if it responds with data. The Primary should ensure a packet read happens within PT timeout, as per clause 5.8.

5.2.2.5 Sequences of multiple MCTP packets and reads without IBIs

If the Secondary has multiple MCTP packets to send to the Primary, it may signal multiple IBIs, one for each packet. This may happen even if waiting for the Primary to initiate a private read request on a prior MCTP packet. A Secondary may only signal multiple ready packets if it is able to service sequential Primary reads separated by a Repeated Start. Slaves that are unable to respond quickly enough to a sequence of reads separated by Repeated Start conditions shall delay IBI notifications of additional packets until after the prior packet is read.

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

- If the Primary receives a multi-packet MCTP message, it may attempt to read subsequent MCTP packets until EOM flag is set in the MCTP header,
- The Primary knows that the Secondary transmits MCTP packets on strictly periodical basis,
- The Primary expects more MCTP packets, so it decides to continue reading until a NACK is received (see clause 5.2.2.6 for more information).

In the above scenarios, the Secondary shall not send IBIs related to packets that have been read by the Primary.

If IBIs are disabled, the Secondary shall still support Primary-initiated reads and provide the next available MCTP packets.

5.2.2.6 NACKs

If the Primary attempts a read when the Secondary has no MCTP packets ready to send, then the Secondary shall NACK the address byte.

The Primary shall follow the flow discussed in clause 5.2.1.2 to differentiate between a Secondary device no longer present and a Secondary device NACKing the transfers.

5.2.2.7 Early terminated or prolonged reads

The Secondary expects the whole MCTP packet to be read by the Primary. It may happen, however, that the Primary terminates the read transfer too early or too late:

- The Primary stops before the Secondary transmits the whole MCTP packet, including the PEC byte, due to unexpected packet content, packet length limit mismatch, or bus errors.
- The Primary continues to drive the clock even after the Secondary indicated the end of transaction due to bit error on T-bit or clock synchronization error.

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

5.2.2.8 Future performance enhancements

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

5.3 Error detection and handling mechanisms

5.3.1 MCTP data packets

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

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

The PEC is calculated independently for each MCTP packet. The PEC calculation restarts with each Start or Repeated Start condition and is inserted after each MCTP packet prior to its termination with Stop or Repeated Start condition.

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

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

- In case of Secondary-to-Primary transfer, if the transfer is terminated too early or too late, the Secondary can retransmit the packet – see clause 5.2.2.7 for more information,
- If the Secondary detects error type S6 or the Primary detects error type M1, then it terminates the data transfer early, as defined in *Specification for I3C Basic[SM], Improved Inter Integrated Circuit – Basic*. In this case, the MCTP packet can be retransmitted.

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

5.3.2 CCC error detection and handling

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

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

Enhancements for optional CCCs are implementation dependent.

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 Basic[SM], Improved Inter Integrated Circuit – Basic*)	An unexpected IBI received from a device indicates a redundant DISEC IBI command is required. If Primary times out when waiting for an IBI from a Secondary, then it is possible that the Secondary interrrupts are unintentionally disabled and the Primary should use GETSTATUS to see if an interrupt is pending and enable IBIs again.
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 consectutive ENTDAAs do not detect additional devices. Confirm address assignments with ACK from directed traffic to that address.

CCC	Error Description	Recommended Behavior
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℠, Improved Inter Integrated Circuit – Basic*)	Primary 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 target (S1 error defined in *Specification for I3C Basic℠, 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 target (S1 error defined in *Specification for I3C Basic℠, 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℠, 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.

5.3.3 “Stuck SDA" condition handling

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

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

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

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

5.4 MCTP support and capabilities discovery

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

5.4.1 Initialization and discovery flow

MCTP devices are identified by their Device Characteristic Registers (DCR) value of 0xCC as uniquely reserved by [MIPI Device Characteristics Register (DCR) Assignments](#). MCTP devices on an I3C bus shall support Dynamic Address Assignment and removable devices shall issue Hot-Join IBIs on power up. Device interrupts shall be enabled by default after the device is powered on or reset.

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

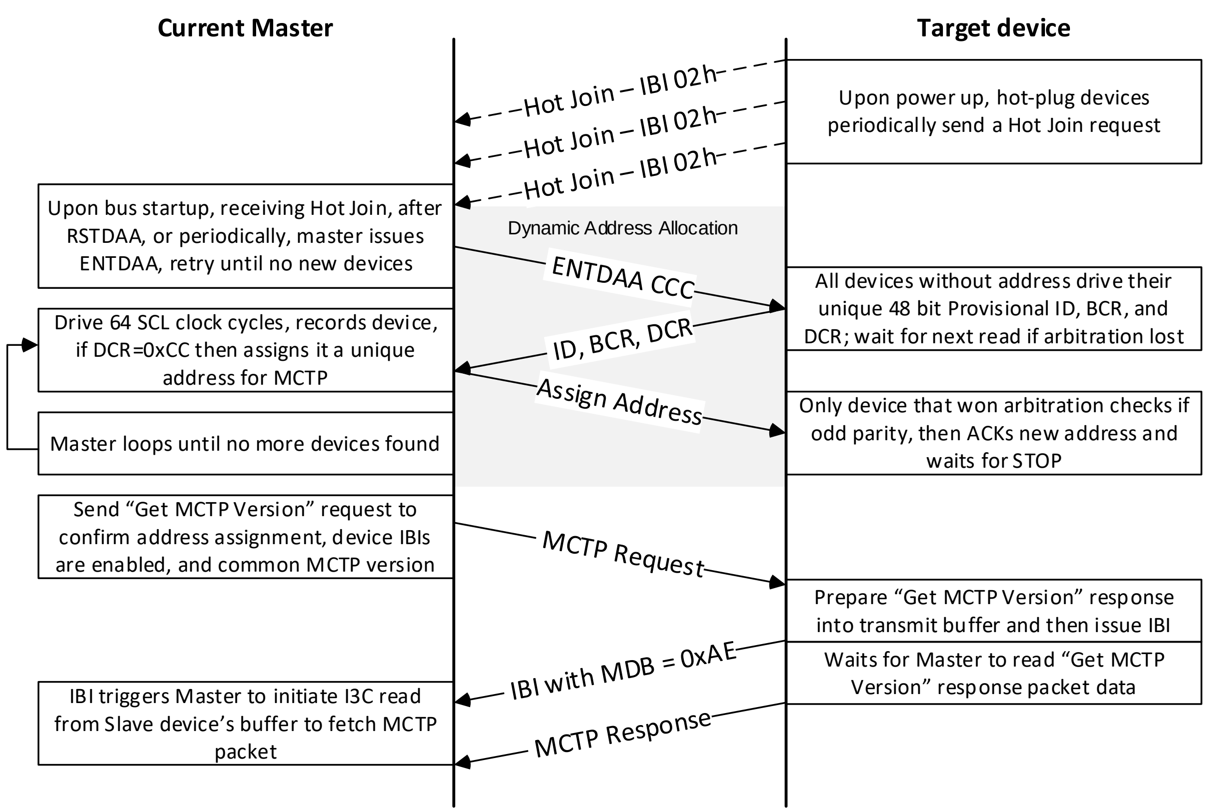

Figure 7 – Sample I3C dynamic address assignment flow and MCTP discovery

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

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

- If the I3C Secondary is waiting for the EID to be assigned by the I3C Primary, it shall send the *Discovery Notify* message to the I3C Primary to trigger the EID assignment process;
- If the I3C Primary anticipates its EID could be assigned by a particular Secondary, it shall send the *Discovery Notify* message directly to the Secondary to trigger the EID assignment process;
- If the I3C Primary does not have the predetermined knowledge about which Secondary assigns the EID, the I3C Primary is allowed to send the *Discovery Notify* message to multiple Slaves.

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

5.4.2 Transmission unit sizes

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

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

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

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

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

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

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

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

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

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

5.5 Supported media

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

5.6 Physical address format for MCTP control messages

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

Table 5 – Physical address format

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

A valid I3C address shall be used to refer to a Secondary. Since the Primary does not really have any address, a special value of zero (7’h00) is used to indicate the Primary 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.

5.7 Get endpoint ID medium-specific information

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

Table 6 – Medium-specific information

Description	
[7:0]	reserved

5.8 MCTP packet and control message timing requirements

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

Slaves should retry packet transmission (i.e., repeat IBI notifications) until the Primary pauses retries with disabled interrupts or reads the packet. In some implementations a queued packet cannot be modified or retracted. If the PT expires, the endpoint may silently discard the packet.

When a Secondary does not accept MCTP packets from a Primary, the Primary may confirm the presence of the Secondary with GETSTATUS CCC, as described in clause 5.2.1.2. If the Secondary is present, the Primary may keep retrying indefinitely or stop after PT elapses.

Table 7 – Timing specifications for MCTP data packets on I3C

Timing Specification	Symbol	Value	Description
Endpoint packet level timeout	PT	100ms	The minimum time an endpoint shall attempt resending an MCTP packet in the following scenarios: • the Secondary shall retry an IBI, when it is NACKed by the Primary, or ACKed without a read, or if interrupts are disabled (IBIs that lose arbitration are not counted as an attempt), • the Primary shall retry writing a packet to a Secondary. If interrupts are enabled, there should be at least 8 retry attempts before timing out.

Table 8 – Timing specifications for MCTP control messages on I3C

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	5 sec	-	Minimum time that a bus owner shall wait before reclaiming the EID for a non-responsive hot-plug endpoint (i.e., not ACKing repeated GETSTATUS CCCs).
Number of request retries	MN1	2	none	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.

Timing Specification	Symbol	Min	Max	Description
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 Secondary initiated transfer or beginning of the write transfer for the Primary initiated transfer). 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.				

ANNEX A
(informative)

Notation

Notations

Examples of notations used in this document are as follows:

- 2:N — In field descriptions, this will typically be used to represent a range of byte offsets starting from byte two and continuing to and including byte N. The lowest offset is on the left, the highest is on the right.
- (6) — Parentheses around a single number can be used in message field descriptions to indicate a byte field that may be present or absent.
- (3:6) — Parentheses around a field consisting of a range of bytes indicates the entire range may be present or absent. The lowest offset is on the left, the highest is on the right.
- PCIe — Underlined, blue text is typically used to indicate a reference to a document or specification called out in 2, "Normative References" or to items hyperlinked within the document.
- [4] — Square brackets around a number are typically used to indicate a bit offset. Bit offsets are given as zero-based values (that is, the least significant bit offset = 0).
- [7:5] — A range of bit offsets. The most significant bit is on the left, the least significant bit is on the right.
- `1b` — The lower case "`b`" following a number consisting of `0`s and `1`s is used to indicate the number is being given in binary format.
- `0x12A` — A leading "`0x`" is used to indicate a number given in hexadecimal format.

ANNEX B
(informative)

Change log

Version	Date	Description
1.0.0	2021-06-23	