



1  
2  
3  
4  
5

**Document Identifier: DSP1063**

**Date: 2018-04-23**

**Version: 1.0.0c**

## 6 **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 its members. Because this document is a Work in Progress, this document may still change, perhaps profoundly and without notice. This document is available for public review and comment until superseded.

**Provide any comments through the DMTF Feedback Portal:**

<http://www.dmtf.org/standards/feedback>

7 **Supersedes: None**  
8 **Document Class: Normative**  
9 **Document Status: Work in Progress**  
10 **Document Language: en-US**  
11

12 Copyright Notice

13 Copyright © 2012-2013, 2018 Distributed Management Task Force, Inc. (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  
17 to 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28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

36

# CONTENTS

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

86

87 **Figures**

88	Figure 1 – Network Management Layer3 Interface Profile: Class diagram .....	10
89	Figure 2 – Registered profile.....	18
90	Figure 3 – IPSubinterface .....	20
91	Figure 4 - Switch Virtual Interface .....	21
92	Figure 5 - Loopback Interface .....	22
93	Figure 6 - IPProtocolEndpoint.....	23
94		

95 **Tables**

96	Table 1 – Referenced profiles.....	9
97	Table 2 – AddIPProtocolEndpoint ( ) Method: Parameters.....	14
98	Table 3 – RemoveIPProtocolEndpoint ( ) Method: Parameters.....	15
99	Table 4 – Operations: CIM_BindsToLANEndpoint .....	15
100	Table 5 – Operations: CIM_HostedService .....	16
101	Table 6 – Operations: CIM_HostedIPInterface .....	16
102	Table 7 – Operations: CIM_IPProtocolEndpoint.....	17
103	Table 8 – CIM Elements: Network Management Layer 3 Interface Profile.....	23
104	Table 9 – Class: CIM_BindsToLANEndpoint .....	24
105	Table 10 – Class: CIM_HostedService .....	24
106	Table 11 – Class: CIM_IPConfigurationService.....	25
107	Table 12 – Class: CIM_IPProtocolEndpoint.....	25
108	Table 13 – Class: CIM_IPSubinterface.....	26
109	Table 14 – Class: CIM_SwitchVirtualInterface.....	26
110	Table 15 – Class: CIM_RegisteredProfile.....	27
111		

112

## Foreword

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

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

## 117 **Acknowledgments**

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

119 Editors:

- 120 • John Parchem – Microsoft Corporation – DMTF Fellow

121 Contributors:

- 122 • John Crandall – Brocade Communications System
- 123 • Dr. Bhumip Khasnabish - ZTE Corporation
- 124 • Lawrence Lamers – VMware
- 125 • John Leung – Intel
- 126 • Steve Neely – Cisco Systems
- 127 • Shishir Pardikar – Citrix
- 128 • Hemal Shah – Broadcom Corporation
- 129 • Alex Zhdankin – Cisco Systems

130

131

## Introduction

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

## 137 Document conventions

### 138 Typographical conventions

139 The following typographical conventions are used in this document:

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

142

143

# Network Management Layer3 Interface Profile

## 144 1 Scope

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

## 149 2 Normative references

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

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

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

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

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

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

164 DMTF DSP1097, *Virtual Ethernet Switch Profile 1.1*,  
165 [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)

166 DMTF DSP1036 *IP Interface Profile 1.1.1*,  
167 [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)

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

## 170 3 Terms and definitions

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

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

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

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

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

186 **3.1**  
187 **conditional**

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

190 **3.2**  
191 **mandatory**

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

194 **3.3**  
195 **optional**

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

197 **3.4**  
198 **pending configuration**

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

201 **3.5**  
202 **referencing profile**

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

205 **3.6**  
206 **unspecified**

207 indicates that this profile does not define any constraints for the referenced CIM element or operation  
208

## 209 **4 Symbols and abbreviated terms**

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

212 **4.1**  
213 **IP**

214 Internet Protocol

215 **4.2**  
216 **VLAN**

217 Virtual Local Area Network

218



219 **4.3**  
 220 **VSI**  
 221 Virtual Switch Interface

222 **5 Synopsis**

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

224 **Version:** 1.0.0.c

225 **Organization:** DMTF

226 **CIM Schema version:** 2.51

227 **Central class:** CIM\_IPConfigurationService

228 **Scoping class:** CIM\_System

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

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

234 **Table 1 – Referenced profiles**

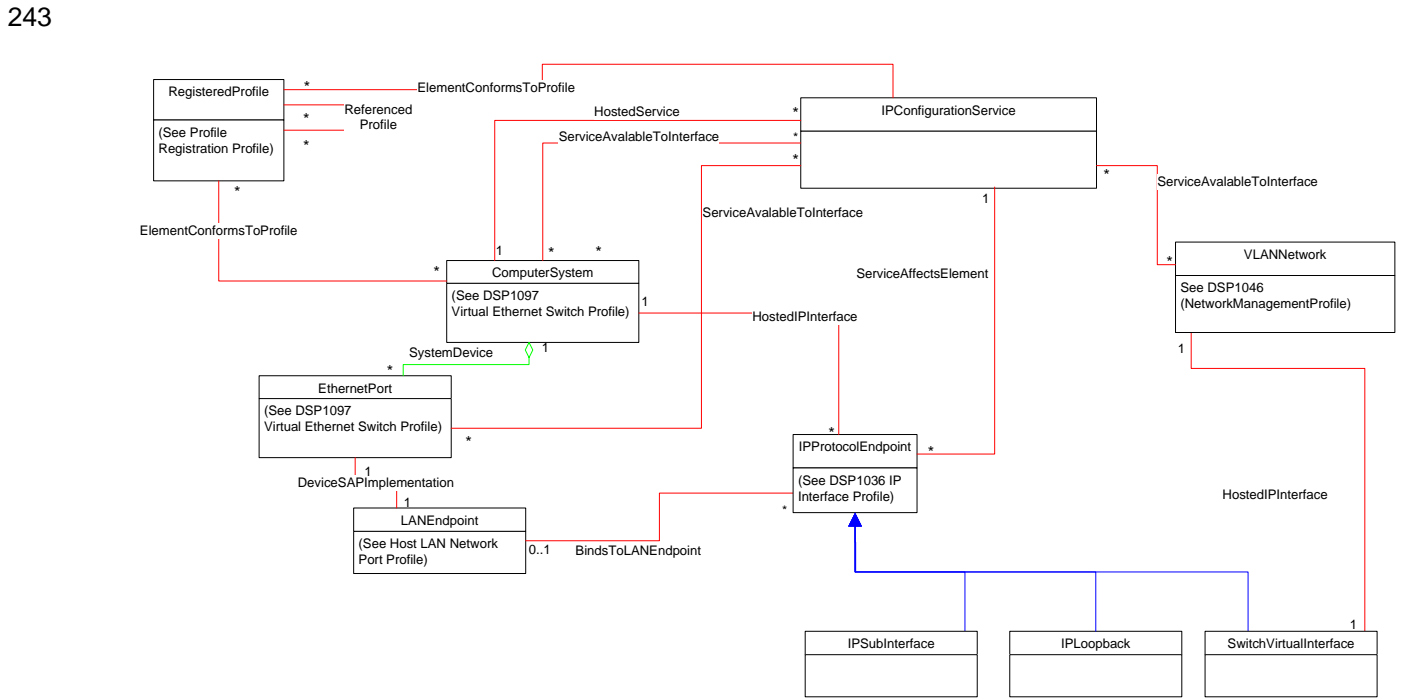
Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None
Virtual Ethernet Switch	DMTF	1.1	Mandatory	None
IP Configuration Profile	DMTF	1.1	Optional	None
IP Interface Profile	DMTF	1.1.1	Mandatory	None
Host LAN Network	DMTF	1.0.2	Mandatory	None
Network Management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

235 **6 Description**

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

240 **6.1 Class diagram**

241 Figure 1 represents the class schema for the *Network Management Layer3 Interface Profile*. For  
 242 simplicity, the CIM\_ prefix has been removed from the names of the classes.



244  
 245

246 **Figure 1 – Network Management Layer3 Interface Profile: Class diagram**

247 Figure 1 is a class diagram for the network layer 3 interfaces within the context of an Ethernet switch  
 248 represented as a [DSP1097](#) Virtual Ethernet Switch Profile compliant switch. The class  
 249 IPProtocolEndpoint and the subclasses of IPProtocolEndpoint, CIM\_Subinterface, CIM\_Loopback and  
 250 CIM\_SwitchVirtualInterface all represent the management aspects the typical layer 3 interfaces found in  
 251 an Ethernet switch to facilitate IP routing capabilities.

252 **6.2 CIM\_IPProtocolEndpoint**

253 In this profile the CIM\_IPProtocolEndpoint is used to provide IP configuration for the Ethernet ports in the  
 254 switch. It is also the super class for all of the Layer 3 interfaces in the switch. The  
 255 CIM\_IPProtocolEndpoints that are representing switch interfaces are associated through an instance of  
 256 CIM\_HostedIPInterface either to the CIM\_ComputerSystem instance representing the scoping class or to  
 257 a Network class or subclass such as CIM\_VLANNetwork for the case of a switch virtual interface.

258 **6.3 CIM\_SwitchVirtualInterface**

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

## 261 6.4 CIM\_IPSubinterface

262 An IPSubinterface subdivides a single switch port into multiple IP subnets. This is typically done using  
263 Dot1Q encapsulation using VLANIds to distinguish the subnets. Even though an IPSubinterface may  
264 have a VLANId within the scoped router this is a layer 3 interface and this interface is not a part of an  
265 internal VLANNetwork with the same VLANId.

## 266 6.5 CIM\_IPLoopback

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

## 270 6.6 CIM\_HostedIPInterface

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

## 273 6.7 CIM\_IPConfigurationService

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

# 278 7 Implementation

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

## 281 7.1 Representing the layer3 interface management capabilities

### 282 7.1.1 CIM\_IPConfigurationService

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

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

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

289 IPProtocolEndpoint instances created through the use of an instance of CIM\_IPConfigurationService shall  
290 be associated to the CIM\_IPConfigurationService instance through an instance of  
291 CIM\_ServiceAffectsElement.

## 292 7.2 Representing Layer 3 Interfaces

### 293 7.2.1 CIM\_IPProtocolEndpoint

294 Instances of CIM\_IPProtocolEndpoint created as a result of the  
295 CIM\_IPConfigurationService.AddIPProtocolEndpoint () shall comply with the requirements of [DSP1036 IP](#)  
296 [Interface Profile 1.1](#) where CIM\_IPProtocolEndpoint is the central class of [DSP1036](#). The additional  
297 requirements listed in this cause and its sub clauses are in addition to requirements in [DSP1036](#).

### 298 7.2.1.1 CIM\_IPProtocolEndpoint (CIM\_EthernetPort)

299 Instances of CIM\_IPProtocolEndpoint created as a result of the  
300 CIM\_IPConfigurationService.AddIPProtocolEndpoint () method targeting an instance of CIM\_EthernetPort  
301 shall be associated with the instance of CIM\_LANEndpoint associated to the CIM\_EthernetPort instance,  
302 that was specified as the TargetInterface of the method call, through an instance of  
303 CIM\_BindsToLANEndpoint. This instance of CIM\_IPProtocolEndpoint shall also be associated through an  
304 instance of CIM\_HostedIPInterface to the scoping instance of CIM\_ComputerSystem.

### 305 7.2.1.2 CIM\_IPLoopbackInterface

306 Represents a single IP endpoint communication channel. CIM\_IPLoopbackInterface shall conform to  
307 **Error! Reference source not found..** The instance of CIM\_System described in **Error! Reference**  
308 **source not found.** shall be the instance of the class scoping class instance of CIM\_ComputerSystem.

### 309 7.2.1.3 CIM\_IPSubinterface

310 Represents the subdivision of a single port into multiple IP subnets. CIM\_IPSubinterface shall conform to  
311 7.2.1.1.

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

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

### 321 7.2.1.4 CIM\_SwitchVirtualInterface

322 Represents the IP settings for a VLAN to allow layer 3 routing between VLANs.  
323 CIM\_SwitchVirtualInterface shall conform to **Error! Reference source not found..** The instance of  
324 CIM\_System described in **Error! Reference source not found.** shall be an instance of the class  
325 CIM\_VLANNetwork.

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

## 330 8 Methods

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

### 333 8.1 Extrinsic methods

334 If synchronous execution of a method succeeds, the implementation shall set a return value of  
335 0 (Completed with No Error).

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

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

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

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

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

- 348 • Set the value of the JobState property to 7 (Completed).
- 349 • Provide an instance of the CIM\_AffectedJobEntity association with property values set as fol-  
350 lows:
  - 351 – The value of the AffectedElement property shall refer to the object that represents the top-  
352 level entity that was created or modified by the asynchronous task. For example, for the  
353 CIM\_IPConfigurationService. AddIPProtocolEndpoint() method, this is an instance of the  
354 CIM\_IPProtocolEndpoint class
  - 355 – The value of the AffectingElement property shall refer to the instance of the  
356 CIM\_ConcreteJob class that represents the completed asynchronous task.
  - 357 – The value of the first element in the ElementEffects[ ] array property (ElementEffects[0])  
358 shall be set to 5 (Create) for the CIM\_IPConfigurationService. AddIPProtocolEndpoint()  
359 method. Otherwise, this value shall be 0 (Unknown).

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

#### 362 **8.1.1.1 Job parameter**

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

- 365 • If the method execution is performed synchronously, the implementation shall set the value to  
366 NULL.
- 367 • If the method execution is performed asynchronously, the implementation shall set the value to  
368 refer to the instance of the CIM\_ConcreteJob class that represents the asynchronous task.

369

#### 370 **8.1.2 CIM\_IPConfigurationService. AddIPProtocolEndpoint()**

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

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

378 Table 2 contains requirements for parameters of this method.

379

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

380 **8.1.2.1.1 TargetInterface**

381 A required reference to an associated target interface, system or network. The supported target interfaces  
 382 for a CIM\_IPProtocolEndpoint class or subclass supported should be as described in the sub clauses of  
 383 7.2.

384 **8.1.2.1.2 IPProtocolEndpoint[]**

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

389 **8.1.2.1.3 EndpointSettings[]**

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

396 **8.1.2.1.4 ResultingEndpoint[]**

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

399 **8.1.2.1.5 Job**

400 See 8.1.1.1

401 **8.1.3 CIM\_IPConfigurationService. RemoveIPProtocolEndpoint()**

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

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

407 Table 3 contains requirements for parameters of this method.

408

**Table 3 – RemoveIPProtocolEndpoint ( ) Method: Parameters**

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

409 **8.1.3.1.1 Endpoint**

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

411 **8.1.3.1.2 Job**

412 See 8.1.1.1

413 **8.2 Profile conventions for operations**

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

416 The default list of operations is as follows:

- 417 • GetInstance
- 418 • EnumerateInstances
- 419 • EnumerateInstanceNames
- 420 • Associators
- 421 • AssociatorNames
- 422 • References
- 423 • ReferenceNames

424 **8.3 CIM\_BindsToLANEndpoint**

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

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

429

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

<b>Operation</b>	<b>Requirement</b>	<b>Messages</b>
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

430 **8.4 CIM\_HostedService**

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

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

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

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

436 **8.5 CIM\_HostedIPInterface**

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

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

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

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

442 **8.6 CIM\_L3InterfaceConfigurationService**

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

444 **8.7 CIM\_IPSubinterface**

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

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

447

448 **8.8 CIM\_IPLoopbackInterface**

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

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



451 **8.9 CIM\_SwitchVirtualInterface**

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

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

454 **8.10 CIM\_IPProtocolInterface**

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

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

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

Operation	Requirement	Messages
ModifyInstance	Conditional. See <a href="#">DSP1036_1.1</a>	None

458 **8.11 CIM\_IPSubinterface**

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

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

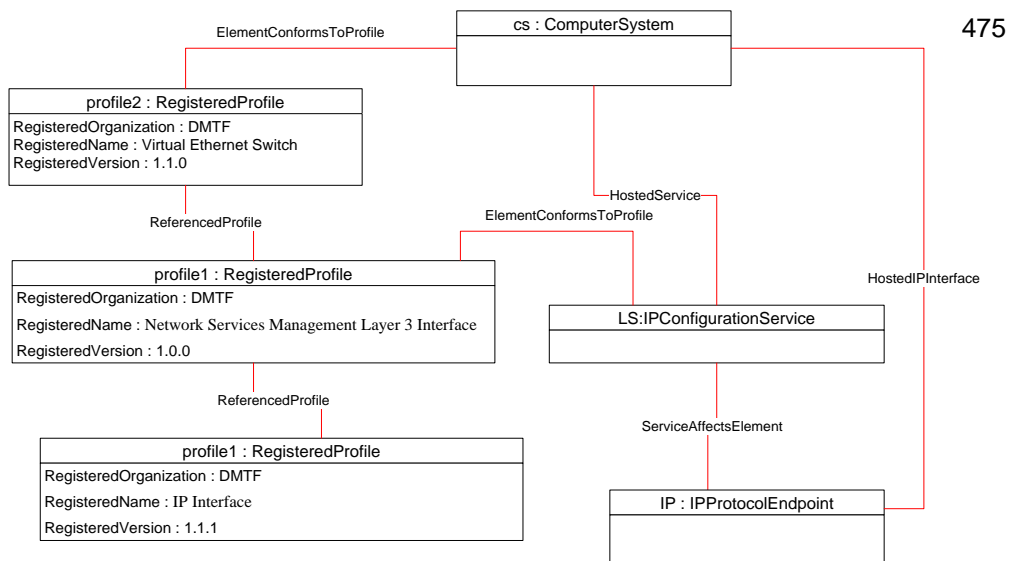
461 **9 Use cases**

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

464 **9.1 Profile registration**

465 The object diagram in Figure 2 shows one possible method for advertising profile conformance. The  
 466 instances of CIM\_RegisteredProfile are used to identify the version of the Network Management Layer3  
 467 Interface Profile with which an instance of CIM\_IPConfigurationService is conformant. An instance of  
 468 CIM\_RegisteredProfile exists for each profile that is instrumented in the system. One instance of  
 469 CIM\_RegisteredProfile identifies the “VirtualEthernetSwitch1.1.0”. The other instance identifies the  
 470 “Network Management Layer3 Interface Profile”. The CIM\_IPConfigurationService instance is scoped to  
 471 an instance of CIM\_ComputerSystem. This instance of CIM\_ComputerSystem is conformant with the  
 472 DMTF *Virtual Ethernet Switch Profile* version 1.1.0 as indicated by the CIM\_ElementConformsToProfile  
 473 association to the CIM\_RegisteredProfile instance.

474



476

477

**Figure 2 – Registered profile**

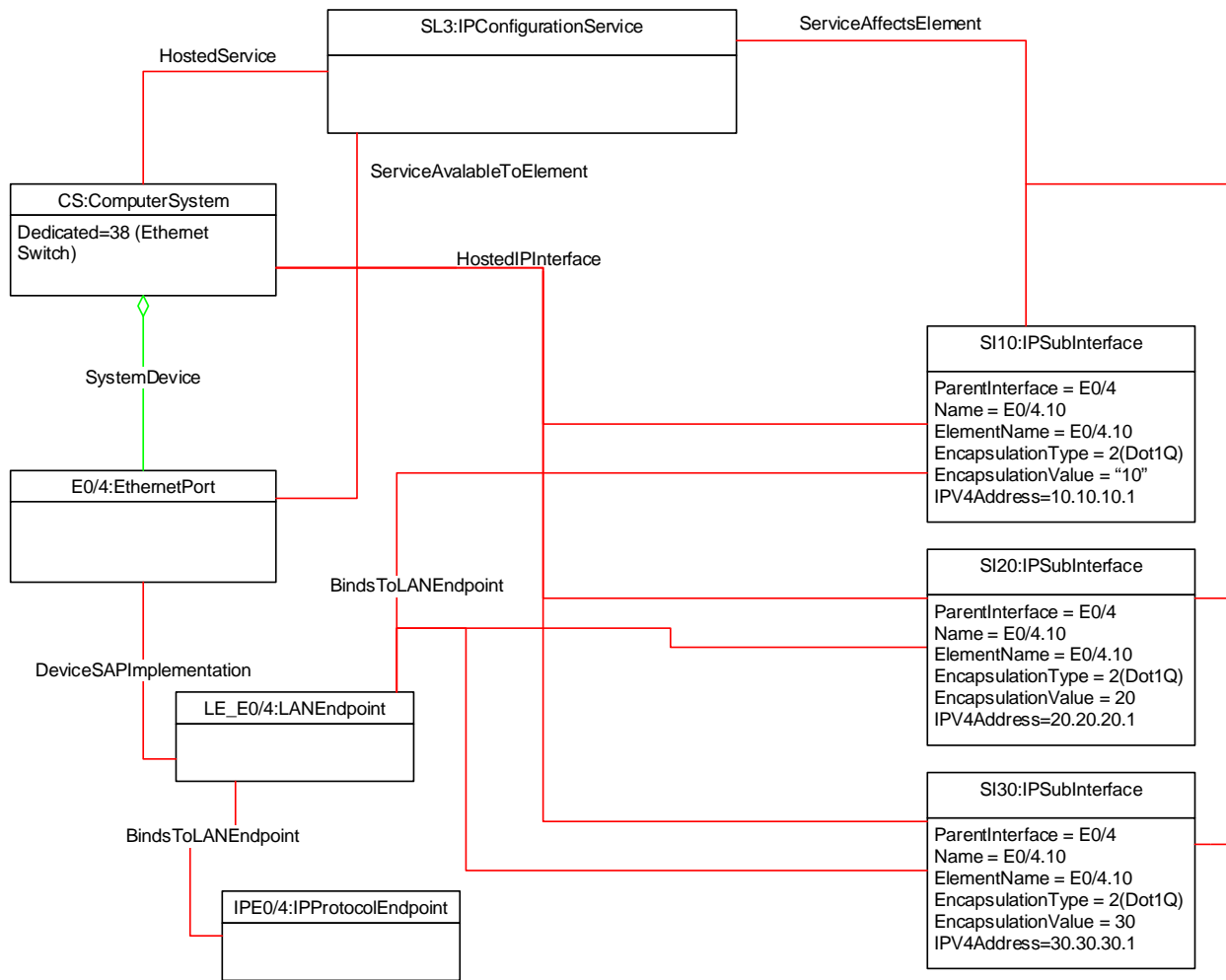
478

479 **9.2 IPSubinterface**

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

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

- 490 • TargetInterface – Wbem URI reference to E0/4
- 491 • IPProtocolEndpoint[] –
  - 492 ○ Embedded Instance of CIM\_IPSubinterface {
  - 493 ElementName = E0/4.10
  - 494 EncapsulationType = 2
  - 495 EncapsulationValue = 10
  - 496 IPv4Address=10.10.10.
  - 497 ProtocolIFType=4060}
  - 498
  - 499 ○ Embedded Instance of CIM\_IPSubinterface {
  - 500 ElementName = E0/4.20
  - 501 EncapsulationType = 2
  - 502 EncapsulationValue = 20
  - 503 IPv4Address=20.20.20.1
  - 504 ProtocolIFType=4060}
  - 505
  - 506 ○ Embedded Instance of CIM\_IPSubinterface {
  - 507 ElementName = E0/4.30
  - 508 EncapsulationType = 2
  - 509 EncapsulationValue = 30
  - 510 IPv4Address=30.30.30.1
  - 511 ProtocolIFType=4060}
  - 512
  - 513



514  
515

516

**Figure 3 – IPSubinterface**

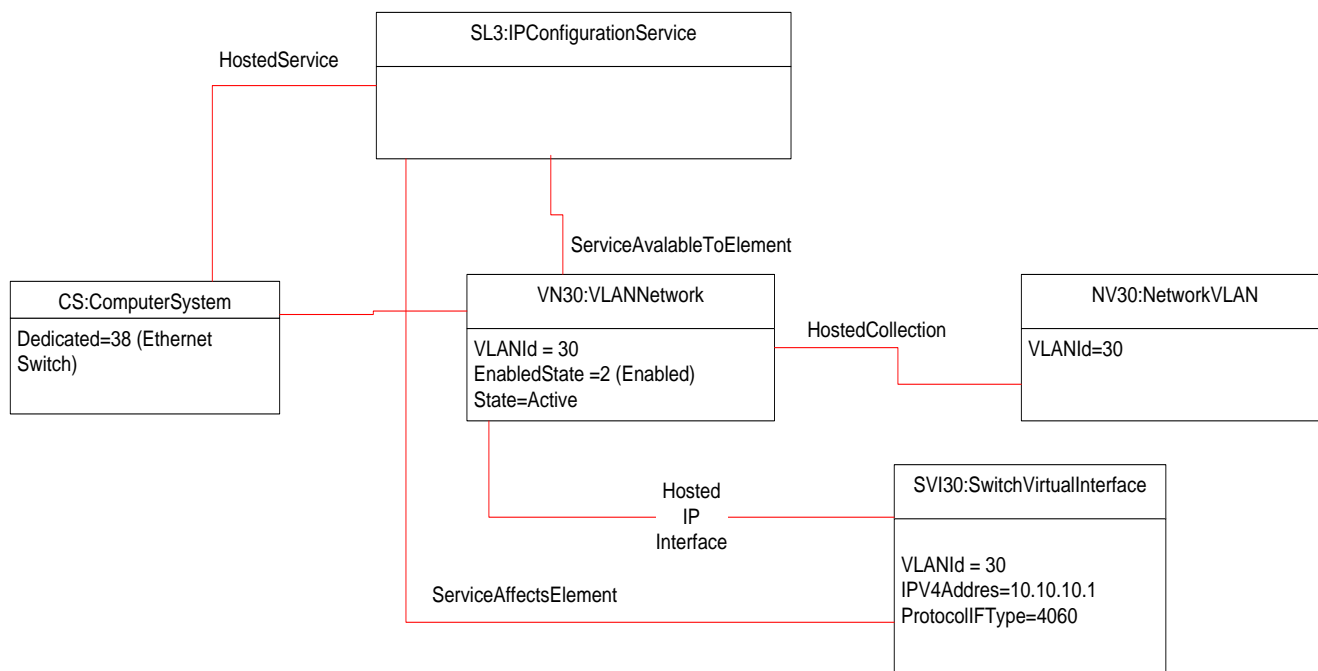
517 **9.3 Switch Virtual Interface**

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

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

- 529       • TargetInterface – WBEM URI reference to VN30  
 530       • IPProtocolEndpoint[] –

- 531 ○ Embedded Instance of CIM\_SwitchVirtualInterface {
- 532 IPv4Address=10.10.10.1
- 533 ProtocolIFType=4060}
- 534



535  
536

Figure 4 - Switch Virtual Interface

537

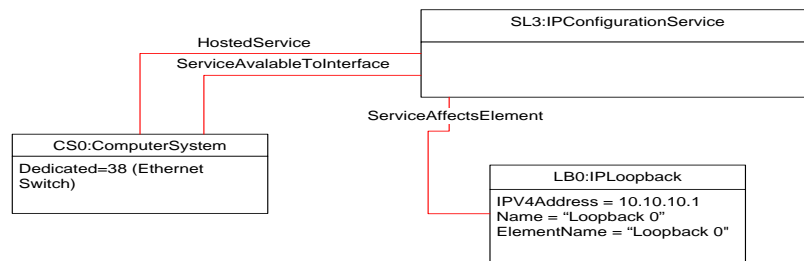
### 538 9.4 Loopback Interface

539 The object diagram shown in Figure 5 contains the basic elements used to model configuration of a  
 540 loopback interface. The diagram shows that CIM\_ComputerSystem has an associated instance through  
 541 the association CIM\_HostedIPInterface. This is a loopback interface CIM\_Loopback:LB0. This is a very  
 542 simple diagram, not shown are many of the required properties of the relative profiles for the objects  
 543 shown.

544 The interface was created with a CIM\_IPConfigurationService.AddProtocolEndpoint() method with the  
 545 following parameters. Note this is for illustration purposes and other properties from the super class  
 546 CIM\_IPProtocolEndpoint and other base classes may be specified as required.

- 547 • TargetInterface – WBEM URI reference to CIM\_ComputerSystem:CS0
- 548 • IPProtocolEndpoint[] –
  - 549 ○ Embedded Instance of CIM\_Loopback {
  - 550 IPv4Address=10.10.10.1
  - 551 ProtocolIFType=4060}
  - 552
  - 553
  - 554

555



556

557

Figure 5 - Loopback Interface

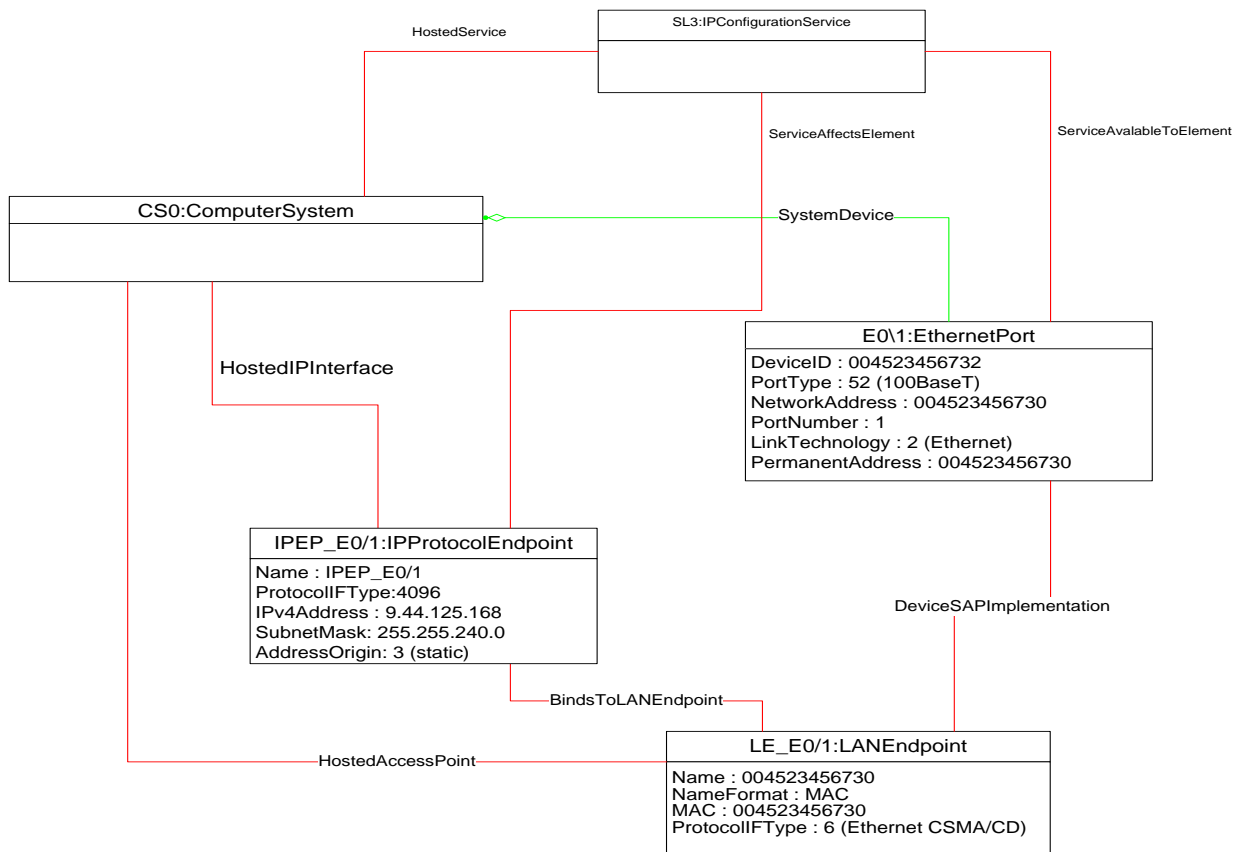
558

### 559 9.5 Add an IPProtocolEndpoint to an Ethernet Port.

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

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

- 571 • TargetInterface – WBEM URI reference to E0\1:CIM\_EthernetPort
- 572 • IPProtocolEndpoint[] –
- 573     o Embedded Instance of CIM\_IPProtocolEndpoint {
- 574         IPv4Address=9.44.125.168
- 575         SubnetMask: 255.255.240.0
- 576         ProtocolIFType=4096}
- 577



578  
579

580

Figure 6 - IPProtocolEndpoint

581 **10 CIM Elements**

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

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

586

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
CIM_HostedIPInterface	Conditional	See 7.2.1.1 and <b>Error! Reference source not found.</b>
CIM_IPProtocolEndpoint	Conditional	See 7.2.1
CIM_IPConfigurationService	Mandatory	See 7.1.1
CIM_IPLoopbackInterface	Conditional	See 7.2.1.2

Element Name	Requirement	Description
CIM_RegisteredProfile	Optional	See
CIM_ServiceAffectsElement	Conditional	See 7.1.1
CIM_ServiceAvalableToElement	Conditional	See 7.1.1
CIM_IPSubinterface	Optional	See 7.2.1.3
CIM_SwitchVirtualInterface	Optional	See 7.2.1.4
<b>Indications</b>		
None defined in this profile		

### 587 10.1 CIM\_BindsToLANEndpoint

588 CIM\_BindsToLANEndpoint relates the CIM\_IPProtocolEndpoint instance with the CIM\_LANEndpoint  
 589 instance on which it depends. Table 9 provides information about the properties of  
 590 CIM\_BindsToLANEndpoint.

591 **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

### 592 10.2 CIM\_HostedService

593 CIM\_HostedService relates the CIM\_IPConfigurationService instance to its scoping  
 594 CIM\_ComputerSystem instance. Table 10 provides information about the properties of  
 595 CIM\_HostedService.

596 **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 *

### 597 10.3 CIM\_IPConfigurationService

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



601

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

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

602 **10.4 CIM\_IPProtocolEndpoint**

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

605

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

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
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>
IPv6SubnetPrefixLength	Conditional	See <a href="#">DSP1036 IP Interface Profile 1.1.1</a>

606

607 10.5 **CIM\_IPSubinterface**

608 CIM\_IPSubinterface represents a sub division of an Ethernet interface interface. Table 13 provides  
 609 information about the additional properties of CIM\_IPSubinterface that are in addition to those in  
 610 CIM\_IPProtocolEndpoint 10.3 Table 12.

611 **Table 13 – Class: CIM\_IPSubinterface**

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 7.2.1.3</b>
ElementName	Mandatory	Pattern ".*"
EncapsulationType	Mandatory	<b>See 7.2.1.3</b>
OtherEncapsulationType	Conditional	<b>See 7.2.1.3</b>
EncapsulationValue	Conditional	<b>See 7.2.1.3</b>

612 10.6 **CIM\_SwitchVirtualInterface**

613 CIM\_SwitchVirtualInterface represents the IP protocol endpoint used to route a VLAN within a switch.  
 614 Table 14 provides information about the additional properties of CIM\_SwitchVirtualInterface that are in  
 615 addition to those in CIM\_IPProtocolEndpoint 10.3, Table 12.

616 **Table 14 – Class: CIM\_SwitchVirtualInterface**

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 ".*"
VLANId	Mandatory	<b>See 7.2.1.4</b>

617 **10.7 CIM\_RegisteredProfile**

618 CIM\_RegisteredProfile identifies the *Network Management Layer3 Interface Profile* in order for a client to  
 619 determine whether an instance of CIM\_IPProtocolEndpoint is conformant with this profile. The  
 620 CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the exception of the  
 621 mandatory values specified for the properties in Table 15, the behavior of the CIM\_RegisteredProfile  
 622 instance is in accordance with the [Profile Registration Profile](#).

623 **Table 15 – 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.c".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

624

625  
626  
627  
628

## **ANNEX A (informative)**

### **Change log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0a	2013-10-01	DMTF Work in Progress
1.0.0b	2017-02-13	DMTF Work in Progress; removed Encapsulation Interface
1.0.0c	2018-04-23	DMTF Work in Progress

629