Evaluation of Underlying Switching Mechanism for Future Networks with P4 and SDN (Workshop Paper)

O. A. Fernando(✉), Hannan Xiao, and Xianhui Che

University of Hertfordshire, Hatfield, UK
{w.k.fernando,h.xiao,x.che}@herts.ac.uk

Abstract. Software Defined Networking (SDN) was introduced with a philosophy of decoupling the control plane from the data plane which facilitates network management while ensuring programmability in order to improve performance and monitoring. OpenFlow which enabled SDN was first introduced to match twelve header fields whilst at current it matches forty one which is expected to grow exponentially. Therefore future networks must have the ability to flexibly parse packets through a common interface. Programming Protocol independent Packet Processing (P4) was introduced to achieve the aforementioned by programming the underlying switch, providing instructions and utilizing APIs to populate the forwarding tables. A P4 programmed switch will forward packets through a parser into multiple stages of match+action tables to find the destination node which is considered the most efficient mechanism for routing. This paper takes into the account the latest platform developed for service providers, Open Networking Operating System (ONOS) to deploy two environments configured in the aforementioned technologies in order to test their performance. Four case studies were drawn which were simulated in Mininet which incorporated SDN + P4 switches. A significant increase of performances were recorded when compared with the performance of cases using SDN only.

Keywords: SDN · P4 · ONOS · Mininet

1 Introduction

By understanding and evaluating the trend of the internet and users intent, it has become evident that the future access of networks will be carried out via utilizing a smart hand held device compared to a red-brick personal computer [13,14]. Cellular network play a crucial part on the aforementioned statement where the users access the medium on-the-go as a habit. The use and application of such technologies has paved the way for researchers to explore new avenues of applications in IoT. They can be listed as mobile cloud computing, vehicular

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 549–568, 2019.
https://doi.org/10.1007/978-3-030-30146-0_38

networking, edge computing etc. [1]. The heterogeneity of applications executing on user devices requires higher access time and availability of the network over latency. According to a comprehensive research by Mobile and wireless communication Enablers for Twenty-twenty Information Society (METIS) have presented the following key performance indicators (KPIs) for the future networks.

- 1000 times higher mobile data volume
- 10 to 100 times higher typical user data rate
- 10 to 100 times higher number of connected devices
- 10 times longer battery life
- 5 times reduced E2E latency, reaching a target of 5 ms for road safety applications [2]

According to [2] most of the delay derive from the internet. In order to achieve the said requirement of latency, more efficient network architectures, signalling and air interface designs must be taken into consideration. The current networks can not accommodate the above mentioned KPIs due to incapable fundamental designs and centralised routing mechanisms [1] etc. Therefore a new underlying switching/forwarding mechanism must be inherited in order to achieve the KPIs presented by METIS [2] to reduce the latency of the core and increase it's performance.

This research consists of technologies and software which will be later discussed in an in-depth manner. Software Defined Networking (SDN), Programming Protocol independent Packet Processing (P4), Mininet, iPerf, Open Networking Operating System (ONOS) and the GUI of ONOS are among them. SDN was introduced as a mechanism of decoupling, disassociating the data plane from the control plane which provides more efficiency breaking away from the decentralization of the predecessor networks. SDN also improves network performance and enables network monitoring.

P4 is a high level programming language, which is domain-specific with a number of constructs designed for the sole purpose of optimizing network's forwarding plane. P4 is an open-source language maintained by a non-profit organization, which goes under the name P4 Language Consortium [15]. The language was originally described and presented in the white paper titled Programming Protocol-Independent Packet Processors [5].

The research will be based on understanding the functionality of the aforementioned two technologies and evaluating the performance. System design or the test bed will be discussed later with illustrations at Sect. 2. In brief, two environments will be designed with the same variables and resources with the exception of the underlying controller. A network which consists of a SDN controller and an independent environment with P4 switches enabled with a SDN controller are the two. Test results, data and statistics are collected by running various experiments. The said experiments are designed to be line with the ISO OSI model, which will be explained in detail at Sect. 2. Data capture is conducted utilizing a well-known software, Wireshark [16].

Wireshark is a free, open-source software designed for packet analysing. Wireshark equips a GUI for ease of understanding where as a terminal CLI based option is also available. Wireshark has the ability to capture packets as described

earlier that can be used for education, troubleshooting, protocol usage, port activities and for conducting various analysis on the network. Whilst conducting the experiments as mentioned above, Wireshark will be activated running on an Ethernet port in promiscuous mode in order to capture the traffic passing through the port. These traffic, data and statistics are later used in the research for analysis.

1.1 Related Work

Future Network Requirements. Future of networking and communication will occupy the space of the current paradigm as predicted by the year 2020. As predicted by technological giants and experts the 5^{th} generation (5G) will be the foundation of the future networking and internet. Due to the complexity and heterogeneity of tomorrow's communication METIS have presented six Horizontal Topics (HT) which can be used to build the foundation of network and communication for tomorrow. The HT's are as follows [18].

- Direct device-to-device communication (D2D)
- Massive machine communication (MMC)
- Moving Networks (MN)
- Ultra-dense networks (UDN)
- Ultra-reliable communication (URC)
- Architecture (Arch)

Table 1 presents a brief summary and a description of the topics that will make future networks possible. Or in a sense, they provide research avenues for both academia and industry to evaluate the current network, invest and implement in features that will enable tomorrows communication possible. As per the author's at [4] *et al.* presents three major use cases or case studies for 5G, which can stated as deduced from the above categories. Although the requirements are similar for the categories, the terminologies given are different. They are

- Enhanced Mobile Broadband (eMBB)
- Ultra-reliable low-latency service (URLLC)
- Massive machine-type communication (mMTC)

By understanding the terminology, it is clear that the above three was derived from the original six aforementioned. Although the requirements for each of the above stated remains the same from the previous, the above are specifically for an environment which facilitates 5G. It is arguable if the future of network presented by various author's were only for 5G or does it falls under the category of 5G and beyond, but following is related research that were conducted in the realm of 5G and future networking.

Related Work. Following is a description of significant work conducted in the realm. Author's at [19] presented a novel approach, known as Softbox. The novel approach was able to reduce the signalling overhead, data plane delay and CPU usage. The author's utilized a P4 core in order to reduce the SDN signalling overhead with a redesigned virtual core. The architecture also facilitates a future networking requirement where the author's were able to achieve a significant reduction in delay.

Research presented by [20] follows a novel approach where the author's distinguishably remark and account the traffic which is present in the network at the time of applying optimization algorithms. Although this research is not in-line with fronthaul or C-RAN, the same logic of accounting traffic which is present in the network is taken into consideration.

Theoretical framework presented by [4] utilizes three different controllers for the future cellular networking infrastructure. Although the paper is in review of the latest applications of technologies, this research understands the utilization of a centralized controller to accommodate future traffic needs. It's application of three different controllers may create traffic overhead in the network. However application of P4 is not present in this framework.

Research presented by [21,22] represents the future of the network in an architecture. This architecture also known as SELFNET utilizes SDN controllers, actuators and sensors in order to carry out functions in the control plane. As stated previously this research encourages the application of a controller to the core of the network but as a flavour of SDN configured using P4. Since P4 has the ability to provide programmability to the forwarding plane, the architecture which ever one presented could benefit from it's application.

1.2 Research Gap and Motivation

The P4 programming language and it's components were delivered in a way to overcome drawbacks of OpenFlow. As per the author's [5] P4 intend to be OpenFlow 2.0. The research is to understand the two potential supporting pillars of future networks and to understand the equilibrium of the two and how it could potentially support the KPI's of future networking and requirements of the software which runs on the network.

To the best of our knowledge, a research has not been conducted in the realm of understanding the performance matrices of the two underlying switching mechanisms. To the best of our knowledge no research has evaluated the two enablers of future networks, SDN independently and SDN with P4. To the best of our knowledge, experiments have not planned nor conducted in accordance with the ISO OSI layer. The experiments will be described in the Sect. 4 in detail with illustration for the ease of understanding.

The primary motivation for the research was derived in-order to understand the performance of the two technological enablers for the networking and the equilibrium of the two. As the gap was derived from referring to the most recent research, the motivation is to understand the performance at it's equilibrium where the P4 has the ability to compliment the performance of a SDN aware

Table 1. Description of Horizontal Topics (HTs) along with the requirements as presented by METIS and elaborated by [18]

HT	Description	Requirements
D2D	Direct communication between devices User plan traffic is hidden, doesn't traverse through the network Minimum interference	Increase coverage Fall-back connectivity Max spectrum utilisation Max capacity Offload backhaul
MMC	Provides two way connectivity to a large number of devices	Data rate Latency Cost
MN	Provide coverage for devices that are part of jointly moving	Communicate with the environment Location awareness
UDN	High traffic demands via infrastructure densification	Increase capacity of radio links Increase energy efficiency of links Better exploitation of spectrum Cost effectiveness Reduced interference Multiple access nodes
URC	Provide high availability	High availability Cost effective Reliability Short response latency
Arch	Platform integrating centralized and decentralized approaches	Heterogeneity Target independence

environment. P4 will deliver independence from underlying protocol and hardware having the ability to provide reconfigurability to the network. Since P4 is designed to conduct the routing based on a match+action table has the ability to compliment the performance of the network. Another motivation can be derived as the need to increase the performance and the productivity of the core network that can reduce the latency of the network as a whole. As per the literature [2], more latency occurs at the core network rather than the latency at the edge or at the end of the network. Hence the motivation was derived.

Since the application of P4 holds merit for networking for tomorrow, more research will follow in future to further support our hypothesis and statistics. The research in future was also considered as a motivation for carrying out this research as a stepping stone. The future work utilizing the P4 enabled SDN environment will be outlined in Sect. 5.

1.3 Contribution

The contribution of the paper are as follows.

- An in-depth analysis of the factors/data/statistics contributing towards increasing performance of the network.
- Experiments benchmarking the ISO-OSI Layer to further clarify and justify the intended use of experiments.
- To the best of our knowledge the two forwarding mechanisms in the data plane have never been tested or evaluated with equal resources given to facilitate a core network for a service provider.
- To the best of our knowledge these data/statistics have not been collected while noise is present in the network.

Structure of the paper is as follows. Section 2 represents the design and the methodology of the experiments which follows a description of the technologies utilized in the domain. Section 3 illustrates a detailed description of the system model. Section 3.1 describes the experiments conducted. Section 4 is a description of the data and an analysis that follows with conclusion and future work in Sect. 5.

2 Design and Methodology

As previously mentioned briefly, the research will consist of two environment with the exception of SDN and P4 against an environment configured with SDN only. The design of the topology is a simple ring with two core switches and two edge switches. An illustration can be found at Figs. 4 and 5. The illustration is self-explanatory for the description provided above. The experiment will consist of independent variables, environments or software such as ONOS, Mininet, Iperf and VLC-wrapper in order to conduct experiments whilst the depending variable will be a SDN flavoured switches in one instantiated environment, P4 switches configured in a SDN environment in the later.

The topology was designed using a python script which is capable of pushing the configuration to Mininet in order to simulate a networking environment. A custom python script was used for the SDN only environment and the bmv2 was used for the later to instantiate P4 switches in the ONOS environment. Since the topology is also an independent variable, the configuration is uniform across both environments.

2.1 System Platforms

Mininet. Mininet [12] is a tool developed at Stanford University which is an open-source, easy-to-deploy and a light-weight network emulator capable of providing a programmable interface to define and build networks and configurations with virtualized elements. It's original intention stands today to alleviate the cost of experimentation by utilizing virtualized resources and software and to perform network testing extensively [6]. Since Mininet has the ability to rapidly prototype large networks and its functions on a single physical computer, this tool has attracted its popularity with the research community to conduct experiments and tests. Tests and experiments not limited to but specializing in OpenFlow [7].

Open Networking Operating System (ONOS). In the recent years, SDN attracted attention from both academia and industry. OpenFlow allows network operators and administrators to replace expensive commodity proprietary hardware with open-source operating systems that has the ability to evolve and scale in time. A such OS has the ability to manage, monitor and programme network switches that facilitates applications and services across a wide range of hardware [8].

To facilitate the above and the KPIs for future networks and applications that is forecast to run on the hardware, ONOS [3] was developed and was launched in April 2013. ONOS is an open source SDN network operating system built for service provider networks [10]. ONOS provides the control plane for a SDN network components such as switches, links etc. and running software, programmes, applications or modules to facilitate and provide communication services to hosts and neighbouring networks [11]. ONOS provides high scalability, high performance and resilience and highly scalable option which makes it the best choice for building next generation SDN and Network Functions Virtualization (NFV) solutions. The GUI provides a global view of the network and its applications whilst maintaining its priority on performance and resilience. Development of ONOS paves the operators and administrators with the ability to vendor neutral control of data plane resources with provisioning capabilities which includes route calculations [9].

ONOS has two APIs, southbound and northbound while its core is responsible for maintaining the network state as mentioned earlier without compromising on the performance. ONOS interacts with the network devices via southbound APIs and Northbound APIs offer services to the applications. ONOS provides three network abstractions at different levels. Flow Rules is responsible for configuring the forwarding logic in devices by abstracting the protocol, Flow Objectives abstract the pipelines of the device and the third level is Intents where it abstracts the topology [10]. Figure 1 is a representation of the different tires of ONOS architecture differentiating the modules and their functions.

ONOS can be configured to run a distributed system (clusters) across multiple servers, allowing it use the CPU and memory of all the underpinned servers whilst maintaining performance and fault tolerance in the face of server failure. This ability of ONOS provides the capability of potential live updates/upgrades without having to reduce/gracefully regrade performance of the system. These updates/upgrades could vary from hardware upgrades to software updates [11].

Methodology is as follows. The environment is pushed utilizing the Mininet API with ONOS as the controller. The distinction between the two environments is detailed in the Table 2. The experiment column will be further explained at Sect. 3. Case study as described in Sect. 2.2 is designed in order to experiment the two platforms which can easily highlight the intended layer in ISO OSI structure. The underlying topology is unique and uniform in both instances.

The experiments which were conducted in both environments were planned meticulously to be in lined with the industry standard. Which is ISO OSI layer. The said statement can be illustrated using the Fig. 2. The experiments were

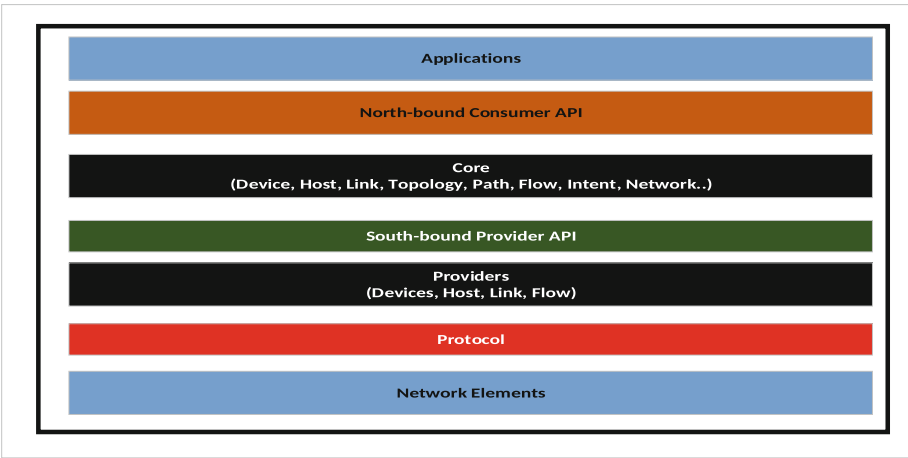

Fig. 1. The illustration represents the ONOS with differentiated tires of functionalities as presented at [11]

Table 2. A summary of the experiments that are conducted in the research, outlining the variable, software and platforms used.

Case	Experiment	ISO OSI Layer	Platform	Software	Data gathered
Case 1	LLDP	Layer II	Ubuntu 16.04 ONOS Mininet	Wireshark	No of Packets Undiscovered Packets
Case 2	ICMP	Layer III		Ping Wireshark	End to End Delay
Case 3	UDP	Layer IV		x-Term Iperf Wireshark	Transmission Delay Throughput
Case 4	Video Streaming	Layer V, VI, VII		x-Term VLC wrapper Wireshark	Quality of Image

built on top of the one which was conducted previously. i.e whilst UDP is passing through the network a separate ICMP is stream and a LLDP stream is present in the background.

2.2 Case Study

The main motivation or the objective of this research is to evaluate the performance of the two forwarding mechanisms that have been presented in literature. Evaluation is conducted in many ways by following several in-depth analysis.

SDN is widely used in current internet infrastructure and forecast to grow even more so with the application and availability of 5G. Whereas P4, a novel approach aims at out performing OpenFlow whilst opening the possibility to achieve the KPIs with respect to the performance inside the core network.

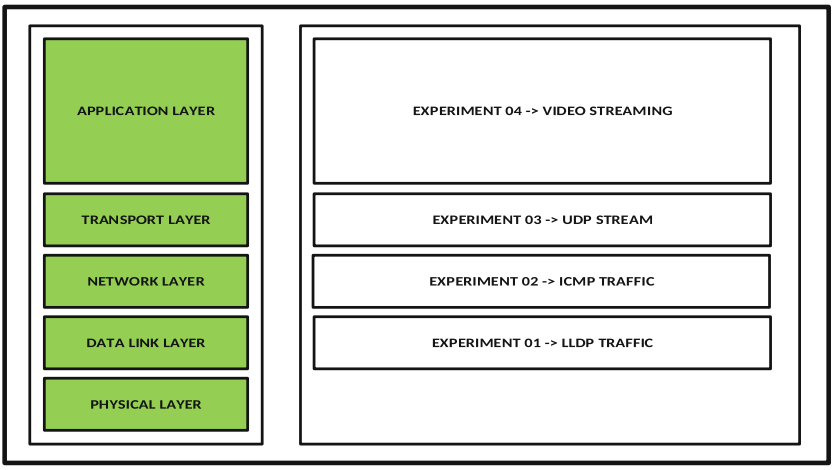

Fig. 2. A visual representation of the experiments which were described in Table 2 are to be in line with ISO OSI layer of communication.

The topology, as illustrated at Fig. 5 is a representation of the underlying network which is configured using python which was pushed towards Mininet. The network consists of four switches, two as edge and two at the core. Two hosts are connected at either ends of the network to the two edge devices. These two hosts (not illustrated) resembles the functionality of the servers, DB, Content Data Centres etc.

The case studies are as follows. There are four primary case studies involved in this research. Each case study involves one or two experiments in their respective environments, which totals the number of experiments to eight. A summary of the experiments along with the software used on each case study is illustrated in Table 2. Following is a description of the experiments conducted.

A core network consists of traffic which is created autonomously without the intention of the administrators. Which is considered as noise, elephant traffic, periodic updates etc. in literature. This traffic is a crucial factor and a metric to consider when an optimization theory is calculated or implemented on the network. Hence the network this research has employed, consists of such traffic simulated in Mininet. The first experiments wraps around the concept of the aforementioned. This experiment functions at the link layer of the network which will be further explained at Sect. 3.

In order to reduce the latency at the core of the network, End-to-End delay must be accounted as a primary experiment. This statistic will significantly

provide evidence, which will illustrate whether the medium or the content will be available to the user with the lowest possible latency. Whist conducting the experiment, traffic previously mentioned (LLDP) is also available in making the data realistic. This experiment will be in-lined with the network layer (Layer III) of the ISO OSI layer. As mentioned previously, traffic will be captured by Wireshark to conduct further analysis.

This research was designed or planned in a way which has the ability to compliment the experiments conducted previously. As such it builds on top of the experiment which was conducted previously. Transport layer is the forth layer of the ISO OSI architecture. With the evolution of applications which will be used in the edge devices or in user equipments, faster converging protocols will be employed in the network. With this concept in mind protocols such as UDP, RUDP will be employed heavily. A stream of traffic will be sent from Iperf client to server in the experiment. Aforementioned transmission will be captured via Wireshark for statistical analysis. This will be further explained at Sect. 3.

The final stage of the case study is to in-cooperate the functions of the session, presentation and application layer of the ISO OSI Layer. This is done via a video stream from one end to the other. As mentioned previously, the noise in the network is still present during transmission of the content.

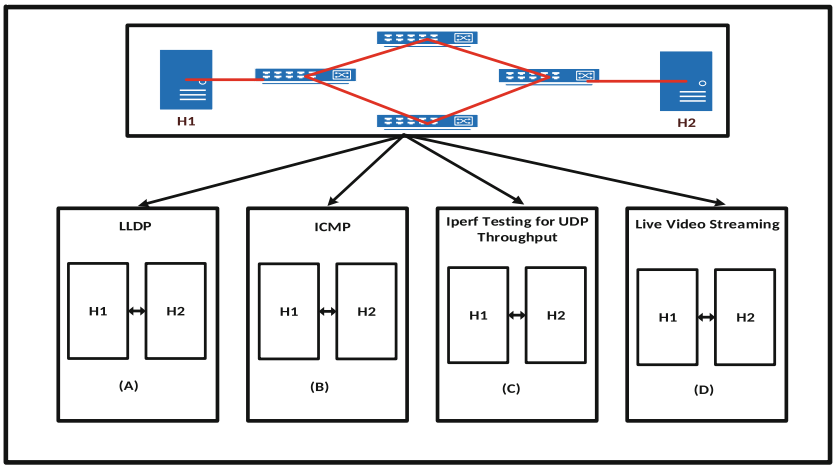

Fig. 3. The experiments conducted under the ring topology utilizing the two distinct variables to test the hypothesis.

3 System Model

In order to test and evaluate the two underlying switching mechanisms, two VMs were deployed using VmWare [17]. Each of this virtual machines are given 16 GB of RAM, 60 GB HDD and 4 CPU cores with Ubuntu 16.04 LTS as the guest operating system. Wireshark was installed on both VMs monitoring the

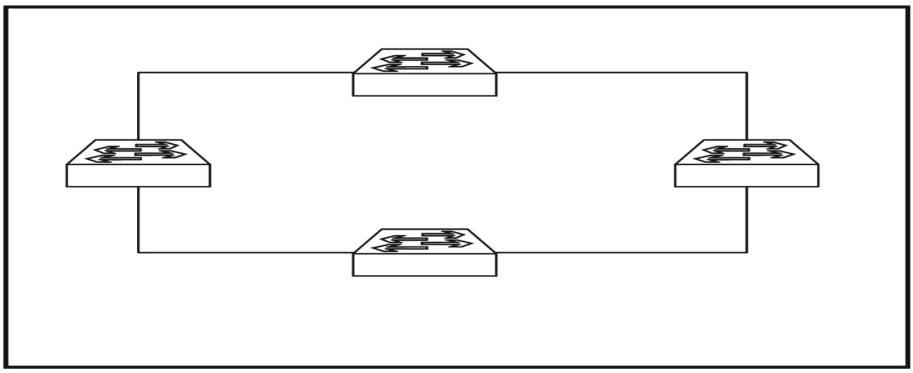

Fig. 4. The underlying network architecture for the core designed to test the two forwarding mechanism.

virtual Ethernet ports that are configured using the Mininet. The topology was described and fed to Mininet using a python script and ONOS was installed in both instances with the exception of one environment configured to operate P4 switches with the former configured to operate under OpenFlow.

Figure 4 is a representation of the environment which was used in order to carry out the tests. It consists of four switches connected in a ring topology with equal weight and bandwidth in each link. The exception as mentioned above is the differentiation of the forwarding mechanism. SDN or SDN+P4. Two Hosts are connected at either ends to the two edge switches. Since the illustrated environment is a representation of the core network, the experiment seeks to answer the KPIs, for the future networks. Hosts in the network are an illustration of a server, end point, processing agent, UE etc. but for the purpose of the experiment the aforementioned hosts are capable of a PC's processors which is supported by Mininet. Figure 5 represents the two environments with variable (P4 or SDN) difference.

Figure 3 represents the experiments which were conducted in order to test the hypothesis of the best underlying forwarding/switching mechanism for the core network of a service provider. These experiments are in-lined with the KPIs presented in section I towards future networks. The list of experiments that have conducted on the environment can be found at Table 2. An illustration of the said experiments in line with the ISO OSI layer can be found at Fig. 2.

Case Study I. The following is an elaboration of the experiments conducted in a detailed manner. Figure 3(a) illustrates the first experiment conducted. The application of a controller requires periodic updates to be sent over the network in order to keep track of links, hosts, servers, elements, traffic etc. This traffic is mandatory for the function of the environment. The future networks requires crucial availability of the medium, hence this type of traffic is mandatory. The periodic traffic ensures that the services and servers are aware/live and kept

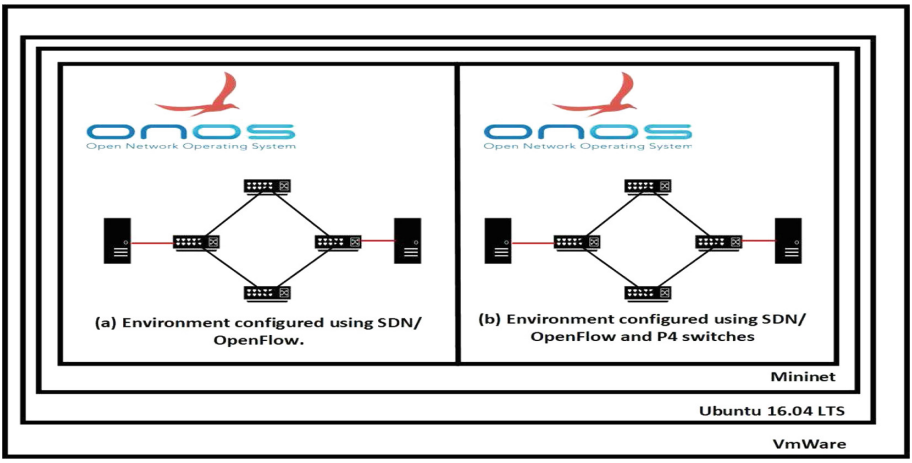

Fig. 5. Illustrated description of the two environments. (a) Represents the environment which is configured using OpenFlow. (b) Represents the environment configured using P4 switches. As illustrated ONOS is acting as the controller with Mininet emulating the network with Ubuntu 16.04 LTS as the guest OS on VmWare.

at ready state. These updates must be accounted for when an improvement algorithm is being placed in the network. Hence this traffic is a crucial factor for administrators. Wireshark was used in order to monitor and capture the LLDP packets travelling inside the network.

Case Study II. Figure 3(b) represents the experiment where a continuous stream of ICMP messages were sent from H1 towards H2. The controllers will decide the best route for the stream to traverse. The selected route is visible via accessing the ONOS GUI. A total of 1300 packets are transmitted from H1 to H2. Wireshark with sudo privileges posses the capability to listen to traffic at a given Ethernet port. Packets are collected at the destination node utilizing Wireshark. Above experiment is conducted on both environments, SDN and SDN + P4.

Case Study III. Figure 3(c) represents the third test/experiment which was conducted in the network. An Iperf test was carried out where H1 acts as the Iperf Server and H2 as the Iperf Client. The future network requirements will provide new avenues for various applications to run on the network and on the host devices. In the instance of such higher data rates must be employed with highly efficient transmission protocols. Hence protocols such as UDP, RUDP and other protocols of same resemblance will have more merit. In order to test the latency and support the hypothesis a UDP stream is generated via accessing the x-term of the hosts utilizing the Mininet simulator. Data transmission can be monitored via accessing the Wireshark with sudo privileges listening to the

relevant Ethernet port. One of an additional advantage is that the protocol functionality along with the port information can also be viewed using Wireshark.

Case Study IV. Forth test compliments the third experiment and have the ability to support the hypothesis even better. Since Iperf sends a randomly generated stream of UDP data between a client and a server, a live video stream amongst the two hosts will carry actual UDP data. The transmission is conducted utilizing the vlc-wrapper by accessing the x-term of hosts. In the experiment H1 will be considered the content network or the content provider whilst H2 will be considered the service or the server/edge server/user equipment/host requesting the content. Wireshark with sudo privileges as mentioned above has the ability to listen to live stream of traffic passing in the network with port information. The test results are qualitative in the underpinned experiment whilst previous tests are consisting quantitative data. The quality of the video is examined by understanding and capturing the frequency of frame lag in the video from original. Displaced pixels and frames arriving late will also be monitored and recorded. Total number of UDP packets will also be recorded in order to test and evaluate the quality of the video.

4 Data and Analysis

The following is a discussion of the data which was gathered at the end of each experiment and the analysis that follows of the results. The results which are presented here are in the same order as it is shown in the Table 2 and the Fig. 3. The distinction of the environment along with the variables that were used to configure the aforementioned is illustrated at Fig. 5. The data were collected utilizing a well-known tool Wireshark as previously mentioned.

4.1 Traffic in the Idle State

The data were collected utilizing the network monitoring tool Wireshark. Upon sending the network configuration file to the Mininet API, by default LLDP packets are generated due to the design of the controller. These packets serve a purpose to the controller, in which the controller is fully aware of the changes in the network since the network devices advertise information about themselves to their neighbours. These traffic is a crucial factor for future networking since the aforementioned traffic will be present in the network at any given time.

Figure 6 represents the traffic in the environment configured with SDN when there's no traffic is available and in idle state where as Fig. 7 represents the same traffic when a P4 switch is present in an environment with SDN.

It is a clear representation of the traffic which is passing through the network when it is in idle state. The Wireshark has the ability to listen to traffic at a given time in a given specific port. By careful observation of Fig. 6, we can observe points where the LLDP have recorded one packet instead of two. These points are highlighted in the plot as well. They are (54,1), (57,1), (64,1), (104,1)

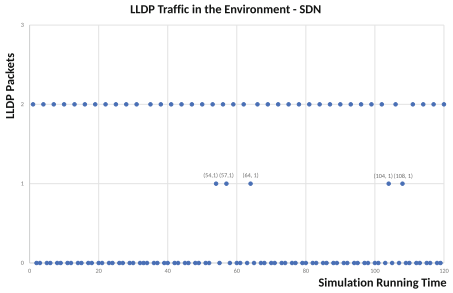

Fig. 6. LLDP traffic in SDN

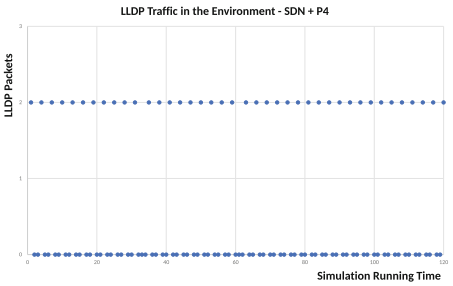

Fig. 7. LLDP traffic in SDN+P4

and (108,1). Although it may seem like a drop in the network, the ambiguity of the network could cause delay or service unavailability in the future applications which requires trivial access and availability of the content and medium. It is evident that the environment when configured with the SDN and P4 has a constant pattern. Since the match+action table consists of neighbour information and routing information the future networks will not have to face moments of ambiguity.

For the purpose of this research, simulation time was 120 s. Hence the noted points of ambiguity in the network is as low as four points. If a higher sample size was chosen with more nodes connected to the controller, more results can be observed. But for the purpose of this research, the chosen sample size is 120 s.

4.2 ICMP

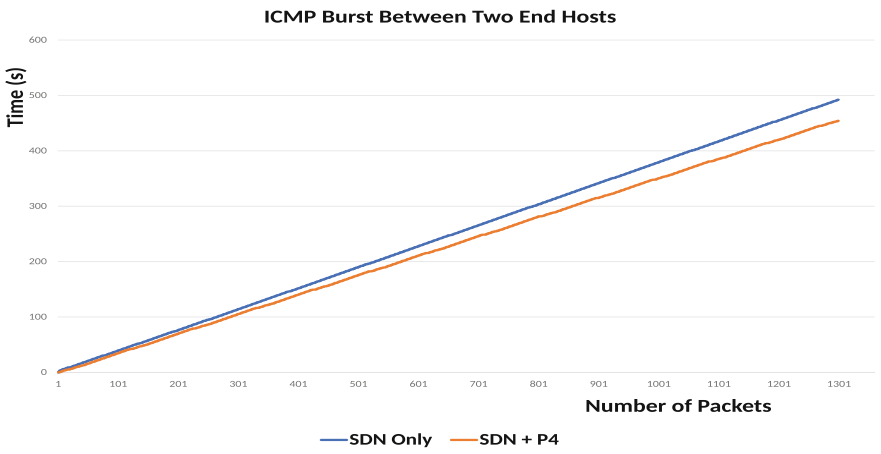

Fig. 8. Illustration of the ICMP bursts in the two environments for 1300 packets (Color figure online)

ICMP is traffic is sent from H1 towards H2 utilizing the Mininet CLI. The traffic can also be generated using the x-term, but for the purpose of the experiment traffic is generated using a simple command in the Mininet CLI. The logic behind the experiment is that the minimum time spent on the process will provide the future environments with the lowest form of latency. Since the networks of tomorrow requires faster traverse of traffic in the core of the network, this experiment can provide feedback and evidence to support.

Figure 8 represents the ICMP burst between the two end hosts in the same graph with SDN only environment in blue and SDN + P4 in amber. It is evident that SDN + P4 requires a smaller time frame (452 s) to traverse the ICMP packets to the end host where as SDN only environment requires a higher time (492 s) comparatively to the latter. Destination host was discovered after 3 s in the SDN environment where as the SDN + P4 environment consumed 0.23 s. Although the time gap is not significantly greater, the future network requires 5 times reduced latency, refer Sect. 1.

Mathematical difference between the two is as little as 40 s. This difference can be observed in a more significant scale if the sample size was increased. But for the purpose of this experiment a sample size of 1300 packets were chosen. If more number of packets were captured containing ICMP traffic more of a difference will be able to identify. Although this is the case, the route request time will have no effect. This is because it represents the time which is required by the controller to define the route for the packets to traverse. Increasing the sample size will have no effect on the end result since it directly speaks of the effectiveness of the controller.

4.3 UDP

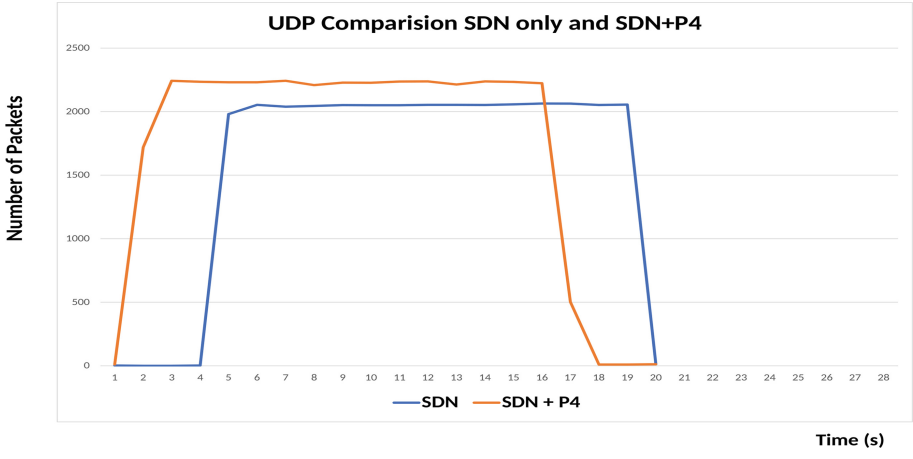

Fig. 9. Illustration of the UDP stream of between the two hosts

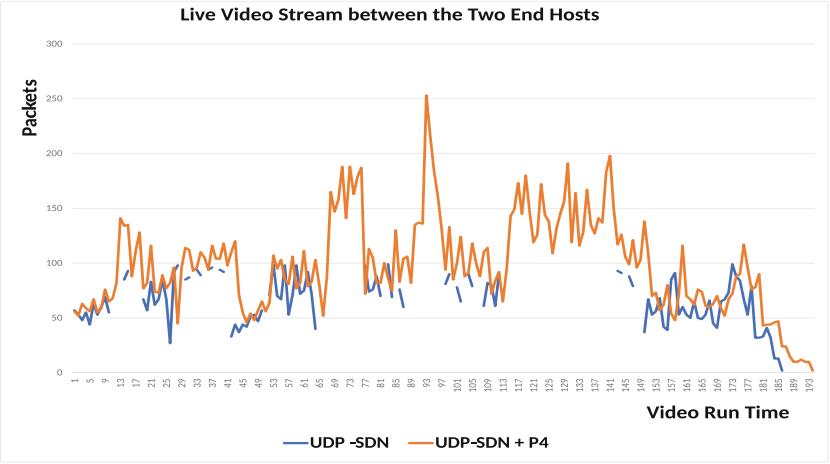

Fig. 10. Live video stream UDP capture utilizing wireshark

Figure 9 is a representation of a capture file which was derived from Wireshark to the traffic which was generated using x-term in the Mininet for an Iperf test of a UDP stream. Out of the two hosts, H1 carried out the Iperf server functionality whilst H2 as the client. A data burst of 25 Mb was sent for the duration of 15 s in order to make the noise in the network create effect and a larger data burst would make the noise negligible.

The line of blue in the Fig. 9 represents the stream of UDP traffic in the environment configured using SDN and a delay of 4 s can be observed which resulted the stream to end with a delay of 3 s. The Data stream started with a delay due to route discovery.

The line in amber represents the data transmission of UDP between the two hosts in the environment with a SDN controller and P4 switches. The data transmission didn't experience a delay and arrived in the desired time frame releasing the network for future transmission. Which indicates efficiency and availability of the network. Also SDN + P4 environment reached a higher data burst as well and was able to maintain the said stream consistently. Figure 10 will provide more evidence to the statements above.

4.4 Live Video Streaming

The Fig. 10 revels the data transmission between the two hosts during the live video stream. For the purpose of fairness and equality same video was used of same size and ratio. For the purpose of the video stream, vlc-wrapper was used and data was captured during the transmission utilizing Wireshark.

As in previous diagrams, SDN is represented using blue lines whilst amber represents the UDP stream of SDN + P4 switches. As represented the blue line of the diagram tend to have a significant drop or below the expected rate

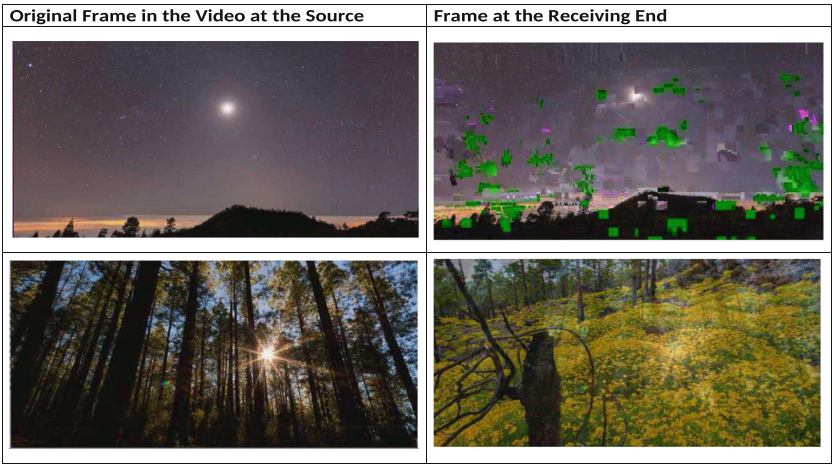

Fig. 11. Live video stream between the two hosts, demonstrating the quality of the frames in the environment configured with SDN

compared to amber line. This cause for the video to experience distorted pixels and become low in quality at unfrequented times. However the video stream which is represented in amber seems to have significant performance since the video show no disoriented pixels or frames with video arriving at the destination as the same time (with an insignificant delay) as it is being displayed at the local source. In both these scenarios background traffic was present in order to simulate a real-life environment. The high points in the Fig. 10 seems to be overlapping with each other but a significant difference is visible at the start of the video where the link tends to be busy with existing traffic. Still the SDN + P4 achieved the first high data burst compared to SDN only environment making the video more accessible/high in quality to the destination. This is because the application of P4 switches and its function of the match+action table assist the controllers with route discovery and routing.

The above paragraph can be further justified by the Figs. 11 and 12. The Fig. 11 represents the video stream of the environment configured using SDN. Left hand two images represents the video displayed locally whilst the right hand side is the video at the destination. As captured, the video is of low quality with disoriented pixels visible on the frame. The low points of the Fig. 10 represents these disoriented images. Figure 9 shows a delay in the start of UDP stream to occur. The third and fourth images of the Fig. 11 can be used as evidence to further support the findings. The image on the right (fourth) can be seen in a mode of transition (trees on the adjacent left hand image are visible) from the existing frame of the video. Hence the significant delay in delivering the packets can be observed.

Figure 12 represents the video stream in the environment configured in SDN + P4. As shown in the figure quality of the picture shows a significant

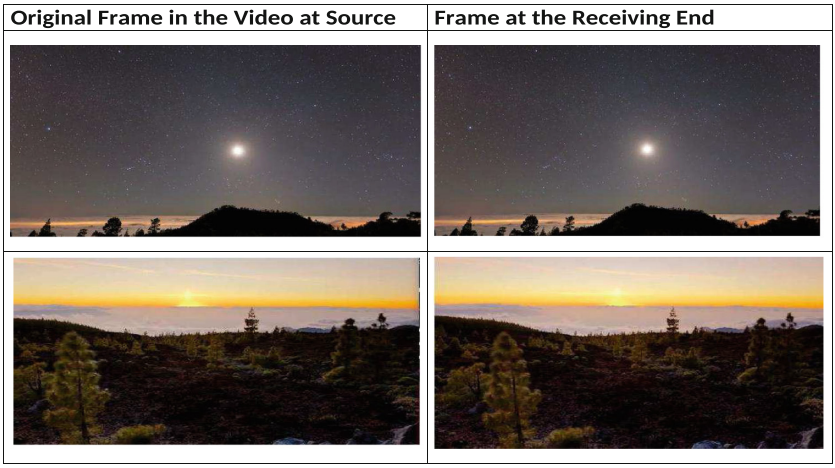

Fig. 12. Live video stream between the two hosts, demonstrating the quality of the frames in the environment configured with SDN+P4

improvement compared to the one configured in SDN only. The frames seem to arrive at the destination with a least significant delay. By close inspection of the images from left to the images on right in the Fig. 12 a slight delay of frames can be observed. But this is insignificant compared to the delay of frames which are arriving at the destination in the Fig. 11. This is further supporting the Figs. 8, 9 and 10. With the background traffic, the controller seems to be complemented with the effect, programmability the P4 has offered.

By further understanding the data that were gathered, following conclusions can be derived. Compared to the SDN only case study, a decrease of 92% can be observed in the time controller spent on destination host discovery. Also a total of 7% reduction in the case study SDN + P4 can be observed in the total E2E compared to the SDN only environment. UDP transmission delay has a significant reduction in SDN + P4 environment of which is 70%. The above statistics significantly improved the quality of the video feed in the last experiment. 8% increase of quality can be marked in the SDN + P4 environment compared to the SDN only case study. Hence the quality of the video was observed. The statistics are all leading up to confirming that the future of networking can benefit greatly from applying a fabric of SDN + P4 as the underlying switching mechanism. The experiments conducted, data gathered and the calculated statistics are all providing evidence to the statement above.

5 Conclusion and Future Work

By a series of test and experiments expanding over the ISO OSI layer of the two environments with four use cases, we can conclude the performances of the environment of SDN + P4 switches stands out superior to the environment

without P4. Hence it is wise to incorporate P4 switching mechanisms in the core of the network for a service provider to better utilize resources and provide services with minimal latency and minimal ambiguity.

Following this research, we aim to utilize the environment to further run experiments with a fully functioning core which includes various services and databases. We also aim to simulate the aforementioned in real equipments as oppose to Mininet to further support the findings of this research.

References

1. Wang, H., Chen, S., Xu, H., Ai, M., Shi, Y.S.N.: A software defined decentralized mobile network architecture toward 5G. IEEE Network **29**(2), 16–22 (2015)
2. Monserrat, J.F., Mange, G., Braun, V., Tullberg, H., Zimmermann, G., Bulakci, Ö.: METIS research advances towards the 5G mobile and wireless system definition. EURASIP J. Wirel. Commun. Network. **2015**(1), 53 (2015)
3. ONOS project. https://onosproject.org/. Accessed 19 Mar 2019
4. Li, R.: Intelligent 5G: when cellular networks meet artificial intelligence. IEEE Wirel. Commun. **24**(5), 175–183 (2017)
5. Bosshart, P., et al.: P4: programming protocol-independent packet processors. ACM SIGCOMM Comput. Commun. Rev. **44**(3), 87–95 (2014)
6. Muelas, D., Ramos, J., de Vergara, J.E.: Lopez assessing the limits of mininet-based environments for network experimentation. IEEE Network **32**(6), 168–176 (2018)
7. De Oliveira, R.L.S., et al.: Using mininet for emulation and prototyping software-defined networks. In: 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), pp. 1–6 (2014). Organization IEEE
8. Berde, P., et al.: ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, pp. 1–6 (2014). Organization ACM
9. Giorgetti, A., Secondini, M., Cugini, F., Sgambelluri, A., Castoldi, P.: ONOS MetroApp for filtering effect assessment in metro optical networks. In: 2018 European Conference on Optical Communication (ECOC), pp. 1–3 (2018). Organization IEEE
10. Sanvito, D., et al.: ONOS Intent Monitor and Reroute service: enabling plug & play routing logic. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 272–276 (2018). Organization IEEE
11. ONOS project. https://wiki.onosproject.org/. Accessed 20 Mar 2019
12. Mininet. http://mininet.org/. Accessed 24 Mar 2019
13. Cisco Visual Networking Index: Forecast and Methodology, 2016to2021, Cisco Public, (2017). Cisco
14. Cisco The Zettabyte Era: Trends and Analysis, Cisco Public (2017). Cisco
15. P4 Language Consortium. https://p4.org/. Accessed 24 Mar 2019
16. Wireshark. https://www.wireshark.org/. Accessed 24 Mar 2019
17. VM-Ware Workstation. https://www.vmware.com/uk/products/workstation-pro/workstation-pro-evaluation.html. Accessed 24 Mar 2019
18. Osseiran, A., et al.: Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE Commun. Mag. **52**(5), 26–35 (2014)

19. Moradi, M., Lin, Y., Mao, Z.M., Sen, S., Spatscheck, O.: SoftBox: a customizable, low-latency, and scalable 5G core network architecture. IEEE J. Sel. Areas Commun. **36**(3), 438–456 (2018)
20. Luong, P., Gagnon, F., Despins, C., Tran, L.-N.: Joint virtual computing and radio resource allocation in limited fronthaul green C-RANs. IEEE Trans. Wirel. Commun. **17**(4), 2602–2617 (2018)
21. Jiang, W., Strufe, M., Schotten, H.D.: Intelligent network management for 5G systems: the SELFNET approach. In: 2017 European Conference on Networks and Communications (EuCNC), pp. 1–5 (2017)
22. Jiang, W., Strufe, M., Schotten, H.: Autonomic network management for software-defined and virtualized 5G systems. In: European Wireless 2017; 23rd European Wireless Conference, pp. 1–6 (2017). Organization IEEE