SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 6, NO. 6
ISSN: 2054 -7420

SDN/NFV Based Internet of Things for Multi-Tenant Networks

Do Sinh[1], Luong-Vy Le[2], Bao-Shuh Paul Lin[1,3], Li-Ping Tung[3]
[1]*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan*
[2]*College of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan*
[3]*Microelectronics & Information Research Center, National Chiao Tung University, Hsinchu, Taiwan*
dosinhuda.cs04g@nctu.edu.tw, leluongvy.eed03g@nctu.edu.tw, bplin@mail.nctu.edu.tw,
lptung@nctu.edu.tw

ABSTRACT

The Internet of Things (IoT) refers to variety of smart devices such as smartphones, tablets, and sensors that can interact and exchange of data among devices through the Internet. The diversity of IoT devices and their services have posed a larger range requirements of availability, throughput, latency, and performance in heterogeneous connectivity environments. Meanwhile, the existing networks often struggle with such of limitations in complex control protocols and difficulty in internetworking with billions of smart devices with different requirements such as latency and bandwidth allocations. These obstacles become substantial barriers to deploy services, as well as isolate between multiple co-existing tenants on the same physical network, deploy simultaneous protocols in the network, be stable to maintain the bandwidth and latency according to predefined QoS demands. These obstacles have recently been facilitated by Software Defined Network (SDN) and Network Function Virtualization (NFV) technologies that enable the programming and monitoring in data plane. In this study, firstly, the authors investigate and propose a SDN/NFV based architecture for multi-tenant networks with plenty of network slices working in a shared physical infrastructure. Secondly, P4 and ONOS Controller are used to implement a deep programming in BMv2 devices to efficiently maintain the network motoring in order to guarantee the E2E latency of communicating channels. Finally, the VXLAN technologies are exploited to for network slicing with different purposes and applications, and Inband Network Telemetry (INT) is used to monitor network latency.

Keywords: Software Defined Network (SDN); Software Defined Mobile Network (SDMN), Network Function Virtualization (NFV); Internet of Things (IoT); Wireless Sensor Network (WSN); P4; PSA(Portable Switch Architecture); ONOS.

1 Introduction

Recently, SDN[1] and NFV[2] are considered as key technology and promising enablers for network deployment, operation, and management with the full capacity of advanced programmability, flexibility, and elasticity. Therefore, they are expected to address different challenges in the next generation network (5G) and the Internet of Things[3][4][5][6].

DOI: 10.14738/tnc.66.5695
Publication Date: 23rd December 2018
URL: http://dx.doi.org/10.14738/tnc.66.5695

SDN is a new paradigm for network processing, it decouples the data and control planes to simplify network configuration and encourage evolution. It provides a new model for network providers to program the control plane for managing data plane devices and optimizing network resources automatically. OpenFlow[7] is a popular API network protocol for SDN to control data plane devices that bring significant benefits for future network deployment such as reduce OPEX (Operating Expense) and CAPEX (Capital Expense). Moreover, P4[8] is a domain-specific language (DSL), multi-platform, protocol, target independent, and re-configurable which is designed to allow the programming of data plane devices with different targets such as software switches, FPGA-based NICs (Field-Programmable Gate Array based Network Interface Cards) or switches based on reconfigurable ASICs (Application-Specific Integrated Circuit)[9]. Therefore, it provides an efficient way to configure the packet processing pipelines to improved network programming in both data and control planes that can achieve such benefit from network performance and encourage high speed network innovation.

NFV (Network Function Virtualization) is an approach to virtualize network element that mean billions of complex dedicated network appliances such as middleware boxes, firewalls, and even routers, switches, etc., are replaced by software running on commodity hardware. As a result, NFV has important roles in reducing the deployment cost and power consumption, and increasing network scalability, network efficiency, and services deployments.

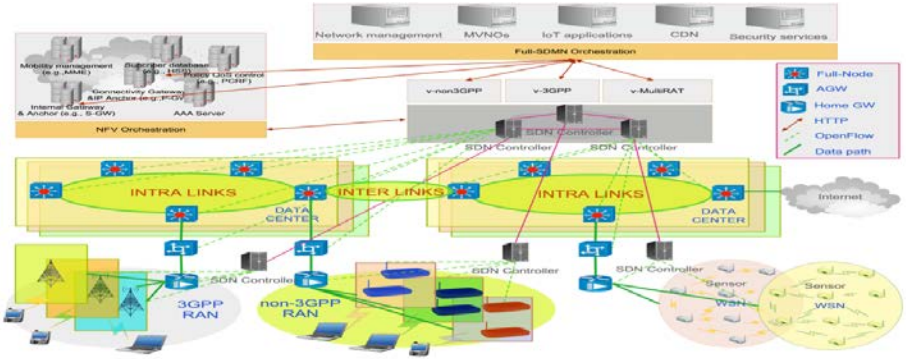


Fig. 1. Full-SDMN Architecture

Figure 1. Full-SDMN Architecture

On the other hand, the Internet of Thing has been growing rapidly in the number of IoT devices, applications, and data collection. With billions of heterogeneous IoT devices connected to networks for exchanging data with different protocols can cause many challenges about the system management and orchestration. Fortunately, SDN/NFV can be exploited as a promising and effective solution to handle the concern. For example, they allow various E2E (End-to-End) communication with different technologies in term of a protocol-independent to interconnect simultaneously through a controller’s application that is used to control and manage underlying devices such as P4. The SDN/NFV architecture for 5G and IoT called Full-SDMN (Full-Software Defined Mobile Network) shown in Fig.1 was proposed and described in our previous study[4], in that SDN/NFV were applied for 5G network and served as a tool for deploying IoT applications.

In this paper, firstly, we propose the Full-SDMN architecture for 5G and IoT, and illustrate how an IoT application can be deployed as an SDN controller application on top of SDN network under the synergy of NFV orchestration, Full-SDMN orchestration, and SDN controller. Secondly, a framework integrating state-of-the-art SDN/NFV components for monitoring IoT applications of multiple tenants with diversity requirements on the same physical network is proposed. In this architecture, the PSA[10] and BMv2[11] P4-based switches are exploited in the data plane to perform multiple protocols services, that mean packets of each path of a service are processed according to its predefined requirements , such as throughput and E2E latency. Moreover, VXLAN (Virtual eXtensible Local Area Network) and VXLAN GPE (Generic Protocol Extension for VXLAN)[12], which have been recently explored by network developers, are considered as key technologies for extending network slicing services to tackle the limitation number of the current VLAN (Virtual Local Area Network). They allow end devices communicate without being limited by distance. In this study, VXLAN and INT[13][14] are chosen as a new integrated solution to monitoring the E2E latency of different services. Finally, several experiments are implemented in the new environment P4-BMv2 target and ONOS Controller[15] to evaluate framework performances.

The remainder of the paper is organized as follows: Section 2 presents the related work; Section 3 proposes and describes the Full-SDMN architecture for the multi-tenant networks; Section 4 implements experiments and performance analysis; Section 5 addresses challenges of the model and discusses the future works; Section 6 concludes the present study.

2 Related Work

SDN/NFV architecture for IoT has recently attracted a lot attention from academia, for example, research [16] proposed a general SDN/NFV-based framework for IoT, in which various IoT devices were connected through SDN/NFV-based gateways, and a framework over wireless sensor networks (WSN) were introduced in study [17] introduced; for more detail, research [18] developed a three-level hierarchical fog architecture applying for smart grids, vehicular networks, and WSN, and research [19] described how SDN-based sensor nodes operate in SDN networks; [20] introduced SDN/NFV-based architecture for supporting network slicing applications. [21] proposed an architecture that can co-operate between SDN and NFV, and they also defined E2E logical network slicing running on a common underlying network. On the other hand, many studies focused on analyzing and deploying the quality of service (QoS) demands for the wide range requirements of diversity E2E slices using SDN/NFV approaches [22][23][24][25][26][27]. However, there are several challenges still need more concentrated effort to overcome: i) a comprehensive IoT framework with a deep programmable data plane to easily implement independent protocols for diversity of IoT devices and services. ii) E2E network virtualization and isolation network or E2E network slicing for multi-tenant applications working under the same physical infrastructure. iii) Effective methods for resource monitoring and optimizing in IoT networks with a huge number of smart devices and connections and complicated service requirements. INT, a new approach based the handshaking between data and control planes, is used to control the E2E latency. iv) Practical models based on SDN/NFV and SDN controller applications to prove the effectiveness of state-of-the- art SDN technologies in providing managing and programing the data plane to supporting different policies on requested IoT services such as multi-tenant services.

To deal with these issues, the Full-SDMN architecture shown in Fig.1 can be considered as a suitable solution supporting independent protocols and less hardware dependent. Especially, the recent integration of P4 based BMv2 switches to the architecture is an essential innovation to overcome the complexity in pipeline of the existing OF- (OpenFlow Data Plane Abstraction)[28] and OpenFlow-based switches. The reprogrammable pipeline is a crucial step motivating service providers to define new protocols so that network programming, controlling, and DPA management become more flexible by using SDN applications with the co-operation of the NFV orchestrations, SDN controllers, and Full-SDMN orchestration. Based on the application, the Full-SDMN orchestration creates original policies to manage and control network elements and services such as MVNOs (Mobile Virtual Network Operators), and IoT applications, and then it generates flows to program the elements to create virtual network slices. Meanwhile, the management of NFV orchestration ensures those element are created and operated normally underlying network. Finally, SDN controller application controls physical components and network function

3 Proposed SDN/NFV Based Internet of Things Architecture for Multi-Tenant Networks

Fig.2 illustrates our proposed SDN/NFV-based IoT architecture for multi-tenant networks, which comprises three main planes: Data plane, Control plane, and Management plane.

3.1 Data plane

WSNs comprise billions of devices/sensors with different types and abilities of computing, sensing, and communicating for various types of applications such as smart-home, e-healthcare, and autonomous cars. In this scenario, the HomeGWs (Home Gateways), which reside at edge network have significant roles connecting the IoT devices, WSNs, and the core network. the HomeGWs work as the gateway supporting for different type of edge networks such as 3GPP RAN (Radio Access Network) and non-3GPP RAN. Therefore, they must satisfy such capacity of flexibility, and scalability for hosting various applications. They are also responsible for

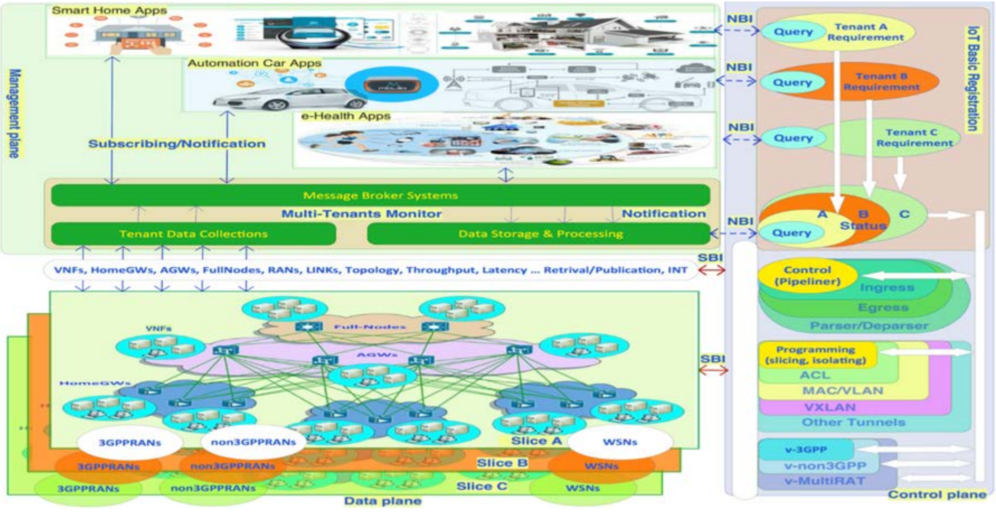


Fig. 2. SDN/NFV Based IoT Architecture for Multi-Tenant networks

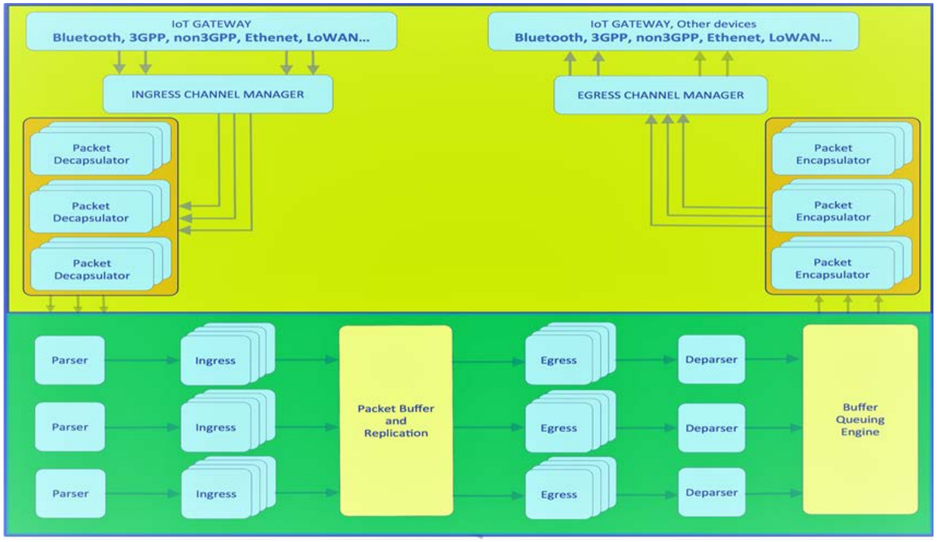


Fig. 3. Multi-protocols and SDN-enabled HomeGWs

processing data generated by WSNs to the Internet. As a result, a huge amount of data generated by IoT devices is pre-processed at each HomeGW so that the data processing at the core network will be significantly reduced. Hence, the HomeGW must be powerful enough to satisfy these requirements. Fig.3 shows our proposed HomeGW architecture based on SDN technology and describes how to process and modify packets of various services in HomeGWs.

In this architecture, Portable Switch Architecture (PSA) model, which has six programmable P4 blocks and two fixed-function blocks with multiple-pipelines, Packet Buffer, and Buffer Queuing Engine are used for installing QoS policies at the data plane. Moreover, the separation of ingress and egress pipelines make the process, modify and management flow packets more flexible and efficient. This is an essential characteristics to support programming interfaces among IoT services regardless of various services with different requirements, that means a service request is independently allocated appropriate resources. For example, Fig.4 describes how to apply different IoT applications in a multi-tenant framework, here, each IoT application such as Automation Cars, Smart Home, and e-HealthCare only intercommunicate with its components (services and IoT devices). Each application usually expects a network performance guarantee like throughput and latency based on the characteristic of services. For example, the vehicular communication, cars need to communicate with one another (Vehicle to Vehicle or V2V), and with the network infrastructure (V2I) for supporting safety services

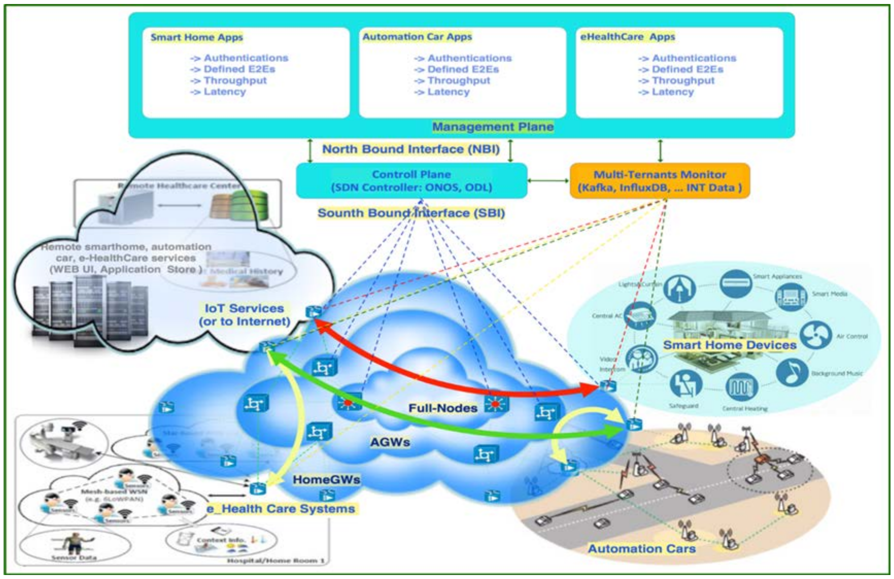


Fig. 4. An example for applying IoT Architecture for Multi-Tenants

(i.e., real-time dangerous warning of on the road) and non-safety services (i.e., local traffic, road map, local weather, car parking areas, popular news)[29][30]. On the other hand, the servers of different applications are often allocated at different positions in the networks. For example, with safety services, their servers must be located at the edge network to reduce latency, while for non-safety services like entertainments, these servers are usually located On the Internet. Therefore, it is necessary to create some virtual RANs (VRAN) and virtual core network (VCN) with different VLAN ID under the data plane of access technologies and core networks such as v-3GGP, v-non3GPP, and v-Multi-RAT. Those VLAN are created and controlled by controller applications based on several methods such as MAC/VLAN tag, VXLAN and VXLAN GPE[12].

Similarly, Applying IoT for eHealth systems is complicated due to network communications, data collection and processing from a massive types of smart sensors and actuators (i.e., heartbeat sensors, body sensors, blood pressure devices, wearable smart medical sensor devices, wire or wireless medical instruments) [31]. Among the common access technologies or WSN for eHealth system, ZigBee, Bluetooth, and LPWAN (Low-Power Wide Area Network) are the most popular due to the fact that they are low energy and low-cost wireless networking standards. [32][33]. In eHealth networks, health data and patient information such as electronic health records and electronic medical records are collected and sent via HomeGWs to Remote Healthcare Center or Internet without caring about latency, while telehealth, telemedicine, and connected health services need a real-time communication [34].

The last IoT scenario discussed in this subsection is IoT for Smart Homes in which smart sensors are embedded in the smart home environment like central power, sight, heating devices, and even other multi-media systems such as background music, and smart appliances to bring more benefits, comfortable, and secure for household users [35][36]. Smart sensors communicate over IoT to systems and usually provide as web server- client modes.

Table 1. IoT applications and requirements

ID	Different Requirements of each IoT services in network			
	Application domains	***Descriptions***	***Data rate***	***Tolerable delay***
1	Automation car and safety services.	real-time warning, roadside functionalities, real-time emergency respond	high	Low (milliseconds)
2	non-safety services for vehicles.	local traffic, road conditions, road map, local weather, car parking areas, popular news.	normal	Normal (seconds)
3	e-Healthcare:alarmand emergencyrespond.	AAL (ambient assisted living), remote diagnosis, disable assistance	normal	low
4	e-Healthcare: data collection and context information storage.	Remote Healthcare Center or Internet for purposes of storing medical history and giving appropriate medical treatments later.	normal	normal
5	Smarthome: data collection, remote controlling, energy management	Central AC, light, central heating, smart appliances, air control.	low	normal
6	Smarthome for Entertainments and comfortable living	Video Intercom, smart Media, Background Music	high	normal
7	Smarthome for safety	security and video surveillances,, access managment	high	low

Table 1 summarizes different requirements (throughput and latency) for typical IoT application domains; therefore, IoT platforms must provide the flexibility for the planners, designers, and implementers that they can employ emerging industrial IoT applications under suitable performance network slices. However, with the increase in IoT devices and services, the current methods for virtualizing network such as VLAN are no longer relevant and substituted by new approach such as QinQ and VXLAN. Moreover, a comprehensive IoT platform supporting a deep programming in the data plane is an essential requirement to implement network slicing and multi-protocol IoT application such as VLAN tagging, VXLAN, and VXLAN GPE.

3.1 Control Plane

The control plane is described as in Fig.2, in which SDN controllers work as a distributed controlling system. They play three significant roles in the network: Program the data plane; manage flows across the network nodes; and provide the platform for hosting controller applications. They co-operate with one another to create on demand and monitor VRAN architectures, such as 3GPP, non-3GPP, and MutiRAT, using SDN controller applications without the need to alter the data plane components. Practical SDN controllers that can meet such wishing of providing the flexibility to easily build and deploy new dynamic IoT services with simplified programmatic interfaces are ONOS and OpenDayLight (ODL)[37]. For example, ONOS, a distributed control architecture, supports both configuration and real-time control of the data plane to deploy of new IoT software, hardware, and applications. In this study, the authors use ONOS controller to build VRAN as v-3GPP, non-3GPP, and v-MultiRAT applications in which the standards are defined by software or SDN applications. These applications modify and control packets though match and actions (parser, ingress, egress, and deparser) in the pipeline of OpenFlow Switch or slice network by using ACL, MAC/VLAN ID Tagging, VNI - VXLAN Network Identifier and VNI-VXLAN GPE).

The first approach use ACL to program the data plane, this application is a built-in ONOS application including rules to allow or deny IP traffic at relevant devices. Through the Multi-

Tenants application, SDN controller conducts real-time measurements over fast varying parameters such as the queues in BMv2 switches, links states. Moreover, it uses Round Robin, Kalman filter algorithm to control each slice to fulfill the services with respect to application characteristics. Thank for P4 - INT [13][38], a powerful network monitoring mechanism implemented in P4, can be injected to every packet to specify the type of metadata such as switch ID, ingress timestamp, hop latency, etc.

The second approach is VLAN, which (Virtual Local Area Network) is an isolation approach to separate different classes of packets. Network providers use VLAN as a group identifier to apply different policies such as QoS to different groups. A VLAN-based slicing is deployed with VLAN IDs tagged to packets to represent for each tenant , [39]. Otherwise, a slice can be defined as a group of devices' MAC addresses or IP addresses pre-defined in the IoT controller application. Moreover, QinQ is used to extend the VLAN numbers up to 4096 x 4096 to solve the limitation of number VLANs (4094 VLAN IDs).

The last approach is VXLAN or VXLAN GPE, which is designed to provide layer 2 overlay network on top of the layer 3 network using MAC-in-UDP encapsulation, in which a 24-bit VNI field can be used to define the number of LAN segments (up to 16 million) to meet the demand on network scales.

Another important component in control plane is the IoT Basic Registration controller application for providing query points to interconnect applications and Multi-Tenants Monitor controller applications to monitor the network slices, which will be explored in the next section.

3.2 Management plane

The management is implemented in two main phases. The first phase aims to set up the virtual E2E transmissions between end users via HomeGWs, AGWs, and Full-Nodes. In this phase, IoT controller applications define the requirements physical resources such as E2E transmissions and its QoS (throughput, latency). Firstly, the requirements are sent to IoT Basic Registration Application to implement application

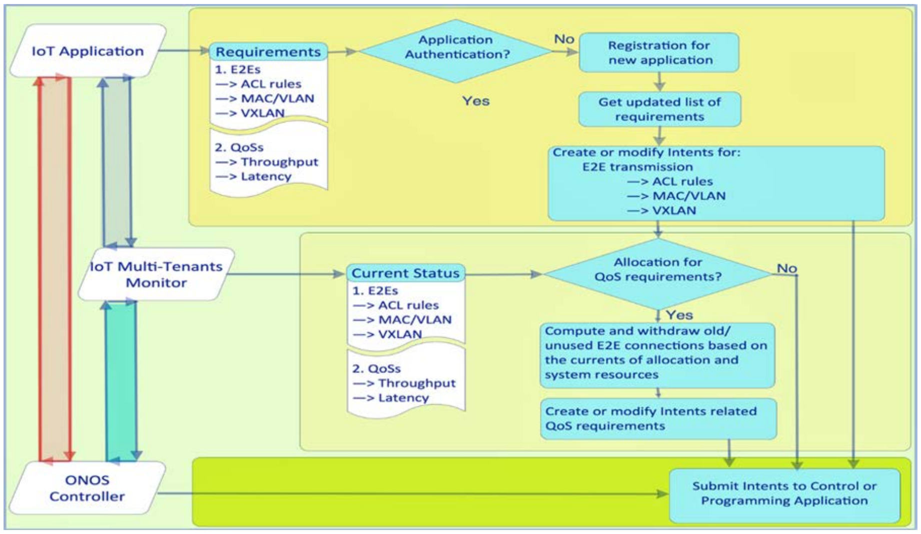

IoT Application
Requirements
1. E2Es
—> ACL rules
—> MAC/VLAN
—> VXLAN
2. QoSs
—> Throughput
—> Latency
Application Authentication?
No
Yes
Registration for new application
Get updated list of requirements
Create or modify Intents for:
E2E transmission
—> ACL rules
—> MAC/VLAN
—> VXLAN
IoT Multi-Tenants Monitor
Current Status
1. E2Es
—> ACL rules
—> MAC/VLAN
—> VXLAN
2. QoSs
—> Throughput
—> Latency
Allocation for QoS requirements?
No
Yes
Compute and withdraw old/ unused E2E connections based on the currents of allocation and system resources
Create or modify Intents related QoS requirements
ONOS Controller
Submit Intents to Control or Programming Application

Fig. 5. Application Registration and Monitoring QoS

authentication, modify Intents as shown in Fig.5, and then create network slices applied to Control and Programming Applications based on several methods such as ACL rules, MAC/VLAN, QinQ, and VXLAN.

The second phase aim to maintain the QoS of E2E transmissions, in this phase the Basic Registration Application continuously check the current QoS of E2E transmission from a Multi-Tenants Monitor Application, and optimize network efficiency by creating or modifying Intents that are related to QoS requirements based on the current system resources.

In this architecture (Fig.5), different IoT applications include smart home, automation car, e-Health applications associate with their requirements and the authentication procedure for their end-users. Moreover, their current E2E channel status are provided via notification of the Multi-Tenants Monitor Application to track their services deployed at data plane.

The Multi-Tenants Application plays the main roles in collecting data from Tenants such as VFNs, HomeGW, AGW, Full-Node devices, links, topology events via SDN Controller. The notations are shifted to IoT applications through the Message Broker Systems. It also processes and stores data for QoS controlling purposes at the IoT Basic Registration Application. In this architecture, P4 INT mechanisms are applied for collecting and reporting network states by data plane without requiring intervention or work from the control plane.

To clearly explore how a user can access and use services and network slicing according to its demand,

Fig.6. shows an example of Automation Car User (UE) that want to access Automation Car Services stored in the Server. The process is explained bellows:

Firstly, the UE sends a packet to its destination via RANs such as SDN based eNBs, or SDN based APs, or SDN based WSN nodes, etc.. And then RANs send a packet-in to SDN Controller (step 2) for authentication and virtual E2E channel establish. Here we assume that Automation Car Controller Application was accepted by IoT Basic Registration Application. Step 3, IoT application implements UE authentication and send back a message to IoT Basic Registration Application for creating or modifying intents of the virtual E2E channel (step 4). Here, if the IoT application does not have any QoS requirement, then the intents created at step 4 will be submitted to Control Application and Programming Application (in SDN Controller) to establish virtual E2E channel (step 5- 12).

Otherwise, if the IoT application requires a QoS control, Multi-Tenants Monitor Application will implement data collection such as devices, links, topology via SDN Controller (step 13-15), then shifts notification to Automation Car Application (step 16) and IoT Basic Registration Application (step 17, 18). After receiving requirements of the IoT application (step 19), the IoT Basic Registration Application re-computes, modifies, and creates Intents for QoS requirements, and then submits it to the Control and Programming Applications to slice and control the virtual E2E channel based on-demand services (step 20-24)

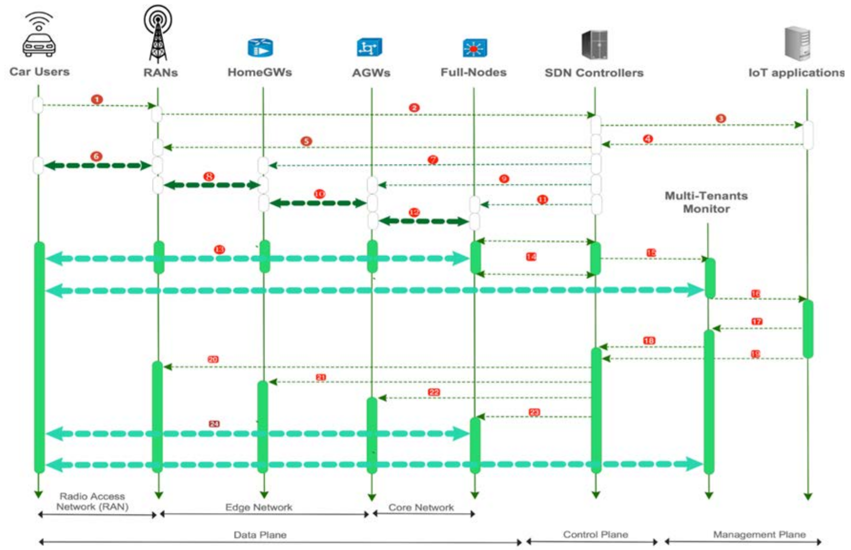


Fig. 6 An example for Automation Car Processing

In summary, the proposed SDN/NFV-based Internet of Things architecture for Multi-Tenants satisfies the trend of 5G, and be completely suitable for multiple providers.

4 Implementation and Evaluation

In this section, we implement an experiment in the standard P4-BMv2 target environment for core networks to perform network slicing and isolating via VXLAN and inject INT for monitoring QoS of multiple E2E channels. The authors used the open source P4 BMv2 switches model available at https://github.com/opennetworkinglab/routing/ and Mininet[40] to create each P4 BMv2 switches in the different trellis topologies to implement the experiments on different spine-and-leaf with the same number of Full-Nodes (spines) and AGWs (leaves). All tests were deployed on the off-the-shelf Dell PowerEdge 1950 III Server platform with two quad-core Intel Xeon 5400, CPU 3.16GHz, and 64GB memory running Linux 4.15, Mininet version 2.2.2, P4 Runtime package, and ONOS Controller version 1.13.2. Moreover, the INT over VXLAN GPE is suitable for virtualized data centers in building the Multi-Tenants Monitor application. Here, the VXLAN GPE (generic protocol extensions) is used to carry INT Headers between VXLAN GPE header and payload. The designed pipelines, INT and VXLAN GPE are used to reduce the complexity in network performance and forward packets transparently over the whole network that be suitable for implementation in virtualized datacenters. Fig.7.a shows the concept of modified BMv2 for multiple pipelines, and the pipeline processing is shown in Fig.7.b. Fig.8. shows a standard packet framework that formats for L2 Frame crossing whole network including the detail explanation of VXLAN header and INT header as bellow:

The modified VXLAN GPE Header is redefined within fields bellow: P bit (Next Protocol Bit) is set to indicate that the Next Protocol field is present; VXLAN Next Protocol bits defined values as 0x01 for next protocol is IPv4; 0x02 for IPv6; 0x03 for Ethernet; and 0x05 for In-band Network Telemetry Header (INT Header); O bit (OAM Flag Bit) is set to indicate that the packet is an OAM packet. In this case, this packet is sent to the ONOS Controller for controlling purposes; otherwise, it is a normal packet that must be forwarded. B bit is defined as BUM bit or a Flag bit. When B bit is set to 1, the packet is marked as a replicated-packet that need to be processed at the Packet Buffer and Replication Function for Multicast traffic purposes.

Instance bit (I bit) is set to indicate a valid VNI, which is a 24-bit field to identify the VXLAN overlay network and indicate that inner packets belonging to different VNIs cannot communicate with each other.

INT header is defined as sub-fields, INT's Type is an 8-bit field. Here, we reserve two types: hop-to-hop INT header and destination INT header type; INT Length is an 8-bit field indicating the actual INT data following the INT header ((0xFF – 4) * 32 bit); Max Hop Count is an 8-bit field indicating the maximum number of hops; Total Hop Count is an 8-bit field indicating the total number of hops that have added their metadata instances to the INT packet; Rep is a 2-bit field for replication requested; C is a 1-bit field indicating the packet that is a copy packet if C is set to 1. The sink HomeGW must be able to distinguish the original packets from replicas to process appropriately; E is a 1-bit flag to indicate that whether a device can prepend its own metadata due to reaching the Max Hop or not; Instruction Count is a 5-bit field indicating the number 1's bit in the Instruction Bitmap; and Instruction Bitmap is a 16-bit INT Instruction field, each bit. Correspond to a specific standard metadata such as bit 0 indicating the Switch ID, bit 1- the Ingress port ID and Egress port ID, bit 2- Hop latency, bit 3 defines Queue Occupancy, bit 4- Ingress timestamp, bit 5- Egress timestamp, bit 6- Queue congestion status, bit 7- Egress port tx utilization, and the remaining bits are reserved.

There are metadata fields defined by HomeGWs, AGWs, and Full-Nodes that enable the P4 program to specify where each packet arrived on, and control where it will go next. The P4 programmer defines objects in P4 that conforms to APIs, and the inputs and outputs of the programmable blocks such as Parser, Ingress, Egress, and Deparser based on user-defined headers and metadata.

After parsed by the Parser, packets are fed to the Ingress block for forwarding step. In this context, the standard packet format and forwarding process are implemented as the main roles of the Ingress block. In the next step, the forwarding process, firstly, verifies the header of the packets to determine what kind of packets and their functionalities. After that, the forwarding process uses a basic forwarding match-action table to find the egress port forwarding packets to destinations. This forwarding mechanism is supported on P4 programming and ONOS services, such as link layer discovery and reactive forwarding.

After processed at two fixed functions: PBR (Packet Buffer and Replication) and BQE (Buffer Queuing Engine), packets are sent to the programmable Egress block for removing the INT header (at HomeGWs) or cloning packets to egress ports. At Egress pipeline, the header is updated and then sent out of Egress to Deparser. Finally, at the Deparser, the packets are serialized and sent out of relevant egress ports.

In this test, the network consists of 16 nodes as shown in Fig.9. a). Fig.9. b) evaluate the latency of E2E communication in the UDP/VXLAN/INT test case, respectively; Fig.9.c), d) and e) show the test result of

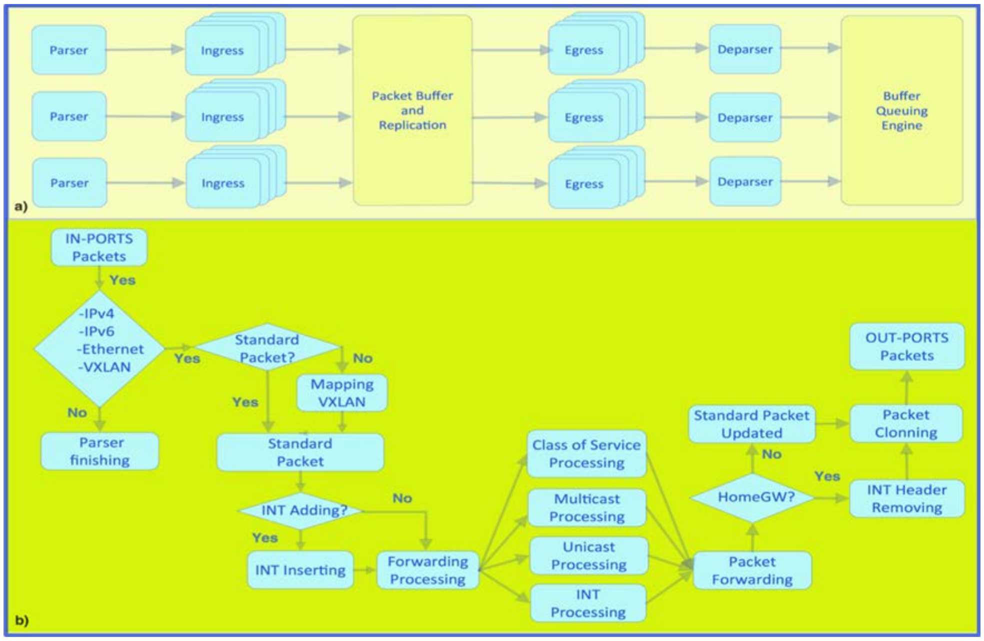


Fig. 7. a) The Concept of Modified BMv2 and b) A modified BMv2 pipeline processing

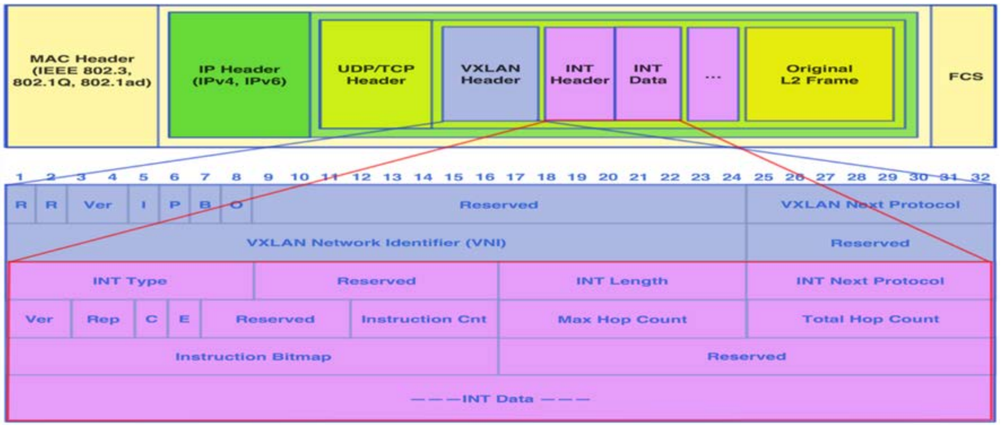
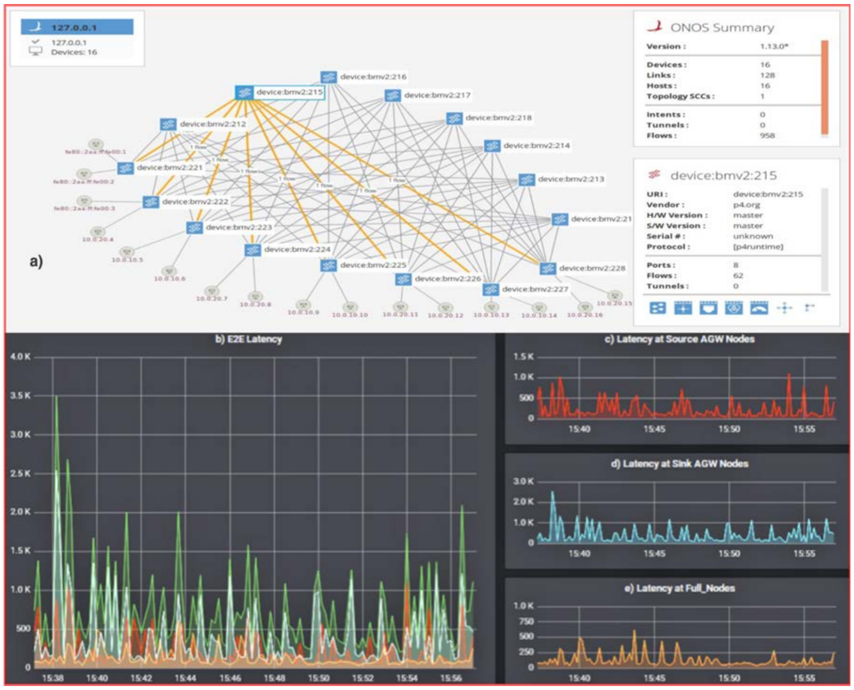


Fig. 8. A standard packet that format for L2 Frame crossing the whole network

UDP/VXLAN/INT after adding latency at source AGWs and AGWs, and Full-Nodes, respectively.

Next experiment, the Multi-Tenants Monitor implements QoS control for 8 different levels based on the 3 bit-QoS priority field predefined from QoS_class0 to QoS_class7, in each class the latency is predefined according to the customer demands. Fig. 10 and 11 show the testing results of some classes with a duration of 20 minutes for each class (QoS_class0, QoS_class2, QoS_class5, and QoS_class7). In the Maximum QoS (QoS_class7) require sink AGWs have to process amount of bigger data than other QoS _classes.

The last experiment, we choose different numbers of VNI-VXLAN at the highest variety of CDF of different cases to test the variable maximum QoS (QoS_class7) as shown in Fig 12. As can be seen, when the numbers of communicating VNIs channels increases, the processing load of the Full-Node also increases. Therefore, the latency at Full-Node is the main latency, result in the increase of E2E latency. Fig. 13. a) and b) show the testing results for 32 VNIs and for 64VNIs in the network, respectively. The E2E latencies are stable until the last duration (n=64; p=0.7), where increase among of source, sink AGWs and Full-Nodes caused by low performance of deploying in this demo.



ONOS Summary

Version : 1.13.0*

Devices : 16
Links : 128
Hosts : 16
Topology SCCs : 1

Intents : 0
Tunnels : 0
Flows : 958

device:bmv2:215

URI : device:bmv2:215
Vendor : p4.org
H/W Version : master
S/W Version : master
Serial # : unknown
Protocol : [p4runtime]

Ports : 8
Flows : 62
Tunnels : 0

Fig. 9. a) A test case on a topology 16 nodes; b) 250 Byte-UDP/VXLAN/INT packets to test the latency at End to End host communication (10-3 ms); c) the latency at source AGWs; d) the latency at sink AGWs; e) the latency at source Full-Nodes.

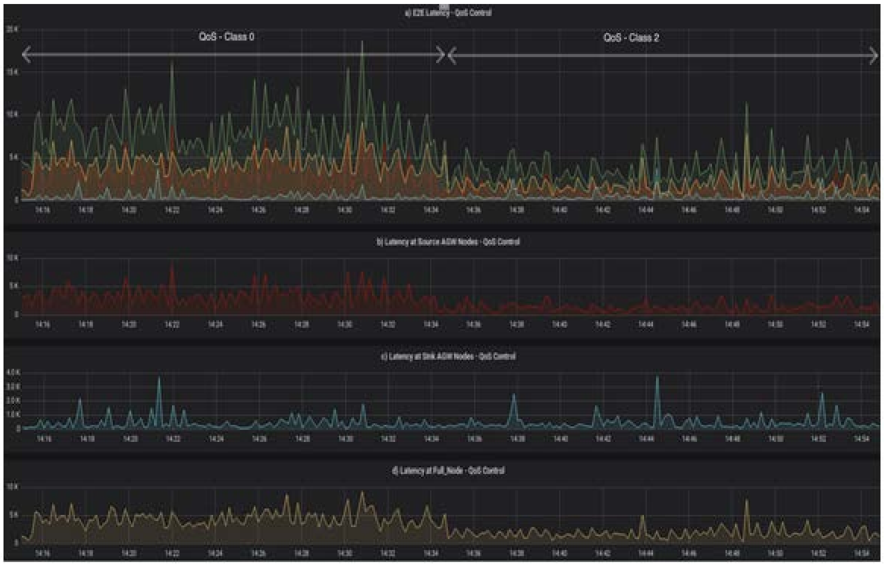


Fig. 10. A test case for QoS_Class0 and Qos_Class2. a) the latency at End to End host communication (10-3 ms); b) the latency at source AGWs; c) the latency at sink AGWs; d) the latency at Full-Nodes.

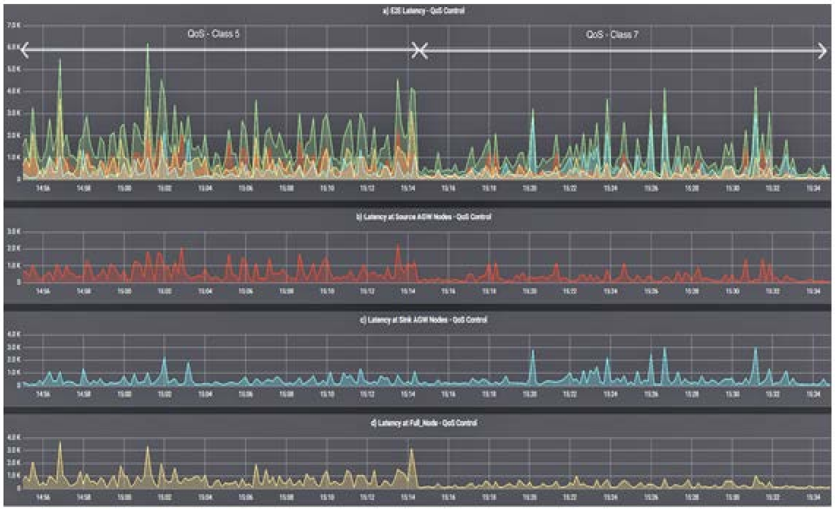


Fig. 11. A test case for QoS_Class5 and Qos_Class7. a) the End to End latency (10-3 ms) hosts communicating; b) the latency at source AGWs; c) the latency at sink AGWs; d) the latency at Full-Nodes

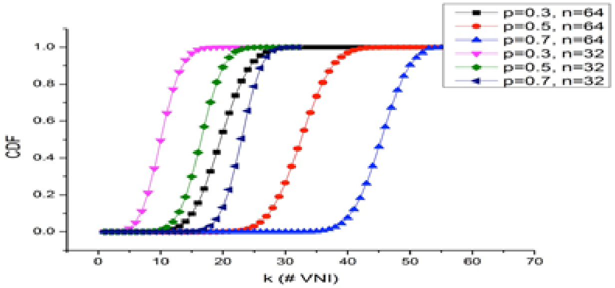


Fig. 12. A test choosing for Qos_Class7 with different VNI cases

5 Future Works

Our future work focuses on investigating and deploying components in customers' side and edge cloud, and integrating different IoT technologies to turn the HomeGWs into a multi-protocols gateway for plenty of IoT technologies and devices. On the other hand, we continue explore programmable data planes like Mobile CORD with P4 and various kinds of SPGW (Serving and Packet Gateway) and build applications on core networks such as traffic flow monitoring, traffic congestion control, QoS and QoE (Quality of Experience) control. Moreover, network slicing implemented at the Full-Nodes and AGWs different RANs is an essential technology for developing multiple protocols (both stateful and stateless protocols) in the network to meet diversity demands of applications in new scenarios.

6 Conclusion

A deep programmable data plane for deploying new network protocols, reprogramming pipelines and developing network services is crucial in opening new features for network innovation, availability, and scalability to satisfy the rapid increase in IoT devices and protocols. In this paper, the authors proposed a

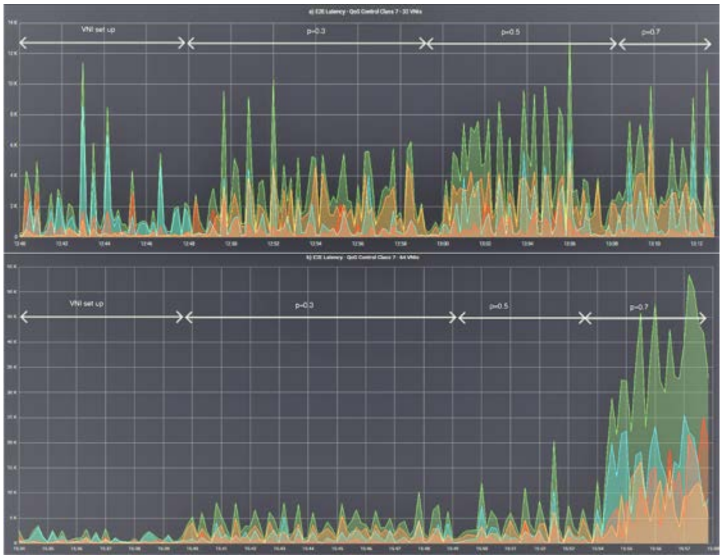


Fig. 13. A test case for QoS_Class7 with different VNIs. a) the End to End latency (10-3 ms) in case 32 VNIs deploying; b) in case 64 VNIs deploying

network architecture that completely suitable for multi-tenant networks, in which SDN/NFV play important roles in orchestrating entire network via SDN controller applications, and then, based on this architecture, network slicing associated with QoS control and network monitoring were implemented. We also recognized that, VXLAN, a new technology supporting a huge number of slices (sixteen millions), is a perfect solution for network slicing in datacenters and metropolitan environments. In addition, INT is a powerful approach for monitoring QoS of channels in the data plane, and then, the controller application can give timely respond to modify the network. The development of open sources such as P4-BMv2, ONOS is considered as key element and solution for new protocols, QoS monitoring mechanism for multi-tenant networks.

ACKNOWLEDGMENT

This paper is particularly supported by “the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project” by the Ministry of Education (MOE) in Taiwan, and the Ministry of Science and Technology of Taiwan under Grants: MOST 107-2221-E-009-056.

REFERENCES

[1] Open Networking Foundation, “SDN Architecture Overview,” *Onf*, no. 1, pp. 1–5, 2013.

[2] ETSI, “Network Functions Virtualisation (NFV); Architectural Framework,” *ETSI GS NFV 002 v1.2.1*, vol. 1, pp. 1–21, 2014.

[3] B. P. Lin, F. J. Lin, and L. Tung, “The Roles of 5G Mobile Broadband in the Development of IoT , Big Data , Cloud and SDN,” no. February, pp. 9–21, 2016.

[4] D. Sinh, L. Le, L. Tung, and B. P. Lin, “The Challenges of Applying SDN / NFV for 5G & IoT,” *14th IEEE – VTS Asia Pacific Wirel. Commun. Symp. (APWCS), Incheon, Korea, Aug 2017*, pp. 138–142, 2017.

[5] L. Le, B. P. Lin, L. Tung, and D. Sinh, “SDN / NFV , Machine Learning , and Big Data Driven Network Slicing for 5G,” IEEE 1st 5G World Forum, Santa Clara, CA, July 9-11,2018

[6] D. Sinh, L. Le, B. P. Lin, and L. Tung, “SDN / NFV - A new approach of deploying network infrastructure for IoT,” *27th Wirel. Opt. Comminications Conf. (WOCC 2018)*, pp. 124–128, 2018.

[7] Z. L. K. Ben Pfaff, Bob Lantz, Brandon Heller, Casey Barker, Dan Cohn, Dan Talayco, David Erickson, Edward Crabbe, Glen Gibb, Guido Appenzeller, Jean Tourrilhes, Justin Pettit, KK Yap, Leon Poutievski, Martin Casado, Masahiko Takahashi, Masayoshi Kobayashi, Nick M, “OpenFlow Switch Specification,” *Open Netw. Found.*, pp. 1–56, 2011.

[8] P. Bosshart, G. Varghese, D. Walker, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, and A. Vahdat, “P4: Programming Protocol-Independent Packet Processors,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.

[9] “P4 Brigade,” no. https://wiki.onosproject.org/display/ONOS/P4+brigade, p. 4.

[10] C. S. Architecture, “P4 16 Portable 1 Switch Architecture,” pp. 1–12, 2016.

[11] “Barefoot Networks. P4-bmv2. Website. https://github.com/p4lang/behavioral-model.,” p. 4.

[12] https://datatracker.ietf.org/doc/draft-ietf-nvo3-vxlan-gpe/?include_text=1, “vxlan-gpe-6.pdf.”

[13] Changhoon Kim, Parag Bhide, Ed Doe, Hugh Holbrook, Anoop Ghanwani, Dan Daly, Mukesh Hira, and Bruce Davie, “In-band Network Telemetry (INT),” no. September, pp. 1–28, 2015.

[14] “https://wiki.onosproject.org/display/ONOS/%28INT%29+In+Band+Network+Telemetry+with+ONOS+and+P4#id-(INT)InBandNetworkTelemetrywithONOSandP4-InstallBPFCollector,” p. 29.

[15] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, and B. Lantz, “ONOS: towards an open, distributed SDN OS,” *Proc. third Work. Hot Top. Softw. Defin. Netw. - HotSDN ’14*, pp. 1–6, 2014.

[16] M. Ojo, D. Adami, and S. Giordano, “A SDN-IoT architecture with NFV implementation,” *2016 IEEE Globecom Work. GC Wkshps 2016 - Proc.*, 2016.

[17] F. Granelli, A. A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Alitska, and P. Chatzimisios, “W IRELESS A CCESS IN F UTURE W IRELESS N ETWORKS : S CENARIOS AND S TANDARDS,”no. June, pp. 26–34, 2015.

[18] I. Stojmenovic, “Fog computing: A cloud to the ground support for smart things and machine-to-machine networks,” *2014 Australas. Telecommun. Networks Appl. Conf. ATNAC 2014*, pp. 117–122, 2015.

[19] H. I. Kobo, A. M. Abu-mahfouz, and G. P. Hancke, “A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements,” *IEEE Commun. Surv. Tutorials*, vol. 5, no. Submitted for publication, 2017.

[20] L. Velasco, L. Gifre, J.-L. Izquierdo-Zaragoza, F. Paolucci, A. P. Vela, A. Sgambelluri, M. Ruiz, and F. Cugini, “An Architecture to Support Autonomic Slice Networking,” *J. Light. Technol.*, vol. 8724, no. c, pp. 1–1, 2017.

[21] J. Ordonez-Lucena, P. Ameigeiras, Di. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, 2017.

[22] W. Miao, F. Agraz, S. Peng, S. Spadaro, G. Bernini, J. Perelló, G. Zervas, R. Nejabati, N. Ciulli, D. Simeonidou, H. Dorren, and N. Calabretta, "SDN-Enabled OPS With QoS Guarantee for Reconfigurable Virtual Data Center Networks," vol. 7, no. 7, pp. 634–643, 2015.

[23] M. Gramaglia, I. Digon, V. Friderikos, D. Von Hugo, C. Mannweiler, and M. A. Puente, "Flexible Connectivity and QoE / QoS Management for 5G Networks : the 5G NORMA view."

[24] S. S. Kumar, "Investigation of Security and QoS on SDN Firewall Using MAC Filtering," pp. 0–4, 2017.

[25] N. Ul Hasan, W. Ejaz, I. Baig, M. Zghaibeh, and A. Anpalagan, "QoS-aware channel assignment for IoT-enabled smart building in 5G systems," *Int. Conf. Ubiquitous Futur. Networks, ICUFN*, vol. 2016– Augus, pp. 924–928, 2016.