



HAL
open science

Exposing an Openflow switch abstraction of the satellite segment to Virtual Network Operators

Slim Abdellatif, Pascal Berthou, Patrick Gelard, Thierry Plesse, Sanae El-Yousfi

► To cite this version:

Slim Abdellatif, Pascal Berthou, Patrick Gelard, Thierry Plesse, Sanae El-Yousfi. Exposing an Openflow switch abstraction of the satellite segment to Virtual Network Operators. IEEE 83rd Vehicular Technology Conference: VTC2016-Spring, May 2016, Nanjing, China. 5p. hal-01267192

HAL Id: hal-01267192

<https://hal.science/hal-01267192>

Submitted on 4 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exposing an Openflow switch abstraction of the satellite segment to Virtual Network Operators

Slim Abdellatif^{b,a}, Pascal Berthou^{a,c}, Patrick Gelard^d, Thierry Plesse^e, Sanae El-Yousfi^a

^a CNRS, LAAS, 7 avenue du colonel Roche, F-31400

^b Univ de Toulouse, INSA, LAAS, F-31400, Toulouse, France

^c Univ de Toulouse, UPS, LAAS, F-31400, Toulouse, France^e

^d CNES, Centre National d'Etudes Spatiales 18 Av. Edouard Belin, 31400, Toulouse, France

^e Directorate General of Armaments (DGA), La Roche Marguerite, 35998 Rennes, France

Abstract— For years, Virtual Network Operators (VNOs) are a major player in the satellite communication market landscape. Typically, they repackage services leased from Satellite Network Operators (SNOs) to provide their customers with added-value end-to-end services that mostly go beyond the satellite network and cover terrestrial networks. However, the level of control and visibility that VNOs have on their purchased satellite services (and the underlying satellite resources) is limited mainly because of SNOs' protective policies and the closed nature of satellite devices. From the VNO perspective, this refrains the development of novel satellite communication services and complicates the provision process of the services that they offer to their customers. To address these limitations, this paper proposes and elaborates on the idea of exposing to VNOs, an Openflow based switch abstraction of the satellite segment. The feasibility and the opportunities brought by such an abstraction to VNOs are presented as well as the technical requirements to make it a reality. Insights on how it can be implemented are presented complemented with a proof of concept implementation on the OpenSAND satellite system emulation platform.

Keywords — *Satellite communications; Software Defined networking; Openflow; Virtual Network Operator*

I. INTRODUCTION

For cost reasons, VNOs are common players in the satellite communication landscape. Usually, in addition to the satellite resources that they lease from SNOs (Satellite Network Operators), they also own a terrestrial infrastructure. Indeed, this latter is needed to support their end-to-end satellite services which typically involve in one side satellite terminals and on the other side terrestrial terminals.

Similarly to terrestrial commercial operators, VNOs selling satellite services are faced with an ever-increasing pressure to deliver new and advanced service offerings. To answer this need, a comprehensive level of visibility and control on their network resources is a necessity. This requirement can be satisfied on the terrestrial network infrastructure since it is usually the property of the VNOs (and if not, it can be achieved with emerging terrestrial network virtualization technologies). This is clearly not the case for satellite resources where the control capabilities and the visibility that are exposed to VNOs are limited and vendor-dependent.

Worse, some control capabilities require intervention from the SNO with potentially a human in the loop.

This paper addresses the problem of providing VNOs with a capability-rich, high-level and satellite-technology-independent interface (for control and monitoring) to their leased satellite resources. It proposes to expose an Openflow-based switch abstraction of the satellite segment to VNOs with the following foreseen benefits. First, VNOs inherit from the general advantages of Software Defined Networking (SDN), notably, simplified (partly automated) network management [1], vendor independent interfaces, and technology independent forwarding abstraction. This latter point is crucial to VNOs that operate a hybrid network infrastructure (composed of terrestrial and satellite networks). Indeed, a better integration of both types of networks can be envisioned. With the increasing adoption of SDN/Openflow in terrestrial networks, the unification of the network management between these networks can also be envisioned. Second, by definition, this abstraction allows VNOs to program their leased satellite resources in real-time by independently developed software. This paves the way to automated, on-demand satellite service provisioning, better satellite resource utilization and as a consequence enables and eases the development of novel dynamic VNO satellite service offers.

This paper discusses some of the questions raised by the adoption of the proposed abstraction. Through simple use cases, it shows the extent to which this abstraction fits into current and forthcoming VNO services. It also highlights some extensions or complements to the abstraction and identifies its main implications on SNOs. Finally, it briefly describes a proof-of-concept implementation on the Opensand satellite telecommunication system emulation platform [2].

This paper is organized as follows. Section II describes the problem under study and presents the terminology adopted in this paper. Section III details our proposed abstraction, discusses the extent to which it answers VNOs' needs. Section IV explains how it can be implemented. Section V concludes the paper.

II. PROBLEM STATEMENT & TERMINOLOGY

The successful delivery of satellite communication services to end-users involves one or many real-life business actors, each

playing one or many roles. Referring to [3], three major roles are distinguished:

- 1) *Satellite Operator (SO)*: it owns the satellite and assumes its operation. It leases satellite capacity at the transponder level (physical layer) to one or several SNOs.
- 2) *Satellite Network Operator (SNO)*: it operates a Broadband Satellite Network (BSN) that provides a multi-beam coverage with forward and return links. The ground segment of the BSN gathers multiple Hubs that are interconnected via a dedicated backbone network with some PoPs (Point of Presence) or gateways to external networks (see Figure 1). Generally, a Hub supports bidirectional traffic on one or many beams. It combines satellite transmit and receive units with a Gateway (GW) to terrestrial networks and a Network Control Centre (NCC). The NCC provides control functions; it typically performs Satellite Terminals (ST) admission and resources control/allocation on the forward and return links. A Network Management Centre (NMC) performs all management functions related to network element's (ST, NCC, GW), i.e. configuration, fault, performance, accounting and security management. Usually, SNOs provide satellite forward and return links to second-tier operators by dividing transponder level bandwidth. The NCC controls this bandwidth sharing. Via the NMC, the SNO provides a management interface to the purchased resources.

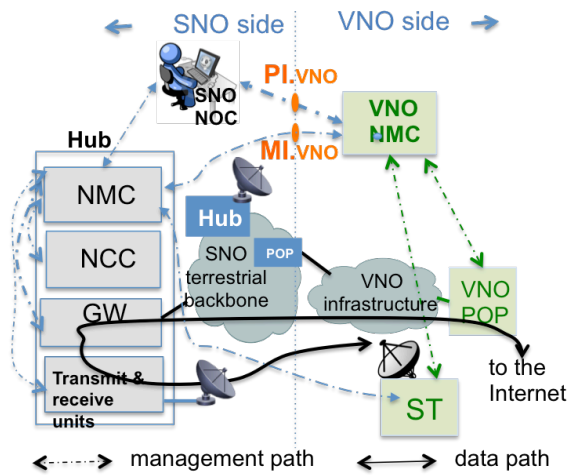


Figure 1 - Typical SNO's Broadband Satellite Network Architecture & SNO to VNO interfaces

- 3) *Satellite Virtual Network Operator* (referred as VNO): VNOs rely on a virtual network infrastructure leased from one or many SNOs complemented with its own terrestrial network infrastructure. As part of its commitments to a customer VNO, an SNO provisions the purchased satellite resources (i.e. connectivity services with the contracted bandwidth or SLA) via provisioning interface (PI.VNO in Figure 1), which is *not automated* and usually *involve humans* from both sides. Moreover, the amount of resources is *predefined* and does not stick to potentially varying VNOs needs.

The SNO also provides a management interface (MI.VNO in Figure 1) to these resources that allows the VNO to implement

its management functions (e.g. monitoring, configuration and accounting) from its own NMC. In fact, the SNO provides to each VNO a partitioned access to its NMC (known as Network Management Domains [4]), which delimits the access of the VNOs to their STs and their resources. In practice, this interface is mainly *focused on monitoring* (reporting status and ST related statistics) with the capability to perform some *very basic configurations* on STs. More advanced management capabilities may be provided under the condition of hardware-based isolation between VNOs, i.e. a VNO get assigned dedicated physical GW and physical transmit/receive units [4]. The MI.VNO interface is generally based on SNMP with *vendor dependent management objects*, complemented with *non-standard vendor specific interfaces*.

III. AN OPENFLOW BASED SWITCH ABSTRACTION OF THE SATELLITE SEGMENT

A. Key idea

The idea is to expose to each VNO a view or an abstraction of the satellite segment that is equivalent to an SDN/Openflow [5] programmable switch that connects VNO's customer STs and the SNO terrestrial backbone network. In other words, a VNO is provided with an interface (which we refer to as I.VNO) to control and manage its satellite segment as if it was programming an Openflow switch (We assume Openflow1.3 or later versions of the protocol). Figure 2 exemplifies this abstraction on a particular VNO. This latter provides communication services to two customers: customer 1 and 2. Customer 1 is using two bidirectional satellite communication services: a first one between its satellite terminals (ST1 and ST2) and another one from ST2 to the Internet. Customer 2 is also purchasing an Internet access for its satellite terminal ST3. With the proposed abstraction, each service is represented as a flow in the switch with typically one flow rule (per direction) installed in the switch's flow table. The flow rule identifies the packets handled by the service (i.e. the *match rule*) and specifies the actions that apply to these packets (i.e. instructions/actions).

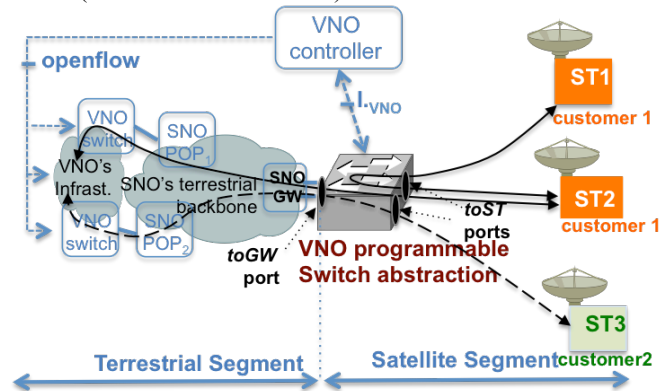


Figure 2 - VNO's view of the satellite segment

Figure 2 highlights the VNO's NMC (or *controller* in the Openflow terminology). From this controller and via the

I.VNO interface, the VNO is able to provision and update on-the-fly satellite communication services as if it was installing, updating or removing flows on an Openflow switch. It is also able to monitor in real-time their status and retrieve some of their statistics. With the flexibility allowed by Openflow in the definition of flows (flexible match rules) and their related actions (instructions/actions), the VNO is able to provide its customers with customized communications, dynamic bandwidth allocations, etc.

Two kinds of switch ports are distinguished: those linked to the STs (*toST* ports) and those linked to the SNO terrestrial GW (*toGW* ports). In the proposed abstraction, the *toST* ports connect all VNO STs that are located on the same satellite beam and using the same carrier frequency band. In the “forward link” direction, the transmission resources of *toST* ports are in line with the resources leased by the VNO on the associated beams. In the “return link” direction, they materialize the traffic submitted by the STs operating on their associated satellite beams. From the instructions and actions programmed in the flow tables, this traffic is scheduled (with potentially traffic drops) and delivered to the GW. Two queues are configured on the ports: A high priority queue that holds the traffic with guaranteed VNO resources (typ. from continuous rate assignment (CRA) and rate-based dynamic capacity (RBDC) service classes) and a best effort queue (typ. From volume-based dynamic capacity (VBDC), and free capacity assignment (FCA) classes).

Figure 2 assumes that the VNO terrestrial infrastructure is Openflow enabled; this is not a commitment, it is simply to stress on one of the potential benefits of the proposed abstraction, which is a unified management between terrestrial and satellite networks.

B. Illustrative example : Application to a commercial satellite service

This section describes the way the proposed abstraction is used by a VNO to request (from the SNO) the provision of a satellite service (with a focus on the satellite segment).

We consider a point-to-point Ethernet service from a satellite terminal ST to a terrestrial customer device connected to a VNO POP. Referring to commercial satellite services, many service attributes may be used to characterize the service: VLAN tagged or untagged, maximum transmission rate (on forward and return link), minimum guaranteed bandwidth, limited monthly traffic volume. These may be applied on per ST or aggregate (a group of STs) basis.

Next, we will start by addressing the case of an 802.1Q Ethernet link with a customer defined VLAN tag, with a maximum transmission rate and a committed (minimum guaranteed) rate. Then, we address the remaining attributes.

1) Service provisioning

With the proposed switch abstraction, provisioning the customer service on the satellite segment resorts to generating Openflow rules that will be delivered to the SNO via the I.VNO interface. One important point is that these rules must clearly identify this particular service against (1) the services used by the other customers from the considered VNO and (2) also from the customers of other VNOs. This is clearly

dependent on the way the SNO and the VNO segregate the traffic belonging to different customers. We will assume a Q-in-Q encapsulation defined in IEEE 802.1ad with the provider VLAN tag (or S-VLAN) used to identify the traffic belonging to a particular customer (in other words, the VNO leases a pool of S-VLAN identifiers and assigns an S-VLAN identifier to each customer). With this assumption, the Openflow “match rule” associated to the service jointly checks for: the ingress-port, S-VLAN-ID, source and destination MAC addresses. It is worth to note that in some very specific scenarios, the service identification may require to check the C-VLAN tag (in addition to the S-VLAN). In this case, two flow-tables are needed to perform the service identification (one for each tag checking). Indeed, standard Openflow rules only allow checking for outermost VLAN tags.

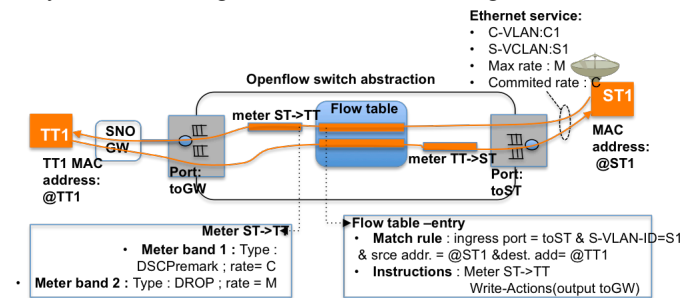


Figure 3 – Application to an Ethernet service

The specification of the maximum and committed transmission rates is achieved by means of an Openflow flow meter, which is triggered on matching packets. Two meter-bands are used: one configured with the committed rate and the second with the maximum rate. If the arrival rate is below the committed rate, none of the meter bands are activated and the packets are forwarded to the output port and buffered on the high priority queue. If the rate is above the maximum rate, the second meter-band is triggered and as a consequence packets are dropped. Finally, if the rate is in the between, the first meter-band marks the packets in such a way that they are buffered in the low priority queue of the output port. Current standard Openflow meter-bands are only capable of dropping packet or remarking the DSCP field by increasing the drop precedence of the packet; there is no explicit way to direct a packet to a specific queue, and, IP traffic is assumed. Different alternatives may address these shortcomings: 1) considering customized meters (with Openflow extensions); or 2) adding a second flow table that dispatches the packets on priority queues based on their marking; or 3) assuming that this dispatching is performed by the output port (through configuration, outside of the scope of Openflow). For simplicity, this latter alternative is considered in Figure 3, which sums up the provisioning of the service.

2) How to limit the service delivery to a monthly traffic volume?

The OpenFlow specification defines a set of counters that apply to different switch elements: flow tables and flow table entries, ports and queues, meters and meter bands, etc. Of

particular interest are: the duration and the byte count of packets that matched a flow table entry, the byte count of packets that crossed a meter and a meter band, etc. By polling periodically these latter counters from its controller, the VNO easily assesses the amount of traffic that was delivered for a particular customer. From this information, it can decide to block the service by removing the associated flow entry.

3) How to establish an aggregate guarantee ?

As cited above, the maximum rate, committed rate or the maximum allowed traffic volume may apply to a single ST or to a set of STs from the same customer. With the proposed abstraction, the aggregate rate can be expressed very easily since a single meter can be associated to multiple flow table entries (and hence to different STs). Also, by following the approach presented in the last section (polling periodically the counters related to appropriate flow entries), a maximum aggregate traffic volume can be enforced.

C. General analysis of Openflow programming capabilities with respect to VNO needs

1) Service provisioning

Satellite service identification is an important aspect that must be captured by the abstraction. In practice, it is usually based on one (or a combination) of the following three fields:

- Layer 2 addresses such as the scheme proposed in [3] relies on a Satellite Virtual Network (SVN) number embedded in the layer 2 addresses, which is used to isolate customer traffic (VNOs are assigned with disjoint pools of SVN numbers)
- Service VLAN Tag as the 802.1Q tag, the Service VLAN tag in IEEE 802.1ad or the Service Instance Tag in IEEE 802.1ah;
- Layer 3 addresses (if disjoint address spaces are assigned to customers).

We argue that the richness and the flexibility of Openflow in the definition of the matching rules clearly goes beyond current VNO needs. Indeed, all relevant protocol fields (layer 2 to layer 4) are supported and can be checked (even partially with bit masking) and combined at will. More fine-grained service identification can be achieved with the ability to rely on transport layer information.

Another important aspect is the expression of the required service behavior (i.e. how the satellite segment handles the packets from the considered service). Generally, it covers the way satellite resources are shared and how the QoS are eventually guaranteed, as well as the desired packet manipulations (packet field modification, addition or removal). We also argue that the “instruction and action sets” are sufficient for that purpose. As shown in the previous section, meter tables can be used to share the forward link and return link resources leased by the VNO between its customers. Bandwidth guarantees can be provided on a service basis or multi-service basis (potentially from different customers). Services that provide a daily or monthly average traffic volume can also be expressed thanks to the byte counts and duration counters associated to flow entries and meters. Since flow table entries can be added/removed/updated and

consulted in real-time, the promise is opening the way to on-demand services as well as “you pay what you use services” that are emerging in terrestrial networks.

2) Service monitoring

VNOs are tied to provide service related status and statistics to their customers. As described in the previous section, Openflow comes with flow-table-related counters that can be polled to derive service status and some high level statistics. But, standard Openflow does not care about low-level signal information (signal strength at STs, etc.). Hence these cannot be made available to the VNO, which is a limitation with respect to VNO needs. Two alternatives may be considered: either extending Openflow, as proposed in the context of optical networks [6], or using accompanying protocols such as SNMP to cope with these low-level metrics.

For some services, OAM functions can also be activated for fault and performance management. Depending on VNO needs, all OAM traffic or part of this traffic (e.g. defect indications, etc.) may be reported to the VNO controller using the Openflow protocol. Referring to Carrier Ethernet OAM (as defined in IEEE 802.1ag), OAM traffic is identified with a specific Ethernet type number. An Openflow entry can be installed in the switch abstraction to direct the desired OAM traffic towards the VNO controller. In case of vendor specific OAM mechanisms (which is quite common in the satellite communication context, even though some vendors are announcing the support of 802.1ag), the SNO NMC can also use the Openflow I.VNO interface to report agreed OAM messages to the controller.

3) ST configuration

Another important aspect for the VNOs is ST configuration, e.g. setting IP configuration, firewalling rules, QoS, OAM mechanisms, satellite frequency band, etc. Clearly, these configurations are out of the scope of our proposed abstraction, which is centered on the services provided to the ST. As a consequence, VNOs will need an additional interface to perform these configurations. Again, different alternatives can be considered: using an accompanying protocol such as SNMP or NETCONF or, more interestingly, deploying an Openflow soft-switch on STs (for firewalling, customer routing and QoS managed by the VNO) with a configuration protocol from the Openflow sphere, namely OF-CONFIG [6] (or OVSDB [8] for specific soft-switches). This latter alternative may open the way to a new kind of services (e.g. on-demand services triggered from STs).

D. Discussion : The abstraction's impact on SNOs

With respect to current practices, the proposed abstraction brings additional flexibility and possibilities to VNOs to express their needs. Also, these needs are submitted to the SNO via high-level synthetic commands (i.e. Openflow rules). Clearly, implementing this abstraction has a cost to the SNO. It is likely that it will lead the SNO: (1) to enhance its orchestration capabilities and automate some of its management procedures in order to map, in real-time, these high-level commands into the appropriate set of actions (configuration, reporting); and (2) to extend the range of configuration actions on some of its network elements (NCC,

GW, etc.) to support the flexibility and possibilities allowed by the abstraction.

Not all the capabilities of an Openflow switch model are useful to all VNOs. For instance, supporting multiple flow-tables and group tables are useful in very specific contexts and hence useless to many VNOs. Hence, exposing a fully functional Openflow switch to all VNOs is questionable. SNOs can provide different Openflow switch abstractions (with different set of supported capabilities) and, hence, monetize their implementation effort in proportion to the supported capabilities.

IV. PROOF-OF-CONCEPT IMPLEMENTATION

Conceptually, the implementation of the switch abstraction requires the integration of an *Openflow to satellite commands adaptor* into the NMC (or the front-end NMC instance devoted to each VNO) of the SNO Hub. The adaptor is in charge of translating (back-and-forth) Openflow commands to satellite hub specific CLI, SNMP (or any other protocol) commands.

For our implementation, we used the OpenSAND platform, which is a tool used to emulate satellite communication systems (mainly DVB-S2/RCS) for performance evaluation and validation of communication network techniques and mechanisms. A typical emulation platform is composed of the following components, which are deployed on separate machines inter-connected via an OpenSAND LAN: the satellite emulator (SE), the OpenSAND Gateway (O-GW) and at least one OpenSAND Satellite Terminal (O-ST). An OpenSAND manager running on the SE machine is used to control the platform (configuration, supervision, etc.); this is achieved with daemon processes (deployed on all above-cited machines) that implement the services to configure and monitor the SE, O-GW and O-ST components.

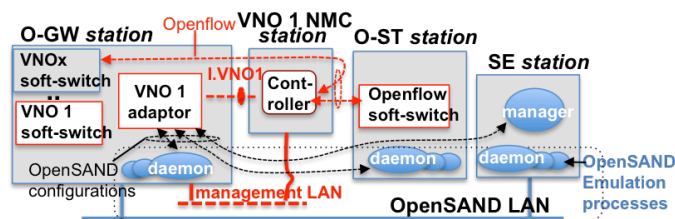


Figure 4 – Implementation on the OpenSAND Platform

Figure 4 describes the general architecture of the implemented prototype. Satellite carriers are exclusively assigned to each VNO. An “Openflow to satellite commands” adaptor (VNO x adaptor in the figure) is deployed in the O-GW for each VNO; it is in charge of translating some Openflow commands in Opensand configuration/monitoring actions by interfacing with the OpenSAND daemons and manager. In fact, the OpenSAND platform does not care about network management and hence do not implement the SNO NMC. This is why the adaptor embraces some of the NMC’s functionalities by interfacing directly with some of the platform components. The adaptor integrates an Openflow

agent [9] in charge of extracting from arriving Openflow packets the corresponding Openflow rules (and vice versa).

Our implementation embeds a software-based Openflow switch in STs; also, in the O-GW, it assigns an Openflow software-switch to each VNO (in place of the classical shared terrestrial GW). These are programmed from the VNO Openflow controller (located on a dedicated machine: VNO1 NMC), which also has access to the VNO adaptor via the Openflow based I.VNO interface. For simplicity, a dedicated management LAN is deployed to support Openflow packets exchanges between the VNO Openflow controllers and the O-GW (who relays the packets to the software switches).

From an experimentation point of view, this prototype was used with two VNOs for satellite service provisioning and dynamic bandwidth allocation. With the presence of the Openflow switch in STs, it was also used to set up on the fly some firewalling rules on STs. For space reasons, they are not presented.

V. CONCLUSION

Extending the capabilities of the interface that SNOs expose to VNOs is a prerequisite to let VNOs enrich their satellite related service catalog, automate their service provisioning procedures, and use their leased resources more efficiently. To this end, we propose in this paper to make this interface evolve to an Openflow-based interface, meaning that a high-level satellite-technology-independent Openflow-based switch abstraction is exposed to VNOs. Some aspects of the proposal were implemented on the Opensand Platform and validated on the use cases presented in the paper. The main perspectives to this work are pursuing the implementation of our prototype to support additional Openflow features.

ACKNOWLEDGMENT

This work is partially funded by and by the French Space Agency (CNES) and by the French National Research Agency (ANR), the French Defense Agency (DGA) under the project ANR DGA ADN (ANR-13-ASTR-0024). The authors are thankful to J. Bernard and A. Thibaud from Viveris Technologies for their contributions to the proof-of-concept implementation on the Opensand platform.

REFERENCES

- [1] H Kim, N. Feamster, “improving network management with software defined networking”. Communications Magazine, IEEE 51 (2), , 2013
- [2] Opensand Platform, available at : <http://opensand.org>
- [3] ETSI TS 101 545-1 V1.1.1, “Digital Video Broadcasting (DVB): Second Generation DVB Interactive Satellite System (DVB-RCS2), Part 1: Overview and System Level specification,” May 2012.
- [4] Deliver World-Class Services without Major Capital expenditures, Hughes, at: <http://global.hughes.com/global-solutions/vno-capabilities>
- [5] Open Networking Foundation, “OpenFlow Switch Specification – version 1.4.0 (Wire Protocol 0x05)”, Oct. 2013
- [6] Open Networking Foundation , “Optical Transport Protocol Extensions,” mar. 2015
- [7] Open Networking Foundation, OpenFlow Management and Configuration Protocol, OF-CONFIG 1.2, 2014
- [8] B. Pfaff and B. Davie, “The open vSwitch database management protocol,” IETF, RFC 7047, Dec. 2013.
- [9] The Revised OpenFlow Library set (ROFL), available at : <http://roflibs.org>