

Proposal for a Feature Automation Solution for an IMS-KMS-IoT Platform based on SDN

Samba DIOUF¹, Kéba GUEYE, Samuel OUYA², Gervais MENDY³
Laboratory LIRT, Higher Polytechnic School, University Cheikh Anta Diop
Dakar, Sénégal

Abstract—The concept of the Internet of Things is a paradigm that is gaining more and more ground and soon the number of connected objects will be counted in billions. This will transform our lives and pose new challenges. To meet the challenges of objects several platforms have been proposed by previous work. Some are based on an IMS-IoT platform and others integrate IMS, IoT and SDN technologies. Our article proposes an architecture that integrates SDN into an IMS-IoT platform and the automation of the control layer (IMS-IoT platform) and the transport layer of the functional architecture in order to meet future requirements related to the configuration of these many objects, mobility and diversity of user terminals (smartphone, tablet, computers, etc.). The proposed architecture makes it possible to benefit from all the simplicity and efficiency provided by an SDN network but also the services offered by the IMS-IoT platform.

Keywords—IMS; IoT; SDN; automation; KMS

I. INTRODUCTION

The concept of the Internet of Things (IoT) incorporates the vision of the ubiquitous virtual connectivity of billions of physical objects or things across a global network infrastructure with interoperable, self-configurable and scalable capabilities. Objects can be part of various application domains and are therefore represented by different types of devices that have heterogeneous technical parameters and communication capabilities. This imposes various technical challenges in terms of adaptation, context awareness, device management discoveries, scalability, large data management, privacy and security.

In addition, the main asset of the IoT idea is its considerable impact on many aspects of everyday life and the behaviour of potential users. The most obvious effects of the Internet of Things, observable by a private user, would be visible in both the home and the workplace. In the first case, some examples of possible application scenarios in which the new paradigm, the Internet of Things, will play a prominent role in the near future are home automation, online health, life support autonomous and improved learning [1], [2]. In the second case, business users have observed similar consequences that can be detected in certain areas such as logistics, intelligent transport of people and goods, automation and industrial manufacturing, as well as business process management.

The "Building the Environment of Things as a Service" (BETaaS) paradigm is the result of a successful research initiative [3] that takes advantage of computer fog and

introduces the innovative concept of Things as a Service (TaaS) into the medium M2M applications. Computer fog [4] was introduced as a concept that reduces the amount of data to be transported to the cloud and reduces the burden of processing cloud systems by implementing data processing, storage, analysis and the edge of the network. BETaaS has extended the computer fog paradigm by introducing a local cloud of M2M gateways that aggregate and analyse data from different heterogeneous IoT devices and expose unified service-oriented interfaces that can be used to access objects connected to the local cloud, thereby introducing the concept of TaaS. This approach offers distinct advantages such as reducing network complexity, reducing transported data, generated by things, and seamless service-oriented interfaces that provide access to different types of devices with technical parameters and capabilities. heterogeneous communication. However, to fully exploit the benefits of exposed IoT services, an effective mechanism for deploying, managing, and delivering end-user services is required.

This article discusses the main challenges of integrating Software Defined Networking (SDN) into an IMS-IoT platform and automating control and transport layers of data collected by objects. IP Multimedia Subsystem (IMS) is a technology that manages session establishment, management, and termination through Session Initiation Protocol (SIP) to enable the deployment of services across all IP networks. IMS provides a common IP interface that simplifies signalling and application development and facilitates the processing of multiple sessions that can be used by different applications. In addition, it enables access control and billing management, which enables consistent management of different applications on the network [5, 6]. The benefits of introducing IMS into the IoT domain have already been examined by several research teams. A cloud-based implementation of the generic framework is described later in [7], where the core IMS components are virtualized in a resource pool and provided as Infrastructure as a Service (IaaS). The framework concept is further extended in [8], where an object management and naming mechanism based on IMS uniform resource identifiers (URIs) is proposed. The article presented a name distribution agent that used SIP messages "MESSAGE" to assign specific URIs to "things". In [9], the integration of IoT devices into a SIPIMS platform is considered in the context of WLAN-3GPP networks. Different SIP message flows are designed to provide different IoT functions, such as recording, streaming, and notifications, based on request / report messages. Finally, in [10], an operator-class IoT service-oriented architecture that integrates IMS with wireless sensor networks (WSN) is

prototyped. The architecture uses the Sensor Web Enablement (SWE) framework to solve the heterogeneity of IoT devices by introducing two main components: a facilitator who introduces IoT services on the telecom operating platform and a gateway that interconnects the IoT device, activator and the WSN. As proof of concept, the proposed design was implemented in an application scenario with 20 WSN nodes and the network load was evaluated, showing that the introduction of the M2M gateway effectively reduces the data traffic generated.

This document proposes an architecture for the provision of IoT services such as e-health, e-learning, etc. on a network of all-IP telecommunications service providers, while facilitating access to services and terminal diversification. We are taking a step closer to integrating SDN into an IoT-IMS platform by introducing the automation of core elements to address the challenges of increased traffic and network complexity. SDN is defined by the Open Network Foundation as a technology where "the control of the network is decoupled from the transfer and is directly programmable".

Previously related works had proposed an implementation of various software modules that can have a dynamic control over network functions, to support network expansion and innovation and for more efficient use of network resources [11]. One of the key benefits of SDN is thought to be its ability to efficiently route traffic flows, making it easier to support Quality of Service (QoS) differentiation. In [12], a converged SDN and IMS framework is presented, which incorporates an Open Flow controller that communicates with the IMS application servers via RESTful programmable interfaces (APIs). This allows for application-based QoS classification and data flow routing. The integration of Open Flow into IMS is also proposed in [13], where an SDN IMS research benchmark is designed.

This paper is structured as follows: Section II discusses the state of the art IMS (OpenIMSCore), SDN and IoT technologies. Section III discusses the composition and characteristics of the proposed architecture across its different layers. Section IV shows the results obtained with the automation of the control and transport layers and an analysis of these results. Finally, Section V conclusion and future work.

II. STATE OF THE ART

A. OpenIMSCore

OpenIMSCore is the implementation of an IMS core network based on the open source solution SIP Express Router (SER). The main elements of the layer are: Home Subscriber Servers (HSS) and different Call Control Session Control (CSCF) servers. The control layer connects to the application servers (AS) in the service layer and to the user equipment (UE) via the transport layer. The CSCF-Proxy (P-CSCF) acts as the point of contact with the UE for access to the main network service; Examination-CSCF (I-CSCF) [12] [13]. See Figure 1.

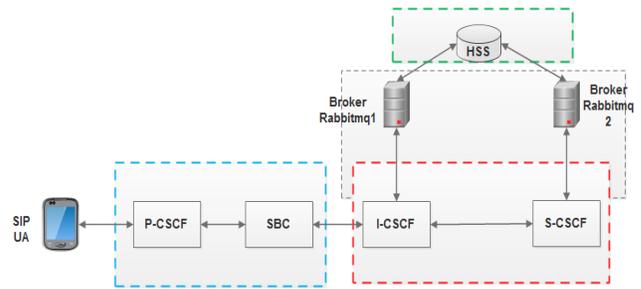


Fig. 1. IMS Architecture with Rabbitmq Servers at HSS.

- The P-CSCF is the point of entry into the IMS network for all signalling of a user's session. It is the point of contact between the user and the central server.
- The I-CSCF constitutes, for a session of a user, the entry point into the network of the operator to which the user has subscribed a service contract (SLA). It acts as an intermediary between the proxy server and the authentication server. It redirects a user's requests to his original operator. The I-CSCF entity assigns an S-CSCF entity to the UE and transfers the request to the selected S-CSCF entity. To perform this function, a DIAMETER exchange with the HSS entity is necessary. It also generates the information needed to generate tax tickets.
- The S-CSCF entity is responsible for authenticating of the user and providing the operational services. It provides routing information, billing information, and maintains session status by controlling a timer (time), querying the HSS to verify user rights to a service. At the same time, the S-CSCF stores the subscriber's position in the HSS in the network and tells the HSS its address so that an entity seeking to reach the subscriber knows which S-CSCF to contact.
- The HSS entity is a database that stores the data specific to each user. The main recorded data include among others
 - a. SVP Table (SerVice Profile): Contains the service profile that is associated with each user.
 - b. The public and private identities of the user (IMSU, IMPU and IMPI) along with the other authentication and authorization information.
 - c. Charging information table (Chginfo): Contains pricing information.
 - d. Application Server Table (APSVR): Contains the address of the AS to contact if the trigger points are compliant.
 - e. Initial Filter Criteria table (IFC); which contains the initial filtering information, characterizing a certain service for the user. This table maps a point trigger that associates a point service set with an application server.

- The RabbitMQ: is a database server that interacts with the I-CSCF, S-CSCF and HSS entities to complete the authentication process before a user is allowed to log on.

B. Software Defined Networking (SDN)

Software Defined Networking (SDN) is a new network paradigm that decouples the data plane and control plane. The SDN is based on the principle of providing a programming interface (API) for data packet transfer devices such as switches and routers, allowing the programming of network elements and the entire network, creating an architecture dynamic, flexible, automated and manageable [14] [15]. The main idea of the SDN is to deport the intelligence of the network in a fundamental element called controller. This controller manages elements of the data plane through APIs [16]. There are several types of SDN controllers. The most used in research and education are POX, OpenDaylight, FloodLight and Ryu [16] [17]. The SDN architecture is structured in three layers that interact with each other via APIs. We have the infrastructure layer or transmission layer. Its main role is to transmit data, monitor local information and collect statistics. We also have the control layer. This layer contains the controller (s) that uses Southern APIs to interact with the infrastructure layer. It uses the north APIs to communicate with the last layer called application layer. The different controllers use East / West APIs to interact with each other [18]. The SDN architecture is as shown in Figure 2:

SDN use the Application Interface (API) for communication between different entities. The development of the APIs allowed the programming of the SDN networks. Among these APIs there is Openflow. Openflow is an open protocol that allows the separation of control plans and data. It acts between the applications and the network. Thus, it allows to have a direct access to the data plane. It is also responsible for managing network flows such as routing, marking, filtering, etc. The OpenFlow protocol is an important, even fundamental element in the design of SDN solutions. It allows equipment and controllers to interact [11]. This approach as presented by the NFB involves deploying routers and switches compatible with OpenFlow protocol and SDN controllers to manage network flows on these devices.

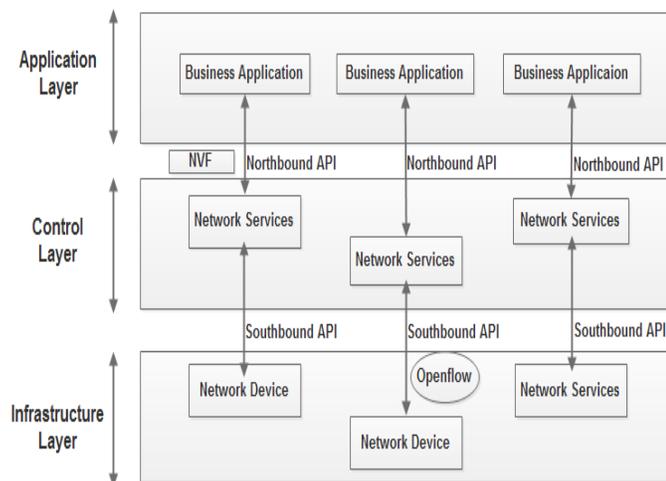


Fig. 2. SDN Architecture.

An SDN network makes it possible to reduce CAPEX and OPEX [10] and can be emulated thanks to a tool like mininet.

For academic purposes, the simplest approach to most features is to use Mininet. It can be easily distributed and deployed due to its packaging in a virtual machine. It also supports multiple controllers and others can be easily added. In addition, it does not have excessive material requirements [16] [18]. For these reasons, we have chosen it as a main tool for our integration of SDN into other platforms. This is what justifies the choice of this simulator.

C. Kurento Media Server (KMS)

Traditional WebRTC applications are standardized so that browsers can communicate directly without the mediation of third-party infrastructures. This is sufficient to provide basic multimedia services, but features such as group communications, stream recording, streaming, or transcoding are difficult to implement. For this reason, the most interesting applications require the use of a multimedia server.

Kurento is an open source WebRTC multimedia server. It makes possible the creation of media processing applications based on the concept of pipelines. Media pipelines are created by interconnect modules called Media Elements. Each Media Element provides a specific feature. KMS contains Media Elements capable of recording and mixing streams, computer vision, etc. Kurento Media Server offers the capabilities of creating media pipelines through a simple JSON-RPC-based network protocol. However, to further simplify developer work, a client API that implements this protocol and directly leverages Media Elements and pipelines is provided. Currently, the Java and JavaScript client API is ready for developers [19]. Taking into account the integrated modules, the Kurento Toolkit is detailed in Figure 3.

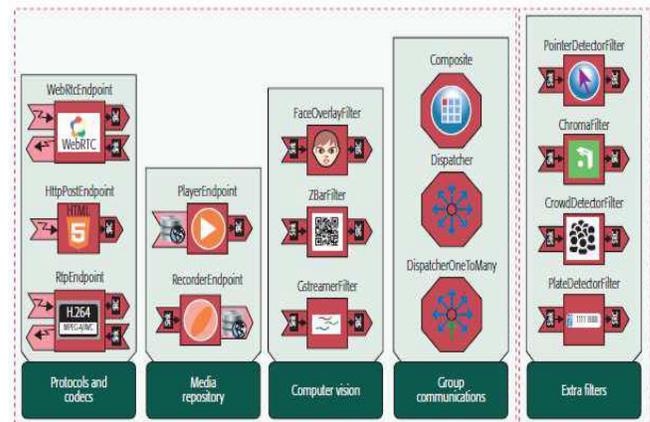


Fig. 3. Kurento Media Elements Toolbox.

D. Web of Things (WoT)

The Web of Things (WoT) is a specialization of the Internet of Things (IoT). On the one hand, it provides an abstraction of the connectivity of smart objects. On the other hand, WoT adds a standard web standards-based application layer to simplify the creation of IoT applications. In IoT, the communication protocols are multiple (MQTT, CoAP, AMQP ...), which creates groups of users.

The main interests of using WoT instead of IoT are the simplicity of development using APIs, standardization and simple coupling. The idea is that all intelligent objects can communicate using a Web language through an API. This API can be present in the intelligent object itself or in an intermediary that can act on behalf of the intelligent object [20].

In the field of research, each of these technologies is currently gaining momentum by proposing innovative solutions. Thus, each technology has developed independently from the others but thanks to IP convergence several technologies are integrating in order to diversify offers and improve the quality of service. Today, many researchers are taking advantage of IP to offer solutions that combine two of these technologies. It is in this case that the work of [21] deals with issues and proposes possible solutions for the Internet of the future through an IoT-IMS communication platform. By coupling SDN and IMS, the authors of [22] set up architecture for a QoS-compliant application in the virtualized IMS (V-IMS) environment. In the same vein [23] is putting in place an integrated ICT service creation platform, the IMS-based ICT service infrastructure is being built and some prototype services are being implemented. Added to this is a converged IMS and SDN ICT service infrastructure that is also presented. The possible use cases of the SDN on the creation of new services and the control of IMS service are also proposed. For some time, the IMS-IoT-SDN integration has been proposed as architecture to give access to information to several types of terminals. In this sense, the authors of [11] have set up a BETaaS M2M development and execution platform and an SDN-based IMS architecture to provide an open environment for the development and delivery of IoT services. All these works have the merit of proposing innovative solutions even revolutionary and constitute an important contribution in the field. However, these solutions can be improved by the automation techniques provided by the SDN. Our article, inspired by this work, proposes the integration of an SDN network into an IMS / IoT platform in order to facilitate access to users and to diversify user terminals. In addition, we also offer the automation of the control layer (IMS / KMS platform) and the transport layer of the functional architecture.

III. PROPOSED ARCHITECTURE

The functional architecture of the proposed solution is the integration of SDN technology into an IMS-IoT platform. This integration makes it easier for users to access the IMS-IoT network but also to diversify terminals (smartphone, tablet, computer, etc.). Figure 4 represents the architecture that we propose:

A. Access Layer

The access layer consists essentially of access network, terminals, gateway and IoT devices. IMS provides a standardized framework for creating and deploying services across all IP networks, regardless of access technology, supporting mobile and wireless telecommunications environments. In the IoT device, we have an IoT gateway to interact with the IMS / KMS platform. This gateway is the equipment responsible for retrieving the data collected by the IoT devices and transmitting them to the access network so

that the terminals can access them. The gateway plays an important role in the integration of SDN and IoT technologies. It can support several types of devices through a pilot architecture [11]. It can also be characterized by its representation of resources in different formats to facilitate its integration with other web applications [12].

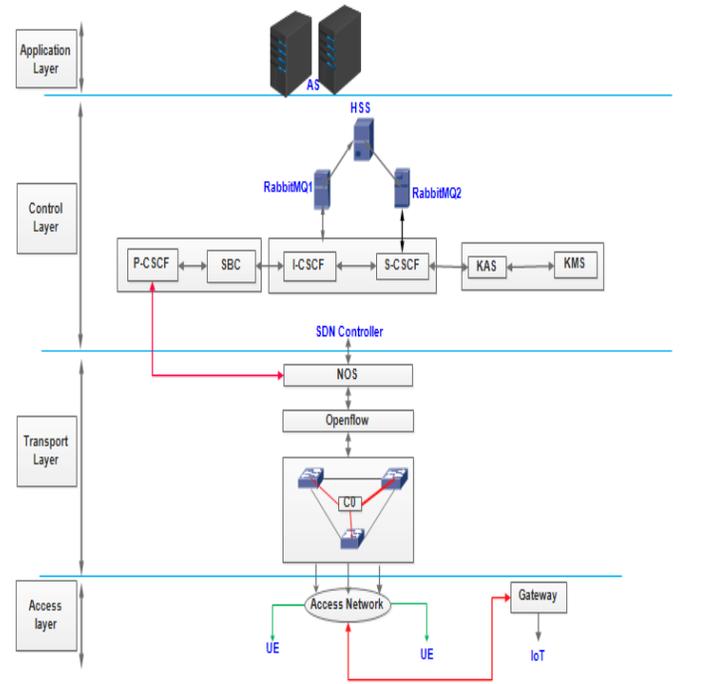


Fig. 4. Proposed Functional Architecture.

B. Transport Layer

The transport layer consists of the SDN part. It is composed of NOS which represents the operating system, different Switch, a controller responsible for controlling the equipment. The SDN concept decouples the control plane and data plane by concentrating data transfer decisions into a central controller (software). A common standard protocol that defines the communication between the control and the data transfer plan is Openflow. In the current architecture, the transport layer corresponds directly to the data transport plan of the SDN network. This plan consists of Openflow switches. Each Openflow switch contains one or more flow tables and an abstraction layer that communicates with the controller.

C. Control Layer

The control layer consists of a condensed IMS and KMS entity. The IMS part consists of P-CSCF, SBC, I-CSCF, S-CSCF, rabbitmq servers and HSS. In the KMS part we have the KMS and KAS entities. These two parts allow us to manage the IMS and WebRTC services for each one. In the SDN scenario, the data transmission decisions and the higher level control functions are executed by a logically centralized POX software controller. The network configuration is performed on a simplified network abstraction and the data flow configurations are communicated to the data routing devices via the Openflow protocol. The control plan is implemented using the SDN (NOS) network operating system

and a set of software modules that perform the various network control functions. The NOS exposes open APIs (North APIs) to enable the development of various applications that can interact and provide better network management. In such a scenario, the basic IMS / KMS elements can be implemented as modules for communication with the NOS via the northern APIs.

D. Application Layer

The IMS / KMS control layer uses multiple interfaces to communicate with service layer that provide the capability to deliver applications implemented by Application Servers (AS). The latter communicate with the S-CSCF using the SIP protocol to perform the session control service. They also communicate with the HSS via the DIAMETER protocol to retrieve or update the subscriber's profile information. AS are the most common type of application servers used in IMS scenarios. They support SIP applications and respond to requests with SIP messages to enable subsequent processing of user requests to S-CSCF. SIP applications can include presence service or sessions involving multimedia elements such as voice, video, chat, games, and so on.

IV. RESULTS AND DISCUSSION

To illustrate the functionality of the proposed architecture and the automation of the control layer and the transport layer we can refer to the following figures 5, 6 and 7. These figures show all the tests we have done. To do these tests we used a mininet simulator. Mininet is an open source network emulator that can run virtual hosts, switches, the controller (s) and virtual machine, or deployed on real hardware. For our case, mininet has enabled us to emulate the different entities of the IMS / KMS platform thanks to the automation of the two intermediate layers (control and transport). Thus, Figures 5 and 6 show the launch of the program via the python automation.py command. This program supports the automation of the control layer and the transport layer. The execution of such a program is as follows: the terminal creates the network and adds the controller whose role is to control the various equipments network Then there is the creation of hosts, switches and links between equipments. It is only after this step that the system starts the network devices (controller, switches, CLI, etc.) just after the computers are set up. If the process proceeds correctly the terminal displays mininet.

Figure 7 highlights the communication tests of the automation of the IMS / KMS platform. By doing a global pingall test we notice an exchange of flows between the different entities.

```
root@sambademo-VirtualBox:~/mininet# python automatisation.py
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
HSS ICSCF KAS KMS PCSCF Rabit1 Rabit2 SBC SCSCF
*** Adding switches:
s1 s2 s3
*** Adding links:
(10.00Mbit 10ms delay 0% loss) (10.00Mbit 10ms delay 0% loss)
```

Fig. 5. First Part of the Program Launch.

```
*** Starting 3 switches
s1 s2 s3 ... (10.00Mbit 10ms delay 0% loss) (10.00Mbit 10ms delay
y 0% loss) (10.00Mbit 10ms delay 0% loss) (10.00Mbit 10ms delay
0% loss) (10.00Mbit 10ms delay 0% loss) (10.00Mbit 10ms delay
0% loss) (10.00Mbit 10ms delay 0% loss)
la connexion des hotes
HSS HSS-eth0:Rabit1-eth1 HSS-eth1:Rabit2-eth1
ICSCF ICSCF-eth0:SBC-eth1 ICSCF-eth1:SCSCF-eth0 ICSCF-eth2:Rab
t1-eth0
KAS KAS-eth0:SCSCF-eth1 KAS-eth1:KMS-eth0
KMS KMS-eth0:KAS-eth1
PCSCF PCSCF-eth0:s1-eth3 PCSCF-eth1:SBC-eth0
Rabit1 Rabit1-eth0:ICSCF-eth2 Rabit1-eth1:HSS-eth0
Rabit2 Rabit2-eth0:SCSCF-eth2 Rabit2-eth1:HSS-eth1
SBC SBC-eth0:PCSCF-eth1 SBC-eth1:ICSCF-eth0
SCSCF SCSCF-eth0:ICSCF-eth1 SCSCF-eth1:KAS-eth0 SCSCF-eth2:Rab
t2-eth0
*** Starting CLI:
mininet>
```

Fig. 6. Launch of the Program Continuation and End.

```
mininet> pingall
*** Ping: testing ping reachability
HSS -> ICSCF KAS KMS PCSCF Rabit1 Rabit2 SBC SCSCF
ICSCF -> HSS KAS KMS PCSCF Rabit1 Rabit2 SBC SCSCF
KAS -> HSS ICSCF KMS PCSCF Rabit1 Rabit2 SBC SCSCF
KMS -> HSS ICSCF KAS PCSCF Rabit1 Rabit2 SBC SCSCF
PCSCF -> HSS ICSCF KAS KMS Rabit1 Rabit2 SBC SCSCF
Rabit1 -> HSS ICSCF KAS KMS PCSCF Rabit2 SBC SCSCF
Rabit2 -> HSS ICSCF KAS KMS PCSCF Rabit1 SBC SCSCF
SBC -> HSS ICSCF KAS KMS PCSCF Rabit1 Rabit2 SCSCF
SCSCF -> HSS ICSCF KAS KMS PCSCF Rabit1 Rabit2 SBC
*** Results: 0% dropped (72/72 received)
mininet>
```

Fig. 7. IMS / KMS Platform Automation Communication Test.

Based on these tests we can say that this proposal is an important contribution to the management of data collected by IoT devices. In that it offers users a lot of ease and diversifies the terminals. It also automates the IMS-KMS platform. This automation is more than important. Indeed, when a component relies on manual processes then it can become a bottleneck. Utilizing administrative tools and data center switches that enable high levels of automation allows IT departments to eliminate such bottlenecks. At the same time, network automation helps to increase agility and reliability, accelerate deployments, and improve the performance of mission-critical business applications.

V. CONCLUSION

Our paper proposes a functional architecture for an IMS-IoT platform, the integration of the SDN and the automation of the transport layer and the control layer of this architecture in order to facilitate the management, the access to the users, to diversify the user terminals etc. The technologies used to do this work are described as well as the essential components of the architecture without forgetting the different challenges posed by these technologies.

This architecture is for us the center of future work that can be focused on the study and load balancing of the various entities and the virtualization of some network functions.

ACKNOWLEDGMENT

The authors would like to thank the LIRT Laboratory Manager for the help he provided during the research.

REFERENCES

- [1] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications", IEEE Journal on Communications Magazine, vol. 53, n°9, 2015, pp. 48-54.
- [2] N. Omnes, M. Bouillon, G. Fromentaux, and O. Le Grand, "A programmable and virtualized network & it infrastructure for the internet of things: How can nfvsdn help for facing the upcoming challenges", Proc. IEEE ICIN, 2015, pp. 64-69
- [3] C. Vallati, E. Mingozzi, G. Tanganelli, N. Buonaccorsi, N. Valdambri, N. Zonidis, B. Martnez, A. Mamelli, D. Sommaccampagna, B.

- Anggorojati, S. Kyriazakos, N. Prasad, F. Nieto, and O. Rodriguez, "Betaas: A platform for development and execution of machine-to-machine applications in the internet of things," WPC, pp. 1–21, 2015.
- [4] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2014, pp. 1–8.
- [5] "IMS Tutorial | IP Multimedia Subsystem | Radio-Electronics.Com." [Online]. Available: http://www.radioelectronics.com/info/telecommunications_networks/ims-ip-multimediasubsystem/tutorial-basics.php. [Accessed: 07-Jul-2015].
- [6] Katov, Anton Nikolaev; Anggorojati, Bayu Kyriazakos, Sofoklis; Mihovska, Albena Dimitrova; Prasad, Neeli R. 18th International Symposium on Wireless Personal Multimedia Communications, 2015 (WPMC)
- [7] H.Liao, K. Chang, C.Chen, J. Chen, H.Chao, T.Wu, "IoTIMS Communication Platform for Future Internet Implementation," The 11th Conference on Information Technology and Applications in Outlying Islands, Taiwan, May 25-26, 2012.
- [8] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Favoriser l'innovation dans les réseaux de campus. *ACM SIGCOMM Comput. Commun. Rev.* 2008, 38, 69-74. [Google Scholar] [CrossRef]
- [9] OpenFlow Switch Specification v1.1.0. Disponible en ligne: <https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf> (consulté le 2 avril 2018).
- [10] CAPEX OPEX Available: <http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf>
- [11] C. Tranoris, S. Denazis, N. Mouratidis, P. Dowling, and J. Tynan, "Integrating OpenFlow in IMS Networks and Enabling for Future Internet Research and Experimentation," in *The Future Internet*, A. Galis and A. Gavras, Eds. Springer Berlin Heidelberg, 2013, pp. 77–88.
- [12] Imène ELLOUM I, 2012. "Management of mobility between access networks and Quality of Service in an NGN / IMS approach." Thèse doctorat : Université de Caen (France)
- [13] N. KAG-TEUBE, Y. P. K. DODOAGNEN, S. OUYA, K. GUEYE, "Proposed solution for improving the reliability of HSS data by integrating a queue manager", international conference on advanced communication technology (ICACT), february 11-14 2018
- [14] Faris Ketli, "Computer networks virtualization cost effective and distance learning achievement", Journal of University of Duhok, Vol. 20, No.1 (Pure and Eng. Sciences), Pp 90-98, 2017 eISSN: 2521-4861 & pISSN: 1812-7568
- [15] Guohui Wang, T. S. Eugene Ng, Anees Shaikh, "Programming your network at run-time for big data applications", HotSDN'12, Helsinki, Finland, August 13, 2012.
- [16] Filip Holik, Josef Horalek, "Implementing SDN into Computer Network Lessons", Journal of Telecommunication, Electronic and Computer Engineering, Vol. 9 No. 1-3
- [17] Enns E et al (2011). "Network Configuration Protocol (NETCONF), RFC 6241". Retrieved from: <https://tools.ietf.org/html/rfc6241>, on 8 Jan 2016
- [18] Rogério Leão Santos de Oliveira, Using Mininet for Emulation and Prototyping Software-Defined Networks, 978-1-4799-4340-1/14/\$31.00 ©2014 IEEE
- [19] B. Garcia, L. Lopez-Fernandez, M. Gallago and F. Gortazar, « Kurento : The Swiss Army Knife of WebRTC Media Servers », in IEEE Communications Standards Magazine, vol. 1, no. 2, pp. 44-51, 2017
- [20] D. Guinard and V. Trifa, Building the web of Things. Manning Publications Co, 2016
- [21] C.-Y. Chen, H.-C. Chao, T.-Y. Wu, C.-I. Fan, J.-L. Chen, Y.-S. Chen and J.-M. Hsu, "IoT-IMS Communication Platform for Future Internet," International Journal of Adaptive, Resilient and Autonomic Systems, vol. 2, no. 4, pp. 74–94, 34 2011.
- [22] HM Yoon, Z Liu, WS Yang, JH Kim, JO Lee, "Architecture for QoS-enabled Application Service Deployment in Virtualized Environment", Advanced Science and Technology Letters Vol.142 (SIT 2016)
- [23] Chung-Shih Tang, Chin-Ywu Twu, Jen-Hong Ju, and Ying-Dian Tsou, "Collaboration of IMS and SDN to Enable New ICT Service Creation", Asia-Pacific Network Operation and Management Symposium (APNOMS) 2014