



1  
2  
3  
4

**Document Number: DSP1063**

**Date: 2013-09-22**

**Version: 1.0.0a**

## 5 **Network Management Layer3 Interface 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 the stated expiration date.

**It expires on: 2013-12-22**

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

6 **Document Type: Specification**  
7 **Document Status: Work In Progress**  
8 **Document Language: en-US**  
9

## 10 Copyright Notice

11 Copyright © 2012-2013 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

32

# CONTENTS

34	Foreword .....	5
35	Introduction.....	6
36	1 Scope .....	7
37	2 Normative references .....	7
38	3 Terms and definitions .....	7
39	4 Symbols and abbreviated terms.....	8
40	5 Synopsis .....	9
41	6 Description .....	9
42	6.1 Class diagram .....	10
43	6.2 CIM_IPProtocolEndpoint .....	10
44	6.3 CIM_SwitchVirtualInterface .....	11
45	6.4 CIM_Subinterface .....	11
46	6.5 CIM_Loopback.....	11
47	6.6 CIM_HostedIPInterface.....	11
48	6.7 CIM_IPEncapsulationInterface .....	11
49	6.8 CIM_IPConfigurationService .....	11
50	7 Implementation.....	11
51	7.1 Representing the layer3 interface management capabilities.....	11
52	7.1.1 CIM_IPConfigurationService .....	11
53	7.2 Representing Layer 3 Interfaces.....	12
54	7.2.1 CIM_IPProtocolEndpoint .....	12
55	8 Methods.....	13
56	8.1 Extrinsic Methods.....	13
57	8.1.2 CIM_IPConfigurationService. AddIPProtocolEndpoint().....	14
58	8.1.3 CIM_IPConfigurationService. RemoveIPProtocolEndpoint() .....	15
59	8.2 Profile conventions for operations .....	16
60	8.3 CIM_BindsToLANEndpoint .....	16
61	8.4 CIM_HostedService .....	16
62	8.5 CIM_HostedIPInterface.....	17
63	8.6 CIM_L3InterfaceConfigurationService .....	17
64	8.7 CIM_IPEncapsulationInterface .....	17
65	8.8 CIM_Subinterface .....	17
66	8.9 CIM_LoopbackInterface.....	17
67	8.10 CIM_SwitchVirtualInterface .....	17
68	8.11 CIM_IPProtocolInterface.....	17
69	8.12 CIM_Subinterface .....	18
70	9 Use cases.....	19
71	9.1 Profile Registration.....	19
72	9.2 Subinterface .....	20
73	9.3 Switch Virtual Interface .....	21
74	9.4 Tunnel Interface and Loopback Interface .....	22
75	9.5 Add an IPProtocolEndpoint to an Ethernet Port. ....	23
76	10 CIM Elements.....	24
77	10.1 CIM_BindsToLANEndpoint .....	25
78	10.2 CIM_HostedService .....	25
79	10.3 CIM_IPConfigurationService .....	26
80	10.4 CIM_IPProtocolEndpoint .....	26
81	10.5 CIM_IPEncapsulationInterface .....	27
82	10.6 CIM_SubInterface .....	27
83	10.7 CIM_SwitchVirtualInterface .....	28
84	10.8 CIM_RegisteredProfile.....	28
85	ANNEX A (informative) Change log.....	29

86

87 **Figures**

88 Figure 1 – Network Management Layer3 Interface Profile: Class diagram ..... 10

89 Figure 2 – Registered profile..... 19

90

91 **Tables**

92 Table 1 – Referenced profiles ..... 9

93 Table 2 – AddIPProtocolEndpoint ( ) Method: Parameters..... 14

94 Table 3 – AddIPProtocolEndpoint ( ) Method: Parameters..... 15

95 Table 4 – Operations: CIM\_BindsToLANEndpoint ..... 16

96 Table 5 – Operations: CIM\_HostedService ..... 16

97 Table 6 – Operations: CIM\_HostedIPInterface ..... 17

98 Table 7 – Operations: CIM\_IPProtocolEndpoint ..... 18

99 Table 8 – CIM Elements: Network Management Layer 3 Interface Profile..... 24

100 Table 9 – Class: CIM\_BindsToLANEndpoint ..... 25

101 Table 10 – Class: CIM\_HostedService ..... 25

102 Table 11 – Class: CIM\_IPConfigurationService..... 26

103 Table 12 – Class: CIM\_IPProtocolEndpoint..... 26

104 Table 13 – Class: CIM\_IPEncapsulationInterface ..... 27

105 Table 14 – Class: CIM\_SubInterface ..... 27

106 Table 15 – Class: CIM\_SwitchVirtualInterface..... 28

107 Table 16 – Class: CIM\_RegisteredProfile ..... 28

108

109

## Foreword

110 The *Network Management Layer3* Interface Profile (DSP1063) was prepared by the Network Services  
111 Management Working Group of the DMTF.

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

## 114 **Acknowledgments**

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

116 Editors:

- 117 • John Parchem - Microsoft

118 Contributors:

- 119 • Hemal Shah – Broadcom Corporation
- 120 • John Crandall – Brocade Communications System
- 121 • Alex Zhdankin – Cisco Systems
- 122 • Steve Neely – Cisco Systems
- 123 • Shishir Pardikar – Citrix
- 124 • John Parchem – Microsoft Corporation
- 125 • Lawrence Lamers – VMware
- 126 • Dr. Bhumip Khasnabish - ZTE Corporation

127

128

## Introduction

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

### 134 Document conventions

#### 135 Typographical conventions

136 The following typographical conventions are used in this document:

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

139

140

# Network Management Layer3 Interface Profile

## 141 1 Scope

142 The *Network Management Layer3 Interface Profile* is a profile that specifies the CIM schema and use  
143 cases associated with the general and common aspects of typical layer 3 interfaces found in an Ethernet  
144 Switch. This profile includes a specification of the Layer 3 interface configuration service, Sub-Interface,  
145 IP Tunnel Interface, switch virtual interface and loopback interface.

## 146 2 Normative references

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

151 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,  
152 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.6.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf)

153 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
154 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

155 DMTF DSP0223, *Generic Operations 1.0*,  
156 [http://www.dmtf.org/standards/published\\_documents/DSP0223\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf)

157 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
158 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

159 DMTF DSP1033, *Profile Registration Profile 1.0*,  
160 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

161 DMTF DSP1097, *Virtual Ethernet Switch Profile 1.1*,  
162 [http://dmtf.org/sites/default/files/standards/documents/DSP1097\\_1.1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1097_1.1.0.pdf)

163 DMTF DSP1036 *IP Interface Profile 1.1.1*,  
164 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1036\\_1.1.1.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1036_1.1.1.pdf)

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

## 167 3 Terms and definitions

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

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

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

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

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

### 183 **3.1**

#### 184 **conditional**

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

### 187 **3.2**

#### 188 **mandatory**

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

### 191 **3.3**

#### 192 **optional**

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

### 194 **3.4**

#### 195 **pending configuration**

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

### 198 **3.5**

#### 199 **referencing profile**

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

### 202 **3.6**

#### 203 **unspecified**

204 indicates that this profile does not define any constraints for the referenced CIM element or operation  
205

## 206 **4 Symbols and abbreviated terms**

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

### 209 **4.1**

#### 210 **IP**

211 Internet Protocol

### 212 **4.2**

#### 213 **VLAN**

214 Virtual Local Area Network  
215

### 216 **4.3**

#### 217 **VSI**

218 Virtual Switch Interface  
219



220  
221  
222

## 223 5 Synopsis

224 **Profile name:** Network Management Layer3 Interface Profile

225 **Version:** 1.0.0

226 **Organization:** DMTF

227 **CIM Schema version:** 2.39e

228 **Central class:** CIM\_IPConfigurationService

229 **Scoping class:** CIM\_System

230 The *Network Management Layer3 Interface Profile* is a profile that specifies the CIM schema and use  
231 cases associated with managing the IP layer 3 interfaces in an Ethernet switch. This profile includes a  
232 specification for configuration and life cycle management of the IP configuration of an Ethernet switch  
233 port, Subinterfaces, Switch Virtual Interfaces, Loopback and IP tunnel interfaces.

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

235

**Table 1 – Referenced profiles**

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None
Virtual Ethernet Switch	DMTF	1.1	Mandatory	None
IP Configuration Profile	DMTF	1.1	Optional	None
IP Interface Profile	DMTF	1.1.1	Mandatory	None
Host LAN Network	DMTF	1.0.2	Mandatory	None
Network Management	DMTF	1.0	Optional	None
Network Management Routing	DMTF	1.0	Optional	None

## 236 6 Description

237 The *Network Management Layer3 Interface Profile* is a profile that will specify the CIM schema and use  
238 cases associated with the general and common aspects of creating and configuring layer 3 interfaces in a  
239 typical Ethernet switch. These interfaces include IP configuration of an Ethernet switch port,  
240 Subinterfaces, Switch Virtual Interfaces, Loopback and IP tunnel interfaces.



### 262 **6.3 CIM\_SwitchVirtualInterface**

263 A switch virtual interface allows IP routing across VLANs. A CIM\_VLANNetwork instance can only have  
264 one CIM\_SwitchVirtualInterface instance associated to it through an instance of CIM\_HostedIPInterface.

### 265 **6.4 CIM\_Subinterface**

266 A Subinterface subdivides a single switch port into multiple IP subnets. This is typically done using Dot1Q  
267 encapsulation using VLANIDs to distinguish the subnets. Even though a Subinterface may have a VLANId  
268 within the scoped router this is a layer 3 interface and this interface is not a part of an internal  
269 VLANNetwork with the same VLANId.

### 270 **6.5 CIM\_Loopback**

271 A loopback interface is a virtual Layer 3 interface typically found in an Ethernet Switch or router. It is has  
272 a single endpoint that is always up. Packets that are transmitted over a loopback interface are  
273 immediately received by this interface.

### 274 **6.6 CIM\_HostedIPInterface**

275 An association allowing for the discovery of all IP interfaces that are hosted by a switch (CIM\_System) or  
276 a network (CIM\_Network).

### 277 **6.7 CIM\_IPEncapsulationInterface**

278 An IP tunnel interface is used to connect two disjointed IP networks.

279

### 280 **6.8 CIM\_IPConfigurationService**

281 The CIM\_IPConfigurationService is the central class of this profile. The service has a set of extrinsic  
282 methods to control the creation and removal layer 3 IP interfaces. The service can be available to  
283 physical interfaces represented with instances of CIM\_EthernetPort, a switch represented by  
284 CIM\_ComputerSystem and VLAN networks represented with instances of CIM\_VLANNetwork.

## 285 **7 Implementation**

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

### 288 **7.1 Representing the layer3 interface management capabilities**

#### 289 **7.1.1 CIM\_IPConfigurationService**

290 One or more instances of CIM\_IPConfigurationService shall be instantiated.

291 These instances of CIM\_IPConfigurationService shall be associated with an instance of the scoping  
292 CIM\_ComputerSystem class through an instance of CIM\_HostedService.

293 The instances of the CIM\_IPConfigurationService class shall also be associated to each  
294 CIM\_ManagedElement subclass instance that may be used as the TargetInterface parameter of its  
295 AddIPProtocolEndpoint () method through an instance of CIM\_ServiceAvailableToElement.

296 IPProtocolEndpoint instances created through the use of an instance of CIM\_IPConfigurationService shall  
297 be associated to the CIM\_IPConfigurationService instance through an instance of  
298 CIM\_ServiceAffectsElement.

299

## 300 7.2 Representing Layer 3 Interfaces

### 301 7.2.1 CIM\_IPProtocolEndpoint

302 Instances of CIM\_IPProtocolEndpoint created as a result of the  
303 CIM\_IPConfigurationService.AddIPProtocolEndpoint () shall comply with the requirements of [DSP1036 IP](#)  
304 [Interface Profile 1.1.1](#) where CIM\_IPProtocolEndpoint is the central class of [DSP1036](#). The additional  
305 requirements listed in this cause and its sub clauses are in addition to requirements in [DSP1036](#).

#### 306 7.2.1.1 CIM\_IPProtocolEndpoint (CIM\_EthernetPort)

307 Instances of CIM\_IPProtocolEndpoint created as a result of the  
308 CIM\_IPConfigurationService.AddIPProtocolEndpoint () method targeting an instance of CIM\_EthernetPort  
309 shall be associated with the instance of CIM\_LANEndpoint associated to the CIM\_EthernetPort instance,  
310 that was specified as the TargetInterface of the method call, through an instance of  
311 CIM\_BindsToLANEndpoint. This instance of CIM\_IPProtocolEndpoint shall also be associated through an  
312 instance of CIM\_HostedIPInterface to the scoping instance of CIM\_ComputerSystem.

#### 313 7.2.1.2 CIM\_IPEncapsulationInterface, CIM\_LoopbackInterface and 314 CIM\_SwitchVirtualInterface

315 Instances of CIM\_IPProtocolEndpoint created as a result of the  
316 CIM\_IPConfigurationService.CreateIPProtocolEndpoint() method where the TargetInterface is an  
317 instance of CIM\_System shall be associated to the TargetInterface through an instance of  
318 CIM\_HostedIPInterface.

#### 319 7.2.1.3 CIM\_LoopbackInterface

320 Represents a single IP endpoint communication channel. CIM\_LoopbackInterface shall conform to  
321 7.2.1.2. The instance of CIM\_System described in 7.2.1.2 shall be the instance of the class scoping class  
322 instance of CIM\_ComputerSystem.

#### 323 7.2.1.4 CIM\_Subinterface

324 Represents the subdivision of a single port into multiple IP subnets. CIM\_Subinterface shall conform to  
325 7.2.1.1.

326 The value of EncapsulationType shall be 1 or 2. If the value matches 1 (Other) the  
327 OtherEncapsulationType property shall be implemented and contain the encapsulation type represented  
328 as a free form string. If the value matches 2 (Dot1Q) the EncapsulationValue property shall be  
329 implemented and contain the 12 bit VLANId value represented as a string.

330 The ParentInterface property shall be implemented and contain a reference to the port interface, the  
331 instance of CIM\_EthernetPort that is being subdivided. This value shall be formatted as a URI per  
332 RFC3986 and should be a WBEM URI (DSP0207). If this interface was created using the  
333 CIM\_IPConfigurationService.AddIPProtocolEndpoint (), this value shall be the reference passed in the  
334 TargetInterface parameter of the method call.

335

#### 336 7.2.1.5 CIM\_SwitchVirtualInterface

337 Represents the IP settings for a VLAN to allow layer 3 routing between VLANs.  
338 CIM\_SwitchVirtualInterface shall conform to 7.2.1.2. The instance of CIM\_System described in 7.2.1.2  
339 shall be an instance of the class CIM\_VLANNetwork.

340 The VLANId property shall be implemented and contain the 12 bit VLANId that this interface is depended  
341 on. If this interface was created using the CIM\_IPConfigurationService.AddIPProtocolEndpoint (), this  
342 value shall be the VLANId of the CIM\_VLANNetwork Instance passed in the TargetInterface parameter  
343 of the method call.

#### 344 **7.2.1.6 CIM\_IPEncapsulationInterface**

345 Represents an interface to tunnel between disjoint IP networks, generally through encapsulation.  
346 CIM\_IPEncapsulationInterface shall conform to 7.2.1.2. The instance of CIM\_System described in 7.2.1.2  
347 shall be the instance of the scoping class instance of CIM\_ComputerSystem. The properties inherited  
348 from the class CIM\_IPProtocolEndpoint shall represent the IP configuration for the represented IP tunnel  
349 Interface and shall conform to [DSP1036 IP Configuration Profile](#).

350 The value of the property TunnelMode shall be implemented and contain a value that represents the type  
351 of tunnel this interface is implementing. If the value matches 1 (Other) the OtherTunnelMode property  
352 shall be implemented and contain the tunnel mode represented as a free form string.

353 If the VRFContext is implemented it may contain an identifier of the virtual routing and forwarding table  
354 (VRF) used to determine the tunnel IP destination address.

355 If the TunnelDestinationIPAddress is implemented it may contain the destination IP address for this tunnel  
356 interface. The value of the property shall be specified in dotted decimal notation as defined in IETF [RFC](#)  
357 [1208](#) if an IPv4 address or if an IPv6 address the value of the property shall be specified in the notation  
358 specified in IETF [RFC 4291](#), section 2.2.

359 If implemented the TunnelSourceInterface may contain the identifier of the source interface, an instance  
360 of CIM\_EthernetPort.

361

362

## 363 **8 Methods**

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

366

### 367 **8.1 Extrinsic Methods**

368 If synchronous execution of a method succeeds, the implementation shall set a return value of  
369 0 (Completed with No Error).

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

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

- 374 • Set a return value of 4096 (Job Started).
- 375 • Set the value of the Job output parameter to refer to an instance of the CIM\_ConcreteJob class  
376 that represents the asynchronous task.
- 377 • Set the values of the JobState and TimeOfLastStateChange properties in that instance to repre-  
378 sent the state and last state change time of the asynchronous task.

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

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

- 382 • Set the value of the JobState property to 7 (Completed).
- 383 • Provide an instance of the CIM\_AffectedJobEntity association with property values set as fol-  
384 lows:
  - 385 – The value of the AffectedElement property shall refer to the object that represents the top-  
386 level entity that was created or modified by the asynchronous task. For example, for the  
387 CIM\_IPConfigurationService.AddIPProtocolEndpoint() method, this is an instance of the  
388 CIM\_IPProtocolEndpoint class
  - 389 – The value of the AffectingElement property shall refer to the instance of the  
390 CIM\_ConcreteJob class that represents the completed asynchronous task.
  - 391 – The value of the first element in the ElementEffects[ ] array property (ElementEffects[0])  
392 shall be set to 5 (Create) for the CIM\_IPConfigurationService.AddIPProtocolEndpoint()  
393 method. Otherwise, this value shall be 0 (Unknown).

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

#### 396 8.1.1.1 Job parameter

397 The implementation shall set the value of the Job parameter as a result of an asynchronous execution of  
398 a method of the CIM\_IPConfigurationService as follows:

- 399 • If the method execution is performed synchronously, the implementation shall set the value to  
400 NULL.
- 401 • If the method execution is performed asynchronously, the implementation shall set the value to  
402 refer to the instance of the CIM\_ConcreteJob class that represents the asynchronous task.

403

#### 404 8.1.2 CIM\_IPConfigurationService.AddIPProtocolEndpoint()

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

407 The successful execution of the AddIPProtocolEndpoint( ) method shall create an index array of instance  
408 of the CIM\_IPProtocolEndpoint class or a subclass of IPProtocolEndpoint and any required associations  
409 as described in the sub clauses of 7.2. In addition if the optional method parameter EndpointSettings is  
410 populated corresponding instances of the embedded CIM\_SettingData classes should be associated with  
411 the newly instantiated CIM\_IPProtocolEndpoint through an instance of CIM\_ElementSettingData.

412 Table 2 contains requirements for parameters of this method.

413 **Table 2 – AddIPProtocolEndpoint ( ) Method: Parameters**

<i>Qualifiers</i>	<i>Name</i>	<i>Type</i>	<i>Description/Values</i>
IN	TargetInterface	CIM_ManagedElement REF	See 8.1.2.1.1
IN	IPProtocolEndpoint	String[]	See 8.1.2.1.2
IN	EndpointSettings	String[]	See 8.1.2.1.3
OUT	ResultingEndpoint	CIM_IPProtocolEndpoint REF[]	See 8.1.2.1.4
OUT	Job	CIM_ConcreteJob REF	See 8.1.1.1

414 **8.1.2.1.1 TargetInterface**

415 A required reference to an associated target interface, system or network. The supported target interfaces  
 416 for a CIM\_IPProtocolEndpoint class or subclass supported should be as described in the sub clauses of  
 417 7.2.

418 **8.1.2.1.2 IPProtocolEndpoint[]**

419 A required array of string an containing one or more embedded instances of the class-subclass of  
 420 CIM\_IPProtocolEndpoint that describes the configuration of the resultant CIM\_IPProtocolEndpoints. The  
 421 populated properties of the embedded CIM\_IPProtocolEndpoints should not contain key properties, and  
 422 any key property values may be ignored.

423 **8.1.2.1.3 EndpointSettings[]**

424 An optional array of strings containing embedded instances of the class-subclass of CIM\_SettingData  
 425 that describes the additional configuration properties for the resultant CIM\_IPProtocolEndpoints. The  
 426 array shall be indexed to the IPProtocolEndpoint array property. The populated properties of the  
 427 embedded CIM\_SettingData instances should not contain key properties, and any key property values  
 428 may be ignored. The resulting CIM\_SettingData instance should be associated with the corresponding  
 429 resultant instance of CIM\_IPProtocolEndpoint through an instance of CIM\_ElementSettingData.

430 **8.1.2.1.4 ResultingEndpoint[]**

431 If the assignment of a protocol endpoint is successfully, an array of references to the resultant instances  
 432 of class CIM\_IPProtocolEndpoint that represents the newly defined endpoints shall be returned.

433 **8.1.2.1.5 Job**

434 See 8.1.1.1

435 **8.1.3 CIM\_IPConfigurationService. RemoveIPProtocolEndpoint()**

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

438 The successful execution of the RemoveIPProtocolEndpoint ( ) method shall remove the instances  
 439 referenced in the methods Endpoint parameter and should remove any associated CIM\_SettingData  
 440 instances.

441 Table 2 contains requirements for parameters of this method.

442 **Table 3 – AddIPProtocolEndpoint ( ) Method: Parameters**

<b>Qualifiers</b>	<b>Name</b>	<b>Type</b>	<b>Description/Values</b>
IN	Endpoint	CIM_IPProtocolEndpoint REF	See 8.1.3.1.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.1.1

443 **8.1.3.1.1 Endpoint**

444 An array of references to instances of the class CIM\_IPProtocolEndpoint that shall be removed.

445 **8.1.3.1.2 Job**

446 See 8.1.1.1

## 447 8.2 Profile conventions for operations

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

450 The default list of operations is as follows:

- 451 • GetInstance
- 452 • EnumerateInstances
- 453 • EnumerateInstanceNames
- 454 • Associators
- 455 • AssociatorNames
- 456 • References
- 457 • ReferenceNames

## 458 8.3 CIM\_BindsToLANEndpoint

459 Table 4 lists implementation requirements for operations. If implemented, these operations shall be  
460 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 4, all operations in  
461 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

463 **Table 4 – Operations: CIM\_BindsToLANEndpoint**

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

## 464 8.4 CIM\_HostedService

465 Table 5 lists implementation requirements for operations. If implemented, these operations shall be  
466 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 5, all operations in  
467 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

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

469 **Table 5 – Operations: CIM\_HostedService**

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



## 470 8.5 CIM\_HostedIPInterface

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

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

475 **Table 6 – Operations: CIM\_HostedIPInterface**

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

476

## 477 8.6 CIM\_L3InterfaceConfigurationService

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

## 479 8.7 CIM\_IPEncapsulationInterface

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

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

## 482 8.8 CIM\_Subinterface

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

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

485

## 486 8.9 CIM\_LoopbackInterface

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

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

489

## 490 8.10 CIM\_SwitchVirtualInterface

491 All operations in 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

## 494 8.11 CIM\_IPProtocolInterface

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

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

497

**Table 7 – Operations: CIM\_IPProtocolEndpoint**

Operation	Requirement	Messages
ModifyInstance	Conditional. See <a href="#">DSP1036 1.1.1</a>	None

498

**499 8.12 CIM\_Subinterface**

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

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

502

503

504

505 **9 Use cases**

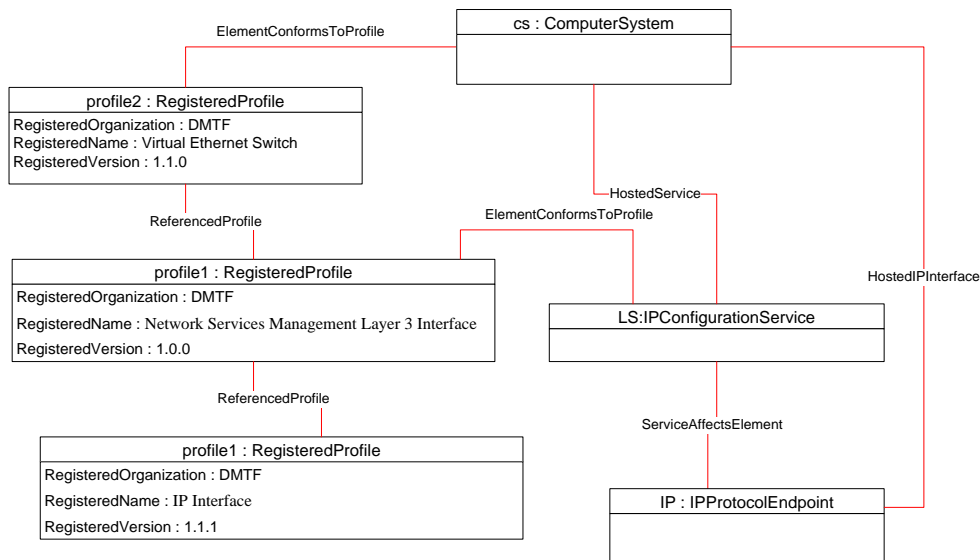
506 This clause contains object diagrams and use cases for the *Network Management Layer3 Interface Profile*.  
 507

508 **9.1 Profile Registration**

509 The object diagram in Figure 2 shows one possible method for advertising profile conformance. The  
 510 instances of CIM\_RegisteredProfile are used to identify the version of the Network Management Layer3  
 511 Interface Profile with which an instance of CIM\_IPConfigurationService is conformant. An instance of  
 512 CIM\_RegisteredProfile exists for each profile that is instrumented in the system. One instance of  
 513 CIM\_RegisteredProfile identifies the “VirtualEthernetSwitch1.1.0”. The other instance identifies the  
 514 “Network Management Layer3 Interface Profile”. The CIM\_IPConfigurationService instance is scoped to  
 515 an instance of CIM\_ComputerSystem. This instance of CIM\_ComputerSystem is conformant with the  
 516 DMTF *Virtual Ethernet Switch Profile* version 1.1.0 as indicated by the CIM\_ElementConformsToProfile  
 517 association to the CIM\_RegisteredProfile instance.

518

519



520

521

**Figure 2 – Registered profile**

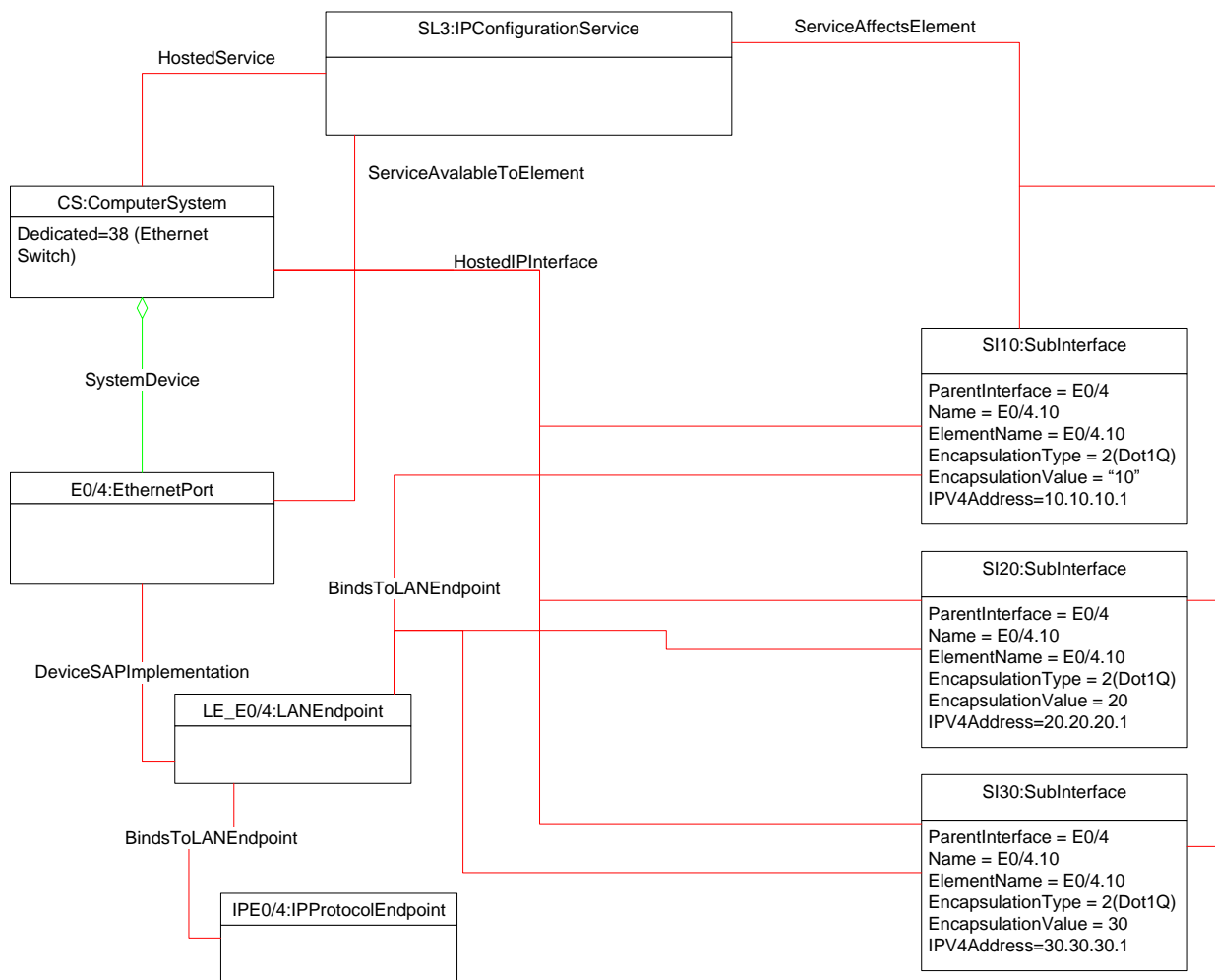
522

## 523 9.2 Subinterface

524 The object diagram shown in Figure 3 contains the basic elements used to model configuration of the  
525 Subinterfaces of an Ethernet switch port. The diagram shows that Ethernet port E0/4 has three  
526 associated instances of CIM\_Subinterface, SI10, SI20, and SI 30 each using Dot1Q encapsulation to  
527 separate the three IP subnets (10.10.10.1, 10.20.20.20.1 and 10.30.30.30.10. The Dot1Q  
528 encapsulation respectively uses VLANId 10, 20 and 30 to provide the isolation in the layer 2 switch. This  
529 is a very simple diagram, not shown are many of the required properties of the relative profiles for the  
530 objects shown.

531 The Subinterfaces were created with a CIM\_IPConfigurationService.AddProtocolEndpoint() method with  
532 the following parameters. Note this is for illustration purposes and other properties from the super class  
533 CIM\_IPProtocolEndpoint and other base classes may be specified as required.

- 534 • TargetInterface – WBEM URI reference to E0/4
- 535 • IPProtocolEndpoint[] –
  - 536 ○ Embedded Instance of CIM\_Subinterface {
  - 537 ElementName = E0/4.10
  - 538 EncapsulationType = 2
  - 539 EncapsulationValue = 10
  - 540 IPv4Address=10.10.10.
  - 541 ProtocolIFType=4060}
  - 542
  - 543 ○ Embedded Instance of CIM\_Subinterface {
  - 544 ElementName = E0/4.20
  - 545 EncapsulationType = 2
  - 546 EncapsulationValue = 20
  - 547 IPv4Address=20.20.20.1
  - 548 ProtocolIFType=4060}
  - 549
  - 550 ○ Embedded Instance of CIM\_Subinterface {
  - 551 ElementName = E0/4.30
  - 552 EncapsulationType = 2
  - 553 EncapsulationValue = 30
  - 554 IPv4Address=30.30.30.1
  - 555 ProtocolIFType=4060}
  - 556
  - 557



558  
559

560

Figure 3 – Subinterface

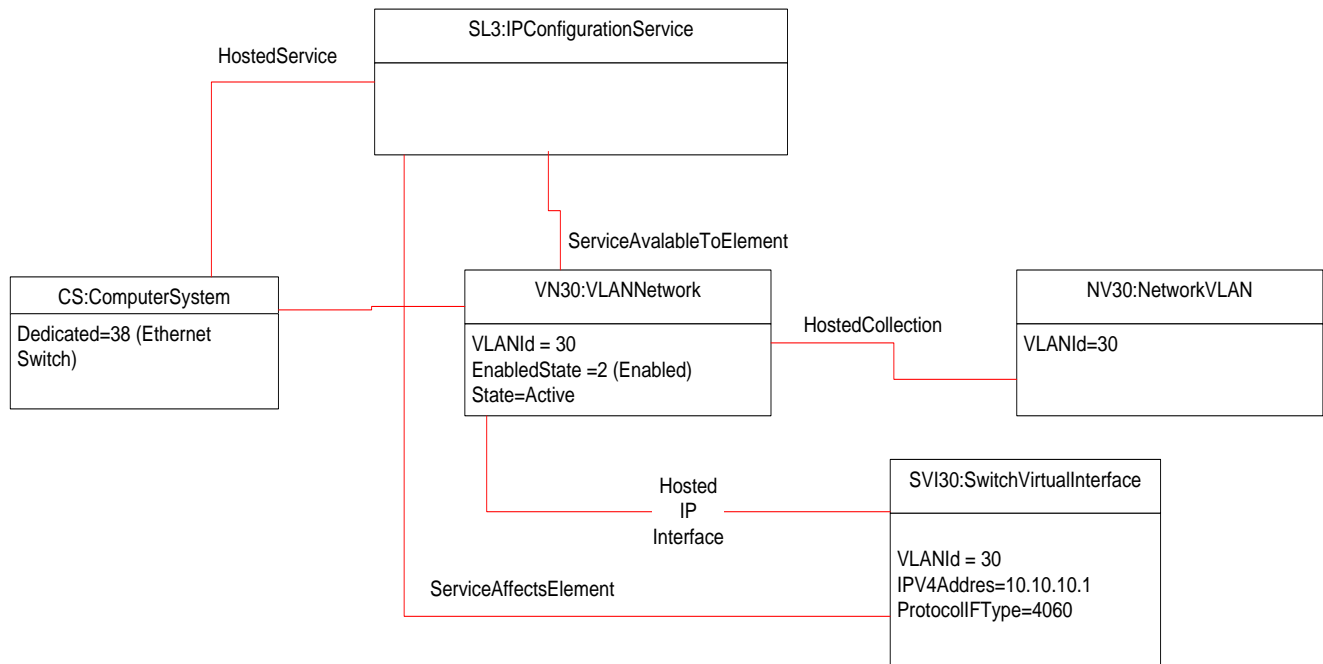
561 **9.3 Switch Virtual Interface**

562 The object diagram shown in Figure 4 contains the basic elements used to model configuration of a  
 563 Switch Virtual Interface (SVI) of a VLAN. The diagram shows that a CIM\_VLANNetwork, VN30 has an  
 564 associated instances of CIM\_SwitchVirtualInterface, SVI30. This interface provides the VLAN an IP  
 565 address allowing a routing component in the switch to bridge VLANs. Note that in the method description  
 566 below the caller did not populate the VLANId property in the embedded instance. In this example the  
 567 provider populated the property in the resultant instance with the value of the VLANId property from the  
 568 TargetInterface. This is a very simple diagram, not shown are many of the required properties of the  
 569 relative profiles for the objects shown.

570 The SVI was created with a IPConfigurationService.AddProtocolEndpoint() method with the following  
 571 parameters. Note this is for illustration purposes and other properties from the super class  
 572 CIM\_IPProtocolEndpoint and other base classes may be specified as required.

- 573       • TargetInterface – WBEM URI reference to VN30  
 574       • IPProtocolEndpoint[] –

- 575           ○ Embedded Instance of CIM\_SwitchVirtualInterface {  
 576            IPv4Address=10.10.10.1  
 577            ProcollIFType=4060}  
 578



579  
 580

581

**Figure 4 - Switch Virtual Interface**

## 582 9.4 Tunnel Interface and Loopback Interface

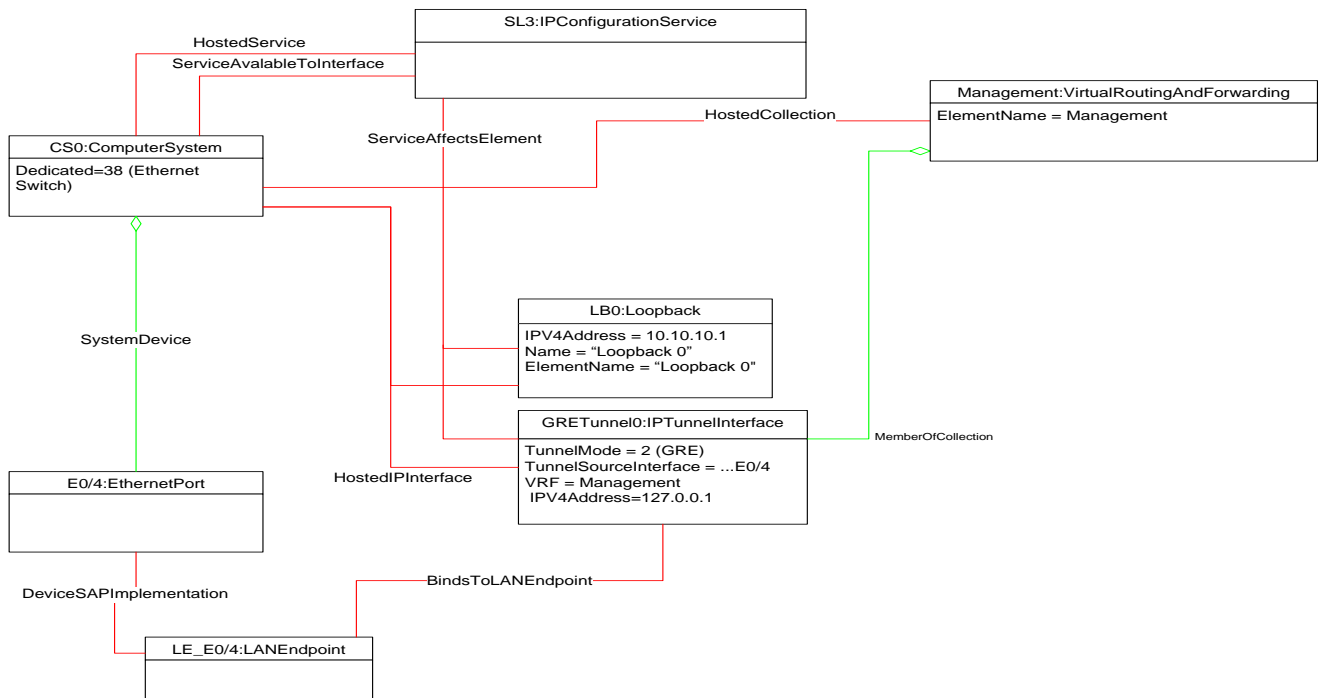
583 The object diagram shown in Figure 5 contains the basic elements used to model configuration of an IP  
 584 tunnel interface and a loopback interface. The diagram shows that CIM\_ComputerSystem has two  
 585 associated instances through the association CIM\_HostedIPInterface. These are a loopback interface  
 586 CIM\_Loopback:LB0 and an IP tunnel interface CIM\_IPEncapsulationInterface:GRETunnel0. This is a  
 587 very simple diagram, not shown are many of the required properties of the relative profiles for the objects  
 588 shown.

589 The two interfaces were created with a CIM\_IPConfigurationService.AddProtocolEndpoint() method with  
 590 the following parameters. Note this is for illustration purposes and other properties from the super class  
 591 CIM\_IPProtocolEndpoint and other base classes may be specified as required.

- 592           • TargetInterface – WBEM URI reference to CIM\_ComputerSystem:CS0  
 593           • IPProtocolEndpoint[] –  
 594            ○ Embedded Instance of CIM\_Loopback {  
 595            IPv4Address=10.10.10.1  
 596            ProcollIFType=4060}  
 597  
 598            ○ Embedded Instance of CIM\_IPEncapsulationInterface {  
 599            TunnelMode=2 (GRE)  
 600            TunnelSourceInterface = WBEM URI reference to CIM\_EthernetPort: E0/4  
 601            VRFContext = WBEM URI reference to CIM\_VirtualRoutingandForwarding: Management  
 602            IPv4Address=127.0.0.1  
 603            ProcollIFType=4060}

604  
 605

606



607  
608

Figure 5 - Tunnel and Loopback Interface

609  
610

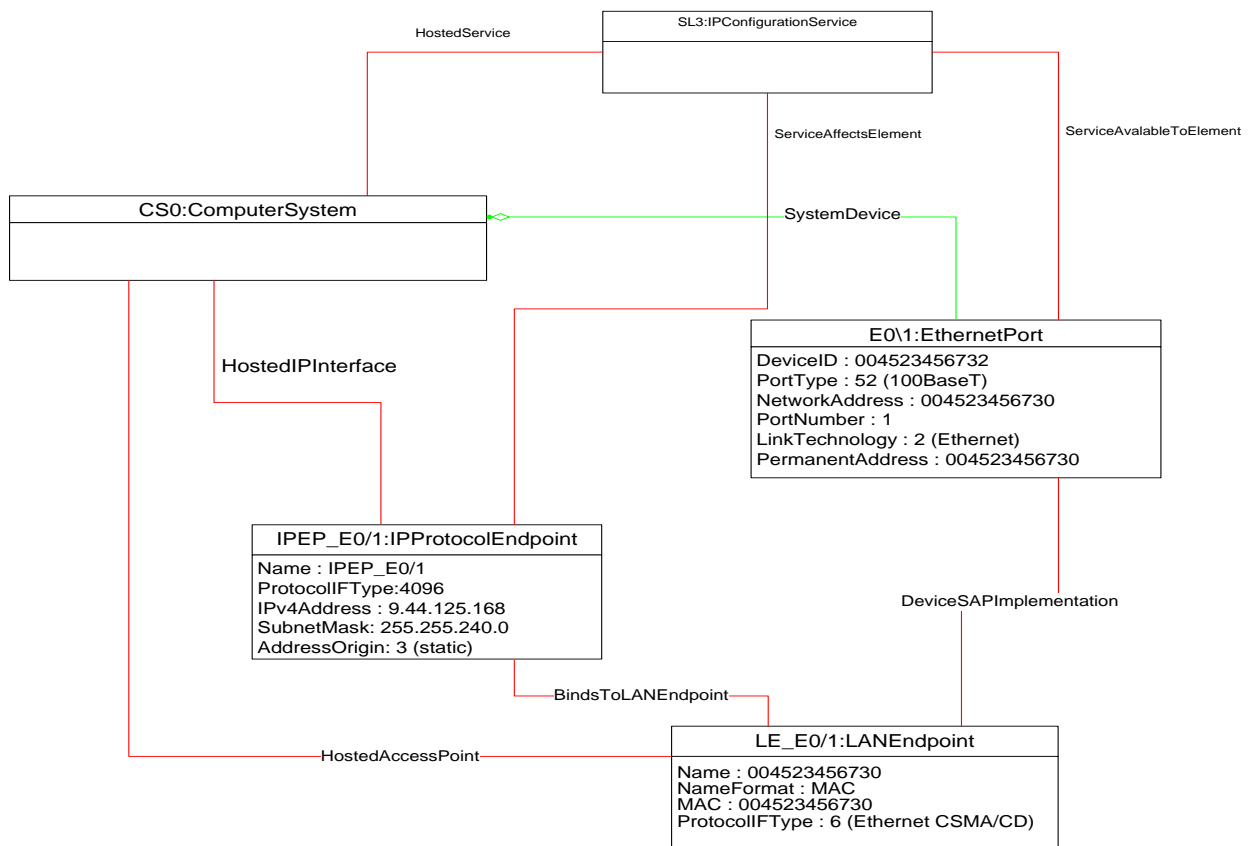
611 **9.5 Add an IPProtocolEndpoint to an Ethernet Port.**

612 The object diagram shown in Figure 6 contains the basic elements used to add an IP address to an  
 613 Ethernet Port. The diagram shows an instance of **CIM\_IPProtocolEndpoint** associated to  
 614 **CIM\_LANEndpoint** instance **LE\_E0/1**, the **CIM\_LANEndpoint** instance for the **CIM\_EthernetPort**  
 615 **E0\1**. The diagram also shows that **CIM\_IPProtocolEndpoint** instance is associated with the scoping  
 616 **CIM\_ComputerSystem** instance through **CIM\_HostedIPInterface**. This is a very simple diagram, not  
 617 shown are many of the required properties of the relative profiles for the objects shown.

618 The **CIM\_IPProtocolEndpoint** interface was created using the **CIM\_IPConfigurationService** instance, **SL3**,  
 619 associated with the target **CIM\_EthernetPort** through **CIM\_ServiceAvailableToElement**. The  
 620 **IPProtocolEndpoint** instance was added through the **CIM\_IPConfigurationService.AddProtocolEndpoint()**  
 621 method with the following parameters. Note this is for illustration purposes and other properties from the  
 622 class **CIM\_IPProtocolEndpoint** and other base classes may be specified as required.

- 623 • TargetInterface – Wbem URI reference to E0\1:CIM\_EthernetPort
- 624 • IPProtocolEndpoint[] –
  - 625 ○ Embedded Instance of **CIM\_IPProtocolEndpoint** {
  - 626 IPv4Address=9.44.125.168
  - 627 SubnetMask: 255.255.240.0
  - 628 ProtocolIFType=4096}

629  
630  
631  
632



633  
634  
635  
636  
637

**Figure 6 - IPProtocolEndpoint**

638 **10 CIM Elements**

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

642  
643  
644

**Table 8 – CIM Elements: Network Management Layer 3 Interface Profile**

645

Element Name	Requirement	Description
<b>Classes</b>		
CIM_BindsToLANEndpoint	Optional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
CIM_HostedService	Conditional	See 7.1.1



Element Name	Requirement	Description
CIM_HostedIPInterface	Conditional	See 7.2.1.1 and 7.2.1.2
CIM_IPProtocolEndpoint	Conditional	See 7.2.1
CIM_IPConfigurationService	Mandatory	See 7.1.1
CIM_LoopbackInterface	Conditional	See 7.2.1.3
CIM_RegisteredProfile	Optional	See
CIM_ServiceAffectsElement	Conditional	See 7.1.1
CIM_ServiceAvalableToElement	Conditional	See <b>Error! Reference source not found.</b>
CIM_Subinterface	Optional	See 7.2.1.4
CIM_SwitchVirtualInterface	Optional	See 7.2.1.5
CIM_IPEncapsulationInterface	Optional	See 7.2.1.6
<b>Indications</b>		
None defined in this profile		

646

647 **10.1 CIM\_BindsToLANEndpoint**

648 CIM\_BindsToLANEndpoint relates the CIM\_IPProtocolEndpoint instance with the CIM\_LANEndpoint  
 649 instance on which it depends. Table 9 provides information about the properties of  
 650 CIM\_BindsToLANEndpoint.

651 **Table 9 – Class: CIM\_BindsToLANEndpoint**

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

652 **10.2 CIM\_HostedService**

653 CIM\_HostedService relates the CIM\_IPConfigurationService instance to its scoping  
 654 CIM\_ComputerSystem instance. Table 10 provides information about the properties of  
 655 CIM\_HostedService.

656 **Table 10 – Class: CIM\_HostedService**

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

657 **10.3 CIM\_IPConfigurationService**

658 CIM\_IPConfigurationService provides the methods to create and delete a Layer 3 interface. Table 11  
 659 provides information about the properties of CIM\_IPConfigurationService that are in addition to those  
 660 specified in [DSP1036 IP Interface Profile 1.1.1](#).

661 **Table 11 – Class: CIM\_IPConfigurationService**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
ElementName	Mandatory	Pattern ".*"
AddIPProtocolEndpoint( )	Mandatory	See 8.1.2.
RemoveIPProtocolEndpoint( )	Mandatory	See 8.1.3.

662

663

664 **10.4 CIM\_IPProtocolEndpoint**

665 CIM\_IPProtocolEndpoint represents an IP interface that is associated with an Ethernet interface. Table 12  
 666 provides information about the properties of CIM\_IPProtocolEndpoint.

667 **Table 12 – Class: CIM\_IPProtocolEndpoint**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
NameFormat	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
ProtocolIFType	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
RequestedState	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
EnabledState	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
ElementName	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
RequestStateChange( )	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
IPv4Address	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
SubnetMask	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
AddressOrigin	Mandatory	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
IPv6Address	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>
IPv6AddressType	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>

Elements	Requirement	Description
IPv6SubnetPrefixLength	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>

668

669 **10.5 CIM\_IPEncapsulationInterface**

670 CIM\_IPEncapsulationInterface represents the IP tunnel interface used to route to connect two disjointed  
 671 IP networks. Table 13 provides information about the additional properties of  
 672 CIM\_IPEncapsulationInterface that are in addition to those in CIM\_IPProtocolEndpoint 10.3 Table 12.

673

**Table 13 – Class: CIM\_IPEncapsulationInterface**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
ElementName	Mandatory	Pattern ".*"
TunnelMode	Mandatory	<b>See 7.2.1.6</b>
VRFContext	Optional	<b>See 7.2.1.6</b>
TunnelDestinationIPAddress	Optional	<b>See 7.2.1.6</b>
TunnelSourceInterface	Optional	<b>See 7.2.1.6</b>

674

675

676

677 **10.6 CIM\_SubInterface**

678 CIM\_SubInterface represents a sub division of an Ethernet interface interface. Table 14 provides  
 679 information about the additional properties of CIM\_SubInterface that are in addition to those in  
 680 CIM\_IPProtocolEndpoint 10.3 Table 12.

681

**Table 14 – Class: CIM\_SubInterface**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
ParentInterface	Mandatory	<b>See 7.2.1.4</b>
ElementName	Mandatory	Pattern ".*"
EncapsulationType	Mandatory	<b>See 7.2.1.4</b>

Elements	Requirement	Description
OtherEncapsulationType	Conditional	See 7.2.1.4
EncapsulationValue	Conditional	See 7.2.1.4

## 682 10.7 CIM\_SwitchVirtualInterface

683 CIM\_SwitchVirtualInterface represents the IP protocol endpoint used to route a VLAN within a switch.  
 684 Table 15 provides information about the additional properties of CIM\_SwitchVirtualInterface that are in  
 685 addition to those in CIM\_IPProtocolEndpoint 10.3 Table 12.

686 **Table 15 – Class: CIM\_SwitchVirtualInterface**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	Pattern ".*"
VLANId	Mandatory	See 7.2.1.5

## 687 10.8 CIM\_RegisteredProfile

688 CIM\_RegisteredProfile identifies the *Network Management Layer3 Interface Profile* in order for a client to  
 689 determine whether an instance of CIM\_IPProtocolEndpoint is conformant with this profile. The  
 690 CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the exception of the  
 691 mandatory values specified for the properties in Table 16, the behavior of the CIM\_RegisteredProfile  
 692 instance is in accordance with the [Profile Registration Profile](#).

693 **Table 16 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Network Management L3 Interface Profile".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

694

695  
696  
697  
698

## **ANNEX A (informative)**

### **Change log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0a	2013-09-22	DMTF Work in Progress

699