Software Defined Networks and Network Function Virtualisation Testbed within FIRE+

First Wave of Experiments on the SoftFIRE Platform

NFV and SDN Experiments for 5G Networks

March 2018

SoftFIRE

Table of Contents

List of Figures

List of Tables

1 Introduction

During its execution, project SoftFIRE [1] has played an incubator role for SMEs in Network Functions Virtualisation (NFV) [2][3] and Software Defined Networking (SDN) [4] technologies, in the context of 5G mobile networks and applications. A number of proposals were received by the SoftFIRE project prior to running its 1st Wave of Experiments [5], which were carefully evaluated for their feasibility on the platform, as well as their novelty and their contribution to the NFV/SDN communities. Most experiments focused on virtualisation solutions, and proposed interesting and innovative technologies.

The project consortium selected a few experiments for its 1st experimentation wave for two purposes: *(1)* To test the capability of the platform to house multiple simultaneously running experiments, *(2)* To gain experience in supporting experimenters using the best possible methodologies and tools, whilst ensuring that experiments run smoothly and reach a successful conclusion. This strategy proved to be extremely useful for the project; with the provided feedback from experimenters, the project later developed more powerful experimentation environment for its later waves of experiments, i.e. Wave 2 [6] and Wave 3 [7].

In this white paper, selected experiments that were successfully deployed on the SoftFIRE platform during its 1st Wave of Experiments are briefly presented. In doing so, the intention is to present what has been achieved by experimenters on the platform, and the types of NFV/SDN experiments that were executed on the platform. These selected experiments are:

- *SoftFire OffLoadIng (SOLId)*, by GridNet, Greece
- *Network Functions Virtualisation at the Edge (NFV@Edge)*, by Politecnico di Torino, Italy
- *Expose*, by National Centre of Scientific Research, Greece
- *SecGENE* (SEmantics driven Code GENEration for 5G networking experimentation), by University of Niš, Faculty of Electronic Engineering, Serbia
- *MARS,* by Level7 S.r.l.u, Italy

The white paper presents summaries of the architecture, experimentation, and contributions of a selected set of experiments.

2 SOLId

The company GridNet [8] performed the SoftFire OffLoadIng (SOLId) experiment during the 1st Wave of Experiments of the SoftFIRE project. The focus of this experiment was on providing an intelligent solution to perform traffic offloading for user data traffic, when multiple access networks are available.

This particular goal has been the aim of a number of solutions so far, such as Multi-path TCP (MP-TCP) [9], Selected IP Traffic Offload (SIPTO) [10], and LTE-WLAN Aggregation (LWA) [11]. The drawback of these schemes is however that they either require significant modifications, or they are not suitable for the most recent releases of mobile core networks. In particular, MC-TCP is generally not allowed by network operators as it allows or is required to pass through firewalls, which poses security risks. SIPTO has a lack of support for Lawful Intercept (LI) and accounting. LWA is a promising approach for multi-access control; however this approach may take several years to become de-facto or it may practically not be adopted by operators, as it requires significant upgrade of Wi-Fi and LTE base stations, as the technique requires modification of the Medium Access Control (MAC) layer.

2.1 Experiment description

The solution offered by GridNet in their experiment SOLId has the objective to build an offloading framework for heterogeneous networks (LTE/LTE-A & Wi-Fi) on an NFV platform, which is driven by end-user perceived Quality of Service (QoS). The experiment used a new network entity, called an *SOLId Framework,* which can make intelligent routing decisions to optimize the usage of available access networks. The experiment in particular had WiFi and LTE-A access, yet the solution is generic, and can be applied to different access networks. The SOLId Framework solution is a modified version of Open Virtual Switch (OVS) [12].

To realise this experiment, project SoftFIRE provided a dedicated user plane network slice that has the functionality for data connectivity over a mobile core network; i.e. the user plane of a mobile core network. Such slices, called a User Plane Node (UPN) are provided to each experimenter that required mobile network data connectivity; hence different experimenters had different slices.

2.2 Experiment architecture

The SOLId Framework solution was connected to the SGi interface of the mobile core, which is now referred to as the N6 interface in the 5G System Architecture [13]. The SOLId Framework was also connected to a WiFi Mesh Gateway solution. The data traffic from the UPN network slice and from the WiFi gateway both passed through the SOLId Framework, making it possible to monitor the traffic through both.

The complete experiment architecture is illustrated in Figure 1. In this figure, the UPN consists of a Cluster Controller (CC) and a Cluster Member (CM), which are the context-aware networking components [14] of the novel core network provided to experimenters by SoftFIRE. The component PPE is the Packet Processing Entity, which is a combination of the user plane functionalities of SGW and PGW. For more details on the core network slices provided by SoftFIRE, readers can refer to the SoftFIRE White Paper on Network Slicing [15].

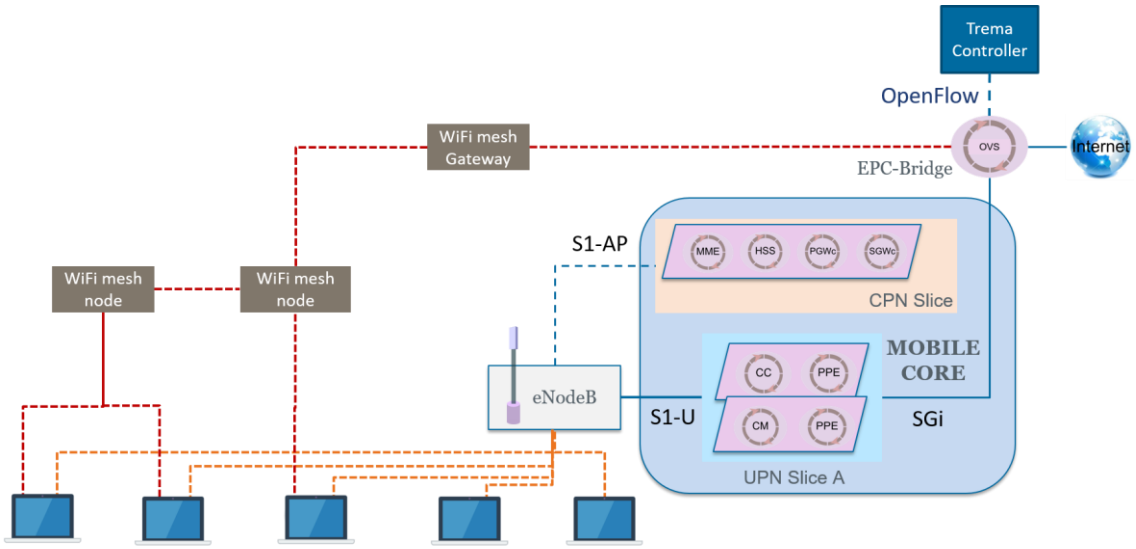

Figure 1. GridNet [8]'s SOLId experiment architecture on the SoftFIRE testbed. The solution SOLId Framework is tagged as "EPC-Bridge" architecturally.

The SOLId Framework was programmed using the Trema [16] controller. The controller and the SOLId Framework were deployed as a single virtual network function (VNF) on the same virtual machine. The SOLId Framework programmed and monitored the UE traffic to the 5G core. A dedicated network function estimated the network performance of the LTE and Wi-Fi networks. Based on the load on the UPN, a decision was made to inform some UEs to switch to the WiFi network.

To make reasonable offloading decisions as to which user equipment (UE) should change network, UE Service Level Agreements (SLA) were taken into account. The SLAs indicated the required downlink throughput.

3 NFV @ Edge

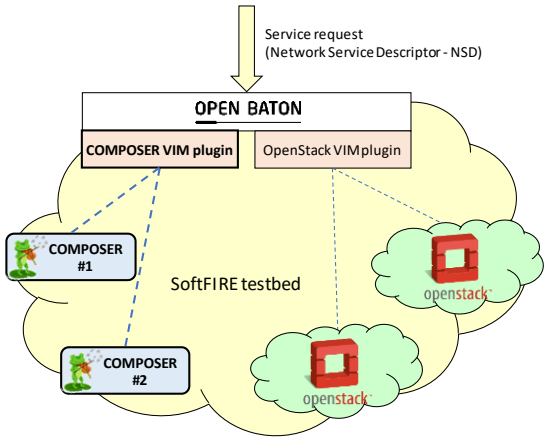

Figure 2. Architecture of the NFV@EDGE experiment.

Current NFV orchestrators such as Open Baton [17] can control multiple and heterogeneous infrastructure domains through the Virtual Infrastructure Managers (VIM), according to the ETSI terminology [18]. However, data centres represent the most common physical infrastructure in use nowadays, hence Open Baton defines a southbound plug-in application programming interface (API) that mimics the most common actions and commands in cloud controllers such as OpenStack [19]. In a nutshell, the VIM plugin provided by Politecnico di Torino in the NFV@Edge experiment implements the basic actions that are needed by Open Baton to control the underlying infrastructure (e.g., create/delete

network, start/stop a virtual machine), translating high-level commands into the ones supported by the involved domain.

As shown in Figure 2, NFV@EDGE requires the extension of Open Baton with a new VIM plug-in that can control the COMPOSER platform [20]; however, due to the nature of the southbound API, the new COMPOSER VIM plugin has to emulate many cloud controller related concepts, such as user/group permissions, availability zones, etc., even if they are not present in a small and lightweight compute node such as a domestic Customer Premises Equipment (CPE). This enables COMPOSER to appear just like another OpenStack domain, hence hiding its different internal architecture.

This experiment extends the ETSI MANO-compliant Open Baton, which is one of the market leading NFV orchestrators. The experiment provides the capability to control resource-constrained devices, such as home CPEs, which are very common at the edge of the network. This plugin can control the COMPOSER platform. Such capability is one of the missing pieces towards an end-to-end programmable network, covering both the edge of the network and its backhaul and reaching to the telco cloud data centre. This requires an overarching orchestrator to set up a complex chain of services encompassing network functions running either at the edge of the network or in the cloud. The solution hence combines the benefits of edge-based services (e.g., reduced latency, no last-mile bandwidth bottleneck, better reliability) with the ones of cloud-based services (e.g., scalability, efficiency, and economy of scale).

3.1 COMPOSER: the COMPact SErvice Router

COMPOSER is a compact service orchestration platform that offers the possibility to compose network functions (NFs) in arbitrary service graphs and deliver virtualised services. In a nutshell, COMPOSER is responsible for deploying NFs (and, in general, managing their lifecycle), and creating traffic steering connections between them.

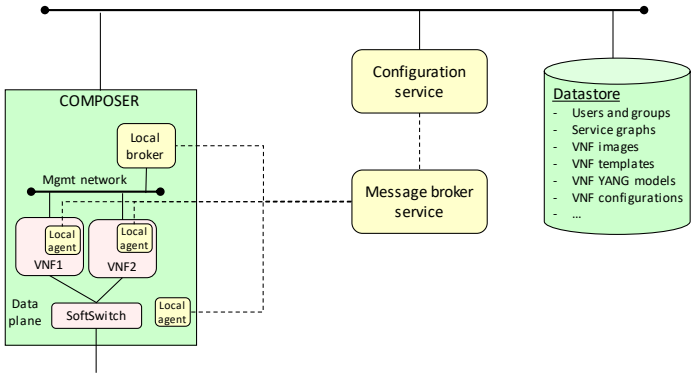

Figure 3. Architecture of COMPOSER [20] for VNF configuration.

The overall architecture of COMPOSER, as depicted in , is composed of four main modules: First, COMPOSER analyzes incoming service requests and instantiates the required VNFs and sets traffic steering primitives to connect them. Second, the configuration service is in charge of both runtime configuration and exporting the run-time state of VNFs. In a nutshell, it provides a REST API [21] that can be used to either set or read a configuration to any controlled object (e.g., VNF). Third, a flexible datastore keeps different kinds of data, such as VNF images and their associated templates, YANG data models [22] of any supported object, user and group permissions, and more. Finally, a message bus connects

the configuration service with all the configuration agents that are running on the different objects.

The delivery of management and control information requires the availability of a (virtual) network infrastructure. The suggested architecture, however, includes a dedicated network for control and management purposes, where only the control/management components are attached.

Given the structure of COMPOSER and the necessity for companion ancillary services, e.g. for configuration purposes, the integration of COMPOSER with OpenBaton requires the setup of multiple components, as shown in Figure 4. In this picture, the new components have been depicted in green, while existing ones (i.e., the one already belonging to the SoftFIRE architecture) are depicted in yellow. An example of a possible service graph is depicted as well, with the corresponding mapping in the infrastructure. The service graph, in fact, has a very different implementation at the infrastructure layer, with three additional VNFs (depicted in light blue) that are required to support the service itself (e.g., local area network (LAN) emulation, router or network address translation (NAT) toward the Internet), and other three VNFs dedicated to the management and configuration tasks (depicted in pink), while apparently the "requested" service includes only two VNFs (orange components).

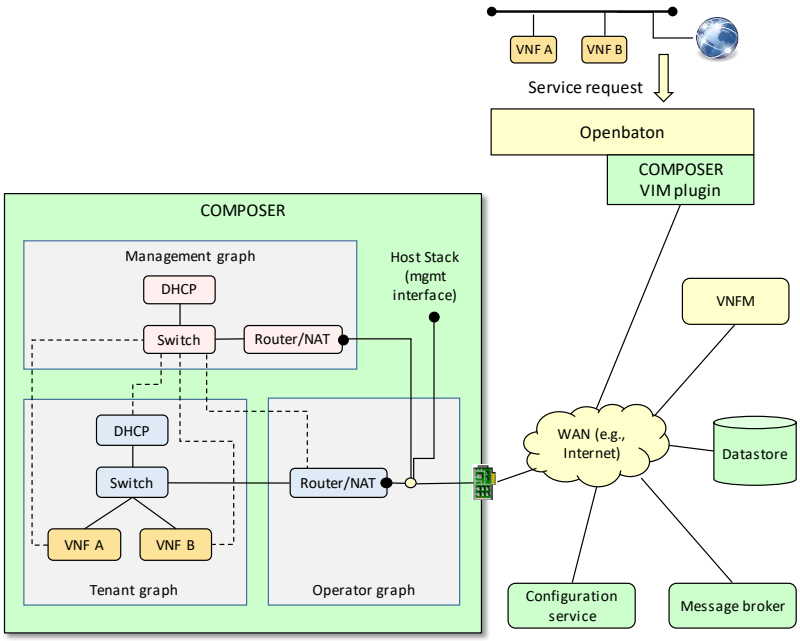

Figure 4. Integration of COMPOSER in the OpenBaton architecture.

In this particular mapping example, COMPOSER is able to accept services through a single NIC (mimicking the typical case of residential gateways, which feature a single network interface card (NIC) toward the operator network); however, at the time of writing, three public IP addresses are required, as depicted by the black spots in Figure 4. For instance, the first is used to connect the requested service to Internet, the second enables the reachability of the services running in the management network (required to configure the VNFs), while the third is used for the general management of COMPOSER itself, e.g., to receive the service graph and to provide a management / control interface to the system.

Figure 5 presents the network creation procedure instantiated by Open Baton. The VIM plugin for COMPOSER first queries COMPOSER to retrieve both the tenant and the operator graphs. Then, it modifies them locally by adding a new switch in the tenant graph and creating a link between it and the router of the operator graph. After communicating such updates to the

COMPOSER, the plugin can finally return to Open Baton an object representing the network just instantiated.

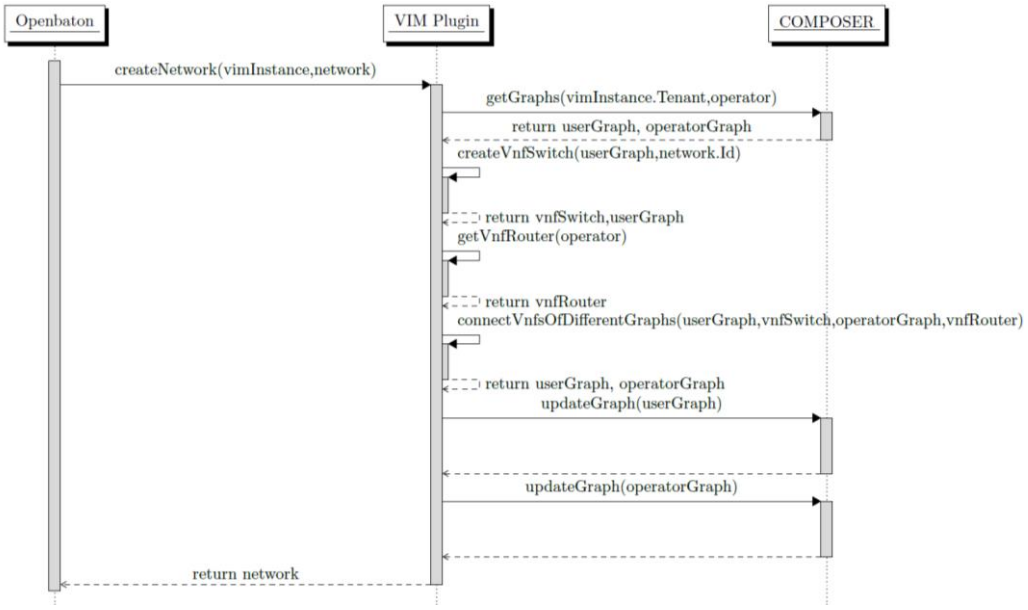

Figure 5. Network creation workflow of the COMPOSER plugin for Open Baton.

3.2 Experimental validation on the SoftFIRE platform

The SoftFIRE platform provided the facilities to perform experimental validation: the OpenBaton orchestrator (in Berlin), an OpenStack remote data centre (also in Berlin), and a physical network infrastructure. In addition, COMPOSER was installed in the Politecnico di Torino lab and connected through a virtual private network (VPN) (GRE over IPsec) to the SoftFIRE testbed.

The validation scenario involves querying OpenBaton to setup a service graph that includes the typical services required to provide domestic Internet access (i.e., a LAN switch, DHCP, storage server, NAT, firewall). Different requests are issued, triggering OpenBaton to deploy the VNFs in different portions of the infrastructure (edge vs. cloud) and measuring the difference in terms of throughput when contacting the storage service in three different operating conditions. This allows to assess the advantages of the NFV@EDGE approach, i.e. how the user experience improves when the service gets closer to the end user. Benchmarking scenarios were:

- *(i)* When the requested service is entirely deployed on OpenStack, hence using COMPOSER as a (dumb) vCPE,
- *(ii)* COMPOSER provides only storage,
- *(iii)* Complete deployment in COMPOSER.

The *iperf* tool was used to simulate a local user connected to the CPE, that establishes a massive data transfer towards the storage server. Average throughput results over 10 experiments in each scenario were found to be as presented in Table 1.

Table 1. Data throughput from the storage server.

Experiment Setting	Throughput (Mbps)
All services on OpenStack (remote)	22.4
Storage service, DHCP and LAN switch on COMPOSER (local), the rest on OpenStack (remote)	206
All services on COMPOSER (local)	112

The tests confirm the advantage of the NFV@EDGE approach, as migrating at least some selected VNFs near the end-user improves its network experience. An improvement of approximately one order of magnitude can be observed when the storage service is local to the user (hence, both located in Torino, Italy) compared to the case in which the storage server is moved to Berlin, Germany.

This initial validation of the "NFV services at the edge of the network" confirms the advantages of delivering edge services, albeit with some limitations. In fact, the experiments showed that, given the possible limited resources available on residential gateways, it necessary to carefully decide which service must be executed on a gateway, and which ones must be moved to the cloud. In fact, performance measurements confirm that the performance are much better when only a *portion* of the services are running on COMPOSER (the others are offloaded in the cloud), compared to the case in which all services are executed on that platform. This result suggest that a careful optimization algorithm has to be envisioned in order to handle edge services, which takes in high consideration also the load on the computing devices under consideration. This highlights a possible future work in the SoftFIRE testbed, involving the definition and the analysis of distributed scheduling algorithms that can decide where to schedule the different VNFs in order to optimize the service.

4 Expose

The combination of satellite and terrestrial networks to form a single/integrated telecoms network is an integral part of future 5G networks. Deployment of such integrated systems will be observed around the globe within the next years, and there is a pressing need to test and measure the integration and performance of these systems. It is also expected that NFV and SDN will have important impact on future satellite communication systems. High cost, low resource availability, and conservative architectures that predominate today in the satellite landscape certainly constitute major obstacles to cross.

The aim of the Expose experiment is to adapt and integrate an open source emulator into the virtualization environment provided by SoftFIRE, and in doing so, to test traffic offloading scenarios between a terrestrial link and a satellite link. By enabling flexible integration of satellite and terrestrial networks, the experiment provides an architectural extension which is aligned with 5G directions.

4.1 Architecture

The experiment involves three software components, based on the Open-source OpenSAND [23] software, namely a Satellite Terminal (ST), a Satellite Emulator (SE), and a DVB-S2/RCS gateway emulator which enables communications between a Satellite network and a terrestrial network. The architecture is illustrated in Figure 6. The ST emulates a DVB-S2/RCS terminal with uplink and downlink communications, and supports various encapsulation schemes. The SE unit is a satellite emulator (transparent or regenerative satellite) and also can support various encapsulation schemes, depending on the payload type, link types (delay, signal distortion, and link budget considerations), and the type of plug-ins used. Finally, the gateway implements encapsulation and decapsulation functions along with many required multiplexing features.

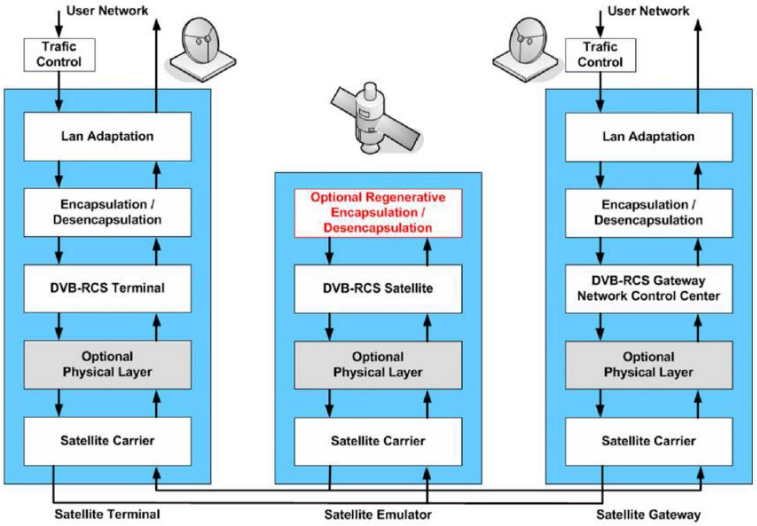

Figure 6. The satellite emulation system deployed on SoftFIRE by Expose.

This experiment executed experiments for satellite communication systems, integrated as VNFs in the SoftFIRE testbed. Figure 7 illustrates the deployed three VNFs on the SoftFIRE platform, as well as the end-to-end experiment setup between an application server and a user equipment terminal. The SoftFIRE virtualisation platform provides two communication paths between these end-points, where the chain of VNFs emulates satellite communications, whereas the

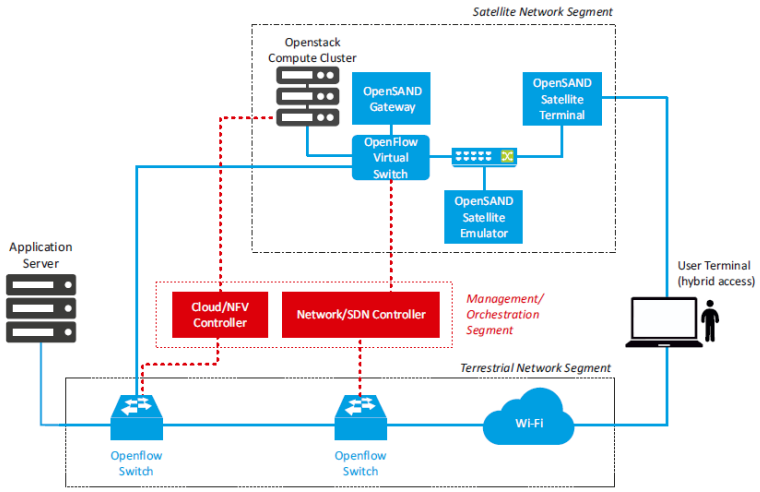

Figure 7. Deployment architecture of the Expose satellite emulation system on the SoftFIRE platform

path through the WiFi and OpenFlow switches emulates a terrestrial link. The experiment scenarios consider a single Satellite Network Operator (SNO) providing a GEO satellite communication services via a transparent satellite.

4.2 Experiment scenarios

The experiment involves two scenarios on this setup:

1) *Hybrid distribution of digital media, offered as a service.* This scenario considers distribution of multimedia content over both satellite and terrestrial paths, enabling high bit-rate and high-quality 2D/3D broadcast content, coupled with interactive personalized services, as illustrated in Figure 8. The ingress and egress points of the two segments (i.e. the satellite and the terrestrial) are connected to SDN compatible Open Virtual Switches (OVS), which are under the control of the OpenDaylight SDN controller. This enables load balancing between the two network paths. The Media Service Providers (MSP) can be offered with management and control capabilities, which enables the MSP to develop own network control logic in order to dynamically configure the network at runtime, allocate resources, and also influence routing/forwarding decisions as desired, i.e. divert streams from the terrestrial to the satellite channel and vice versa on-the-fly or adjust the load balancing between the two networks. In the experiments, a video service is initially delivered over the terrestrial network. Once the link quality degrades due to the background traffic, SDN rules are applied and video quality is reinstated. Structural Similarity Index Measure (SSIM) has been improved from 0.21 to 0.85.

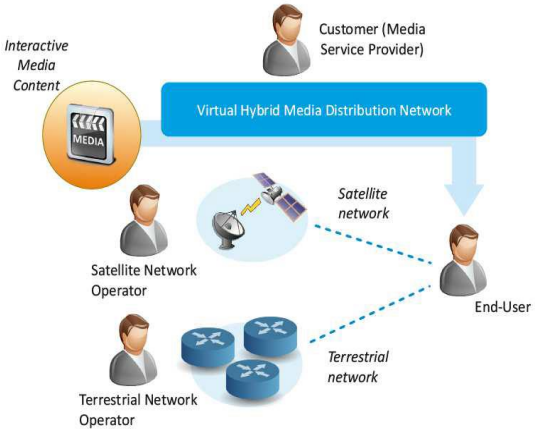

Figure 8. Hybrid distribution of media as a service.

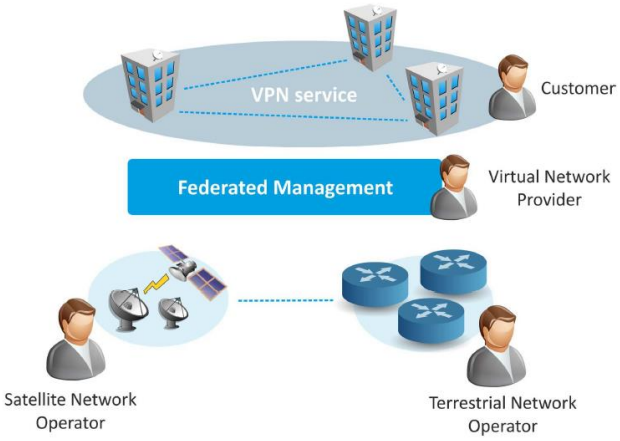

Figure 9. Federated satellite/terrestrial VPN as-a-service scenario.

2) *Federated terrestrial and satellite networks over VPN.* This scenario covers the use cases in which a customer is an enterprise or institution with several distributed Points of Presence (PoPs), which need to be

interconnected via VPN. Satellite links are considered for both coverage extension and reliability improvement. When some of these PoPs are outside the terrestrial network coverage satellite communications enables integration. Reliability improvement is an inherit outcome of the extension of terrestrial networks via satellite links, in cases where terrestrial coverage is inadequate for those PoPs where both types of routes are available, which is especially useful for mission-critical applications. The scenario includes three VNFs that implement open virtual switches, one VNF of a video server, one VNF of a virtual transcoder, and finally one VNF of an end-user/client of the media service. The VPN tunnel is diverted from the terrestrial to the satellite channel and vice versa on-the-fly to adjust the load balancing between the two networks. A VPN tunnel is established over the terrestrial network. Migration event to the satellite link is triggered, and an SDN rule is applied. This steers the VPN tunnel to the satellite link.

3) *Satellite edge processing.* This scenario involves deployment of instances of specific services of the terrestrial network, such as LTE eNodeB components, as VNFs on the satellite access segment. The topic is "*news aggregation*", where user-generated content (e.g. video) is transmitted over a satellite towards a news aggregator server. In the presence of multiple users requesting multimedia content, due to bandwidth limitations on the satellite link, network congestion occurs, leading to quality degradation or even service interruption. Edge computing capabilities, which are deployed as a VNF instantiated at the SDN/NFV-enabled satellite terminal, can improve bandwidth utilization. To facilitate video transmission dynamically, the experimenter monitored quality degradation instances, and instantiated a transcoder as a VNF at the SDN/NFV-enabled terminal. SDN traffic rules transparently steer media traffic flows through the VNF-based transcoder to be forwarded over the satellite to the news aggregator.

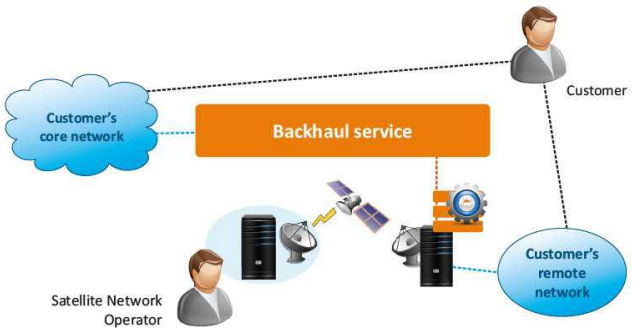

Figure 10. Dynamic backhauling with edge processing scenario.

SDN and NFV, the two main concepts investigated in this work, have different implications for satellite communications. SDN is mainly intended to be implemented at the border of a satellite communication system, possibly without any impact on its core service in mid-term applications. SDN operations still need to be integrated with satellite communications NMS and OSS/BSS. NFV could have shorter-term applications, related to the operations and management of specific features, wherever they are implemented. With the advent of projects aiming at developing low-cost LEO constellation composed of many small satellites, the opportunities to develop and operate on-board SDN-compatible routers may become a reality.

5 SecGENE

This experiment, SecGENE (SEmantics driven Code GENEration for 5G networking experimentation), was conducted by researchers from the University of Niš, Faculty of Electronic Engineering and provided an automatic code generation platform [25] based on semantic experiment descriptions that can be easily generated for experiments on testbeds. By adopting domain and system ontologies, a formal representation of the semantics of an experiment is derived. It is envisioned that such technology can reduce the time-to-experiment on a new online testbed platform, reducing the initial time when experimenters attempt to familiarise themselves with a new platform.

5.1 Code generation

The provided simple an easy to learn environment enables experimenters to describe their experiments quickly and efficiently. The experiment involved a SecGENE server, which automatically generated code that then was testbed on the SoftFIRE platform.

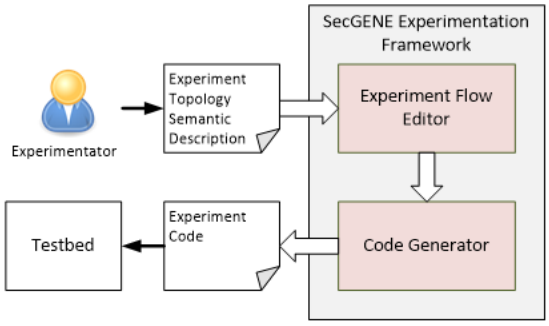

Figure 11. Experiment code generation using SEFE.

Experimentation using in SecGENE is illustrated in Figure 11. First, experimenters define an experiment topology using the JFed [26] tool. The generated RSpec [27] file that describes the topology is then processed by FitEagle [28], which converts it to a semantic description. This is then imported into the SecGENE Experiment Flow Editor (SEFE), which has an intuitive GUI, and contains semantic knowledge of the networking domain, based on *Network Sensing* and *Network Capability* ontologies. A snapshot of an experiment on the GUI of SEFE is presented in Figure 12.

5.2 Experiment flow definition

SEFE contains experiment components that user can use to construct an experiment by performing simple drag/drop and connect actions on these components, as shown in Figure 12. Each component describes a specific task in the experiment flow, where the components are:

Metric – implements measurement action on the network resources. When using this component, the user can select a metric for the measurement that will be performed.
Transfer – implements data transfer from a single node over the given link. When using this component, the user selects one of the existing topology nodes and one interface that exists on the selected node.
Select – implements selection of one value in a set of multiple values.
Set – implements mechanism for sending one value to the given node.
Start/End – are responsible for annotating start and end of the experiment flow.

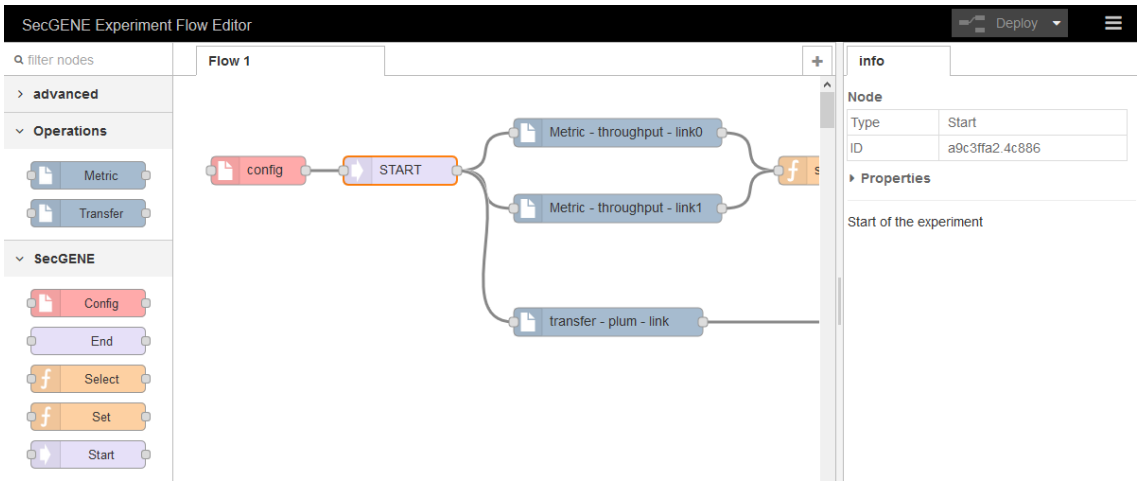

Figure 12. Experiment Flow Editor in SecGENE.

For each flow component in the flow, SEFE automatically generates code for the targeted testbed execution environment. During this process component's high level conceptualization parameters are mapped to the lover levels that are required by execution environment. Commands that are generated follows experiment flow, e.g. they are set to be executed on the proper nodes and in the desired order.

Shell script and OMF execution environments are both supported by the SEFE. In the case of shell script execution environment (such are VNF capable testbeds) output is provided in the form of the shell scripts files for each of the network nodes. Scripts can be directly included in the corresponding virtual network function descriptor (VNFD) packages, in which case they will be executed when VNFD is deployed. For object module format (OMF) capable testbeds, a single file is generated that contains appropriate OMF Experiment Description Language (OEDL) code. The experiment is then executed by running that script file on the experiment coordinator node.

In virtualisation testbeds, such as SoftFIRE, SEFE generated shell scripts for experiment execution of network nodes, which can be included in VNFDs, and executed at VNF deployment time.

6 MARS

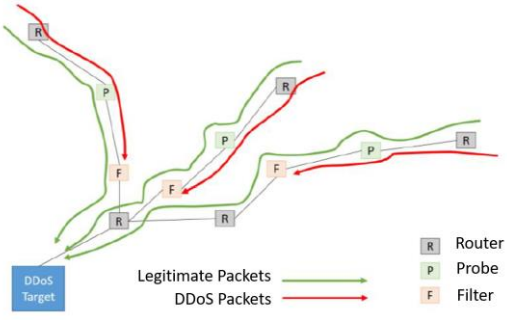

Figure 13. MARS solution against DDoS attacks.

This experiment tested a solution to detect distributed denial of service (DDoS) attacks that would reduce the quality of service from legitimate data sources or impede it completely. To overcome this, Level7 designed a simple but effective solution which includes data probes on data paths. The concept is illustrated in Figure 13.

DDoS target represents a server or a resource on the network that could be attacked from various sources. The sources send TCP/SYN

packets in order to saturate the resources on the target. The Probes send periodic reports to the Controller that has a general overview of the network status. The Filter (also called Enforcer) has no real intelligence but it applies the policies that are sent from the Controller to the Enforcers in the network.

The experiment used two sites of the SoftFIRE platform, where one of the sites housed a legitimate data source and a DDoS attacker as VNFs. The site also had instantiated probe VNFs. The second site ran a VNF for a target node, and also had the SDN controller in place. Performance measurements were made to demonstrate the effectiveness of this DDoS prevention scheme, including two scenarios: No DDoS prevention in place, DDoS prevention with Probes and Enforcers activated. Results showed that almost all (99.468%) of the DDoS packets were blocked, whereas no legitimate traffic was blocked.

The main finding of the experiment was that distributing monitoring probes is an effective way to detect and prevent DDoS attacks. Level7 later participated at SoftFIRE's Innovation Hackathon event [29], improving some concepts from micro DDoS techniques and implementing a brand new SDN based solution.

7 Conclusions

Various experiments were executed on the federated virtualisation testbed provided by SoftFIRE. The selected set of experiments in this white paper have a variety of themes, i.e. WiFi-LTE traffic off-loading, satellite-LTE traffic off-loading, orchestration of edge devices, and automatic code generation for experimentation on virtualisation platforms.

The SOLId experiment provided an EPC-Bridge solution to realise dynamic WiFi-LTE offloading mechanism, which is triggered by the load on LTE links. With the use of traffic monitoring and SDN, it is shown that non-3GPP technologies can effectively help reduce the load on mobile core networks. This is a supporting technology for 5G mobile networks, which are expected to receive an large aggregated traffic from a much larger number of devices than LTE networks receive today.

The experiment NFV@Edge provides a virtual infrastructure manager (VIM) plugin for the ETSI MANO orchestrator Open Baton, so that edge devices, especially customer premises equipment (CPE), can be orchestrated. This gives OpenBaton the capability to control resource-constrained devices. As a result, the solution combines the benefits of edge-based services (e.g., reduced latency, no last-mile bandwidth bottleneck, better reliability) with the ones of cloud-based services (e.g., scalability, efficiency, and economy of scale).

Another experiment, called EXPOSE, adapted and integrate an open source emulator OpenSAND into the virtualization environment provided by SoftFIRE, and tested traffic offloading scenarios between a terrestrial link and a satellite link for user terminals with capability to support both. SDN-based traffic steering was performed on Open Virtual Switches (OVS) to redirect traffic from the terrestrial to the satellite link, in three different scenarios.

The experiment SecGENE provided an automatic code generation platform based on semantic experiment descriptions that can be easily generated for experiments on testbeds. This platform is expected to reduce the time-to-experiment on a new online testbed platform, reducing the initial time when experimenters attempt to familiarise themselves with a new experimentation platform.

Finally, the MARS experiment showed that having distributed probes that monitor data flows is an effective method to prevent DDoS attacks, and built an SDN-based solution to dynamically filter malicious packet flows.

These projects demonstrated the power of NFV and SDN in supporting near-future 5G technologies, and the SMEs and academic researchers involved in these experiments have benefited from the availability of a virtualisation testbed for performance and functionality testing, before investments can be made in such technologies by the industry.

Bibliography

[1] EU SoftFIRE project, https://www.softfire.eu/

[2] "Network Function Virtualisation: State-of-the-Art and Research Challenges", Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, Raouf Boutaba, *IEEE Communications Surveys & Tutorials*, vol 18, no 1, 236-262, September 2015,

[3] "Network Functions Virtualisation— Introductory White Paper", ETSI, 22 October 2012, retrieved 20 June 2013.

[4] "Software-Defined Networking: A Comprehensive Survey", Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, *Proceedings of the IEEE*, vol 103, no 1, pp 14-76, January 2015,

[5] 1st Wave of Experiments on the SoftFIRE platform, https://www.softfire.eu/open-calls/first-open-call/

[6] 2nd Wave of Experiments on the SoftFIRE platform, https://www.softfire.eu/open-calls/second-open-call/

[7] 3rd Wave of Experiments on the SoftFIRE platform, https://www.softfire.eu/open-calls/third-open-call/

[8] GridNet, http://gridnet.gr/

[9] Multi-Path TCP (MP-TCP), https://www.multipath-tcp.org/

[10] Selected-IP Traffic Offload (SIPTO), http://blog.3g4g.co.uk/2010/09/selected-ip-traffic-offload-sipto.html

[11] LTE HetNet (LTE-H) a.k.a. LTE WiFi Link Aggregation, http://blog.3g4g.co.uk/2015/04/lte-hetnet-lte-h-aka-lte-wi-fi-link.html

[12] Open Virtual Switch (OVS), http://openvswitch.org/

[13] TS 23.501, System Architecture for the 5G System, 3GPP, http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/

[14] "Flat Distributed Cloud (FDC) architecture", 5G Innovation Centre, https://www.surrey.ac.uk/sites/default/files/5G-Network-Architecture-Whitepaper-(Jan-2016).pdf

[15] *"5G Mobile Core Network Slicing on an Orchestrated and Virtualised Infrastructure"*, SoftFIRE White Paper, available at: https://www.softfire.eu/wp-content/uploads/SoftFIRE-White-Paper-on-Network-Slicing.pdf

[16] Trema, https://trema.github.io/trema/

[17] OpenBaton, https://openbaton.github.io/

[18] ETSI MANO specification, http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[19] OpenStack open source cloud computing software, https://www.openstack.org/

[20] Ivano Cerrato, Fulvio Risso, Roberto Bonafiglia, Kostas Pentikousis, Gergely Pongráczc, Hagen Woesner. "COMPOSER: A Compact Open-source Service Platform", Under Review. Available online at http://fulvio.frisso.net/files/18-Composer.pdf

[21] Representational State Transfer (REST), http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
[22] YANG – A data modelling language for the network configuration protocol (NETCONF), IETF RFC 6020, https://tools.ietf.org/html/rfc6020
[23] OpenSAND, http://opensand.org/content/home.php
[24] OpenDaylight, https://www.opendaylight.org/
[25] Valentina Nejkovic, Milorad Tosic, Filip Jelenkovic, Nenad Milosevic, Zorica Nikolic, "The SecGENE ontologies framework ", in Proc. XVI International Scientific – Professional Symposium INFOTEH-JAHORINA 2017, Jahorina, 22-24 March, 2017
[26] JFed, https://jfed.ilabt.imec.be/
[27] RSpec, http://rspec.info/
[28] FitEagle, http://fiteagle.github.io/
[29] SoftFIRE Innovation Hackathon, https://www.softfire.eu/events/innovation-hackathon/, April 2018.

List of Acronyms and Abbreviations

Acronym	Meaning
5G	Fifth Generation Mobile Network
API	Application Programming Interface
BSS	Base Station Subsystem
CC	Cluster Controller
CM	Cluster Member
CPE	Customer Premises Equipment
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DVB-S	Digital Video Broadcasting - Satellite
DVB-RTS	Digital Video Broadcasting Return Channel via Satellite
EPS	Evolved Packet Core
ETSI	European Telecommunications Standard Institute
GEO	Geosynchronous Equatorial Orbit
GRE	Generic Routing Encapsulation
HSS	Home Subscriber Server
LAN	Local Area Network
LEO	Low-Earth-Orbit
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LWA	LTE-WLAN Aggregation
MAC	Medium Access Control
MANO	Management and Orchestration
MP-TCP	MultiPath TCP
MME	Mobility Management Entity
MSP	Media Service Provider
NIC	Network Interface Card
NAT	Network Address Translation
NFV	Network Function Virtualisation
NMS	Network Management System

PoP	Point of Presence
ODL	OpenDaylight
OEDL	OMF Experiment Description Language
OMF	Object Module Format
OSS	Operations Support System
OVS	Open Virtual Switch
PGW	Packet data network Gateway
PGWc	Packet data network Gateway control
PoP	Point of Presence
PPE	Packet Processing Entity
QoS	Quality of Service
REST	Representational State Transfer
SDN	Software Defined Network
SE	Satellite Emulator
SLA	Service Level Agreement
SGW	Serving Gateway
SGWc	Serving Gateway control
SIPTO	Selected IP Traffic Offload
SNO	Satellite Network Operator
SSIM	Structural SIMilarity
ST	Satellite Terminal
TCP	Transport Control Protocol
UE	User Equipment
UPN	User Plane Node
vCPE	Virtual CPE
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VPN	Virtual Private Network

SoftFIRE

Disclaimer

SoftFIRE has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 687860.