

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2020

D. Shyti
L. Beylier
SFR
L. Iannone
Telecom ParisTech
October 19, 2019

**A YANG Module for uCPE management.
draft-shyti-opsawg-vysm-04**

Abstract

This document provides a YANG data model for uCPE management (VYSM) and definition of the uCPE equipment. The YANG Service Model serves as a base framework for managing an universal Customer-Premises Equipment (uCPE) subsystem. The model can be used by a Network Service Orchestrator.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4](#).e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	2
3.	Universal CPE	3
3.1.	uCPE purpose	4
3.2.	uCPE VNF ecosystem example	4
3.3.	Internal uCPE service example	5
4.	YANG Service Model for uCPE management	6
5.	uCPE YANG Service Model tree diagram overview	7
6.	Specification of the VNF YANG Service Model	9
6.1.	uCPE main yang module	10
6.2.	uCPE ports yang module for each interface	15
6.3.	uCPE interfaces yang module	16
7.	XML example	39
8.	Security Considerations	41
9.	IANA Considerations	41
10.	Acknowledgements	41
11.	Normative References	42
	Authors' Addresses	42

[1.](#) Introduction

Network Function Virtualization is a technology that allows to virtualize the network services running on dedicated hardware. This technology became a base for universal Customer-Premises Equipment (uCPE). This document defines the uCPE as hardware with x86 capabilities that has a hypervisor. In other words, uCPE is a host that may run multiple Virtual Machines with guest OSs, where each Guest OS may represent a Physical Network Function. This document presents the YANG Service Model (VYSM) to manage from an Orchestrator the infrastructure inside the uCPE.

[2.](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Link - is an entity that enables link layer communication of nodes.

Port - node connector to the link.

NE - Network Element.

NSYM - Network Service Yang Module.

VYSM - VNF YANG Service Model.

3. Universal CPE

Firstly, this document defines the platform that is controlled with VYSM - universal CPE (uCPE). The uCPE as hardware with x86 capabilities that is generally running Linux distribution with additional virtualisation layer. Virtualization layer provides virtual compute, virtual storage and virtual network resources. Each VNF running in the uCPE requires the amount of virtual resources (for example: 4 vCPUs, 4GB RAM, 40GB storage, 4 vPorts). VNFs MAY be interconnected between each other and physical ports via Virtual Networks. Topology construction and VM lifecycle management is allowed via high level interface (Configuration can be done in the same transaction). The figure below presents the uCPE architecture.

VNF1	VNF2	VNF3	
Virtual Compute	Virtual Storage	Virtual Networks	uCPE software
PHY x86 processor	RAM+PHY storage	PHYsical ports	uCPE Hardware

The next elements can be managed in the uCPE:

o Virtual Network Functions:

- * Number of assigned vCPUs.
- * Size of allocated RAM.
- * VNF day0 config (bootstrap).
- * vLinks that are attached to the VNF.

o Virtual Switches:

- * vLinks that are attached to the vSW.

o Virtual Links(vLinks).

- o Physical Ports of the uCPE.

3.1. uCPE purpose

- o uCPE replaces multiple types of equipment (Node#1 - Node#5) with 1 unit by virtualizing them as Virtual Network Functions on the top of NFVIs:

```

:      NODE #1      :      NODE #2 :      NODE #3      : NODE #4 : NODE #5 :
:      +-----+ :      +-----+ :      +-----+ :      +---+ :      +-----+ :
...-----|Aggregation|-----|CE-L2 |-----|CPE-L3|-----|FW|-----|SDWAN|---LAN
:      | switch  | :      |      | :      |      | :      | | :      |      | :
:      +-----+ :      +-----+ :      +-----+ :      +---+ :      +-----+ :

```

```

:      NODE #1      :      NODE #2      :
:      :      +.....+ :
:      +-----+ : | +-----+ +-----+ +---+ +-----+ | :
...---|Aggregation|---|--|CE-L2 |-----|CPE-L3|-----|FW|---|SDWAN|---|LAN
:      | switch  | : | |      |      |      |      | | :
:      +-----+ : | +-----+ +-----+ +---+ +-----+ | :
:      :      | universal Customer-Premises Equipment | :
:      :      +-----+ :

```

- o uCPE facilitates the interconnection between the Network Functions (NF) as interconnection between NF is performed via virtual links (that is part of the uCPE management). That means that no need to hire technician to cable the equipment, it could be done via orchestrator.
- o uCPE facilitates the 0day configuration of the VNFs as its 0day configuration can be putted remotely.

3.2. uCPE VNF ecosystem example

uCPE supports a Virtual Network Functions of different type:

- o SD-WAN
- o vRouter(vCPE)
- o vFirewall
- o vLB(vLoad Balancer)

- o vCGNAT(vCarrier Grade NAT)
- o virtual WAN Optimistaion
- o vWireless LAN controller
- o Other...

3.3. Internal uCPE service example

The VNF in the uCPE could be a vRouter or vFirewall or an SD-WAN that is not a default part of virtual network resources of the uCPE. Multiple VNFs MAY be instantiated in the uCPE. With support of links and swithes, VNFs MAY participate a service chains. Example of service chains (Note that virtual switch "vs(WAN)" connected to LAN ports and vSW(WAN) is connected to WAN ports):

- o vSW(WAN)-l1-vRouter(vCPE)-l2-vSW(LAN).
- o vSW(WAN)-l1-vRouter(vCPE)-l2-vSW(Service)-l3-vFirewall-l4-vSW(LAN).
- o vSW(WAN)-l1-vRouter(vCPE)-l2-vSW(Service1)-l3-vFirewall-l4-vSW(Service2)-l5-SD-WAN-l6-vSW(LAN).
- o vSW(WAN)-l1-SDWAN-l2-vSW(Service)-l3-vFirewall-l4-vSW(LAN).
- o

```

vSW(WAN1)--vRouter--+
                        +--vLoadBalance  vFirewall--vSW(LAN)
vSW(WAN2)--vRouter--+  |                |
                        +-vSW(Service1)+

```

o

```

vSW(WAN1)--vRouter(ISP1)--+
                        +--SD-WAN          vFirewall--vSW(LAN)
vSW(WAN2)--vRouter(ISP2)--+  |                |
                        +-vSW(Service1)+

```


4. YANG Service Model for uCPE management

Secondly, this document defines and classifies the VYSM as Network Service YANG Module(NSYM) layer component [RFC 8199](#) [[RFC8199](#)]. Thus it inherits the characteristics of the NSYM Layer. VYSM is a modeled representation of the specific service requirements. It provides abstraction of services configuration and operations that MAY be implemented in Network Elements (NEs). Thus VYSM does not describe all configuration to be performed on the devices, but provides the configuration that is required for the "Network Service to Network Element(s)" decomposition process [RFC 8199](#) [[RFC8199](#)]. Example of the decomposition is presented in the figure below.

The Network Service YANG module exposes the configuration commands via the Northbound interfaces of the orchestrator. Therefore the set of the commands modeled in the VYSM can be inputted via Northbound interfaces(for example CLI). In the example the command "vm VNF1" is passed via Northbound interface to the orchestrator. It defines the virtual machine name. Further the same configuration MAY be transformed to the one or multiple Network Element payloads (for example xml for NETCONF) that carry an equivalent of commands such as "nf nf-name VNF1"


```

+-----+
|               |
|  config t     |
|    vm VNF1    |
+-----+
#
#

+-----+
:
: +-----+
: | Network Service YANG Module |
: +-----+
: #
: ##### orchestrator
: # # #
: '-----' '-----' '-----'
: 'Module1' 'Module 2' 'Module3' <= Network Element
: '-----' '-----' '-----' YANG Modules
: # # #
: # # #####
: #### ##### #
: # # #
+-----+
# # #
Network # element 1   Network # element 2   Network # element3
+-----+ +-----+ +-----+
| domains domain VNF1| |tenants tenant name VNF1| |nf nf-name VNF1|
+-----+ +-----+ +-----+

```

5. uCPE YANG Service Model tree diagram overview

This section provides an overview of the Service YANG Model (VSYM) that MAY be made with "pyang" utility. The figure below presents the tree diagram of VSYM.

```

module: ietf-ucpe
+--rw ietf-ucpe:ucpe* [name]
  +--rw ietf-ucpe:Name          string
  +--rw ietf-ucpe:device*       -> ../name
  +--rw ietf-ucpe:links* [link]
    | +--rw ietf-ucpe:link      string
  +--rw ietf-ucpe:switches* [switch]
    | +--rw ietf-ucpe:switch    string
    | +--rw ietf-ucpe:ports* [port]
    |   +--rw ietf-ucpe:port    uint64

```



```

|     +--rw ietf-ucpe:name?   string
|     +--rw ietf-ucpe:link?   -> ../../../../links/link
+--rw ietf-ucpe:vms* [vm]
|   +--rw ietf-ucpe:vm        string
|   +--rw ietf-ucpe:ports* [port]
|   |   +--rw ietf-ucpe:port   string
|   |   +--rw ietf-ucpe:name?  string
|   |   +--rw ietf-ucpe:link?  -> ../../../../links/link
|   +--rw ietf-ucpe:ram?      string
|   +--rw ietf-ucpe:cpu?      string
|   +--rw ietf-ucpe:storages* [id]
|   |   +--rw ietf-ucpe:id      string
|   |   +--rw ietf-ucpe:location? string
|   +--rw ietf-ucpe:day0-config
|   |   +--rw ietf-ucpe:location? string
|   |   +--rw ietf-ucpe:day0-var-path? string
|   |   +--rw ietf-ucpe:variable* [name]
|   |   |   +--rw ietf-ucpe:name    string
|   |   |   +--rw ietf-ucpe:value?  string
+--rw ucpe-if:interfaces
|   +--rw ucpe-if:interface* [name]
|   |   +--rw ucpe-if:name                string
|   |   +--rw ucpe-if:description?        string
|   |   +--rw ucpe-if:type                 identityref
|   |   +--rw ucpe-if:enabled?             boolean
|   |   +--rw ucpe-if:link-up-down-trap-enable? enumeration {if-mib}?
|   |   +--ro ucpe-if:admin-status          enumeration {if-mib}?
|   |   +--ro ucpe-if:oper-status           enumeration
|   |   +--ro ucpe-if:last-change?          yang:date-and-time
|   |   +--ro ucpe-if:if-index              int32 {if-mib}?
|   |   +--ro ucpe-if:phys-address?         yang:phys-address
|   |   +--ro ucpe-if:higher-layer-if*      interface-ref
|   |   +--ro ucpe-if:lower-layer-if*       interface-ref
|   |   +--ro ucpe-if:speed?                yang:gauge64
|   |   +--ro ucpe-if:statistics
|   |   |   +--ro ucpe-if:discontinuity-time yang:date-and-time
|   |   |   +--ro ucpe-if:in-octets?         yang:counter64
|   |   |   +--ro ucpe-if:in-unicast-pkts?   yang:counter64
|   |   |   +--ro ucpe-if:in-broadcast-pkts? yang:counter64
|   |   |   +--ro ucpe-if:in-multicast-pkts? yang:counter64
|   |   |   +--ro ucpe-if:in-discards?       yang:counter32
|   |   |   +--ro ucpe-if:in-errors?         yang:counter32
|   |   |   +--ro ucpe-if:in-unknown-protos? yang:counter32
|   |   |   +--ro ucpe-if:out-octets?        yang:counter64
|   |   |   +--ro ucpe-if:out-unicast-pkts?  yang:counter64
|   |   |   +--ro ucpe-if:out-broadcast-pkts? yang:counter64
|   |   |   +--ro ucpe-if:out-multicast-pkts? yang:counter64
|   |   |   +--ro ucpe-if:out-discards?      yang:counter32

```



```

|      | +--ro ucpe-if:out-errors?          yang:counter32
|      +--rw ucpe-interface:ports* [port]
|          +--rw ucpe-interface:port      string
|          +--rw ucpe-interface:link?-> ../../../ietf-nfv:links/link
x--ro ucpe-if:interfaces-state
  x--ro ucpe-if:interface* [name]
    x--ro ucpe-if:name                string
    x--ro ucpe-if:type                identityref
    x--ro ucpe-if:admin-status        enumeration {if-mib}?
    x--ro ucpe-if:oper-status         enumeration
    x--ro ucpe-if:last-change?        yang:date-and-time
    x--ro ucpe-if:if-index            int32 {if-mib}?
    x--ro ucpe-if:phys-address?       yang:phys-address
    x--ro ucpe-if:speed?              yang:gauge64
    x--ro ucpe-if:statistics
      x--ro ucpe-if:discontinuity-time yang:date-and-time
      x--ro ucpe-if:in-octets?         yang:counter64
      x--ro ucpe-if:in-unicast-pkts?   yang:counter64
      x--ro ucpe-if:in-broadcast-pkts? yang:counter64
      x--ro ucpe-if:in-multicast-pkts? yang:counter64
      x--ro ucpe-if:in-discards?       yang:counter32
      x--ro ucpe-if:in-errors?         yang:counter32
      x--ro ucpe-if:in-unknown-protos? yang:counter32
      x--ro ucpe-if:out-octets?        yang:counter64
      x--ro ucpe-if:out-unicast-pkts?  yang:counter64
      x--ro ucpe-if:out-broadcast-pkts? yang:counter64
      x--ro ucpe-if:out-multicast-pkts? yang:counter64
      x--ro ucpe-if:out-discards?      yang:counter32
      x--ro ucpe-if:out-errors?        yang:counter32

```

6. Specification of the VNF YANG Service Model

This section presents the specification of the VYSM. There are 3 yang modules, where:

- o yang module "ietf-vysm-interfaces" augments module "ietf-interfaces"
- o yang module "ietf-interfaces" augments module "ietf-vysm-service".


```
1. module ietf-ucpe{
    ...
    list ucpe {
        ...
    }
}

2. module ietf-ucpe-interfaces {
    import ietf-ucpe { prefix ietf-vysm; }
    ...
    augment "/ietf-vysm:ucpe"{
        container interfaces {
            description "Interface parameters.";
            list interface {
                ...
            }
        }
    }
}

3. module ietf-ucpe-interfaces-ports {
    import ietf-ucpe { prefix ietf-nfv; }
    import ietf-ucpe-interfaces { prefix ietf-if; }
    ...
    augment "/ietf-nfv:ucpe/ietf-if:interfaces/ietf-if:interface" {
    }
}
}
```

[6.1.](#) uCPE main yang module

```
<CODE BEGINS> file "ietf-ucpe@2019-10-19.yang"
module ietf-ucpe {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe";
    prefix ietf-ucpe;

    organization "SFR";
    contact
        "Dmytro Shytyi
        EMail:ietf.dmytro@shytyi.net";
    description
        "This is a Network Function Virtualization (NFV) YANG
        service model.
```

Copyright (c) 2019 IETF Trust and the persons identified as
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself for full legal notices."

```
revision 2019-10-19 {
  description
    "Yang model was cleaned.Ietf interfaces added";
  reference "draft-shytyi-opsawg-vysm-04";
}
revision 2019-09-16 {
  description
    "Added 0day config for VNFs.
    Yang model modified according
    to the received comments.";
  reference "draft-shytyi-opsawg-vysm-00";
}
revision 2018-01-07 {
  description
    "Initial revision.";
  reference "draft-shytyi-netmod-vysm-01";
}

list ucpe {
  key "name";
  leaf name {
    type string;
    description
      "Device identifier";
  }
  description
    "Name of the instance of the service";
  list links {
    key "link";
    leaf link {
      type string {
        pattern "[a-zA-Z0-9]*";
      }
      description
        "Name of the virtual link from the pool of
        the links";
    }
  }
}
```



```
    description
      "Pool of the virtual links that connect
      VMS and Interfaces";
  }
  list switches {
    key "switch";
    leaf switch {
      type string;
      description
        "Name of the forwarding domain";
    }
    list ports {
      key "port";
      leaf port {
        type uint64 {
          range "10..15";
        }
        description
          "Name of the connector";
      }
      leaf name {
        type string;
        description
          "Name of the subconnector";
      }
      leaf link {
        type leafref {
          path "../..../links/link";
        }
        description
          "Link that is connected to the switch
          via port";
      }
      description
        "Set of the connectors the forwarding
        domain has";
    }
    description
      "Set of the forwarding domains";
  }
  list vms {
    key "vm";
    leaf vm {
      type string {
        pattern "[a-zA-Z]*";
      }
      description
        "Name of the Virtual Machine";
    }
  }
```



```
}
list ports {
  key "port";
  leaf port {
    type string;
    description
      "Name of the connector";
  }
  leaf name {
    type string;
    description
      "Name of the subconnector";
  }
  leaf link {
    type leafref {
      path "../../links/link";
    }
    description
      "Link that connects the VM with a
      switch or Interface via connector";
  }
  description
    "Set of Virtual Machine connectors";
}
leaf ram {
  type uint64;
  description
    "Size of RAM to allocate for the
    Guest OS";
}
leaf cpu {
  type uint64;
  description
    "Number of vCPUs to allocate for the
    Guest OS";
}
list storages {
  key "id";
  leaf id {
    type string;
    description
      "Name of the Storage";
  }
  leaf location {
    type string;
    description
      "External location where the image is
      saved.";
  }
}
```



```
    }
    description
      "Virtual storge of the image for the
       Virtual Machine";
  }
  container day0-config {
    leaf location {
      type string;
      description
        "0day configuration location";
    }
    leaf day0-var-path {
      type string;
      description
        "path of the file that contains the
         0day variables";
    }
    list variable {
      key "name";
      leaf name {
        type string;
        description
          "variable name";
      }
      leaf value {
        type string;
        description
          "variable value";
      }
      description
        "name:value";
    }
  }
  description
    "0day configuration:init config";
}
description
  "Set of the Virtual Machines configured on
   the universal Customer-Premises Equipment";
}
}
```

<CODE ENDS>

6.2. uCPE ports yang module for each interface

```
<CODE BEGINS> file "ietf-ucpe-interfaces-ports@2019-10-19.yang"
module ietf-ucpe-interfaces-ports {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-interfaces-ports";
  prefix ucpe-interface;

  import ietf-ucpe {
    prefix ietf-nfv;
  }
  import ietf-ucpe-interfaces {
    prefix ietf-if;
  }

  organization "SFR";
  contact
    "Dmytro Shytyi
     EMail:ietf.dmytro@shytyi.net";
  description
    "This is a Network Function Virtualization (NFV) YANG
     model for interfaces.

     Copyright (c) 2019 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
     (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
     for full legal notices.";

  revision 2019-10-19 {
    description
      "Yang model is cleaned";
    reference "draft-shytyi-opsawg-vysm-04";
  }
  revision 2019-10-03 {
    description
      "Initial revision.";
    reference "draft-shytyi-opsawg-vysm-01";
  }
}
```



```
augment "/ietf-nfv:ucpe/ietf-if:interfaces/ietf-if:interface" {
  list ports {
    key "port";
    leaf port {
      type string;
      description
        "Name of the connector";
    }
    leaf link {
      type leafref {
        path "../../../../../ietf-nfv:links/ietf-nfv:link";
      }
      description
        "Link that is connected to the port
        via connector";
    }
    description
      "Set of the connectors the physical
      interface has";
  }
  description
    "ucpe ports of the interface";
}
```

<CODE ENDS>

6.3. uCPE interfaces yang module

```
<CODE BEGINS> file "ietf-ucpe-interfaces@2019-10-19.yang"
module ietf-ucpe-interfaces {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-interfaces";
  prefix ucpe-if;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-ucpe {
    prefix ietf-vysm;
  }

  organization "IETF NETMOD (Network Modeling) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/netmod/>
```


WG List: <mailto:netmod@ietf.org>

Editor: Martin Bjorklund
<mailto:mbj@tail-f.com>;

description

"This module contains a collection of YANG definitions for managing network interfaces.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of [RFC 8343](#); see the RFC itself for full legal notices.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in [BCP 14](#) ([RFC 2119](#)) ([RFC 8174](#)) when, and only when, they appear in all capitals, as shown here.";

```
revision 2019-10-19 {
  description
    "Yang model updated according to the ucpe needs";
  reference "RFC 8343: A YANG Data Model for Interface Management";
}
revision 2018-02-20 {
  description
    "Updated to support NMDA.";
  reference "RFC 8343: A YANG Data Model for Interface Management";
}
revision 2014-05-08 {
  description
    "Initial revision.";
  reference "RFC 7223: A YANG Data Model for Interface Management";
}
/*
 * Typedefs
 */

typedef interface-ref {
  type leafref {
```



```
    path "/ietf-vysm:ucpe/"+
        "ucpe-if:interfaces/ucpe-if:interface/ucpe-if:name";
  }
  description
    "This type is used by data models that need to reference
    interfaces.";
}

/*
 * Identities
 */

identity interface-type {
  description
    "Base identity from which specific interface types are
    derived.";
}

/*
 * Features
 */

feature arbitrary-names {
  description
    "This feature indicates that the device allows user-controlled
    interfaces to be named arbitrarily.";
}

feature pre-provisioning {
  description
    "This feature indicates that the device supports
    pre-provisioning of interface configuration, i.e., it is
    possible to configure an interface whose physical interface
    hardware is not present on the device.";
}

feature if-mib {
  description
    "This feature indicates that the device implements
    the IF-MIB.";
  reference "RFC 2863: The Interfaces Group MIB";
}

/*
 * Data nodes
 */

augment "/ietf-vysm:ucpe" {
```



```
container interfaces {  
  description  
    "Interface parameters.";  
  list interface {  
    key "name";  
    description  
      "The list of interfaces on the device.
```

The status of an interface is available in this list in the operational state. If the configuration of a system-controlled interface cannot be used by the system (e.g., the interface hardware present does not match the interface type), then the configuration is not applied to the system-controlled interface shown in the operational state. If the configuration of a user-controlled interface cannot be used by the system, the configured interface is not instantiated in the operational state.

System-controlled interfaces created by the system are always present in this list in the operational state, whether or not they are configured.";

```
leaf name {  
  type string;  
  description  
    "The name of the interface.
```

A device MAY restrict the allowed values for this leaf, possibly depending on the type of the interface. For system-controlled interfaces, this leaf is the device-specific name of the interface.

If a client tries to create configuration for a system-controlled interface that is not present in the operational state, the server MAY reject the request if the implementation does not support pre-provisioning of interfaces or if the name refers to an interface that can never exist in the system. A Network Configuration Protocol (NETCONF) server MUST reply with an rpc-error with the error-tag 'invalid-value' in this case.

If the device supports pre-provisioning of interface configuration, the 'pre-provisioning' feature is advertised.

If the device allows arbitrarily named user-controlled interfaces, the 'arbitrary-names' feature is advertised.

When a configured user-controlled interface is created by the system, it is instantiated with the same name in the operational state.

A server implementation MAY map this leaf to the ifName MIB object. Such an implementation needs to use some mechanism to handle the differences in size and characters

allowed between this leaf and ifName. The definition of such a mechanism is outside the scope of this document.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifName";

}

leaf description {

type string;

description

"A textual description of the interface.

A server implementation MAY map this leaf to the ifAlias MIB object. Such an implementation needs to use some mechanism to handle the differences in size and characters

allowed between this leaf and ifAlias. The definition of such a mechanism is outside the scope of this document.

Since ifAlias is defined to be stored in non-volatile storage, the MIB implementation MUST map ifAlias to the value of 'description' in the persistently stored configuration.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifAlias";

}

leaf type {

type identityref {

base interface-type;

}

mandatory true;

description

"The type of the interface.

When an interface entry is created, a server MAY initialize the type leaf with a valid value, e.g., if it is possible to derive the type from the name of the interface.

If a client tries to set the type of an interface to a value that can never be used by the system, e.g., if the type is not supported or if the type does not match the name of the interface, the server MUST reject the request. A NETCONF server MUST reply with an rpc-error with the


```
        error-tag 'invalid-value' in this case.";
        reference "RFC 2863: The Interfaces Group MIB - ifType";
    }
    leaf enabled {
        type boolean;
        default "true";
        description
            "This leaf contains the configured, desired state of the
            interface.

            Systems that implement the IF-MIB use the value of this
            leaf in the intended configuration to set
            IF-MIB.ifAdminStatus to 'up' or 'down' after an ifEntry
            has been initialized, as described in RFC 2863.

            Changes in this leaf in the intended configuration are
            reflected in ifAdminStatus.";
        reference "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
    }
    leaf link-up-down-trap-enable {
        if-feature if-mib;
        type enumeration {
            enum "enabled" {
                value 1;
                description
                    "The device will generate linkUp/linkDown SNMP
                    notifications for this interface.";
            }
            enum "disabled" {
                value 2;
                description
                    "The device will not generate linkUp/linkDown SNMP
                    notifications for this interface.";
            }
        }
    }
    description
        "Controls whether linkUp/linkDown SNMP notifications
        should be generated for this interface.
```

If this node is not configured, the value 'enabled' is operationally used by the server for interfaces that do not operate on top of any other interface (i.e., there are no 'lower-layer-if' entries), and 'disabled' otherwise.";

```
reference
```



```
    "RFC 2863: The Interfaces Group MIB -
      ifLinkUpDownTrapEnable";
  }
  leaf admin-status {
    if-feature if-mib;
    type enumeration {
      enum "up" {
        value 1;
        description
          "Ready to pass packets.";
      }
      enum "down" {
        value 2;
        description
          "Not ready to pass packets and not in some test mode.";
      }
      enum "testing" {
        value 3;
        description
          "In some test mode.";
      }
    }
  }
  config false;
  mandatory true;
  description
    "The desired state of the interface.

    This leaf has the same read semantics as ifAdminStatus.";
  reference "RFC 2863: The Interfaces Group MIB ifAdminStatus";
}
leaf oper-status {
  type enumeration {
    enum "up" {
      value 1;
      description
        "Ready to pass packets.";
    }
    enum "down" {
      value 2;
      description
        "The interface does not pass any packets.";
    }
    enum "testing" {
      value 3;
      description
        "In some test mode.  No operational packets can
        be passed.";
    }
  }
}
```



```
    enum "unknown" {
      value 4;
      description
        "Status cannot be determined for some reason.";
    }
    enum "dormant" {
      value 5;
      description
        "Waiting for some external event.";
    }
    enum "not-present" {
      value 6;
      description
        "Some component (typically hardware) is missing.";
    }
    enum "lower-layer-down" {
      value 7;
      description
        "Down due to state of lower-layer interface(s).";
    }
  }
  config false;
  mandatory true;
  description
    "The current operational state of the interface.

    This leaf has the same semantics as ifOperStatus.";
  reference "RFC 2863: The Interfaces Group MIB - ifOperStatus";
}
leaf last-change {
  type yang:date-and-time;
  config false;
  description
    "The time the interface entered its current operational
    state. If the current state was entered prior to the
    last re-initialization of the local network management
    subsystem, then this node is not present.";
  reference "RFC 2863: The Interfaces Group MIB - ifLastChange";
}
leaf if-index {
  if-feature if-mib;
  type int32 {
    range "1..2147483647";
  }
  config false;
  mandatory true;
  description
    "The ifIndex value for the ifEntry represented by this
```



```
        interface.";
        reference "RFC 2863: The Interfaces Group MIB - ifIndex";
    }
    leaf phys-address {
        type yang:phys-address;
        config false;
        description
            "The interface's address at its protocol sub-layer. For
            example, for an 802.x interface, this object normally
            contains a Media Access Control (MAC) address. The
            interface's media-specific modules must define the bit
            and byte ordering and the format of the value of this
            object. For interfaces that do not have such an address
            (e.g., a serial line), this node is not present.";
        reference "RFC 2863: The Interfaces Group MIB ifPhysAddress";
    }
    leaf-list higher-layer-if {
        type interface-ref;
        config false;
        description
            "A list of references to interfaces layered on top of this
            interface.";
        reference "RFC 2863: The Interfaces Group MIB
                    - ifStackTable";
    }
    leaf-list lower-layer-if {
        type interface-ref;
        config false;
        description
            "A list of references to interfaces layered
            underneath this interface.";
        reference "RFC 2863: The Interfaces Group MIB - ifStackTable";
    }
    leaf speed {
        type yang:gauge64;
        units "bits/second";
        config false;
        description
            "An estimate of the interface's current bandwidth in bits
            per second. For interfaces that do not vary in
            bandwidth or for those where no accurate estimation can
            be made, this node should contain the nominal bandwidth.
            For interfaces that have no concept of bandwidth, this
            node is not present.";
        reference
            "RFC 2863: The Interfaces Group MIB -
                ifSpeed, ifHighSpeed";
    }
}
```



```
container statistics {
  config false;
  description
    "A collection of interface-related statistics objects.";
  leaf discontinuity-time {
    type yang:date-and-time;
    mandatory true;
    description
      "The time on the most recent occasion at which any one
      or more of this interface's counters suffered a
      discontinuity. If no such discontinuities have occurred
      since the last re-initialization of the local management
      subsystem, then this node contains the time the local
      management subsystem re-initialized itself.";
  }
  leaf in-octets {
    type yang:counter64;
    description
      "The total number of octets received on the interface,
      including framing characters.

      Discontinuities in the value of this counter can occur
      at re-initialization of the management system and at
      other times as indicated by the value of
      'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB ifHCInOctets";
  }
  leaf in-unicast-pkts {
    type yang:counter64;
    description
      "The number of packets, delivered by this sub-layer to a
      higher (sub-)layer, that were not addressed to a
      multicast or broadcast address at this sub-layer.

      Discontinuities in the value of this counter can occur
      at re-initialization of the management system and at
      other times as indicated by the value of
      'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB -
      ifHCInUcastPkts";
  }
  leaf in-broadcast-pkts {
    type yang:counter64;
    description
      "The number of packets, delivered by this sub-layer to a
      higher (sub-)layer, that were addressed to a broadcast
      address at this sub-layer.
```



```
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
        "RFC 2863: The Interfaces Group MIB -
        ifHCInBroadcastPkts";
}
leaf in-multicast-pkts {
    type yang:counter64;
    description
        "The number of packets, delivered by this sub-layer to a
        higher (sub-)layer, that were addressed to a multicast
        address at this sub-layer.  For a MAC-layer protocol,
        this includes both Group and Functional addresses.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
        "RFC 2863: The Interfaces Group MIB -
        ifHCInMulticastPkts";
}
leaf in-discards {
    type yang:counter32;
    description
        "The number of inbound packets that were chosen to be
        discarded even though no errors had been detected to
        prevent their being deliverable to a higher-layer
        protocol.  One possible reason for discarding such a
        packet could be to free up buffer space.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB -
        ifInDiscards";
}
leaf in-errors {
    type yang:counter32;
    description
        "For packet-oriented interfaces, the number of inbound
        packets that contained errors preventing them from being
        deliverable to a higher-layer protocol.  For character-
        oriented or fixed-length interfaces, the number of
```


inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifInErrors";

}

leaf in-unknown-protos {

type yang:counter32;

description

"For packet-oriented interfaces, the number of packets received via the interface that were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing, the number of transmission units received via the interface that were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter is not present.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of

'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifInUnknownProtos";

}

leaf out-octets {

type yang:counter64;

description

"The total number of octets transmitted out of the interface, including framing characters.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of

'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifHCOutOctets";

}

leaf out-unicast-pkts {

type yang:counter64;

description

"The total number of packets that higher-level protocols

requested be transmitted and that were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.
reference "[RFC 2863](#): The Interfaces Group MIB - ifHCOutUcastPkts";

```
}
leaf out-broadcast-pkts {
  type yang:counter64;
  description
    "The total number of packets that higher-level protocols
    requested be transmitted and that were addressed to a
    broadcast address at this sub-layer, including those
    that were discarded or not sent.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCOutBroadcastPkts";
}
leaf out-multicast-pkts {
  type yang:counter64;
  description
    "The total number of packets that higher-level protocols
    requested be transmitted and that were addressed to a
    multicast address at this sub-layer, including those
    that were discarded or not sent. For a MAC-layer
    protocol, this includes both Group and Functional
    addresses.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCOutMulticastPkts";
}
leaf out-discards {
  type yang:counter32;
  description
```


"The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of

'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifOutDiscards";

}

leaf out-errors {

type yang:counter32;

description

"For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of

'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifOutErrors";

}

}

}

}

/*

* Legacy typedefs

*/

// typedef interface-state-ref {

// type leafref {

// path "/ietf-vysm:ucpe"+

// "ucpe-if:interfaces-state/ucpe-if:interface/ucpe-if:name";

// }

// status deprecated;

// description

// "This type is used by data models that need to reference
// the operationally present interfaces.";

// }

/*

* Legacy operational state data nodes


```
*/
container interfaces-state {
  config false;
  status deprecated;
  description
    "Data nodes for the operational state of interfaces.";
  list interface {
    key "name";
    status deprecated;
    description
      "The list of interfaces on the device.

      System-controlled interfaces created by the system are
      always present in this list, whether or not they are
      configured.";
    leaf name {
      type string;
      status deprecated;
      description
        "The name of the interface.

        A server implementation MAY map this leaf to the ifName
        MIB object. Such an implementation needs to use some
        mechanism to handle the differences in size and characters
        allowed between this leaf and ifName. The definition of
        such a mechanism is outside the scope of this document.";
      reference "RFC 2863: The Interfaces Group MIB - ifName";
    }
    leaf type {
      type identityref {
        base interface-type;
      }
      mandatory true;
      status deprecated;
      description
        "The type of the interface.";
      reference "RFC 2863: The Interfaces Group MIB - ifType";
    }
    leaf admin-status {
      if-feature if-mib;
      type enumeration {
        enum "up" {
          value 1;
          description
            "Ready to pass packets.";
        }
        enum "down" {
          value 2;
```



```
        description
            "Not ready to pass packets and not in some test mode.";
    }
    enum "testing" {
        value 3;
        description
            "In some test mode.";
    }
}
mandatory true;
status deprecated;
description
    "The desired state of the interface.

    This leaf has the same read semantics as ifAdminStatus.";
    reference "RFC 2863: The Interfaces Group MIB -
        ifAdminStatus";
}
leaf oper-status {
    type enumeration {
        enum "up" {
            value 1;
            description
                "Ready to pass packets.";
        }
        enum "down" {
            value 2;
            description
                "The interface does not pass any packets.";
        }
        enum "testing" {
            value 3;
            description
                "In some test mode.  No operational packets can
                be passed.";
        }
        enum "unknown" {
            value 4;
            description
                "Status cannot be determined for some reason.";
        }
        enum "dormant" {
            value 5;
            description
                "Waiting for some external event.";
        }
        enum "not-present" {
            value 6;
```



```
        description
            "Some component (typically hardware) is missing.";
    }
    enum "lower-layer-down" {
        value 7;
        description
            "Down due to state of lower-layer interface(s).";
    }
}
mandatory true;
status deprecated;
description
    "The current operational state of the interface.

    This leaf has the same semantics as ifOperStatus.";
    reference "RFC 2863: The Interfaces Group MIB -
        ifOperStatus";
}
leaf last-change {
    type yang:date-and-time;
    status deprecated;
    description
        "The time the interface entered its current operational
        state.  If the current state was entered prior to the
        last re-initialization of the local network management
        subsystem, then this node is not present.";
        reference "RFC 2863: The Interfaces Group MIB -
            ifLastChange";
}
leaf if-index {
    if-feature if-mib;
    type int32 {
        range "1..2147483647";
    }
    mandatory true;
    status deprecated;
    description
        "The ifIndex value for the ifEntry represented by this
        interface.";
        reference "RFC 2863: The Interfaces Group MIB - ifIndex";
}
leaf phys-address {
    type yang:phys-address;
    status deprecated;
    description
        "The interface's address at its protocol sub-layer.  For
        example, for an 802.x interface, this object normally
        contains a Media Access Control (MAC) address.  The
```



```
interface's media-specific modules must define the bit
and byte ordering and the format of the value of this
object. For interfaces that do not have such an address
(e.g., a serial line), this node is not present.";
reference "RFC 2863: The Interfaces Group MIB -
ifPhysAddress";
}
//      leaf-list higher-layer-if {
//          type interface-state-ref;
//          status deprecated;
//          description
//              "A list of references to interfaces layered on
//              top of this
//              interface.";
//          reference
//              "RFC 2863: The Interfaces Group MIB -
//              ifStackTable";
//      }
//
//      leaf-list lower-layer-if {
//          type interface-state-ref;
//          status deprecated;
//          description
//              "A list of references to interfaces
//              layered underneath this
//              interface.";
//          reference
//              "RFC 2863: The Interfaces Group MIB -
//              ifStackTable";
//      }
leaf speed {
    type yang:gauge64;
    units "bits/second";
    status deprecated;
    description
        "An estimate of the interface's current bandwidth in bits
        per second. For interfaces that do not vary in
        bandwidth or for those where no accurate estimation can
        be made, this node should contain the nominal bandwidth.
        For interfaces that have no concept of bandwidth, this
        node is not present.";
    reference
        "RFC 2863: The Interfaces Group MIB -
        ifSpeed, ifHighSpeed";
}
container statistics {
    status deprecated;
```



```
description
  "A collection of interface-related statistics objects.";
leaf discontinuity-time {
  type yang:date-and-time;
  mandatory true;
  status deprecated;
  description
    "The time on the most recent occasion at which any one or
    more of this interface's counters suffered a
    discontinuity. If no such discontinuities have occurred
    since the last re-initialization of the local management
    subsystem, then this node contains the time the local
    management subsystem re-initialized itself.";
}
leaf in-octets {
  type yang:counter64;
  status deprecated;
  description
    "The total number of octets received on the interface,
    including framing characters.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB -
    ifHCInOctets";
}
leaf in-unicast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "The number of packets, delivered by this sub-layer to a
    higher (sub-)layer, that were not addressed to a
    multicast or broadcast address at this sub-layer.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB -
    ifHCInUcastPkts";
}
leaf in-broadcast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "The number of packets, delivered by this sub-layer to a
    higher (sub-)layer, that were addressed to a broadcast
```


address at this sub-layer.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 2863](#): The Interfaces Group MIB - ifHCInBroadcastPkts";

}

leaf in-multicast-pkts {

type yang:counter64;

status deprecated;

description

"The number of packets, delivered by this sub-layer to a higher (sub-)layer, that were addressed to a multicast address at this sub-layer. For a MAC-layer protocol, this includes both Group and Functional addresses.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 2863](#): The Interfaces Group MIB - ifHCInMulticastPkts";

}

leaf in-discards {

type yang:counter32;

status deprecated;

description

"The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifInDiscards";

}

leaf in-errors {

type yang:counter32;

status deprecated;

description

"For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifInErrors";

}

leaf in-unknown-protos {

type yang:counter32;

status deprecated;

description

"For packet-oriented interfaces, the number of packets received via the interface that were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing, the number of transmission units received via the interface that were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter is not present.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifInUnknownProtos";

}

leaf out-octets {

type yang:counter64;

status deprecated;

description

"The total number of octets transmitted out of the interface, including framing characters.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of 'discontinuity-time'.";

reference "[RFC 2863](#): The Interfaces Group MIB - ifHCOutOctets";


```
}
leaf out-unicast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "The total number of packets that higher-level protocols
    requested be transmitted and that were not addressed
    to a multicast or broadcast address at this sub-layer,
    including those that were discarded or not sent.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
    reference "RFC 2863: The Interfaces Group MIB -
               ifHCOutUcastPkts";
}
leaf out-broadcast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "The total number of packets that higher-level protocols
    requested be transmitted and that were addressed to a
    broadcast address at this sub-layer, including those
    that were discarded or not sent.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
    'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB -
       ifHCOutBroadcastPkts";
}
leaf out-multicast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "The total number of packets that higher-level protocols
    requested be transmitted and that were addressed to a
    multicast address at this sub-layer, including those
    that were discarded or not sent. For a MAC-layer
    protocol, this includes both Group and Functional
    addresses.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system and at
    other times as indicated by the value of
```



```
        'discontinuity-time'.";
    reference
        "RFC 2863: The Interfaces Group MIB -
         ifHCOutMulticastPkts";
}
leaf out-discards {
    type yang:counter32;
    status deprecated;
    description
        "The number of outbound packets that were chosen to be
        discarded even though no errors had been detected to
        prevent their being transmitted. One possible reason
        for discarding such a packet could be to free up buffer
        space.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system and at
        other times as indicated by the value of
        'discontinuity-time'.";
        reference "RFC 2863: The Interfaces Group MIB -
         ifOutDiscards";
}
leaf out-errors {
    type yang:counter32;
    status deprecated;
    description
        "For packet-oriented interfaces, the number of outbound
        packets that could not be transmitted because of errors.
        For character-oriented or fixed-length interfaces, the
        number of outbound transmission units that could not be
        transmitted because of errors.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system and at
        other times as indicated by the value of
        'discontinuity-time'.";
        reference "RFC 2863: The Interfaces Group MIB -
         ifOutErrors";
}
}
}
}
description
    "ucpe physical interfaces";
}
```



```
}
    <CODE ENDS>
```

7. XML example

The XML example below presents the configuration of the next service in the uCPE, where: vSW(LAN), vSW(WAN), vSW(Service) - virtual switches; l1,l2,l3,l4 - virtual links; VMs represent PNFs (Physical Network Functions) that could be bootstrapped with 0day config/license.

```
+-----+      +-----+      +-----+
|vSW(LAN)|--l2--|VNF-vFirewall|--l3--|      |
+-----+      +-----+      |          |
+-----+      +-----+      |vSW(Service)|
|vSW(WAN)|--l1--|  VNF_vCPE  |--l4--|      |
+-----+      +-----+      +-----+
```

```
<ucpe xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe">
  <name>ucpe1</name>
  <links>
    <link>l1</link>
  </links>
  <links>
    <link>l2</link>
  </links>
  <links>
    <link>l3</link>
  </links>
  <links>
    <link>l4</link>
  </links>
  <switches>
    <switch>lan</switch>
    <ports>
      <port>10</port>
      <name>l2p10</name>
      <link>l2</link>
    </ports>
  </switches>
  <switches>
    <switch>service</switch>
    <ports>
      <port>10</port>
```



```
    <name>l3p10</name>
    <link>l3</link>
  </ports>
  <ports>
    <port>11</port>
    <name>l4p10</name>
    <link>l4</link>
  </ports>
</switches>
<switches>
  <switch>wan</switch>
  <ports>
    <port>10</port>
    <link>l1</link>
  </ports>
</switches>
<vms>
  <vm>VNF-vCPE</vm>
  <ports>
    <port>1</port>
    <name>l1p1</name>
    <link>l1</link>
  </ports>
  <ports>
    <port>2</port>
    <name>l4p2</name>
    <link>l4</link>
  </ports>
  <ram>2048</ram>
  <cpu>2</cpu>
  <storages>
    <id>1</id>
    <location>http://192.168.2.1/vCPE-x86.qcow2</location>
  </storages>
  <day0-config>
    <location>https://192.168.2.1/vCPE-day0.iso</location>
    <day0-var-path>/config.rom</day0-var-path>
    <variable>
      <name>hostname</name>
      <value>IETF-vCPE</value>
    </variable>
    <variable>
      <name>ipaddress</name>
      <value>192.168.1.2 255.255.255.0</value>
    </variable>
  </day0-config>
</vms>
<vms>
```



```
<vm>VNF-vFirewall</vm>
<ports>
  <port>1</port>
  <name>l3p1</name>
  <link>l3</link>
</ports>
<ports>
  <port>2</port>
  <name>l2p2</name>
  <link>l2</link>
</ports>
<ram>2048</ram>
<cpu>2</cpu>
<storages>
  <id>1</id>
  <location>http://192.168.2.1/vFirewall-x86.qcow2</location>
</storages>
<day0-config>
  <location>https://192.168.2.1/vFirewall-day0.iso</location>
  <day0-var-path>/config.rom</day0-var-path>
  <variable>
    <name>hostname</name>
    <value>vFirewall</value>
  </variable>
  <variable>
    <name>ipaddress</name>
    <value>192.168.1.3 255.255.255.0</value>
  </variable>
</day0-config>
</vms>
</ucpe>
```

8. Security Considerations

At this time, no security considerations are addressed by this memo.

9. IANA Considerations

No request to IANA at this time.

10. Acknowledgements

The authors would like to thank:

- o Mahesh Jethanandani.
- o Robert Varga.

- o Bill Wu.
- o Joe Clarke.
- o Tom Petch.

for their valuable comments.

11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", [RFC 8199](#), DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.

Authors' Addresses

Dmytro Shytyi
SFR
Paris , Ile-de-France
France

Email: ietf.dmytro@shytyi.net
URI: <https://dmytro.shytyi.net>

Laurent Beylier
SFR
Paris , Ile-de-France
France

Email: laurent.beylier@sfr.com

Luigi Iannone
Telecom ParisTech
Paris , Ile-de-France
France

Email: luigi.iannone@telecom-paristech.fr

