Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 361–368

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Future Networks and Communications (FNC 2017)

S6: a Smart, Social and SDN-based Surveillance System for Smart-cities

Corrado Rametta*, Gabriele Baldoni, Alfio Lombardo,

Sergio Micalizzi and Alessandro Vassallo

DIEEI – University of Catania, V. le Doria 6, Catania 95125, Italy (name.surname@dieei.unict.it)

Abstract

In the last few years, Software Defined Networks (SDN) and Network Functions Virtualization (NFV) have been introduced in the Internet as a new way to design, deploy and manage networking services. Working together, they are able to consolidate and deliver the networking components using standard IT virtualization technologies not only on high-volume servers, but also in end user premises, Telco operator edge and access nodes thus allowing the emergence of new services.
In this context, this paper presents a smart video surveillance platform designed to exploit the facilities offered by full SDN-NFV networks. This platform is based on free and open source software running on Provider Equipment (PE), so allowing function deployment simplification and management cost reduction.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Software Defined Networking; Network Functions Virtualization; Fog/Edge Computing; Live Video Broadcasting.

* Corresponding author. Tel.: +39 347 8322202.
E-mail address: crametta@dieei.unict.it

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.078

1. Introduction

The new paradigms of Software Defined Networks (SDN)[1] and Network Functions Virtualization (NFV)[2,3] have recently redefined the vision of the Internet: the power of SDN is based on its characteristic of decoupling control and data planes, moving the network intelligence to a centralized controller. On the other hand, the emerging technology of NFV introduces an important change in the network service provisioning approach, leveraging on standard IT virtualization technology to consolidate many network equipment facilities and application services onto standard servers that could be located in data centers, network nodes and even in the end user premises[4].

Therefore, a joint application of SDN/NFV framework allows a Telco Operator to run network and application functions within virtual machines, by using NFV, and to dynamically steer traffic flows through the requested virtual network functions (VNFs) thanks to the underlying SDN network. By so doing, Telco Operators are migrating their networks from a completely hardware platform made up of hardware middle boxes[5] or software routers[6,7], towards a more flexible softwarized network where VNFs can be instantiated and migrated according to specific policies aimed at optimizing energy efficiency, costs and performance[8,9], and taking into account congestion of parts of the network, or even faults, at run-time.

Moving from this technical background, this paper proposes an SDN/NFV-based video surveillance platform allowing to easily deploy a huge number of IP cameras in the territory of a smart city, and associate the related video streams to interested users that may be local police, security forces, administrative entities and even simple citizens.

Against the classical approach used by the legacy video surveillance systems[10,11], here, thanks to the presence of the SDN/NFV interconnection network, the video stream generated by each IP camera is automatically rerouted directly to the "interested receivers" in a point-to-multipoint fashion. Thanks to this peculiarity, installation of new cameras is trivial because cameras do not need to be configured since where sending the video stream is automatically decided by the network. Moreover, thanks to the contribution of SDN, video stream generated from a camera is not replicated for each destination, while thanks to the contribution of NFV, new plugins can be easily added in the form of service chains of Virtual Functions (VFs) between the source and the destination of a data stream. For example, additional virtual machines can be run in the network to provide network- and application-layer services, like for example video rate control[12,13,14], flow encryption[15,16], or TCP flow control[17]. Let us stress that point-to-multipoint communication is not realized with approaches that can be now considered obsolete, as for example peer-to-peer (P2P) multipoint communication, which may present some instability problems[18,19,20]. On the contrary, point-to-multipoint communication in the system proposed in this paper is realized within the network, so minimizing traffic and maximizing performance thanks to the possibility of orchestrating network-level and application-level resources at the same time.

2. Platform description

The target of the proposed platform is to develop a video surveillance platform that presents the following main peculiarities: smart, plug-and-play, flexible, scalable in terms of number of transmitting and receiving devices.

More specifically, the access to the platform is achieved by positioning SDN/NFV-compliant Smart Access Node (SAN) devices realized by using general-purpose hardware providing WiFi or 4G connectivity, each being in charge of covering a small/medium area (e.g. car park, square, school and so on), and allowing the connection of both video transmitters and receivers. Each user connected to the platform, through a web application or a mobile app, has a map of the territory covered by the platform (i.e. the smart city), with all the active cameras represented with a green circle. Association of one or more cameras to a registered user is done very easily by clicking on the map viewed on the screen, or through a QR-code that is present in proximity of the camera. The user can then customize the received video and the events associated to each camera, for example requiring the system to be alerted in case of a motion detection from a specific camera. Other tools are available like for example a mosaic view conveying the flow stream of multiple cameras. Of course, more than one user can be "interested" to the same camera installed in a given area covered by the service. Thanks to the presence of the SDN/NFV underlay network, the video stream generated by each IP Camera is automatically rerouted directly to the "interested receivers" in a point-to-multipoint fashion. Thanks to this peculiarity, installation of a new camera is trivial because no additional configuration is needed: the destination of a video stream is automatically decided by the Platform Orchestrator. Moreover, thanks to

the contribution of SDN, video stream generated from a camera is not replicated for each destination, while thanks to the contribution of NFV, the platform is able to support a large number of personalized services based on the users' requirements: more in detail, each data flow during the path from source to destination will traverse a set of VFs (service chain) according to the kind of service requested by the end user. Furthermore, the platform is able to easily integrate new plugins in order to support new capabilities and provide new functionalities.

The proposed video surveillance platform (VSP), as shown in Figure 1, is constituted by five main blocks: video sources, users or video receivers, SDN/NFV Network, Virtual Functions or Service Plugins and Video Service Manager.

Video transmitters are any networked device that is able to transmit video to an IP address, like for example webcams, IP cams, smartphones and tablets. It can use any standard encoders, at any resolution. Video receivers, on the other hand, include smartphones, tablets and personal computers.

The Service Manager represents the front-end of the platform. It manages the platform users, with their profile and their requirements, and the registered video transmitters. Moreover, by connecting to it through a web interface

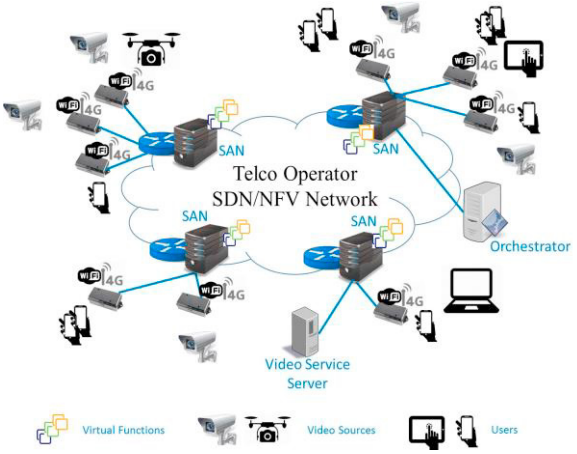

Figure 1. Application scenario.

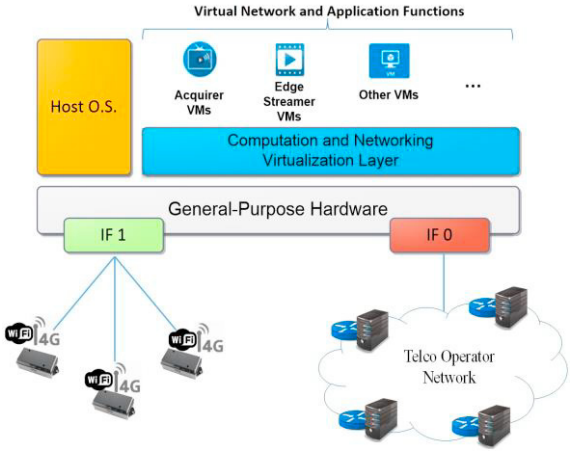

Figure 2. Smart Access Node Architecture.

or a mobile app, users can customize the received services, as explained below. One of the main characteristics of platform is its extendibility. This is achieved by installation of additional optional elements called Plugins. Plugins are software tools that are compliant with a specific interface towards the network. These plugins, together with the basic services already provided by the platform at the beginning of the service setup, can be easily concatenated to realize more sophisticated services.

Example of possible plugins are:

- Video cryptography: this plugin is constituted by two pieces of software, one to be inserted at the beginning of the chain to encrypt the transmitted video, and the other at the end of the chain, to decrypt the received video.
- Area monitoring: this plugin allows the user to define a portion of the monitored area in order to receive an alert if some movement is revealed in it.
- Area obscuring: this plugin allows obscuring a portion of the monitored area for privacy reasons.
- Target follower: this plugin allows to point a given target (for example a person viewed in a camera), and follow it even if moves to areas covered by other cameras, useful in security scenarios.
- Mosaic: this plugin allows a user to compose more than one video streams, each coming from a different camera, into one composite video stream, so achieving the possibility of watching more than one flow simultaneously.

Finally, the SDN/NFV network is a communication infrastructure that is compliant with the software defined network (SDN) and network function virtualization (NFV) paradigms. Thanks to these paradigms, it is possible to

"softwarize" both network and application functions in the network nodes, and centralize the intelligence to coordinate, manage and allocate resources in a central entity called Orchestrator.

By using an SDN/NFV network, it will be possible to easily connect video transmitters and receivers to the platform in a "plug-and-play" way, by means of Smart Access Nodes. In fact, thanks to the NFV paradigm, it is possible to create a service chain to manage video flows while, thanks to the SDN paradigm, video flows generated by video transmitters are intercepted at the entrance of the network, and automatically redirected to a required service chain, and finally redirected to the interested receivers.

The application of the SDN/NFV paradigms provides the platform with a scalability characteristic, making the most important difference in respect to the current state of art. In fact, in the state-of-art video surveillance systems, video processing is done in a centralized server deployed over-the-top (OTT) in respect to the underlying network. Therefore, in those systems, the underlying network has to be able to convey to the centralized server as many input flows as the number of input cameras, and from the centralized server to the users as many flows as the number of receiving users. On the contrary, in this case, video streams produced by cameras can be processed directly on the edge nodes, close to the video sources as much as possible, according to the fog computing approach. Moreover, users interested in the same video flow and accessing the network through the same edge access node cause a load of only one flow, the one directed towards the egress node. The point-to-multipoint communications is realized by the egress node, so avoiding any waste of bandwidth in the core network. For this reason, increasing the number of both video transmitters and receivers do not cause an increasing of the load and the complexity of the underlying network.

3. Benefits achieved with this technology

Taking into account the state of art of video-surveillance systems that are present on the market today, the proposed platform has the following peculiarities that, in the Authors' view, can stimulate the interest of the main system stakeholders, i.e. Telco Providers, Users and third-party video-surveillance service providers. In fact, thanks to the presence of the SDN/NFV technology used to realize the SANs and the NANs, the platform presents the following key advantages:

1) *Reduction of network traffic*, with consequent performance improvements. In fact, the data stream generated by each Data flow Sender is automatically rerouted only and directly to the "interested receivers" in a point-to-multipoint fashion, within the network, avoiding any need of over-the-top (OTT) servers, which cause flow replication even for users accessing the network through the same ingress node.
2) *Scalability*. Network traffic does not increases when a user requesting a given data flow accesses the network from an access node where at least one user is receiving the same flow. Moreover, network traffic increase is linear with the number of Data flow Senders.
3) *Low end-to-end latency*. This advantage derives from the application of the fog-computing paradigm, given that many VFs are provided to the users by their access nodes.
4) *OpEx and CapEX reduction*, since it is realized by software tools running on general-purpose hardware.
5) *Plug-and-Play*. Installation of new cameras or other Data flow sources is trivial because they do not need to be configured: destinations of their video streams are automatically decided by the SDN/NFV Orchestrator.
6) *Platform add-ons*. Platform is able to support a large number of personalized services (e.g. video cryptography, area monitoring, area obscuring, target follower, mosaic) installed as plugins on some SANs or core nodes according to the users' requirements. More in detail, according to the kinds of service requested by the end users, each data flow is routed through the set of required VFs organized in service chains; furthermore, the platform is able to easily integrate new plugins in order to integrate new capabilities and provide new functionalities.

4. Proof of concept

4.1. Emulation framework

The emulation framework has been realized by means of the interconnection of the network emulator Mininet with real devices, such as wireless access points, 4G femtocell, PCs and smartphones. The SDN controller involved

in the proposed network topology is a customized version of OpenDaylight. By the way, all the various component of the platform will be described.

4.1.1. Mininet (computing and networking virtualization environment)

Mininet[21] is an open-source network emulator that allows users to create virtual software-defined networks consisting of an OpenFlow[22] controller, flat Ethernet network of multiple OpenFlow-enabled Ethernet switches, and multiple hosts connected to those switches.

The platform set up in our environment has the peculiarity of binding, by means of Python-language configuration scripts, OpenFlow switch ports with network interfaces of physical devices that result, in such a way, directly connected to the emulation framework. This solution allows the interconnection of physical wireless Access Points (APs) and 3G/4G Radio Access Networks (RANs) to the emulated network and, consequently, any physical device (such as smartphone, pc or any other internetworking appliances) connected to the APs/RANs results as being part of the emulated network.

4.1.2. OpenDaylight (SDN controller)

The SDN controller involved in the emulated platform is the open-source OpenDayLight[23]. It is a Network Operating System for SDN-NFV developed under the auspices of the Linux Foundation and written in Java.

It acts as a Controller in an SDN-NFV enabled infrastructure, also allowing the management of networks divided into slices. In the current demonstrator, it was used the version Hydrogen of OpenDayLight which support OpenFlow 1.0 specifications. OSGi, a Java framework that allows having a modular system, has been exploited. Thereby, it is possible to extend the functionalities of ODL by using so-called Bundles. A Bundle is a component written in Java, using the OSGi Framework and OpenDayLight Java API, which allows the creation of a module for the Controller; a Bundle is executed inside the Controller and can interact with switches in the network.

OpenDayLight interacts with L2 Switches and Applications using Southbound and Northbound interfaces, respectively. Northbound exposes a REST API service that allows managing the network.

To realize the interaction between Southbound and L2 Switches, ODL supports different protocols; in the proposed framework it has been employed OpenFlow 1.0 to guarantee the compatibility with the Open VSwitches emulated by Mininet.

4.1.3. 4G femtocell + Accuver XCore (LTE access network + EPC emulator)

In order to provide the emulated platform with a mobile access network, a “4G femtocell in a box” has been connected to the platform. The attached femtocell, providing LTE connection to mobile devices moving in the area covered by it, was connected to the EPC Emulator Accuver XCORE[24] running in a Laptop through a Wi-Fi router. The connection between these two elements is realized by means of a VPN preconfigured in the femtocell firmware and in the laptop. The XCORE EPC Emulator implements the function of the whole EPC infrastructure on a single PC, allowing in this way the possibility to test and develop solutions for LTE systems without the need of a real network infrastructure. We used Samsung Galaxy S4 smartphones that were preconfigured to work with the femtocell by mounting a specific SIM card.

4.1.4. KVM (virtual functions hypervisor)

Kernel-based Virtual Machine (KVM)[25] has been chosen as virtual functions hypervisor. A client-server application has been developed with the aim of interfacing Network Orchestrator and hypervisor. Clients run in the compute nodes, i.e. the smart access node of the proposed architecture, and communicate with the server from which they receive commands to create, destroy, suspend or migrate virtual functions.

4.1.5. Topology

A testbed topology, shown in Figure 3, has been realized by means of two Intel NUC MiniPCs DC53427HYE (Ubuntu 14.04 as Operating System), named *MiniNet PC* and *PC B*, running MiniNet network emulator and the

OpenDaylight controller, respectively. The MiniNet PC is equipped with three USB-Ethernet adapters in order to provide the required network interfaces for the data and management plane; three WiFi access points act as CPE for 802.11 compliant devices such as PCs, tablets and so on; finally, a 4G FemtoCell coupled with the related EPC Emulator provides access to 4G smartphones connected to the platform.

The machine hosting OpenDayLight controller is also used to host the Orchestrator as well, so acting as the SDN network Controller and as the front-end Server for our service. It was connected with built-in Ethernet interfaces to the other NUC to the manage network, and with a USB-Ethernet adapter to an access point (is this way it can be addressed for nodes inside the emulated network).

A customized launching and configuration script for MiniNet, written in Python language, allows binding among the three physical Ethernet ports of the MiniNet PC and the Open vSwitches emulated by the network emulator.

The overall emulated infrastructure is therefore composed by a fully-virtualized section (provided by the MiniNet tool) and a real network section, connected to the previous part by means of the configuration script thanks to which physical devices result attached to virtual switches.

The virtual network is composed by 8 SDN switches, subdivided into 4 core nodes connected in full mesh manner and 4 edge nodes acting as Smart Access Nodes (SANs), one for each core node.

The physical devices involved in the platform are three WiFi access points, connected to the Smart Access Nodes SAN1, SAN2 and SAN4, each providing wireless connectivity to WiFi devices; the Accuver Femtocell provides LTE radio access and it has been connected to the SAN 2 of the virtualized network. In such a way, it is possible to test the SDN-NFV service also involving 4G access technology.

4.2. Testbed

A simple proof of concept has been launched in the above topology. More in detail an IP camera has been connected to the Smart Access Node 1 meanwhile an Android smartphone running the mobile App has been connected to the Smart Access Node 2. SANs host Open vSwitches and NFV Manager Clients: once started, the first connect to the Open Daylight Controller, the latter connect to the NFV Manager Server, both two hosted in PC B.

We developed a User mobile app for Android with the purpose of testing the platform functionalities. The mobile app has been developed in Android Studio v2.3, ensuring frontward compatibility with Android 5.0 (Lollipop). By means of the mobile App it is possible to:

- Register an IP camera to the Video Surveillance System;
- Join the service.

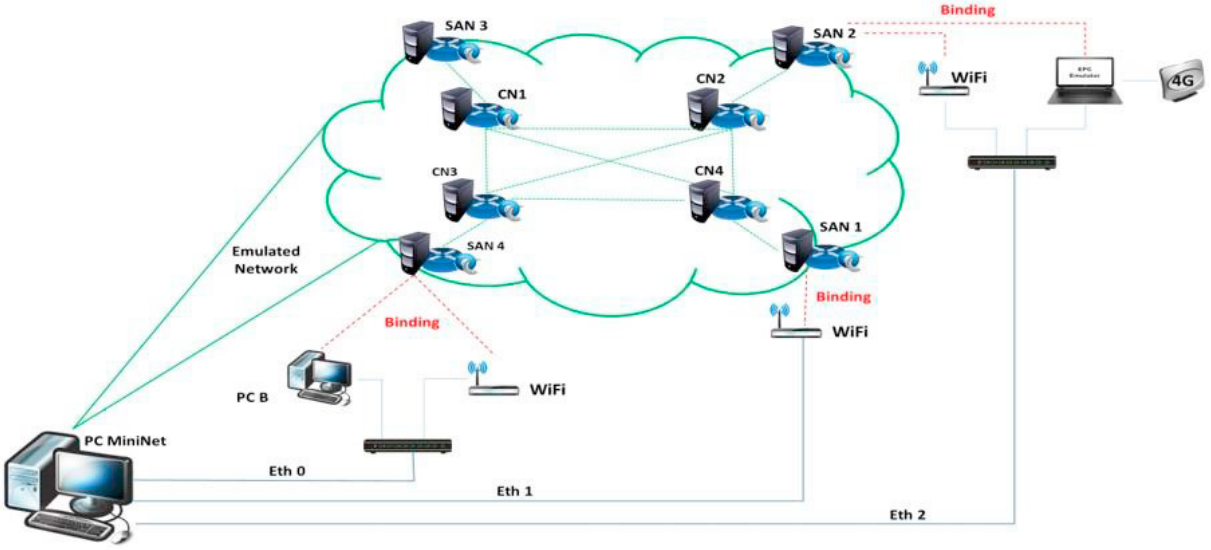

Figure 3. Testbed topology.

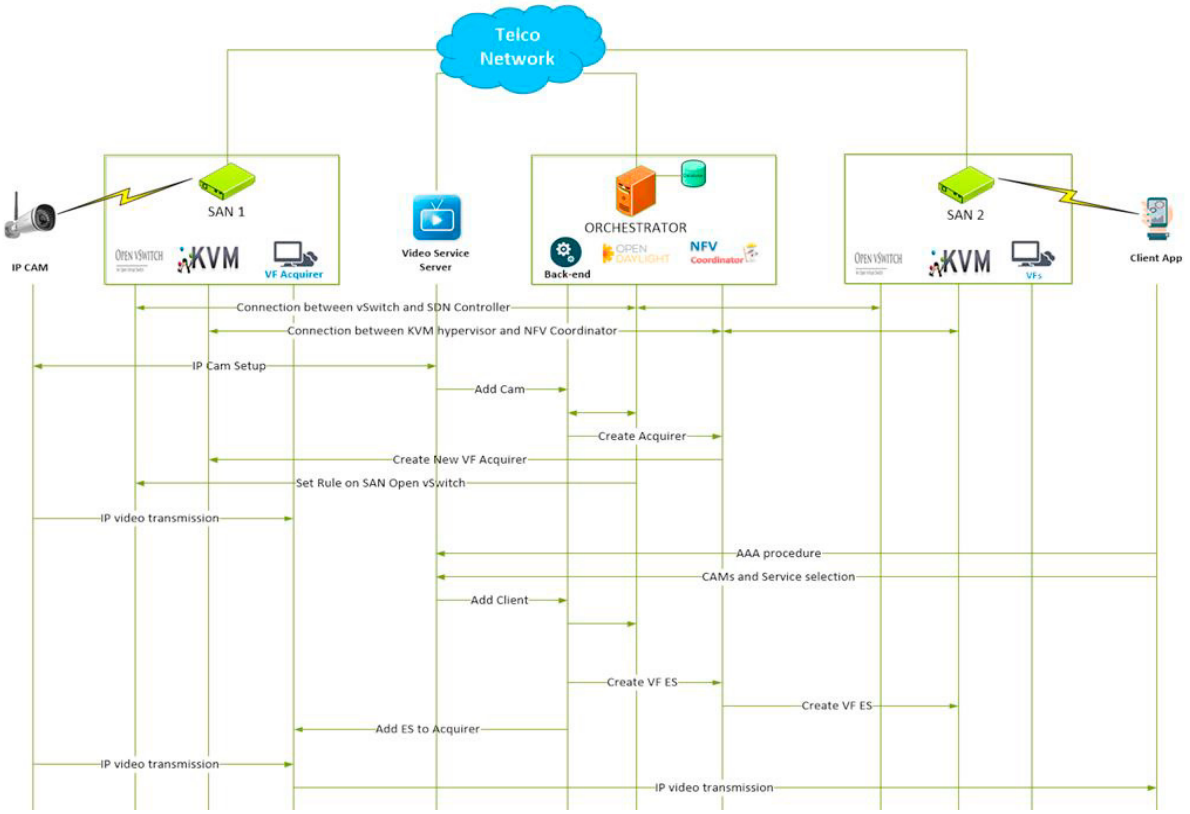

Figure 4. Sequence diagram of the proposed proof of concept.

An IP camera is registered by inserting its MAC address, location name (i.e. a symbolic reference to the place where the camera is deployed), geographic coordinates (to enable the geolocation by means of Google Maps), QR-code (user can choose cameras he is interested to by means of a QR-code reader integrated within the app), available services (motion detection, recording, live streaming, face recognition, etc.), and restrictions (users or group of users allowed to exploit services).

Once registered the IP camera, the Video Service Server communicates its IP address to the back-end server that, through the SDN controller, knows the SAN the IP cam is connected to. In our example a virtual function *Acquirer*, enabling the live video streaming, is launched in the SAN 1. It acts as the root of the logical content delivery tree and its main tasks are recognizing the media streams (coming from one or more cameras), receiving transmission instructions from the Orchestrator (in the form of a list of IP addresses where the media stream have to be forwarded) and forwarding video flows towards the destinations.

In the second phase of the testbed an User joins the service, by means of his mobile app, accessing the network from the SAN 2. The mobile app connects to the Video Service Server allowing User to: select the IP camera by means of its QR-code, its location name or selecting it by exploiting a map; choose the service among the ones associated to the selected camera. Once selected the camera and the service, the Video Service Server communicates the triple (IP User, ID Camera, ID Requested Service) to the Orchestrator; the latter knows the SAN where the User is connected to by sending a query to the SDN controller and communicates with the NFV Coordinator to instantiate a virtual function *Edge Streamer* within the SAN 2. Actually, Edge Streamer performs a transcoding of the video flow by making it compliant with the video player running on the mobile phone and based on the Vitamio library [26]. The Edge Streamer is realized by using a customized version of the open source software Multicat[27].

After instantiated the Edge Streamer, the Orchestrator sends the new destination IP address to the virtual function Acquirer that updates the list of the video flow receivers. From now on it begins the video flow transmission from the IP camera to Users who requested it. Workflow of the above described testbed is shown in Figure 4.

Conclusions

Design and deployment of the proposed platform involves and stimulate a multidisciplinary future work, since it involves expertizes in the fields of telecommunications network, computer science, computer programming, data center management, mobile 4G and 5G networks, video encoding, computer forensics, security, web and mobile app design. For this reason, deployment of this platform can be seen as a first seed to create an infrastructure of competences in further extending the platform for a more sophisticated service with better performance.

Acknowledgements

This work has been partially supported by the H2020 INPUT project.

References

1. White paper on “Software-Defined Networking: The New Norm for Networks”, available at *https://www.opennetworking.org/*.
2. White paper on “Network Functions Virtualisation”, available at *http://portal.etsi.org/NFV/NFV_White_Paper.pdf*.
3. A. Manzalini *et al.,* “Software-Defined Networks for Future Networks and Services,” White Paper based on the IEEE Workshop SDN4FNS, 2014.
4. A. Manzalini and R. Saracco, "Software Networks at the Edge: A Shift of Paradigm," *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Trento, 2013.
5. V. Sekar, S. Ratnasamy, M. K. Reiter, N. Egi, G. Shi, "The middlebox manifesto: Enabling innovation in middlebox deployment", Proc. ACM HotNets-X, pp. 1-6, 2011.
6. G. Calarco, C. Raffaelli, G. Schembra, G. Tusa, “Comparative Analysis of SMP Click Scheduling Techniques,” Proc. of QoSIP 2005, Catania (Italy), February 2-4, 2005, pp. 379-389.
7. R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, “The Click modular router,” Proc. of the 17th ACM Symposium on Operating Systems Principles (SOSP '99), pages 217--231, Kiawah Island, South Carolina, December 1999.
8. A. Lombardo, C. Panarello, D. Reforgiato, G. Schembra, “Measuring and modeling Energy Consumption to design a Green NetFPGA Giga-Router,” in Proc. of IEEE Globecom 2012, Anaheim, California, USA, 3-7 December 2012.
9. P. Sharma, S. Banerjee, D. Demir, S. Natarajan and S. Mandavilli, "NEEM: Network energy efficiency manager," *2012 IEEE Network Operations and Management Symposium*, Maui, HI, 2012.
10. A. Lombardo, *et al.*, “Multipath Routing and Rate-Controlled Video Encoding in Wireless Video Surveillance Networks,” Multimedia Systems, Volume 14, Number 3, pp. 155-165.
11. X. Fu and B. l. Guo, "Framework for Distributed Video Surveillance in Heterogeneous Environment," *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, 2008.
12. L. Galluccio, *et al.*, “An analytical framework for the design of intelligent algorithms for adaptive-rate MPEG video encoding in next generation time-varying wireless networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 23 No. 2, February 2005.
13. Thung-Hiung Tsai and Jin-Jang Leou, "A rate control scheme for H.264 video transmission," *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, Taipei, 2004.
14. A. Lombardo, G. Schembra, “Performance evaluation of an Adaptive-Rate MPEG encoder matching IntServ Traffic Constraints,” *IEEE Transactions on Networking*, vol. 11, no. 1, pp. 47-65, February 2003.
15. D. M. Dumbere and N. J. Janwe, "Video encryption using AES algorithm," *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, Coimbatore, 2014.
16. M. Li, C. Yang and J. Tian, "Video Selective Encryption Based on Hadoop Platform," *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, 2015, pp. 208-212.
17. A. Lombardo, M. Barbera, C. Panarello, G. Schembra, “Active Window Management: an efficient gateway mechanism for TCP traffic control”, Proc. IEEE ICC 2007, GLASGOW, Scotland (UK), 24-28 June 2007.
18. M. Barbera, A. Lombardo, G. Schembra, M. Tribastone, “A Markov Model of a Freerider in a BitTorrent P2P Network,” Proc. IEEE Globecom 2005, St. Louis, MO, USA, 28 Nov. – 2 Dec. 2005, pp. 985-989.
19. N. Magharei, Y. Guo and R. Rejaie, "Issues in Offering Live P2P Streaming Service to Residential Users," 2007 4th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 2007.
20. A. G. Busà, A. Lombardo, M. Barbera, G. Schembra, “CLAPS: A Cross-Layer Analysis Platform for P2P video Streaming,” *Proc. IEEE ICC 2007*, GLASGOW, Scotland (UK), 24-28 June 2007.
21. Mininet, Available online at http://mininet.org/
22. Openflow, Available online at http://www.opennetworking.org/sdn-resources/openflow
23. OpenDaylight, Available online at http://www.opendaylight.org
24. Xcore LTE EPC Net. Emulator, http://www.accuver.com
25. Kernel Virtual Machine, https://www.linux-kvm.org/page/Main_Page
26. Vitamio SDK, https://www.vitamio.org/en/
27. Multicat, http://www.videolan.org/projects/multicat.html