Document Number: DSP1120

Date: 2015-06-19

Version: 1.0.0a

Network Management - Tunnel Management Profile

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, it may still change, perhaps profoundly. This document is available for public review and comment until superseded.

Provide any comments through the DMTF Feedback Portal:
http://www.dmtf.org/standards/feedback

Supersedes: None

Document Type: Specification

Document Class: Normative

Document Status: Work in Progress

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Network Management* - Tunnel Management Profile (DSP 1120) was prepared by the Network Services Management Working Group of the DMTF.

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

Acknowledgments

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

Editors:

- John Parchem – DMTF Fellow
- Bhumip Khasnabish - ZTE Corporation

Contributors:

- John Crandall – Brocade Communications System
- Bhumip Khasnabish - ZTE Corporation
- Lawrence Lamers – VMware
- John Parchem – DMTF Fellow
- Shishir Pardikar – Citrix
- Hemal Shah – Broadcom Corporation
- Eric Wells – Hitachi
- Alex Zhdankin – Cisco Systems

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage Network Services and the associated configuration information. The target audience for this specification is implementers who are writing CIM-based providers or consumers of management interfaces that represent the component described in this document.

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.
- ABNF rules are in `monospaced font`.

Network Management - Tunnel Management Profile

1 Scope

The *Network Management - Tunnel Management Profile* is a profile that will specify the CIM schema and use cases associated with the general and common aspects of tunneling management. In general, the tunnel interface includes switch virtual interface and loopback interface.

2 Normative references

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

DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

DMTF DSP0200, *CIM Operations over HTTP 1.3,*
http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

DMTF DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

DMTF DSP1033, *Profile Registration Profile 1.0,*
http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

DMTF DSP1097, *Virtual Ethernet Switch Profile 1.1*,
http://dmtf.org/sites/default/files/standards/documents/DSP1097_1.1.0.pdf

DMTF DSP1036 *IP Interface Profile 1.1.1,*
http://www.dmtf.org/sites/default/files/standards/documents/DSP1036_1.1.1.pdf

IETF WG, Network Virtualization Overlays (NVO3), Sept. 2011,
https://datatracker.ietf.org/wg/nvo3/charter/

IETF Draft, A Stateless Transport Tunneling (STT) Protocol, April 2014,
http://tools.ietf.org/html/draft-davie-stt-06/

IETF Draft, Network Virtualization using Generic Routing Encapsulation (NVGRE), July 2014,
http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-05/

IETF RFC 7348, Virtual eXtensible Local Area Network (VXLAN), August 2014,
http://tools.ietf.org/html/rfc7348/

IETF Draft, Generic Network Virtualization Encapsulation (Geneve), August 2014,
http://tools.ietf.org/html/draft-gross-geneve-01/

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

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, Annex H. The terms in parenthesis 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, Annex H 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 5.

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.

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

3.1

conditional

indicates requirements to be followed strictly to conform to the document when the specified conditions are met

3.2

mandatory

indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.3

optional

indicates a course of action permissible within the limits of the document

3.4

pending configuration

indicates the configuration that will be applied to an IP network connection the next time the IP network connection accepts a configuration

3.5

referencing profile

indicates a profile that owns the definition of this class and can include a reference to this profile in its “Referenced Profiles” table

3.6

unspecified

indicates that this profile does not define any constraints for the referenced CIM element or operation

4 Symbols and abbreviated terms

The abbreviations defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional abbreviations are used in this document.

4.1

Customer Network

The customer network in an overlay is the network as seen by a virtual machines or a physical server. In an overlay network the customer network traffic is the encapsulated payload in a packet on the provider network.

4.2

IP

Internet Protocol

4.3

Gateway

interconnects networks with different network protocol technologies or separate IP address networks by performing the required protocol or IP address mapping conversions.

4.4

IP Gateway

Uses the IP protocol as an underlay protocol to tunnel unroutable network segments, allowing the traffic to routed on a an IP network.

4.5

Tunnel

A Tunnel is a path across unroutable network segments. These can be segments on different IP networks or networks using different protocols.

4.6

IP Encapsulation

a method of creating IP packets in which logically separate networks are abstracted from their underlying structures by inclusion of inside network packet within higher level network packet.

4.7

Overlay Network\Customer

A network containing un-encapsulated network traffic from the perspective of the encapsulation interfaces specified in this profile.

4.8

Provider Network

The underlay network of a tunnel. The network contains the encapsulated network traffic from the perspective of the encapsulation gateways specified in this profile.

4.9

Network Overlay/Underlay

Network overlay/underlay allows encapsulation of one packet into another using "packet-in-a-packet" technique. The encapsulated packet is forwarded to an endpoint where it is de-encapsulated. Network overlay/underlay is commonly used to (a) support secure multi-tenancy and (b) extend one network across another

5 Synopsis

Profile name: Network Management - Tunnel Management Profile

Version: 1.0.0

Organization: DMTF

CIM Schema version: 2.42e

Central class: CIM_IPEncapsulationManagementSevice

Scoping class: CIM_ComputerSystem

The *Network Management - Tunnel Management Profile* is a profile that specifies the CIM schema and use cases associated with Tunneling Management where a layer 2 or layer 3 overlay network is carried over a tunnel interface where layer 3 is used as the tunnel underlay. This profile includes a specification of the IPEncapsulationGateway and their associated setting data interfaces among others.

Table 1 identifies profiles on which this profile has a dependency.

Table 1 – Referenced profiles

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None
Network Management	DMTF	1.0	Mandatory	None

6 Description

The *Network Management - Tunnel Management Profile* is a profile that will specify the CIM schema and use cases associated with the general and common aspects of Tunneling Management. This profile includes a specification of the CIM_IPEncapsulationManagementService and a set of associated CIM classes to configure and manage a Tunnel gateway.

6.1 Class diagram

Figure 1 represents the class schema for the *Network Management - Tunnel Management Profile*. For simplicity, the CIM_ prefix has been removed from the names of the classes.

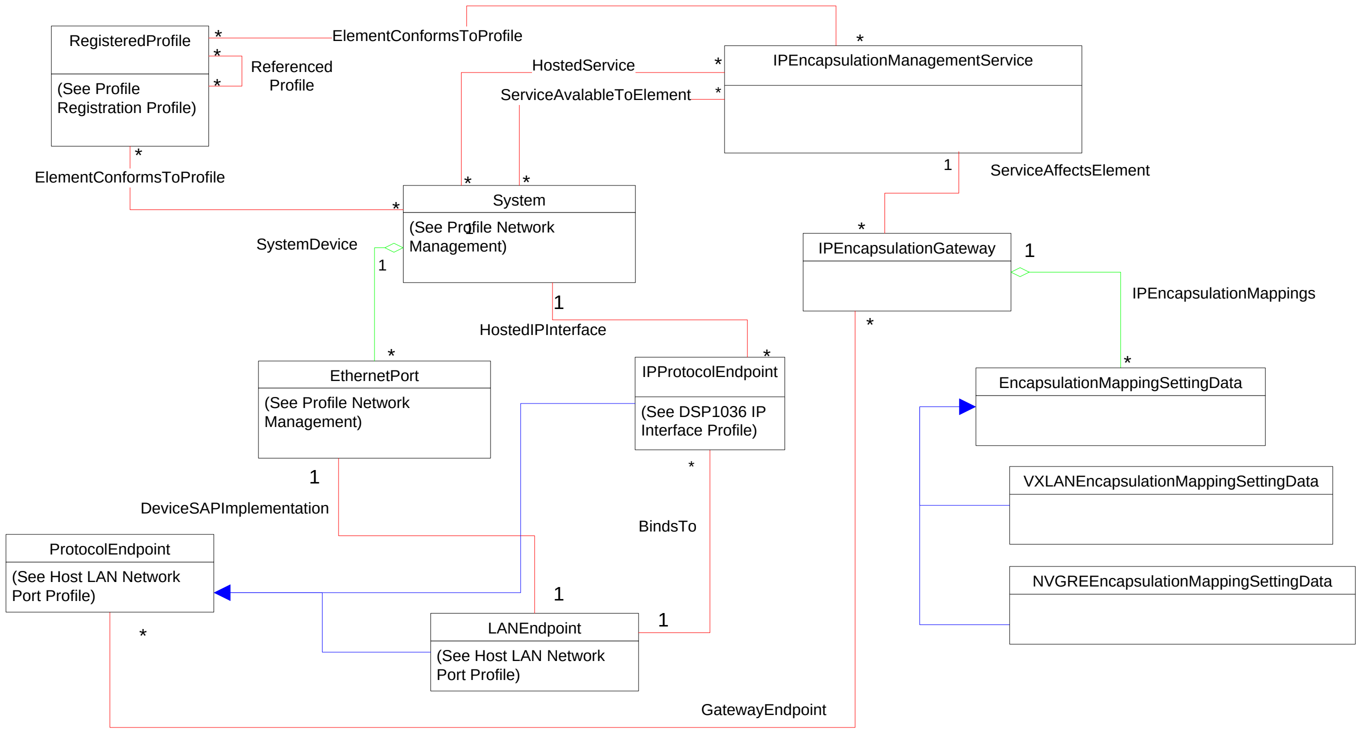
RegisteredProfile
(See Profile Registration Profile)
ElementConformsToProfile
Referenced Profile
IPEncapsulationManagementService
HostedService
ServiceAvalableToElement
ElementConformsToProfile
System
(See Profile Network Management)
ServiceAffectsElement
IPEncapsulationGateway
IPEncapsulationMappings
SystemDevice
HostedIPInterface
EthernetPort
(See Profile Network Management)
IPProtocolEndpoint
(See DSP1036 IP Interface Profile)
EncapsulationMappingSettingData
VXLANEncapsulationMappingSettingData
NVGREEncapsulationMappingSettingData
DeviceSAPImplementation
BindsTo
ProtocolEndpoint
(See Host LAN Network Port Profile)
LANEndpoint
(See Host LAN Network Port Profile)
GatewayEndpoint
1
*

Figure 1 – Network Management - Tunnel Management Profile

6.2 Tunneling Protocol Variants

A variety of tunneling protocols can be utilized in (virtual) overlay networks in order to extend disparate network segments between hosts (servers) for multi-tenant data center networks. These include Virtual Extensible LAN (VXLAN), Network Virtualization using Generic Encapsulation (NVGRE), state-less transport tunneling (STT) and IETF Network Virtualization Overlays 3 (NVO3). Currently this profile supports VXLAN and NVGRE although the profile generically supports other L2 and L3 tunneling protocols such as STT, LISP, L2TPv3, MPLS, GRE, VXLAN-GPE, GENEVE and GUE.

6.3 IPEncapsulationGateway

In this profile the IPEncapsulationGateway is used to provide the configuration for an instance pf an encapsulation service. This includes the ability to configure the endpoints that are on the provider network and the customer network and the virtual routing and forwarding tables that are to be used to determine the next hop routes required to route traffic between the two networks. An instance of IPEncapsulationGateway also hosts the collection of CIM_EncapsulationMappingSettingData that are used to describe the policy to map each tunnel.

6.4 CIM_EncapsulationMappingSettingData, CIM_NVGREEncapsulationSettingData, CIM_VXLANEncapsultionSettingData

CIM_EncapsulationSettingData represents a lookup record contained in a mapping table, represented by the aggregation IPEncapsulationMappings. This table of records provide the required information generally indexed off of the target IP address of an incoming packet from the provider network. It provides the information required to construct the encapsulation header for the underlay network. CIM_NVGREEncapsulationSettingData and CIM_VXLANEncapsulationSettingData specializes

CIM_EncapsulationSettingData to provide encapsulation information respectively for NVGRE and VXLAN overlay networks.

6.5 CIM_IPEncapsulationManagementService

The CIM_IPEncapsulationManagementService is the central class of this profile. The service has a set of extrinsic methods to control the creation and removal of the instances required to create an IP encapsulation gate.

7 Implementation

This clause details the requirements related to the arrangement of instances and the properties of instances for implementations of this profile

7.1 Representing IP encapsulation management services

7.1.1 CIM_IPEncapsulationManagementService

One or more instances of CIM_IPEncapsulationManagementService shall be instantiated.

These instances of CIM_IPEncapsulationManagementService shall be associated with an instance of the scoping CIM_System class through an instance of CIM_HostedService.

The instances of the CIM_IPEncapsulationManagementService class shall also be associated to each CIM_System subclass instance that may be used as the TargetInterface parameter of its AddIPEncapsulationGateway () method through an instance of CIM_ServiceAvalableToElement.

IPEncapsulationGateway instances managed by or created through the use of an instance of CIM_IPEncapsulationManagementService shall be associated to the CIM_IPEncapsulationManagementService instance through an instance of CIM_ServiceAffectsElement.

7.2 CIM_IPEncapsulationGateway

Instances of CIM_IPEncapsulationGateway created as part of the execution of AddIPEncapsulationGateway() method shall be associated with the instance of CIM_IPEncapsulationManagementService from which the method call was made through an instance CIM_ServiceAffectsElement.

An instance of CIM_IPEncapsulationGateway directly associated to an instance CIM_ProtocolEndpoint as configured through the CASourceEndpoint or the PASourceEndpoint property in the representative CIM_IPEncapsulationGateway instance shall be associated through an instance of a CIM_GatewayEndpoint association.

Instances of CIM_IPProtocolEndpoint created as part of the execution of AddIPEncapsulationGateway() method shall be associated to the instantiated instance of CIM_IPEncapsulationGateway from the same method call with an instance of the CIM_GatewayEndpoint.

7.3 Representing the Protocol Endpoints of an Encapsulation gateway.

Each instance of CIM_EncapsulationGateway shall have two associated instances of CIM_ProtocolEndpoint or a subclass of CIM_ProtocolEndpoint one representing an endpoint that is a part of the customer network and one representing an endpoint that is a part of the provider network. These instances are associated through the CIM_GatewayConnection association as described in 7.2. The association is made to these endpoints after the successful completion of a CIM_IPEncapsulationManagementService. AddIPEncapsulationGateway() method.

If an instance of CIM_IPProtocolEndpoint is instantiated through a successful completion of a CIM_IPEncapsulationManagementService. AddIPEncapsulationGateway() method where either a CAIPEndpoint or an PAIPEndpoint parameter was populated, that instance shall be associated through an instance of CIM_HostedIPInterface to the instance CIM_System that was specified in the TargetSystem parameter of the method call.

7.4 Representing an encapsulation mapping table

7.4.1 CIM_EncapsulationMappingSettingData

Instances of CIM_EncapsulationMappingSettingData created as a result of the AddIPEncapsulationGateway() or AddEncapsulationMappingSettingData () method shall be associated to the instance of CIM_IPEncapsulationGateway contained in the EncapsulationGateway parameter of the respective method through an aggregation instance of CIM_IPEncapsulationMappings.

8 Methods

This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM elements defined by this profile.

8.1 Extrinsic Methods

If synchronous execution of a method succeeds, the implementation shall set a return value of 0 (Completed with No Error).

If synchronous execution of a method fails, the implementation shall set a return value of 2 (Failed) or a more specific return code as specified with the respective method.

If a method is executed as an asynchronous task, the implementation shall perform all of the following actions:

- Set a return value of 4096 (Job Started).
- Set the value of the Job output parameter to refer to an instance of the CIM_ConcreteJob class that represents the asynchronous task.
- Set the values of the JobState and TimeOfLastStateChange properties in that instance to represent the state and last state change time of the asynchronous task.

In addition, the implementation may present state change indications as task state changes occur.

If the method execution as an asynchronous task succeeds, the implementation shall perform all of the following actions:

- Set the value of the JobState property to 7 (Completed).
- Provide an instance of the CIM_AffectedJobEntity association with property values set as follows:
  - The value of the AffectedElement property shall refer to the object that represents the top-level entity that was created or modified by the asynchronous task. For example, for the CIM_IPConfigurationService. AddIPProtocolEndpoint() method, this is an instance of the CIM_IPProtocolEndpoint class
  - The value of the AffectingElement property shall refer to the instance of the CIM_ConcreteJob class that represents the completed asynchronous task.
  - The value of the first element in the ElementEffects[ ] array property (ElementEffects[0]) shall be set to 5 (Create) for the CIM_IPConfigurationService. AddIPProtocolEndpoint() method. Otherwise, this value shall be 0 (Unknown).

If the method execution as an asynchronous task fails, the implementation shall set the value of the JobState property to 9 (Killed) or 10 (Exception).

8.1.1 Job parameter

The implementation shall set the value of the Job parameter as a result of an asynchronous execution of a method of the CIM_IPConfigurationService as follows:

- If the method execution is performed synchronously, the implementation shall set the value to NULL.
- If the method execution is performed asynchronously, the implementation shall set the value to refer to the instance of the CIM_ConcreteJob class that represents the asynchronous task.

8.1.2 CIM_IPConfigurationService. AddIPEncapsulationGateway()

The implementation of the AddIPEncapsulationGateway ( ) method is optional, the provisions in this sub clause apply in addition behavior applicable to all extrinsic methods as specified in 8.1.

The successful execution of the AddIPEncapsulationGateway ( ) method shall create an instance of the CIM_IPEncapsulationGateway class or a subclass of CIM_IPEncapsulationGateway and any required associations as described in the sub clauses 7.2 required to instantiate a complete encapsulation interface. In addition if the optional method parameter EncapsulationMappings is populated with corresponding instances of the embedded CIM_EncapsulationMappingSettingData classes the instantiated instances should be associated with the newly instantiated IPEncapsulationGateway through an instance of CIM_IPEncapsulationMappings. If the optional method parameters PAIPEndpoint or CAIPEndpoint are populated with corresponding instances of the embedded CIM_IPProtocolEndpoint classes the instantiated instances should be associated with the newly instantiated IPEncapsulationGateway through an instance of CIM_GatewayEndpoint.

Table 2 contains requirements for parameters of this method.

Table 2 – AddIPEncapsulationGateway ( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	TargetSystem	CIM_System REF	See 8.1.2.1.1
IN	EncapsulationGateway	String	See 8.1.2.1.2
IN	PAIPEndpoint	String	See 8.1.2.1.2
IN	CAIPEndpoint	String	See 8.1.2.1.4
IN	EncapsulationMappings	String[]	See 8.1.2.1.5
OUT	ResultingGateway	IPEncapsulationGateway REF	See 8.1.2.1.5
OUT	Job	CIM_ConcreteJob REF	See 8.1.1

8.1.2.1.1 TargetSystem

A required reference to a, system or network. The supported target interfaces for a CIM_IPEncapsulationGateway class or subclass supported should be as described in the sub clauses of **Error! Reference source not found.**.

8.1.2.1.2 EncapsulationGateway

A required string containing an embedded instance of the class-subclass of CIM_IPEncapsulationGateway describes the initial configuration of the resulting

CIM_IPEncapsulationGateway instance. The populated properties of the embedded instance should not contain key properties, and any key property values may be ignored.

8.1.2.1.3 PAIPEndpoint

An optional string containing an embedded instances of the class-subclass of CIM_IPProtocolEndpoint that describes the initial configuration of an CIM_IPProtocolEndpoint that is on the provider network. The populated properties of the embedded instance should not contain key properties, and any key property values may be ignored.

8.1.2.1.4 CAIPEndpoint

An optional string containing an embedded instances of the class-subclass of CIM_IPProtocolEndpoint that describes the initial configuration of an CIM_IPProtocolEndpoint that is on the customer network. The populated properties of the embedded instance should not contain key properties, and any key property values may be ignored.

8.1.2.1.5 EncapsulationMapping[]

An optional array of strings containing embedded instances of the class-subclass of CIM_EncapsulationMappingSettingData that describes entries in a mapping table used by the encapsulation gateway to provide the encapsulation header information used by the gateway. The populated properties of the embedded CIM_EncapsulationMappingSettingData instances should not contain key properties, and any key property values may be ignored.

8.1.2.1.6 ResultingInterface

If the creation if the CIM_IPEncapsulationGateway is successfully, the resultant instance of the class CIM_IPEncapsulationGateway that represents the instantiated instance of CIM_IPEncapsulationInteface is returned.

8.1.2.1.7 Job

See 8.1.1

8.1.3 CIM_IPConfigurationService. AddIPEncapsulationMappings()

The implementation of the AddIPEncapsulationMappings ( ) method is optional, the provisions in this sub clause apply in addition behavior applicable to all extrinsic methods as specified in 8.1.

The successful execution of the AddIPEncapsulationMappings ( ) method shall create or add to an array of instances of the CIM_EncapsulationMappingSettingData. The added instances of CIM_EncapsulationMappingSettingData shall be associated to the target IPEncapsulationGateway through an instance of CIM_IPEncapsulationMappings.

Table 3 contains requirements for parameters of this method.

Table 3 – AddIPEncapsulationMappings ( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	TargetInterface	CIM_IPEncapsulationGateway REF	See 8.1.3.1.1
IN	EncapsulationMappings	String[]	See 8.1.3.1.2
OUT	Job	CIM_ConcreteJob REF	See 8.1.3.1.3

8.1.3.1.1 TargetInterface

A required reference to an interface IPEncapsulationGateway class or subclass.

8.1.3.1.2 EncapsulationMapping[]

An optional array of strings containing embedded instances of the class-subclass of CIM_EncapsulationMappingSettingData that describes entries in a mapping table used by the encapsulation interface to provide the encapsulation header for the resultant IPEncapsulationGateway. The populated properties of the embedded CIM_EncapsulationMappingSettingData instances should not contain key properties, and any key property values may be ignored. The resulting CIM_EncapsulationMappingSettingData instance should be associated with the target instance of IPEncapsulationGateway configured in the EncapsulationGateway parameter through an instance of CIM_IPEncapsulationMapping.

8.1.3.1.3 Job

See 8.1.1

8.1.4 CIM_IPConfigurationService. RemoveIPEncapsulationGateway()

The implementation of the RemoveIPEncapsulationGateway( ) method is optional, the provisions in this sub clause apply in addition behavior applicable to all extrinsic methods as specified in 8.1.

The successful execution of the RemoveIPEncapsulationGateway ( ) method shall remove the instances referenced in the methods Gateway parameter and should remove any associated CIM_SettingData instances.

Table 4 contains requirements for parameters of this method.

Table 4 – RemoveIPEncapsulationGateway ( ) Method: Parameters

Qualifiers	Name	Type	*Description/Values*
IN	Gateway	CIM_IPIPEncapsulationGateway REF[]	See 8.1.4.1.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.1

8.1.4.1.1 Endpoint

An array of references to the pair ofinstances of the class CIM_IPIPEncapsulationGateway that shall be removed.

8.1.4.1.2 Job

See 8.1.1

8.1.5 CIM_IPConfigurationService. RemoveIPEncapsulationMappings()

The implementation of the RemoveIPEncapsulationMappings( ) method is optional, the provisions in this sub clause apply in addition to behavior applicable to all extrinsic methods as specified in 8.1.

The successful execution of the RemoveIPEncapsulationMappings ( ) method shall remove the instances referenced in the methods EncapsulationMapping parameter.

Table 5 contains requirements for parameters of this method.

Table 5 – RemoveIPEncapsulationMappings ( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Encapsulation Mappings	CIM_EncapsulationMappingSettingData REF[]	See 8.1.5.1.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.5.1.2

8.1.5.1.1 EncapsulationMappings

An array of references to the pair ofinstances of the class CIM_EncapsulationMappingSettingData that shall be removed.

8.1.5.1.2 Job

See 8.1.1

8.2 Profile conventions for operations

For each profile class (including associations), the implementation requirements for operations, including those in the following default list, are specified in class-specific subclauses of this clause.

The default list of operations is as follows:

- GetInstance
- EnumerateInstances
- EnumerateInstanceNames
- Associators
- AssociatorNames
- References
- ReferenceNames

8.3 CIM_GatewayEndpoint

Table 6 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 6, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 6 – Operations: CIM_GatewayEndpoint

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.4 CIM_BindsTo

Table 7 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 6, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

Table 7 – Operations: CIM_BindsToLANEndpoint

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.5 CIM_HostedService

Table 8 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 8, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 8 – Operations: CIM_HostedService

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.6 CIM_HostedIPInterface

Table 9 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 9 – Operations: CIM_HostedIPInterface

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.7 CIM_IPEncapsulationManagementService

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

8.8 IPEncapsulationGateway

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

9 Use cases

This clause contains object diagrams and use cases for the *Network Management - Tunnel Management Profile*.

9.1 Profile Registration

The object diagram in Figure 2 shows one possible method for advertising profile conformance. The instances of CIM_RegisteredProfile are used to identify the version of the Network Management - Tunnel Management Profile with which an instance of CIM_IPEncapsulationManagementService is conformant. An instance of CIM_RegisteredProfile exists for each profile that is instrumented in the system. One instance of CIM_RegisteredProfile identifies the System conforming to the Network Management Profile. The other instance identifies an instance of CIM_IPEncapsulationManagmentService. The CIM_IPEncapsulationManagmentService instance is scoped to an instance of CIM_System. This instance of CIM_System is conformant with the DMTF Network Management Profile version 1.0.0 as indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance.

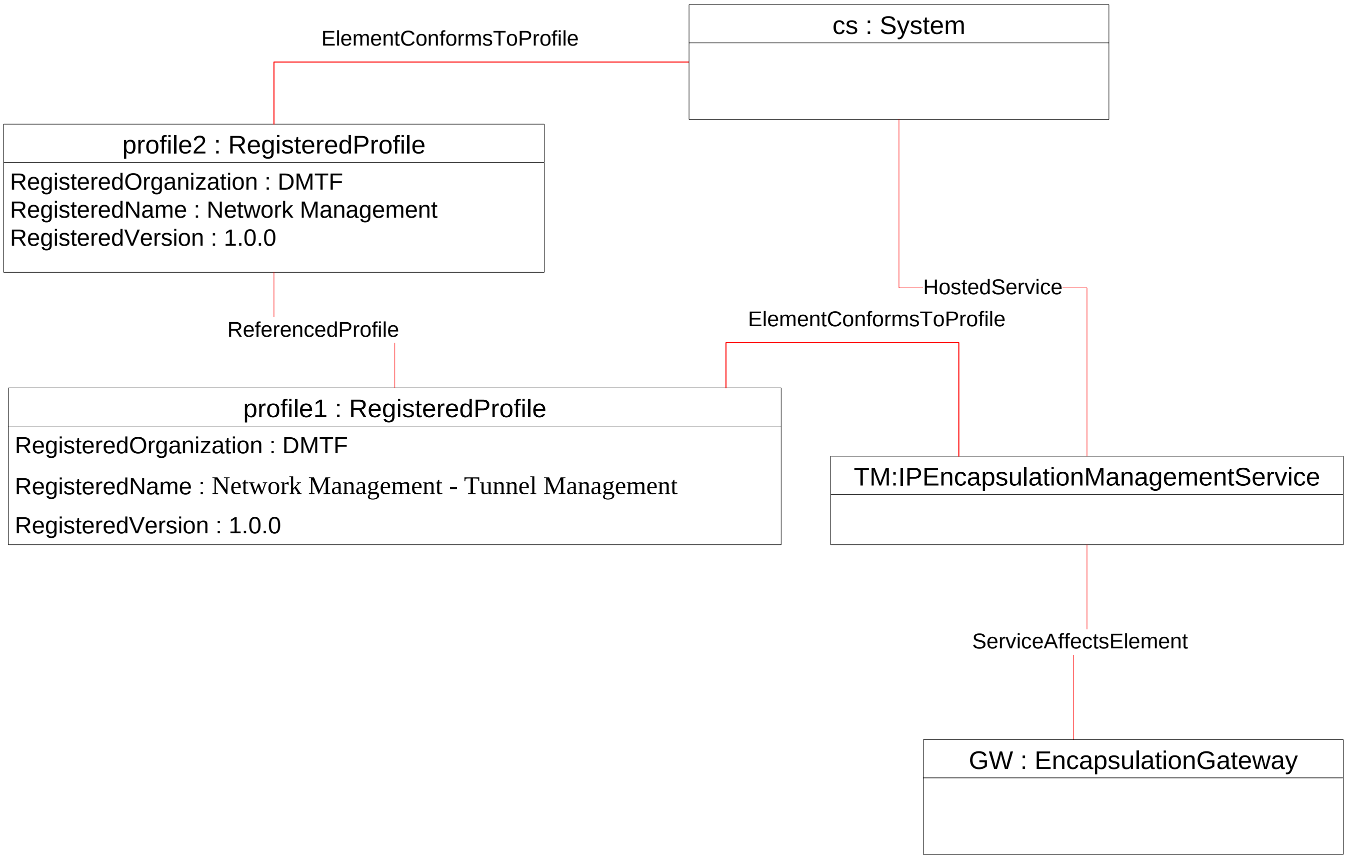
cs : System
ElementConformsToProfile
profile2 : RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Network Management
RegisteredVersion : 1.0.0
ReferencedProfile
HostedService
ElementConformsToProfile
profile1 : RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Network Management - Tunnel Management
RegisteredVersion : 1.0.0
TM:IPEncapsulationManagementService
ServiceAffectsElement
GW : EncapsulationGateway

Figure 2 – Registered profile

9.2 L2 NVGre Tunnel Gateway

The object diagram shown in Figure 3 contains the basic elements used to model a simple L2 NVGRE tunnel gateway. The gateway as shown could be an example of an NVGRE tunnel gateway as part of a virtual Ethernet switch. CIM_EthernetPort: E0/4 is a port available to a virtual computer system and CIM_EthernetPort: E1/1 is the uplink to the physical Ethernet switch. In this example the virtual system port (E0/4) would be on the customer network and the uplink port (E1/1) would be on the provider network.

This gateway takes all network traffic from CIM_EthernetPort: E0/4 and encapsulates it based on the mappings found in the instances of CIM_EncapsulationMappingSettingData and sends the encapsulated traffic out on the provider network through CIM_EthernetPort: E1/1. Any traffic from the provider network is de-capsulated and sent to the virtual machine through CIM_EthernetPort: E0/4.

This is a very simple instance diagram, not shown are many of the required properties of the relative profiles for the objects shown.

The IPEncapsulationGateway was created with a CIM_IPConfigurationService.AddEncapsulationGateway() method with the following parameters.

- TargetInterface – WBEM URI reference to CIM_ComputerSystem:CS0
- EncapsulationGateway
    - ○ Embedded Instance of IPEncapsulationGateway {
      TunnelMode=4 (NVGRE)
      CASourceInterface = WBEM URI reference to CIM_EthernetPort: E0/4
      PASourceInterface = WBEM URI reference to CIM_EthernetPort: E1/1 }

- EncapsulationMappings
    - ○ Embedded Instance of EncapsulationMappingSettingData {
      CustomerAddress 10.1.0.125
      ProviderAddress 198.168.56.255
      MACAddress = 01:23:45:67:89:ab
      VSID = GUID for Virtual Subnet ID.

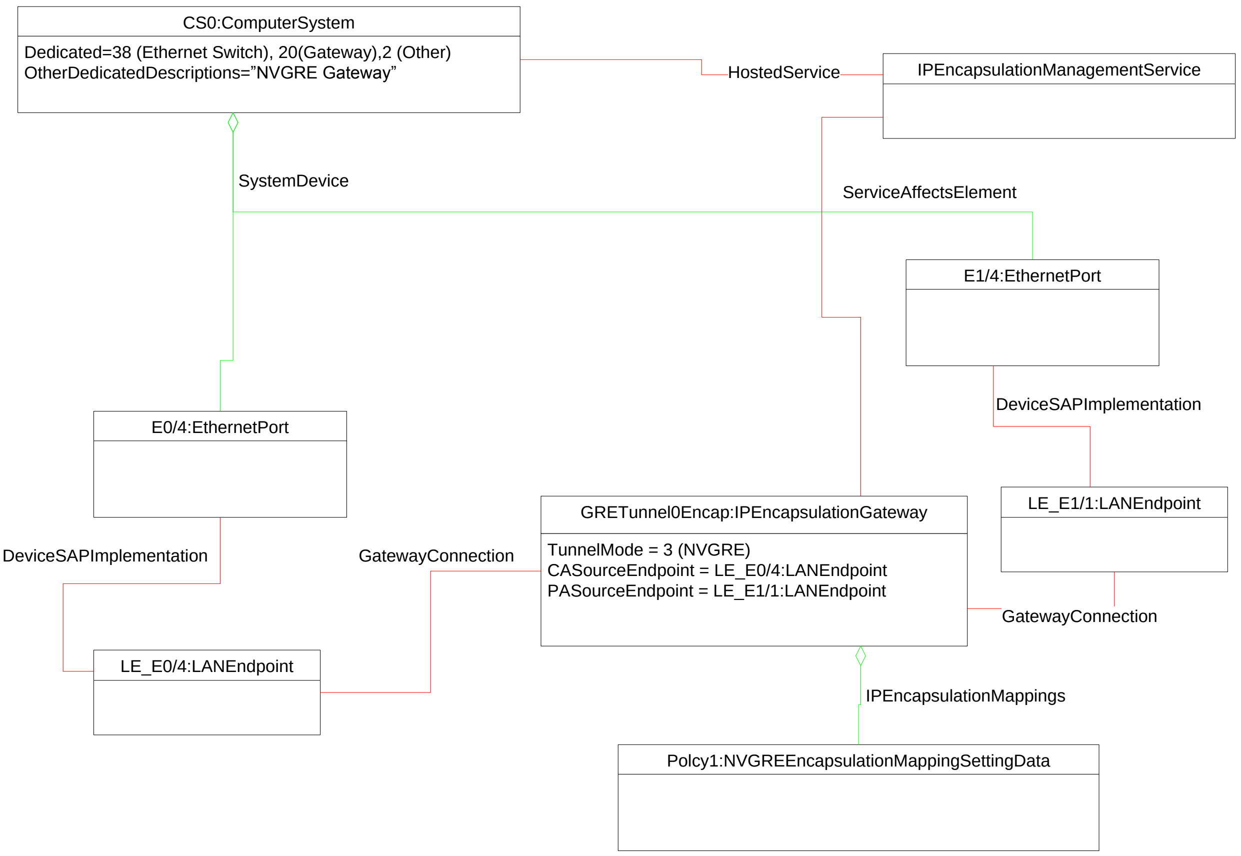
CS0:ComputerSystem
Dedicated=38 (Ethernet Switch), 20(Gateway),2 (Other)
OtherDedicatedDescriptions="NVGRE Gateway"
HostedService
IPEncapsulationManagementService
SystemDevice
ServiceAffectsElement
E1/4:EthernetPort
DeviceSAPImplementation
E0/4:EthernetPort
LE_E1/1:LANEndpoint
GRETunnel0Encap:IPEncapsulationGateway
TunnelMode = 3 (NVGRE)
CASourceEndpoint = LE_E0/4:LANEndpoint
PASourceEndpoint = LE_E1/1:LANEndpoint
DeviceSAPImplementation
GatewayConnection
GatewayConnection
LE_E0/4:LANEndpoint
IPEncapsulationMappings
Polcy1:NVGREEncapsulationMappingSettingData

Figure 3 – NVGRE Tunnel Gateway

9.3 Routed NVGre Tunnel Gateway

The instance diagram shown in contains the basic elements used to model a routable NVGRE tunnel gateway. The gateway as shown could be an example of an NVGRE tunnel gateway as part of a network router. While this is a simplified instance diagram, the example shows a port E0/4 that is on the customer network and is configured with an instance of CIM_SwitchVirtualInterface an IP endpoint that is configured through VLAN encapsulation to accept traffic tagged with a specific VALN ID. Another port E1/4 is an IP enabled port on the provider network. The customer network and the information required to determine the next hop routes are represented in the Virtual Routing and Forwarding table VRF1_CA:VirtualRoutingAndForwarding. The provider network and the information required to determine the next hop routes for the provider network are represented in the Virtual Routing and Forwarding table VRF0_PA:VirtualRoutingAndForwarding.

LE_E0/4:SwitchedVirtualInterface, an IP protocol endpoint for port E0/1, is a member of the VRF1_CA:VirtualRoutingAndForwarding table. IPE_E1/4:IPProtocolEndpoint is a member of the VRF0_PA:VirtualRoutingAndForwarding table.

The router has an IPEncapsultaionManagementService, ServiceNVGRE. This example shows the result of a AddEncapsulationGateway() method call on that service that creates a routable IPEncapsulationGateway with IPProtocolEndpoint instances for both the customer and the provider networks respectively GRETunnel0CA and GRETunnel0PA. As shown in Figure 4 these IP protocol endpoints are also members of their respective virtual routing and formatting tables.

This example shows the method call used to add a NVGRE IPEncapsulationGateway

The IPEncapsulationGateway was created with a CIM_IPConfigurationService.AddEncapsulationGateway() method with the following parameters.

- TargetInterface – WBEM URI reference to CIM_ComputerSystem:CS0
- EncapsulationGateway
  - Embedded Instance of IPEncapsulationGateway {
    TunnelMode=4 (NVGRE)
    CASourceInterface = null
    PASourceInterface = null
    CAVRFContext = WBEM URI reference to VRF1_CA:VirtualRoutingAndForwarding
    PAVRFContext = WBEM URI reference to VRF0_PA:VirtualRoutingAndForwarding

- CAEndpoint
  - Embedded Instance of CIM_IPProtocolEndpoint {
    IPv4Address=10.1.0.0
    ProtocolIFType=4060}

- PAEndpoint
  - Embedded Instance of CIM_IPProtocolEndpoint {
    IPv4Address=198.168.0.0
    ProtocolIFType=4060}

- EncapsulationMappings
  - CustomerAddress 10.1.0.125
    ProviderAddress 198.168.56.255
    MACAddress = 01:23:45:67:89:ab
    VSID = GUID for Virtual Subnet ID.

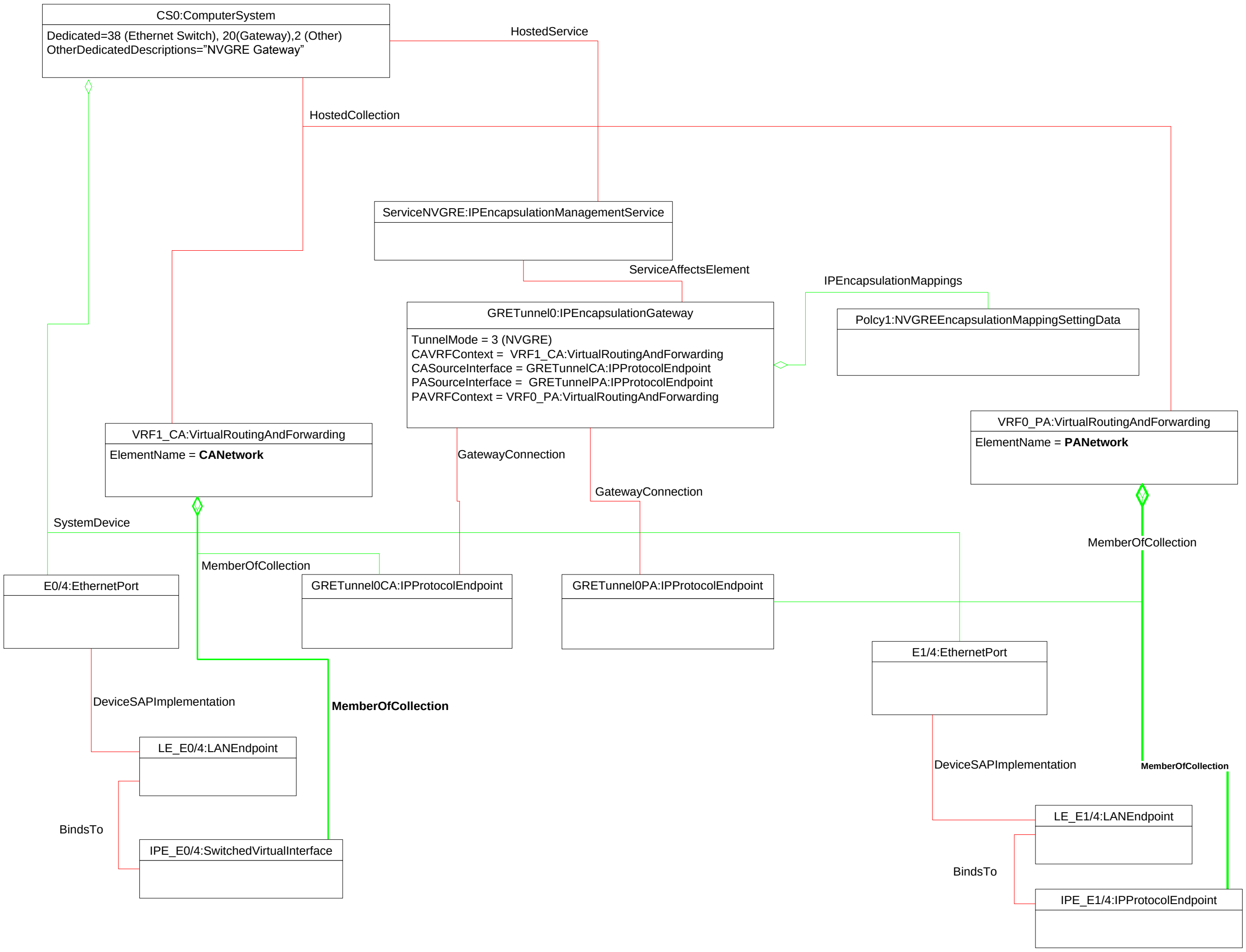
CS0:ComputerSystem
Dedicated=38 (Ethernet Switch), 20(Gateway),2 (Other)
OtherDedicatedDescriptions="NVGRE Gateway"
HostedService
HostedCollection
ServiceNVGRE:IPEncapsulationManagementService
ServiceAffectsElement
IPEncapsulationMappings
GRETunnel0:IPEncapsulationGateway
TunnelMode = 3 (NVGRE)
CAVRFContext = VRF1_CA:VirtualRoutingAndForwarding
CASourceInterface = GRETunnelCA:IPProtocolEndpoint
PASourceInterface = GRETunnelPA:IPProtocolEndpoint
PAVRFContext = VRF0_PA:VirtualRoutingAndForwarding
Polcy1:NVGREEncapsulationMappingSettingData
VRF1_CA:VirtualRoutingAndForwarding
ElementName = CANetwork
VRF0_PA:VirtualRoutingAndForwarding
ElementName = PANetwork
GatewayConnection
GatewayConnection
SystemDevice
MemberOfCollection
MemberOfCollection
E0/4:EthernetPort
GRETunnel0CA:IPProtocolEndpoint
GRETunnel0PA:IPProtocolEndpoint
E1/4:EthernetPort
DeviceSAPImplementation
MemberOfCollection
LE_E0/4:LANEndpoint
BindsTo
IPE_E0/4:SwitchedVirtualInterface
DeviceSAPImplementation
MemberOfCollection
LE_E1/4:LANEndpoint
BindsTo
IPE_E1/4:IPProtocolEndpoint

Figure 4 – NVGRE Routable Tunnel Gateway

10 CIM Elements

Table 10 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 10. Clauses 7 ("Implementation") and 8 ("Methods") may impose additional requirements on these elements.

Table 10 – CIM Elements: Network Management – Tunnel Management Profile

Element Name	Requirement	Description
Classes		
CIM_BindsTo	Optional	See DSP1036 IP Interface Profile 1.1.1
CIM_BindsToLANEndpoint	Optional	See DSP1036 IP Interface Profile 1.1.1
CIM_HostedService	Conditional	See **Error! Reference source not found.**
CIM_HostedIPInterface	Conditional	See **Error! Reference source not found.** nd **Error! Reference source not found.**
CIM_IPProtocolEndpoint	Conditional	See **Error! Reference source not found.**

Element Name	Requirement	Description
CIM_IPEncapsulationManagementService	Mandatory	See **Error! Reference source not found.**
CIM_RegisteredProfile	Optional	See
CIM_ServiceAffectsElement	Conditional	See **Error! Reference source not found.**
CIM_ServiceAvalableToElement	Conditional	See **Error! Reference source not found.**
IPEncapsulationGateway	Required	See **Error! Reference source not found.**
Indications		
None defined in this profile		

10.1 CIM_BindsToLANEndpoint

CIM_BindsToLANEndpoint relates the CIM_IPProtocolEndpoint instance with the CIM_LANEndpoint instance on which it depends. Table 11 provides information about the properties of CIM_BindsToLANEndpoint.

Table 11 – Class: CIM_BindsToLANEndpoint

Elements	Requirement	Description
Antecedent	Mandatory	**Key:** This shall be a reference to an instance of CIM_LANEndpoint. Cardinality 0..1
Dependent	Mandatory	**Key:** This shall be a reference to the Central Instance. Cardinality 1

10.2 CIM_BindsTo

CIM_BindsTo relates two pairs of CIM_ProtocolEndpoints together. Table 12 provides information about the properties of CIM_BindsTo.

Table 12 – Class: CIM_BindsTo

Elements	Requirement	Description
Antecedent	Mandatory	**Key:** This shall be a reference to an instance of CIM_EncapsulationGateways. Cardinality 1
Dependent	Mandatory	**Key:** This shall be a reference to the paired CIM_EncapsulationGateways 1

10.3 CIM_EncapsulationMappingSettingData

Contains one mapping lookup record for an IP encapsulation gateway. Table 13 provides information about the properties of CIM_EncapsulationSettingData.

Table 13 – Class: CIM_EncapsulationMappingSettingData

Elements	Requirement	Description
InstanceID	Mandatory	**Key**

10.4 CIM_NVGREEncapsulationMappingSettingData

Contains one mapping lookup record for an NVGRE IP encapsulation gateway. Table 14 provides information about the properties of CIM_NVGREEncapsulationSettingData.

Table 14 – Class: CIM_NVGREEncapsulationMappingSettingData

Elements	Requirement	Description
InstanceID	Mandatory	**Key**

10.5 CIM_VXLANEncapsulationMappingSettingData

Contains one mapping lookup record for an VXLAN IP encapsulation gateway. Table 15 provides information about the properties of CIM_VXLANEncapsulationSettingData.

Table 15 – Class: CIM_VXLANEncapsulationMappingSettingData

Elements	Requirement	Description
InstanceID	Mandatory	**Key**

10.6 CIM_HostedService

CIM_HostedService relates the CIM_IPEncapsulationManagmentService instance to its scoping CIM_System instance. Table 16 provides information about the properties of CIM_HostedService.

Table 16 – Class: CIM_HostedService

Elements	Requirement	Description
Antecedent	Mandatory	**Key:** This shall be a reference to the Central Instance. Cardinality 1
Dependent	Mandatory	**Key:** This shall be a reference to an instance of CIM_IPEncapsulationManagementService. Cardinality *

10.7 CIM_IPEncapsulationManagementService

CIM_IPEncapsulationManagementService provides the methods to create and delete an encapsulation gateway interface. Table 17 provides information about the properties of CIM_IPEncapsulationManagementService.

Table 17 – Class: CIM_IPEncapsulationManagmentService

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	Pattern ".*"
AddIPEncapsulationGateway( )	Optional	See 8.1.2.
RemoveIPEncapsulationGateway( )	Optional	See 8.1.4
AddIPEncapsulationMappings()	Optional	See 8.1.3
RemoveIPEncapsulationMappings()	Optional	See 8.1.5

10.8 CIM_IPProtocolEndpoint

CIM_IPProtocolEndpoint represents an IP interface that is associated with an IP encapsulation gateway or an Ethernet interface. Table 18 provides information about the properties of CIM_IPProtocolEndpoint.

Table 18 – Class: CIM_IPProtocolEndpoint

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
NameFormat	Mandatory	See DSP1036 IP Interface Profile 1.1.1
ProtocolIFType	Mandatory	See DSP1036 IP Interface Profile 1.1.1
RequestedState	Mandatory	See DSP1036 IP Interface Profile 1.1.1
EnabledState	Mandatory	See DSP1036 IP Interface Profile 1.1.1
ElementName	Mandatory	See DSP1036 IP Interface Profile 1.1.1
RequestStateChange( )	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv4Address	Conditional	See DSP1036 IP Interface Profile 1.1.1
SubnetMask	Conditional	See DSP1036 IP Interface Profile 1.1.1
AddressOrigin	Mandatory	See DSP1036 IP Interface Profile 1.1.1
IPv6Address	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv6AddressType	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv6SubnetPrefixLength	Conditional	See DSP1036 IP Interface Profile 1.1.1

10.9 IPEncapsulationGateway

IPEncapsulationGateway represents the either the encapsulation or the decapsulation IP encapsulation interface used to route to connect two disjoined IP networks. Table 19 provides information about the additional properties of IPEncapsulationGateway that are in addition to those in CIM_IPProtocolEndpoint 10.7.

Table 19 – Class: IPEncapsulationGateway

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	Pattern ".*"
TunnelMode	Mandatory	**See** Error! Reference source not found.
CAVRFContext	Optional	**See** Error! Reference source not found.
CASourceInterface	Required	**See** Error! Reference source not found.
PAVRFContext	Optional	
PASourceInterface	Required	**See** Error! Reference source not found.

10.10 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Network Management - Tunnel Management Profile* in order for a client to determine whether an instance of CIM_IPProtocolEndpoint is conformant with this profile. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 20, the behavior of the CIM_RegisteredProfile instance is in accordance with the *Profile Registration Profile*.

Table 20 – Class: CIM_RegisteredProfile

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Tunnel Management".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

ANNEX A (informative)

Change log

Version	Date	Description
1.0.0a	2015-06-19	DMTF Work in Progress

Bibliography