Document Identifier: DSP2054

Date: 2019-12-18

Version: 1.0.0

PLDM NIC Modeling

Supersedes: None

Document Class: Normative

Document Status: Published

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Platform Level Data Model (PLDM) NIC Modeling Specification* (DSP2054 was prepared by the Platform Management Components Intercommunications (PMCI) of the DMTF.

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

Acknowledgments

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

Editor:

- Yuval Itkin – Mellanox Technologies

Contributors:

- Balaji Natrajan – Microchip Technology Inc.
- Bill Scherer – Hewlett Packard Enterprise
- Bob Stevens – Dell
- Brett Scrivner - Lenovo
- Dov Goldstein – Intel Corporation
- Edward Newman - Hewlett Packard Enterprise
- Eliel Louzoun – Intel Corporation
- James Smart – Broadcom Inc.
- Hemal Shah – Broadcom Inc.
- Ira Kalman – Intel Corporation
- Jeffrey Plank – Microchip Technology Inc.
- Kaijie Guo – Lenovo
- Patrick Caporale – Lenovo
- Patrick Schoeller – Hewlett Packard Enterprise
- Richelle Ahlvers – Broadcom Inc.
- Scott Dunham – Lenovo

Introduction

The *Platform Level Data Model (PLDM) NIC Modeling document* defines the PLDM data structures for modeling a NIC using PLDM for Monitoring and Control semantics. Additional information related to modeling configuration options for NICs are also defined.

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.

PLDM NIC Modeling

1 Scope

This document defines messages and data structures for modeling a NIC using PLDM for Monitoring and Control semantics. NIC modeling allows implementers of NIC and MC to better understand how to use PLDM for Monitoring and Control in a real system. Implementers using the model described in this document can assure interoperability at the system level. The model also provides for scalability in terms of the number of controllers, ports and connectors in the given NIC hardware. For model simplicity, entity-types are fabric-agnostic, and simplicity over accuracy is preferred where possible.

This specification is not a system-level requirements document. The modeling and messages which are stated in this document are implemented through PLDM messaging using PLDM for Platform Monitoring and Control semantics. PLDM NIC Modeling does not specify whether a given NIC is required to implement every property included in the model. For example, this model does not specify whether a given NIC shall support PLDM for Platform Monitoring and Control. However, implementing PLDM NIC Modeling per this document requires using messages and data model structures defined in PLDM for Platform Monitoring and Control.

Portions of this reference model specification rely on information and definitions from other specifications, which are identified in clause 2. Five of these references are particularly relevant:

- DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification*, provides definitions of common terminology, conventions, and notations used across the different PLDM specifications as well as the general operation of the PLDM messaging protocol and message format.
- DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification*, defines the values that are used to represent different type codes defined for PLDM messages.
- DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification,* defines the messages and data structures for discovering, describing, initializing, and accessing sensors and effecters within the management controllers and management devices of a platform management subsystem
- DMTF DSP0249, *Platform Level Data Model (PLDM) State Set Specification,* defines the collection of state sets, each having a set of enumeration values. PLDM for Monitoring and Control uses the state set to report the discrete values from PLDM sensors.
- DMTF DSP0257, *Platform Level Data Model (PLDM) FRU Data Specification 1.0,* defines a FRU data format that provides platform asset information including part number, serial number and manufacturer.

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.

ANSI/IEEE Standard 754-1985, *Standard for Binary Floating Point Arithmetic*

DMTF DSP0236, *MCTP Base Specification 1.2*, http://dmtf.org/sites/default/files/standards/documents/DSP0236_1.2.pdf

DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification 1.0*, http://dmtf.org/sites/default/files/standards/documents/DSP0240_1.0.pdf

DMTF DSP0241, *Platform Level Data Model (PLDM) Over MCTP Binding Specification 1.0*, http://dmtf.org/sites/default/files/standards/documents/DSP0241_1.0.pdf

DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification 1.2*, http://dmtf.org/sites/default/files/standards/documents/DSP0245_1.2.pdf

DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification 1.1*, http://dmtf.org/sites/default/files/standards/documents/DSP0248_1.1.pdf

DMTF DSP0249, *Platform Level Data Model (PLDM) State Sets Specification 1.0,* http://dmtf.org/sites/default/files/standards/documents/DSP0249_1.0.pdf

DMTF DSP0257, *Platform Level Data Model (PLDM) FRU Data Specification 1.0,* http://dmtf.org/sites/default/files/standards/documents/DSP0257_1.0.pdf

DMTF DSP0267, *Platform Level Data Model (PLDM) for Firmware Update Specification 1.0,* http://dmtf.org/sites/default/files/standards/documents/DSP0267_1.0.pdf

IETF RFC2781, *UTF-16, an encoding of ISO 10646*, February 2000, http://www.ietf.org/rfc/rfc2781.txt

IETF STD63, *UTF-8, a transformation format of ISO 10646* http://www.ietf.org/rfc/std/std63.txt

IETF RFC4122, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005, http://www.ietf.org/rfc/rfc4122.txt

IETF RFC4646, *Tags for Identifying Languages*, September 2006, http://www.ietf.org/rfc/rfc4646.txt

ISO 8859-1, *Final Text of DIS 8859-1, 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No.1,* February 1998

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards,* http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

SFF Committee Management Interface for Cabled Environments SFF-8636, https://www.snia.org/technology-communities/sff/specifications

SFF Committee Diagnostic Monitoring Interface for Optical Transceivers SFF-8472, https://www.snia.org/technology-communities/sff/specifications

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 DSP0240 for terms and definitions that are used across the PLDM specifications. For the purposes of this document, the following additional terms and definitions apply.

3.1

Cable

one of: Active copper, Passive-Copper, Optical fiber of an AOC, optical fiber connected to an AOC module

3.2

Break-out Cable

a set of physical cables which are connected to the same connector. Breakout cable is a physical cable type.

3.3

Communication channel

a logical representation of a networking connection path that conveys information between physical entities as described in 6.6.5.

3.4

Connector

a physical element which is part of the NIC. A pluggable Module is connected to the NIC by a physical connection to the connector.

3.5

Interconnect

a physical connection between a pluggable module and a connector on the NIC

3.6

NIC

Network Interface Card (NIC). A NIC is an entity in a system that provides network connectivity to the system. The network can be of any type, such as Ethernet, Fibre-Channel, InfiniBand or any other type.

3.7

Pluggable Module

a module which is plugged into the NIC network connection connector. Pluggable modules may be integrated with a cable as one unit or may be separate elements. A pluggable module can be an active device with embedded active-components, or it can be a passive device with none. The type of a pluggable module depends on the type of the physical connector for which it is designed.

3.8

LOM

LAN-On-Motherboard, a NIC which is embedded on the motherboard.

3.9

Network Controller

an active device which includes the equivalent of MAC and PHY of the specific network connection, this device typically connects to a host CPU over a bus such as PCIe

3.10

Network Port

a physical interface on a network controller, used to convey network-communication. The type of a network port depends on the type of the communication network to which it is connected.

3.11

PHY

an electronic circuit, usually implemented as a chip, required to implement physical layer interface function.

3.12

Record Handle

an opaque numeric value used to access individual PDR within the PDRs repository.

3.13

TID

Terminus ID as defined in DSP0240.

4 Symbols and abbreviated terms

Refer to DSP0240 for symbols and abbreviated terms that are used across the PLDM specifications. For the purposes of this document, the following additional symbols and abbreviated terms apply.

4.1

NIC

Network Interface Card

4.2

LOM

LAN On Motherboard

4.3

PHY

Physical layer interface

5 Conventions

Refer to DSP0240 for conventions, notations, and data types that are used across the PLDM specifications.

5.1 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.

5.2 Byte ordering

Unless otherwise specified, as for all PLDM specifications byte ordering of multibyte numeric fields or multibyte bit fields is "Little Endian" (that is, the lowest byte offset holds the least significant byte, and higher offsets hold the more significant bytes).

6 PLDM NIC Modeling overview

This document describes a modeling scheme for a NIC using PLDM for Monitoring and Control DSP0248 semantics. The model is scalable, allowing consistent modeling of NICs with different configuration options such as the number of network-controllers, number of ports, and number of connectors. PLDM NIC Modeling supports different types of networks, including devices supporting multiple network-types concurrently.

While PLDM for Platform Monitoring and Control is a public standard, using the model as defined in this document simplifies interoperability by establishing a consistent schema. The model is also intended to serve as a template for modeling other system hardware elements.

The basic format that is used for sending PLDM messages is defined in DSP0240. The format that is used for carrying PLDM messages over a transport-layer protocol or medium is given in companion documents to the base specification. For example, DSP0241 defines how PLDM messages are formatted and sent using MCTP as the transport. PLDM NIC Modeling defines the data structures and their relations which together describe a given NIC hardware configuration and state.

The model supports the following:

- Consistent modeling of a NIC regardless of the specific configuration and resources count
- NIC hardware structure description
- Defining the group of resources used to form a network connection
- Associating a network connection to a specific controller and cable
- Representing any type of physical connection, including cables, break-out cables and backplane connections
- Reporting of configuration changes

Unlike static systems, a NIC use external connections. For that reason, the same NIC can operate in different settings depending on the combination of NIC hardware and connected network cable. This dynamism requires dynamic modeling capability. For NIC hardware that supports pluggable modules, the model reflects both the NIC hardware as well as any connected pluggable modules. A NIC may support a backplane-connection; in this case, no pluggable module exists. The model equally supports these different hardware configurations.

The model is hierarchical, with each subgroup including elements grouped to form a physical element.

6.1 Model elements

6.1.1 Terminus Locator(s)

PLDM for Platform Monitoring and Control defines a single root for every model, referred to as Terminus Locator.

In a typical implementation of PLDM for Platform Monitoring and Control, the network controller is the active component which communicates with the MC. The network controller is therefore serving as a terminus locator. When there are multiple Network controllers assembled on the same card, there is no single device which reports all the sensors of all the elements in the system to the MC.

PLDM for Platform Monitoring and Control does not allow associating components reported via different TIDs since every database is relative to a given TID. To overcome this constraint, the standard method allowing the MC to correctly associate multiple TIDs to the same NIC hardware requires the use of PLDM for FRU (DSP0267). When the MC reads multiple TIDs and observes the same board part number and serial number and thus the same globally unique ID, it can recognize these TIDs as belonging to the same card.

All PLDM model IDs used in a given card shall be consistent across all TIDs. This avoids conflict from duplication of IDs in the combined model, generated by merging the TID-specific model elements reported as part of the overall model.

6.1.2 NIC

In this model, the NIC is the top-level element of the hierarchy.

When modelling a LOM (LAN On Motherboard) instead of a NIC, instead of being part of the system level, the NIC model will be defined as part of the system main board (Type-ID 64 in DSP0249). In this case, a NIC will not be a stand-alone card (Type-ID 68) but will rather be declared as a module (Type-ID 62) which is part of the motherboard.

6.1.3 Network controller

The network controller is an active component which performs the networking control function of either MAC and PHY layers or only the MAC layer. A network controller always includes at least one network port.

A controller contains sensors for its health state, power-consumption, and temperature. The temperature of a network controller can be reported by one or more temperature sensors typically located in thermally sensitive areas on the card. In addition, state sensors for each of the MAC elements is monitored for link state, link speed, and link type.

Network controllers with more than one network interface port are modeled with a separate set of sensors for each port. In this case each port will be monitored independently through its set of sensors.,

The first network controller in a NIC reports all NIC level sensors under its terminus ID.

6.1.4 Connector

The connector is a physical component into which a cable or a pluggable module may be attached. In a typical use case, the connector is accessible through the system front or rear panel to allow the connection of a pluggable module. A connector is only included in the model of a NIC that is using that connector. Therefore connector is included in the model only when the network is physically connected via a Pluggable module or a cable. When using a backplane connection there is no connector in the model.

6.1.5 Pluggable module

A pluggable module is the element which is plugged into the NIC network connector. Pluggable modules and the cables connected to them may be modeled as a single compound unit or be composed of separate elements. A pluggable module can be active or passive. When there is a pluggable module, the presence of the module is reported in the model via a state sensor. When active, supporting pluggable module reports, the power envelope and temperature of the module.

6.1.6 Cable

A cable is a passive element used to connect the network signal from a pluggable module or connector to the network. A cable can be electrical (such as copper) or optical (such as fiber-optic). Cables do not typically include any sensors and do not have presence indication; therefore, their state cannot be reported by any sensor. For this reason, when using a passive cable, such as RJ45, connected without a pluggable module, there is no way to report the cable presence, health, or temperature. Some DSP based PHY devices may sense a cable presence allowing to report the presence state of a cable indirectly.

6.1.7 Break-out cable

A break-out cable is a group of network cables connected to the same pluggable module at one end with the other end of each cable is connected to a potentially separate pluggable module. When break-out cable is used, the model includes multiple cables which are all connected through the same pluggable module. When using a break-out cable, multiple communication channels are associated with the same break-out cable. Each of these channels is assumed to use a separate cable within the break-out cable.

6.1.8 Backplane connection

A backplane connection refers to a network connection that does not use a pluggable module or any physical cables. When using a backplane connection, the network connection signals are carried through a connector on the NIC to the system. When a backplane connection is used, there is no associated cable and there is no other sensor to reflect the physical connection state. As there is no additional monitor and control information in the connection to the backplane, there is no need to reflect this connector in the model.

6.2 Model sensors

Attributes are reported by means of sensors. Numeric sensors are used to report specific measured attribute. State sensors report operational and/or health state.

6.2.1 NIC temperature sensor

Temperature sensors in the NIC reports the card’s physical temperature. There may be multiple temperature sensors installed on the PCB.

The temperature sensor is a numeric sensor. It is not included in the NIC container PDR as sensors are defined by directly referencing the entity being measured.

6.2.2 NIC power sensor

The power sensor in the NIC reports the estimated or measured aggregate power consumption of all the different elements included in the model. This includes mainly the network controller and the pluggable modules power. A NIC which cannot accurately report its real-time power shall report its expected maximal power at the respective operating mode. When there are multiple network controllers on the same NIC, there may be no visibility for any network controller to the real-time information of the other network controllers. For this reason, this sensor is only available when there is only one network controller in the NIC, or when there is a hardware sensor which does allow measuring and reporting the total power consumption. Note that network controllers which cannot report real-time information may report the expected maximal power for the operating mode in use.

6.2.3 NIC FAN speed sensor

The NIC FAN speed sensor reports the speed of an active cooling FAN. A NIC may have multiple FANs installed on the PCB, each with its own speed sensor. The thresholds reported for this numeric sensor shall be set by the hardware vendor.

6.2.4 NIC composite state sensor

A composite state sensor is used to report the NIC thermal state, configuration state, and aggregate health state of all the components included in the reported database. The reported aggregate health state reflects the worst of the reported health states for each one of the elements monitored in the model.

When there are multiple network controllers, there may be no visibility from any network controller to the real-time information of other network controllers. For this reason, the composite state sensor is only available when there is only a single network controller in the NIC or when the reporting network controller has the needed visibility.

The configuration state reported in this sensor relates to the change of pluggable modules or to the network controller device. When a pluggable module is inserted or removed, the card configuration changes.

The NIC thermal state sensor, NIC configuration state sensor, and the NIC health state sensor are collected into the NIC composite state sensor.

6.2.5 Network controller temperature sensor

The temperature sensor of the network controller reflects the device temperature at a physical location. The thresholds used by the sensor to define its normal, warning, critical, and fatal ranges are design specific and should be defined by the device manufacturer.

6.2.6 Network controller power sensor

The network controller power sensor reflects the present value of the device power consumption. The thresholds which may be used by the sensor to define its normal, warning, critical, and fatal ranges are design specific and should be defined by the device manufacture.

Note that network controllers that cannot report real-time information may report the expected maximal power for the operating mode in use.

6.2.7 Network controller composite state sensor

The network controller’s composite state sensor reports the operational state of the network controller. The use of composite state sensor allows combining multiple metrics into a single sensor with a complete view of the operational and health state of the controller. The MC can use this sensor to identify issues with the controller and to identify the specific maintenance operations that need to perform. These operations may include network controller reset, system-level shut-down for thermal protection, and other system-level maintenance.

Using the configuration change indication, the network controller notifies the MC to retrieve PDRs updated by the configuration change.

When FW Update is detected, the composite state sensor can reflect this event to the MC, allowing the MC to take any action needed to respond to the update. Note that reading the new FW version shall be performed by the MC using protocols other than PLDM for Platform Monitoring and Control, such as DSP0257 and/or DSP0267. Please note that FW update only reflects the conclusion of the FW programming operation; it is device-specific whether this detection additionally implies that new FW is already active.

6.2.8 Network port link speed sensor

The network port may operate at various communication speeds. This numeric sensor is used to report the actual operating link speed.

6.2.9 Network port link state sensor

A state sensor is used to reflect the operational state of the port. The MC uses the attributes reported by this sensor to monitor the state of the port. Possible states for the link are Connected and Disconnected as defined in DSP0249.

6.2.10 Pluggable module temperature sensor

This sensor reflects the pluggable module temperature. The thresholds used to define the thermal operating ranges are read from the module parameters. Note that due to some terminology gaps between SFF and DMTF PLDM for Platform Monitoring and Control, some terms require translation as shown in Table 1**.**

Table 1 – SFF8636 and DSP0248 thresholds definitions

SFF8636 / SFF8472	DSP0248	Description
Warning	Warning	The reading is outside of normal expected operating range but the monitored entity is expected to continue to operate normally.
Alarm	Critical	The reading is outside of supported operating range. Monitored entities might operate abnormally, have transient failures, or propagate errors to other entities under this condition. Prolonged operation under this condition might result in degraded lifetime for the monitored entity.
N/A	Fatal	The reading is outside of rated operating range. Monitored entities might experience permanent failures or cause permanent failures to other entities under this condition.

6.2.11 Pluggable module power sensor

Power reporting for the pluggable module shall use the information from the module itself. As a reference, SFF8636 and SFF8472 defines power classes that can be used to report the expected maximal power consumption of the modules. If there is a module that can report its actual real-time power consumption, this information should be used as it provides more accuracy.

6.2.12 Pluggable module composite state sensor

The composite state sensor within the pluggable module is used to report the overall operational state for the pluggable module. This sensor reports the pluggable module’s presence as well as its temperature operational state and the pluggable module health state.

6.3 Hierarchy description of the NIC model elements

PLDM NIC Modeling uses a hierarchical model. The hierarchy is described using two types of associations as described in the following clauses. Associating entities is done hierarchically, by associating the containing entities rather than associating all the contained entities within that container.

In PLDM modeling, except for the entity that represents an overall system, all entities are contained within at least one other physical entity. Each level within the resulting hierarchy is an individual numeric space.

Identification of the numeric space in which a given element in the hierarchy is declared uses a parameter called the container ID. Container ID is defined as an opaque number that identifies the containing Entity that the Instance number is defined relative to. If this value is 0x0000, then the containing Entity is considered to be the overall system.

An entity association PDR uses 3 references to container IDs:

- containerID – An opaque number that identifies a particular container entity in the hierarchy of containment.
- containerEntityContainerID – a reference to the higher level that contains the declared namespace. The top-level PDR shall always use containerEntityContainerID=0 (System)
- containedEntityContainerID – a reference to the numeric space at which a contained entity is instantiated.

6.3.1 Physical entities association

Physical association is defined in DSP0248 as a method to associate components which are physically connected to each other. The model uses this concept to describe the following structures:

- Content of the NIC PCB
- Content of the network controllers
- Content of a pluggable module, including the associated cable(s) of that module
- Association of a pluggable module with the connector into which it is plugged

A hierarchy entity is defined using an entity association PDR identified with a unique ***containerID*** identifier parameter. The entity association PDR's ***containerEntityContainerID*** references the PDR in which the entity is contained.

Figure 1 shows how a contained entity PDR references its containing entity PDR:

Network Controller Association PDR

Container ID	1000
Record Handle	1150

Container Entity		
Entity Type	144	Network controller
Entity Instance Number	1	
Container Entity Container ID	100	NIC

Association Type	**Physical to Physical containment**

Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	1	
Contained Entity Container ID	1000	Network Controller

Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	2	
Contained Entity Container ID	1000	Network Controller

NIC Entity Association PDR

Container ID	100
Record Handle	1100

Container Entity		
Entity Type	68	Add-In card
Entity Instance Number	1	
Container Entity Container ID	0	System

Association Type	Physical to Physical containment

Contained Entity - Network Controller		
Entity Type	144	Network controller
Entity Instance Number	1	
Contained Entity Container ID	100	NIC

Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	1	
Contained Entity Container ID	100	NIC

Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	2	
Contained Entity Container ID	100	NIC

Figure 1 – Hierarchy description using containerEntityContainerID referencing the containedEntityContainerID

6.3.2 Logical entity association

Logical association is defined in DSP0248 as a method to associate components which collectively form a shared property yet are not physically part of the same component. This model uses logical association to describe the following structures:

- Sharing a MAC, PHY (if on a separate device than the MAC), and cable to form a network connection

Figure 2 shows logical association between a network controller's Ethernet network port and a cable within a pluggable module:

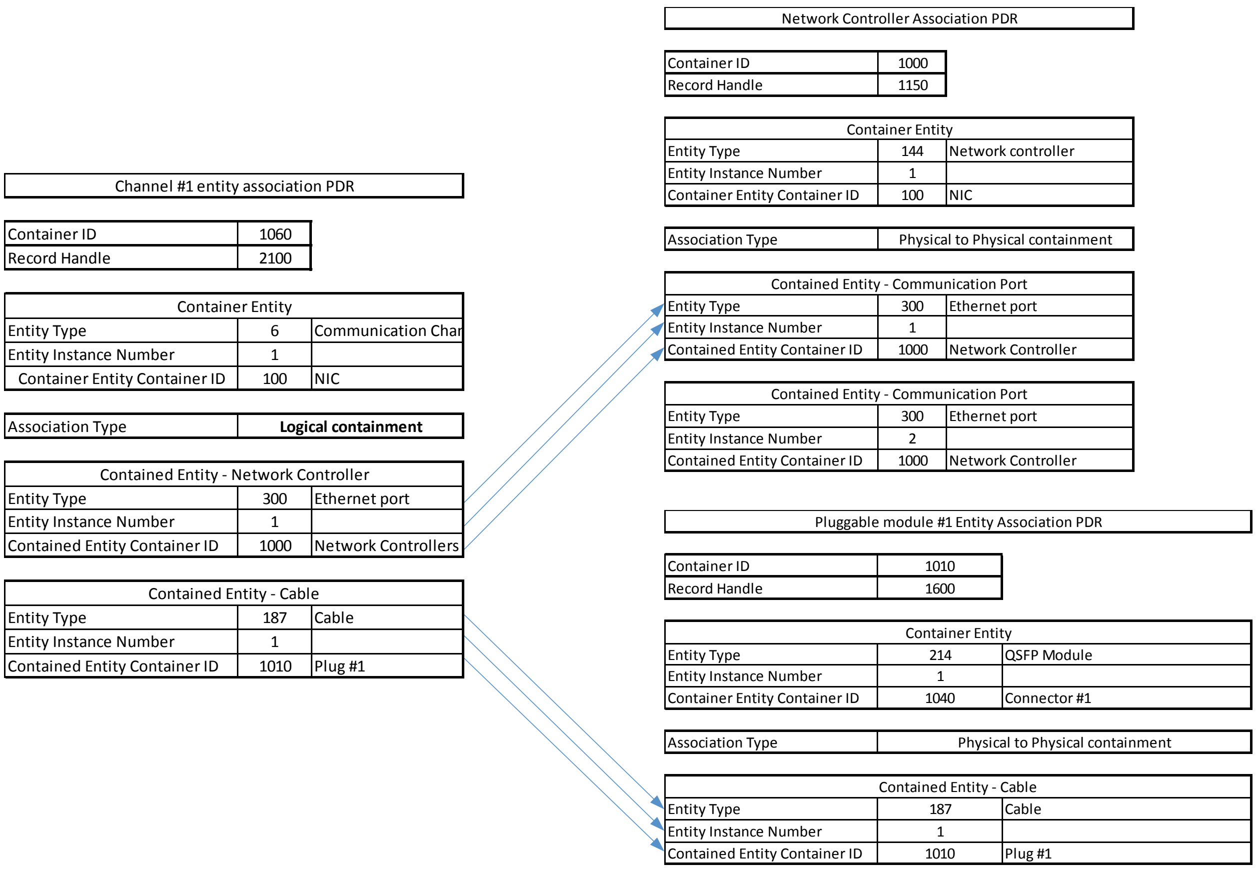

Figure 2 – Defining a communication channel using logical association

6.3.3 Sensors association

Associating a numeric sensor to the measured entity is done by directly referencing the measured entity in an entity association PDR with its containedEntityContainerID, containedEntityType, and containedEntityInstanceNumber. A sensor is identified by a unique Sensor ID value. In PLDM for Platform Monitoring and Control, numeric and state sensors are not included in entity association PDRs.

Figure 3 illustrates the association of a temperature sensor to a network controller in the model:

Network Controller Temperature sensor PDR		
Record Handle	1500	
Sensor ID	300	
Entity Type	144	Network controller
Entity Instance	1	Network Controller Instance #
Container ID	100	NIC
Base Units	2	Degrees C

Network Controller Entity Association PDR		
Container ID	1000	
Record Handle	1150	
Container Entity		
Entity Type	144	Network controller
Entity Instance Number	1	
Container Entity Container ID	100	NIC
Association Type	Physical to Physical containment	
Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	1	
Contained Entity Container ID	1000	Network Controller
Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	2	
Contained Entity Container ID	1000	Network Controller

Figure 3 – Sensor association

6.3.3.1 Associating a sensor at the top level

When associating a sensor to the top-level entity which is the system the association uses the top-level ***containerEntityType containerEntityInstanceNumber*** and **containerEntityContainerID** parameters.

Figure 4 illustrates the association of a temperature sensor to the NIC in the model.

Ambient Temperature sensor PDR		
Record Handle	1130	
Sensor ID	20	
Entity Type	68	Add-In card
Entity Instance	1	NIC Card Instance #
Container ID	0	System
Base Units	2	Degrees C

NIC Entity Association PDR		
Container ID	100	
Record Handle	1100	
Container Entity		
Entity Type	68	Add-In card
Entity Instance Number	1	
Container Entity Container ID	0	System
Association Type	Physical to Physical containment	
Contained Entity - Network Controller		
Entity Type	144	Network controller
Entity Instance Number	1	
Contained Entity Container ID	100	NIC
Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	1	
Contained Entity Container ID	100	NIC
Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	2	
Contained Entity Container ID	100	NIC

Figure 4 – Top-level sensor association

6.4 Element PLDM Type IDs

The model uses the following Type ID for each component in the model, selected from the available types defined in DSP0249. The following table lists the chosen Type IDs used in the model:

Table 2 – Type IDs used in the NIC model

Component	Type ID
Communication channel	6
NIC1[)]1)	68/62
Network controller	144
Connector	185
Cable	187
QSFP Module[1)3), 1)4)]	214
Ethernet port [1)2), 1)5)]	300

Notes:

1) The Type ID for the NIC is 68. If the NIC is a LOM, then Type ID 62 shall be used, as described in 6.1.2.

2) The Type ID for the network controller ports shall match the type of network that is in use. The example in the above table relates to an Ethernet network.

3) The Type ID which identifies the pluggable module type, shall match the actual type of the pluggable module.

4) QSFP is used as an example. For additional types of pluggable modules types see DSP0249

5) Ethernet port is used as an example. For additional types of network port connection types see DSP0249

6.5 Enumeration

PLDM for Monitoring and Control uses enumerated IDs to define elements in the database. These IDs are labeled as:

- Container ID – unique for each container PDR in the model database
- Instance ID – unique for each entity type within a given hierarchy level
- Handle ID – unique ID for each PDR in the model database
- Sensor ID – unique for each sensor in the model database

The proposed model provides an example enumeration scheme for these IDs, allowing a reasonably scalable formulation.

6.5.1 Enumeration scheme

The model assumes some maximal limits to define the enumerated values. These limits where chosen based on industry practice, which restricts the number of network controllers, connectors, and sensors used in the same NIC hardware. These limits are provided as an example and can be adjusted according to the specific NIC requirements.

The example model enumeration is designed to support a NIC that does not exceed the following limits:

Table 3 – Chosen enumeration limits in the model

Model Limit	Value
Max network controllers	10
Max connectors count	20
Max board temperature sensors	10
Max temperature sensors/controller	10
Max temperature sensors/plug	10

Note:

If one of the above limits is insufficient for a NIC, only the enumerated values will be affected; the model structure will not have to change.

Table 4 illustrates the enumeration scheme, calculated based on the above limits.

Table 4 – Example Enumeration Scheme with Type IDs

Item	Max count	Container ID	Container ID	Base Handle	Max Handle	Base Sensor ID	Max Sensor-ID	Base Instance	Max instance	Type-ID
NIC	1	100		1100				1	1	68
Card Composite State Sensor	1			1101	1101	5	5	1	1	68
NIC Power Sensor	1			1102	1102	6	6		1	68
Connectors	20	1040	1059	1110	1129			1	4	185
NIC Temp sensors	10			1130	1139	20	29		10	68
NIC FAN speed sensor	10			1140	1149	30	39		10	68
Network Controllers	10	1000	1009	1150	1159			1	10	144
Network Controller power	1			1160	1169	50	59	1	1	144
Network Controller State	1			1170	1179	60	69	1	1	144
Ports of Network Controller	10			1200	1299			1	2	300
Link speed of network controller	10			1300	1399	100	199		2	300
Port State of network controller	10			1400	1499	200	299		2	300
Temp sensors per network controller	10			1500	1599	300	399		10	144
Plugs	20	1010	1029	1600	1619			1	2	214
Plug Power Sensor	20			1700	1719	400	419			214
Plug Temp sensor	10			1800	1999	500	699			214
Plug composite Sensor	1			2000	2019	700	719	1	1	214
Cable	16							1	16	187
Communication Channel	100	1060	1159	2100	2199			1	100	6

Calculated
Model Constant
NA

6.6 Model illustration

The PLDM NIC model is hierarchical model. The following subclauses describe the model for each of the hierarchy levels:

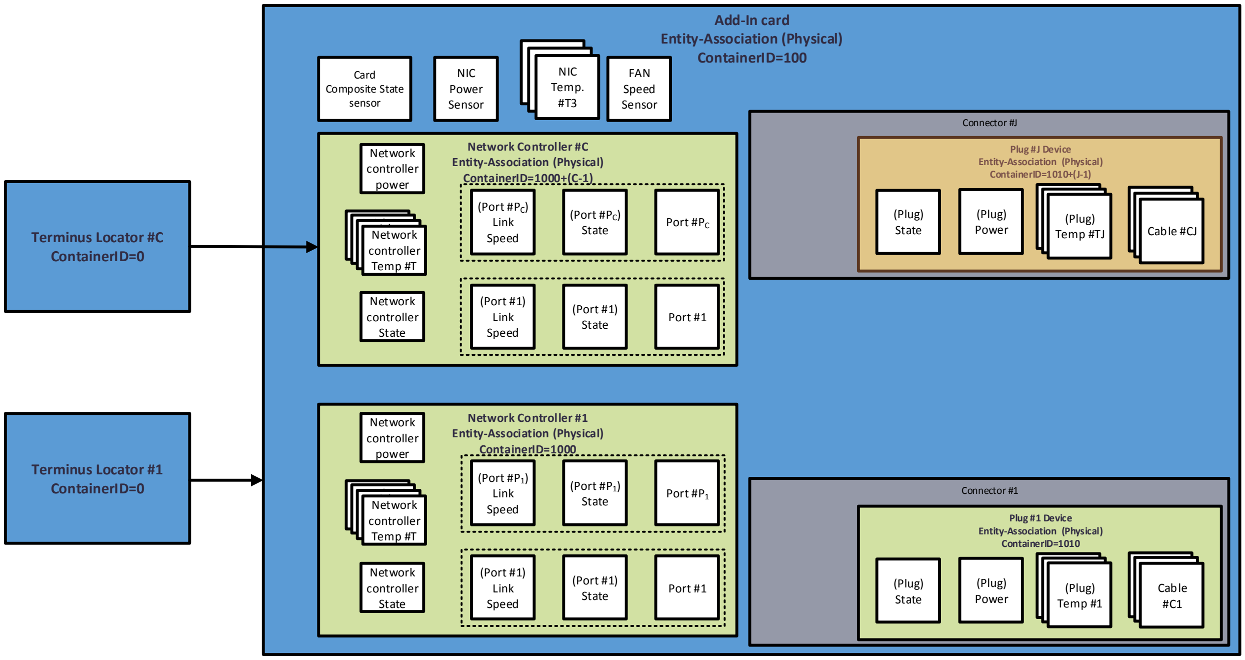

Figure 5 – NIC PLDM model diagram

6.6.1 NIC

The NIC level contains the PCB card, network controllers, connectors, and one or more thermal sensors. The PCB power consumption is represented with a power sensor. The NIC operational state is represented by a composite state sensor. When there are multiple network controllers on the same card, NIC sensors are typically only reported by the first network controller. Note that the top-level health state sensor relates to card level sensors and may not reflect the health states of network controllers beyond the first.

6.6.2 Network controller

The network controller hierarchy represents the active device (or one of multiple devices) that performs the network control interface (such as the MAC and PHY layers). A network controller is represented as a collection of ports and sensors associated with the controller as well as sensors associated with specific network ports. Each port has its own set of sensors.

6.6.3 Pluggable module

Pluggable module is the element attached to the NIC connector that optionally includes the electronics of the network cable. In single link module, a pluggable module is attached to one cable. When a breakout cable is used, the same pluggable module is connected to multiple cables, each carrying an independent network link.

The pluggable module is represented as a set of sensors, which reflect its operational state and power consumption, and cables. Since the pluggable module is not part of the PCB, it may be attached or detached from the NIC dynamically. The model reflects this occurrence with a PLDM configuration change event. Configuration change events can be used to reflect both insertion and removal of a pluggable module.

While a pluggable module is disconnected from the NIC, a query to the pluggable module numeric sensors (power and temperature) shall be responded to with ***sensorOperationalState*** set to ***unavailable*** as defined in DSP0248. Note that when a pluggable module is (re-)inserted into a connector, a configuration change event directs the MC to re-read the PDRs of the new module. This ensures that the MC sees the parameters settings for the newly inserted module.

6.6.4 Associating a pluggable module with connector

A pluggable module is physically attached to a specific connector on the PCB. To reflect this physical connection, the NIC model includes the pluggable module in the respective connector entity association PDR using physical association.

6.6.5 Associating a cable with a network port

A given cable is used to carry the traffic of a specific port on a given network controller. The network port is embedded within a given network controller, and the cable is attached to a given pluggable module. As there is no physical direct connection between the network port and the cable, the logical connection between the cable and the network port is declared as a communication channel. This declaration is performed using a communication channel entity association PDR, with association type set to logical association. As described in clause 6.1.8, cables are not included in the model when using a backplane connection.

Figure 6 illustrates a logical association of a cable and a network port:

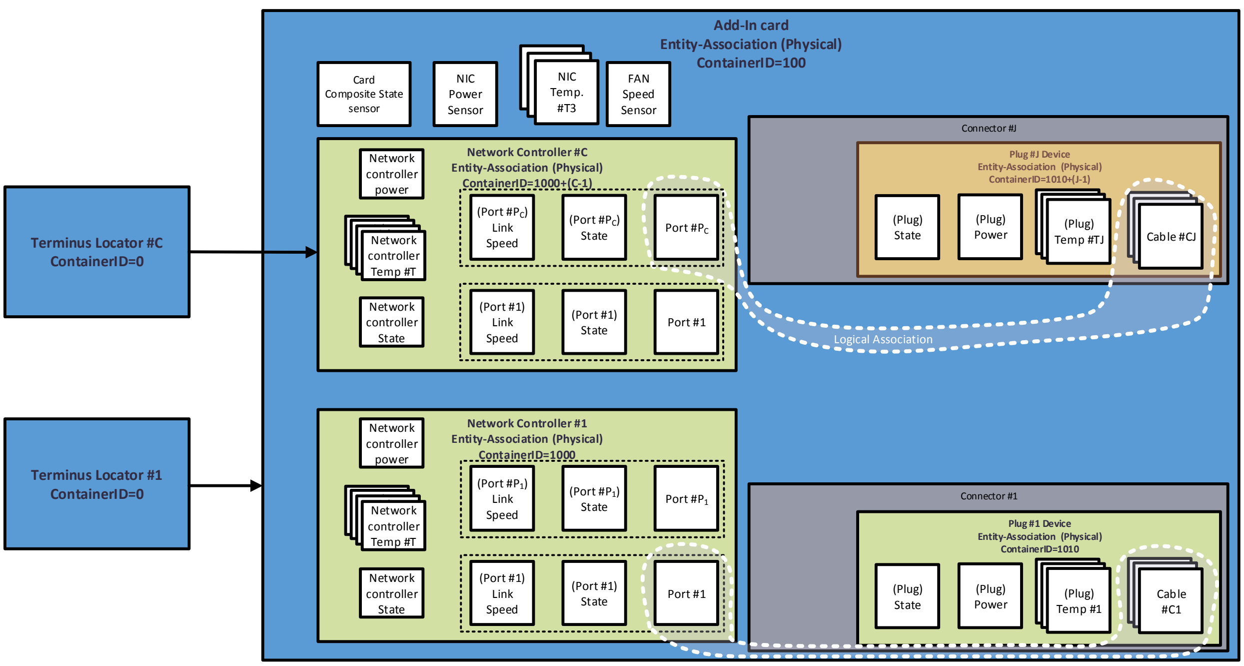

Figure 6 – Cable and network port entity association

The cable is a contained entity within the pluggable module. To associate the cable from a pluggable module to the correct network port, the communication channel entity association PDR associates the port entity in the network controller with the cable in the pluggable module.

Notes:

1) When a cable with no pluggable module is used (such as an RJ45 cable) there is no pluggable module defined, and the cable is declared as directly attached to the connector. In this case, the association of the cable to the network controller's network-port should be adjusted accordingly.

2) Even though every hierarchy is an independent numeric space, the example uses unique instances for the cables to allow matching the cable number to the marking on the NIC bracket.

6.7 Events

The model supports using PLDM events as a method to notify the MC upon changes to a model setting or to any of the model PDRs. The following events can be used with the model:

6.7.1 Network controller configuration change

This event indicates to the MC that some of the configuration parameters of the network controller have changed. Such changes could relate to link settings and/or enablement of a network port. The MC may use the ***GetPDRRepositoryInfo*** command and check if the ***timestamp*** parameter value has changed since it read the PDRs. The MC may update the whole PDRs repository by re-reading all the PDRs, or only update its repository. The value used for the ***timestamp*** shall be a virtual time value initialized by the network controller at device initialization.

An alternative approach for the MC to track PDRs change is using the newly defined ***pldmNewPDRAdded***, ***pldmExistingPDRDeleted*** and ***pldmPDRRepositoryChgEvent*** platform events.

The MC should re-read any changed PDRs to get the new information.

6.7.2 Pluggable module insertion and removal notification

This event is important to notify the MC on pluggable module presence change. It is needed for both thermal threshold management as well as for module's presence indication. When the MC receives notification of new pluggable module insertion it shall read the parameters of the newly inserted pluggable module as it may have different power class information and/or thermal thresholds. Note that while the model reflects common sensors for pluggable modules, there could be additional sensors outside the scope of this document. Additionally, when changing from a single-cable pluggable module to one with a break-out cable, the whole NIC configuration may have to change accordingly. This may induce a change in the PDR repository.

As described in clause 6.1.8, pluggable modules are not included in the model when using a backplane connection.

6.7.3 Health and state sensors events notifications

The NIC may report a change to any of its health or state sensors using a PLDM state or numeric sensor event. Providing such a notification can significantly shorten the response time, compared to waiting for the MC to poll the sensors, for an occurrence that requires the MC to take an action such as increasing the airflow from a cooling FAN.

7 Model use example

The following example for modeling a NIC using PLDM for Platform Monitoring and Control describes a NIC with the following attributes:

- Dual-port NIC
- Single Network controller
  - Dual Ethernet port
  - Single on-chip temp sensor
- Single ambient temperature sensor on the PCB
- A QSFP pluggable module is attached to each network connector
  - The QSFP pluggable module has a single temp sensor and a single cable

Figure 7 illustrates the model which is used in the example.

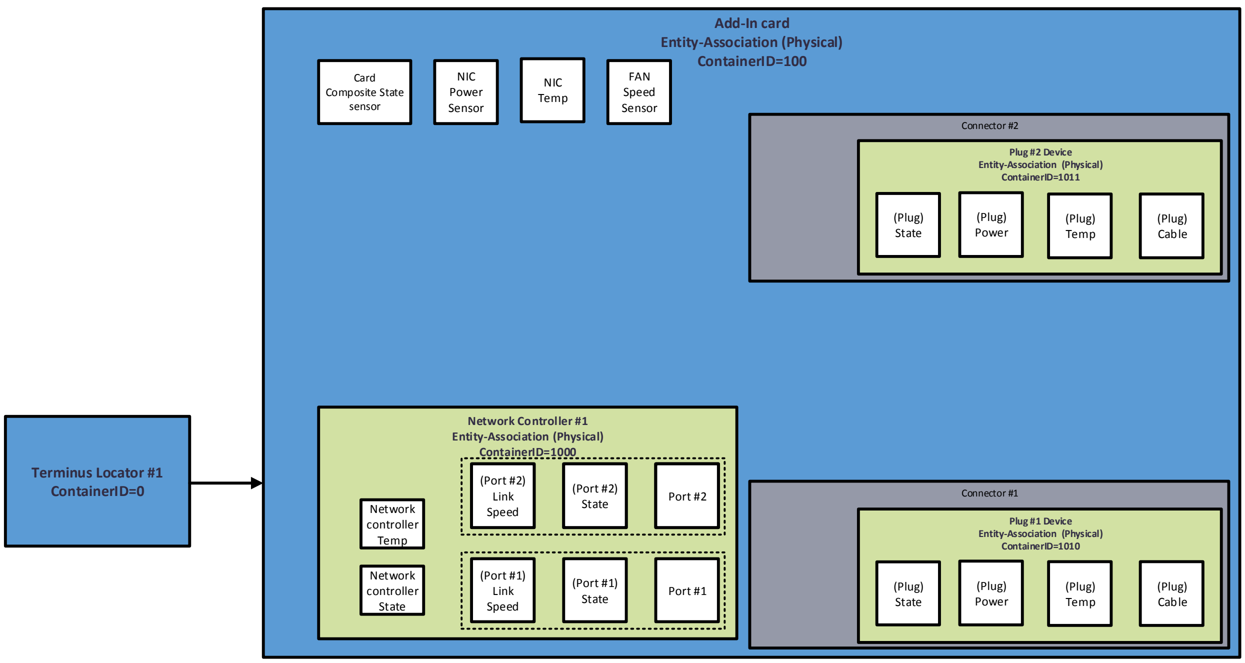

Figure 7 – Example model diagram

7.1 Model hierarchy

The model PDRs identify the elements depicted in Figure 5. The hierarchies are illustrated in the following diagram. For simplicity, Figure 8 does not show sensors. The physical connections between pluggable modules and their corresponding connectors are modeled using physical entity association. The linkages between cables and their corresponding network ports to form the communication channels are modeled using logical entity association.

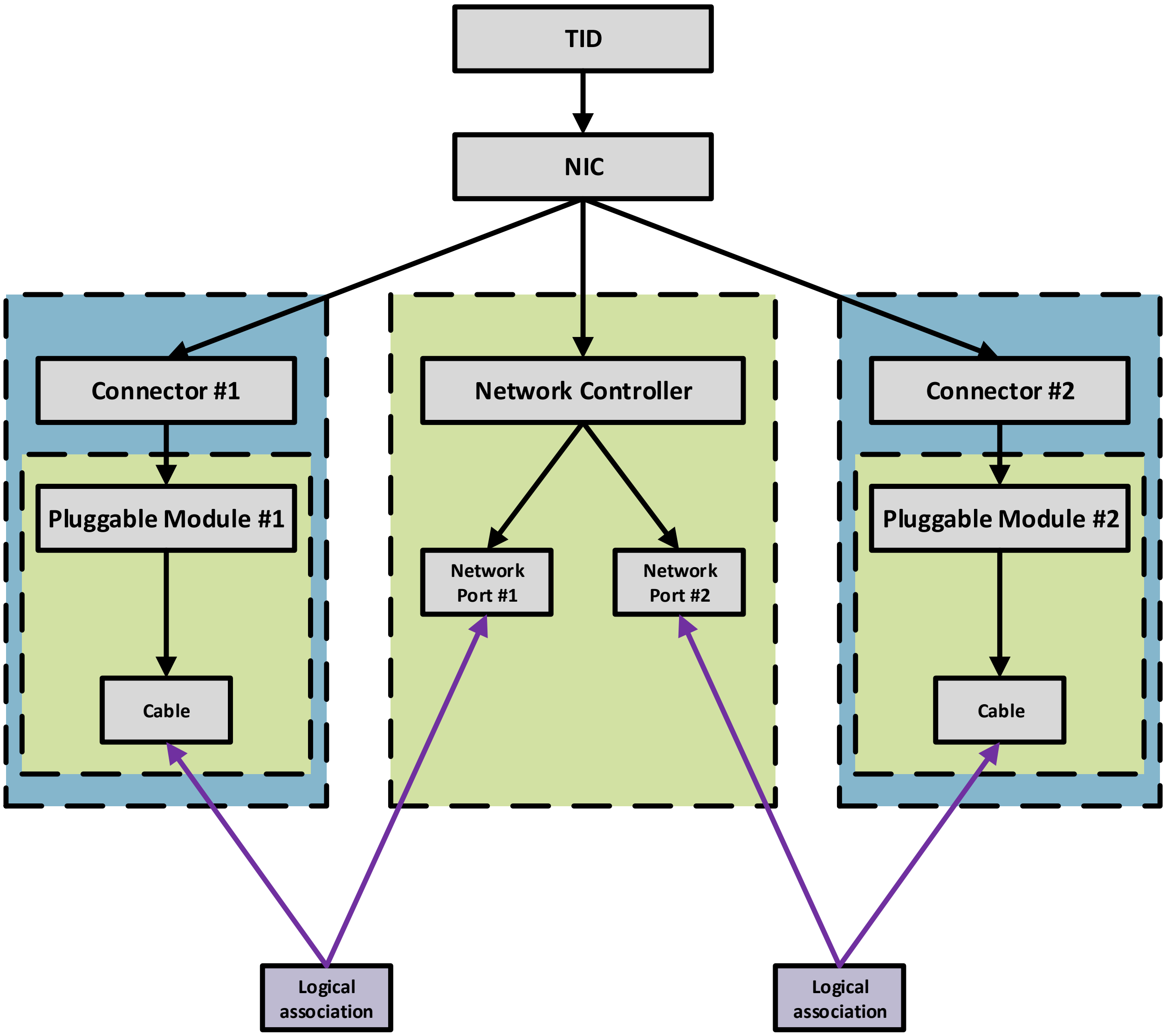

Figure 8 – NIC model hierarchy

7.2 Top-level TID

The terminus ID is identified by the terminus locator PDR. The TID defines the top-level entry point to the PLDM model. Because there is only one network controller on the NIC, there is only one TID in this example.

Table 5 – TID PDR

Field name	Value	Description
Container ID	0	System
TID		Assigned by System
Record Handle	10	Opaque number
Terminus Locator Size	1	Size of(EID) or size of(UID)
Terminus Locator Type	1/0	MCTP EID/RBT UID
EID	EID	MCTP assigned EID Value
UID	UID	Vendor provided UUID format value

The TID value is assigned to the terminus by the system controller. When the transport layer is MCTP then the identification of the terminus is performed using the Endpoint ID (EID) value. When using PLDM over RBT the terminus locator PDR shall use the UID (instead of EID). The UID value in the terminus locator PDR uses the device UUID value as the termini UID, for more information regarding terminus locator PDR see DSP0248.

7.3 NIC

The top level of the model is the NIC level. The NIC includes the physical elements which are the network controller (only one controller in this example) and the connectors.

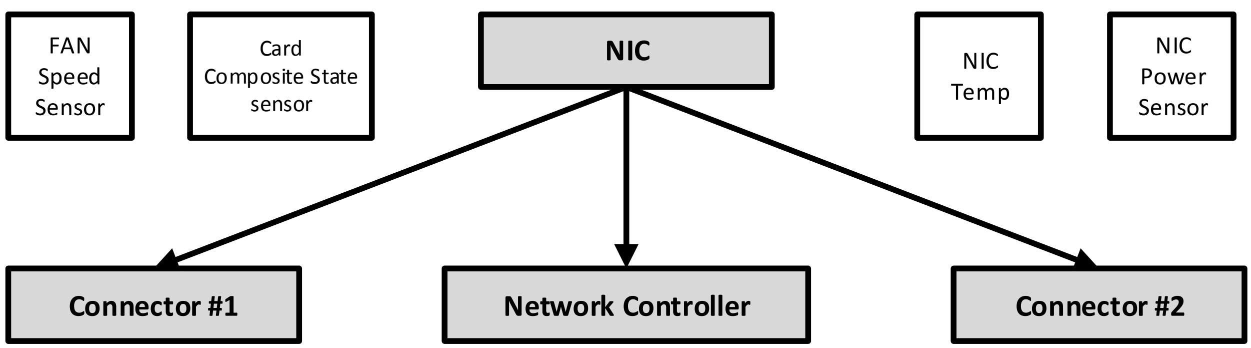

Figure 9 – NIC level elements

The sensors in the NIC level are described using a reference to the measured entity, independently of the container that includes all the physical elements on the NIC.

NIC Entity Association PDR

Container ID	100
Record Handle	1100

Container Entity		
Entity Type	68	Add-In card
Entity Instance Number	1	
Container Entity Container ID	0	System

Association Type	Physical to Physical containment

Contained Entity - Network Controller		
Entity Type	144	Network controller
Entity Instance Number	1	
Contained Entity Container ID	100	NIC

Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	1	
Contained Entity Container ID	100	NIC

Contained Entity - Connector		
Entity Type	185	Connector
Entity Instance Number	2	
Contained Entity Container ID	100	NIC

743 **Figure 10 – NIC container PDR**

744 Note that the NIC's ID, 100, will be referenced by the sensors not included in the entity association PDR.
745 The enumeration model shown in

Table 4 includes the container ID for every hierarchy level.

7.3.1 NIC power sensor

The NIC power sensor is a numeric sensor. It is not included in the NIC container PDR as sensors are defined by directly referencing the entity being measured.

Using a Container ID value of 100 implies that this PDR is reporting a sensor that is part of the container ID 100, which in this model relates to the NIC level shown in Figure 7.

NIC Power sensor PDR		
Field	**Value**	**Description**
Record Handle	1102	
Sensor ID	6	
Entity Type	68	Add-In card
Entity Instance	1	NIC Instance #
Container ID	0	System
Base Units	7	Watt
Unit Modifier	-1	0.1Watt resolution

Figure 11 – NIC power sensor PDR

7.3.2 NIC temperature sensor

The NIC temperature sensor reports the card's temperature. While it is possible to have multiple temperature-sensors installed on the PCB, this example has only one.

The temperature sensor is a numeric sensor. It is not included in the NIC container PDR as sensors are defined by directly referencing the entity being measured.

Ambient Temperature sensor PDR		
Field	**Value**	**Description**
Record Handle	1130	
Sensor ID	20	
Entity Type	68	Add-In card
Entity Instance	1	NIC Instance #
Container ID	0	System
Base Units	2	Degrees C

Figure 12 – Ambient Temperature sensor PDR

Using a Container ID value of 100 implies that this PDR is reporting a sensor that is part of container ID 100, which in this model relates to the NIC level shown in Figure 7.

7.3.3 NIC FAN speed sensor

The FAN speed sensor in the NIC reports the fan speed of an active cooling FAN. While it is possible to have multiple FANs installed on the PCB, each with its own speed sensor, this example has only one.

The FAN speed sensor is a numeric sensor. It is not included in the NIC container PDR as sensors are defined by directly referencing the entity being measured.

NIC FAN speed sensor PDR		
Field	**Value**	**Description**
Record Handle	1140	
Sensor ID	30	
Entity Type	68	Add-In card
Entity Instance	1	NIC Instance #
Container ID	0	System
Base Units	19	RPM
Unit Modifier	0	no need for scaling

Figure 13 – FAN speed sensor PDR

Using a Container ID value of 100 implies that this PDR is reporting a sensor that is part of container ID 100, which in this model relates to the NIC level shown in Figure 7.

7.3.4 NIC composite state sensor

The configuration state change reported in this sensor relates to changes in pluggable modules or in the network controller device. When a pluggable module is inserted or removed, the card configuration changes. In this example, there is a single network controller device, which allows complete visibility of configuration changes from the NIC level. Invalid configuration is applicable to cases where the pluggable module cannot be supported for any reason such as installing a pluggable module with breakout cable to a card which does not support a breakout cable.

When there are multiple network controllers, it may not be possible to report an overall NIC configuration state. In this case, the NIC configuration change and configuration state sensors should not be included in the NIC composite state sensor.

The state sensor is not included in the NIC container PDR as sensors are defined by directly referencing the entity being measured.

NIC composite State Sensor PDR		
Record Handle	1101	
Entity Type	68	Add-In card
Entity Instance Number	1	
Container Entity Container ID	0	System

Terminus Handle	0
Sensor ID	5
Composite Sensor Count	4

Sensor Type	1	Health state
Possible States	1=Normal, 3=Critical, 5=Upper_Non_Critical, 4=Fatal	

Sensor Type	15	Configuration
Possible States	1=Valid Configuration, 2=Invalid Configuration	

Sensor Type	16	Configuration Change
Possible States	1=Normal, 2=Change in Configuration	

Sensor Type	21	Thermal Trip
Possible States	1=Normal, 2=Over-Temp Shutdown	

Figure 14 – NIC composite state sensor PDR

Using a Container ID value of 0 implies that this PDR is reporting a sensor that is part of the top level container ID 0, which relates to the NIC level.

7.3.5 NIC connectors

The connectors in the model represent the physical elements into which pluggable modules are installed. It is assumed that the instance IDs of the connectors will be set to match the port number as marked on the hardware bracket. This ensures consistency between the physical marking and the logical reporting of the PLDM model.

The connector type reflects one of the possible types of pluggable modules that can be used with the specific NIC. The enumerated values used for connector types are defined in DSP0249 in the PLDM Entity ID Code Tables.

The connectors are part of the NIC physical elements and are thus included within the NIC container PDR.

7.4 Network controller

The network controller is the active device in charge of the network connection. In the given example of an Ethernet NIC the network controller device includes the MAC and PHY layers of the network ports.

Being a physical entity, the network controller is already declared within the NIC container PDR. The content of the network controller includes a set of sensors related to the network ports, as well as a set of device-level sensors. The following diagram illustrates the model elements for the network controller in the example model:

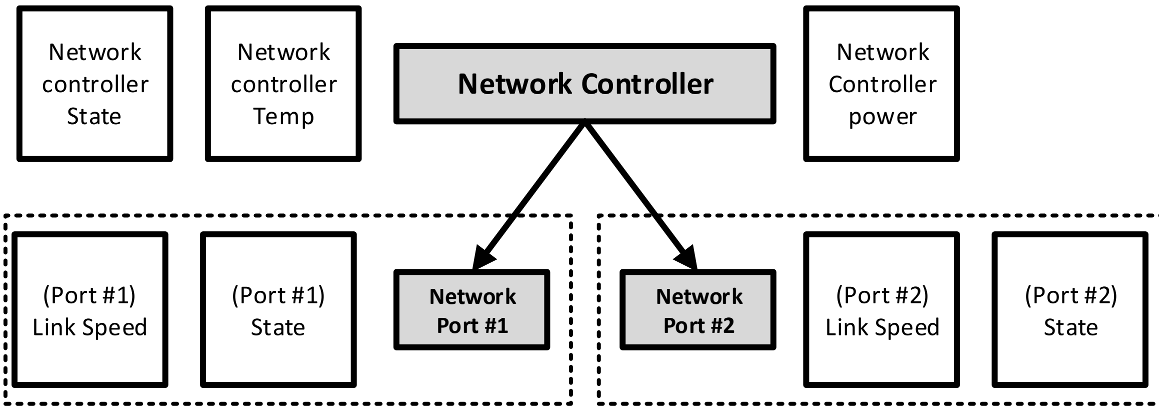

Figure 15 – Example model network controller

The network controller content is declared using an entity-association PDR that includes the hierarchical description of the network controller. The device-level sensors as well as the network port sensors are declared with separate PDRs using direct references to the measured entities. The dotted lines in the diagram are used to illustrate the association of the link and state sensors to their network port. In this example use case the network port is an Ethernet port; for different network port types, the corresponding port type ID should be used.

Network Controller Association PDR

Container ID	1000
Record Handle	1150

Container Entity		
Entity Type	144	Network controller
Entity Instance Number	1	
Container Entity Container ID	100	NIC

Association Type	Physical to Physical containment

Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	1	
Contained Entity Container ID	1000	Network Controller

Contained Entity - Communication Port		
Entity Type	300	Ethernet port
Entity Instance Number	2	
Contained Entity Container ID	1000	Network Controller

Figure 16 – Network controller association PDR

The network controller is contained within the NIC level (ID 100) and has ID 1000. This creates a hierarchy that allows sensors to be associated with the network controller, as described in the clauses 7.4.1, 7.4.3 and 7.4.4.

7.4.1 Network controller temperature sensor

The network controller temperature sensor reflects the device’s temperature. The thresholds that define its normal, warning, critical and fatal ranges are design specific and should be defined by the device manufacturer.

Network Controller Temperature sensor PDR		
Field	**Value**	**Description**
Record Handle	1500	
Sensor ID	300	
Entity Type	144	Network controller
Entity Instance	1	Network Controller Instance #
Container ID	100	NIC
Base Units	2	Degrees C

Figure 17 – Network controller temp sensor PDR

In this example there is only one temperature sensor on the device. There may be more than 1 temperature sensor in a given device. It is recommended that every network controller device contain at least one temperature sensor to allow the MC to perform thermal monitoring and system control.

The container ID in this case is 100 which references the NIC, as defined in 7.4.

7.4.2 Network controller power sensor

The network controller power sensor reflects the present value of the device’s power consumption. The thresholds which may be used by the sensor to define its normal, warning, critical and fatal ranges are design specific and should be defined by the device manufacture.

Field	Value	Comment
Record Handle	1160	
Sensor ID	50	
Entity Type	144	Network controller
Entity Instance	1	Network Controller Instance #
Container ID	100	NIC
Base Units	7	Watt
Unit Modifier	-1	0.1Watt resolution

Figure 18 – Network controller power sensor PDR

Network controllers that cannot report real-time information may report the expected maximal power for the present operating mode.

7.4.3 Network controller composite state sensor

The network controller’s composite state sensor reports the operational state of the network controller. Composite state sensors aggregate multiple metrics into a single sensor that provides an overview of the operational and health state of the controller. The MC can use this sensor to identify issues with the controller, as well as to identify which maintenance operations are required to be performed by the MC. Such operations may include reset to the network controller, system-level shut-down for thermal protection and other system-level maintenance operations.

Using the configuration change indication, the controller can trigger notification to the MC so that it can retrieve the updated PDRs which are affected by the configuration change.

When FW Update is detected, the composite state sensor can reflect this event to the MC, so that the MC can take the needed action to respond to the update. Note that reading the new FW version should be performed by the MC using protocols other than PLDM for Platform Monitoring and Control, such as DSP0257 and/or DSP0267. Please note that FW update only reflect the conclusion of the FW programming operation. It is device-specific dependent if this detection also implies that the new FW is already active or not.

Network Controller composite State Sensor PDR		
Record Handle	1170	
Entity Type	144	Network controller
Entity Instance Number	1	
Container Entity Container ID	100	NIC

Terminus Handle	0
Sensor ID	60
Composite Sensor Count	5

Sensor Type	1	Health state
Possible States	1=Normal, 3=Critical, 5=Upper_Non_Critical, 4=Fatal	

Sensor Type	15	Configuration
Possible States	1=Valid Configuration, 2=Invalid Configuration	

Sensor Type	16	Configuration Change
Possible States	1=Normal, 2=Change in Configuration	

Sensor Type	21	Thermal Trip
Possible States	1=Normal, 2=Over-Temp Shutdown	

Sensor Type	18	Firmware Version
Possible States	1=Normal, 2=Version change detected - Compatible, 3=Version change detected Incompatible	

Figure 19 – Network controller composite state sensor PDR

The container ID in this case, 100, references the network controller defined in clause 7.4.

7.4.4 Network controller Ethernet port

The Ethernet network port is declared as an entity within the network controller. The sensors within the network controller that monitor a given channel are declared by directly referencing the corresponding port ID.

7.4.4.1 Network controller port state

A state sensor is used to reflect the operational state of the port. The following diagrams show composite state sensors for the two network controller ports in the example NIC:

Network Controller Port #1 State Sensor PDR		
Record Handle	1400	
Entity Type	300	Ethernet port
Entity Instance Number	1	
Container Entity Container ID	1000	Network Controllers

Terminus Handle	0
Sensor ID	200
Composite Sensor Count	1

Sensor Type	33	Port State
Possible States	1=Connected, 2=Disconnected	

Figure 20 – Network port 1 state sensor PDR

Network Controller Port #2 State Sensor		
Record Handle	1401	
Entity Type	300	Ethernet port
Entity Instance Number	2	
Contained Entity Container ID	1000	Network Controllers

Terminus Handle	0
Sensor ID	201
Composite Sensor Count	1

Sensor Type	33	Port State
Possible States	1=Connected, 2=Disconnected	

Figure 21 – Network port 2 state sensor PDR

As can be seen from the PDR diagrams, links can be characterized as either connected or disconnected. Note that the disconnected link state implies simply that the link operation is not enabled; in particular, it does not imply that the physical link connection is disconnected.

The container ID in this case, 1000, references the network controller defined in clause 7.4.

7.4.4.2 Network controller port speed

The network port may operate at various communication speeds. This numeric sensor reports the actual operating link speed.

The following diagrams show the link speed PDRs of the two network ports in the example:

Network controller link speed sensor Port #1 PDR		
Field	**Value**	**Description**
Record Handle	1300	
Sensor ID	100	
Entity Type	300	Ethernet port
Entity Instance	1	Port instance #1 in controller
Container ID	1000	Network Controller
Base Unit	60	Bits
Unit Modifier	6	Mbits
Rate Unit	3	Per Second

Figure 22 – Network port 1 link speed sensor PDR

Network controller link speed sensor Port #2 PDR		
Field	**Value**	**Description**
Record Handle	1301	
Sensor ID	101	
Entity Type	300	Ethernet port
Entity Instance	2	Port instance #2 in controller
Container ID	1000	Network Controller
Base Unit	60	Bits
Unit Modifier	6	Mbits
Rate Unit	3	Per Second

Figure 23 – Network port 2 link speed sensor PDR

PLDM numeric sensor PDRs require specification of both the units of measure and the scaling method of the reported value. In this case, the units of measure are bits/second and the scaling multiplier of the measured value is 1,000,000. Together, these yield a reported value in Mbps.

The container ID in this case, 1000, references the network controller defined in clause 7.4.

7.5 Pluggable module

As defined in clause 6.6.3, the pluggable module includes one or more cables as well as some sensors. The following diagram shows the content of the first pluggable module in the example model. The pluggable module type in the model matches the network interface connector type; in this example, QSFP. Note that the numeric sensors and the composite state sensor in the pluggable module are not described in the pluggable module hierarchy itself: these sensors are only declared by referencing the measured entity.

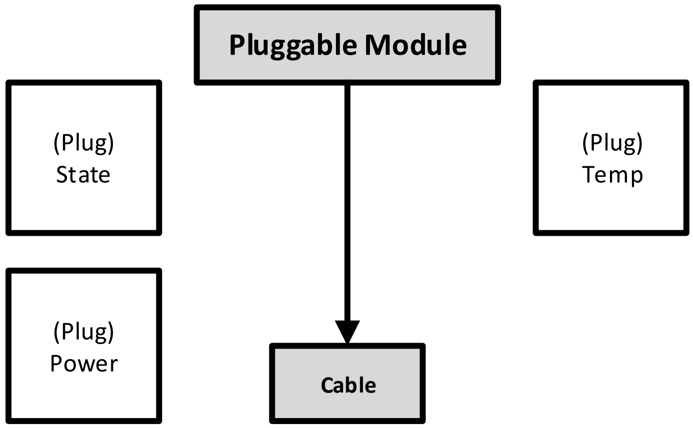

Figure 24 – Example pluggable module structure

The following PDR defines the content of the pluggable module device using physical association between the cables and the pluggable modules.

Pluggable module #1 Entity Association PDR

Container ID	1010
Record Handle	1600

Container Entity		
Entity Type	214	QSFP Module
Entity Instance Number	1	
Container Entity Container ID	1040	Connector #1

Association Type	Physical to Physical containment

Contained Entity - Cable		
Entity Type	187	Cable
Entity Instance Number	1	
Contained Entity Container ID	1010	Plug #1

Figure 25 – Pluggable Module #1 entity association

Pluggable module #2 Entity Association PDR

Container ID	1011
Record Handle	1601

Container Entity		
Entity Type	214	QSFP Module
Entity Instance Number	1	
Container Entity Container ID	1041	Connector #2

Association Type	Physical to Physical containment

Contained Entity - Cable		
Entity Type	187	Cable
Entity Instance Number	1	
Contained Entity Container ID	1011	Plug #2

Figure 26 – Pluggable Module #2 entity association

The pluggable modules are part of their respective connectors; this is indicated because they point to their connectors' container ID values. Each of the pluggable modules has its own content and its own hierarchy ID. In the example, these are 1010 for the 1st pluggable module and 1011 for the 2nd pluggable module.

7.5.1 Pluggable module temperature sensor

The pluggable module temperature sensor reports the pluggable module temperature. As with the other sensors, the sensor is declared by referencing the measured entity.

Plug #1 Temperature sensor PDR		
Field	**Value**	**Description**
Record Handle	1800	
Sensor ID	500	
Entity Type	214	QSFP Module
Entity Instance	1	Temp sensor #1 in Plug
Container ID	1040	Connector #1
Base Units	2	Degrees C

Figure 27 – Plug #1 temperature sensor PDR

Plug #2 Temperature sensor PDR		
Field	**Value**	**Description**
Record Handle	1801	
Sensor ID	501	
Entity Type	214	QSFP Module
Entity Instance	1	Temp sensor #1 in Plug
Container ID	1041	Connector #2
Base Units	2	Degrees C

Figure 28 – Plug #2 temperature sensor PDR

Note that as the instance ID of each element is enumerated within its hierarchy, both sensors can have instance ID of 1 as they are in different pluggable modules, while both are uniquely defined. The container ID of each of the sensors matches the corresponding pluggable module Container ID.

If a pluggable module is turned off by the network controller - for thermal protection or for any other reason -- the reported temperature shall reflect the last measured value read before the pluggable module was turned off.

7.5.2 Pluggable module power sensor

The pluggable module power sensor reports the pluggable module's expected or measured power consumption. As with other sensors, the sensor is declared by referencing the measured entity.

Plug #1 Power sensor PDR		
Field	**Value**	**Description**
Record Handle	1700	
Sensor ID	400	
Entity Type	214	QSFP Module
Entity Instance	1	Power sensor #1 in Plug
Container ID	1040	Connector #1
Base Units	7	Watt
Unit Modifier	-1	0.1Watt resolution

Figure 29 – Pluggable module #1 power sensor

Plug #2 Power sensor PDR		
Field	**Value**	**Description**
Record Handle	1701	
Sensor ID	401	
Entity Type	0	#N/A
Entity Instance	1	Power sensor #1 in Plug
Container ID	1041	Connector #2
Base Units	7	Watt
Unit Modifier	-1	0.1Watt resolution

Figure 30 – Pluggable module #2 power sensor

The unit of measure in this case is Watts and the multiplication scaling factor is 0.1; therefore, the reported value will use tenths of a Watt.

7.5.3 Pluggable module composite state sensor

The pluggable module's composite state sensor reports the overall operational state of the pluggable module.

Plug #1 composite State Sensor PDR		
Record Handle	2000	
Entity Type	214	QSFP Module
Entity Instance Number	1	
Container Entity Container ID	1040	Connector #1

Terminus Handle	0
Sensor ID	700
Composite Sensor Count	3

Sensor Type	1	Health state
Possible States	1=Normal, 3=Critical, 5=Upper_Non_Critical, 4=Fatal	

Sensor Type	13	Presence
Possible States	1=Present, 2=Not_Present	

Sensor Type	21	Thermal Trip
Possible States	1=Normal, 2=Over-Temp Shutdown	

Figure 31 – Pluggable Module #1 composite state sensor PDR

Plug #2 composite State Sensor PDR		
Record Handle	2001	
Entity Type	214	QSFP Module
Entity Instance Number	2	
Container Entity Container ID	1041	Connector #2

Terminus Handle	0
Sensor ID	701
Composite Sensor Count	3

Sensor Type	1	Health state
Possible States	1=Normal, 3=Critical, 5=Upper_Non_Critical, 4=Fatal	

Sensor Type	13	Presence
Possible States	1=Present, 2=Not_Present	

Sensor Type	21	Thermal Trip
Possible States	1=Normal, 2=Over-Temp Shutdown	

Figure 32 – Pluggable Module #2 composite state sensor PDR

The composite state sensor uses temperature thresholds detailed in Table 1 to report the health state and the thermal state rating. When there is no pluggable module in the NIC, the presence sensor will report the module's absence.

If a module is turned off by the network controller for thermal protection or for any other reason, the reported health state shall reflect the last known state of the module prior to being turned off.

7.6 Connector association to a Pluggable module

Pluggable modules are defined within the Connector hierarchy level. The association of pluggable modules with their connectors is done using entity association PDRs as described in Figure 34 and Figure 35. The following diagram illustrates the entity association in the example model:

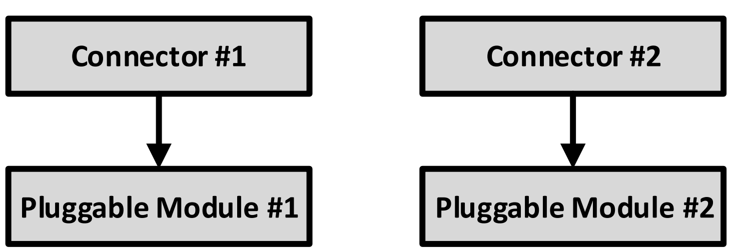

Figure 33 – Pluggable module association with connectors

937 The corresponding entity association PDRs are shown below:

938

Connector #1 entity association PDR		
Container ID	1040	
Record Handle	1110	
Container Entity		
Entity Type	185	Connector
Entity Instance Number	1	
Container Entity Container ID	100	NIC
Association Type	Physical containment	
Contained Entity - Cable		
Entity Type	214	QSFP Module
Entity Instance Number	1	
Contained Entity Container ID	1040	Connector #1

939 **Figure 34 – Connector #1 entity association PDR**

Connector #2 entity association PDR

Container ID	1041
Record Handle	1111

Container Entity		
Entity Type	185	Connector
Entity Instance Number	2	
Container Entity Container ID	100	NIC

Association Type	Physical containment

Contained Entity - Cable		
Entity Type	214	QSFP Module
Entity Instance Number	1	
Contained Entity Container ID	1041	Connector #2

Figure 35 – Connector #2 entity association PDR

As can be seen from the provided PDRs, Connector 1 is used with pluggable module 1 and connector 2 is used with pluggable module 2. It is strongly recommended to match pluggable module instance numbers and connector numbers to the port numbers physically marked on the PCB card and/or bracket to ensure coherent database enumeration.

7.7 Logical association of a cable with a network port

The PLDM NIC model associates cables with network ports in a network controller via logical association as described in clause 6.6.5. The following diagram illustrates entity association in the example model:

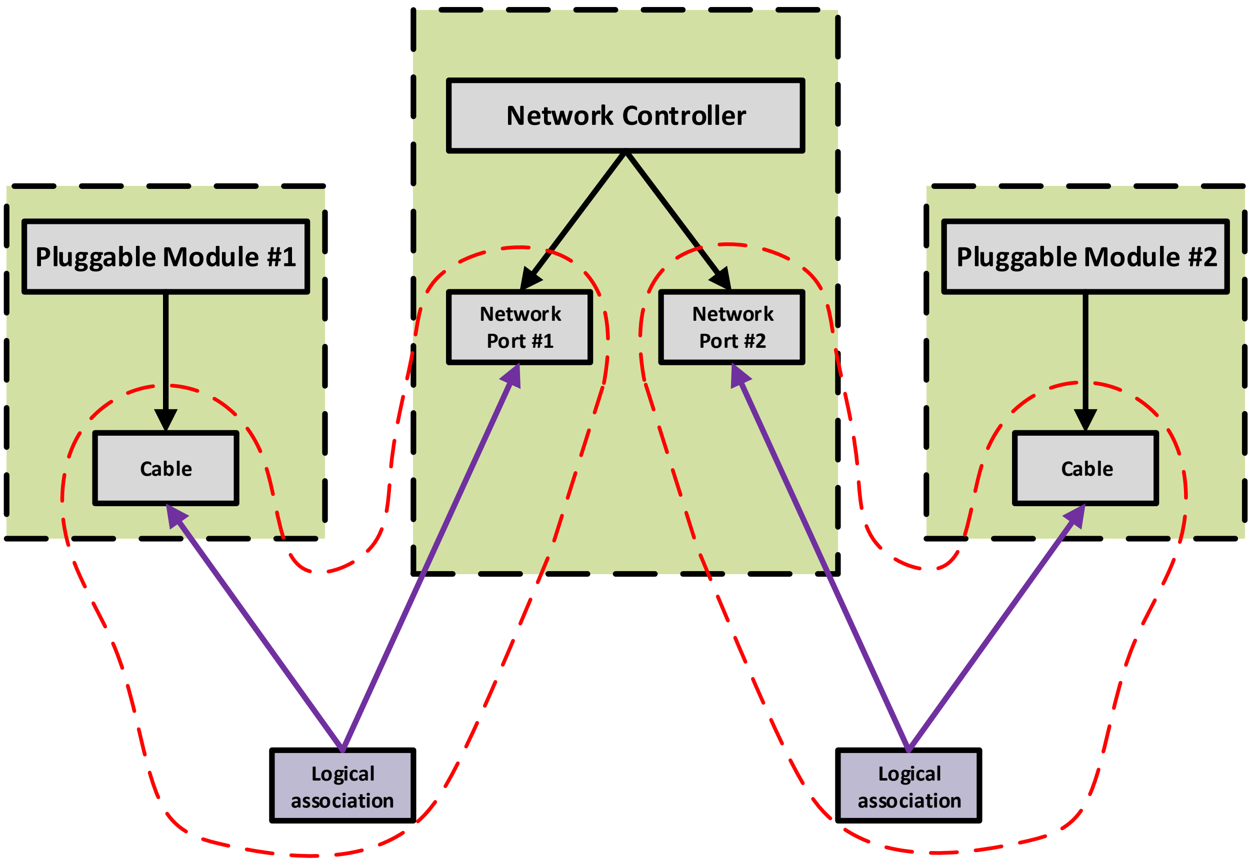

Figure 36 – Logical association of cables with network controller ports

The logical associations of the network ports to the cables are shown by the dashed red lines. Different NIC implementations may map their network ports to the physical connectors and to the associated cabled in different ways. Entity association PDRs allows modeling any NIC implementation. For example, note the different associations of the cables here in Figure 36 as compared to in Figure 6, above.

The corresponding entity association PDRs are shown below:

Channel #1 entity association PDR

Container ID	1060
Record Handle	2100

Container Entity		
Entity Type	6	Communication Channel
Entity Instance Number	1	
Container Entity Container ID	100	NIC

Association Type	Logical containment

Contained Entity - Network Controller		
Entity Type	300	Ethernet port
Entity Instance Number	1	
Contained Entity Container ID	1000	Network Controllers

Contained Entity - Cable		
Entity Type	187	Cable
Entity Instance Number	1	
Contained Entity Container ID	1010	Plug #1

957 **Figure 37 – Cable #1 entity association with controller network port #1**

Channel #2 entity association PDR

Container ID	1061
Record Handle	2101

Container Entity		
Entity Type	6	Communication Channel
Entity Instance Number	2	
Container Entity Container ID	100	NIC

Association Type	Logical containment

Contained Entity - Network Controller		
Entity Type	300	Ethernet port
Entity Instance Number	2	
Contained Entity Container ID	1000	Network Controllers

Contained Entity - Cable		
Entity Type	187	Cable
Entity Instance Number	1	
Contained Entity Container ID	1011	Plug #2

Figure 38 – Cable #2 entity association with controller network port #2

The Cable instance number is 1 for both PDRs. The reasoning for this is that the enumeration for every entity is performed within its hierarchy. As there is only 1 cable in each pluggable module, in our example, both are having the same instance ID value of 1, but each is referenced within a different hierarchy. If a breakout cable were in use, the component cables within it would be numbered with instance numbers 1, 2, 3, etc.

ANNEX A
(informative)

Change log

Version	Date	Description
1.0.0	2019-12-18	