Journal of Mobile Multimedia, Vol. 13, No.3&4 (2017) 183-196

TOWARDS ENHANCED QOS MANAGEMENT SDN-BASED FOR NEXT GENERATION NETWORKS WITH QOE EVALUATION: IMS USE CASE

SARA KHAIRI
Dept. Telecommunications Systems, Networks and Services, STRS Laboratory
National Institute of Posts and Telecommunications
Rabat, Morocco
sara.khairi@gmail.com

BRAHIM RAOUYANE
Dept. Information Technology, Networks and Telecommunications, IT&NT Laboratory
Hassan II University, FS Ain Chock, Casablanca, Morocco

MOSTAFA BELLAFKIH
Dept. Telecommunications Systems, Networks and Services, STRS Laboratory
National Institute of Posts and Telecommunications
Rabat, Morocco

The increase of mobile subscribers' requests and the explosion of multimedia services such as provided by the IP Multimedia Subsystem (IMS) are making QoS management more and more complex. In fact, with the emergence of the Software Defined Network (SDN) paradigm which enables dynamic configuration of network resources and flexible QoS management, Next Generation Networks (NGN) have found necessary to integrate this new concept. The aim of this paper is twofold. First, to present a new architecture SDN-based for Next Generation Networks (NGN) to enhance the QoS management. The architecture is implemented and evaluated for a Video on Demand (VoD) in IMS network. Second, to demonstrate the implementation of the proposed architecture by testing the Quality of Experience (QoE) which is evaluated in terms of the Video Mean Opinion Score (VMOS).

Keywords: IP Multimedia Subsystem (IMS), Quality of Service (QoS), Software Defined Networking (SDN), Quality of Experience (QoE), Video Mean Opinion Score (VMOS), DiffServ, Queuing

1. Introduction

The IP multimedia subsystem (IMS) is a standardized Next Generation Network (NGN) architecture for telecom operators that provide fixed and mobile multimedia services incorporating voice, video and data services. With the convergence of the Internet, fixed and mobile communications and the explosion of multimedia applications, the issue of maximizing the resource utilization and the continuity of Quality of Service (QoS) while satisfying user's Quality of Experience (QoE) requirements has been gaining importance. Indeed, the QoS management entities defined for the IMS are the Policy Control Rule Function (PCRF) and the Policy and Charging Enforcement Function (PCEF). The PCRF supports policy and charging control decision based on session and media related information (e.g. forwarding rules, queues configurations, bandwidth limitation, etc.) obtained from the P-CSCF according to the predefined Service Level Agreements (SLA). The PCRF provides all the policies to the PCEF to be applied according to a QoS management mechanism (i.e. DiffServ, IntServ, etc.). However, in the legacy IMS architecture the user has to deploy and configure QoS mechanisms in all routers without having a global view of the network. With the use of SDN, all settings and configurations are done in the controller which commands all of the network topology. In this work; we will use the PCRF to provide solely policies from the IMS network while the management will be done using a new paradigm called Software Defined Networking (SDN).

The SDN has emerged to provide flexible network programmability by decoupling the control plane from the data plane. The control plane offers a generous northbound Application Programming Interface (API) to communicate with the applications; and it provides a southbound API to manage the forwarding layer and have a global view of the network. The Controller have much knowledge of the network topology and resource availability while the PCRF understands better the context and needs of the user. To this purpose, the overall objective of our work is to study the use of SDN as a way to improve the QoS management for Next Generation Networks. The architecture is evaluated for a VoD service in the IMS networkby measuring the VMOS for three different network states.

Mainly, this paper defines the QoS and QoE concepts in IMS and SDN, and sets out their integration. After, the paper proposes a flexible architecture for QoS management for NG networks with SDN. A new module QoS Manager is implemented in the SDN controller. This module provides a rich API that implements a dynamic QoS configuration of per-flow queuing and Diffserv according to PCRF requirements. Finally, a testbed implementation is evaluated by the mean of the QoE by calculating the Video Mean Opinion Score (VMOS) versus the Quality Scale for an IMS service a use case. The test bed results demonstrate the effectiveness of the proposed architecture.

2. Related Work

Various studies have been done to enhance the QoS in the IMS and SDN networks [1] [2] [3] [4] [5] [6] [7] [8]. The most of the existing works target architectures with limited network view. Some works treat the QoS only in the SDN without experiments for IMS services [5] [6] [7] [8]. Others provide methods for QoS management in IMS without taking advantage of the flexibility of the SDN [1] or only discuss the integration without propositions for QoS management mechanisms [2] [3] [4].

In [1], authors have implemented full-Diffserv architecture for QoS management in the IMS network. The architecture shows good performance for IMS services but it is implemented only for one QoS model that makes it limited. This work [3] presents experience and an implementation effort in integrating OpenFlow mechanisms within IMS, the work is promising; however, it doesn't take advantage of the QoS management in the SDN. In paper [5], they propose a new module in SDN controller using an approach similar to 'Aggregation of RSVP for IPv4 and IPv6 Reservations'. This module tried an intermediate approach where the QoS flows are directed to a DiffServ queue with a fixed rate and the reservation of the queue is reconfigured only when the total reservation for the QoS flows exceed the initial reservation. This solution is dynamic and promising; however, it is tested only for cloud applications and does not consider IMS services. OpenQoS framework [6] dynamically computes QoS routes where the multimedia traffic can be routed without delay. It does the route computation based on the current packet delay and congestion information collected in the network. This framework is used for multimedia traffic and it is interesting to implement its integration with IMS network.

Paper [7] proposes a QoS solution for internet system. The SDN controller configures DiffServ Queues on OpenFlow routers at startup. It also adds flow entries to enqueue packets with ToS field in the appropriate queue. The edge routers identify traffic that needs QoS and add appropriate DSCP value in the ToS field so that the routers configured by SDN controller enqueue the packet in QoS queues. This method is a little similar to the solution proposed in this paper, since they considered the same principal of queuing and separating the edge from the core routers, however, our conception is different. The main difference between this work and ours is that ours proposed a new solution for integrating IMS and SDN via a new interactive API. The IMS services are managed via new modules implemented in SDN. Our proposal configures DiffServ in edge switch and queuing with the incoming parameters from the PCEF in core switch.

In comparison with the existing works above, the solution proposed in this paper provides a new module QoS Manager with an interactive API that mainly uses configurations with dynamic queues (i.e. changeable according to the policies specific to each IMS traffic). The queues are configured with a max and min bandwidth, and packet loss and delay allowed. The DiffServ, using Differentiated Services Code Point (DSCP) value, is set for the edge switches. Using OpenFlow together with policies provides the ability to dynamically adapt and reroute the network flow according to bandwidth traffic. Thereby, a converged IMS and SDN architecture is explored and an approach to fulfill QoS differentiation for IMS services is investigated. The proposed architecture is detailed in section 4.

3. QoS management in IMS and SDN and QoE model

3.1. QoS management in IMS

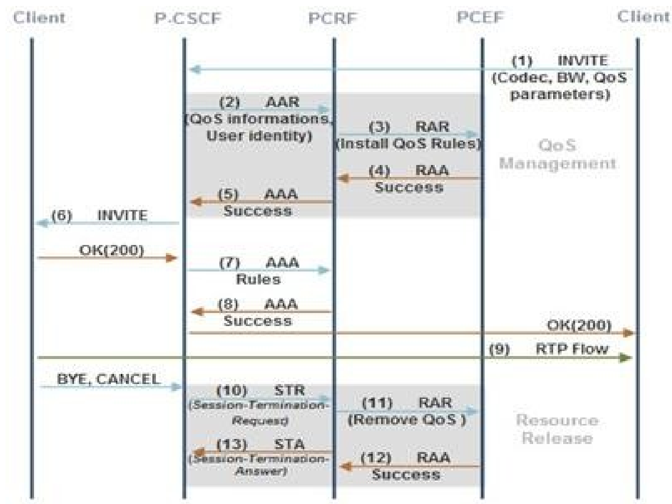

Figure 1 QoS management in IMS for an incoming INVITE.

QoS management in the IMS network is based on the interaction between three entities namely: P-CSCF, PCRF and PCEF. When an IMS call flow starts, the P-CSCF receives from a terminal, an INVITE request (1) that is transmitted to another terminal (6). The P-CSCF sends an Authentication Authorization Request (AAR) message to the PCRF (2) describing the session to be established. The PCRF provides the PCC rules to PCEF by sending a Re- Authentication Request (RAR) message (3). The PCC rules provided by the PCRF are executed by the PCEF which controls the flow of each stream individually. The PCEF ensures that the authorized QoS will not be exceeded, and returns a RAA (Re-Authentication Request) message (4). Figure 1 illustrates this scenario.

3.2. QoS management in SDN

QoS management in the IMS network is based on the interactiThe SDN architecture is based on three layers namely: Applications, control (SDN controller) and infrastructure (SDN switches). In the SDN network, the QoS can be managed statically or dynamically according to the network state. The QoS management is realized by means of control modules centralized in the SDN Controller. The remote controls of these modules are managed through REST APIs in where we implement all instructions about QoS, security and routing. The controller translates the request from REST API to OpenFLow and commands SDN switches. This interaction is presented in Fig. 2.

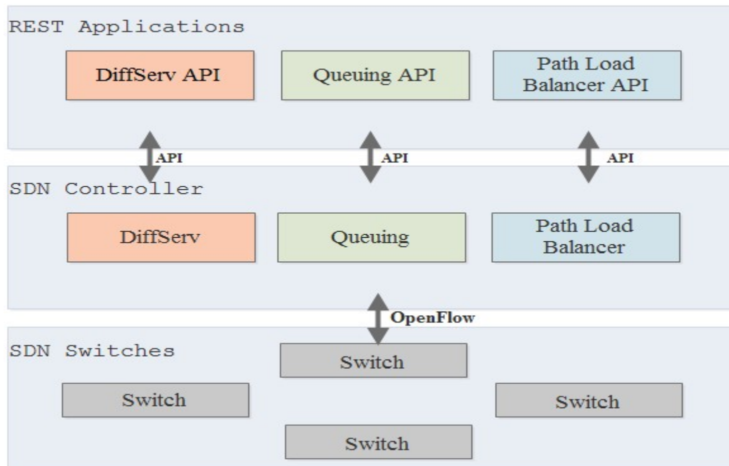

Figure 2 QoS management in SDN.

Since the SDN has more knowledge of the network state and can implement several models and methods through his controller, it will be promising to benefit from their functionalities in the IMS network. Several works propose to evaluate the QoS parameters to perform their architecture. In this work, we propose to estimate another crucial and important model for multimedia services evaluation which is the Quality of Experience (QoE). The widely used metric to estimate the QoE of video applications is the Video Mean Opinion Score (VMOS).

3.3. QoE evaluation model: VMOS

The video quality measurements based on VMOS values are used to derive non-intrusive QoE prediction model and sender bitrate adaptive control mechanism based on non-linear regression methods from [16]. The VMOS is a quantitative indicator of five levels of satisfaction of the potential customers based on the quality of service. The five levels are illustrated in Table I.

TABLE I. LEVELS VERSUS QUALITY SCALE.

Level	Quality scale
1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

The VMOS equation is given in [16] as follow:

$$VMOS = \frac{a_1 + a_2 FR + a_3 \ln(SBR)}{1 + a_4 PER + a_5 (PER)^2} \quad (1)[16]$$

Where: FR: Frame Rate, SBR: Send Bitrate, PER: Packet Error Rate. Those values depend on the video resolution and specifications.

The coefficients according to the type of movement are presented in Table II.

TABLE II. COEFFICIENTS FOR ALL CONTENT TYPES.

	Slight Movement	**Normal Movement**	**Fast Movement**
a_1	4.5796	3.4757	3.0946
a_2	-0.0065	0.0022	-0.0065
a_3	0.0573	0.0407	0.1464
a_4	2.2073	2.4984	10.0437

The proposed architecture is explained in the following section.

4. Experiments and discussion

In comparison with legacy IMS architectures that target limited QoS mechanisms, heavy configurations and restricted view of the network, our proposed architecture provides a new API between IMS and SDN that controls new implemented modules in a programmable controller. In the IMS network, the PCRF examines the incoming messages from the P-CSCF, retrieves appropriate policies, and informs the OpenFlow Controller via the implemented REST API. In this work we have removed the PCEF server which allows a direct communication between PCRF and SDN controller. This API, at this step, replace the PCEF traffic enforcement. In addition, the OF controller contains a set of APIs based applications, here, DiffServ, per-flow queuing and Path Load balancer. The policies provide to the controller the directives to dynamically adapt and reroute the network flows. Fig. 5 highlights the architecture proposed and the added modules.

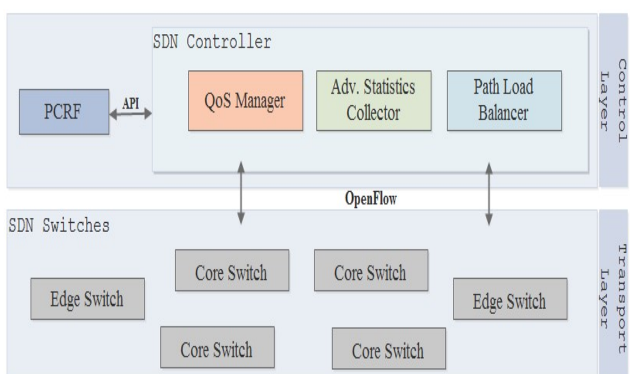

Figure 3 Proposed combined architecture of IMS and SDN with new modules.

The SDN facilitates QoS management at each switch, for more suppleness we have defined two types of switch namely: Edge Switch (short E-S) and Core Switch (short C-S). The edge switch is defined to provide enhanced services, such as QoS by using DiffServ with DSCP, and to connect servers and clients. The core switch represents the middle of the network. At this step, we didn't yet implement all PCEF functionalities in the SDN controller, this will be done in future works. The PCRF sends information about IMS traffic, via the implemented API, to the SDN controller to make appropriate decisions. To this aim, we have defined three modules in the SDN controller.

Advanced Statistics Collector: In order to have a global view of the network state, this module has been implemented, based on the legacy Statistics Collector module [12]. In addition to functionalities defined in the legacy module, the new module defines the type of incoming traffics using an implemented sniffer and identifies them in the network with iptables and iptables layer 7, specifies a Bandwidth threshold for some links to avoid traffic congestion (i.e. we assumed that the appropriate threshold to avoid congestion is fixed to 80% of bandwidth use), and calculates Delay and Packet loss for each link. The Advanced Statistics Collector informs the QoS Manager module of the network state so that to make appropriate decisions.

QoS Manager: This module is implemented based on the existing QoS Manager defined in [8] with major modifications. Our QoS Manager gathers information about network state from the Advanced Statistics collector via a Java API and policies from the PCRF via a REST API to route traffics according to their QoS requirements. The module implements a hybrid QoS mechanism which is Diffserv and per-flow queuing. DiffServ with DSCP is considered in the edge routers (seen in red in Fig. 6 below). Each queue uses three main methods, as their names indicate, to Add, Modify and Delete queue. The queues are identified by the following parameters: a priority, a max and min bandwidth, IP source and destination, packet loss and delay allowed, and Traffic Control (TC) configurations. The implemented API enforces the constraints required by the PCRF, in other words, all of the queues parameters are gathered from the PCRF via the implemented API to form a dynamic QoS configuration for IMS traffics. When the flow packets exceed the threshold defined in Advanced Statistic Collector, the Path Load Balancer is supposed to start work.

Path Load Balancer: This module is used when the threshold specified is exceeded. By default, the controller used in our testbed uses the Dijkstra's which is not suitable in case of congestion. To avoid congested links every switch must act as a dispatcher. The evaluation of this module is subject to future work. The workflow of the interaction of the proposed modules is presented below.

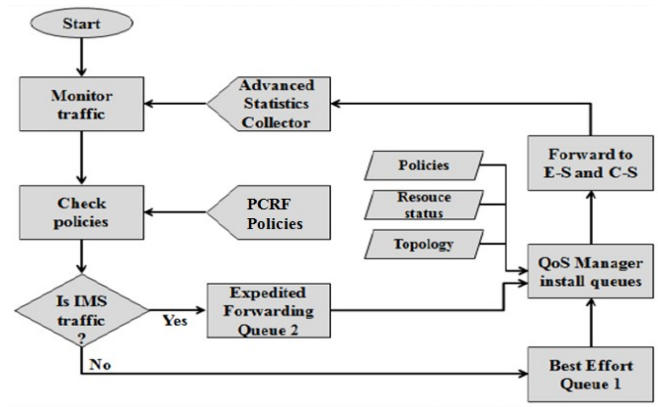

Figure 4 Proposed architecture workflow.

The feasibility and performance aspect of our proposal are evaluated through the following experiments. The reliability and scalability aspects will be evaluated in future work.

5. Implementation

5.1. TestBed description

The test bed is made up of 4 layers implemented with open source solutions:

Service layer - We have used two types of services: VoD AS [15] and Iperf. The Video on Demand (VoD) represents the IMS service. The VoD allows users to watch video content of their own choice at a time of their own. Iperf is an open source performance measuring tool used here to generate TCP traffic and load the network.

Control layer - The control entities of IMS and SDN are situated here. The IMS servers (i.e. HSS and CSCF) are deployed via the open source called OpenIMSCore [14]. The entity which maintains policies control is the PCRF, presented with the solution UCT PCRF without the PCEF server. The SDN controller that we have chosen to allow the management of the QoS in the transport layer is Floodlight [12].

Transport layer - This layer represents the infrastructure devices. All commands and configurations applied to this layer are received from the control layer (i.e. Floodlight Controller). We have used Mininet [13] with a Python script to simulate the topology (OpenVswitch and hosts) and test the traffic flows. The edge switch ensures operations of Classifying, metering, queuing, and scheduling. The Core Switches at the entrance, performs classifying flow and applies PerHop Behavior (PHB). The tool Traffic Control (TC) is used manipulate traffic control settings and ensure each queue gets level of services required for its class. A set of queue disciplines are implemented such as First-In-First-Out

(FIFO), Class Based Queuing (CBQ) and Hierarchical Token Buckets (HTB). The iptables tool is used for packets classifying.

Access layer - This layer contains two types of clients namely: An IMS client and another client non-IMS. The UCT IMS Client [15] which has ample functionality including VoD/IPTV support, is used to represent IMS client. The IMS traffic behavior is evaluated by the mean of VoD service.

To realize the testbed shown in Fig. 5, a configured UCT IMS Core is connected to the UCT PCRF which communicates to the Floodlight Controller via the implemented API. The controller is connected to Mininet in which the topology is created. We have deployed a mesh topology with six switches and four hosts. Four switches are considered as Core Switches (in Fig. 5 C-S), and two others are the Edge Switches (E-S). The video specifications are illustrated in Table III.

TABLE III. VIDEO SPECIFICATIONS.

Name	Resolution	Codec	SBR (kbps)	FR (fps)	PER (%)	Time (s)
Avatar	176x144	H.264	64	25	<5 & >20	60

Two hosts run VoD server and client to generate streaming traffic from S1 to S6, and two other hosts run iperf server and client to generate data traffic from S1 to S6 and overload the network. To make the network in the congestion state, we run a VLC service (VideoLan Client) instead of iperf service. We assume that the VLC video streaming has the same properties of the VoD. The testbed is described in Fig. 5.

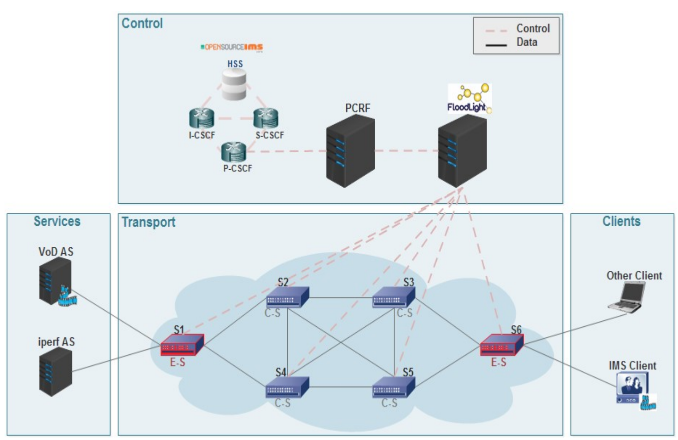

Figure 5 Testbed Architecture.

We have configured two queues with different rates. Queue1 is configured for core switches, Queue2 is configured for edge switches (i.e. S1 and S6) in which we have deployed DiffServ with DSCP. The parameters of Queue2 are provided by PCRF via the implemented API. TABLE IV presents the correspondence between the policies provided by PCRF and the Queues implemented.

TABLE IV. CORRESPONDENCE BETWEEN POLICIES AND QUEUES.

Policies / Queues	*Traffic Type*	*Min BW (Mb)*	*Max BW (Mb)*	*Max Packet Loss (%)*	*DSCP*	*TC*
Queue1	BE1	10	10	20	-	HTB3
Queue2	EF2	10	40	5	46	HTB3

The testbed architecture is evaluated for three network state namely: Normal network with perfect load conditions (generally under 50%), Overloaded network by TCP data with iperf (load conditions under 80%), and network congestion where the network is receiving two types of videos from IMS network and from SDN. The queues configured are tested for the three states in Fig. 6.

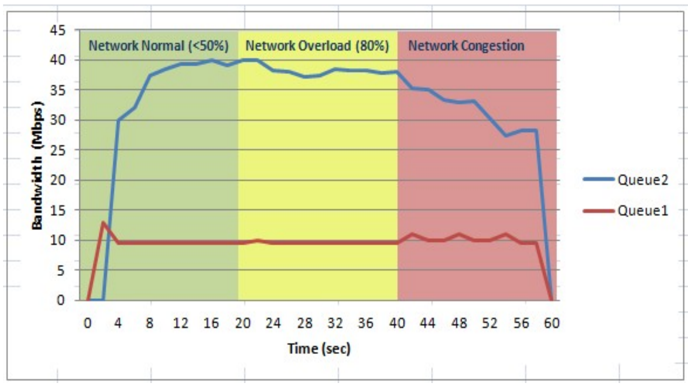

Figure 6 Bandwidth behavior for installed queues for three network states.

Figure 6 shows that the privileged flow uses more bandwidth than the other one; the configured queues are respected even when the both flows take the same path. We assume, in this case, that the queues parameters are the same as when they are provided by PCRF. We studied three use cases to demonstrate why the added modules are suitable to achieve better performance in the IMS network. We have first measured the most known and critical QoS parameter which is Packet Loss for the three network states. After, we have measured another significant indicator of QoE which is evaluated by the VMOS [16].

5.2. TestBed experiments: QoS parameters evaluation

In this section, we simulate three experiments to observe the network response in different situations (normal, overload and congestion). The results present the packet loss percentage for three experiments defined as follow:

Test 1 - QoS management in SDN: In this case, the queues are configured for each type of flow only in the SDN Floodlight Controller (i.e. without interaction with PCRF). To discover what is the privileged flow, both flows take the same path from S1 to S6 (i.e. S1 - S2 - S4 - S6).

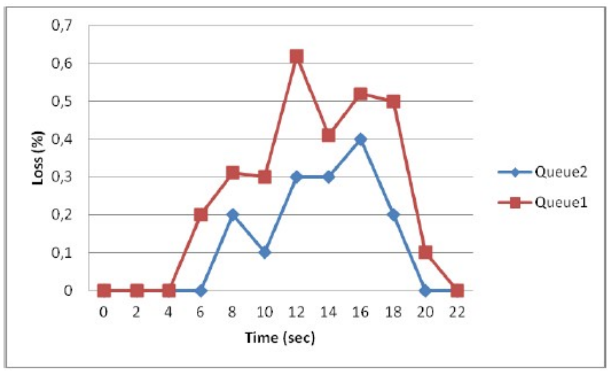

Figure 7 Loss at switch S1 for Queue1 and Queue2.

Figure 7 highlights the percentage of the packet loss for Queue2 which is acceptable for Video streaming, and packet loss for Queue1 with more loss without exceeding the packet loss allowed.

Test 2 - QoS Management in IMS: QoS policies are provided by PCRF depending on user requirement and enforced by the PCEF. In this case, the SDN controller is not used which means that the queues are not involved. We run the VoD as an IMS service and we overload the network with iperf traffic. The parameters used are shown in TABLE IV.

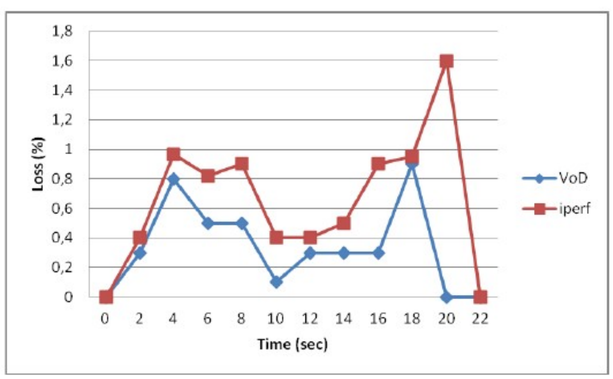

Figure 8 Loss at switch S1 for VoD vs. iperf.

We observe in figure 8 that the loss for VoD streaming traffic in the IMS network is more important than loss in figure 7. This is due to the overloaded network with iperf traffic without applying specific queues to privilege the IMS service.

Test 3 - QoS management in the proposed architecture: The testbed presented in figure 5 have been deployed to demonstrate the performance of our proposal. The policies are provided by the PCRF, and configured by the controller via the implemented API. In this case, we run two hosts one for VoD as the IMS service and the other run another video streaming non-IMS to overload the network. While the VoD is the IMS service our solution takes into account that is the privileged flow so it has to be differentiated.

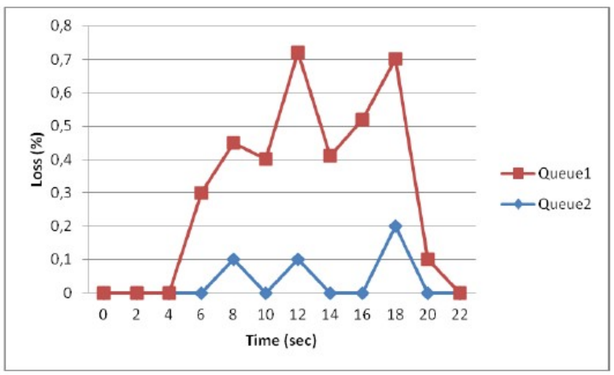

Figure 9 Loss at switch S1 for VoD vs. Video non-IMS.

From figure 9 we can see that the loss for queue2 (i.e. VoD) is inconsiderable and less than the loss in Fig.7. The solution proposed in this paper allows to specific flows such as the IMS flow to be privilege. The QoS is dynamically managed thanks to the enforced policies and the differentiated queues implemented in the SDN controller. More details are provided in [10].

In this part we have evaluated the QoS parameters to perform our architecture. In the coming part, we propose to estimate, for the same experiments discussed earlier, another crucial and important model for multimedia services evaluation which is the Quality of Experience (QoE).

5.3. TestBed results: QoE evaluation

For the 20 first seconds, the services are operating in normal conditions. For the IMS traffic (i.e. VoD streaming) we observe that the QoE evaluated by the VMOS is guaranteed, since the VMOS level presents an amount of 5 which is equivalent to an Excellent quality scale. When an IMS packet arrives to the head of the queue it is marked with the higher priority that's why the VMOS IMS and VMOS IMS&SDN values are better than VMOS SDN value.

For the 20 second seconds, the network is overloaded with other services, a degradation of the VMOS IMS&SDN value is seen, but it still presents a Good quality scale. The VoD service for the IMS remains perfect; this is due to the configuration executed in edge switches (S1 and S6) that prevents non-IMS to monopolize traffic bandwidth.

In the last 20 seconds, all switches are overloaded with video traffics non-IMS. The video quality of IMS network without SDN interaction is degraded because of that. With the configuration deployed for the IMS and SDN integration, the IMS services are always privileged even when other streaming videos are presented. The VMOS IMS&SDN remains above 4 which correspond to a Good quality scale. Indeed, this is making instructions to the transport plan very strict to reject all future services to protect existing privileged services whatever the type of QoS needed. However, this has a direct impact on customer SLA which will be addressed in future work. This scenario is shown in Fig. 10.

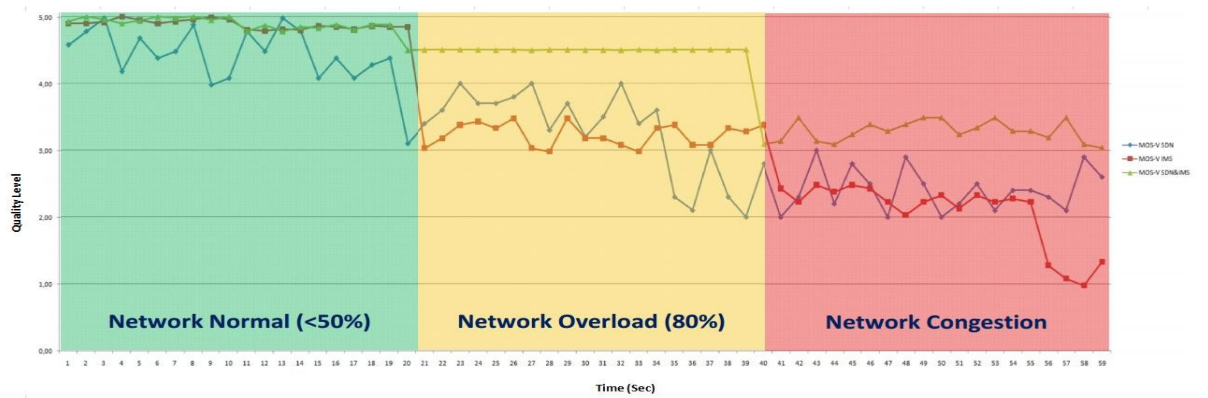

Figure 10 VMOS for a VoD for three tests in three network states.

6. Discussion

The experiments prove that the integration of the SDN in the IMS transport layer has been advantageous for the QoS management. From QoE results, we can consider that the performance of the architecture proposed, at this step, has been guaranteed. From this study, we can come out with the comparison summarized in TABLE IV. By combining the advantages of the both technologies we can predict a bright future for this work.

7. Conclusion

The paper presents a performance evaluation of the use of SDN concepts as a way to improve the QoS management in NG Networks generally, and IMS network specially as a use case. Mainly, the architecture is tested by the means of Packet Loss as a critical QoS parameter. Secondary, the architecture is evaluated in terms of QoE by measuring the VMOS for three different network states. The objective of the paper is twofold. First, to implement an API for PCRF to interact with the controller; it allows reliable communication between PCRF rules and SDN Controller. The controller also includes a set of APIs namely: QoS Manager and Advanced statistics. Thus, the controller becomes interactive with a global view and a dynamic network configuration. Second, to evaluate the QoS of the proposed architecture by the means of Packet Loss parameter and QoE by the means of the

VMOS metric. The experiments highlight our objective to manage the QoS for the NG networks resources in a more efficient way via SDN. Future works aim mainly to enhance the implemented APIs in order to allow SLA negotiation, add PCRF functionalities in the interactive API, differentiate IMS from non-IMS services without neglecting this last, and evaluate the proposed architecture for a mobile network (i.e. LTE network) with more experiments and different conditions.

References

1. B.RAOUYANE and M.BELLAFKIH, "Towards an Autonomous System for QoS management of IMS networks", in IJCSI, Vol. 8 Issue 4, p481, 2011.
2. Z. Liu and al., “A Model of SDN Controllers Supporting for Processing of Flows Based on the IMS “. In APNOMS, 16th Asia-Pacific, 2015.
3. C. TRANOR and al., "Integrating OpenFlow in IMS Networks and Enabling for Future Internet Research and Experimentation". In Future Internet Assembly 2013: Validated Results and New Horizons,Vol. 7858, pp 77-88, 2013.
4. C-S Tan and al., "Collaboration of IMS and SDN to Enable New ICT". In Network Operations and Management Symposium (APNOMS), pp 4-8,16th Asia-Pacific, 2014.
5. S. Dwarakanathan and al., "Cloud Application HA using SDN to ensure QoS". In IEEE 8th International Conference on Cloud Computing, New York, 2015.
6. H. Egilmez and al., "OpenQos: An OpenFlow Controller Design for Multimedia Delivery with EndtoEnd Quality of Service over Software-Defined Networks". In Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), Asia-Pacific, 2013.
7. S. Sharma and al.,"Implementing Quality of Service for the Software Defined Networking Enabled Future Internet". In Third European Workshop on Software Defined Networks, Budapest, 2014.
8. 8. R.Wallner and R. Cannistra, "An SDN Approach: Quality of Service using Big Switch’s Floodlight Open-source Controller". Proceedings of the Asia-Pacific Advanced Network, v. 35, p. 14-19, 2013.
9. B. Nunes and al., "A survey of software-defined networking: Past, present, and future of programmable networks. In IEEE Communications Surveys & Tutorials", vol. 16, no. 3, pp. 1617-1634, Aug. 2014.
10. S. Khairi, B. Raouyane and M. Bellafkih., "Towards a Dynamic QoS management in the IMS Transport Layer via SDN", In IEEE/ACS AICCSA, November 2017.
11. M. F. Bari and al., "Dynamic controller provisioning in software defined network ". In International Conference on Network and Service Management, pp 188- May 2013.
12. Floodlight home page, https://floodlight.atlassian.net/wiki/display/floodlightcontroller/
13. Mininet home page http://mininet.org/walkthrough/
14. OpenIMSCore home page, htp://www.ims-way.com/2011/04/open-ims- first-install/
15. Uct ims client home page, https://yulexs.wordpress.com/tag/uct/
16. A. Khan, L. Sun, E. Ifeachor, J. Fajardo and F. Liberal, “Video quality prediction model for H.264 video over UMTS networks and their application in mobile video streaming ”, IEEE ICC, Cape Town, South Africa, 23-27 May 2010.