

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 22, 2021

D. Shytyi
L. Beylier
SFR
L. Iannone
Telecom ParisTech
November 18, 2020

A YANG Module for uCPE management.
draft-shytyi-opsawg-vysm-09

Abstract

This document provides a YANG data model for uCPE management (VYSM) and definition of the uCPE equipment. The YANG Model serves as a base framework for managing an universal Customer-Premises Equipment (uCPE) subsystem. The model can be used by a Network 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 May 22, 2021.

Copyright Notice

Copyright (c) 2020 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 Model for uCPE management	6
5. Components for uCPE Management	7
6. Diagram overview of YANG Data Model tree for uCPE management	9
7. Logical Network Elements extension YANG Model	11
8. Security Considerations	20
9. IANA Considerations	20
10. Acknowledgements	20
11. Normative References	21
Appendix A. Example of the uCPE resources management	21
Appendix B. Example of the uCPE resources management (deprecated)	25
Appendix C. Deprecated VNF YANG Model	26
Appendix D. XML example of deprecated YANG model	32
Authors' Addresses	34

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 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 Yang Module.

VYSM - VNF YANG 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

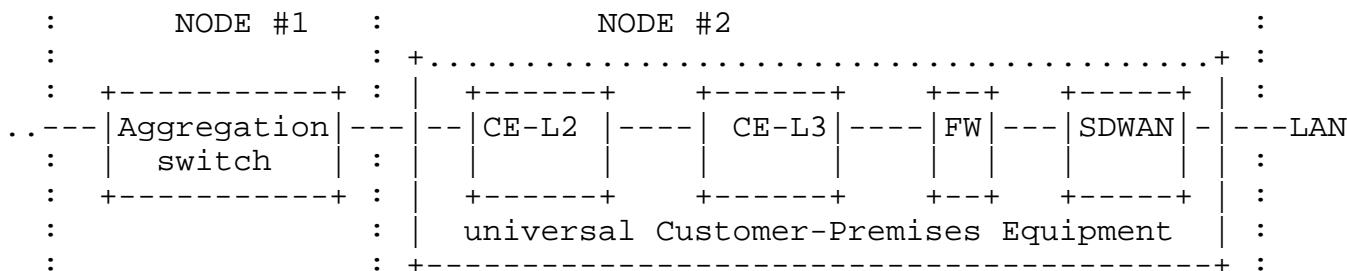
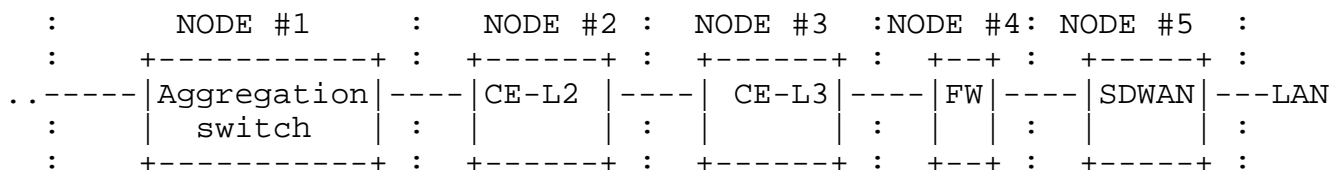
The next elements can be managed in the uCPE:

- 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:



- 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
- 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-l2-vSW(LAN).
- o vSW(WAN)-l1-vRouter-l2-vSW(Service)-l3-vFirewall-l4-vSW(LAN).
- o vSW(WAN)-l1-vRouter-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 Model for uCPE management

Secondly, this document defines and classifies the YANG Model for uCPE Management. This Module is modeled representation of the specific network requirements. It provides abstraction of network configuration and operations. The YANG Model for uCPE Management does not describe all configuration to be performed on the devices, but provides the configuration that is required for the "Network to Network Element(s)" decomposition process RFC 8199 [RFC8199]. Example of the decomposition is presented in the figure below.

The Network 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 YANG Module | <= scope of this document |
: +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
: #
: #####
: # # #
: '-----' '-----' '-----'
: 'Module1' ' Module 2' ' Module3' :
: '-----' '-----' '-----' :
: # # #
: # # #####
: #### ##### #
: # # #
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
# # #
Network # element 1 Network # element 2 Network # element3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| domains domain VNF1| |tenants tenant name VNF1| |nf nf-name VNF1|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

5. Components for uCPE Management

This section provides a components overview to manage the uCPE.

There are multiple RFCs and drafts produced by the IETF community, that are referenced in the YANG tree to manage the uCPE. Each document produced by the IETF covers a part of uCPE Management. The list of the documents is provided below:

- o [RFC8530] - logical network elements (VNFs) properties.
- o [RFC8345] - definition of networks, nodes, node-termination-points: network includes the uCPE with uCPE's physical termination points.
- o [I-D.ietf-teas-sf-aware-topo-model]physical ports and service functions (VNFs) interconnection matrixes (PhyPort-VNF, VNF-VNF).

- o This document itself provides yang modules that completes the existing documents produced by IETF.

This document introduces yang modules for 'logical network elements properties(VNFs)" part:

- o day0-info: mapping between variables inside of the bootstrap config and required values in the list "day0-info". In the bootstrap config the variable could be putted instead value. The value could be set in the day0-info part (check the YANG model) and after the value in the list will be mapped to the variable in the bootstrap config.
- o vCPU/vRAM/vDisk/VNF-ports leafs and lists.

The minimal list of yang models required for compilation of the YANG tree to manage the uCPE is presented below:

- o ieee-dot1Q-types
- o ietf-interfaces
- o ietf-ip
- o ietf-logical-network-element
- o ietf-network
- o ietf-network-instance
- o ietf-ietf-network-topology
- o ietf-routing-types
- o ietf-te-topology
- o ietf-te-topology-sf
- o ietf-te-types
- o ietf-yang-schema-mount
- o The YANG modules introduced in this document:
- o
 - * ietf-ucpe-lne-properties

- * ietf-ucpe-lt-virtual-link-id
- * ietf-ucpe-ni-properties
- * ietf-ucpe-node-type

6. Diagram overview of YANG Data Model tree for uCPE management

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

```

module: ietf-network
  +--rw networks
    +--rw network* [network-id]
      +--rw network-id          network-id
      +--rw network-types
        | +--rw tet:te-topology!
        |   +--rw tet:sf:sf!
      +--rw supporting-network* [network-ref]
        | +--rw network-ref      -> /networks/network/network-id
      +--rw node* [node-id]
        +--rw node-id            node-id
        +--rw supporting-node* [network-ref node-ref]
          | +--rw network-ref      ->
          |   | ..../../supporting-network/network-ref
          | +--rw node-ref         -> /networks/network/node/node-id
      +--rw nt:termination-point* [tp-id]
        +--rw nt:tp-id            tp-id
        +--rw nt:supporting-termination-point*
          | [network-ref node-ref tp-ref]
          +--rw nt:network-ref
            | -> ..../../nw:supporting-node/network-ref
          +--rw nt:node-ref
            | -> ..../../nw:supporting-node/node-ref
          +--rw nt:tp-ref
            | -> /nw:networks/network[nw:network-id=
            |   current()/../network-ref]/node
            |   [nw:node-id=current()/../node-ref]/
            |   termination-point/tp-id
      +--rw tet:te-node-id?        te-types:te-node-id
      +--rw tet:te!
      +--rw tet:te-node-template*
        | -> ..../../te/templates/
        |   node-template/name {template}?
      +--rw tet:te-node-attributes
        |
        |   ...
  
```

```

+--rw tet-sf:service-function
  +--rw tet-sf:connectivity-matrices
    |   +--rw tet-sf:connectivity-matrix* [id]
    |   |   +--rw tet-sf:id                uint32
    |   |   +--rw tet-sf:from
    |   |   |   +--rw tet-sf:service-function-id?  string
    |   |   |   +--rw tet-sf:sf-connection-point-id? string
    |   |   +--rw tet-sf:to
    |   |   |   +--rw tet-sf:service-function-id?  string
    |   |   |   +--rw tet-sf:sf-connection-point-id? string
    |   |   +--rw tet-sf:enabled?                boolean
    |   |   +--rw tet-sf:direction? connectivity-direction
    |   |   +--rw tet-sf:virtual-link-id?        string
    +--rw tet-sf:link-terminations
      +--rw tet-sf:link-termination* [id]
      |   +--rw tet-sf:id                uint32
      |   +--rw tet-sf:from
      |   |   +--rw tet-sf:tp-ref?    -> ../../../../
      |   |   |   ../../../../nt:termination-point/tp-id
      |   +--rw tet-sf:to
      |   |   +--rw tet-sf:service-function-id?  string
      |   |   +--rw tet-sf:sf-connection-point-id? string
      |   +--rw tet-sf:enabled?                boolean
      |   +--rw tet-sf:direction? connectivity-direction

```

logical-network-elements

```

+--rw logical-network-element* [name]
  +--rw name                string
  +--rw managed?            boolean
  +--rw description?        string
  +--rw root
  +--rw logical-network-elements-properties
  +--rw sf-connection-points* [sf-connection-point-id]
    |   +--rw sf-connection-point-id  string
    |   +--rw ram?                    uint64
    |   +--rw cpu?                    uint64
    +--rw storages* [id]
      |   +--rw id                    string
      |   +--rw location?             string
    +--rw day0-config
      +--rw location?                 string
      +--rw day0-var-path?           string
      +--rw variable* [name]
        +--rw name                    string
        +--rw value?                 string

```

```

ietf-network-instance
  +--rw network-instances
    +--rw network-instance* [name]
      +--rw name string
      +--rw enabled? boolean
      +--rw description? string
      +--rw (ni-type)?
      +--rw (root-type)
        +--:(vrf-root)
          | +--rw vrf-root
        +--:(vsi-root)
          | +--rw vsi-root
          | +--rw ietf-ucpe-ni:network-instance-properties
          |   +--rw ietf-ucpe-ni:sf-connection-points*
          |     | [sf-connection-point-id]
          |     +--rw ietf-ucpe-ni:sf-connection-point-id
          |       | string
          |     +--rw ietf-ucpe-ni:dot1q-vlan
          |       +--rw ietf-ucpe-ni:access-tag?
          |         | dlq:vid-range
          |       +--rw ietf-ucpe-ni:trunk-allowed-vlans?
          |         | dlq:vid-range
          |       +--rw ietf-ucpe-ni:port-mode?
          |         | enumeration
          |     +--rw ietf-ucpe-ni:supporting-node?
          |       -> /nw:networks/network/node/node-id
        +--:(vv-root)
          +--rw vv-root

```

7. Logical Network Elements extension YANG Model

This section provides a YANG models that address uCPE resources management.

```

<CODE BEGINS> file "ietf-ucpe-lne-properties@2019-11-21.yang"
module ietf-ucpe-lne-properties {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-lne-properties";
  prefix ietf-ucpe;

  import ietf-logical-network-element {
    prefix lne;
    reference
      "RFC 8530: YANG Model for Logical Network Elements";
  }
}

```

```
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 2019-11-21 {
  description
    "Augmentation of RFC 8530";
  reference
    "draft-shytyi-opsawg-vysm-07";
}
revision 2019-10-28 {
  description
    "Yang model with vPorts assigned to the interfaces";
  reference
    "draft-shytyi-opsawg-vysm-05";
}
revision 2019-10-19 {
  description
    "Yang model was cleaned. Interfaces added";
  reference
    "draft-shytyi-opsawg-vysm-04";
}
revision 2019-09-16 {
  description
    "Added Oday 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";
}

augment "/lne:logical-network-elements/lne:logical-network-element" {
  container logical-network-element-properties {
    list sf-connection-points {
      key "sf-connection-point-id";
      leaf sf-connection-point-id {
        type string;
        description
          "Name of the connector";
      }
    }
  }
  description
    "Connection points of logical-network-element";
  }
  description
    "Set of Virtual Network Function 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
        "Number of
        vDisk attached to the VM";
    }
  }
  leaf location {
    type string;
    description
      "External location where
      the image (ex.qcow2) is saved.";
  }
}
```

```
    description
      "Virtual storage/vDisk
       attached to 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
        "list of variables";
    }
    description
      "0day configuration:init config";
  }
}
description
  "Properties of logic-network-element";
}
```

<CODE ENDS>

```
<CODE BEGINS> file "ietf-ucpe-ni-properties@2019-11-27.yang"
module ietf-ucpe-ni-properties {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-ni-properties";
```

```
prefix ietf-ucpe-ni;
import ieee-dot1Q-types{
  prefix dlq;
  reference
    "Prefix dot1Q-types.
    Institute of Electrical and Electronics Engineers";
}
import ietf-logical-network-element {
  prefix lne;
  reference
    "RFC 8530: YANG Model for Logical Network Elements";
}
import ietf-network{
  prefix nw;
  reference
    "RFC 8345: YANG Data Model for Network Topologies";
}
import ietf-network-instance{
  prefix ni;
  reference
    "RFC 8529: YANG Data Model for Network Instances";
}
```

organization

"SFR";

contact

"Dmytro Shytyi
EMail:ietf.dmytro@shytyi.net";

description

"This is a YANG data model that
extends Network Instance properties.

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 2019-11-27 {
  description
```

```
    "Added VLANs";
  reference
    "draft-shytyi-opsawg-vysm-07";
}
revision 2019-11-21 {
  description
    "Augmentation of RFC 8530";
  reference
    "draft-shytyi-opsawg-vysm-07";
}
revision 2019-10-28 {
  description
    "Yang model with vPorts assigned to the interfaces";
  reference
    "draft-shytyi-opsawg-vysm-05";
}
revision 2019-10-19 {
  description
    "Yang model was cleaned. Interfaces added";
  reference
    "draft-shytyi-opsawg-vysm-04";
}
revision 2019-09-16 {
  description
    "Added 0day config for VNFs.";
  reference
    "draft-shytyi-opsawg-vysm-00";
}
revision 2018-01-07 {
  description
    "Initial revision.";
  reference
    "draft-shytyi-netmod-vysm-01";
}

augment "/ni:network-instances/ni:network-instance/"
+ "ni:root-type/ni:vsi-root" {
  container network-instance-properties {
    list sf-connection-points {
      key "sf-connection-point-id";
      leaf sf-connection-point-id {
        type string;
        description
          "Name of the connector";
      }
    }
  }
  container dot1q-vlan{
    leaf access-tag{
      type dlq:vid-range;
    }
  }
}
```



```
        description
            "Access tag";
    }
    leaf trunk-allowed-vlans{
        type dlq:vid-range;
    }
    leaf port-mode{
        type enumeration{
            enum trunk;
            enum access;
        }
    }
}
description
    "Dot1q encapsulation parameters";
}
    description
        "Connection points of logical-network-element";
    }
    description
        "Set of Virtual Network Function connectors";
    leaf supporting-node {
        type leafref{
            path "/nw:networks/nw:network/nw:node/nw:node-id";
        }
    }
    description
        "The reference to a physical node which
        hosts the logical element";
    }
}
description
    "Properties of network instance";
}
}
<CODE ENDS>
```

```
<CODE BEGINS> file "ietf-ucpe-node-type@2020-02-14.yang"
module ietf-ucpe-node-type {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-node-type";
    prefix node-type;
    import ietf-network{
        prefix nw;
        reference
            "RFC 8345: YANG Data Model for Network Topologies";
    }
}
```

```
organization
```

```
  "SFR";
```

```
contact
```

```
  "Dmytro Shytyi
```

```
  EMail:ietf.dmytro@shytyi.net";
```

```
description
```

```
  "This is a YANG data model that extends node-type.
```

```
  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 2020-02-14 {
```

```
  description
```

```
    "Added node type";
```

```
  reference
```

```
    "draft-shytyi-opsawg-vysm-08";
```

```
}
```

```
identity node-type {
```

```
  description
```

```
    "Base identity for node-type.";
```

```
}
```

```
identity ucpe {
```

```
  base node-type;
```

```
  description
```

```
    "Identity for mail application.";
```

```
}
```

```
augment "/nw:networks/nw:network/nw:node" {
```

```
  leaf node-type {
```

```
    type identityref{
```

```
      base node-type;
```

```
    }
```

```
  description
```

```
    "Properties of network instance";
```

```
}
```

```
}
```

```
}
```

<CODE ENDS>

```
<CODE BEGINS> file "ietf-ucpe-lt-virtual-link-id@2020-02-14.yang"
module ietf-ucpe-lt-virtual-link-id {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ucpe-lt-virtual-link-id";
  prefix lt-vlink-id;
  import ietf-network{
    prefix nw;
    reference
      "RFC 8345: YANG Data Model for Network Topologies";
  }
  import ietf-te-topology {
    prefix "tet";
    reference
      "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
      Engineering (TE) Topologies";
  }
  import ietf-te-topology-sf{
    prefix "tet-sf";
    reference "RFC XXXX: SF Aware TE Topology YANG Model";
  }
  organization
    "SFR";
  contact
    "Dmytro Shytyi
    EMail:ietf.dmytro@shytyi.net";
  description
    "This is a YANG data model that extends link-termination of
    service function with virtual link.
```

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 2020-02-14 {
  description
    "Added virtual link id";
  reference
    "draft-shytyi-opsawg-vysm-08";
}
augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet-sf:service-function/"
  + "tet-sf:link-terminations/tet-sf:link-termination"{
leaf virtual-link-id {
  type string;
  description
    "Virtual link id between equipment and service function";
}
description
  "Virtual link id between equipment and service function";
}
}
<CODE ENDS>
```

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.
- o Martin Bjorklund.
- o Schonwalder Jurgen.
- o Dean Bogdanovic.

o Bo Wu.

for their valuable comments.

11. Normative References

- [I-D.ietf-teas-sf-aware-topo-model]
 Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-03 (work in progress), March 2019.
- [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>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <<https://www.rfc-editor.org/info/rfc8530>>.

Appendix A. Example of the uCPE resources management

This section provides an overview of the YIN format.

```
<networks xmlns="urn:ietf:params:xml:ns:yang:ietf-network">
  <network>
    <network-id>network-1</network-id>
    <network-types>
      <te-topology xmlns="urn:ietf:params:xml:ns:yang:ietf-te-topology">
        <sf xmlns="urn:ietf:params:xml:ns:yang:ietf-te-topology-sf"/>
      </te-topology>
    </network-types>
    <node>
      <node-id>ucpel</node-id>
      <te-node-id xmlns="urn:ietf:params:xml:ns:yang:ietf-te-topology" \
        >0.0.0.0</te-node-id>
```

```
<te xmlns="urn:ietf:params:xml:ns:yang:ietf-te-topology">
  <te-node-attributes>
    <service-function
      xmlns="urn:ietf:params:xml:ns:yang:ietf-te-topology-sf">
      <connectivity-matrices>
        <connectivity-matrix>
          <id>1</id>
          <from>
            <service-function-id>VMone</service-function-id>
            <sf-connection-point-id>1</sf-connection-point-id>
          </from>
          <to>
            <service-function-id>SwitchOne</service-function-id>
            <sf-connection-point-id>11</sf-connection-point-id>
          </to>
          <virtual-link-id>l11</virtual-link-id>
        </connectivity-matrix>
        <connectivity-matrix>
          <id>2</id>
          <from>
            <service-function-id>VMtwo</service-function-id>
            <sf-connection-point-id>1</sf-connection-point-id>
          </from>
          <to>
            <service-function-id>SwitchOne</service-function-id>
            <sf-connection-point-id>12</sf-connection-point-id>
          </to>
          <virtual-link-id>l12</virtual-link-id>
        </connectivity-matrix>
        <connectivity-matrix>
          <id>3</id>
          <from>
            <service-function-id>VMthree</service-function-id>
            <sf-connection-point-id>1</sf-connection-point-id>
          </from>
          <to>
            <service-function-id>SwitchOne</service-function-id>
            <sf-connection-point-id>13</sf-connection-point-id>
          </to>
          <virtual-link-id>l13</virtual-link-id>
        </connectivity-matrix>
        <connectivity-matrix>
          <id>4</id>
          <from>
            <service-function-id>VMfour</service-function-id>
            <sf-connection-point-id>1</sf-connection-point-id>
          </from>
          <to>
```

```

        <service-function-id>SwitchOne</service-function-id>
        <sf-connection-point-id>14</sf-connection-point-id>
    </to>
    <virtual-link-id>l14</virtual-link-id>
</connectivity-matrix>
</connectivity-matrices>
</service-function>
</te-node-attributes>
</te>
</node>
</network>
</networks>

<logical-network-elements \
  xmlns="urn:ietf:params:xml:ns:yang:ietf-logical-network-element">
  <logical-network-element>
    <name>VMfour</name>
    <logical-network-element-properties \
      xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe-lne-properties">
      <sf-connection-points>
        <sf-connection-point-id>1</sf-connection-point-id>
      </sf-connection-points>
      <supporting-node>ucpel</supporting-node>
      <ram>1024</ram>
      <cpu>4</cpu>
      <storages>
        <id>1</id>
        <location>vm4.qcow2</location>
      </storages>
    </logical-network-element-properties>
  </logical-network-element>
  <logical-network-element>
    <name>VMone</name>
    <logical-network-element-properties \
      xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe-lne-properties">
      <sf-connection-points>
        <sf-connection-point-id>1</sf-connection-point-id>
      </sf-connection-points>
      <supporting-node>ucpel</supporting-node>
      <ram>1024</ram>
      <cpu>4</cpu>
      <storages>
        <id>1</id>
        <location>vm1.qcow2</location>
      </storages>
    </logical-network-element-properties>
  </logical-network-element>
</logical-network-element>

```

```
<name>VMthree</name>
<logical-network-element-properties \
  xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe-lne-properties">
  <sf-connection-points>
    <sf-connection-point-id>1</sf-connection-point-id>
  </sf-connection-points>
  <supporting-node>ucpe</supporting-node>
  <ram>1024</ram>
  <cpu>4</cpu>
  <storages>
    <id>1</id>
    <location>vm3qcow2</location>
  </storages>
</logical-network-element-properties>
</logical-network-element>
<logical-network-element>
  <name>VMtwo</name>
  <logical-network-element-properties \
    xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe-lne-properties">
    <sf-connection-points>
      <sf-connection-point-id>1</sf-connection-point-id>
    </sf-connection-points>
    <supporting-node>ucpel</supporting-node>
    <ram>1024</ram>
    <cpu>4</cpu>
    <storages>
      <id>1</id>
      <location>vm4.iso</location>
    </storages>
  </logical-network-element-properties>
</logical-network-element>
</logical-network-elements>
<network-instances \
  xmlns="urn:ietf:params:xml:ns:yang:ietf-network-instance">
<network-instance>
  <name>SwitchOne</name>
  <network-instance-properties \
    xmlns="urn:ietf:params:xml:ns:yang:ietf-ucpe-ni-properties">
    <sf-connection-points>
      <sf-connection-point-id>10</sf-connection-point-id>
      <dot1q-vlan>
        <trunk-allowed-vlans>112</trunk-allowed-vlans>
        <trunk-allowed-vlans>113</trunk-allowed-vlans>
        <trunk-allowed-vlans>114</trunk-allowed-vlans>
        <port-mode>trunk</port-mode>
      </dot1q-vlan>
    </sf-connection-points>
  </network-instance-properties>
</network-instance>
</network-instances>
```



```
    <sf-connection-point-id>11</sf-connection-point-id>
  </sf-connection-points>
  <dot1q-vlan>
    <access-tag>111</access-tag>
  </dot1q-vlan>
<sf-connection-points>
  <sf-connection-point-id>12</sf-connection-point-id>
</sf-connection-points>
<sf-connection-points>
  <sf-connection-point-id>13</sf-connection-point-id>
</sf-connection-points>
<sf-connection-points>
  <sf-connection-point-id>14</sf-connection-point-id>
</sf-connection-points>
  <supporting-node>ucpel</supporting-node>
</network-instance-properties>
</network-instance>
</network-instances>
```

Appendix B. Example of the uCPE resources management (deprecated)

This section provides an overview of the deprecated YANG Model that MAY give an alternative view on the uCPE management.

```

module: ietf-example-ucpe
  +--rw ucpe* [name]
    +--rw name          string
    +--rw links* [link]
      | +--rw link      string
    +--rw phyInterfaces* [interface]
      | +--rw interface  string
      | +--rw ports* [port]
      |   +--rw port      string
      |   +--rw link?    -> ../../../../links/link
    +--rw switches* [switch]
      | +--rw switch     string
      | +--rw ports* [port]
      |   +--rw port      string
      |   +--rw name?     string
      |   +--rw link?    -> ../../../../links/link
    +--rw vms* [vm]
      +--rw vm           string
      +--rw ports* [port]
        | +--rw port      string
        | +--rw name?     string
        | +--rw link?    -> ../../../../links/link
      +--rw ram?         uint64
      +--rw cpu?         uint64
      +--rw storages* [id]
        | +--rw id         string
        | +--rw location?  string
      +--rw day0-config
        +--rw location?    string
        +--rw day0-var-path? string
        +--rw variable* [name]
          +--rw name       string
          +--rw value?     string

```

Appendix C. Deprecated VNF YANG Model

This section provides a deprecated yang model that addresses the configuration of the uCPE resources presented above.

```

<CODE BEGINS> file "ietf-example-ucpe@2019-10-28.yang"
module ietf-example-ucpe {
  namespace "urn:ietf:params:xml:ns:yang:ietf-example-ucpe";
  prefix ietf-example-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 2019-10-28 {
  description
    "Yang model with vPorts assigned to the interfaces";
  reference
    "draft-shytyi-opsawg-vysm-05";
}
revision 2019-10-19 {
  description
    "Yang model was cleaned. 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
      "ID of uCPE where
       a service is instantiated";
  }
  list links {
    key "link";
    leaf link {
      type string;
      description
        "Name of the virtual link from the pool
         of the links";
    }
    description
      "Pool of the virtual links that connect VMs and
       Interfaces";
  }
  list phyInterfaces {
    key "interface";
    leaf interface {
      type string;
      description
        "Name of physical interface";
    }
    list ports {
      key "port";
      leaf port {
        type string;
        description
          "Name of the connector";
      }
      leaf link {
        type leafref {
          path "../.../links/link";
        }
        description
          "Link that is connected to
           the port via connector";
      }
      description
        "Set of the connectors the
         physical interface has";
    }
    description
      "Set of physical interfaces";
  }
}
```

```
}
list switches {
  key "switch";
  leaf switch {
    type string;
    description
      "Name of the forwarding domain";
  }
  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 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;
    description
      "ID of the Virtual Machine";
  }
  list ports {
    key "port";
    leaf port {
      type string;
      description

```

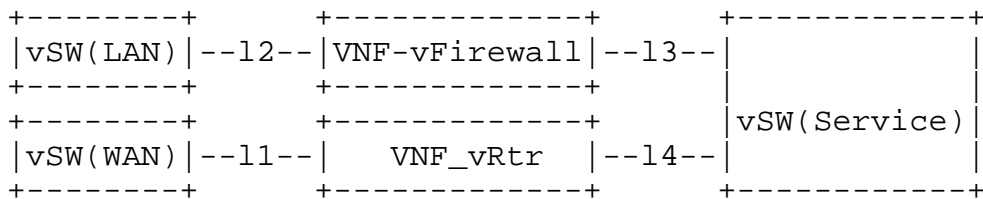
```
        "Name of the connector";
    }
    leaf name {
        type string;
        description
            "Name of
             the subconnector";
    }
    leaf link {
        type leafref {
            path "../..../links/link";
        }
        description
            "Link that connects the
             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
            "Number of
             vDisk attached to the VM";
    }
    leaf location {
        type string;
        description
            "External location where
             the image (ex.qcow2) is saved.";
    }
    description
        "Virtual storage/vDisk
```

```
        attached to 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
                "list of variables";
        }
        description
            "0day configuration:init config";
    }
    description
        "Set of the Virtual Machines configured
        on the universal Customer-Premises Equipment";
}
description
    "This is an uCPE management service";
}
}
```

<CODE ENDS>

Appendix D. XML example of deprecated YANG model

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; VNFs represent PNFs (Physical Network Functions) that could be bootstrapped with 0day config/license.



```

<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>
  </switches>
</ucpe>

```



```
    <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-vRtr</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/vRtr-x86.qcow2</location>
  </storages>
  <day0-config>
    <location>https://192.168.2.1/vRtr-day0.iso</location>
    <day0-var-path>/config.rom</day0-var-path>
    <variable>
      <name>hostname</name>
      <value>IETF-vRtr</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>
```

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