Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2020

D. Shytyi
L. Beylier
SFR
L. Iannone
Telecom ParisTech

October 29, 2019

A YANG Module for uCPE management. draft-shytyi-opsawg-vysm-06

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 May 1, 2020.

Table of Contents

1. Introduction

Network Function Virtualization is a technology that allows to virtualize the network services running on dedicaded hardware. This technology became a base for universal Customer-Premises Equipment (uCPE). This document defines the uCPE as harware 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.

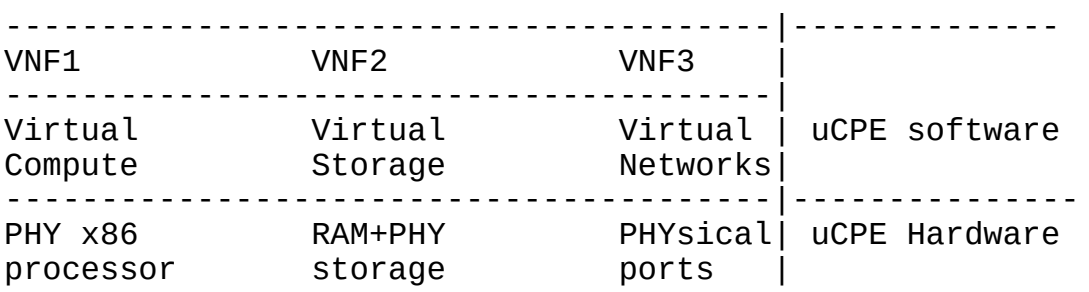
NE - Network Element.

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 harware with x86 capabilities that is generally running Linux distibution with additinal virtualisation layer. Virtualization layer provides virtual compute, virtual storage and virtual network resources. Each VNF runnning in the uCPE requires the amount of virtual resources (for example: 4 vCPUs, 4GB RAM, 40GB storege, 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.

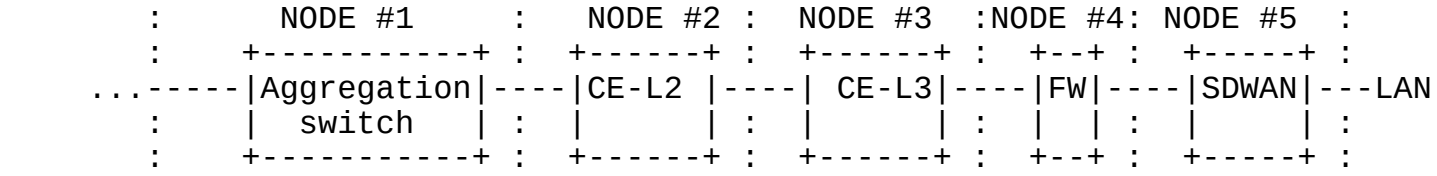


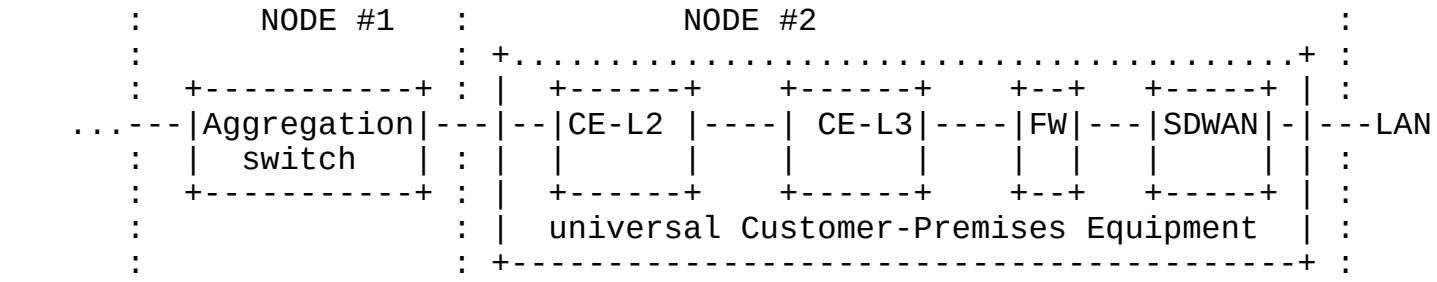
The next elements can be managed in the uCPE:

- Virtual Network Funcitons:
  - Number of assigned vCPUs.
  - Size of allocated RAM.
  - VNF day0 config (bootstrap).
  - vLinks that are attached to the VNF.
- Virtual Switches:
  - vLinks that are attached to the vSW.
- Virtual Links(vLinks).
- Physical Ports of the uCPE.

3.1. uCPE purpose

- 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:





- uCPE falicitates the interconnection between the Network Funtions (NF) as interconnection between NF is performed via virtual links(that is part of the uCPE management). That meens that no need to hire technichian to cable the equipment, it could be done via orchestrator.

- uCPE falicitates 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 Funcitons of different type:

- SD-WAN

- vRouter(vCPE)

- vFirewall

- vLB(vLoad Balancer)

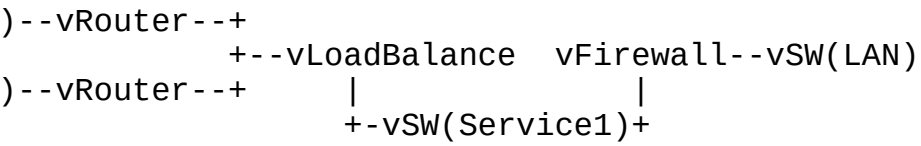
- vCGNAT(vCarrier Grade NAT)

- virtual WAN Optimistaion
- vWireless LAN controller
- Other...

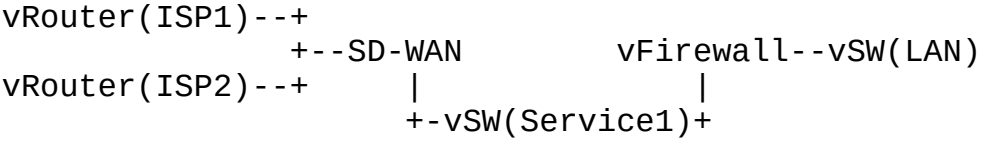
3.3. Internal uCPE service example

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):

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



-

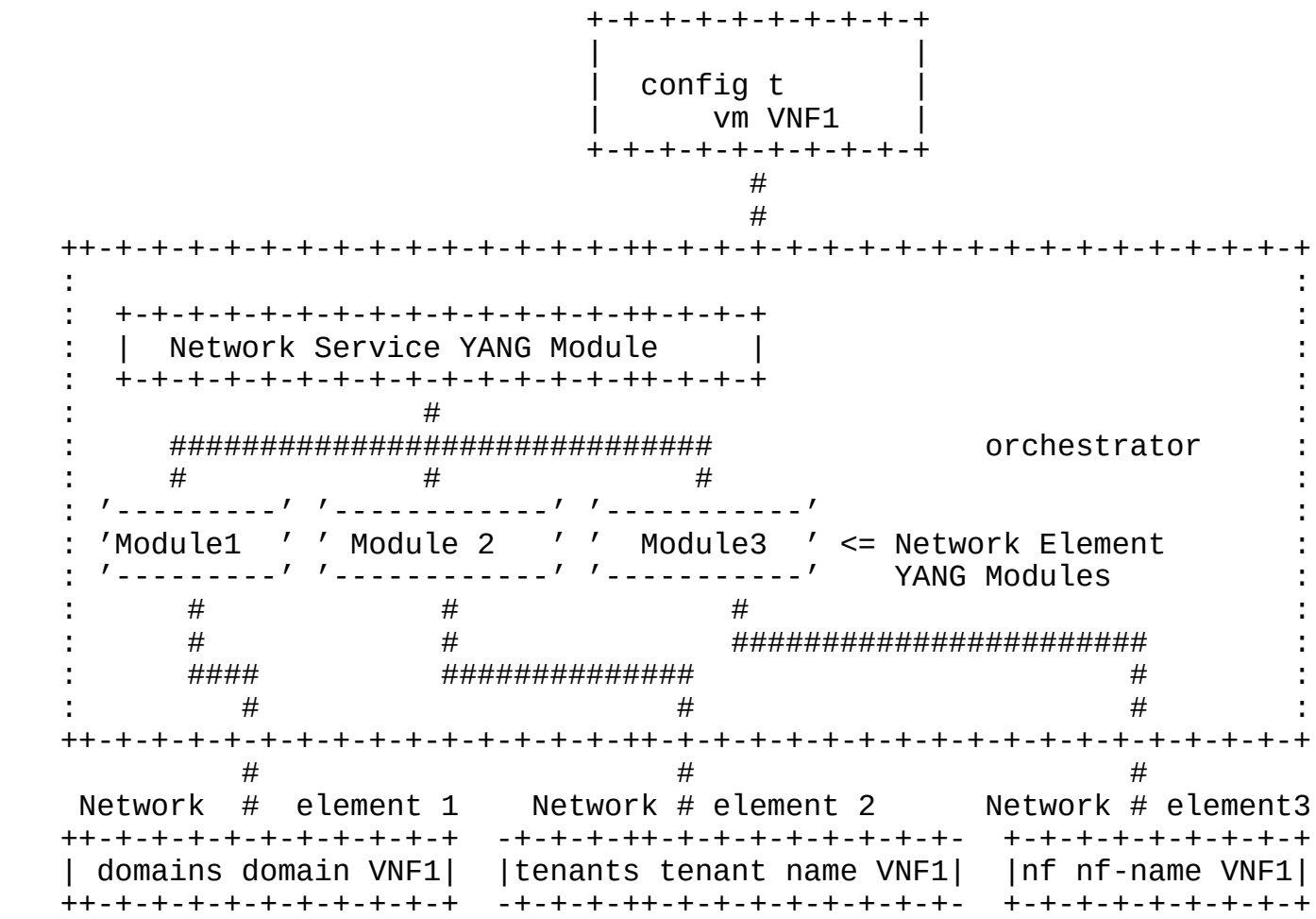


4. YANG Service Model for uCPE management

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 Elemets (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 inputed via Notrhbound 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 transormed 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"



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 VYSM.

```
module: ietf-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
```

6. Specification of the VNF YANG Service Model

This section presents the specification of the VYSM.

```
<CODE BEGINS> file "ietf-ucpe@2019-10-28.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 2019-10-28 {
      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 storge/vDisk
               attached to the Virtual Machine";
```

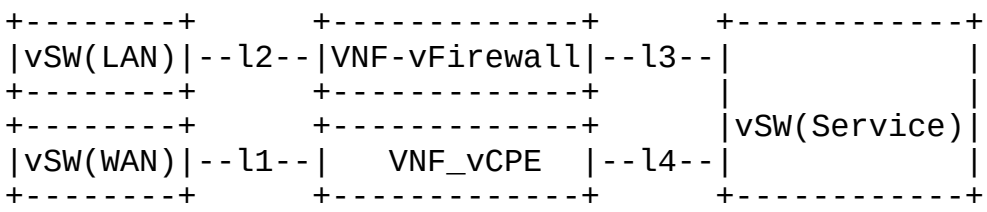
```
        }
        container day0-config {
          leaf location {
            type string;
            description
              "0day configuration location";
          }
          leaf day0-var-path {
            type string;
            description
              "path of the file
               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>
```

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 Fuctions) 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>
      <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:

- Mahesh Jethanandani.
- Robert Varga.
- Bill Wu.
- Joe Clarke.
- Tom Petch.
- Martin Bjorklund.

o Schonwalder Jurgen.

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