

Article

Routing Performance Evaluation of a Multi-Domain Hybrid SDN for Its Implementation in Carrier Grade ISP Networks

Babu R. Dawadi *, Abhishek Thapa, Roshan Guragain, Dilochan Karki, Sandesh P. Upadhaya and Shashidhar R. Joshi

Department of Electronics and Computer Engineering, Pulchowk Campus, Tribhuvan University, Kathmandu 19758, Nepal; acevalentyne5@gmail.com (A.T.); guragainroshan0@gmail.com (R.G.); karkidilochan.dk@gmail.com (D.K.); sandeshupadhaya204@gmail.com (S.P.U.); srjoshi@ioe.edu.np (S.R.J.)

* Correspondence: baburd@ioe.edu.np

Abstract: Legacy IPv4 networks are strenuous to manage and operate. Network operators are in need of minimizing the capital and operational expenditure of running network infrastructure. The implementation of software-defined networking (SDN) addresses those issues by minimizing the expenditures in the long run. Legacy networks need to integrate with the SDN networks for smooth migration towards the fully functional SDN environment. In this paper, we compare the network performance of the legacy network with the SDN network for IP routing in order to determine the feasibility of the SDN deployment in the Internet Service provider (ISP) network. The simulation of the network is performed in the Mininet test-bed and the network traffic is generated using a distributed Internet traffic generator. An open network operating system is used as a controller for the SDN network, in which the SDN-IP application is used for IP routing. Round trip time, bandwidth, and packet transmission rate from both SDN and legacy networks are first collected and then the comparison is made. We found that SDN-IP performs better in terms of bandwidth and latency as compared to legacy routing. The experimental analysis of interoperability between SDN and legacy networks shows that SDN implementation in a production level carrier-grade ISP network is viable and progressive.

Keywords: ISP; SDN; SDN-IP; legacy network; performance comparison



Citation: Dawadi, B.R.; Thapa, A.; Guragain, R.; Karki, D.; Upadhaya, S.P.; Joshi, S.R. Routing Performance Evaluation of a Multi-Domain Hybrid SDN for Its Implementation in Carrier Grade ISP Networks. *Appl. Syst. Innov.* **2021**, *4*, 46. <https://doi.org/10.3390/asi4030046>

Academic Editors: Gregorio Procissi and Friedhelm Schwenker

Received: 23 May 2021
Accepted: 15 July 2021
Published: 21 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Networks are organized groups of devices or nodes with communication links among them. Traditional IP networks are termed legacy networks. Despite the world-wide use, the traditional network is complex and tedious to manage. Adding a new device or changing the network configuration is complex in traditional networking, that is, it should be implemented using low-level languages and rigid commands thus taking days and months for large networks. Thus, the expansion of the existing legacy network is fairly expensive. The demand for network based services is increasing rapidly. To meet those growing demands, the numbers of running hardware should be increased and changes should be made in software leading to a higher cost in system upgrades. Manual configuration, inconsistent policies and inability to scale are the major problems of the traditional IP networks. Vertical integration has made traditional IP networks more complex in their operation and management. Legacy networking devices have the control plane and data plane integrated into the same physical device. The data plane, also called the forwarding plane, is a hardware unit whose job is to collect the network packets and forward them to the destination based on the entries provided in the routing table. Network policies are enforced in the control plane. The control plane determines how the packets in the data plane should be handled, for example, drop, reject, forward and so forth.

In SDN, network devices are controlled by an SDN controller and the applications are installed on the top of the controller. When the control and data plane functionalities are segregated, the network switch becomes simply a data plane forwarding device. The control logic is implemented in a logically centralized controller, which runs a network operating system (NOS) that runs on commodity server hardware to provide necessary resources to facilitate the proper control and operation of data plane devices based on a logically centralized, abstract network view [1]. The logical centralization of the control logic makes it flexible and provides error-free policy deployment through high-level programming languages as an application programming interface (API), compared to the low-level device-specific configurations frequently used in legacy networks. The controller provides the global view of the overall network, which simplifies the development of more robust network functions, services and applications. The resulting network is also programmable through software applications running on top of the controller OS, which interact with the underlying data plane devices. The APIs at the controller can automatically react to spurious changes of the network state and keep high-level policies intact [1].

Though SDN implementation in datacenter networks is popular, its proper implementation with the migration of a legacy network to carrier grade ISP (CG-ISP) and telecommunication (Telcos) networks is becoming a central challenge for service providers due to the need for real time migration as well as the higher cost of investment [2,3]. Additionally, real time implementation of SDN in production networks is the subject of ongoing research. Hybrid SDN is the only solution for the smooth migration of legacy ISP/Telcos networks into SDN [4,5]. In this paper, an emulated network environment is created in Mininet for hybrid SDN implementation and routing performance evaluation by comparing legacy routing and SDN routing with their interoperability through an experimental analysis using an open network operating system (ONOS)/SDN-IP [6]. The viability of the SDN implementation in ISP/Telcos networks is then evaluated. The major contributions of this paper are as follows.

- A multi-domain hybrid SDN environment is created and implemented in the routing between SDN and legacy networking domains.
- Routing performance is evaluated, comparing legacy routing and SDN routing using ONOS/SDN-IP. Routing using SDN-IP has a better performance than legacy routing.
- This experimental analysis evaluates the viability of SDN implementation in ISPs and Telcos networks. This encourages the service providers to smoothly migrate their legacy networks into SDN.

The rest of this paper is organized as follows. Section 2 presents the background and related work in the field of SDN and legacy network integration. Section 3 describes the research problem and the proposed method. We present the details of the experimental setup, analysis, and evaluations in Section 4. Section 5 provides a discussion and suggests future work, and the paper is concluded in Section 6.

2. Background and Related Work

The cost of new technology implementation is generally higher in terms of capital and operational expenditure (CapEX and OpEX) investment and the development of technical human resources (HR) to manage and operate those newer technologies for ISPs and Telcos service providers. There are also certain implementation challenges with respect to management, availability of technological standards and user interface provisioning when providing new technology-based services to customers during and after network migration [7,8].

A legacy network is less flexible to customized programming and is more vendor specific, leading to higher dependency on support, management and operation. A better solution would be the implementation of SDN. In SDN, networks are controlled by software applications running on the top of an SDN controller. The separation of control and data plane operations in SDN simply transforms the network switches into simple forwarding devices.

NOS provides the essential resources and abstractions to facilitate the programming of forwarding devices based on a logically centralized, abstract network view. The resulting network is also programmable through software applications running on top of the NOS, which interact with the underlying data plane devices. ONOS [9] is the robust and distributed controller OS, which provides a better solution for building next-generation SDN/NFV solutions. The need for ISPs and Telcos service providers can simply be fulfilled by the introduction of ONOS to carrier-grade solutions that leverage the economics of white box merchant silicon hardware while offering the flexibility to create and deploy new dynamic network services with simplified programmatic interfaces [9]. ONOS is developed with a set of several other applications that make it flexible, modular, and scalable in terms of both the architecture and the cluster, and the distributed SDN controller. SDN-IP is one of the ONOS applications developed to implement routing with external legacy networks using the standard border gateway protocol (BGP) to enable interoperability with legacy routing. In ONOS/SDN-IP, the SDN can be treated as a single autonomous system (AS) and communicates simply with external AS, as with traditional routing. SDN-IP integrated BGP and ONOS enable communication with external AS in the hybrid SDN so that SDN-IP behaves like a regular BGP speaker and uses its services to install and update the appropriate forwarding state in the SDN data plane devices.

Dawadi et al. [3,10–12] presented approaches for legacy network transformation to SDN and IPv6 networking so that service providers can smoothly plan for network transformation with optimised migration cost. The authors implemented IPv4/IPv6 routing in a multi-domain SoDIP6 network using ONOS/SDN-IP.

Friyanto [13] presented the use of multiple ONOS controllers in a cluster for high available services using ONOS/SDN-IP reactive routing. Efficient routing implementations in the different aspects of wired and wireless SDN [14–17] are the primary concerns, but their implementations in production level networks have to be evaluated considering the smooth transition from legacy to hybrid SDN to pure SDN.

There are many studies on hybrid SDN implementations. For example, HARMLESS [18], Panopticon [19], RouteFlow [20], Fibbing [21] and OSHI [22] are some of the attempts, but most of this research has not been implemented at the production level. OSHI considers the hybrid switch, while our implementation only considers the hybrid SDN/legacy network as a mix network and routes implementations on that mixed network. Tomovic et al. [23] presented a modified version of three dynamic routing algorithms viz. MIRA, DORA and SWP, to be implemented with the OpenFlow controller. These routing modules are implementable only in an SDN environment.

Similarly, few studies have dealt with the implementation of SDN in CG-ISP networks using ONOS/SDN-IP [6,24–26], even though ONOS is the best controller in terms of its features and dedication towards transport SDN over a wide area network [27]. Kong et al. (2013) [28] evaluated the performance of an OpenFlow switch in SDN by comparing it with legacy networking and found that OpenFlow based communication did not perform well. During that period, SDN was just at the emerging stage but now it is well matured [29–31] and robust. Research on implementing SDN in an ISP network is in progress [24,32]. SDN is popularly implemented in data center networks [33,34] and its implementation in ISP networks is part of ongoing research. Because we are at the early stage of SDN implementation in service provider networks, we found limited or no research related to routing performance comparison in terms of BGP implementation in legacy networks and SDN in the service provider networks. Our approach in this study is not scoped towards migration techniques, instead we focused on the production level implementation of routing in hybrid SDN and its performance in comparison with legacy network to measure the viability of legacy network transition towards SDN by implementing ONOS/SDN-IP.

3. Problem Description and Proposed Method

The overall structure of our experimental evaluation is based on the generic architecture of ONOS/SDN-IP implementation [35] as shown in Figure 1 and generated the experimental output of our research use case using Mininet as shown in Figure 2. All the external networks, that is, external ASs, are supposed to be the legacy networks, directly connected with backbone transit AS, which is SDN based.

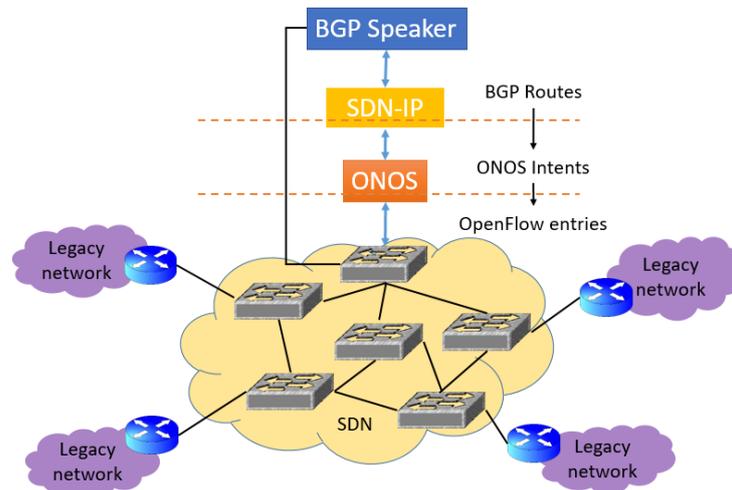


Figure 1. Use case network of SDN integration with legacy networks .

For routing purposes, the Quagga routing suite [36] was used, in which BGP was implemented. For SDN integration with a legacy network, the ONOS controller was configured to implement BGP and SDN-IP with reactive routing enabled.

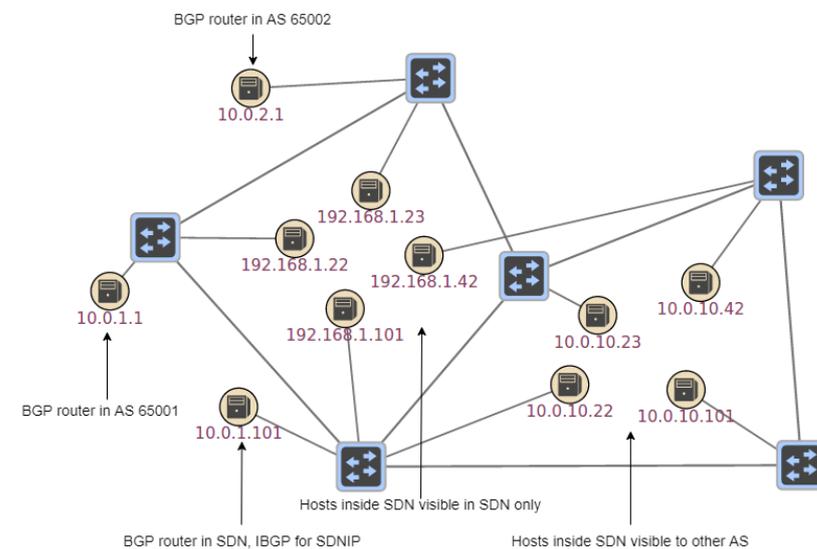


Figure 2. Experimental topology of SDN integrated with legacy networks. AS, Autonomous System, BGP, Border Gateway Protocol, SDN, Software-Defined Networking.

A complete legacy network was created as shown in Figure 3, where BGP and OSPF were used for routing purposes. Each router is a virtual machine loaded with BGP and OSPF configurations. In Figure 3, r5 is a route reflector router [37] and r4, r3, r6, r7, and r8 are its clients, whereas r4-r2 and r3-r1 are BGP peers.

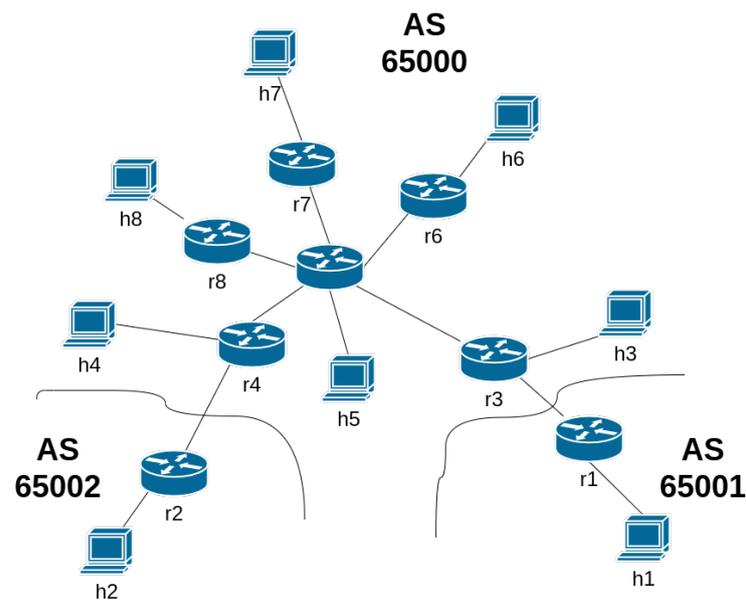


Figure 3. Use-case of legacy routing network.

The SDN integrated legacy network and pure legacy network were chosen in the present study to analyze network performance. For comparison between them (hybrid SDN and legacy network), a distributed internet traffic generator (D-ITG) [38], a Packet InterNet Groper (PING), Iperf, and statistical rules were applied. A varying number and size of packets were created and transmitted from one host to another in each network, and the corresponding delay, latency and total packet transmission were observed. The PING utility was used to check the network connectivity between hosts. In both networks, first, PING tool was used to ensure total communication in the network; then it was used for finding out RTT. We used IPERF tool to determine the maximum bandwidths offered by the network. The compatibility and performance of SDN integration with the legacy network for multi-domain routing purposes were clearly depicted by the statistical values and comparison techniques required for SDN and the legacy network. Mininet emulated the hosts as a Linux machine. ONOS, running on the same machine, was used as an SDN controller.

The experimental setup of this study is depicted in Figure 4. The experiment was executed in a virtual environment for which we used Oracle VirtualBox as the hyper-visor. The guest operating system was allocated with 4 GB of base memory and two processor cores. The experiment was carried out on the Ubuntu 14.10 (64 bit) operating system. The network topology file used Mininet to create routers, hosts and OpenFlow switches. Quagga version 0.99.23 was configured in the routers for BGP routing in SDN, while OSPF and BGP were configured for the legacy network. Routers and hosts received data routed through the OpenFlow switches. The ONOS (version 1.2.1 user) interface provided the visualization of the network. DITG, PING, and IPERF tools were used for the data collection. The collected data were analyzed and visualized using the Python Matplotlib module. A brief descriptions of the APIs enabled over ONOS are follows.

- **OpenFlow:** It is the southbound protocol in the SDN control plane. It enables communications between the data plane and the control plane devices so that the control plane deploys flow rules in the OpenFlow table of the data plane and provides a decision for every packet incoming to the data plane devices.
- **Config:** This application is activated for loading the network configuration.
- **ProxyARP:** It is the process of resolving the address of a host. It responds to address resolution protocol (ARP) or neighbor discovery protocol requests on behalf of a target host if the target host is known to ONOS (i.e., it is present and active in the host store). ProxyARP considers the network to be a single IP network, therefore, if it receives

a request for which the target host is not known, it will flood the request to every edge port in the network (apart from the input port) in the hope that the host exists in the network and will reply.

- **SDN-IP:** It is an ONOS application for routing implementation that allows an SDN to connect to external networks on the internet using the standard BGP.

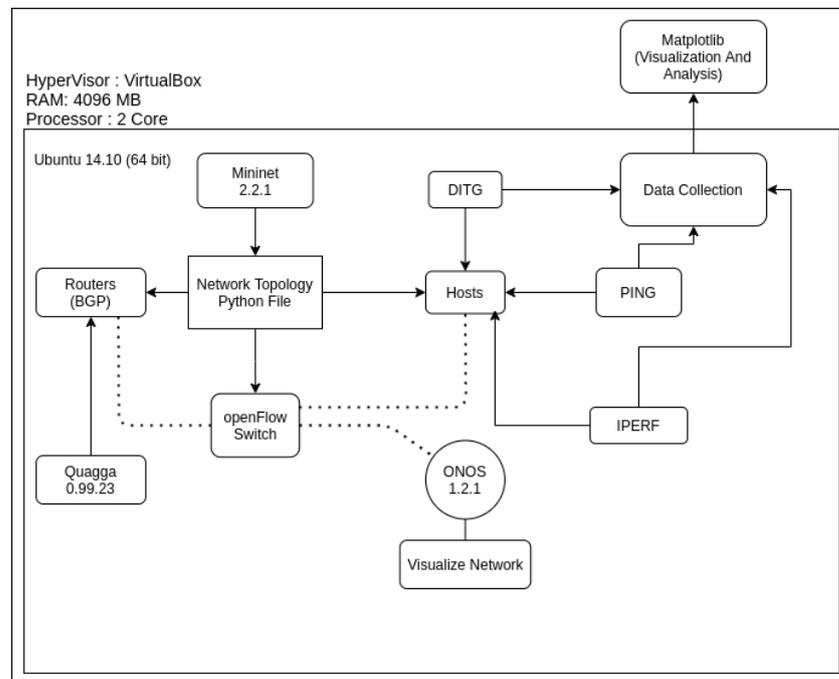


Figure 4. System setup simulation environment.

After running those APIs, we can see the routes learned by ONOS, which can be seen using the 'routes' command on the ONOS console. In the legacy ASs, an eBGP router is connected with a single host. The SDN network in our proposed use case consisted of a BGP border router, which acts as iBGP to SDN-IP and eBGP speaker to the external legacy networks. The BGP policies advertised by external network border routers will be received by the ONOS/BGP speakers in the transit SDN and re-advertised to other external legacy networks by SDN-IP as an iBGP peer. SDN-IP translates the best route policies into the ONOS application intent request (AIR) and deploys them into data plane devices as forwarding rules to route the transit traffic into appropriate external domains. Internal BGP speakers were configured to use the TCP port at 179 to communicate with other BGP speakers and the TCP port at 2000 to send the route information to ONOS/SDN-IP instances, and the external BGP speakers were connected to OpenFlow enabled switches.

The BGP routers were emulated using the Linux hosts in which the Quagga routing suite was running. The BGP daemon of the Quagga routing suite was used for the BGP routers. Figure 2 shows SDN-IP implementation for an ISP network, where BGP speakers 10.0.1.1, 10.0.2.1, and 10.0.2.101 were connected to OpenFlow switches. BGP speaker 10.0.1.101 was connected to ONOS, that is, SDN-IP. It advertises route information to SDN-IP; using this, the SDN network acts as a transit AS. BGP speakers 10.0.1.1 and 10.0.2.1 were connected to the external AS. These speakers advertise routes to the AS and to other BGP speakers. iBGP speaker 10.0.1.101 listens to these routes and sends them to SDN-IP. SDN-IP changes the known BGP routes to intents and ONOS turns the intents to OpenFlow entries. The OpenFlow entries are then deployed into the OpenFlow switches using the OpenFlow protocol.

Using D-ITG, UDP flows were generated with varying packet size and packet rate for 15 s. Two log files were generated for both the sender and receiver side. The log file generated at the receiver side contained values for different performance indicators such

as delay, jitter, bit-rate, bytes received, packets dropped and average loss-burst size. On decoding the log file, we received a stream of data containing these performance indicators. The values of these parameters were taken from the log file on receiver's side. Multiple unidirectional flows were sent for fifteen seconds between pairs of hosts in turns by varying transmission rate and packet size to study the performance of the network under low and high load. The packet size set were 512 and 1024 bytes and the packet transmission rates were varied to 100, 1000, and 3000 packets per second. These flows were sent in the traffic via UDP where each host acted both as a client and a server. Ultimately, the log files of flows generated from ten hosts (ref. Figure 2) in the SDN network were further processed to a CSV file using a Python script. The D-ITG data samples generation snapshot is shown in Figure 5.

```

-----
Flow number: 1
From 10.0.0.4:34771
To   10.0.0.3:9501
-----
Total time           =          9.999002 s
Total packets        =           10000
Minimum delay        =          0.000000 s
Maximum delay        =          0.000000 s
Average delay        =          0.000000 s
Average jitter       =          0.000000 s
Delay standard deviation =          0.000000 s
Bytes received       =          7498028
Average bitrate      =          5999.021102 Kbit/s
Average packet rate  =          1000.099810 pkt/s
Packets dropped      =                   0 (0.00 %)
-----
***** TOTAL RESULTS *****
Number of flows      =                   1
Total time           =          9.999002 s
Total packets        =           10000
Minimum delay        =          0.000000 s
Maximum delay        =          0.000000 s
Average delay        =          0.000000 s
Average jitter       =          0.000000 s
Delay standard deviation =          0.000000 s
Bytes received       =          7498028
Average bitrate      =          5999.021102 Kbit/s
Average packet rate  =          1000.099810 pkt/s
Packets dropped      =                   0 (0.00 %)
Error lines          =                   0
-----

```

Figure 5. Snapshot of a D-ITG output.

Moreover, 507 data samples were captured during the simulation. Figure 6a–d shows the pattern and distribution interval of the data samples for delay, jitter, bit rate and the packet rate, respectively.

The bi-variate distribution of numerical attributes can be observed in the pair-plot shown in Figure 7. A pair-plot visualizes given data to understand the best set of features to explain a relationship between two variables (attributes) or to form the most separated clusters. The plot is in matrix format where the x -axis is represented by row name and the y -axis is represented by column name. The main-diagonal subplots are the univariate histograms (distributions) for each attribute. For example, the plot in the fourth row and the fourth column shows the univariate histogram for average-jitter, where the minimum and maximum of the parameter can be observed. On the non-diagonal subplots, bi-variate distribution between attributes can be observed. Some plots, such as those in the first column

and the first row, represent separate clusters, which show that no relationship exists between the attributes. Certain scatter plots that follow a clear linear pattern with an increasing or decreasing slope can be seen, which shows that some conclusions can be drawn from the distribution. Plots with a linear relationship between the two attributes make it possible to identify in which space the classes will be well separated from each other. For example, the plot in the second row and the fourth column shows a linear relationship between the two variables. Such linear patterns represent dependencies among the attributes, bringing us to the conclusion that average delay and average jitter are correlated.

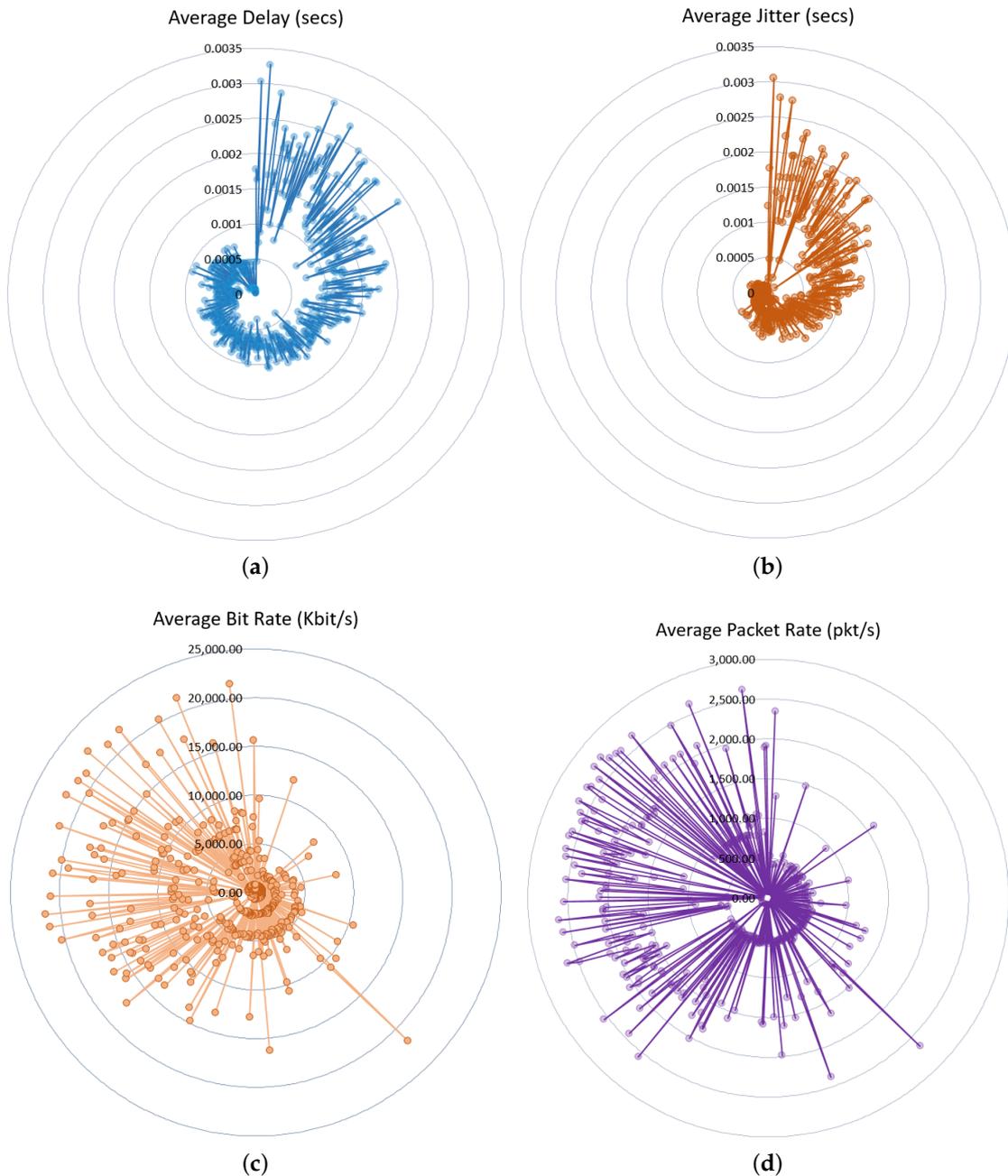


Figure 6. Distribution range of sample dataset (507 samples), (a) Data samples for average delay, (b) Data samples for average jitter, (c) Data samples for average bit rate, (d) Data samples for average packet rate.

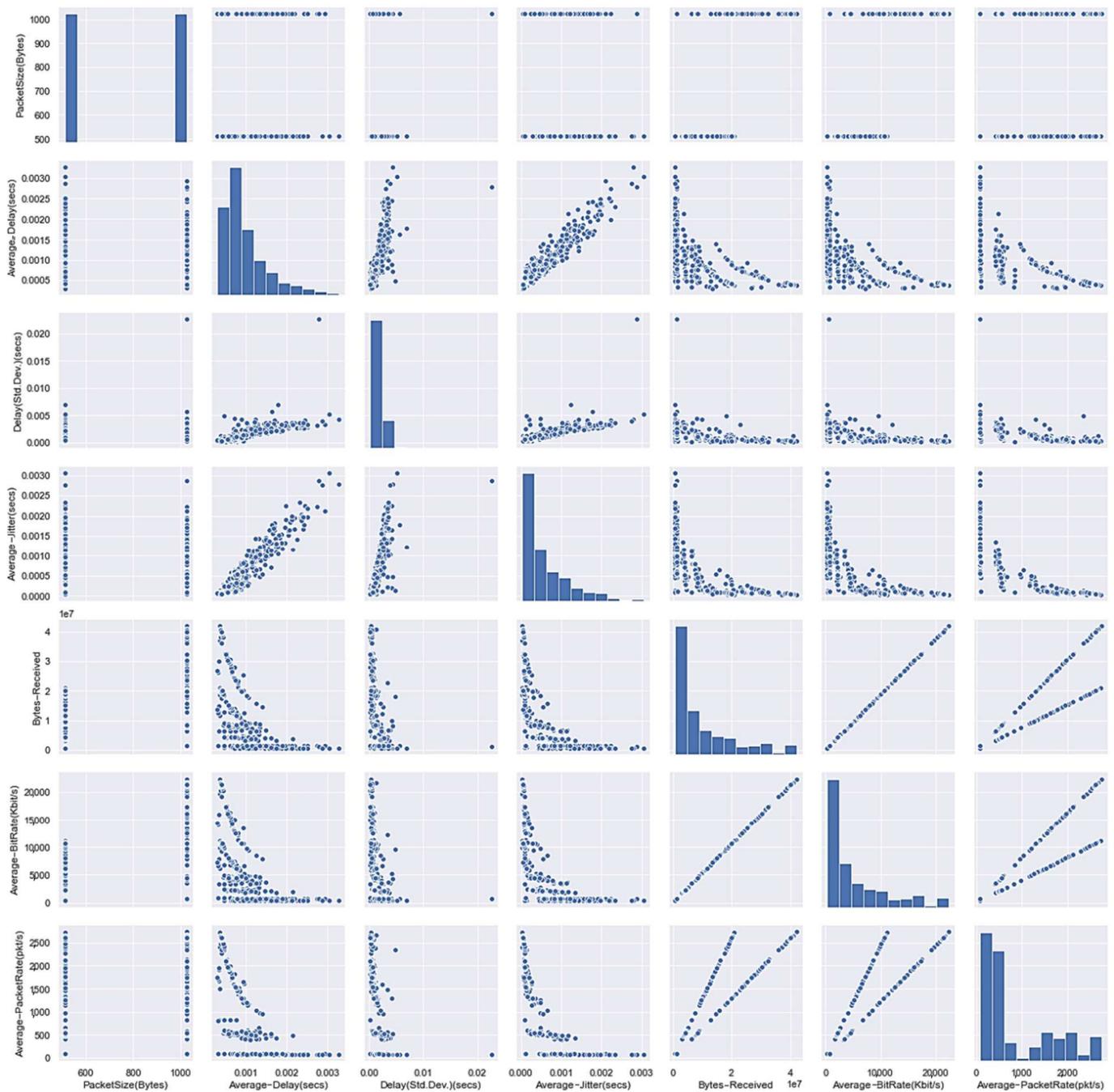


Figure 7. Pair-plot of dataset.

The pair-plot is an efficient tool for exploratory data visualization and analysis but does not show the strength of correlation between the attributes with dependencies. This can be achieved through a correlation heat-map using the heatmap() function available in the Seaborn package. The correlation of all the numerical attributes as a heat-map using the Spearman correlation coefficient is shown in Figure 8. The scale of color in the heat-map varies from blue to yellow, where blue signifies a correlation of +1 and yellow signifies a correlation of -1. The Spearman correlation coefficient measures the extent to which two variables tend to change together. The coefficient describes both the strength and the direction of the relationship. The Spearman’s rank-order correlation is the non-parametric version of the Pearson correlation. Spearman’s correlation coefficient (r_s) measures the strength and direction of association or the monotonic relationship between two ranked variables. It requires two variables that are either ordinal, interval, or ratio. For a sample size of n ,

the attributes X_i and Y_i are converted to ranks rg_x and rg_y . Spearman correlation coefficient (r_s) is computed by Equation (1),

$$r_s = cov(rg_x, rg_y) / \sigma_{rg_x} \cdot \sigma_{rg_y}, \tag{1}$$

where $cov(rg_x, rg_y)$ is the co-variance of the rank variables. σ_{rg_x} and σ_{rg_y} are standard deviations of the rank variables.

It can be observed from Figure 8 that average delay and average jitter are highly correlated with a Spearman coefficient of 0.92. Similarly, bytes received, average bit rate, and average packet rate are highly correlated as well. Parameters, for example, average delay and average packet rate, can be observed to be negatively correlated, with a Spearman coefficient of -0.75 .



Figure 8. Spearman correlation heat-map.

4. Analysis and Evaluations

This section aims to highlight the differences between the integrated SDN network and the legacy network by comparing them in terms of various performance indicators. Additionally, the performance of the integrated SDN network was further observed by taking note of the network quality of service (QoS) parameters defined.

4.1. Comparison between SDN and Legacy Network

In this section, we provide a comparative analysis of the use case network proposed as a hybrid SDN and legacy network in terms of the QoS parameters defined. The comparison is done by analysing the performance of both networks (legacy and SDN) on the basis of bandwidth capacity, packet transmission rate and time required to transmit the packet from the source node to the destination node.

4.1.1. Bandwidth

The results of bandwidth capacity in both the SDN and the legacy network were obtained by using *IPERF* tool in Mininet. Here, we show the maximum bandwidth between hosts in neighbouring ASs and within the same AS for both the SDN and the legacy network. In addition to that, the maximum bandwidth when the hosts use AS65000 or SDN as a transit network to connect is also shown. As shown in Figure 9, the maximum bandwidth seems to be higher in the case of the SDN network, which can highlight the ability of SDN to transfer more data per second compared to their legacy counterparts.

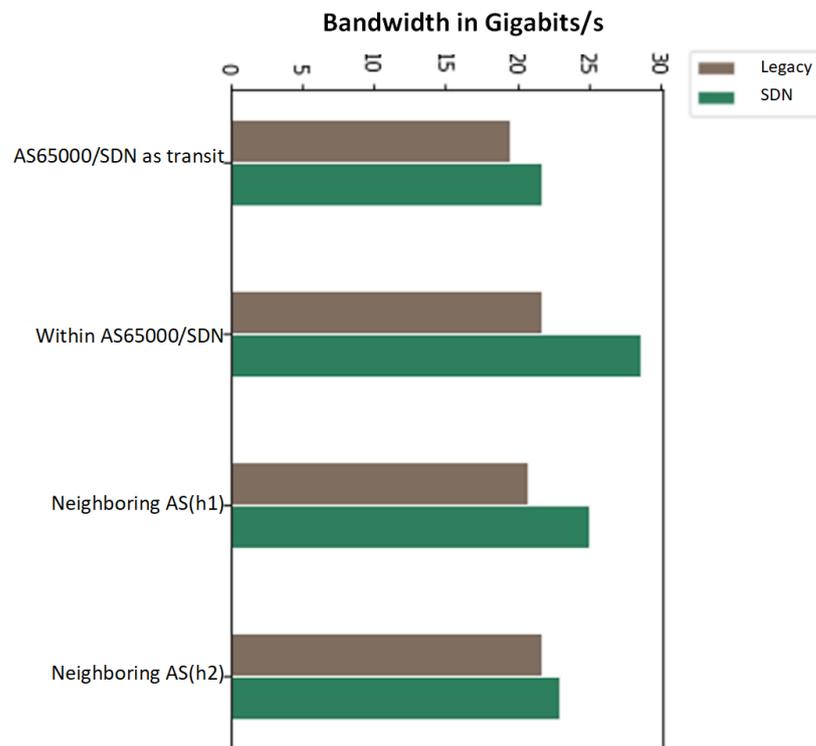


Figure 9. Bandwidth capacity for SDN and legacy network.

It can also be seen that the maximum bandwidth is higher for nodes within AS65000 or within the SDN compared to the neighbouring AS or when AS65000 and SDN are used as a transit network.

4.1.2. Packet Transmission Rate (PTR)

We compared both networks based on the PTR obtained by executing the *PING* tool in Mininet. PTR for both topologies is shown in Figure 10 for source and destination hosts that use AS65000 or the SDN network as a transit network for a varying number of packets.

The results obtained for the PTR indicate that the total time taken by both networks for a given number of packets is almost equal. It is noted that both networks are active for the same number of time intervals to execute the command.

4.1.3. Round Trip Time (RTT)

Next, we compare the delay between nodes in each network. The *PING* tool was used to obtain the RTT between the source and the destination nodes. The networks are compared on the basis of the variance of RTT, and the minimum and maximum values of RTT.

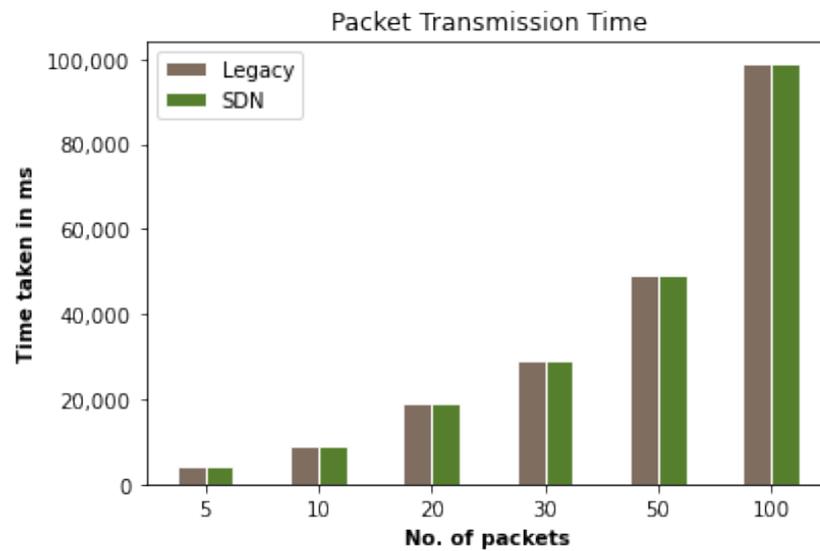


Figure 10. PTR for AS65000/SDN as a transit network.

Here, the standard deviation of the values of RTT for both the networks are plotted in a graph. This is done for the source and destination nodes within an AS, in a neighbouring AS, and when AS65000 and the SDN network are used as transit networks.

Standard deviation is simply the average of how far each RTT value is from the mean RTT. This shows how the round trip time varies with time. A higher value of variation in the RTT value results in a more unstable network. As shown in the plots of Figures 11a,b, 12a,b, 13a,b and 14a,b, the standard deviation values with a varying number of packets for SDN is lesser than that of the legacy network. This signifies that SDN can be more reliable and stable than their legacy counterparts.

The values used in the plots are used as point estimates to provide a rough estimate of the population parameter (i.e., mean RTT variance of the population) within a confidence interval. Two different samples for the legacy (ref. Figures 11a, 12a, and 13a,b) and SDN networks (ref. Figures 11b, 12b, and 14a,b) were taken, respectively. The T-distribution of the samples are determined at a confidence level of 95% as the standard deviation of the population is unknown to us. For the legacy network, a confidence interval of (0.02495, 0.06938) is obtained with a sample mean equal to 0.0472 and a *t*-critical value equal to 2.200985. Similarly, for the SDN network, a confidence interval of (0.0304, 0.0551) is obtained with a sample mean equal to 0.042767 and a *t*-critical value equal to 2.000995.

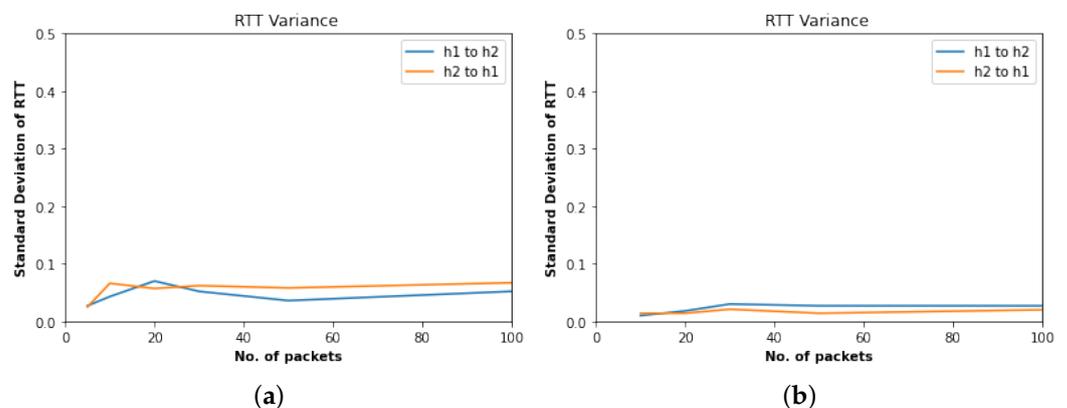


Figure 11. RTT Variance, (a) in Legacy (AS65000 as transit) and (b) AS65000/SDN as Transit in SDN.

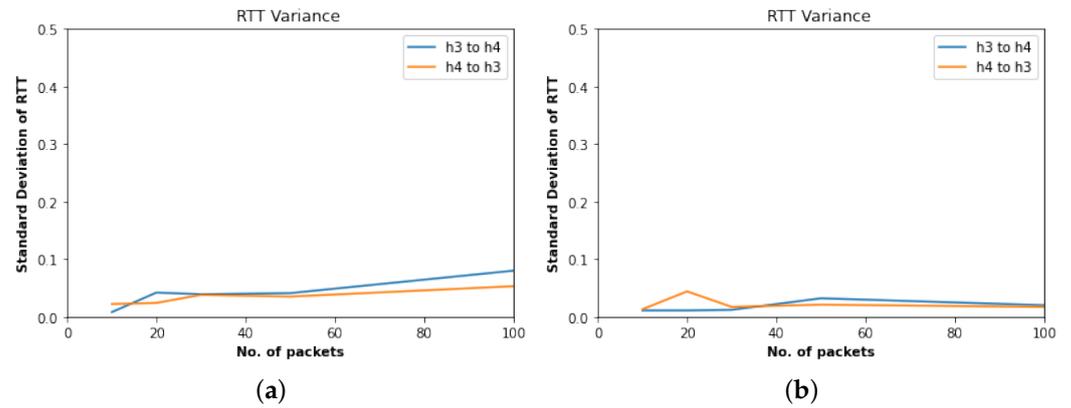


Figure 12. RTT Variance, (a) Within AS65000 in Legacy, and (b) within SDN. RTT, Round Trip Time.

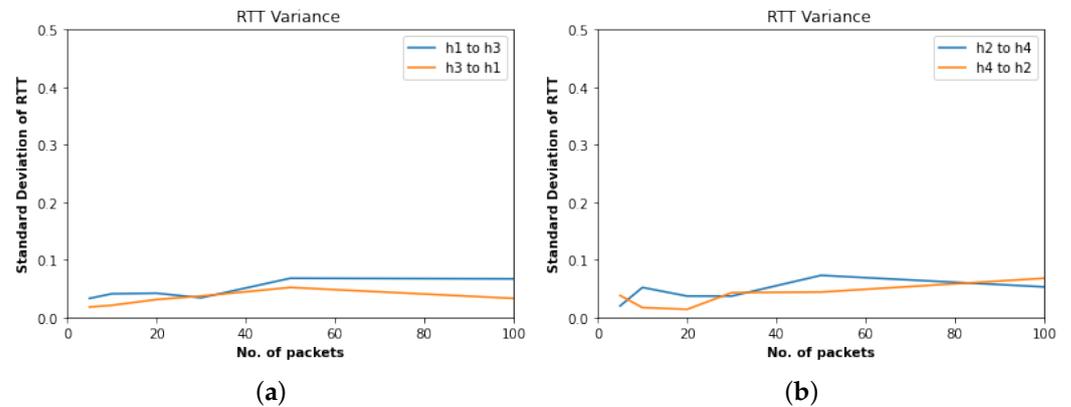


Figure 13. RTT Variance for Neighbouring AS of Legacy Network, (a) h1 <-> h3 and (b) h2 <-> h4.

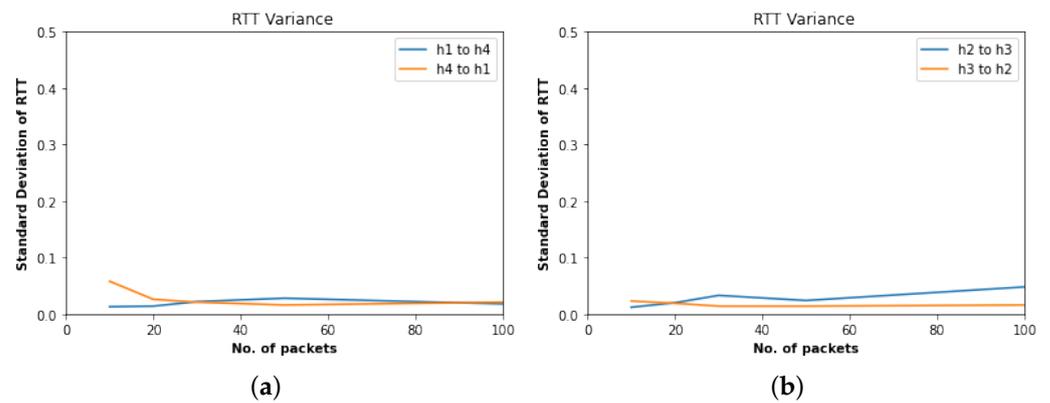


Figure 14. RTT variance for neighbouring AS of SDN, (a) h1 <-> h4 and (b) h2 <-> h3.

Some of the observations presented in the graphs have spikes at the beginning because of the fact that, in both the networks, the pinging host first needs to discover the target to be communicated, which takes additional time. Initial readings of the packets are avoided in the plots to show the actual deviation in RTT. From the observations, the variability in the standard deviation of RTT is small and smooth. We run multiple rounds of executions to see the pattern of variation in the standard deviation, while no significant pattern changes have been found. Additionally, the experiments were executed in virtual environments of higher memory and processing capacity host. The results of the additional round of tests are provided in our GitHub page mentioned in the supplementary section of this article.

Furthermore, we run the experiments in the virtual machine hosted on the azure and DNS servers of Google and Cloud-flare to demonstrate the presence of variability in RTT values over time and the results are plotted in Figure 15. This shows that the RTT variance of the PING results between machines in a real scenario. We observed the values of RTT with a deviation from 0.1 to 0.8. There are other communications and the protocol's internal signaling (e.g., OpenFlow communication, Host_Location_service_Provider etc.), which affect the RTT values.

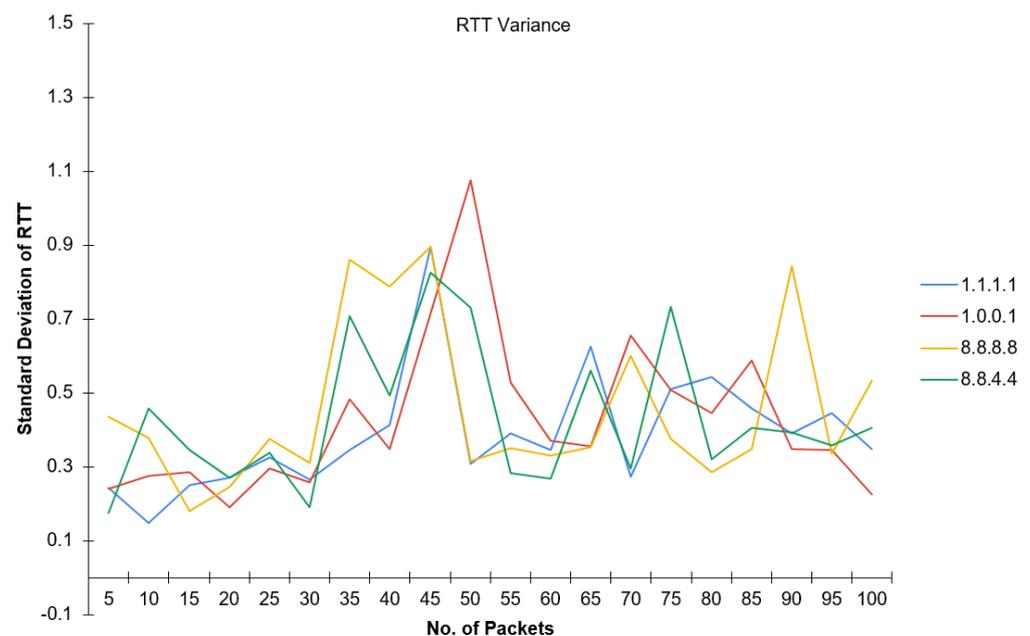


Figure 15. RTT Variance of PING results with remote DNS server.

The minimum and maximum values of RTT for both networks are plotted in the graph in Figures 16a–d, 17a–d, and 18a–d. This is done for source and destination nodes within an AS, in a neighbouring AS, and when AS65000 and the SDN network are used as transit networks. The traffic were captured at multiple rounds and more than one plot is provided in Figures 16–18 respectively.

From the plots of Figures 16a–d, 17a–d, 18a–d and 19a–d, it can be observed that, for a varying number of packets, the minimum and maximum values of RTT are lesser in SDN compared to that of the legacy network. A higher RTT value signifies more time taken for the transmission of packets that can affect the speed and reliability of the network. Thus, it can be concluded that routing in SDN using ONOS/SDN-IP performs much better than the legacy network in terms of speed and reliability.

4.1.4. Learning Time

Figures 20 and 21 show the routes learnt by the ONOS controller in SDN and our BGP router in legacy topology, respectively. It is noted that the controller took about ten to fifteen seconds to learn the complete routes, whereas the BGP routers in the legacy network took around one minute. This shows that the learning rate of SDN is faster than that of the legacy network. In the legacy network, every router needs to learn about the changes in the topology, whereas in SDN, only the controller needs to know about the changes and it will install corresponding flow entries in the forwarding devices if required. This improves network performance as no routing advertisement messages are advertised in the network.

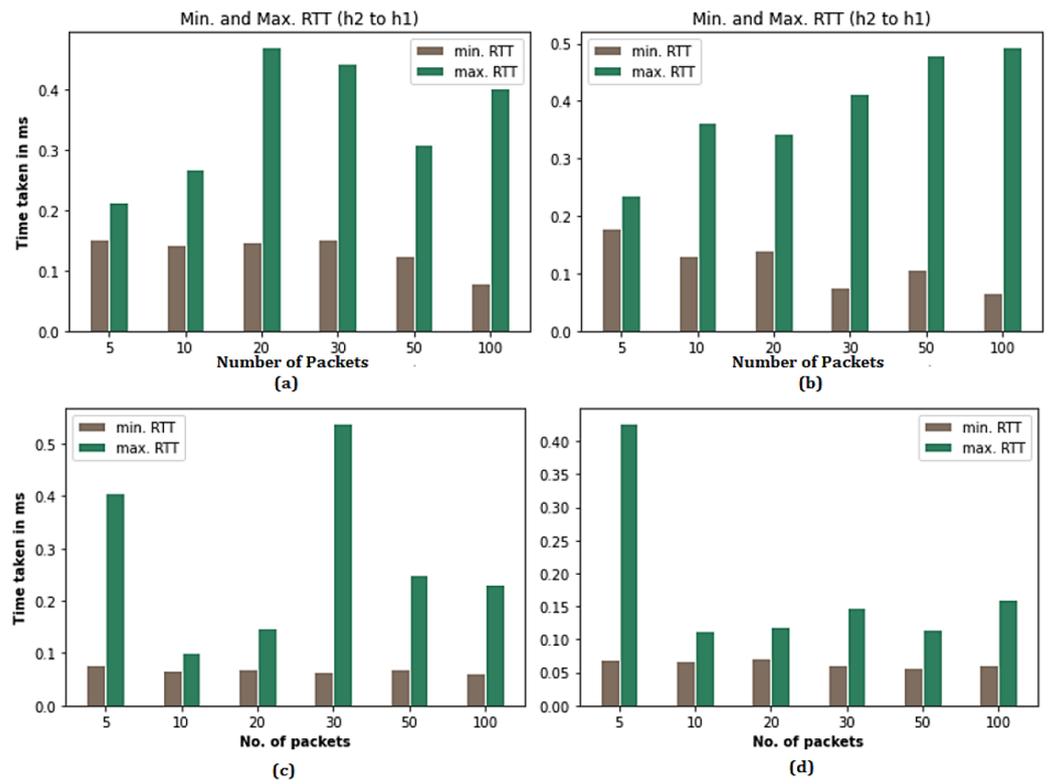


Figure 16. Min. and Max.RTT between h2 and h1, (a,b) AS65000/SDN as transit in legacy, (c,d) Min. and Max.RTT between h2 and h1 in SDN.

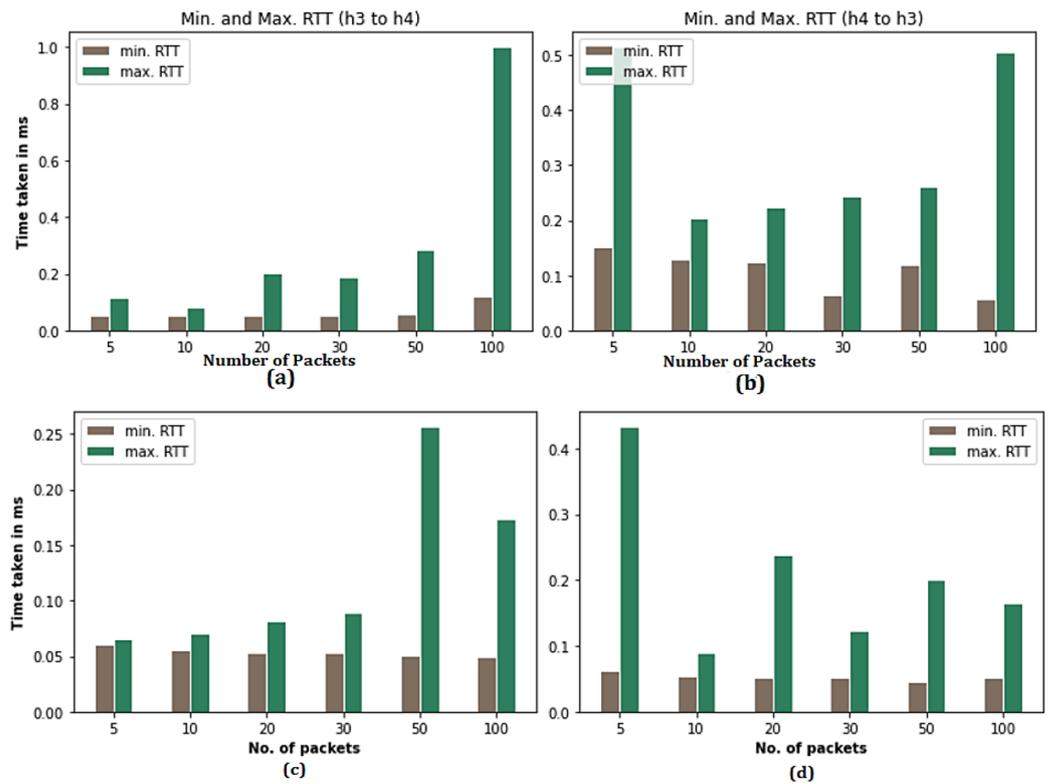


Figure 17. Min. and Max.RTT, (a,b) Within AS65000/SDN in Legacy, and (c,d) in SDN.

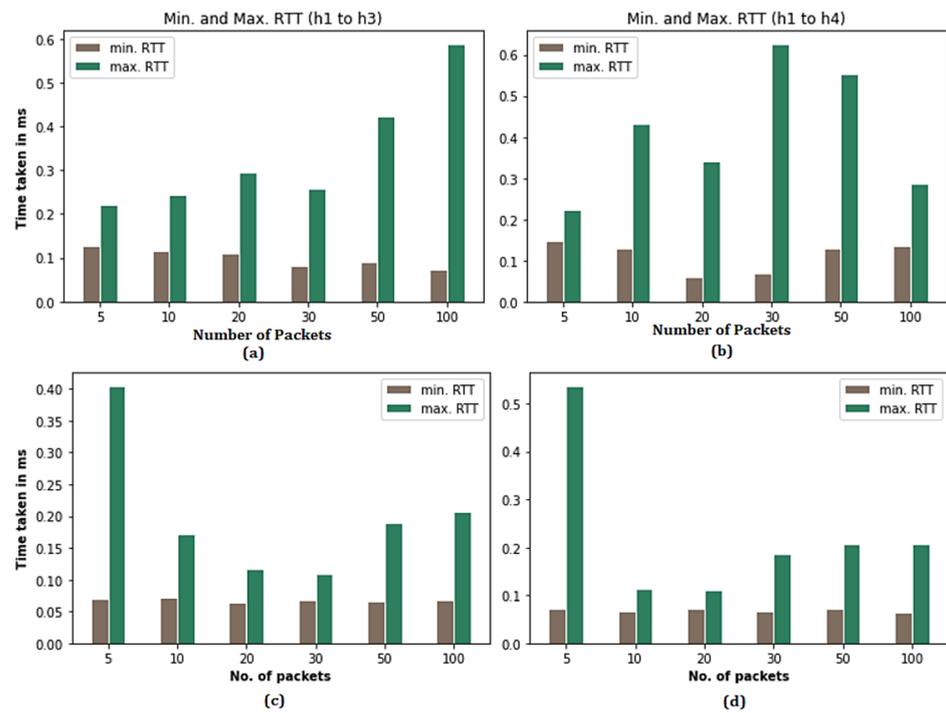


Figure 18. Min. and Max.RTT for neighbouring AS for Host h1, (a,b) in Legacy, and (c,d) in SDN.

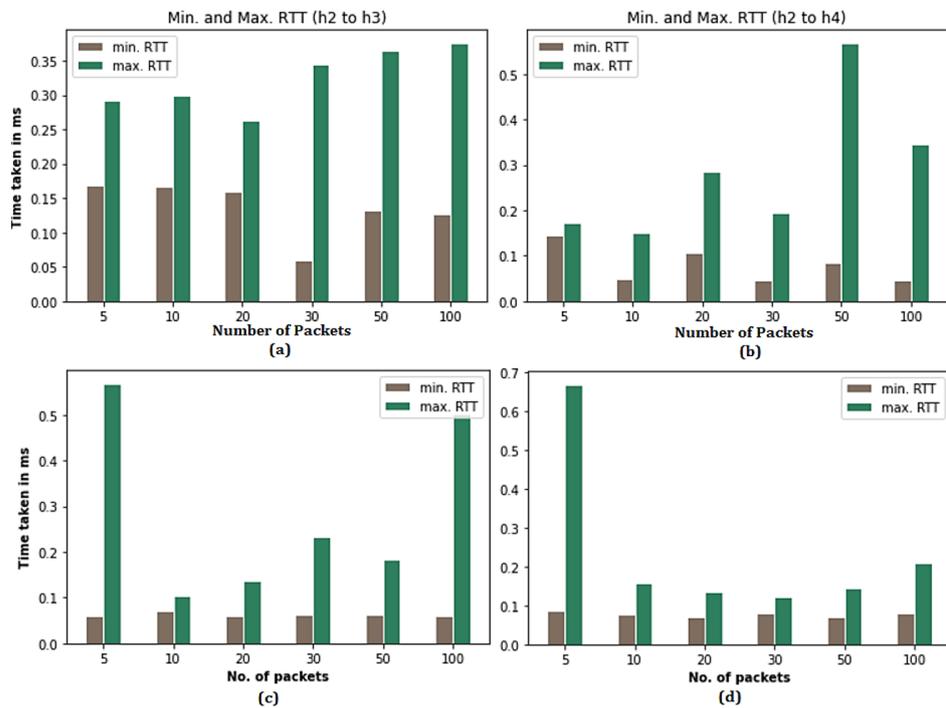


Figure 19. Min. and Max.RTT for neighbouring AS for host h2, (a,b) for Host h1 in legacy, and (c,d) in SDN.

```

onos> bgp-routes
Network          Next Hop          Origin LocalPref          MED BGP-ID
10.100.200.0/24  10.0.2.1          IGP      100                      0 10.10.10.1
                  AsPath 65002
192.168.3.0/24   10.10.10.1        IGP      100                      0 10.10.10.1
                  AsPath [none]
192.168.4.0/24   10.10.10.1        IGP      100                      0 10.10.10.1
                  AsPath [none]
10.100.100.0/24  10.0.1.1          IGP      100                      0 10.10.10.1
                  AsPath 65001
10.0.10.0/24     10.10.10.1        IGP      100                      0 10.10.10.1
                  AsPath [none]
Total BGP IPv4 routes = 5

Network          Next Hop          Origin LocalPref          MED BGP-ID
Total BGP IPv6 routes = 0
onos>

```

Figure 20. Routes Learnt by ONOS.

```

"Node: r5"
O>* 10.0.100.0/30 [110/20] via 10.0.100.14, r5-eth2, 00:18:41
O 10.0.100.4/30 [110/10] is directly connected, r5-eth1, 00:19:31
C>* 10.0.100.4/30 is directly connected, r5-eth1
O>* 10.0.100.8/30 [110/20] via 10.0.100.6, r5-eth1, 00:18:41
O 10.0.100.12/30 [110/10] is directly connected, r5-eth2, 00:19:31
C>* 10.0.100.12/30 is directly connected, r5-eth2
O 10.0.100.16/30 [110/10] is directly connected, r5-eth3, 00:19:31
C>* 10.0.100.16/30 is directly connected, r5-eth3
O 10.0.100.20/30 [110/10] is directly connected, r5-eth4, 00:19:31
C>* 10.0.100.20/30 is directly connected, r5-eth4
O 10.0.100.24/30 [110/10] is directly connected, r5-eth5, 00:19:31
C>* 10.0.100.24/30 is directly connected, r5-eth5
C>* 127.0.0.0/8 is directly connected, lo
B> 192.168.1.0/24 [200/0] via 10.0.100.1 (recursive), 00:18:31
*
   via 10.0.100.14, r5-eth2, 00:18:31
B> 192.168.2.0/24 [200/0] via 10.0.100.10 (recursive), 00:18:31
*
   via 10.0.100.6, r5-eth1, 00:18:31
B>* 192.168.3.0/24 [200/0] via 10.0.100.14, r5-eth2, 00:19:22
B>* 192.168.4.0/24 [200/0] via 10.0.100.6, r5-eth1, 00:19:22
C>* 192.168.5.0/24 is directly connected, r5-eth0
B>* 192.168.6.0/24 [200/0] via 10.0.100.26, r5-eth5, 00:19:24
B>* 192.168.7.0/24 [200/0] via 10.0.100.22, r5-eth4, 00:19:26
B>* 192.168.8.0/24 [200/0] via 10.0.100.18, r5-eth3, 00:19:24
zebra#

```

Figure 21. Routes learnt by reflector router (r5) in the legacy network.

5. Discussion and Future Work

Virtualization capability of SDN enables powerful APIs, which can be used to control many network functionalities with intelligence. The lack of backward compatibility of SDN with the legacy network enforces research communities to develop a robust system for interoperability between SDN and legacy networks for smooth and low cost migration approaches. Hybrid SDN is the only solution for having a smooth transition to pure SDN for service providers so that customers can receive uninterrupted services during the migration period. SDN-IP is an ONOS application that is used to peer SDN networks with external networks on the Internet using the standard BGP so that service providers can run hybrid SDN in their existing network during the transition period of their network migration. The SDN-IP controlled network acts as a transit AS that interconnects different legacy IP networks, considering each external network as a different AS domain, and interfaces with the SDN network through its BGP-speaking border routers.

In this study, an SDN integrated network was created with Mininet and experimental tests were carried out to ensure successful communication between the hosts in different ASs. The output of the tests showed a smooth transmission of data between hosts, thus providing testament to the possibilities of interoperability between two different networking paradigms. A similar legacy network was also created and observed to note the difference in the characteristics of an integrated SDN and legacy network.

To give statistical significance to this experiment, correlation coefficients of the QoS parameters were determined. These values can be further utilized with artificial intelligence

technologies or to develop machine learning models or rules about the network. These insights can be useful in wide areas of applications such as traffic classification, routing optimization, quality of service and quality of experience prediction, resource management, security enhancement and many other purposes.

The comparison between legacy and SDN integrated networks aimed at verifying the speculations about SDN performing better than legacy networks. By studying both the networks based on key performance indicators, such as bandwidth, packet transmission rate, and round trip time, the performance of the SDN network was observed to be much better than its legacy counterpart. Similarly, the exploratory data analysis of the values of the QoS parameters collected from the network provided us with insights about the network.

The scope of this experimental analysis was limited to an IPv4 addressing network only. A routing performance analysis with the networks enabled with IPv6 addresses could further help to establish the significance and prospects of SDN implementation with future networking technologies. Additionally, this study was limited to comparing the performance of hybrid SDN and a legacy network to determine the feasibility of the migration of the legacy networks into SDN. We read some of the literature related to SDN routing approaches [22,23], already mentioned in Section 2, since we are particularly dealing with routing in hybrid networks. Reference [23] deals with dynamic routing algorithm implementation in a pure SDN environment, while the article OSHI [22] considers hybrid switch networks. However, our scenario of routing implementation does not particularly match with the concepts provided in [22,23]. OSHI [22] considers hybrid or dual stack switch, that is, legacy routing and OpenFlow implementation, in a single switch. The dual stack switch consumes a higher load as compared with a single stack switch because it has to handle dual packets routing/forwarding at once as well as a separate access control list needing to be maintained. Hence, we will consider the comparison of performance evaluation of routing in an SDN only environment using ONOS/SDN-IP with other SDN routing approaches for future work.

The next generation wireless networking viz. 5G/6G network is considered to be fully SDN based [39,40]. The paradigm shifts in mobile communications by the conceptualization and implementation of 5G/6G added a strong requirement for SDN in the modern network environment. The wired/wireless network and server virtualization, high speed communications at highly dense smart devices with ultra low latency for real time mission critical applications, energy efficient smart network deployment with efficient radio access network (RAN) design, smart spectrum management and implementation are the part of 5G and beyond networks that directly concern people's modern living standards. Hence, from core network to end-access network service provisioning, cloud computing to fog computing to edge computing—they are mostly related to the availability of 5G and SDN, and the 4G to 5G migration and the performance evaluation of SDN based 5G network as compared with 4G networks are considered to be future work. Though network congestion optimization is not within the scope of this study, queue length analysis of the network device provides the trade-off between cost and performance of the network to understand the impact of congestion in SDN. Queue length analysis is also considered as future work.

Routing implementations over hybrid SDN with multiple controllers and their placement, a large and varying number of data plane devices, and so forth, could be the subject of future work to reflect the situations of CG-ISP networks when implemented into reality. A single instance of an ONOS controller is used in the present study. Multiple instances can be run at the same time for high available services. Multiple instances help in the load balancing of the whole network and add reliability to the network. Additionally, other available network controllers can be used to replicate the results for comparison in addition to ONOS.

6. Conclusions

The implementation of SDN is regarded as the best solution to meet modern communication requirements and to avoid the issues of legacy networks. The data plane devices that support the SDN network generally use the OpenFlow protocol to establish communication with the controller. However, the legacy devices generally already in operation these days do not support the OpenFlow protocol. The replacement of the existing networking devices at once by the OpenFlow supporting devices is not feasible and hence, a phase-wise migration is the only tangible solution. The features of SDN are compelling enough to encourage the migration of larger networks of ISPs from legacy to SDN. This study has demonstrated the successful integration of SDN networks with legacy networks to show the possibility of smooth migration and interoperability with legacy networks during migration. Multi-domain routing using ONOS/SDN-IP in the SDN environment has a better performance as compared with the legacy routing. The experimental analysis presented contributes to verifying the seamless interoperability of legacy and SDN networks so that service providers can be confident in the SDN implementation in their ISP and Telcos networks. This study is also a testament to SDN-integrated networks performing better than traditional legacy networks based on QOS parameters viz. bandwidth, PTR, and RTT.

Supplementary Materials: The program code in Python and the dataset of this research work are available at this GitHub page: <https://github.com/Abhishekthapa/SDN-Comparison-with-Legacy>, accessed on 11 April 2021.

Author Contributions: Conceptualization, B.R.D.; Methodology, B.R.D.; Software, A.T., R.G., D.K., and S.P.U.; Validation, B.R.D., and S.R.J.; Formal Analysis, B.R.D., A.T., R.G., D.K., and S.P.U.; Investigation, B.R.D., S.R.J.; Resources, B.R.D., S.R.J.; Data Curation, B.R.D., A.T., R.G., D.K., and S.P.U.; Writing—Original Draft Preparation, B.R.D., A.T., R.G., D.K., and S.P.U.; Writing—Review and Editing, B.R.D.; Visualization, A.T., R.G., D.K., and S.P.U.; Supervision, B.R.D., and S.R.J.; Project Administration, B.R.D. and S.R.J.; Funding Acquisition, B.R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University Grant Commission, Nepal (Grant-ID: FRG-074/75-Engg-01). Additionally, partial funding support was provided by NTNU-MSESSD-PhD program under the financial support from EnPe (NORAD), Norway.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are thankful to anonymous reviewers for their constructive comments to refine our article to meet the standards.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76. [[CrossRef](#)]
2. Dawadi, B.R.; Rawat, D.B.; Joshi, S.R. Software Defined IPv6 Network: A New Paradigm for Future Networking. *J. Inst. Eng.* **2019**, *15*, 1–13. [[CrossRef](#)]
3. Dawadi, B.R.; Rawat, D.B.; Joshi, S.R.; Manzoni, P.; Keitsch, M.M. Migration cost optimization for service provider legacy network migration to software-defined IPv6 network. *Int. J. Netw. Manag.* **2020**, *31*, e2145. [[CrossRef](#)]
4. Amin, R.; Reisslein, M.; Shah, N. Hybrid SDN networks: A survey of existing approaches. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3259–3306. [[CrossRef](#)]
5. Huang, X.; Cheng, S.; Cao, K.; Cong, P.; Wei, T.; Hu, S. A survey of deployment solutions and optimization strategies for hybrid SDN networks. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1483–1507. [[CrossRef](#)]
6. Sanvito, D.; Moro, D.; Gulli, M.; Filippini, I.; Capone, A.; Campanella, A. Enabling external routing logic in ONOS with Intent Monitor and Reroute service. In Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 25–29 June 2018; pp. 332–334.

7. Dawadi, B.R.; Rawat, D.B.; Joshi, S.R.; Manzoni, P. Legacy Network Integration with SDN-IP Implementation towards a Multi-Domain SoDIP6 Network Environment. *Electronics* **2020**, *9*, 1454. [CrossRef]
8. Mostafaei, H.; Lospoto, G.; Di Lallo, R.; Rimondini, M.; Di Battista, G. A framework for multi-provider virtual private networks in software-defined federated networks. *Int. J. Netw. Manag.* **2020**, *30*, e2116. [CrossRef]
9. ONOS-Project. ONOS Home Page. Available online: <https://opennetworking.org/onos/> (accessed on 22 April 2020).
10. Dawadi, B.R.; Rawat, D.B.; Joshi, S.R.; Manzoni, P. Evolutionary Gaming Approach for Decision Making of Tier-3 ISP Networks Migration to SoDIP6 Networks. *Int. J. Commun. Syst.* **2020**. [CrossRef]
11. Dawadi, B.; Rawat, D.; Joshi, S.; Keitsch, M. Joint cost estimation approach for service provider legacy network migration to unified software defined IPv6 network. In Proceedings of the IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Philadelphia, PA, USA, 18–20 October 2018. [CrossRef]
12. Dawadi, B.; Rawat, D.; Joshi, S. *Evolutionary Dynamics of Service Provider Legacy Network Migration to Software Defined IPv6 Network*; Boonyopakorn, P., Meesad, P., Sodsee, S., Unger, H., Eds.; Springer: Cham, Switzerland, 2020; Volume 936. [CrossRef]
13. Friyanto, A. High Availability Aspects of SDN-IP Reactive Routing. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bangdung, Indonesia, 2020; Volume 879, p. 012070.
14. Sheu, J.P.; Zeng, Q.X.; Jagadeesha, R.; Chang, Y.C. Efficient unicast routing algorithms in Software-Defined Networking. In Proceedings of the 2016 European Conference on Networks and Communications (EuCNC), Athens, Greece, 27–30 June 2016; pp. 377–381.
15. Guck, J.W.; Van Bemten, A.; Reisslein, M.; Kellerer, W. Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 388–415. [CrossRef]
16. Huin, N.; Rifai, M.; Giroire, F.; Pacheco, D.L.; Urvoy-Keller, G.; Moulrierac, J. Bringing energy aware routing closer to reality with SDN hybrid networks. *IEEE Trans. Green Commun. Netw.* **2018**, *2*, 1128–1139. [CrossRef]
17. Gilani, S.S.A.; Qayyum, A.; Rais, R.N.B.; Bano, M. SDNMesh: An SDN based routing architecture for wireless mesh networks. *IEEE Access* **2020**, *8*, 136769–136781. [CrossRef]
18. Csikor, L.; Szalay, M.; Rétvári, G.; Pongrácz, G.; Pezaros, D.P.; Toka, L. Transition to SDN is HARMLESS: Hybrid Architecture for Migrating Legacy Ethernet Switches to SDN. *IEEE/ACM Trans. Netw.* **2020**, *28*, 275–288. [CrossRef]
19. Levin, D.; Canini, M.; Schmid, S.; Schaffert, F.; Feldmann, A. Panopticon: Reaping the Benefits of Incremental SDN Deployment in Enterprise Networks. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*; USENIX Association: Philadelphia, PA, USA, 2014; pp. 333–345.
20. Vidal, A.; Verdi, F.; Fernandes, E.L.; Rothenberg, C.E.; Salvador, M.R. Building upon RouteFlow: A SDN development experience. In *XXXI Simposio Brasileiro de Redes de Computadores*; SBRC: Bracilia, Brazil, 2013; Volume 2013.
21. Vissicchio, S.; Tilmans, O.; Vanbever, L.; Rexford, J. Central control over distributed routing. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, London, UK, 17–21 August 2015; Volume 45, pp. 43–56. [CrossRef]
22. Salsano, S.; Ventre, P.L.; Lombardo, F.; Siracusano, G.; Gerola, M.; Salvadori, E.; Santuari, M.; Campanella, M.; Prete, L. Hybrid IP/SDN networking: Open implementation and experiment management tools. *IEEE Trans. Netw. Service Manag.* **2015**, *13*, 138–153. [CrossRef]
23. Tomovic, S.; Lekic, N.; Radusinovic, I.; Gardasevic, G. A new approach to dynamic routing in SDN networks. In Proceedings of the 2016 18th Mediterranean Electrotechnical Conference (MELECON), Lemesos, Cyprus, 18–20 April 2016; pp. 1–6.
24. Ventre, P.L.; Salsano, S.; Gerola, M.; Salvadori, E.; Usman, M.; Buscaglione, S.; Prete, L.; Hart, J.; Snow, W. SDN-Based IP and Layer 2 Services with an Open Networking Operating System in the GEANT Service Provider Network. *IEEE Commun. Mag.* **2017**, *55*, 71–79. [CrossRef]
25. Lee, H.L.; Liu, T.L.; Chen, M. Deploying SDN-IP over Transnational Network Testbed. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW), Yilan, Taiwan, 20–22 May 2019; pp. 1–2. [CrossRef]
26. Galiza, H.; Schwarz, M.; Bezerra, J.; Ibarra, J. Moving an IP Network to SDN: A Global Use Case Deployment Experience at Amlight. Available online: <https://amlight.net/wp-content/uploads/2015/03/wpeif2016-onos.pdf> (accessed on 11 April 2021)
27. Mamushiane, L.; Lysko, A.; Dlamini, S. A comparative evaluation of the performance of popular SDN controllers. In Proceedings of the 2018 Wireless Days (WD), Dubai, United Arab Emirates, 3–5 April 2018; pp. 54–59.
28. Kong, X.; Wang, Z.; Shi, X.; Yin, X.; Li, D. Performance evaluation of software-defined networking with real-life isp traffic. In Proceedings of the 2013 IEEE Symposium on Computers and Communications (ISCC), Split, Croatia, 7–10 July 2013; pp. 000541–000547.
29. Shirmarz, A.; Ghaffari, A. Performance issues and solutions in SDN-based data center: A survey. *J. Supercomput.* **2020**, *76*, 7545–7593. [CrossRef]
30. Rotsos, C.; Antichi, G.; Bruyere, M.; Owezarski, P.; Moore, A.W. OFLOPS-Turbo: Testing the next-generation OpenFlow switch. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5571–5576.
31. Yazdinejad, A.; Bohlooli, A.; Jamshidi, K. Performance improvement and hardware implementation of open flow switch using FPGA. In Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 28 February–1 March 2019; pp. 515–520.
32. Li, T.; Chen, J.; Fu, H. Application scenarios based on SDN: An overview. *J. Phys. Conf. Ser.* **2019**, *1187*, 052067. [CrossRef]
33. Dai, B.; Xu, G.; Huang, B.; Qin, P.; Xu, Y. Enabling network innovation in data center networks with software defined networking: A survey. *J. Netw. Comput. Appl.* **2017**, *94*, 33–49. [CrossRef]

34. Khorsandroo, S.; Sanchez, A.G.; Tosun, A.S.; Rodríguez, J.M.A.; Doriguzzi-Corin, R. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. *Comput. Netw.* **2021**, *192*, 107981. [CrossRef]
35. ONOS SDN-IP Description. Available online: <https://wiki.onosproject.org/display/ONOS/SDN-IP+Architecture> (accessed on 11 April 2021).
36. QUAGGA ROUTING SUITE. Available online: <https://www.quagga.net/> (accessed on 11 April 2021).
37. ROUTE REFLECTOR Description. Available online: <https://networklessons.com/bgp/bgp-route-reflector> (accessed on 11 April 2021).
38. Avallone, S.; Guadagno, S.; Emma, D.; Pescapè, A.; Ventre, G. D-ITG distributed internet traffic generator. In Proceedings of the First International Conference on the Quantitative Evaluation of Systems, Enschede, The Netherlands, 27–30 September 2004; pp. 316–317.
39. Zakeri, A.; Gholipour, N.; Tajallifar, M.; Ebrahimi, S.; Javan, M.R.; Mokari, N.; Sharafat, A.R. E2e migration strategies towards 5g: Long-term migration plan and evolution roadmap. *arXiv* **2020**, arXiv:2002.08984.
40. Long, Q.; Chen, Y.; Zhang, H.; Lei, X. Software defined 5G and 6G networks: A survey. *Mob. Netw. Appl.* **2019**, 1–21. [CrossRef]