

A Virtualized Application Service for QoS Management

Hyun-Min Yoon, Zeqi Liu, Woo-Suk Yang, Jung-Ho Kim and Jae-Oh Lee

*Chungcheongnam-do dongnam-gu, Cheonan-si Byeongcheon-myeon Gajeon-ri
Korea University of Technology and Education, Korea
[yhm00700, fbxorrl, joos13, junggho32, jolee]@koreatech.ac.kr*

Abstract

IP Multimedia Subsystem (IMS) is a framework that provides multimedia service to standard UMT-based mobile user. The IMS requires the assurance of QoS (Quality of Service) for the network flow between users, but QoS of network resource depends on existing protocol scheme. In recent, the research on reliability and QoS of the multimedia service based on NFV/SDN is claimed for the deployment of 5G network environment. Therefore, the design and analysis of QoS-enabled application service deployment is needed for the accomplishment for resource allocation of network resources. In this paper, we propose the architecture for QoS-enabled application in the environment of Virtualized IMS (V-IMS).

Keywords: SDN, IMS, QoS, NFV, ONOS, SLS

1. Introduction

The current communication network environment is developing at a rapid pace. Internet users have exchanged peta-byte data over the network. A few years ago, storage, computing, and network resources were physically separate from each other, and even the resource management systems were physically separate. Virtualization is a technology that uses software technology to logically and physically aggregate hardware resources, abstracts the resources of computer resources from a computer, and integrates and separates the resources into a desired form. In other words, it can be defined as "a technology that hides the characteristics of physical computer resources from the way other systems, applications, and end users interact with resources." This creates a single physical resource, such as a server, operating system, application program, or storage device that appears to function as multiple logical resources [12]. In order to efficiently manage these resources, faster and better application is required. The IP Multimedia Subsystem (IMS), which is a wired/wireless integrated control network, has appeared, and users are increasingly using IMS services in a variety of terminals. However, network resources are limited and not infinite. Therefore, management of resources is getting more and more important. The Software Defined Network (SDN) and Network Function Virtualization (NFV) technologies allow software to control the network by separating the data plane and the control plane of the network. This is a technology that enables more flexible and inexpensive network infrastructures by applying virtualization technology to the commercial product base. The SDN controller is responsible for the configuration of physical resources. It also maintains the rules of the network and distributes network resources according to appropriate commands.

In this paper, we use QoS-enabled Intent Forwarding App and SLS Moderator to control traffic. And a new virtualization model that combines IMS with an SDN controller that flexibly handles this controlled traffic and efficiently uses network resources. The

Received (January 25, 2017), Review Result (May 11, 2017), Accepted (June 4, 2017)

remaining pages provide background inspections of related studies such as IMS, NFV / SDN, and ONOS [1].

2. Related Works

2.1. IMS

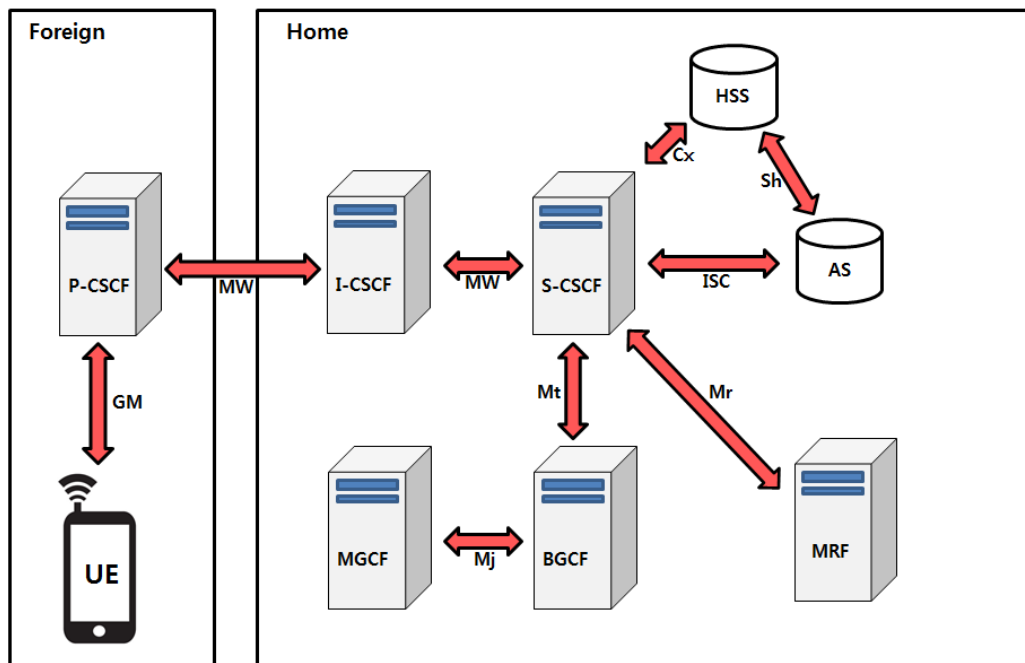


Figure 1. SDN/NFV-based Flexible Architecture

All The basic goal of IMS is to provide a complex combination of multimedia, such as voice, audio, video and data, based on the IP protocol. The P-CSCF in the IMS is a grass-landing point for IMS but not a starting point for users. This function accepts SIP messages and receives all validating SIP messages. It can perform forwarding the sending SIP message, maintaining an emergency session, compressing and decompressing all SIP messages as needed. It maintains user equipment security and approved QoS [2].

Figure 1 shows the Components of the IMS architecture that are relevant to our investigation are explained as follow:

- AS: As a server that contains the applications or services provided, the advantage of the IMS architecture is a mix of applications that can benefit in real time.
- Home Subscriber Server (HSS): HSS is a key data storage device for all subscriber and service related data of IMS. Key data stored in the HSS include user identity, registration information, access parameters, and service triggering information.
- Call Session Control Function (CSCF): It is the person in charge of the session and provides control of the session. The CSCF is divided into three components.
- P-CSCF (Proxy CSCF): The P-CSCF is a grass-landing point for IMS but not a starting point for users. This function accepts SIP messages and receives all validating SIP messages. It can perform forwarding the sending SIP message, maintaining an emergency session, compressing and decompressing all SIP messages as needed. It maintains user equipment security and approved QoS [2].
- Interrogating CSCF (I-CSCF): Receives the SIP message delivered from the P-CSCF and determines the S-CSCF assigned by the user. This is done by querying the user's information stored in the HSS.

- S-CSCF (Serving CSCF): CSCF is the brain of IMS and is in the home network. Performs session control and registration services for the UE. It is responsible for routing all SIP messages to their final destination. Among the functions, the UE is authenticated using the HSS.
- Interrogating CSCF (I-CSCF): Receives the SIP message delivered from the P-CSCF and determines the S-CSCF assigned by the user. This is done by querying the user's information stored in the HSS.
- S-CSCF (Serving CSCF): CSCF is the brain of IMS and is in the home network. Performs session control and registration services for the UE. It is responsible for routing all SIP messages to their final destination. Among the functions, the UE is authenticated using the HSS.

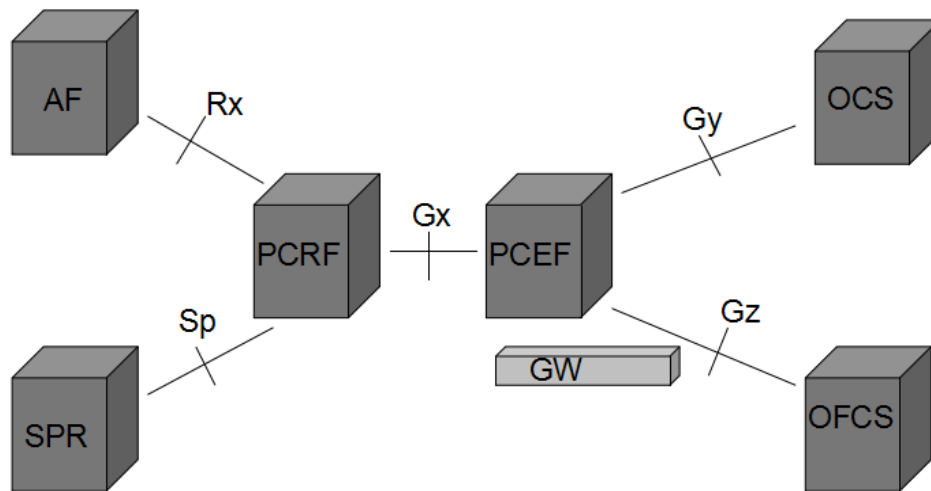


Figure 2. PCC Architecture of IMS

Figure 2 shows the PCC architecture of the IMS. It consists of the following elements:

- Policy and Charging Rules Function (PCRF) [11]: It is the point where policy decisions are made. In order to make decisions, the PCRF receives information from AFs and from the Subscription Profile Repository (SPR)
- Application Functions (AF): An AF (*i.e.*, a P-CSCF or an Application Server) provides the PCRF with information obtained from the session signalling (especially from the Session Description Protocol (SDP) in the offer/answer exchanges) over the Diameter-based Rx interface.
- SPR: The SPR provides the PCRF with QoS related information about the user's subscription over the Sp interface. The Sp interface has not been standardized yet but current deployments use Lightweight Directory Access Protocol (LDAP) or Diameter on that interface.
- Policy and Charging Enforcement Function (PCEF) [2]: The PCEF is a logical function within the gateway (*e.g.*, a General Packet Radio Service (GGSN)). PCEF enforces the decisions made by the PCRF. The PCRF communicates with the PCEF over the Diameter-based Gx interface.
- OCS: The OCS performs online credit control over Gy interface. That is, it can instruct the gateway to close a particular gate if the user runs out of credit. The PCEF communication with the OCS over the Diameter based Gy interface.
- Offline Charging System (OFCS): The OFCS obtains information about data flows to be used for charging purposes. The PCEF communications with the OFCS over the Diameter based Gz interface.

2.2. NFV

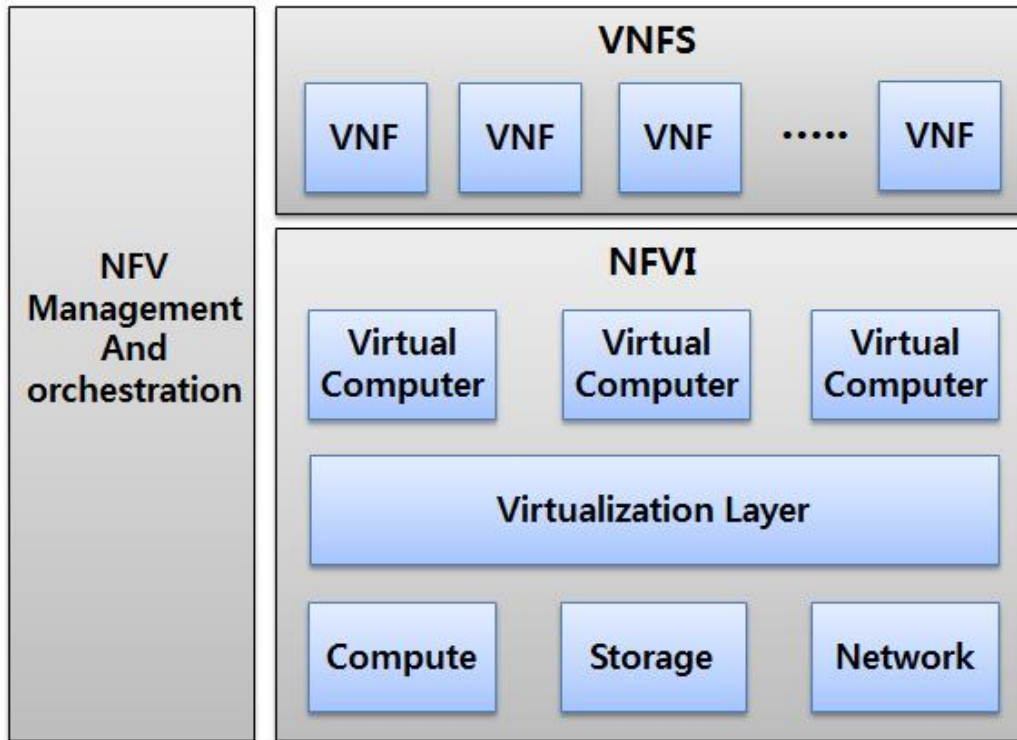


Figure 3. High-level NFV Framework

Network Function Virtualization (NFV) is the virtualization of core network functions. The NFV was initiated by network operators to facilitate the deployment of new network services by virtualizing networking devices and hardwares. The major telecommunications carriers and service providers around the world have created the Specification Group for NFV. The European Telecommunications Standards Institute (ETSI) is leading the way in creating technology standards and is currently in the Proof of Concept (PoC) stage. Through this, ETSI seeks to "define the requirements and architecture for virtualization of network functions". This purpose of Virtualization is to help reduce operational costs and shorten time to market for service components while delivering flexible, faster solutions from industry-standard server hardware components [4].

Virtualization removes the dependency between network function (NF) and hardware, creates a standardized execution environment as shown in typical network devices, and creates a management interface for virtualized network functions (VNFs) do. This is due to the fact that several VNFs share physical hardware in the form of a virtual machine (VM). Moreover, the phenomenon that has already been seen in cloud computing infrastructures, the pool of hardware enables large, agile sharing of NFV infrastructure resources by VNFs. This creates new business opportunities similar to the cloud computing service model, similar to Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Here, VNF owners do not necessarily own an NFV infrastructure that is functional enough to operate them. Once different and non-virtualized NFs are virtualized into VNFs, these VNFs need to be organized into ordered graphs to realize network services. In NFV, these graphs are called VNF Forwarding Graphs. "Forwarding Graph" is preferred over the word "Service Chain" to explain that end-to-end delivery within a virtualized overlay network is not entirely

organized into a one-dimensional chain. Service networks will have branches instead, and multidimensional concepts will be implied.

Ultimately, the Key point of NFV is the virtualization of IT resources in software. This eliminates the need for additional physical devices. And we can use virtualization technology to provide networking capabilities. Figure 3 shows the high-level NFV framework, there are three domains as we can see. Network functions are virtualized so that VNFs operate on a virtualized network infrastructure (NFVI). NFVI has various hardware, physical resources and the ability to virtualize these resources, and supports the execution of VNFs. Finally, NFV management and orchestration supports and manages the lifecycle of VNFs, the software resources that support infrastructure virtualization, the lifecycle of physical resources, and orchestration. NFV management and orchestration focuses on all aspects of virtualization that require specific management and operations in the NFV architecture [5].

But there are still challenges to be addressed by the NFV. For example, there are various technical difficulties in the NFV that must implement the packet processing function of the data plane in software. It must be able to move between different hardware vendors and between different hypervisors, and the virtualized network appliances must be high-performance. In addition, there must be a failover and recovery function in case of hardware and software failure [10].

2.3. ONOS

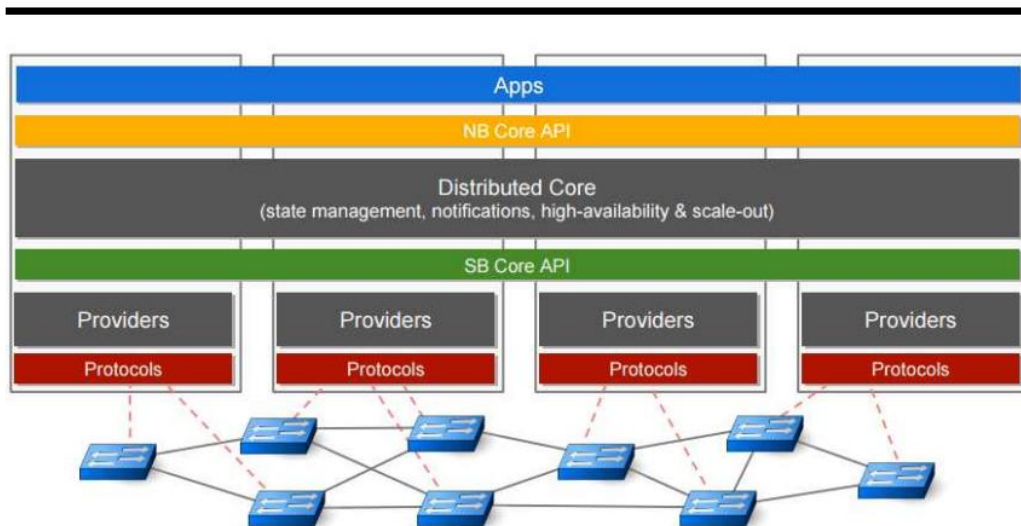


Figure 4. Distributed Controller Architecture of ONOS

ONOS is an abbreviation of Open Network Operating System. It is designed to overcome the problems of existing centralized SDN controller structure and to be suitable for actual large network environment. ONOS is designed to enhance the reliability, performance, and scale-out of the network. We have also created useful Northbound/Southbound concepts and interfaces with OpenFlow APIs for easy application development. The ONOS goal is largely three-fold: it uncovers the service provider's CAPEX and OPEX, and helps service providers move their current network to the white box and provide agile services [4].

Figure 4 is a conceptual diagram showing the structure of ONOS. As shown in Figure 4, the ONOS controller with distributed architecture controls the network switching devices at the bottom. The layer of ONOS consists of application, distributed core, provider, and protocol layer. The protocol layer is a layer for communicating with various network devices with the SDN function. In addition to OpenFlow, various SDN protocols

such as NetConf are defined. The provider layer is a layer that operates independently for each ONOS controller, and instructs each network device by using a protocol such as OpenFlow that defines the network commands requested from the upper layer in the protocol layer. On the contrary, it plays a role of informing information received from network devices to higher layers. The distributed core layer is a layer in which various functions for controlling and managing the SDN network are defined. Each ONOS controller is actually run independently on the network, but the distributed core layer is defined to be able to run as a single controller through the store of this layer. Finally, the application layer defines various applications to perform network control and management functions defined in the distributed core layer through a graphical user interface (GUI) and a command line interface (CLI). Meanwhile, SBI (South Bound Interface) and NBI (North Bound Interface) are defined for communication and call between the provider layer and the distributed core layer and between the distributed core layer and the application layer [6].

2.3.1. Distributed Core

Distributed Core provides carrier-class SDN control with scalability, high efficiency and performance. ONOS acts as a single system, bringing Web-style agility to SDN control in the service provider's network. Failover is also possible where the standby system is automatically shut down when the primary system is down. ONOS instances work together to look like one platform to the rest of the network and applications. ONOS scalability is ensured because applications and network devices do not need to know if they work with a single instance or multiple instances. Distributed Core provides communication, state manager, and leader election between instances. As a result, multiple instances can behave logically as one independent chain. The leader election service ensures that one switch puts one master instance. Communications, state managers, and leader election mechanisms also enable much of the cluster's throughput, low latency, and high efficiency.

2.3.2. Northbound Abstraction/APIs

There are two powerful northbound concepts that show the Internet framework and the global network. The Internet framework allows an application to request services from a network without knowing specifically how services are provided. This enables network providers to develop high-level applications. Developers can explicitly simplify their intentions. A global network View provides applications with status information about the network, such as hosts, switches, cables, and other utilities. An application can program this Network View via the API. One API has the ability to show network graphs. Technically, the northbound concept and API maintains a distance from the details of the network that an application does not need to know. Conversely, it allows an application to take care of multiple conflicting applications instead of an OS. From a business perspective, it is possible to speed application development and transform networks without application downtime.

2.3.3. Southbound Abstraction/APIs

The southbound concept is designed to use network elements such as switches, hosts, or cables. The Southbound of ONOS represents each network element as a generic entity. This concept does not require the Distributed Core to know the obvious elements represented by the underlying drivers of network elements. Thereby allowing the core to use the software or system regardless of the terminal and become a southbound protocol. The ONOS and Southbound concepts enable plug-ins of various southbound protocols and devices. Southbound thus allows ONOS to manage and manipulate many different devices using different protocols. At the heart of Southbound is the ability to maintain and

maintain network elements without having to know specifically about the protocol and its components. Through the adapter API, Distributed Core is able to keep the network elements up to date.

3. Architecture and Implementation Model

3.1. Architecture of SDN Controller

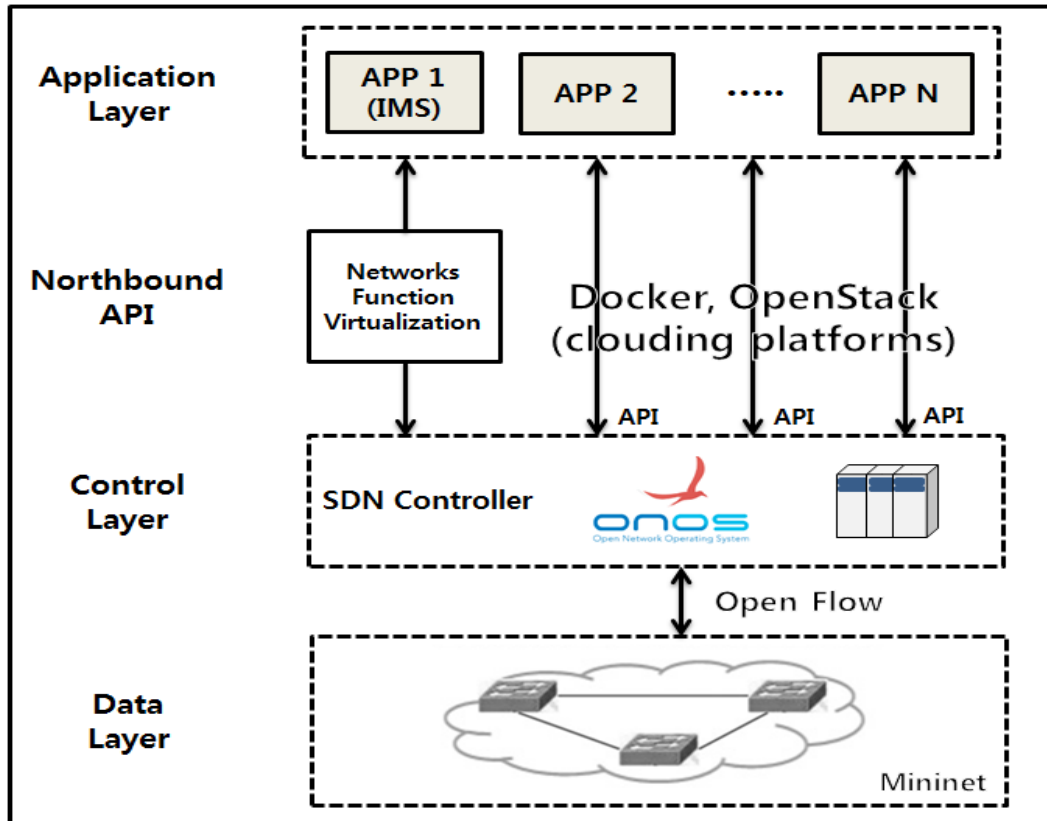


Figure 5. SDN/NFV-based Flexible Architecture

SDN is the logical separation of network control from network hardware. SDN began with a large-scale IP infrastructure to simplify traffic management and achieve operational efficiency by establishing and enforcing centralized management through packet delivery. Since then, SDN has evolved into an open networking environment in which components such as switches, servers, and storage are run on standard hardware and configured and managed centrally. The Open Networking Foundation (ONF), representative of SDN, is a single architecture that relies on the "control infrastructure" developed as an "accessible computing device," in which the underlying infrastructure is abstracted for application and network services and treated as a logical or virtual entity. This means that the SDN control hardware is located on top of the physical infrastructure layer that consists of networking devices and that it communicates via a control surface interface such as OpenFlow. The key point of SDN is to transform the network into a flexible, programmable platform to optimize resource utilization to make it more cost-effective and scalable.

Figure 5 shows a logical view of SDN architecture in OpenFlow. Information networks are logically centralized to software-based SDN controllers that maintain a global view of the network. As a result, networks are represented by logical switches, a single policy engine and applications. By using SDN, companies and forwarders can simplify network

design and operation, and operate the entire network from a single logical point in a vendor-independent manner. SDN can also operate network devices simply by utilizing SDN controllers without using one-to-one devices of thousands of protocol standards any more. The difference between NFV and SDN is as follows. NFV adds virtual tunnels and functions to the physical network, so it can be located on the server and run on the existing network. On the other hand, SDN requires a new network structure in which the data plane and the control plane are separated as a new external means for modifying, provisioning and managing physical networks [7].

3.2. Architecture of Virtualized IMS Platform

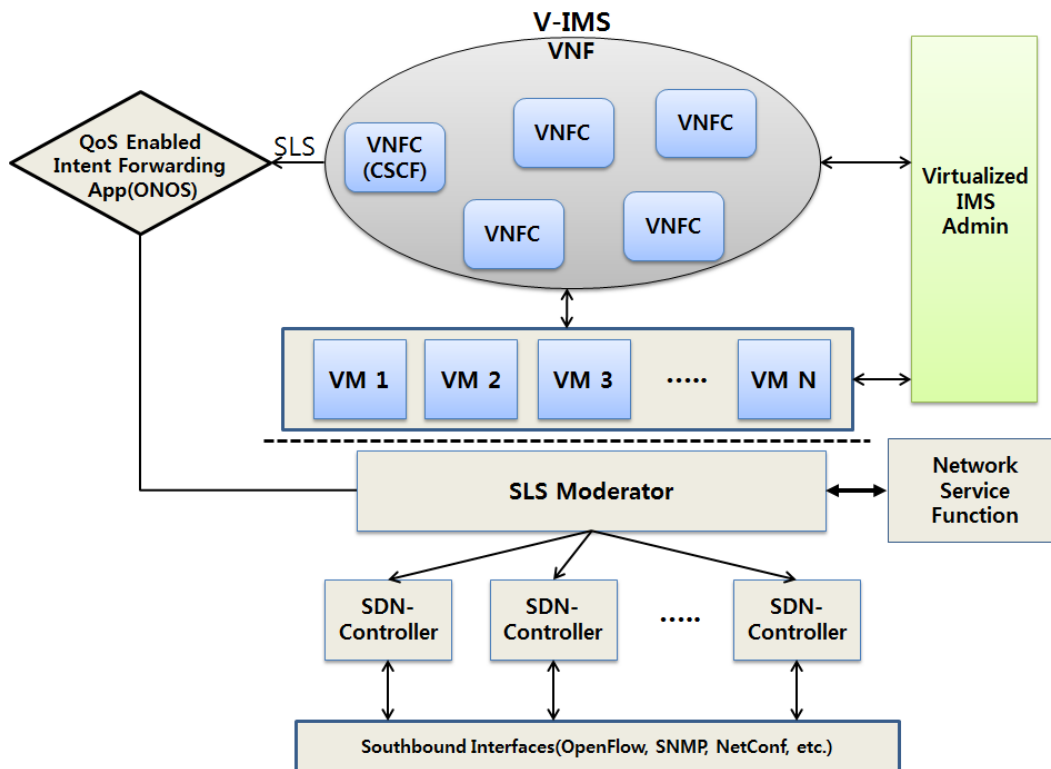


Figure 6. SDN/NFV-based Flexible Architecture

The platform provides a combination of several devices. The details are as follows.

(1) The IMS can perform their roles by virtualization, and management of each virtualized IMS (V-IMS) is performed by an administrator virtualized IMS.

(2) The virtualization manager performs creation, removal and configuration of a virtual machine (VM) and configuration of a new virtualization system. In addition, IMS can perform operations such as initialization configuration, generation, suspension / resumption, and termination of virtualized IMS through a VM and monitoring functions to understand the operation status of virtualized IMS.

(3) V-IMS-Admin consists of configuration management module and monitoring module of virtualized IMS and performs operation and monitoring through VM operating on virtualized IMS system.

(4) The virtualized structure supports the execution of multiple VMs at the same time, and monitors / manages / manages / VMs efficiently through the hypervisor.

(5) V-IMS sends SLS signal to QoS Enabled Intent Forwarding App. The deployed VNF for QoS Control is in charge of classifying the flow, and detecting its QoS class and compliance with the respective SLS. Following the classification, the DPI communicates with the several SDN controllers via the interface between SLS moderator and SDN

controllers set of information required to identify the flow (*e.g.*, Medium Access Control (MAC) and IP addresses, protocol, TCP/UDP ports) and its classification. As a result of the classification, each SDN controller changes the flow state of its own domain [6].

(6) In routing, media traffic such as video has different traffic performance due to various factors like delay. QoS Enabled Intent Forwarding App provides this traffic performance information to SLS Moderator.

(7) The SLS Moderator collects the traffic performance information and Forwarding it to the multiple SDN-controllers.

(8) The SDN controller performs basic functions for centralized network control such as forwarding control, topology and resource status management, and routing control based on the global view of the network status. In addition, differentiated forwarding and packet processing rules are determined according to the upper application or policy request and the forwarding rules are applied to the lower SDN switch boxes.

(9) The SDN controllers provide a variety of interfaces. These interfaces can be used to program, interrogate, and configure the controller's numerous network service functions, such as Forwarding Manager, Switch Manager, Slicing Manager, Topology Manager and Host Tracker.

The above description can be represented by an algorithm. The algorithm is shown in Figure 7.

3.2.1. Network Service Function

Forwarding Manager is responsible for forwarding multipath based on the OpenFlow technology and dynamically makes a decision about rerouting the network flow depending on the bandwidth traffic, in the mesh network Equal Cost Multi Path (ECMP), routing path planning is done.

The Slicing Manager of controllers is to alternate network slice, and to control network data via configuring the OpenFlow based on OpenVSwitches. Providing connectivity to other Application servers and connecting cloud resources is also the function of configuring the OpenFlow. Slicing Manager can also be used to perform traffic shaping and slice isolation.

Host Tracker and Topology Manager also can be used in this architecture. Host Tracker tracks the up/down status of network switches and their ports by listening to the asynchronous status messages exchanged between the OpenFlow controller and switches. The data collected by this application helps the management plane to maintain a global view of the network. Topology Manager collects data from the Host Tracker. It maintains a complete view of the network, which is used by the policy adaptation modules to make resource provisioning decisions.

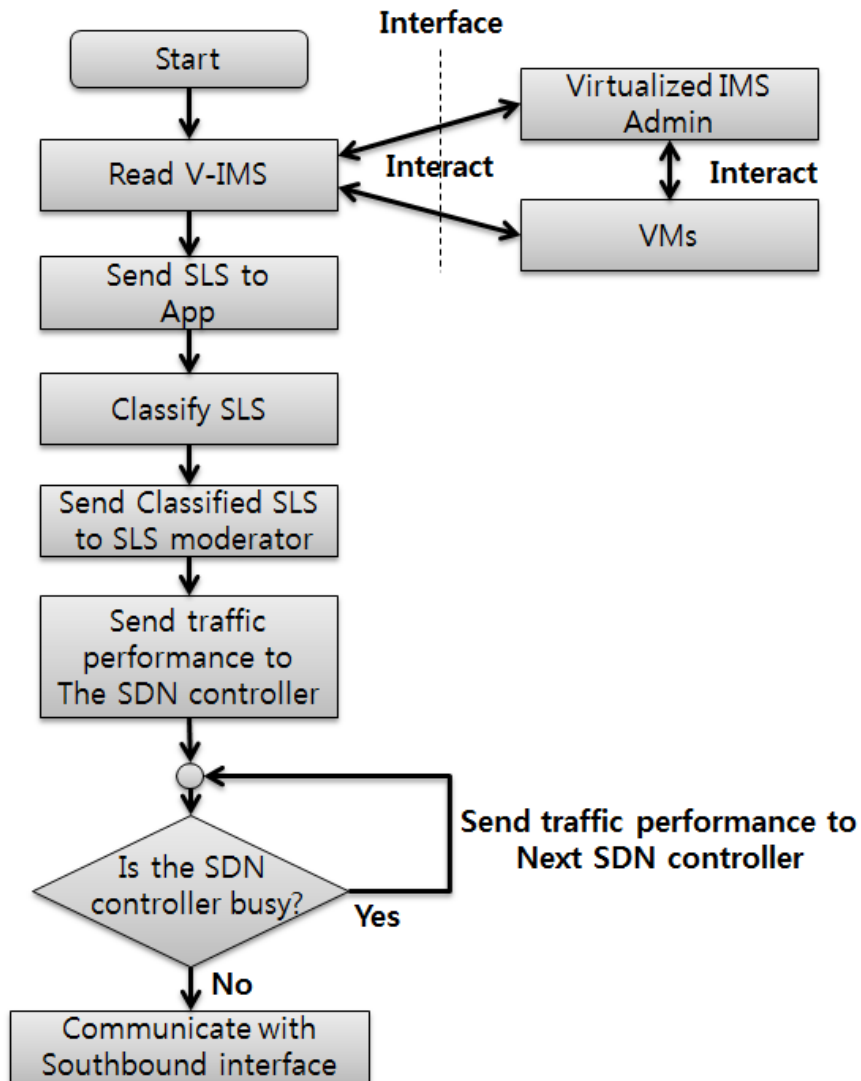
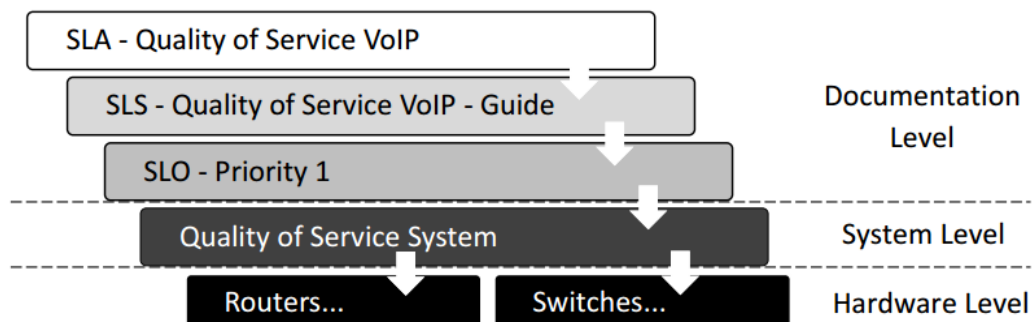


Figure 7. Algorithm of Implementation Model

3.2.2. QoS Enabled Intent Forwarding App and SLS Moderators



SLAs are generally business-oriented, and they leave aside the technical details, which are guided by a service level specification (SLS) and a service level objective (SLO). The SLS is a technical interpretation of the SLA. The SLO is a sub-item of the SLS that

contains the parameters to achieve the SLS [15]. Figure 8 presents an overview of the process of refining SLAs. The SLA, SLS and SLO represent, respectively, the documentation describing the service in a formal way, the technical form (guide) for its functioning requirements, and the parameters aimed at quality and satisfaction. At the system level, a quality of service system interprets the management requirements, and enforces policies for the configuration of elements at the hardware level.

SLAs are specified by means of Service Level Specifications (SLS) technically. A SLS is a set of parameters (throughput, delay, jitter *etc.*) and their values which together define the service offered to a traffic stream (*e.g.*, round-trip delay, throughput, packet loss probability, *etc.*). TEQUILA defined a set of semantics for SLSs as the basic component for defining QoS-based IP connectivity services [14]. A SLA may not include a service specification itself, but may contain a list of services the customer can request on demand [13]. The QoS Enabled Intent Forwarding App could communicate with the V-IMS for that user policies traffic. Resource provisioning and allocation can be implemented or may be adjusted dynamically on user SLS demand. The messages exchanged between these components could be presented here as follow:

- SLS Request: This message is the customer or service provider request for a service based on an SLS from a provider.
- SLS Response: This message answers the SLS Request message to indicate success/failure session. This may trigger further negotiation by PDF for QoS inter-domain connections.
- SLS Confirmation: This message indicates whether the requested SLS can be fulfilled or not. It may confirm or modifies some criteria.

4. Conclusion

Currently, network management methods have been proposed to make traffic management more efficient due to increased traffic overload. Native/Web client technology, cloud technology and NFV/SDN are combined technologies including interface to various IoT/E products. In a next-generation network environment, it is essential to develop a virtualization-based platform.

Therefore, this paper proposes a new virtualized IMS model by combining NFV/SDN. The core of this proposal is to classify the traffic using QoS Enabled Intent Forwarding App and SLS Moderator, which are created using ONOS. In addition, it will be possible to efficiently use network resources and structure by flexibly transmitting traffic classified into a plurality of combined SDN controllers. We will enhance QoS-enabled Intent Forwarding App and algorithms to use it and compare it with the existing models

Acknowledgments

This paper was (partially) supported by the Sabbatical Year Research Program of KOREATECH in 2017.

This paper is a revised and expanded version of a paper entitled Architecture for QoS-enabled Application Service Deployment in Virtualized Environment presented at Hyun-Min Yoon, Zeqi Liu, Woo-Suk Yang, Jung-Ho Kim and Jae-Oh Lee, Jeju Island and Dec. 23 of conference

References

- [1] ONOS Project, <http://onosproject.org/>.
- [2] V. P. J. Edison, J. H. Kim and J. O. Lee, "A 3GPP IP Multimedia Subsystem-Based System Architecture for a M2M Horizontal Service Platform", *Journal of Next Generation Information Technology*, (2003), pp. 299-311.

- [3] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao, "5G on the horizon: key challenges for the radio-access network", *Vehicular Technology Magazine, IEEE*, vol. 8, no. 3, (2013), pp. 47–53.
- [4] ETSI, GSNFV, "Network functions virtualisation (NFV): architectural framework", ETSI GS NFV, vol. 2, no. 2, V1.
- [5] <http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf>.
- [6] K-ONE Technical Document # 11 Design and Implementation of OpenFlow Monitoring System for ONOS
- [7] Jae-Man Been, Jae-Oh Lee "The Platform of Virtualized IMS using the NFV/SDN", *KNOM Review*, Vol. 18, No. 1 pp. 49-57.
- [8] Franco Callegati, Walter Cerroni, Chiara Contoli, Rossella Cardone, Matteo Nocentini, and Antonio Manzalini "SDN for Dynamic NFV Deployment" *IEEE Communications Magazine* • October 2016.
- [9] Hyun-Min Yoon, Zeqi Liu, Woo-Suk Yang, Jung-Ho Kim and Jae-Oh Lee, "Architecture for QoS-enabled Application Service Deployment in Virtualized Environment" ,
- [10] Jung-Hee Lee, Sang-Min Lee, Gang-Il Choi, Byeom-Cheol Lee, *The Journal of The Korean Institute of Communication Sciences* 30(3), 2013.2, 51-58 (8 pages)
- [11] Camarillo, Gonzalo and Miguel-Angel Garcia-Martin, *The 3G IP multimedia subsystem (IMS): merging the Internet and the cellular worlds*, John Wiley & Sons, pp. 245~249, 2007.
- [12] Jung-Yeon Hwang, Ho-Yong Ryu, *Korea Institute of Information Technology Magazine* 14(1), 2016.6, 1-13 (13 pages)
- [13] Nguyen Thi Minh Hien, Jae-Oh Lee, *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, vol. 11, (2007), pp. 384-387.
- [14] D. Goderis, *Service level specifications semantics, parameters, and negotiation requirements*, IETF Internet-Draft, draft-tequilasls-02.txt, (2002).
- [15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, (2008), pp. 69–74.