UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

COMPARATIVE ANALYSIS OF SOFTWARE DEFINED NETWORKS (SDN) AND CONVENTIONAL NETWORKS USING ROUTING PROTOCOLS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN TELECOMMUNICATIONS ENGINEERING

By

DEEPTHI GOPI
Norman, Oklahoma
2017

COMPARATIVE ANALYSIS OF SOFTWARE DEFINED NETWORKS (SDN) AND CONVENTIONAL NETWORKS USING ROUTING PROTOCOLS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

____
Dr. Samuel Cheng, Chair

____
Dr. Robert Huck

____
Dr. Ali Imran

Acknowledgements

I would like to express my deepest gratitude to my committee chair, Dr. Samuel Cheng for the unceasing supervision of my thesis work and research, for his patience, motivation, enthusiasm, and immense knowledge. He has always been a very supportive and encouraging advisor, taking time to sharing his views with me and always giving me the apt guidance and assistance in my research work. His invaluable guidance has helped me all the time steering me in the right direction of my research and in writing of this thesis. Thank you for giving the opportunity to enhance my learning under you! You have a unique and critical contribution to my thesis work and I'm really very grateful for that.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Robert Huck, whose in-depth understanding of the research topic has helped me to approach the research from numerous different angles, the TCOM Lab project in SDN completed under your supervision inspired me and ignited my interest to pursue research in SDN.

I would like to express my sincere thanks to Dr. Ali Imran, for his unconditional support and help which enabled me to apply myself thoroughly in successfully completing this thesis. Both of them have always given me encouragement and insightful comments on my work thus aiding me in creating this fruitful, prolific and valuable thesis.

My heartfelt thanks also goes to Dr. Pramode Verma, who gave me this wonderful opportunity to pursue my graduate degree here at OU, for always supporting and administrating me in my research. I will carry forward your valuable teachings and guidance throughout my life.

I would like to express my deep profound gratitude to Dr. Daisaku Ikeda, my mentor for always providing me consistent and unfailing support, guidance, inspiration, and encouragement to accomplish my goals and ambitions with firm determination, unshakable conviction and indomitable “Never Give Up” spirit. I will forever carry on your mission, teachings, and values in my heart. I’m proud to proclaim that I’m your disciple. Thank you so much for being my mentor and coming into my life, Sensei!

My deepest and heartfelt gratitude to my grandfather L. Pandurangaiah Chetty, Hon. Lt. who made my dream of pursuing Master’s degree come true, your precious blessings, encouragement, support, and sponsorship helped me to realize my dream goal. I’m greatly indebted to you for all you have done for me and will be forever grateful and appreciative for your presence in my life.

Last but not the least, I’m grateful to my parents, sister and other family members who have been the tenacious pillar of support and strength, motivating me throughout my Master’s degree program and my life in general. My family’s support was an extremely essential factor that shaped my personality during this work, their unconditional love and affection always motivated me and gave me the strength to pursue my goals to the very end. I would like to thank my entire SGI family for providing me with unfailing support and continuous encouragement throughout my life and years of education. I would like to thank my friends for their feedback, cooperation, and help during my research.

I confidently say that I owe everything I am today to my mentor, Dr. Daisaku Ikeda, to my family, to esteemed professors who have taught me over my years of study and to the SGI. I dedicate this thesis to all the people in my life who have directly and indirectly contributed for the successful completion and realization of this work.

Table of Contents

1. **List of Tables**

List of Figures

List of Graphs

Abstract

Conventional routing protocols such as RIP, OSPF, EIGRP and BGP have a very rigid and intricate system thus narrowing the adaptability of networks to the ever changing Internet, the emergence of Software Defined Networking (SDN) provides a solution for this problem. Due to the handiness of a centralized controller, SDN has provided an effective method in terms of routing computation and fine control over data packets. Due to the increase in unpredicted failures taking place the ability to predict/ know the approximate maximum time it takes for these networks to converge in order to avoid and/or minimize loss of packets/data during these failures has become crucial in today's world. This time that the routers in the network take to converge via the implemented routing protocol to resume communication or transfer of information again is called the routing convergence time.

In this thesis, the performance is evaluated by measuring the routing convergence time during link failure with respect to the topology scale of the networks to show that SDN routing/forwarding is better compared to conventional routing. Further the results indicate that the routing convergence time is less in SDN networks on comparison with conventional networks when the topology scale is increased, indicating that SDN networks converge faster during link/node failures in comparison with Conventional networks and that routing convergence time is greatly influenced with the changing topological size/increasing network size. I believe that this work can throw light upon many advantages in SDN with regards to faster convergence during failures in contrast to archaic conventional networks.

1. Introduction

The Internet has a very deep relationship in every nook and corner of our lives. The routing protocols play a very important role in TCP/IP communication, the architecture and topology of conventional networks is very rigid, inflexible as shown in Figure 1 and has not changed in the past decades even when there are huge leaps in network speed and topology size. Any change in network caused due to an external event often takes long lead times to adapt to such simple changes, during such changes it is of vital importance for these networks to be able to converge seamlessly without forklifting [1] the existing network. Adding up to such limitations, these networks are built on closed, proprietary routing protocols further augmenting to the rigidity and inflexibility, thus programming and configuring the routers and switches is an arduous task [2].

1.1 Background

The routing protocols have contributed in communication over the Internet architecture, the data packets sent from the source traverse through the Internet constituting routers, switches, etc. to reach their respective destination. Due to the proprietary routing protocols and topology of conventional networks, it is wide-open to lot of bottlenecks leading to performance degradation.

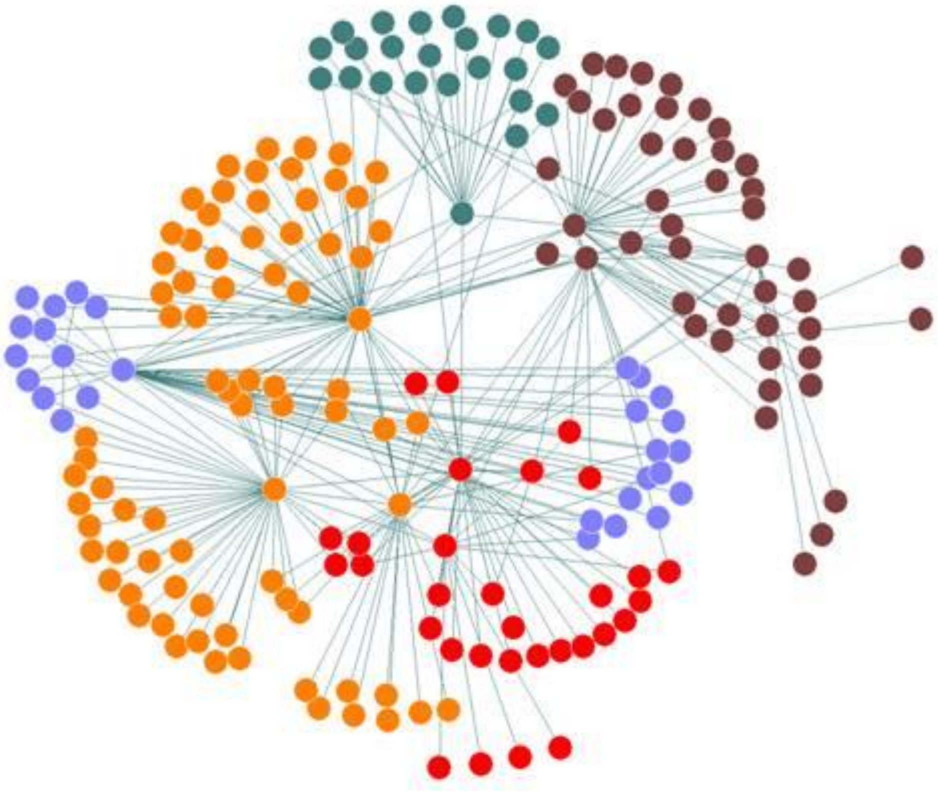


Figure 1. Rigid Conventional Architecture

With the growth in unforeseen failures and attacks, the ability of failure detection and recovery has become critical in today's world. Similarly, the need to transfer information from a source to a given destination during link failures or when changes in the topological information occur is also very crucial. Given a circumstance, it is important to be able to predict/ know the approximate maximum time it takes for a network to converge in order to avoid and/or minimize loss of packets/data [3], [4]. Routing convergence time is considered as one of the vital performance indicator and design goal for determining the performance of the routing protocol [5] and is of prime importance for networks, the faster the routers running the protocol help the network to converge during failure the more reliable it is to be used in real time applications [6]. Every routing protocol should be able to quickly adapt to topological changes and deliver data to the destination [7]. When a link/node failure occurs, the main job of these protocols is to quickly detect link failure and find an alternative route to reach the destination [4]. In conventional networks, every router possesses the entire topology information of the network in its routing table, it has to transmit/send the link change information to different parts of the network which affects the routing convergence time [3].

The emergence of SDN provides a hope in solving the above stated problem of conventional networks.

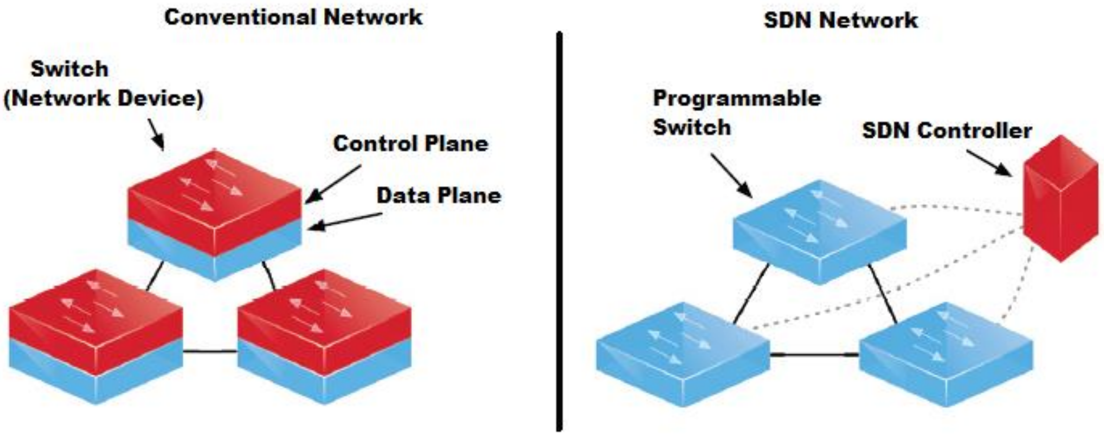


Figure 2. Conventional Network vs SDN Network Architecture [8]

From Figure 2, in SDN, the control plane (network plane) and the data plane (forwarding plane) are decoupled thus enabling direct provision of programming the network plane [9], [10]. Due to the presence of the controller in SDN networks, the controller transfers the control power of the data packets [3], [10] from the data plane switches to the central controller, the controller changes the topology information and performs routing thus stipulating faster convergence in comparison to conventional networks. The main purpose of SDN is moving towards centralized network architecture breaking free from the outdated rigid and inflexible architecture thus introducing agile and dexterous rerouting based on network conditions [11] to improve network performance and convergence mechanism.

1.2 Related Work

Many papers in past indicate research in comparative study of routing convergence time of different routing protocols for conventional networks and analyzing which routing protocol has the least routing convergence time/which routing protocol converges faster and how it will affect the performance of the networks. Reference [5] used OPNET simulation tool and real equipment to compare the convergence duration of routing protocols RIP, OSPF and EIGRP in conventional networks, and analyzed how it would affect the packet loss and quality of real time application. From this work, they drew conclusions that in both using simulation and real time the convergence for EIGRP is much faster compared to OSPF and RIP whereas RIP took the longest time to converge in both the scenarios. Similarly, in the paper referenced [12], it discussed the process of choosing the routing protocols (involves distance vector/link state or both) by capturing the traffic generated by each of the protocols and analyzing it, the conclusion drawn was convergence time of OSPF was faster than others.

Paper [13] indicates they developed a model which could achieve better network convergence based on the traffic variations thus improving network dependency and traffic performance. Along similar lines, from [14], they used SSFNet simulator to build conventional networks where they tried to investigate the relationship between BGP routing convergence time and the configuration of the Minimum Route Advertisement Interval (MRAI) timer for every simulated conventional network topology. Likewise, in [3] they studied ping response time w.r.t to varying packet forwarding delay using Open Shortest Path First (OSPF) protocol and OpenFlow protocol to study the behavior of routing convergence time performance [3].

1.3 Motivation

Up to now in Section 1.2, most of the papers which have been discussed previously have tried to study, assess and analyze the routing convergence time of dynamic routing protocols only in the conventional networks. Through this thesis, we believe that our study can throw light upon many advantages in SDN with regards to faster convergence during node and link failures in contrast to archaic conventional networks, thus the main contributions of this thesis is evaluating/comparing the performance of two different technologies namely SDN routing/forwarding using OpenFlow protocol and conventional routing using Border Gateway Protocol (BGP) with respect to routing convergence time and topological size. On comparison we study the convergence time behavior with different network sizes i.e. continuous increase in network topology size.

1.4 Problem statement

As described in Section 1.1, routing convergence time is considered as one of the vital performance indicator and design goal for determining the performance of the routing protocol [5] and is of prime importance for networks, the faster the routers running the protocol help the network to converge during failure the more dependable it is to be used in real time applications [6].

We know, from the reference [15] that the size of the network (topology size) plays a significant role in routing convergence time, a larger network will converge slower than a smaller one. Likewise, in [16], [17] where they compared convergence time for different network sizes w.r.t time function and the results showed a logarithmic relationship between topology size and routing convergence time in peer-to-peer (P2P) networks, there was a consistent change in the convergence time with continuous increase in network size. Thus, in this thesis, we study the performance of SDN routing using OpenFlow protocol and conventional routing using BGP protocol with respect to routing convergence time and increase in topological size.

1.5 Thesis Contribution

This thesis aims to show that the routing convergence time is improved in SDN as compared to conventional networks. The contributions of this thesis are listed as follows:

1. It discusses about SDN networks and its advantages over conventional networks.
2. It discusses the importance of convergence process and routing convergence time.
3. Provides an assessment between the BGP and SDN routing and convergence process analysis.
4. It evaluates the performance in terms of routing convergence time with increasing topology size.
5. A comparison between SDN and conventional networks is performed in terms of routing convergence time to reveal the superior performance of SDN networks.

1.6 Organization of the thesis

This thesis titled "***Comparative analysis of SDN and Conventional networks using routing protocols***" is arranged into six chapters. The chapters have been organized in the following manner.

In Chapter 1, discusses the brief history of conventional networks in terms of routing protocol mechanism, its drawbacks leading to acknowledgement of SDN networks.

In Chapter 2, explains the concept of convergence and the importance of routing convergence time that this thesis addresses along with details about the software tools used, experimental topologies implemented, the investigational experiment scheme, and the necessary settings.

In Chapter 3, introduces the BGP protocol in conventional networks and gives a detailed discussion about the concept of routing process in BGP. It explains and analyses the convergence process of BGP.

In Chapter 4, gives a comprehensive discussion about the concept of software defined networking. It elucidates the main highlights of SDN in terms routing mechanism using OpenFlow protocol.

In Chapter 5, illustrates graphs and discusses the results and of the experimental implementation.

Chapter 6, concludes the thesis with summarizable results and a discussion about the future work.

2. Convergence Process

Convergence is defined as the state in which the routers come to an agreement on the best paths for sending packets to the destination thus in turn completely updating their routing tables and possessing similar topological intelligence about the network in which they function [18], [19], [20], [21], [22]. For the routers operating on dynamic protocols in a network, convergence is an essential parameter to operate correctly [23]. Whenever a link or node failure occurs, basically any change in the topology of the network gives rise to convergence [18], [19], and [20].

During convergence every router will independently re-compute alternative paths and construct a new routing table based on the new information attained, once these tables have been updated with the changes, convergence is completed and the transfer of data packets resumes from the source to destination [20], [21]. The data attained by the routers must not conflict with any other router's routing table information, they must possess the correct topology information exchanged with each other [24], [25]. A network is said to be converged if the routers know how the network looks like, which links are up/down and which are the best routes to reach every destination [12].

Routers are intelligent devices which make their own routing decisions, this intelligence is a big boon as it allows the networks to be more efficient, faster, and robust without any human interference, but due to poor designing of networks it could lead to malicious attacks, threats, overloading and instability.

The concept of convergence helps in planning for network capacity, service capacity and criticality of infrastructure mainly in terms of network designing in order to avoid network overloading or suspicious attacks leading to instability and uncertainty

[21], [26]. Nowadays, telephone lines are connected to switches, if a switch goes down in the network then convergence is necessary for recovery/failover situations to transfer to alternate lines.

2.1 Routing Convergence Time

The time the routers take to come to an agreement with regards to the new topology after their routing tables are completely updated is called routing convergence time.

From [27], it describes routing convergence time as the sum of failure detection time, flooding of information time, processing the routing updates time, computation & installation paths time and rerouted (alternative) path time. In other words, elaborating reference [27], the basic definition of routing convergence time can be defined as the time taken by the network to re-establish its connectivity after a failure event occurs until the traffic is rerouted through an alternative path.

It depends on various parameters such as:

a) topology size i.e. the number of routers using the routing protocols within the network,

b) distance of routers (from point of link failure),

c) bandwidth and traffic load on the network links,

d) static/ dynamic routing protocol used [15], [21].

The routing convergence time is predominately affected by: link failure detection, link change propagation, wait time for comparing topology information, best path computation time, RIB (Routing Information Base), and FIB (Forwarding Information Base) update time [1] in a network.

Especially in conventional networks, the network stability and convergence time are mostly affected because of the speed at which the failures are identified and transmitted over the entire network depending on the number of routers present and the processing capabilities of all the routers in the network [1]. The size of the network is quarantined as the major contributor to network performance [1].

2.2 Software Tools and Specifications

For SDN networks, we use Floodlight v1.2 [28] controller as the main controller for the network in the first VM (Virtual Machine) using OpenFlow 1.1, Mininet 2.1.0 (a network simulator) [29], [30], [28] is used to create the network consisting of switches, routers and hosts [3], [29] as well as to measure the routing convergence time in the second VM. Using commands in Mininet the entire network (second VM) is linked to the Floodlight Controller (first VM) using OpenFlow protocol. The entire process and source code is explained in detail in the Appendices Section.

For conventional networks, we use Packet Tracer 6.1 (network simulation and visualization tool) for creating the network consisting of switches, routers and hosts enabling manual programming of routing protocols in the routers. In both networks, the real timer (ms), ping and traceroute commands are used for measuring the routing convergence time. The ping command is used to establish an interaction between the hosts in the network, the total amount of time taken by the source to send a packet to the destination and for the destination to send/echo a response back to the source about receiving it is called the response time [31].

The traceroute command prints out the complete path taken by a packet to reach a particular destination from the source, basically the path output here indicates the IP addresses of all the forwarding entities (like routers, switches in between) of the connection. This route related information that the command prints out can be very handy while debugging any network related issues [32].

2.3 Topologies Implemented

In this work, using Mininet network simulator the three topologies are created: 8 nodes, 16 nodes and 80 nodes with 2 hosts for conventional networks and SDN networks. The routers in the conventional network are manually configured with BGP protocol and assigned appropriate AS (Autonomous System) numbers. In SDN network, all the switches in the network are directly connected to the controller. The topology diagrams for conventional networks are shown below in Figure 3, 4, and 5:

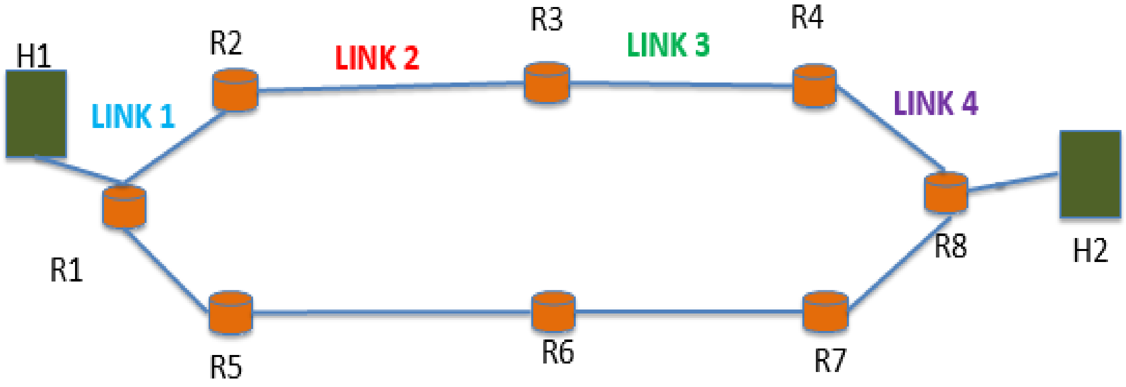


Figure 3. 8 Node Topology (Conventional)

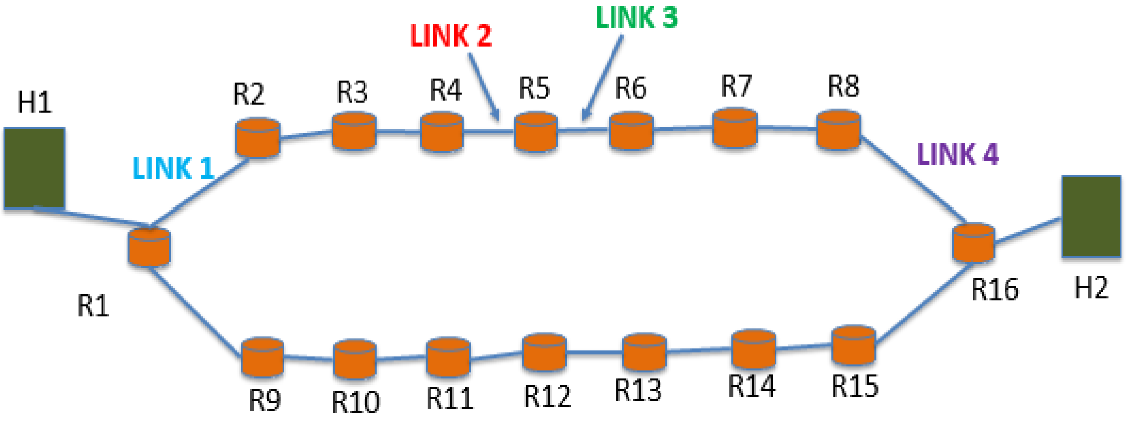


Figure 4. 16 Node Topology (Conventional)

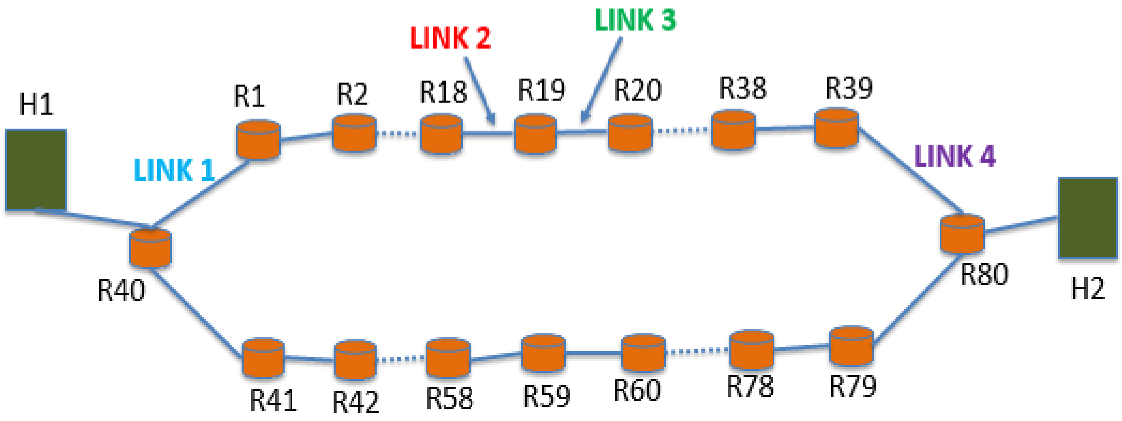


Figure 5. 80 Node Topology (Conventional)

The topology diagrams for SDN networks are shown below in Figure 6, 7, and 8:

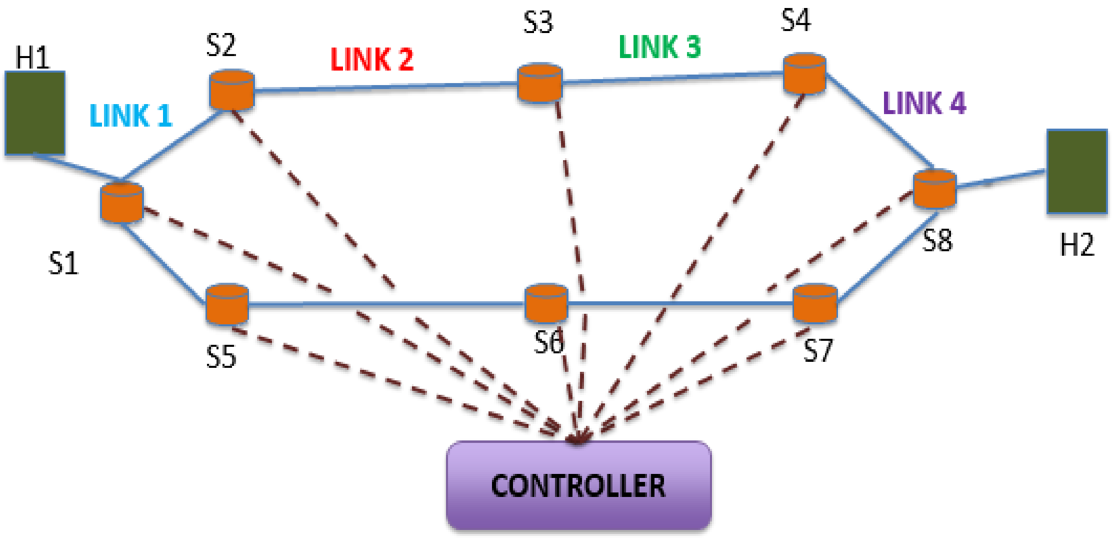


Figure 6. 8 Node Topology (SDN)

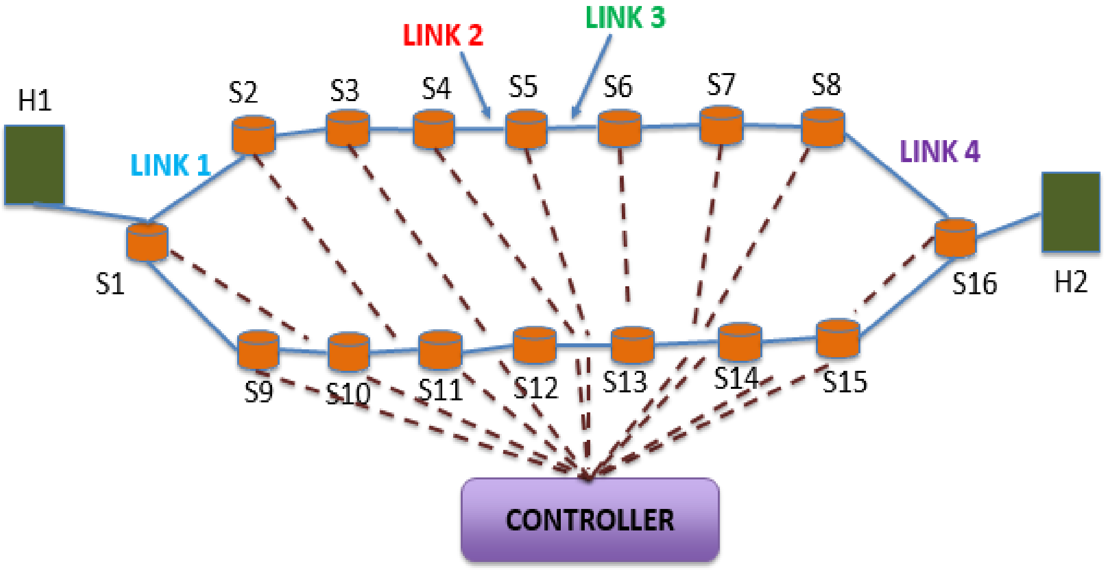


Figure 7. 16 Node Topology (SDN)

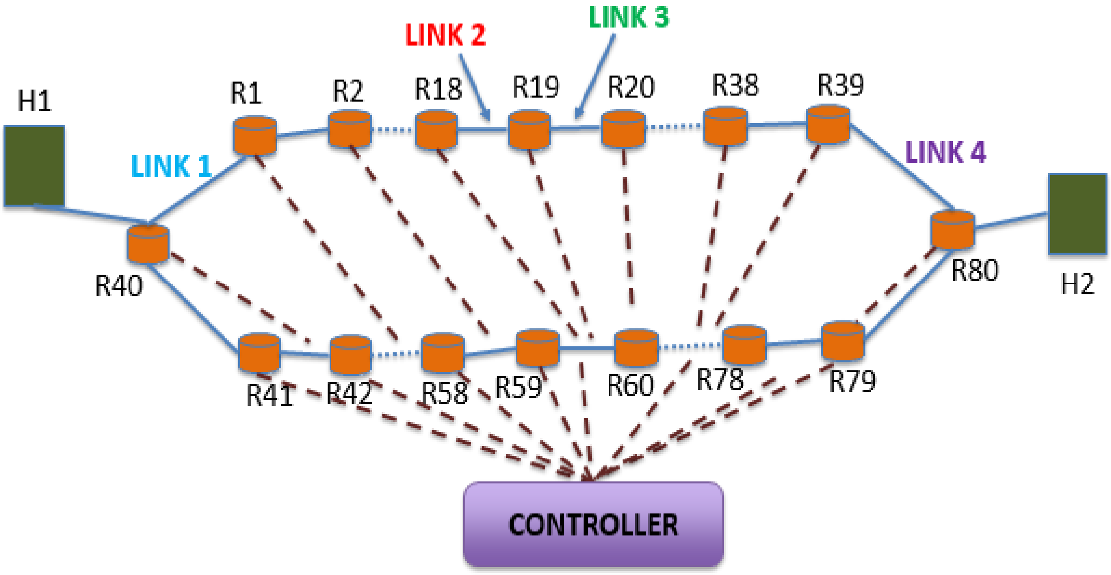


Figure 8. 80 Node Topology (SDN)

2.4 Experimental Scheme

In the simulation experiment, the bandwidth is set as 10 Mbps and link delay is 0 ms to calculate the routing convergence time. Table 1 and Figure 18 in Section 5 show the various values obtained for routing convergence time at different link positions for both conventional and SDN networks.

The experimental scheme for this work is explained below:

1. First a stable communication is established between host 1 (H1) and host 2 (H2) by firing a continuous ping test over the main path established, the main path is detected using traceroute command as shown in Figure 9. The main path remains the same for all the topologies indicated.
2. The **Link 1** as shown in Figure 10 is broken/disrupted down and wait until the connectivity for ping is restored via alternative path. The time from the disruption till again the connectivity is established, this time is the routing convergence time and it is recorded by observing the ping statistics.
3. After the disruption, since the alternative path is selected, communication is established via the alternative path as shown in Figure 11.
4. Then the **Link 1** is restored back up, the communication resumes via the main path.
5. The same process from step 1 to 4 is repeated 50 times and finally the average of all the readings is taken.
6. Likewise, the above process is repeated for links at different positions (**i.e. Link 2, 3 and 4**) for the given networks with different topology sizes as shown in Figure 3, 4, 5, 6, 7, and 8.

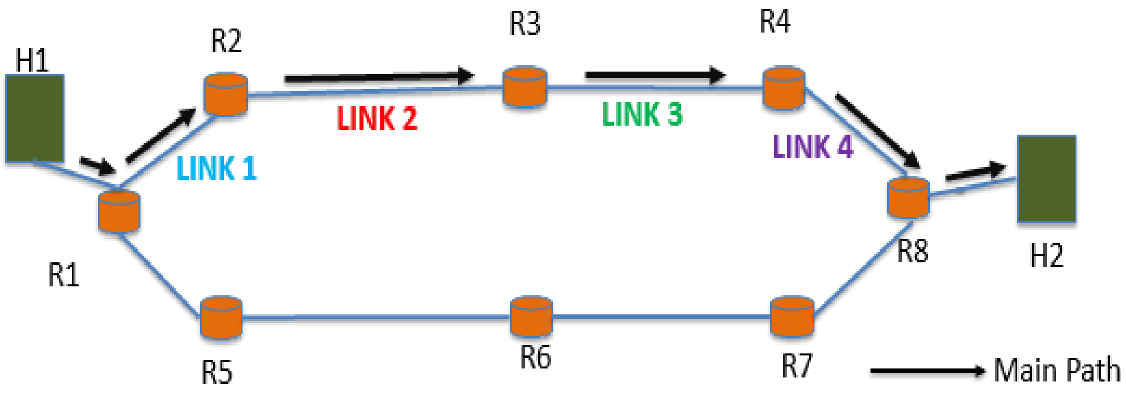


Figure 9. 8 Node Topology indicating the Main Path

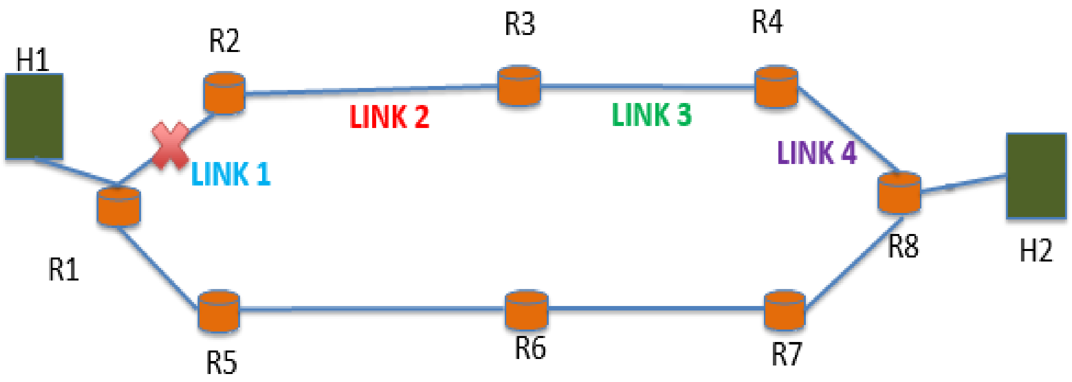


Figure 10. Link 1 is broken down

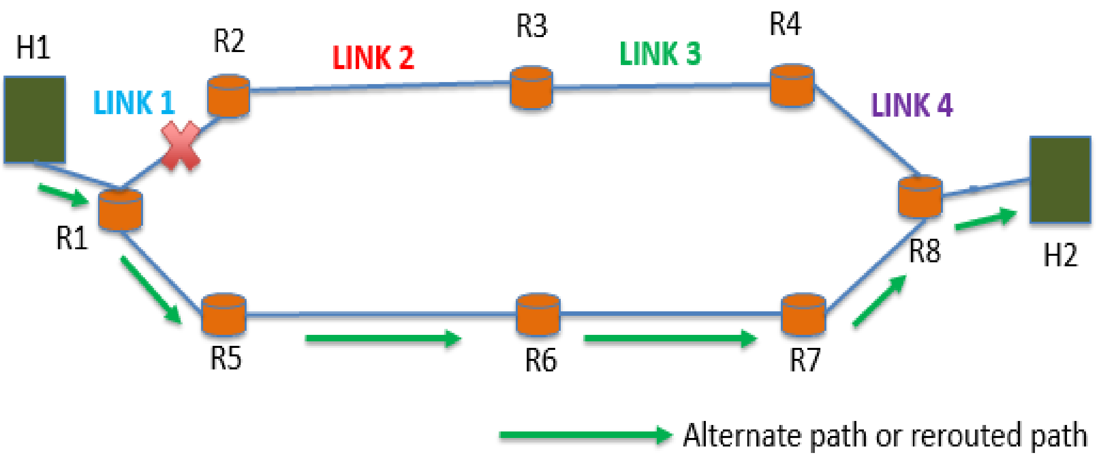


Figure 11. 8 Node Topology indicating the Alternate/ rerouted path

3. BGP Convergence Analysis

The Border Gateway Protocol (BGP) is globally used, since the early days of the Internet, it is a shortest path-vector protocol – in other words, a distance-vector protocol to route traffic between Autonomous Systems (AS) and domains, i.e., networks belonging to different administrative entities. BGP is a distributed protocol, over which ASes exchange routing information with their neighbors, and establish route paths.

An Autonomous System (AS) is a single entity/collection of entities whose prefixes and routing policies are under common administrative control (e.g. network service provider, a large company, a university, a division of a company, or a group of companies) [33].

A prefix is referred to as a route announcement [34]. A route contains prefixes (which is composed of IP addresses being broadcasted and also a path of AS numbers (ASN), indicating which ASes the packet must pass through in order to reach a respective destination [35].

In short, a prefix is a part of a BGP route, and will be exchanged between BGP neighbors in a BGP update message [35].

If an AS exchanges routing information with other ASes on the Internet, it needs to have an ASN, for exchanging routing information, particularly in identifying paths through multiple ASes via BGP, the Border Gateway Protocol [33].

3.1 Routing Process

The BGP routing process takes place as described below:

a) Once a session is established and BGP messages are exchanged, R1 enters the "BGP Read-Only Mode" [36], this indicates that R1 will not start the BGP Best-

Path selection process until it either receives all prefixes from R2. The reason to hold the BGP best-path selection process is to ensure that the peer has supplied us all routing information [36], [37], [38].

b) This implementation sends a KEEPALIVE [37] message once the updates are sent to the peer. If there are larger routing tables then the best-path selection process will cause a longer delay due to the exchange of tables.

c) Once R1 leaves the read-only mode, it compares the new information with its routing table (RIB) contents and starts the best-path selection, selecting the best-path for every prefix. This process takes time proportional to the amount of the new informational learned [36], [37], and [38].

d) Once the best-path is elected, BGP has to upload all routes to the RIB, before advertising them to the peers. Before propagating the best path information, it will upload the routing information in its RIB before advertising [36], [37], and [38].

e) Then RIB upload will in turn trigger forwarding table (FIB) information upload, both RIB and FIB updates are time-consuming [36], [37].

f) After information has been uploaded to FIB, R1 needs to send the best-path information to every peer that should receive it.

g) R1 starts sending updates to all the routers in the network and they in turn will upload the best path information in their RIB and FIB.

3.2 BGP Convergence Analysis

When a routing change occurs (e.g., a link is down, etc.), it will take some time for R1 to realize that the connection is no longer valid [39].

1) R1 will first remove the invalid routes from its routing and forwarding tables, then it sends BGP MESSAGE updates to its neighbors to inform them about the link down [39]. In BGP there are two types of update messages sent and they are:

a) Advertisement: This message informs neighboring routers of a new path added to the network for a destination. Transmits new route advertisement from the originating source to a destination [14].

b) Withdrawal: It is an update indicating that a previously advertised destination is no longer available [14].

Every BGP message contains new information pertaining to a set of path attributes. Therefore, if new information about a disrupted path needs to be transmitted then it requires a separate UPDATE message to be sent. In this work, more emphasis is given for interruption or withdrawal of the connection.

2) When the neighboring AS will receive these updates, it will calculate and change any needed updates (if any are there) for its routing table, it will in turn send updates (with withdrawal messages) to its own neighbors withdrawing the lost routes. Thus the BGP updates will propagate over the entire network in this way [36], [37], and [38].

3) The withdrawal updates are processed by the neighbors, they will choose the alternate best paths, if there are larger routing tables then the best-path selection process will cause a longer delay due to the exchange of tables.

4) The elected paths are added to their own routing and forwarding tables (FIB and RIB tables) [14], [36], [37], and [38], both RIB and FIB updates are time-consuming.

5) Then the neighbors will broadcast their new best paths.

6) The router processes incoming BGP updates, elects new best paths, and adds them in routing and forwarding tables and continues to propagate these updates to other ASes [36], [37], and [38]. Larger the topology size, longer it takes to propagate the updates to different parts of the network thus increasing the routing convergence time.

7) Once a new path is elected then the connection is established and transfer of information resumes [36], [37], and [38].

4. SDN Convergence Analysis

In SDN, the controller is the "brains" of the SDN network (control plane), the controller uses the OpenFlow protocol (Southbound API) to connect and configure the network devices (routers, switches, etc.) to determine the best path for transmitting information. The OpenFlow protocol defines the communication between the Floodlight controller and the switches via a set of messages which are sent to and from the controller to the switch [40]. The controller programs the switch via these messages thus providing fine-grained control over the data traffic as shown in Figure 12.

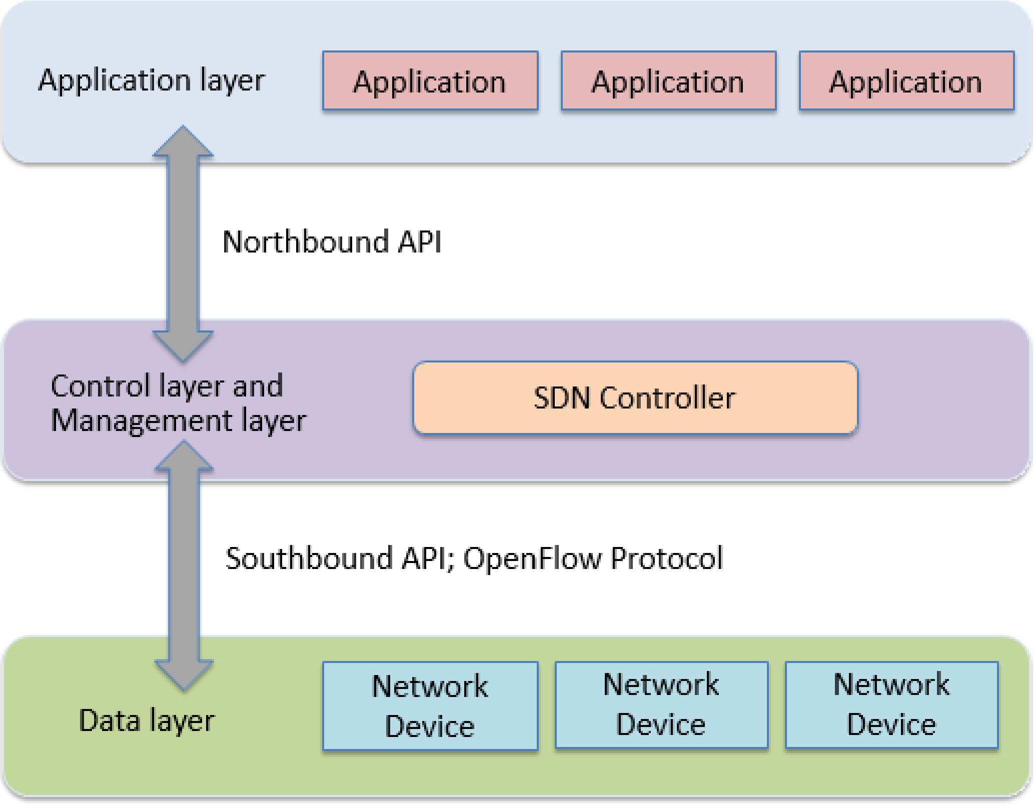
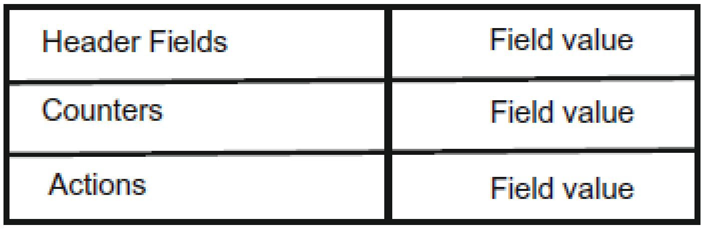


Figure 12. SDN Network Architecture [40]

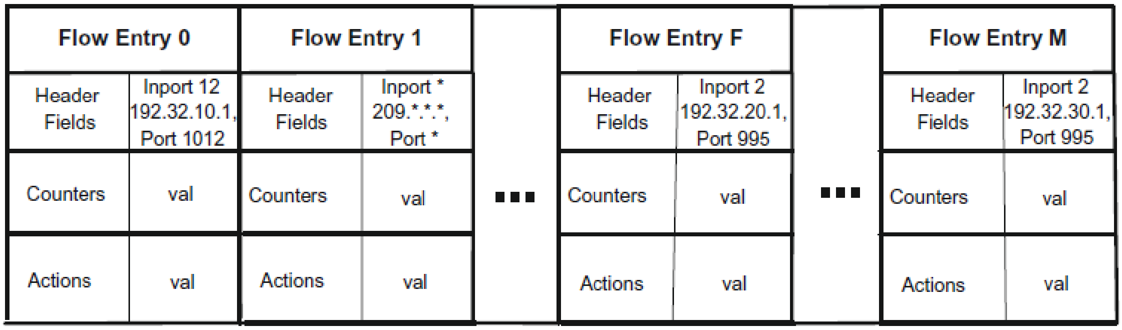
A secure channel which is a path between the controller and the devices via which the messages flow takes place is used by controller for communications [40]. The controller performs basic programming like define, add, update, modify, and delete flows, here a flow is a set of packets transferred from one network endpoint(s) to another endpoint(s). The OpenFlow Switch (data plane) [40], [41] shown in Figure 17 consists of one or more flow tables and a group table as shown in Figure 13, 14, and 16.

The switch matches headers, modifies packets, and forwards them based on a set of forwarding/flow tables and associated instructions for the particular flows at a very high speed. These tables are programmed by the controller, it sets all the packet-matching and forwarding rules in the switch [40].



Header Fields	Field value
Counters	Field value
Actions	Field value

Figure 13. Flow entries [40]



Flow Entry 0		Flow Entry 1			Flow Entry F			Flow Entry M	
Header Fields	Inport 12 192.32.10.1, Port 1012	Header Fields	Inport * 209.*.*.*, Port *		Header Fields	Inport 2 192.32.20.1, Port 995		Header Fields	Inport 2 192.32.30.1, Port 995
Counters	val	Counters	val	■■■	Counters	val	■■■	Counters	val
Actions	val	Actions	val		Actions	val		Actions	val

Figure 14. Generic flow table [40]

The flow table consists of flow entries as shown in Figure 13 and 14.

4.1 Routing/ Forwarding Process

The SDN routing/ forwarding process is described as below:

1) Using the secure channel, the messaging between the controller and switch takes place.
2) If the IP address of the controller is known then the switch will initiate this connection. The messages can be sent by either the controller or the switch without being implored by the other.
3) Once the secure channel has been set up, hello messages are exchanged between them.
4) The controller sets the configuration parameters of the switch through the SET_CONFIG [40] message during the primary phase of the controller-switch dialogue.
5) The controller supervises the switch via the OpenFlow protocol, the controller can add, update, and delete flow entries. These flows describe a set of rules that the switch should take when a packet enters the incoming port of the switch.
6) In the switch, each flow table has a set of flow entries; each flow entry consists of match/header fields, counters, and a set of instructions to apply to matching packets [40] as shown in Figure 13, 14 and 17.
    a. **Match/header fields**: used to match against packets. If the packet matches a flow entry in a flow table, the corresponding instruction set is executed like directing the packet to another flow table.
    b. **Counters**: used to track statistics about how many packets have been forwarded or dropped for this flow.

c. **Instructions/actions**: what the switch must do with the packet after the matching is performed.

7) There are three essential options for the packet which arrives at the incoming port of the switch as shown in Figure 15 and 17:

   •A. Forward the packet out to a local port possibly if header field is modified.

   • B. The packet is dropped.

   • C. Passing the packet to the controller [40].

8) If a table miss occurs where the packet does not match a flow entry in a flow table, by default the switch sends the packets to the controller via a packet-in message [40], another options is to drop the packet as shown in Figure 17.

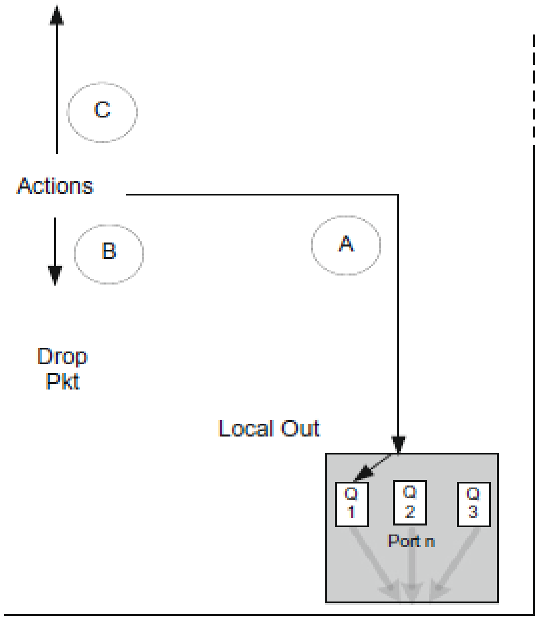


Figure 15. Essential options for the packet arriving at the switch [40]

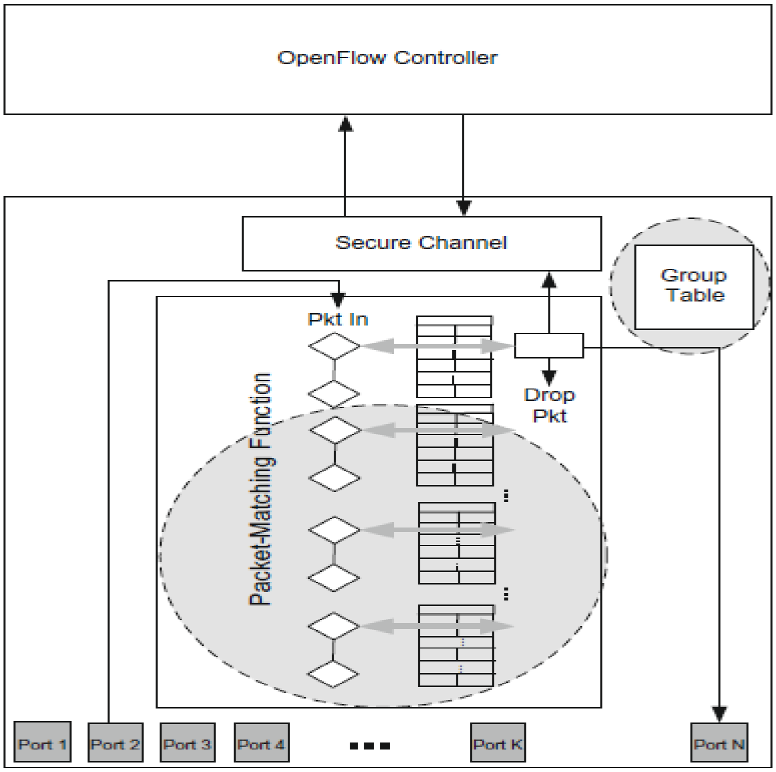


Figure 16. Group Table [40]

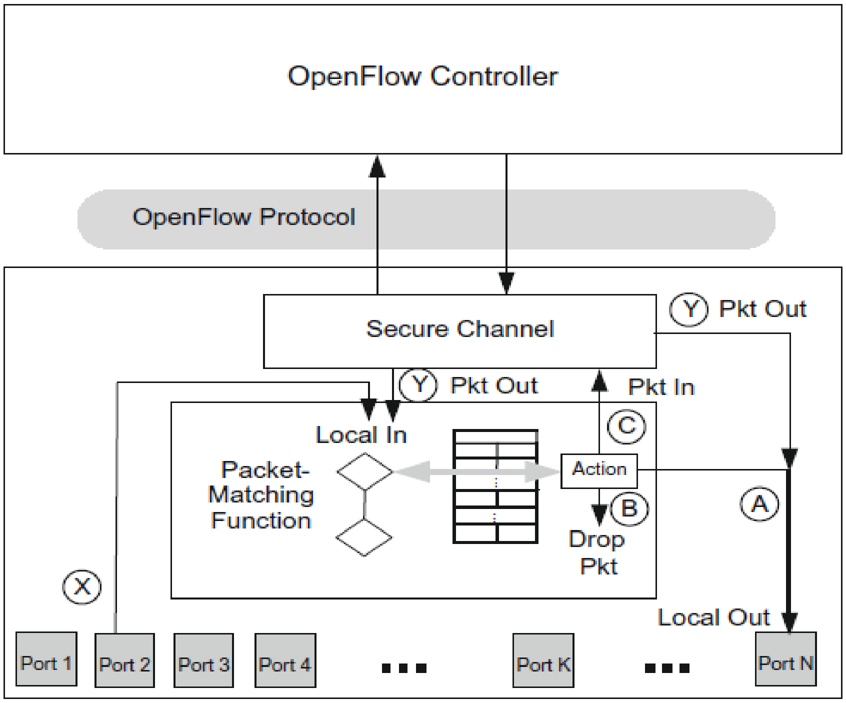


Figure 17. OpenFlow V.1.1 switch [40]

4.2 SDN Convergence Analysis

The SDN convergence process is as follows:

1) Whenever a failure occurs in a network like port down/link failure/neighbor fails, the SDN switch will first detect that a failure has occurred through PORT_STATUS which communicates changes [40], [42].

2) The switch uses the ERROR message to notify the controller about the failure by sending a message to the controller [40].

3) Once controller is notified about a failure the controller will use the knowledge of the entire network to compute new flows which do not use the failed component during transfer [42]. The controller chooses the rerouted/alternate path from the flow tables maintained by it.

4) Using the FLOW_MOD message, the controller changes the current flow entries in the switch such that all flows will avoid the failed element such that the switch that identified the failure will reroute the flows [40].

5) Using the FLOW_MOD(MODIFY) command [40] controller seeks to modify the corresponding flow entry where the MODIFY command will notify the switch to change/modify the field headers like VLAN headers, Ethernet source and destination address, and IP source and destination address [40] may be changed.

6) The controller will update its own flow table information and then pushes the same data onto the switch in the network.

7) Switches which are affected by the failure will receive the updated flow table information from the controller and will in turn update their flow tables with the information received [42].

8) Next when a packet enters the switch, the switch matches the packet’s header field with its new flow table information, if a match is found in the flow table then it forwards the packet out to the necessary local port [40].

9) If the packet does not match the flow entry then the packet is dropped or passed to the controller for further processing [40].

10) The switch also consists of a group table having group entries, each entry consisting of one or more actions buckets. These actions buckets are defined by the controller that are applied before the packet arrives at the port then the switch forwards to the correct port.

11) If there is a change in next hop of the IP routing tables in the controller, the controller can change all the flows by reprogramming the single group entry for rerouting.

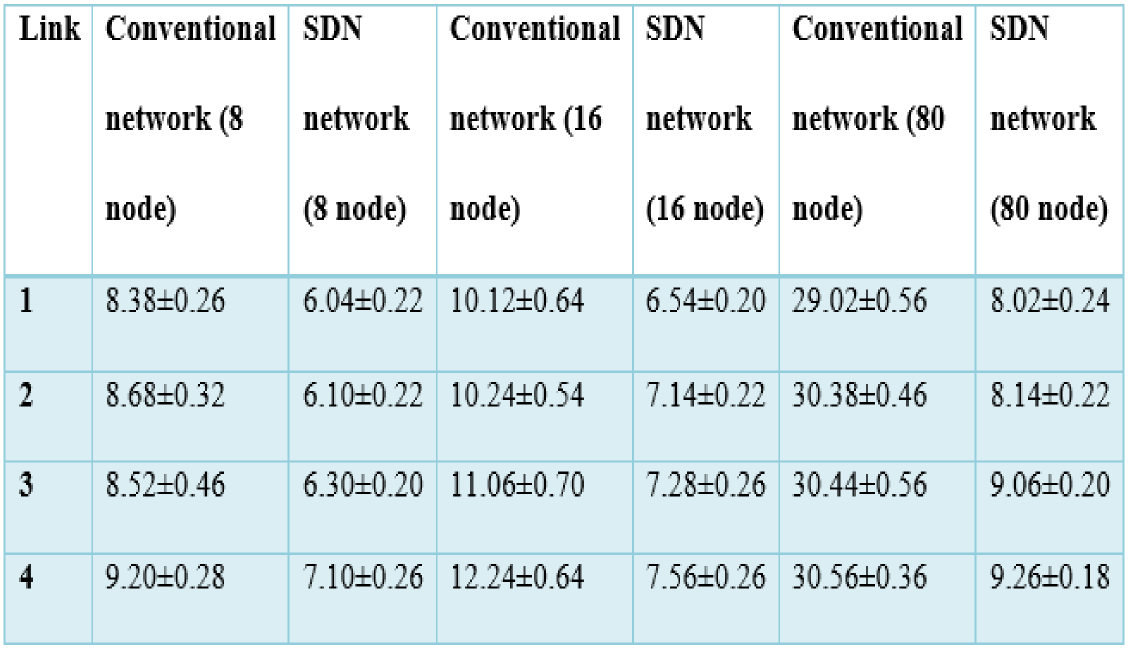
12) Flow entries may also point to a group, which specifies additional processing. These groups are used to represent set of actions for flooding even complex forwarding functions e.g. multipath, fast reroute.

13) When the next hop has been altered due to a single routing update, changing a single group entry’s action bucket is clearly faster than updating the potentially large number of flow entries. Whenever a link fails, the switch detects it and the controller traverses all the flows so that they don't pass through the failed element.

Rather than relying on decentralized and distributed negotiations between routers, in SDN the flow tables are instead computed and calculated by a centralized controller, thus routing convergence can be accelerated by replacing the decentralized routing protocols and rigid system architecture with a centralized controller scheme and arrangement.

5. Results and Graphical analysis

The table consisting of values for routing convergence calculated for all the topologies is below:

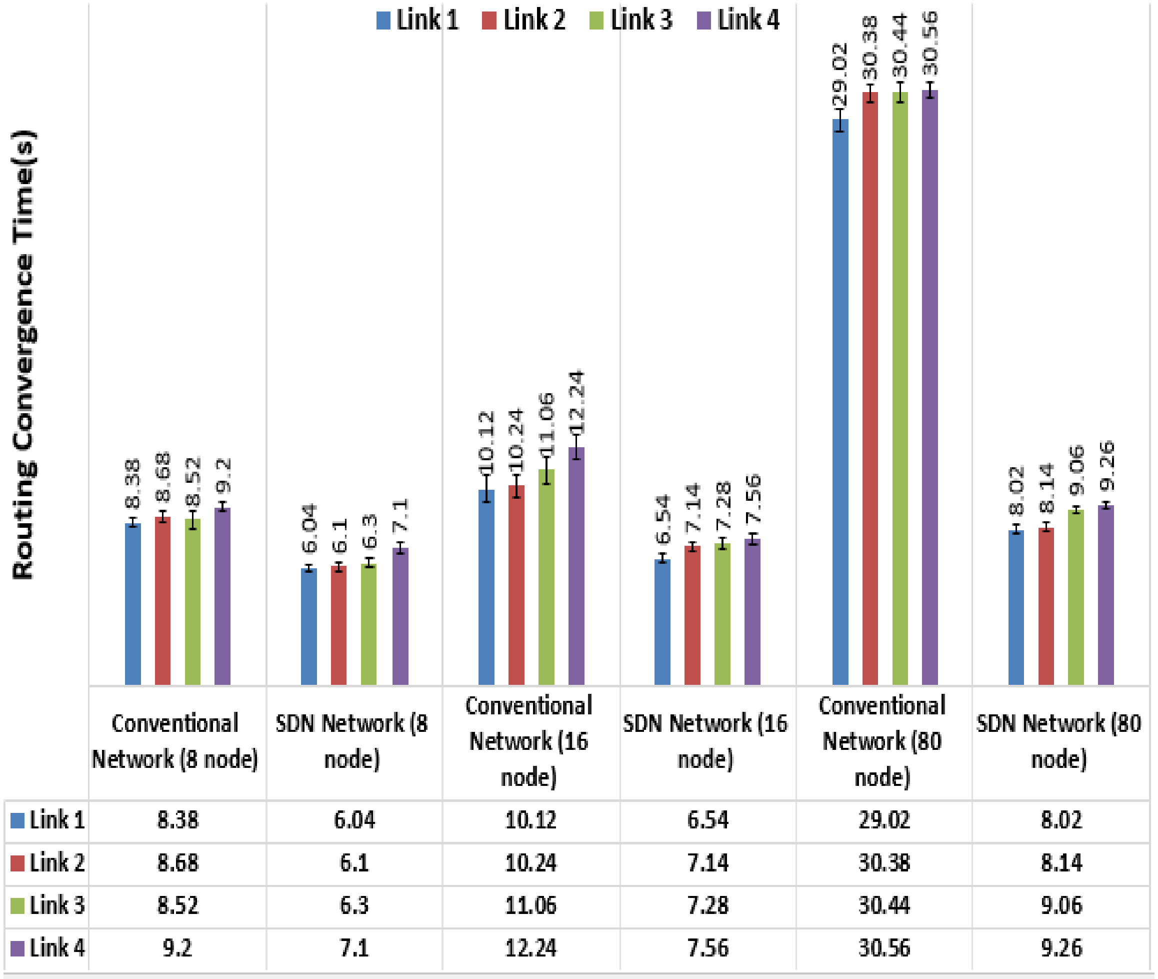


Link	Conventional network (8 node)	SDN network (8 node)	Conventional network (16 node)	SDN network (16 node)	Conventional network (80 node)	SDN network (80 node)
1	8.38±0.26	6.04±0.22	10.12±0.64	6.54±0.20	29.02±0.56	8.02±0.24
2	8.68±0.32	6.10±0.22	10.24±0.54	7.14±0.22	30.38±0.46	8.14±0.22
3	8.52±0.46	6.30±0.20	11.06±0.70	7.28±0.26	30.44±0.56	9.06±0.20
4	9.20±0.28	7.10±0.26	12.24±0.64	7.56±0.26	30.56±0.36	9.26±0.18

Table 1. Comparison of Routing Convergence Time with Error bounds

5.1 Graphical Representation

Graph depicting the Routing convergence time for all the topologies



	Conventional Network (8 node)	SDN Network (8 node)	Conventional Network (16 node)	SDN Network (16 node)	Conventional Network (80 node)	SDN Network (80 node)
🟦 Link 1	8.38	6.04	10.12	6.54	29.02	8.02
🟥 Link 2	8.68	6.1	10.24	7.14	30.38	8.14
🟩 Link 3	8.52	6.3	11.06	7.28	30.44	9.06
🟪 Link 4	9.2	7.1	12.24	7.56	30.56	9.26

Figure 18. Comparison of Routing Convergence Time with Error bounds

5.2 Result Analysis

From Figure 18 and Table 1 we see that the routing convergence time of SDN networks is lesser compared to Conventional Networks for all the three topologies. The explanation for this phenomenon lies in the convergence process of SDN and conventional networks as indicated in Section 3.2 and 4.2. From the graph in Figure 5 the following two outcomes/results can be drawn, the first result is explained below:

The first result observed is the convergence time SDN networks is lesser compared to Conventional Networks for all the three topologies, the explanation is below:

From Section 3.2, the causes for delayed routing convergence time in conventional networks occur due to the process of uploading both the RIB and FIB updates are time-consuming and the new information obtained is always been compared with the local routing table information on a consistent basis [14], [36], [37], [38]. Further before sending the new updates about the rerouted path, the withdrawal messages are sent to the neighbors withdrawing the lost routes in turn enhancing the routing convergence time [36], [37], and [38].

The process of failure detection and propagation by means of BGP mechanics is slow, in BGP if the damage is severe, the information about it is transmitted at a slow pace.

In contrast to conventional networks, from Section 4.2 the controller performs the routing convergence work in the SDN network since whenever a failure occurs in a network like port down/link failure/neighbor fails, the SDN switch will detect that a failure and communicate that change to the controller, and the controller performs the job of routing thus the routing convergence time in SDN networks depends on link down detection, topology message update time (from controller to switch and vice versa) and flow table update time, thus the routing convergence time is less compared to conventional networks.

The second result observed from the Figure 18and Table 1, the convergence time keeps increasing as the topology size increases from 8 to 80 nodes in conventional networks, this occurs because when there is a change in a network then the BGP protocol must send updates to all the routers to update their routing tables through the links. Flooding [43], [44], is similar to broadcasting, it is a way of distributing routing information updates quickly to every node in a large network [43], [44]. Thus the information of the failed link is propagated through flooding. Flooding, of course, scales linearly with topological scale since it uses every path in the network [45].Thus as the topology size/ number of routers keeps increasing, the protocol has to send updates and advertise new routes to all the routers, withdraw all the failed routes from all the routers in the network and also the routers in the network take time to update their own RIB and FIB tables in turn affecting and enhancing the routing convergence time [3].

In contrast, in SDN networks, instead of flooding the only device which is updated is the controller and the routing updates are not propagated over the entire network [4], thus the routing convergence time is less affected in this case. Once the controller finds the rerouted path it will push to the affected switches instead of transmitting to all devices. Next when a packet enters the switch, the switch matches the packet's header and forwards the packet out to the necessary local port [4], [41]. If no match is found in a flow table, the default is to forward the packet to the controller over the OpenFlow channel or to drop the packet [40].

From Table 1, after computing the error bounds, assuming that it is a Gaussian distribution with 95% confidence, in SDN networks, even though there is a trend in the routing convergence time values, these values are within the permissible acceptable error bound.

6. Conclusion and Future Work

This section of the thesis presents a precise summarizable conclusion of the comparison done in terms of routing convergence time and topology scale along with future research for this work.

6.1 Conclusion

The conclusion drawn from this work is when there is a change in a network like link/node failures the routing convergence time in conventional networks continues to increase with increase in topology scale, because the information of the failed link/node is propagated through flooding in order to update all the routing tables of the routers. Since flooding scales linearly with topology scale, it affects the routing convergence time. Thus as the topology size/ number of routers keeps increasing, the protocol has to send updates and advertise new routes to all the routers, withdraw all the failed routes from all the routers in the network and also the routers in the network take time to update their own RIB and FIB tables in turn affecting and enhancing the routing convergence time.

In case of SDN networks, with increase in topology scale there is not much significant change in routing convergence time. Since the controller performs the routing convergence work and the topology information of the network is not maintained by the switches, when there is a change in the network instead of being flooded, the information goes to the controller, which sends the updated routing tables to the affected switches and doesn't have to update the information to all the devices in the network so the routing convergence time is not much affected and moderately stable.

In conventional networks, the routing convergence time is greatly influenced by the following parameters: failure detection and propagation, BGP message update time, FIB and RIB update time, withdrawal process (of lost routes) and advertising new routes to neighbors in BGP protocol, these parameters are the causes for delayed routing convergence time in conventional networks. Moreover, failure detection and propagation by means of BGP mechanics is slow.

The routing convergence time in SDN networks depends on: link down detection, topology message update time (from controller to switch and vice versa) and flow table update time, thus the routing convergence time is less compared to conventional networks.

6.2 Future Work

The research work for the routing convergence time of the two networks can be analyzed from the aspects of other routing protocols available, varying different parameters in the network like in terms of link delay, maximum queue size, routing table scale and flow table scale with appropriate and available experimental settings and tools.

In this work, ring topology has been used, for future research different topologies like mesh, hybrid or even complex topologies can be used to measure and study the behavior of the routing convergence time taking various distinctive factors into consideration when using different topologies varying in sizes and shapes.

7. References

[1] Abe Martey and Scott Sturgess, IS-IS network design solutions. Indianapolis,Indiana: Cisco Press, 2002.

[2] The Challenge With Legacy Networks - ReadWrite, http://readwrite.com/2015/05/13/brocade-the-new-ip-challenge-sponsored/

[3] Hailong Zhang and Jinyao Yan, “Performance of SDN Routing in Comparison with Legacy Routing Protocols,” *In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on,* pp. 491–494. IEEE, 2015.

[4] Gotz Lichtwald, Uwe Walter and Martina Zitterbart, “Improving Convergence Time of Routing Protocols,” March 2004. *In Proceedings of 3 International Conference on Networking (ICN’04).*

[5] D. Sankar and D. Lancaster, “Routing Protocol Convergence Comparison using Simulation and Real Equipment,” *Advances in Communications, Computing, Networks and Security, 10(2013),* pp.186–194.

[6] Joseph Kobina Panford, Kwabena Riverson, Boansi Kufuor Oliver and Rasheeda Mendeeya Yehuza, “Comparative Analysis Of Convergence Times Between RIP And EIGRP Routing Protocols In A Network,” *Researchjournali’s Journal of Computer Science*, vol. 2, no. 3 April 2015, ISSN 2349-5391 pp.1–10.

[7] Beichuan Zhang, Daniel Massey, and Lixia Zhang, “Destination reachability and BGP convergence time [Border gateway routing protocol],” *In Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE,* vol. 3, pp. 1383–1389. IEEE, 2004.

[8] Martin Casado, Nate Foster, and Arjun Guha, “Abstractions for software-defined networks,” *Communications of the ACM 57, no. 10 (2014):* 86–95.

[9] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig, “Software defined networking: A comprehensive survey,” *Proceedings of the IEEE 103, no. 1 (2015):* 14–76.

[10] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” *IEEE Communications Surveys and Tutorials 16,* no. 3 (2014): 1617–1634.

[11] Commscope SDN and NFV facilities network convergence, https://telecomreseller.com/2017/03/13/sdn-and-nfv-facilitate-network-convergence/,www.commscope.com/Docs/SDN_NFV_Facilitate_Convergence_BA-111351-EN.

[12] Mustafa Abdulkadhim, "Routing protocols convergence activity and protocols related traffic simulation with it's impact on the network," *International Journal of Science, Engineering and Computer Technology 5, no. 3 (2015): 40.*

[13] Dan Zhao, Hongjun Liu, Xiaofeng Hu, and Chunqing Wu, "Towards network convergence and traffic engineering optimization," *In Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International,* pp. 448–455, IEEE.

[14] Timothy G. Griffin and Brian J. Premore, "An experimental analysis of BGP convergence time," *In Network Protocols, 2001. Ninth International Conference on,* pp. 53–61. IEEE.

[15] Sumit Kasera and Nishit Narang, Communication networks: principles and practice. New York City, New York: Tata McGraw-Hill Education, 2005.

[16] Sven A. Brueckner, Giovanni Di Marzo Serugendo, and David Hales, "Engineering Self-Organising Systems," *Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers. vol. 3910. Springer, 2006.*

[17] Mark Jelasity and Ozalp Babaoglu, "T-Man: Gossip-based overlay topology management," *In International Workshop on Engineering Self-Organising Applications, pp. 1–15. Springer Berlin Heidelberg, 2005.*

[18] Sahrish Khan, Abdul Wahid and Sadaf Tanvir, "Comparative study of routing strategies in software defined networking," *In Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp.* 696–702. ACM, 2016.

[19] Jonathan Wellons, Liang Dai, Yuan Xue, and Yi Cui, "Predictive or oblivious: a comparative study of routing strategies for wireless mesh networks under uncertain demand," *In Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on,* pp. 215–223. IEEE, 2008.

[20] Wolfgang Braun, and Michael Menth, "Software-defined networking using OpenFlow: Protocols, applications and architectural design choices," *2014 Future Internet 6, no. 2 (2014):* 302–336.

[21] Convergence definition by The Linux Information Project, http://www.linfo.org/convergence.html.

[22] H. Berkowitz, E. Davies, S. Hares, P. Krishnaswamy, and M. Lepp, *Terminology for benchmarking bgp device convergence in the control plane. No. RFC 4098. 2005.*

[23] John P. John, Ethan Katz-Bassett, Arvind Krishnamurthy, Thomas Anderson, and Arun Venkataramani, "Consensus routing: The Internet as a distributed system," *In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 351–364. 2008.

[24] Patricia A. Morreale and James M. Anderson, Software Defined Networking: Design and Deployment. Boca Raton, Florida: *CRC Press, 2015.*

[25] David Kim and Michael G. Solomon, "Fundamentals of information systems security," Burlington, Massachusetts: Jones and Bartlett Learning, 2016.

[26] James G Stavridis, "Convergence: illicit networks and national security in the age of globalization," Edited by Michael Miklaucic, and Jacqueline Brewer. Washington, D.C. :Government Printing Office, 2013.

[27] Amir Siddiqi and Biswajit Nandy, "Consensus routing: The Internet as a distributed system.Improving network convergence time and network stability of an OSPF-routed IP network," *In International Conference on Research in Networking, pp. 469–485. Springer Berlin Heidelberg, 2005.*

[28] Floodlight OpenFlow Controller - Project Floodlight, www.projectfloodlight.org/floodlight/.

[29] Casimer DeCusatis, Aparicio Carranza, and Jean Delgado-Caceres, "Modeling Software Defined Networks using Mininet," *Proceedings of the 2nd International Conference on Computer and Information Science and Technology (CIST'16) Ottawa, Canada*, May 11 – 12, 2016 Paper No. 133.

[30] Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet, www.mininet.org/walkthrough/

[31] What is response time? SearchNetworking, http://searchnetworking.techtarget.com/definition/response-time.

[32] How Traceroute Works? http://www.thegeekstuff.com/2012/05/traceroute-examples/

[33] Understanding Autonomous Systems, https://www.cs.rutgers.edu/~pxk/352/notes/autonomous_systems.html.

[34] BGP prefix concept, https://supportforums.cisco.com/discussion/9739831/bgp-prefix-concept.

[35] What's the meaning of BGP prefix? http://stackoverflow.com/questions/34803922/whats-the-meaning-of-bgp-prefix

[36] BGP CONVERGENCE OPTIMIZATION, https://www.ipspace.net/BGP Convergence Optimization.

[37] Petr Lapukhov, Understanding BGP Convergence. http://blog.ine.com/2010/11/22/understandingbgp-convergence/

[38] BGP Routing Table Analysis Reports, http://bgp.potaroo.net/

[39] Pavlos Sermpezis and Xenofontas Dimitropoulos, “Can SDN Accelerate BGP Convergence?A Performance Analysis of Inter-domain Routing Centralization,” *arXiv preprint arXiv:1702.00188 (2017).*

[40] Paul Goransson, Chuck Black, and Timothy Culver, Software Defined Networks: A Comprehensive Approach. Burlington, Massachusetts: Morgan Kaufmann, 2016.

[41] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, “OpenFlow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review 38*, no. 2 (2008): 69–74.

[42] Naga Katta, Haoyu Zhang, Michael Freedman, and Jennifer Rexford, “Ravana: Controller fault-tolerance in software-defined networking,” *In Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, p. 4. ACM, 2015.

[43] What is flooding? - Definition from WhatIs.com - SearchNetworking, http://searchnetworking.techtarget.com/definition/flooding.

[44] What is Flooding? - Definition from Techopedia, https://www.techopedia.com/definition/16190/flooding-networking.

[45] Joseph P.Macker and Justin W. Dean, “A study of link state flooding optimizations for scalable wireless networks,” *In Military Communications Conference, 2003. MILCOM’03.* 2003 IEEE, vol. 2, pp. 1262–1267. IEEE, 2003.

8. Appendices

8.1 Experimental Procedure

In SDN networks, first build and run Floodlight within the first VM using the following commands:

```
floodlight $ ant                               → build Floodlight
floodlight $ java -jar target/floodlight.jar → run Floodlight
```

The distance between the controller and the nodes is equivalent to 1 hop.

In Mininet (second VM), the topology is created and connected to the Floodlight using the following commands,

```
cd mininet
mininet$ cd custom
mininet/custom$ nano ring.py  → type in the source code for the 8 node topology
mininet/custom$ sudo mn --custom ring.py --topo mytopo --controller=remote, ip=156.110.167.188, port=6653 --switch ovsk, protocols=OpenFlow13
mininet> h1 ping -c h2     → ping command
mininet> link s1 s2 down → disable the link to measure the routing convergence time in Section 2.4
mininet> link s1 s2 up      → to bring the link back up
```

Likewise, in conventional networks, select the Command prompt of Host 1/PC 1, first stable communication is established using ping command between Host 1 (H1) and Host 2 (H2) in the CLI, then using the Del option the link is disabled to measure routing convergence time as indicated in Section 2.4.

8.2 Source Code

The source code for the 8 node topology with file name ‘ring.py’ is as shown below:

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
class MyTopo ( Topo ):
    “Simple topology.”
  def  __init__( self ):
        “Create custom topology.”
        # Initialize topology
        Topo.__init__( self )
        # Add hosts and switches
        H1 = self.addHost (‘H1’)
        H2 = self.addHost (‘H2’)
        S1 = self.addSwitch (‘S1’)
        S2 = self.addSwitch (‘S2’)
        S3 = self.addSwitch (‘S3’)
        S4 = self.addSwitch (‘S4’)
        S5 = self.addSwitch (‘S5’)
        S6 = self.addSwitch (‘S6’)
        S7 = self.addSwitch (‘S7’)
        S8 = self.addSwitch (‘S8’)
        # Add links
        self.addLink (H1, S1)
        self.addLink (S1, S2)
        self.addLink (S2, S3)
        self.addLink (S3, S4)
        self.addLink (S4, S8)
        self.addLink (S1, S5)
        self.addLink (S5, S6)
        self.addLink (S6, S7)
        self.addLink (S7, S8)
        self.addLink (S8, H2)
topos = { ‘mytopo’ : (lambda: MyTopo( ) ) }
```

Similarly, the above source code can be edited and connected to the Floodlight controller accordingly w.r.t the varying topology size (in this work, 16 nodes and 80 nodes topology is used, the Add links sections in the code is edited w.r.t. to the topology diagrams available in Section 2.3).