MDC MANAGEMENT DEVELOPERS CONFERENCE

December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

Management Component Transport Protocol (MCTP)

Tom Slaight

Intel Corporation

Abstract

- The Management Component Transport Protocol (MCTP) is a protocol defined by the DMTF Platform Management Component Intercommunications sub-team of the DMTF Pre-OS Workgroup. MCTP is designed to support communications between different intelligent hardware components that make up a platform management subsystem that provides monitoring and control functions inside a managed system. This protocol is independent of the underlying physical bus properties, as well as the "data-link" layer messaging used on the bus. The MCTP communication model includes a message format, transport description, message exchange patterns, and operational endpoint characteristics. This talk provides a technical overview of the architectural goals and general operation of MCTP and how MCTP is used over SMBus/I2C and PCIe media.
- Thursday, 10/6, 11:40-12:30

MCTP – what is it?

Management

Component

Transport

Protocol

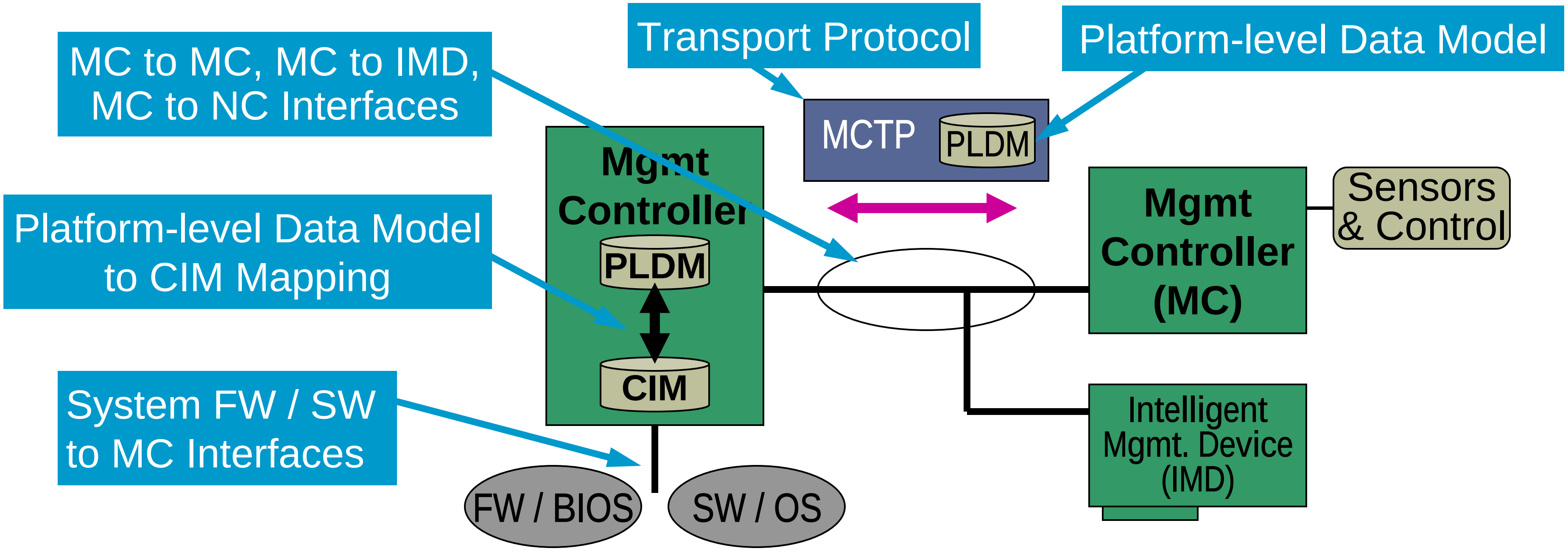
- A media independent, message transport protocol for *intra-system* communication of management traffic between:
  - Management Controllers and Intelligent Management Devices
  - Management Controllers and other Management Controllers
  - Management Controllers & system f/w
- Designed for use over multiple bus types
  - e.g. SMBus/I2C, PCIe VDM, USB, etc.
- Suitable for all types of computer platforms:
  - server, desktop, mobile, comms
- DMTF Industry Specification
  - From PMCI subteam of the DMTF Pre-OS workgroup
  - Specs 1.0.0 Preliminary as of June/July 2007

MCTP Overview

Management Component Transport Protocol

- For ‘inside the box’ communication of platform management traffic
- Carries multiple message types
  - e.g. MCTP Control, Platform Level Data Model, Network sideband, etc.
- Suitable for use with multiple media types
- Suitable to all computer platform types: server, desktop, mobile, comms
- Designed for implementation using low-cost microcontrollers

MCTP - where it fits



MC to MC, MC to IMD, MC to NC Interfaces

Transport Protocol

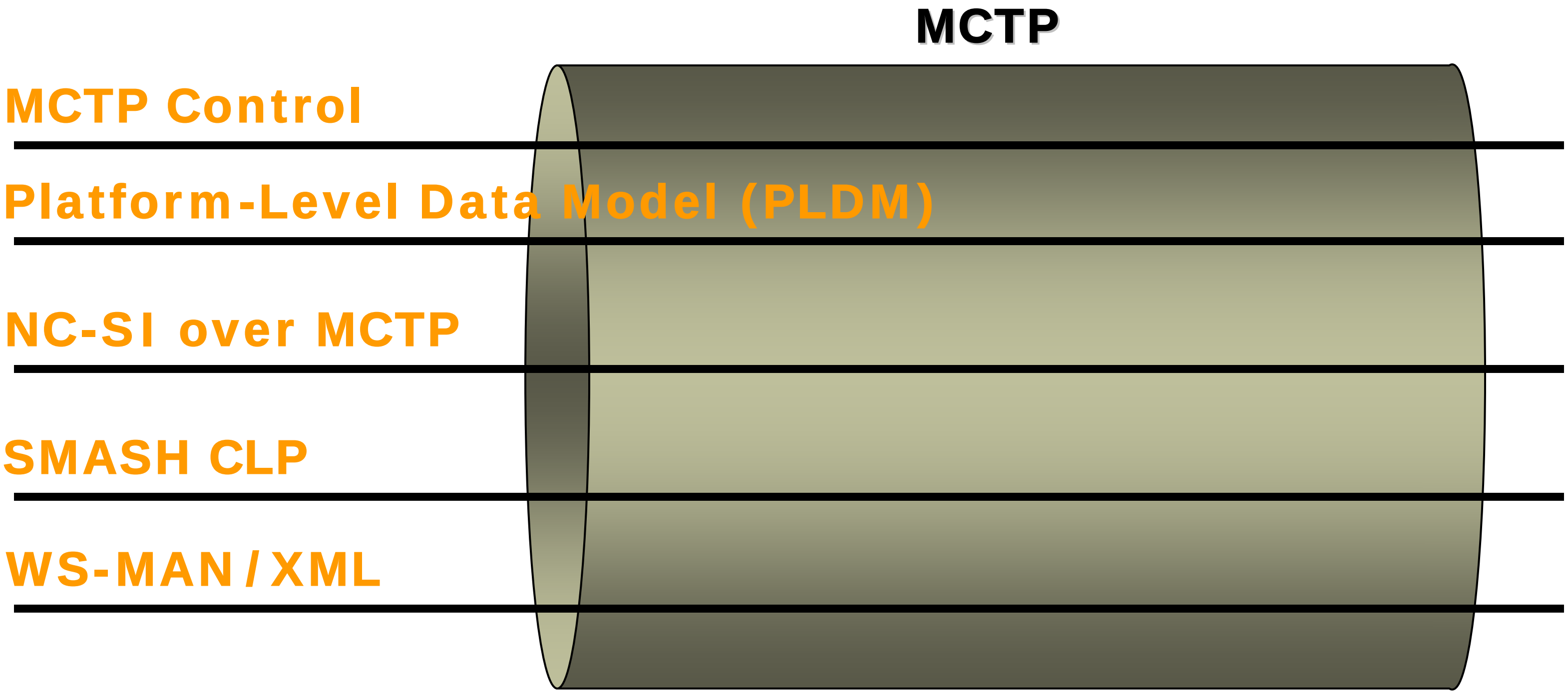
Platform-level Data Model

Platform-level Data Model to CIM Mapping

System FW / SW to MC Interfaces

- **Interfaces:** types of interconnects defined for platform management
- **Transport Protocol:** how mgmt data is moved across the interfaces
- **Platform-level Data Model:** how low level platform management hardware monitoring and control functions are abstracted and accessed
- **PLDM to CIM Mapping:** how the low-level data model is used under the CIM Profiles

MCTP Overview



MCTP

MCTP Control

Platform-Level Data Model (PLDM)

NC-SI over MCTP

SMASH CLP

WS-MAN / XML

MCTP Supports Multiple Message Types over a Common Transport

MCTP Overview

Goal:

Capture and integrate learning's from other management bus protocols

e.g. IPMB, ICMB, SMBus, ATCA-IPMB, ASF

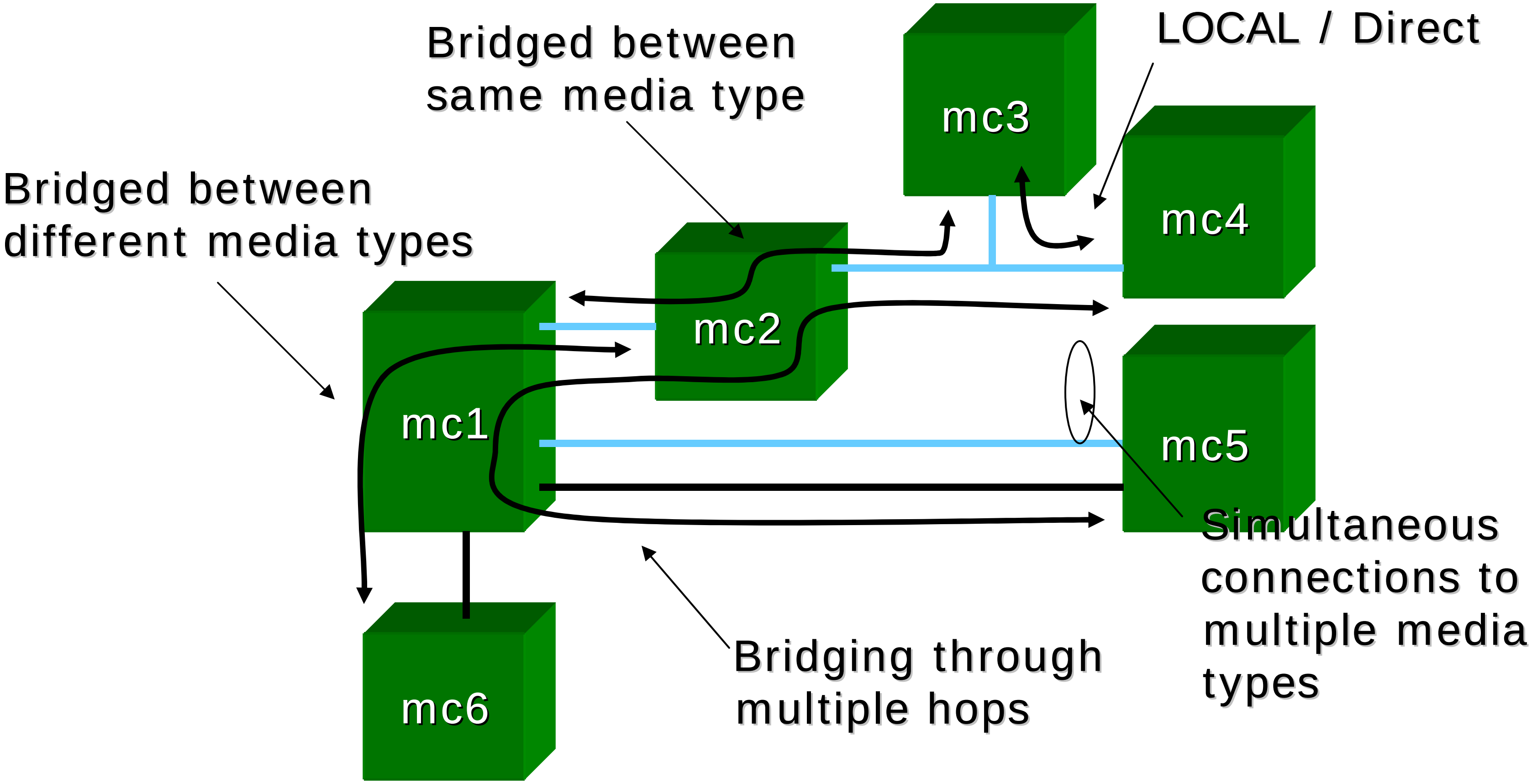
- ability to handle multiple message types
- improved addressing
- simpler routing
- better discovery
- hot plug support

MCTP Overview

- flexible and scalable
  - packet based transfers suitable for multiple types of media
  - message assembly/disassembly reduces buffer size requirements for simple devices & protocols while supporting larger messages for more sophisticated protocols and larger data transfers
- compact, low overhead
  - addressing, integrity checks and header sizes optimized to fit needs of internal management communication. E.g. only ~10 bytes overhead for MCTP on SMBus versus 20 bytes or more for IP.
- optimized for internal communication
  - takes advantage of internal communication characteristics to simplify the protocol, increase efficiency, and reduce firmware overhead
  - E.g. no need for temporal re-ordering, windowed protocols, nor per packet acknowledge.
- supports fixed address / simple devices

Optimized for platform management communication

MCTP Overview

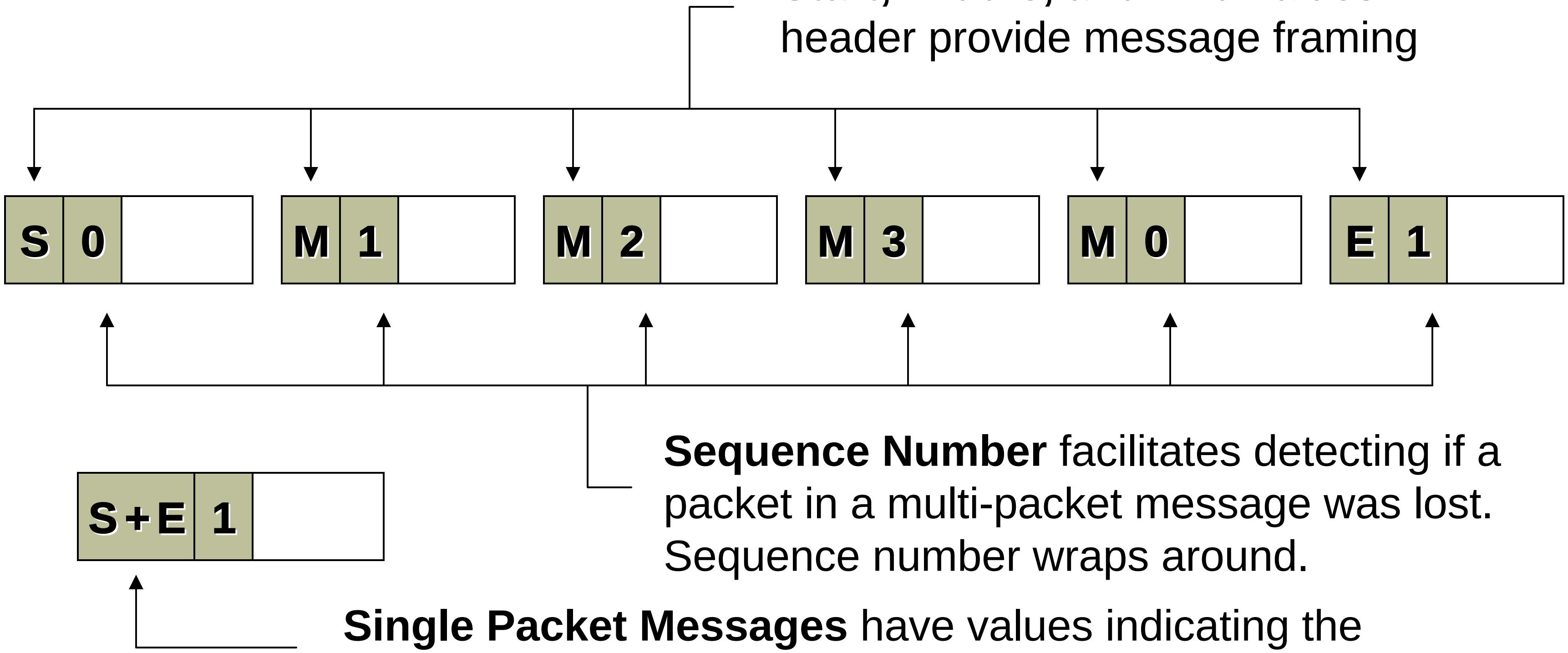


Supports routing across multiple bus segments and between different media types

Message Assembly

An MCTP Message can be formed of multiple MCTP packets.

- Bits in packet identify start, middle, and end packets for a given message.



Start, Middle, and End values in header provide message framing

Sequence Number facilitates detecting if a packet in a multi-packet message was lost. Sequence number wraps around.

Single Packet Messages have values indicating the packet is both a START packet and an END packet.

The Source Endpoint ID, Msg Tag, and TO bit fields uniquely identify the packets for a given message

MANAGEMENT DEVELOPERS CONFERENCE

Logical Addressing

MCTP uses Logical Addressing between communication *Endpoints*

- Called the “Endpoint ID” (EID)
  - Similar to an IP Address
- Provides Media-independent addressing
  - Enables devices on different media to intercommunicate without having to deal with physical address format conversions

Bus Owners and Bridges

An MCTP Bus Owner

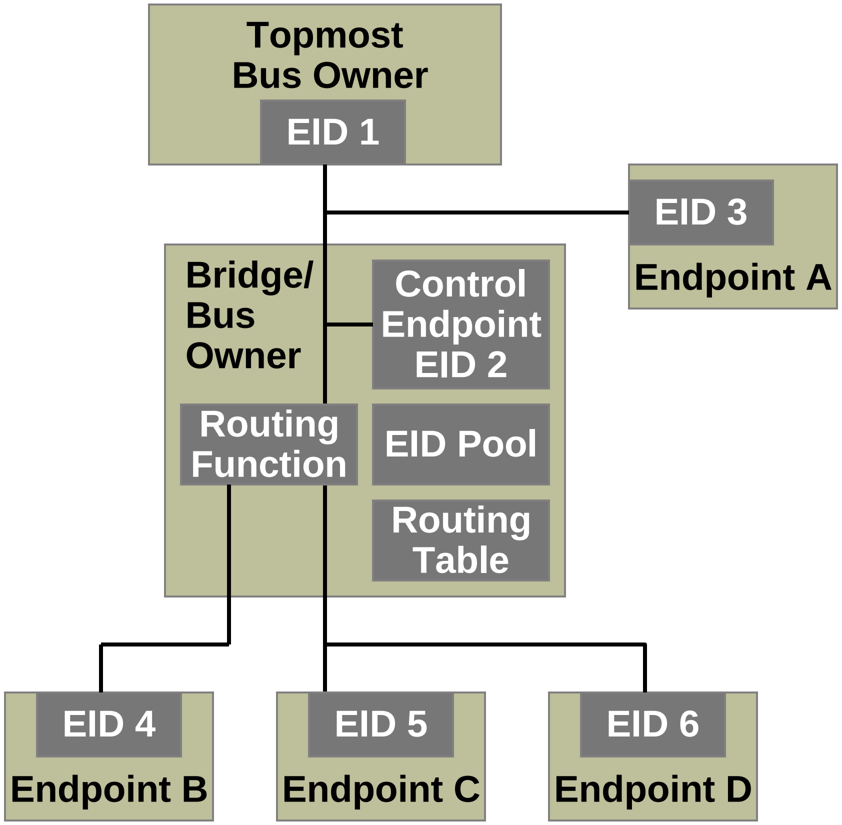
- Assigns EIDs to devices on busses it directly ‘owns’
- Allocates EID pools to MCTP Bridges

The “Topmost Bus Owner”

- Is a bus owner that is the source of all EIDs used in an MCTP bus hierarchy

An MCTP Bridge

- Is responsible for routing MCTP packets between two or more busses
- An MCTP Bridge is the bus owner of at least one bus
- Bridges are responsible for assigning EIDs and allocating EID pools to any devices on busses that the Bridge owns.



Bridging
Routing Tables

- Each MCTP Bridge maintains a logical routing table
  - For devices that are on a bus that is directly connected to the bridge the table contains entries that are used to associate an Endpoint ID with the corresponding physical address and bus ID for the Endpoint
  - For other bridges that are on a bus that is directly connected to the bridge the tables contain entries that associate a *range* of Endpoint IDs with the physical address and bus ID for the other bridge.

- Each Routing Table entry associates an Endpoint ID, or a range of Endpoint IDs to a single physical address and bus ID.

- It is possible to have more than one entry for a given Endpoint ID.

EID Range	Bus ID	Bus Address

Basic Routing Table Entry fields

Bridging

When a Bridge receives an MCTP Packet:

- It checks to see whether the Destination Endpoint ID in the packet matches or falls within the range of Endpoint IDs in the table
  - If the Endpoint ID is for the Bridge itself, the Bridge internally consumes the packet
- Otherwise, if there's a match:
  - The Bridge changes the physical addresses in the packet and reformats the medium -specific header and trailer fields as needed for the target bus:
    - The destination physical address is changed to the to the physical address obtained from the entry in the routing table
    - The Bridge substitutes the source physical in the packet it received with the Bridge's own physical address on the target bus
      - This is necessary to enable messages to be routed back to the originator
  - Packet-specific Data Integrity fields are updated as required
- If there's no match, Packets with out-of-range Endpoint IDs are silently discarded

Bridging and Endpoints

- Most simple MCTP Endpoints will never need to know about Bridges
  - Typically, another Endpoint will initiate communication with them
  - Receiving Endpoints extract the physical address and Source EID from the incoming message use it to format a message back to the originator

- MCTP has no general support for broadcast to all nodes in a given MCTP network
  - Eliminates the need to deal with issues of handling broadcasts through bridges and across different media
  - Avoids definition of special broadcast addresses for media such as SMBus
  - Eliminates need for devices to support additional decode for a special broadcast address

Bridging and Endpoints

- An Endpoint that originates sending a message to another MCTP Endpoint does need to know what physical address to send the message to
  - This could be the physical address of a bridge
  - Or the direct physical address of the target device if the target device is on the same bus as the originator

- An Endpoint gets the physical address for accessing a given EID by sending a *Resolve Endpoint ID* command to the Bus Owner.
  - The *Resolve Endpoint ID* command requests a bus owner to return the physical address that is to be used to route packets to a given Endpoint ID. (This is essentially the MCTP equivalent of ARP)
    - The address will either be the actual physical address for the device implementing the Endpoint
    - Or it will be the physical address for bridge to be used to route packets to the desired Endpoint

- How does the Endpoint know the address for the Bus Owner?
  - An Endpoint knows the physical address of the bus owner either a-priori, or because it saved that information when it got an Endpoint ID assigned to it.

MCTP Common Fields

- Source Endpoint ID - Logical Address of the message originator
- Destination Endpoint ID - Logical Address of the message target
- SOM flag - Identifies first packet in message
- EOM flag - Identifies last packet in message
  (SOM & EOM = 0 indicates a ‘middle’ packet)
- Seq # - Sequence number. (Used to detect if there are missing packets in a message)
- Hdr Version - Identifies MCTP transport packet format and fields for a given medium
- Msg Tag - Identifies packets belonging to a particular message
- TO bit - “Tag Owner bit” - Identifies whether an endpoint is the originator of the Msg Tag value or is using a Msg Tag value given to it by another endpoint.
  (for MCTP Control messages, this is used to also differentiate requests & responses)
- Message Type - Identifies the payload format used for higher level protocols
  - only in the header for the first packet
- IC bit - Integrity Check bit. Indicates whether an optional overall message integrity check field is present.

SMBus Packet Format

<table>
<tr><th></th><th colspan="8">+0</th><th colspan="8">+1</th><th colspan="8">+2</th><th colspan="8">+3</th></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Byte 1 &gt;</td><td colspan="7">Destination Slave Address</td><td>0</td><td colspan="8">Command Code = MCTP = 0Fh</td><td colspan="8">Byte Count</td><td colspan="7">Source Slave Address</td><td>1</td></tr>
<tr><td>Byte 5 &gt;</td><td colspan="4">MCTP Reserved</td><td colspan="4">Hdr Version</td><td colspan="8">Destination Endpoint ID</td><td colspan="8">Source Endpoint ID</td><td>SOM</td><td>EOM</td><td colspan="2">Pkt Seq #</td><td>TO</td><td colspan="3">Msg Tag</td></tr>
<tr><td>Byte 9 &gt;</td><td>IC</td><td colspan="7">Msg Type</td><td colspan="8">Message Header</td><td colspan="8">Message Data</td><td colspan="8">Message Integrity Check</td></tr>
<tr><td>Byte N &gt;</td><td colspan="8">PEC</td><td colspan="24"></td></tr>
</table>

MCTP Message Header (Varies based on Message Type)

Message Type byte (only required in first packet header of message.)

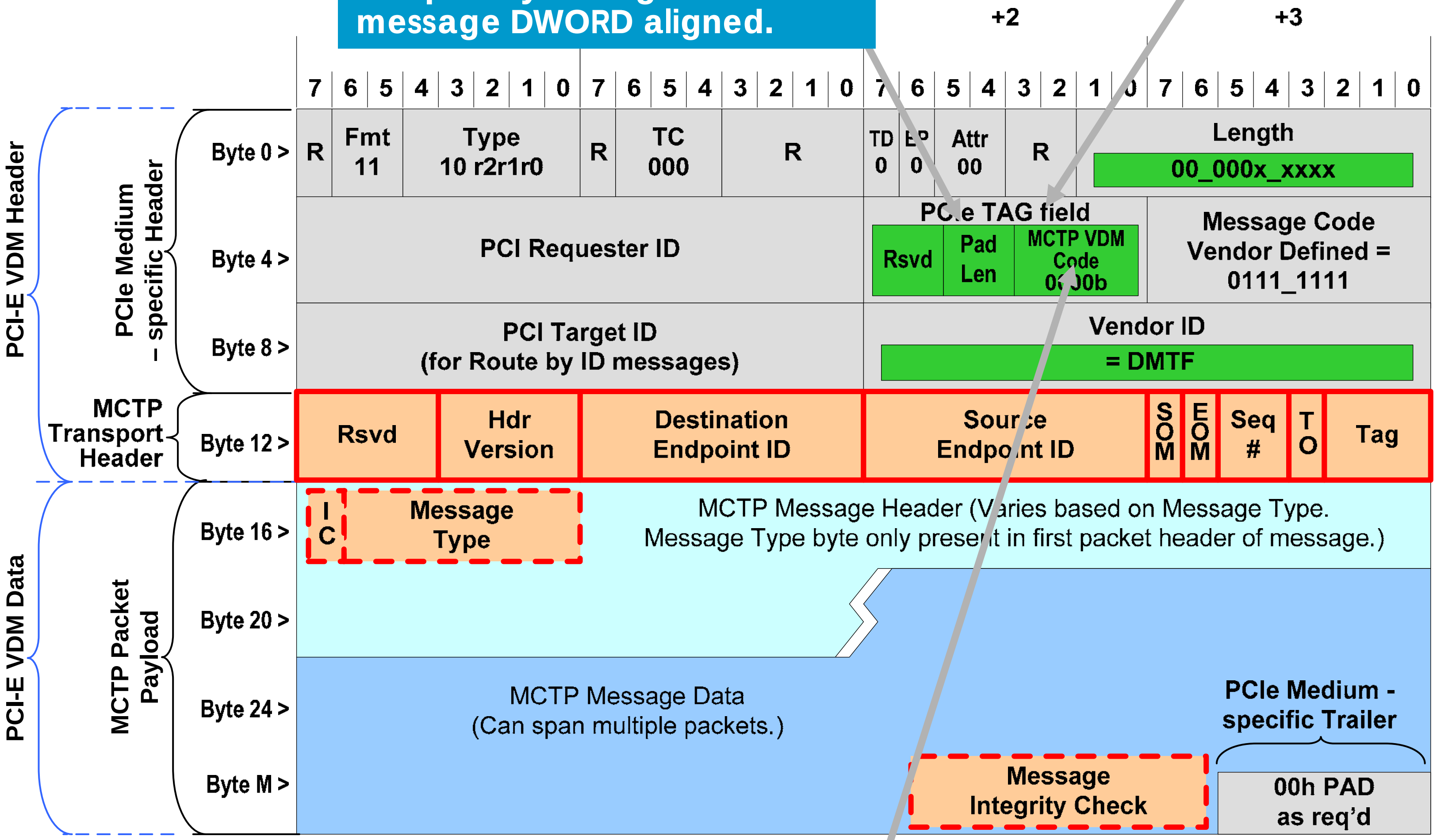
🟧 = common fields for all MCTP messages

Copyright DMTF 2007

PCIe VDM Packet Format

Pad Length - (2-bits) indicates # of pad bytes to get PCIe message DWORD aligned.

MCTP defined usage of PCIe TAG field



+2 +3

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

PCI-E VDM Header

PCIe Medium – specific Header

Byte 0 > R | Fmt 11 | Type 10 r2r1r0 | R | TC 000 | R | TD 0 | EP 0 | Attr 00 | R | Length 00_000x_xxxx

Byte 4 > PCI Requester ID | PCIe TAG field: Rsvd | Pad Len | MCTP VDM Code 0000b | Message Code Vendor Defined = 0111_1111

Byte 8 > PCI Target ID (for Route by ID messages) | Vendor ID = DMTF

MCTP Transport Header

Byte 12 > Rsvd | Hdr Version | Destination Endpoint ID | Source Endpoint ID | SOM | EOM | Seq # | TO | Tag

PCI-E VDM Data

MCTP Packet Payload

Byte 16 > IC | Message Type | MCTP Message Header (Varies based on Message Type. Message Type byte only present in first packet header of message.)

Byte 20 >

Byte 24 > MCTP Message Data (Can span multiple packets.)

Byte M > Message Integrity Check | PCIe Medium - specific Trailer: 00h PAD as req'd

MCTP VDM Code uniquely identifies MCTP VDMs from other possible VDMs that may be defined under the DMTF Vendor ID

Physical Addressing & Packet Data Integrity

- Physical Addressing is specific to given medium
  - E.g. 7-bit address for SMBus, Bus-Device-Function for PCIe, etc.
- MCTP does not define an overall message data integrity check
  - Covered by per-packet data integrity, specific to medium
    - E.g. SMBus 8-bit PEC (packet error check code CRC)
  - Overall message integrity can be added at higher level protocols if needed
    - MCTP does have a provision for a common location for an overall message integrity check, identified using the “IC” bit.
- MCTP Error Detection and retries are handled on MESSAGE basis.
  - MCTP does not define packet -level retries
  - Packet level retries can occur at physical level as required by medium, e.g. SMBus loss of arbitration.

MCTP Control Messages

MCTP Control Messages are for the initialization and support of MCTP communications, including:

- Endpoint Discovery

  Prepare for Endpoint Discovery, Endpoint Discovery, Discovery Notify, Get Endpoint UUID

- EID assignment and EID Pool allocation

  Set Endpoint ID, Get Endpoint ID, Allocate Endpoint IDs

- EID to physical address resolution

  Resolve Endpoint ID

- Bridge Routing Table initialization and updates

  Routing Information Update

- MCTP Network Topology and Path Transmission Unit discovery

  Get Routing Table Entries, Query Hop

- MCTP Message Type and Vendor Defined Message Support

  Get MCTP Version Support, Get Message Type Support, Get Vendor Defined Message Support

MCTP Control Message Protocol

- Used for transfer of MCTP Control Messages
- Built on top of MCTP packet transport specs
- Request / Response protocol with message level retry support
  - Instance ID field supports message retries
- Datagram supported for unacknowledged ‘alert’ messages
- Control Messages fit in single Baseline MTU sized MCTP packets
  - No need for assembly / disassembly

MCTP Specifications

Common Specifications:

DSP0236, Management Component Transport Protocol (MCTP) Base Specification

- Describes MCTP packet routing, message assembly/disassembly, and initialization
- Defines the MCTP base protocol, MCTP Control Message protocol, and the MCTP Control Commands.

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

- Central collection of IDs and codes used across the different MCTP specifications.

Transport Binding Specifications:

DSP0237, Management Component Transport Protocol (MCTP) SMBus / I2C Transport Binding Specification

- Defines how MCTP is carried using SMBus or I2C as the transport medium.

DSP0238, Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification

- Defines how MCTP is carried using PCI Express™ Vendor Defined Messages (VDMs) as the transport medium.

Summary

- designed to carry multiple message types over a single physical connection
- flexible and scalable
  - packet based transfers and message assembly/disassembly enable tailoring transfers to the payload needs of different message types.
- compact, low overhead
  - addressing, integrity checks and header sizes optimized to fit needs of internal