SDN/NFV Architecture for IoT Networks

Ángel Leonardo Valdivieso Caraguay[1], Patricia Jeanneth Ludeña-González[1],
Rommel Vicente Torres Tandazo[1] and Lorena Isabel Barona López[2]
[1]*Department of Electronics and Computer Science (DCCE), Universidad Técnica Particular de Loja, Loja, Ecuador*
[2]*Department of Informatics and Computer Science (DICC), Escuela Politécnica Nacional, Quito, Ecuador*

Keywords: IoT, NFV, SDN, QoS.

Abstract: Since the appearance of the Internet of Things concept the research community has been focused on how to allow an efficient communication and data analysis of millions of devices connected to Internet. However, the current efforts have not enough in order to cover the IoT requirements and challenges. In recent years, technologies like SDN, NFV, edge computing, among others, have been pushing not only traditional networks but also the IoT environments. One of the main concerns is to ensure the quality of the services provided by IoT devices, for its purpose the decouple of data and control plane, the virtualization of network functions, advanced data analysis capacity and the ability to share and put closer the services are considered promising characteristics for this kind of environment.This paper presents an architecture that integrates SDN and NFV focused on IoT environments and a proof of concept to enhance the quality of services. The experiment takes advantage of the controller capabilities in order to modify QoE/QoS flows in real time by means the configuration of SDN-App.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The growth of devices connected to Internet has brought uncertainty and concerns about the real capacity of current networks. According the Leading IoT Gartner report (Gartner, 2017), the number of connected devices will exceed the 20 billions by 2020 and thus the industry and the research community are introducing novel technologies in order to ensure the correct and efficiently operation of the network. Software Defined Network (SDN), Network Function Virtualization (NFV), Self-organized network (SON), Mobile Edge Network (MEC), cloud and fog computing, artificial intelligent, among others, are considered key enabled technologies in this totally networked world.

The Internet of Things (IoT) could take advantage from the combination of these technologies in order to provide a context aware environment (Palattella et al., 2016),(Li et al., 2015). The main current IoT challenges are related to technology convergence, processing of huge volume data, network programmability, limited capacity, energy scarcity, context modeling, reasoning, ensuring Quality of Services (QoS), security, trust and privacy, among others (Sheng et al., 2013), (Perera et al., 2014). The key idea behind NFV is to share the network infrastructure between different physical infrastructures/vendors in order to deploy a traditional network function as a virtual instance. For its part, SDN separates the data plane (forwarding tasks) and control plane (decision and control tasks) of current network devices, allowing the network programmability. In the IoT context, the synergy between SDN and NFV aims to enhance the control and deployment of IoT devices or sensors in an efficient and cost-effective way (Bizanis and Kuipers, 2016), (Barona López et al., 2015).

Furthermore, IoT devices not only have a critical problem in terms of network, computing and storage capacity but also scalability, data access and complex analysis requirements (Díaz et al., 2016). So, IoT could take advantage from edge and cloud computing due their facility to provide unlimited resources (storage, computing and network) closer to the required service or where the user is located. These resources can be quickly deployed with a minimal effort or interaction of the service administrator, establishing on demand business model.

IoT environments require the processing, correlation and analysis of raw data acquired by differed devices such as sensors, but because of their limited resources, these processes must be done outside. In

Caraguay, Á., Ludeña-González, P., Tandazo, R. and López, L.
SDN/NFV Architecture for IoT Networks.
DOI: 10.5220/0007234804250429
In *Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018)*, pages 425-429
ISBN: 978-989-758-324-7
Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

this context, advanced analysis capabilities and artificial intelligent can aid to build knowledge based on the massive data sending by IoT devices. This approach lets to offer new kind of applications knowing as Cloud of Things or Everything as a Service (Cavalcante et al., 2016). One of the main issues in this regard is to ensure the quality of the provisioned services. From the Quality of Service point of view, SDN and NFV play an important role in order to ensure the service level agreements (SLAs) through their elasticity to manage and deploying new network functions or sensors when the QoS levels are decreasing or if the user needs additional services. This lets not only enhance the QoS of the provisioned service but also to provide a better Quality of Experience (QoE) to end users. Al the same time, the service and telecommunication providers can be benefited by the reduction of both capital and operational expenditures (CAPEX and OPEX).

This paper presents an IoT architecture based on the combination of SDN and NFV with the objective to realize proofs of concept related to QoS challenges in this kind of environments. This document is organized into four sections, being the first of them the present introduction. Section 2 describes the proposed SDN/NFV Architecture. Section 3 shows the results of the flow modification and its influence in the QoS/QoE levels, in real time. Finally the conclusions are presented in Section 4.

2 SDN/NFV ARCHITECTURE

The global description of the architecture is described in Figure 1. The proposed architecture follows and integrates the design principles promoted by ONF (Ong, 2017) and ETSI (Matias et al., 2015). The model is composed by well-defined four layers: Infrastructure, Control and Virtualization, Application and Orchestration and Management.

The different layers are composed by sublayers (Figure 2), as follows:

1. *Infrastructure Layer.* It includes the hardware and basic software components needed to forwarding the traffic. In contrast to traditional network devices, IoT devices also originates their own traffic provided by the internal sensors. In this context, a dual or hybrid functionality is proposed. IoT nodes execute traditional forwarding protocols such as (AODV (Chakeres and Belding-Royer, 2004), BCHP (Gondaliya and Kathiriya, 2016), DSDV (Perkins et al., 2001)), among others, depending of their standard capabilities. If the nodes have connection with the IoT controller, the nodes operate as a data plane waiting for the configuration provided by control plane. The southbound API enables the communication between data and control plane.

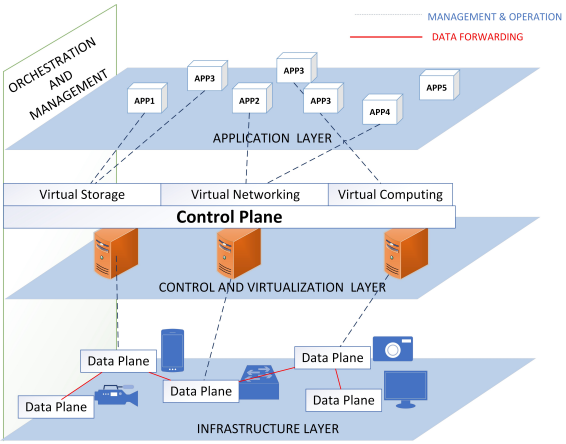

Figure 1: IoT SDN/NFV Architecture.

2. *Control and Virtualization Layer.* It provides the control of the forwarding data behavior and the virtualized resources for the instantiation of the NFV Apps. For this purpose, it is composed by the control plane and virtualized elements. The control plane has basic control functions. The Device Manager registers the available nodes that has connection with the controller and can receive control plane messages. The unique identification of the nodes (e.g. IP, MAC address) and other identification parameters will be available for the other modules. Similarly, the Topology Manager uses the information of the Device Manager and attempts to map the location and connectivity capabilities of the devices. The Statistics Manager listens the communication messages between data and control plane and infers some basic statistics information (e.g. bytes sent by node links). Furthermore, the Flow Manager registers the active data flows in the network.

   The administrator also has the capability of include their own control plane modules for special purposes. These applications are known as SDN Apps. For its part, the virtualized resources (storage, networking and computing), known as NFV Apps, are available for their download, installation or configuration in the virtual infrastructure. These Apps are considered high level applications.

3. *Application Layer.* The different NFV applications are located on this layer. This layer follows the virtualization principles applied in computer science, where different user applications

can be executed in different operating systems (OS) sharing the same hardware resources. In this context, the different users can share virtualized resources (storage, network and computing) to execute their applications in a isolated environment. Examples of NFV Apps include security apps, QoE, data analysis, among others.

4. *Orchestration and Management.* The paradigms of separate data and control planes, network function softwarization and resources virtualization require the coordination of the different layers of the infrastructure. For this reason, the Orchestration and Management layer operates and has the possibility of take actions on the other layers of the infrastructure. It is responsible to ensure the enough resources for the instantiation and operation of the VNF Apps and Control Apps. For this purpose, it is composed by three sublayers: VNF Manager, Orchestrator and Virtual Infrastructure Manager (VIM). The VIM has a clear and updated view of the installed infrastructure and the available virtualized resources (storage, networking, computing). This information is sent to the Orchestrator.

   For its part, the Orchestrator receives the VIM information and ensures the good operation and isolation of virtualized elements. Moreover, the Orchestrator authorizes the instantiation of Virtual Network Functions (VNF) only if virtual resources are available. Then, the VNF Manager organizes and registers the instantiation of available VNFs. It also receives the NFV instantiation requests from different customers.

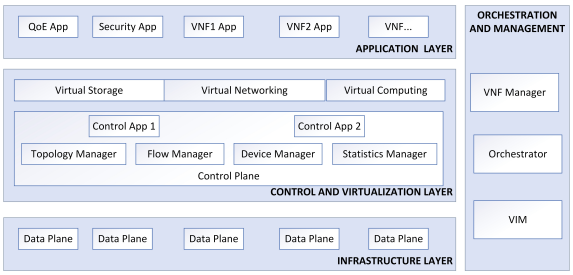

Figure 2: IoT SDN/NFV Architecture.

3 RESULTS

In order to demonstrate the feasibility of the proposed framework, the simulation results are focused on the behavior of Infrastructure and Control and Virtualization Layer. For this purpose, mininet (Lantz et al., 2010) emulations were applied. Mininet is the SDN emulator widely adopted by research community. Mininet uses python scripts to emulate custom network topologies and includes virtual hosts, OF-enabled switches and virtual links. However, the emulation performance in real time applications is dependent on the underlying host system capacity. In this context, the use of small scale topologies guarantee the accuracy results and CPU/memory isolation.

The topology is emulated using a server Acer Swift 3 (Intel i7, 2.7 Ghz, 8GB) running a virtual machine with Linux Ubuntu 16.04. The topology (Figure 3) emulates a common data center with core, aggregation and edge OF-enabled switches (s1-s7). The switches are connected to the SDN Floodlight (Floodlight Controller,) controller. Floodlight provides a REST-API to request the main SDN services required to implement SDN-Apps. The present experiment uses the QoS-App (Wallner and Cannistra, 2013), (Wallnerryan,). Each edge switch (s4-s7) is connected with two hosts (h1-h8). The virtual hosts represent the IoT devices and generate video streaming information. The virtual hosts use a VLC server to stream the video file “highway_cif ”(Telecommunication Networks Group,) using RTP/UDP. The video is encoded in MPEG4 (highway_cif.mp4) with a size file of 4.23 MB and a resolution of 352 x 288. The video is composed by 2000 frames and a total duration of 66 seconds.

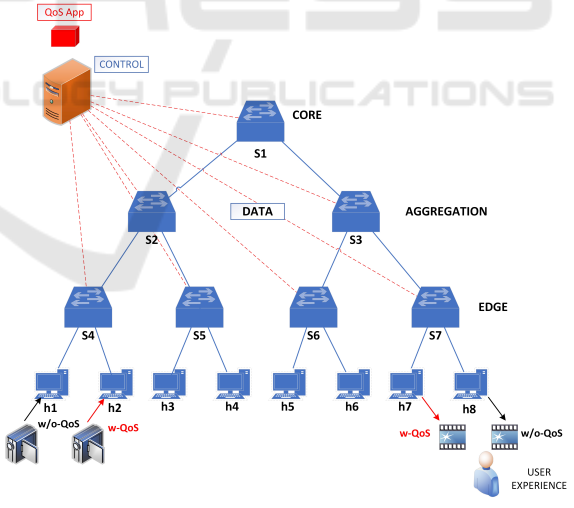

Figure 3: Test Topology.

The objective of the experiment is to demonstrate the controller capability to customize the switches behavior without affect the normal operation of the network. The experiment uses two streaming flows. The first streaming (w/o-QoS) is sent from h1 to h8 and the second stream (w-QoS) sents the video from h2 to h7. Both videos are simultaneously sent through the network. During the streaming, after 45 seconds, the QoS-App is downloaded and installed in the control-

ler. Then, the QoS-App is automatically configured to control the bandwidth of the links depending of the flow. The w-QoS streaming (flow from h2 to h7) is limited with a maximal datarate of 2Mbps and the second flow w/o-Qos (h1 to h8) is limited to a maximal datarate of 0.4 Mbps. Then, the received streams are saved in different video files.

The received files are processed using the Evalvid Tool (Department of Telecommunication Systems,), (Klaue et al., 2003). For this purpose, the files are decoded as .yuv files and Evalvid evaluates the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Metric (SSIM) between both, sent and received streams. Since several processes are executed simultaneously in the same VMs (mininet, Floodlight, VLC, Evalvid), slight variations between test are depicted depending on the CPU and memory resources. For this reason, the Monte-Carlo method is used. That is, the scenario is tested 20 times and the corresponding average is evaluated.

The results of the experiments are depicted in the Figures 4, 5 and 6. With the purpose of reduce distortions, the trend line with an average of 20 frames is depicted. Figure 4 shows the PSNR average against the number of frames. The red line (solid) represents the w-QoS streaming and the blue line (dotted) represents the w/o-QoS. At the beginning of the experiment, both streaming flows have similar behavior (best effort). For this reason, the average PSNR is clearly similar between them. The vertical black line depicts the point at which the QoS-App is downloaded and configured in the Floodlight controller. As expected, after the QoS-App configuration, the behavior of the switches are automatically modified to identify the flows and assign different QoS policies. As a result, the h2-h7 traffic shows better levels of PSNR compared with h1-h8 flow. In this context, the average w-QoS is 26,54 dB against 23,73 dB of the w/o-QoS stream.

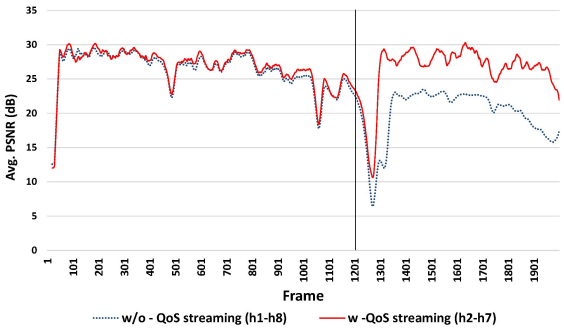

Figure 4: Peak Signal to Noise Ratio.

The Figure 4 shows a considerable trough in case plot. The experiments show that the unexpected effect is mainly caused by the fast moving scenes in this time. The network load is increased and the PSNR decreases.

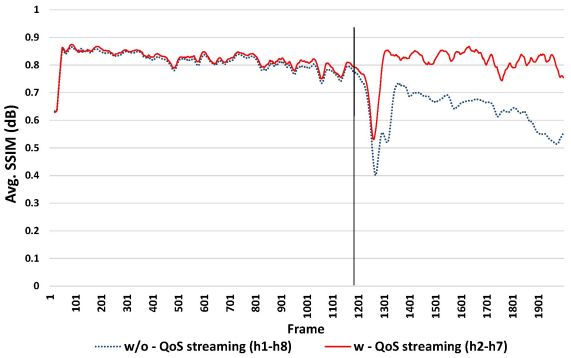

Figure 5: Structural Similarity Index Metric.

For its part, the Figure 5 outlines the SSIM of the test flows. The average SSIM of w-QoS is 0,814 against the w/o-QoS streaming with 0,738. Finally, Figure 6 summarizes the above results with the calculation of MOS (Mean Opinion Score). The MOS (ITU, 1996) attempts to estimate the grade of acceptability of the user (QoE). The scale is proposed as follows: (5) excellent, (4) good, (3) fair, (2) poor and (1) bad. Using this background information, the average MOS during the streaming is calculated. The average for w/o- QoS is 2,302 against 2,755 of the w-QoS streaming. The obtained results demonstrates the controller capability to dynamically modify the network behavior and balance the data flows depending of the user requirements.

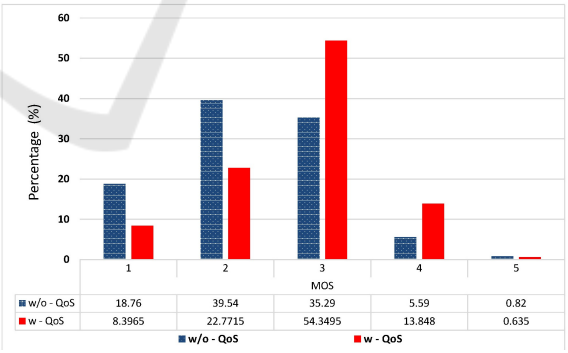

Figure 6: Mean Opinion Score (MOS).

4 CONCLUSIONS

The present work analyzes the advantages that SDN and NFV paradigms introduce in IoT environments. Moreover, a SDN/NFV architecture for IoT networks is proposed. In this context, the controller capability to dynamically control the network behavior is

tested. The mininet based test topology and the video streaming analysis demonstrate that the floodlight controller is capable to modify the QoS/QoE of different flows in real time. Therefore, the challenges for future research include the balance and orchestration of virtual resources for IoT environments. Similarly, the optimization of algorithms for real streaming in SDN/NFV architectures is challenging.

ACKNOWLEDGMENT

The authors are with the Group of Robust, Sustainable and Secure Networks (SRSNet), Department of Electronics and Computer Science (DCCE), Universidad Técnica Particular de Loja (UTPL), Loja, Ecuador.

REFERENCES

Barona López, L. I., Valdivieso Caraguay, Á. L., Villalba, L. J. G., and López, D. (2015). Trends on Virtualisation with Software Defined Networking and Network Function Virtualisation. *IET Networks*, 4(5):255–263.

Bizanis, N. and Kuipers, F. A. (2016). SDN and Virtualization Solutions for the Internet of Things: A Survey. *IEEE Access*, 4:5591–5606.

Cavalcante, E., Pereira, J., Alves, M. P., Maia, P., Moura, R., Batista, T., Delicato, F. C., and Pires, P. F. (2016). On the Interplay of Internet of Things and Cloud Computing: A Systematic Mapping Study. *Computer Communications*, 89:17–33.

Chakeres, I. D. and Belding-Royer, E. M. (2004). AODV Routing Protocol Implementation Design. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 698–703. IEEE.

Department of Telecommunication Systems. Evalvid. Available at http://www.tkn.tu-berlin.de/menue/research/evalvid.

Díaz, M., Martín, C., and Rubio, B. (2016). State-of-the-art, Challenges, and Open Issues in the Integration of Internet of Things and Cloud Computing. *Journal of Network and Computer Applications*, 67:99–117.

Floodlight Controller. Open Source Software for Building Software-Defined Networks. Available at http://www.projectfloodlight.org/floodlight (2018/04/19).

Gartner (2017). *Leading IoT Gartner Report*.

Gondaliya, N. and Kathiriya, D. (2016). Map Based DTN Architecture and an Efficient Routing Protocol in Delay Tolerant Networks for Post Disaster Situation. *International Journal of Computer Science and Information Security*, 14(8):980.

ITU (1996). Methods for Subjective Determination of Transmission Quality. Recommendation P.800, International Telecommunication Union, Geneva.

Klaue, J., Rathke, B., and Wolisz, A. (2003). Evalvid–A Framework for Video Transmission and Quality Evaluation. In *International conference on modelling techniques and tools for computer performance evaluation*, pages 255–272. Springer.

Lantz, B., Heller, B., and McKeown, N. (2010). A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM.

Li, S., Da Xu, L., and Zhao, S. (2015). The Internet of Things: a Survey. *Information Systems Frontiers*, 17(2):243–259.

Matias, J., Garay, J., Toledo, N., Unzilla, J., and Jacob, E. (2015). Toward an SDN-enabled NFV Architecture. *IEEE Communications Magazine*, 53(4):187–193.

Ong, L. (2017). ONF SDN Architecture and Standards for Transport Networks: Control Architecture and Network Modeling I. In *Optical Fiber Communication Conference*, pages M2H–1. Optical Society of America.

Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., and Ladid, L. (2016). Internet of Things in the 5G era: Enablers, Architecture, and Business Models. *IEEE Journal on Selected Areas in Communications*, 34(3):510–527.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context Aware Computing for the Internet of Things: A Survey. *IEEE communications surveys & tutorials*, 16(1):414–454.

Perkins, C. E. et al. (2001). *Ad Hoc Networking*, volume 1. Addison-wesley Reading.

Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J., and Leung, K. (2013). A survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities. *IEEE Wireless Communications*, 20(6):91–98.

Telecommunication Networks Group. Evalvid Video File. Available at http://www2.tkn.tu-berlin.de/research/evalvid/cif/highway.

Wallner, R. and Cannistra, R. (2013). An SDN Approach: Quality of Service using Big Switchs Floodlight Open-source Controller. *Proceedings of the Asia-Pacific Advanced Network*, 35:14–19.

Wallnerryan. Floodlight with QoS module. Available at https://github.com/wallnerryan/floodlight-qos-beta (2018/04/19).