CCAMP Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 April 2022

C. Yu
I. Busi
Huawei Technologies
A. Guo
Futurewei Technologies
S. Belotti
Nokia
J-F. Bouquier
Vodafone
F. Peruzzini
TIM
O.G.d. Dios
Telefonica
V. Lopez
Nokia

25 October 2021

A YANG Data Model for Optical Network Inventory
draft-yg3bp-ccamp-optical-inventory-yang-00

Abstract

This document defines a YANG data model for optical network inventory data information.

The YANG data model presented in this document is intended to be used as the basis toward a generic YANG data model for network inventory data information which can be augmented, when required, with technology-specific (e.g., optical) inventory data, to be defined either in a future version of this document or in another document.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA).

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/.

Copyright Notice

Table of Contents

1. Introduction

Network inventory management is a key component in operators' OSS architectures.

Network inventory is a fundamental functionality in network management and was specified many years ago. Given the emerging of data models and their deployment in operator's management and control systems, the traditional function of inventory management is also requested to be defined as a data model.

Network inventory management and monitoring is a critical part of ensuring the network stays healthy, well-planned, and functioning in the operator's network. Network inventory management allows the operator to keep track of what physical network devices are staying in the network including relevant software and hardware.

The network inventory management also helps the operator to know when to acquire new assets and what is needed, or to decommission old or faulty ones, which can help to improve network performance and capacity planning.

In [I-D.ietf-teas-actn-poi-applicability] a gap was identified regarding the lack of a YANG data model that could be used at ACTN MPI interface level to report whole/partial hardware inventory information available at PNC level towards north-bound systems (e.g., MDSC or OSS layer).

[RFC8345] initial goal was to make possible the augmentation of the YANG data model with network inventory data model but this was never developed and the scope was kept limited to network topology data only.

It is key for operators to drive the industry towards the use of a standard YANG data model for network inventory data instead of using vendors proprietary APIs (e.g., REST API).

In the ACTN architecture, this would bring also clear benefits at MDSC level for packet over optical integration scenarios since this would enable the correlation of the inventory information with the links information reported in the network topology model.

The intention is to define a generic YANG data model that would be as much as possible technology agnostic (valid for IP, optical and microwave networks) and that could be augmented, when required, to include some technology-specific inventory details.

[RFC8348] defines a YANG data model for the management of the hardware on a single server and therefore it is more applicable to the PNC South Bound Interface (SBI) towards the network elements rather than at the PNC MPI. However, the YANG data model defined in [RFC8348] has been used as a reference for defining the YANG network inventory data model.

For optical network inventory, the network inventory YANG data model should support the use cases (4a and 4b) and requirements defined in [ONF_TR-547], in order to guarantee a seamless integration at MDSC/OSS/orchestration layers.

The proposed YANG data model has been analysed to cover the requirements and use cases for Optical Network Inventory.

Being based on [RFC8348], this data model should be a good starting point toward a generic data model and applicable to any technology. However, further analysis of requirements and use cases is needed to extend the applicability of this YANG data model to other types of networks (IP and microwave) and to identify which aspects are generic and which aspects are technology-specific for optical networks.

This document defines one YANG module: ietf-network-inventory.yang (Section 4).

Note: review in future versions of this document the related modules, depending on the augmentation relationship.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture [RFC8342].

1.1. Terminology and Notations

Refer to [RFC7446] and [RFC7581] for the key terms used in this document. The following terms are defined in [RFC7950] and are not redefined here:

* client
* server
* augment
* data model
* data node

The following terms are defined in [RFC6241] and are not redefined here:

* configuration data
* state data

The terminology for describing YANG data models is found in [RFC7950].

TBD: Recap the concept of chassis/slot/component/board/... in [TMF-MTOSI].

Following terms are used for the representation of the hierarchies in the optical network inventory.

Network Element: a device installed on one or several shelves and can afford some specific transmission function independently.

Cabinet: a holder of the device and provides power supply for the device in it.

Chassis: a holder of the device installation.

Slot: a holder of the board.

Component: holders and equipments of the network element, including rack, shelf, slot, sub-slot, board and port.

Board/Card: a pluggable equipment on the network element and can afford a specific transmission function independently.

Port: an interface on board

1.2. Tree Diagram

A simplified graphical representation of the data model is used in Section 3 of this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. Prefix in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in the following table.

Prefix	Yang Module	Reference
ianahw	iana-hardware	[RFC8348]
ni	ietf-network-inventory	RFCXXX
yang	ietf-yang-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

RFC Editor Note: Please replace XXXX with the RFC number assigned to this document. Please remove this note.

2. YANG Data Model for Optical Network Inventory

2.1. YANG Model Overview

Based on TMF classification in [TMF-MTOSI], inventory objects can be divided into two groups, holder group and equipment group. The holder group contains rack, shelf, slot, sub-slot while the equipment group contains network-element, board and port. With the requirement of GIS and on-demand domain controller selection raised, the equipment room becomes a new inventory object to be managed besides TMF classification.

Logically, the relationship between these inventory objects can be described by Figure 1 below:

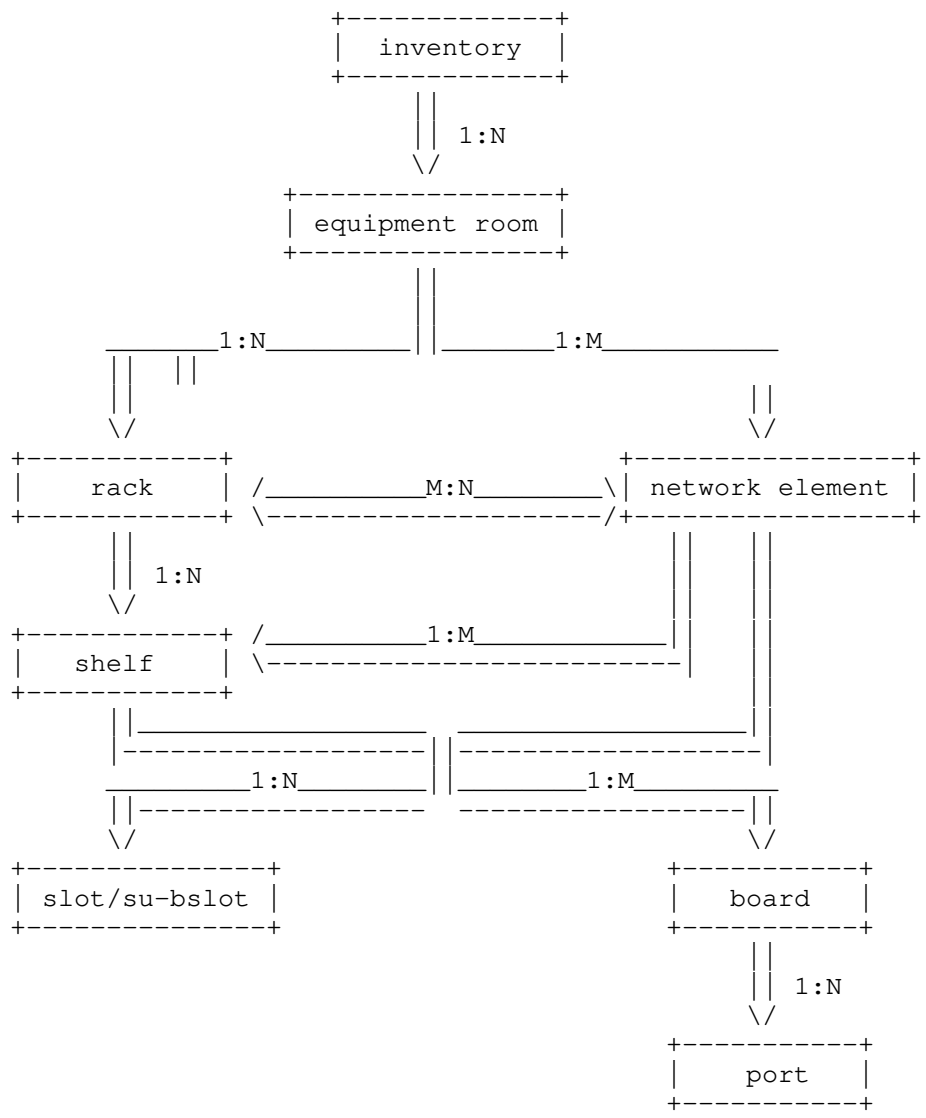

Figure 1: Relationship between inventory objects

In [RFC8348], rack, shelf, slot, sub-slot, board and port are defined as components of network elements with generic attributes.

While [RFC8348] is used to manage the hardware of a single server (e.g., a Network Element), the Network Inventory YANG data model is used to retrieve the network inventory information that a controller discovers from multiple Network Elements under its control.

However, the YANG data model defined in [RFC8348] has been used as a reference for defining the YANG network inventory data model. This approach can simplify the implementation of this network inventory model when the controller uses the YANG data model defined in [RFC8348] to retrieve the hardware configuration from the network elements under its control.

Note: review in future versions of this document which attributes from [RFC8348] are required also for network inventory and whether there are attributes not defined in [RFC8348]which are required for network inventory

Note: review in future versions of this document whether to re-use definitions from [RFC8348] or use schema-mount.

```
  +--ro network-inventory
     +--ro equipment-rooms
     |  +--ro equipment-room* [uuid]
     |     +--ro uuid        yang:uuid
     |     ...................................
     |     +--ro rack* [uuid]
     |        +--ro uuid           yang:uuid
     |        ...................................
     |        +--ro shelves* [uuid]
     |           +--ro uuid            yang:uuid
     |           ...................................
     |           +--ro chassis-ref
     |              +--ro ne-ref?          leafref
     |              +--ro component-ref?   leafref
     |
     +--ro network-elements
        +--ro network-element* [uuid]
           +--ro uuid          yang:uuid
           ...................................
           +--ro components
              +--ro component* [uuid]
                 +--ro uuid              yang:uuid
                 ...................................
```

The YANG data model for network inventory follows the same approach of [RFC8348] and reports the network inventory as a list of components of different types (e.g., chassis, module, port).

```
  +--ro components
     +--ro component* [uuid]
        +--ro uuid              yang:uuid
        +--ro name?             string
        +--ro description?      string
        +--ro class?            identityref
        +--ro parent-rel-pos?   int32
        +--ro children* [child-ref]
        |  +--ro child-ref    -> ../../../../uuid
        +--ro parent
           +--ro parent-ref?   -> ../../../../uuid
```

Note: review in future versions of this document whether the component list should be under the network-inventory instead of under the network-element container

However, considering there are some special scenarios, the relationship between the rack and network elements is not 1 to 1 nor 1 to n. The network element cannot be the direct parent node of the rack. So there should be n to m relationship between racks and network elements. And the shelves in the rack should have some reference information to the component.

Note that in [RFC8345], topology and inventory are two subsets of network information. However, considering the complexity of the existing topology models and to have a better extension capability, we define a separate root for the inventory model. We will consider some other ways to do some associations between the topology model and inventory model in the future.

Note: review in future versions of this document whether network inventory should be defined as an augmentation of the network model defined in [RFC8345] instead of under a new network-inventory root.

The proposed YANG data model has been analysed to cover the requirements and use cases for Optical Network Inventory.

Further analysis of requirements and use cases is needed to extend the applicability of this YANG data model to other types of networks (IP and microwave) and to identify which aspects are generic and which aspects are technology-specific for optical networks.

3. Optical Network Inventory Tree Diagram

Figure 2 below shows the tree diagram of the YANG data model defined in module ietf-network-inventory.yang (Section 4).

```
module: ietf-network-inventory
  +--ro network-inventory
     +--ro equipment-rooms
     |  +--ro equipment-room* [uuid]
     |     +--ro uuid        yang:uuid
     |     +--ro name?       string
     |     +--ro location?   string
     |     +--ro rack* [uuid]
     |        +--ro uuid           yang:uuid
     |        +--ro name?          string
     |        +--ro row-number?    uint32
     |        +--ro rack-number?   uint32
     |        +--ro shelves* [uuid]
     |           +--ro uuid            yang:uuid
     |           +--ro name?           string
     |           +--ro shelf-number?   uint8
     |           +--ro chassis-ref
     |              +--ro ne-ref?          leafref
     |              +--ro component-ref?   leafref
     +--ro network-elements
        +--ro network-element* [uuid]
           +--ro uuid          yang:uuid
           +--ro name?         string
           +--ro components
              +--ro component* [uuid]
                 +--ro uuid              yang:uuid
                 +--ro name?             string
                 +--ro description?      string
                 +--ro class?            identityref
                 +--ro parent-rel-pos?   int32
                 +--ro children* [child-ref]
                 |  +--ro child-ref    -> ../../../../uuid
                 +--ro parent
                    +--ro parent-ref?   -> ../../../../uuid
```

Figure 2: Network inventory tree diagram

4. YANG Model for Optical Network Inventory

```
<CODE BEGINS> file "ietf-network-inventory@2021-10-25.yang"
module ietf-network-inventory {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-inventory";
  prefix ni;

  import ietf-yang-types {
    prefix yang;
    reference
```

```
        "RFC6991: Common YANG Data Types.";
    }

    import iana-hardware {
      prefix ianahw;
      reference
        "RFC 8348: A YANG Data Model for Hardware Management.";
    }

    organization
      "IETF CCAMP Working Group";
    contact
      "WG Web:   <https://datatracker.ietf.org/wg/ccamp/>
       WG List:  <mailto:ccamp@ietf.org>

       Editor:   Chaode Yu
                 <yuchaode@huawei.com>

       Editor:   Italo Busi
                 <italo.busi@huawei.com>

       Editor:   Aihua Guo
                 <aihuaguo.ietf@gmail.com>

       Editor:   Sergio Belotti
                 <sergio.belotti@nokia.com>

       Editor:   Jean-Francois Bouquier
                 <jeff.bouquier@vodafone.com>

       Editor:   Fabio Peruzzini
                 <fabio.peruzzini@telecomitalia.it>";

    description
      "This module defines a model for retrieving network inventory.

      The model fully conforms to the Network Management
      Datastore Architecture (NMDA).

      Copyright (c) 2021 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; 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.";

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note.
    // RFC Ed.: update the date below with the date of RFC publication
    // and remove this note.

    revision 2021-10-25 {
      description
        "Initial revision.";
      reference
        "draft-yg3bp-ccamp-optical-inventory-yang-00: A YANG Data
        Model for Optical Network Inventory.";
    }

    container network-inventory {
      config false;
      description
        "The top-level container for the network inventory
        information.";
      uses equipment-rooms-grouping;
      uses network-elements-grouping;
    }

    grouping common-entity-attributes {
      description
        "A set of attributes which are common to all the entities
        (e.g., component, equipment room) defined in this module.";
      leaf uuid {
        type yang:uuid;
        description
          "Uniquely identifies an entity (e.g., component).";
      }
      leaf name {
        type string;
        description
          "A name for an entity (e.g., component), as specified by
          a network manager, that provides a non-volatile 'handle'
          for the entity and that can be modified anytime during the
          entity lifetime.
```

```
          If no configured value exists, the server MAY set the value
          of this node to a locally unique value in the operational
          state.";
      }
    }
    grouping network-elements-grouping {
      description
        "The attributes of the network elements.";
      container network-elements {
        description
          "The container for the list of network elements.";
        list network-element {
          key uuid;
          description
            "The list of network elements within the network.";
          uses common-entity-attributes;
          uses components-grouping;
        }
      }
    }

    grouping equipment-rooms-grouping {
      description
        "The attributes of the equipment rooms.";
      container equipment-rooms {
        description
          "The container for the list of equipment rooms.";
        list equipment-room {
          key uuid;
          description
            "The list of equipment rooms within the network.";
          uses common-entity-attributes;
          leaf location {
            type string;
            description
              "compared with the location information of the other
              inventory objects, a GIS address is preferred for
              equipment room";
          }
          list rack {
            key uuid;
            description
              "The list of racks within an equipment room.";
            uses common-entity-attributes;
            leaf row-number {
              type uint32;
              description
                "Identifies the row within the equipment room where
```

```
                the rack is located.";
            }
            leaf rack-number {
              type uint32;
              description
                "Identifies the physical location of the rack within
                the row.";
            }
            list shelves {
              key uuid;
              description
                "The list of shelves within a rack.";
              uses common-entity-attributes;
              leaf shelf-number {
                type uint8;
                description
                  "Identifies the location of the shelf within the
                  rack.";
              }
              container chassis-ref {
                description
                  "The reference to the network element component
                  representing this shelf.";
                leaf ne-ref {
                  type leafref {
                    path "/ni:network-inventory/ni:network-elements"
                      + "/ni:network-element/ni:uuid";
                  }
                  description
                    "The reference to the network element containing
                    the component.";
                }
                leaf component-ref {
                  type leafref {
                    path "/ni:network-inventory/ni:network-elements"
                       + "/ni:network-element[ni:uuid"
                       + "=current()/../ne-ref]/ni:components"
                       + "/ni:component/ni:uuid";
                  }
                  description
                    "The reference to the component within the network
                    element.";
                }
              }
            }
          }
        }
      }
```

```
    }

    grouping components-grouping {
      description
        "The attributes of the hardware components.";
      container components {
        description
          "The container for the list of components.";
        list component {
          key uuid;
          description
            "The list of components within a network element.";
          uses common-entity-attributes;
          leaf description {
            type string;
            description
              "A textual description of the component.";
            reference
              "RFC 8348: A YANG Data Model for Hardware Management.";
          }
          leaf class {
            type identityref {
              base ianahw:hardware-class;
            }
            description
              "An indication of the general hardware type of the
               component.";
            reference
              "RFC 8348: A YANG Data Model for Hardware Management.";
          }
          leaf  parent-rel-pos {
            type int32 {
              range "0 .. 2147483647";
            }
            description
              "An indication of the relative position of this child
               component among all its sibling components.  Sibling
               components are defined as components that:

                 o share the same value of the 'parent' node and

                 o share a common base identity for the 'class' node.";
            reference
              "RFC 8348: A YANG Data Model for Hardware Management.";
          }
          list children {
            key child-ref;
            description
```

```
              "The child components that are physically contained by
              this component.";

            leaf child-ref {
              type leafref {
                path "../../../../ni:uuid";
              }
              description
                "The reference to the child component.";
            }
          }
          container parent {
            description
              "The parent component that physically contains this
               component.

               If this container is not instantiated, it indicates
               that this component is not contained in any other
               component.

               In the event that a physical component is contained by
               more than one physical component (e.g., double-wide
               modules), this container contains the data of one of
               these components.  An implementation MUST use the same
               component every time this container is instantiated.";
            leaf parent-ref {
              type leafref {
                path "../../../../ni:uuid";
              }
              description
                "The reference to the parent component.";
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 3: Network inventory YANG module

5. Manageability Considerations

<Add any manageability considerations>

6. Security Considerations

<Add any security considerations>

7. IANA Considerations

<Add any IANA considerations>

8. References

8.1. Normative References

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7446] Lee, Y., Ed., Bernstein, G., Ed., Li, D., and W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, DOI 10.17487/RFC7446, February 2015, <https://www.rfc-editor.org/info/rfc7446>.

[RFC7581] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", RFC 7581, DOI 10.17487/RFC7581, June 2015, <https://www.rfc-editor.org/info/rfc7581>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", RFC 8348, DOI 10.17487/RFC8348, March 2018, <https://www.rfc-editor.org/info/rfc8348>.

[TMF-MTOSI] TM Forum (TMF), "TMF MTOSI 4.0 Equipment Model", TMF SD2-20_EquipmentModel , 2008, <https://www.tmforum.org/resources/suite/mtosi-4-0/>.

8.2. Informative References

[I-D.ietf-teas-actn-poi-applicability] Peruzzini, F., Bouquier, J., Busi, I., King, D., and D. Ceccarelli, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Packet Optical Integration (POI)", Work in Progress, Internet-Draft, draft-ietf-teas-actn-poi-applicability-03, 12 July 2021, <https://www.ietf.org/archive/id/draft-ietf-teas-actn-poi-applicability-03.txt>.

[ONF_TR-547] Open Networking Foundation (ONF), "TAPI v2.1.3 Reference Implementation Agreement", ONF TR-547 TAPI RIA v1.0 , July 2020, <https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf>.

[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>.

Acknowledgments

The authors of this document would like to thank the authors of [I-D.ietf-teas-actn-poi-applicability] for having identified the gap and requirements to trigger this work.

This document was prepared using kramdown.

Authors' Addresses

Chaode Yu
Huawei Technologies

Email: yuchaode@huawei.com

Italo Busi
Huawei Technologies

Email: italo.busi@huawei.com

Aihua Guo
Futurewei Technologies

Email: aihuaguo.ietf@gmail.com

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com

Jean-Francois Bouquier
Vodafone

Email: jeff.bouquier@vodafone.com

Fabio Peruzzini
TIM

Email: fabio.peruzzini@telecomitalia.it

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Victor Lopez
Nokia

Email: victor.lopez@nokia.com