



1
2
3
4
5

Document Identifier: DSP1120

Date: 2018-09-07

Version: 1.0.0

6 **Network Management - Tunnel Management**
7 **Profile**

8 **Supersedes: None**

9 **Document Class: Normative**

10 **Document Status: Published**

11 **Document Language: en-US**

12 Copyright Notice

13 Copyright © 2018 DMTF. All rights reserved.

14 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
15 management and interoperability. Members and non-members may reproduce DMTF specifications and
16 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to
17 time, the particular version and release date should always be noted.

18 Implementation of certain elements of this standard or proposed standard may be subject to third party
19 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations
20 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,
21 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or
22 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to
23 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,
24 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or
25 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any
26 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent
27 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is
28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party
29 implementing the standard from any and all claims of infringement by a patent owner for such
30 implementations.

31 For information about patents held by third-parties which have notified the DMTF that, in their opinion,
32 such patent may relate to or impact implementations of DMTF standards, visit
33 <http://www.dmtf.org/about/policies/disclosures.php>.

34 This document's normative language is English. Translation into other languages is permitted.

35

CONTENTS

36	Foreword	5
37	Introduction.....	6
38	1 Scope	7
39	2 Normative references	7
40	3 Terms and definitions	8
41	4 Symbols and abbreviated terms.....	9
42	5 Synopsis	10
43	6 Description	10
44	6.1 Class diagram	10
45	6.2 Tunneling protocol variants.....	11
46	6.3 IPEncapsulationInterface	11
47	6.4 CIM_EncapsulationMappingSettingData	11
48	6.5 CIM_IPEncapsulationManagementService	12
49	7 Implementation	12
50	7.1 Representing IP encapsulation management services	12
51	7.1.1 CIM_IPEncapsulationManagementService	12
52	7.2 CIM_IPEncapsulationInterface	12
53	7.3 Representing the Protocol Endpoints of an encapsulation gateway	12
54	7.4 Representing an encapsulation mapping table	13
55	7.4.1 CIM_EncapsulationMappingSettingData	13
56	8 Methods.....	13
57	8.1 Extrinsic methods.....	13
58	8.1.1 Job parameter.....	14
59	8.1.2 CIM_IPConfigurationService.AddIPEncapsulationInterface()	14
60	8.1.3 CIM_IPConfigurationService.AddIPEncapsulationMappings().....	15
61	8.1.4 CIM_IPConfigurationService.RemoveIPEncapsulationInterface()	16
62	8.1.5 CIM_IPConfigurationService.RemoveIPEncapsulationMappings().....	16
63	8.2 Profile conventions for operations	17
64	8.3 CIM_GatewayEndpoint.....	17
65	8.4 CIM_BindsTo	18
66	8.5 CIM_HostedService	18
67	8.6 CIM_HostedIPInterface.....	18
68	8.7 CIM_IPEncapsulationManagementService	19
69	8.8 IPEncapsulationInterface	19
70	9 Use cases.....	20
71	9.1 Profile Registration.....	20
72	9.2 L2 NVGRE tunnel gateway	21
73	9.3 Routed NVGRE tunnel gateway	22
74	10 CIM Elements	24
75	10.1 CIM_BindsToLANEndpoint.....	25
76	10.2 CIM_BindsTo	25
77	10.3 CIM_EncapsulationMappingSettingData	25
78	10.4 CIM_HostedService	26
79	10.5 CIM_IPEncapsulationManagementService	26
80	10.6 CIM_IPProtocolEndpoint	26
81	10.7 CIM_IPEncapsulationInterface	27
82	10.8 CIM_RegisteredProfile.....	28
83	ANNEX A (informative) Change log.....	29

84

85 **Figures**

86	Figure 1 – Network Management - Tunnel Management Profile	11
87		

88 **Tables**

89	Table 1 – Referenced profiles	10
90	Table 2 – AddPEncapsulationInterface () Method: Parameters	14
91	Table 3 – AddIPEncapsulationMappings () Method: Parameters	15
92	Table 4 – RemovePEncapsulationInterface() Method: Parameters	16
93	Table 5 – RemovePEncapsulationMappings() Method: Parameters	17
94	Table 6 – Operations: CIM_GatewayEndpoint	17
95	Table 7 – Operations: CIM_BindsToLANEndpoint	18
96	Table 8 – Operations: CIM_HostedService	18
97	Table 9 – Operations: CIM_HostedIPInterface	18
98	Table 10 – CIM Elements: Network Management – Tunnel Management Profile	24
99	Table 11 – Class: CIM_BindsToLANEndpoint	25
100	Table 12 – Class: CIM_BindsTo	25
101	Table 13 – Class: CIM_EncapsulationMappingSettingData	25
102	Table 14 – Class: CIM_HostedService	26
103	Table 15 – Class: CIM_IPEncapsulationManagementService	26
104	Table 16 – Class: CIM_IPProtocolEndpoint	26
105	Table 17 – Class: CIM_IPEncapsulationInterface	27
106	Table 18 – Class: CIM_RegisteredProfile	28
107		

108

Foreword

109 The *Network Management - Tunnel Management Profile* (DSP1120) was prepared by the Network
110 Services Management Working Group of the DMTF.

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

113 Acknowledgments

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

115 Editors:

- 116 • John Parchem – DMTF Fellow

117 Contributors:

- 118 • John Crandall – Brocade Communications System
- 119 • Bhumip Khasnabish - ZTE Corporation
- 120 • Lawrence Lamers – VMware
- 121 • John Parchem – DMTF Fellow
- 122 • Shishir Pardikar – Citrix
- 123 • Hemal Shah – Broadcom Corporation
- 124 • Eric Wells – Hitachi
- 125 • Alex Zhdankin – Cisco Systems

126

127

Introduction

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

133 Document conventions

134 Typographical conventions

135 The following typographical conventions are used in this document:

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

138

139 Network Management - Tunnel Management Profile

140 1 Scope

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

144 2 Normative references

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

149 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
150 http://www.dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

151 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
152 http://www.dmtf.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

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

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

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

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

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

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

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

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

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

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

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

175 3 Terms and definitions

176 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
177 are defined in this clause.

178 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
179 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
180 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
181 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
182 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
183 alternatives shall be interpreted in their normal English meaning.

184 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as
185 described in [ISO/IEC Directives, Part 2](#), Clause 5.

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

189 The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following additional
190 terms are used in this document.

191 3.1

192 **conditional**

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

195 3.2

196 **mandatory**

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

199 3.3

200 **optional**

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

202 3.4

203 **pending configuration**

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

206 3.5

207 **referencing profile**

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

210 3.6

211 **unspecified**

212 indicates that this profile does not define any constraints for the referenced CIM element or operation
213

214 4 Symbols and abbreviated terms

215 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following
216 additional abbreviations are used in this document.

217 4.1

218 Customer Network

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

222 4.2

223 IP

224 Internet Protocol

225 4.3

226 Gateway

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

229

230 IP Gateway

231 Uses the IP protocol as an underlay protocol to tunnel non-routable network segments, allowing the
232 traffic to be routed on an IP network.

233 4.4

234 Tunnel

235 A Tunnel is a path across non-routable network segments. These can be segments on different IP
236 networks or networks using different protocols.

237 4.5

238 IP Encapsulation

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

241 4.6

242 Overlay Network\Customer

243 A network containing non-encapsulated network traffic from the perspective of the encapsulation
244 gateways specified in this profile.

245 4.7

246 Provider Network

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

249 4.8

250 Network Overlay/Underlay

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

255 5 Synopsis

256 **Profile name:** Network Management - Tunnel Management Profile

257 **Version:** 1.0.0

258 **Organization:** DMTF

259 **CIM Schema version:** 2.52

260 **Central class:** CIM_IPEncapsulationManagementSevice

261 **Scoping class:** CIM_ComputerSystem

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

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

267 **Table 1 – Referenced profiles**

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

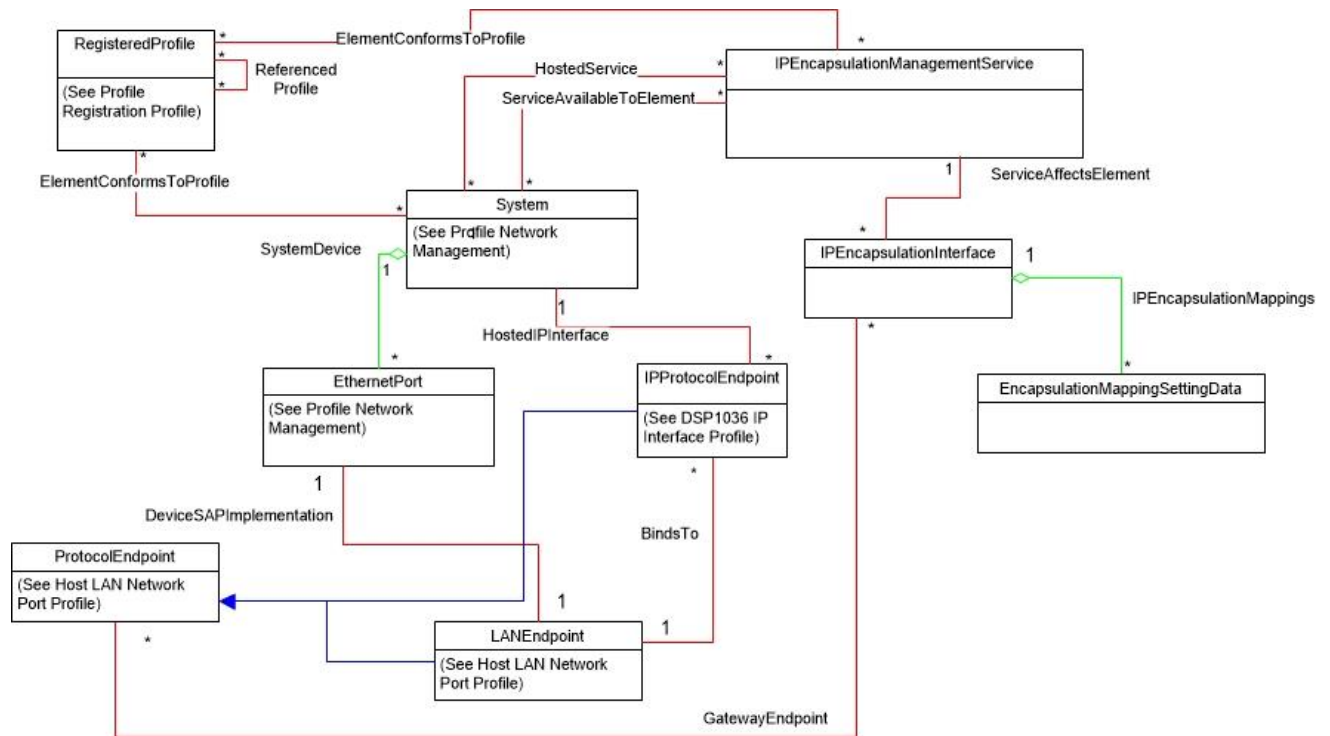
268 6 Description

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

273 6.1 Class diagram

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

276



277
278

279

Figure 1 – Network Management - Tunnel Management Profile

280 **6.2 Tunneling protocol variants**

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

287 **6.3 IPEncapsulationInterface**

288 In this profile the IPEncapsulationInterface is used to provide the configuration for an instance of an
 289 encapsulation service. This includes the ability to configure the endpoints that are on the provider network
 290 and the customer network and the virtual routing and forwarding tables that are to be used to determine
 291 the next hop routes required to route traffic between the two networks. An instance of
 292 IPEncapsulationInterface is associated with the collection of CIM_EncapsulationMappingSettingData that
 293 is used to describe the policy to map each tunnel.

294 **6.4 CIM_EncapsulationMappingSettingData**

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

309 **6.5 CIM_IPEncapsulationManagementService**

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

303 **7 Implementation**

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

306 **7.1 Representing IP encapsulation management services**

307 **7.1.1 CIM_IPEncapsulationManagementService**

308 One or more instances of CIM_IPEncapsulationManagementService shall be instantiated.

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

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

314 IPEncapsulationInterface instances managed by or created through the use of an instance of
315 CIM_IPEncapsulationManagementService shall be associated to the
316 CIM_IPEncapsulationManagementService instance through an instance of CIM_ServiceAffectsElement.

317 **7.2 CIM_IPEncapsulationInterface**

318 Instances of CIM_IPEncapsulationInterface created as part of the execution of
319 AddIPEncapsulationInterface() method shall be associated with the instance of
320 CIM_IPEncapsulationManagementService from which the method call was made through an instance of
321 CIM_ServiceAffectsElement association.

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

326 Instances of CIM_IPProtocolEndpoint created as part of the execution of AddIPEncapsulationInterface()
327 method shall be associated to the instantiated instance of CIM_IPEncapsulationInterface from the same
328 method call through an instance of the CIM_GatewayEndpoint association.

329 **7.3 Representing the Protocol Endpoints of an encapsulation gateway**

330 Each instance of CIM_EncapsulationGateway shall have two associated instances of
331 CIM_ProtocolEndpoint or a subclass of CIM_ProtocolEndpoint one representing an endpoint that is a part
332 of the customer network and another representing an endpoint that is a part of the provider network.
333 These instances are associated through the CIM_GatewayEndpoint association instances. The
334 association is made to these endpoints after the successful completion of a
335 CIM_IPEncapsulationManagementService.AddIPEncapsulationInterface() method.

336 If an instance of CIM_IPProtocolEndpoint is instantiated through a successful completion of a
337 CIM_IPEncapsulationManagementService.AddIPEncapsulationInterface() method where either a
338 CAIPEndpoint or an PAIPEndpoint parameter was populated, that instance shall be associated through

339 an instance of CIM_HostedIPInterface to the instance CIM_System that was specified in the
340 TargetSystem parameter of the method call.

341 7.4 Representing an encapsulation mapping table

342 7.4.1 CIM_EncapsulationMappingSettingData

343 Instances of CIM_EncapsulationMappingSettingData created as a result of the
344 AddIPEncapsulationInterface() or AddEncapsulationMappingSettingData () method shall be associated
345 to the instance of CIM_IPEncapsulationInterface contained in the EncapsulationGateway parameter of
346 the respective method through an aggregation instance of CIM_IPEncapsulationMappings.

347 8 Methods

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

350 8.1 Extrinsic methods

351 If synchronous execution of a method succeeds, the implementation shall set a return value of
352 0 (Completed with No Error).

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

355 If a method is executed as an asynchronous task, the implementation shall perform all of the following ac-
356 tions:

- 357 • Set a return value of 4096 (Job Started).
- 358 • Set the value of the Job output parameter to refer to an instance of the CIM_ConcreteJob class
359 that represents the asynchronous task.
- 360 • Set the values of the JobState and TimeOfLastStateChange properties in that instance to repre-
361 sent the state and last state change time of the asynchronous task.

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

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

- 365 • Set the value of the JobState property to 7 (Completed).
- 366 • Provide an instance of the CIM_AffectedJobEntity association with property values set as fol-
367 lows:
 - 368 – The value of the AffectedElement property shall refer to the object that represents the top-
369 level entity that was created or modified by the asynchronous task. For example, for the
370 CIM_IPConfigurationService. AddIPProtocolEndpoint() method, this is an instance of the
371 CIM_IPProtocolEndpoint class
 - 372 – The value of the AffectingElement property shall refer to the instance of the
373 CIM_ConcreteJob class that represents the completed asynchronous task.
 - 374 – The value of the first element in the ElementEffects[] array property (ElementEffects[0])
375 shall be set to 5 (Create) for the CIM_IPConfigurationService. AddIPProtocolEndpoint()
376 method. Otherwise, this value shall be 0 (Unknown).

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

379 **8.1.1 Job parameter**

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

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

386 **8.1.2 CIM_IPConfigurationService.AddIPEncapsulationInterface()**

387 The implementation of the AddIPEncapsulationInterface () method is optional, the provisions in this
 388 subclause apply in addition to behavior applicable to all extrinsic methods as specified in 8.1.

389 The successful execution of the AddIPEncapsulationInterface () method shall create an instance of the
 390 class or a subclass of CIM_IPEncapsulationInterface and any required associations as described in the
 391 subclauses 7.2 required to instantiate a complete encapsulation interface. In addition if the optional
 392 method parameter EncapsulationMappings is populated with corresponding instances of the embedded
 393 CIM_EncapsulationMappingSettingData classes the instantiated instances should be associated with the
 394 newly instantiated IPEncapsulationInterface through an instance of CIM_IPEncapsulationMappings. If the
 395 optional method parameters PAIPEndpoint or CAIPEndpoint are populated with corresponding instances
 396 of the embedded CIM_IPProtocolEndpoint classes the instantiated instances shall be associated with the
 397 newly instantiated IPEncapsulationInterface through an instance of CIM_GatewayEndpoint.

398 Table 2 contains requirements for parameters of this method.

399 **Table 2 – AddIPEncapsulationInterface () Method: Parameters**

<i>Qualifiers</i>	<i>Name</i>	<i>Type</i>	<i>Description/Values</i>
IN	TargetSystem	CIM_System REF	See 8.1.2.1
IN	EncapsulationGateway	String	See 8.1.2.2
IN	PAIPEndpoint	String	See 8.1.2.2
IN	CAIPEndpoint	String	See 8.1.2.4
IN	EncapsulationMappings	String[]	See 8.1.2.5
OUT	ResultingGateway	IPEncapsulationInterface REF	See 8.1.2.5
OUT	Job	CIM_ConcreteJob REF	See 8.1.1

400 **8.1.2.1 TargetSystem**

401 A required reference to a system or network. The supported target interfaces for a
 402 CIM_IPEncapsulationInterface class or subclass shall be as described in the subclauses of 7.2.

403 **8.1.2.2 EncapsulationGateway**

404 A required string containing an embedded instance of the class or subclass of
 405 CIM_IPEncapsulationInterface describes the initial configuration of the resulting
 406 CIM_IPEncapsulationInterface instance. The populated properties of the embedded instance should not
 407 contain key properties, and any key property values may be ignored.

408 **8.1.2.3 PAIPEndpoint**

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

413 **8.1.2.4 CAIPEndpoint**

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

418 **8.1.2.5 EncapsulationMapping[]**

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

424 **8.1.2.6 ResultingInterface**

425 If the creation of the CIM_IPEncapsulationInterface is successful, the instance of the class
 426 CIM_IPEncapsulationInterface that represents the instantiated instance of CIM_IPEncapsulationInteface
 427 is returned.

428 **8.1.2.7 Job**

429 See 8.1.1

430 **8.1.3 CIM_IPConfigurationService.AddIPEncapsulationMappings()**

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

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

437 Table 3 contains requirements for parameters of this method.

438 **Table 3 – AddIPEncapsulationMappings () Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	TargetInterface	CIM_IPEncapsulationInterface REF	See 8.1.3.1
IN	EncapsulationMappings	String[]	See 8.1.3.2
OUT	Job	CIM_ConcreteJob REF	See 8.1.3.3

439 **8.1.3.1 TargetInterface**

440 A required reference to an instance of IPEncapsulationInterface class or subclass.

441 **8.1.3.2 EncapsulationMapping[]**

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

450 **8.1.3.3 Job**

451 See 8.1.1

452 **8.1.4 CIM_IPConfigurationService.RemoveIPEncapsulationInterface()**

453 The implementation of the RemoveIPEncapsulationInterface() method is optional, the provisions in this
 454 subclause apply in addition to behavior applicable to all extrinsic methods as specified in 8.1.

455 The successful execution of the RemoveIPEncapsulationInterface() method shall remove the instances
 456 referenced in the methods Gateway parameter and shall remove any associated CIM_SettingData
 457 instances.

458 Table 4 contains requirements for parameters of this method.

459 **Table 4 – RemoveIPEncapsulationInterface() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Gateway	CIM_IPEncapsulationInterface REF[]	See 8.1.4.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.1

460 **8.1.4.1 Endpoint**

461 An array of references to the pair of the class CIM_IPIIPEncapsulationInterface instances that shall be
 462 removed.

463 **8.1.4.2 Job**

464 See 8.1.1

465 **8.1.5 CIM_IPConfigurationService.RemoveIPEncapsulationMappings()**

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

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

470 Table 5 contains requirements for parameters of this method.

471

Table 5 – RemoveIPEncapsulationMappings() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Encapsulation Mappings	CIM_EncapsulationMappingSettingData REF[]	See 8.1.5.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.5.2

472 **8.1.5.1 EncapsulationMappings**

473 An array of references to the set of CIM_EncapsulationMappingSettingData instances that shall be
 474 removed.

475 **8.1.5.2 Job**

476 See 8.1.1

477 **8.2 Profile conventions for operations**

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

480 The default list of operations is as follows:

- 481 • GetInstance
- 482 • EnumerateInstances
- 483 • EnumerateInstanceNames
- 484 • Associators
- 485 • AssociatorNames
- 486 • References
- 487 • ReferenceNames

488 **8.3 CIM_GatewayEndpoint**

489 Table 6 lists implementation requirements for operations. If implemented, these operations shall be
 490 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in
 491 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

493 **Table 6 – Operations: CIM_GatewayEndpoint**

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

494 **8.4 CIM_BindsTo**

495 Table 7 lists implementation requirements for operations. If implemented, these operations shall be
 496 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in
 497 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

499 **Table 7 – Operations: CIM_BindsToLANEndpoint**

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

500 **8.5 CIM_HostedService**

501 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
 502 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in
 503 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

505 **Table 8 – Operations: CIM_HostedService**

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

506 **8.6 CIM_HostedIPInterface**

507 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 508 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 509 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

511 **Table 9 – Operations: CIM_HostedIPInterface**

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

512

513 **8.7 CIM_IPEncapsulationManagementService**

514 All operations in the default list in 8.2 shall be implemented as defined in [DSP0200](#).

515 **8.8 IPEncapsulationInterface**

516 All operations in the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

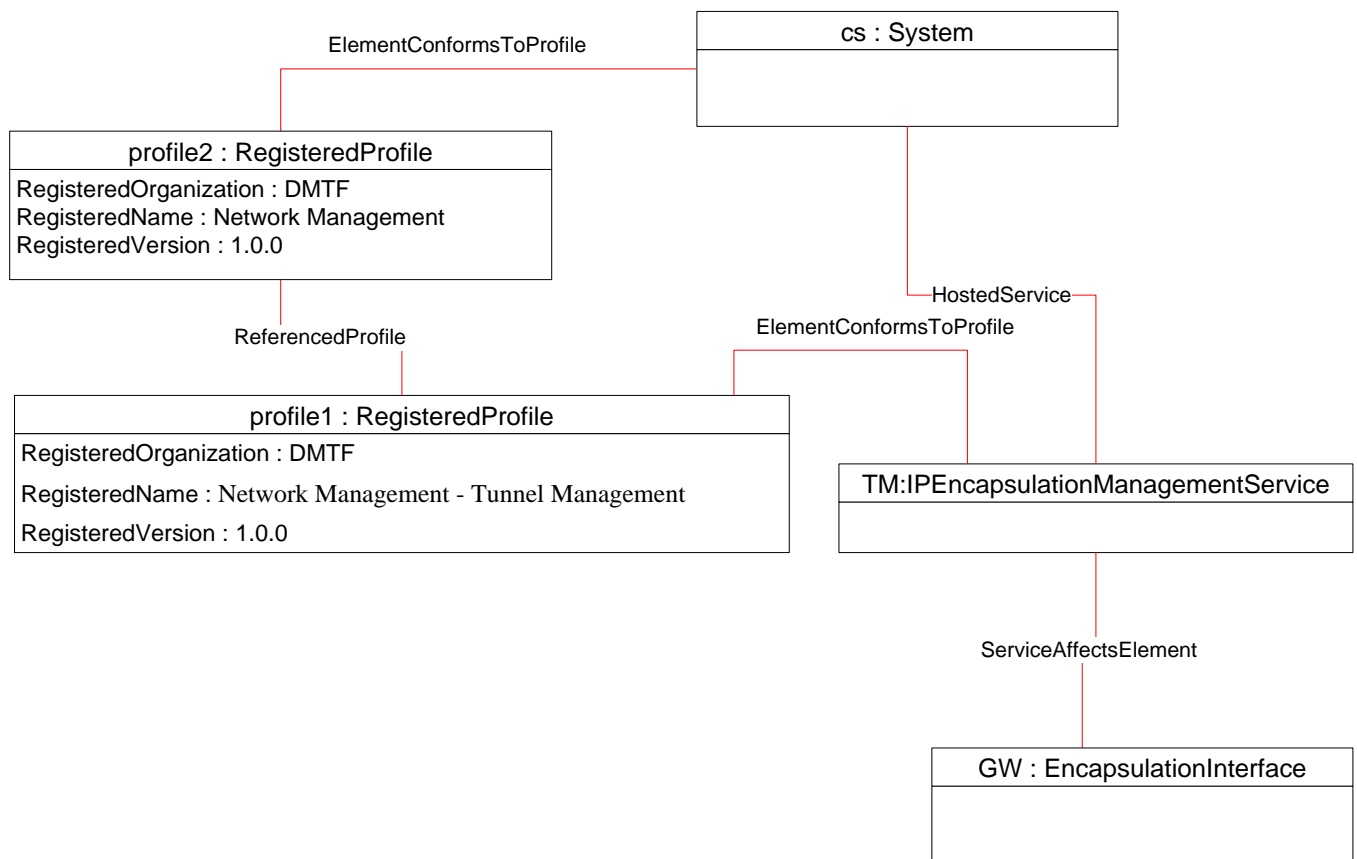
518 **9 Use cases**

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

521 **9.1 Profile Registration**

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

531



532
 533
 534

535 **Figure 2 – Registered profile**

536

537 9.2 L2 NVGRE tunnel gateway

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

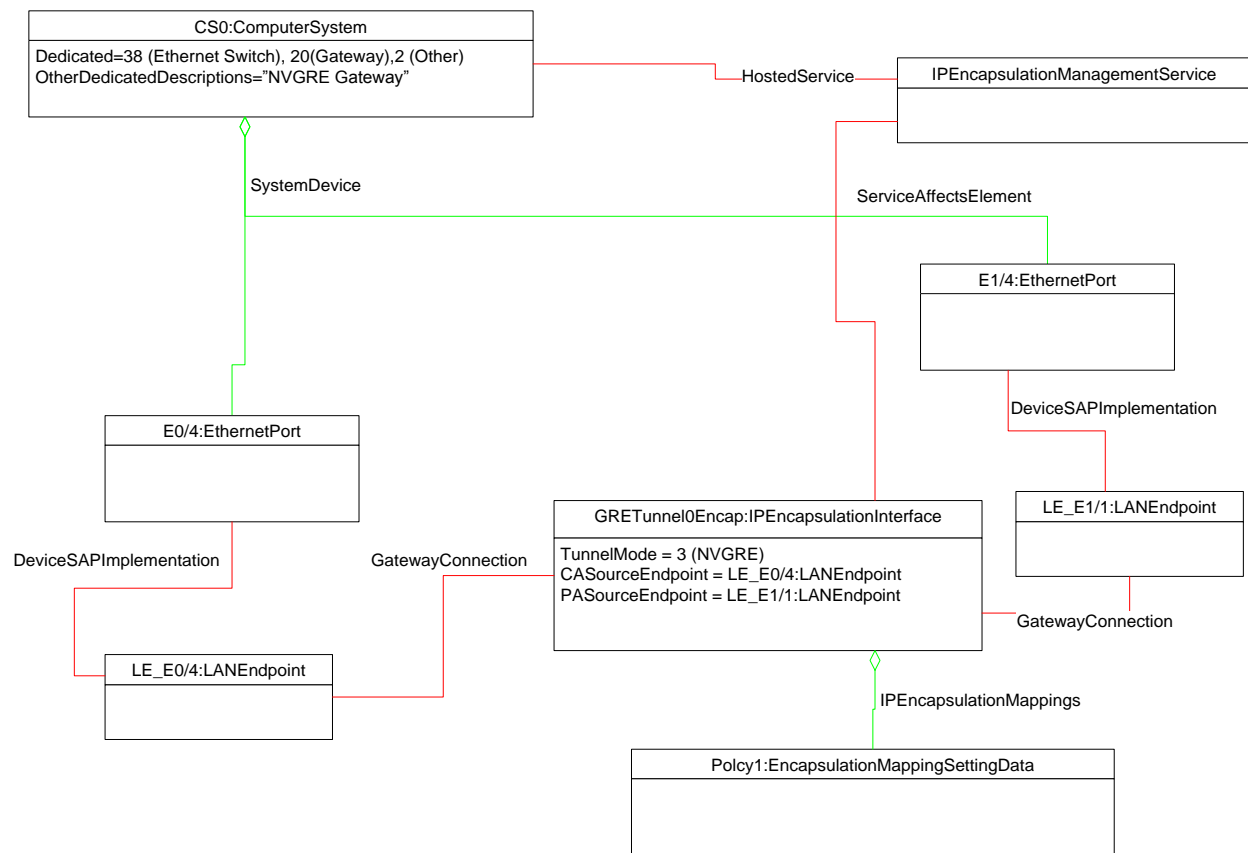
544 This gateway takes all network traffic from CIM_EthernetPort: E0/4 and encapsulates it based on the
545 mappings found in the instances of CIM_EncapsulationMappingSettingData and sends the encapsulated
546 traffic out on the provider network through CIM_EthernetPort: E1/1. Any traffic from the provider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

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

550 The IPEncapsulationInterface was created with a
551 CIM_IPConfigurationService.AddEncapsulationGateway() method with the following parameters.

- 552 • TargetInterface – WBEM URI reference to CIM_ComputerSystem:CS0
- 553 • EncapsulationGateway
 - 554 – Embedded Instance of IPEncapsulationInterface {
 - 555 TunnelMode=4 (NVGRE)
 - 556 CASourceInterface = WBEM URI reference to CIM_EthernetPort: E0/4
 - 557 PASourceInterface = WBEM URI reference to CIM_EthernetPort: E1/1 }
- 558 • EncapsulationMappings
- 559
 - 560 – Embedded Instance of EncapsulationMappingSettingData {
 - 561 CustomerAddress 10.1.0.125
 - 562 ProviderAddress 198.168.56.255
 - 563 MACAddress = 01:23:45:67:89:ab
 - 564 VSID = GUID for Virtual Subnet ID.
 - 565

566



567

568

Figure 3 – NVGRE tunnel gateway

569

570 9.3 Routed NVGRE tunnel gateway

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

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

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

589 This example shows the method call used to add a NVGRE IPEncapsulationInterface.

590

591 The IPEncapsulationInterface was created with a
592 CIM_IPConfigurationService.AddEncapsulationGateway() method with the following parameters.

- 593 • TargetInterface – WBEM URI reference to CIM_ComputerSystem:CS0
- 594 • EncapsulationGateway
 - 595 – Embedded Instance of IPEncapsulationInterface {
 - 596 TunnelMode=4 (NVGRE)
 - 597 CASourceInterface = null
 - 598 PASourceInterface = null
 - 599 CAVRFContext = WBEM URI reference to VRF1_CA:VirtualRoutingAndForwarding
 - 600 PAVRFContext = WBEM URI reference to VRF0_PA:VirtualRoutingAndForwarding

601

- 602 • CAEndpoint
 - 603 – Embedded Instance of CIM_IPProtocolEndpoint {
 - 604 IPv4Address=10.1.0.0
 - 605 ProtocolIFType=4060}

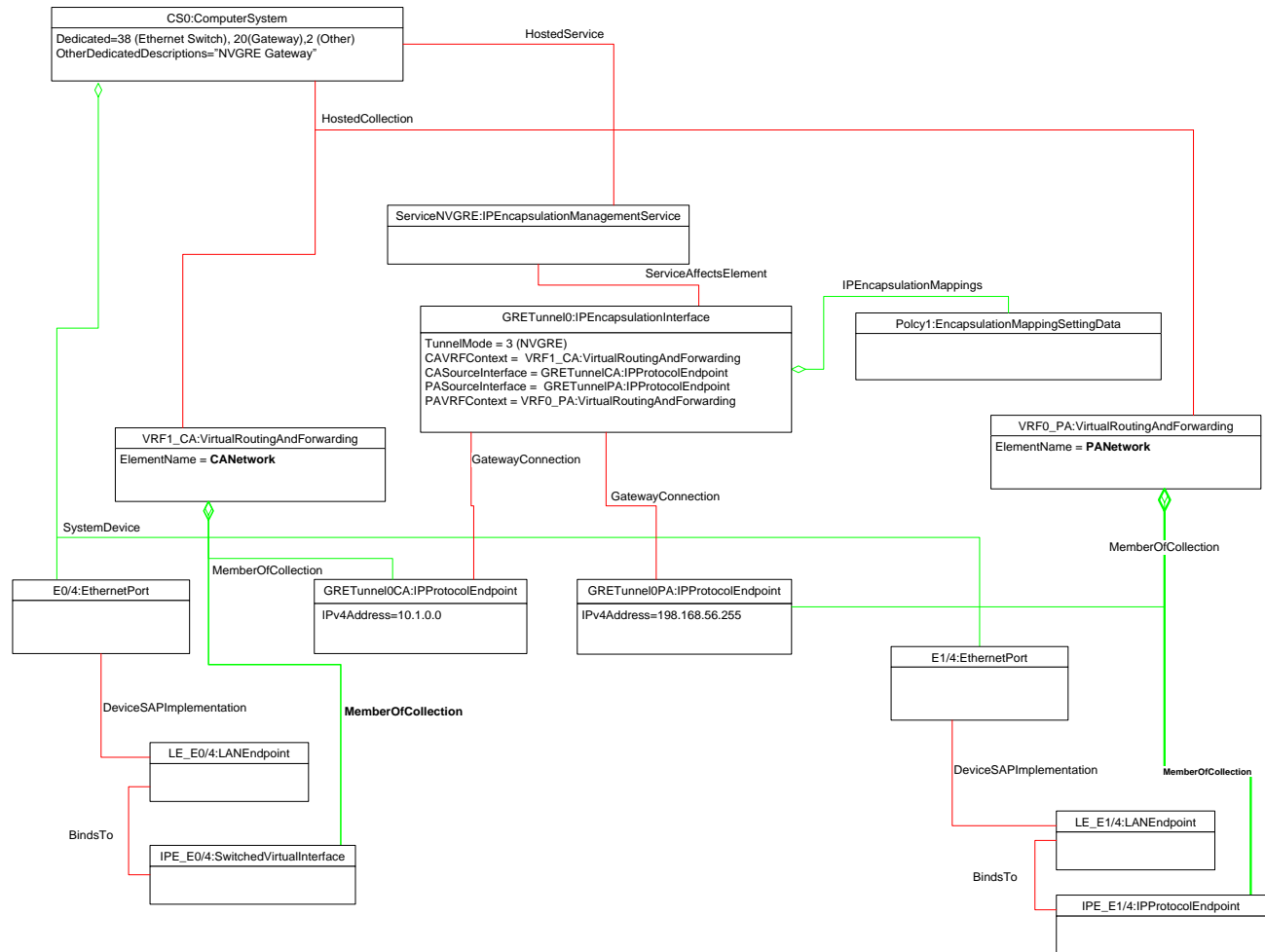
606

- 607 • PAEndpoint
 - 608 – Embedded Instance of CIM_IPProtocolEndpoint {
 - 609 IPv4Address=198.168.0.0
 - 610 ProtocolIFType=4060}

611

- 612 • EncapsulationMappings
 - 613 – CustomerAddress 10.1.0.125
 - 614 ProviderAddress 198.168.56.255
 - 615 MACAddress = 01:23:45:67:89:ab
 - 616 VSID = GUID for Virtual Subnet ID.

617



618
619

Figure 4 – NVGRE routed tunnel gateway

620 **10 CIM Elements**

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

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

625

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 DSP1036 IP Interface Profile 1.1.1
CIM_HostedIPInterface	Conditional	See DSP1036 IP Interface Profile 1.1.1
CIM_IPProtocolEndpoint	Conditional	See DSP1036 IP Interface Profile 1.1.1
CIM_IPEncapsulationManagementService	Mandatory	See 7.1

Element Name	Requirement	Description
CIM_RegisteredProfile	Optional	See DSP1036 IP Interface Profile 1.1.1
CIM_ServiceAffectsElement	Conditional	See DSP1036 IP Interface Profile 1.1.1
CIM_ServiceAvailableToElement	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPEncapsulationInterface	Required	See 7.2
Indications		
None defined in this profile		

626 **10.1 CIM_BindsToLANEndpoint**

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

630 **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

631 **10.2 CIM_BindsTo**

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

634 **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

635 **10.3 CIM_EncapsulationMappingSettingData**

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

638 **Table 13 – Class: CIM_EncapsulationMappingSettingData**

Elements	Requirement	Description
InstanceID	Mandatory	Key

639 **10.4 CIM_HostedService**

640 CIM_HostedService relates the CIM_IPEncapsulationManagementService instance to its scoping
641 CIM_System instance. Table 14 provides information about the properties of CIM_HostedService.

642 **Table 14 – 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 *

643 **10.5 CIM_IPEncapsulationManagementService**

644 CIM_IPEncapsulationManagementService provides the methods to create and delete an encapsulation
645 gateway interface. Table 15 provides information about the properties of
646 CIM_IPEncapsulationManagementService.

647 **Table 15 – Class: CIM_IPEncapsulationManagementService**

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

648 **10.6 CIM_IPProtocolEndpoint**

649 CIM_IPProtocolEndpoint represents an IP interface that is associated with an IP encapsulation gateway
650 or an Ethernet interface. Table 16 provides information about the properties of CIM_IPProtocolEndpoint.

651 **Table 16 – Class: CIM_IPProtocolEndpoint**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key

Elements	Requirement	Description
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

652 **10.7 CIM_IPEncapsulationInterface**

653 IPEncapsulationInterface represents either the encapsulation or the decapsulation IP encapsulation
 654 interface used to route to connect two disjointed IP networks. Table 17 provides information about the
 655 additional properties of IPEncapsulationInterface that are in addition to those in CIM_IPProtocolEndpoint,
 656 clause 10.5.

657 **Table 17 – Class: CIM_IPEncapsulationInterface**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	Pattern ".*"
TunnelMode	Mandatory	See 8.1.2 .
CAVRFCContext	Optional	See 8.1.2 .
CASourceInterface	Required	See 8.1.2 ..
PAVRFCContext	Optional	See 8.1.2 ..
PASourceInterface	Required	See 8.1.2 ..

658

659 **10.8 CIM_RegisteredProfile**

660 CIM_RegisteredProfile identifies the *Network Management - Tunnel Management Profile* in order for a
661 client to determine whether an instance of CIM_IPProtocolEndpoint is conformant with this profile. The
662 CIM_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the exception of the
663 mandatory values specified for the properties in Table 18, the behavior of the CIM_RegisteredProfile
664 instance is in accordance with the [Profile Registration Profile](#).

665 **Table 18 – 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 "1.0.0d".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

666
667
668
669

**ANNEX A
(informative)**

Change log

Version	Date	Description
1.0.0	2018-09-07	

670