

Redfish Requirements for the Ultra Ethernet Server Endpoint

v1.0

October 28, 2026

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Copyright © 2026 Ultra Ethernet Consortium.

The Ultra Ethernet Consortium (UEC) is a consensus-based standards organization operating under the Joint Development Foundation. Its members collaborate openly to define and promote high-performance Ethernet technologies for modern computing environments. This specification ("Specification") contains Approved Deliverables of UEC as such term is defined in the UEC Charter.

This document is made available under the terms of the Creative Commons Attribution-NoDerivatives 4.0 International License (CC BY-ND 4.0). As stated in the license, you may copy and redistribute the document, provided that attribution is given to the Ultra Ethernet Consortium. However, if you create a derivative work from the material, you may not distribute that work. A copy of the license is available at https://creativecommons.org/licenses/by-nd/4.0/.

This Specification is provided on an "AS IS" basis. UEC and The Joint Development Foundation (the "Parties") make no warranties of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose or noninfringement of any third party intellectual property rights, or compliance with applicable laws. Use of the information in this Specification is at your own risk. The entire risk as to the results and performance of this Specification is assumed by the user. In no event will the Parties (as defined above) be liable to any other party for lost profits or any form of indirect, special, incidental, or consequential damages of any character from any causes of action of any kind with respect to this Specification or its governing document, whether based on breach of contract, tort (including negligence), or otherwise, and whether or not the recipient of this Specification has been advised of the possibility of such damage.

This Specification may include references to technologies for which patent rights might be claimed. UEC does not make any representation regarding the existence or status of any such rights. Implementers of this Specification are solely responsible for obtaining any necessary licenses or permissions from the appropriate rights holders.

UEC reserves the right to adopt any changes or alterations to this Specification as it deems necessary or appropriate.

Ultra Ethernet, Ultra Ethernet Consortium, and the Ultra Ethernet Consortium logo are the unregistered trademarks of Ultra Ethernet Consortium in the United States and other countries. All rights reserved. Other brands, names or logos contained herein are the property of their respective owners.

For more information about the Ultra Ethernet Consortium, visit: https://ultraethernet.org.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Contents

1 Terms 7

2 Redfish Requirements 7

2.1 Scope 7

2.2 Managing the UE Layers 7

2.3 Redfish Overview 8

2.3.1 Services and Resources 9

2.3.2 Packet Delivery Sublayer (PDS) Management 11

2.3.3 Managing FEP 11

2.4 Transport Extensions 12

2.4.1 Managing Congestion Signaling (CSIG) 13

2.5 Redfish Model for Managing the Server Network Endpoint 14

2.5.1 Network Adapter Resource 15

2.5.2 Port Resource 17

2.5.3 Network Device Function Resource 19

2.5.4 Ethernet Interface Resource 20

2.5.5 PCIe Device Resource 21

2.5.6 PCIe Function Resource 21

2.6 Normative Requirements 22

2.7 Management Tasks for an Ethernet NIC 22

2.8 Management Tasks for an Ultra Ethernet NIC 23

2.8.1 Ultra Ethernet to Redfish Mapping 23

2.8.2 Get CSIG Config 25

2.8.3 Set CSIG Config 25

2.8.4 Get FEP Config 26

3 References 26

3.1 Normative References 26

3.2 Informative References 27

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Tables

Table 1-1 - Terms 7
Table 2-1 - FEP Attributes 12
Table 2-2 - Server Ethernet NIC Endpoint Management Tasks 22
Table 2-3 - Server UE NIC Endpoint Management Tasks 23
Table 2-4 - Ultra Ethernet to Redfish Mapping 23

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Figures

Figure 2-1 - UE Layers 8
Figure 2-2 - Redfish Hierarchy Model 14
Figure 2-3 - Redfish NIC Model 15

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Acknowledgments

This specification is developed under the Process, Procedures, and Guidelines of the Ultra Ethernet Consortium (UEC). The organization would like to thank all that are involved. A special acknowledgment goes out to the following individuals for their leadership, authoring, and technical contributions to this specification.

Steering Committee at Time of Publication

Role	Name	Affiliation
Chair	J Metz	AMD
Vice Chair	Abdul Kabbani	Microsoft
TAC Chair	Tom Emmons	Arista
TAC Vice Chair	Brad Matthews	Marvell

Working Group Leadership at Time of Publication

Working Group	Name	Affiliation
Physical Layer WG	Jannik Hammel Nielsen	Huawei
	Adee Ran	Cisco
Link Layer WG	Bob Alverson	HPE
	Yan Zhuang	Huawei
Transport Layer WG	Torsten Hoefler	Microsoft
	Karen Schramm	Broadcom
Software Layer WG	Rip Sohan	Broadcom
	Jianxin Xiong	Intel
Management WG	Ravindranath Kanakarajan	Marvell
	Hemal Shah	Broadcom
Compliance & Test WG	Dave Estes	Viavi Solutions
	Cliff Tavares	Keysight
Performance & Debug WG	Rajalaxmi Angadi	Intel
	Nathan Hanford	LLNL
Storage WG	Venkat Goli	Broadcom
	Claudio DeSanti	Dell

Specification Development

Role	Name	Affiliation
Author	John Leung	Intel
Author	Vivek Kashyap	Intel
Contributor	Jeff Hilland	HPE
Editor	Paul Congdon	Linux Foundation
Program Manager	Rob Craig	Linux Foundation

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

1 Terms

Table 1-1 - Terms

Term	Description
NIC	Network interface card. In Redfish, the NIC is modeled as a network adapter resource.
Redfish	An interface for managing the physical chassis and compute services and resource.
Redfish profile	A JSON-formatted file that prescribes the requirements on a Redfish implementation for a specific management domain. The requirements specify the resources, resource properties, and property values of a conformant Redfish implementation.
Redfish mockup	A folder structure that contains example responses that would be returned for RESTful GET requests.

2 Redfish Requirements

2.1 Scope

This document specifies how the Redfish interface will be used to manage an Ultra Ethernet NIC endpoint on the server. An Ultra Ethernet NIC is referred to as a fabric endpoint (FEP) [7] .

2.2 Managing the UE Layers

The Redfish interface is used to manage the following UE components:

- A UE NIC (FEP) endpoint on the server
- UE fabric
- Storage attached via UE

Figure 2-1 shows the components of the UE layers and the UE components within each layer. Each component may need to be managed (e.g., configure, enable, etc.).

- Managing the physical layer includes managing the physical port.
- Managing the link layer includes managing the logical port.
- Managing the transport layer includes managing the transport protocols, multi-pathing, packet spraying, congestion control, and trimming.
- Managing the software layer includes managing firmware and drivers.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

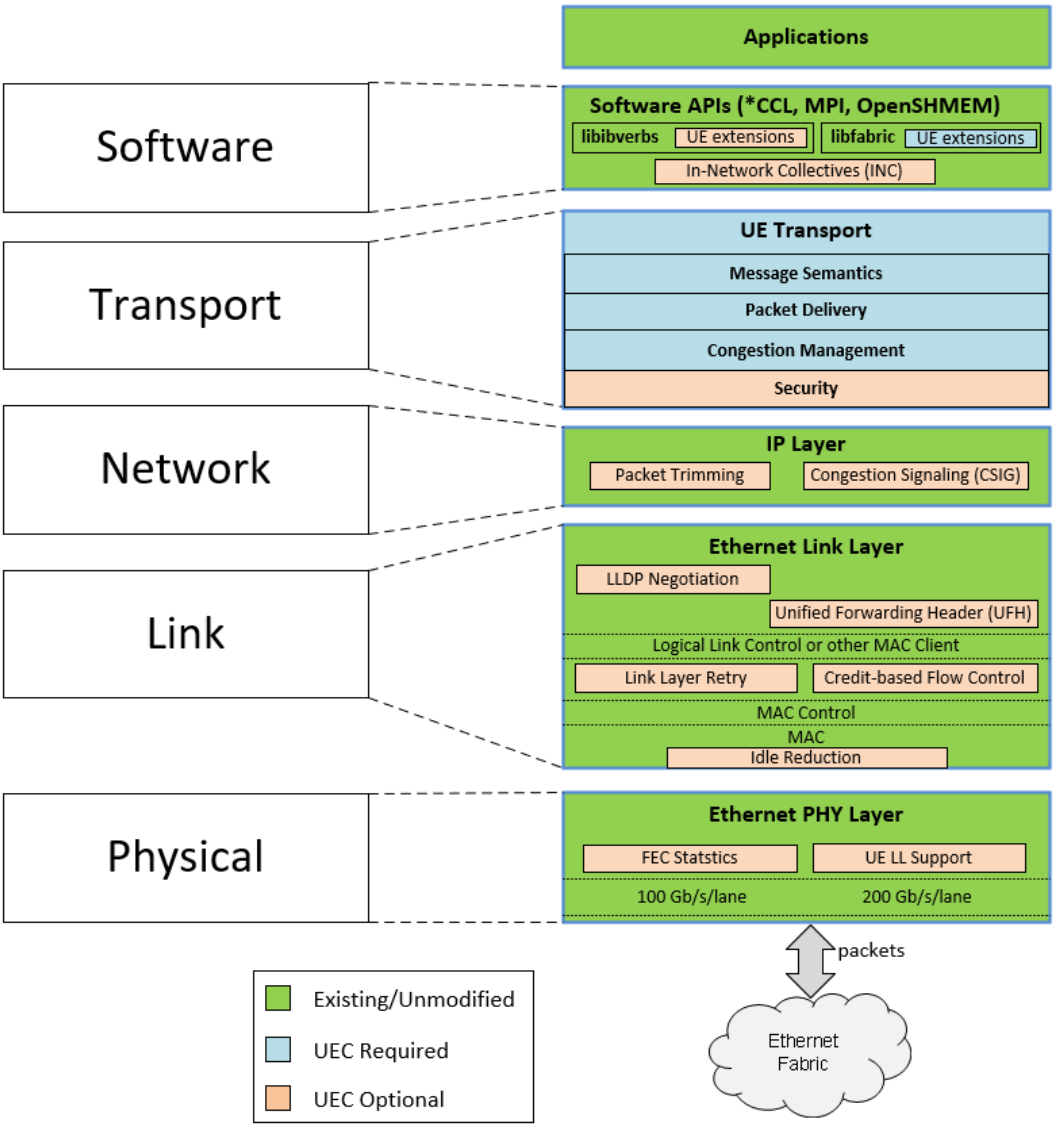


Figure 2-1 - UE Layers

2.3 Redfish Overview

The Redfish® standard specifies an HTTP RESTful interface for exposing system manageability. A remote client can use the Redfish interface to manage a conformant Redfish service.

The Redfish standard consists of the Redfish Specification [1] and the Redfish Data Model Specification [2] . The Redfish Specification defines the RESTful behavior of the interface. The model specification defines the structure of each resource. The resource representation contains properties of the resource, actions that can be performed on the resource, and references to other resources within the model.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

A Redfish service can emit messages as an event, alert, or log entry. Redfish contains a set of message registries that describes the messages that can be emitted. In the registry, each entry contains the string and fields of a message. The message registries are described in the “Redfish Message Registry Guide” [12] .

The Redfish model is expressed in three schema formats: OpenAPI, JSON-schema and CSDL (common schema definition language). This allows the three tool chains to support the Redfish interface.

A Redfish service that implements the Redfish interface can:

- Manage the physical, functional aspect of multiple systems, and the connectivity between the systems (fabrics).
- Administer the management hierarchy, which represents the layers of management (for example from datacenter manager to rack manager to node manager to component manager).
- Manage sessions, events, accounts, telemetry, and logs.
- Manage the composition and aggregation of systems.
- Manage certificates.
- Update firmware, reset the system, configure BIOS, and configure network ports.

The Redfish model expands as other standards bodies adopt it for their own management domains. DMTF has received extension requests for storage services, fabric management, data center infrastructure management (DCIM), industrial IOT, and process automation organizations.

2.3.1 Services and Resources

Redfish provides mechanisms to manage various aspects of manageability. These mechanisms are available in the Redfish model. Redfish contains resources that model management services and resources that model the logical, physical, and managerial aspects of a system.

- Logical, physical, manager model: The Redfish model contains a logical model, a physical model, and a manager model. The latter two models model the physical and management hierarchies, respectively. The logical model (or functional model) covers compute, storage, and fabric functions.
- Fabric model: The fabric model enables the user to manage various fabrics using the same management pattern. This includes PCIe, CXL, Ethernet, and SANs. DMTF worked with the OpenFabric Alliance on this model. The model is described in the “Redfish Fabrics White Paper” [20] .
- OS, VMs, and container model: The OS/VM/container model was created from a contribution by the Open Process Automation Forum for managing software entities above the hardware platform. Currently, the model provides only basic manageability.
- Telemetry service: The telemetry service provides a mechanism to describe metrics and to specify metric reports and triggers. This model is described in the “Redfish Telemetry White Paper” [5] .

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

- Composition service: The composition service enables a user to request a composition be created from the resources managed by a Redfish service. The requests can be made by specifying constraints or specifying intent. The model is described in the “Redfish Composability White Paper” [8] .
- Firmware update service: The firmware update service supports updating by pushing or pulling one or more firmware images. The model is described in the “Redfish Firmware Update White Paper” [9] .
- Certificate service: The certificate management service describes certificates for devices, services, and other resources. It also provides interface for clients to manage certificates. The model is described in the "Redfish Certificate Management White Paper” [10] .
- Aggregation service: The aggregation service provides a mechanism for performing actions on an ad-hoc or persistent group of resources. Access points define the properties needed to access the entity being aggregated, and connection methods describe the protocol or other semantics of the connection.
- Account, session, event, task, job services: These services are specified to support the remote client. The account service is the basis for providing privilege. The session service shows each session connected to the Redfish service. The event service enables a user to subscribe to asynchronous messages. The task service enables the user to track asynchronous actions. The job service enables the user to schedule and execute operations.

The Redfish model has been utilized for managing datacenter platforms (compute, storage, and networking), DCIM (data center infrastructure management), and process automation components. The model covers managing the physical and functional aspects of multiple systems.

Redfish has been extended for usage in management domains beyond compute. Each management domain can prescribe the subset of the Redfish model for their management domain by creating a profile document. The profile document is a machine-readable file. The Open Compute Project [16] , the Open Process Automation Forum [17] , OpenStack, SNIA, and DMTF [18] have created profiles.

The DMTF supports a validator—the Redfish Interop Validator—that can be used to test an implementation for conformance to requirements included in the profile document. The validator reads the profile document, autogenerates tests, executes them against an implementation, and generates a detailed test report.

2.3.1.1 Managing the UE server endpoint

This section describes how the Redfish interface is used to manage an Ultra Ethernet (UE) server endpoint implemented by a UE-capable NIC function (fabric endpoint, FEP). The goal is to enable consistent out-of-band discovery, configuration, and operational monitoring of UE endpoint transport. UE transport configuration is surfaced primarily through the *NetworkDeviceFunction* resource associated with the UE-capable NIC function.

UE endpoint manageability includes:

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

- core endpoint transport configuration and observability, and
- optional transport extensions.

The core transport configuration is described first using Ultra Ethernet's Packet Delivery Sublayer (PDS) and FEP concepts, while optional capabilities (for example, Congestion Signaling (CSIG)) are called out explicitly under "Transport Extensions." This separation allows a client to implement baseline workflows while conditionally enabling extension workflows when the endpoint reports the corresponding capabilities.

2.3.2 Packet Delivery Sublayer (PDS) Management

2.3.2.1 Overview

PDS management covers endpoint configuration and state related to packet delivery behavior and delivery-related policies that are relevant to operational management. The PDS configuration surfaces those endpoint parameters that an operator may need to set or audit to ensure interoperable and predictable delivery behavior.

2.3.2.2 PDS Configuration

PDS configuration and state are exposed by the UE FEP under *NetworkDeviceFunction* resource, as the *UltraEthernet.FEP* property.

The following parameters represent the illustrative PDS configuration set used by this document for UE endpoint management. These parameters are presented as PDS properties and reflect endpoint management list.

Illustrative PDS configuration categories include:

- Encapsulation / protocol selection
    - (e.g., UET over UDP vs UET over IP, next-header/protocol identifier)
- Data protection mode
    - (e.g., none, CRC, security-tagged modes if applicable)
- Sequence/range/windowing controls
    - (e.g., PSN range limiting, receiver-advertised packet range)
- Acknowledgement behavior
    - (e.g., ACK-on-ECN behavior, ACK coalescing support/enablement, ACK thresholds)
- ACK payload sizing and trimming-related thresholds
    - (e.g., maximum ACK return data size, threshold for trimmable ACK classification)
- Timer initialization
    - (e.g., initial retransmission timeout parameter)

2.3.3 Managing FEP

Table 2-1 lists the attributes of UE FEP (fabric endpoint). These attributes are derived from the parameters defined in Table 3-28 of the UE Specification v1.1.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

Table 2-1 - FEP Attributes

Name	Type	Default (Range)	Description
UET_OVER_UDP	boolean	TRUE	TRUE indicates UET over UDP. FALSE indicates UET over IP. The default is TRUE.
UET_data_protect	uint8	1 (0-2)	Fabric: 0 – No CRC or TSS 1 – CRC 2 – TSS
Limit_PSN_Range	boolean	TRUE	Sender-side limit on PSN gap => PDC closes when PSN = start_PSN+2^{31}. If 0, PSN can continue to grow and wrap.
Default_MPR	uint8	8 (1-255)	The default MPR assumed when creating a PDC.
Max_Ack_data_size	uint16	0-8 KB	Maximum amount of return data that can be carried with a PDS ACK sent in response to a PDS Request sent in the forward direction.
Trimmable_ACK_size	uint16	10 (0-10 KB)	ACK packets carrying read response data which are larger than this size use a trimmable DSCP.
ACK_on_ECN	boolean	TRUE	ECN marked packet triggers an ACK.
Coalesced ACK support	boolean	TRUE	Coalesced ACK is supported.
Enable ACK per packet	boolean	FALSE	If coalesced ACKs are supported, then when TRUE, use coalesced ACKs; otherwise generate an ACK per packet.
ACK_Gen_Trigger	uint8	16 KB (0-32) Quanta: 256	When the amount of received data accumulated since the last ACK reaches ACK_Gen_Trigger, an ACK is generated. This parameter is applicable when ACK coalescing is enabled.
ACK_Gen_Min_pkt_add	uint8	1 KB (0-2) Quanta: 64 B	When a packet is received at a PDC, the received data contributes toward ACK_Gen_Trigger. ACK_Gen_Min_pkt_add specifies the minimum amount that shall be credited toward ACK_Gen_Trigger for each packet received. This parameter is applicable when ACK coalescing is enabled.
RTO_Init_Time	uint8	(0-8) sec Quanta 128 ns	Retransmit packet after this amount of time if no ACK or NACK within this configured time at source. No default. Depends on fabric scale and CC mode.

2.4 Transport Extensions

This section describes UE endpoint manageability features that are not required for core transport operation. A client consults capability indicators in the UE extension to determine whether these extensions are supported and applicable.

When an optional extension is not supported, the associated properties should be absent or read-only, and management clients should skip extension-specific configuration steps while continuing to rely on the baseline management surfaces for operational workflows.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

2.4.1 Managing Congestion Signaling (CSIG)

2.4.1.1 Overview

CSIG provides optional in-band congestion signaling capabilities and associated endpoint configuration/state. CSIG behavior may depend on a domain-wide policy (for example, a preferred tag format selection) and requires endpoints to reflect and apply that policy consistently when operating within a CSIG-enabled domain.

2.4.1.2 CSIG

The following discussion describes the fabric-wide considerations that influence endpoint CSIG configuration and the endpoint's responsibility to reflect the domain policy when operating within a CSIG-enabled fabric domain.

The UE server endpoint is part of the UE fabric. The UE fabric needs to be globally configured regarding congestion signaling (CSIG). This global configuration setting indicates the preferred tag format for the UE CSIG domain. The tag format can be compact or wide. All fabric endpoints (FEPs) SHOULD use this global configuration setting to determine whether to insert a compact- or wide-format CSIG tag.

There SHOULD NOT be a mix of compact and wide CSIG-capable paths in the same UE CSIG fabric domain.

A global configuration setting for the entire UE fabric and set of FEPs will indicate whether CSIG is enabled for the entire fabric or disabled for the entire fabric.

The CSIG spec defines an architecture for the in-band telemetry protocol.

The theory of operations is to insert a tag in the Ethernet packet, after the VLAN tag. Each tag carries a specific signal type (example delay or available bandwidth). The values are added by the intermediate switches. The values are overwritten or left alone based on the function defined as minima or maxima. The server endpoint inserts the tag and terminates it.

The global configuration setting specifies the configuration for the entire UE fabric and the set of FEPs.

- Indicates whether CSIG is enabled or disabled for the entire fabric.
- Indicates whether the preferred tag format for the UE CSIG domain is the compact or wide format.

There will be no mix of compact and wide CSIG-capable paths in same UE CSIG domain. All FEPs SHOULD use this global configuration setting to determine whether to insert a compact- or wide-format CSIG tag.

2.4.1.3 Link Negotiation Options for CSIG

The UE LLDP Specification defines a link negotiation options TLV that includes a 2-bit telemetry (TEL) field, used to exchange CSIG capability information over peer links.

Each device, after learning the CSIG capability of its peer device over their peering link, performs an intersection operation to determine the final state that should be configured on its local link.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

The TLV exchange on the link will produce a link state for current CSIG capability state of the link after LLDP negotiation. The link state may be “Disabled", "CompactEnabled", "WideEnabled", or "BothEnabled".

The "StripAndForward“ state also exists but this state is only applicable to switches and not endpoints (FEPs in servers).

2.5 Redfish Model for Managing the Server Network Endpoint

The Redfish interface is used to manage a server platform. Figure 2-2 shows the Redfish hierarchy model of a server with a network interface. A Redfish client would provide the resource path (URI) to the resource of interest with the RESTful operation. The resource hierarchy starts at /redfish/v1 (also called the service root).

Redfish models the physical aspect and functional aspects of a managed entity via chassis and systems hierarchies.

The figure shows only the system hierarchy. However, other functional aspects are also modeled in

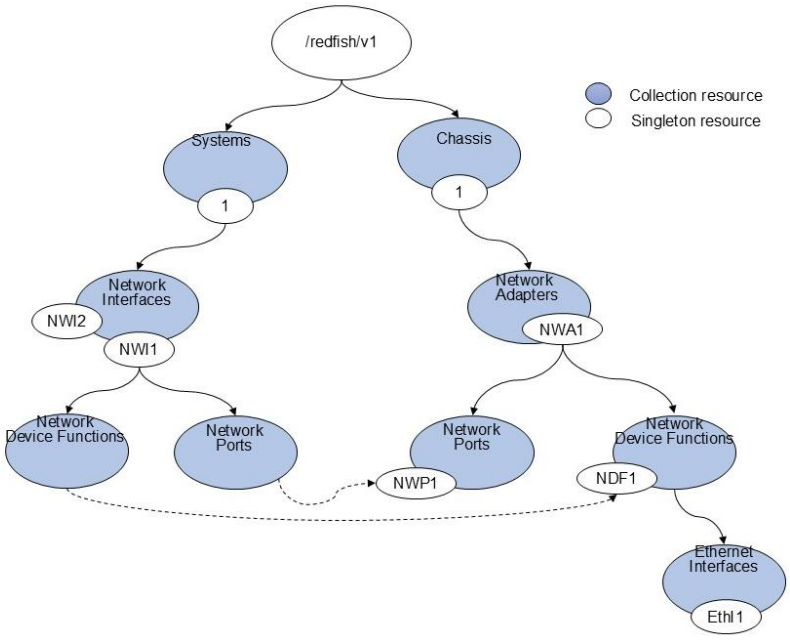


Figure 2-2 - Redfish Hierarchy Model

other hierarchies subordinate to the service root. Examples of these functional aspects are managers, fabrics, and storage services. The number of functional aspects is determined by the complexity of what the Redfish service is managing.

In the physical context, a chassis may contain one or more network adapters. Each network adapter is composed of a collection of ports and a collection of network device functions. Each collection may have one or more ports and network device functions. In the functional context, a system may contain one or

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

more network interfaces. Each network interface is composed of a collection of ports and a collection of network device functions. The port collection may have one or more links to a port or network device function. The construct allows a straightforward traversal from the functional to physical context.

The network adapter may contain an Ethernet interface collection. Figure 2-3 shows the Redfish NIC model.

The following sections show examples (or mockups) of the resource. These examples can be found on the OCP Profiles GitHub [8] .

The mockups show only a subset of the properties available for a resource. The Redfish Data Model Specification [2] provides a description of the Redfish data model and describes all the properties available for each resource.

The Redfish mockup server [19] can be used to quickly emulate a Redfish service and respond to RESTful GET requests. The mockup is specified on the invoking command line.

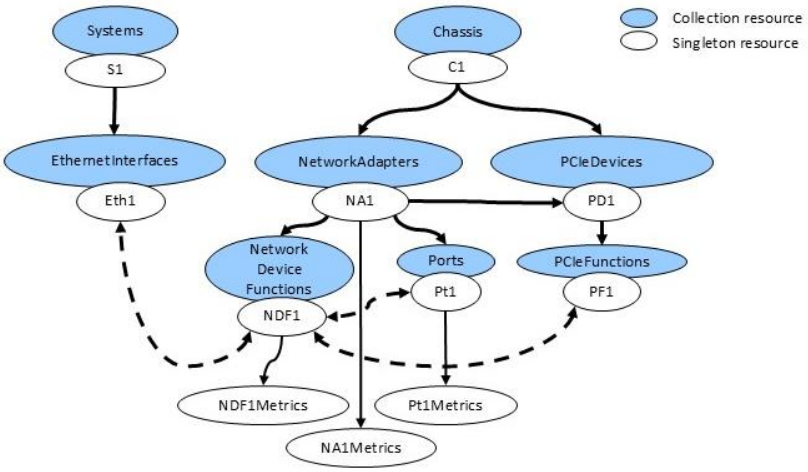


Figure 2-3 - Redfish NIC Model

2.5.1 Network Adapter Resource

The *NetworkAdapter* resource contains references to the port collection and the network device function collections. The resource also contains the properties of any controllers on the network adapter.

After the properties, the resource contains the actions that can be performed on the resource. For this resource, the action is *ResetSettingsToDefault*. Within the *Action* property, the *Target* property contains the URI on which to perform an HTTP POST.

At the bottom of the resource is a *@Redfish.Settings* property. The *Settings* property allows multiple properties to be changed in the resource referenced by the *SettingsObject* property. Once the changes

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

are made to the *Settings* resource, the client can apply the changes by performing an HTTP POST to the *Settings* resource. The property also indicates the parameter(s) that can be included in the POST request to the *Settings* resource. In the mockup, the only supported apply time is *OnReset*.


```
{
    "@odata.type": "#NetworkAdapter.v1_9_0.NetworkAdapter",
    "Id": "DE07A000",
    "Name": "Network Adapter",
    "Manufacturer": "Contoso",
    "Model": "Contoso 2",
    "SKU": "Contoso 2 function adapter",
    "SerialNumber": "LMNOP4279",
    "PartNumber": "ABCDEFG2",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "Ports": {
        "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports"
    },
    "NetworkDeviceFunctions": {
        "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/NetworkDeviceFunctions"
    },
    "Metrics": {
        "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Metrics"
    },
    "Controllers": [
        {
            "FirmwarePackageVersion": "229.1.123.0",
            "Links": {
                "PCIeDevices": [
                    {
                        "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000"
                    }
                ]
            },
            "ControllerCapabilities": {
                "NetworkPortCount": 4,
                "NetworkDeviceFunctionCount": 16,
                "DataCenterBridging": {
                    "Capable": true
                },
                "NPAR": {
                    "NparCapable": true,
                    "NparEnabled": true
                },
                "VirtualizationOffload": {
                    "SRIOV": {
                        "SRIOVVEPACapable": true
                    },
                    "VirtualFunction": {
                        "DeviceMaxCount": 256,
                        "MinAssignmentGroupSize": 8,
                        "NetworkPortMaxCount": 256
                    }
                }
            },
            "PCIeInterface": {
                "LanesInUse": 8,
                "MaxLanes": 16,
                "MaxPCIeType": "Gen4",
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
                "PCIeType": "Gen4"
            }
        }
    ],
    "LLDPEnabled": true,
    "Actions": {
        "#NetworkAdapter.ResetSettingsToDefault": {
            "target":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Actions/NetworkAdapter.ResetSettingsTo
Default",
            "@Redfish.OperationApplyTimeSupport": {
                "@odata.type": "#Settings.v1_3_3.OperationApplyTimeSupport",
                "SupportedValues": [ "OnReset" ]
            }
        }
    },
    "@odata.context": "/redfish/v1/$metadata#NetworkAdapter.NetworkAdapter",
    "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000",
    "@odata.etag": "W/\"4DFAAF27\"",
    "@Redfish.Settings": {
        "@odata.type": "#Settings.v1_3_3.Settings",
        "SettingsObject": {
            "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Settings"
        },
        "SupportedApplyTimes": [ "OnReset" ]
    }
}
```


2.5.2 Port Resource

The *Port* resource contains properties for the port and associated link.


```
{
    "@odata.type": "#Port.v1_6_0.Port",
    "Id": "1",
    "Name": "Ethernet Port 1",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "Metrics": {
        "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1/Metrics"
    },
    "PortId": "1",
    "PortProtocol": "Ethernet",
    "PortType": "BidirectionalPort",
    "Enabled": true,
    "Ethernet": {
        "SupportedEthernetCapabilities": [ "WakeOnLAN" ],
        "AssociatedMACAddresses": [
            "9c:dc:71:c3:bb:0a",
            "9c:dc:71:c3:bb:0e",
            "9c:dc:71:c3:bb:12",
            "9c:dc:71:c3:bb:16"
        ],
        "FlowControlConfiguration": "None",
        "FlowControlStatus": "None",
        "WakeOnLANEnabled": true,
        "LLDPEnabled": true,
        "LLDPReceive": {
            "ChassisId": "2c:23:3a:48:2f:5d",
            "ChassisIdSubtype": "MacAddr",
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
            "ManagementAddressIPv4": "",
            "ManagementAddressIPv6": "",
            "ManagementAddressMAC": "2c:23:3a:48:2f:ae",
            "ManagementVlanId": 4095,
            "PortId":
"54:65:6E:2D:47:69:67:61:62:69:74:45:74:68:65:72:6E:65:74:31:2F:32:2F:39",
            "PortIdSubtype": "IfName"
        },
        "LLDPTransmit": {
            "ChassisId": "9c:dc:71:c3:bb:16",
            "ChassisIdSubtype": "MacAddr",
            "ManagementAddressIPv4": "",
            "ManagementAddressIPv6": "",
            "ManagementAddressMAC": "9c:dc:71:c3:bb:16",
            "ManagementVlanId": 4095,
            "PortId": "9C:DC:71:C3:BB:16",
            "PortIdSubtype": "MacAddr"
        }
    },
    "LinkConfiguration": [
        {
            "AutoSpeedNegotiationCapable": true,
            "AutoSpeedNegotiationEnabled": true,
            "CapableLinkSpeedGbps": [ 25.0, 10.0 ],
            "ConfiguredNetworkLinks": [
                {
                    "ConfiguredLinkSpeedGbps": 25.0,
                    "ConfiguredWidth": 1
                },
                {
                    "ConfiguredLinkSpeedGbps": 10.0,
                    "ConfiguredWidth": 1
                }
            ]
        }
    ],
    "LinkNetworkTechnology": "Ethernet",
    "MaxFrameSize": 9622,
    "MaxSpeedGbps": 25.0,
    "Width": 1,
    "InterfaceEnabled": true,
    "SignalDetected": true,
    "PortMedium": "Optical",
    "LinkState": "Enabled",
    "LinkStatus": "LinkUp",
    "LinkTransitionIndicator": 1,
    "CurrentSpeedGbps": 10.0,
    "ActiveWidth": 1,
    "SFP": {
        "SupportedSFPTypes": [
            "SFP",
            "SFPPlus",
            "SFP28"
        ],
        "Status": {
            "Health": "OK",
            "State": "Enabled"
        },
        "Manufacturer": "Mellanox",
        "PartNumber": "844483-B21",
        "SerialNumber": "THY1020240",
        "MediumType": "FiberOptic",
        "FiberConnectionType": "SingleMode",
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
        "Type": "SFP28"
    },
    "Actions": {
        "#Port.Reset": {
            "target":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1/Actions/Port.Reset",
            "ResetType@Redfish.AllowableValues": [
                "ForceRestart",
                "ForceOn",
                "ForceOff"
            ],
            "@Redfish.OperationApplyTimeSupport": {
                "@odata.type": "#Settings.v1_3_3.OperationApplyTimeSupport",
                "SupportedValues": [
                    "Immediate"
                ]
            }
        }
    },
       "@odata.context": "/redfish/v1/$metadata#Port.Port",
    "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1",
    "@odata.etag": "W/\"B40342B6\"",
    "@Redfish.Settings": {
        "@odata.type": "#Settings.v1_3_3.Settings",
        "SettingsObject": {
            "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1/Settings"
        },
        "SupportedApplyTimes": [ "OnReset" ]
    }
}
```


2.5.3 Network Device Function Resource

The *NetworkDeviceFunction* resource contains properties describing the capabilities supported. In the example shown, the capabilities are Ethernet and FibreChannel.


```
{
    "@odata.type": "#NetworkDeviceFunction.v1_8_0.NetworkDeviceFunction",
    "Id": "1",
    "Name": "Network Device Function 1",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "Metrics": {
        "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/NetworkDeviceFunctions/1/Metrics"
    },
    "NetDevFuncType": "Ethernet",
    "DeviceEnabled": true,
    "NetDevFuncCapabilities": [ "Ethernet" ],
    "Ethernet": {
        "MACAddress": "9c:dc:71:c3:bb:0a",
        "PermanentMACAddress": "9c:dc:71:c3:bb:0a",
        "MTUSizeMaximum": 9600
    },
    "BootMode": "PXE",
    "VirtualFunctionsEnabled": true,
    "MaxVirtualFunctions": 8,
    "AssignablePhysicalNetworkPorts": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1"
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
        }
    ],
    "Links": {
        "PCIeFunction": {
            "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000/PCIeFunctions/1"
        },
        "PhysicalNetworkPortAssignment": {
            "@odata.id": "/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/Ports/1"
        },
        "EthernetInterfaces": [
            {
                "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/5"
            }
        ]
    },
    "@odata.context":
"/redfish/v1/$metadata#NetworkDeviceFunction.NetworkDeviceFunction",
    "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/NetworkDeviceFunctions/1",
    "@odata.etag": "W/\"80212509\""
}
```


2.5.4 Ethernet Interface Resource

The *EthernetInterface* resource contains properties describing the interface. Most of the properties are shown because they are of interest to UE.


```
{
    "@odata.type": "#EthernetInterface.v1_4_1.EthernetInterface",
    "Id": "5",
    "FullDuplex": false,
    "IPv4Addresses": [],
    "IPv4StaticAddresses": [],
    "IPv6AddressPolicyTable": [],
    "IPv6Addresses": [],
    "IPv6StaticAddresses": [],
    "IPv6StaticDefaultGateways": [],
    "InterfaceEnabled": null,
    "LinkStatus": "LinkUp",
    "MACAddress": "9c:dc:71:c3:bb:0a",
    "Name": "",
    "NameServers": [],
    "SpeedMbps": null,
    "StaticNameServers": [],
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "Links": {
        "NetworkDeviceFunctions": [
            {
                "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/NetworkDeviceFunctions/1"
            }
        ]
    },
    "UefiDevicePath": "PciRoot(0x3)/Pci(0x1,0x1)/Pci(0x0,0x0)",
    "@odata.context": "/redfish/v1/$metadata#EthernetInterface.EthernetInterface",
    "@odata.etag": "W/\"C4D4ADE7\"",
    "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/5"
}
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

2.5.5 PCIe Device Resource

The *PCIeDevice* resource contains the properties of a PCIe device that is attached to a system.

```
{
    "@odata.type": "#PCIeDevice.v1_9_0.PCIeDevice",
    "Id": "DE07A000",
    "Name": "PCIe Device",
    "Manufacturer": "Contoso",
    "Model": "Contoso 2",
    "SKU": "Contoso 2 function adapter",
    "SerialNumber": "LMNOP4279",
    "PartNumber": "ABCDEFG2",
    "UUID": "00000000-0000-1000-8000-9cdc71c3bb0a",
    "DeviceType": "MultiFunction",
    "FirmwareVersion": "192.168.59.0",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "PCIeFunctions": {
        "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000/PCIeFunctions"
    },
    "PCIeInterface": {
        "LanesInUse": 8,
        "MaxLanes": 16,
        "MaxPCIeType": "Gen4",
        "PCIeType": "Gen4"
    },
    "@odata.context": "/redfish/v1/$metadata#PCIeDevice.PCIeDevice",
    "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000",
    "@odata.etag": "W/\"70C0367C\""
}
```

2.5.6 PCIe Function Resource

The *PCIeFunction* resource contains the properties of a PCIe function that is attached to a system.

```
{
    "@odata.type": "#PCIeFunction.v1_3_0.PCIeFunction",
    "Id": "1",
    "Name": "PCIe Function 1",
    "FunctionType": "Physical",
    "DeviceClass": "NetworkController",
    "FunctionId": 1,
    "DeviceId": "0x1801",
    "VendorId": "0x14e4",
    "ClassCode": "0x020000",
    "RevisionId": "0x11",
    "SubsystemId": "0x1598",
    "SubsystemVendorId": "0x14e4",
    "Status": {
        "Health": "OK",
        "State": "Enabled"
    },
    "Links": {
        "NetworkDeviceFunctions": [
            {
                "@odata.id":
"/redfish/v1/Chassis/1/NetworkAdapters/DE07A000/NetworkDeviceFunctions/1"
            }
        ],
        "PCIeDevice": {
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
            "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000"
        }
    },
    "@odata.context": "/redfish/v1/$metadata#PCIeFunction.PCIeFunction",
    "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/DE07A000/PCIeFunctions/1",
    "@odata.etag": "W/\"2D186009\""
}
```


2.6 Normative Requirements

The resource representations above illustrate where UE-capable NIC functions and related objects appear in the Redfish hierarchy. The next sections build on these examples and show the expected request/response patterns a client uses to retrieve configuration, apply configuration changes, and verify operational state using the same resource URIs.

In Redfish, normative requirements for an implementation are specified in a Redfish profile. A Redfish profile is a file whose format conforms to the “Redfish Interoperability Profiles Specification” [3] .

Conformance to a profile can be tested by executing the “Redfish Interop Validator” [15] and specifying the profile as the input file. The validator reads the specified profile, auto-generates and executes tests against an implementation, and generates a test report.

The manageability of the server UE NIC endpoint is specified by extending the manageability of the OCP Ethernet NIC and specifying the manageability unique to the UE NIC.

2.7 Management Tasks for an Ethernet NIC

OCP prescribes the manageability of a generic Ethernet NIC in their Ethernet NIC profile [4] . A usage guide is associated with the profile [20] . The usage guide describes the management tasks that can be accomplished with a conformant implementation. For each task, the usage guide shows the request/response interaction across the Redfish interface. Table 2-2 is extracted from the usage guide and enumerates the management tasks.

Table 2-2 - Server Ethernet NIC Endpoint Management Tasks

Use Case	Management Task	Requirement
NIC endpoint	Get NIC information	Mandatory
	Get device function information	Mandatory
	Get port information	Mandatory
	Get PCIe information	Mandatory
	Get PCIe function information	Mandatory
	Set NIC information	Mandatory
	Set device function information	Mandatory
	Set port information	Mandatory
	Set NIC settings to default	Mandatory
NIC hardware	Get FRU information	Mandatory
System NIC	Get System NIC information	Mandatory
Telemetry	Get NIC metrics	Mandatory
	Get device function metrics	Mandatory
	Get port metrics	Mandatory
NIC Firmware	Get firmware version	Mandatory
	Update firmware	Mandatory

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

2.8 Management Tasks for an Ultra Ethernet NIC

This section lists the manageability tasks that apply to an Ultra Ethernet NIC endpoint.

The following provides only an overview of managing Ultra Ethernet for the purpose of understanding what is required on Redfish.

UEC prescribes the manageability of a UE endpoint in their UE Ethernet NIC profile [8] .

Table 2-3 contains a list of the manageability tasks that are specific to a UE NIC. Each manageability task has a link to a section that describes the interaction between the Redfish client with the Redfish service for that task.

Table 2-3 - Server UE NIC Endpoint Management Tasks

Use Case	Management Task	Requirement	Section
FEP	Get FEP configuration	Mandatory	Section 2.8.4
CSIG	Get CSIG configuration	Mandatory	Section 2.8.2
	Set CSIG configuration	Mandatory	Section 2.8.3

The following task descriptions use the *NetworkDeviceFunction* resource as the primary anchor for UE-specific configuration. Each task is expressed as a minimal GET or PATCH interaction to retrieve the current effective configuration and, where supported, to update configuration using standard Redfish semantics.

2.8.1 Ultra Ethernet to Redfish Mapping

Table 2-4 contains the mapping between UE properties/attributes and Redfish resource properties. The Redfish convention is for property names to append the units. The unit naming conforms to the Unified Code for Units of Measure (UCUM) [6] .

Table 2-4 - Ultra Ethernet to Redfish Mapping

Section	UE Property	Redfish resource	Redfish property	Detail
CSIG	Discovery	NetworkDeviceFunction	Configuration	2.8.1.1
CSIG	ActiveTag Format	NetworkDeviceFunction	TagFormat	2.8.1.2
CSIG	LinkState	NetworkDeviceFunction	LinkCSIGState	2.8.1.3
FEP	UET_OVER_UDP	NetworkDeviceFunction	ProtocolUsed	2.8.1.4
FEP	UET_data_protect	NetworkDeviceFunction	DataProtection	2.8.1.5
FEP	Limit_PSN_Range	NetworkDeviceFunction	LimitPSNRangeEnable	2.8.1.6
FEP	Default_MPR	NetworkDeviceFunction	ReceiverMaxPacketRangeBytes	2.8.1.7
FEP	Max_Ack_data_size	NetworkDeviceFunction	MaxAckDataSize	2.8.1.8
FEP	Trimmable_ACK_size	NetworkDeviceFunction	TrimmableAckSize	2.8.1.9
FEP	ACK_on_ECN	NetworkDeviceFunction	AckOnEcnEnabled	2.8.1.10
FEP	Coalesced ACK support	NetworkDeviceFunction	CoalescedAckSupported	2.8.1.11
FEP	Enable ACK per packet	NetworkDeviceFunction	CoalescedAckEnabled	2.8.1.12
FEP	ACK_Gen_Trigger	NetworkDeviceFunction	AckGenTrigger	2.8.1.13
FEP	ACK_Gen_Min_pkt_add	NetworkDeviceFunction	AckGenMinPacketAdd	2.8.1.14
FEP	RTO_Init_Time	NetworkDeviceFunction	RTOInitTime	2.8.1.15

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

2.8.1.1 Configuration property

The configuration property SHALL indicate the list of CSIG tag formats supported by this interface. The values SHALL be either "Compact" or "Both".

2.8.1.2 TagFormat property

The *TagFormat* property SHALL indicate the currently active CSIG tag format on this interface. The values SHALL be either "Compact" or "Wide".

2.8.1.3 LinkState property

The *LinkState* property SHALL indicate the current CSIG link state after LLDP negotiation. The value SHALL be one of the following values: "Disabled", "CompactEnabled", "WideEnabled", or "BothEnabled".

2.8.1.4 ProtocolUsed property

The *ProtocolUsed* property SHALL indicate the protocol currently being used. The value SHALL be either UDP or IP.

2.8.1.5 DataProtection property

The *DataProtection* property SHALL indicate the data protection currently being used. The value SHALL be one of the following: “None”, “CRC”, or “TSS”.

2.8.1.6 LimitPSNRangeEnable property

The *LimitPSNRangeEnable* property SHALL indicate that PSN range is fixed or can grow and wrap. The property is Boolean.

2.8.1.7 ReceiverMaxPacketRangeBytes property

The *ReceiverMaxPacketRangeBytes* property SHALL specify the maximum packet range in bytes.

2.8.1.8 MaxAckDataSize property

The MaxAckDataSize property SHALL specify the value, which is multiplied by 16 bytes, to produce the maximum size of the return data with a PDS ACK. The resulting value SHALL be in the range of 0-8 kibibytes.

2.8.1.9 TrimmableAckSize property

The TrimmableAckSize property SHALL specify the value, which is multiplied by 16 bytes, to produce the ACK-packet size at which the use of the trimmable DSCP is permitted. The resulting value SHALL be in the range of 0-10 kibibytes.

2.8.1.10 AckOnEcnEnabled property

The *AckOnEcnEnabled* property SHALL indicate that an ACK is sent when the ECN is marked. The property is Boolean.

2.8.1.11 CoalescedAckSupported property

The *CoalescedAckSupport* property SHALL indicate that coalescing of ACKs is supported. The property is Boolean.

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

2.8.1.12 CoalescedAckEnabled property

The *CoalescedAckEnabled* property SHALL indicate that ACKs will be coalesced. The property is Boolean.

2.8.1.13 AckGenTrigger property

The *AckTriggerThreshold* property SHALL specify the value, which is multiplied by 256 bytes, to produce the amount of data to be received since the last ACK, before a new ACK is generated. The resulting value shall be in the range of 0-32 kibibytes. This parameter SHALL be applicable when ACK coalescing is enabled.

2.8.1.14 AckGenMinPacketAdd property

The *AckGenMinPacketAdd* property SHALL specify the value, which is multiplied by 64 bytes, to produce the minimum number of bytes added to the *ACKGenTrigger* property for each packet received. The resulting value SHALL be in the range of 0-2 kibibytes. This parameter is applicable when ACK coalescing is enabled.

2.8.1.15 RTOInitTime property

The *RTOInitTime* property SHALL specify the value, which is multiplied by 128 ns, to produce the amount of time after which no ACK or NACK is received, when a packet is retransmitted. The resulting value SHALL be in the range of 0-8 seconds.

2.8.2 Get CSIG Config

The CSIG configuration is obtained by retrieving the *NetworkDeviceFunction* resource.

```
GET
/redfish/v1/Chassis/NVMeChassis/NetworkAdapters/8fd725a1/NetworkDeviceFunctions/11100
```

The response is:

```
{
   "@odata.id":
"/redfish/v1/Chassis/1U/NetworkAdapters/8fd725a1/NetworkDeviceFunctions/11100",
   "@odata.type": "#NetworkDeviceFunction.v1_9_2.NetworkDeviceFunction",
    "Id": "11100",
    "UltraEthernet" : {
        "CSIG": {
            "Configuration": "Compact",
            "TagFormat": "Wide",
            "LinkCSIGState": "BothEnabled"
        },
        "FEP": { … }
    }
}
```

2.8.3 Set CSIG Config

The CSIG configuration is set with the following command:

```
PATCH
/redfish/v1/Chassis/NVMeChassis/NetworkAdapters/8fd725a1/NetworkDeviceFunctions/11100
```

With the following request body:

```
"CSIG": {
```


Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

```
    {
        "TagFormat": "Wide"
    }
}
```

The response would be the same as for the Get CSIG config.

2.8.4 Get FEP Config

The FEP configuration is obtained by retrieving the *NetworkDeviceFunction* resource.

```
GET
/redfish/v1/Chassis/NVMeChassis/NetworkAdapters/8fd725a1/NetworkDeviceFunctions/11100
```

The response is:

```
{
   "@odata.id":
"/redfish/v1/Chassis/1U/NetworkAdapters/8fd725a1/NetworkDeviceFunctions/11100",
   "@odata.type": "#NetworkDeviceFunction.v1_9_2.NetworkDeviceFunction",
   "Id": "11100",
   "UltraEthernet": {
      "CSIG": { … },
      "FEP": {
         "ProtocolUsed": "UDP",
         "DataProtection": "CRC",
         "LimitPSNRangeEnable": true,
         "ReceiverMaxPacketRangeBytes":  8,
         "MaxAckDataSize": "2",
         "TrimmableAckSize": 10,
         "AckOnEcnEnabled": true,
         "CoalescedAckSupported": true,
         "CoalescedAckEnabled": true,
         "AckTriggerThreshold": 8,
         "AckGenMinPacketAdd": 1,
         "RTOInitTime": 8
      }
   }
}
```

3 References

This specification makes both normative and informative references. Normative references are required to assure interoperability among UE components. Informative references are intended to provide additional background and further understanding of the UE operating environment and the Redfish manageability interface.

3.1 Normative References

[1] DMTF DSP0266, Redfish Specification, https://www.dmtf.org/dsp/DSP0266

[2] DMTF DSP0268, Redfish Data Model Specification, https://www.dmtf.org/dsp/DSP0268

[3] DMTF DSP0272, Redfish Interoperability Profiles Specification, https://www.dmtf.org/dsp/DSP0272

[4] OCP Ethernet NIC Profile, https://github.com/opencomputeproject/HWMgmt-OCP-Profiles/blob/master/EthernetNIC/OCP_NIC.v1_0_0.json

[5] UEC, UE NIC Profile, https://github.com/ultraethernet/uec-management-public/tree/main/Redfish/profiles

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.

[6] G. Schadow and C.J. McDonald, "The Unified Code for Units of Measure (UCUM) Specification, ver. 2.2," 17 June 2024. [Online]. Available: https://ucum.org.
[7] UEC, UE Specification v1.1, https://ultraethernet.org/wp-content/uploads/sites/20/2026/10/UE-Specification-1.1.pdf

3.2 Informative References

[8] UEC, Mockups for the UE Ethernet NIC endpoint management model, https://github.com/ultraethernet/uec-management-public/tree/main/Redfish/mockups/NIC
[9] DMTF DSP2066, Redfish Fabrics White Paper, https://www.dmtf.org/dsp/DSP2066
[10] DMTF DSP2051, Redfish Telemetry White Paper, https://www.dmtf.org/dsp/DSP2051
[11] DMTF DSP2050, Redfish Composability White Paper, https://www.dmtf.org/dsp/DSP2050
[12] DMTF DSP2062, Redfish Firmware Update White Paper, https://www.dmtf.org/dsp/DSP2062
[13] DMTF DSP2059, Redfish Certificate Management White Paper, https://www.dmtf.org/dsp/DSP2059
[14] DMTF DSP2065, Redfish Message Registry Guide, https://www.dmtf.org/dsp/DSP2065
[15] DMTF Redfish Interop Validator, https://github.com/DMTF/Redfish-Interop-Validator
[16] Open Compute Project Profiles, https://github.com/opencomputeproject/HWMgmt-OCP-Profiles)
[17] Open Process Automation (OPAF) Profiles, http://open-process-automation.projects.opengroup.org/twg/part-5-system-management/profiles
[18] DMTF DSP8013, Redfish Interoperability Profiles Bundle, https://www.dmtf.org/dsp/DSP8013
[19] Redfish Mockup Server, https://github.com/DMTF/Redfish-Mockup-Server
[20] OCP Usage Guide for Managing the Ethernet NIC, https://www.opencompute.org/documents/usageguide-ethernetnic-1-0-0-final-pdf
[21] DMTF 2066, Redfish Fabrics White Paper, https://www.dmtf.org/dsp/DSP2066
[22] DMTF 2051, Redfish Telemetry White Paper, https://www.dmtf.org/dsp/DSP2051
[23] DMTF 2050, Redfish Composability White Paper, https://www.dmtf.org/dsp/DSP2050
[24] DMTF 2062, Redfish Firmware Update White Paper, https://www.dmtf.org/dsp/DSP2062
[25] DMTF 2062, Redfish Certificate Management White Paper, https://www.dmtf.org/dsp/DSP2059

Ultra Ethernet Consortium - Proprietary and Confidential © 2026. All rights reserved.